LA-UR-24-24602

Approved for public release; distribution is unlimited.

Title:	MCNP® Code Version 6.3.1 Theory & User Manual
Author(s):	Kulesza, Joel A.; Adams, Terry R.; Armstrong, Jerawan Chudoung; Bolding, Simon R.; Brown, Forrest B.; Bull, Jeffrey S.; Burke, Timothy Patrick; Clark, Alexander Rich; Forster, Robert Arthur Iii; Giron, Jesse Frank; Grieve, Tristan Sumner; Josey, Colin James; Martz, Roger Lee; McKinney, Gregg W.; Pearson, Eric J.; Rising, Michael Evan; Solomon, Clell Jeffrey Jr.; Swaminarayan, Sriram; Trahan, Travis John; Weaver, Colin Andrew; Wilson, Stephen C.; et al.
Intended for:	Report
Issued:	2024-05-28 (rev.1)

Los Alamos NATIONAL LABORATORY

LA-UR-24-24602, Rev. 1

MCNP®

Code Version 6.3.1

Theory & User Manual

Document compiled from `git` hash `2c1c635d4b` on May 16, 2024.

This document cannot be made Section-508 compliant (accessible to persons with disabilities) at this time. If you need help accessing this document, please email mcnp_help@lanl.gov.

MCNP® Code Version 6.3.1 Theory & User Manual

LA-UR-24-24602, Rev. 1

May 16, 2024

Los Alamos National Laboratory

Editor: Joel A. Kulesza

Terry R. Adams
Jerawan C. Armstrong
Simon R. Bolding
Forrest B. Brown
Jeffrey S. Bull
Timothy P. Burke
Alexander R. Clark
Robert A. (Art) Forster III
Jesse F. Giron
Avery S. Grieve
Colin J. Josey
Joel A. Kulesza
Roger L. Martz
Gregg W. McKinney
Eric J. Pearson
Michael E. Rising
Clell J. (CJ) Solomon Jr.
Sriram Swaminarayan
Travis J. Trahan
Colin A. Weaver
Stephen C. Wilson
Anthony J. Zukaitis

Table of Contents

List of Figures

List of Tables

Preface

Introduction

This document acts as a repository of knowledge for the Monte Carlo N-Particle (MCNP) transport computer code. It is maintained alongside the source code and attempts to introduce new users and re-familiarize experienced users with the theory and practices of using the MCNP code for the wide range of particle transport analyses that it is appropriate for. The latest version of the MCNP code, version 6.3.1, provides the Monte Carlo particle transport community with the latest feature developments and bug fixes in the MCNP code. The MCNP code version 6.0 and later is also known as the MCNP6 code. This document is organized in four parts:

Part I focuses on theory and is based largely on the MCNP5 theory manual [1],

Part II focuses on user guidance and input specification and is largely based on the MCNP6 user manual [2]

Part III focuses on primers and examples and is largely based on the examples in the MCNP6 user manual [Chapter 4 of 2]

Part IV contains appendices that provide details such as file formats, constants, etc.

Getting Additional Help with the MCNP Code

A website providing information on upcoming MCNP classes, a reference collection, email forum, and development team contact information is maintained at https://mcnp.lanl.gov. To seek additional help with all aspects of the MCNP code, users are encouraged to:

1. Review the reference collection to become familiar with related work,
2. Attend MCNP classes relevant to their technical focus area(s), and
3. Subscribe to, monitor, and post to the MCNP forum, as appropriate. Current instructions to subscribe to the forum are available on the MCNP website at https://mcnp.lanl.gov. Users must have a valid MCNP license to subscribe to the forum, and only subscribers may post to the forum.

Finally, users may contact the MCNP development team directly using the contact information provided.

Reporting Bugs in the MCNP Code

If users identify a bug in the MCNP code, they should:

1. Create the simplest possible input file that demonstrate the misbehavior observed,
2. Attach the input file, and any necessary supporting files (weight-window input files, unstructured mesh files, etc.) to an email describing:
   (a) The computer operating system and version,
   (b) The MCNP code version and compilation options, if known,
   (c) The parallel computing environment (MPI, OpenMP, etc.), and
   (d) Steps to reproduce the behavior.
3. Send the email and attachments to mcnp_help@lanl.gov.

Document Conventions

This document follows certain typographic conventions:

Hyperlinks	are used to improve the navigability of this document. Text in this style can be clicked in the PDF rendition of this document to take the reader elsewhere in the document or to the internet.
`CARD`	is used to indicate an input file card, which is usually hyperlinked to the card's definition elsewhere in this document.
`This Text Style`	is used to show the contents of files or commands that should be typed literally.
`This Text Style`	is used to show text that should be replaced with user-supplied values in examples and card definitions.
`This Text Style`	is used to indicate electronic construct names of such as file names, script names, subroutine names, and short library name abbreviations.
(x, y, z)	is used to indicate coordinates either in a local or global coordinate system, either abstractly as (x, y, z) or directly as, e.g., $(1, 3, 5)$.
$[i, j, k]$	is used to indicate lattice elements, either abstractly as $[i, j, k]$ or directly as, e.g., $[1, 3, 5]$.
$\mathscr{P}$	is used to indicate a particle variable that should be replaced with a specific user-specified particle type such as "n" for neutrons, "p" for photons, etc. See Table 4.3 for the full list of particle type identifiers.

Important details are often accumulated in lists such as the following. If a detail is relevant elsewhere, a hyperlink may exist to it such as (1).

Details:

1. The MCNP code cannot solve your problem, it can only answer your question.
   (a) It is important that a user understand how to ask the right question.
2. Often, the most difficult part of using the MCNP code is calculating a quantity that can be measured.

The stylistic box that follows designates a cautionary comment, which is important to its nearby text. For example...

⚠ Caution

When you see this box, you know the text within it is important to keep in mind.

The stylistic box that follows designates information that is relevant to feature deprecation. A unique identifier that relates to the MCNP issue-tracking system (e.g., DEP-12345) is used to cross-reference this box from elsewhere in the document and from within the code (e.g., in deprecation warning messages).

Because the MCNP code has undergone significant modernization work, it is important to recognize, understand, and test new features that make old features deprecated. Deprecated features should not be relied upon because they may be removed in the next release of the code. By removing these features, the MCNP development team can reduce its maintenance burden and instead focus on providing new features (and code releases) more quickly.

Deprecation Notice **DEP-12345**

When you see this box, particularly about a feature you use, a plan should be developed and executed to migrate away from the old feature/behavior to the new feature/behavior while maintaining proper quality assurance and quality control.

The MCNP input file contains entries that are commonly referred to as cards. The word "card" used throughout this manual describes a single constituent of user input that is typically a single line of a file. Cards are usually structured to take a list of whitespace-delimited numbers or keyword-value pairs. This terminology is historical and refers to a time when input was processed on punched cards. As cards are described (primarily in Chapter 5), they appear as shown below.

Cell-card Form: `CARDNAME:`$\mathscr{P}$ x
or
Data-card Form: `CARDNAME:`$\mathscr{P}$ x_1 x_2 ... x_J

$\mathscr{P}$	Particle designator.
x	Value to assign to the cell.
x_j	Value to assign to cell j. Number of entries equals number of cells in problem.

Data-card Form: `CARDNAME` *`keyword`* = `value(s)`

Description: here

Use: here

Default: here

`keyword1` = *`value`*	Description of parameter. If
	`keyword1` = `foo` do one behavior
	`keyword1` = `bar` do another behavior
`keyword2` = *`value`*	Description of parameter. If
	`keyword2` = `foo` do one behavior
	`keyword2` = `bar` do another behavior

File Listings and PDF Attachments

Source-code and MCNP file listings are given in the format shown in Listing 1.

Often, these files are electronically attached to the PDF rendition of this document such that they can be easily retrieved by the reader.

These can be accessed using Adobe Acrobat through the menu path shown in Fig. 1. The Evince software provides a similar capability to retrieve electronic attachments.

Recommended Citation

The recommended citation for this document is [3].

A recommended BibTEX entry is given in Listing 2.

Listing 1: Godiva MCNP Input Example (g1.txt)

```
g1 - Godiva critical
c
c CELL CARDS
10   100  -18.74  -1  imp:n=1
20   0             1  imp:n=0

c SURFACE CARDS
1    so   8.741

c DATA CARDS
kcode 1000  1.0  10  50
ksrc  0.0  0.0  0.0
m100  92235 -.9473
      92238 -.0527
```

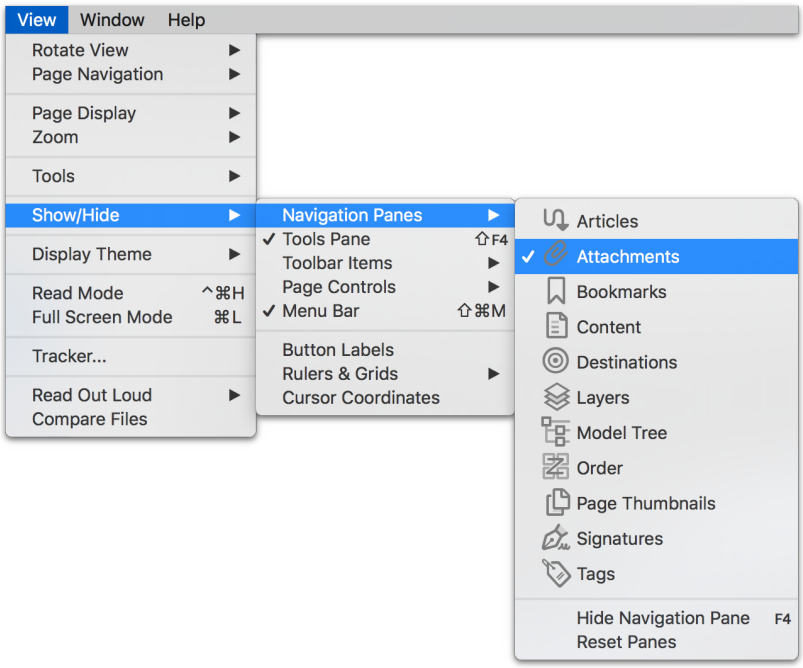


Figure 1: Adobe Acrobat Menu Path to Access PDF Attachments

Listing 2: Recommended BibTeX Entry for this Document (recommended_citation.bib.txt)

```
@TechReport{MCNP631_Manual,
  author      = {Joel A. Kulesza and Terry R. Adams and Jerawan C. Armstrong and Simon R. Bolding and
    Forrest B. Brown and Jeffrey S. Bull and Timothy P. Burke and Alexander R. Clark and Forster, III,
    Robert Arthur and Jesse F. Giron and Avery S. Grieve and Colin J. Josey and Roger L. Martz and Gregg
     W. McKinney and Eric J. Pearson and Michael E. Rising and Solomon, Jr., Clell J. and Sriram
    Swaminarayan and Travis J. Trahan and Colin A. Weaver and Stephen C. Wilson and Anthony J. Zukaitis
    },
  editor      = {Joel A. Kulesza},
  title       = {{MCNP\textsuperscript{\textregistered} Code Version 6.3.1 Theory \& User Manual}},
  institution = {Los Alamos National Laboratory},
  number      = {LA-UR-24-24602, Rev.~1},
  address     = {Los Alamos, NM, USA},
  year        = {2024},
  month       = may,
}
```

Acknowledgements

The editor and authors are grateful to the support provided by Los Alamos National Laboratory and other sponsors to maintain this resource and other resources for the worldwide MCNP user community.

These people provided review and feedback for this version of the manual:

Michael J. Berninger
Kermit A. Bunde
Yurdunaz Çelik
Osman Şahin Çelikten
Ming Fang
Jeffrey A. Favorite
Chad J. Forrest
R. Boone Gilbreath
Avery S. Grieve
Timothy M. Kelley
Jean-Marc Lasgouttes
Ronald LaVera
Michael A. Lively
Mark Miner
Donald K. (Kent) Parsons
Olaf Schumann
Mich Sakai
Ciara B. Sivels

These people provided review and feedback for prior versions of the manual:

Jeremy L. Conlin
Jeffrey A. Favorite
John S. Hendricks
Tucker C. McClanahan
Steven D. Nolen
Donald K. (Kent) Parsons
Mara M. Watson

In addition to some of the current authors, previous versions of this manual were written by:

Casey A. Anderson
Thomas E. Booth
Laura Casswell
Lawrence J. Cox
David A. Dixon
Joe W. Durkee
Jay S. Elson
Jeffrey A. Favorite
Michael L. Fensin
John T. Goorley
John S. Hendricks
Henry G. (Grady) Hughes III
Michael R. James
Russell C. Johns
Brian C. Kiedrowski
Stepan G. Mashnik
Garrett E. McMath
Denise B. Pelowitz
Richard E. Prael
Avneet Sood
Jeremy E. Sweezy
Christopher J. Werner
Trevor A. Wilcox

Abbreviations

1-D	one-dimensional
2-D	two-dimensional
3-D	three-dimensional
8-D	eight-dimensional
ACE	a compact ENDF
ACTI	Advanced Computational Technology Initiative
ACTL	Activation Library
ANS	American Nuclear Society
ANSI	American National Standards Institute
AP	anterior-posterior
ASCII	American Standard Code for Information Interchange
ASTM	American Society of Testing and Materials, now ASTM International
AWR	atomic weight ratio
CAD	computer-aided design
CAE	computer-aided engineering
CDF	cumulative distribution function
CE	continuous energy
CEM	Cascade-Exciton Model
CERN	Conseil Européen pour la Recherche Nucléaire (European Organization for Nuclear Research)
CGM	Cascading Gamma-ray and Multiplicity code
CLI	command-line interface
CLT	Central Limit Theorem
CNDC	Chinese Nuclear Data Center
CSDA	continuous slowing down approximation
CSEWG	Cross Section Evaluation Working Group

CSG	constructive solid geometry
CSR	compressed sparse row
DBRC	Doppler broadening resonance correction
DOP	delayed on production
DR	dominance ratio
EALF	energy of the average neutron lethargy causing fission
EEOUT	Elemental Edit Output File
EHSPDF	empirical history-score probability density function
ELA	Event Log Analyzer
ENDF	Evaluated Nuclear Data File
ENDL	Evaluated Nuclear Data Library
ENIAC	Electronic Numerical Integrator and Computer
EOF	end of file
EPDL	Evaluated Photon Data Library
FEA	finite element analysis
FIC	flux image on a cylinder
FIP	flux image by pinhole
FIR	flux image radiograph
FLUKA	FLUktuierende KAskade
FOM	figure of merit
FREYA	Fission Reaction Event Yield Algorithm
FWHM	full width at half maximum
GDR	giant dipole resonance
GEM	Generalized Evaporation Model
GEM2	Generalized Evaporation/Fission Model
GMV	General Mesh Viewer
GUI	graphical user interface
GWD	gigawatt days
HDF	hierarchical data format
HEU	highly enriched uranium
HPGe	high-purity germanium
HTGR	high-temperature gas reactor
IAEA	International Atomic Energy Agency

ICN	Integrated Computing Network
ICRP	International Commission on Radiological Protection
ICRS	International Conference on Radiation Shielding
ICRU	International Commission on Radiation Units and Measurements
IEEE	Institute of Electrical and Electronics Engineers
INC	Intra-Nuclear Cascade
INCL	Intra-Nuclear Cascade model developed at Liege
ISO	isotropic
ISRD	International Symposium on Reactor Dosimetry
ITS	Integrated TIGER Series
JAERI	Japan Atomic Energy Research Institute
JINR	Joint Institute for Nuclear Research
KWV	keyword-value
LANL	Los Alamos National Laboratory, formerly Los Alamos Scientific Laboratory
LANSCE	Los Alamos Neutron Science Center
LAQGSM	Los Alamos Quark-gluon String Model
LASL	Los Alamos Scientific Laboratory
LAT	lateral
LCG	linear congruential generator
LCS	LAHET Code System
LEU	low enriched uranium
LLAT	left-lateral
MCNP	Monte Carlo N-Particle
MEM	Modified Pre-equilibrium Model
MG	multigroup
MPI	Message Passing Interface
MPM	Multistage Pre-equilibrium Model
MTU	metric tons of uranium
MW	megawatt
NCIA	neutron capture ion algorithm
NCRP	National Council on Radiation Protection and Measurements
NEA	Nuclear Energy Agency
NFS	network file system

ORNL	Oak Ridge National Laboratory
OTF	on the fly
OTFDB	on-the-fly Doppler broadening
PA	posterior-anterior
PDF	probability density function; portable document format
PDS	position and direction sampling
PHT	pulse-height tally
PNG	pseudorandom number generator
QD	quasi-deuteron
QF	(Birk's) quenching factor
RAL	Rutherford Appleton Laboratory
REGL	Revised Eolus Grid Library
RLAT	right-lateral
RNG	random number generator
ROC	receiver-operator characterization
ROT	rotational
RSICC	Radiation Safety Information Computational Center
SATIF	Shielding Aspects of Accelerators, Targets, and Irradiation Facilities
SF	spontaneous fission
SSH	secure shell
TFC	tally fluctuation chart
TTB	thick-target bremsstrahlung
UM	unstructured mesh
VAA	visually accurate area
VOV	variance of the variance
VR	variance reduction
VTK	Visualization ToolKit
XCP	X-Computational Physics (a LANL organizational unit)
XDMF	eXtensible Data Model and Format
XML	Extensible Markup Language
XS	cross section
XSDIR	cross-section directory
ZAID	isotopic proton (Z) and mass number (A) (and optionally nuclear-data library) identifier

Part I

MCNP Theory Manual

Chapter 1

MCNP Code Overview

This chapter provides an overview of the MCNP code with brief summaries of the material covered in-depth in later chapters. First, §1.1 briefly describes the MCNP code and Monte Carlo particle transport method. The following five features of MCNP code are introduced in §1.2: (1) nuclear data and reactions, (2) source specifications, (3) tallies and output, (4) estimation of errors, and (5) variance reduction. Finally, §1.3 explains MCNP geometry setup and the concept of cells and surfaces.

1.1 The MCNP Code and the Monte Carlo Method

The MCNP code is a general-purpose, continuous-energy, generalized-geometry, time-dependent code designed to track 37 particle types over broad range of energies. The code was first created in 1977 when a series of special-purpose Monte Carlo codes were combined to create the first generalized Monte Carlo particle transport code. The worldwide user community's high confidence in the MCNP code's predictive capabilities are based on its performance with verification and validation test suites, comparisons to its predecessor codes, underlying high quality nuclear and atomic databases, and significant use by its users across the world in hundreds of applications. The MCNP code has become a repository for physics knowledge where the knowledge and expertise contained in the MCNP code is formidable.

The user creates an MCNP input file containing information about the problem in areas such as:

- the geometry specification,
- the description of materials and selection of cross-section evaluations,
- the location and characteristics of the source,
- the type of answers or tallies desired, and
- any variance reduction techniques used to improve efficiency.

An introduction to each area is given in Chapter 3, with more detailed discussion in the MCNP primers [Part III].

There are five guiding principles to keep in mind when developing and running a Monte Carlo particle transport calculation. They will be more meaningful as you read this manual and gain experience with the MCNP code, but no matter how sophisticated a user you may become, never forget the following five points:

1. Define and sample the geometry and source well.
2. You cannot recover lost information.

3. Question the statistical convergence, stability, and reliability of results.
4. Be conservative when applying variance reduction.
5. The number of histories run is not indicative of the quality of the answer.

The following subsections compare Monte Carlo and deterministic methods and provide a simple description of the Monte Carlo method.

1.1.1 Monte Carlo Methods vs. Deterministic Methods

Monte Carlo methods are different from deterministic transport methods. Deterministic methods, the most common of which is the discrete ordinates method, solve the transport equation for the average particle behavior. By contrast, Monte Carlo obtains answers by simulating individual particles and recording some aspects (tallies) of their average behavior. The average behavior of particles in the physical system is then inferred (using the Central Limit Theorem) from the average behavior of the simulated particles. Not only are Monte Carlo and deterministic methods very different ways of solving a problem, even what constitutes a solution is different. Deterministic methods typically give fairly complete information (for example, flux) throughout the phase space of the problem. Monte Carlo supplies information only about specific tallies requested by the user.

When Monte Carlo and discrete ordinates methods are compared, it is often said that Monte Carlo solves the integral transport equation, whereas discrete ordinates solves the integro-differential transport equation. Two things are misleading about this statement. First, the integral and integro-differential transport equations are two different forms of the same equation; if one is solved, the other is solved. Second, Monte Carlo "solves" a transport problem by simulating particle histories. A transport equation need not be written to solve a problem by Monte Carlo. Nonetheless, one can derive an equation that describes the probability density of particles in phase space; this equation turns out to be the same as the integral transport equation.

Without deriving the integral transport equation, it is instructive to investigate why the discrete ordinates method is associated with the integro-differential equation and Monte Carlo with the integral equation. The discrete ordinates method visualizes the phase space to be divided into many small regions, and the particles move from one region to another. In the limit, as the regions get progressively smaller, particles moving from region to region take a differential amount of time to move a differential distance in space. In the limit, this approaches the integro-differential transport equation, which has derivatives in space and time. By contrast, Monte Carlo transports particles between events (for example, collisions) that are separated in space and time. Neither differential space nor time are inherent parameters of Monte Carlo transport. The integral equation does not have terms involving time or space derivatives.

Monte Carlo is well suited to solving complicated three-dimensional, time-dependent problems. Because the Monte Carlo method does not use phase space regions, there are no averaging approximations required in space, energy, and time. This is especially important in allowing detailed representation of all aspects of physical data.

1.1.2 The Monte Carlo Method

Monte Carlo can be used to duplicate theoretically a random walk process (such as the interaction of nuclear particles with materials) and is particularly useful for complex problems that cannot be modeled by computer codes that use deterministic methods. The individual probabilistic events that comprise a particle history from birth to death are simulated sequentially, but particle histories can be simulated in parallel. The probability distributions governing these events are statistically sampled to describe the total phenomenon.

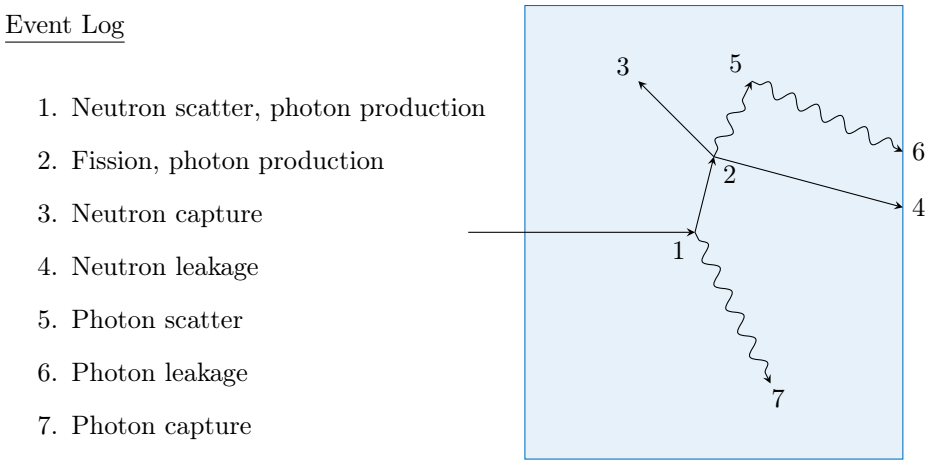


Figure 1.1: Various particle random walks. The zigzag lines are used to represent the moving of photons in the MCNP user manual, but the MCNP code treats a photon movement as a straight line between collisions.

In general, the simulation is performed on a computer because the number of trials necessary to adequately describe the phenomenon is usually quite large. The statistical sampling process is based on the selection of random numbers—analogous to throwing dice in a gambling casino—hence the name "Monte Carlo." In particle transport, the Monte Carlo technique is pre-eminently realistic (a numerical experiment). It consists of actually following each of many particles from a source throughout its life to its death in some terminal category (absorption, escape, etc.). Probability distributions are randomly sampled using nuclear data to determine the outcome at each step of its life.

The MCNP code treats neutrons and photons as particles moving in a straight line between collisions. Figure 1.1 represents the random history of a neutron incident on a slab of material that can undergo fission. Numbers between 0 and 1 are selected randomly to determine what (if any) and where interaction takes place, based on the rules (physics) and probabilities (nuclear data) governing the processes and materials involved. In this particular example, a neutron collision occurs at event 1. The neutron is scattered in the direction shown, which is selected randomly from the physical scattering distribution. A photon is also produced and is temporarily stored, or banked, for later analysis. At event 2, fission occurs, resulting in the termination of the incoming neutron and the birth of two outgoing neutrons and one photon. One neutron and the photon are banked for later analysis. The first fission neutron is captured at event 3 and terminated. The banked neutron is now retrieved and, by random sampling, leaks out of the slab at event 4. The fission-produced photon has a collision at event 5 and leaks out at event 6. The remaining photon generated at event 1 is now followed with a capture at event 7. Note that the MCNP code retrieves banked particles such that the last particle stored in the bank is taken out first (i. e., last-in-first-out stack). This neutron history is now complete. As more and more such histories are followed, the neutron and photon distributions become better known. The quantities of interest (whatever the user requests) are tallied, along with estimates of the statistical precision (uncertainty) of the results.

1.2 Introduction to Features of the MCNP Code

Various features, concepts, and capabilities of the MCNP code are summarized in this section. More detail concerning each topic is available in later chapters or appendices.

1.2.1 Nuclear Data and Reactions

The MCNP code uses continuous-energy nuclear and atomic data libraries. The primary sources of nuclear data are evaluations from the Evaluated Nuclear Data File (ENDF) [4] system, Advanced Computational Technology Initiative (ACTI) [5], the Evaluated Nuclear Data Library (ENDL) [6], Evaluated Photon Data Library (EPDL) [7], the Activation Library (ACTL) [8] compilations from Livermore, and evaluations from the Nuclear Physics (T–16) Group [9–11] at Los Alamos. Evaluated data are processed into a format appropriate for the MCNP code by codes such as NJOY [12–14]. The processed nuclear data libraries retain as much detail from the original evaluations as is feasible to faithfully reproduce the evaluator's intent. The ACE nuclear data libraries used by the MCNP code are publicly available at https://nucleardata.lanl.gov. Note that while "ACE" is an acronym for "A Compact ENDF," a better description of ACE is that it is the processed data for use in the MCNP code, as these files are often not compact.

Nuclear data tables exist for neutron interactions, neutron-induced photons, photon interactions, neutron dosimetry or activation, and thermal particle scattering $S(\alpha, \beta)$. Most of the photon and electron data are atomic rather than nuclear in nature; photonuclear data are also included. Each data table available to the MCNP code is listed on a cross-section directory file, typically referred to as the **xsdir** file. Users may select specific data tables through unique identifiers for each table described in §1.2.3. These identifiers generally contain the atomic number Z, mass number A, and library specifier ID.

Over 836 neutron interaction tables are available for approximately 100 different isotopes and elements. Multiple tables for a single isotope are provided primarily because data have been derived from different evaluations, but also because of different temperature regimes and different processing tolerances. More neutron interaction tables are constantly being added as new and revised evaluations become available. Neutron-induced photon production data are given as part of the neutron interaction tables when such data are included in the evaluations.

Photon interaction tables exist for all elements from $Z = 1$ through $Z = 100$. The data in the photon interaction tables allow the MCNP code to account for coherent and incoherent scattering, photoelectric absorption with the possibility of fluorescent emission, and pair production. Scattering angular distributions are modified by atomic form factors and incoherent scattering functions. Cross sections for nearly 2,000 dosimetry or activation reactions involving over 400 target nuclei in ground and excited states are part of the MCNP data package. These cross sections can be used as energy-dependent response functions in the MCNP code to determine reaction rates but cannot be used as transport cross sections.

Thermal data tables are appropriate for use with the $S(\alpha, \beta)$ scattering treatment in the MCNP code. The data include chemical (molecular) binding and crystalline effects that become important as the neutron's energy becomes sufficiently low. The thermal scattering library based on ENDF/B-VIII.0 contains 34 materials and 253 evaluations [15].

1.2.2 Target Identifiers

The MCNP code supports three formats for identifying targets: the ZAID format, the SZAID format, and the name format. In this section, Z is the atomic number, A is the mass number, and S is the isomeric state. For natural materials or atomic data, A can be zero.

The oldest supported format is the ZAID format, which stands for "Z-A identifier." It is a 6-digit number given as

$$\text{ZAID} = Z \times 1000 + A + (S > 0) \times 300 + S \times 100. \tag{1.1}$$

For non-metastable nuclides, this conventionally appears as ZZZAAA. The first metastable is 400 higher than the non-metastable ZZZAAA.

Table 1.1: Examples of Target Identifiers

Target	ZAID	SZAID	Name (All possibilities, first is recommended)
^{1}H	1001	1001	H-1 / H1
$^{\mathrm{Nat}}$C	6000	6000	C-0 / C0
^{99}Tc	43099	43099	Tc-99 / Tc99
$^{99\mathrm{m}}$Tc	43499	1043099	Tc-99m1/ Tc-99m / Tc99m1 / Tc99m
$^{177\mathrm{m}2}$Hf	72677	2072177	Hf-177m2 / Hf177m2
^{238}U	92238	92238	U-238 / U238
^{242}Am	95642	1095242	Am-242 / Am242
$^{242\mathrm{m}}$Am	95242	95242	Am-242m1 / Am-242m / Am242m1 / Am242m

The second supported format is the SZAID format, which stands for "*S*-*Z*-*A* identifier." It is a 7-digit number given as

$$\mathrm{SZAID} = S \times 1000000 + Z \times 1000 + A. \tag{1.2}$$

In text representation, it is commonly shown as SZZZAAA.

Caution

As a historical quirk, $^{242\mathrm{m}1}$Am and ^{242}Am are swapped in the ZAID and SZAID formats, so that the former is 95242 and the latter is 95642 for ZAID and 1095242 for SZAID. It is important to verify if a data library follows this convention. To date, all LANL-published libraries do. The name format does *not* swap these isomers. As such, `Am-242m1` can load a table labeled `95242`.

The final form is the name format. It takes the form Nn-AAAmS, where Nn is the one- or two-letter case-insensitive isotopic symbol. Isotopic symbols for all $Z \leq 118$ are available. The hyphen is optional. The metastable indicator is absent for non-metastables, can be *m* or *m1* for the first metastable state, and can be as high as *m4*. The C++ regular expression (**std::regex**, ECMAScript syntax) used is

```
"^([a-zA-Z]{1,2})-?([0-9]{1,3})(m[1-4]?)?"
```

In general, the MCNP code uses the name format whenever possible. When not possible, most typically due to breaking compatibility with input files with previous versions of the code or changing defined file formats, ZAIDs specifically will be used. Examples of all three formats can be found in Table 1.1. Input cards list which of these formats they support.

One can add analytic or otherwise non-physical data tables by using an identifier with $Z = 999$ or with an isotopic symbol of Xx. Any A will be accepted, allowing up to 999 possible data tables. The code will prevent the use of such data with model physics, which relies on valid Z and A values.

1.2.3 Table Identifiers

Table identifiers are listed in the **xsdir** file described in Appendix B and is composed of three parts. The first is the target identifier, which for single targets are described in §1.2.2. If there are multiple targets, as is typical of thermal scattering data, the target is simply a string. The second is the library identifier, which can either be a 2-digit or more integer or an arbitrary string. The third is the physics identifier [Table B.1] which defines the projectile and the governed physics.

Table 1.2: Examples of Table Identifiers

Identifier	Projectile	Physics	Target	Library Identifier
92238.80c	Neutron	Continuous Energy	^{238}U	80
h-h2o.40t	Neutron	$S(\alpha,\beta)$ Thermal Scattering	H in H_2O	40
U-238.Lib80x-293.6K.c	Neutron	Continuous Energy	^{238}U	Lib80x-293.6K
U-0.eprdata14.p	Photon	Continuous Photoatomic	Elemental U	eprdata14
1001.810h	Proton	Continuous Energy	^{1}H	810

The table identifier takes the form of

[target identifier].[library identifier][physics identifier]

if the library identifier is an integer. It takes the form of

[target identifier].[library identifier].[physics identifier]

if the library identifier is a non-integer string. Note that there is a period required between the library and physics identifiers in this form. Several examples are shown in Table 1.2 (the last three options became available in version 6.3.1).

1.2.4 Source Specification

The MCNP code's generalized user-input source capability allows the user to specify a wide variety of source conditions without having to make a code modification. Independent probability distributions may be specified for the source variables of energy, time, position, and direction, and for other parameters such as starting cell(s) or surface(s). Information about the geometric extent of the source can also be given. In addition, source variables may depend on other source variables (for example, energy as a function of angle) thus extending the built-in source capabilities of the code. The user can bias all input distributions.

In addition to input probability distributions for source variables, certain built-in functions are available. These include various analytic functions for fission and fusion energy spectra such as Watt, Maxwellian, and Gaussian spectra; Gaussian for time; and isotropic, cosine, and monodirectional for direction. Biasing may also be accomplished by special built−in functions.

A surface source allows particles crossing a surface in one problem to be used as the source for a subsequent problem. The decoupling of a calculation into several parts allows detailed design or analysis of certain geometric regions without having to rerun the entire problem from the beginning each time. The surface source has a fission volume source option that starts particles from fission sites where they were written in a previous run.

The MCNP code provides the user three methods to define an initial criticality source to estimate k_{eff}, the ratio of neutrons produced in successive generations in fissile systems.

1.2.5 Tallies and Output

The user can instruct the MCNP code to make various tallies related to particle current, particle flux, and energy deposition. MCNP tallies are normalized to be per starting particle except for a few special cases with criticality sources. Currents can be tallied as a function of direction across any set of surfaces, surface segments, or sum of surfaces in the problem. Charge can be tallied for charged particles. Fluxes across any set of surfaces, surface segments, sum of surfaces, and in cells, cell segments, or sum of cells are also available.

Similarly, the fluxes at designated detectors (points or rings) are standard tallies, as well as radiography detector tallies. Fluxes can also be tallied on a mesh superimposed on the problem geometry. Heating and fission tallies give the energy deposition in specified cells. A pulse height tally provides the energy distribution of pulses created in a detector by radiation. In addition, particles may be flagged when they cross specified surfaces or enter designated cells, and the contributions of these flagged particles to the tallies are listed separately. Tallies such as the number of fissions, the number of absorptions, the total helium production, or any product of the flux times the approximately 100 standard ENDF reactions plus several nonstandard ones may be calculated with any of the MCNP tallies. In fact, any quantity of the form

$$C = \int \phi(E) f(E) \mathrm{d}E \tag{1.3}$$

can be tallied, where $\phi(E)$ is the energy-dependent fluence, and $f(E)$ is any product or summation of the quantities in the cross-section libraries or a response function provided by the user. The tallies may also be reduced by line-of-sight attenuation. Tallies may be made for segments of cells and surfaces without having to build the desired segments into the actual problem geometry. All tallies are functions of time and energy as specified by the user and are normalized to be per starting particle. Mesh tallies are functions of energy and are also normalized to be per starting particle.

In addition to the tally information, the output file contains tables of standard summary information to give the user a better idea of how the problem ran. This information can give insight into the physics of the problem and the adequacy of the Monte Carlo simulation. If errors occur during the running of a problem, detailed diagnostic prints for debugging are given. Printed with each tally is also its statistical relative error corresponding to one standard deviation. Following the tally is a detailed analysis to aid in determining confidence in the results. Ten pass/no-pass checks are made for the user-selectable tally fluctuation chart (TFC) bin of each tally. The quality of the confidence interval still cannot be guaranteed because portions of the problem phase space possibly still have not been sampled. Tally fluctuation charts, described in the following section, are also automatically printed to show how a tally mean, error, variance of the variance, and slope of the largest history scores fluctuate as a function of the number of histories run.

All tally results, except for mesh tallies, can be displayed graphically, either while the code is running or in a separate post-processing mode.

1.2.6 Estimation of Monte Carlo Errors

MCNP tallies are normalized to be per starting particle and are printed in the output accompanied by a second number R, which is the estimated relative error defined to be one estimated standard deviation of the mean $S_{\overline{x}}$ divided by the estimated mean $\overline{x}$. In the MCNP code, the quantities required for this error estimate—the tally and its second moment—are computed after each complete Monte Carlo history, which accounts for the fact that the various contributions to a tally from the same history are correlated. For a well-behaved tally, R will be proportional to $1/\sqrt{N}$ where N is the number of histories. Thus, to halve R, we must increase the total number of histories fourfold. For a poorly behaved tally, R may increase as the number of histories increases.

The estimated relative error can be used to form confidence intervals about the estimated mean, allowing one to make a statement about what the true result is. The Central Limit Theorem states that as N approaches infinity there is a 68% chance that the true result will be in the range $\overline{x}(1 \pm R)$ and a 95% chance in the range $\overline{x}(1 \pm 2R)$. It is extremely important to note that these confidence statements refer only to the precision of the Monte Carlo calculation itself and not to the accuracy of the result compared to the true physical value. A statement regarding accuracy requires a detailed analysis of the uncertainties in the physical data, modeling, sampling techniques, and approximations, etc., used in a calculation.

The guidelines for interpreting the quality of the confidence interval for various values of R are listed in Table 1.3.

Table 1.3: Guidelines for Interpreting the Relative Error, R^*.

Range of R	Quality of the Tally
0.50 to 1.00	Not meaningful
0.20 to 0.50	Factor of a few
0.10 to 0.20	Questionable
<0.10	Generally reliable
<0.05	Generally reliable for point detectors

*$R = S_{\overline{x}}/\overline{x}$ and represents the estimated relative error at the 1σ level. These interpretations of R assume that all portions of the problem phase space are being sampled well by the Monte Carlo process.

For all tallies except next-event estimators, hereafter referred to as point detector tallies, the quantity R should be less than 0.10 to produce generally reliable confidence intervals. Point detector results tend to have larger third and fourth moments of the individual tally distributions, so a smaller value of R, < 0.05, is required to produce generally reliable confidence intervals. The estimated uncertainty in the Monte Carlo result must be presented with the tally so that all are aware of the estimated precision of the results.

Keep in mind the footnote to Table 1.3. For example, if an important but highly unlikely particle path in phase space has not been sampled in a problem, the Monte Carlo results will not have the correct expected values and the confidence interval statements may not be correct. The user can guard against this situation by setting up the problem so as not to exclude any regions of phase space and by trying to sample all regions of the problem adequately.

Despite one's best effort, an important path may not be sampled often enough, causing confidence interval statements to be incorrect. To try to inform the user about this behavior, the MCNP code calculates a figure of merit (FOM) for one tally bin of each tally as a function of the number of histories and prints the results in the tally fluctuation charts at the end of the output. The FOM is defined as

$$FOM \equiv 1/\left(R^2 T\right) \tag{1.4}$$

where T is the computer time in minutes. The more efficient a Monte Carlo calculation is, the larger the FOM will be because less computer time is required to reach a given value of R.

The FOM should be approximately constant as N increases because R^2 is proportional to $1/N$ and T is proportional to N. *Always examine the tally fluctuation charts to be sure that the tally appears well behaved, as evidenced by a fairly constant FOM.* A sharp decrease in the FOM indicates that a seldom-sampled particle path has significantly affected the tally result and relative error estimate. In this case, the confidence intervals may not be correct for the fraction of the time that statistical theory would indicate. Examine the problem to determine what path is causing the large scores and try to redefine the problem to sample that path much more frequently.

After each tally, an analysis is done and additional useful information is printed about the TFC tally bin result. The nonzero scoring efficiency, the zero and nonzero score components of the relative error, the number and magnitude of negative history scores, if any, and the effect on the result if the largest observed history score in the TFC were to occur again on the very next history are given. A table just before the TFCs summarizes the results of these checks for all tallies in the problem. Ten statistical checks are made and summarized in Table 160 after each tally, with a pass yes/no criterion. The empirical history score probability density function (PDF) for the TFC bin of each tally is calculated and displayed in printed plots.

The TFCs at the end of the problem include the variance of the variance (an estimate of the error of the relative error), and the slope (the estimated exponent of the PDF large score behavior) as a function of the number of particles started.

All this information provides the user with statistical information to aid in forming valid confidence intervals for Monte Carlo results. There is no GUARANTEE, however. The possibility always exists that some as yet unsampled portion of the problem may change the confidence interval if more histories were calculated. Chapter 2 contains more information about estimation of Monte Carlo precision.

1.2.7 Variance Reduction

As noted in the previous section, R (the estimated relative error) is proportional to $1/\sqrt{N}$, where N is the number of histories. For a given MCNP run, the computer time T consumed is proportional to N. Thus $R = C/\sqrt{T}$, where C is a positive constant. There are two ways to reduce R: (1) increase T and/or (2) decrease C. Computer budgets often limit the utility of the first approach. For example, if it has taken 2 hours to obtain $R = 0.10$, then 200 hours will be required to obtain $R = 0.01$. For this reason the MCNP code has special variance reduction techniques for decreasing C (variance is the square of the standard deviation). The constant C depends on the tally choice and/or the sampling choices.

1.2.7.1 Tally Choice

As an example of the tally choice, note that the fluence in a cell can be estimated either by a collision estimate or a track-length estimate. The collision estimate is obtained by tallying $1/\Sigma_t$ (Σ_t is the macroscopic total cross section) at each collision in the cell and the track-length estimate is obtained by tallying the distance the particle moves while inside the cell. Note that as Σ_t gets very small, very few particles collide but give enormous tallies when they do, producing a high variance situation [§2.6.6]. In contrast, the track-length estimate gets a tally from every particle that enters the cell. For this reason the MCNP code has track length tallies as standard tallies, whereas the collision tally is not standard in the MCNP code, except for estimating k_{eff}.

1.2.7.2 Non-analog Monte Carlo

Explaining how sampling affects C requires understanding of the non-analog Monte Carlo model.

The simplest Monte Carlo model for particle transport problems is the analog model that uses the natural probabilities that various events occur (for example, collision, fission, capture, etc.). Particles are followed from event to event by a computer, and the next event is always sampled (using the random number generator) from a number of possible next events according to the natural event probabilities. This is called the analog Monte Carlo model because it is directly analogous to the naturally occurring transport.

The analog Monte Carlo model works well when a significant fraction of the particles contribute to the tally estimate and can be compared to detecting a significant fraction of the particles in the physical situation. There are many cases for which the fraction of particles detected is very small, less than 10^{-6}. For these problems analog Monte Carlo fails because few, if any, of the particles tally, and the statistical uncertainty in the answer is unacceptable.

Although the analog Monte Carlo model is the simplest conceptual probability model, there are other probability models for particle transport that estimate the same average value as the analog Monte Carlo model, while often making the variance (uncertainty) of the estimate much smaller than the variance for the analog estimate. This means that problems that would be impossible to solve in days of computer time with analog methods can be solved in minutes of computer time with non-analog methods.

A non-analog Monte Carlo model attempts to follow "interesting" particles more often than "uninteresting" ones. An "interesting" particle is one that contributes a large amount to the quantity (or quantities) that

needs to be estimated. There are many non-analog techniques, and all are meant to increase the odds that a particle scores (contributes). To ensure that the average score is the same in the non-analog model as in the analog model, the score is modified to remove the effect of biasing (changing) the natural odds. Thus, if a particle is artificially made q times as likely to execute a given random walk, then the particle's score is weighted by (multiplied by) $1/q$. The average score is thus preserved because the average score is the sum, over all random walks, of the probability of a random walk multiplied by the score resulting from that random walk.

A non-analog Monte Carlo technique will have the same expected tallies as an analog technique if the expected weight executing any given random walk is preserved. For example, a particle can be split into two identical pieces and the tallies of each piece are weighted by 1/2 of what the tallies would have been without the split. Such non-analog, or variance reduction, techniques can often decrease the relative error by sampling naturally rare events with an unnaturally high frequency and weighting the tallies appropriately.

1.2.7.3 Variance Reduction Tools in the MCNP Code

There are four categories of variance reduction techniques [16] that range from the trivial to the esoteric.

1.2.7.3.1 Truncation Methods

Truncation methods are the simplest of variance reduction methods. They speed up calculations by truncating parts of phase space that do not contribute significantly to the solution. The simplest example is geometry truncation in which unimportant parts of the geometry are simply not modeled. Specific truncation methods available in the MCNP code are the energy cutoff and time cutoff.

1.2.7.3.2 Population Control Methods

Population control methods use particle splitting and Russian roulette to control the number of samples taken in various regions of phase space. In important regions many samples of low weight are tracked, while in unimportant regions few samples of high weight are tracked. A weight adjustment is made to ensure that the problem solution remains unbiased. Specific population control methods available in the MCNP code are geometry splitting and Russian roulette, energy splitting/ roulette, time splitting/roulette, weight cutoff, and weight windows.

1.2.7.3.3 Modified Sampling Methods

Modified sampling methods alter the statistical sampling of a problem to increase the number of tallies per particle. For any Monte Carlo event it is possible to sample from any arbitrary distribution rather than the physical probability as long as the particle weights are then adjusted to compensate. Thus, with modified sampling methods, sampling is done from distributions that send particles in desired directions or into other desired regions of phase space such as time or energy, or change the location or type of collisions. Modified sampling methods in the MCNP code include the exponential transform, implicit capture, forced collisions, source biasing, and neutron-induced photon production biasing.

1.2.7.3.4 Partially Deterministic Methods

Partially deterministic methods are the most complicated class of variance reduction methods. They circumvent the normal random walk process by using deterministic-like techniques, such as next-event estimators, or by controlling the random number sequence. In the MCNP code these methods include point detectors, DXTRAN, and correlated sampling.

Variance reduction techniques, used correctly, can greatly help the user produce a more efficient calculation. Used poorly, they can result in a wrong answer with good statistics and few clues that anything is amiss. Some variance reduction methods have general application and are not easily misused. Others are more specialized and attempts to use them carry high risk. The use of weight windows tends to be more powerful than the use of importances but typically requires more input data and more insight into the problem. The exponential transform for thick shields is not recommended for the inexperienced user; rather, use many cells with increasing importances (or decreasing weight windows) through the shield. Forced collisions are used to increase the frequency of random walk collisions within optically thin cells but should be used only by an experienced user. The point detector estimator should be used with caution, as should DXTRAN.

For many problems, variance reduction is not just a way to speed up the problem but is necessary to get any answer at all. Deep penetration problems and pipe detector problems, for example, will run too slowly by factors of trillions without adequate variance reduction. Consequently, users have to become skilled in using the variance reduction techniques in the MCNP code.

The following summarizes briefly the main MCNP variance reduction techniques. Detailed discussion is in §2.7.

1. Energy cutoff: Particles whose energy is out of the range of interest are terminated so that computation time is not spent following them.
2. Time cutoff: Like the energy cutoff but based on time.
3. Geometry splitting with Russian roulette: Particles transported from a region of higher importance to a region of lower importance (where they will probably contribute little to the desired problem result) undergo Russian roulette; that is, some of those particles will be killed a certain fraction of the time, but survivors will be counted more by increasing their weight the remaining fraction of the time. In this way, unimportant particles are followed less often, yet the problem solution remains undistorted. On the other hand, if a particle is transported to a region of higher importance (where it will likely contribute to the desired problem result), it may be split into two or more particles (or tracks), each with less weight and therefore counting less. In this way, important particles are followed more often, yet the solution is undistorted because, on average, total weight is conserved.
4. Energy splitting/Russian roulette: Particles can be split or rouletted upon entering various user−supplied energy ranges. Thus important energy ranges can be sampled more frequently by splitting the weight among several particles and less important energy ranges can be sampled less frequently by rouletting particles.
5. Time splitting/Russian roulette: Like energy splitting/roulette, but based on time.
6. Weight cutoff/Russian roulette: If a particle weight becomes so low that the particle becomes insignificant, it undergoes Russian roulette. Most particles are killed, and some particles survive with increased weight. The solution is unbiased because total weight is conserved, but computer time is not wasted on insignificant particles.
7. Weight window: As a function of energy, geometric location, or both, low−weighted particles are eliminated by Russian roulette and high−weighted particles are split. This technique helps keep the weight dispersion within reasonable bounds throughout the problem. An importance generator is available that estimates the optimal limits for a weight window.

8. Exponential transformation: To transport particles long distances, the distance between collisions in a preferred direction is artificially increased and the weight is correspondingly artificially decreased. Because large weight fluctuations often result, it is highly recommended that the weight window be used with the exponential transform.

9. Implicit absorption: When a particle collides, there is a probability that it is absorbed by the nucleus. In analog absorption, the particle is killed with that probability. In implicit absorption, also known as implicit capture or survival biasing, the particle is never killed by absorption; instead, its weight is reduced by the absorption probability at each collision. Important particles are permitted to survive by not being lost to absorption. On the other hand, if particles are no longer considered useful after undergoing a few collisions, analog absorption efficiently gets rid of them.

10. Forced collisions: A particle can be forced to undergo a collision each time it enters a designated cell that is almost transparent to it. The particle and its weight are appropriately split into two parts, collided and uncollided. Forced collisions are often used to generate contributions to point detectors, ring detectors, and/or DXTRAN spheres.

11. Source variable biasing: Source particles with phase-space variables of more importance are emitted with a higher frequency but with a compensating lower weight than are less important source particles.

12. Point and ring detectors: When the user wishes to tally a flux—related quantity at a point in space, the probability of transporting a particle precisely to that point is vanishingly small. Therefore, pseudoparticles are directed to the point instead. Every time a particle history is born in the source or undergoes a collision, the user may require that a pseudoparticle be tallied at a specified point in space. In this way, many pseudoparticles of low weight reach the detector, which is the point of interest, even though no particle histories could ever reach the detector. For problems with rotational symmetry, the point may be represented by a ring to enhance the efficiency of the calculation.

13. DXTRAN: DXTRAN, which stands for deterministic transport, improves sampling in the vicinity of detectors or other tallies. It involves deterministically transporting particles on collision to some arbitrary, user-defined sphere in the neighborhood of a tally and then calculating contributions to the tally from these particles. Contributions to the detectors or to the DXTRAN spheres can be controlled as a function of a geometric cell or as a function of the relative magnitude of the contribution to the detector or DXTRAN sphere. The DXTRAN method is a way of obtaining large numbers of particles on user-specified "DXTRAN spheres." DXTRAN makes it possible to obtain many particles in a small region of interest that would otherwise be difficult to sample. Upon sampling a collision or source density function, DXTRAN estimates the correct weight fraction that should scatter toward, and arrive without collision at, the surface of the sphere. The DXTRAN method then puts this correct weight on the sphere. The source or collision event is sampled in the usual manner, except that the particle is killed if it tries to enter the sphere because all particles entering the sphere have already been accounted for deterministically.

14. Correlated sampling: The sequence of random numbers in the Monte Carlo process is chosen so that statistical fluctuations in the problem solution will not mask small variations in that solution resulting from slight changes in the problem specification. The ith history will always start at the same point in the random number sequence no matter what the previous $i-1$ particles did in their random walks.

Note: weight cutoff/Russian roulette and implicit absorption are the only two variance reduction techniques enabled by default in an MCNP calculation.

1.3 MCNP Geometry

We will present here only basic introductory information about geometry setup, surface specification, and cell and surface card inputs. Areas of further interest would be the complement operator, use of parentheses, and

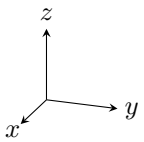


Figure 1.2: Right-handed Cartesian coordinate system.

repeated structure and lattice definitions, found in Chapter 2. Chapter 10 contains geometry examples and is recommended as a next step. Chapter 5 has detailed information about the format and entries on cell, surface (including macrobody), and data cards.

The geometry of the MCNP code treats an arbitrary three-dimensional configuration of user-defined materials in geometric cells bounded by first- and second-degree surfaces and fourth-degree elliptical tori. The cells are defined by the intersections, unions, and complements of the regions bounded by the surfaces. Surfaces are defined by supplying coefficients to the analytic surface equations or, for certain types of surfaces, known points on the surfaces. The MCNP code also provides a "macrobody" capability, where basic shapes such as spheres, boxes, cylinders, etc., may be combined using Boolean operators. This capability is essentially the same as the combinatorial geometry provided by other codes such as MORSE, KENO, and VIM.

The MCNP code has a more general geometry than is available in most combinatorial geometry codes. In addition to the capability of combining several predefined geometric bodies, as in a combinatorial geometry scheme, the MCNP code gives the user the added flexibility of defining geometric regions from all the first and second degree surfaces of analytical geometry and elliptical tori and then of combining them with Boolean operators. The code does extensive internal checking to find input errors. In addition, the geometry-plotting capability in the MCNP code helps the user check for geometry errors.

The MCNP code treats geometric cells in a Cartesian coordinate system. The surface equations recognized by the MCNP code are listed in Table 5.1. The particular Cartesian coordinate system used is arbitrary and user defined, but the right–handed system shown in Figure 1.2 is usually chosen.

Using the bounding surfaces specified on cell cards, the MCNP code tracks particles through the geometry, calculates the intersection of a track's trajectory with each bounding surface, and finds the minimum positive distance to an intersection. If the distance to the next collision is greater than this minimum distance and there are no DXTRAN spheres along the track, the particle leaves the current cell. At the appropriate surface intersection, the MCNP code finds the correct cell that the particle will enter by checking the sense of the intersection point for each surface listed for the cell. When a complete match is found, the MCNP code has found the correct cell on the other side and the transport continues.

1.3.1 Cells

When cells are defined, an important concept is that of the sense of all points in a cell with respect to a bounding surface. Suppose that $s = f(x, y, z) = 0$ is the equation of a surface in the problem. For any set of points (x, y, z), if $s = 0$ the points are on the surface. However, for points not on the surface, if $s < 0$, the points are said to have a negative sense with respect to that surface and, conversely, a positive sense if $s > 0$. For example, a point at $x = 3$ has a positive sense with respect to the plane $x–2 = 0$. That is, the equation $x–D = 3–2 = s = 1$ is positive for $x = 3$ (where D is a constant).

Cells are defined on cell cards. Each cell is described by a cell number, material number, and material density followed by a list of operators and signed surfaces that bound the cell. If the sense is positive, the sign can be omitted. The material number and material density can be replaced by a single zero to indicate a void cell. The cell number must begin in columns 1–5. The remaining entries follow, separated by blanks. A complete description of the cell card format can be found in §5.2. Each surface divides all space into two regions, one

(a) Single, complicated, cell. (b) Many, simple, cells.

Figure 1.3: Complicated versus simple cell example.

with positive sense with respect to the surface and the other with negative sense. The geometry description defines the cell to be the intersection, union, and/or complement of the listed regions.

The subdivision of the physical space into cells is not necessarily governed only by the different material regions, but may be affected by problems of sampling and variance reduction techniques (such as splitting and Russian roulette), the need to specify an unambiguous geometry, and the tally requirements. The tally segmentation feature may eliminate most of the tally requirements.

Be cautious about making any one cell very complicated. With the union operator and disjointed regions, a very large geometry can be set up with just one cell. The problem is that for each track flight between collisions in a cell, the intersection of the track with each bounding surface of the cell is calculated, a calculation that can be costly if a cell has many surfaces. As an example, consider Figure 1.3a. It is just a lot of parallel cylinders and is easy to set up. However, the cell containing all the little cylinders is bounded by twelve surfaces (counting a top and bottom). A much more efficient geometry is seen in Figure 1.3b, where the large cell has been broken up into a number of smaller cells.

1.3.1.1 Cells Defined by Intersections of Regions of Space

The intersection operator in the MCNP code is implicit; it is simply the blank space between two surface numbers on the cell card.

If a cell is specified using only intersections, all points in the cell must have the same sense with respect to a given bounding surface. This means that, for each bounding surface of a cell, all points in the cell must remain on only one side of any particular surface. Thus, there can be no concave corners in a cell specified only by intersections. Figure 1.4, a cell formed by the intersection of five surfaces (ignore surface 6 for the time being), illustrates the problem of concave corners by allowing a particle (or point) to be on two sides of a surface in one cell. Surfaces 3 and 4 form a concave corner in the cell such that points p_1 and p_2 are on the same side of surface 4 (that is, have the same sense with respect to 4) but point p_3 is on the other side of surface 4 (opposite sense). Points p_2 and p_3 have the same sense with respect to surface 3, but p_1 has the opposite sense. One way to remedy this dilemma (and there are others) is to add surface 6 between the 3/4 corner and surface 1 to divide the original cell into two cells.

With surface 6 added to Figure 1.4, the cell to the right of surface 6 is number 1 (cells indicated by circled numbers); to the left number 2; and the outside cell number 3. The cell cards (in two dimensions, all cells void) are given in Listing 1.1.

Listing 1.1: Example cell definitions.

```
1 0 1 -2 -3 6
2 0 1 -6 -4 5
```

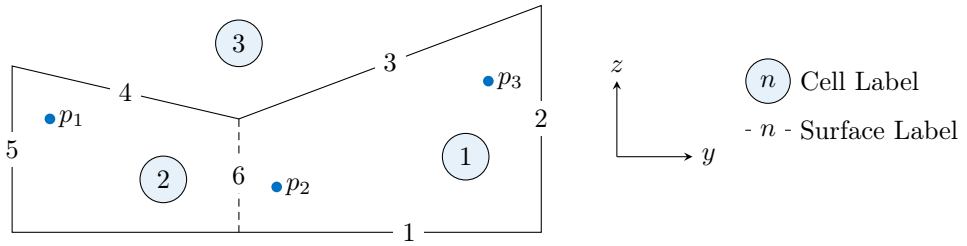
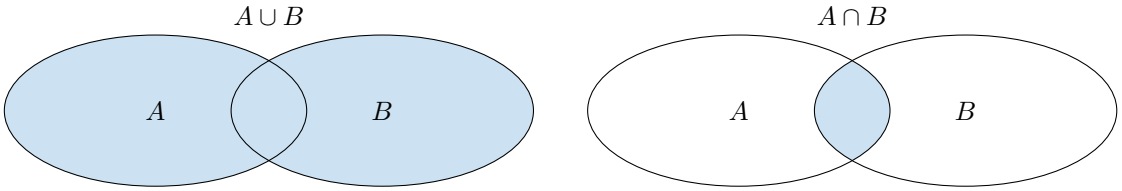


Figure 1.4: Geometry example, A.



(a) Union of all space.

(b) Intersection of common space.

Figure 1.5: Cells from unions and intersections.

Cell 1 is a void and is formed by the intersection of the region above (positive sense) surface 1 with the region to the left (negative sense) of surface 2, intersected with the region below (negative sense) surface 3, and finally intersected with the region to the right (positive sense) of surface 6. Cell 2 is described similarly.

Cell 3 cannot be specified with the intersection operator. The following section about the union operator is needed to describe cell 3.

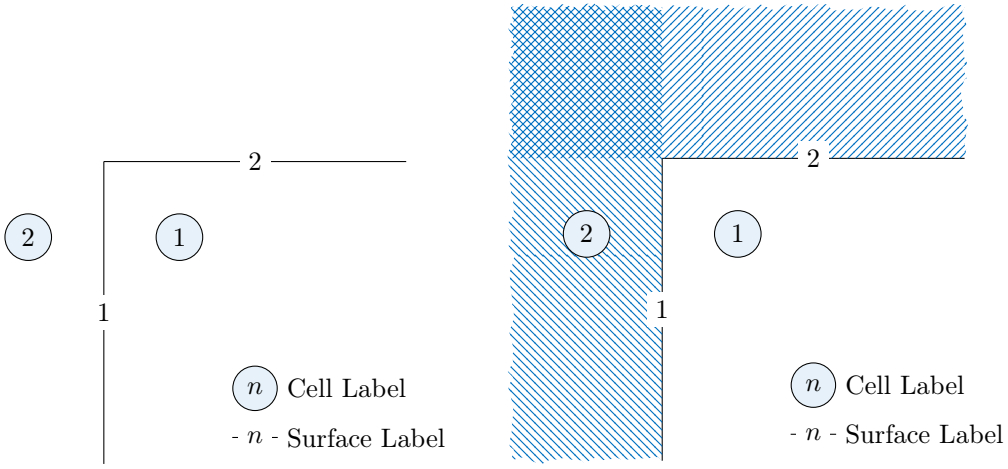
1.3.1.2 Cells Defined by Unions of Regions of Space

The union operator, signified by a colon on the cell cards, allows concave corners in cells and also cells that are completely disjoint. The intersection and union operators are binary Boolean operators, so their use follows Boolean algebra methodology; unions and intersections can be used in combination in any cell description.

Spaces on either side of the union operator are irrelevant, but remember that a space without the colon signifies an intersection. In the hierarchy of operations, intersections are performed first and then unions. There is no left to right ordering. Parentheses can be used to clarify operations and in some cases are required to force a certain order of operations. Innermost parentheses are cleared first. Spaces are optional on either side of a parenthesis. A parenthesis is equivalent to a space and signifies an intersection.

For example, let A and B be two regions of space. The region containing points that belong to both A and B is called the intersection of A and B. The region containing points that belong to A alone or to B alone or to both A and B is called the union of A and B. The shaded area in Figure 1.5a represents the union of A and B (or A : B), and the shaded area in Figure 1.5b represents the intersection of A and B (or A B). The only way regions of space can be added is with the union operator. An intersection of two spaces always results in a region no larger than either of the two spaces. Conversely, the union of two spaces always results in a region no smaller than either of the two spaces.

A simple example will further illustrate the concept of Figure 1.5 and the union operator to solidify the concept of adding and intersecting regions of space to define a cell. See also the second example in §10.1.1.2. In Figure 1.6 we have two infinite planes that meet to form two cells. Cell 1 is easy to define; it is everything in the universe to the right of surface 1 (that is, a positive sense) that is also in common with (or intersected



(a) Inclusive region.

(b) Exclusive region, showing definition complication.

Figure 1.6: More complicated cells from unions and intersections.

with) everything in the universe below surface 2 (that is, a negative sense). Therefore, the surface relation of cell 1 is 1 -2.

Cell 2 is everything in the universe to the left (negative sense) of surface 1 plus everything in the universe above (positive sense) surface 2, or -1 : 2, illustrated in Figure 1.6b by all the shaded regions of space. If cell 2 were specified as -1 2, that would represent the region of space common to -1 and 2, which is only the cross-hatched region in the figure and is obviously an improper specification for cell 2.

Returning to Figure 1.4, if cell 1 is inside the solid black line and cell 2 is the entire region outside the solid line, then the MCNP cell cards in two dimensions are (assuming both cells are voids) given in Listing 1.2.

Listing 1.2: Example cell definitions with intersections.

```
1 0 1 -2 (-3 : -4) 5
2 0 -5 : - 1 : 2 : 3 4
```

Cell 1 is defined as the region above surface 1 intersected with the region to the left of surface 2, intersected with the union of regions below surfaces 3 and 4, and finally intersected with the region to the right of surface 5. Cell 2 contains four concave corners (all except between surfaces 3 and 4), and its specification is just the converse (or complement) of cell 1. Cell 2 is the space defined by the region to the left of surface 5 plus the region below 1 plus the region to the right of 2 plus the space defined by the intersections of the regions above surfaces 3 and 4.

A simple consistency check can be noted with the two cell cards in Listing 1.2. All intersections for cell 1 become unions for cell 2 and vice versa. The senses are also reversed.

Note that in this example, all corners less than 180 degrees in a cell are handled by intersections, and all corners greater than 180 degrees are handled by unions.

To illustrate some of the concepts about parentheses, assume an intersection is thought of mathematically as multiplication and a union is thought of mathematically as addition. Parentheses are removed first, with multiplication being performed before addition. The cell cards for the example cards from Figure 1.4 may be written as shown in Listing 1.3.

Listing 1.3: Example cell definitions mathematical rendition.

```
1 a · b · (c + d) · e
2 e + a + b + c · d
```

Note that parentheses are required for the first cell but not for the second, although the second could have been written as $e + a + b + (c \cdot d)$, $(e + a + b) + (c \cdot d)$, $(e) + (a) + (b) + (c \cdot d)$, etc.

Several more examples using the union operator are given in §10.1.1. Study them to get a better understanding of this powerful operator that can greatly simplify geometry setups.

1.3.2 Surface Type Specification

The first- and second-degree surfaces plus the fourth-degree elliptical and degenerate tori of analytical geometry are all available in the MCNP code. The surfaces are designated by mnemonics such as C/Z for a cylinder parallel to the z-axis. A cylinder at an arbitrary orientation is designated by the general quadratic (GQ) mnemonic. A paraboloid parallel to a coordinate axis is designated by the special quadratic (SQ) mnemonic. The 29 mnemonics representing various types of surfaces are listed in Table 5.1.

1.3.3 Surface Parameter Specification

There are two ways to specify surface parameters in the MCNP code: (1) by supplying the appropriate coefficients needed to satisfy the surface equation, and (2) by specifying known geometric points on a surface that is rotationally symmetric about a coordinate axis.

1.3.3.1 Coefficients for the Surface Equations

The first way to define a surface is to use one of the surface-type mnemonics from Table 5.1 and to calculate the appropriate coefficients needed to satisfy the surface equation.

For example, a sphere of radius 3.62 cm with the center located at the point $(4, 1, -3)$ is specified by

```
S 4 1 -3 3.62
```

An ellipsoid whose axes are not parallel to the coordinate axes is defined by the GQ mnemonic plus up to 10 coefficients of the general quadratic equation. Calculating the coefficients can be (and frequently is) nontrivial, but the task is greatly simplified by defining an auxiliary coordinate system whose axes coincide with the axes of the ellipsoid. The ellipsoid is easily defined in terms of the auxiliary coordinate system, and the relationship between the auxiliary coordinate system and the main coordinate system is specified on a TRn card, described in §5.5.3.

The use of the SQ and GQ surfaces is determined by the orientation of the axes. One should always use the simplest possible surface in describing geometries; for example, using a GQ surface instead of an S to specify a sphere will require more computational effort for the MCNP code.

1.3.3.2 Points that Define a Surface

The second way to define a surface is to supply known points on the surface. This method is convenient if you are setting up a geometry from something like a blueprint where you know the coordinates of intersections of surfaces or points on the surfaces. When three or more surfaces intersect at a point, this second method also produces a more nearly perfect point of intersection if the common point is used in the surface specification. It is frequently difficult to get complicated surfaces to meet at one point if the surfaces are specified by the equation coefficients. Failure to achieve such a meeting can result in the unwanted loss of particles.

There are, however, restrictions that must be observed when specifying surfaces by points that do not exist when specifying surfaces by coefficients. Surfaces described by points must be either skew planes or surfaces rotationally symmetric about the x, y, or z axes. They must be unique, real, and continuous. For example, points specified on both sheets of a hyperboloid are not allowed because the surface is not continuous. However, it is valid to specify points that are all on one sheet of the hyperboloid. See the X, Y, Z, and P input card descriptions in §5.3.2 for additional explanation.

Chapter 2

Geometry, Data, Physics, and Mathematics

2.1 Introduction

This chapter discusses the mathematics and physics of the MCNP code, including geometry, cross-section libraries, sources, variance reduction schemes, Monte Carlo simulation of particle transport, and tallies. This discussion is not meant to be exhaustive; many details of the particular techniques and of the Monte Carlo method itself will be found elsewhere. Carter and Cashwell's book [17], a good general reference on radiation transport by Monte Carlo, is based upon what is in the MCNP code. Another reference is Lux and Koblinger's book [18]. Methods of sampling from standard probability densities are discussed in the Monte Carlo samplers by Everett and Cashwell [19].

The MCNP code is currently developed by Monte Carlo Codes Group (XCP-3) in X-Computational Physics Division (XCP) at Los Alamos National Laboratory (LANL). The MCNP code development team maintains and improves the MCNP code, supports and deploys it at LANL and at other Department of Energy (DOE) laboratories and government agencies where we have collaborators or sponsors, offers online and in-person MCNP training classes, and provides limited free consulting and support for MCNP users. The MCNP code is typically distributed to other users through the Radiation Safety Information Computational Center (RSICC) at Oak Ridge National Laboratory (`https://rsicc.ornl.gov`).

The MCNP code is comprised of hundreds of subroutines written in Fortran, C, and C++. The source code has been made as system independent as possible to enhance its portability, and follows the Fortran 2018 [20], C 99 [21], and C++ 14 [22] standards. The MCNP code takes advantage of parallel computer architectures using two parallel models: task-based threading using the OpenMP model and distributed processing is supported through the use of the Message Passing Interface (MPI) model. The MCNP code also combines threading with MPI, but some features of the code are only available with MPI-based parallelism.

2.1.1 History of the Monte Carlo Method and the MCNP Code

The Monte Carlo method is generally attributed to scientists working on the development of nuclear weapons in Los Alamos during the 1940s. However, its roots go back much farther.

Perhaps the earliest documented use of random sampling to solve a mathematical problem was that of Comte de Buffon in 1772 [23]. A century later people performed experiments in which they threw a needle in a haphazard manner onto a board ruled with parallel straight lines and inferred the value of π from observations of the number of intersections between needle and lines [24, 25]. Laplace suggested in 1786 that π could be evaluated by random sampling [26]. Lord Kelvin appears to have used random sampling to aid in evaluating some time integrals of the kinetic energy that appear in the kinetic theory of gasses [27] and acknowledged his secretary for performing calculations for more than 5000 collisions [28].

According to Emilio Segrè, Enrico Fermi's student and collaborator, Fermi invented a form of the Monte Carlo method when he was studying the moderation of neutrons in Rome [28, 29]. Though Fermi did not publish anything, he amazed his colleagues with his predictions of experimental results. After indulging himself, he would reveal that his "guesses" were really derived from the statistical sampling techniques that he performed in his head when he couldn't fall asleep.

During World War II at Los Alamos, Fermi joined many other eminent scientists to develop the first atomic bomb. It was here that Stan Ulam became impressed with electromechanical computers used for implosion studies. Ulam realized that statistical sampling techniques were considered impractical because they were long and tedious, but with the development of computers they could become practical. Ulam discussed his ideas with others like John von Neumann and Nicholas Metropolis. Statistical sampling techniques reminded everyone of games of chance, where randomness would statistically become resolved in predictable probabilities. It was Nicholas Metropolis who noted that Stan had an uncle who would borrow money from relatives because he "just had to go to Monte Carlo" and thus named the mathematical method "Monte Carlo" [29].

Meanwhile, a team of wartime scientists headed by John Mauchly was working to develop the first electronic computer at the University of Pennsylvania in Philadelphia. Mauchly realized that if Geiger counters in physics laboratories could count, then they could also do arithmetic and solve mathematical problems. When he saw a seemingly limitless array of women cranking out firing tables with desk calculators at the Ballistic Research Laboratory at Aberdeen, he proposed [29] that an electronic computer be built to deal with these calculations. The result was ENIAC (Electronic Numerical Integrator and Computer), the world's first computer, built for Aberdeen at the University of Pennsylvania. It had 18,000 double triode vacuum tubes in a system with 500,000 solder joints [29].

John von Neumann was a consultant to both Aberdeen and Los Alamos. When he heard about ENIAC, he convinced the authorities at Aberdeen that he could provide a more exhaustive test of the computer than mere firing-table computations. In 1945 John von Neumann, Stan Frankel, and Nicholas Metropolis visited the Moore School of Electrical Engineering at the University of Pennsylvania to explore using ENIAC for thermonuclear weapon calculations with Edward Teller at Los Alamos [29]. After the successful testing and dropping of the first atomic bombs a few months later, work began in earnest to calculate a thermonuclear weapon. On March 11, 1947, John von Neumann sent a letter to Robert Richtmyer, leader of the Theoretical Division at Los Alamos, proposing use of the statistical method to solve neutron diffusion and multiplication problems in fission devices [29]. His letter was the first formulation of a Monte Carlo computation for an electronic computing machine. In 1947, while in Los Alamos, Fermi invented a mechanical device called FERMIAC [30] to trace neutron movements through fissionable materials by the Monte Carlo Method.

By 1948 Stan Ulam was able to report to the Atomic Energy Commission that not only was the Monte Carlo method being successfully used on problems pertaining to thermonuclear as well as fission devices, but also it was being applied to cosmic ray showers and the study of partial differential equations [29]. In the late 1940s and early 1950s, there was a surge of papers describing the Monte Carlo method and how it could solve problems in radiation or particle transport and other areas [31–33]. Many of the methods described in these papers are still used in Monte Carlo today, including the method of generating random numbers [34] used in the MCNP code. Much of the interest was based on continued development of computers such as the Los Alamos MANIAC (Mechanical Analyzer, Numerical Integrator, and Computer) in March 1952.

The Atomic Energy Act of 1946 created the Atomic Energy Commission to succeed the Manhattan Project. In 1953 the United States embarked upon the "Atoms for Peace" program with the intent of developing nuclear energy for peaceful applications such as nuclear power generation. Meanwhile, computers were advancing rapidly. These factors led to greater interest in the Monte Carlo method. In 1954 the first comprehensive review of the Monte Carlo method was published by Herman Kahn [35] and the first book was published by Cashwell and Everett [19] in 1959.

At Los Alamos, Monte Carlo computer codes developed along with computers. The first Monte Carlo code was the simple 19-step computing sheet in John von Neumann's letter to Richtmyer. But as computers became

more sophisticated, so did the codes. At first the codes were written in machine language and each code would solve a specific problem. In the early 1960s, better computers and the standardization of programming languages such as Fortran made possible more general codes. The first Los Alamos general-purpose particle transport Monte Carlo code was MCS [36], written in 1963. Scientists who were not necessarily experts in computers and Monte Carlo mathematical techniques now could take advantage of the Monte Carlo method for radiation transport. They could run the MCS code to solve modest problems without having to do either the programming or the mathematical analysis themselves. MCS was followed by MCN [37] in 1965. MCN could solve the problem of neutrons interacting with matter in a three-dimensional geometry and used physics data stored in separate, highly developed, libraries.

In 1973 MCN was merged with MCG [38], a Monte Carlo gamma code that treated higher energy photons, to form MCNG, a coupled neutron-gamma code. In 1977 MCNG was merged with MCP [38], a Monte Carlo Photon code with detailed physics treatment down to 1 keV, to accurately model neutron-photon interactions. The code has been known as the MCNP code (often referred to, incorrectly, as just "MCNP") ever since. Though at first "MCNP" stood for Monte Carlo Neutron Photon, now it stands for Monte Carlo N-Particle. Other major advances in the 1970s included the present generalized tally structure, automatic calculation of volumes, and a Monte Carlo eigenvalue algorithm to determine k_{eff} for nuclear criticality (KCODE).

In 1983 MCNP3 was released, entirely rewritten in ANSI standard Fortran 77. MCNP3 was the first MCNP code version internationally distributed through the Radiation Shielding and Information Center at Oak Ridge, Tennessee. Other 1980s versions of the MCNP code were MCNP3A (1986) and MCNP3B (1988), that included tally plotting graphics (MCPLOT), the present generalized source, surface sources, repeated structures/lattice geometries, and multi-group/adjoint transport. MCNP4 was released in 1990 and was the first UNIX version of the code. It accommodated N-particle transport and multitasking on parallel computer architectures. MCNP4 added electron transport (patterned after the Integrated TIGER Series (ITS) electron physics) [39], the pulse height tally (F8), a thick-target bremsstrahlung approximation for photon transport, enabled detectors and DXTRAN with the $S(\alpha, \beta)$ thermal treatment, provided greater random number control, and allowed plotting of tally results while the code was running.

MCNP4A, released in 1993, featured enhanced statistical analysis, distributed processor multitasking for running in parallel on a cluster of scientific workstations, new photon libraries, ENDF-6 capabilities, color X-Windows graphics, dynamic memory allocation, expanded criticality output, periodic boundaries, plotting of particle tracks via SABRINA, improved tallies in repeated structures, and many smaller improvements.

MCNP4B, released in 1997, featured differential operator perturbations, enhanced photon physics equivalent to ITS3.0, PVM load balance and fault tolerance, cross-section plotting, postscript file plotting, 64-bit workstation upgrades, PC X-windows, inclusion of LAHET HMCNP, lattice universe mapping, enhanced neutron lifetimes, coincident-surface lattice capability, and many smaller features and improvements.

MCNP4C, released in 2000, featured an unresolved resonance treatment, macrobodies, superimposed importance mesh, perturbation enhancements, electron physics enhancements, plotter upgrades, cumulative tallies, parallel enhancements and other small features and improvements.

MCNP5, released in 2003, is rewritten in ANSI standard Fortran 90. It includes the addition of photonuclear collision physics, superimposed mesh tallies, time splitting, and plotter upgrades. MCNP5 also includes parallel computing enhancements with the addition of support for OpenMP and MPI.

The MCNPX program began in 1994 as an extension of MCNP4B and LAHET 2.8, extending the MCNP code to 34 particle types at nearly all energies. The INCL, CEM, and LAQGSM physics models were added along with heavy ion transport. New sources, tallies, output, graphics and variance reduction capabilities were developed and added.

The merger of MCNP5 and MCNPX began in 2006 and the first version of the merged code, MCNP6.1 (i.e., the MCNP code, version 6.1.0), was released in 2013 (which followed a release of the MCNP code, version 6 beta 2, in 2012 and was later followed by a release of the MCNP code, version 6.1.1, in 2014).

The MCNP code, version 6.2, that released in 2018, contains 39 new features in addition to 172 bug fixes and code enhancements. Two new utility tools, Whisper and MCNPTools, were released with the MCNP6.2 code. Details of MCNP6.2 features and bug fixes are in the release notes [40].

MCNP6.3, released in January 2023 and available to the public in August 2023, transitioned to the CMake build system, added numerous HDF5-formatted output files (several with complementary XDMF files to permit immediate open source and cross platform visualization), an internal fission matrix to accelerate k-eigenvalue calculations, and other new features, resources, and bug fixes described in the release notes [41]. In addition, a cross-platform Qt-based plotter built upon MCNP6.3 was released as a technology preview and to solicit user feedback.

Large production codes such as the MCNP code have revolutionized science—not only in the way it is done, but also by becoming the repositories for physics knowledge. The knowledge and expertise contained in the MCNP code is formidable. Current MCNP development is characterized by a strong emphasis on quality control, documentation, and research. New features continue to be added to the MCNP code to reflect new advances in computer architectures, improvements in Monte Carlo methodology, and better physics models. The MCNP code has a proud history and a promising future.

2.1.2 Structure of the MCNP Code

The MCNP code is currently written using a mixed Fortran/C/C++ programming paradigm. It can be built with any Fortran compiler supporting the Fortran 2018 standard [20] and any C++ compiler supporting the C++14 standard [22]. Fortran global data is shared via modules. The general internal structure of the MCNP code is as follows:

Initiation (IMCN):

- Initialize global variables to default values;
- Read input two times to get user inputs;
- Set up variable dimensions or dynamically allocated storage;
- Read input file to load input;
- Initialize random number generator;
- Process geometry;
- Process source;
- Process tallies;
- Process materials specifications including masses without loading the data files;
- Calculate cell volumes and surface areas.

Interactive Geometry Plot (PLOTG).

Cross-section Processing (XACT):

- Load libraries;
- Eliminate excess nuclear data outside problem energy range;

- Doppler broaden elastic and total cross sections to the proper temperature if the problem temperature is higher than the library temperature;
- Process multigroup libraries if requested;
- Process electron libraries including calculation of range tables, straggling tables, scattering angle distributions, and bremsstrahlung.

MCRUN sets up multitasking and multiprocessing, runs histories, and returns to print, write

RUNTPE dumps, or process another criticality cycle.

Under MCRUN, the MCNP code runs particle histories. The following procedures are for neutron and/or photon transport

- Start a source particle;
- Find the distance to the next boundary, cross the surface and enter the next cell;
- Find the total neutron cross section and process neutron collisions producing photons as appropriate;
- Find the total photon cross section and process photon collisions producing electrons as appropriate;
- Use the optional thick-target bremsstrahlung approximation if no electron transport;
- Follow electron tracks;
- Process optional multigroup collisions;
- Process detector tallies or DXTRAN;
- Process surface, cell, and pulse height tallies.

Periodically write output file, restart dumps, update to next criticality cycle, rendezvous for multitasking and updating detector and DXTRAN Russian roulette criteria, etc.:

- Go to the next criticality cycle;
- Print output file summary tables;
- Print tallies;
- Generate weight windows.

Plot tallies, cross sections, and other data (MCPLOT).

MPI distributed processor multiprocessing routines.

Random number generator and control.

Mathematics, character manipulation, and other routines.

2.1.2.1 History Flow

The history flow of heavy charged particles is described in [42]. The basic flow of a particle history for a coupled neutron/photon/electron problem is handled as follows:

For a given history, the random number sequence is set up and the number of the history is incremented. The particle-state arrays are reset. Then, the particle identifier (**ipt**) is set for the type of particle being run: 1 for a neutron, 2 for a photon, etc. (with the full set of integer identifiers given in Table 4.3). The branch of the history is set to 1.

Next, the appropriate source routine is called. Source options are the standard fixed sources, the surface source, the criticality source, or a user-provided source. All of the parameters describing the particle are set in these source routines, including position, direction of flight, energy, weight, time, and starting cell (and possibly surface), by sampling the various distributions described on the source input control cards. Several checks are made at this time to verify that the particle is in the correct cell or on the correct surface, and directed toward the correct cell.

Next, the initial parameters of the first fifty particle histories are printed. Then some of the summary information is incremented. Energy, time, and weight are checked against cutoffs. A number of error checks are made. Detector contributions are scored, and then the DXTRAN subroutine is called (if used in the problem) to create particles on the spheres. The particles are saved in the bank for later tracking. Bookkeeping is started for the pulse height cell tally energy balance. The weight window game is played, with any additional particles from splitting put into the bank and any losses to Russian roulette terminated.

Then the actual particle transport is started. For an electron source, electrons are run separately. For a neutron or photon source, the intersection of the particle trajectory with each bounding surface of the cell is calculated. The minimum positive distance to the cell boundary indicates the next surface the particle is heading toward. The distance to the nearest DXTRAN sphere is calculated, as is the distance to time cutoff, and energy boundary for multigroup charged particles. The cross sections for a current cell are calculated using a binary table lookup in data tables for neutrons or photons. The total photon cross section may include the photonuclear portion of the cross section if photonuclear physics is in use. See §5.7.2.3 for a discussion of turning photonuclear physics on. The total cross section is modified by the exponential transformation if necessary. The distance to the next collision is determined (if a forced collision is required, the uncollided part is banked). The track length of the particle in the cell is found as the minimum of the distance to collision, the distance to the cell surface, one mean free path (in the case of a mesh-based weight window), the distance to a DXTRAN sphere, the distance to time cutoff, or the distance to energy boundary. Track length cell tallies are then incremented. Some summary information is incremented. The particle's parameters (time, position, and energy) are then updated. If the particle's distance to a DXTRAN sphere (of the same type as the current particle) is equal to the minimum track length, the particle is terminated because particles reaching the DXTRAN sphere are already accounted for by the DXTRAN particles from each collision. If the particle exceeds the time cutoff, the track is terminated. If the particle was detected leaving a DXTRAN sphere, the DXTRAN flag is set to zero and the weight cutoff game is played. The particle is either terminated to weight cutoff or survives with an increased weight. Weight adjustments then are made for the exponential transformation.

If the minimum track length is equal to the distance-to-surface crossing, the particle is transported to the cell surface, any surface tallies are processed, and the particle is processed for entering the next cell. Reflecting surfaces, periodic boundaries, geometry splitting, Russian roulette from importance sampling, and loss to escape are treated. The bank entries or retrievals are made on a last-in, first-out basis. The history is continued by going back to the previous paragraph and repeating the steps.

If the distance to collision is less than the distance to surface, or if a multigroup charged particle reaches the distance to energy boundary, the particle undergoes a collision. For neutrons, the collision analysis determines which nuclide is involved in the collision, samples the target velocity of the collision nuclide for

the free gas thermal treatment, generates and banks any photons, handles analog capture or capture by weight reduction, plays the weight cutoff game, and handles $S(\alpha, \beta)$ thermal collisions and elastic or inelastic scattering. For criticality problems, fission sites are stored for subsequent generations. Any additional tracks generated in the collision are put in the bank. The energies and directions of particles exiting the collision are determined. Multigroup and multigroup/adjoint collisions are treated separately. The collision process and thermal treatments are described in more detail in §2.4.3.1.

The collision analysis for photons is similar to that for neutrons, but includes either the simple or the detailed physics treatments. See §5.7.2.3 for a discussion of turning photonuclear physics on. The simple physics treatment is valid only for photon interactions with free electrons, i.e. it does not account for electron binding effects when sampling emission distributions; the detailed treatment is the default and includes form factors and Compton profiles for electron binding effects, coherent (Thomson) scatter, and fluorescence from photoelectric capture [§2.4.4]. There may also be photonuclear physics (if photonuclear physics is in use). Additionally, photonuclear biasing is available (similar to forced collisions) to split the photon (updating the weight by the interaction probabilities) and force one part to undergo a photoatomic collision and the second part to undergo a photonuclear collision. The collision analysis samples for the collision nuclide, treats photonuclear collisions, treats photoelectric absorption, or capture (with fluorescence in the detailed physics treatment), incoherent (Compton) scatter (with Compton profiles and incoherent scattering factors in the detailed physics treatment to account for electron binding), coherent (Thomson) scatter for the detailed physics treatment only (again with form factors), and pair production. Secondary particles from photonuclear collisions (either photons or neutrons) are sampled using the same routines as for inelastic neutron collisions [§2.4.3.5]. Electrons are generated for incoherent scatter, pair production, and photoelectric absorption. These electrons may be assumed to deposit all their energy instantly if *ides* = 1 on the PHYS:p card, or they may produce electrons with the thick-target bremsstrahlung approximation (default for MODE p problems, *ides* = 0 on the PHYS:p card), or they may undergo full electron transport (default for MODE p e problems, *ides* = 0 on the PHYS:p card.) Multigroup or multigroup/adjoint photons are treated separately.

After the surface crossing or collision is processed, transport continues by calculating the distance to cell boundary, and so on. Or if the particle involved in the collision was killed by capture or variance reduction, the bank is checked for any remaining progeny, and if none exists, the history is terminated. Appropriate summary information is incremented, the tallies of this particular history are added to the total tally data, the history is terminated, and a return is made.

After each history, several checks are made to see if other actions need to be performed before additional histories can be run. For continuation, the subroutine is called again. Otherwise a return is made and the summary information and tally data are printed.

2.2 Geometry

The basic MCNP geometry concepts, discussed in Chapter 1, include the sense of a cell, the intersection and union operators, and surface specification. Covered in this section are the complement operator; the repeated structure capability; an explanation of two surfaces, the cone and the torus; and a description of ambiguity, reflecting, white, and periodic boundary surfaces.

2.2.1 Complement Operator

The complement operator provides no new capability over the intersection and union operators. It is just a shorthand cell-specifying method that implicitly uses the intersection and union operators.

The complement operator is the # symbol. The complement operator can be thought of as standing for not in. There are two basic uses of the operator:

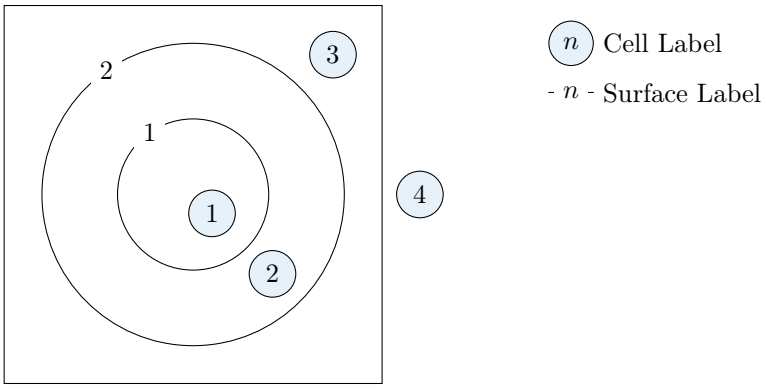


Figure 2.1: Illustration of poor use of complement operator.

1. $\#n$ means that the description of the current cell is the complement of the description of cell n.
2. $\#(\dots)$ means complement the portion of the cell description in the parentheses (usually just a list of surfaces describing another cell).

In the first of the two above forms, the MCNP code performs five operations: (1) the symbol # is removed, (2) parentheses are placed around n, (3) any intersections in n become unions, (4) any unions in n are replaced by back-to-back parentheses, ")(", which is an intersection, and (5) the senses of the surfaces defining n are reversed.

A simple example is a cube. We define a two-cell geometry with six surfaces, where cell 1 is the cube and cell 2 is the outside world:

```
1 0 -1 2 -3 4 -5 6
2 0 1:-2: 3:-4: 5:-6
```

Note that cell 2 is everything in the universe that is **not in** cell 1, or

```
2 0 #1
```

The form `#(n)` is not allowed; it is functionally available as the equivalent of `-n`.

⚠ Caution

Using the complement operator can destroy some of the necessary conditions for some cell volume and surface area calculations by the MCNP code. See §10.1.1.14 for an example.

The complement operator can be easily abused if it is used indiscriminately. A simple example can best illustrate the problems. Figure 2.1 consists of two concentric spheres inside a box. Cell 4 can be described using the complement operator as

```
4 0 #3 #2 #1
```

Although cells 1 and 2 do not touch cell 4, to omit them would be incorrect. If they were omitted, the description of cell 4 would be everything in the universe that is not in cell 3. Since cells 1 and 2 are not part of cell 3, they would be included in cell 4. Even though surfaces 1 and 2 do not physically bound cell 4, using the complement operator as in this example causes the MCNP code to think that all surfaces involved with the complement do bound the cell. Even though this specification is correct and required by the MCNP code, the disadvantage is that when a particle enters cell 4 or has a collision in cell 4, the MCNP code must calculate the intersection of the particle's trajectory with all real bounding surfaces of cell 4 plus any extraneous ones brought in by the complement operator. This intersection calculation is very expensive and can add significantly to the required computer time.

A better description of cell 4 would be to complement the description of cell 3 (omitting surface 2) by reversing the senses and interchanging union and intersection operators as illustrated in the cell cards that describe the simple cube in the preceding paragraphs.

2.2.2 Repeated Structure Geometry

The repeated structure geometry feature is explained in detail starting on §5.5.5. The capabilities are only introduced here. Examples are shown in Chapter 10. The cards associated with the repeated structure feature are `U` (universe), `FILL`, `TRCL`, `URAN`, and `LAT` (lattice) and cell cards with LIKE m BUT.

The repeated structure feature makes it possible to describe only once the cells and surfaces of any structure that appears more than once in a geometry. This unit then can be replicated at other locations by using the "LIKE m BUT" construct on a cell card. The user specifies that a cell is filled with something called a universe. The `U` card identifies the universe, if any, to which a cell belongs. The `FILL` card specifies with which universe a cell is to be filled. A universe is either a lattice or an arbitrary collection of cells. The two types of lattice shapes, hexagonal prisms and hexahedra, need not be rectangular nor regular, but they must fill space exactly. Several concepts and cards combine in order to use this capability.

2.2.3 Surfaces

2.2.3.1 Explanation of Cone and Torus

Two surfaces, the cone and torus, require more explanation. The quadratic equation for a cone describes a cone of two sheets (just like a hyperboloid of two sheets): one sheet is a cone of positive slope, and the other has a negative slope. A cell whose description contains a two-sheeted cone may require an ambiguity surface to distinguish between the two sheets. The MCNP code provides the option to select either of the two sheets; this option frequently simplifies geometry setups and eliminates any ambiguity. The $+1$ or the -1 entry on the cone surface card causes the one sheet cone treatment to be used. If the sign of the entry is positive, the specified sheet is the one that extends to infinity in the positive direction of the coordinate axis to which the cone axis is parallel. The converse is true for a negative entry. This feature is available only for cones whose axes are parallel to the coordinate axes of the problem.

The treatment of fourth degree surfaces in Monte Carlo calculations has always been difficult because of the resulting fourth order polynomial ("quartic") equations. These equations must be solved to find the intersection of a particle's line of flight with a toroidal surface. In the MCNP code these equations must also be solved to find the intersection of surfaces in order to compute the volumes and surface areas of geometric regions of a given problem. In either case, the quartic equation,

$$x + Bx + Cx + Dx + E = 0, \tag{2.1}$$

is difficult to solve on a computer because of roundoff errors. For many years the MCNP toroidal treatment required 30 decimal digits (CDC double-precision) accuracy to solve quartic equations. Even then there were

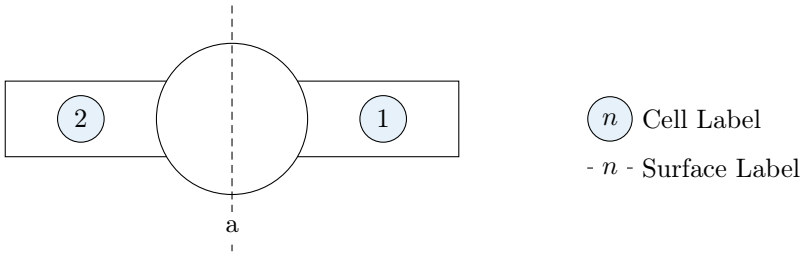


Figure 2.2: Cell demonstrating two different senses.

round-off errors that had to be corrected by Newton-Raphson iterations. Schemes using a single-precision quartic formula solver followed by a Newton-Raphson iteration were inadequate because if the initial guess of roots supplied to the Newton-Raphson iteration is too inaccurate, the iteration will often diverge when the roots are close together.

The single-precision quartic algorithm in the MCNP code basically follows the quartic solution of Cashwell and Everett [43]. When roots of the quartic equation are well separated, a modified Newton-Raphson iteration quickly achieves convergence. But the key to this method is that if the roots are double roots or very close together, they are simply thrown out because a double root corresponds to a particle's trajectory being tangent to a toroidal surface, and it is a very good approximation to assume that the particle then has no contact with the toroidal surface. In extraordinarily rare cases where this is not a good assumption, the particle would become "lost." Additional refinements to the quartic solver include a carefully selected finite size of zero, the use of a cubic rather than a quartic equation solver whenever a particle is transported from the surface of a torus, and a gross quartic coefficient check to ascertain the existence of any real positive roots. As a result, the single-precision quartic solver is substantially faster than double-precision schemes, portable, and also somewhat more accurate.

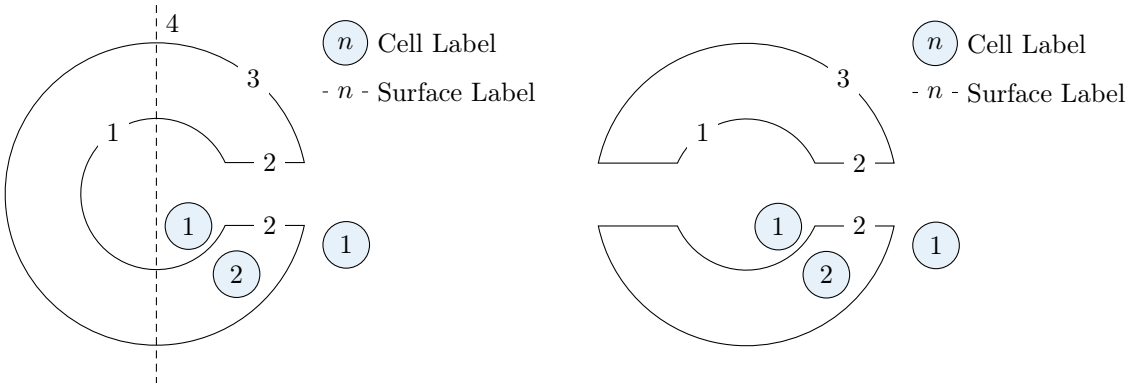
In the MCNP code, elliptical tori symmetric about any axis parallel to a coordinate axis may be specified. The volume and surface area of various tallying segments of a torus usually will be calculated automatically.

2.2.3.2 Ambiguity Surfaces

The description of the geometry of a cell must eliminate any ambiguities as to which region of space is included in the cell. That is, a particle entering a cell should be able to determine uniquely which cell it is in from the senses of the bounding surfaces. This is not possible in a geometry such as shown in Figure 2.2 unless an ambiguity surface is specified. Suppose the figure is rotationally symmetric about the y-axis.

A particle entering cell 2 from the inner spherical region might think it was entering cell 1 because a test of the senses of its coordinates would satisfy the description of cell 1 as well as that of cell 2. In such cases, an ambiguity surface is introduced such as plane a. An ambiguity surface need not be a bounding surface of a cell, but it may be and frequently is. It can also be the bounding surface of some cell other than the one in question. However, the surface must be listed among those in the problem and must not be a reflecting surface [§2.2.3.3]. The description of cells 1 and 2 in Figure 2.2 is augmented by listing for each its sense relative to surface a as well as that of each of its other bounding surfaces. A particle in cell 1 cannot have the same sense relative to surface a as does a particle in cell 2. More than one ambiguity surface may be required to define a particular cell.

A second example may help to clarify the significance of ambiguity surfaces. We would like to describe the geometry of Figure 2.3a. Without the use of an ambiguity surface, the result will be Figure 2.3b. Surfaces 1 and 3 are spheres about the origin, and surface 2 is a cylinder around the y-axis. Cell 1 is both the center and outside world of the geometry connected by the region interior to surface 2.



(a) With ambiguity surface.

(b) Without ambiguity surface.

Figure 2.3: Example geometry demonstrating ambiguity surface.

At first glance it may appear that cell 1 can easily be specified by -1 : -2 : 3 whereas cell 2 is simply #1. This results in Figure 2.3b, in which cell 1 is everything in the universe interior to surface 1 plus everything in the universe interior to surface 2 (remember the cylinder goes to plus and minus infinity) plus everything in the universe exterior to surface 3.

An ambiguity surface (plane 4 at $y = 0$) will solve the problem. Everything in the universe to the right of the ambiguity surface intersected with everything in the universe interior to the cylinder is a cylindrical region that goes to plus infinity but terminates at $y = 0$. Therefore, -1 : (4 -2) : 3 defines cell 1 as desired in Figure 2.3a. The parentheses in this last expression are not required because intersections are done before unions. Another expression for cell 2 rather than #1 is 1 -3 #(4 -2).

For the user, ambiguity surfaces are specified the same way as any other surface—simply list the signed surface number as an entry on the cell card. For the MCNP code, if a particular ambiguity surface appears on cell cards with only one sense, it is treated as a true ambiguity surface. Otherwise, it still functions as an ambiguity surface but the TRACK subroutine will try to find intersections with it, thereby using a little more computer time.

2.2.3.3 Reflecting Surfaces

A surface can be designated a reflecting surface by preceding its number on the surface card with an asterisk. Any particle hitting a reflecting surface is specularly (mirror) reflected. Reflecting planes are valuable because they can simplify a geometry setup (and also tracking) in a problem. They can, however, make it difficult (or even impossible) to get the correct answer. The user is cautioned to check the source weight and tallies to ensure that the desired result is achieved. Any tally in a problem with reflecting planes should have the same expected result as the tally in the same problem without reflecting planes.

⚠ Caution

Point detectors or DXTRAN regions used with reflecting surfaces give wrong answers [§2.5.6.4.2].

The following example illustrates the above points and should make MCNP users very cautious in the use of reflecting surfaces. Reflecting surfaces should never be used in any situation without a lot of thought.

Consider a cube of carbon 10 cm on a side sitting on top of a 5-MeV neutron source distributed uniformly in volume. The source cell is a 1-cm-thick void completely covering the bottom of the carbon cube and no

more. The average neutron flux across any one of the sides (but not top or bottom) is calculated to be 0.150 (±0.5%) per cm^2 per starting neutron from an MCNP F2 tally, and the flux at a point at the center of the same side is 1.55×10^{-03} n/cm^2 (±1%) from an MCNP F5 tally. The cube can be modeled by half a cube and a reflecting surface. All dimensions remain the same except the distance from the tally surface to the opposite surface (which becomes the reflecting surface) is 5 cm. The source cell is cut in half also. Without any source normalization, the flux across the surface is now 0.302 (±0.5%), which is twice the flux in the nonreflecting geometry. The detector flux is 2.58×10^{-03} (±1%), which is less than twice the point detector flux in the nonreflecting problem.

The problem is that for the surface tally to be correct, the starting weight of the source particles has to be normalized; it should be half the weight of the non-reflected source particles. The detector results will always be wrong (and lower) for the reason discussed in §2.5.6.4.2.

In this particular example, the normalization factor for the starting weight of source particles should be 0.5 because the source volume is half of the original volume. Without the normalization, the full weight of source particles is started in only half the volume. These normalization factors are problem dependent and should be derived very carefully.

Another way to view this problem is that the tally surface has doubled because of the reflecting surface; two scores are being made across the tally surface when one is made across each of two opposite surfaces in the nonreflecting problem. The detector has doubled too, except that the contributions to it from beyond the reflecting surface are not being made [§2.5.6.4.2].

2.2.3.4 White Boundaries

A surface can be designated a white boundary surface by preceding its number on the surface card with a plus. A particle hitting a white boundary is reflected with a cosine distribution, $p(\mu) = \mu$, relative to the surface normal; that is, $\mu = \sqrt{\xi}$, where ξ is a random number. White boundary surfaces are useful for comparing MCNP results with other codes that have white boundary conditions. They also can be used to approximate a boundary with an infinite scatterer. They make no sense in problems with next-event estimators such as detectors or DXTRAN [§2.5.6.4.2] and should always be used with caution.

2.2.3.5 Periodic Boundaries

Periodic boundary conditions can be applied to pairs of planes to simulate an infinite lattice. Although the same effect can be achieved with an infinite lattice, the periodic boundary is easier to use, simplifies comparison with other codes having periodic boundaries, and can save considerable computation time. There is approximately a 55% run-time penalty associated with repeated structures and lattices that can be avoided with periodic boundaries. However, collisions and other aspects of the Monte Carlo random walk usually dominate running time, so the savings realized by using periodic boundaries are usually much smaller. A simple periodic boundary problem is illustrated in Figure 2.4.

It consists of a square reactor lattice infinite in the z direction and 4 cm on a side in the x and y directions with an off-center 0.5 cm radius cylindrical fuel pin. The MCNP surface cards are given in Listing 2.1.

Listing 2.1: periodic_boundary.mcnp.inp.txt

```
1 -2 px  -2
2 -1 px   2
3 -4 py  -2
4 -3 py   2
5    c/z  0.75 0.75 0.5
```

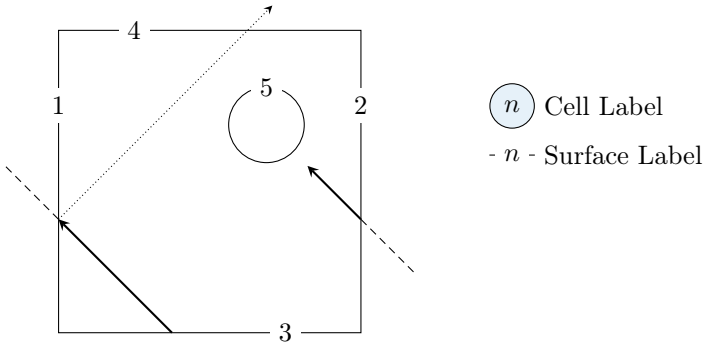


Figure 2.4: Demonstration of periodic boundary conditions.

The negative entries before the surface mnemonics specify periodic boundaries. Card one says that surface 1 is periodic with surface 2 and is a `px` plane. Card two says that surface 2 is periodic with surface 1 and is a `px` plane. Card three says that surface 3 is periodic with surface 4 and is a `py` plane. Card four says that surface 4 is periodic with surface 3 and is a `py` plane. Card five says that surface 5 is an infinite cylinder parallel to the z axis. A particle leaving the lattice out the left side (surface 1) reenters on the right side (surface 2). If the surfaces were reflecting, the reentering particle would miss the cylinder, shown by the dotted line. In a fully specified lattice and in the periodic geometry, the re-entering particle will hit the cylinder as it should.

Much more complicated examples are possible, particularly hexagonal prism lattices. In all cases, the MCNP code checks that the periodic surface pair matches properly and performs all the necessary surface rotations and translations to put the particle in the proper place on the corresponding periodic plane.

The following limitations apply:

- Periodic boundaries cannot be used with next-event estimators such as detectors or DXTRAN [§2.5.6.4.2];
- All periodic surfaces must be planes;
- Periodic planes cannot also have a surface transformation;
- The periodic cells may be infinite or bounded by planes on the top or bottom that must be reflecting or white boundaries but not periodic;
- Periodic planes can only bound other periodic planes or top and bottom planes;
- A single zero-importance cell must be on one side of each periodic plane; and
- All periodic planes must have a common rotational vector normal to the geometry top and bottom.

2.3 Cross Sections

The MCNP code package is incomplete without the associated nuclear data tables. The kinds of tables available and their general features are outlined in this section. The manner in which information contained on nuclear data tables is used in the MCNP code is described in §2.4.

There are two broad objectives in preparing nuclear data tables for the MCNP code. First, the data available to the MCNP code should reproduce the original evaluated data as much as is practical. Second, new data should be brought into the MCNP package in a timely fashion, thereby giving users access to the most recent evaluations. The nuclear data needed by the MCNP code are available at the LANL nuclear data website `https://nucleardata.lanl.gov`.

Ten classes of data tables exist for the MCNP code. They are:

1. continuous-energy neutron interaction data;
2. discrete reaction neutron interaction data;
3. continuous-energy photoatomic interaction data;
4. continuous-energy photonuclear interaction data;
5. neutron dosimetry cross sections;
6. neutron $S(\alpha, \beta)$ thermal data;
7. multigroup neutron, coupled neutron/photon, and charged particles masquerading as neutrons;
8. multigroup photon;
9. electron interaction data; and
10. charged particle interaction data.

In a given simulation, one typically needs data for every projectile-target pair. For nuclear reactions, targets need to be specified at the nuclide or isomer level. This is the case for neutron interactions, charged particle interactions, and photonuclear interactions. For atomic reactions such as electron or photoatomic, targets need to be specified on the atomic level. The code can convert a nuclide specification to atomic automatically. Finally, for neutron thermal scattering, the target is the atom within the molecule of interest. For example, water can have ^{1}H, ^{2}H, ^{3}H,^{16}O, ^{17}O, and ^{18}O as nuclear targets, H and O as atomic targets, and H within H_2O and O within H_2O as thermal scattering targets.

⚠ Caution

Some historical data libraries provide "elemental" nuclear data, in which each nuclide is not cleanly separated out. In general, these datasets are of lower quality than using separate nuclear data for each target and should be avoided.

The means of selecting each class of data through MCNP input is described next. In the remainder of this section, the characteristics of each class of data such as evaluated sources, processing tools, differences between data on the original evaluation and on the MCNP data tables, are described and some hints are provided on how to select the appropriate data tables.

2.3.1 Dataset Selection

The MCNP code needs data tables for a wide variety of physics within the simulation. Most of the data available to the MCNP code is listed in a file called the **xsdir** ("cross section directory"). This file contains a list of table identifiers [§1.2.3], information on where they are stored on disk, and a few physical parameters the code needs to work. Further information on the **xsdir** file can be found in Appendix B.

Through the combination of material specification cards (M, MT, MX) and physics options (MODE, PHYS), the code builds up a list of datasets it needs to load. This list contains target information, physics identifiers, and library identifiers if provided. It then scans through the **xsdir** for tables that match all input information. The first match in the **xsdir** is used. If a dataset is not found, the code will report an error.

The target information is not quite the target identifier described in §1.2.2. When loading nuclear data, the target is decomposed into its constituent Z, A, and S values and comparisons are performed with these. When loading atomic data, the target is also decomposed into Z, A, and S, but when searching, A and S are

set to zero. All currently known available data follows this convention for atomic data. Finally, for neutron thermal scattering, the precise target identifier string is searched for (case insensitive).

This approach has a notable property in that target identifier formats are interchangeable. For example, `U-238` will match the table `92238` and vice versa. This was done to allow the more modern formats to be used to access data provided in older formats.

By convention, only one dataset with a given combination of target information (Z, A, S for nuclear/atomic data, name for $S(\alpha, \beta)$), physics identifier, and library identifier should be present in the **`xsdir`** so that the combination is unique. There may be multiple libraries with a given set of target information and physics identifier made distinct by the library identifier.

2.3.2 Neutron Interaction Data: Continuous-energy and Discrete-reaction

In neutron problems, one neutron interaction table is required for each isotope (or element if using the older "elemental" tables) in the problem. Continuous-energy tables use a physics identifier of `c`, and the discrete reaction tables use `d`.

For most materials, there are many cross-section sets available (represented by different library identifiers) because of multiple sources of evaluated data and different parameters used in processing the data. An evaluated nuclear data set is produced by analyzing experimentally measured cross sections and combining those data with the predictions of nuclear model calculations in an attempt to extract the most accurate interaction description. Preparing evaluated cross-section sets has become a discipline in itself and has developed since the early 1960s. In the US, researchers at many of the national laboratories as well as several industrial firms are involved in such work. The American evaluators joined forces in the mid-1960s to create the national ENDF system [44].

There has been some confusion due to the use of the term ENDF to refer to both a library and a format. The US effort to create a national evaluated nuclear data library led to formation of the Cross Section Evaluation Working Group (CSEWG) in the 1960s. This body standardized the ENDF format, which is used to store evaluated nuclear data files, and created the US ENDF/B library that contains the set of data evaluations currently recommended by CSEWG. Each update of the ENDF/B library receives a unique identifier (discussed below). While ENDF began as a US effort, over time other data centers have adopted the ENDF storage format for their own use (this international standardization has encouraged and facilitated many collaborations). The ENDF-6 format [45] (note that the Arabic number 6 indicates the ENDF format version) has become the international standard for storing evaluated nuclear data and is used by data centers in Europe, Japan, China, Russia, Korea and elsewhere. The user should be aware that there are many evaluated nuclear data libraries of which ENDF/B is only one.

It is worth discussing the ENDF/B library for a moment. The US-based CSEWG meets once a year to discuss and approve changes to the ENDF/B library. In order to track the updates to the ENDF/B library, the following notation has been adopted. The "/B" in ENDF/B is used to indicate the US data library as recommended by CSEWG. There was at one time an ENDF/A that was a repository for other, possibly useful, data. However, this is no longer used. The major version of the library is indicated by a Roman numeral, e.g. ENDF/B-V or ENDF/B-VI. Changes in the major version are generally tied to changes in the standard cross sections. Many cross-section measurements are made relative to the standard cross sections, e.g. elastic scattering off hydrogen or the ^{235}U(n,f) cross section. When one of the standard cross sections is changed, the evaluated data that were based on that standard must be updated. Within a major release, revisions are generally indicated as ENDF/B-VI.2 or ENDF/B-VI.6 where the ".2" and ".6" indicate release 2 and release 6, respectively. A release indicates that some evaluations have been revised, added or deleted. Users should note that neither a major release nor an interim release guarantee that a particular evaluation has been updated. In fact, only a few evaluations change in each release and often the change is limited to a certain energy region. This numbering scheme simply indicates that something within the data library

has changed. It is up to the user to read the accompanying documentation to determine exactly what, if anything, changed. Each ACE table provided with the MCNP package is listed in [46] where its lineage (e.g. ENDF/B-V.0 or ENDF/B-VI.2) is given. The ENDF/B evaluations are available through the National Nuclear Data Center at Brookhaven National Laboratory [http://www.nndc.bnl.gov/].

In addition to the ENDF/B library, many other data centers provide libraries of evaluated data. These include the Japanese Atomic Energy Research Institute's (JAERI) JENDL library, the European JEFF library maintained by the Nuclear Energy Agency (NEA), the Chinese Nuclear Data Center's (CNDC) CENDL library, and the Russian BOFOD library. Other libraries also exist. These centers may provide processed versions of their library in MCNP ACE format. Contact the appropriate center for more information.

In recent years the primary evaluated source of neutron interaction data provided as part of the MCNP code package has been the ENDF/B library (i.e. ENDF/B-V and ENDF/B-VI). However, these have been supplemented with evaluated neutron interaction data tables from other sources, in particular data from Lawrence Livermore National Laboratory's Evaluated Nuclear Data Library (ENDL) library [6] and supplemental evaluations performed in the Nuclear Physics Group in the Theoretical Division at Los Alamos [9–11]. The package also includes older evaluations from previous versions of ENDF/B, ENDL, the Los Alamos Master Data File [47], and the Atomic Weapons Research Establishment in Great Britain.

The MCNP code does not access evaluated data directly from the ENDF format; these data must first be processed into ACE format. The very complex processing codes used for this purpose include NJOY [13, 14] for evaluated data in ENDF format and MCPOINT [48] for evaluated data in the ENDL format.

Data on the MCNP neutron interaction tables include cross sections and emission distributions for secondary particles. Cross sections for all reactions given in the evaluated data are specified. For a particular table, the cross sections for each reaction are given on one energy grid that is sufficiently dense that linear-linear interpolation between points reproduces the evaluated cross sections within a specified tolerance. Over the years this tolerance has been tightened as computer memory has increased. In general, the tables currently available have cross sections that are reproduced to a tolerance of 1% or less, although many recent tables have been created with tolerances of 0.1%. Depending primarily on the number of resolved resonances for each isotope, the resulting energy grid may contain up to ≈100,000 points (see [46] for information about specific tables).

Angular distributions for neutron (and photonuclear) collisions are given in each table for all reactions emitting neutrons or photons (note that older neutron tables may not include photon distributions). The distributions are typically given in the center-of-mass system for elastic scattering and discrete-level inelastic scattering. Other distributions may be given in either the center-of-mass or laboratory system as specified by the ENDF-6 scattering law from which they are derived. Angular distributions are given on a reaction-dependent grid of incident energies.

The sampled angle of scattering uniquely determines the secondary energy for elastic scattering and discrete-level inelastic scattering. For other inelastic reactions, energy distributions of the scattered particles are provided in each table. As with angular distributions, the energy distributions are given on a reaction-dependent grid of incident energies. The energy and angle of particles scattered by inelastic collisions is sampled in a stochastic manner such that the overall emission distribution and energy are preserved for many collisions but not necessarily for any single collision.

When neutron evaluations contain data about secondary photon production, that information appears in the MCNP neutron interaction tables. Many processed data sets contain photon production cross sections, photon angular distributions, and photon energy distributions for each neutron reaction that produces secondary photons. However, the user should be aware that not all evaluations include this information and the information is sometimes approximate, e.g. individual gamma lines may be lumped into average photon emission bins.

Other miscellaneous information on the neutron (and photonuclear) interaction tables includes the atomic weight ratio of the target nucleus, the Q-values of each reaction, and $\overline{\nu}$ data (the average number of neutrons per fission) for fissionable isotopes. In many cases both prompt and total ν are given. Total ν is the default and the TOTNU card can be used to change the default.

Approximations must be made when processing an evaluated data set into ACE format. As mentioned above, cross sections are reproduced to within a certain tolerance, generally less than 1%. Until recently, evaluated angular distributions for non-isotropic secondary particles could only be approximated on ACE tables by 32 equally probable cosine bins. This approximation is extremely fast to use but may not adequately represent a distribution originally given as a 20^{th}-order Legendre polynomial. Starting with the MCNP code, version 4C, tabular angular distributions may be used to represent the scattering angle with a tolerance generally between 0.1% to 1% or better. On the whole, the approximations within more recent ACE tables are small, and MCNP interaction data tables for neutron (and photonuclear) collisions are extremely faithful representations of the original evaluated data.

Discrete-reaction tables are identical to continuous-energy tables except that in the discrete reaction tables all cross sections have been averaged into 262 groups. The averaging is done with a flat weighting function. This is not a multigroup representation; the cross sections are simply given as histograms rather than as continuous curves. The remaining data (angular distributions, energy distributions, ν, etc.) are identical in discrete-reaction and continuous-energy neutron tables. Discrete-reaction tables have been provided in the past as a method of shrinking the required data storage to enhance the ability to run the MCNP code on small machines or in a time-sharing environment. Given the advances in computing speed and storage, they are no longer necessary and should not be used. There original purpose was for preliminary scoping studies. They were never recommended as a substitute for the continuous-energy tables when performing final calculations.

Careful users will want to think about what neutron interaction tables to choose. There is, unfortunately, no strict formula for choosing the tables. The following guidelines and observations are the best that can be offered:

1. Users should, in general, use the most recent data available. The nuclear data evaluation community works hard to continually update these libraries with the most faithful representations of the cross sections and emission distributions.

2. Consider checking the sensitivity of the results to various sets of nuclear data. Try, for example, a calculation with ENDF/B cross sections, and then another with ENDL cross sections. If the results of a problem are extremely sensitive to the choice of nuclear data, it is advisable to find out why.

3. Consider differences in evaluators' philosophies. The Physical Data Group at Livermore is justly proud of its extensive cross-section efforts; their evaluations manifest a philosophy of reproducing the data with the fewest number of points. Livermore evaluations are available mainly in the ".40C" series. We at Los Alamos are particularly proud of the evaluation work being carried out in the Nuclear Data team; generally, these evaluations are the most complex because they are the most thorough.

4. Be aware of the neutron energy spectrum in your problem. For high-energy problems, the "thinned" and discrete reaction data are probably not bad approximations. Conversely, it is essential to use the most detailed continuous-energy set available for problems influenced strongly by transport through the resonance region.

5. Check the temperature at which various data tables have been processed. Do not use a set that is Doppler broadened to 3,000 K for a room temperature calculation.

6. For a coupled neutron/photon problem, be careful that the tables you choose have photon production data available. If possible, use the more-recent sets that have been processed into expanded photon production format.

7. Users should be aware of the differences between the ".50C" series of data tables and the ".51C" series. Both are derived from ENDF/B-V. The ".50C" series is the most faithful reproduction of the evaluated data. The ".51C" series, also called the "thinned" series, has been processed with a less rigid tolerance than the ".50C" series. As with discrete reaction data tables, although not to the same extent, users should be careful when using the "thinned" data for transport through the resonance region.
8. In general, use the best data available. It is understood that the latest evaluations tend to be more complex and therefore require more memory and longer execution times. If you are limited by available memory, try to use smaller data tables such as thinned or discrete-reaction for the minor isotopes in the calculation. Discrete reaction data tables might be used for a parameter study, followed by a calculation with the full continuous-energy data tables for confirmation.

In conclusion, the additional time necessary to choose appropriate neutron interaction data tables rather than simply to accept the defaults often will be rewarded by increased understanding of your calculation.

2.3.3 Photon Interaction Data

Photon interaction cross sections are required for all photon problems. Photon interactions can now account for both photoatomic and photonuclear events. Because these events are different in nature, i.e. elemental versus isotopic, the data are stored on separate tables. Photoatomic data are stored on ACE tables that use the physics identifier `p`.

The "01p" ACE tables were introduced in 1982 and combine data from several sources. The incoherent, coherent, photoelectric and pair production cross sections, the coherent form factors, and incoherent scattering function for this data set come from two sources. For Z equal to 84, 85, 87, 88, 89, 91, and 93, these values are based on the compilation of Storm and Israel [49] and include data for incident photon energies from 1 keV to 15 MeV. For all other elements from Z equal to 1 through 94, the data are based on ENDF/B-IV33 and include data for incident photon energies from 1 keV to 100 MeV. Fluorescence data for Z equal to 1 through 94 are taken from work by Everett and Cashwell [50] as derived from multiple sources.

The "02p" ACE tables were introduced in 1993 and are an extension of the "01p" to higher incident energies [51]. Below 10 MeV the data are identical to the "01p" data (i.e. the cross sections, form factors, scattering function, and fluorescence data in this region are identical). From 10 MeV to the top of the table (either 15 or 100 MeV, depending on the table) the cross-section values are smoothly transitioned from the "01p" values to the values from the Livermore Evaluated Photon Data Library (EPDL89) [7]. Above this transition region, the cross section values are derived from the EPDL89 data and are given for incident energies up to 100 GeV. The pair production threshold was also corrected for some tables.

The "03p" ACE tables were introduced in 2002 and are an extension of the "02p" tables to include additional data. The energy of a photon after an incoherent (Compton) collision is a function of the momentum of the bound electron involved in the collision. To calculate this effect (which is seen as a broadening of the Compton peak), it is necessary to know the probability with which a photon interacts with an electron from a particular shell and the momentum profile for the electrons of each shell. The probabilities and momentum profile data of Biggs et al. [52] are included in the "03p" tables. All other data in the "03p" are identical to the "02p" data. The ability to use the new data for broadening of the Compton scattering energy requires MCNP5 or later; however, these tables are compatible with older versions of the code (you simply will not access or use the new data).

The "04p" ACE tables were introduced in 2002 and contain the first completely new data set since 1982. These tables were processed from the ENDF/B-VI.8 library. The ENDF/B-VI.8 photoatomic and atomic relaxation data are in turn based upon the EPDL97 [53] library. They include incoherent, coherent, photoelectric and pair production cross sections for incident energies from 1 keV to 100 GeV and Z equal to 1 to 100. They

also include coherent form factors, incoherent scattering functions, and fluorescence data derived from the ENDF/B-VI.8 data. It should be noted that the form factor and scattering data have been evaluated and are hard-coded in the MCNP code (in the GETXST subroutine). The fluorescence data use the traditional scheme defined by Everett and Cashwell [50] but updated and consistent with the new data. Also included are the bound electron momenta of Biggs et al. [52] (i.e. identical to those data in the "03p" tables). This is the recommended data set. More information on the "04p" ACE tables can be found in [54].

For each element the photoatomic interaction libraries contain an energy grid—explicitly including the photoelectric edges and the pair production threshold—the incoherent, coherent, photoelectric and pair production cross sections (all stored as the logarithm of the value to facilitate log-log interpolation). The total cross section is not stored; instead it is calculated from the partial cross sections as needed. The energy grid for each table is tailored specifically for that element. The average material heating due to photon scattering is calculated by the processing code and included as a tabulation on the main energy grid. The incoherent scattering function and coherent form factors are tabulated as a function of momentum transfer on a predefined, fixed-momenta grid. Average fluorescence data (according to the scheme of Everett and Cashwell [50]) are also included. The most recent data (on the 03p and 04p libraries) also include momentum profile data for broadening of the photon energy from Compton scattering from bound electrons.

The determination of directions and energies of atomically scattered photons requires information different from the sets of angular and energy distributions found on neutron interaction tables. The angular distribution for fluorescence x-rays from the relaxation cascade after a photoelectric event is isotropic. The angular distributions for coherent and incoherent scattering come from sampling the well-known Thomson and Klein-Nishina formulas, respectively. By default, this sampling accounts for the form factor and scattering function data at incident energies below 100 MeV. Above, 100 MeV (or at the user's request) the form factor and scattering function data are ignored (a reasonable approximation for high-energy photons). The energy of an incoherently scattered photon is calculated from the sampled scattering angle. If available, this energy is modified to account for the momentum of the bound electron.

Very few approximations are made in the various processing codes used to transfer photon data from ENDF into the format of MCNP photon interaction tables. Cross sections are reproduced exactly as given (except as the logarithm of the value). Form factors and scattering functions are reproduced as given; however, the momentum transfer grid on which they are tabulated may be different from that of the original evaluation. Heating numbers are calculated values, not given in evaluated sets, but inferred from them. Fluorescence data are calculated using the scheme developed by Everett and Cashwell [50].

Photonuclear data tables use the physics identifier `u`. Photon interactions can include photonuclear events. Early data distribution included tables for only 13 nuclides. Because of this, photonuclear physics must be explicitly turned on. If on, a table must be provided for each nuclide of every material or a fatal error will occur and the simulation will not run. Around 2000, more than 150 other photonuclear data evaluations were created as part of an IAEA collaboration [55]. Around 2020, evaluations for 219 nuclides became available through the IAEA's Nuclear Data Services website (https://www-nds.iaea.org/photonuclear) [56].

Photonuclear interaction data describe nuclear events with specific isotopes. The reaction descriptions use the same ENDF-6 format as used for neutron data. Their processing, storage as ACE tables, and sampling in a simulation are completely analogous to what is done for neutrons. See the previous discussion of the neutron data for more details. Note that the photonuclear data available so far are complete in the sense that they provide secondary particle distributions for all light-particles, i.e. photons, neutrons, protons, alphas, etc. At this time, the MCNP code makes use of the photon and neutron emission distributions.

The selection of photon interaction data has become more complicated. Let us first examine the simple cases. Photon or photon/electron problems where photonuclear events are to be ignored (i.e. photonuclear physics is explicitly off) should specify the material composition on the `M`n card by mass or weight fraction of each element, i.e. setting $A = 0$ as shown in §1.2.2. The next most simple case is a coupled neutron-photon problem that will explicitly ignore photonuclear events. In this case, one should specify the material composition

according to the rules discussed in the previous section on neutron data tables. Given an isotopic material component, e.g. Al-27, the appropriate elemental photoatomic table will be selected, e.g. Al-0. If no evaluation identifier is given, the default (first) table from the **xsdir** file will be used. If a particular evaluation set is desired, the PLIB option on the Mn card may be used to select all photoatomic tables from a given library. It is recommended in all cases that the photoatomic tables for any given problem all be from the same library (these data sets are created in masse and thus are self-consistent across a library).

The most complicated case for material definition is the selection of tables for coupled neutron-photon problems where photonuclear events are not ignored. In such a case, the composition must be chosen based on the availability of most appropriate isotopic neutron and photonuclear tables as needed for the specific problem at hand. The MXn card may be used to accommodate mismatches in the availability of specific isotopes [§5.6.3]. As always, a fully specified identifier, e.g. Al-27.24u, will ensure that a specific table is selected. The PNLIB option on the material card may be used to select all photonuclear tables from a specific library. Otherwise, the code will select the first match in the **xsdir** file. Note that if no photonuclear table is available for the given target, the problem will report the error and will not run. See the discussion in the description of the MXn card for more information [§5.6.3].

2.3.4 Electron Interaction Data

Electron interaction data tables are required both for problems in which electrons are actually transported, and for photon problems in which the thick-target bremsstrahlung model is used. Electron data tables use the physics identifier e, and are selected by default when the problem mode requires them. There are two electron interaction data libraries: **el** (data suffix of .01e) and **el03** (data suffix of .03e).

The electron libraries contain data on an element-by-element basis for atomic numbers from Z equal 1 to 94. The library data contain energies for tabulation, radiative stopping power parameters, bremsstrahlung production cross sections, bremsstrahlung energy distributions, K-edge energies, Auger electron production energies, parameters for the evaluation of the Goudsmit-Saunderson theory for angular deflections based on the Riley cross-section calculation, and Mott correction factors to the Rutherford cross sections also used in the Goudsmit-Saunderson theory. The **el03** library also includes the atomic data of Carlson used in the density effect calculation. Internal to the code at run-time, data are calculated for electron stopping powers and ranges, K x-ray production probabilities, knock-on probabilities, bremsstrahlung angular distributions, and the Landau-Blunck-Leisegang theory of energy-loss fluctuations. The **el03** library is derived from the ITS3.0 code system [57]. Discussions of the theoretical basis for these data and references to the relevant literature are presented in [§2.4.5].

The hierarchy rules for electron cross sections require that each material must use the same electron library. If a specific library identifier is selected on a material card, that choice of library will be used as the default for all elements in that material. Alternatively, the default electron library for a given material can be chosen by specifying the ELIB option on the M card. In the absence of any specification, the MCNP code will use the first electron data table listed in the **xsdir** file for the relevant element.

⚠ Caution

Under no circumstances should data tables from different libraries be specified for use in the same material (e.g., "m6 12000.01e 1 20000.03e 1" should not be used). This will result in a fatal error as reported at run time. Overriding this error with a FATAL option [Table 3.6] will result in unreliable results.

2.3.5 Neutron Dosimetry Cross Sections

Dosimetry cross-section tables cannot be used for transport through material. These incomplete cross-section sets provide energy-dependent neutron cross sections to the MCNP code for use as response functions with

the FM tally feature, e.g. they may be used in the calculation of a reaction rate. Identifiers for dosimetry tables have the physics identifier `y`. Remember, dosimetry cross-section tables have no effect on the particle transport of a problem.

The available dosimetry cross sections are from three sources: ENDF/B-V Dosimetry Tape 531, ENDF/B-V Activation Tape 532, and ACTL [8]—an evaluated neutron activation cross-section library from the Lawrence Livermore National Laboratory. Various codes have been used to process evaluated dosimetry data into the format of MCNP dosimetry tables.

Data on dosimetry tables are simply energy-cross-section pairs for one or more reactions. The energy grids for all reactions are independent of each other. Interpolation between adjacent energy points can be specified as histogram, linear-linear, linear-log, log-linear, or log-log. With the exception of the tolerance involved in any reconstruction of point-wise cross sections from resonance parameters, evaluated dosimetry cross sections can be reproduced on the MCNP data tables with no approximation.

When specifying a dosimetry dataset on a material card, the full specifier must be used including the library identifier. There are no defaults for dosimetry tables. The code will prevent materials with dosimetry data from being used in the geometry. These materials can only be used as the multiplier in `FM` cards.

2.3.6 Neutron Thermal $S(\alpha, \beta)$ Tables

Thermal $S(\alpha, \beta)$ tables are not required, but they are essential to get correct answers in problems involving neutron thermalization. The thermal scattering library based on ENDF/V-VIII.0 provides the material identifiers for use on the `MT`n card(s). The data on these material identifier tables encompass those required for a complete representation of thermal neutron scattering by molecules and crystalline solids. The source of $S(\alpha, \beta)$ data is a special set of ENDF tapes [58]. The THERMR and ACER modules of the NJOY [13, 14] system have been used to process the evaluated thermal data into a format appropriate for the MCNP code.

Data are for neutron energies generally less than 4 eV. Cross sections are tabulated on table-dependent energy grids; inelastic scattering cross sections are always given and elastic scattering cross sections are sometimes given. Correlated energy-angle distributions are provided for inelastically scattered neutrons. A set of equally probable final energies is tabulated for each of several initial energies. Further, a set of equally probable cosines or cosine bins is tabulated for each combination of initial and final energies. Elastic scattering data can be derived from either an incoherent or a coherent approximation. In the incoherent case, equally probable cosines or cosine bins are tabulated for each of several incident neutron energies. In the coherent case, scattering cosines are determined from a set of Bragg energies derived from the lattice parameters. During processing, approximations to the evaluated data are made when constructing equally probable energy and cosine distributions.

2.3.7 Multigroup Tables

Multigroup cross-section libraries are the only libraries allowed in multi-group/adjoint problems. Neutron multigroup problems cannot be supplemented with $S(\alpha, \beta)$ thermal libraries; the thermal effects must be included in the multigroup neutron library. Photon problems cannot be supplemented with electron libraries; the electrons must be part of the multigroup photon library. Neutron multigroup data has the `m` physics suffix, and photons use `g`.

Although continuous-energy data are more accurate than multigroup data, the multigroup option is useful for a number of important applications: (1) comparison of deterministic (S_N) transport codes to Monte Carlo; (2) use of adjoint calculations in problems where the adjoint method is more efficient; (3) generation of adjoint importance functions; (4) cross-section sensitivity studies; (5) solution of problems for which continuous-cross

sections are unavailable; and (6) charged particle transport using the Boltzmann-Fokker-Planck algorithm in which charged particles masquerade as neutrons.

Multigroup cross sections are very problem dependent. Some multigroup libraries are available from the Transport Methods Group at Los Alamos but must be used with caution. Users are encouraged to generate or get their own multigroup libraries and then use the supplementary code CRSRD [59] to convert them to the MCNP code format. Reference [59] describes the conversion procedure. This report also describes how to use both the multigroup and adjoint methods in the MCNP code and presents several benchmark calculations demonstrating the validity and effectiveness of the multigroup/adjoint method.

To generate cross-section tables for electron/photon transport problems that will use the multigroup Boltzmann-Fokker-Planck algorithm [60], the CEPXS [61–63] code developed by Sandia National Laboratory and available from RSICC can be used. The CEPXS manuals describe the algorithms and physics database upon which the code is based; the physics package is essentially the same as ITS version 2.1. The keyword "MONTE-CARLO" is needed in the CEPXS input file to generate a cross-section library suitable for input into CRSRD; this undocumented feature of the CEPXS code should be approached with caution.

2.4 Physics

The physics of neutron, photon, and electron interactions is the very essence of the MCNP code. A review of charged particle transport capabilities in the MCNP code can be found in [42]. For a description of all high-energy event generators used by the MCNP code, see [64]. This section may be considered a software requirements document in that it describes the equations the MCNP code is intended to solve. All the sampling schemes essential to the random walk are presented or referenced. But first, particle weight and particle tracks, two concepts that are important for setting up the input and for understanding the output, are discussed in the following sections.

2.4.1 Statistical Weight

At the most fundamental level, weight is a tally multiplier. That is, the tally contribution for a weight w is the unit weight tally contribution multiplied by w. Weight is an adjustment for deviating from a direct physical simulation of the transport process. Note that if a Monte Carlo code always sampled from the same distributions as nature does, then the Monte Carlo code would have the same mean and variance as seen in nature. Quite often, the natural variance is unacceptably high and the Monte Carlo code modifies the sampling using some form of "variance reduction" [§2.7]. The variance reduction methods use weighting schemes to produce the same mean as the natural transport process, but with lower calculational variance than the natural variance of the transport process.

With the exception of the pulse height tally (`F8`), all tallies in the MCNP code are made by individual particles. In this case, weight is assigned to the individual particles as a "particle weight." The manual discusses the "particle weight" cases first and afterward discusses the weight associated with the `F8` tally.

2.4.1.1 Particle Weight

If the MCNP code were used only to simulate exactly physical transport, then each MCNP particle would represent one physical particle and would have unit weight. However, for computational efficiency, the MCNP code allows many techniques that do not exactly simulate physical transport. For instance, each MCNP particle might represent a number w of particles emitted from a source. This number w is the initial weight of the MCNP particle. The w physical particles all would have different random walks, but the one MCNP

particle representing these w physical particles will only have one random walk. Clearly this is not an exact simulation; however, the true number of physical particles is preserved in the MCNP code in the sense of statistical averages and therefore in the limit of a large number of MCNP source particles (of course including particle production or loss if they occur). Each MCNP particle result is multiplied by the weight so that the full results of the w physical particles represented by each MCNP particle are exhibited in the final results (tallies). This procedure allows users to normalize their calculations to whatever source strength they desire. The default normalization is a weight of one per MCNP source particle. A second normalization to the number of Monte Carlo histories is made in the results so that the expected means will be independent of the number of source particles actually initiated in the MCNP calculation.

The utility of particle weight, however, goes far beyond simply normalizing the source. Every Monte Carlo biasing technique alters the probabilities of random walks executed by the particles. The purpose of such biasing techniques is to increase the number of particles that sample some part of the problem of special interest (1) without increasing (and sometimes actually decreasing) the sampling of less interesting parts of the problem, and (2) without erroneously affecting the expected mean physical result (tally). This procedure, properly applied, increases precision in the desired result compared to an unbiased calculation taking the same computing time. For example, if an event is made $\sqrt{2}$ times as likely to occur (as it would occur without biasing), the tally ought to be multiplied by $1/\sqrt{2}$ so that the expected average tally is unaffected. This tally multiplication can be accomplished by multiplying the particle weight by $1/\sqrt{2}$ because the tally contribution by a particle is always multiplied by the particle weight in the MCNP code. Note that weights need not be integers.

In short, particle weight is a number carried along with each MCNP particle, representing that particle's relative contribution to the final tallies. Its magnitude is determined to ensure that whenever the MCNP code deviates from an exact simulation of the physics, the expected physical result nonetheless is preserved in the sense of statistical averages, and therefore in the limit of large MCNP particle numbers. Its utility is in the manipulation of the number of particles, sampling just a part of the problem to achieve the same results and precision, obviating a full unbiased calculation which has a longer computing time.

2.4.1.2 Pulse-height Tally (F8) Weight

Unlike other tallies in the MCNP code, the pulse height tally depends on a collection of particles instead of just individual particles. Because of this, a weight is assigned to each collection of tallying particles. It is this "collective weight" that multiplies the F8 tally, not the particle weight.

When variance reduction is used, a "collective weight" is assigned to every collection of particles. If variance reduction techniques have made a collection's random walk q times as likely as without variance reduction, then the collective weight is multiplied by $1/q$ so that the expected F8 tally of the collection is preserved. The interested reader should consult [65, 66] for more details.

2.4.2 Particle Tracks

When a particle starts out from a source, a particle track is created. If that track is split 2 for 1 at a splitting surface or collision, a second track is created and there are now two tracks from the original source particle, each with half the single track weight. If one of the tracks has an (n,2n) reaction, one more track is started for a total of three. A track refers to each component of a source particle during its history. Track length tallies use the length of a track in a given cell to determine a quantity of interest, such as fluence, flux, or energy deposition. Tracks crossing surfaces are used to calculate fluence, flux, or pulse-height energy deposition (surface estimators). Tracks undergoing collisions are used to calculate multiplication and criticality (collision estimators).

Within a given cell of fixed composition, the method of sampling a collision along the track is determined using the following theory. The probability of a first collision for a particle between l and $l + \mathrm{d}l$ along its line of flight is given by

$$p(l)\mathrm{d}l = \exp(-\Sigma_\mathrm{t} l)\Sigma_\mathrm{t}\mathrm{d}l, \tag{2.2}$$

where Σ_t is the macroscopic total cross section of the medium and is interpreted as the probability per unit length of a collision. Setting ξ the random number on $[0, 1)$, to be

$$\xi = \int_0^l \exp(-\Sigma_\mathrm{t} s)\Sigma_\mathrm{t}\mathrm{d}s = 1 - \exp(-\Sigma_\mathrm{t} l), \tag{2.3}$$

it follows that

$$l = -\frac{1}{\Sigma_\mathrm{t}} \ln(1 - \xi). \tag{2.4}$$

However, because $1 - \xi$ is distributed in the same manner as ξ and hence may be replaced by ξ, we obtain the well-known expression for the distance to collision,

$$l = -\frac{1}{\Sigma_\mathrm{t}} \ln(\xi). \tag{2.5}$$

2.4.3 Neutron Interactions

When a particle (representing any number of neutrons, depending upon the particle weight) collides with a nucleus, the following sequence occurs:

1. the collision nuclide is identified;
2. either the $S(\alpha, \beta)$ treatment is used or the velocity of the target nucleus is sampled for low-energy neutrons;
3. photons are optionally generated for later transport;
4. neutron capture (that is, neutron disappearance by any process) is modeled;
5. if the energy of the neutron is low enough and an appropriate $S(\alpha, \beta)$ table is present, the collision is modeled by the $S(\alpha, \beta)$ treatment;
6. otherwise, either elastic scattering or an inelastic reaction (including fission) is selected, and the new energy and direction of the outgoing track(s) are determined.

2.4.3.1 Selection of Collision Nuclide

If there are n different nuclides forming the material in which the collision occurred, and if ξ is a random number on the unit interval $[0, 1)$, then the k^{th} nuclide is chosen as the collision nuclide if

$$\sum_{i=1}^{k-1} \Sigma_{\mathrm{t},i} < \xi \sum_{i=1}^{n} \Sigma_{\mathrm{t},i} \leq \sum_{i=1}^{k} \Sigma_{\mathrm{t},i}, \tag{2.6}$$

where $\Sigma_{\mathrm{t},i}$ is the macroscopic total cross section of nuclide i. If the energy of the neutron is low enough (below about 4 eV) and the appropriate $S(\alpha, \beta)$ table is present, the total cross section is the sum of the capture cross section from the regular cross-section table and the elastic and inelastic scattering cross sections from the $S(\alpha, \beta)$ table. Otherwise, the total cross section is taken from the regular cross-section table and is adjusted for thermal effects [§2.4.3.2].

2.4.3.2 Free Gas Thermal Treatment

A collision between a neutron and an atom is affected by the thermal motion of the atom, and in most cases, the collision is also affected by the presence of other atoms nearby. The thermal motion cannot be ignored in many applications of the MCNP code without serious error. The effects of nearby atoms are also important in some applications. The MCNP code uses a thermal treatment based on the free gas approximation to account for the thermal motion. It also has an explicit $S(\alpha,\beta)$ capability that takes into account the effects of chemical binding and crystal structure for incident neutron energies below about 4 eV, but is available for only a limited number of substances and temperatures. The $S(\alpha,\beta)$ capability is described in §2.4.3.6.

The free gas thermal treatment in the MCNP code assumes that the medium is a free gas and also that, in the range of atomic weight and neutron energy where thermal effects are significant, the elastic scattering cross section at zero temperature is nearly independent of the energy of the neutron and that the reaction cross sections are nearly independent of temperature. These assumptions allow the MCNP code to have a thermal treatment of neutron collisions that runs almost as fast as a completely non-thermal treatment and that is adequate for most practical problems.

With the above assumptions, the free gas thermal treatment consists of adjusting the elastic cross section and taking into account the velocity of the target nucleus when the kinematics of a collision are being calculated. The MCNP free gas thermal treatment effectively applies to elastic scattering only.

Cross-section libraries processed by NJOY already include Doppler broadening of elastic, capture, fission, and other low-threshold absorption cross-sections (< 1 eV). Inelastic cross sections are never broadened by NJOY.

2.4.3.2.1 Adjusting the Elastic Cross Section

The first aspect of the free gas thermal treatment is to adjust the zero-temperature elastic cross section by raising it by the factor

$$F = \left(1 + 0.5/a^2\right)\mathrm{erf}(a) + \exp\left(-a^2\right)/\left(a\sqrt{\pi}\right), \tag{2.7}$$

where $a = \sqrt{AE/kT}$, A is the atomic weight of the nucleus, E is the incident neutron energy, and T is the material temperature. For speed, F is approximated by $F = 1 + 0.5/a^2$ when $a \geq 2$ and by linear interpolation in a table of 51 values of aF when $a < 2$. Both approximations have relative errors less than 0.0001. The total cross section also is increased by the amount of the increase in the elastic cross section.

The adjustment to the elastic and total cross sections is done partly in the setup of a problem and partly during the actual transport calculation. No adjustment is made if the elastic cross section in the data library was already processed to the temperature that is needed in the problem. If all of the cells that contain a particular nuclide have the same temperature, which is constant in time, that is different from the temperature of the library, the elastic and total cross sections for that nuclide are adjusted to that temperature during the setup so that the transport will run a little faster. Otherwise, these cross sections are reduced, if necessary, to zero temperature during the setup and the thermal adjustment is made when the cross sections are used. For speed, the thermal adjustment is omitted if the neutron energy is greater than $500\,kT/A$. At that energy the adjustment of the elastic cross section would be less than 0.1%.

Note that this adjustment of the nuclear data is less accurate than the one used within NJOY, as NJOY will handle more reactions and does not assume constant data. As such, it is recommended to use datasets Doppler-broadened to the temperature of interest, rather than relying on this adjustment. See the discussion in the TMP card for more information.

2.4.3.2.2 Sampling the Velocity of the Target Nucleus

The second aspect of the free gas thermal treatment takes into account the velocity of the target nucleus when the kinematics of a collision are being calculated. The target velocity is sampled and subtracted from the velocity of the neutron to get the relative velocity. The collision is sampled in the target-at-rest frame and the outgoing velocities are transformed to the laboratory frame by adding the target velocity.

There are different schools of thought as to whether the relative energy between the neutron and target, E_r, or the laboratory frame incident neutron energy (target-at-rest), E_o, should be used for all the kinematics of the collision. E_o is used in the MCNP code to obtain the distance-to-collision, select the collision nuclide, determine energy cutoffs, generate photons, generate fission sites for the next generation of a KCODE criticality problem, for $S(\alpha,\beta)$ scattering, and for capture. E_r is used for everything else in the collision process, namely elastic and inelastic scattering, including fission and $(\mathrm{n},x\mathrm{n})$ reactions. It is shown in Eq. (2.8) that E_r is based upon $v_{\mathrm{rel.}}$ that is based upon the elastic scattering cross section, and, therefore, E_r is truly valid only for elastic scatter. However, the only significant thermal reactions for stable isotopes are absorption, elastic scattering, and fission. ^{181}Ta has a 6 keV threshold inelastic reaction; all other stable isotopes have higher inelastic thresholds. Metastable nuclides like $^{242\mathrm{m}}$Am have inelastic reactions all the way down to zero, but these inelastic reaction cross sections are neither constant nor $1/v$ cross sections and these nuclides are generally too massive to be affected by the thermal treatment anyway. Furthermore, fission is very insensitive to incident neutron energy at low energies. The fission secondary energy and angle distributions are nearly flat or constant for incident energies below about 500 keV. Therefore, it makes no significant difference if E_r is used only for elastic scatter or for other inelastic collisions as well. At thermal energies, whether E_r or E_o is used only makes a difference for elastic scattering.

If the energy of the neutron is greater than $400\,kT$ and the target is not ^{1}H, the velocity of the target is set to zero. Otherwise, the target velocity is sampled as follows. The free-gas kernel is a thermal interaction model that results in a good approximation to the thermal flux spectrum in a variety of applications and can be sampled without tables. The effective scattering cross section in the laboratory system for a neutron of kinetic energy E is

$$\sigma_{\mathrm{s}}^{\mathrm{eff}}(E) = \frac{1}{v_n}\iint \sigma_{\mathrm{s}}(v_{\mathrm{rel.}})v_{\mathrm{rel.}}p(V)\mathrm{d}v\frac{\mathrm{d}\mu_t}{2}. \tag{2.8}$$

Here, $v_{\mathrm{rel.}}$ is the relative velocity between a neutron moving with a scalar velocity v_n and a target nucleus moving with a scalar velocity V, and μ_t is the cosine of the angle between the neutron and the target direction-of-flight vectors. The equation for $v_{\mathrm{rel.}}$ is

$$v_{\mathrm{rel.}} = \left(v_n^2 + V^2 - 2v_nV\mu_t\right)^{1/2}. \tag{2.9}$$

The scattering cross section at the relative velocity is denoted by $\sigma_{\mathrm{s}}(v_{\mathrm{rel.}})$, and $p(V)$ is the probability density function for the Maxwellian distribution of target velocities,

$$p(V) = \frac{4}{\pi^{1/2}}\beta^3V^2\exp\left(-\beta^2V^2\right), \tag{2.10}$$

with β defined as

$$\beta = \left(\frac{AM_n}{2kT}\right)^{1/2}, \tag{2.11}$$

where A is the mass of a target nucleus in units of the neutron mass, M_n is the neutron mass in MeV-sh^2/cm^2, and kT is the equilibrium temperature of the target nuclei in MeV.

The most probable scalar velocity V of the target nuclei is $1/\beta$, which corresponds to a kinetic energy of kT for the target nuclei. This is not the average kinetic energy of the nuclei, which is $3kT/2$. The quantity that the MCNP code expects on the TMPn input card is kT and not just T [§5.7.5]. Note that kT is not a function of the particle mass and is therefore the kinetic energy at the most probable velocity for particles of any mass.

Equation (2.8) implies that the probability distribution for a target velocity V and cosine μ_t is

$$P(V,\mu_t) = \frac{\sigma_s(v_{rel.})v_{rel.}p(V)}{2\sigma_s^{eff.}(E)v_n}. \tag{2.12}$$

It is assumed that the variation of $\sigma_s(v)$ with target velocity can be ignored. The justification for this approximation is that (1) for light nuclei, $\sigma_s(v_{rel.})$ is slowly varying with velocity, and (2) for heavy nuclei, where $\sigma_s(v_{rel.})$ can vary rapidly, the moderating effect of scattering is small so that the consequences of the approximation will be negligible. As a result of the approximation, the probability distribution actually used is

$$P(V,\mu_t) = \sqrt{v_n^2V^2 - 2Vv_n\mu_t}V^2\exp\left(-\beta^2V^2\right). \tag{2.13}$$

Note that the above expression can be written as

$$P(V,\mu_t) = \frac{\sqrt{v_n^2V^2 - 2Vv_n\mu_t}}{v_n+V}\left[V^3\exp\left(-\beta^2V^2\right) + v_nV^2\exp\left(-\beta^2V^2\right)\right]. \tag{2.14}$$

As a consequence, the following algorithm is used to sample the target velocity.

1. With probability $\alpha = 1/(1+(\sqrt{\pi}\beta v_n/2))$, the target velocity V is sampled from the distribution

$$P_1(V) = 2\beta^4V^3\exp\left(-\beta^2V^2\right). \tag{2.15}$$

   The transformation $V = \sqrt{y}/\beta$ reduces this distribution to the sampling distribution $P(y) = y\exp(-y)$. The MCNP code actually codes $1-\alpha$.

2. With probability $1-\alpha$, the target velocity is sampled from the distribution

$$P_2(V) = \left(4\beta^3/\sqrt{\pi}\right)V^2\exp\left(-\beta^2V^2\right). \tag{2.16}$$

   Substituting $V = y/\beta$ reduces the distribution to the sampling distribution for y to

$$P(y) = \left(4/\sqrt{\pi}\right)y^2\exp\left(-y^2\right).$$

3. The cosine of the angle between the neutron velocity and the target velocity is sampled uniformly on the interval $-1 \le \mu_t \le 1$.

4. The rejection function $R(V,\mu_t)$ is computed using

$$R(V,\mu_t) = \frac{\sqrt{v_n^2+V^2-2Vv_n\mu_t}}{v_n+V} \le 1. \tag{2.17}$$

With probability $R(V,\mu_t)$, the sampling is accepted; otherwise, the sampling is rejected and the procedure is repeated. The minimum efficiency of this rejection algorithm corresponding to assuming $V = v_n = v_{rel.}$ averaged over μ_t is

$$\frac{\int_{-1}^{1}R(v_{rel.},\mu_t)\mathrm{d}\mu_t}{\int_{-1}^{1}\mathrm{d}\mu_t} = \frac{1}{2}\int_{-1}^{1}\frac{\sqrt{v_{rel.}^2+v_{rel.}^2-2v_{rel.}^2\mu_t}}{2v_{rel.}}\mathrm{d}\mu_t = \frac{\sqrt{2}}{4}\int_{-1}^{1}\sqrt{1-\mu_t}\mathrm{d}\mu_t = \frac{2}{3}, \tag{2.18}$$

which approaches 100% as either the incident neutron energy approaches zero or becomes much larger than kT.

For more accuracy, the probability distribution in Equation 2.12 can be directly sampled without the constant cross-section approximation. This is enabled through the DBRC card. This is not enabled by default.

2.4.3.3 Optional Generation of Photons

Photons are generated if the problem is a combined neutron/photon run and if the collision nuclide has a nonzero photon production cross section. The number of photons produced is a function of neutron weight, neutron source weight, photon weight limits (entries on the PWT card), photon production cross section, neutron total cross section, cell importance, and the importance of the neutron source cell. No more than 10 photons may be born from any neutron collision. In a KCODE calculation, secondary photon production from neutrons is turned off during the inactive cycles.

Because of the many low-weight photons typically created by neutron collisions, Russian roulette is played for particles with weight below the bounds specified on the PWT card, resulting in fewer particles, each having a larger weight. The created photon weight before Russian roulette is

$$W_{\mathrm{p}} = \frac{W_{\mathrm{n}}\sigma_{\gamma}}{\sigma_{\mathrm{t}}}, \tag{2.19}$$

where

W_{p}	is the photon weight,
W_{n}	is the neutron weight,
σ_{γ}	is the photon production cross section, and
σ_{t}	is the total neutron cross section.

Both σ_{γ} and σ_{t} are evaluated at the incoming neutron energy without the effects of the thermal free gas treatment because nonelastic cross sections are assumed independent of temperature.

The Russian roulette game is played according to neutron cell importances for the collision and source cell. For a photon produced in cell i where the minimum weight set on the PWT card is $W_i^{\mathrm{min.}}$, let I_i be the neutron importance in cell i and let I_s be the neutron importance in the source cell. If $W_{\mathrm{p}} > W_i^{\mathrm{min.}} I_s/I_i$, one or more photons will be produced. The number of photons created is N_{p}, where

$$N_{\mathrm{p}} = \frac{W_{\mathrm{p}} I_i}{5 W_i^{\mathrm{min.}} I_s} + 1\,,\ N_{\mathrm{p}} \leq 10. \tag{2.20}$$

Each photon is stored in the bank with weight $W_{\mathrm{p}}/N_{\mathrm{p}}$. If $W_{\mathrm{p}} < W_i^{\mathrm{min.}} I_s/I_i$, Russian roulette is played and the photon survives with probability $W_{\mathrm{p}} I_i/\left(W_i^{\mathrm{min.}} I_s\right)$ and is given the weight $W_i^{\mathrm{min.}} I_s/I_i$.

If weight windows are not used and if the weight of the starting neutrons is not unity, setting all the $W_i^{\mathrm{min.}}$ on the PWT card to negative values will make the photon minimum weight relative to the neutron source weight. This will make the number of photons being created roughly proportional to the biased collision rate of neutrons. It is recommended for most applications that negative numbers be used and be chosen to produce from one to four photons per source neutron. The default values for $W_i^{\mathrm{min.}}$ on the PWT card are -1, which should be adequate for most problems using cell importances.

If energy-independent weight windows are used, the entries on the PWT card should be the same as on the WWN1:p card. If energy-dependent photon weight windows are used, the entries on the PWT card should be the minimum WWNn:p entry for each cell, where n refers to the photon weight window energy group. This will cause most photons to be born within the weight window bounds.

Any photons generated at neutron collision sites are temporarily stored in the bank. There are two methods for determining the exiting energies and directions, depending on the form in which the processed photon production data are stored in a library. The first method has the evaluated photon production data processed

into an "expanded format" [67]. In this format, energy-dependent cross sections, energy distributions, and angular distributions are explicitly provided for every photon-producing neutron interaction. In the second method, used with data processed from older evaluations, the evaluated photon production data have been collapsed so that the only information about secondary photons is in a matrix of 20 equally probable photon energies for each of 30 incident neutron energy groups. The sampling techniques used in each method are now described.

2.4.3.3.1 Expanded Photon Production Method

In the expanded photon production method, the reaction n responsible for producing the photon is sampled from

$$\sum_{i=1}^{n-1} \sigma_i < \xi \sum_{i=1}^{N} \sigma_i \leq \sum_{i=1}^{n} \sigma_i, \tag{2.21}$$

where ξ is a random number on the interval $[0, 1)$, N is the number of photon production reactions, and σ_i is the photon production cross section for reaction i at the incident neutron energy. Note that there is no correlation between the sampling of the type of photon production reaction and the sampling of the type of neutron reaction described in §2.4.3.5.

Just as every neutron reaction (for example, $(\mathrm{n}, 2\mathrm{n})$) has associated energy-dependent angular and energy distributions for the secondary neutrons, every photon production reaction (for example, $(\mathrm{n}, \mathrm{p}\gamma)$) has associated energy-dependent angular and energy distributions for the secondary photons. The photon distributions are sampled in much the same manner as their counterpart neutron distributions.

All non-isotropic secondary photon angular distributions are represented by either 32 equiprobable cosine bins or by a tabulated angular distribution. The distributions are given at a number of incident neutron energies. All photon-scattering cosines are sampled in the laboratory system. The sampling procedure is identical to that described for secondary neutrons in §2.4.3.5.1.

Secondary photon energy distributions are also a function of incident neutron energy. There are two representations of secondary photon energy distributions allowed in ENDF-6 format: tabulated spectra and discrete (line) photons. Correspondingly, there are two laws used in the MCNP code for the determination of secondary photon energies. Law 4 provides for representation of a tabulated photon spectra possibly including discrete lines. Law 2 is used solely for discrete photons. These laws are described in more detail beginning in §2.4.3.5.4.1.

The expanded photon production method has clear advantages over the original 30×20 matrix method [§2.4.3.3.2]. In coupled neutron/photon problems, users should attempt to specify data sets that contain photon production data in expanded format. Such data sets are identified by "yes" entries in the GPD column in [46]. However, it should be noted that the evaluations from which these data are processed may not include all discrete lines of interest; evaluators may have binned sets of photons into average spectra that simply preserve the energy distribution.

2.4.3.3.2 30×20 Photon Production Method

For lack of better terminology, we will refer to the photon production data contained in older libraries as "30×20 photon production" data. In contrast to expanded photon production data, there is no information about individual photon production reactions in the 30×20 data. This method is not used in modern tables and is only included to maintain backwards compatibility for very old data libraries.

The only secondary photon data are a 30×20 matrix of photon energies; that is, for each of 30 incident neutron energy groups there are 20 equally probable exiting photon energies. There is no information regarding secondary photon angular distributions; therefore, all photons are taken to be produced isotropically in the laboratory system.

There are several problems associated with 30×20 photon production data. The 30×20 matrix is an inadequate representation of the actual spectrum of photons produced. In particular, discrete photon lines are not well represented, and the high-energy tail of a photon continuum energy distribution is not well sampled. Also, the multigroup representation is not consistent with the continuous-energy nature of the MCNP code. Finally, not all photons should be produced isotropically. None of these problems exists for data processed into the expanded photon production format.

2.4.3.4 Absorption

Absorption is treated in one of two ways: analog or implicit. Either way, the incident incoming neutron energy does not include the relative velocity of the target nucleus from the free gas thermal treatment because nonelastic reaction cross sections are assumed to be nearly independent of temperature. That is, only the scattering cross section is affected by the free gas thermal treatment. The terms "absorption" and "capture" are used interchangeably for non-fissile nuclides, both meaning $(\mathrm{n}, 0\mathrm{n})$. For fissile nuclides, "absorption" includes both capture and fission reactions.

2.4.3.4.1 Analog Absorption

In analog absorption, the particle is killed with probability $\sigma_\mathrm{a}/\sigma_\mathrm{t}$, where σ_a and σ_t are the absorption and total cross sections, respectively, of the collision nuclide at the incoming neutron energy. The absorption cross section is specially defined for the MCNP code as the sum of all (n, x) cross sections, where x is anything except neutrons. Thus σ_a is the sum of $\sigma_{\mathrm{n,g}}$, $\sigma_{\mathrm{n,a}}$, $\sigma_{\mathrm{n,d}}$, σ_f, etc. For all particles killed by analog absorption, the entire particle energy and weight are deposited in the collision cell.

2.4.3.4.2 Implicit Absorption

For implicit absorption, also called survival biasing, the neutron weight W_n is reduced to W'_n as

$$W'_\mathrm{n} = \left(1 - \frac{\sigma_\mathrm{a}}{\sigma_\mathrm{t}}\right) W_\mathrm{n}. \tag{2.22}$$

If the new weight W'_n is below the problem weight cutoff (specified on the CUT card), Russian roulette is played, resulting overall in fewer particles with larger weight.

For implicit absorption, a fraction $\sigma_\mathrm{a}/\sigma_\mathrm{t}$ of the incident particle weight and energy is deposited in the collision cell corresponding to that portion of the particle that was absorbed. Implicit absorption is the default method of neutron absorption in the MCNP code.

2.4.3.4.3 Implicit Absorption Along a Flight Path

Implicit absorption also can be done continuously along the flight path of a particle trajectory as is the common practice in astrophysics. In this case, the distance to scatter, rather than the distance to collision, is sampled. The distance to scatter is

$$l = -\frac{1}{\Sigma_\mathrm{s}} \ln(1 - \xi). \tag{2.23}$$

The particle weight at the scattering point is reduced to account for the expected absorption along the flight path,

$$W' = W \exp(-\Sigma_a l), \tag{2.24}$$

where

W'	is the reduced weight after expected absorption along flight path,
W	is the weight at the start of the flight path,
σ_a	is the absorption cross section,
σ_s	is the scattering cross section,
σ_t	is the total cross section ($\sigma_a + \sigma_s$),
l	is the distance to scatter, and
ξ	is a uniformly sampled random number.

Implicit absorption along a flight path is a special form of the exponential transformation coupled with implicit absorption at collisions. See the description of the exponential transform in §5.12.7. The path length is stretched in the direction of the particle, $\mu = 1$, and the stretching parameter is $p = \Sigma_a/\Sigma_t$. Using these values the exponential transform and implicit absorption at collisions yield the identical equations as does implicit absorption along a flight path.

Implicit absorption along a flight path is invoked in the MCNP code as a special option of the exponential transform variance reduction method. It is most useful in highly absorbing media, that is, $\Sigma_a/\Sigma_t \to 1$. When almost every collision results in absorption, it is very inefficient to sample distance to collision. However, implicit absorption along a flight path is discouraged. In highly absorbing media, there is usually a superior set of exponential transform parameters. In relatively non-absorbing media, it is better to sample the distance to collision than the distance to scatter.

2.4.3.5 Elastic and Inelastic Scattering

If the conditions for the $S(\alpha, \beta)$ treatment are not met, the particle undergoes either an elastic or inelastic collision. The selection of an elastic collision is made with the probability

$$\frac{\sigma_{el}}{\sigma_{in} + \sigma_{el}} = \frac{\sigma_{el}}{\sigma_t - \sigma_a}, \tag{2.25}$$

where

σ_{el}	is the elastic scattering cross section.
σ_{in}	is the inelastic cross section, including any neutron-out process such as (n,n'), (n,f), (n,np), etc.
σ_a	is the absorption cross section; $\Sigma_a(n, x)$, where $x \neq n$, that is, all neutron disappearing reactions.
σ_t	is the total cross section, $\sigma_t = \sigma_{el} + \sigma_{in} + \sigma_a$.

Both $\sigma_{\rm el}$ and $\sigma_{\rm t}$ are adjusted for the free gas thermal treatment at thermal energies.

The selection of an inelastic collision is made with the remaining probability,

$$\frac{\sigma_{\rm in}}{\sigma_{\rm t}-\sigma_{\rm a}}$$

If the collision is determined to be inelastic, the type of inelastic reaction, n, is sampled from

$$\sum_{i=1}^{n-1}\sigma_i < \xi\sum_{i=1}^{N}\sigma_i \le \sum_{i=1}^{n}\sigma_i, \tag{2.26}$$

where ξ is a random number on the interval $[0,1)$, N is the number of inelastic reactions, and σ_i is the $i^{\rm th}$ inelastic reaction cross section at the incident neutron energy.

Directions and energies of all outgoing particles from neutron collisions are determined by sampling data from the appropriate cross-section table. Angular distributions are provided and sampled for scattered neutrons resulting from either elastic or discrete-level inelastic events; the scattered neutron energy is then calculated from two-body kinematics. For other reaction types, a variety of data representations is possible. These representations may be divided into two types: (a) the outgoing energy and outgoing angle are sampled independently of each other, or (b) the outgoing energy and outgoing angle are correlated. In the latter case, the outgoing energy may be specified as a function of the sampled outgoing angle, or the outgoing angle may be specified as a function of the sampled outgoing energy. Details of the possible data representations and sampling schemes are provided in the following sections.

2.4.3.5.1 Sampling of Angular and Energy Distributions

The cosine of the angle between incident and exiting particle directions, μ, is sampled from angular distribution tables in the collision nuclide's cross-section library. The cosines are either in the center-of-mass or target-at-rest system, depending on the type of reaction. Data are provided at a number of incident neutron energies. If E is the incident neutron energy, if E_n is the energy of table n, and if E_{n+1} is the energy of table $n+1$, then a value of μ is sampled from table $n+1$ with probability $(E-E_n)/(E_{n+1}-E_n)$ and from table n with probability $(E_{n+1}-E)/(E_{n+1}-E_n)$. There are two options in the MCNP code for representing and sampling a non-isotropic scattering cosine. The first method involves the use of 32 equally probable cosine bins. The second method is to sample a tabulated distribution as a function of μ.

When the method with 32 equiprobable cosine bins is employed, a random number ξ on the interval $[0,1)$ is used to select the $i^{\rm th}$ cosine bin such that $I = 32+1$. The value of μ is then computed as

$$\mu = \mu_i + (32\xi - i)(\mu_{i+1}-\mu_i). \tag{2.27}$$

The method of 32 equiprobable cosine bins accurately represents high-probability regions of the scattering probability; however, it can be a very crude approximation in low-probability regions. For example, it accurately represents the forward scattering in a highly forward-peaked distribution, but may represent all the back angle scattering using only one or a few bins.

A new, more rigorous angular distribution representation was implemented in MCNP4C. This new representation features a tabulation of the probability density function (PDF) as a function of the cosine of the scattering angle. Interpolation of the PDF between cosine values may be either by histogram or linear-linear interpolation. The new tabulated angular distribution allows more accurate representations of original evaluated distributions (typically given as a set of Legendre polynomials) in both high-probability and low-probability regions.

Tabular angular distributions are equivalent to tabular energy distribution (as defined using ENDF Law 4) except that the sampled value is the cosine of the scattering angle, and discrete lines are not allowed. For

each incident neutron energy E_i there is a pointer to a table of cosines $\mu_{i,k}$, probability density functions $p_{i,k}$, and cumulative density functions $c_{i,k}$. The index i and the interpolation fraction r are found on the incident energy grid for the incident energy E_{in} such that

$$E_i < E_{\text{in}} < E_{i+1} \tag{2.28}$$

and

$$E_{\text{in}} = E_i + r(E_{i+1} - E_i). \tag{2.29}$$

A random number, ξ_1, on the unit interval $[0, 1)$ is used to sample a cosine bin k from the cumulative density function

$$c_{l,k} < \xi_1 < c_{l,k+1}, \tag{2.30}$$

where $l = i$ if $\xi_2 > r$ and $l = i + 1$ if $\xi_2 < r$, and ξ_2 is a random number on the unit interval. For histogram interpolation the sampled cosine is

$$\mu' = \mu_{l,k} + \frac{\xi_1 - c_{l,k}}{p_{l.k}}. \tag{2.31}$$

For linear-linear interpolation the sampled cosine is

$$\mu' = \mu_{l,k} + \left\{ \frac{\sqrt{P_{l,k}^2 + 2\left[\frac{p_{l,k+1}-p_{l,k}}{\mu_{l,k+1}-\mu_{l,k}}\right](\xi_1 - c_{l,k})} - p_{l,k}}{\left[\frac{p_{l,k+1}-p_{l,k}}{\mu_{l,k+1}-\mu_{l,k}}\right]} \right\} \tag{2.32}$$

If the emitted angular distribution for some incident neutron energy is isotropic, μ is chosen from $\mu = \xi'$, where ξ' is a random number on the interval $[-1, 1)$. Strictly, in the MCNP code, random numbers are always furnished on the interval $(0, 1)$. Thus, to compute ξ' on $(-1, 1)$ we calculate $\xi' = 2\xi - 1$, where ξ is a random number on $(0, 1)$.

The ENDF-6 format also has various formalisms to describe correlated secondary energy-angle spectra. These are discussed later in this chapter.

For elastic scattering, inelastic level scattering, and some ENDF-6 inelastic reactions, the scattering cosine is chosen in the center-of-mass system. Conversion must then be made to μ_{lab}, the cosine in the target-at-rest system. For other inelastic reactions, the scattering cosine is sampled directly in the target-at-rest system.

The incident particle direction cosines (u_o, v_o, w_o) are rotated to new outgoing target-at-rest system cosines (u, v, w) through a polar angle whose cosine is μ_{lab}, and through an azimuthal angle sampled uniformly. For random numbers ξ_1 and ξ_2 on the interval $[-1, 1)$ with rejection criterion $\xi_1\xi_2 \leq 1$, the rotation scheme is [page 54 of 18]

$$u = u_o\mu_{\text{lab}} + \frac{\sqrt{1-\mu_{\text{lab}}^2}(\xi_1 u_o w_o - \xi_2 v_o)}{\sqrt{(\xi_1^2+\xi_2^2)(1-w_o^2)}}, \tag{2.33a}$$

$$v = v_o\mu_{\text{lab}} + \frac{\sqrt{1-\mu_{\text{lab}}^2}(\xi_1 v_o w_o + \xi_2 u_o)}{\sqrt{(\xi_1^2+\xi_2^2)(1-w_o^2)}}, \tag{2.33b}$$

$$w = w_o\mu_{\text{lab}} - \frac{\xi_1\sqrt{(1-\mu_{\text{lab}}^2)(1-w_o^2)}}{\sqrt{(\xi_1^2+\xi_2^2)}}. \tag{2.33c}$$

If $1 - w_o^2 \sim 0$, then

$$u = u_o\mu_{\text{lab}} + \frac{\sqrt{1-\mu_{\text{lab}}^2}(\xi_1 u_o v_o + \xi_2 w_o)}{\sqrt{(\xi_1^2+\xi_2^2)(1-v_o^2)}}, \tag{2.34a}$$

$$v = v_o\mu_{\text{lab}} - \frac{\xi_1\sqrt{(1-\mu_{\text{lab}}^2)(1-v_o^2)}}{\sqrt{(\xi_1^2+\xi_2^2)}}, \tag{2.34b}$$

$$w = w_o\mu_{\text{lab}} + \frac{\sqrt{1-\mu_{\text{lab}}^2}(\xi_1 w_o v_o - \xi_2 u_o)}{\sqrt{(\xi_1^2+\xi_2^2)(1-v_o^2)}}. \tag{2.34c}$$

If the scattering distribution is isotropic in the target-at-rest system, it is possible to use an even simpler formulation that takes advantage of the exiting direction cosines, (u, v, w), being independent of the incident direction cosines, (u_o, v_o, w_o). In this case,

$$u = 2\xi_1^2 + 2\xi_2^2 - 1, \tag{2.35a}$$

$$v = \xi_1 \sqrt{\frac{1-u^2}{\xi_1^2+\xi_2^2}}, \tag{2.35b}$$

$$w = \xi_2 \sqrt{\frac{1-u^2}{\xi_1^2+\xi_2^2}}, \tag{2.35c}$$

where ξ_1 and ξ_2 are rejected if $\xi_1^2 + \xi_2^2 > 1$.

2.4.3.5.2 Energy from Elastic Scattering

Once the particle direction is sampled from the appropriate angular distribution tables, then the exiting energy, E_{out}, is dictated by two-body kinematics,

$$E_{\text{out}} = \frac{1}{2} E_{\text{in}}[(1-\alpha)\mu_{\text{cm}} + 1 + \alpha] \tag{2.36}$$

$$= E_{\text{in}} \left[\frac{1 + A^2 + 2A\mu_{\text{cm}}}{(1+A)^2} \right], \tag{2.37}$$

where E_{in} is the incident neutron energy, μ_{cm} is the center-of-mass cosine of the angle between incident and exiting particle directions,

$$\alpha = \left(\frac{A-1}{A+1} \right)^2, \tag{2.38}$$

and A is the mass of nuclide nucleus in units of the mass of a neutron (atomic weight ratio).

In the case of point detectors, Equation (2.36) is solved for a fixed cosine scattering angle in the laboratory frame, μ_{lab}.

$$E_{out} = \frac{E_{in}}{(A+1)^2} \left[\mu_{lab} + \sqrt{\mu_{lab}^2 + A^2 - 1} \right]^2 \tag{2.39}$$

If the nuclear data is in the center-of-mass frame, then for point detectors it is necessary to convert

$$p(\mu_{\text{lab}}) = p(\mu_{\text{cm}}) \frac{\mathrm{d}\mu_{\text{cm}}}{\mathrm{d}\mu_{\text{lab}}} \tag{2.40}$$

where

$$\begin{aligned} \mu_{\text{cm}} &= \frac{1}{A} \left[\mu_{\text{lab}} (A+1) \sqrt{\frac{E_{out}}{E_{in}}} - 1 \right] \\ &= \frac{1}{A} \left[\mu_{lab}^2 + \mu_{lab} \sqrt{\mu_{lab}^2 + A^2 - 1} - 1 \right] \end{aligned} \tag{2.41}$$

and

$$\frac{\mathrm{d}\mu_{\text{cm}}}{\mathrm{d}\mu_{\text{lab}}} = \frac{1}{A\sqrt{\mu_{lab}^2 + A^2 - 1}} \left(\mu_{lab} + \sqrt{\mu_{lab}^2 + A^2 - 1} \right)^2 \tag{2.42}$$

In the case of the free gas thermal treatment, the kinematics for the point detector must take into account the velocity of the target nucleus as described in [68].

2.4.3.5.3 Inelastic Reactions

The treatment of inelastic scattering depends upon the particular inelastic reaction chosen. Inelastic reactions are defined as (n, y) reactions such as $(\mathrm{n}, \mathrm{n}')$, $(\mathrm{n}, 2\mathrm{n})$, (n, f), $(\mathrm{n}, \mathrm{n}'\alpha)$ in which y includes at least one neutron.

For many inelastic reactions, such as $(\mathrm{n}, 2\mathrm{n})$, more than one neutron can be emitted for each incident neutron. The weight of each exiting particle is always the same as the weight of the incident particle minus any implicit capture. The energy of exiting particles is governed by various scattering laws that are sampled independently from the cross-section files for each exiting particle. Which law is used is prescribed by the particular cross-section evaluation used. In fact, more than one law can be specified, and the particular one used at a particular time is decided with a random number. In an $(\mathrm{n}, 2\mathrm{n})$ reaction, for example, the first particle emitted may have an energy sampled from one or more laws, but the second particle emitted may have an energy sampled from one or more different laws, depending upon specifications in the nuclear data library. Because emerging energy and scattering angle is sampled independently for each particle, there is no correlation between the emerging particles. Hence energy is not conserved in an individual reaction because, for example, a 14-MeV particle could conceivably produce two 12-MeV particles in a single reaction. The net effect of many particle histories is unbiased because on the average the correct amount of energy is emitted. Results are biased only when quantities that depend upon the correlation between the emerging particles are being estimated.

Users should note that the MCNP code follows a very particular convention. The exiting particle energy and direction are always given in the target-at-rest (laboratory) coordinate system. For the kinematical calculations in the MCNP code to be performed correctly, the angular distributions for elastic, discrete inelastic level scattering, and some ENDF-6 inelastic reactions must be given in the center-of-mass system, and the angular distributions for all other reactions must be given in the target-at-rest system. The MCNP code does not stop if this convention is not adhered to, but the results will be erroneous. In the checking of the cross-section libraries prepared for the MCNP code at Los Alamos, however, careful attention has been paid to ensure that these conventions are followed.

The exiting particle energy and direction in the target-at-rest (laboratory) coordinate system are related to the center-of-mass energy and direction as [17]:

$$E' = E'_{cm} + \left[\frac{E + 2\mu_{\mathrm{cm}}(A+1)\sqrt{EE'_{cm}}}{(A+1)^2} \right] \tag{2.43}$$

and

$$\mu_{\mathrm{lab}} = \mu_{\mathrm{cm}} \sqrt{\frac{E'_{cm}}{E'}} + \frac{1}{A+1}\sqrt{\frac{E}{E'}}, \tag{2.44}$$

where

E'	is the exiting particle energy (laboratory),
E'_{cm}	is the exiting particle energy (center-of-mass),
E	is the incident particle energy (laboratory),
μ_{cm}	is the cosine of center-of-mass scattering angle,
μ_{lab}	is the cosine of laboratory scattering angle, and
A	is the atomic weight ratio (mass of nucleus divided by mass of incident particle).

For point detectors, Equation (2.43) is solved for a fixed μ_{lab} resulting in a quadratic equation with two possible solutions for the exiting particle energy [69],

$$E'_{\pm} = \frac{E}{(A+1)^2}[\mu_{lab} \pm D]^2, \tag{2.45}$$

where

$$D = \sqrt{\mu_{lab} + \frac{E'_{cm}}{E}(A+1)^2 - 1} \tag{2.46}$$

The existence of the roots of the quadratic are subject to the following conditions:

1. If $E'_{cm} \leq 0$, no roots are valid.
2. If $E'_{cm} > E/(A+1)^2$, only the root with the + sign is valid.
3. If $0 < E'_{cm} < E/(A+1)^2$ and $\sqrt{1-(A+1)^2 E'_{cm}/E} < \mu_{lab} < 1$, both roots are valid.

If the nuclear data is in the center-of-mass frame, then for point detectors it is necessary to convert

$$p(\mu_{\text{lab}}) = p(\mu_{\text{cm}})\frac{\mathrm{d}\mu_{\text{cm}}}{\mathrm{d}\mu_{\text{lab}}} \tag{2.47}$$

where

$$\mu_{\text{cm}} = \mu_{\text{lab}}\sqrt{\frac{E'}{E'_{\text{cm}}}} - \frac{1}{A+1}\sqrt{\frac{E}{E'_{\text{cm}}}} \tag{2.48}$$

and

$$\frac{\mathrm{d}\mu_{\text{cm}}}{\mathrm{d}\mu_{\text{lab}}} = \frac{E'(A+1)}{\sqrt{EE'_{\text{cm}}D^2}}. \tag{2.49}$$

2.4.3.5.4 Non-fission Inelastic Scattering and Emission Laws

Non-fission inelastic reactions are handled differently from fission inelastic reactions. For each non-fission reaction N_p particles are emitted, where N_p is an integer quantity specified for each reaction in the cross-section data library of the collision nuclide. The direction of each emitted particle is independently sampled from the appropriate angular distribution table, as was described earlier. The energy of each emitted particle is independently sampled from one of the following scattering or emission laws. Energy and angle are correlated only for ENDF-6 Laws 44 and 67. For completeness and convenience, all the laws are listed together, regardless of whether the law is appropriate for non-fission inelastic scattering (for example, Law 3), fission spectra (for example, Law 11), both (for example, Law 9), or neutron-induced photon production (for example, Law 2). The conversion from center-of-mass to target-at-rest (laboratory) coordinate systems is given in the above equations.

2.4.3.5.4.1 Law 1 (ENDF Law 1): Equiprobable Energy Bins

The index i and the interpolation fraction r are found on the incident energy grid for the incident energy E_{in} such that

$$E_i < E_{\text{in}} < E_{i+1} \tag{2.50}$$

and

$$E_{\text{in}} = E_i + r(E_{i+1} - E_i). \tag{2.51}$$

A random number on the unit interval ξ_1 is used to select an equiprobable energy bin k from the K equiprobable outgoing energies $E_{i,k}$ where

$$k = \xi_i K + 1. \tag{2.52}$$

Then scaled interpolation is used with random numbers ξ_2 and ξ_3 on the unit interval. Let

$$E_1 = E_{i,1} + r(E_{i+1,1} - E_{i,1}) \tag{2.53}$$

and

$$E_K = E_{i,K} + r(E_{i+1,K} - E_{i,K}) \tag{2.54}$$

and

$$l = \begin{cases} i & \xi_3 > r \\ i+1 & \xi_3 < r \end{cases} \tag{2.55}$$

and

$$E' = E_{l,k} + \xi_2(E_{l,k+1} - E_{l.k}) \tag{2.56}$$

then

$$E_{\text{out}} = E_1 + \frac{(E' - E_{l,1})(E_K - E_1)}{E_{l,k} - E_{l,1}}. \tag{2.57}$$

2.4.3.5.4.2 Law 2: Discrete Photon Energy

The value provided in the library is E_g. The secondary photon energy E_{out} is either

$$E_{\text{out}} = E_g \tag{2.58}$$

for non-primary photons or

$$E_{\text{out}} = E_g + [A/(A+1)]E_{\text{in}} \tag{2.59}$$

for primary photons, where A is the atomic weight to neutron weight ratio of the target and E_{in} is the incident neutron energy.

2.4.3.5.4.3 Law 3 (ENDF Law 3): Inelastic Scattering $(\mathbf{n}, \mathbf{n}')$ From Nuclear Levels

The value provided in the library is Q and as a result

$$E_{\text{out}} = \left(\frac{A}{A+1}\right)^2 \left[E_{\text{in}} - \frac{Q(A+1)}{A}\right]. \tag{2.60}$$

2.4.3.5.4.4 Law 4 (ENDF Law 4): Tabular Distribution

For each incident neutron energy E_i there is a pointer to a table of secondary energies $E_{i,k}$, probability density functions $p_{i,k}$, and cumulative density functions $c_{i,k}$. The index i and the interpolation fraction r are found on the incident energy grid for the incident energy E_{in} such that

$$E_i < E_{\text{in}} < E_{i+1} \tag{2.61}$$

and

$$E_{\text{in}} = E_i + r(E_{i+1} - E_i). \tag{2.62}$$

The tabular distribution at each E_i may be composed of discrete lines, a continuous spectra, or a mixture of discrete lines superimposed on a continuous background. If discrete lines are present, there must be the same number of lines (given one per bin) in each table. The sampling of the emission energy for the discrete lines (if present) is handled separately from the sampling for the continuous spectrum (if present). A random number, ξ_1, on the unit interval $[0, 1)$ is used to sample a second energy bin k from the cumulative density function.

If discrete lines are present, the algorithm first checks to see if the sampled bin is within the discrete line portion of the table as determined by

$$c_{i,k} + r(c_{i+1,k} - c_{i,k}) < \xi_1 < c_{i,k+1} + r(c_{i+1,k+1} - c_{i,k+1}).$$

If this condition is met, then the sampled energy E' for the discrete line is interpolated between incident energy grids as

$$E' = E_{i,k} + r(E_{i+1,k} - E_{i,k}). \tag{2.63}$$

If a discrete line has been sampled, the energy sampling is finished. If a discrete line has not been sampled, the secondary energy is sampled from the remaining continuous background.

For continuous distributions, the secondary energy bin k is sampled from

$$c_{l.k} < \xi_1 < c_{l.k+1}, \tag{2.64}$$

where $l = i$ if $\xi_2 > r$ and $l = i + 1$ if $\xi_2 < r$, and ξ_2 is a random number on the unit interval. For histogram interpolation the sampled energy is

$$E' = E_{l,k} + \frac{\xi_1 - c_{l.k}}{p_{l,k}}. \tag{2.65}$$

For linear-linear interpolation the sampled energy is

$$E' = E_{l,k} + \left\{ \frac{\sqrt{P_{l,k}^2 + 2\left[\frac{p_{l,k+1} - p_{l,k}}{E_{l,k+1} - E_{l,k}}\right](\xi_1 - c_{l,k})} - p_{l,k}}{\left[\frac{p_{l,k+1} - p_{l,k}}{E_{l,k+1} - E_{l,k}}\right]} \right\}. \tag{2.66}$$

The secondary energy is then interpolated between the incident energy bins i and $i + 1$ to properly preserve thresholds. Let

$$E_1 = E_{i,1} + r(E_{i+1,1} - E_{i,1}) \tag{2.67}$$

and

$$E_K = E_{i,K} + r(E_{i+1,K} - E_{i,K}) \tag{2.68}$$

then

$$E_{\text{out}} = E_1 + \frac{(E' - E_{l,1})(E_K - E_1)}{(E_{l,K} - E_{l,1})}. \tag{2.69}$$

The final step is to adjust the energy from the center-of-mass system to the laboratory system, if the energies were given in the center-of-mass system.

Law 4 is an independent distribution, i.e. the emission energy and angle are not correlated. The outgoing angle is selected from the angular distribution as described in §2.4.3.5.1. Data tables built using this methodology are designed to sample the distribution correctly in a statistical sense and will not necessarily sample the exact distribution for any specific collision.

2.4.3.5.4.5 Law 5 (ENDF Law 5): General Evaporation Spectrum

The function $g(x)$ is tabulated versus χ and the energy is tabulated versus incident energy E_{in}. The law is then

$$f(E_{\text{in}} \to E_{\text{out}}) = g\left(\frac{E_{\text{out}}}{T(E_{\text{in}})}\right). \tag{2.70}$$

This density function is sampled by $E_{\text{out}} = \chi(\xi)T(E_{\text{in}})$, where $T(E_{\text{in}})$ is a tabulated function of the incident energy and $c(\xi)$ is a table of equiprobable χ values.

2.4.3.5.4.6 Law 7 (ENDF Law 7): Simple Maxwell Fission Spectrum

The law is

$$f(E_{\text{in}} \to E_{\text{out}}) = C\sqrt{E_{\text{out}}}\exp\left(-\frac{E_{\text{out}}}{T(E_{\text{in}})}\right). \tag{2.71}$$

The nuclear temperature $T(E_{\text{in}})$ is a tabulated function of the incident energy. The normalization constant C is given by

$$C^{-1} = T^{3/2}\left[\left(\frac{\sqrt{\pi}}{2}\right)\text{erf}\left(\sqrt{\frac{E_{\text{in}} - U}{T}}\right) - \sqrt{\frac{E_{\text{in}} - U}{T}}\exp\left(-\frac{E_{\text{in}} - U}{T}\right)\right], \tag{2.72}$$

where U is a constant provided in the library and limits E_{out} to $0 \le E_{\text{out}} \le E - U$. In the MCNP code this density function is sampled by the rejection scheme

$$E_{\text{out}} = -T(E_{\text{in}})\left[\frac{\xi_1^2\ln(\xi_3)}{\xi_1^2 + \xi_2^2} + \ln(\xi_4)\right], \tag{2.73}$$

where ξ_1, ξ_2, ξ_3, and ξ_4 are random numbers on the unit interval. ξ_1 and ξ_2 are rejected if $\xi_1^2 + \xi_2^2 > 1$.

2.4.3.5.4.7 Law 9 (ENDF Law 9): Evaporation Spectrum

The law is

$$f(E_{\text{in}} \to E_{\text{out}}) = CE_{\text{out}} \exp\left(-\frac{E_{\text{out}}}{T(E_{\text{in}})}\right), \tag{2.74}$$

where the nuclear temperature $T(E_{\text{in}})$ is a tabulated function of the incident energy. The energy U is provided in the library and is assigned so that E_{out} is limited by $0 \le E_{\text{out}} \le E_{\text{in}} - U$. The normalization constant C is given by

$$C^{-1} = T^2\left[1 - \exp\left(-\frac{E_{\text{in}} - U}{T}\right)\left(1 + \frac{E_{\text{in}} - U}{T}\right)\right]. \tag{2.75}$$

In the MCNP code this density function is sampled by

$$E_{\text{out}} = -T(E_{\text{in}}) \ln(\xi_1 \xi_2), \tag{2.76}$$

where ξ_1 and ξ_2 are random numbers on the unit interval, and ξ_1 and ξ_2 are rejected if $E_{\text{out}} > E_{\text{in}} - U$.

2.4.3.5.4.8 Law 11 (ENDF Law 11): Energy Dependent Watt Spectrum

The law is

$$f(E_{\text{in}} \to E_{\text{out}}) = C \exp\left(-\frac{E_{\text{out}}}{a(E_{\text{in}})}\right) \sinh\left(\sqrt{b(E_{\text{in}})E_{\text{out}}}\right). \tag{2.77}$$

The constants a and b are tabulated functions of incident energy and U is a constant from the library. The normalization constant C is given by

$$C^{-1} = \frac{1}{2}\sqrt{\frac{\pi a^3 b}{4}} \exp\left(\frac{ab}{4}\right)\left[\text{erf}\left(\sqrt{\frac{E_{\text{in}} - U}{a}} - \sqrt{\frac{ab}{4}}\right) + \text{erf}\left(\sqrt{\frac{E_{\text{in}} - U}{a}} + \sqrt{\frac{ab}{4}}\right)\right] - a \exp\left(-\frac{E_{\text{in}} - U}{a}\right) \sinh\left(\sqrt{b(E_{\text{in}} - U)}\right), \tag{2.78}$$

where the constant U limits the range of outgoing energy so that $0 \le E_{\text{out}} \le E_{\text{in}} - U$. This density function is sampled as follows. Let

$$g = \sqrt{\left(1 + \frac{ab}{8}\right)^2 - 1} + \left(1 + \frac{ab}{8}\right). \tag{2.79}$$

Then $E_{\text{out}} = -ag \ln(\xi_1)$. E_{out} is rejected if

$$[(1 - g)(1 - \ln(\xi_1)) - \ln(\xi_2)]^2 > bE_{\text{out}}, \tag{2.80}$$

where ξ_1 and ξ_2 are random numbers on the unit interval.

2.4.3.5.4.9 Law 22 (UK Law 2): Tabular Linear Functions of Incident Energy Out

Tables of $P_{i,j}$, $C_{i,j}$, and $T_{i,j}$ are given at a number of incident energies E_i. If $E_i \le E_{\text{in}} < E_{i+1}$ then the i^{th} $P_{i,j}$, $C_{i,j}$, and $T_{i,j}$ tables are used and

$$E_{\text{out}} = C_{i,k}(E_{\text{in}} - T_{i,k}), \tag{2.81}$$

where k is chosen according to

$$\sum_{j=1}^{k} P_{i,j} < \xi \le \sum_{j=1}^{k+1} P_{i,j},$$

where ξ is a random number on the unit interval $[0, 1)$.

2.4.3.5.4.10 Law 24 (UK Law 6): Equiprobable Energy Multipliers

The law is

$$E_{\text{out}} = E_{\text{in}} T(E_{\text{in}}). \tag{2.82}$$

The library provides a table of K equiprobable energy multipliers $T_{i,k}$ for a grid of incident neutron energies E_i. For incident energy E_{in} such that

$$E_i < E_{\text{in}} < E_{i+1}.$$

The random numbers ξ_1 and ξ_2 on the unit interval are used to find T with

$$k = \xi_1 K + 1 \tag{2.83}$$

and

$$T = T_{i,k} + \xi_2 (T_{i,k+1} - T_{i,k}) \tag{2.84}$$

so

$$E_{\text{out}} = E_{\text{in}} T \tag{2.85}$$

2.4.3.5.4.11 Law 44 Tabular Distribution (ENDF Law=1 Lang=2): Kalbach-87 Correlated Energy-angle Scattering)

Law 44 is an extension of Law 4. For each incident energy E_i there is a pointer to a table of secondary energies $E_{i,k}$, probability density functions $p_{i,k}$, cumulative density functions $c_{i,k}$, pre-compound fractions $R_{i,k}$, and angular distribution slope values $A_{i,k}$. The secondary emission energy is found exactly as stated in the Law 4 description in §2.4.3.5.4.4. Unlike Law 4, Law 44 includes a correlated angular distribution associated with each incident energy E_i as given by the Kalbach parameters $R_{i,k}$, and $A_{i,k}$. Thus, the sampled emission angle is dependent on the sampled emission energy.

The sampled values for R and A are interpolated on both the incident and outgoing energy grids. For discrete spectra,

$$A = A_{i,k} + r(A_{i+1,k} - A_{i,k}) \tag{2.86}$$

and

$$R = R_{i,k} + r(R_{i+1,k} - R_{i,k}). \tag{2.87}$$

For continuous spectra with histogram interpolation,

$$A = A_{i,k} \tag{2.88}$$

and

$$R = R_{i,k} \tag{2.89}$$

For continuous spectra with linear-linear interpolation,

$$A = A_{l,k} + (A_{l,k+1} - A_{l,k}) \frac{E' - E_{l,k}}{E_{l,k+1} - E_{l,k}} \tag{2.90}$$

and

$$R = R_{l,k} + (R_{l,k+1} - R_{l,k}) \frac{E' - E_{l,k}}{E_{l,k+1} - E_{l,k}}. \tag{2.91}$$

The outgoing neutron center-of-mass scattering angle μ is sampled from the Kalbach density function

$$p(\mu, E_{\text{in}}, E_{\text{out}}) = \frac{1}{2}\frac{A}{\sinh(A)}[\cosh(A\mu) + R\sinh(A\mu)] \tag{2.92}$$

using the random numbers ξ_3 and ξ_4 on the unit interval as follows. If $\xi_3 > R$, then let

$$T = (2\xi_4 - 1)\sinh(A) \tag{2.93}$$

and

$$\mu = \frac{\ln\left(T + \sqrt{T^2+1}\right)}{A}, \tag{2.94}$$

or if $\xi_3 < R$, then

$$\mu = \frac{\ln[\xi_4 \exp(A) + (1-\xi_4)\exp(-A)]}{A}. \tag{2.95}$$

As with Law 4, the emission energy and angle are transformed from the center-of-mass to the laboratory system as necessary.

2.4.3.5.4.12 Law 61 Tabular Distribution (ENDF Law=1 Lang=0, 12, or 14): Correlated Energy-angle Scattering

Law 61 is an extension of Law 4. For each incident energy E_i there is a pointer to a table of secondary energies $E_{i,k}$, probability density functions $p_{i,k}$, cumulative density functions $c_{i,k}$, and pointers to tabulated angular distributions $L_{i,k}$. The secondary emission energy is found exactly as stated in the Law 4 description in §2.4.3.5.4.4. Unlike Law 4, Law 61 includes a correlated angular distribution associated with each incident energy E_i as given by the tabular angular distribution located using the pointers $L_{i,k}$. Thus, the sampled emission angle is dependent on the sampled emission energy.

If the secondary distribution is given using histogram interpolation, the angular distribution located at $L_{i,k}$ is used to sample the emission angle. If the secondary distribution is specified as linear interpolation between energy points, $L_{i,k}$ is chosen by selecting the bin closest to the randomly sampled cumulative distribution function (CDF) point. If the value of $L_{i,k}$ is zero, the angle is sampled from an isotropic distribution as described on page 81. If the value of $L_{i,k}$ is positive, it points to a tabular angular distribution which is then sampled as described on page 81.

As with Law 4, the emission energy and angle are transformed from the center-of-mass to the laboratory system as necessary.

2.4.3.5.4.13 Law 66 (ENDF Law 6): N-body Phase Space Distribution

The phase space distribution for particle i in the center-of-mass coordinate system is:

$$P_i(\mu, E_{\text{in}}, T) = C_n\sqrt{T}(E_i^{\max} - T)^{3n/2-4}, \tag{2.96}$$

where all energies and angles are also in the center-of-mass system and $E_i^{\max}$ is the maximum possible energy for particle i, μ, and T. T is used for calculating E_{out}. The C_n normalization constants for $n = 3, 4, 5$ are:

$$C_3 = \frac{4}{\pi(E_i^{\max})^2}, \tag{2.97a}$$

$$C_4 = \frac{105}{32(E_i^{\max})^{7/2}}, \tag{2.97b}$$

$$C_5 = \frac{256}{14\pi(E_i^{\max})^5}. \tag{2.97c}$$

$E_i^{\max}$ is a fraction of the energy available, E_{avail}, given as

$$E_i^{\max} = \frac{M - m_i}{M} E_{\text{avail}}, \tag{2.98}$$

where M is the total mass of the n particles being treated, m_i is the mass of particle i, and

$$E_{\text{avail}} = \frac{m_{\text{t}}}{m_{\text{p}} + m_{\text{t}}} E_{\text{in}} + Q, \tag{2.99}$$

where m_{t} is the target mass and m_{p} is the projectile mass. For neutrons,

$$\frac{m_{\text{t}}}{m_{\text{p}} + m_{\text{t}}} = \frac{A}{A+1} \tag{2.100}$$

and for a total mass ratio $A_{\text{p}} = M/m_i$,

$$\frac{M - m_i}{M} = \frac{A_{\text{p}} - 1}{A_{\text{p}}}. \tag{2.101}$$

Thus,

$$E_i^{\max} = \frac{A_{\text{p}} - 1}{A_{\text{p}}} \left(\frac{A}{A+1} E_{\text{in}} + Q \right). \tag{2.102}$$

The total mass A_{p} and the number of particles in the reaction n are provided in the data library. The outgoing energy is sampled as follows.

Let ξ_i, $i = 1, 10$ be random numbers on the unit interval. Then from rejection technique R28 from the Monte Carlo Sampler [70], accept ξ_1 and ξ_2 if $\xi_1^2 + \xi_2^2 \leq 1$ and accept ξ_3 and ξ_4 if $\xi_3^2 + \xi_4^2 \leq 1$.

Then let

$$x = \frac{-\xi_1 \ln\left(\xi_1^2 + \xi_2^2\right)}{\xi_1^2 + \xi_2^2} - \ln(\xi_9) \tag{2.103}$$

$$y = \begin{cases} \frac{-\xi_3 \ln\left(\xi_3^2+\xi_4^2\right)}{\xi_3^2+\xi_4^2} - \ln(\xi_5) & n = 3 \\ -\ln(\xi_5 \xi_6 \xi_7) & n = 4 \\ \frac{-\xi_3 \ln\left(\xi_3^2+\xi_4^2\right)}{\xi_3^2+\xi_4^2} - \ln(\xi_5 \xi_6 \xi_7 \xi_8) & n = 5 \end{cases} \tag{2.104}$$

and

$$T = \frac{x}{x + y}. \tag{2.105}$$

Then

$$E_{\text{out}} = T E_i^{\max} \tag{2.106}$$

The cosine of the scattering angle is always sampled isotropically in the center-of-mass system using another random number ξ_{10} on the unit interval as

$$\mu = 2\xi_{10} - 1. \tag{2.107}$$

2.4.3.5.4.14 Law 67 (ENDF Law 7): Correlated Energy-angle Scattering

For each incident neutron energy, first the exiting particle direction μ is sampled as described in §2.4.3.5.1. In other law data, first the exiting particle energy is sampled and then the angle is sampled. The index i and the interpolation fraction r are found on the incident energy grid for the incident energy E_{in} such that

$$E_i < E_{\text{in}} < E_{i+1}$$

and

$$E_{\text{in}} = E_i + r(E_{i+1} - E_i). \tag{2.108}$$

For each incident energy E_i there is a table of exiting particle direction cosines $\mu_{i,j}$ and locators $L_{i,j}$. This table is searched to find which ones bracket μ, namely,

$$\mu_{i,j} < \mu < \mu_{i,j+1}. \tag{2.109}$$

Then the secondary energy tables at $L_{i,j}$ and $L_{i,j+1}$ are sampled for the outgoing particle energy. The secondary energy tables consist of a secondary energy grid $E_{i,j,k}$, probability density functions $p_{i,j,k}$, and cumulative density functions $c_{i,j,k}$. A random number ξ_1 on the unit interval is used to pick between incident energy indices: if $\xi_1 < r$ then $l = i + 1$; otherwise, $l = i$. Two more random numbers ξ_2 and ξ_3 on the unit interval are used to determine interpolation energies. As such,

$$E_{i,k} = \begin{cases} E_{i,j+1,k}\,,\ m = j+1 & \xi_2 < \frac{\mu-\mu_{1,j}}{\mu_{1,j+1}-\mu_{i,j}}\,,\ l = i \\ E_{i,j,k}\,,\ m = j & \xi_2 \geq \frac{\mu-\mu_{1,j}}{\mu_{1,j+1}-\mu_{i,j}}\,,\ l = i \end{cases}. \tag{2.110}$$

Similarly,

$$E_{i+1,k} = \begin{cases} E_{i+1,j+1,k}\,,\ m = j+1 & \xi_3 < \frac{\mu-\mu_{i+1,j}}{\mu_{i+1,j+1}-\mu_{i+1,j}}\,,\ l = i+1 \\ E_{i+1,j,k}\,,\ m = j & \xi_3 \geq \frac{\mu-\mu_{i+1,j}}{\mu_{i+1,j+1}-\mu_{i+1,j}}\,,\ l = i+1 \end{cases}. \tag{2.111}$$

A random number ξ_4 on the unit interval is used to sample a secondary energy bin k from the cumulative density function

$$c_{l,m,k} < \xi_4 < c_{l,m,k+1}. \tag{2.112}$$

For histogram interpolation the sampled energy is

$$E' = E_{l,m,k} + \frac{\xi_4 - c_{l,m,k}}{p_{l,m,k}}. \tag{2.113}$$

For linear-linear interpolation the sampled energy is

$$E' = E_{l,m,k} + \left\{ \frac{\sqrt{P_{l,m,k}^2 + 2\left[\frac{p_{l,m,k+1}-p_{l,m,k}}{E_{l,m,k+1}-E_{l,m,k}}\right](\xi_4 - c_{l,m,k})} - p_{l,m,k}}{\left[\frac{p_{l,m,k+1}-p_{l,m,k}}{E_{l,m,k+1}-E_{l,m,k}}\right]} \right\}. \tag{2.114}$$

The final outgoing energy E_{out} uses scaled interpolation. Let

$$E_1 = E_{i,1} + r(E_{i+1,1} - E_{i,1}) \tag{2.115}$$

and

$$E_K = E_{i,K} + r(E_{i+1,K} - E_{i,K}) \tag{2.116}$$

then

$$E_{\text{out}} = E_1 + \frac{(E' - E_{l,1})(E_K - E_1)}{E_{l,K} - E_{l,1}}. \tag{2.117}$$

2.4.3.5.5 Emission from Fission

For any fission reaction a number of neutrons, n, is emitted according to the value of $\overline{\nu}(E_{\text{in}})$. Depending on the type of problem (fixed source or KCODE) and on user input (TOTNU card), the MCNP code may use either

prompt $\overline{\nu}_{\mathrm{p}}(E_{\mathrm{in}})$ or total $\overline{\nu}_{\mathrm{t}}(E_{\mathrm{in}})$. For either case, the average number of neutrons per fission, $\overline{\nu}(E_{\mathrm{in}})$, may be a tabulated function of energy or a polynomial function of energy.

If DATA entry on the FMULT card is zero (default), then n is sampled between I (the largest integer less than $\overline{\nu}(E_{\mathrm{in}})$) and $I+1$ by

$$n = \begin{cases} I+1 & \xi \leq \overline{\nu}(E_{\mathrm{in}}) - 1 \\ I & \xi > \overline{\nu}(E_{\mathrm{in}}) - 1 \end{cases}, \tag{2.118}$$

where ξ is a random number drawn from the unit interval.

If more microscopically correct fission neutron multiplicities are desired for fixed source problems, the DATA entry on the FMULT card can be used to select which set of Gaussian widths are used to sample the actual number of neutrons from fission that typically range from 0 to 7 or 8 [71]. For a given fission event, there is a probability P_n that n neutrons are emitted. This distribution is generally called the neutron multiplicity distribution. Fission neutron multiplicity distributions are known to be well reproduced by simple Gaussian distributions [72],

$$P_0 = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{1/2} \exp\left(-\frac{(x-\overline{\nu}+b)^2}{2\sigma^2}\right) \mathrm{d}x, \tag{2.119}$$

and

$$P_{n\neq 0} = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{n-1/2}^{n+1/2} \exp\left(-\frac{(x-\overline{\nu}+b)^2}{2\sigma^2}\right) \mathrm{d}x, \tag{2.120}$$

where $\overline{\nu}$ is the mean multiplicity, b is a small adjustment to make the mean equal to $\overline{\nu}$, and σ is the Gaussian width. For the range of realistic widths, the adjustment b can be accurately expressed as a single smooth function of $(\overline{\nu}+0.5)/\sigma$ [73]. To determine the value of σ from experimental data, many authors have minimized the chi-squared

$$\chi^2(\sigma) = \sum_n \left[\frac{P_n^{\mathrm{exp}} - P_n(\sigma)}{\Delta P_n^{\mathrm{exp}}}\right]^2, \tag{2.121}$$

where $\Delta P_n^{\mathrm{exp}}$ is the uncertainty in the experimentally measured multiplicity distribution P_n^{exp}. The factorial moments of the neutron multiplicity distribution ($\overline{\nu}_i = \sum P_n n!/(n-i)!$) emitted by a multiplying sample can be expressed as a function of the factorial moments for spontaneous and induced fission [74]. Therefore, for many applications it is not necessary to know the details of the neutron multiplicity distribution, but it is more important to know the corresponding first three factorial moments. A reevaluation of the existing spontaneous fission and neutron induced fission data has been performed [73] where the widths of Gaussians are adjusted to fit the measured second and third factorial moments. This reevaluation was done by minimizing the chi-squared

$$\chi^2(\sigma) = \sum_{i=2}^{3} \left[\frac{\nu_i(P_n^{\mathrm{exp}}) - \nu_i(P_n(\sigma))}{\Delta\nu_i^{\mathrm{exp}}}\right]^2. \tag{2.122}$$

These results are summarized in Table 2.1. Despite the change in emphasis from the detailed shape to the moments of the distributions, the inferred widths are little changed from those obtained by others. However, by minimizing the chi-squared in Eq. (2.122) the inferred widths are guaranteed to be in reasonable agreement with the measured second and third factorial moments. The widths obtained using Eq. (2.122) give Gaussian distributions that reproduce the experimental second and third factorial moments to better than 0.6%. The adjustment parameter b ensures that the first moment ($\overline{\nu}$) is accurately reproduced. If the chi-squared in Eq. (2.121) is used, then the second and third factorial moments can differ from the experimental values by as much as 10%.

Assuming that the widths of the multiplicity distributions are independent of the initial excitation energy of the fissioning system [73], the relationship between different factorial moments is easily calculated as a function of $\overline{\nu}$. The corresponding calculated relationships between the first three factorial moments are in

Table 2.1: Recommended Gaussian Widths [73] from Eq. (2.122)

Reaction	σ
^{233}U(n,f)	1.070
^{235}U(n,f)	1.088
^{238}U(n,f)	1.116
^{239}Pu(n,f)	1.140
^{241}Pu(n,f)	1.150
^{238}Pu SF*	1.135
^{240}Pu SF	1.151
^{242}Pu SF	1.161
^{242}Cm SF	1.091
^{244}Cm SF	1.103
^{246}Cm SF	1.098
^{248}Cm SF	1.108
^{250}Cf SF	1.220
^{252}Cf SF	1.245
^{254}Cf SF	1.215
^{254}Fm SF	1.246

*SF: Spontaneous Fission

good agreement with experimental neutron induced fission data up to an incoming neutron energy of 10 MeV [73]. This implies that an energy independent width can be used with confidence up to an incoming neutron energy of at least 10 MeV. The Gaussian widths in Table 2.1 are used for fission multiplicity sampling in the MCNP code when the `DATA` entry on the `FMULT` card is 1. Induced fission multiplicities for isotopes not listed in Table 2.1 use a Gaussian width that is linearly dependent on the mass number of the fissioning system [73].

The direction of each emitted neutron is sampled independently from the appropriate angular distribution table by the procedure described in §2.4.3.5.1.

The energy of each fission neutron is determined from the appropriate emission law. These laws are discussed in the preceding section. The MCNP code then models the transport of the first neutron out after storing all other neutrons in the bank.

2.4.3.5.6 Prompt and Delayed Neutron Emission

If (1) the MCNP code is using $\overline{\nu}_t$, (2) the data for the collision isotope includes delayed-neutron spectra, and (3) the use of detailed delayed-neutron data has not been preempted (on the `PHYS`:`n` card), then each fission neutron is first determined by the MCNP code to be either a prompt fission neutron or a delayed fission neutron. Assuming analog sampling, the type of emitted neutron is determined from the ratio of delayed $\overline{\nu}(E_{in})$ to total $\overline{\nu}_t(E_{in})$ where a delayed neutron is produced if

$$\xi \leq \frac{\overline{\nu}_d(E_{in})}{\overline{\nu}_t(E_{in})} \tag{2.123}$$

and a prompt neutron is produced if

$$\xi > \frac{\overline{\nu}_d(E_{in})}{\overline{\nu}_t(E_{in})}, \tag{2.124}$$

where $\overline{\nu}_d$ is the expected number of delayed neutrons.

If the neutron is determined to be a prompt fission neutron, it is emitted instantaneously, and the emission laws (angle and energy) specified for prompt fission are sampled.

If the neutron is determined to be a delayed fission neutron, then the MCNP code first samples for the decay group by using the specified abundances. Then, the time delay is sampled from the exponential density with decay constant specified for the sampled decay group.

Finally, the emission laws (angle and energy) specified for that decay group are then sampled. Since the functionality in the MCNP code to produce delayed neutrons using appropriate emission data is new, we include next a somewhat more expanded description.

A small but important fraction ($\approx 1\%$) of the neutrons emitted in fission events are delayed neutrons emitted as a result of fission-product decay at times later than prompt fission neutrons. the MCNP code users have always been able to specify whether or not to include delayed fission neutrons by using either $\overline{\nu}_{\mathrm{t}}$ (prompt plus delayed) or $\overline{\nu}_{\mathrm{p}}$ (prompt only). However, in versions of the MCNP code up through and including 4B, all fission neutrons (whether prompt or delayed) were produced instantaneously and with an energy sampled from the spectra specified for prompt fission neutrons.

For many applications this approach is adequate. However, it is another example of a data approximation that is unnecessary. Therefore, Versions 4C and later of the MCNP code allow delayed fission neutrons to be sampled (either analog or biased) from time and energy spectra as specified in nuclear data evaluations. The libraries with detailed delayed fission neutron data are listed in [46] with a "yes" in the "DN" column.

The explicit sampling of a delayed-neutron spectrum implemented in MCNP4C has two effects. One is that the delayed neutron spectra have the correct energy distribution; they tend to be softer than the prompt spectra. The second is that experiments measuring neutron decay after a pulsed source can now be modeled with the MCNP code because the delay in neutron emission following fission is properly accounted for. In this treatment, a natural sampling of prompt and delayed neutrons is implemented as the default and an additional delayed neutron biasing control is available to the user via the PHYS:n card. The biasing allows the number of delayed neutrons produced to be increased artificially because of the low probability of a delayed neutron occurrence. The delayed neutron treatment is intended to be used with the TOTNU card in the MCNP code, giving the user the flexibility to use the time-dependent treatment of delayed neutrons whenever the delayed data are available.

The impact of sampling delayed-neutron energy spectra on reactivity calculations has been studied [75]. As expected, most of the reactivity impacts are very small, although changes of 0.1-0.2% in k_{eff} were observed for certain cases. Overall, inclusion of delayed-neutron spectra can be expected to produce small positive reactivity changes for systems with significant fast neutron leakage and small negative changes for some systems in which a significant fraction of the fissions occurs in isotopes with an effective fission threshold (e.g., ^{238}U and ^{240}Pu).

2.4.3.6 The $S(\alpha, \beta)$ Treatment

The $S(\alpha, \beta)$ thermal scattering treatment is a complete representation of thermal neutron scattering by molecules and crystalline solids. Two processes are allowed: (1) inelastic scattering with cross section σ_{in} and a coupled energy-angle representation derived from an ENDF $S(\alpha, \beta)$ scattering law, and (2) elastic scattering with no change in the outgoing neutron energy for solids with cross section σ_{el} and an angular treatment derived from lattice parameters. The elastic scattering treatment is chosen with probability $\sigma_{\mathrm{el}}/(\sigma_{\mathrm{el}} + \sigma_{\mathrm{in}})$. This thermal scattering treatment also allows the representation of scattering by multi-atomic molecules (for example, BeO).

For the inelastic treatment, the distribution of secondary energies is represented by a set of equally probable final energies (typically 16 or 32) for each member of a grid of initial energies from an upper limit of typically 4 eV down to 10^{-5} eV, along with a set of angular data for each initial and final energy. The selection of a final energy E' given an initial energy E can be characterized by sampling from the distribution

$$p(E'|E_i < \xi < E_{i+1}) = \frac{1}{N} \sum_{j=1}^{N} \delta[E' - \rho E_{i,j} - (1-\rho) E_{i+1,j}], \tag{2.125}$$

where E_i and E_{i+1} are adjacent elements on the initial energy grid,

$$\rho = \frac{E_{i+1} - E}{E_{i+1} - E_i}, \tag{2.126}$$

$N$is the number of equally probable final energies, and $E_{i,j}$ is the j^{th} discrete final energy for incident energy E_i.

There are two allowed schemes for the selection of a scattering cosine following selection of a final energy and final energy index j. In each case, the $(i,j)^{\text{th}}$ set of angular data is associated with the energy transition $E = E_i \rightarrow E' = E_{i,j}$.

In the first scheme, the data consist of sets of equally probable discrete cosines $\mu_{i,j,k}$ for $k = 1, \ldots, \nu$ with ν typically 4 or 8. An index k is selected with probability $1/\nu$, and μ is obtained by the relation

$$\mu = \rho\mu_{i,j,k} + (1 - \rho)\mu_{i+1,j,k}. \tag{2.127}$$

In the second scheme, the data consist of bin boundaries of equally probable cosine bins. In this case, random linear interpolation is used to select one set or the other, with ρ being the probability of selecting the set corresponding to incident energy E_i. The subsequent procedure consists of sampling for one of the equally probable bins and then choosing μ uniformly in the bin.

For elastic scattering, the above two angular representations are allowed for data derived by an incoherent approximation. In this case, one set of angular data appears for each incident energy and is used with the interpolation procedures on incident energy described above. For elastic scattering, when the data have been derived in the coherent approximation, a completely different representation occurs. In this case, the data actually stored are the set of parameters D_k, where

$$\sigma_{eI} = \begin{cases} D_k/E & E_{bk} \le E < E_{bk+1} \\ 0 & E < E_{B1} \end{cases} \tag{2.128}$$

and E_{Bk} are Bragg energies derived from the lattice parameters. For incident energy E such that $E_{Bk} \le E \le E_{Bk+1}$,

$$P_i = D_i/D_k\,,\ i = 1, \ldots, k \tag{2.129}$$

represents a discrete cumulative probability distribution that is sampled to obtain index i, representing scattering from the i^{th} Bragg edge. The scattering cosine is then obtained from the relationship

$$\mu = 1 - 2E_{Bi}/E. \tag{2.130}$$

Using next-event estimators such as point detectors with $S(\alpha, \beta)$, scattering cannot be done exactly because of the discrete scattering angles. The MCNP code uses an approximate scheme [76, 77] that in the next-event estimation calculation replaces discrete lines with histograms of width $\delta\mu < 0.1$.

See also §2.5.6.4.7.

2.4.3.7 Probability Tables for the Unresolved Resonance Range

Within the unresolved resonance range (e.g., in ENDF/B-VI, 2.25–25 keV for ^{235}U, 10–149.03 keV for ^{238}U, and 2.5–30 keV for ^{239}Pu), continuous-energy neutron cross sections appear to be smooth functions of energy. This behavior occurs not because of the absence of resonances, but rather because the resonances are so close together that they are unresolved. Furthermore, the smoothly varying cross sections do not account

for resonance self-shielding effects, which may be significant for systems whose spectra peak in or near the unresolved resonance range.

Fortunately, the resonance self-shielding effects can be represented accurately in terms of probabilities based on a stratified sampling technique. This technique produces tables of probabilities for the cross sections in the unresolved resonance range. Sampling the cross section in a random walk from these probability tables is a valid physics approximation so long as the average energy loss in a single collision is much greater than the average width of a resonance; that is, if the narrow resonance approximation [78] is valid. Then the detail in the resonance structure following a collision is statistically independent of the magnitude of the cross sections prior to the collision.

The utilization of probability tables is not a new idea in Monte Carlo applications. A code [79] to calculate such tables for Monte Carlo fast reactor applications was utilized in the early 1970s. Temperature-difference Monte Carlo calculations [80] and a summary of the VIM Monte Carlo code [81] that uses probability tables are pertinent early examples. Versions of the MCNP code up through and including 4B did not take full advantage of the unresolved resonance data provided by evaluators. Instead, smoothly varying average cross sections were used in the unresolved range. As a result, any neutron self-shielding effects in this energy range were unaccounted for. Better utilizations of unresolved data have been known and demonstrated for some time, and the probability table treatment has been incorporated [82] into MCNP4C and its successors. The column "UR" in [46] lists whether unresolved resonance probability table data is available for each nuclide library.

Sampling cross sections from probability tables is straightforward. At each of a number of incident energies there is a table of cumulative probabilities (typically 20) and the value of the near-total, elastic, fission, and radiative capture cross sections and heat deposition numbers corresponding to those probabilities. These data supplement the usual continuous data; if probability tables are turned off (`PHYS`:`n` card), then the usual smooth cross section is used. But if the probability tables are turned on (default), if they exist for the nuclide of a collision, and if the energy of the collision is in the unresolved resonance energy range of the probability tables, then the cross sections are sampled from the tables. The near-total is the total of the elastic, fission, and radiative capture cross sections; it is not the total cross section, which may include other absorption or inelastic scatter in addition to the near-total. The radiative capture cross section is not the same as the usual MCNP capture cross section, which is more properly called "destruction" or absorption and includes not only radiative capture but all other reactions not emitting a neutron. Sometimes the probability tables are provided as factors (multipliers of the average or underlying smooth cross section) which adds computational complexity but now includes any structure in the underlying smooth cross section.

It is essential to maintain correlations in the random walk when using probability tables to properly model resonance self-shielding. Suppose we sample the 17^{th} level (probability) from the table for a given collision. This position in the probability table must be maintained for the neutron trajectory until the next collision, regardless of particle splitting for variance reduction or surface crossings into various other materials whose nuclides may or may not have probability table data. Correlation must also be retained in the unresolved energy range when two or more cross-section sets for an isotope that utilize probability tables are at different temperatures.

The impact of the probability-table approach has been studied [83] and found to have negligible impact for most fast and thermal systems. Small but significant changes in reactivity may be observed for plutonium and ^{233}U systems, depending upon the detailed shape of the spectrum. However, the probability-table method can produce substantial increases in reactivity for systems that include large amounts of ^{238}U and have high fluxes within the unresolved resonance region. Calculations for such systems will produce significantly nonconservative results unless the probability-table method is employed.

2.4.4 Photon Interactions

Sampling of a collision nuclide, analog capture, implicit capture, and many other aspects of photon interactions such as variance reduction, are the same as for neutrons. The collision physics are completely different.

The MCNP code has two photon interaction models: simple and detailed.

The simple physics treatment ignores coherent (Thomson) scattering and fluorescent photons from photoelectric absorption. It is intended for high-energy photon problems or problems where electrons are free and is also important for next-event estimators such as point detectors, where scattering can be nearly straight ahead with coherent scatter. The simple physics treatment uses implicit capture unless overridden with the CUT:p card, in which case it uses analog capture.

The detailed physics treatment includes coherent (Thomson) scattering and accounts for fluorescent photons after photoelectric absorption. Form factors and Compton profiles are used to account for electron binding effects. Analog capture is always used. The detailed physics treatment is used below energy EMCPF on the PHYS:p card, and because the default value of EMCPF is 100 MeV, that means it is almost always used by default. It is the best treatment for most applications, particularly for high-Z nuclides or deep penetration problems.

The generation of electrons from photons is handled three ways. These three ways are the same for both the simple and detailed photon physics treatments.

1. If electron transport is turned on (MODE P E), then all photon collisions except coherent scatter can create electrons that are banked for later transport.
2. If electron transport is turned off (no E on the MODE card), then a thick-target bremsstrahlung model (TTB) is used. This model generates electrons, but assumes that they are locally slowed to rest. Any bremsstrahlung photons produced by the non-transported electrons are then banked for later transport. Thus electron-induced photons are not neglected, but the expensive electron transport step is omitted. The TTB production model contains many approximations compared to models used in actual electron transport. In particular, the bremsstrahlung photons inherit the direction of the parent electron.
3. If IDES = 1 on the PHYS:p card, then all electron production is turned off, no electron-induced photons are created, and all electron energy is assumed to be locally deposited.

The TTB approximation is the default for MODE P problems. In MODE P E problems, it plays a role when the energy cutoff for electrons is greater than that for photons. In this case, the TTB model is used in the terminal processing of the electrons to account for the few low-energy bremsstrahlung photons that would be produced at the end of the electrons' range.

2.4.4.1 Simple Physics Treatment

The simple physics treatment is intended primarily for higher energy photons. It is inadequate for high-Z nuclides or deep penetration problems. The physical processes treated are photoelectric effect, pair production, Compton scattering from free electrons, and (optionally) photonuclear interactions (described in §2.4.4.3). The photoelectric effect is regarded as an absorption (without fluorescence). The kinematics of Compton scattering is assumed to be with free electrons (without the use of form factors or Compton profiles). The total scattering cross section, however, includes the incoherent scattering factor regardless of the use of simple or detailed physics. Thus, strict comparisons with codes using only the Klein-Nishina differential cross section are not valid. Highly forward coherent Thomson scattering is ignored. Thus the total cross section σ_t is regarded as the sum of three components:

$$\sigma_{\mathrm{t}} = \sigma_{\mathrm{pe}} + \sigma_{\mathrm{pp}} + \sigma_{\mathrm{s}}. \tag{2.131}$$

2.4.4.1.1 Photoelectric Effect

This is treated as a pure absorption by implicit capture with a corresponding reduction in the photon weight WGT, and hence does not result in the loss of a particle history except for Russian roulette played on the weight cutoff. The non-captured weight $WGT(1-\sigma_{\rm pe}/\sigma_{\rm s})$ is then forced to undergo either pair production or Compton scattering. The captured weight either is assumed to be locally deposited or becomes a photoelectron for electron transport or for the TTB approximation.

2.4.4.1.2 Pair Production

In a collision resulting in pair production [probability $\sigma_{\rm pp}/(\sigma_{\rm t}-\sigma_{\rm pe})$], either an electron-positron pair is created for further transport (or the TTB treatment) and the photon disappears, or it is assumed that the kinetic energy $WGT(E-1.022)$ MeV of the electron-positron pair produced is deposited as thermal energy at the time and point of collision, with isotropic production of one photon of energy 0.511 MeV headed in one direction and another photon of energy 0.511 MeV headed in the opposite direction. The rare single 1.022-MeV annihilation photon is ignored. The relatively unimportant triplet production process is also ignored. The simple physics treatment for pair production is the same as the detailed physics treatment that is described in §2.4.4.2.4.

2.4.4.1.3 Compton Scattering

The alternative to pair production is Compton scattering on a free electron, with probability $\sigma_{\rm s}/(\sigma_{\rm t}-\sigma_{\rm pe})$. In the event of such a collision, the objective is to determine the energy E' of the scattered photon, and $\mu=\cos(\theta)$ for the angle θ of deflection from the line of flight. This yields at once the energy $WGT(E–E')$ deposited at the point of collision and the new direction of the scattered photon. The energy deposited at the point of collision can then be used to make a Compton recoil electron for further transport or for the TTB approximation. The differential cross section for the process is given by the Klein-Nishina formula [17]

$$K(\alpha,\mu)\mathrm{d}\mu=\pi r_o^2\left(\frac{\alpha'}{\alpha}\right)^2\left[\frac{\alpha'}{\alpha}+\frac{\alpha}{\alpha'}+\mu^2-1\right]\mathrm{d}\mu, \tag{2.132}$$

where r_o is the classical electron radius 2.817938×10^{-13} cm , α and α' are the incident and final photon energies in units of 0.511 MeV [$\alpha=E/\left(mc^2\right)$, where m is the mass of the electron and c is the speed of light], and $\alpha'=\alpha/[1+\alpha(1-\mu)]$.

The Compton scattering process is sampled exactly by Kahn's method [35] below 1.5 MeV and by Koblinger's method [84] above 1.5 MeV as analyzed and recommended by Blomquist and Gelbard [85].

2.4.4.2 Detailed Physics Treatment

The detailed physics treatment includes coherent (Thomson) scattering and accounts for fluorescent photons after photoelectric absorption. Again, photonuclear interactions may (optionally) be included [§2.4.4.3]. Form factors are used with coherent and incoherent scattering to account for electron binding effects. Photo-neutron reactions can also be included for select isotopes. Analog capture is always used, as described in §2.4.4.2.3. The detailed physics treatment is used below energy `EMCPF` on the `PHYS`:`p` card, and because the default value of `EMCPF` is 100 MeV, that means it is almost always used by default. It is the best treatment for most applications, particularly for high-Z nuclides or deep penetration problems.

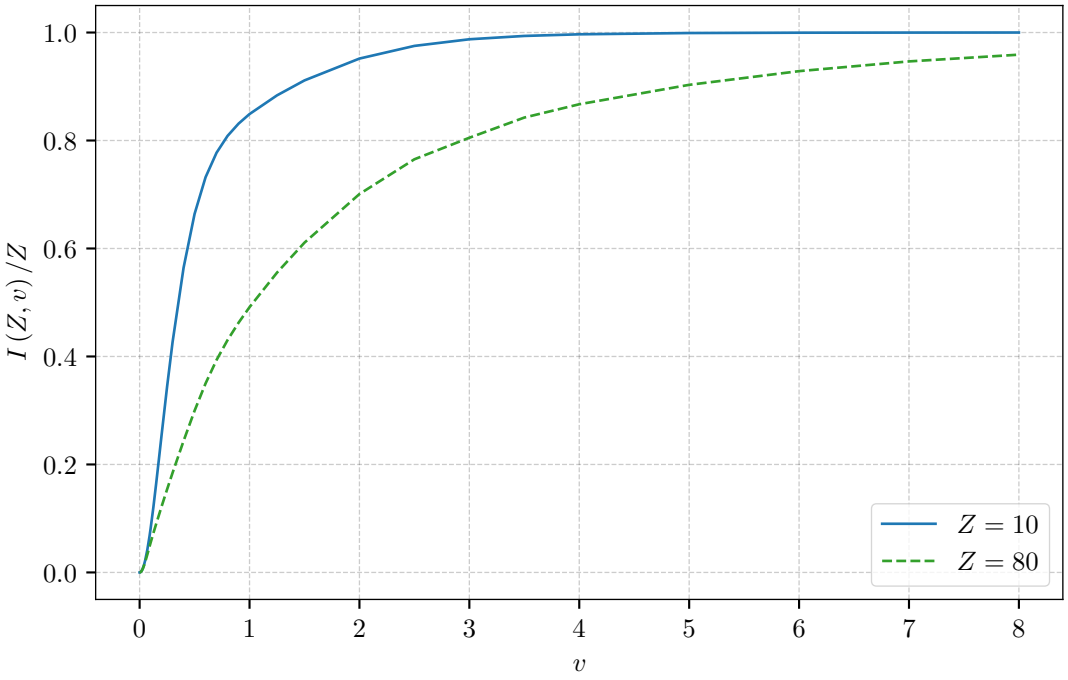


Figure 2.5: Scattering factor modifying the Klein-Nishina cross section from [86].

When using next-event estimators (point detectors, ring detectors, or image detectors) with detailed physics treatment, one should enable one of two options, as explained in §2.4.4.2.5. That is, one can turn off coherent scattering with the NOCOH = 1 option on the PHYS:p card. Or, one can prevent large scores from rare coherent scattering contributions by enabling next-event estimator contributions from all possible photon reactions at each collision by setting the PDS = 1 option on the FT card.

2.4.4.2.1 Incoherent (Compton) Scattering

To model Compton scattering it is necessary to determine the angle θ of scattering from the incident line of flight (and thus the new direction), the new energy E' of the photon, and the recoil kinetic energy of the electron, $E - E'$. The recoil kinetic energy can be deposited locally, can be transported in MODE P E problems, or (default) can be treated with the TTB approximation.

Incoherent scattering is assumed to have the differential cross section

$$\sigma_I(Z, \alpha, \mu)\mathrm{d}\mu = I(Z, v)K(\alpha, \mu)\mathrm{d}\mu, \tag{2.133}$$

where $I(Z, v)$ is an appropriate scattering factor modifying the Klein-Nishina cross section in Eq. (2.119).

Qualitatively, the effect of $I(Z, v)$ is to decrease the Klein-Nishina cross section (per electron) more extremely in the forward direction, for low E and for high-Z independently. For any Z, $I(Z, v)$ increases from $I(Z, 0) = 0$ to $I(Z, \infty) = Z$. The parameter v is the inverse length $v = \sin(\theta/2)/\lambda = \kappa\alpha\sqrt{1-\mu}$, where $\kappa = m_o c/\left(h\sqrt{2}\right) = 29.1434$ Å^{-1}. The maximum value of v is $v_{\max} = \kappa\alpha\sqrt{2} = 41.2149\alpha$ Å^{-1} at $\mu = -1$. The essential features of $I(Z, v)$ are indicated in Figure 2.5. These parameters are calculated using MCNP internal variables, which may not match the current best values.

For hydrogen, an exact expression for the form factor is used [86, 87], which is

$$I(1,v) = 1 - \frac{1}{\left(1 + \frac{1}{2}\frac{f^2 v^2}{\kappa^2}\right)^4}, \tag{2.134}$$

where f is the inverse fine structure constant, $f = 137.0393$, and $f/\sqrt{2} = 96.9014$.

The Klein-Nishina formula is sampled exactly by Kahn's method [35] below 1.5 MeV and by Koblinger's method [84] above 1.5 MeV as analyzed and recommended by Blomquist and Gelbard [85]. The outgoing energy E' and angle μ are rejected according to the form factors.

For next-event estimators such as detectors and DXTRAN, the probability density for scattering toward the detector point must be calculated as

$$p(\mu) = \frac{1}{\sigma_1(Z,\alpha)} I(Z,v) K(\alpha,\mu) = \frac{\pi r_o^2}{\sigma_1(Z,\alpha)} I(Z,v) \left(\frac{\alpha'}{\alpha}\right)^2 \left(\frac{\alpha}{\alpha'} + \frac{\alpha'}{\alpha} + \mu^2 - 1\right), \tag{2.135}$$

where $\pi r_o = 0.2494351$ and $\sigma_1(Z,\alpha)$ and $I(Z,v)$ are looked up in the data library.

The new energy, E', of the photon accounts for the effects of a bound electron. The electron binding effect on the scattered photon's energy distribution appears as a broadening of the energy spectrum due to the pre-collision momentum of the electron. This effect on the energy distribution of the incoherently scattered photon is called Doppler broadening.

The Hartree-Fock Compton profiles, $J(p_z)$, are used to account for the effects of a bound electron on the energy distribution of the scattered photon. These Compton profiles are a collection of orbital and total atom data tabulated as a function of the projected pre-collision momentum of the electron. Values of the Compton profiles for the elements are published in tabular form by Biggs et al. [52] as a function of p_z.

The scattered energy of a Doppler broadened photon can be calculated by selecting an orbital shell, sampling the projected momentum from the Compton profile, and calculating the scattered photon energy, E', from

$$p_z = -f \frac{E - E' - EE'(1 - \cos(\theta))/mc^2}{\sqrt{E^2 + E'^2 - 2EE'\cos(\theta)}}. \tag{2.136}$$

The Compton profiles are related to the incoherent scattering function, $I(Z,v)$, by

$$I(Z,v) = \sum_k \int_{-\infty}^{p_z^{\max}} J_k(p_z, Z)\mathrm{d}p_z, \tag{2.137}$$

where k refers to the particular electron subshell, $J_k(p_z, Z)$ is the Compton profile of the k^{th} shell for a given element, and $p_z^{\max}$ is the maximum momentum transferred and is calculated using $E' = E - E_{\text{binding}}$.

2.4.4.2.2 Coherent (Thomson) Scattering

Thomson scattering involves no energy loss, and thus is the only photon process that cannot produce electrons for further transport and that cannot use the TTB approximation. Only the scattering angle θ is computed, and then the transport of the photon continues.

The differential cross section is $\sigma_2(Z,\alpha,\mu)\mathrm{d}\mu = C^2(Z,v)T(\mu)\mathrm{d}\mu$, where $C(Z,v)$ is a form factor modifying the energy-independent Thomson cross section $T(\mu) = \pi r_o^2(1+\mu^2)\mathrm{d}\mu$.

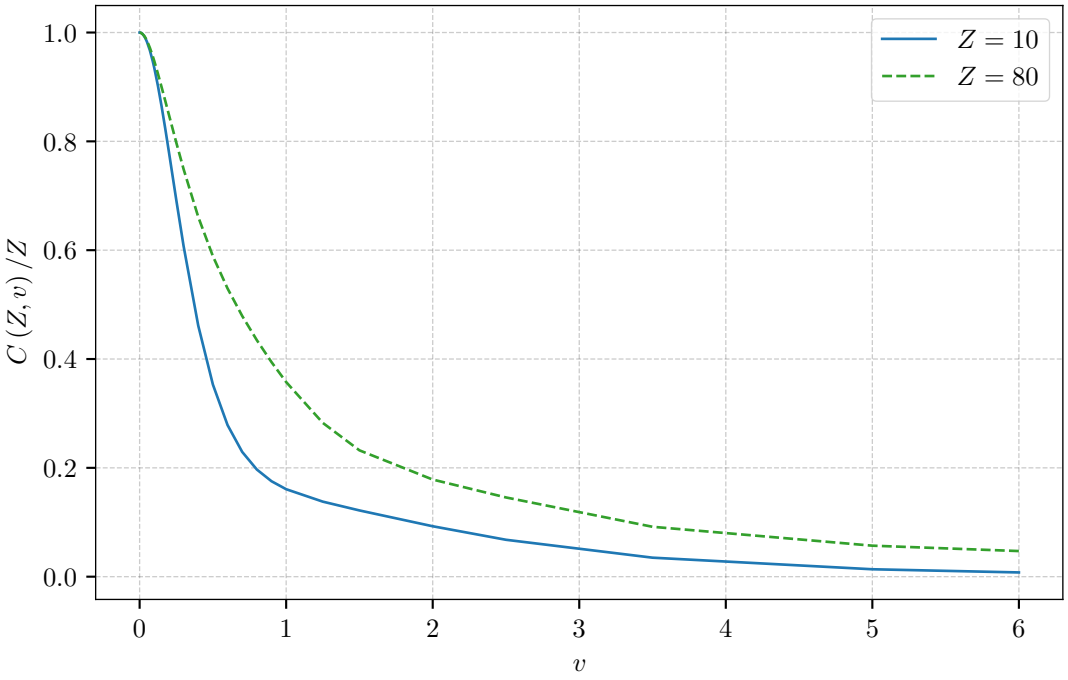


Figure 2.6: Form factor modifying the energy-dependent Thomson cross section from [86].

The general effect of $C^2(Z,v)/Z^2$ is to decrease the Thomson cross section more extremely for backward scattering, for high E, and low Z. This effect is opposite in these respects to the effect of $I(Z,v)/Z$ on $K(\alpha,\mu)$ in incoherent (Compton) scattering. For a given Z, $C(Z,v)$ decreases from $C(Z,0) = Z$ to $C(Z,\infty) = 0$. For example, $C(Z,v)$ is a rapidly decreasing function of μ as μ varies from $+1$ to -1, and therefore the coherent cross section is peaked in the forward direction. At high energies of the incoming photon, coherent scattering is strongly forward and can be ignored. The parameter v is the inverse length $v = \sin(\theta/2)/\lambda = \kappa\alpha\sqrt{1-\mu}$, where $\kappa = m_o c/\left(h\sqrt{2}\right) = 29.1434$ Å^{-1}. The maximum value of v is $v_{\max} = \kappa\alpha\sqrt{2} = 41.2149\alpha$ Å^{-1} at $\mu = -1$. The square of the maximum value is $v^2_{\max} = 1698.6679\alpha^2$ Å^{-2}. The qualitative features of $C(Z,v)$ are shown in Figure 2.6. These parameters are calculated using MCNP internal variables, which may not match the current best values.

For next-event estimators, one must evaluate the probability density function

$$p(\mu) = \pi r_o^2\left(1+\mu^2\right)C^2(Z,v)/\sigma_2(Z,\alpha) \tag{2.138}$$

for a given μ. Here $\sigma_2(Z,\alpha)$ is the integrated coherent cross section. The value of $C(Z,v)$ at $v = \kappa\alpha\sqrt{1-\mu}$ must be interpolated in the original $C^2(Z,v_i)$ tables separately stored on the cross-section library for this purpose.

Note that at high energies, coherent scattering is virtually straight ahead with no energy loss; thus, it appears from a transport viewpoint that no scattering took place. For a point detector to sample this scattering, the point must lie on the original track ($\mu \cong 1$), which is seldom the case. Thus, photon point detector variances generally will be much greater with detailed photon physics than with simple physics unless coherent scattering is turned off with `NOCOH` $= 1$ on the `PHYS`:`p` card or by enabling next-event estimator contributions from all possible photon reactions at each collision by setting the `PDS` option on the `FT` card, as explained in §2.4.4.2.5.

2.4.4.2.3 Photoelectric Effect

The photoelectric effect consists of the absorption of the incident photon of energy E, with the consequent emission of several fluorescent photons and the ejection (or excitation) of an orbital electron of binding energy $e < E$, giving the electron a kinetic energy of $E - e$. Zero, one, or two fluorescent photons are emitted. These three cases are now described.

(1) Zero photons greater than 1 keV are emitted. In this event, the cascade of electrons that fills up the orbital vacancy left by the photoelectric ejection produces electrons and low-energy photons (Auger effect). These particles can be followed in `MODE P E` problems, or be treated with the TTB approximation, or be assumed to deposit energy locally. Because no photons are emitted by fluorescence (some may be produced by electron transport or the TTB model), the photon track is terminated. This photoelectric "capture" of the photon is scored like analog capture in the summary table of the output file. Implicit capture is not possible.

(2) One fluorescent photon of energy greater than 1 keV is emitted. The photon energy E' is the difference in incident photon energy E, less the ejected electron kinetic energy $E - e$, less a residual excitation energy e' that is ultimately dissipated by further Auger processes. This dissipation leads to additional electrons or photons of still lower energy. The ejected electron and any Auger electrons can be transported or treated with the TTB approximation. In general,

$$E' = E - (E - e) - e' = e - e'. \tag{2.139}$$

These primary transactions are taken to have the full fluorescent yield from all possible upper levels e', but are apportioned among the x-ray lines $K\alpha_1$, $(L_3 \to K)$; $K\alpha_2$, $(L_2 \to K)$; $K\beta_1'$, $(\text{mean}\, M \to K)$; and $k\beta_2'$, $(\text{mean}\, N \to K)$.

(3) Two fluorescence photons can occur if the residual excitation e' of process (2) exceeds 1 keV. An electron of binding energy e'' can fill the orbit of binding energy e', emitting a second fluorescent photon of energy $E'' = e'$–e''. As before, the residual excitation e'' is dissipated by further Auger events and electron production that can be modeled with electron transport in `MODE P E` calculations, approximated with the TTB model, or assumed to deposit all energy locally. These secondary transitions come from all upper shells and go to L shells. Thus the primary transitions must be $K\alpha_1$ or $K\alpha_2$ to leave an L shell vacancy.

Each fluorescent photon born as discussed above is assumed to be emitted isotropically and can be transported, provided that $E', E'' > 1$ keV. The binding energies e, e', and e'' are very nearly the x-ray absorption edges because the x-ray absorption cross section takes an abrupt jump as it becomes energetically possible to eject (or excite) the electron of energy $E \cong e''$, then e', then e, etc. The jump can be as much as a factor of 20 (for example, K-carbon).

A photoelectric event is terminal for elements $Z < 12$ because the possible fluorescence energy is below 1 keV. The event is only a single fluorescence of energy above 1 keV for $31 > Z \geq 12$, but double fluorescence (each above 1 keV) is possible for $Z \geq 31$. For $Z \geq 31$, primary lines $K\alpha_1$, $K\alpha_2$, and $K\beta_1'$ are possible and, in addition, for $Z \geq 37$, the $K\beta_2'$ line is possible.

In all photoelectric cases where the photon track is terminated because either no fluorescent photons are emitted or the ones emitted are below the energy cutoff, the termination is considered to be caused by analog capture in the output file summary table (and not energy cutoff).

2.4.4.2.4 Pair Production

This process is considered only in the field of a nucleus. The threshold is $2mc^2[1 + (m/M)] \cong 1.022$ MeV, where M is the nuclear mass and m is the mass of the electron. There are three cases:

1. In the case of electron transport (MODE P E), the electron and positron are created and banked and the photon track terminates.

2. For MODE P problems with the TTB approximation, both an electron and positron are produced but not transported. Both particles can make TTB approximation photons. The positron is then considered to be annihilated locally and a photon pair is created as in case (3).

3. For MODE P problems when positrons are not created by the TTB approximation, the incident photon of energy E vanishes. The kinetic energy of the created positron/electron pair, assumed to be $E - 2mc^2$, is deposited locally at the collision point. The positron is considered to be annihilated with an electron at the point of collision, resulting in a pair of photons, each with the incoming photon weight, and each with an energy of $mc^2 = 0.511$ MeV. The first photon is emitted isotropically, and the second is emitted in the opposite direction. The very rare single-annihilation photon of 1.022 MeV is ignored.

2.4.4.2.5 Caution for Detectors and Coherent Scattering

Users should understand the implications of using detailed photon physics with either next-event estimators or DXTRAN. The user should take care to enable either one of three possible solutions when using the detailed physics treatment with photon next-event estimators (such as point detector, ring detectors, and image detectors). The user can use two of these options for photon simulations with DXTRAN. For both next-event estimators and DXTRAN, one can turn off coherent scattering with the NOCOH = 1 option on the PHYS:p card or enable the simple physics treatment (EMCPF < 0.001 on the PHYS:p card). Alternatively, for next-event estimators only, one can enable sampling all reactions at each collision event with the FT PDS option for each tally.

Turning off coherent scattering can improve the figure of merit (FOM) [§2.6.5] by more than a factor of 10 for tallies with small relative errors because coherent scattering is highly peaked in the forward direction. Consequently, coherent scattering becomes undersampled because the photon must be traveling directly at the detector point and undergo a coherent scattering event. When the photon is traveling nearly in the direction of the point detector or the chosen point on a ring detector or DXTRAN sphere, the $p(\mu)$ term (called "PSC" in the MCNP code) of the point detector [§2.5.6.1] becomes large, which causes a large score for the event and may significantly increase the variance of the tally.

The reason that a large score can occur is that $p(\mu)$ is a probability density function, not a probability ($p(\mu)\mathrm{d}\mu$, which can be no larger than unity), and can approach infinity for highly forward-peaked scattering. Thus, the under-sampled coherent scattering event is characterized by many low contributions to the detector when the photon trajectory is directed away from the detector (when $p(\mu)$ is small) and a few large contributions when the trajectory is nearly pointed toward the detector (and $p(\mu)$ is large). Such under-sampled events cause a large score and commensurate increase in both the tally mean and the variance, a decrease in the FOM, and likely a failure to pass the statistical checks for the tally as described in §2.6.9.2.3. One way to improve sampling for point-detector tallies of these forward-peaked behaviors is with the FT PDS option. The FT PDS option enables contributions from all possible reaction types (and optionally all isotopes as well) at each collision. With this option, the statistical weight of each reaction contribution is based on the reaction probability, which results in many small contributions from coherent scattering instead of a few large contributions.

2.4.4.3 Photonuclear Physics Treatment

Photonuclear physics may be included when handling a photon collision. A photonuclear interaction begins with the absorption of a photon by a nucleus. There are several mechanisms by which this can occur. The nuclear data files currently available focus on the energy range up to 150 MeV incident photon energy. The

value of 150 MeV was chosen as this energy is just below the threshold for the production of pions and the subsequent need for much more complicated nuclear modeling. Below 150 MeV, the primary mechanisms for photoabsorption are the excitation of either the giant dipole resonance or a quasi-deuteron nucleon pair.

The giant dipole resonance (GDR) absorption mechanism can be conceptualized as the electromagnetic wave, the photon, interacting with the dipole moment of the nucleus as a whole. This results in a collective excitation of the nucleus. It is the most likely process (that is, the largest cross section) by which photons interact with the nucleus. Expected peak cross sections of 6–10 millibarns are seen for the light isotopes and 600–800 millibarns are not uncommon for the heavy elements. Thus, photonuclear collisions may account for a theoretical maximum of 5–6% of the photon collisions. The GDR occurs with highest probability when the wavelength of the photon is comparable to the size of the nucleus. This typically occurs for photon energies in the range of 5–20 MeV and has a resonance width of a few MeV. For deformed nuclei, a double peak is seen due to the variation of the nuclear radius. Outside of this resonance region, the cross section for a GDR reaction becomes negligible. A complete description of this process can be found in the text by Bohr and Mottelson [88].

The quasi-deuteron (QD) absorption mechanism can be conceptualized as the electromagnetic wave interacting with the dipole moment of a correlated neutron-proton pair. In this case, the neutron-proton pair can be thought of as a QD having a dipole moment with which the photon can interact. This mechanism is not as intense as the GDR but it provides a significant background cross section for all incident photon energies above the relevant particle separation threshold. The seminal work describing this process was published by Levinger [89, 90]. Recent efforts to model this process include the work of Chadwick et al. [91].

Once the photon has been absorbed by the nucleus, one or more secondary particle emissions can occur. For the energy range in question (that is, below 150 MeV) these reactions may produce a combination of gamma rays, neutrons, protons, deuterons, tritons, helium-3 particles, alphas, and fission fragments. The threshold for the production of a given secondary particle is governed by the separation energy of that particle, typically a few MeV to as much as a few 10s of MeV. Most of the these particles are emitted via pre-equilibrium and equilibrium mechanisms though it is possible, but rare, to have a direct emission.

Pre-equilibrium emission can be conceptualized as a particle within the nucleus that receives a large amount of energy from the absorption mechanism and escapes the binding force of the nucleus after at least one but very few interactions with other nuclei. This is in contrast to a direct emission where the emission particle escapes the nucleus without any interactions. Typically this occurs from QD absorption of the photon where the incident energy is initially split between the neutron-proton pair. Particles emitted by this process tend to be characterized by higher emission energies and forward-peaked angular distributions.

Equilibrium emission can be conceptualized as particle evaporation. This process typically occurs after the available energy has been generally distributed among the nucleons. In the classical sense, particles boil out of the nucleus as they penetrate the nuclear potential barrier. The barrier may contain contributions from coulomb potential (for charged particles) and effects of angular momentum conservation. It should be noted that for heavy elements, evaporation neutrons are emitted preferentially as they are not subject to the coulomb barrier. Particles emitted by this process tend to be characterized by isotropic angular emission and evaporation energy spectra. Several references are available on the general emission process after photoabsorption [92–94].

For all of the emission reactions discussed thus far, the nucleus will most probably be left in an excited state. It will subsequently relax to the ground state by the emission of one or more gamma rays. The gamma-ray energies follow the well known patterns for relaxation. The only reactions that do not produce gamma-rays are direct reactions where the photon is absorbed and all available energy is transferred to a single emission particle leaving the nucleus in the ground state.

Reactions at higher energies (above the pion production threshold) require more thorough descriptions of the underlying nuclear physics. The delta resonance and other absorption mechanisms become significant and the

amount of energy involved in the reaction presents the opportunity for the production of more fundamental particles. While beyond the scope of this current work, descriptions of the relevant physics may be found in the paper by Fasso et al. [95].

New photonuclear data tables are used to extend the traditional photon collision routines. Because of the sparsity of photonuclear data, the user is allowed to toggle photonuclear physics on or off (with the fourth entry on the PHYS:p card) and the code defaults to off. Once turned on, the total photon cross section, photoatomic plus photonuclear (i.e. the photonuclear cross section is absent from this calculation when photonuclear physics is off), is used to determine the distance to the next photon collision. For simple physics, this implies the sum of the photoelectric, pair production, incoherent and photonuclear cross sections. Detailed physics includes the additional coherent cross section in this sum.

The toggle for turning on and off photonuclear physics is also used to select biased or unbiased photonuclear collisions. For the unbiased option, the type of collision is sampled as either photonuclear or photoatomic based on the ratio of the partial cross sections. The biased option is similar to forced collisions. At the collision site, the particle is split into two parts, one forced to undergo photoatomic interaction and the other photonuclear. The weight of each particle is adjusted by the ratio of their actual collision probability. The photoatomic sampling routines (as described in §2.4.4.1 and §2.4.4.2) are used to sample the emission characteristics for secondary electrons and photons from a photoatomic collision. The emission characteristics for secondary particles from photonuclear collisions are handled independently.

Once it has been determined that a photon will undergo a photonuclear collision, the emission particles are sampled as follows. First, the appropriate collision isotope is selected based on the ratio of the total photonuclear cross section from each relevant table. Note that photoatomic collisions are sampled from a set of elemental tables whereas photonuclear collisions are sampled from a set of isotopic tables. Next, the code computes the ratio of the production cross section to the total cross section for each secondary particle undergoing transport. Based on this ratio, an integer number of emission particles are sampled. If weight games (i.e. weight cutoffs or weight windows) are being used, these secondary particles are subjected to splitting or roulette to ensure that the sampled particles will be of an appropriate weight. The emission parameters for each secondary particle are then sampled independently from the reaction laws provided in the data. Last, tallies and summary information are appropriately updated, applicable variance reduction games are performed, and the emitted particle is banked for further transport.

Note that photonuclear physics was implemented in the traditional Monte Carlo style as a purely statistical based process. This means that photons undergoing a photonuclear interaction produce an average number of emission particles. For multiple particle emission, the particles may not be sampled from the same reaction; for example, if two neutrons are sampled, one may be from the $(\gamma, 2n)$ distributions and the second from the (γ, np) distributions. Note that the photonuclear data use the same energy/angle distributions that have been used for neutrons and the same internal coding for sampling. See §2.4.3.5.4. This generalized particle production method is statistically correct for large sampling populations and lends itself to uncomplicated biasing schemes. It is (obviously) not microscopically correct. It is not possible to perform microscopically correct sampling given the current set of data tables.

Because of the low probability of a photon undergoing a photonuclear interaction, the use of biased photonuclear collisions may be necessary. However, caution should be exercised when using this option as it can lead to large variations in particle weights. It is important to check the summary tables to determine if appropriate weight cutoff or weight windows have been set. That is, check to see if weight cutoffs or weight windows are causing more particle creation and destruction than expected. It is almost always necessary to adjust the default neutron weight cutoff (when using only weight cutoffs with photonuclear biasing) as it will roulette a large fraction of the attempts to create secondary photoneutrons.

More information about the photonuclear physics included in the MCNP code can be found in White [96, 97].

2.4.5 Electron Interactions

The transport of electrons and other charged particles is fundamentally different from that of neutrons and photons. The interaction of neutral particles is characterized by relatively infrequent isolated collisions, with simple free flight between collisions. By contrast, the transport of electrons is dominated by the long-range Coulomb force, resulting in large numbers of small interactions. As an example, a neutron in aluminum slowing down from 0.5 MeV to 0.0625 MeV will have about 30 collisions, while a photon in the same circumstances will experience fewer than ten. An electron accomplishing the same energy loss will undergo about 105 individual interactions. This great increase in computational complexity makes a single-collision Monte Carlo approach to electron transport infeasible for most situations of practical interest.

Considerable theoretical work has been done to develop a variety of analytic and semi-analytic multiple-scattering theories for the transport of charged particles. These theories attempt to use the fundamental cross sections and the statistical nature of the transport process to predict probability distributions for significant quantities, such as energy loss and angular deflection. The most important of these theories for the algorithms in the MCNP code are the Goudsmit-Saunderson [98] theory for angular deflections, the Landau [99] theory of energy-loss fluctuations, and the Blunck-Leisegang [100] enhancements of the Landau theory. These theories rely on a variety of approximations that restrict their applicability, so that they cannot solve the entire transport problem. In particular, it is assumed that the energy loss is small compared to the kinetic energy of the electron.

In order to follow an electron through a significant energy loss, it is necessary to break the electron's path into many steps. These steps are chosen to be long enough to encompass many collisions (so that multiple-scattering theories are valid) but short enough that the mean energy loss in any one step is small (so that the approximations necessary for the multiple-scattering theories are satisfied). The energy loss and angular deflection of the electron during each of the steps can then be sampled from probability distributions based on the appropriate multiple-scattering theories. This accumulation of the effects of many individual collisions into single steps that are sampled probabilistically constitutes the "condensed history" Monte Carlo method.

The most influential reference for the condensed history method is the 1963 paper by Berger [101]. Based on the techniques described in that work, Berger and Seltzer developed the ETRAN series of electron/photon transport codes [102]. These codes have been maintained and enhanced for many years at the National Bureau of Standards (now the National Institute of Standards and Technology). The ETRAN codes are also the basis for the Integrated TIGER Series [103], a system of general-purpose, application-oriented electron/photon transport codes developed and maintained by Halbleib and his collaborators at Sandia National Laboratories in Albuquerque, New Mexico. The electron physics in the MCNP code is essentially that of the Integrated TIGER Series, Version 3.0. The ITS radiative and collisional stopping power and bremsstrahlung production models were integrated into MCNP4C.

2.4.5.1 Electron Steps and Substeps

The condensed random walk for electrons can be considered in terms of a sequence of sets of values

$$(0, E_0, t_0, \mathbf{u}_0, \mathbf{r}_0),\ (s_1, E_1, t_1, \mathbf{u}_1, \mathbf{r}_1),\ (s_2, E_2, t_2, \mathbf{u}_2, \mathbf{r}_2), \ldots$$

where s_n, E_n, t_n, $\mathbf{u}_n$, and $\mathbf{r}_n$ are the total path length, energy, time, direction, and position of the electron at the end of n steps. On the average, the energy and path length are related by

$$E_{n-1} - E_n = -\int_{s_{n-1}}^{s_n} \frac{\mathrm{d}E}{\mathrm{d}s}\mathrm{d}s, \tag{2.140}$$

where $-\mathrm{d}E/\mathrm{d}s$ is the total stopping power in energy per unit length. This quantity depends on energy and on the material in which the electron is moving. ETRAN-based codes customarily choose the sequence of path lengths $\{s_n\}$ such that

$$\frac{E_n}{E_{n-1}} = k, \tag{2.141}$$

for a constant k. The most commonly used value is $k = 2^{-1/8}$, which results in an average energy loss per step of 8.3%.

Electron steps with (energy-dependent) path lengths $s = s_n - s_{n-1}$ determined by Eqs. (2.140)–(2.141) are called major steps or energy steps. The condensed random walk for electrons is structured in terms of these energy steps. For example, all pre-calculated and tabulated data for electrons are stored on an energy grid whose consecutive energy values obey the ratio in Eq. (2.141). In addition, the Landau and Blunck-Leisegang theories for energy straggling are applied once per energy step. See §2.4.5.6 for a more detailed option. For a single step, the angular scattering could also be calculated with satisfactory accuracy, since the Goudsmit-Saunderson theory is valid for arbitrary angular deflections. However, the representation of the electron's trajectory as the result of many small steps will be more accurate if the angular deflections are also required to be small. Therefore, the ETRAN codes and the MCNP code further break the electron steps into smaller substeps. A major step of path length s is divided into m substeps, each of path length s/m. Angular deflections and the production of secondary particles are sampled at the level of these substeps. The integer m depends only on material (average atomic number Z). Appropriate values for m have been determined empirically, and range from $m = 2$ for $Z < 6$ to $m = 15$ for $Z > 91$.

In some circumstances, it may be desirable to increase the value of m for a given material. In particular, a very small material region may not accommodate enough substeps for an accurate simulation of the electron's trajectory. In such cases, the user can increase the value of m with the `ESTEP` option on the material card `M`. The user can gain some insight into the selection of m by consulting `PRINT` Table 85 in the MCNP output. Among other information, this table presents a quantity called DRANGE as a function of energy. DRANGE is the size of an energy step in g/cm^2. Therefore, DRANGE/m is the size of a substep in the same units, and if ρ is the material density in g/cm^3, then DRANGE/$(m\rho)$ is the length of a substep in centimeters. This quantity can be compared with the smallest dimension of a material region. A reasonable rule of thumb is that an electron should make at least ten substeps in any material of importance to the transport problem.

2.4.5.2 Condensed Random Walk

In the initiation phase of a transport calculation involving electrons, all relevant data are either precalculated or read from the electron data file and processed. These data include the electron energy grid, stopping powers, electron ranges, energy step ranges, substep lengths, and probability distributions for angular deflections and the production of secondary particles. Although the energy grid and electron steps are selected according to Eqs. (2.140)–(2.141), energy straggling, the analog production of bremsstrahlung, and the intervention of geometric boundaries and the problem time cutoff will cause the electron's energy to depart from a simple sequence s_n satisfying Eq. (2.141). Therefore, the necessary parameters for sampling the random walk will be interpolated from the points on the energy grid.

At the beginning of each major step, the collisional energy loss rate is sampled (unless the logic described in §2.4.5.6 is being used). In the absence of energy straggling, this will be a simple average value based on the nonradiative stopping power described in the next section. In general, however, fluctuations in the energy loss rate will occur. The number of substeps m per energy step will have been preset, either from the empirically determined default values, or by the user, based on geometric considerations. At most m substeps will be taken in the current major step with the current value for the energy loss rate. The number of substeps may be reduced if the electron's energy falls below the boundary of the current major step, or if the electron reaches a geometric boundary. In these circumstances, or upon the completion of m substeps, a new major step is begun, and the energy loss rate is resampled.

With the possible exception of the energy loss and straggling calculations, the detailed simulation of the electron history takes place in the sampling of the substeps. The Goudsmit-Saunderson [98] theory is used to sample from the distribution of angular deflections, so that the direction of the electron can change at the end of each substep. Based on the current energy loss rate and the substep length, the projected energy for the electron at the end of the substep is calculated. Finally, appropriate probability distributions are sampled for the production of secondary particles. These include electron-induced fluorescent X-rays, "knock-on" electrons (from electron-impact ionization), and bremsstrahlung photons.

Note that the length of the substep ultimately derives from the total stopping power used in Eq. 2.140, but the projected energy loss for the substep is based on the nonradiative stopping power. The reason for this difference is that the sampling of bremsstrahlung photons is treated as an essentially analog process. When a bremsstrahlung photon is generated during a substep, the photon energy is subtracted from the projected electron energy at the end of the substep. Thus the radiative energy loss is explicitly taken into account, in contrast to the collisional (nonradiative) energy loss, which is treated probabilistically and is not correlated with the energetics of the substep. Two biasing techniques are available to modify the sampling of bremsstrahlung photons for subsequent transport. However, these biasing methods do not alter the linkage between the analog bremsstrahlung energy and the energetics of the substep.

The MCNP code uses identical physics for the transport of electrons and positrons, but distinguishes between them for tallying purposes, and for terminal processing. Electron and positron tracks are subject to the usual collection of terminal conditions, including escape (entering a region of zero importance), loss to time cutoff, loss to a variety of variance-reduction processes, and loss to energy cutoff. The case of energy cutoff requires special processing for positrons, which will annihilate at rest to produce two photons, each with energy $mc^2 = 0.511008$ MeV.

2.4.5.3 Collisional Stopping Power

Berger [101] gives the restricted electron collisional stopping power, i.e., the energy loss per unit path length to collisions resulting in fractional energy transfers ϵ less than an arbitrary maximum value ϵ_m, in the form

$$-\left(\frac{\mathrm{d}E}{\mathrm{d}s}\right)_{\epsilon_m} = NZC\left\{\ln\left(\frac{E^2(\tau+2)}{2I^2}\right) + f^-(\tau,\epsilon_m) - \delta\right\}, \tag{2.142}$$

where

$$f^-(\tau,\epsilon_m) = -1-\beta^2 + \left(\frac{\tau}{\tau+1}\right)^2\frac{\epsilon_m^2}{2} + \frac{2\tau+1}{(\tau+1)^2}\ln(1-\epsilon_m) + \ln[4\epsilon_m(1-\epsilon_m)] + \frac{1}{1-\epsilon_m}. \tag{2.143}$$

Here ϵ and ϵ_m represent energy transfers as fractions of the electron kinetic energy E; I is the mean ionization potential in the same units as E; β is v/c; τ is the electron kinetic energy in units of the electron rest mass; δ is the density effect correction (related to the polarization of the medium); Z is the average atomic number of the medium; N is the atom density of the medium in cm^{-3}; and the coefficient C is given by

$$C = \frac{2\pi e^4}{mv^2} \tag{2.144}$$

where m, e, and v are the rest mass, charge, and speed of the electron, respectively. The density effect correction δ is calculated using the prescriptions of Sternheimer, Berger and Seltzer [104] when using data from the **el03** library and using the method of Sternheimer and Peierls [105] when using data from the **el** library.

The ETRAN codes and the MCNP code do not make use of restricted stopping powers, but rather treat all collisional events in an uncorrelated, probabilistic way. Thus, only the total energy loss to collisions is

needed, and Eqs. (2.142)–(2.143) can be evaluated for the special value $\epsilon_m = 1/2$. The reason for the $1/2$ is the indistinguishability of the two outgoing electrons. The electron with the larger energy is, by definition, the primary. Therefore, only the range $\epsilon < 1/2$ is of interest. With $\epsilon_m = 1/2$, Eq. (2.143) becomes

$$f^-(\tau, \epsilon_m) = -\beta^2 + [1 - \ln(2)] + \left[\frac{1}{8} + \ln(2)\right]\left(\frac{\tau}{\tau+1}\right)^2. \tag{2.145}$$

On the right side of Eq. (2.142), we can express both E and I in units of the electron rest mass. Then E can be replaced by τ on the right side of the equation. We also introduce supplementary constants

$$C_2 = \ln\bigl(2I^2\bigr), \tag{2.146a}$$
$$C_3 = 1 - \ln(2), \tag{2.146b}$$
$$C_4 = \frac{1}{8} + \ln(2), \tag{2.146c}$$

so that Eq. (2.142) becomes

$$-\left(\frac{\mathrm{d}E}{\mathrm{d}s}\right) = NZ^2\frac{2\pi e^4}{mv^2}\left\{\ln\bigl[\tau^2(\tau+2)\bigr] - C_2 + C_3 - \beta^2 + C_4\left(\frac{\tau}{\tau+1}\right)^2 - \delta\right\}. \tag{2.147}$$

This is the collisional energy loss rate in MeV/cm in a particular medium. In the MCNP code, we are actually interested in the energy loss rate in units of MeV barns (so that different cells containing the same material need not have the same density). Therefore, we divide Eq. (2.147) by N and multiply by the conversion factor 10^{24} barns/cm^2. We also use the definition of the fine structure constant

$$\alpha = \frac{2\pi e^2}{hc}, \tag{2.148}$$

where h is Planck's constant, to eliminate the electronic charge e from Eq. (2.147). The result is as follows:

$$-\left(\frac{\mathrm{d}E}{\mathrm{d}s}\right) = \frac{10^{24}\alpha^2h^2c^2}{2\pi mc^2}Z\left\{\ln\bigl[\tau^2(\tau+2)\bigr] - C_2 + C_3 - \beta^2 + C_4\left(\frac{\tau}{\tau+1}\right)^2 - \delta\right\}\frac{1}{\beta^2}. \tag{2.149}$$

This is the form actually used in the MCNP code to preset the collisional stopping powers at the energy boundaries of the major energy steps.

The mean ionization potential and density effect correction depend upon the state of the material, either gas or solid. In the fit of Sternheimer and Peierls [105] the physical state of the material also modifies the density effect calculation. In the Sternheimer, Berger and Seltzer [104] treatment, the calculation of the density effect uses the conduction state of the material to determine the contribution of the outermost conduction electron to the ionization potential. The occupation numbers and atomic binding energies used in the calculation are from Carlson [106].

2.4.5.4 Radiative Stopping Power

The radiative stopping power is

$$-\left.\frac{\mathrm{d}E}{\mathrm{d}s}\right|_{\mathrm{rad}} = 10^{24}Z(Z+\overline{\eta})\bigl(\alpha r_e^2\bigr)\bigl(T+mc^2\bigr)\Phi_{\mathrm{rad}}^{(n)}, \tag{2.150}$$

where $\Phi_{\mathrm{rad}}^{(n)}$ is the scaled electron-nucleus radiative energy-loss cross section based upon evaluations by Berger and Seltzer for data from either the **el** or the **el03** library (details of the numerical values of the data on the **el03** library can be found in [107–109]); $\overline{\eta}$ is a parameter to account for the effect of electron-electron bremsstrahlung (it is unity when using data from the **el** library and, when using data from the **el03** library, it is based upon the work of Seltzer and Berger [107–109] and can be different from unity); α is the fine structure constant; mc^2 is the mass energy of an electron; and r_e is the classical electron radius. The dimensions of the radiative stopping power are the same as the collisional stopping power.

2.4.5.5 Energy Straggling

Because an energy step represents the cumulative effect of many individual random collisions, fluctuations in the energy loss rate will occur. Thus the energy loss will not be a simple average $\overline{\Delta}$; rather there will be a probability distribution $f(s,\Delta)\mathrm{d}\Delta$ from which the energy loss Δ for the step of length s can be sampled. Landau [99] studied this situation under the simplifying assumptions that the mean energy loss for a step is small compared with the electron's energy, that the energy parameter ξ defined below is large compared with the mean excitation energy of the medium, that the energy loss can be adequately computed from the Rutherford [110] cross section, and that the formal upper limit of energy loss can be extended to infinity. With these simplifications, Landau found that the energy loss distribution can be expressed as

$$f(s,\Delta)\mathrm{d}\Delta = \phi(\lambda)\mathrm{d}\lambda \tag{2.151}$$

in terms of $\phi(\lambda)$, a universal function of a single scaled variable

$$\lambda = \frac{\Delta}{\xi} - \ln\left[\frac{2\xi mv^2}{(1-\beta^2)I^2}\right] + \delta + \beta^2 - 1 + \gamma. \tag{2.152}$$

Here m and v are the mass and speed of the electron, δ is the density effect correction, β is v/c, I is the mean excitation energy of the medium, and γ is Euler's constant ($\gamma = 0.5772157\ldots$) . The parameter ξ is defined by

$$\xi = \frac{2\pi e^4 NZ}{mv^2}s, \tag{2.153}$$

where e is the charge of the electron and NZ is the number density of atomic electrons, and the universal function is

$$\phi(\lambda) = \frac{1}{2\pi i}\int_{x-i\infty}^{x+i\infty} \exp(\mu\ln(\mu) + \lambda\mu)\mathrm{d}\mu, \tag{2.154}$$

where x is a positive real number specifying the line of integration.

For purposes of sampling, $\phi(\lambda)$ is negligible for $\lambda < -4$, so that this range is ignored. Börsch-Supan [111] originally tabulated $\phi(\lambda)$ in the range $-4 \le \lambda \le 100$, and derived for the range $\lambda > 100$ the asymptotic form

$$\phi(\lambda) \approx \frac{1}{w^2+\pi^2}, \tag{2.155}$$

in terms of the auxiliary variable w, where

$$\lambda = w + \ln(w) + \gamma - \frac{3}{2}. \tag{2.156}$$

Recent extensions [57] of Börsch-Supan's tabulation have provided a representation of the function in the range $-4 \le \lambda \le 100$ in the form of five thousand equally probable bins in λ. In the MCNP code, the boundaries of these bins are saved in the array **`eqlm(mlam)`**, where **`mlam`** = 5001. Sampling from this tabular distribution accounts for approximately 98.96% of the cumulative probability for $\phi(\lambda)$. For the remaining large-λ tail of the distribution, the MCNP code uses the approximate form $\phi(\lambda) \approx w$, which is easier to sample than $(w^2+\pi^2)^{-1}$, but is still quite accurate for $\lambda > 100$.

Blunck and Leisegang [100] have extended Landau's result to include the second moment of the expansion of the cross section. Their result can be expressed as a convolution of Landau's distribution with a Gaussian distribution:

$$f^*(s,\Delta) = \frac{1}{\sqrt{2\pi\sigma}}\int_{-\infty}^{\infty} f(s,\Delta')\exp\left[\frac{(\Delta-\Delta')^2}{2\sigma^2}\right]\mathrm{d}\Delta'. \tag{2.157}$$

Blunck and Westphal [112] provided a simple form for the variance of the Gaussian:

$$\sigma_{\mathrm{BW}}^2 = 10 \text{ eV} \cdot Z^{4/3}\overline{\Delta}. \tag{2.158}$$

Subsequently, Chechin and Ermilova [113] investigated the Landau/Blunck-Leisegang theory, and derived an estimate for the relative error

$$\epsilon_{\mathrm{CE}} \approx \left[\frac{10\xi}{I}\left(1+\frac{\xi}{10I}\right)^3\right]^{-1/2}, \tag{2.159}$$

caused by the neglect of higher-order moments. Based on this work, Seltzer [114] describes and recommends a correction to the Blunck-Westphal variance as

$$\sigma = \frac{\sigma_{\mathrm{BW}}}{1+3\epsilon_{\mathrm{CE}}}. \tag{2.160}$$

This value for the variance of the Gaussian is used in the MCNP code.

Examination of the asymptotic form for $\phi(\lambda)$ shows that unrestricted sampling of λ will not result in a finite mean energy loss. Therefore, a material- and energy-dependent cutoff λ_{c} is imposed on the sampling of λ. In the initiation phase of an MCNP calculation, the code makes use of two preset arrays, **`flam(mlanc)`** and **`avlm(mlanc)`**, with **`mlanc`** = 1591. The array **`flam`** contains candidate values for λ_{c} in the range $-4 \le \lambda_{\mathrm{c}} \le 50000$; the array **`avlm`** contains the corresponding expected mean values for the sampling of λ. For each material and electron energy, the code uses the known mean collisional energy loss $\overline{\Delta}$, interpolating in this tabular function to select a suitable value for λ_{c}, which is then stored in the dynamically allocated array **`flc`**. During the transport phase of the calculation, the value of **`flc`** applicable to the current material and electron energy is used as an upper limit, and any sampled value of λ greater than the limit is rejected. In this way, the correct mean energy loss is preserved.

2.4.5.6 Logic for Sampling Energy Straggling

The Landau theory described in the previous section provides an energy-loss distribution determined by the energy E of the electron, the path-length s to be traversed, and the properties of the material. Let us symbolize a sampling of this distribution as an application of a straggling operator $L(E, s, \Delta)$ that provides a sampled value of the energy loss Δ. In the MCNP code earlier than version 5.1.40, all parameters needed for sampling straggling were precomputed and associated with the standard energy boundaries E_n and the corresponding ranges s_n. In effect the code was restricted to calculations based on discrete arguments of the operator $L\left(E_n, s_n, \overline{\Delta}_n\right)$. As a result, the proper assignment of an electron transport step to an energy group n required a rather subtle logic. Eventually, two algorithms for apportioning straggled energy loss to electron substeps were made available. With MCNP code version 5.1.40, a third algorithm is provided, as discussed in §2.4.5.6.3.

2.4.5.6.1 Energy Indexing Algorithm in the MCNP Code

The first energy indexing algorithm (also called the "bin-centered" treatment) developed for the MCNP code is arguably the less successful of the two existing algorithms, but for historical reasons remains the default option. It was an attempt to keep the electron substeps aligned as closely as possible with the energy groups that were used for their straggling samples. A simplified description of the MCNP algorithm is as follows. An electron of energy E is assigned to the group n such that $E_n > E \ge E_{n+1}$. A straggled energy loss Δ is sampled from $L\left(E_n, s_n, \overline{\Delta}_n\right)$. The electron attempts to traverse m substeps, each of which is assigned the energy loss Δ/m. If m substeps are completed, the process starts over with the assignment of a new energy group. However, if the electron crosses a cell boundary, or if the electron energy falls below the current group,

the loop over m is abandoned, even if fewer than m substeps have been completed, and the energy group is reassigned.

Since the straggling parameters are pre-computed at the midpoints of the energy groups, this algorithm does succeed in assigning to each substep a straggled energy loss based on parameters that are as close as possible to the beginning energy of the substep. However, there are two problems with the current MCNP approach. First, there is a high probability that the electron will not actually complete the expected range s_n for which the energy loss was sampled, in which case the energy loss relies on a linear interpolation in a theory that is clearly nonlinear. Second, the final substep of each sequence using the sampled energy loss from $L\left(E_n, s_n, \overline{\Delta}_n\right)$ will frequently fall partially in the next-lower energy group $n+1$, but no substep using the sample from $L\left(E_n, s_n, \overline{\Delta}_n\right)$ will ever be partially in the higher group n–1.

⚠ Caution

This results in a small, but potentially significant, systematic error.

See for example the investigations of Schaart et al. [115] and references therein.

2.4.5.6.2 Energy Indexing Algorithm in the ITS Code

Developed for the ITS codes earlier than the MCNP algorithm, this method (also called the "nearest-group-boundary" treatment) was added to the MCNP code in order to explore some of the energy-dependent artifacts of the condensed history approach, and in order to offer more consistency with the TIGER Series codes. This algorithm differs from the default treatment in two ways. First, the electron is initially assigned to a group n such that

$$(E_{n-1} + E_n)/2 > E \geq (E_n + E_{n+1})/2. \tag{2.161}$$

In other words, the electron is assigned to the group whose upper limit is closest to the electron's energy. Second, although the electron will be reassigned when it enters a new geometric cell, it will not be reassigned merely for falling out of the current energy group. These differences serve to reduce the number of times that unwanted imposition of linear interpolation on partial steps occurs, and to allow more equal numbers of excursions above and below the energy group from which the Landau sampling was made. As [115] shows, these advantages make the ITS algorithm a more accurate representation of the energy loss process, as indicated in comparisons with reference calculations and experiments. Nevertheless, although the reliance on linear interpolation and the systematic errors are reduced, neither is completely eliminated. It is straightforward to create example calculations that show unphysical artifacts in the ITS algorithm as well as in the MCNP logic.

The "nearest-group-boundary" treatment is selected by setting the 18th entry of the `DBCN` card to 1. For example, the card "`DBCN 17J 1`" selects this straggling logic without affecting any of the other `DBCN` options.

2.4.5.6.3 New Energy- and Step-specific Method

It is easy to express what we would like to see in the straggling logic. For an electron with energy E about to traverse a step of length s, we would like to sample the straggling from the operator $L\left(E, s, \overline{\Delta}\right)$ without regard to the prearranged energy boundaries E_n. In the MCNP code, version 5.1.40, we have now brought this situation about. A new Fortran 90 module has been installed to deal with straggling data. Those parameters that are separate from the individual straggling events are still precomputed, but each electron transport step can now sample its energy loss separately from adjacent steps, and specifically for its current energy and planned step length. Using this approach, we largely eliminate the linear interpolations and energy misalignments of the earlier algorithms and obviate the need for a choice of energy group. As of the MCNP

code, version 5.1.40, the new straggling logic is included in the code, but is still being tested. Preliminary results [116] indicate that a more accurate and stable estimate of the straggling is obtained, and a variety of unphysical artifacts are eliminated.

The new logic is selected by setting the 18th entry of the DBCN card to 2, for example with the card "DBCN 17J 2".

2.4.5.7 Angular Deflections

The ETRAN codes and the MCNP code rely on the Goudsmit-Saunderson [98] theory for the probability distribution of angular deflections. The angular deflection of the electron is sampled once per substep according to the distribution

$$F(s,\mu) = \sum_{l=0}^{\infty} \left(l + \frac{1}{2}\right) \exp(-sG_l) P_l(\mu), \tag{2.162}$$

where s is the length of the substep, $\mu = \cos(\theta)$ is the angular deflection from the direction at the beginning of the substep, $P_l(\mu)$ is the l^{th} Legendre polynomial, and G_l is

$$G_l = 2\pi N \int_{-1}^{1} \frac{\mathrm{d}\sigma}{\mathrm{d}\Omega} [1 - P_l(\mu)] \mathrm{d}\mu, \tag{2.163}$$

in terms of the microscopic cross section $\mathrm{d}\sigma/\mathrm{d}\Omega$, and the atom density N of the medium.

For electrons with energies below 0.256 MeV, the microscopic cross section is taken from numerical tabulations developed from the work of Riley [117]. For higher-energy electrons, the microscopic cross section is approximated as a combination of the Mott [118] and Rutherford [110] cross sections, with a screening correction. Seltzer [102] presents this "factored cross section" in the form

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega} = \frac{Z^2 e^2}{p^2 v^2 (1 - \mu + 2\eta)^2} \left[\frac{(\mathrm{d}\sigma/\mathrm{d}\Omega)_{\text{Mott}}}{(\mathrm{d}\sigma/\mathrm{d}\Omega)_{\text{Rutherford}}} \right], \tag{2.164}$$

where e, p, and v are the charge, momentum, and speed of the electron, respectively. The screening correction η was originally given by Molière [119] as

$$\eta = \frac{1}{4} \left(\frac{\alpha mc}{0.885p} \right)^2 Z^{2/3} \left[1.13 + 3.76 \left(\frac{\alpha Z}{\beta} \right)^2 \right], \tag{2.165}$$

where α is the fine structure constant, m is the rest mass of the electron, and $\beta = v/c$. The MCNP code now follows the recommendation of Seltzer [102], and the implementation in the Integrated TIGER Series, by using the slightly modified form

$$\eta = \frac{1}{4} \left(\frac{\alpha mc}{0.885p} \right)^2 Z^{2/3} \left[1.13 + 3.76 \left(\frac{\alpha Z}{\beta} \right)^2 \sqrt{\frac{\tau}{\tau + 1}} \right], \tag{2.166}$$

where τ is the electron energy in units of electron rest mass. The multiplicative factor in the final term is an empirical correction which improves the agreement at low energies between the factored cross section and the more accurate partial-wave cross sections of Riley.

2.4.5.8 Bremsstrahlung

When using data from the **el** library, for the sampling of bremsstrahlung photons, the MCNP code relies primarily on the Bethe-Heitler [120] Born-approximation results that have been used until rather recently [107] in ETRAN. A comprehensive review of bremsstrahlung formulas and approximations relevant to the present level of the theory in the MCNP code can be found in the paper of Koch and Motz [121]. Particular prescriptions appropriate to Monte Carlo calculations have been developed by Berger and Seltzer [122]. For the ETRAN-based codes, this body of data has been converted to tables including bremsstrahlung production probabilities, photon energy distributions, and photon angular distributions.

For data tables on the **el03** library, the production cross section for bremsstrahlung photons and energy spectra are from the evaluation by Seltzer and Berger [107–109]. The evaluation uses detailed calculations of the electron-nucleus bremsstrahlung cross section for electrons with energies below 2 MeV and above 50 MeV. The evaluation below 2 MeV uses the results of Pratt, Tseng, and collaborators, based on numerical phase-shift calculations [123–126]. For 50 MeV and above, the analytical theory of Davies, Bethe, Maximom, and Olsen [127, 128] is used and is supplemented by the Elwert-Coulomb [129] correction factor and the theory of the high-frequency limit or tip region given by Jabbur and Pratt [130, 131]. Screening effects are accounted for by the use of Hartree-Fock atomic form factors [86, 132]. The values between these firmly grounded theoretical limits are found by a cubic-spline interpolation as described in [107, 108]. Seltzer reports good agreement between interpolated values and those calculated by Tseng and Pratt [133] for 5- and 10-MeV electrons in aluminum and uranium. Electron-electron bremsstrahlung is also included in the cross-section evaluation based on the theory of Haug [134] with screening corrections derived from Hartree-Fock incoherent scattering factors [86, 132]. The energy spectra for the bremsstrahlung photons are provided in the evaluation. No major changes were made to the tabular angular distributions, which are internally calculated when using the **el** library, except to make finer energy bins over which the distribution is calculated.

The MCNP code addresses the sampling of bremsstrahlung photons at each electron substep. The tables of production probabilities are used to determine whether a bremsstrahlung photon will be created. For data from the **el03** library, the bremsstrahlung production is sampled according to a Poisson distribution along the step so that none, one or more photons could be produced; the **el** library allows for either none or one bremsstrahlung photon in a substep. If a photon is produced, the new photon energy is sampled from the energy distribution tables. By default, the angular deflection of the photon from the direction of the electron is also sampled from the tabular data. The direction of the electron is unaffected by the generation of the photon because the angular deflection of the electron is controlled by the multiple scattering theory. However, the energy of the electron at the end of the substep is reduced by the energy of the sampled photon because the treatment of electron energy loss, with or without straggling, is based only on non-radiative processes.

There is an alternative to the use of tabular data for the angular distribution of bremsstrahlung photons. If the fourth entry on the PHYS:e card is 1, then the simple, material-independent probability distribution

$$p(\mu)\mathrm{d}\mu = \frac{1-\beta^2}{2(1-\beta\mu)^2}\mathrm{d}\mu, \tag{2.167}$$

where $\mu = \cos(\theta)$ and $\beta = v/c$, will be used to sample for the angle of the photon relative to the direction of the electron according to the formula

$$\mu = \frac{2\xi - 1 - \beta}{2\xi\beta - 1 - \beta}, \tag{2.168}$$

where ξ is a random number drawn from the unit interval. This sampling method is of interest only in the context of detectors and DXTRAN spheres. A set of source contribution probabilities $p(\mu)$ consistent with the tabular data is not available. Therefore, detector and DXTRAN source contributions are made using Eq. (2.167). Specifying that the generation of bremsstrahlung photons rely on Eq. (2.167) allows the user to force the actual transport to be consistent with the source contributions to detectors and DXTRAN.

2.4.5.9 K-shell Electron Impact Ionization and Auger Transitions

Data tables in the **el03** library use the same K-shell impact ionization calculation (based upon ITS1.0) as data tables on the **el** library, except for how the emission of relaxation photons is treated; the **el03** evaluation model has been modified to be consistent with the photo-ionization relaxation model. In the **el** evaluation, a K-shell impact ionization event generated a photon with the average K-shell energy. The **el03** evaluation generates photons with energies given by Everett and Cashwell [50]. Both **el03** and **el** treatments only take into account the highest Z component of a material. Thus inclusion of trace high Z impurities could mask K-shell impact ionization from other dominant components.

Auger transitions are handled the same for data tables from the **el03** and **el** libraries. If an atom has undergone an ionizing transition and can undergo a relaxation, if it does not emit a photon it will emit an Auger electron. The difference between **el** and **el03** is the energy with which an Auger electron is emitted, given by $E_A = E_{\overline{K}}$ or $E_A = E_{\overline{K}} - 2E_{\overline{L}}$ for **el** or **el03**, respectively. The **el** value is that of the highest energy Auger electron while the **el03** value is the energy of the most probable Auger electron. It should be noted that both models are somewhat crude.

2.4.5.10 Knock-on Electrons

The Møller cross section [135] for scattering of an electron by an electron is

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\epsilon} = \frac{C}{E}\left\{\frac{1}{\epsilon^2} + \frac{1}{(1-\epsilon)^2} + \left(\frac{\tau}{\tau+1}\right)^2 - \frac{2\tau+1}{(\tau+1)^2}\frac{1}{\epsilon(1-\epsilon)}\right\}, \tag{2.169}$$

where ϵ, τ, E, and C have the same meanings as in Eqs. (2.142)–(2.145). When calculating stopping powers, one is interested in all possible energy transfers. However, for the sampling of transportable secondary particles, one wants the probability of energy transfers greater than some ϵ_c representing an energy cutoff, below which secondary particles will not be followed. This probability can be written

$$\sigma(\epsilon_c) = \int_{\epsilon_c}^{1/2} \frac{\mathrm{d}\sigma}{\mathrm{d}\epsilon}\mathrm{d}\epsilon. \tag{2.170}$$

The reason for the upper limit of 1/2 is the same as in the discussion of Eq. (2.145). Explicit integration of Eq. (2.169) leads to

$$\sigma(\epsilon_c) = \frac{C}{E}\left\{\frac{1}{\epsilon_c} + \frac{1}{1-\epsilon_c} + \left(\frac{\tau}{\tau+1}\right)^2\left(\frac{1}{2}-\epsilon_c\right) - \frac{2\tau+1}{(\tau+1)^2}\ln\left(\frac{1-\epsilon_c}{\epsilon_c}\right)\right\}. \tag{2.171}$$

Then the normalized probability distribution for the generation of secondary electrons with $\epsilon > \epsilon_c$ is given by

$$g(\epsilon,\epsilon_c)\mathrm{d}\epsilon = \frac{1}{\sigma(\epsilon_c)}\frac{\mathrm{d}\sigma}{\mathrm{d}\epsilon}\mathrm{d}\epsilon. \tag{2.172}$$

At each electron substep, the MCNP code uses $\sigma(\epsilon_c)$ to determine randomly whether knock-on electrons will be generated. If so, the distribution of Eq. (2.172) is used to sample the energy of each secondary electron. Once an energy has been sampled, the angle between the primary direction and the direction of the newly generated secondary particle is determined by momentum conservation. This angular deflection is used for the subsequent transport of the secondary electron. However, neither the energy nor the direction of the primary electron is altered by the sampling of the secondary particle. On the average, both the energy loss and the angular deflection of the primary electron have been taken into account by the multiple scattering theories.

Table 2.2: Tally Quantities Scored.

Tally	Description	Score	Physical Quantity	Units
F1	surface current	W	$J = \int \mathrm{d}E \int \mathrm{d}t \int \mathrm{d}A \int \mathrm{d}\Omega \lvert\mathbf{\Omega}\cdot\mathbf{n}\rvert \psi(\mathbf{r},\mathbf{\Omega},E,t)$	particles
F2	surface fluence	$\frac{W}{\lvert\mu\rvert A}$	$\overline{\phi}_{\mathrm{S}} = \frac{1}{A}\int \mathrm{d}E \int \mathrm{d}t \int \mathrm{d}A \int \mathrm{d}\Omega\, \psi(\mathbf{r},\mathbf{\Omega},E,t)$	particles/cm^2
F4	cell fluence	$W\frac{T_l}{V}$	$\overline{\phi}_{\mathrm{V}} = \frac{1}{V}\int \mathrm{d}E \int \mathrm{d}t \int \mathrm{d}V \int \mathrm{d}\Omega\, \psi(\mathbf{r},\mathbf{\Omega},E,t)$	particles/cm^2
F5	detector fluence	$\frac{W\cdot p(\mathbf{\Omega}_{\mathrm{P}})\exp(-\lambda)}{L^2}$	$\phi_{\mathrm{P}} = \int \mathrm{d}E \int \mathrm{d}t \int \mathrm{d}\Omega\, \psi(\mathbf{r}_{\mathrm{P}},\mathbf{\Omega},E,t)$	particles/cm^2
F6	energy deposition	$WT_l\sigma_{\mathrm{t}}(E)H(E)\frac{\rho_{\mathrm{a}}}{m}$	$H_{\mathrm{t}} = \frac{\rho_{\mathrm{a}}}{m}\int \mathrm{d}E \int \mathrm{d}t \int \mathrm{d}V \int \mathrm{d}\Omega\, \sigma_{\mathrm{t}}(E)H(E)\psi(\mathbf{r},\mathbf{\Omega},E,t)$	MeV/g
F7	fission-energy deposition	$WT_l\sigma_{\mathrm{f}}(E)Q\frac{\rho_{\mathrm{a}}}{m}$	$H_{\mathrm{f}} = \frac{\rho_{\mathrm{a}}}{m}Q\int \mathrm{d}E \int \mathrm{d}t \int \mathrm{d}V \int \mathrm{d}\Omega\, \sigma_{\mathrm{f}}(E)\psi(\mathbf{r},\mathbf{\Omega},E,t)$	MeV/g
F8	pulse-height tally	W_C put in bin E_D	pulses	pulses

2.4.5.11 Multigroup Boltzmann-Fokker-Planck Electron Transport

The electron physics described above can be implemented into a multigroup form using a hybrid multigroup and continuous-energy method for solving the Boltzmann-Fokker-Planck equation as described by Morel [60]. The multigroup formalism for performing charged particle transport was pioneered by Morel and Lorence [61–63] for use in deterministic transport codes. With a first-order treatment for the continuous slowing down approximation (CSDA) operator, this formalism is equally applicable to a standard Monte Carlo multigroup transport code as discussed by Sloan [136]. Unfortunately, a first-order treatment is not adequate for many applications. Morel, et al. have addressed this difficulty by developing a hybrid multigroup/continuous energy algorithm for charged particles that retains the standard multigroup treatment for large-angle scattering, but treats exactly the CSDA operator. As with standard multigroup algorithms, adjoint calculations are performed readily with the hybrid scheme.

The process for performing an MCNP/MGBFP calculation for electron/photon transport problems involves executing three codes. First the CEPXS [61–63] code is used to generate coupled electron-photon multigroup cross sections. Next the CRSRD code casts these cross sections into a form suitable for use in the MCNP code by adjusting the discrete ordinate moments into a Radau quadrature form that can be used by a Monte Carlo code. CRSRD also generates a set of multigroup response functions for dose or charge deposition that can be used for response estimates for a forward calculation or for sources in an adjoint calculation. Finally, the MCNP code is executed using these adjusted multigroup cross sections. Some applications of this capability for electron/photon transport have been presented in [137].

2.5 Tallies

The MCNP code automatically creates standard summary information that gives the user a better insight into the physics of the problem and the adequacy of the Monte Carlo simulation including: a complete accounting of the creation and loss of all tracks and their energy; the number of tracks entering and reentering a cell plus the track population in the cell; the number of collisions in a cell; the average weight, mean free path, and energy of tracks in a cell; the activity of each nuclide in a cell (that is, how particles interacted with each nuclide, not the radioactivity); and a complete weight balance for each cell.

The MCNP code also provides seven standard tally types that can be specified in an MCNP input file by using F cards (see §5.9 for the tally type specification). These tallies are normalized to be per source particle unless a different normalization has been specified with the WGT keyword on the SDEF card, changed by the user with a **TALLYX** subroutine, and by weight in a criticality (KCODE) calculation. The MCNP tally plotter provides graphical displays of the results (see §6.3). The seven standard tally quantities actually scored in the MCNP code before the final normalization are presented in Table 2.2. The table also gives the physical quantity that corresponds to each tally, and it defines much of the notation used in the remainder of this section. The F2, F4, and F5 tallies in Table 2.2 are described as fluence tallies with the associated units. However, depending

on the source units, these may also be fluence rate (i.e., flux) tallies with units of particles/$\left(\text{cm}^2 \cdot \text{s}\right)$. For Table 2.2, the variables used are

W	particle weight,
W_{C}	collective weight from a history for pulse-height tally [§2.5.5],
$\mathbf{r}, \mathbf{\Omega}, E, t$	particle position vector (cm), direction unit vector, energy (MeV), and time (shakes, sh; $1\,\text{sh} = 10^{-8}\,\text{s}$),
μ	$\mathbf{\Omega} \cdot \mathbf{n}$, cosine of angle between surface normal $\mathbf{n}$ and particle trajectory $\mathbf{\Omega}$,
A, V	surface area (cm^2) and volume (cm^3), calculated by the code or input by the user,
T_l	track length (cm), event transit time multiplied by the particle velocity,
$p(\mathbf{\Omega}_{\text{P}})$	probability density function for scattering (or starting) in the direction $\mathbf{\Omega}_{\text{P}}$ towards the point detector (azimuthal symmetry is assumed),
λ	total number of mean free paths from particle location to detector (i.e., the optical distance),
L	distance to detector from the source or collision event (cm),
$\sigma_{\text{t}}(E)$	microscopic total cross section (barns),
$\sigma_{\text{f}}(E)$	microscopic fission cross section (barns),
$H(E)$	heating number (MeV/collision),
E_D	total energy deposited by a history in a detector (MeV); see [§2.5.5],
ρ_{a}	atom density (atoms/barn-cm),
ρ_{g}	mass density (g/cm^3); not used in Table 2.2 but used later in this chapter,
m	cell mass (g),
Q	total prompt energy release per fission (MeV),
ψ	angular flux as typically defined in nuclear reactor theory [78, 138]; $\psi(\mathbf{r}, \mathbf{\Omega}, E, t) = vn(\mathbf{r}, \mathbf{\Omega}, E, t)$, where n is the particle density (particles/cm^3/MeV/steradian) and v is the velocity (cm/sh), so the units of ψ are particles/cm^2/sh/MeV/steradian,
J	total (not net) current crossing a surface,
$\overline{\phi}_{\text{S}}$	average flux on a surface,
$\overline{\phi}_{\text{V}}$	average flux in a cell (i.e., in a volume),
ϕ_{P}	flux at a point,
$\mathbf{r}_{\text{P}}$	point at which ϕ_{P} is estimated (i.e., the location of the point detector),
H_{t}	total energy deposition in a cell (MeV/g),
H_{f}	total fission energy deposition in a cell (MeV/g).

Table 2.3: Tallies Modified with an Asterisk or Plus.

Tally	Scores	Units
*F1	WE	MeV
*F2	$\frac{WE}{\lvert\mu\rvert A}$	MeV/cm^2
*F4	$W\frac{T_l E}{V}$	MeV/cm^2
*F5	$\frac{W\cdot p(\mathbf{\Omega}_\mathrm{P})\exp(-\lambda)E}{L^2}$	MeV/cm^2
*F6	$1.60219\times10^{-22}\frac{\text{jerks}}{\text{MeV}}WT_l\sigma_\mathrm{t}(E)H(E)\frac{\rho_\mathrm{a}}{m}$	jerks/g
+F6	total energy deposition from all particles	MeV/g
*F7	$1.60219\times10^{-22}\frac{\text{jerks}}{\text{MeV}}WT_l\sigma_\mathrm{f}(E)Q\frac{\rho_\mathrm{a}}{m}$	jerks/g
*F8	$E_D\times W_C$ put in bin E_D	MeV
+F8	$\pm W_C$ put in bin E_D	charge

The units of each tally are derived from the units of the source. If the source has units of particles per unit time, current tallies are particles per unit time and flux tallies are particles per unit time per unit area. When the source has units of particles, current tallies have units of particles and flux tallies actually represent fluences with units of particles per unit area. A steady-state flux solution can be obtained by having a source with units of particles per unit time and integrating the tally over all time (that is, omitting the `T`n card). The average flux in a time bin can be obtained from the fluence tally for a time-dependent source by dividing the tally by the time bin width in shakes. These tallies can all be made per unit energy by dividing each energy bin by the energy bin width.

Adding an asterisk (*`F`n) changes the units into an energy tally and multiplies each tally as indicated in Table 2.3. For an `F8` pulse height tally, the asterisk changes the tally from deposition of pulses to an energy deposition tally. A plus sign can only be used with `F6` and `F8` cards. A +`F6` tally is a total energy position tally from all particles (2.5.3) and a +`F8` tally is a charge deposition tally.

Extensive statistical analysis of tally convergence is applied to the tally fluctuation bin of each tally [§5.9.19]. Ten statistical checks are made, including the variance of the variance and the Pareto slope of the history score probability density function. These checks are described in §2.6.

In addition to the standard tallies, the MCNP code has superimposed mesh tallies. This feature allows the user to tally particles on a mesh independent of the problem geometry. Track-length quantities such as fluence, heating, energy deposition, point-detector and DXTRAN sphere contribution rays or other data such as source points can be calculated. Mesh tallies are invoked by using the `FMESH` and `TMESH` cards. When a track-length quantity is computed over the mesh tally cells, it is typically normalized to be per starting particle, except in `KCODE` criticality calculations.

Not all features of the standard tallies have been implemented in the mesh tallies. For example, no tally fluctuation statistics are given for mesh tallies; the only error information provided is the relative error for each mesh cell. Features that can be used with the mesh tallies are multiplying the result by the particle energy (*`FMESH` card), dose functions, and tally multipliers. Time binning is not a feature of the `TMESH` tally.

The definitions of the current and flux in the sections that follow come from nuclear reactor theory [78, 138] but are related to similar quantities in radiative transfer theory [139, 140]. The MCNP angular flux multiplied by the particle energy is the same as the intensity in radiative transfer theory. The MCNP total flux at energy E multiplied by the particle energy E equals the integrated energy density times the speed of light in radiative transfer theory. The MCNP current multiplied by the particle energy is analogous to the radiative flux crossing an area in radiative transfer theory. The MCNP particle fluence multiplied by the particle energy is the same as the fluence in radiative transfer theory.

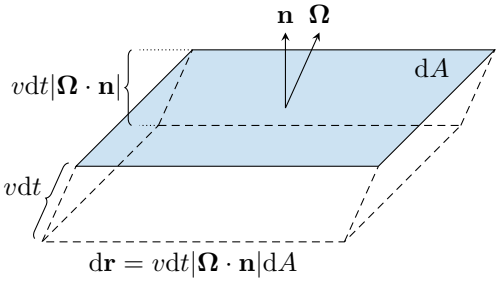


Figure 2.7: Diagram for description of the surface current tally.

Nuclear reactor theory has given the terms flux and current quite different meanings [78, 138] than they have in other branches of physics; terminology from other fields should not be confused with that used in this manual.

Rigorous mathematical derivations of the basic tallies are given in [141]. Somewhat heuristic derivations follow. Note that the surface current is a total but the cell and surface fluxes are averages.

2.5.1 Surface Current Tally

The surface current (F1) tally is a simple count of the number of particles, represented by the Monte Carlo weight, crossing a surface in specified bins as illustrated in Figure 2.7. The number of particles at time t, in a volume element $\mathrm{d}\mathbf{r}$, with directions within $\mathrm{d}\Omega$, and energies within $\mathrm{d}E$ is $n(\mathbf{r},\mathbf{\Omega},E,t)\mathrm{d}\mathbf{r}\mathrm{d}\Omega\mathrm{d}E$. Let the volume element $\mathrm{d}\mathbf{r}$ contain the surface element $\mathrm{d}A$ (with surface normal $\mathbf{n}$) and along $\mathbf{\Omega}$ for a distance $v\mathrm{d}t$, as depicted in Figure 2.7. Then the differential volume element is $\mathrm{d}\mathbf{r} = v\mathrm{d}t|\mathbf{\Omega}\cdot\mathbf{n}|\mathrm{d}A$. All the particles within this volume element (with directions within $\mathrm{d}\Omega$ and energies within $\mathrm{d}E$) will cross surface $\mathrm{d}A$ in time $\mathrm{d}t$. Thus, the number of particles crossing surface $\mathrm{d}A$ in time $\mathrm{d}t$ is $|\mathbf{\Omega}\cdot\mathbf{n}|vn(\mathbf{r},\mathbf{\Omega},E,t)\mathrm{d}\Omega\mathrm{d}E\mathrm{d}t\mathrm{d}A$. The number of particles crossing surface A in energy bin i, time bin j, and angle bin k is thus

$$\int_{E_i}\mathrm{d}E\int_{t_j}\mathrm{d}t\int_{\Omega_k}\mathrm{d}\Omega\int\mathrm{d}A|\mathbf{\Omega}\cdot\mathbf{n}|vn(\mathbf{r},\mathbf{\Omega},E,t). \tag{2.173}$$

The range of integration over energy, time, and angle (cosine) is controlled by E, T, and C cards. If the range of integration is over all angles (no C card), then the surface current tally is a count of the number of particles with any trajectory crossing the surface (in each energy and time bin) and thus has no direction associated with it.

Note that the MCNP current J of Table 2.2 is the total current and not the net current. It is the total number of particles crossing a surface. Frequently, the net current, rather than the total current, is desired. Defining the partial currents crossing in the positive and negative directions ("right" and "left" or "up" and "down") as [138]

$$J_\pm = \int\mathrm{d}E\int\mathrm{d}t\int\mathrm{d}A\int_{\substack{\mathbf{\Omega}\cdot\mathbf{n}>0\\ \mathbf{\Omega}\cdot\mathbf{n}<0}}\mathrm{d}\Omega|\mathbf{\Omega}\cdot\mathbf{n}|\psi(\mathbf{r},\mathbf{\Omega},E,t), \tag{2.174}$$

where the net current across the surface is $J_{\mathrm{net}} = J_+ - J_-$. The total current of Table 2.2 is $J_{\mathrm{net}} = J_+ + J_-$. The partial currents $J_\pm$ across a surface can be calculated in the MCNP code using the surface current tally with two cosine bins, one each for $-1 \le \mu < 0$ and $0 < \mu \le 1$.

The units of the surface current tally are those of the source. If the source has units of particles per unit time, the tally has units of particles per unit time. When the source has units of particles, the tally has units of particles. The SD card can be used to input a constant that divides the tally. In other words, if x is input on the SD card, the tally will be divided by x.

2.5.2 Flux Tallies

Defining the scalar flux as $\phi(\mathbf{r},E,t) \equiv \int \mathrm{d}\Omega\, \psi(\mathbf{r},\mathbf{\Omega},E,t)$ where $\phi(\mathbf{r},E,t)\mathrm{d}\mathbf{r}\mathrm{d}E$ is the total scalar flux in volume element $\mathrm{d}\mathbf{r}$ about $\mathbf{r}$ and energy element $\mathrm{d}E$ about E and, introducing energy and time bins, the integrals of Table 2.2 for the surface flux (F2), cell flux (F4), and detector flux (F5) tallies can be recast as

$$\mathrm{F2} = \frac{1}{A}\int_{E_i}\mathrm{d}E\int_{t_j}\mathrm{d}t\int\mathrm{d}A\,\phi(\mathbf{r},E,t), \tag{2.175a}$$

$$\mathrm{F4} = \frac{1}{V}\int_{E_i}\mathrm{d}E\int_{t_j}\mathrm{d}t\int\mathrm{d}V\,\phi(\mathbf{r},E,t), \tag{2.175b}$$

$$\mathrm{F5} = \int_{E_i}\mathrm{d}E\int_{t_j}\mathrm{d}t\,\phi(\mathbf{r}_\mathrm{P},E,t). \tag{2.175c}$$

The range of integration over energy and time can be tailored by E and T cards. If no E card is present, the integration limits are the same as the limits for the corresponding cross sections used. The cell flux and surface flux tallies are discussed in this section. The detector flux tally is discussed in §2.5.6.

2.5.2.1 Track-length Estimate of Cell Flux

The average particle flux in a cell (from Table 2.2) can be written

$$\begin{aligned}\overline{\phi}_\mathrm{V} &= \frac{1}{V}\int\mathrm{d}E\int\mathrm{d}t\int\mathrm{d}V\int\mathrm{d}\Omega\,\psi(\mathbf{r},\mathbf{\Omega},E,t)\\ &= \frac{1}{V}\int\mathrm{d}E\int\mathrm{d}V\int\mathrm{d}\Omega\int\mathrm{d}t\,vn(\mathbf{r},\mathbf{\Omega},E,t)\\ &= \frac{1}{V}\int\mathrm{d}E\int\mathrm{d}V\int\mathrm{d}t\,vN(\mathbf{r},E,t),\end{aligned} \tag{2.176}$$

where $N(\mathbf{r},E,t) = \int\mathrm{d}\Omega\,n(\mathbf{r},\mathbf{\Omega},E,t)$ is the density of particles, regardless of their trajectories, at a point. Defining $\mathrm{d}s$ to be the differential unit of track length and noting that $\mathrm{d}s = v\mathrm{d}t$ yields

$$\overline{\phi}_\mathrm{V} = \frac{1}{V}\int\mathrm{d}E\int\mathrm{d}V\int\mathrm{d}s\,N(\mathbf{r},E,t). \tag{2.177}$$

The quantity $N(\mathbf{r},E,t)\mathrm{d}s$ may be thought of as a track-length density; thus, the average flux can be estimated by summing track lengths. The MCNP code estimates $\overline{\phi}_\mathrm{V}$ by summing WT_l/V for all particle tracks in the cell. Time- and energy-dependent subdivisions of $\overline{\phi}_\mathrm{V}$ are made by binning the track lengths in appropriate time and energy bins. The track length estimator is generally quite reliable because there are frequently many tracks in a cell (compared to the number of collisions), leading to many contributions to this tally.

The SD card can be used to input a new volume that divides the tally. In other words, if V' is input on the SD card, the tally will be divided by V' instead of V. See the SD card information on how the MCNP code can handle the volumes used to compute the tallies.

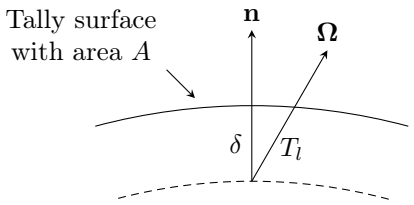


Figure 2.8: Diagram for description of the surface flux tally.

2.5.2.2 Surface Flux

The average particle scalar flux on a surface ($\overline{\phi}_{\mathrm{S}}$ of Table 2.2) is estimated using a surface crossing estimator that may be thought of as the limiting case of the cell flux or track length estimator when the cell becomes infinitely thin, as illustrated in Figure 2.8.

As the cell thickness δ approaches zero, the cell volume approaches $A\delta$ and the track length through the cell approaches $\delta/|\mathbf{\Omega}\cdot\mathbf{n}|$. Thus,

$$\begin{aligned}\overline{\phi}_{\mathrm{S}} &= \lim_{\delta\to 0}\overline{\phi}_{\mathrm{V}}\\ &= \lim_{\delta\to 0}\frac{WT_l}{V}\\ &= \lim_{\delta\to 0}\frac{W\delta}{A\delta|\mathbf{\Omega}\cdot\mathbf{n}|}\\ &= \frac{W}{A|\mu|}.\end{aligned}$$

A more formal derivation of the surface flux estimator may be found in [141].

For particles grazing the surface, $1/|\mu|$ is very large and the MCNP code approximates the surface flux estimator in order to ensuring a finite variance for the sampled population.

⚠ Caution

An unmodified surface flux estimator has an infinite variance when $1/|\mu|$ is very large, and thus confidence intervals could not be formed via the central limit theorem because the central limit theorem requires a finite variance. For this reason, the MCNP code sets $\mu = 0.0005$ when $\mu < 0.001$; because of this approximation, the F2 surface flux tally is not an exact estimate of the surface flux. This value can be adjusted with the 24th entry on the DBCN card.

The SD card can be used to input a new area that divides the tally. In other words, if A' is input on the SD card, the tally will be divided by A' instead of A. See information in the SD card section on how the MCNP code handles the areas used by the tallies.

The surface flux tally is essential for stochastic calculation of surface areas when the normal analytic procedure fails [§2.9.2].

2.5.3 Energy Deposition Tally

The energy-deposition family of tallies are used to estimate cell heating. The F6 and TMESH cards can be used to tally energy deposition; the F6 card is for a cell-based tally and the TMESH card is for a mesh-based

tally. The F6 card provides the energy deposition for a single particle type in units of MeV/g per source particle. The *F6 card is equivalent to the F6 card except in units of jerks/g per source particle (1 MeV = 1.60219×10^{-22} jerks). The +F6 card provides an estimate of the total energy deposition from all particles. These tallies are implemented as hybrid track-length and collision tallies. The mass normalization of the cell-based energy deposition tally can be adjusted by the SD card. Multiple particles can be listed as follows: F6:p,n.

These tallies operate slightly differently depending on the incident particle, the MODE card, and if model physics are used. An overview is listed in Table 2.4. The heating numbers, which are the probability of a reaction multiplied by all kinetic energy carried away by the secondary particles, are generated by NJOY [12].

⚠ Caution

The use of heating numbers can result in negative energy deposition tallies in two cases. First, in the case in which collision tallies are used to subtract energies of secondary particles, this can result in negative +F6 tallies when the tally is undersampled. Second, older data may have poor separation of neutron and photon heating resulting in one of the two having negative values. The total energy deposition is still consistent in this second case.

⚠ Caution

The way the MCNP code handles F6 tallies results in double counting in a variety of cases, such as with a combination of photons and electrons, or with light ion recoil. As such, the sum of F6 tallies should not be used, with the exception of F6:n+F6:p which are designed to be compatible. For total energy deposition, +F6 should be considered instead.

This hybrid tallying approach was designed to minimize the cost of computing energy deposition for neutral particle problems. As the charged-particle contribution is contained within the neutral particle components, one does not need to simulate charged particles to get reasonable estimates. However, the use of heating numbers results in a number of caveats that one should be aware of:

1. The energy from non-transported secondary charged particles is deposited along the track (for projectiles with heating numbers) or at the point of collision (for everything else). If the mean free path of these secondary products would have been larger than the geometry of interest and as a result would have been deposited elsewhere, this can result in incorrect energy deposition.
2. Heating numbers ignore the energy deposition from secondary particles undergoing further reactions beyond slowing down.
3. Photonuclear reactions are not included in the photon data.
4. Heating from radioactive decay is not included.

The first three caveats can be remedied by adjusting the MODE and PHYS cards to include the necessary particles and physics. The more comprehensive both are, the more accurate energy deposition will be.

Radioactive decay is partially handled by the MCNP code in a variety of ways. Using TOTNU on (default), delayed neutrons from fission will be produced and transported, and will deposit energy in the same fashion as prompt neutrons. The generation of delayed neutrons from capture, as well as generation of other particles, is done via the ACT card.

With these caveats and remedies in mind, there are a few rules of thumb for computing accurate energy deposition:

Table 2.4: Physics-dependent Energy Deposition Methods

Neutrons		
	Table Physics	Track-length tallies are performed using heating numbers. These heating numbers include the kinetic energy for all secondary particles except photons. If available, the partial heating numbers of particles on the `MODE` card are removed to ensure consistency. If not, the energy of the secondary particle is subtracted out from *only* the `+F6` tally at the point of collision. This second case typically occurs during light ion recoil.
	Model Physics	Kinematics are tallied using collision tallies.
Photons		
	Table Physics	Track-length tallies are performed using heating numbers. These heating numbers include the kinetic energy for all secondary particles except neutrons. For secondary particles other than neutrons and electrons, energy balance is achieved using the same approach as for tabular neutrons above. Electron heating is never removed from the heating number. As such, `F6:p` and `F6:e` will double count the electron contribution. For `+F6` tallies, kinematic collision tallies are used for photons instead to guarantee consistency.
	Model Physics	Kinematics are tallied using collision tallies.
Charged Particles	The slowing down energy deposition is tallied by taking the start energy and end energy of a track and performing a track-length tally assuming a constant dE/dx. For cell-based tallies, this results in no approximation as particles will always stop at a surface crossing. For mesh-based tallies, this can lead to localized inconsistencies between neighboring mesh elements.	
	Table Physics (proton only)	Track-length tallies are performed using heating numbers. These heating numbers include the kinetic energy for all secondary particles. Energy balance is achieved using the same approach as for tabular neutrons above.
	Model Physics	Kinematics are tallied using collision tallies. If neutral daughter products (which includes neutrons, photons, neutrinos, π^0, and neutral kaons) are not on the `MODE` card, their energy will not be deposited.
Other Neutral Particles	Kinematics are tallied using collision tallies. If neutral daughter products (which includes neutrons, photons, neutrinos, π^0, and neutral kaons) are not on the `MODE` card, their energy will not be deposited.	

1. For low energy neutron sources and k-eigenvalue problems, a MODE n p, F6:n+F6:p or +F6 tally will provide reasonably accurate prompt total + fission delay neutron energy deposition values.
2. For low energy photon fixed-source problems, MODE p, F6:p or +F6 will provide reasonably accurate energy deposition. One should enable photofission if necessary.
3. The ACT card can allow computing non-fission delayed neutrons and other delayed particles. It can only be used for fixed-source simulations.
4. If the geometry is thin relative to the mean free path of generated secondary particles (such as electrons from photons, or recoil nuclei from any nuclear reaction), and the energy deposition in this component is important, one should add those particles to the simulation and use +F6 tallies to prevent double-counting energy deposition. In addition, light ion recoil may need to be enabled (see the PHYS:n and PHYS:h cards) for some problems.
5. If a given particle type is expected to undergo important reactions beyond slowing down, it should be added to the simulation.
6. If neutral particles can be generated, they should be included on the MODE card or the energy will not be tracked.

2.5.4 Track-length Fission Energy Deposition

The fission-energy deposition (F7) tally is a track-length estimate of neutron-induced fission energy deposition, and is given in units of MeV/g per source particle. The *F7 tally is identical to the F7 tally, but converted to jerks/g per source particle (see Table 2.2 and Table 2.3). The Q values used to compute F7 tallies are printed in PRINT Table 98 in an MCNP output file.

2.5.4.1 Equivalence of F4, F6, and F7 Tallies

For neutrons and photons, the F6 and F7 heating tallies are special cases of the F4 track length estimate of cell flux with energy-dependent multipliers. The tally combinations given in Listing give equivalent results.

Listing 2.2: tally_equivalence.mcnp.inp.txt

```
c Tally Definitions
f14:n 1
fm14 0.0025621 9  1 -4
f16:n 1
c
f24:n 1
fm24 0.0025621 9 -6 -8
f27:n 1
c
f34:p 1
fm34 0.0025621 9 -5 -6
f36:p 1
```

That is, the F14/FM14 and F16 tallies are equivalent, the F24/FM24 and F27 tallies are equivalent, and the F34/FM34 and F36 tallies are equivalent. In this example, material 9 in cell 1 is ^{235}U with an atom density (ρ_a) of 0.02 atoms/barn-cm and a mass density (ρ_g) of 7.80612 g/cm^3 for an atom/gram ratio of 0.0025621. Note that using $-1/\rho_g$ will give the same result as using ρ_a/ρ_g and is a better choice if perturbations are used. For more information on perturbations see §2.12.

For the photon results to be identical, both electron transport and the thick-target bremsstrahlung approximation (PHYS:p j 1 must be turned off. In the F6:p tally, if a photon produces an electron that produces a photon, the second photon is not counted again. It is already tallied in the first photon heating. In the F4: p tally, the second photon track is counted, so the F4 tally will slightly overpredict the tally.

The photon heating tally also can be checked against the *F8 energy deposition tally by dividing the F6 tally by a unit mass with the SD card. Results will only be statistically identical because the tallies are totally independent and use different estimators. The FM card can also be used to make the surface flux tally (F2) and point and ring detector tallies (F5) calculate heating, on a surface or at a point, respectively.

2.5.5 Pulse-height Tallies

The pulse height tally provides the energy distribution of pulses created in a cell that models a physical detector. It also can provide the energy deposition in a cell. Although the entries on the F8 card are cells, this is not a track length cell tally. The pulse-height tallies are made at source points and at surface crossings. The *F8 card changes the tally from deposition of pulses to an energy deposition tally and the +F8 card changes the tally to a charge deposition tally. The pulse height tally is analogous to a physical detector. The F8 energy bins (E_D) correspond to the total energy deposited in a detector in the specified channels by each computational particle (history). All the other MCNP tallies record the energy of a scoring track in the energy bin.

In an experimental configuration, suppose a source emits 100 photons at 10 MeV, and ten of these get to the detector cell. Further, suppose that the first photon (and any of its progeny created in the cell) deposits 1 keV in the detector before escaping, the second deposits 2 keV, and so on up to the tenth photon which deposits 10 keV. Then the pulse height measurement at the detector would be one pulse in the 1-keV energy bin, 1 pulse in the 2-keV energy bin, and so on up to 1 pulse in the 10-keV bin.

In the analogous MCNP pulse height tally, the source cell is credited with the energy times the weight of the source particle. When a particle crosses a surface, the energy times the weight of the particle is subtracted from the account of the cell that it is leaving and is added to the account of the cell that it is entering. The energy is the kinetic energy of the particle plus $2m_oc^2 = 1.022016$ MeV if the particle is a positron. At the end of the history, the account in each tally cell is divided by the source weight. The resulting energy determines which energy bin the score is put in. The value of the score is the source weight (W_C) for an F8 tally and the source weight times the energy in the account for a *F8 tally. The value of the score is zero if no track entered the cell during the history. Another aspect of the pulse height tally that is different from other MCNP tallies is that F8:p, F8:e and F8:p,e are all equivalent. All the energy from both photons and electrons, if present, will be deposited in the cell, no matter which tally is specified.

When the pulse height tally is used with energy bins, care must be taken because of negative scores from non-analog processes and zero scores caused by particles passing through the pulse height cell without depositing energy. In some codes, like the Integrated TIGER Series, these events cause large contributions to the lowest energy bin pulse height score. In other codes no contribution is made. The MCNP code compromises by counting these events in a zero bin and an epsilon bin so that these scores can be segregated out. It is recommended that energy binning for an F8 tally be something like

```
E8 0 1.e-5 1.0 2.0 3.0 4.0 5.0 ...
```

Knock-on electrons in the MCNP code are non-analog in that the energy loss is included in the multiple scattering energy loss rate rather than subtracted out at each knock-on event. Thus knock-ons can cause negative energy pulse height scores. These scores will be caught in the 0 energy bin. If they are a large fraction of the total F8 tally, then the tally is invalid because of non-analog events. Another situation is

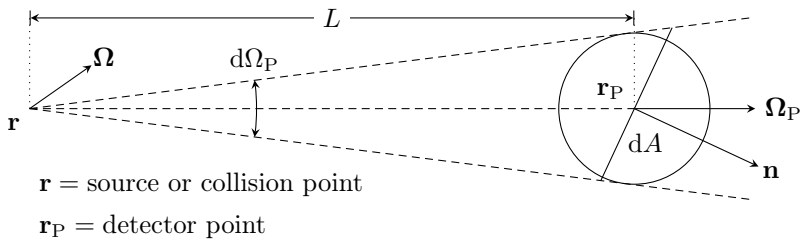


Figure 2.9: Illustration of point detector contributions.

differentiating zero contributions from particles not entering the cell and particles entering the cell but not depositing any energy. These are differentiated in the MCNP code by causing an arbitrary 1.e-12 energy loss for particles just passing through the cell. These will appear in the 0-epsilon bin.

2.5.6 Flux at a Detector

The neutral particle flux can be estimated at a point (or ring) using the point (or ring) detector next-event estimator. Neutral particle flux images using an array of point detectors—one detector for each pixel—can also be estimated. Detectors can yield anomalous statistics and must be used with caution. Detectors also have special variance reduction features, such as a highly advantageous DD card Russian roulette game. Whenever a user-supplied source is specified, a user-supplied source angle probability density function must also be provided.

2.5.6.1 Point Detector

A point detector is a deterministic estimate (from the current event point) of the flux at a point in space. Contributions to the point detector tally are made at source and collision events throughout the random walk. The point detector tally (F5) may be considered a limiting case of a surface flux tally (F2), as shown in Figure 2.9.

Consider the point detector to be a sphere whose radius is shrinking to zero. Let $\mathbf{\Omega}_\mathrm{P}$ be in the direction to the center of the sphere, i.e., in the direction $\mathbf{r}_\mathrm{P} - \mathbf{r}$. Let $\mathrm{d}\Omega_\mathrm{P}$ be the solid angle subtended by the sphere from $\mathbf{r}$, and let $\mathrm{d}A$ be defined by the intersection of an arbitrary plane (passing through the detector point) and the collapsing cone.

In order to contribute to a flux tally upon crossing $\mathrm{d}A$, the particle has to do two things. First, the particle must scatter toward $\mathrm{d}A$ (i.e. into solid angle $\mathrm{d}\Omega_\mathrm{P}$); this occurs with probability $p(\mathbf{\Omega}_\mathrm{P})\mathrm{d}\Omega_\mathrm{P}$. Second, the particle must have a collision-less free flight for the distance $L = |\mathbf{r}_\mathrm{P} - \mathbf{r}|$ (along $\mathbf{\Omega}_\mathrm{P}$) to the sphere; this occurs with probability $\exp\left(-\int_0^L \Sigma_\mathrm{t}(s)\mathrm{d}s\right)$, where $\Sigma_\mathrm{t}(s)$ is the total macroscopic cross section at a distance s (along $\mathbf{\Omega}_\mathrm{P}$) from the source or collision point. The probability that these two events both occur is

$$p(\mathbf{\Omega}_\mathrm{P})\mathrm{d}\Omega_\mathrm{P} \exp\left(-\int_0^L \Sigma_\mathrm{t}(s)\mathrm{d}s\right).$$

Define η to be the cosine of the angle between the particle direction and the unit normal ($\mathbf{n}$) to area $\mathrm{d}A$ as

$$\eta = \mathbf{\Omega}_\mathrm{P} \cdot \mathbf{n}. \tag{2.178}$$

If a particle of weight w reaches $\mathrm{d}A$, it will contribute $w/|\eta|\mathrm{d}A$ to the flux (compare to the F2 tally in §2.5.2.2).

As the sphere shrinks to a point, the solid angle subtended by $\mathrm{d}A$ is $\mathrm{d}\Omega_\mathrm{P} = |\eta|\mathrm{d}A/L^2$. The sides of the cone in the figure become parallel and the cone resembles a cylinder near the shrinking sphere. Thus the tally becomes

$$\begin{aligned}\mathrm{F5} &= p(\mathbf{\Omega}_\mathrm{P})\mathrm{d}\Omega_\mathrm{P} \exp\left[-\int_0^L \Sigma_\mathrm{t}(s)\mathrm{d}s\right]\frac{w}{|\eta|\mathrm{d}A} \\ &= wp(\mathbf{\Omega}_\mathrm{P})\frac{|\eta|\mathrm{d}A}{L^2}\frac{1}{|\eta|\mathrm{d}A}\exp\left[-\int_0^L \Sigma_\mathrm{t}(s)\mathrm{d}s\right]\end{aligned}$$

or

$$\mathrm{F5} = w\frac{p(\mathbf{\Omega}_\mathrm{P})}{L^2}\exp\left[-\int_0^L \Sigma_\mathrm{t}(s)\mathrm{d}s\right]. \tag{2.179}$$

In all the scattering distributions and in the standard sources, the MCNP code assumes azimuthal symmetry. This provides some simplification. The vector $\mathbf{\Omega}_\mathrm{P}$ can be expressed in spherical coordinates with respect to the particle's direction of travel, $\mathbf{\Omega}$, being the polar axis. The azimuthal angle is ϕ and the cosine of the polar angle is μ. The probability of scattering into $\mathrm{d}\Omega_\mathrm{P}$ can then be written in terms of a probability in μ, ϕ. That is,

$$p(\mathbf{\Omega}_\mathrm{P})\mathrm{d}\Omega_\mathrm{P} = p(\mu,\phi)\mathrm{d}\mu\mathrm{d}\phi. \tag{2.180}$$

Defining the probability density function for scattering about μ as

$$p(\mu) \equiv \int_0^{2\pi} p(\mu,\phi)\mathrm{d}\phi \tag{2.181}$$

and, recalling that $p(\mu,\phi)$ is independent of ϕ, yields

$$p(\mu,\phi) = \frac{p(\mu)}{2\pi}. \tag{2.182}$$

Substituting this into Eq. (2.179) yields

$$\mathrm{F5} = w\frac{p(\mu)}{2\pi L^2}\exp\left[-\int_0^L \Sigma_\mathrm{t}(s)\mathrm{d}s\right]. \tag{2.183}$$

A point detector tally is known as a "next-event estimator" because it is a tally of the flux at a point as if the "next event" were a particle trajectory directly to the detector point without further collision.

A contribution to the point detector is made at every source or collision event. The free-flight probability term, $\exp\left[-\int_0^L \Sigma_\mathrm{t}(s)\mathrm{d}s\right]$, term accounts for attenuation between the present event and the detector point. The $1/2\pi L^2$ term accounts for the solid angle effect. The $p(\mu)$ term accounts for the probability of scattering toward the detector instead of the direction selected in the random walk. For an isotropic source or scatter, $p(\mu) = 0.5$, and the solid-angle terms reduce to the expected $1/4\pi L^2$. Note that $p(\mu)$ can be larger than unity because it is the value of a density function and not a probability. Each contribution to the detector can be thought of as the transport of a pseudoparticle to the detector.

The L^2 term in the denominator of the point detector causes a singularity that makes the theoretical variance of this estimator infinite. That is, if a source or collision event occurs near the detector point, L approaches zero and the flux approaches infinity. The technique is still valid and unbiased, but convergence is slower and often impractical. If the detector is not in a source or scattering medium, a source or collision close to the detector is impossible. For problems where there are many scattering events near the detector, a cell or surface estimator should be used instead of a point detector tally. If there are so few scattering events near the detector that cell and surface tallies are impossible, a point detector can still be used with a specified average flux region close to the detector. This region is defined by a fictitious sphere of radius R_o surrounding the point detector. R_o can be specified either in centimeters or in mean free paths. If R_o is specified in centimeters and if $L < R_o$, the point detector estimation inside R_o is assumed to be the average flux uniformly distributed in volume. That is

$$\Phi(L < R_o) = \frac{\int_V \Phi(r,\theta,\phi)\mathrm{d}V}{\int_V \mathrm{d}V} \tag{2.184}$$

$$= \frac{\int_0^{R_o} w \frac{p(\mu)}{2\pi \cancel{r^2}} \exp\left[-\int_0^r \Sigma_\mathrm{t}(s)\mathrm{d}s\right] \cancel{r^2}\mathrm{d}r \int_0^\pi \sin(\phi)\mathrm{d}\phi \int_0^{2\pi} \mathrm{d}\theta}{\int_0^{R_o} r^2\mathrm{d}r \int_0^\pi \sin(\phi)\mathrm{d}\phi \int_0^{2\pi} \mathrm{d}\theta} \tag{2.185}$$

$$= \frac{\frac{4\pi}{2\pi} w\, p(\mu) \int_0^{R_o} \exp\left[-\int_0^r \Sigma_\mathrm{t}(s)\mathrm{d}s\right]\mathrm{d}r}{\frac{4\pi}{3} R_o^3} \tag{2.186}$$

$$= \frac{3w\, p(\mu) \int_0^{R_o} \exp\left[-\int_0^r \Sigma_\mathrm{t}(s)\mathrm{d}s\right]\mathrm{d}r}{2\pi R_o^3}, \tag{2.187}$$

where we can assume that the total cross section is constant within the sphere, so

$$\Phi(L < R_o) = \frac{3w\, p(\mu) \int_0^{R_o} \exp[-\Sigma_\mathrm{t} r]\mathrm{d}r}{2\pi R_o^3} \tag{2.188}$$

$$= \frac{3w\, p(\mu)[1 - \exp(-\Sigma_\mathrm{t} R_o)]}{2\pi R_o^3 \Sigma_\mathrm{t}}. \tag{2.189}$$

If $\Sigma_\mathrm{t} = 0$, the detector is not in a scattering medium, no collision can occur, and

$$\Phi(L < R_o, \Sigma_\mathrm{t} = 0) = \lim_{\Sigma_\mathrm{t} \to 0} \frac{3w\, p(\mu)[1 - \exp(-\Sigma_\mathrm{t} R_o)]}{2\pi R_o^3 \Sigma_\mathrm{t}} = \frac{3w\, p(\mu)}{2\pi R_o^2}. \tag{2.190}$$

If the fictitious sphere radius is specified in mean free paths λ_0, then $\lambda_0 = \Sigma_\mathrm{t} R_o$ and

$$\Phi(\lambda < \lambda_0) = \frac{3w\, p(\mu)[1 - \exp(-\lambda_0)]\Sigma_\mathrm{t}^2}{2\pi \lambda_0^3}. \tag{2.191}$$

The choice of R_o may require some experimentation. For a detector in a void region R_o can be set to zero. However, one is cautioned against using $R_o = 0$ in regions with very few collisions (such as air), because this can cause rare scattering events near the detector that result in large scores. For a typical problem, setting R_o to a mean free path or some fraction thereof is usually adequate. If R_o is specified in centimeters, it should correspond to the mean free path for some average energy in the sphere.

⚠ Caution

Be certain when defining R_o that the sphere it defines does not encompass more than one material unless you understand the consequences. This is especially true when defining R_o in terms of mean free path because R_o becomes a function of energy and can vary widely. If the sphere does contain multiple materials, the total cross section used corresponds to the material at the center of the sphere.

In particular, if R_o is defined in terms of mean free paths and if a detector is on a surface that bounds a void on one side and a material on the other, the contribution to the detector from the direction of the void

will be zero even though the importance of the void is nonzero. The reason is simply that the volume of the artificial sphere is infinite in a void. Contributions to the detector from the other direction (that is, across the material) will be accounted for.

Detectors differing only in R_o are coincident detectors [§2.5.6.4.4], and there is little cost incurred by experimenting with several detectors that differ only by R_o in a single problem.

2.5.6.2 Ring Detector

A ring detector [142] tally is a point detector tally in which the point detector location is not fixed but rather sampled from some location on a ring. Most of the previous section on point detectors applies to ring detectors as well. In the MCNP code, three ring detector tallies (`Fx`, `Fy`, and `Fz`) correspond to rings located rotationally symmetric about the x-, y-, and z-coordinate axes. A ring detector usually enhances the efficiency of point detectors for problems that are rotationally symmetric about a coordinate axis. Ring detectors also can be used for problems where the user is interested in the average flux at a point on a ring about a coordinate axis.

Although the ring detector is based on the point detector that has a $1/L^2$ singularity and an unbounded variance, the ring detector has a finite variance and only a $1/L_{\min}$ singularity, where $L_{\min}$ is the minimum distance between the contributing point and the detector ring [143].

In a cylindrically symmetric system, the flux is constant on a ring about the axis of symmetry. Hence, one can sample uniformly for positions on the ring to determine the flux at any point on the ring. The ring detector efficiency is improved by biasing the selection of point detector locations to favor those near the contributing collision or source point. This bias results in the same total number of detector contributions, but the large contributions are sampled more frequently, reducing the relative error.

For isotropic scattering in the lab system, experience has shown that a good biasing function is proportional to $\exp(-P)L^{-2}$, where P is the number of mean free paths and L is the distance from the collision point to the detector point. For most practical applications, using a biasing function involving P presents prohibitive computational complexity except for homogeneous medium problems. For air transport problems, a biasing function resembling $\exp(-P)$ has been used with good results. A biasing function was desired that would be applicable to problems involving dissimilar scattering media and would be effective in reducing variance. The function L^{-2} meets these requirements.

In Figure 2.10, consider a collision point, (x_o, y_o, z_o) at a distance L from a point detector location (x, y, z). The point (x, y, z) is to be selected from points on a ring of radius r that is symmetric about the y-axis in this case.

To sample a position (x, y, z) on the ring with a $1/L^2$ bias, we pick φ from the density function $p(\varphi) = C/\left(2\pi L^2\right)$, where C is a normalization constant. To pick φ from $p(\varphi)$, let ξ be a random number on the unit interval. Then

$$\begin{aligned}
\xi &= \frac{C}{2\pi}\int_{-\pi}^{\varphi}\frac{\mathrm{d}\varphi'}{L^2} \\
&= \frac{C}{2\pi}\int_{-\pi}^{\varphi}\frac{\mathrm{d}\varphi'}{\left(x_o - r\cos(\varphi')\right)^2 + \left(y_o - y\right)^2 + \left(x_o - r\sin(\varphi')\right)^2} \\
&= \frac{C}{2\pi}\int_{-\pi}^{\varphi}\frac{\mathrm{d}\varphi'}{a + b\cos(\varphi') + c\sin(\varphi')} \\
&= \frac{1}{\pi}\tan^{-1}\left\{\frac{1}{C}\left[(a-b)\tan\left(\frac{\varphi}{2}\right) + c\right]\right\} + \frac{1}{2}
\end{aligned} \tag{2.192}$$

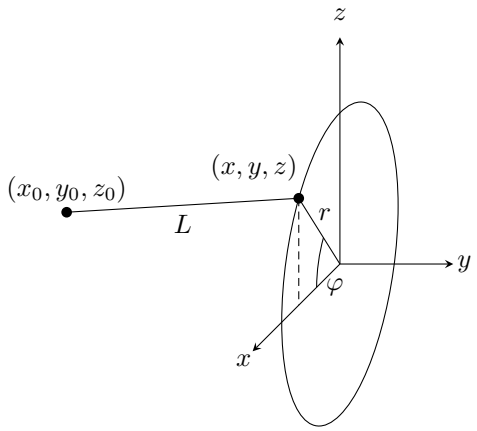


Figure 2.10: Illustration of ring detector contributions.

where

a	$= r^2 + x_o^2 + (y - y_o)^2 + z_o^2$,
b	$= -2rx_o$,
c	$= -2rz_o$, and
C	$= \left(a^2 - b^2 - c^2\right)^{1/2}$.

The above equation is valid if $a^2 > b^2 + c^2$, which is true except for collisions exactly on the ring.

Solving for $\tan(\varphi/2)$, one obtains

$$\tan\left(\frac{\varphi}{2}\right) = \frac{1}{a-b}\left\{C\tan\left[\pi\left(\xi - \frac{1}{2}\right)\right] - c\right\}. \tag{2.193}$$

Letting $t = \tan(\varphi/2)$, then

$$x = r\cos(\varphi) = r\frac{1-t^2}{1+t^2}, \tag{2.194a}$$

$$y = y \text{ (fixed)}, \tag{2.194b}$$

$$z = r\sin(\varphi) = \frac{2rt}{1+t^2}. \tag{2.194c}$$

For ring detectors, the $1/L^2$ biasing has been supplemented when it is weak to include a biasing based on angle to select the point on the ring. This angle is in the plane of the ring and is relative to the shortest line from the collision point to the detector ring. The angle that would most likely be selected would pick the same point on the ring as a straight line through the axis of the problem, the collision point, and the ring. The angle least likely to be picked would choose the point on the opposite side of the ring. This approach will thus make scores with smaller attenuations more often. This supplemental biasing is achieved by requiring that $a \le 3/2\left(b^2 + c^2\right)^{1/2}$ in Eq. (2.193).

If the radius of the ring is very large compared to the dimensions of the scattering media (such that the detector sees essentially a point source in a vacuum), the ring detector is still more efficient than a point detector. The reason for this unexpected behavior is that the individual scores to the ring detector for a

specific history have a mean closer to the true mean than to the regular point detector contributions. That is, the point detector contributions from one history will tend to cluster about the wrong mean because the history will not have collisions uniformly in volume throughout the problem, whereas the ring detector will sample many paths through the problem geometry to get to different points on the ring.

2.5.6.3 Flux Image Detectors

Flux image detector tallies are an array of point detectors close enough to one another to generate an image based on the point detector fluxes. Each detector point represents one pixel of the flux image. The source need not be embedded in the object. The particle creating the image does not have to be the source particle type. Three types of neutral particle flux image tallies can be made [144, 145]:

- Flux Image Radiograph (`FIR`), a flux image radiograph on a planar image surface;
- Flux Image on a Cylinder (`FIC`), a flux image on a cylindrical image surface; and
- Flux Image by Pinhole (`FIP`), a flux image by pinhole on a planar image surface.

When these flux image tallies are used with `FS`n and `C`n cards to construct a virtual image grid, millions of point detectors can be created-one detector for each pixel-to produce a flux image. The `FS`n card is used to define the image pixels along the s-axis. The `C`n card defines the pixels along the t-axis. The relationship of the s-axis, t-axis, and reference direction for the planar image grid is calculated by the MCNP code and follows the right-hand rule. Since the orientation of the s-axis and the t-axis is dependent on the reference direction in the geometry coordinate system, the MCNP tally output should be examined to see the direction cosines of these two planar image grid axes.

⚠ Caution

The image grid SHOULD NOT be in a scattering material because the point detector average flux neighborhood is not used for flux image tallies.

2.5.6.3.1 Radiograph Image Tallies

Both the Flux Image Radiograph (`FIR`) and Flux Image on a Cylinder (`FIC`) tallies act like film for an x-ray type image (that is, a transmitted image for neutrons or photons). The diagram in Figure 2.11 shows how the `FIR` planar rectangular grid image is defined for a source particle passing through an object and scattering in an object. An `FIC` cylindrical surface grid generates an image on a cylinder as shown in Figure 2.12 for the particles generated inside the object.

In both cases, a ray-trace point-detector flux contribution is made to every image grid bin (pixel) from each source and scatter event. Allowing each event to contribute to all pixels reduces statistical fluctuations across the grid that would occur if the grid location for the contribution were selected randomly. For each source and scatter event, the direction cosines to a pixel detector point are determined. The option exists to select a random position in the pixel. The same relative random offset is used for all pixels for a source or scatter event. The random detector location in a pixel changes from event to event. The option also exists to select the point detector location at the center of each pixel when the center flux is desired.

A standard point detector attenuated ray-trace flux contribution to the image pixel is then made. A new direction cosine is determined for each pixel followed by the new ray-trace flux calculation. These tallies automatically create a source-only contribution and a total for each pixel. Standard point detector tally modifications can be made to the image tally, for example, by using the `FM`, `PD`, and `FT` cards.

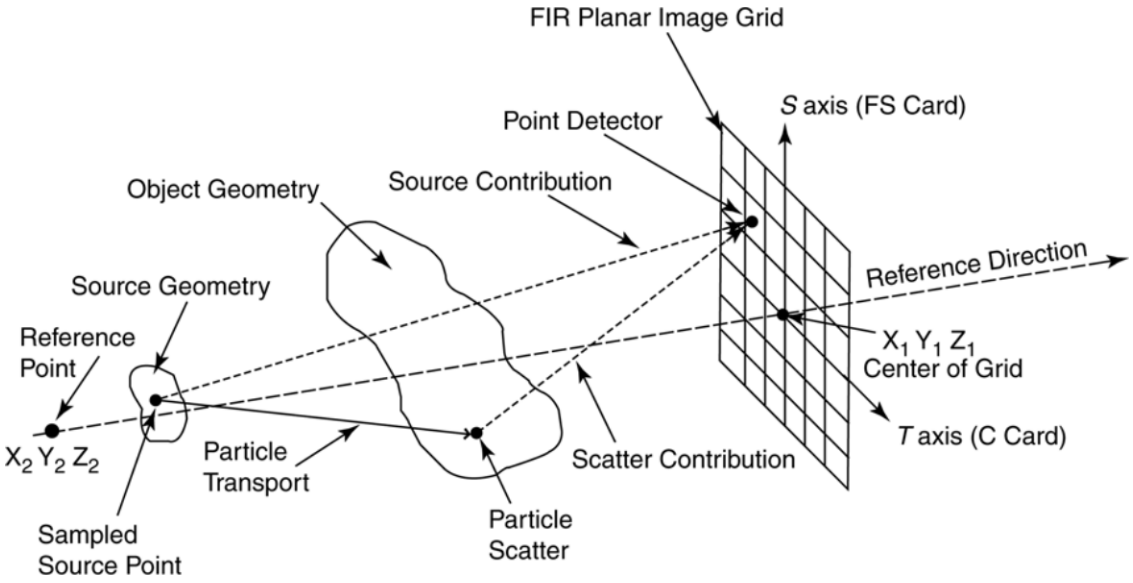


Figure 2.11: Diagram of an FIR (Flux Image Radiograph) tally for a source external to the object. The directions of the orthogonal S- and T-axes depend on the reference-direction vector in the geometry coordinate system.

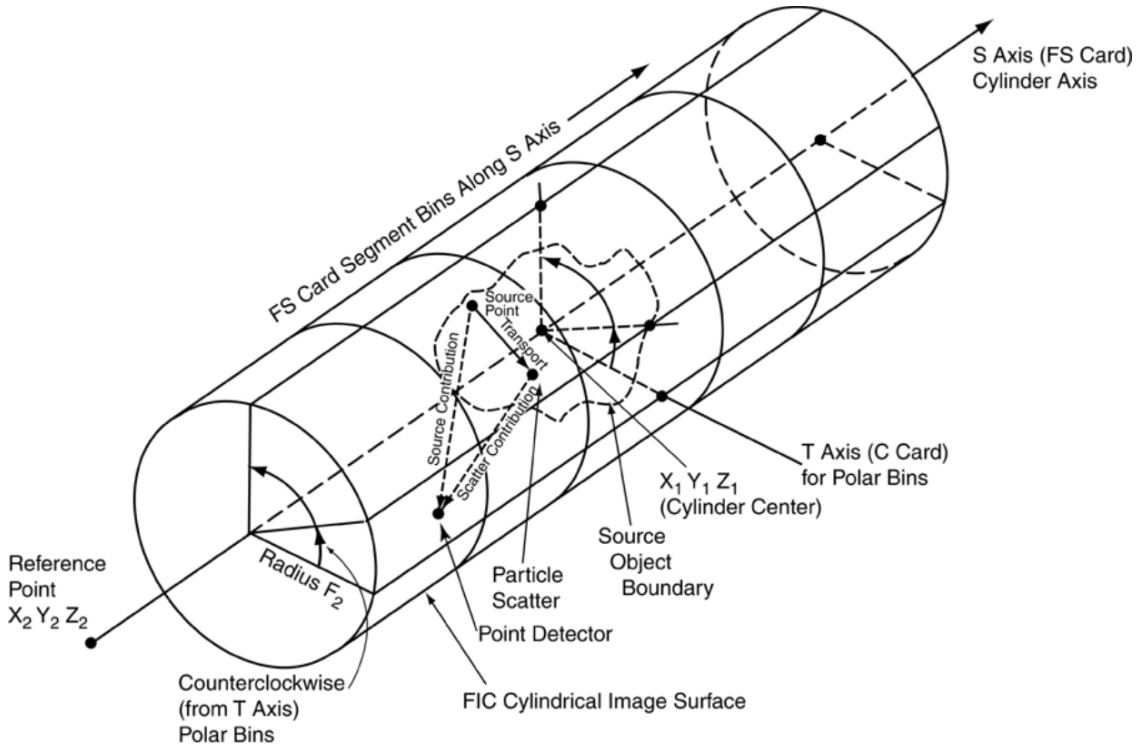


Figure 2.12: Diagram of an FIC (Flux Image on a Cylinder) tally for a source internal to the object.

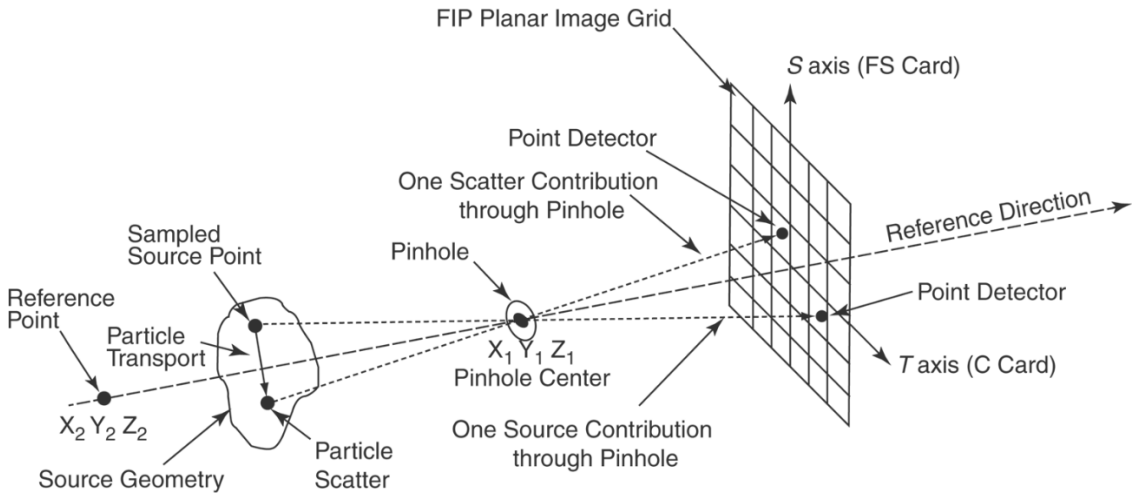


Figure 2.13: Diagram of an FIP (Flux Image by Pinhole) tally for a source internal to the object. The directions of the orthogonal S- and T-axes depend on the reference-direction vector in the geometry coordinate system.

2.5.6.3.2 Pinhole Image Tally

The Flux Image by Pinhole (FIP) tally uses a pinhole (as in a pinhole camera) to create a neutron or photon image onto a planar rectangular grid that acts much like photographic film. Figure 2.13 is a diagram of the FIP image tally. Each source and scatter event contributes to one point detector on the image grid pixel intersected by the particle trajectory through the pinhole.

The particle event point and the virtual pinhole point (sampled uniformly in area if a radius is specified) are used to define the direction cosines of the contribution to be made from the source or scatter location through the pinhole to one image grid element (pixel). Once this direction is established, a ray-trace point detector flux contribution is made to the intersected pixel including attenuation by any material along that path. No source or scattering events on the image grid side of the pinhole will contribute to the image.

The pinhole and associated grid will image both direct source contributions and the direct plus any scattered contributions. Standard tally modifications can be made to the image tally, for example, by using the FM, PD, and FT cards.

The magnitude of the flux contribution through the pinhole to a pixel is calculated as follows. The flux at a pinhole point P is $\phi_P(\mathbf{\Omega})$, where $\mathbf{\Omega}$ is the direction that intersects the pinhole at point P. Define μ to be the cosine of the angle between the detector trajectory and the reference direction, which is perpendicular to the plane of the pinhole. The particle weight per unit pinhole area (or the particle current per unit pinhole area) is $\phi_P(\mathbf{\Omega})\mu$. The weight in a small area $\mathrm{d}A$ in the pinhole is $\phi_P(\mathbf{\Omega})\mu\mathrm{d}A$. The total particle weight W integrated over the pinhole area A_P is

$$W = \int_{A_P} \phi_P(\mathbf{\Omega})\mu\mathrm{d}A. \tag{2.195}$$

The FIP tally selects one particle trajectory to carry this weight. This trajectory should be sampled in $\mathrm{d}A$ from

$$p(\mathbf{\Omega})\mathrm{d}\Omega = \frac{\phi_P(\mathbf{\Omega})\mu\mathrm{d}A}{\int_{A_P} \phi_P(\mathbf{\Omega})\mu\mathrm{d}A}. \tag{2.196}$$

Instead, the pinhole point P sampling is biased to be uniform in the pinhole area A_P; that is,

$$b(\mathbf{\Omega})\mathrm{d}\Omega = \frac{\mathrm{d}A}{A_P}. \tag{2.197}$$

To account for this biased sampling, the weight W of the sample must be multiplied by

$$w_m(\mathbf{\Omega}) = \frac{p(\mathbf{\Omega})}{b(\mathbf{\Omega})} = \frac{A_P \phi_P(\mathbf{\Omega})\mu}{\int_{A_P} \phi_P(\mathbf{\Omega})\mu \mathrm{d}A}. \tag{2.198}$$

Thus, an unbiased estimate of the sampled weight going through $\mathrm{d}A$ at the pinhole is $W_P(\mathbf{\Omega}) = W w_m(\mathbf{\Omega})$ or

$$W_P(\mathbf{\Omega}) = \left[\int_{A_P} \phi_P(\mathbf{\Omega})\mu \mathrm{d}A\right]\left[\frac{A_P \phi_P(\mathbf{\Omega})\mu}{\int_{A_P} \phi_P(\mathbf{\Omega})\mu \mathrm{d}A}\right] = A_P \phi_P(\mathbf{\Omega})\mu. \tag{2.199}$$

Now that an unbiased estimate of the weight through $\mathrm{d}A$ is obtained, an unbiased estimate of the weight arriving on the image plane can also be obtained. If $\lambda(\mathbf{\Omega})$ is the optical path along $\mathbf{\Omega}$ from the sampled pinhole point to the image plane, then the weight $W_{\text{pixel}}(\mathbf{\Omega})$ arriving at the pixel in the image plane is

$$W_{\text{pixel}}(\mathbf{\Omega}) = W_P(\mathbf{\Omega}) \exp[-\lambda(\mathbf{\Omega})] = A_P \phi_P(\mathbf{\Omega})\mu \exp[-\lambda(\mathbf{\Omega})]. \tag{2.200}$$

The surface flux at the image plane is estimated by the $W_{\text{pixel}}(\mathbf{\Omega})$ divided by μ (note that the pinhole plane and image plane are parallel) divided by pixel area A_{pixel}. Therefore, the surface flux at the intersected pixel is

$$\phi_{\text{pixel}}(\mathbf{\Omega}) = \frac{A_P \phi_P(\mathbf{\Omega}) \exp[-\lambda(\mathbf{\Omega})]}{A_{\text{pixel}}}. \tag{2.201}$$

Thus, the flux at the pixel is just the $\exp[-\lambda(\mathbf{\Omega})]$-attenuated flux at the pinhole scaled by the ratio of A_P (where the weight W passes through) to the A_{pixel} (the pixel where the flux $\phi_{\text{pixel}}(\mathbf{\Omega})$ is scored). If a perfect pinhole with no pinhole area is used, then A_P is defined to be unity.

2.5.6.4 General Considerations of Point Detector Estimators

2.5.6.4.1 Pseudoparticles and Detector Reliability

Point and ring detectors are Monte Carlo methods wherein the simulation of particle transport from one place to another is deterministically short-circuited. Transport from the source or collision point to the detector is replaced by a deterministic estimate of the potential contribution to the detector. This transport between the source or collision point and the detector can be thought of as being via "pseudoparticles." Pseudoparticles undergo no further collisions. These particles do not reduce the weight or otherwise affect the random walk of the particles that produced them. They are merely estimates of a potential contribution. The only resemblance to Monte Carlo particles is that the quantity they estimate requires an attenuation term that must be summed over the trajectory from the source or collision to the detector. Thus most of the machinery for transporting particles can also be used for the pseudoparticles. No records (for example, tracks entering) are kept about pseudoparticle passage.

⚠ Caution

Because detectors rely on pseudoparticles rather than particle simulation by random walk, they should be considered only as a very useful last resort. Detectors are unbiased estimators, but their use can be tricky, misleading, and occasionally unreliable.

Consider the problem illustrated in Figure 2.14. The monoenergetic isotropic point source always will make the same contribution to the point detector, so the variance of that contribution will be zero. If no particles have yet collided in the scattering region, the detector tally will be converged to the source contribution, which is wrong and misleading. But as soon as a particle collides in the scattering region, the detector tally

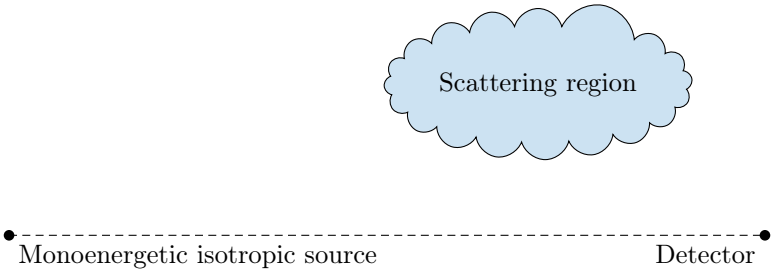


Figure 2.14: Demonstration of inappropriate source-point detector-scatterer configuration.

and its variance will jump. Then the detector tally and variance will steadily decrease until the next particle collides in the scattering region, at which time there will be another jump.

These jumps in the detector score and variance are characteristic of undersampling important regions. Next-event estimators are prone to undersampling as already described in §2.4.4.2.5 for the $p(\mu)$ term of photon coherent scattering. The jump discussed here is from the sudden change in the L and possibly λ terms. Jumps in the tally caused by undersampling can be eliminated only by better sampling of the undersampled scattering region that caused them.

Biasing Monte Carlo particles toward the tally region would cause the scattering region to be sampled better, thus eliminating the jump problem. It is recommended that detectors be used with caution and with a complete understanding of the nature of next-event estimators. When detectors are used, the tally fluctuation charts printed in the output file should be examined closely to see the degree of the fluctuations. Also the detector diagnostic tables in the MCNP output file should be examined to see if any one pseudoparticle trajectory made an unusually large contribution to the tally. **Detector results should be viewed suspiciously if the relative error is greater than 5%.** Close attention should be paid to the tally statistical analysis and the ten statistical checks described in §2.6.9.2.3.

2.5.6.4.2 Detectors and Reflecting, White, or Periodic Surfaces

⚠ Caution

Detectors used with reflecting, white, or periodic surfaces give wrong answers because pseudoparticles travel only in straight lines.

Consider Figure 2.15, with a point detector and six source cells. The imaginary cells and point detector are also shown on the other side of the mirror. The solid line shows the source contribution from the indicated cell. The MCNP code does not allow for the dashed-line contribution on the other side of the reflecting surface. The result is that contributions to the detector will always be from the solid path instead of from a mixture of solid and dashed contributions. This same situation occurs at every collision. Therefore, the detector tally will be lower (with the same starting weight) than the correct answer and should not be used with reflecting, white, or periodic surfaces. The effect is even worse for problems with multiple reflecting, white, or periodic surfaces.

2.5.6.4.3 Variance-reduction Schemes for Detectors

Pseudoparticles of point detectors are not subject to the variance reduction schemes applied to particles of the random walk. They do not split according to importances, weight windows, etc., although they are

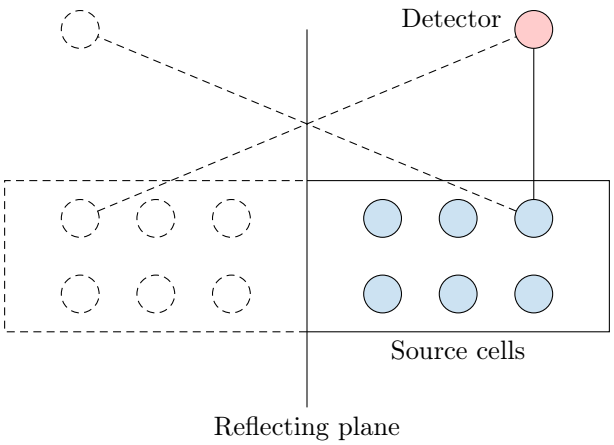


Figure 2.15: Demonstration of inappropriate source-point detector-reflecting boundary scenario.

terminated by entering zero importance cells. However, two Russian roulette games are available specifically for detector pseudoparticles.

The PD card can be used to specify the pseudoparticle generation probability for each cell. The entry for each cell i is p_i where $0 \le p_i \le 1$. Pseudoparticles are created with probability p_i and weight $1/p_i$. If $p_i = 1$, which is the default, every source or collision event produces a pseudoparticle. If $p_i = 0$, no pseudoparticle is produced.

Caution

Setting $p_i = 0$ in a cell that can actually contribute to a detector erroneously biases the detector tally by eliminating such contributions.

Thus $p_i = 0$ should be used only if the true probability of scoring is zero or if the score from cell i is unwanted for some legitimate reason such as problem diagnostics. Fractional entries of p_i should be used with caution because the PD card applies equally to all pseudoparticles. The DD card can be used to Russian roulette just the unimportant pseudoparticles. However, the DD card roulette game often requires particles to travel some distance along their trajectory before being killed. When cells are many mean-free paths from the detector, the PD card may be preferable.

The DD card controls both the detector diagnostic printing and a Russian roulette game played on pseudoparticles in transit to detectors. The Russian roulette game is governed by the input parameter k that controls a comparison weight w_c internal to the MCNP code, such that

$$w_c = \begin{cases} -k & k < 0 \\ 0 & k = 0 \\ 0 & k > 0 \text{ and } N \le 200 \\ \frac{k}{N}\Sigma_i^I \varphi_i & k > 0 \text{ and } N > 200 \end{cases} \tag{2.202}$$

where

N	is the number of histories run thus far,
I	is the number of pseudoparticles started so far,

φ_i	$= \frac{Wp(\mu)\exp(-\lambda)}{2\pi L^2}$, and
I	is the contribution from the ith pseudoparticle to the detector tally.

When each pseudoparticle is generated, W, $p(\mu)$, and L are already known before the expensive tracking process is undertaken to determine λ. If $Wp(\mu)/\left(2\pi L^2\right) < w_c$, the pseudoparticle contribution to the detector φ_i will be less than the comparison weight. Playing Russian roulette on all pseudoparticles with $\varphi_i < w_c$ avoids the expensive tracking of unimportant pseudoparticles. Most are never started. Some are started but are rouletted as soon as λ has increased to the point where $Wp(\mu)e^{-\lambda}/\left(2\pi L^2\right) < w_c$. Rouletting pseudoparticles whose expected detector contribution is small also has the added benefit that those pseudoparticles surviving Russian roulette now have larger weights, so the disparity in particle weights reaching the detector is reduced. Typically, using the DD card will increase the efficiency of detector problems by a factor of ten. This Russian roulette is so powerful that it is one of two MCNP variance reduction options that is turned on by default. The default value of k is 0.1. The other default variance reduction option is implicit capture.

The DD card Russian roulette game is almost foolproof. Performance is relatively insensitive to the input value of k. For most applications the default value of $k = 0.1$ is adequate. Usually, choose k so that there are 1–5 transmissions (pseudoparticle contributions) per source history. If k is too large, too few pseudoparticles are sampled; thus $k \geq 1$ is a fatal error.

⚠ Caution

Because a random number is used for the Russian roulette game invoked by $k > 0$, the addition of a detector tally affects the random walk tracking processes.

Detectors are the only tallies that affect results. If any other tally type is added to a problem, the original problem tallies remain unchanged. Because detectors use the default DD card Russian roulette game, and that game affects the random number sequence, the whole problem will track differently and the original tallies will agree only to within statistics. Because of this tracking difference, it is recommended that $k < 0$ be used once a good guess at w_c can be made. This is especially important if a problem needs to be debugged by starting at some history past the first one. Also, $k < 0$ makes the first 200 histories run faster.

There are two cases when it is beneficial to turn off the DD card Russian roulette game by setting $k = 0$. First, when looking at the tail of a spectrum or some other low probability event, the DD card roulette game will preferentially eliminate small scores and thus eliminate the very phenomenon of interest. For example, if energy bias is used to preferentially produce high energy particles, these biased particles will have a lower weight and thus preferentially will be rouletted by the DD card game. Second, in very deep penetration problems, pseudoparticles will sometimes go a long way before being rouletted. In this rare case it is wasteful to roulette a pseudoparticle after a great deal of time has been spent following it and perhaps a fractional PD card should be used or, if possible, a cell or surface tally.

2.5.6.4.4 Coincident Detectors

Because tracking pseudoparticles is very expensive, the MCNP code uses a single pseudoparticle for multiple detectors, known as coincident detectors, that must be identical in:

- geometric location,
- particle type (that is, neutron or photon),
- upper time bin limit,

- DD card Russian Roulette control parameter, k, and
- PD card entries, if any.

Energy bins, time bins, tally multipliers, response functions, fictitious sphere radii, user-supplied modifications (**TALLYX**), etc., can all be different. Coincident detectors require little additional computational effort because most detector time is spent in tracking a pseudoparticle. Multiple detectors using the same pseudoparticle are almost "free."

2.5.6.4.5 Direct vs. Total Contribution

Unless specifically turned off by the user, the MCNP code automatically prints out both the direct and total detector contribution. Recall that pseudoparticles are generated at source and collision events. The direct contribution is that portion of the tally from pseudoparticles born at source events. The total contribution is the total tally from both source and collision events. For MODE N P problems with photon detectors, the direct contribution is from pseudophotons born in neutron collisions. The direct contributions for detailed photon physics will be smaller than the simple physics direct results because coherent scattering is included in the detailed physics total cross section and omitted in the simple physics treatment.

2.5.6.4.6 Angular Distribution Functions for Point Detectors

All detector estimates require knowledge of the $p(\mu)$ term, the value of the probability density function at an angle θ, where $\mu = \cos(\theta)$. This quantity is available to the MCNP code for the standard source and for all kinds of collisions. For user-supplied source subroutines, the MCNP code assumes an isotropic distribution,

$$p(\mu)\mathrm{d}\mu = \frac{\mathrm{d}\Omega}{4\pi} = \int_0^{2\pi} \frac{\mathrm{d}\mu\mathrm{d}\varphi}{4\pi} = \frac{1}{2}\mathrm{d}\mu. \tag{2.203}$$

Therefore, the variable PSC $= p(\mu) = 1/2$. If the source distribution is not isotropic in a user-supplied source subroutine, the user must also supply a subroutine **SRCDX** if there are any detectors or DXTRAN spheres in the problem. In subroutine **SRCDX**, the variable PSC must be set for each detector and DXTRAN sphere. An example of how this is done and also a description of several other source angular distribution functions is in §10.3.5.

2.5.6.4.7 Detectors and the $S(\alpha, \beta)$ Thermal Treatment

The $S(\alpha, \beta)$ thermal treatment poses special challenges to next-event estimators because the probability density function for angle has discrete lines to model Bragg scattering and other molecular effects. Therefore, the MCNP code has an approximate model [76] that, for the PSC calculation (not the transport calculation), replaces the discrete lines with finite histograms of width $\mu < 0.1$.

This approximation has been demonstrated to accurately model the discrete line $S(\alpha, \beta)$ data. In cases where continuous data is approximated with discrete lines, the approximate scheme cancels the errors and models the scattering better than the random walk [77]. Thus the $S(\alpha, \beta)$ thermal treatment can be used with confidence with next-event estimators like detectors and DXTRAN.

2.5.7 Additional Tally Features

The standard MCNP tally types can be controlled, modified, and beautified by other tally cards. These cards are described in detail in §3.2.5.4; an overview is given here.

2.5.7.1 Bin Limit Control

The integration limits of the various tally types can be controlled by `E`, `T`, `C`, and `FS` cards. The `E` card establishes energy bin ranges; the `T` card establishes time bin ranges; the `C` card establishes cosine bin ranges; and the `FS` card segments the surface or cell of a tally into subsurface or subcell bins.

2.5.7.2 Flagging

Cell and surface flagging cards, `CF` and `SF`, determine where the different portions of a tally originate. For example:

```
F4  1
CF4 2 3 4
```

The flux tally for cell 1 is output twice: first, the total flux in cell 1; and second, the flagged tally, or that portion of the flux caused by particles having passed through cells 2, 3, or 4.

2.5.7.3 Multipliers and Modification

MCNP tallies can be modified in many different ways. The `EM`, `TM`, and `CM` cards multiply the quantities in each energy, time, or cosine bin by a different constant. This capability is useful for modeling response functions or changing units. For example, a surface current tally can have its units changed to per steradian by entering the inverse steradian bin sizes on the `CM` card.

The `DE` and `DF` cards allow modeling of an energy-dependent dose function that is a continuous function of energy from a table whose data points need not coincide with the tally energy bin structure (`E` card).

The `FM` card multiplies the `F1`, `F2`, `F4`, and `F5` tally cards by any continuous-energy quantity available in the data libraries. For example, average heating numbers $H_{\text{avg}}(E)$ and total cross section $\sigma_{\text{t}}(E)$ are stored on the MCNP data libraries. An `F4` tally multiplied by $\sigma_{\text{t}}H_{\text{avg}}(E)\rho_{\text{a}}/\rho_{\text{g}}$ converts it to an `F6` tally, or an `F5` detector tally multiplied by the same quantity calculates heating at a point [§2.5.6.1]. The `FM` card can modify any flux or current tally of the form $\int \varphi(E)\mathrm{d}E$ into $\int R(E)\varphi(E)\mathrm{d}E$, where $R(E)$ is any combination of sums and products of energy-dependent quantities known to the MCNP code.

The `FM` card can also model attenuation. Here the tally is converted to $\int \varphi(E)\exp[-\sigma_{\text{t}}(E)\rho_{\text{a}}x]\mathrm{d}E$, where x is the thickness of the attenuator, ρ_{a} is its atom density, and σ_{t} is its total cross section. Double parentheses allow the calculation of $\int \varphi(E)\exp[-\sigma_{\text{t}}(E)\rho_{a}x]R(E)\mathrm{d}E$. More complex expressions of $\sigma_{\text{t}}(E)\rho_{\text{a}}x$ are allowed so that many attenuators may be stacked. This is useful for calculating attenuation in line-of-sight pipes and through thin foils and detector coatings, particularly when done in conjunction with point and ring detector tallies. Beware, however, that attenuation assumes that the attenuated portion of the tally is lost from the system by capture or escape and cannot be scattered back in.

Two special `FM` card options are available. The first option sets $R(E) = 1/\varphi(E)$ to score tracks or collisions. The second option sets $R(E) = 1/v$ (where v is scalar velocity) to score population or prompt removal lifetime.

2.5.7.4 Special Treatments

A number of special tally treatments are available using the FT card. A brief description of each one follows.

2.5.7.4.1 Change Current Tally Reference Vector

F1 current tallies measure bin angles relative to the surface normal. They can be binned relative to any arbitrary vector defined with the FRV option.

2.5.7.4.2 Gaussian Energy Broadening

The GEB option can be used to better simulate a physical radiation detector in which energy peaks exhibit Gaussian energy broadening. The tallied energy is broadened by sampling from the Gaussian,

$$f(E) = C \exp\left[-\left(\frac{E - E_0}{A}\right)^2\right], \tag{2.204}$$

where

E	is the broadened energy,
E_0	is the unbroadened energy of the tally,
C	is a normalization constant, and
A	is the Gaussian width.

The Gaussian width is related to the full width half maximum (FWHM) by

$$A = \frac{FWHM}{2\sqrt{\ln 2}} \approx 0.60056120439322 \times FWHM. \tag{2.205}$$

The desired FWHM is specified by the user-provided constants, a, b, and c, where

$$FWHM = a + b\sqrt{E + cE^2}. \tag{2.206}$$

The FWHM is defined as $FWHM = 2(E_{FWHM} - E_0)$, where E_{FWHM} is such that $f(E_{FWHM}) = {}^1\!/_2 f(E_0)$ and $f(E_0)$ is the maximum value of $f(E)$.

2.5.7.4.3 Time Convolution

Because the geometry and material compositions are independent of time, except in the case of time-dependent temperatures, the expected tally $T(t, t+\tau)$ at time $t+\tau$ from a source particle emitted at time t is identical to the expected tally $T(0,\tau)$ from a source particle emitted at time 0. Thus, if a calculation is performed with all source particles started at $t = 0$, one has an estimate of $T(0,\tau)$ and the tallies T_{Q_i} from a number of time-distributed sources. $Q_i(t)$ can be calculated at time η as

$$T_{Q_i}(\eta) = \int_a^b Q_i(t)T(t,\eta)\mathrm{d}t = \int_a^b Q_i(t)T(0,\eta - t)\mathrm{d}t \tag{2.207}$$

by sampling t from $Q_i(t)$ and recording each particle's tally (shifted by t), or after the calculation by integrating $Q_i(t)$ multiplied by the histogram estimate of $T(0,\eta - t)$. The latter method is used in the MCNP code to simulate a source as a square pulse starting at time a and ending at time b, where a and b are supplied by the TMC option.

2.5.7.4.4 Binning by the Number of Collisions

Tallies can be binned by the number of collisions that caused them with the `INC` option and an `FU` card. A current tally, for example, can be subdivided into the portions of the total current coming from particles that have undergone zero, one, two, three, ... collisions before crossing the surface. In a point detector tally, the user can determine what portion of the score came from particles having their 1st, 2nd, 3rd, ... collision. Collision binning is particularly useful with the exponential transform because the transform reduces variance by reducing the number of collisions.

⚠ Caution

If particles undergoing many collisions are the major contributor to a tally, then the exponential transform is ill-advised. When the exponential transform is used, the portion of the tally coming from particles having undergone many collisions should be small.

2.5.7.4.5 Binning by Detector Cell

The `ICD` option with an `FU` card is used to determine what portion of a detector tally comes from what cells. This information is similar to the detector diagnostics print, but the `FT` card can be combined with energy and other binning cards. The contribution to the normalized rather than unnormalized tally is printed.

2.5.7.4.6 Binning by Source Distribution

The `SCX` and `SCD` options are used to bin a tally score according to what source distribution caused it.

2.5.7.4.7 Binning by Multigroup Particle Type

The `PTT` option with an `FU` card is used to bin multigroup tallies by particle type. The MCNP multigroup treatment is available for neutron, coupled neutron/photon, and photon problems. However, charged particles or any other combinations of particles can be run with the various particles masquerading as neutrons and are printed out in the MCNP output file as if they were neutrons. With the `PTT` option, the tallies can be segregated into particle types by entering atomic weights in units of MeV on the `FU` card. The `FU` atomic weights must be specified to within 0.1% of the true atomic weight in MeV units; thus `FU 0.511` specifies an electron, but 0.510 is not recognized.

2.5.7.4.8 Binning by Particle Charge

The `ELC` option allows binning `F1` current tallies by particle charge. There are three `ELC` options:

1. Cause negative electrons to make negative scores and positrons to make positive scores. Note that by tallying positive and negative numbers the relative error is unbounded and this tally may be difficult to converge.
2. Segregate electrons and positrons into separate bins plus a total bin. There will be three bins (positron, electron, and total) all with positive scores. The total bin will be the same as the single tally bin without the `ELC` option.

3. Segregate electrons and positrons into separate bins plus a total bin, with the electron bin scores being all negative to reflect their charge. The bins will be for positrons (positive scores), electrons (negative scores), and total. The total bin will be the same as the single bin with the first `ELC` option above (usually with negative scores because there are more electrons than positrons).

2.5.7.5 User Modification

If the above capabilities do not provide exactly what is desired, tallies can be modified by a user-supplied **`TALLYX`** subroutine (`FU` card). As with a user-supplied **`SOURCE`** subroutine, which lets the users provide their own specialized source, the **`TALLYX`** subroutine lets the user modify any tally, with all the programming changes conveniently located in a single subroutine.

2.5.7.6 Tally Output Format

Not only can users change the contents of MCNP tallies, the output format can be modified as well. Any desired descriptive comment can be added to the tally title by the tally comment (`FC`) card. The printing order can be changed (`FQ` card) so that instead of, for instance, getting the default output blocks in terms of time vs. energy, they could be printed in blocks of segment vs. cosine. The tally bin that is monitored for the tally fluctuation chart printed at the problem end and used in the statistical analysis of the tally can be selected (`TF` card). Detector tally diagnostic prints are controlled with the `DD` card. Finally, the `PRINT` card controls what optional tables are displayed in the output file.

2.6 Estimation of the Monte Carlo Precision

Monte Carlo results represent an average of the contributions from many histories sampled during the problem. An important quantity equal in stature to the Monte Carlo answer (or tally) itself is the statistical error or uncertainty associated with the result. The importance of this error and its behavior versus the number of histories cannot be overemphasized because the user not only gains insight into the quality of the result, but also can determine if a tally appears statistically well behaved. If a tally is not well behaved, the estimated error associated with the result generally will not reflect the true confidence interval of the result and, thus, the answer could be completely erroneous. The MCNP code contains several quantities that aid the user in assessing the quality of the confidence interval [146].

The purpose of this section is to educate MCNP users about the proper interpretation of the MCNP estimated mean, relative error, variance of the variance, and history score probability density function. Carefully check tally results and the associated tables in the tally fluctuation charts to ensure a well-behaved and properly converged tally.

2.6.1 Monte Carlo Means, Variances, and Standard Deviations

Monte Carlo results are obtained by sampling possible random walks and assigning a score x_i (for example, x_i is the energy deposited by the ith random walk) to each random walk. Random walks typically will produce a range of scores depending on the tally selected and the variance reduction chosen.

Suppose $f(x)$ is the history score probability density function for selecting a random walk that scores x to the tally being estimated. The true answer (or mean) is the expected value of x, $E(x)$, where

$$E(x) = \int x\, f(x) \mathrm{d}x = \text{true mean}. \tag{2.208}$$

The function $f(x)$ is seldom explicitly known; thus, $f(x)$ is implicitly sampled by the Monte Carlo random walk process. The true mean $E(x)$ then is estimated by the sample mean $\overline{x}$ where

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{2.209}$$

where x_i is the value of x selected from $f(x)$ for the ith history and N is the number of histories calculated in the problem. The Monte Carlo mean $\overline{x}$ is the average value of the scores x_i for all the histories calculated in the problem. The relationship between $E(x)$ and $\overline{x}$ is given by the Strong Law of Large Numbers [17] that states that if $E(x)$ is finite, $\overline{x}$ tends to the limit $E(x)$ as N approaches infinity.

The variance of the population of x values is a measure of the spread in these values and is given by [17]:

$$\sigma^2 = \int [x - E(x)]^2 f(x)\mathrm{d}x = E\left(x^2\right) - [E(x)]^2. \tag{2.210}$$

The square root of the variance is σ, which is called the standard deviation of the population of scores. As with $E(x)$, σ is seldom known but can be estimated by Monte Carlo as S

$$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{2.211}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{2.212}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}\left(x_i^2 - 2\overline{x}x_i + \overline{x}^2\right) \tag{2.213}$$

$$\approx \frac{1}{N}\left[\sum_{i=1}^{N} x_i^2 - 2\overline{x}\left(\sum_{i=1}^{N} x_i\right) + \overline{x}^2\left(\sum_{i=1}^{N} 1\right)\right] \tag{2.214}$$

$$\approx \frac{N}{N}\left[\frac{1}{N}\sum_{i=1}^{N} x_i^2 - 2\overline{x}\left(\frac{1}{N}\sum_{i=1}^{N} x_i\right) + \overline{x}^2\left(\frac{1}{N}\sum_{i=1}^{N} 1\right)\right] \tag{2.215}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} x_i^2 - 2(\overline{x})(\overline{x}) + \overline{x}^2 \tag{2.216}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} x_i^2 - 2\overline{x}^2 + \overline{x}^2 \tag{2.217}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} x_i^2 - \overline{x}^2 \tag{2.218}$$

$$\approx \overline{x^2} - \overline{x}^2 \tag{2.219}$$

where

$$\overline{x^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2 \tag{2.220}$$

and more generally

$$\overline{x^m} = \frac{1}{N}\sum_{i=1}^{N} x_i^m. \tag{2.221}$$

When the number of samples is large, the approximation introduced by removing Bessel's correction ($N-1 \to N$) as done in Eq. (2.212) is generally valid. This approach is used within the MCNP code for calculations

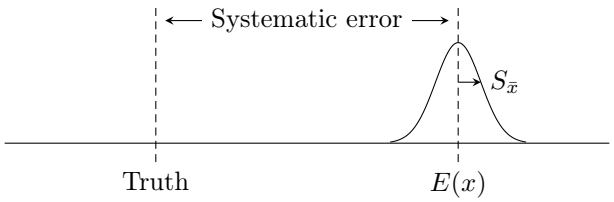


Figure 2.16: Inaccuracy caused by systematic error versus statistical precision.

that rely on history-based statistics. For components that compute batch statistics, where N is not expected to be very large, Bessel's correction is retained.

The quantity S is the estimated standard deviation of the population of x based on the values of x_i that were actually sampled (i.e., S is the sample standard deviation).

The estimated variance of $\overline{x}$ is given by

$$S_{\overline{x}}^2 = \frac{S^2}{N}. \tag{2.222}$$

These formulas do not depend on any restriction on the distribution of x or $\overline{x}$ (such as normality) beyond requiring that $E(x)$ and σ^2 exist and are finite. The estimated standard deviation of the mean $\overline{x}$ is given by $S_{\overline{x}}$.

It is important to note that $S_{\overline{x}}$ is proportional to $1/\sqrt{N}$, which is the inherent drawback to the Monte Carlo method. To halve $S_{\overline{x}}$, four times the original number of histories must be calculated, a calculation that can be computationally expensive. The quantity $S_{\overline{x}}$ can also be reduced for a specified N by making S smaller, reducing the inherent spread of the tally results. This can be accomplished by using variance reduction techniques such as those discussed in §2.7.

2.6.2 Precision and Accuracy

There is an extremely important difference between precision and accuracy of a Monte Carlo calculation. As illustrated in Figure 2.16, precision is the uncertainty in $\overline{x}$ caused by the statistical fluctuations of the x_is for the portion of physical phase space sampled by the Monte Carlo process. Important portions of physical phase space might not be sampled because of problem cutoffs in time or energy, inappropriate use of variance reduction techniques, or an insufficient sampling of important low-probability events. Accuracy is a measure of how close the expected value of $\overline{x}$, $E(x)$, is to the true physical quantity being estimated. The difference between this true value and $E(x)$ is called the systematic error, which is seldom known. Error or uncertainty estimates for the results of Monte Carlo calculations refer only to the precision of the result and not to the accuracy. It is possible to calculate a highly precise result that is far from the physical truth because nature has not been modeled faithfully.

2.6.2.1 Factors Affecting Problem Accuracy

Three factors affect the accuracy of a Monte Carlo result: (1) the code and data, (2) problem modeling, and (3) the user. Code factors encompass: the physics features included in a calculation as well as the mathematical models used; uncertainties in the data, such as the transport and reaction cross sections, Avogadro's number, atomic weights, etc.; the quality of the representation of the differential cross sections in energy and angle; and coding errors (bugs). All of the applicable physics must be included in a calculation to produce accurate results. Even though the evaluations are not perfect, more faithful representation of the evaluator's data should produce more accurate results. The descending order of preference for Monte

Carlo data for calculations is continuous energy, thinned continuous energy, discrete reaction, and multigroup. Coding errors can always be a problem because no large code is bug-free. The MCNP code, however, is a very mature and heavily used production code. With steadily increasing use over the years, the likelihood of a serious coding error continues to diminish.

The second area, problem-modeling factors, can quite often contribute to a decrease in the accuracy of a calculation. Many calculations produce seemingly poor results because the model of the energy and angular distribution of the radiation source is not adequate. Two other problem-modeling factors affecting accuracy are the geometric description and the physical characteristics of the materials in the problem.

The third general area affecting calculational accuracy involves user errors in the problem input or in user-supplied subroutines and patches to the MCNP code. The user can also abuse variance reduction techniques such that portions of the physical phase space are not allowed to contribute to the results. Checking the input and output carefully can help alleviate these difficulties. A last item that is often overlooked is a user's thorough understanding of the relationship of the Monte Carlo tallies to any measured quantities being calculated. Factors such as detector efficiencies, data reduction and interpretation, etc., must be completely understood and included in the calculation, or the comparison is not meaningful.

2.6.2.2 Factors Affecting Problem Precision

The precision of a Monte Carlo result is affected by four user-controlled choices: (1) forward vs. adjoint calculation, (2) tally type, (3) variance reduction techniques, and (4) number of histories run.

The choice of a forward vs. adjoint calculation depends mostly on the relative sizes of the source and detector regions. Starting particles from a small region is easy to do, whereas transporting particles to a small region is generally hard to do. Because forward calculations transport particles from source to detector regions, forward calculations are preferable when the detector (or tally) region is large and the source region is small. Conversely, because adjoint calculations transport particles backward from the detector region to the source region, adjoint calculations are preferable when the source (or tally) region is large and the detector region is small. The MCNP code can be run in multigroup adjoint mode. There is no continuous-energy adjoint capability.

As alluded to above, the smaller the tally region, the harder it becomes to get good tally estimates. An efficient tally will average over as large a region of phase space as practical. In this connection, tally dimensionality is extremely important. A one-dimensional tally is typically 10 to 100 times easier to estimate than a two-dimensional tally, which is 10 to 100 times easier than a three-dimensional tally. This fact is illustrated in Fig. 2.22 later in this section.

Variance reduction techniques can be used to improve the precision of a given tally by increasing the nonzero tallying efficiency and by decreasing the spread of the nonzero history scores. These two components are depicted in a hypothetical $f(x)$ shown in Fig. 2.17. See §2.6.8 for more discussion about the empirical $f(x)$ for each tally fluctuation chart bin. A calculation will be more precise when the history-scoring efficiency is high and the variance of the nonzero scores is low. The user should strive for these conditions in difficult Monte Carlo calculations. Examples of these two components of precision are given in §2.6.6.

More histories can be run to improve precision [§2.6.3]. Because the precision is proportional to $1/\sqrt{N}$, running more particles is often costly in computer time and therefore is viewed as the method of last resort for difficult problems.

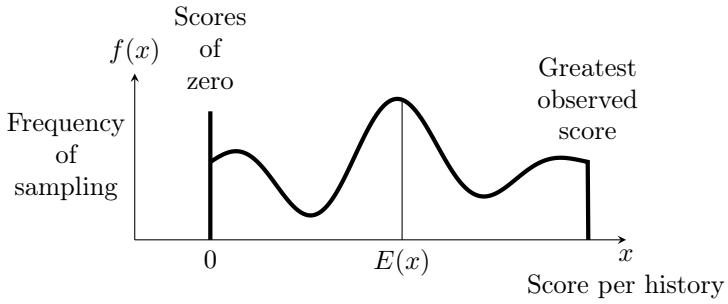


Figure 2.17: Hypothetical history-score probability density function.

2.6.3 Monte Carlo Confidence Intervals and the Central Limit Theorem

To define confidence intervals for the precision of a Monte Carlo result, the Central Limit Theorem [17] of probability theory is used, stating that

$$\lim_{N\to\infty} \Pr\left[E(x)+\alpha\frac{\sigma}{\sqrt{N}} < \overline{x} < E(x)+\beta\frac{\sigma}{\sqrt{N}}\right] = \frac{1}{\sqrt{2\pi}}\int_{\alpha}^{\beta}\exp\left(-\frac{t^2}{2}\right)\mathrm{d}t, \tag{2.223}$$

where α and β can be any arbitrary values and $\Pr[Z]$ means the probability of Z. In terms of the estimated standard deviation of $\overline{x}$, $S_{\overline{x}}$, this may be rewritten in the following approximation for large N:

$$\Pr\left(\left[\alpha S_{\overline{x}} < \frac{\overline{x}-E(x)}{\sigma/\sqrt{N}} < \beta S_{\overline{x}}\right] \approx \frac{1}{\sqrt{2\pi}}\int_{\alpha}^{\beta}\exp\left(-\frac{t^2}{2}\right)\mathrm{d}t\right). \tag{2.224}$$

This crucial theorem states that for large values of N (that is, as N tends to infinity) and identically distributed independent random variables x_i with finite means and variances, the distribution of the $\overline{x}$s approaches a normal distribution. Therefore, for any distribution of tallies (an example is shown in Figure 2.17), the distribution of resulting $\overline{x}$s will be approximately normally distributed, as shown in Figure 2.16, with a mean of $E(x)$. If S is approximately equal to σ, which is valid for a statistically significant sampling of a tally (that is, N has tended to infinity), then

$$\overline{x} - S_{\overline{x}} < E(x) < \overline{x} + S_{\overline{x}}\,, \sim 68\% \text{ of the time and} \tag{2.225a}$$

$$\overline{x} - 2S_{\overline{x}} < E(x) < \overline{x} + 2S_{\overline{x}}\,, \sim 95\% \text{ of the time} \tag{2.225b}$$

from standard tables for the normal distribution function. Equation (2.225a) is a 68% confidence interval and Eq. (2.225b) is a 95% confidence interval.

The key point about the validity of these confidence intervals is that the physical phase space must be adequately sampled by the Monte Carlo process. If an important path in the geometry or a window in the cross sections, for example, has not been well sampled, both $\overline{x}$ and $S_{\overline{x}}$ will be unknowingly incorrect and the results will be wrong, usually tending to be too small. The user must take great care to be certain that adequate sampling of the source, transport, and any tally response functions have indeed taken place. Additional statistical quantities to aid in the assessment of proper confidence intervals are described in later portions of this section beginning in §2.6.9.1.

2.6.4 Estimated Relative Errors in the MCNP Code

All standard MCNP tallies are normalized to be per starting particle history (except for some criticality calculations) and are printed in the output with a second number, which is the estimated relative error

Table 2.5: Estimated Relative Error R vs. Number of Identical Tallies n for Large N

n	1	4	16	25	100	400
R	1.0	0.5	0.25	0.20	0.10	0.05

defined as

$$R \equiv \frac{S_{\overline{x}}}{\overline{x}}. \tag{2.226a}$$

The relative error is a convenient number because it represents statistical precision as a fractional result with respect to the estimated mean.

Combining Eqs. (2.209), (2.219), and (2.220), R can be written (for large N) as

$$R = \sqrt{\frac{1}{N}\left(\frac{\overline{x^2}}{\overline{x}^2} - 1\right)} = \sqrt{\frac{\sum_{i=1}^{N} x_i^2}{\left(\sum_{i=1}^{N} x_i\right)^2} - \frac{1}{N}}. \tag{2.226b}$$

Several important observations about the relative error can be made from Eq. (2.226b). First, if all the x_is are nonzero and equal, then R is zero. Thus, low-variance solutions should strive to reduce the spread in the x_is. If the x_is are all zero, then R is defined to be zero. If only one nonzero score is made, R approaches unity as N becomes large. Therefore, for x_is of the same sign, $S_{\overline{x}}$ can never be greater than $\overline{x}$ because R never exceeds unity. For positive and negative x_is, R can exceed unity. The range of R values for x_is of the same sign is therefore between zero and unity.

To determine what values of R lead to results that can be stated with confidence, consider Eq. (2.226b) for a difficult problem in which nonzero scores occur very infrequently. In this case,

$$\frac{1}{N} \ll \frac{\sum_{i=1}^{N} x_i^2}{\left(\sum_{i=1}^{N} x_i\right)^2}. \tag{2.227a}$$

For clarity, assume that there are n out of N ($n \ll N$) nonzero scores that are identical and equal to x. With these two assumptions, R for "difficult problems" becomes

$$R_{\text{D.P.}} \sim \sqrt{\frac{nx^2}{n^2x^2}} = \frac{1}{\sqrt{n}}\,,\ n \ll N \tag{2.227b}$$

This result is expected because the limiting form of a binomial distribution with infrequent nonzero scores and large N is the Poisson distribution used in detector "counting statistics."

Through use of Eq. (2.227b), a table of R values versus the number of tallies or "counts" can be generated as shown in Table 2.5. A relative error of 0.5 is the equivalent of four counts, which is hardly adequate for a statistically significant answer. Sixteen counts is an improvement, reducing R to 0.25, but still is not a large number of tallies. The same is true for n equals 25. When n is 100, R is 0.10, so the results should be much improved. With 400 tallies, an R of 0.05 should be quite good indeed, except possibly for point-detector and ring-detector tallies.

Based on this qualitative analysis and the experience of Monte Carlo practitioners, Table 2.6 presents the recommended interpretation of the estimated 1σ confidence interval $\overline{x}(1 \pm R)$ for various values of R associated with an MCNP tally. These guidelines were determined empirically, based on years of experience using the MCNP code on a wide variety of problems. Just before the tally fluctuation charts, a "Status of Statistical Checks" table prints how many tally bins of each tally have values of R exceeding these recommended guidelines.

Table 2.6: Guidelines for Interpreting the Relative Error, R^*.

Range of R	Quality of the Tally
0.50 to 1.00	Not meaningful
0.20 to 0.50	Factor of a few
0.10 to 0.20	Questionable
<0.10	Generally reliable
<0.05	Generally reliable for point detectors

*$R = S_{\overline{x}}/\overline{x}$ and represents the estimated relative error at the 1σ level. These interpretations of R assume that all portions of the problem phase space are being sampled well by the Monte Carlo process.

Point detector tallies generally require a smaller value of R for valid confidence interval statements because some contributions, such as those near the detector point, are usually extremely important and may be difficult to sample well. Experience has shown that for R less than 0.05, point detector results are generally reliable. For an R of 0.10, point detector tallies may only be known within a factor of a few and sometimes not that well (see the pathological example §2.6.10).

The MCNP code calculates the relative error for each tally bin in the problem using Eq. (2.226b). Each x_i is defined as the total contribution from the ith starting particle and all resulting progeny. This definition is important in many variance reduction methods, multiplying physical processes such as fission or (n,xn) neutron reactions that create additional neutrons, and coupled neutron/photon/ electron problems. The ith source particle and its offspring may thus contribute many times to a tally and all of these contributions are correlated because they are from the same source particle. The x_is are all independent from each other.

Figure 2.18 represents the MCNP process of calculating the first and second moments of each tally bin and relevant totals using three tally storage blocks of equal length. The hypothetical grid of tally bins in the bottom half of Figure 2.18 has 24 tally bins including the time and energy totals. During the course of the ith history, sums are performed in the first MCNP tally storage block. Some of the tally bins receive no contributions and others receive one or more contributions. At the conclusion of the ith history, the sums are added to the second MCNP tally storage block. The sums in the first MCNP tally storage block are squared and added to the third tally storage block. The first tally storage block is then filled with zeros and history $i+1$ begins. After the last history N, the estimated tally means are computed using the second MCNP tally storage block and Eq. (2.209). The estimated relative errors are calculated using the second and third MCNP tally storage blocks and Eq. (2.226b). This method of estimating the statistical uncertainty of the result produces the best estimate because the batch size is one, which minimizes the variance of the variance [147–149].

Note that there is no guarantee that the estimated relative error will decrease inversely proportional to the N as required by the Central Limit Theorem because of the statistical nature of the tallies. Early in the problem, R will generally have large statistical fluctuations. Later, infrequent large contributions may cause fluctuations in $S_{\overline{x}}$ and to a lesser extent in $\overline{x}$ and therefore in R. The MCNP code calculates a figure of merit for one bin of each numbered tally to aid the user in determining the statistical behavior as a function of N and the efficiency of the tally.

2.6.5 MCNP Figure of Merit

The estimated relative error squared, R^2, should be proportional to $1/N$, as shown by Eq. (2.226b). The computer time T used in an MCNP problem should be directly proportional to N; therefore, R^2T should be approximately a constant within any one Monte Carlo calculation. It is convenient to define a FOM of a tally to be

$$FOM \equiv \frac{1}{R^2 T}. \tag{2.228a}$$

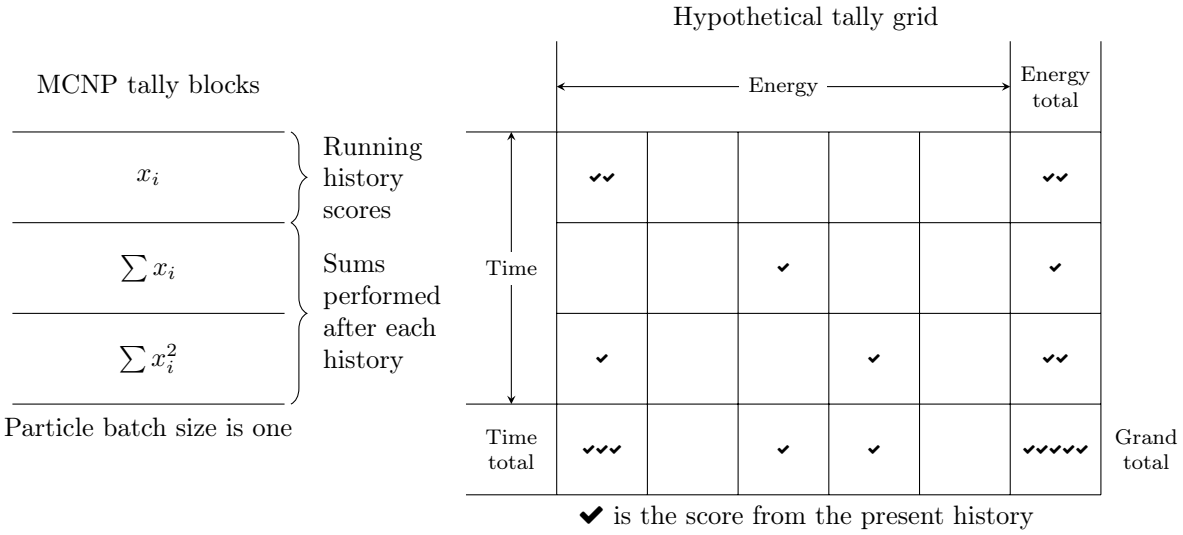


Figure 2.18: Hypothetical energy-time-binned tally scores.

Table 2.7: R Values as a Function of the FOM for $T = 1$ Minute

FOM	1	10	100	1000	10000
R	1.0	0.32	0.10	0.032	0.010

The MCNP code prints the FOM for the Tally Fluctuation Chart (TFC) bin of each numbered tally as a function of N, where the unit of computer time T is minutes. The table is printed in particle increments of 1000 up to 20,000 histories. Between 20,000 and 40,000 histories, the increment is doubled to 2000. This trend continues, producing a table of up to 20 entries. The default increment can be changed by the 5th entry on the PRDMP card.

The FOM is a very important statistic about a tally bin and should be studied by the user. It is a tally reliability indicator in the sense that if the tally is well behaved, the FOM should be approximately a constant with the possible exception of statistical fluctuations very early in the problem. An order-of-magnitude estimate of the expected fractional statistical fluctuations in the FOM is $2R$. This result assumes that both the relative statistical uncertainty in the relative error is of the order of the relative error itself and the relative error is small compared to unity. The user should *always* examine the tally fluctuation charts at the end of the problem to check that the FOMs are approximately constant as a function of the number of histories for each tally.

The numerical value of the FOM can be better appreciated by considering the relation

$$R = \frac{1}{\sqrt{FOM \cdot T}}. \tag{2.228b}$$

Table 2.7 shows the expected value of R that would be produced in a one-minute problem ($T = 1$) as a function of the value of the FOM. It is clearly advantageous to have a large FOM for a problem because the computer time required to reach a desired level of precision is proportionally reduced. Examination of Eq. (2.228a) shows that doubling the FOM for a problem will reduce the computer time required to achieve the same R by a factor of two.

Another interpretation for the FOM involves defining the problem's particle computation rate t as

$$t = \frac{N}{T}, \tag{2.228c}$$

where t is the number of particles per minute for a problem on a specific computer and N is the number of particles run in the problem. Substituting Eq. (2.228c) into Eq. (2.228a) and using Eqs. (2.209), (2.222), and (2.226a), the FOM becomes

$$FOM = t\left(\frac{\overline{x}}{S}\right)^2 \tag{2.228d}$$

where S is the sample standard deviation (not the estimated standard deviation of the mean, $S_{\overline{x}}$). The squared quantity is a ratio of the desired result divided by a measure of the spread in the sampled values. This ratio is called the tally signal-to-noise ratio:

$$\text{signal-to-noise ratio} = \frac{\overline{x}}{S}. \tag{2.228e}$$

The quantity $\overline{x}/S$ approaches the expected value of the signal-to-noise ratio for a problem tally bin as N becomes large. Using Eq. (2.228e), the FOM becomes

$$FOM = t(\text{signal-to-noise ratio})^2. \tag{2.228f}$$

The FOM is directly proportional to the particles per minute t (as would be expected) and the tally bin signal-to-noise ratio squared. The tally bin signal-to-noise ratio is dependent on the shape of the underlying history score probability density function $f(x)$ for the tally bin [§2.6.8]. To increase the FOM, t and/or the signal-to-noise ratio can be increased. Because $\overline{x}$ should be the same for the problems with different variance reduction, increasing the FOM is equivalent to increasing t/S^2 (decreasing S with variance reduction techniques often decreases t). It is usually worthwhile to optimize the tally efficiency by intelligently running various variance reduction methods and using the largest FOM consistent with good phase-space sampling (good sampling can often be inferred by examining the cell particle activity in `PRINT` Table 126). The MCNP code prints both the empirical $f(x)$ and signal-to-noise ratio for the tally fluctuation chart bin of each tally in `PRINT` Table 161.

In summary, the FOM has three uses. One important use is as a tally reliability indicator. If the FOM is not approximately a constant (except for statistical fluctuations early in the problem), the confidence intervals may not overlap the expected score value, $E(x)$, the expected fraction of the time (see Eqs. (2.225a) and (2.225b)). A second use for the FOM is to optimize the efficiency of the Monte Carlo calculation by making several short test runs with different variance reduction parameters and then selecting the problem with the largest FOM. Remember that the statistical behavior of the FOM (that is, R) for a small number of histories may cloud the selection of techniques competing at the same level of efficiency. A third use for the FOM is to estimate the computer time required to reach a desired value of R by using $T \sim 1/\left(R^2 FOM\right)$.

2.6.6 Separation of Relative Error into Two Components

Three factors that affect the efficiency of a Monte Carlo calculation are (1) history-scoring efficiency, (2) dispersions in non-zero history scores, and (3) computer time per history. All three factors are included in the FOM. The first two factors control the value of R; the third is T.

The relative error can be separated into two components: the non-zero history-scoring efficiency component R^2_{eff} and the intrinsic spread of the nonzero x_i scores R^2_{int}. Defining q to be the fraction of histories producing nonzero x_is, Eq. (2.220) can be rewritten as

$$R = \frac{\sum_{i=1}^{N} x_i^2}{\left(\sum_{i=1}^{N} x_i\right)^2} - \frac{1}{N} = \frac{\sum_{x_i \neq 0} x_i^2}{\left(\sum_{x_i \neq 0} x_i\right)^2} - \frac{1}{N} = \frac{\sum_{x_i \neq 0} x_i^2}{\left(\sum_{x_i \neq 0} x_i\right)^2} - \frac{1}{qN} + \frac{1-q}{qN}. \tag{2.229a}$$

Table 2.8: Expected Values of R_{eff} as a Function of the Fraction of Histories Producing Non-zero Scores (q) and the Number of Histories (N)

N	q			
	0.001	0.01	0.1	0.5
10^3	0.999	0.315	0.095	0.032
10^4	0.316	0.099	0.030	0.010
10^5	0.100	0.031	0.009	0.003
10^6	0.032	0.010	0.003	0.001

Note by Eq. (2.220) that the first two terms are the relative error of the qN non-zero scores. Thus defining,

$$R_{\text{int}}^2 = \frac{\sum_{x_i \neq 0} x_i^2}{\left(\sum_{x_i \neq 0} x_i\right)^2} - \frac{1}{qN}, \tag{2.229b}$$

$$R_{\text{eff}}^2 = \frac{1-q}{qN}, \tag{2.229c}$$

$$R^2 = R_{\text{eff}}^2 + R_{\text{int}}^2. \tag{2.229d}$$

For identical nonzero x_is, R_{int}^2 is zero and for a 100% scoring efficiency, R_{eff}^2 is zero. It is usually possible to increase q for most problems using one or more of the MCNP variance reduction techniques. These techniques alter the random walk sampling to favor those particles that produce a nonzero tally. The particle weights are then adjusted appropriately so that the expected tally is preserved. This topic is described in §2.7. The sum of the two terms of Eq. (2.229d) produces the same result as Eq. (2.220). Both R_{int}^2 and R_{eff}^2 are printed for the tally fluctuation chart bin of each tally so that the dominant component of R can be identified as an aid to making the calculation more efficient.

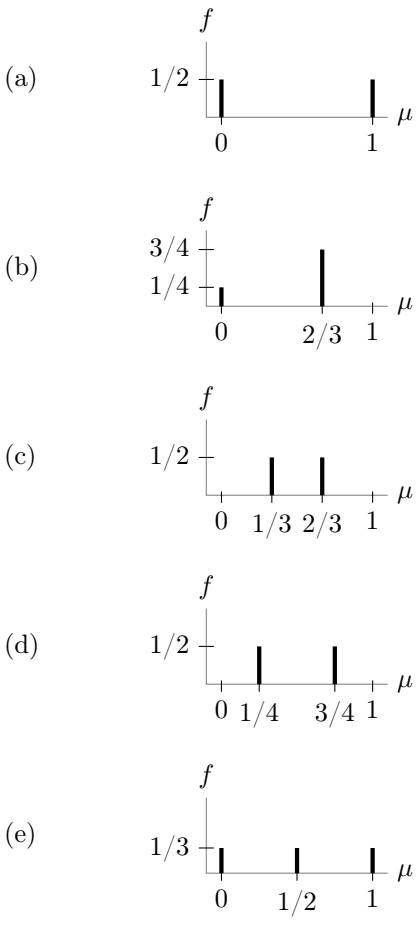
These equations can be used to better understand the effects of scoring inefficiency; that is, those histories that do not contribute to a tally. Table 2.8 shows the expected values of R_{eff} as a function of q and the number of histories N. This table is appropriate for identical nonzero scores and represents the theoretical minimum relative error possible for a specified q and N. It is no surprise that small values of q require a correspondingly large number of particles to produce precise results.

A practical example of scoring inefficiency is the case of infrequent high-energy particles in a down-scattering-only problem. If only a small fraction of all source particles has an energy in the highest energy tally bin, the dominant component of the relative error will probably be the scoring efficiency because only the high-energy source particles have a nonzero probability of contributing to the highest energy bin. For problems of this kind, it is often useful to run a separate problem starting only high-energy particles from the source and to raise the energy cutoff. The much improved scoring efficiency will result in a much larger FOM for the high-energy tally bins.

To further illustrate the components of the relative error, consider the five examples of selected discrete probability density functions shown in Fig. 2.19. Cases (a) and (b) have no dispersion in the nonzero scores, cases (c) and (d) have 100% scoring efficiency, and case (e) contains both elements contributing to R. The most efficient problem is case (c). Note that the scoring inefficiency contributes 75% to R in case (e), the second worst case of the five.

2.6.7 Variance of the Variance

Previous sections have discussed the relative error R and figure of merit FOM as measures of the quality of the mean. A quantity called the relative variance of the variance (VOV) is another useful tool that can assist the user in establishing more reliable confidence intervals. The VOV is the estimated relative variance



(a)
$E(\mu) = 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{1}{2}$
$R = R_{\text{eff}} = 1/\sqrt{N}$
$R_{\text{int}} = 0$

(b)
$E(\mu) = 0 \cdot \frac{1}{4} + \frac{2}{3} \cdot \frac{3}{4} = \frac{1}{2}$
$R = R_{\text{eff}} = 0.58/\sqrt{N}$
$R_{\text{int}} = 0$

(c)
$E(\mu) = \frac{1}{3} \cdot \frac{1}{2} + \frac{2}{3} \cdot \frac{1}{2} = \frac{1}{2}$
$R = R_{\text{int}} = 0.33/\sqrt{N}$
$R_{\text{eff}} = 0$

(d)
$E(\mu) = \frac{1}{4} \cdot \frac{1}{2} + \frac{3}{4} \cdot \frac{1}{2} = \frac{1}{2}$
$R = R_{\text{int}} = 0.5/\sqrt{N}$
$R_{\text{eff}} = 0$

(e)
$E(\mu) = 0 \cdot \frac{1}{3} + \frac{1}{2} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} = \frac{1}{2}$
$R = 0.82/\sqrt{N}$
$R_{\text{int}} = 0.41/\sqrt{N}$ 25%
$R_{\text{eff}} = 0.71/\sqrt{N}$ 75%

Figure 2.19: Five various distributions with an identical mean of 0.5.

of the estimated R. The VOV involves the estimated third and fourth moments of the empirical history score PDF $f(x)$ and is much more sensitive to large history score fluctuations than is R. The magnitude and trend of the VOV versus the number of particle histories are indicators of tally fluctuation chart (TFC) bin convergence. Early work was done by Estes and Cashwell [147] and Pederson [150] later reinvestigated this statistic to determine its usefulness. Pederson concluded [146] that the VOV is a much better indicator of confidence interval validity than R.

The VOV is a quantity that is analogous to the square of the R of the mean, except it is for R instead of the mean. The estimated relative VOV of the mean is defined as

$$\text{VOV} = \frac{S^2\left(S_{\overline{x}}^2\right)}{S_{\overline{x}}^4}, \tag{2.230}$$

where $S_{\overline{x}}^2$ is the estimated variance of $\overline{x}$ and $S^2\left(S_{\overline{x}}^2\right)$ is the estimated variance in $S_{\overline{x}}^2$. The VOV is a measure of the relative statistical uncertainty in the estimated R and is important because S must be a good approximation of σ to use the Central Limit Theorem to form confidence intervals.

The VOV for a tally bin [150] is

$$\text{VOV} = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^4}{\left[\sum_{i=1}^{N}(x_i - \overline{x})^2\right]^2} - \frac{1}{N}. \tag{2.231}$$

This is the fourth central moment minus the second central moment squared, normed by the product of N and the second central moment squared.

When Eq. (2.231) is expanded in terms of sums of powers of x_i, it becomes

$$\text{VOV} = \frac{\sum x_i^4 - \frac{4}{N}\sum x_i \sum x_i^3 + \frac{6}{N^2}\sum x_i^2 \left(\sum x_i\right)^2 - \frac{3}{N^3}\left(\sum x_i\right)^4}{\left(\sum x_i^2 - \frac{\left(\sum x_i\right)^2}{N}\right)^2} - \frac{1}{N} \tag{2.232}$$

or

$$\text{VOV} = \frac{\sum x_i^4 - \frac{4}{N}\sum x_i \sum x_i^3 + \frac{8}{N^2}\sum x_i^2 \left(\sum x_i\right)^2 - \frac{4}{N^3}\left(\sum x_i\right)^4 - \frac{1}{N}\left(\sum x_i^2\right)^2}{\left(\sum x_i^2 - \frac{\left(\sum x_i\right)^2}{N}\right)^2}. \tag{2.233}$$

Now consider the truncated Cauchy formula for the following analysis. The truncated Cauchy is similar in shape to some difficult Monte Carlo tallies. After numerous statistical experiments on sampling a truncated positive Cauchy distribution,

$$\text{Cauchy } f(x) = \frac{2}{\pi(1+x^2)}, \; 0 \le x \le x_{\max}, \tag{2.234}$$

it is concluded that the VOV should be below 0.1 to improve the probability of forming a reliable confidence interval [146]. The quantity 0.1 is a convenient value and is why the VOV is used for the statistical check and not the square root of the VOV. Multiplying numerator and denominator of Eq. (2.233) by $1/N$ converts the terms into $\overline{x}^n$, averages, and shows that the VOV is expected to decrease as $1/N$.

It is interesting to examine the VOV for the n identical history scores x ($n \ll N$) that were used to analyze R in Table 2.5. The VOV behaves as $1/N$ in this limit. Therefore, ten identical history scores would be enough to satisfy the VOV criterion, a factor of at least ten less than the R criterion. There are two reasons for this phenomenon: (1) the VOV is a squared quantity, so it is naturally smaller; and (2) the history scores will ordinarily not be identical and thus the fourth-moment terms in the VOV will increase rapidly over the second-moment terms in R.

The behavior of the VOV as a function of N for the TFC bin is printed in the MCNP output file. Because the VOV involves third and fourth moments, the VOV is a much more sensitive indicator to large history

scores than the R, which is based on first and second moments. The desired VOV behavior is to decrease inversely with N. This criterion is deemed to be a necessary, but not sufficient, condition for a statistically well-behaved tally result. A tally with a VOV that matches this criteria is NOT guaranteed to produce a high quality confidence interval because under sampling of high scores will also underestimate the higher score moments.

To calculate the VOV of every tally bin, put a nonzero 15th entry on the DBCN card. This option creates two additional history score moment tables to sum x_i^3 and x_i^4 (see Fig. 2.18). This option is not the default because it increases tally storage, which could be prohibitive for a problem with many tally bins. The magnitude of the VOV in each tally bin is reported in the "Status of Statistical Checks" table. History-dependent checks of the VOV of all tally bins can be done by printing the tallies to the output file at some frequency using the first entry on the PRDMP card.

2.6.8 Empirical History-score Probability Density Function $f(x)$

2.6.8.1 Introduction

This section discusses another statistic that is useful in assessing the quality of confidence intervals from Monte Carlo calculations. Consider a generic Monte Carlo problem with difficult to sample, but extremely important, large history scores. This type of problem produces three possible scenarios [146].

The first, and obviously desired, case is a correctly converged result that produces a statistically correct confidence interval. The second case is the sampling of an infrequent, but very large, history score that causes the mean and R to increase and the FOM to decrease significantly. This case is easily detectable by observing the behavior of the FOM and the R in the TFCs.

The third and most troublesome case yields an answer that appears statistically converged based on the accepted guidelines described previously, but in fact may be substantially smaller than the correct result because the large history tallies were not well sampled. This situation of too few large history tallies is difficult to detect. The following sections discuss the use of the empirical history score PDF $f(x)$ to gain insight into the TFC bin result. A pathological example to illustrate the third case follows.

2.6.8.2 The History-score Probability Density Function $f(x)$

A history score posted to a tally bin can be thought of as having been sampled from an underlying and generally unknown history score PDF $f(x)$, where the random variable x is the score from one complete particle history to a tally bin. The history score can be either positive or negative. The quantity $f(x)\mathrm{d}x$ is the probability of selecting a history score between x and $x + \mathrm{d}x$ for the tally bin. Each tally bin will have its own $f(x)$.

The most general form for expressing $f(x)$ mathematically is

$$f(x) = f_c(x) + \sum_{i=1}^{m} p_i \delta(x - x_i), \tag{2.235}$$

where $f_c(x)$ is the continuous non-zero part and $\sum_{i=1}^{m} p_i \delta(x - x_i)$ represents the m different discrete components occurring at x_i with probability p_i. An $f(x)$ could be composed of either or both parts of the distribution. A history score of zero is included in $f(x)$ as the discrete component $\delta(x - 0)$.

By the definition of a PDF,

$$\int_{-\infty}^{\infty} f(x)\mathrm{d}x \equiv 1. \tag{2.236}$$

As discussed in §2.6.1, $f(x)$ is used to estimate the mean, variance, and higher moment quantities such as the VOV.

2.6.8.3 The Central Limit Theorem and $f(x)$

As discussed in §2.6.3, the Central Limit Theorem (CLT) states that the estimated mean will appear to be sampled from a normal distribution with a known standard deviation $\sigma/\sqrt{N}$ when N approaches infinity. In practice, σ is NOT known and must be approximated by the estimated standard deviation S. The major difficulty in applying the CLT correctly to a Monte Carlo result to form a confidence interval is knowing when N has approached infinity.

The CLT requires the first two moments of $f(x)$ to exist. Nearly all MCNP tally estimators (except point detectors with zero neighborhoods in a scattering material and some exponential transform problems) satisfy this requirement. Therefore, the history score PDF $f(x)$ also exists. One can also examine the behavior of $f(x)$ for large history scores to assess if $f(x)$ appears to have been "completely" sampled. If "complete" sampling has occurred, the largest values of the sampled xs should have reached the upper bound (if such a bound exists) or should decrease faster than $1/x^3$ so that $E(x^2) = \int_{-\infty}^{\infty} x^2 f(x)\mathrm{d}x$ exists (σ is assumed to be finite in the CLT). Otherwise, N is assumed not to have approached infinity in the sense of the CLT. This is the basis for the use of the empirical $f(x)$ to assess Monte Carlo tally convergence.

The argument should be made that since S must be a good estimate of σ, the expected value of the fourth history score moment $E(x^2) = \int_{-\infty}^{\infty} x^4 f(x)\mathrm{d}x$ should exist. It will be assumed that only the second moment needs to exist so that the $f(x)$ convergence criterion will be relaxed somewhat. Note that [146] states that the VOV is still a good convergence metric even if four moments do not exist. Nevertheless, this point should be kept in mind.

2.6.8.4 Analytic Study of $f(x)$ for Two-state Monte Carlo Problems

Booth [151, 152] examined the distribution of history scores analytically for both an analog two-state splitting problem and two exponential transform problems. This work provided the theoretical foundation for statistical studies [153] on relevant analytic functions to increase understanding of confidence interval coverage rates for Monte Carlo calculations.

It was found that the two-state splitting problem $f(x)$ decreases geometrically as the score increases by a constant increment. This is equivalent to a negative exponential behavior for a continuous $f(x)$. The $f(x)$ for the exponential transform problem decreases geometrically with geometrically increasing x. Therefore, the splitting problem produces a linearly decreasing $f(x)$ for the history score on a lin-log plot of the score probability versus score. The exponential transform problem generates a linearly decreasing score behavior (with high score negative exponential roll off) on a log-log plot of the score probability versus score plot. In general, the exponential transform problem is the more difficult to sample because of the larger impact of the low-probability high scores.

The analytic shapes were compared with a comparable problem calculated with a modified version of the MCNP code. These shapes of the analytic and empirical $f(x)$s were in excellent agreement [153].

2.6.8.5 Proposed Uses for the Empirical $f(x)$ in Each TFC Bin

Few papers discuss the underlying or empirical $f(x)$ for Monte Carlo transport problems [138, 146, 154]. The MCNP code provides a visual inspection and analysis of the empirical $f(x)$ for the TFC bin of each tally. This analysis helps to determine if there are any unsampled regions (holes) or spikes in the empirical history score PDF $f(x)$ at the largest history scores.

The most important use for the empirical $f(x)$ is to help determine if N has approached infinity in the sense of the CLT so that valid confidence intervals can be formed. It is assumed that the underlying $f(x)$ satisfies the CLT requirements; therefore, so should the empirical $f(x)$. Unless there is a largest possible history score, the empirical $f(x)$ must eventually decrease more steeply than x^{-3} for the second moment $\left(\int_{-\infty}^{\infty} x^2 f(x)\mathrm{d}x\right)$ to exist. It is postulated [155] that if such decreasing behavior in the empirical $f(x)$ with no upper bound has not been observed, then N is not large enough to satisfy the CLT because $f(x)$ has not been completely sampled. Therefore, a larger N is required before a confidence interval can be formed. It is important to note that this convergence criterion is NOT affected by any correlations that may exist between the estimated mean and the estimated R [146]. In principle, this lack of correlation should make the $f(x)$ diagnostic robust in assessing "complete" sampling.

Both the analytic and empirical history score distributions suggest that large score fill-in and one or more extrapolation schemes for the high score tail of the $f(x)$ could provide an estimate of scores not yet sampled to help assess the impact of the unsampled tail on the mean. The magnitude of the unsampled tail will surely affect the quality of the tally confidence interval.

2.6.8.6 Creation of $f(x)$ for TFC Bins

The creation of the empirical $f(x)$ in the MCNP code automatically covers nearly all TFC bin tallies that a user might reasonably be expected to make, including the effect of large and small tally multipliers. A logarithmically spaced grid is used for accumulating the empirical $f(x)$ because the tail behavior is assumed to be of the form $1/x^n$, $n > 3$ (unless an upper bound for the history scores exists). This grid produces an equal width histogram straight line for $f(x)$ on a log-log plot that decreases n decades in $f(x)$ per decade increase in x.

Ten bins per x decade are used and cover the unnormalized tally range from 10^{-30} to 10^{30}. The term "unnormalized" indicates that normalizations that are not performed until the end of the problem, such as cell volume or surface area, are not included in $f(x)$. The user can multiply this range at the start of the problem by the 16th entry on the DBCN card when the range is not sufficient. Both history score number and history score for the TFC bin are tallied in the x grid.

With this x grid in place, the average empirical $f(\overline{x}_i)$ between x_i and x_{i+1} is defined to be

$$f(\overline{x}_i) = \frac{\text{number of history scores in } i\text{th score bin}}{N(x_{i+1} - x_i)}, \tag{2.237}$$

where $x_{i+1} = 1.2589x_i$. The quantity 1.2589 is $10^{0.1}$ and comes from 10 equally spaced log bins per decade. The calculated $f(x_i)$s are available on printed plots or by using the "z" plot option (MCPLOT) with the TFC command mnemonics. Any history scores that are outside the x grid are counted as either above or below to provide this information to the user.

Negative history scores can occur for some electron charge deposition tallies. The default MCNP behavior is that any negative history score will be lumped into one bin below the lowest history score in the built-in grid (the default is 10^{-30}). If the 16th entry on the DBCN card is negative, $f(-x)$ will be created from the negative scores and the absolute value of the 16th entry on the DBCN card will be used as the score grid multiplier. Positive history scores then will be lumped into the lowest bin because of the sign change.

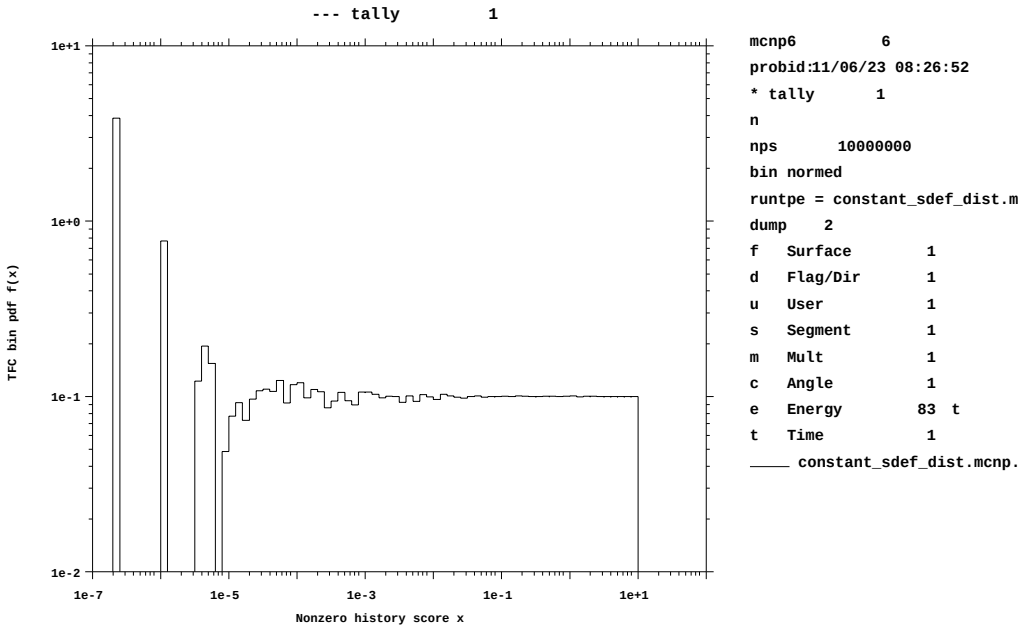


Figure 2.20: Example empirical history-score PDF for a uniform 0–10 MeV source.

Figure 2.20 and Fig. 2.21 show two simple examples of empirical $f(x)$s from the MCNP code for 10 million histories each. Figure 2.20 is from an energy leakage tally directly from a source that is uniform in energy from 0 to 10 MeV. The analytic $f(x)$ is a constant 0.1 between 0 and 10 MeV. The empirical $f(x)$ shows the sampling, which is 0.1 with statistical noise at the lower x (i.e., E, because the energy of the particle is the score) bins where fewer samples are made in the smaller-width energy bins. The MCNP input file for Fig. 2.20 is given in Listing 2.3 and plotting with the command input file given in Listing 2.4.

Listing 2.3: constant_sdef_dist.mcnp.inp.txt

```
Constant source energy distribution example
10 0 -1 imp:n=1
20 0  1 imp:n=0

1 so 10

sdef pos= 0 0 0 erg=d1
si1 0 10                $ Sample between 0 and 10.
sp1 0 1                 $ Sample uniformly.
*f1:n 1
e0 1e-7 80ilog 10       $ Bin tally by energy.
print                   $ Print all output file tables.
prdmp 2j 1              $ Write MCTAL file at conclusion of calculation.
nps 1e7
```

Listing 2.4: constant_sdef_dist.mcnp.comin.txt

```
tally 1
```

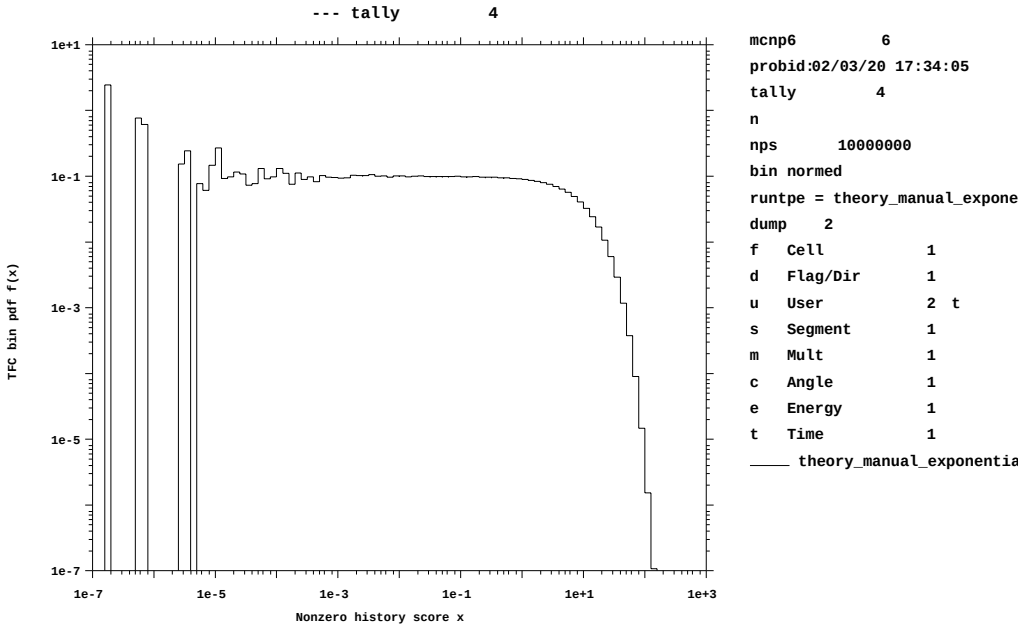


Figure 2.21: Example empirical history-score PDF for the first collision flux.

```
tfc p
end
```

Figure 2.21 shows the sampled distance to first collision in a material that has a macroscopic cross section of about 0.1 cm^{-1}. This analytic function is a negative exponential given by $f(x) = \Sigma \exp(-\Sigma x)$ (see §2.4.2) with a mean of 10. The empirical $f(x)$ transitions from a constant 0.1 at values of x less than unity to the expected negative exponential behavior for larger values of x. The MCNP input file for Fig. 2.21 is given in Listing 2.5 and plotting with the command input file given in Listing 2.6.

Listing 2.5: exponential_track_len_dist.mcnp.inp.txt

```
Exponential distance-to-collision distribution example
10 1 0.005 -1 imp:n=1
20 0        1 imp:n=0

1 so 1e6

sdef pos= 0 0 0 erg=355e-5
m1 1001 1
f4:n 10
ft4 inc                 $ Assign user binning by collision.
fu4 0                   $ Consider only uncollided values.
print                   $ Print all output file tables.
prdmp 2j 1              $ Write MCTAL file at conclusion of calculation.
cut:n j 350e-5
nps 1e7
```

Listing 2.6: exponential_track_len_dist.mcnp.comin.txt

```
tally 4
tfc p
loglog
xlims 1e-7 1e3
noerrbar
end
```

2.6.8.7 Pareto Fit to the Largest History Scores for the TFC Bin

The slope n in $1/x^n$ of the largest history tallies x must be estimated to determine when the largest history scores decrease faster than $1/x^3$. The 201 largest history scores for each TFC bin are continuously updated and saved during the calculation. A generalized Pareto function [156],

$$\text{Pareto } f(x) = a^{-1}\left(1 + \frac{kx}{a}\right)^{-(1/k)-1}, \tag{2.238}$$

is used to fit the largest xs. This function fits a number of extreme value distributions including $1/x^n$, exponential ($k = 0$), and constant ($k = -1$). The large history score tail fitting technique uses the robust "simplex" algorithm [157], which finds the values of a and k that best fit the largest history scores by maximum likelihood estimation.

The number of history score tail points used for the Pareto fit is a maximum of 201 points because this provides about 10% precision [156] in the slope estimator at $n = 3$. The precision increases for smaller values of n and vice versa. The number of points actually used in the fit is the lesser of 5% of the nonzero history scores or 201. The minimum number of points used for a Pareto fit is 25 with at least two different values,

which requires 500 nonzero history scores with the 5% criterion. If less than 500 history scores are made in the TFC bin, no Pareto fit is made.

From the Pareto fit, the slope of $f(x_{\text{large}})$ is defined to be

$$\text{SLOPE} \equiv (1/k) + 1. \tag{2.239}$$

A slope value of zero is defined to indicate that not enough $f(x_{\text{large}})$ tail information exists for a SLOPE estimate. The SLOPE is not allowed to exceed a value of 10 (a "perfect score"), which would indicate an essentially negative exponential decrease. If the 100 largest history scores all have values with a spread of less than 1%, an upper limit is assumed to have been reached and the SLOPE is set to 10. The SLOPE should be greater than 3 to satisfy the second moment existence requirement of the CLT. Then, $f(x)$ will appear to be "completely" sampled and hence N will appear to have approached infinity.

A printed plot of $f(x)$ is automatically generated in the MCNP output file if the SLOPE is less than 3 (or if any of the other statistical checks described in the next section do not pass). If $0 < \text{SLOPE} < 10$, several Ss appear on the printed plot to indicate the Pareto fit, allowing the quality of the fit to the largest history scores to be assessed visually. If the largest scores are not Pareto in shape, the SLOPE value may not reflect the best estimate of the largest history score decrease. A new SLOPE can be estimated graphically, as described in §2.6.8.8. A blank or 162 on the PRINT card also will cause printed plots of the first two cumulative moments of the empirical $f(x)$ to be made. Graphical plots of various $f(x)$ quantities can be made using the "z" plot option (MCPLOT) with the TFC plot command. These plots should be examined for unusual behavior in the empirical $f(x)$, including holes or spikes in the tail. The MCNP code tries to assess both conditions and prints a message if either condition is found.

2.6.8.8 Graphical Estimation of the Tally Slope when the Slope Test Fails

When the SLOPE test fails (SLOPE is less than or equal to three), the calculation should not be rejected without further analysis. Sometimes the SLOPE test fails because, although the MCNP code uses a Pareto distribution to fit the tally tail, the tally tail may not be well represented by a Pareto distribution. In this case, the user can manually assess the slope using a ruler and MCNP PRINT Table 161.

The slope estimator in the MCNP code is designed to estimate the number of score moments that exist in a calculation. Note that if for large x the score density $f(x)$ doesn't go down at least as fast as Cx^{-s} for some $x > tail$ then the rth score moment,

$$\int_{\text{tail}} f(x) r^x \mathrm{d}x > \int_{\text{tail}} Cx^{-s} \mathrm{d}x = \int_{\text{tail}} Cx^{r-s} \mathrm{d}s, \tag{2.240}$$

is not finite unless $r - s < -1$. That is, $s > r + 1$.

Thus for the second moment ($r = 2$) to exist $s > 3$ (needed to use the Central Limit Theorem) and for the fourth moment ($r = 4$) to exist $s > 5$ (desirable so that the VOV is finite, so that the sample variance is a good estimate of the true variance in the Central Limit Theorem.) If the tail score density were $f(x) < Cx^{-s}$, then

$$-\frac{\mathrm{d}(\log f(x))}{\mathrm{d}(\log x)} > s. \tag{2.241}$$

This derivative measures the number of decades change in $f(x)$ per decade change in x.

The Pareto fit to the score probability density is

$$f(x) = a^{-1}\left(1 + k\frac{x}{a}\right)^{-\frac{1}{k+1}}. \tag{2.242}$$

For large enough x, this becomes (essentially the Cx^{-s} mentioned earlier):

$$f(x) = a^{-1}\left(k\frac{x}{a}\right)^{-\frac{1}{k+1}}, \tag{2.243a}$$

$$\log f(x) = \log\left[a^{-1}\left(k\frac{x}{a}\right)^{-\frac{1}{k+1}}\right] = -\log a - \frac{1}{k+1}\left(\log\frac{k}{a} + \log x\right), \tag{2.243b}$$

$$\frac{\mathrm{d}(\log f(x))}{\mathrm{d}(\log x)} = -\frac{1}{k+1} = -(\text{MCNP slope}). \tag{2.243c}$$

Thus the MCNP slope estimator is a measure of the number of decades decline in $f(x)$ per decade decline in x.

MCNP PRINT Table 161 (see Listing 2.7) is a log-log plot, so the user can check whether the estimate of the tail slope looks reasonable. Suppose that the MCNP code tells the user:

```
the estimated inverse power slope of the 198 largest tallies starting at 2.99875E+00 is 1.4253
```

so the MCNP estimate of the tail in this case is from the last three bins in the chart. Note that PRINT Table 161 has the number on each bin. Note the vertical lines on PRINT Table 161 labeled with a "d". Each vertical line is an additional decade in $f(x)$.

Note that taking a ruler and drawing an extrapolation line through the `s s s` on the chart from $x = 0.501$ to $x = 5.01$ gives about 1.5 decades in $f(x)$. This graphically derived line through the `s s s` thus has 1.5 decades in $f(x)$ per decade in x (i.e. slope = 1.5); this is roughly consistent with the MCNP code's slope estimate of 1.4.

When a straight line is passed through the tail (***), the extrapolated line from $x = 0.501$ to $x = 5.01$ is off the chart at $x = 0.501$. Instead of using a full decade to get the "ruler" slope estimate, use the 0.5 decade from $x = 1.58$ to $x = 5.01$. That is, extrapolate a straight line through the tail and look at the slope of this line. The line changes well over 3 decades (perhaps 3.3 decades) in $f(x)$ in a 0.5 decade in x indicating that the slope is at least 6. Thus the user can conclude that the Pareto fit was not a good fit to $f(x)$ and the user can be fairly confident that at least 5 moments of the score distribution exist. It appears this calculation can thus be accepted despite the slope estimate warning.

Listing 2.7: Sample MCNP Print Table 161

```
abscissa  ordinate  log plot of tally probability density function in tally fluctuation chart bin(d=decade,slope= 1.4)
 tally number num den log den:d-----------------------d-----------------------d-----------------------d--------------------
2.51-01      5 6.45-07 -6.190 ***************          |                       |                       |
3.16-01    338 3.46-05 -4.460 ************************|************************|********               |
3.98-01    388 3.16-05 -4.500 ************************|************************|*******                |
5.01-01   1345 8.70-05 -4.061 ************************|************************|******************     |
6.31-01    629 3.23-05 -4.491 ************************|************************|*******                |
7.94-01   1881 7.68-05 -4.115 ************************|************************|******************     |
1.00+00  32836 1.06-03 -2.973 ************************|************************|************************|**********************
1.26+00   9820 2.53-04 -3.597 mmmmmmmmmmmmmmmmmmmmmmmm|mmmmmmmmmmmmmmmmmmmmmmmm|mmmmmmmmmmmmmmmmmmmmmmmm|mmmm
1.58+00    719 1.47-05 -4.833 ************************|************************ |                       |
2.00+00   4453 7.23-05 -4.141 ************************|************************|****************       |
2.51+00   1412 1.82-05 -4.739 ************************|************************|                       |
3.16+00    789 8.09-06 -5.092 ************************|******************       |s                      |
3.98+00     88 7.17-07 -6.145 ***************          |                      s  |                       |
5.01+00     30 1.94-07 -6.712 *                      |                    s     |                       |
 total   54733 3.65-04       d-----------------------d-----------------------d-----------------------d--------------------
```

2.6.9 Forming Statistically Valid Confidence Intervals

The goal of a Monte Carlo calculation is to produce a valid confidence interval for each tally bin. Section 2.6 has described different statistical quantities and the recommended criteria to form a valid confidence interval. Detailed descriptions of the information available in the output for all tally bins and the TFC bins are now discussed.

2.6.9.1 Information Available for Forming Statistically Valid Confidence

The R is calculated for every user-specified tally bin in the problem. The VOV and the shifted confidence interval center, discussed below, can be obtained for all bins with a nonzero entry for the 15th entry on the DBCN card at problem initiation.

2.6.9.1.1 R Magnitude Comparisons with MCNP Guidelines

The quality of MCNP tallies historically has been associated with two statistical checks that have been the responsibility of the user: (1) for all tally bins, the estimated relative error magnitude rules-of-thumb that are shown in Fig. 2.5 (that is, $R < 0.1$ for non-point detector tallies and $R < 0.05$ for point detector tallies); and (2) a statistically constant FOM in the user-selectable (TFn card) TFC bin so that the estimated R is decreasing by $1/\sqrt{N}$ as required by the CLT.

In an attempt to make the user more aware of the seriousness of checking these criteria, the MCNP code provides checks of the R magnitude for all tally bins. A summary of the checks is printed in the "Status of Statistical Checks" table. Messages are provided to the user giving the results of these checks.

2.6.9.1.2 Asymmetric Confidence Intervals

A correlation exists between the estimated mean and the estimated uncertainty in the mean [150]. If the estimated mean is below the expected value, the estimated uncertainty in the mean $S_{\overline{x}}$ will most likely be below its expected value. This correlation is also true for higher moment quantities such as the VOV. The worst situation for forming valid confidence intervals is when the estimated mean is much smaller than the expected value, resulting in smaller than predicted coverage rates. To correct for this correlation and improve coverage rates, one can estimate a statistic shift in the midpoint of the confidence interval to a higher value. The estimated mean is unchanged.

The shifted confidence interval midpoint is the estimated mean plus a term proportional to the third central moment. The term arises from an Edgeworth expansion [150] to attempt to correct the confidence interval for non-normality effects in the estimate of the mean. The adjustment term is given by

$$SHIFT = \frac{\sum(x_i - \overline{x})^3}{2S^2N}. \tag{2.244}$$

Substituting for the estimated mean and expanding produces

$$SHIFT = \frac{\left(\sum x_i^3 - \frac{3}{N}\sum x_i^2 \sum x_i + \frac{2}{N^2}\left(\sum x_i\right)^3\right)}{2\left(N\sum x_i^2 - \left(\sum x_i\right)^2\right)}. \tag{2.245}$$

The $SHIFT$ should decrease as $1/N$. This term is added to the estimated mean to produce the midpoint of the now asymmetric confidence interval about the mean. This value of the confidence interval midpoint can be used to form the confidence interval about the estimated mean to improve coverage rates of the true, but unknown, mean $E(x)$. The estimated mean plus the $SHIFT$ is printed automatically for the TFC bin for all tallies. A nonzero entry for the 15th DBCN card entry produces the shifted value for all tally bins.

This correction approaches zero as N approaches infinity, which is the condition required for the CLT to be valid. Kalos and Whitlock [158] uses a slightly modified form of this correction to determine if the requirements of the CLT are "substantially satisfied." Their relation is

$$\left|\sum(x_i - \overline{x})^3\right| \ll S^3\sqrt{N}, \tag{2.246}$$

which is equivalent to

$$SHIFT \ll S_{\overline{x}}/2. \tag{2.247}$$

The user is responsible for applying this check.

2.6.9.1.3 Forming Valid Confidence Intervals for Non-TFC Bins

The amount of statistical information available for non-TFC bins is limited to the mean and R. The VOV and the center of the asymmetric confidence can be obtained for all tally bins with a nonzero 15th entry on the DBCN card in the initial problem. The magnitude criteria for R (and the VOV, if available) should be met before forming a confidence interval. If the shifted confidence interval center is available, it should be used to form asymmetric confidence intervals about the estimated mean.

History dependent information about R (and the VOV, if available) for non-TFC bins can be obtained by printing out the tallies periodically during a calculation using the first entry on the PRDMP card. The N-dependent behavior of R can then be assessed. The complete statistical information available can be obtained by creating a new tally and selecting the desired TFC bin with the TFn card.

2.6.9.2 Information Available for Forming Statistically Valid Confidence Intervals for TFC Bins

Additional information about the statistical behavior of each TFC bin result is available. A TFC bin table is produced by the MCNP code after each tally to provide the user with detailed information about the apparent quality of the TFC bin result. The contents of the table are discussed in the following subsections, along with recommendations for forming valid confidence intervals using this information.

2.6.9.2.1 TFC Bin Tally Information

The first part of the TFC bin table contains information about the TFC bin result including the mean, R, scoring efficiency, the zero and nonzero history score components of R [§2.6.6], and the shifted confidence interval center. The two components of R can be used to improve the problem efficiency by either improving the history scoring efficiency or reducing the range of nonzero history scores.

2.6.9.2.2 The Largest TFC Bin History Score Occurs on the Next History

There are occasions when the user needs to make a conservative estimate of a tally result. Conservative is defined so that the results will not be less than the expected result. One reasonable way to make such an estimate is to assume that the largest observed history score would occur again on the very next history, $N + 1$.

The MCNP code calculates new estimated values for the mean, R, VOV, FOM, and shifted confidence interval center for the TFC bin result for this assumption. The results of this proposed occurrence are summarized in the TFC bin information table. The user can assess the impact of this hypothetical happening and act accordingly.

Table 2.9: Summary of MCNP Tally 10 Statistical Checks

Value	Test #	Description
Mean	1	a non-monotonic behavior (no up or down trend) in the estimated mean as a function of the number histories N for the last half of the problem
R	2	an acceptable magnitude of the estimated R of the estimated mean (< 0.05 for a point detector tally or < 0.10 for a non-point detector tally)
	3	a monotonically decreasing R as a function of the number histories N for the last half of the problem
	4	a $1/N$ decrease in the R as a function of N for the last half of the problem
VOV	5	the magnitude of the estimated VOV should be less than 0.10 for all types of tallies
	6	a monotonically decreasing VOV as a function of N for the last half of the problem
	7	a $1/N$ decrease in the VOV as a function of N for the last half of the problem
FOM	8	a statistically constant value of the FOM as a function of N for the last half of the problem
	9	a non-monotonic behavior in the FOM as a function of N for the last half of the problem
$f(x)$	10	the $SLOPE$ [Eq. (2.239)] of the 25 to 201 largest positive (negative with a negative DBCN(16) entry) history scores x should be greater than 3.0 so that the second moment $\int x\, f(x)\mathrm{d}x$ will exist if the $SLOPE$ is extrapolated to infinity

2.6.9.2.3 Description of the 10 Statistical Checks for the TFC Bin

The MCNP code prints the results of ten statistical checks of the tally in the TFC bin at each print. In a "Status of Statistical Checks" table, the results of these ten checks are summarized at the end of the output for all TFC bin tallies. The quantities involved in these checks are the estimated mean, R, VOV, FOM, and the large history score behavior of $f(x)$. Passing all of the checks should provide additional assurance that any confidence intervals formed for a TFC bin result will cover the expected result the correct fraction of the time. At a minimum, the results of these checks provide the user with more information about the statistical behavior of the result in the TFC bin of each tally.

The 10 statistical checks are made on the TFCs printed at the end of the output for desirable statistical properties of Monte Carlo solutions as shown in Table 2.9.

The seven N-dependent checks for the TFC bin are for the last half of the problem. The last half of the problem should be well behaved in the sense of the CLT to form the most valid confidence intervals. "Monotonically decreasing" in checks 3 and 5 allows for some increases in both R and the VOV. Such increases in adjacent TFC entries are acceptable and usually do not, by themselves, cause poor confidence intervals. A TFC bin R that does not pass check 3, by definition in the MCNP code, does not pass check 4. Similarly, a TFC bin VOV that does not pass check 6, by definition, does not pass check 7.

A table is printed after each tally for the TFC bin result that summarizes the results and the pass or no-pass status of the checks. Both asymmetric and symmetric confidence intervals are printed for the one, two, and three σ levels when all of the statistical checks are passed. These intervals can be expected to be correct with improved probability over historical rules of thumb. This is NOT A GUARANTEE, however; there is always a possibility that some as-yet-unsampled portion of the problem would change the confidence interval if more

histories were calculated. A WARNING is printed if one or more of these ten statistical checks is not passed, and one page of printed plot information about $f(x)$ is produced for the user to examine.

An additional information-only check is made on the largest five $f(x)$ score grid bins to determine if there are bins that have no samples or if there is a spike in an $f(x)$ that does not appear to have an upper limit. The result of the check is included in the TFC summary table for the user to consider. This check is not a pass or no-pass test because a hole in the tail may be appropriate for a discrete $f(x)$ or an exceptional sample occurred with so little impact that none of the ten checks was affected. The empirical $f(x)$ should be examined to assess the likelihood of "complete" sampling.

2.6.9.2.4 Forming Valid TFC Bin Confidence Intervals

For TFC bin results, the highest probability of creating a valid confidence interval occurs when all of the statistical checks are passed. Not passing several of the checks is an indication that the confidence interval is less likely to be correct. A monotonic trend in the mean for the last half of the problem is a strong indicator that the confidence interval is likely to produce incorrect coverage rates. The magnitudes of R and the VOV should be less than the recommended values to increase the likelihood of a valid confidence interval. Small jumps in the R, VOV, and/or the FOM as a function of N are not threatening to the quality of a result. The slope of $f(x)$ is an especially strong indicator that N has not approached infinity in the sense of the CLT. If the slope appears too shallow (< 3), check the printed plot of $f(x)$ to see that the estimated Pareto fit is adequate. The use of the shifted confidence interval is recommended, although it will be a small effect for a well-converged problem.

The last half of the problem is determined from the TFC. The more information available about the last half of the problem, the better the N-dependent checks will be. Therefore, a problem that has run 40,000 histories will have 20 TFC N entries, which is more N entries than a 50,000 history problem with 13 entries. It is possible that a problem that passes all tests at 40,000 may not pass all the tests at 40,001. As is always the case, the user is responsible for deciding when a confidence interval is valid. These statistical diagnostics are designed to aid in making this decision.

2.6.10 A Statistically Pathological Output Example

A statistically pathological test problem [147] is discussed in this section. The problem calculates the surface-leakage flux for neutrons above 12 MeV from an isotropic steady-state 14-MeV neutron point source of one particle/second at the center of a 30-cm-thick concrete shell with an outer radius of 390 cm. The input is shown in Listing 10.48 in §10.6.1. Point and ring detectors are deliberately used to estimate the surface neutron leakage flux with highly inefficient, long-tailed, $f(x)$s. The largest point detector history scores are those that have many collisions near the detector, which rarely occurs. A more-efficient volumetric track-length leakage-flux tally in a thin shell at the outer surface of the concrete sphere is also used to compare with the detector results. The variance-reduction methods used are implicit capture with weight cutoff, low-score point detector Russian roulette, and a 0.5-mean-free-path (approximately 8 cm for 12–14 MeV neutrons) neighborhood around the detectors to ensure finite tally higher moments.

Figure 2.22 shows MCNP plots of the estimated mean, R, VOV, and slope of the empirical history score PDF as a function of N for 10^4 (left column) and 10^8 (right column) histories. The track-length results are shown as a solid black line, ring-detector results as a dashed blue line, and the point-detector results as a dotted red line.

The left column shows the results as a function of N for 10^4 histories. The track-length flux tally appears well converged at 10^4 histories with a mean of 6.40×10^{-8} neutrons/cm^2/sec ($R = 0.029$, VOV $= 0.007$, and slope greater than 3). The point-detector result at 6,000 histories is 1.48×10^{-8} ($R = 0.023$), which is

a factor of 4 below the correct result. With no other information, this result could be accepted by even a careful Monte Carlo practitioner. However, the VOV never gets close to the recommended 0.1 value and the slope of $f(x)$ is 1.5. This slope could not continue indefinitely because the mean of $f(x)$ would not exist. Therefore, a confidence interval should not be formed for this tally. The ring detector is much better at sampling collisions close to the detector. Consequently, the ring detector tally results do not exhibit the point-detector small-mean behavior and are not yet converged ($R = 0.26$, VOV $= 0.56$, and SLOPE $= 1.7$).

The right column shows the results as a function of N for 10^8 histories. The Rs for this case should be 100 times smaller than the 10^4-history calculation for converged results. The 10^8-history track-length flux is 6.16×10^{-8} ($R = 0.0003$, VOV $= 0.0007$, and slope greater than 3), the ring-detector result of 6.27×10^{-8} ($R = 0.0033$, VOV $= 0.0005$, and SLOPE greater than 3), and the point-detector result is 6.15×10^{-8} ($R = 0.050$, VOV $= 0.085$, and slope is 2.3). Both the track-length and ring-detector tallies appear well converged, but the point-detector tally needs more sampling to more completely sample $f(x)$ and increase the slope. The ring-detector result differs from the track-length result slightly because of the uniform collision approximation in the neighborhood around to the detector.

The empirical $f(x)$s for the three tallies at 10^4 and 10^8 histories are shown in Figs. 2.23a and 2.23b. All 3 tally $f(x)$s have larger sampled history scores, but large scores are most prevalent for the point-detector tally. When one compares the empirical point detector $f(x)$s for 10^4 and 10^8 histories, the 10^4 history $f(x)$ has unsampled regions in the tail, which indicates incomplete $f(x)$ sampling [155]. For the point-detector tally, seven decades of x have been sampled with 10^8 histories compared to only three decades for 10^4 histories. The point-detector $f(x)$ slope is increasing, but it still does not yet appear to be completely sampled. The most compact (and most efficient) $f(x)$ is the track-length tally, followed by the ring-detector tally, and then the point-detector tallies. The track-length tally is 100 times more efficient than the ring-detector tally, which is 4,000 times more efficient than the point-detector tally, as measured using the FOM.

For difficult-to-sample problems such as this example, it is possible that an even larger history score could occur that would cause the VOV and possibly the slope to have unacceptable values. The mean and R will be much less affected than the VOV. The additional calculation time required to improve problem sampling and to reach acceptable values for the VOV and the slope could be prohibitive.

⚠ Caution

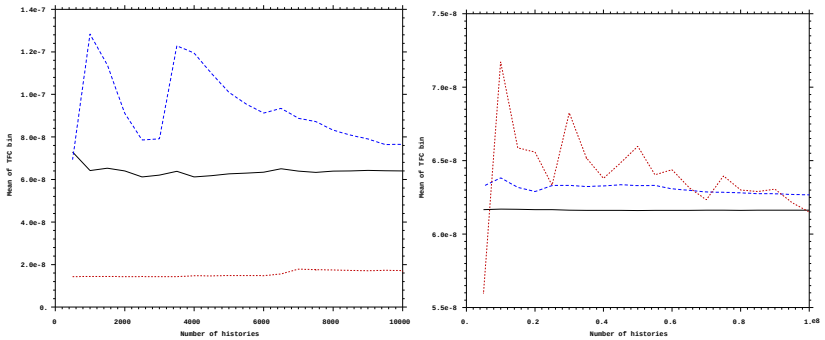
The large history score should NEVER be discarded from the tally result. It is important that the reason for the large history score be completely understood.

If the large history score is created by a poorly sampled region of phase space, the problem should be modified to provide improved phase-space sampling. If a conservative (large) answer is required, the printed result that assumes the largest history score occurs on the very next history can be used; however, there still may be yet unsampled but important regions of phase space that should be explored further.

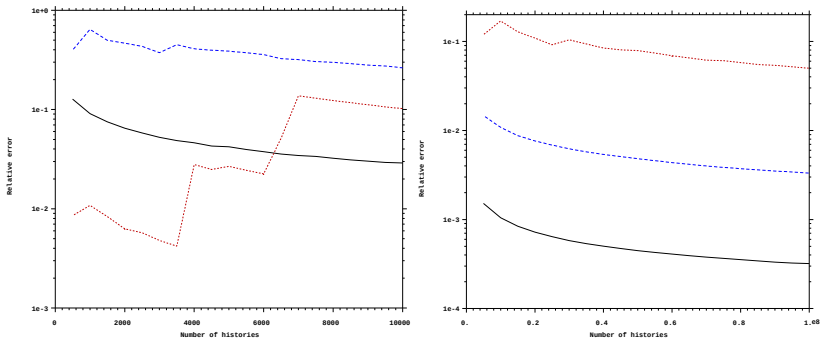
2.6.11 Batch Statistics

A small number of features in the MCNP code use batch statistics rather than history statistics for performance reasons. With batch statistics, the mean value of a number of histories is used as the score for computing the statistical moments. So long as this number of histories is constant, the total mean is the same as history statistics. Additionally, the standard error describes the same value, albeit with fewer degrees of freedom.

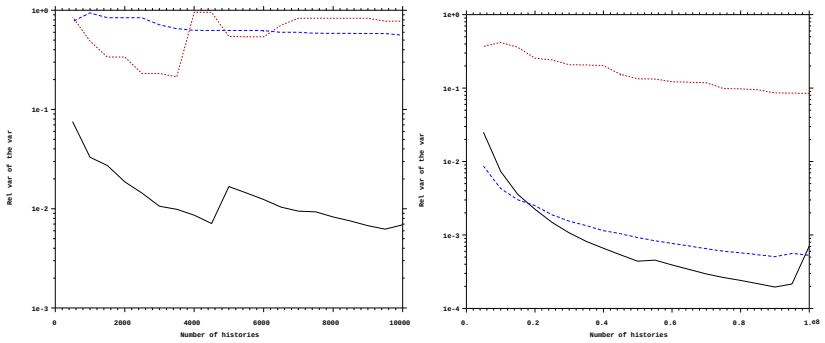
The reduction in degrees of freedom has a few effects. First, the standard error, as computed as the square root of the unbiased variance, is itself a biased estimator [159]. Using a small number of batches can result in a significant bias. However, as one goes beyond 100 batches this effect quickly becomes negligible. For a normal distribution, five samples results in an expected bias of 7.9%, and 100 samples results in an expected



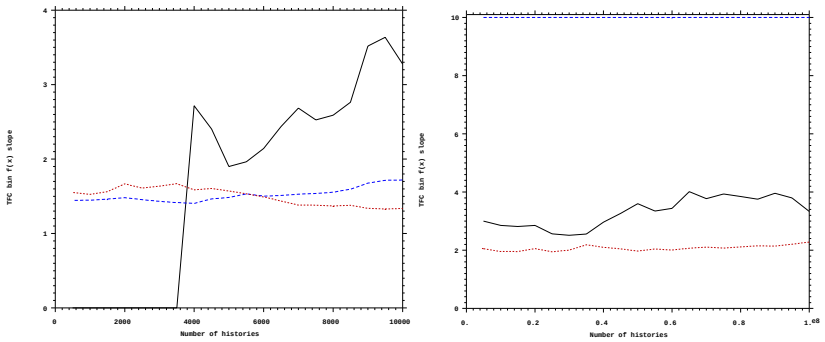
(a) Mean, 10^4 (b) Mean, 10^8



(c) Rel. error, 10^4 (d) Rel. error, 10^8

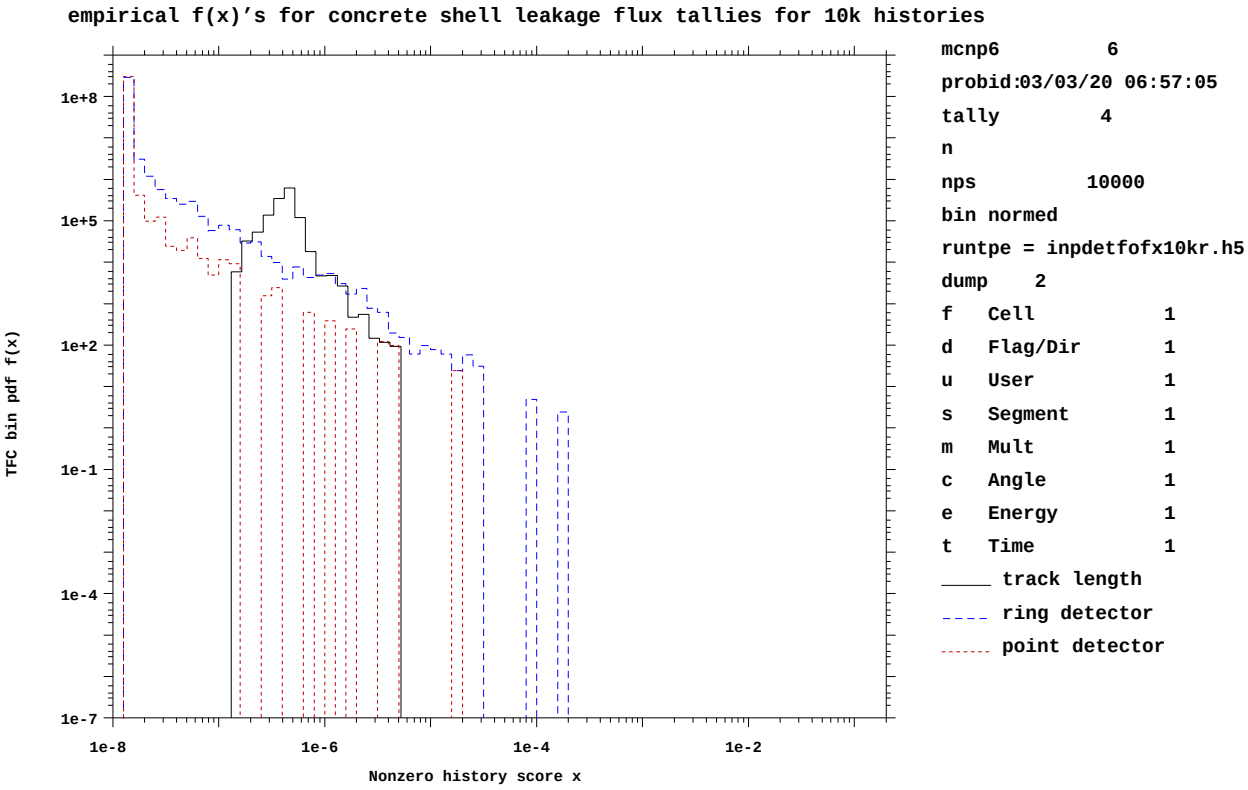


(e) VOV, 10^4 (f) VOV, 10^8

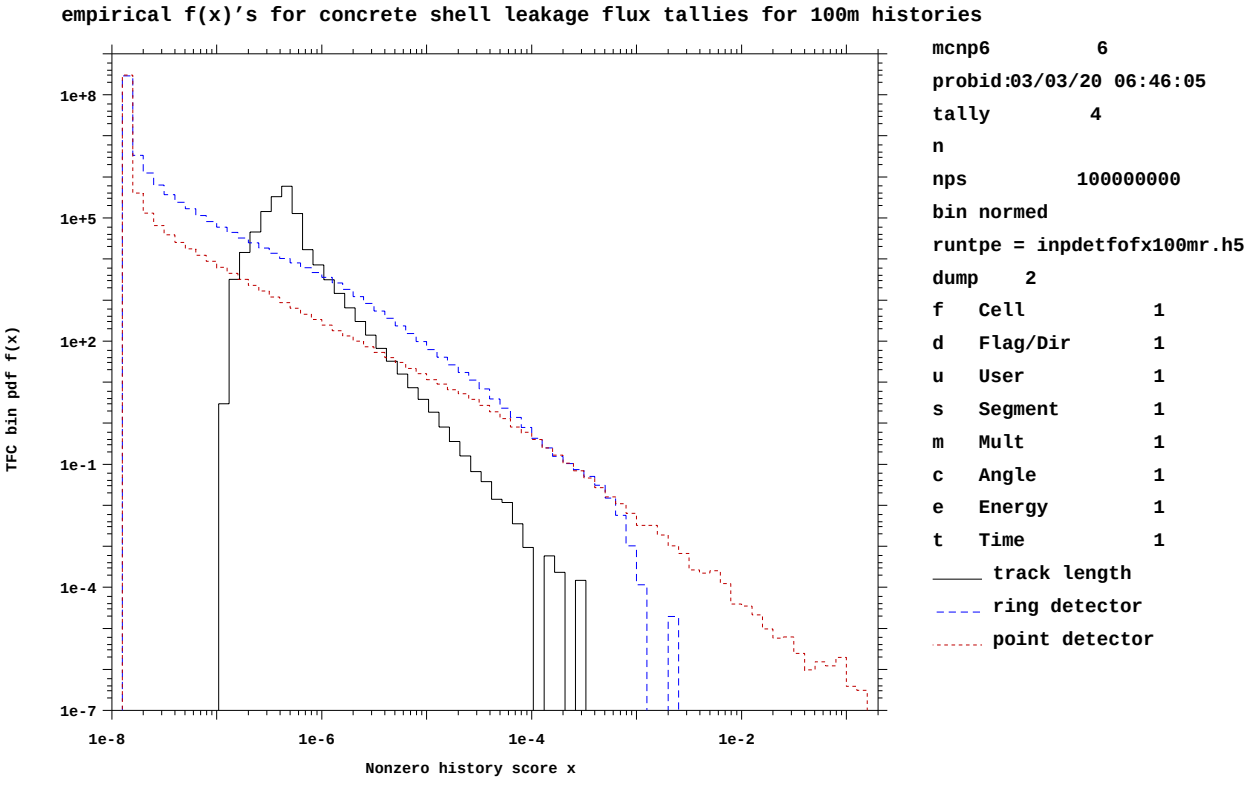


(g) Slope, 10^4 (h) Slope, 10^8

Figure 2.22: Mean, relative error, variance of the variance, and tally slope for 10,000 histories (left) and 100 million histories (right). The track length tally is the solid black line, ring detector is the dashed blue line and the point detector is the dotted red line.



(a) 10^4 histories



(b) 10^8 histories

Figure 2.23: The empirical $f(x)$s for 3 tallies with 10^4 and 10^8 histories.

bias of 0.25%. The second issue is that the variance of the standard error will generally increase when one partitions a fixed number of histories into fewer and fewer batches.

One advantage batch statistics has over history statistics is in KCODE problems. As discussed in §2.8.2.9, there is correlation between generations and between histories in an eigenvalue calculation. By summing together all histories in a generation, the correlation between histories within a single generation is eliminated. The correlation between generations, which is typically much larger, is not. This can yield slightly more conservative tally standard error estimates in KCODE.

Overall, with batch statistics, users should maximize both batch count (to avoid standard error bias and variance, with more than 100 batches recommended) and batch size (to ensure that batch statistics is faster than history statistics). Further, in KCODE problems, one needs to ensure the batch size is also sufficiently large to avoid renormalization bias on the mean as per usual.

2.7 Variance Reduction

2.7.1 General Considerations

2.7.1.1 Variance Reduction and Accuracy

Variance-reducing techniques in Monte Carlo calculations reduce the computer time required to obtain results of sufficient precision. Note that precision is only one requirement for a good Monte Carlo calculation. Even a zero variance calculation cannot accurately predict natural behavior if other sources of error are not minimized. Factors affecting accuracy were discussed in §2.6.

2.7.1.2 Two Choices that Affect Efficiency

The efficiency of a Monte Carlo calculation is affected by two choices: tally type and random walk sampling. The tally choice (for example, point detector flux tally vs. surface crossing flux tally) amounts to trying to obtain the best results from the random walks sampled. The chosen random walk sampling amounts to preferentially sampling "important" random walks at the expense of "unimportant" random walks. A random walk is important if it has a large affect on a tally. These two choices usually affect the time per history and the history variance as described next in §2.7.1.3. The MCNP code estimates tallies of the form

$$\langle T \rangle = \int \mathrm{d}\mathbf{r} \int \mathrm{d}\mathbf{v} \int \mathrm{d}t N(\mathbf{r},\mathbf{v},t) T(\mathbf{r},\mathbf{v},t) \tag{2.248}$$

by sampling particle histories that statistically produce the correct particle density $N(\mathbf{r},\mathbf{v},t)$. The tally function $T(\mathbf{r},\mathbf{v},t)$ is zero except where a tally is required. For example, for a surface crossing tally (F1), T will be one on the surface and zero elsewhere. MCNP variance reduction techniques allow the user to try to produce better statistical estimates of N where T is large, usually at the expense of poorer estimates where T is zero or small.

There are many ways to statistically produce $N(\mathbf{r},\mathbf{v},t)$. Analog Monte Carlo simply samples the events according to their natural physical probabilities. In this way, an analog Monte Carlo calculation estimates the number of physical particles executing any given random walk. Non-analog techniques do not directly simulate nature. Instead, non-analog techniques are free to do anything if N, hence $\langle T \rangle$, is preserved. This preservation is accomplished by adjusting the weight of the particles. The weight can be thought of as the number of physical particles represented by the MCNP particle [§2.4.1]. Every time a decision is made, the non-analog techniques require that the expected weight associated with each outcome be the same as in the

analog game. In this way, the expected number of physical particles executing any given random walk is the same as in the analog game.

For example, if an outcome "A" is made q times as likely as in the analog game, when a particle chooses outcome "A," its weight must be multiplied by q^{-1} to preserve the expected weight for outcome "A." Let p be the analog probability for outcome "A"; then pq is the non-analog probability for outcome "A." If w_0 is the current weight of the particle, then the expected weight for outcome "A" in the analog game is $w_0 \cdot p$ and the expected weight for outcome "A" in the non-analog game is $(w_0/q) \cdot pq$.

The MCNP code uses three basic types of non-analog games: (1) splitting, (2) Russian roulette, and (3) sampling from non-analog probability density functions. The previous paragraph discusses type 3. Splitting refers to dividing the particle's weight among two or more daughter particles and following the daughter particles independently. Usually the weight is simply divided evenly among k identical daughter particles whose characteristics are identical to the parent except for a factor $1/k$ in weight (for example, splitting in the weight window). In this case the expected weight is clearly conserved because the analog technique has one particle of weight w_0 at $(\mathbf{r}, \mathbf{v}, t)$, whereas the splitting results in k particles of weight w_0/k at $(\mathbf{r}, \mathbf{v}, t)$. In both cases the outcome is weight w_0 at $(\mathbf{r}, \mathbf{v}, t)$.

Other splitting techniques split the parent particle into k, typically two, differing daughter particles. The weight of the jth daughter represents the expected number of physical particles that would select outcome j from a set of k mutually exclusive outcomes. For example, the MCNP forced collision technique considers two outcomes: (1) the particle reaches a cell boundary before collision, or (2) the particle collides before reaching a cell boundary. The forced collision technique divides the parent particle representing w_0 physical particles into two daughter particles, representing w_1 physical particles that are uncollided and w_2 physical particles that collide. The uncollided particle of weight w_1 is then put on the cell boundary. The collision site of the collided particle of weight w_2 is selected from a conditional distance-to-collision probability density, the condition being that the particle must collide in the cell. This technique preserves the expected weight colliding at any point in the cell as well as the expected weight not colliding. A little simple mathematics is required to demonstrate this technique.

Russian roulette takes a particle at $(\mathbf{r}, \mathbf{v}, t)$ of weight w_0 and turns it into a particle of weight $w_1 > w_0$ with probability w_0/w_1 and kills it (that is, weight = 0) with probability $(1 - w_0/w_1)$. The expected weight at $(\mathbf{r}, \mathbf{v}, t)$ is

$$w_1 \cdot (w_0/w_1) + 0 \cdot (1 - w_0/w_1) = w_0, \tag{2.249}$$

the same as in the analog game.

Some techniques use a combination of these basic games and DXTRAN [§2.7.2.18] uses all three.

2.7.1.3 Efficiency, Time per History, and History Variance

Recall from §2.6.5 that the measure of efficiency for MCNP calculations is the FOM : $FOM \equiv 1/\left(R^2 T\right)$, where

R^2	is the sample relative standard deviation of the mean and
T	is the computer time for the calculation in minutes.

Recall from Eqs. (2.222) and (2.226a) that $R = \left(S/\sqrt{N}\right)/\overline{x}$, where

S^2	is the sample history variance,

N	is the number of particles, and
$\overline{x}$	is the sample mean.

Generally we are interested in obtaining the smallest R in a given time T. The equation above indicates that to decrease R it is desirable to: (1) decrease S and (2) increase N; that is, decrease the time per particle history. Unfortunately, these two goals usually conflict. Decreasing S normally requires more time because better information is required. Increasing N normally increases S because there is less time per history to obtain information. However, the situation is not hopeless. It is often possible either to decrease S substantially without decreasing N too much or to increase N substantially without increasing S too much, so that R decreases.

Many variance reduction techniques in the MCNP code attempt to decrease R by either producing or destroying particles. Some techniques do both. In general, techniques that produce tracks work by decreasing S (we hope much faster than N decreases) and techniques that destroy tracks work by increasing N (we hope much faster than S increases).

2.7.1.4 Strategy

Successful use of MCNP variance reduction techniques is often difficult, tending to be more art than science. The introduction of the weight window generator has improved things, but the user is still fundamentally responsible for the choice and proper use of variance reducing techniques. Each variance reduction technique has its own advantages, problems, and peculiarities. However, there are some general principles to keep in mind while developing a variance reduction strategy.

Not surprisingly, the general principles all have to do with understanding both the physical problem and the variance reduction techniques available to solve the problem. If an analog calculation will not suffice to calculate the tally, there must be something special about the particles that tally. The user should understand the special nature of those particles that tally. Perhaps, for example, only particles that scatter in particular directions can tally. After the user understands why the tallying particles are special, MCNP techniques can be selected (or developed by the user) that will increase the number of special particles followed.

After the MCNP techniques are selected the user typically has to supply appropriate parameters to the variance reduction techniques. This is probably more difficult than is the selection of techniques. The first guess at appropriate parameters typically comes either from experience with similar problems or from experience with an analog calculation of the current problem. It is usually better to err on the conservative side; that is, too little biasing rather than too much biasing. After the user has supplied parameters for the variance reduction techniques, a short Monte Carlo calculation is done so that the effectiveness of the techniques and parameters can be monitored with the MCNP output.

The MCNP output contains much information to help the user understand the sampling. This information should be examined to ensure that

1. the variance reduction techniques are improving the sampling of the particles that tally;
2. the variance reduction techniques are working cooperatively; that is, one is not destructively interfering with another;
3. the FOM table is not erratic, which would indicate poor sampling; and
4. there is nothing that looks obviously ridiculous.

Unfortunately, analyzing the output information requires considerable thought and experience. Reference [160] shows in detail strategies and analysis for a particular problem.

After ascertaining that the techniques are improving the calculation, the user makes a few more short calculations to refine the parameters until the sampling no longer improves. The weight window generator can also be turned on to supply information about the importance function in different regions of the phase space. This rather complex subject is described in §2.7.2.12.2.

2.7.1.5 Erratic Error Estimates

Erratic error estimates are sometimes observed in MCNP calculations. In fact, the primary reason for the Tally Fluctuation Chart (TFC) table in the MCNP output is to allow the user to monitor the FOM and the relative error as a function of the number of histories. With few exceptions, such as an analog point detector embedded in a scattering medium with $R_0 = 0$ (a practice highly discouraged), MCNP tallies are finite variance tallies. For finite variance tallies the relative error should decrease roughly as $\sqrt{N}$ so the FOM should be roughly constant and the ten statistical checks of the tallies [§2.6.9.2.3] should all be passed. If the statistical checks are not passed, the error estimates should be considered erratic and unreliable, no matter how small the relative error estimate is.

Erratic error estimates occur typically because a high-weight particle tallies from an important region of phase space that has not been well sampled. A high-weight particle in a given region of phase space is a particle whose weight is some nontrivial fraction of all the weight that has tallied from that region because of all previous histories. A good example is a particle that collides very close to a point or ring detector. If not much particle weight has previously collided that close to the detector, the relative error estimate will exhibit a jump for that history. Another example is coherent photon scattering towards a point detector [§2.4.4.2.5].

To avoid high-weight particles in important regions, the user should try to ensure that these regions are well sampled by many particles and try to minimize the weight fluctuation among these particles. Thus the user should try to use biasing techniques that preferentially push particles into important regions without introducing large weight fluctuations in these regions. The weight window can often be very useful in minimizing weight fluctuations caused by other variance reduction techniques.

If, despite a user's efforts, an erratic error estimate occurs, the user should obtain event logs for those particles causing the estimate to be erratic. The event logs should be studied to learn what is special about these particles. When the special nature of these particles is understood, the user can adjust the variance reduction techniques to sample these particles more often. Thus their weight will be smaller and they will not be as likely to cause erratic estimates.

⚠ Caution

Under no circumstances should these particles be discarded or ignored! The fact that these particles contribute very heavily to the tally indicates that they are important to the calculation and the user should try to sample more of them.

2.7.1.6 Biasing Against Random Walks of Presumed Low Importance

It was mentioned earlier that one should be cautious and conservative when applying variance reduction techniques. Many more people get into trouble by overbiasing than by underbiasing. Note that preferentially sampling some random walks means that some walks will be sampled (for a given computer time) less frequently than they would have been in an analog calculation. Sometimes these random walks are so heavily

biased against that very few, or even none, are ever sampled in an actual calculation because not enough particles are run.

Suppose that (on average) for every million histories only one track enters cell 23. Further suppose that a typical calculation is 100,000 histories. On any given calculation it is unlikely that a track enters cell 23. Now suppose that tracks entering cell 23 turn out to be much more important than a user thought. Maybe 10% of the answer should come from tracks entering cell 23. The user could run 100,000 particles and get 90% of the true tally with an estimated error of 1%, with no indication that anything is amiss. However, suppose the biasing had been set such that (on average) for every 10,000 particles, one track entered cell 23, about 10 tracks total. The tally probably will be severely affected by at least one high weight particle and will hover closer to the true tally with a larger and perhaps erratic error estimate. The essential point is this: following ten tracks into cell 23 does not cost much computer time and it helps ensure that the estimated error cannot be low when the tally is seriously in error. Always make sure that all regions of the problem are sampled enough to be certain that they are unimportant.

2.7.2 Variance-reduction Techniques

There are four classes of variance reduction techniques [16] that range from the trivial to the esoteric.

2.7.2.1 Truncation Methods

These are the simplest of variance reduction methods. They speed up calculations by truncating parts of phase space that do not contribute significantly to the solution. The simplest example is geometry truncation in which unimportant parts of the geometry are simply not modeled. Specific truncation methods available in the MCNP code are energy cutoff and time cutoff.

2.7.2.2 Population Control Methods

These use particle splitting and Russian roulette to control the number of samples taken in various regions of phase space. In important regions many samples of low weight are tracked, while in unimportant regions few samples of high weight are tracked. A weight adjustment is made to ensure that the problem solution remains unbiased. Specific population control methods available in the MCNP code are geometry splitting and Russian roulette, energy splitting/ roulette, time splitting/roulette, weight cutoff, and weight windows.

2.7.2.3 Modified Sampling Methods

These alter the statistical sampling of a problem to increase the number of tallies per particle. For any Monte Carlo event it is possible to sample from any arbitrary distribution rather than the physical probability as long as the particle weights are then adjusted to compensate. Thus with modified sampling methods, sampling is done from distributions that send particles in desired directions or into other desired regions of phase space such as time or energy, or change the location or type of collisions. Modified sampling methods in the MCNP code include the exponential transform, implicit capture, forced collisions, source biasing, and neutron-induced photon production biasing.

2.7.2.4 Partially Deterministic Methods

These are the most complicated class of variance reduction methods. They circumvent the normal random walk process by using deterministic-like techniques, such as next-event estimators, or by controlling the random number sequence. In the MCNP code these methods include point detectors, DXTRAN, and correlated sampling.

The available MCNP variance reduction techniques are described next.

2.7.2.5 Energy Cutoff

The energy cutoff in the MCNP code is either a single user-supplied, problem-wide energy level or a cell-dependent energy level. Particles are terminated when their energy falls below the energy cutoff. The energy cutoff terminates tracks and thus decreases the time per history. The energy cutoff should be used only when it is known that low-energy particles are either of zero or almost zero importance. An energy cutoff is like a Russian roulette game with zero survival probability. A number of pitfalls exist.

1. Remember that low-energy particles can often produce high-energy particles (for example, fission or low-energy neutrons inducing high-energy photons). Thus, even if a detector is not sensitive to low-energy particles, the low-energy particles may be important to the tally.
2. The CUT card energy cutoff is the same throughout the problem. Often low-energy particles have zero importance in some regions and high importance in others, and so a cell-dependent energy cutoff is also available with the ELPT card.
3. The answer will be biased (low) if the energy cutoff is killing particles that might otherwise have contributed. Furthermore, as $N \to \infty$ the apparent error will go to zero and therefore mislead the unwary. Serious consideration should be given to two techniques discussed later, energy roulette and space-energy weight window, that are always unbiased.

The energy cutoff has one advantage not directly related to variance reduction. A lower energy cutoff requires more cross sections so that computer memory requirements go up and interactive computing with a time-sharing system is degraded.

2.7.2.6 Time Cutoff

The time cutoff in the MCNP code, controlled with the MCNP CUT card, is a single user-supplied, problem-wide time value. Particles are terminated when their time exceeds the time cutoff. The time cutoff terminates tracks and thus decreases the computer time per history. A time cutoff is like a Russian roulette game with zero survival probability. The time cutoff should only be used in time-dependent problems where the last time bin will be earlier than the cutoff.

Although the energy and time cutoffs are similar, more caution must be exercised with the energy cutoff because low energy particles can produce high energy particles, whereas a late time particle cannot produce an early time particle.

2.7.2.7 Geometry Splitting with Russian Roulette

Geometry splitting/Russian roulette is one of the oldest and most widely used variance-reducing techniques in Monte Carlo codes. When used judiciously, it can save substantial computer time. As particles migrate in an important direction, they are increased in number to provide better sampling, but if they head in an unimportant direction, they are killed in an unbiased manner to avoid wasting time on them. Oversplitting, however, can substantially waste computer time. Splitting generally decreases the history variance but increases the time per history, whereas Russian roulette generally increases the history variance but decreases the time per history.

Each cell in the problem geometry setup is assigned an importance I by the user on the IMP input card. The number I should be proportional to the estimated value that particles in the cell have for the quantity being scored. When a particle of weight W passes from a cell of importance I to one of higher importance I', the particle is split into a number of identical particles of lower weight according to the following recipe. If I'/I is an integer n ($n \geq 2$), the particle is split into n identical particles, each weighing W/n. Weight is preserved in the integer splitting process. If I'/I is not an integer but still greater than 1, splitting is done probabilistically so that the expected number of splits is equal to the importance ratio. Denoting $n = \lfloor I'/I \rfloor$ to be the largest integer in I'/I, $p = I'/I - n$ is defined. Then with probability p, $n+1$ particles are used, and with probability $1-p$, n particles are used. For example, if I'/I is 2.75, 75% of the time split 3 for 1 and 25% of the time split 2 for 1. The weight assigned to each particle is $W \cdot I/I'$, which is the expected weight, to minimize dispersion of weights.

On the other hand, if a particle of weight W passes from a cell of importance I to one of lower importance I', so that $I'/I < 1$, Russian roulette is played and the particle is killed with probability $1-(I'/I)$, or followed further with probability I'/I and weight $W \cdot I/I'$.

Geometry splitting with Russian roulette is very reliable. It can be shown that the weights of all particle tracks are the same in a cell no matter which geometric path the tracks have taken to get to the cell, assuming that no other biasing techniques, e.g. implicit capture, are used. The variance of any tally is reduced when the possible contributors all have the same weight.

The assigned cell importances can have any value—they are not limited to integers. However, adjacent cells with greatly different importances place a greater burden on reliable sampling. Once a sample track population has deteriorated and lost some of its information, large splitting ratios (like 20 to 1) can build the population back up, but nothing can regain the lost information. It is generally better to keep the ratio of adjacent importances small (for example, a factor of a few) and have cells with optical thicknesses in the penetration direction less than about two mean free paths. The MCNP code prints a warning message if adjacent importances or weight windows have a ratio greater than 4. PRINT Table 120 in the output file lists the affected cells and ratios.

Generally, in a deep penetration shielding problem the sample size (number of particles) diminishes to almost nothing in an analog simulation, but splitting helps keep the size built up. A good rule is to keep the population of tracks traveling in the desired direction more or less constant—that is, approximately equal to the number of particles started from the source. A good initial approach is to split the particles 2 for 1 wherever the track population drops by a factor of 2. Near-optimum splitting usually can be achieved with only a few iterations and additional iterations show strongly diminishing returns. Note that in a combined neutron/photon problem, importances will probably have to be set individually for neutrons and for photons.

The MCNP code never splits into a void, although Russian roulette can be played entering a void. Splitting into a void accomplishes nothing except extra tracking because all the split particles must be tracked across the void and they all make it to the next surface. The split should be done according to the importance ratio of the last non-void cell departed and the first non-void cell entered. Note four more items:

1. Geometry splitting/Russian roulette works well only in problems that do not have extreme angular dependence. In the extreme case, splitting/Russian roulette can be useless if no particles ever enter an important cell where the particles can be split.
2. Geometry splitting/Russian roulette will preserve weight variations. The technique is "dumb" in that it never looks at the particle weight before deciding appropriate action. An example is geometry splitting/Russian roulette used with source biasing.
3. Geometry splitting/Russian roulette are turned on or off together.
4. Particles are killed immediately upon entering a zero importance cell, acting as a geometry cutoff.

2.7.2.8 Energy Splitting/Roulette

Energy splitting and roulette is controlled with the MCNP `ESPLT` card. Energy splitting/roulette is independent of spatial cell. If the problem has a space-energy dependence, the space-energy dependent weight window is normally a better choice.

2.7.2.8.1 Energy Splitting

In some cases, particles are more important in some energy ranges than in others. For example, it may be difficult to calculate the number of ^{235}U fissions because the thermal neutrons are also being captured and not enough thermal neutrons are available for a reliable sample. In this case, once a neutron falls below a certain energy level it can be split into several neutrons with an appropriate weight adjustment. A second example involves the effect of fluorescent emission after photoelectric absorption. With energy splitting, the low-energy photon track population can be built up rather than rapidly depleted, as would occur naturally with the high photoelectric absorption cross section.

2.7.2.9 Energy Roulette

In many cases the number of tracks increases with decreasing energy, especially neutrons near the thermal energy range. These tracks can have many collisions requiring appreciable computer time. They may be important to the problem and cannot be completely eliminated with an energy cutoff, but their number can be reduced by playing a Russian roulette game to reduce their number and computer time.

If a track's energy is below a prescribed energy level, the roulette game is played, based on the input value of the survival probability. If the game is won, the track's history is continued, but its weight is increased by the reciprocal of the survival probability to conserve weight.

2.7.2.10 Time Splitting/Roulette

Time splitting/roulette, controlled with the MCNP `TSPLT` card, is similar to the energy splitting and roulette game just discussed, except a particle's time can only increase, in contrast with a particle's energy that may increase or decrease. Time splitting/roulette is independent of spatial cell. If the problem has a space-time dependence, the space-time dependent weight window is normally a better choice.

1. Splitting: In some cases, particles are more important later in time. For example, if a detector responds primarily to late time particles, then it may be useful to split the particles as time increases.
2. Russian roulette: In some cases there may be too many late time particles for optimal calculational efficiency, and the late time particles can be rouletted.

2.7.2.11 Weight Cutoff

In weight cutoff, Russian roulette is played if a particle's weight drops below a user-specified weight cutoff. The particle is either killed or its weight is increased to a user-specified level. The weight cutoff was originally envisioned for use with geometry splitting/Russian roulette and implicit capture [§2.7.2.14]. Because of this intent,

1. The weight cutoffs in cell j depend not only on $w_{c,1}$ and $w_{c,2}$ on the CUT card, but also on the cell importances.
2. Implicit capture is always turned on (except in detailed photon physics) whenever a nonzero $w_{c,1}$ is specified.

Referring to item 1 above, the weight cutoff is applied when the particle's weight falls below $R_j \cdot w_{c,2}$, where R_j is the ratio of the source cell importance (IMP card) to cell j's importance. With probability $W/(w_{c,1} \cdot R_j)$ the particle survives with new weight $w_{c,1} \cdot R_j$; otherwise the particle is killed.

As mentioned earlier, the weight cutoff game was originally envisioned for use with geometry splitting and implicit capture. To illustrate the need for a weight cutoff when using implicit capture, consider what can happen without a weight cutoff. Suppose a particle is in the interior of a very large medium and there are neither time nor energy cutoffs. The particle will go from collision to collision, losing a fraction of its weight at each collision. Without a weight cutoff, a particle's weight would eventually be too small to be representable in the computer, at which time an error would occur. If there are other loss mechanisms (for example, escape, time cutoff, or energy cutoff), the particle's weight will not decrease indefinitely, but the particle may take an unduly long time to terminate.

Weight cutoff's dependence on the importance ratio can be easily understood if one remembers that the weight cutoff game was originally designed to solve the low-weight problem sometimes produced by implicit capture. In a high-importance region, the weights are low by design, so it makes no sense to play the same weight cutoff game in high- and low-importance regions.

Comments: Many techniques in the MCNP code cause weight change. The weight cutoff was really designed with geometry splitting and implicit capture in mind. Care should be taken in the use of other techniques.

Weight cutoff games are unlike time and energy cutoffs. In time and energy cutoffs, the random walk is always terminated when the threshold is crossed. Potential bias may result if the particle's importance was not zero. A weight cutoff (weight roulette would be a better name) does not bias the game because the weight is increased for those particles that survive.

Setting the weight cutoff is not typically an easy task, and it requires thought and experimentation. Essentially, the user must guess what weight is worth following and start experimenting with weight cutoffs in that vicinity.

2.7.2.12 Weight Window

The weight window, shown qualitatively in Fig. 2.24, is a phase space splitting and Russian roulette technique. The phase space may be space-energy, space-time, or space.

For each phase space cell, the user supplies a lower weight bound. The upper weight bound is a user-specified multiple of the lower weight bound. These weight bounds define a window of acceptable weights. If a particle is below the lower weight bound, Russian roulette is played and the particle's weight is either increased to a

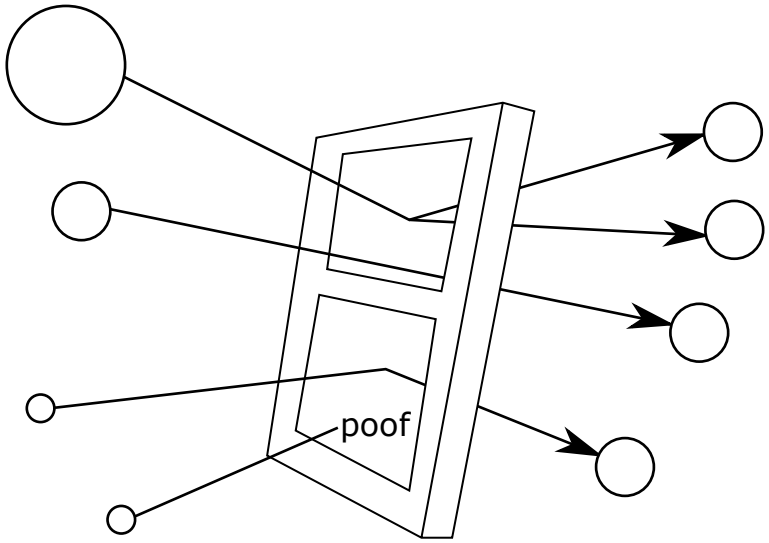
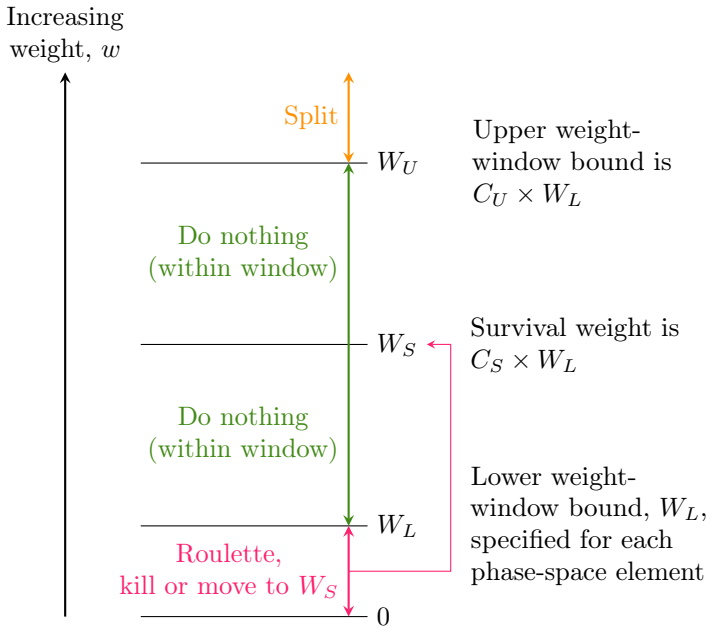


Figure 2.24: Qualitative illustration of weight window splitting and rouletting.



Note: Constants C_U and C_S apply throughout the entire problem.

Figure 2.25: Implementation diagram of MCNP weight window splitting and rouletting ranges.

value within the window or the particle is terminated. If a particle is above the upper weight bound, it is split so that all the split particles are within the window. No action is taken for particles within the window.

Figure 2.25 is a more detailed picture of the weight window. Three important weights define the weight window in a phase space cell:

1. W_L, the lower weight bound,
2. W_S, the survival weight for particles playing roulette, and
3. W_U, the upper weight bound.

The user specifies W_L for each phase space cell on WWN cards. W_S and W_U are calculated using two problem-wide constants, C_S and C_U (entries on the WWP card), as $W_S = C_S W_L$ and $W_U = C_U W_L$. Thus all cells have an upper weight bound C_U times the lower weight bound and a survival weight C_S times the lower weight bound.

Although the weight window can be effective when used alone, it was designed for use with other biasing techniques that introduce a large variation in particle weight. In particular, a particle may have several "unpreferred" samplings, each of which will cause the particle weight to be multiplied by a weight factor substantially larger than one. Any of these weight multiplications by itself is usually not serious, but the cumulative weight multiplications can seriously degrade calculational efficiency. Worse, the error estimates may be misleading until enough extremely high-weight particles have been sampled. Monte Carlo novices are prone to be misled because they do not have enough experience reading and interpreting the summary information on the sampling supplied by the MCNP code. Hence, a novice may put more faith in an answer than is justified.

Although it is impossible to eliminate all pathologies in Monte Carlo calculations, a properly specified weight window goes far toward eliminating the pathology referred to in the preceding paragraph. As soon as the weight gets above the weight window, the particle is split and subsequent weight multiplications will thus be multiplying only a fraction of the particle's weight (before splitting). Thus, it is hard for the tally to be severely perturbed by a particle of extremely large weight. In addition, low-weight particles are rouletted, so time is not wasted following particles of trivial weight.

One cannot ensure that every history contributes the same score (a zero variance solution), but by using a window inversely proportional to the importance, one can ensure that the mean score from any track in the problem is roughly constant. A weight window generator exists to estimate these importance reciprocals [§2.7.2.12.2]. In other words, the window is chosen so that the track weight times the mean score (for unit track weight) is approximately constant. Under these conditions, the variance is due mostly to the variation in the number of contributing tracks rather than the variation in track score.

Thus far, two things remain unspecified about the weight window: the constant of inverse proportionality and the width of the window. It has been observed empirically that an upper weight bound five times the lower weight bound works well, but the results are reasonably insensitive to this choice anyway. The constant of inverse proportionality is chosen so that the lower weight bound in some reference cell is chosen appropriately. In most instances the constant should be chosen so that the source particles start within the window.

2.7.2.12.1 Weight Window Compared to Geometry Splitting

Although both techniques use splitting and Russian roulette, there are some important differences.

1. The weight window is space-energy dependent or space-time dependent. Geometry splitting is only space dependent.
2. The weight window discriminates on particle weight before deciding appropriate action. Geometry splitting is done regardless of particle weight.
3. The weight window works with absolute weight bounds. Geometry splitting is done on the ratio of the importance across a surface.
4. The weight window can be applied at surface crossings, collisions, or both. In addition, a weight window can be applied after a given distance of travel in a material (e.g., using the *nmfp* entry on the WWP card). Geometry splitting is applied only at surface crossings.
5. The weight window can control weight fluctuations introduced by other biasing techniques by requiring all particles in a cell to have weight $W_L \leq W \leq W_U$ [161]. The geometry splitting will preserve any weight fluctuations because it is weight independent.
6. In the rare case where no other weight modification schemes are present, importances will cause all particles in a given cell to have the same weight. Weight windows will merely bound the weight.
7. The weight windows can be turned off for a given cell or energy regime by specifying a zero lower bound. This is useful in long or large regions where no single importance function applies. Care should be used because when the weight window is turned off at collisions, the weight cutoff game is turned on, sometimes causing too many particles to be killed.
8. For repeated structures, the geometry splitting uses the product of the importances at the different levels. No product is used for the weight windows.

2.7.2.12.2 The Stochastic Weight-window Generator

The generator, controlled with the MCNP WWG card, is a method that automatically generates weight window importance functions [161]. The values generated may be thought of as estimates of a forward-calculated adjoint solution and can provide considerable insight into the physics of a problem. The task of choosing importances by guessing, intuition, experience, or trial and error is simplified and insight into the Monte Carlo calculation is provided.

Low weight-window values indicate important regions. A low weight-window value near the boundary with the outside world often indicates that the geometry was truncated and more cells need to be added outside the present geometry. Weight-window values that differ greatly between adjacent cells indicate poor weight window convergence and/or a need to subdivide the geometry into smaller phase space units that will have different importances.

Although the window generator has proved very useful, two caveats are appropriate. The generator is not a panacea for all importance sampling problems and certainly is not a substitute for thinking on the user's part. In fact, in most instances, the user will have to decide when the generator's results look reasonable and when they do not. After these disclaimers, one might wonder what use to make of a generator that produces both good and bad results. To use the generator effectively, it is necessary to remember that the generated parameters are only statistical estimates and that these estimates can be subject to considerable error. Nonetheless, practical experience indicates that a user can learn to use the generator effectively to solve some very difficult transport problems.

Examples of the weight-window generator are given in [160, 161] and should be examined before using the generator. Note that this importance estimation scheme works regardless of what other variance reduction techniques are used in a calculation.

2.7.2.12.3 Theory

The importance of a particle at a point P in phase space equals the expected score a unit weight particle will generate. Imagine dividing the phase space into a number of phase space "cells" or regions. The importance of a cell then can be defined as the expected score generated by a unit weight particle after entering the cell. Thus, with a little bookkeeping, the cell's importance can be estimated as

$$\begin{matrix}\text{Importance}\\(\text{expected score})\end{matrix} = \frac{\begin{matrix}\text{total score because of particles (and}\\\text{their progeny) entering the cell}\end{matrix}}{\text{total weight entering the cell}}. \tag{2.250}$$

After the importances have been generated, the MCNP code assigns weight windows inversely proportional to the importances. Then the MCNP code supplies the weight windows in an output file suitable for use as an input file in a subsequent calculation. The spatial portion of the phase space is divided using either standard MCNP cells or a superimposed mesh grid, which can be either rectangular or cylindrical. The energy portion of the phase space is divided using the WWGE card. The time portion of the phase space can be divided also. The constant of proportionality is specified on the WWG card.

2.7.2.12.4 Limitations of the Weight-window Generator

The principal problem encountered when using the generator is bad estimates of the importance function because of the statistical nature of the generator. In particular, unless a phase space region is sampled adequately, there will be either no generator importance estimate or an unreliable one. The generator often needs a very crude importance guess just to get any tally; that is, the generator needs an initial importance function to estimate a (we hope) better one for subsequent calculations.

Fortunately, in most problems the user can guess some crude importance function sufficient to get enough tallies for the generator to estimate a new set of weight windows. Because the weight windows are statistical, several iterations usually are required before the optimum importance function is found for a given tally. The first set of generated weight windows should be used in a subsequent calculation, which generates a better set of windows, etc.

In addition to iterating on the generated weight windows, the user must exercise some degree of judgment. Specifically, in a typical generator calculation, some generated windows will look suspicious and will have to be reset. In the MCNP code, this task is simplified by an algorithm that automatically scrutinizes cell-based importance functions, either input by the user or generated by a generator. By flagging the generated windows that are more than a factor of 4 different from those in adjacent spatial regions, often it is easy to determine which generated weight windows are likely to be statistical flukes that should be revised before the next generator iteration. For example, suppose the lower weight bounds in adjacent cells were 0.5, 0.3, 0.9, 0.05, 0.03, 0.02, etc.; here the user would probably want to change the 0.9 to something like 0.1 to fit the pattern, reducing the 18:1 ratio between cells 3 and 4.

The weight window generator also will fail when phase space is not sufficiently subdivided and no single set of weight window bounds is representative of the whole region. It is necessary to turn off the weight windows (by setting a lower bound of zero) or to further subdivide the geometry or energy phase space. Use of a superimposed importance mesh grid for weight window generation is a good way to subdivide the spatial portion of the phase space without complicating the MCNP cell geometry.

On the other hand, the weight window generator will also fail if the phase space is too finely subdivided and subdivisions are not adequately sampled. Adequate sampling of the important regions of phase space is always key to accurate Monte Carlo calculations, and the weight window generator is a tool to help the user determine the important phase space regions. When using the mesh-based weight window generator, resist the temptation to create mesh cells that are too small.

2.7.2.13 Exponential Transform

The exponential transform, controlled with the MCNP EXT card, samples the distance to collision from a non-analog probability density function. Although many impressive results are claimed for the exponential transform, it should be remembered that these results are usually obtained for one-dimensional geometries and quite often for energy-independent problems. A review article by Clark [162] gives theoretical background and sample results for the exponential transform. Sarkar and Prasad [163] have done a purely analytical analysis for the optimum transform parameter for an infinite slab and one energy group. The exponential transform allows particle walks to move in a preferred direction by artificially reducing the macroscopic cross section in the preferred direction and increasing the cross section in the opposite direction according to

$$\Sigma_{\mathrm{t}}^{*} = \Sigma_{\mathrm{t}}(1 - p\mu), \tag{2.251}$$

where

Σ_{t}^{*}	is the fictitious transformed cross section,
Σ_{t}	is the true total cross section,
Σ_{a}	is the absorption cross section,
Σ_{s}	is the scattering cross section,
p	is the exponential transform parameter used to vary the degree of biasing $\lvert p\rvert < 1$ can be a constant or $p = \Sigma_{\mathrm{a}}/\Sigma_{\mathrm{t}}$, in which case $\Sigma_{\mathrm{t}}^{*} = \Sigma_{\mathrm{s}}$, and
μ	is the cosine of the angle between the preferred direction and the particle's direction with $\mu \leq 1$. The preferred direction can be specified on a VECT card.

At a collision a particle's weight is multiplied by a factor w_c (derived below) so that the expected weight colliding at any point is preserved. The particle's weight is adjusted such that the weight multiplied by the probability that the next collision is in ds about s remains constant.

The probability of colliding in ds about s is $\Sigma \exp(-\Sigma s)\mathrm{d}s$ where Σ is either Σ_{t} or Σ_{t}^{*}, so that preserving the expected collided weight requires

$$\Sigma_{\mathrm{t}} \exp(-\Sigma_{\mathrm{t}} s)\mathrm{d}s = w_c \Sigma_{\mathrm{t}}^{*} \exp(-\Sigma_{\mathrm{t}}^{*} s)\mathrm{d}s, \tag{2.252}$$

or

$$w_c = \frac{\Sigma_{\mathrm{t}} \exp(-\Sigma_{\mathrm{t}} s)}{\Sigma_{\mathrm{t}}^{*} \exp(-\Sigma_{\mathrm{t}}^{*} s)} = \frac{\exp(-\rho\Sigma_{\mathrm{t}}\mu s)}{1 - p\mu}. \tag{2.253}$$

If the particle reaches a cell surface, time cutoff, DXTRAN sphere, or tally segment instead of colliding, the particle's weight is adjusted so that the weight, multiplied by the probability that the particle travels a distance s to the surface, remains constant. The probability of traveling a distance s without collision is $\exp(-\Sigma s)$ so that preserving the expected uncollided weight requires

$$\exp(-\Sigma_{\mathrm{t}} s) = w_s \exp(-\Sigma_{\mathrm{t}}^{*} s), \tag{2.254}$$

or

$$w_s = \frac{\exp(-\Sigma_{\mathrm{t}} s)}{\exp(-\Sigma_{\mathrm{t}}^{*} s)} = \exp(-\rho\Sigma_{\mathrm{t}}\mu s). \tag{2.255}$$

For one-dimensional deep penetration through highly absorbing media, the variance typically will decrease as p goes from zero to some p', and then increase as p goes from p' to one. For $p < p'$, the solution is "underbiased" and for $p > p'$, the solution is "overbiased."

Choosing p' is usually a matter of experience, although some insight may be gleaned by understanding what happens in severely underbiased and severely overbiased calculations. For illustration, apply the variance analysis in §2.6.6 to a deep penetration problem when the exponential transform is the only non-analog technique used. In a severely underbiased calculation ($p \to 0$), very few particles will score, but those that do will all contribute unity. Thus the variance in an underbiased system is caused by a low scoring efficiency rather than a large dispersion in the weights of the penetrating particles. In a severely overbiased system ($p \to 1$) particles will score, but there will be a large dispersion in the weights of the penetrating particles with a resulting increase in variance.

Comments: the MCNP code gives a warning message if the exponential transform is used without a weight window. There are numerous examples where an exponential transform without a weight window gives unreliable means and error estimates. However, with a good weight window both the means and errors are well behaved. The exponential transform works best on highly absorbing media and very poorly on highly scattering media. For neutron penetration of concrete or earth, experience indicates that a transform parameter $p = 0.7$ is about optimal. For photon penetration of high-Z material, even higher values such as $p = 0.9$ are justified.

The following explains what happens with an exponential transform without a weight window. For simplicity consider a slab of thickness T with constant Σ_t. Let the tally be a simple count (F1 tally) of the weight penetrating the slab and let the exponential transform be the only non-analog technique used. Suppose for a given penetrating history that there are k flights, m that collide and n that do not collide. The penetrating weight is thus

$$w_p = \prod_{i=1}^{m} \frac{\exp(-\rho\Sigma_t \mu_i s_i)}{(1 - p\mu_i)} \prod_{j=m+1}^{k} \exp(-\rho\Sigma_t \mu_j s_j). \tag{2.256}$$

However, the particle's penetration of the slab means that

$$\sum_{l=1}^{k} \mu_l s_l = T \tag{2.257}$$

and hence

$$w_p = \exp(-\rho\Sigma_t T) \prod_{i=1}^{m} (1 - p\mu_i)^{-1}. \tag{2.258}$$

The only variation in w_p is because of the $(1 - p\mu)^{-1}$ factors that arise only from collisions. For a perfectly absorbing medium, every particle that penetrates scores exactly $T \exp(-p\Sigma_t)$. If a particle has only a few collisions, the weight variation will be small compared to a particle that has many collisions. The weight window splits the particle whenever the weight gets too large, depriving the particle of getting a whole series of weight multiplications upon collision that are substantially greater than one.

By setting $p = \Sigma_a/\Sigma_t$ and $\mu = 1$ so that $\Sigma^* = \Sigma_s$, we sample distance to scatter rather than distance to collision. It is preferable to sample distance to scatter in highly absorbing media—in fact, this is the standard procedure for astrophysics problems. Sampling distance to scatter is also equivalent to implicit capture along a flight path [§2.4.3.4.3]. However, in such highly absorbing media there is usually a more optimal choice of transform parameter, p, and it is usually preferable to take advantage of the directional component by not fixing $\mu = 1$.

2.7.2.14 Implicit Capture

"Implicit capture," "survival biasing," and "absorption by weight reduction" are synonymous. Implicit capture, controlled with the MCNP CUT card, is a variance reduction technique applied in the MCNP code after the collision nuclide has been selected. Let

σ_{ti}	be the total macroscopic cross section for nuclide i and
σ_{ai}	be the microscopic absorption cross section for nuclide i.

When implicit capture is used rather than sampling for absorption with probability σ_{ai}/σ_{ti}, the particle always survives the collision and is followed with new weight $W(1-\sigma_{ai}/\sigma_{ti})$. Implicit capture is a splitting process where the particle is split into absorbed weight (which need not be followed further) and surviving weight.

Implicit capture can also be done along a flight path rather than at collisions when a special form of the exponential transform is used. See §2.4.3.4.3 for details.

Two advantages of implicit capture are

1. a particle that has finally, against considerable odds, reached the tally region and is not absorbed just before a tally is made, and
2. the history variance, in general, decreases when the surviving weight (that is, 0 or W) is not sampled, but an expected surviving weight is used instead (see weight cutoff, §2.7.2.11).

Two disadvantages are

1. a fluctuation in particle weight is introduced, and
2. the time per history is increased (see weight cutoff, §2.7.2.11).

2.7.2.15 Forced Collisions

The forced collision method, controlled with the MCNP `FCL` card, is a variance reduction scheme that increases sampling of collisions in specified cells. Because detector contributions and DXTRAN particles arise only from collisions and at the source, it is often useful in certain cells to increase the number of collisions that can produce large detector contributions or large weight DXTRAN particles. Sometimes we want to sample collisions in a relatively thin cell (a fraction of a mean free path) to improve the estimate of quantities like a reaction rate or energy deposition or to cause collisions that are important to some other part of the problem.

The forced collision method splits particles into collided and uncollided parts. The collided part is forced to collide within the current cell. The uncollided part exits the current cell without collision and is stored in the bank until later when its track is continued at the cell boundary. Its weight is

$$W = W_0 \exp(-\Sigma_t d), \tag{2.259}$$

where

W_0	is the current particle weight before forced collision,
d	is the distance to cell surface in the particle's direction, and
Σ_t	is the macroscopic total cross section of the cell material.

That is, the uncollided part is the current particle weight multiplied by the probability of exiting the cell without collision.

The collided part has weight $W = W_0(1 - \exp(-\Sigma_t d))$, which is the current particle weight multiplied by the probability of colliding in the cell. The uncollided part is always produced. The collided part may be produced only a fraction f of the time, in which case the collided weight is $W_0(1 - \exp(-\Sigma_t d))/f$. This is useful when several forced collision cells are adjacent or when too much time is spent producing and following forced collision particles.

The collision distance is sampled as follows. If $P(x)$ is the unconditional probability of colliding within a distance x, $P(x)/P(d)$ is the conditional probability of colliding within a distance x given that a collision is known to occur within a distance d. Thus the position x of the collision must be sampled on the interval $0 < x < d$ within the cell according to $\xi = P(x)/P(d)$, where and ξ is a random number. Solving for x, one obtains

$$x = -\frac{1}{\Sigma_t} \ln\{1 - \xi[1 - \exp(-\Sigma_t d)]\}. \tag{2.260}$$

Because a forced collision usually yields a collided particle having a relatively small weight, care must be taken with the weight-cutoff game [§2.7.2.11], the weight-window game [§2.7.2.12], and subsequent collisions of the particle within the cell. The weight window game is not played on the surface of a forced collision cell that the particle is entering. For collisions inside the cell the user has two options.

Option 1	(negative entry for the cell on the forced collision card) After the forced collision, subsequent collisions of the particle are sampled normally. The weight cutoff game is turned off and detector contributions and DXTRAN particles are made before the weight window game is played. If weight windows are used, they should be set to the weight of the collided particle weight or set to zero if detector contributions or DXTRAN particles are desired.
Option 2	(positive entry for the cell on the forced collision card) After the forced collision, detector contributions or DXTRAN particles are made and either the weight cutoff or weight window game is played. Surviving collided particles undergo subsequent forced collisions. If weight windows are used, they should bracket the weight of particles entering the cell.

2.7.2.16 Source Variable Biasing

Provision is made for biasing the MCNP sources in any or all of the source variables specified. The MCNP code's source biasing, although not completely general, allows the production of more source particles, with suitably reduced weights, in the more important regimes of each variable. For example, one may start more "tracks" at high energies and in strategic directions in a shielding problem while correcting the distribution by altering the weights assigned to these tracks. Sizable variance reductions may result from such biasing of the source. Source biasing samples from a non-analog probability density function.

If negative weight cutoff values are used on the `CUT` card, the weight cutoff is made relative to the lowest value of source particle weight generated by the biasing schemes. Two approaches are available:

1. Biasing by Specifying Explicit Sampling Frequencies: The `SB` input card determines source biasing for a particular variable by specifying the frequency at which source particles will be produced in the variable regime. If this fictitious frequency does not correspond to the fraction of actual source particles in a variable bin, the corrected weight of the source particles in a particular bin is determined by the ratio of the actual frequency (defined on the `SP` card) divided by the fictitious frequency (defined on the `SB` card) except for the lin-lin interpolation where it is defined to be the ratio of the actual to fictitious frequency evaluated at the exact value of the interpolated variable. The total weight of particles started in a given `SI` bin interval is thus conserved.

2. Biasing by Standard Prescription: Source biasing can use certain built-in prescriptions similar in principle to built-in analytic source distributions. These biasing options are detailed in §2.7.2.16.1, §2.7.2.16.2, §2.7.2.16.3, §2.7.2.16.4 for the appropriate source variables. The SB card input is analogous to that of an SP card for an analytic source distribution; that is, the first entry is a negative prescription number for the type of biasing required, followed by one or more optional user-specified parameters, which are discussed in the following sections.

2.7.2.16.1 Direction Biasing

The source direction can be biased (about a reference axis) by sampling from a continuous exponential function or by using cones of fixed size and starting a fixed fraction of particles within each cone. The user can bias particles in any arbitrary direction or combination of directions. The sampling of the azimuthal angle about the reference axis is not biased.

In general, continuous biasing is preferable to fixed cone biasing because cone biasing can cause problems from the discontinuities of source track weight at the cone boundaries. However, if the cone parameters (cone size and fraction of particles starting in the cone) are optimized through a parameter study and the paths that tracks take to contribute to tallies are understood, fixed cone biasing sometimes can outperform continuous biasing. Unfortunately, it is usually time consuming (both human and computer) and difficult to arrive at the necessary optimization.

Source directional biasing can be sampled from an exponential probability density function $p(\mu) = C\exp(K\mu)$, where C is a norming constant equal to $K/[\exp(K) - \exp(-K)]$ and $\mu = \cos\theta$, where θ is an angle relative to the biasing direction. K is typically about 1; $K = 3.5$ defines the ratio of weight of tracks starting in the biasing direction to tracks starting in the opposite direction to be 1/1097. This ratio is equal to $[1 - \exp(-2K)]/[\exp(2K) - 1]$.

Table 2.10 may help to give the user a feel for the biasing parameter K.

From this table for $K = 1$, we see that half the tracks start in a cone of 64° opening about the axis, and the weight of tracks at 64° is 0.762 times the unbiased weight of source particles. $K = 0.01$ is almost equivalent to no biasing, and $K = 3.5$ is very strong.

Cone directional biasing can be invoked by specifying cone cosines on the SI card, the true distribution on the SP card, and the desired biasing probabilities on the SB card. Both histogram and linear interpolation can be used. For example, consider the following case in which the true distribution is isotropic:

SIn	-1	ν	1
SPn	0	$\frac{1+\nu}{2}$	$\frac{1-\nu}{2}$
SBn	0	p_1	p_2

The direction cosine relative to the reference direction, say ν, is sampled uniformly within the cone $\nu' < \nu < 1$ with probability p_2 and within $-1 < \nu < \nu'$ with the complementary probability p_1. The weights assigned are $w(1-\nu)/(2p_2)$ and $w(1+\nu)/(2p_1)$, respectively. Note that for a very small cone defined by ν' and a high probability $p_2 \gg p_1$ for being within the cone, the few source particles generated outside the cone will have a very high weight that can severely perturb a tally.

2.7.2.16.2 Covering Cylinder Extent Biasing

This biasing prescription for the SDEF EXT variable allows the automatic spatial biasing of source particles in a cylindrical-source-covering-volume along the axis of the cylinder. Such biasing can aid in the escape of source particles from optically thick source regions and thus represents a variance reduction technique.

Table 2.10: Exponential Biasing Parameter

K	Cumulative Probability	θ	Weight
0.01	0	0	0.990
	0.25	60	0.995
	0.50	90	1.000
	0.75	120	1.005
	1.00	180	1.010
1.0	0	0	0.432
	0.25	42	0.552
	0.50	64	0.762
	0.75	93	1.230
	1.00	180	3.195
2.0	0	0	0.245
	0.25	31	0.325
	0.50	48	0.482
	0.75	70	0.931
	1.00	180	13.40
3.5	0	0	0.143
	0.25	23	0.190
	0.50	37	0.285
	0.75	53	0.569
	1.00	180	156.5

2.7.2.16.3 Covering Cylinder or Sphere Radial Biasing

This biasing prescription for the SDEF RAD variable allows for the radial spatial biasing of source particles in either a spherical or cylindrical source covering volume. Like the previous example of extent biasing, this biasing can be used to aid in the escape of source particles from optically thick source regions.

2.7.2.16.4 Biasing Standard Analytic Source Functions [164]

The preceding examples discuss the biasing of source variables by either input of specific sampling frequencies corresponding to SP card entries or by standard analytic biasing functions. A third biasing category can be used in conjunction with standard analytic source probability functions (for example, a Watt fission spectrum).

A negative entry on an SP card, that is,

```
SPn -i a b
```

causes the MCNP code to sample source distribution n from probability function i with input variables $a, b, \ldots$ Sampling schemes cannot typically be biased. For example, for

```
SPn -5 a
```

the evaporation spectrum $f(E) = CE\exp(-E/a)$ is sampled according to the sampling prescription $E = -a\log(\xi_1 \cdot \xi_2)$, where ξ_1 and ξ_2 are random numbers. Biasing this sampling scheme is usually very difficult

or impossible. Fortunately, there is an approximate method available in the MCNP code for biasing any arbitrary probability function [164]. The code approximates the function as a table, then uses the usual SB card biasing scheme to bias this approximate table function. The user inputs a coarse bin structure to govern the bias and the code adds up to 300 additional equiprobable bins to assure accuracy. For example, suppose we wish to sample the function

$$f(E) = CE \exp\left(-\frac{E}{a}\right) \tag{2.261}$$

and suppose that we want half the source to be in the range $0.005 < E < 0.1$ and the other half to be in the range $0.1 < E < 20$. Then the input is

SIn	0.005	0.1	20	
SPn		-5	a	
SBn	C	0	0.5	1

The MCNP code breaks up the function into 150 equiprobable bins below $E = 0.1$ and 150 more equiprobable bins above $E = 0.1$. Half the time E is chosen from the upper set of bins and half the time it is chosen from the lower set. Particles starting from the upper bins have a different weight from that of particles starting from the lower bins in order to adjust for the bias, and a detailed summary is provided when the PRINT option is used.

Note that in the above example the probability distribution function is truncated below $E = 0.005$ and above $E = 20$. The MCNP code prints out how much of the distribution is lost in this manner and reduces the weight accordingly.

It is possible for the user to choose a foolish biasing scheme. For example,

SIn	0.005	297I	0.1	20
SPn		-5	a	
SBn	0	1	298R	

causes each of the 299 bins to be chosen with equal probability. This would be all right except that since there are never more than 300 equiprobable bins, this allocates only 1 equiprobable bin per user-supplied bin. The single equiprobable bin for $0.1 < E < 20$ is inadequate to describe the distribution function over this range. Thus the table no longer approximates the function and the source will be sampled erroneously. The MCNP code issues an error message whenever too much of the source distribution is allocated to a single equiprobable bin, alerting users to a poor choice of binning which might inadequately represent the function. The coarse bins used for biasing should be chosen so that the probability function is roughly equally distributed among them.

2.7.2.17 Point Detector Tally

The point detector is a tally and does not bias random-walk sampling. Recall from §2.6, however, that the tally choice affects the efficiency of a Monte Carlo calculation. Thus, a little will be said here in addition to the discussion in the tally section.

Although flux is a point quantity, flux at a point cannot be estimated by either a track-length tally (F4) or a surface flux tally (F2) because the probability of a track entering the volume or crossing the surface of a point is zero. For very small volumes, a point detector tally can provide a good estimate of the flux where it would be almost impossible to get either a track-length or surface-crossing estimate because of the low probability of crossing into the small volume.

It is interesting that a DXTRAN sphere of vanishingly small size with a surface-crossing tally across the diameter normal to the particle's trajectory is equivalent to a point detector. Thus, many of the comments on DXTRAN are appropriate and the DXC cards essentially are identical to the PD cards.

For a complete discussion of point detectors, see §2.5.6.1.

2.7.2.18 DXTRAN

DXTRAN, controlled with the MCNP DXT card, typically is used when a small region is being inadequately sampled because particles have a very small probability of scattering toward that region. To ameliorate this situation, the user can specify in the input file a DXTRAN sphere that encloses the small region. Upon collision (or exiting the source) outside the sphere, DXTRAN creates a special "DXTRAN particle" and deterministically scatters it toward the DXTRAN sphere and deterministically transports it, without collision, to the surface of the DXTRAN sphere. The collision itself is otherwise treated normally, producing a non-DXTRAN particle that is sampled in the normal way, with no reduction in weight. However, the non-DXTRAN particle is killed if it tries to enter the DXTRAN sphere on its next free flight. DXTRAN uses a combination of splitting, Russian roulette, and sampling from a non-analog probability density function.

The subtlety about DXTRAN is how the extra weight created for the DXTRAN particles is balanced by the weight killed as non-DXTRAN particles cross the DXTRAN sphere. The non-DXTRAN particle is followed without any weight correction, so if the DXTRAN technique is to be unbiased, the extra weight put on the DXTRAN sphere by DXTRAN particles must somehow (on average) balance the weight of non-DXTRAN particles killed on the sphere.

2.7.2.18.1 DXTRAN Viewpoint 1

One can view DXTRAN as a splitting process (much like the forced collision technique) wherein each particle is split upon departing a collision (or source point) into two distinct pieces:

1. the weight that does not enter the DXTRAN sphere on the next flight, either because the particle is not pointed toward the DXTRAN sphere or because the particle collides before reaching the DXTRAN sphere, and
2. the weight that enters the DXTRAN sphere on the next flight.

Let w_0 be the weight of the particle before exiting the collision, let p_1 be the analog probability that the particle does not enter the DXTRAN sphere on its next flight, and let p_2 be the analog probability that the particle does enter the DXTRAN sphere on its next flight. The particle must undergo one of these mutually exclusive events, thus $p_1 + p_2 = 1$. The expected weight not entering the DXTRAN sphere is $w_1 = w_0 p_1$, and the expected weight entering the DXTRAN sphere is $w_2 = w_0 p_2$. Think of DXTRAN as deterministically splitting the original particle with weight w_0 into two particles, a non-DXTRAN (particle 1) particle of weight w_1 and a DXTRAN (particle 2) particle of weight w_2. Unfortunately, things are not quite that simple.

Recall that the non-DXTRAN particle is followed with unreduced weight w_0 rather than weight $w_1 = w_0 p_1$. The reason for this apparent discrepancy is that the non-DXTRAN particle (particle 1) plays a Russian roulette game. Particle 1's weight is increased from w_1 to w_0 by playing a Russian roulette game with survival probability $p_1 = w_1/w_0$. The reason for playing this Russian roulette game is simply that p_1 is not known, so assigning weight $w_1 = p_1 w_0$ to particle 1 is impossible. However, it is possible to play the Russian roulette game without explicitly knowing p_1. It is not magic, just slightly subtle.

The Russian roulette game is played by sampling particle 1 normally and keeping it only if it does not enter (on its next flight) the DXTRAN sphere; that is, particle 1 survives (by definition of p_1) with probability p_1. Similarly, the Russian roulette game is lost if particle 1 enters (on its next flight) the DXTRAN sphere; that is, particle 1 loses the roulette with probability p_2. To restate this idea, with probability p_1, particle 1 has weight w_0 and does not enter the DXTRAN sphere and with probability p_2, the particle enters the DXTRAN sphere and is killed. Thus, the expected weight not entering the DXTRAN sphere is $w_0 p_1 + 0 \cdot p_2 = w_1$, as desired.

So far, this discussion has concentrated on the non-DXTRAN particle and ignored exactly what happens to the DXTRAN particle. The sampling of the DXTRAN particle will be discussed after a second viewpoint on the non-DXTRAN particle.

2.7.2.18.2 DXTRAN Viewpoint 2

This second way of viewing DXTRAN does not see DXTRAN as a splitting process but as an accounting process in which weight is both created and destroyed on the surface of the DXTRAN sphere. In this view, DXTRAN estimates the weight that should go to the DXTRAN sphere upon collision and creates this weight on the sphere as DXTRAN particles. If the non-DXTRAN particle does not enter the sphere, its next flight will proceed exactly as it would have without DXTRAN, producing the same tally contributions and so forth. However, if the non-DXTRAN particle's next flight attempts to enter the sphere, the particle must be killed or there would be (on average) twice as much weight crossing the DXTRAN sphere as there should be because the weight crossing the sphere has already been accounted for by the DXTRAN particle.

2.7.2.18.3 The DXTRAN Particle

Although the DXTRAN particle does not confuse people nearly as much as the non-DXTRAN particle, the DXTRAN particle is nonetheless subtle.

The most natural approach for scattering particles toward the DXTRAN sphere would be to sample the scattering angle Ω proportional to the analog density. This approach is not used because it is too much work to sample proportional to the analog density and because it is sometimes useful to bias the sampling.

To sample Ω in an unbiased fashion when it is known that Ω points to the DXTRAN sphere, one samples the conditional density

$$P_{\text{con}}(\Omega) = \frac{P(\Omega)}{\int_{S(\Omega)} P(\Omega)\text{d}\Omega}, \tag{2.262}$$

where $S(\Omega)$ is the set of directions pointed toward the DXTRAN sphere, and multiplies the weight by $\int_{S(\Omega)} P(\Omega)\text{d}\Omega$, the probability of scattering into the cone (see Fig. 2.26). However, it is too much work to calculate the above integral for each collision. Instead, an arbitrary density function $P_{\text{arb}}(\Omega)$ is sampled and the weight is multiplied by

$$\frac{P_{\text{con}}(\Omega)}{P_{\text{arb}}(\Omega)} = \frac{P(\Omega)}{P_{\text{arb}}(\Omega)\int_{S(\Omega)} P(\Omega)\text{d}\Omega} \tag{2.263}$$

The total weight multiplication is the product of the fraction of the weight scattering into the cone, $\int_{S(\Omega)} P(\Omega)\text{d}\Omega$, and the weight correction for sampling $P_{\text{arb}}(\Omega)$ instead of $P_{\text{con}}(\Omega)$. Thus, the weight correction on scattering is

$$\frac{P(\Omega)}{P_{\text{arb}}(\Omega)}.$$

If μ is the cosine of the angle between the scattering direction and the particle's incoming direction, then $P(\Omega) = P(\mu)/2\pi$ because the scattering is symmetric in the azimuthal angle. If η is the cosine of the angle with respect to the cone axis (see Fig. 2.26) and if the azimuthal angle about the cone axis is uniformly sampled, then $P_{\text{arb}}(\Omega) = P_{\text{arb}}(\eta)/2\pi$. Thus

$$\frac{P(\mu)}{P_{\text{arb}}(\eta)} = \text{weight multiplier for DXTRAN particle.} \tag{2.264}$$

This result can be obtained more directly, but the other derivation does not explain why $P_{\text{con}}(\Omega)$ is not sampled.

Because $P_{\text{arb}}(\eta)$ is arbitrary, the MCNP code can choose a scheme that samples η from a two-step density that favors particles within the larger η interval. In fact, the inner DXTRAN sphere has to do only with this arbitrary density and is not essential to the DXTRAN concept. The DXTRAN particles are always created on the outside DXTRAN sphere, with the inner DXTRAN sphere defining only the boundary between the two steps in the density function.

After $\eta = \cos\theta$ has been chosen, the azimuthal angle φ is sampled uniformly on $[0, 2\pi)$; this completes the scattering. Recall, however, that the DXTRAN particle arrives at the DXTRAN sphere without collision. Thus the DXTRAN particle also has its weight multiplied by the negative exponential of the optical path between the collision site and the sphere. Thus the DXTRAN weight multiplication is

$$\frac{P(\mu)}{P_{\text{arb}}(\eta)} \exp(-\lambda), \tag{2.265}$$

where λ is the number of mean free paths from the exit site to the chosen point on the DXTRAN sphere.

2.7.2.18.4 Inside the DXTRAN Sphere

So far, only collisions outside the DXTRAN sphere have been discussed. At collisions inside the DXTRAN sphere, the DXTRAN game is not played because first, the particle is already in the desired region, and second, it is impossible to define the angular cone of Fig. 2.26. If there are several DXTRAN spheres and the collision occurs in sphere i, DXTRAN will be played for all spheres except sphere i.

2.7.2.18.5 Real Particles vs. Pseudoparticle

Sometimes the DXTRAN particle is called a pseudoparticle and the non-DXTRAN particle is called the original or real particle. The terms “real particle” and “pseudoparticle” are potentially misleading. Both particles are equally real: both execute random walks, both carry nonzero weight, and both contribute to tallies. The only sense in which the DXTRAN particle should be considered "pseudo" or “not real” is during creation. A DXTRAN particle is created on the DXTRAN sphere, but creation involves determining what weight the DXTRAN particle should have upon creation. Part of this weight determination requires calculating the optical path between the collision site and the DXTRAN sphere. This is done in the same way as point detectors (see point detector pseudoparticles in §2.5.6.4.1). The MCNP code determines the optical path by tracking a pseudoparticle from the collision site to the DXTRAN sphere. This pseudoparticle is deterministically tracked to the DXTRAN sphere simply to determine the optical path. No distance to collision is sampled, no tallies are made, and no records of the pseudoparticle’s passage are kept (for example, tracks entering). In contrast, once the DXTRAN particle is created at the sphere’s surface, the particle is no longer a pseudoparticle. The particle has real weight, executes random walks, and contributes to tallies.

2.7.2.18.6 DXTRAN Details

To explain how the scheme works, consider the neighborhood of interest to be a spherical region surrounding a designated point in space. In fact, consider two spheres of arbitrary radii about the point $P_0 = (x_0, y_0, z_0)$. Further, assume that the particle having direction (u, v, w) collides at the point $P_1 = (x, y, z)$, as shown in Fig. 2.26. The quantities θ_I, θ_O, η_I, η_O, R_I, R_O are defined in the figure. Thus L, the distance between the collision point and center of the spheres, is

$$L = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}. \tag{2.266}$$

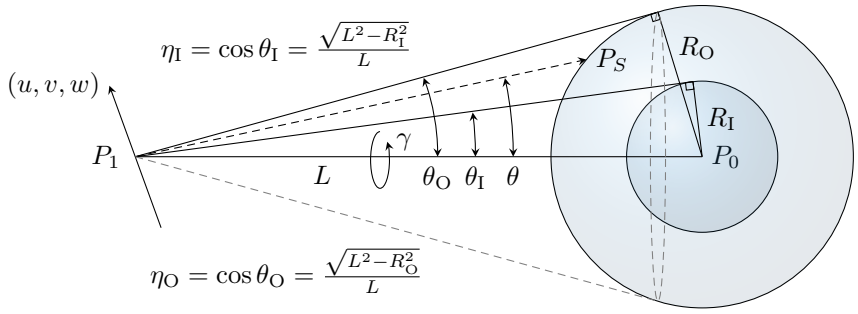


Figure 2.26: Diagram of DXTRAN inner and outer spheres.

On collision, a DXTRAN particle is placed at a point on the outer sphere of radius R_O as described below. Provision is made for biasing the contributions of these DXTRAN particles on the outer sphere within the cone defined by the inner sphere. The weight of the DXTRAN particle is adjusted to account for the probability of scattering in the direction of the point on the outer sphere and traversing the distance with no further collision.

The steps in sampling the DXTRAN particles are outlined next. First, sample

$$\eta_I = \cos\theta_I = \frac{\sqrt{L^2 - R_I^2}}{L}, \tag{2.267a}$$

$$\eta_O = \cos\theta_O = \frac{\sqrt{L^2 - R_O^2}}{L}. \tag{2.267b}$$

Next, sample $\eta = \eta_I + \xi(1 - \eta_I)$ uniformly in $[\eta_I, 1)$ with probability

$$\frac{Q(1 - \eta_I)}{Q(1 - \eta_I) + \eta_I + \eta_O}$$

and with probability

$$\frac{\eta_I - \eta_O}{Q(1 - \eta_I) + \eta_I + \eta_O}$$

sample $\eta = \eta_O + \xi(\eta_I - \eta_O)$ uniformly in $[\eta_O, \eta_I)$. The quantity Q (equal to 5 in the MCNP code) is a factor that measures the importance assigned to scattering in the inner cone relative to the outer cone. Therefore, Q is also the ratio of weights for particles put in the two different cones.

With $\eta = \cos\theta$ chosen, a new direction (u', v', w') is computed by considering the rotation through the polar angle θ (and a uniform azimuthal angle φ) from the reference direction

$$\left(\frac{x_0 - x}{L}, \frac{y_0 - y}{L}, \frac{z_0 - z}{L}\right)$$

The particle is advanced in the direction (u', v', w') to the surface of the sphere of radius R_O. The new DXTRAN particle with appropriate direction and coordinates is banked. The weight of the DXTRAN particle is determined by multiplying the weight of the particle at collision by

$$\nu \cdot \frac{P(\mu)\{Q(1 - \eta_I) + \eta_I - \eta_O\} \exp\left(-\int_{P_I}^{P_S} \sigma_{\mathrm{t}}(s)\mathrm{d}s\right)}{Q}, \ \eta_I \le \eta < 1$$

and

$$\nu \cdot P(\mu)\{Q(1 - \eta_I) + \eta_I - \eta_O\} \exp\left(-\int_{P_I}^{P_S} \sigma_{\mathrm{t}}(s)\mathrm{d}s\right), \ \eta_O \le \eta < \eta_I,$$

where

μ	is $uu' + vv' + ww'$,
$P(\mu)$	is the scattering probability density function for scattering through the angle $\cos^{-1}\mu$ in the lab system for the event sampled at (x, y, z),
ν	is the number of particles emitted from the event, and
$\exp\left(-\int_{P_I}^{P_S}\sigma_t(s)\mathrm{d}s\right)$	is the attenuation along the line between $P_I(x, y, z)$ and P_S, the point on the sphere where the particle is placed.

In arriving at the weight factor, note that the density function for sampling η is given by

$$p(\eta) = \begin{cases} \frac{Q}{Q(1-\eta_I)+\eta_I-\eta_O} & \eta_I < \eta < 1 \\ \frac{1}{Q(1-\eta_I)+\eta_I-\eta_O} & \eta_O \le \eta < \eta_I \end{cases} \tag{2.268}$$

Thus the weight of the DXTRAN particle is the weight of the incoming particle at P_I modified by the ratio of the probability density function for actually scattering from P_I and arriving at P_S without collision to the density function actually sampled in choosing P_S. Therefore, particles in the outer cone have weights $Q = 5$ times higher than the weights of similar particles in the inner cone.

The attenuation is calculated at the energy obtained by scattering through the angle μ. The energy is uniquely determined from μ in elastic scattering (and also in level scattering), whereas for other nonelastic events, the energy is sampled from the corresponding probability density function for energy, and may not depend on μ.

2.7.2.18.7 Auxiliary Games for DXTRAN

The major disadvantage to DXTRAN is the extra time consumed following DXTRAN particles with low weights. Three special games can control this problem:

1. DXTRAN weight cutoffs,
2. DXC games, and
3. DD game.

Particles inside a DXTRAN sphere are not subject to the normal MCNP weight cutoff or weight window game. Instead DXTRAN spheres have their own weight cutoffs, allowing the user to roulette DXTRAN particles that, for one reason or another, do not have enough weight to be worth following.

Sometimes low-weighted DXTRAN particles occur because of collisions many free paths from the DXTRAN sphere. The exponential attenuation causes these particles to have extremely small weights. The DXTRAN weight cutoff will roulette these particles only after much effort has been spent producing them. The DXC cards are cell dependent and allow DXTRAN contributions to be taken only some fraction of the time. They work just like the PD cards for detectors [§2.5.6.4.3]. The user specifies a probability p_i that a DXTRAN particle will be produced at a given collision or source sampling in cell i. The DXTRAN result remains unbiased because when a DXTRAN particle is produced its weight is multiplied by p_i^{-1}. The non-DXTRAN particle is treated exactly as before, unaffected unless it enters the DXTRAN sphere, whereupon it is killed. To see the utility, suppose that the DXTRAN weight cutoff was immediately killing 99% of the DXTRAN particles from cell i. Only 1% of the DXTRAN particles survive anyway, so it might be appropriate to produce only 1% ($p_i = 0.01$) and have these not be killed immediately by the DXTRAN weight cutoff. Or the p_is can often be set such that all DXTRAN particles from all cells are created on the DXTRAN sphere with roughly the same

weight. Choosing the p_is is often difficult and the method works well typically when the material exponential attenuation is the major source of the weight fluctuation.

Often the weight fluctuation arises because the probability $P(\mu)$ of scattering toward the DXTRAN sphere varies greatly, depending on what nuclide is hit and what the collision orientation is with respect to the DXTRAN sphere. For example, consider a highly forward-peaked scattering probability density. If the DXTRAN sphere were close to the particle's pre-collision direction, $P(\mu)$ will be large; if the DXTRAN sphere were at 105° to the pre-collision direction, $P(\mu)$ will be small. The DD game can be used to reduce the weight fluctuation on the DXTRAN sphere caused by these geometry effects, as well as the material exponential attenuation effects.

The DD game selectively roulettes the DXTRAN pseudoparticles during creation, depending on the DXTRAN particles' weight compared to some reference weight. This is the same game that is played on detector contributions, and is described in §2.5.6.4.3. The reference weight can be either a fraction of the average of previous DXTRAN particle weights or a user input reference weight. Recall that a DXTRAN particle's weight is computed by multiplying the exit weight of the non-DXTRAN particle by a weight factor having to do with the scattering probability and the negative exponential of the optical path between the collision site and DXTRAN sphere. The optical path is computed by tracking a pseudoparticle from collision to the DXTRAN sphere. The weight of the pseudoparticle is monotonically decreasing, so the DD game compares the pseudoparticle's weight at the collision site and, upon exiting each cell, against the reference weight. A roulette game is played when the pseudoparticle's weight falls below the reference weight. The DD card stops tracking a pseudoparticle as soon as the weight becomes inconsequential, saving time by eliminating subsequent tracking.

2.7.2.18.8 Final Comments on DXTRAN

1. DXTRAN should be used carefully in optically thick problems. Do not rely on DXTRAN to do penetration.
2. If the source is user supplied, some provision must be made for obtaining the source contribution to particles on the DXTRAN sphere.
3. Extreme care must be taken when more than one DXTRAN sphere is in a problem. Cross-talk between spheres can result in extremely low weights and an excessive growth in the number of particle tracks.
4. Never put a zero on the DXC card. A zero will bias the calculation by not creating DXTRAN particles but still killing the non-DXTRAN particle if it enters the DXTRAN sphere.
5. Usually there should be a rough balance in the summary table of weight created and lost by DXTRAN.
6. DXTRAN cannot be used with reflecting surfaces for the same reasons that point detectors cannot be used with reflecting surfaces. See §2.5.6.4.2 for further explanation.
7. Both DXTRAN and point detectors track pseudoparticles to a point. Therefore, most of the discussion about detectors applies to DXTRAN. Refer to the section on detectors, §2.5.6, for more information.

2.7.2.19 Correlated Sampling

Correlated sampling estimates the change in a quantity resulting from a small alteration of any type in a problem. This technique enables the evaluation of small quantities that would otherwise be masked by the statistical errors of uncorrelated calculations. The MCNP code correlates a pair of runs by providing each new history in the original and altered problems with the same starting pseudorandom number. The same sequence of subsequent numbers is used and each history tracks identically until the alteration causes

the tracking to diverge. The sequencing of random numbers is done by incrementing the random number generator, controlled with the MCNP RAND card, at the beginning of each history by a stride S of random numbers from the beginning of the previous history. The stride should be a quantity greater than would be needed by most histories [§2.11].

The MCNP code does not provide an estimate of the error in the difference. Reference [160] shows how the error in the difference between two correlated runs can be estimated. A post-processor code would have to be written to do this.

Correlated sampling should not be confused with more elaborate Monte Carlo perturbation schemes that calculate differences and their variances directly. The MCNP code also has a sophisticated perturbation capability.

2.8 Criticality Calculations

Nuclear criticality, the ability to sustain a chain reaction by fission neutrons, is characterized by k_{eff}, the eigenvalue to the time-independent neutron transport equation. In reactor theory, k_{eff} is thought of as the ratio between the number of neutrons in successive generations, with the fission process regarded as the birth event that separates generations of neutrons [78]. For critical systems, $k_{\text{eff}} = 1$ and the chain reaction will just sustain itself. For subcritical systems, $k_{\text{eff}} < 1$ and the chain reaction will not sustain itself. For supercritical systems, $k_{\text{eff}} > 1$ and the number of fissions in the chain reaction will increase with time. In addition to the geometry description and material cards, all that is required to run a criticality problem is a KCODE card, described below, and an initial spatial distribution of fission points using either the KSRC card, the SDEF card, or a SRCTP file.

Calculating k_{eff} consists of estimating the mean number of fission neutrons produced in one generation per fission neutron started. A generation is the life of a neutron from birth in fission to death by escape, parasitic capture, or absorption leading to fission. In the MCNP code, the computational equivalent of a fission generation is a k_{eff} cycle; that is, a cycle is a computed estimate of an actual fission generation. Processes such as (n,2n) and (n,3n) are considered internal to a cycle and do not act as termination. Because fission neutrons are terminated in each cycle to provide the fission source for the next cycle, a single history can be viewed as continuing from cycle to cycle. The effect of the delayed neutrons is included by using the total $\overline{\nu}$ when the data are available. In a MODE N P problem, secondary photon production from neutrons is turned off during inactive cycles. The MCNP code uses three different estimators for k_{eff}. We recommend using, for the final k_{eff} result, the statistical combination of all three [165].

It is extremely important to emphasize that the result from a criticality calculation is a confidence interval for k_{eff} that is formed using the final estimated k_{eff} and the estimated standard deviation. A properly formed confidence interval from a valid calculation should include the true answer the fraction of time used to define the confidence interval. For example, 68% of confidence intervals formed at the 68% confidence level, which corresponds to roughly one standard deviations of the mean for a normal distribution, will contain the true answer. There will always be some probability that the true answer lies outside of a confidence interval.

Reference [166] is an introduction to using the MCNP code for criticality calculations, focusing on the unique aspects of setting up and running a criticality problem and interpreting the results. A quick-start chapter gets the new MCNP user on the computer running a simple criticality problem as quickly as possible.

2.8.1 Criticality Program Flow

Because the calculation of k_{eff} entails running successive fission cycles, criticality calculations have a different program flow than MCNP fixed source problems. They require a special criticality source that is incompatible with the surface source and user-supplied sources. Unlike fixed source problems, where the source being sampled throughout the problem never changes, the criticality source changes from cycle to cycle.

2.8.1.1 Criticality Problem Definition

To set up a criticality calculation, the user initially supplies an MCNP input file that includes the KCODE card with the following information:

1. the nominal number of source histories, N, per k_{eff} cycle;
2. an initial guess of k_{eff} ;
3. the number of source cycles, I_c, to skip before k_{eff} accumulation; and
4. the total number of cycles, I_t, in the problem.

Other KCODE entries are discussed in §5.8.10. The initial spatial distribution of fission neutrons can be entered by using (1) the KSRC card with sets of x, y, z point locations, (2) the SDEF card to define points uniformly in volume, or (3) a file (SRCTP) from a previous MCNP criticality calculation. If the SDEF card is used, the default WGT value should not be changed. Any KSRC points in geometric cells that are void or have zero importance are rejected. The remaining KSRC points are duplicated or rejected enough times so the total number of points M in the source spatial distribution is approximately the nominal source size N. The energy of each source particle for the first k_{eff} cycle is selected from a generic Watt thermal fission distribution if it is not available from the SRCTP file.

2.8.1.2 Particle Transport for Each k_{eff} Cycle

In each k_{eff} cycle, M (varying with cycle) source particles are started isotropically. For the first cycle, these M points come from one of three user-selected source possibilities. For subsequent cycles, these points are the ones written at collision sites from neutron transport in the previous cycle. The total source weight of each cycle is a constant N. That is, the weight of each source particle is N/M, so all normalizations occur as if N rather than M particles started in each cycle.

Source particles are transported through the geometry by the standard random walk process, except that fission is treated as capture, either analog or implicit, as defined on the PHYS:N or CUT:N card. At each collision point the following four steps are performed for the cycle:

1. the three prompt neutron lifetime estimates are accumulated;
2. if fission is possible, the three k_{eff} estimates are accumulated; and
3. if fission is possible, $n \geq 0$ fission sites (including the sampled outgoing energy of the fission neutron) at each collision are stored for use as source points in the next cycle,

   where

n	$= W\overline{\nu}(\sigma_{\text{f}}/\sigma_{\text{t}})(1/k_{\text{eff}} + \xi)$;
W	is the particle weight (before implicit capture weight reduction or analog capture);
$\overline{\nu}$	is the average number of neutrons produced per fission at the incident energy of this collision, with either prompt $\overline{\nu}$ or total $\overline{\nu}$ (default) used;
σ_{f}	is the microscopic material fission cross section;
σ_{t}	is the microscopic material total cross section; and
k_{eff}	is the estimated collision k_{eff} from previous cycle. For the first cycle, the second KCODE card entry is used.

$M = \sum n$ is the number of fission source points to be used in the next cycle, unless a tally with batch statistics is enabled. In that case, the fission bank is resampled to be exactly `nsrck` at the end of each cycle. The number of fission sites n stored at each collision is rounded up or down to an integer (including zero) with a probability proportional to its closeness to that integer. If the initial guess of k_{eff} is too low or too high, the number of fission sites written as source points for the next cycle will be, respectively, too high or too low relative to the desired nominal number N. A bad initial guess of k_{eff} causes only this consequence.

A very poor initial guess for the spatial distribution of fissions can cause the first cycle estimate of k_{eff} to be extremely low. This situation can occur when only a fraction of the fission source points enter a cell with a fissionable material. As a result, one of two error messages can be printed: (1) no new source points were generated, or (2) the new source has overrun the old source. The second message occurs when the MCNP code's storage for the fission source points is exceeded because the small k_{eff} that results from a poor initial source causes n to become very large.

The fission energy of the next-cycle neutron is sampled separately for each source point and stored for the next cycle. It is sampled from the same distributions as fissions would be sampled in the random walk based on the incident neutron energy and fissionable isotope. The geometric coordinates and cell of the fission site are also stored.

4. The collision nuclide and reaction are sampled (after steps 1, 2, and 3) but the fission reaction is not allowed to occur because fission is treated as capture. The fission neutrons that would have been created are accrued by three different methods to estimate k_{eff} for this cycle. The three estimators are a collision estimator, an absorption estimator and a track-length estimator as discussed in §2.8.2.

2.8.1.3 k_{eff} Cycle Termination

At the end of each k_{eff} cycle, a new set of M source particles has been written from fissions in that cycle. The number M varies from cycle to cycle but the total starting weight in each cycle is a constant N. These M particles are written to the SRCTP file at certain cycle intervals. The SRCTP file can be used as the initial source in a subsequent criticality calculation with a similar, though not identical, geometry. Also, k_{eff} quantities are accumulated, as is described below.

2.8.1.4 Convergence

The first I_c cycles in a criticality calculation are inactive cycles, where the spatial source changes from the initial definition to the correct distribution for the problem. No k_{eff} accumulation, summary table, activity table, or tally information is accrued for inactive cycles. Photon production, perturbations, and DXTRAN are turned off during inactive cycles. I_c is the third entry on the `KCODE` card for the number of k_{eff} cycles to be skipped before k_{eff} and tally accumulation. After the first I_c cycles, the fission source spatial distribution is assumed to have achieved equilibrium, active cycles begin, and k_{eff} and tallies are accumulated. Cycles are run until either a time limit is reached or the total cycles on the `KCODE` card have been completed.

Criticality calculations with the MCNP code are based on an iterative procedure called "power iteration" [167, 168]. After assuming an initial guess for the fission source spatial distribution (i.e., first generation), histories are followed to produce a source for the next fission neutron generation and to estimate a new value for k_{eff}. The new fission source distribution is then used to follow histories for the second generation, producing yet another fission source distribution and estimate of k_{eff}. These generations (also called cycles or batches) are repeated until the source spatial distribution has converged. Once the fission source distribution has converged to its stationary state, tallies for reaction rates and k_{eff} may be accumulated by running additional cycles until the statistical uncertainties have become sufficiently small.

Analysis of the power iteration procedure for solving k_{eff} eigenvalue calculations [167] shows that the convergence of the fission source distribution, $\mathbf{S}$, and the estimated eigenvalue, k_{eff}, can be modeled as

$$\mathbf{S}^{(n+1)} \approx \mathbf{S}_0 + a\left(\frac{k_1}{k_0}\right)^{n+1}\mathbf{S_1} + \cdots \tag{2.269a}$$

$$k_{\text{eff}}^{(n+1)} \approx k_0\left[1 - b\left(\frac{k_1}{k_0}\right)^{n}\left(1 - \frac{k_1}{k_0}\right) + \cdots\right], \tag{2.269b}$$

where $\mathbf{S}_0$ and k_0 are the fundamental eigenfunction and eigenvalue of the exact transport solution, $\mathbf{S}_1$ and k_1 are the eigenfunction and eigenvalue of the first higher mode, a and b are constants, and n is the number of cycles performed in the power iteration procedure. Note that k_0 is the expected value of k_{eff}, and that $k_0 > k_1 > 0$, so that $k_1/k_0 < 1$. The quantity k_1/k_0 is called the *dominance ratio* (DR), and is the key physical parameter that determines the convergence rate of the power iteration procedure. The DR is a function of problem geometry and materials. As the number of cycles n becomes large, the error terms due to higher modes die off as DR^n, and the source distribution and k_{eff} approach their stationary, equilibrium values. For typical light-water reactor systems, the DR is often in the range 0.8–0.99, and 50–100 inactive cycles may be required for errors in the initial guess to die away sufficiently that the source and k_{eff} converge. For some critical systems (e.g., heavy-water reactors, fuel storage vaults), however, the DR may be very close to 1 (e.g., 0.99 or higher), and hundreds or thousands of inactive cycles may be required to attain source convergence.

It should also be noted that the source distribution $\mathbf{S}$ and the eigenvalue k_{eff} do not converge in the same manner. The expression for $k_{\text{eff}}^{(n+1)}$ has the additional factor $1 - (k_1/k_0)$ on the higher-mode error. For problems where the DR is very close to 1, the source distribution may take hundreds or thousands of cycles to converge (because of errors dying out as DR^n), while k_{eff} may converge rapidly (because its higher-mode error is damped by the additional factor $1 - \text{DR}$, which may be very small). That is, k_{eff} will converge more rapidly than the source distribution. Thus, it is very important to examine the behavior of both k_{eff} and the source distribution when assessing problem convergence. **Both k_{eff} and the fission source distribution must converge before starting active cycles for tallies.** It is up to the user to specify the number of inactive cycles I_c to run in order to attain convergence. Most users will make a trial calculation (using a small number of histories per cycle, such as 1000) to examine the convergence behavior of k_{eff} and the source distribution, to determine a proper value for I_c, and then make a final calculation using a larger number of histories per cycle (e.g., 5000 or more) and sufficient active cycles to attain small uncertainties. To assist users in assessing convergence of criticality calculations, the MCNP code provides several statistical checks on k_{eff}, as discussed in the next sections. In addition, the MCNP code calculates a quantity called the entropy of the source distribution, H_{src} [169, 170] to assist users in assessing the convergence of the source distribution.

2.8.2 Estimation of k_{eff} Confidence Intervals and Prompt Neutron Lifetimes

The criticality eigenvalue k_{eff} and various prompt neutron lifetimes, along with their standard deviations, are automatically estimated in every criticality calculation in addition to any user-requested tallies. k_{eff} and the lifetimes are estimated for every active cycle, as well as averaged over all active cycles. k_{eff} and the lifetimes are estimated in three different ways. These estimates are combined [165] using observed statistical correlations to provide the optimum final estimate of k_{eff} and its standard deviation.

It is known [171] that the power iteration method with a fixed source size produces a very small negative bias Δk_{eff} in k_{eff} that is proportional to $1/N$. This bias is negligible [171] for all practical problems where N is greater than about 200 neutrons per cycle and as long as too many active cycles are not used. It has been shown [171] that this bias is less, probably much less, than one-half of one standard deviation for 400 active cycles when the ratio of the true k_{eff} standard deviation to k_{eff} is 0.0025 at the problem end.

In the MCNP code, the definition of k_{eff} is:

$$k_{\text{eff}} = \frac{\text{fission neutrons in generation } i+1}{\text{fission neutrons in generation } i} = \frac{\rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \nu\sigma_{\text{f}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega}{\int_V \int_0^\infty \int_E \int_\Omega \nabla \cdot J \text{d}V \text{d}t \text{d}E \text{d}\Omega + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega (\sigma_{\text{c}} + \sigma_{\text{f}} + \sigma_{\text{m}}) \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega}, \quad (2.270)$$

where the phase-space variables are t, E, and Ω for time, energy, direction, and implicitly r for position with incremental volume $\text{d}V$ around r. The denominator is the loss rate, which is the sum of leakage, capture (n,0n), fission, and multiplicity (n,xn) terms. By particle balance, the loss rate is also the source rate, which is unity in a criticality calculation. If the number of fission neutrons produced in one generation is equal to the number in the previous generation, then the system is critical. If it is greater, the system is supercritical. If it is less, then the system is subcritical. The multiplicity term is:

$$\begin{aligned} &\rho_{\text{m}} \int_V \int_0^\infty \int_E \int_\Omega \sigma_{\text{m}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega \\ &\quad = \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \sigma_{\text{n,2n}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega - 2\rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \sigma_{\text{n,2n}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega \\ &\qquad + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \sigma_{\text{n,3n}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega - 3\rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \sigma_{\text{n,3n}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega + \cdots . \end{aligned} \quad (2.271)$$

The above definition of k_{eff} comes directly from the time-integrated Boltzmann transport equation (without external sources),

$$\begin{aligned} &\int_V \int_0^\infty \int_E \int_\Omega \nabla \cdot J \text{d}V \text{d}t \text{d}E \text{d}\Omega + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \sigma_{\text{t}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega \\ &\quad = \frac{1}{k_{\text{eff}}} \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \nu\sigma_{\text{f}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \sigma_{\text{s}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega, \end{aligned} \quad (2.272)$$

which may be rewritten to look more like the definition of k_{eff} as

$$\begin{aligned} &\int_V \int_0^\infty \int_E \int_\Omega \nabla \cdot J \text{d}V \text{d}t \text{d}E \text{d}\Omega + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega (\sigma_{\text{c}} + \sigma_{\text{f}} + \sigma_{\text{n,2n}} + \sigma_{\text{n,3n}} + \cdots) \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega \\ &\quad = \frac{1}{k_{\text{eff}}} \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \nu\sigma_{\text{f}} \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega (2\sigma_{\text{n,2n}} + 3\sigma_{\text{n,3n}} + \cdots) \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega. \end{aligned} \quad (2.273)$$

The loss rate is on the left and the production rate is on the right.

The neutron prompt removal lifetime is the average time from the emission of a prompt neutron in fission to the removal of the neutron by some physical process such as escape, capture, or fission. Also, even with the TOTNU card to produce delayed neutrons as well as prompt neutrons (KCODE default), the neutrons are all born at time zero, so the removal lifetimes calculated in the MCNP code are prompt removal lifetimes, even if there are delayed neutrons.

The definition of the prompt removal lifetime [172] is

$$\tau_{\text{r}} = \frac{\int_V \int_0^\infty \int_E \int_\Omega \eta \text{d}V \text{d}t \text{d}E \text{d}\Omega}{\int_V \int_0^\infty \int_E \int_\Omega \nabla \cdot J \text{d}V \text{d}t \text{d}E \text{d}\Omega + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega (\sigma_{\text{c}} + \sigma_{\text{f}} + \sigma_{\text{m}}) \Phi \text{d}V \text{d}t \text{d}E \text{d}\Omega}, \quad (2.274)$$

where η is the population per unit volume per unit energy per unit solid angle. In a multiplying system in which the population is increasing or decreasing on an asymptotic period, the population changes in accordance with

$$\eta = \eta_0 \exp\left[\frac{(k_{\text{eff}} - 1)t}{\tau_{\text{r}}^+}\right], \tag{2.275}$$

where τ_{r}^+ is the adjoint-weighted removal lifetime. The MCNP code calculates the non-adjoint-weighted prompt removal lifetime τ_{r} that can be significantly different in a multiplying system. In a non-multiplying system, $k_{\text{eff}} = 0$ and $\tau_{\text{r}} \rightarrow \tau_{\text{r}}^+$, the population decays as

$$\eta = \eta_0 \exp(-t/\tau_{\text{r}}), \tag{2.276}$$

where the non-adjoint-weighted removal lifetime τ_{r} is also the relaxation time.

Noting that the flux is defined as

$$\Phi = \eta v, \tag{2.277}$$

where v is the speed, the non-adjoint-weighted prompt removal lifetime in the MCNP code, τ_{r}, is defined as

$$\tau_{\text{r}} = \frac{\int_V \int_0^\infty \int_E \int_\Omega \frac{\Phi}{v} \text{d}V\,\text{d}t\,\text{d}E\,\text{d}\Omega}{\int_V \int_0^\infty \int_E \int_\Omega \nabla \cdot J \text{d}V\,\text{d}t\,\text{d}E\,\text{d}\Omega + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega (\sigma_{\text{c}} + \sigma_{\text{f}} + \sigma_{\text{m}})\Phi \text{d}V\,\text{d}t\,\text{d}E\,\text{d}\Omega}. \tag{2.278}$$

The prompt removal lifetime is a fundamental quantity in the nuclear engineering point kinetics equation. It is also useful in nuclear well-logging calculations and other pulsed source problems because it gives the population time-decay constant.

2.8.2.1 Collision Estimators

The collision estimate for k_{eff} for any active cycle is

$$k_{\text{eff}}^{\text{C}} = \frac{1}{N} \sum_i W_i \left[\frac{\sum_j f_j \overline{\nu}_j \sigma_{\text{f}_j}}{\sum_j f_j \sigma_{\text{t}_j}}\right], \tag{2.279}$$

where

i	is summed over all collisions in a cycle where fission is possible;
j	is summed over all nuclides of the material involved in the ith collision;
σ_{t_j}	is the total microscopic cross section;
σ_{f_j}	is the microscopic fission cross section;
$\overline{\nu}_j$	is the average number of prompt or total neutrons produced per fission by the collision nuclide at the incident energy;
f_j	is the atomic fraction for nuclide j;
N	is the nominal source size for cycle; and
W_i	is the weight of particle entering collision.

Because W_i represents the number of neutrons entering the ith collision,

$$W_i \left[\frac{\sum_j f_j \overline{\nu}_j \sigma_{\mathrm{f}_j}}{\sum_j f_j \sigma_{\mathrm{t}_j}} \right] \tag{2.280}$$

is the expected number of neutrons to be produced from all fission processes in the collision. Thus $k_{\mathrm{eff}}^{\mathrm{C}}$ is the mean number of fission neutrons produced per cycle. The collision estimator tends to be best, sometimes only marginally so, in very large systems.

The collision estimate of the prompt removal lifetime for any active cycle is the average time required for a fission source neutron to be removed from the system by either escape, capture (n,0n), or fission.

$$\tau_{\mathrm{r}}^{\mathrm{C}} = \frac{\sum W_{\mathrm{e}} T_{\mathrm{e}} + \sum (W_{\mathrm{c}} + W_{\mathrm{f}}) T_{\mathrm{x}}}{\sum W_{\mathrm{e}} + \sum (W_{\mathrm{c}} + W_{\mathrm{f}})}, \tag{2.281}$$

where T_{e} and T_{x} are the times from the birth of the neutron until escape or collision. W_{e} is the weight lost at each escape. $W_{\mathrm{c}} + W_{\mathrm{f}}$ is the weight lost to (n,0n) and fission at each collision,

$$W_{\mathrm{c}} + W_{\mathrm{f}} = W_i \frac{\sum_j f_j \left(\sigma_{\mathrm{c}_j} + \sigma_{\mathrm{f}_j} \right)}{\sum_j f_j \sigma_{\mathrm{t}_j}}, \tag{2.282}$$

where σ_{c_j} is the microscopic capture (n,0n) cross section, and W_i is the weight entering the collision.

2.8.2.2 Absorption Estimators

The absorption estimator for k_{eff} for any active cycle is made when a neutron interacts with a fissionable nuclide. The estimator differs for analog and implicit absorption. For analog absorption,

$$k_{\mathrm{eff}}^{\mathrm{A}} = \frac{1}{N} \sum_i W_i \overline{\nu}_j \frac{\sigma_{\mathrm{f}_j}}{\sigma_{\mathrm{c}_j} + \sigma_{\mathrm{f}_j}}, \tag{2.283}$$

where i is summed over each analog absorption event in the jth nuclide. Note that in analog absorption, the weight is the same both before and after the collision. Because analog absorption includes fission in criticality calculations, the frequency of analog absorption at each collision with nuclide j is $\left(\sigma_{\mathrm{c}_j} + \sigma_{\mathrm{f}_j} \right) / \sigma_{\mathrm{t}_j}$. The analog absorption k_{eff} estimate is very similar to the collision estimator of k_{eff} except that only the jth absorbing nuclide, as sampled in the collision, is used rather than averaging over all nuclides.

For implicit absorption, the following is accumulated:

$$k_{\mathrm{eff}}^{\mathrm{A}} = \frac{1}{N} \sum_i W_i' \overline{\nu}_j \frac{\sigma_{\mathrm{f}_j}}{\sigma_{\mathrm{c}_j} + \sigma_{\mathrm{f}_j}}, \tag{2.284}$$

where i is summed over all collisions in which fission is possible and

$$W_i' = W_i \frac{\sigma_{\mathrm{c}_j} + \sigma_{\mathrm{f}_j}}{\sigma_{\mathrm{t}_j}} \tag{2.285}$$

is the weight absorbed in the implicit absorption. The difference between the implicit absorption estimator $k_{\mathrm{eff}}^{\mathrm{A}}$ and the collision estimator $k_{\mathrm{eff}}^{\mathrm{C}}$ is that only the nuclide involved in the collision is used for the absorption k_{eff} estimate rather than an average of all nuclides in the material for the collision k_{eff} estimator.

The absorption estimator with analog absorption is likely to produce the smallest statistical uncertainty of the three estimators for systems where the ratio $\overline{\nu}_j \sigma_{\mathrm{f}_j} / \left(\sigma_{\mathrm{c}_j} + \sigma_{\mathrm{f}_j} \right)$ is nearly constant. Such would be the case for a thermal system with a dominant fissile nuclide such that the 1/velocity cross-section variation would tend to cancel.

The absorption estimate differs from the collision estimate in that the collision estimate is based upon the expected value at each collision, whereas the absorption estimate is based upon the events actually sampled at a collision. Thus all collisions will contribute to the collision estimate of $k_{\text{eff}}^{\text{C}}$ and $\tau_{\text{r}}^{\text{C}}$ by the probability of fission (or capture for $\tau_{\text{r}}^{\text{C}}$) in the material. Contributions to the absorption estimator will only occur if an actual fission (or capture for $\tau_{\text{r}}^{\text{A}}$) event occurs for the sampled nuclide in the case of analog absorption. For implicit absorption, the contribution to the absorption estimate will only be made for the nuclide sampled.

The absorption estimate of the prompt removal lifetime for any active cycle is again the average time required for a fission source neutron to be removed from the system by either escape, capture (n,0n), or fission.

For implicit absorption,

$$\tau_{\text{r}}^{\text{A}} = \frac{\sum W_{\text{e}} T_{\text{e}} + \sum (W_{\text{c}} + W_{\text{f}}) T_{\text{x}}}{\sum W_{\text{e}} + \sum W_{\text{c}} + \sum W_{\text{f}}}, \tag{2.286}$$

where

$$W_{\text{c}} + W_{\text{f}} = W_i \frac{\sigma_{\text{c}_j} + \sigma_{\text{f}_j}}{\sigma_{\text{t}_j}}. \tag{2.287}$$

For analog absorption,

$$\tau_{\text{r}}^{\text{A}} = \frac{\sum W_e T_{\text{e}} + \sum W_c T_{\text{c}} + \sum W_{\text{f}} T_{\text{f}}}{\sum W_{\text{e}} + \sum W_{\text{c}} + \sum W_{\text{f}}}, \tag{2.288}$$

where T_{e}, T_{c}, T_{f}, and T_{x} are the times from the birth of the neutron until escape, capture (n,0n), fission, or collision. W_{e} is the weight lost at each escape. W_{c} and W_{f} are the weights lost to capture (n,0n) and fission at each capture (n,0n) or fission event with the nuclide sampled for the collision.

2.8.2.3 Track-length Estimators

The track length estimator of k_{eff} is accumulated every time the neutron traverses a distance d in a fissionable material cell:

$$k_{\text{eff}}^{\text{TL}} = \frac{1}{N} \sum_i W_i \rho d \sum_j f_j \overline{\nu}_j \sigma_{\text{f}_j}, \tag{2.289}$$

where

i	is summed over all neutron trajectories;
ρ	is the atomic density in the cell; and
d	is the trajectory track length from the last event.

Because $\rho d \sum_j f_j \overline{\nu}_j \sigma_{\text{f}_j}$ is the expected number of fission neutrons produced along trajectory d, $k_{\text{eff}}^{\text{TL}}$ is a third estimate of the mean number of fission neutrons produced in a cycle per nominal fission source neutron.

The track length estimator tends to display the lowest variance for optically thin fuel cells (for example, plates) and fast systems where large cross-section variations because of resonances may cause high variances in the other two estimators.

The track length estimator for the prompt removal lifetime for each cycle is accumulated every time the neutron traverses a distance d in any material in any cell:

$$\tau_{\text{r}}^{\text{TL}} = \frac{\sum_i W_i d/v}{W_s} \tag{2.290}$$

where W_s is the source weight summed over all histories in the cycle and v is the velocity. Note that d/v is the time span of the track. Note further that:

$$\sum_i W_i d/v = \int_V \int_0^\infty \int_E \int_\Omega \frac{\Phi}{v} \mathrm{d}V \mathrm{d}t \mathrm{d}E \mathrm{d}\Omega \tag{2.291}$$

and in criticality problems:

$$\begin{aligned} W_s &= \frac{1}{k_{\text{eff}}} \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega \nu\sigma_{\text{f}} \Phi \mathrm{d}V \mathrm{d}t \mathrm{d}E \mathrm{d}\Omega \\ &= \int_V \int_0^\infty \int_E \int_\Omega \nabla \cdot J \mathrm{d}V \mathrm{d}t \mathrm{d}E \mathrm{d}\Omega + \rho_{\text{a}} \int_V \int_0^\infty \int_E \int_\Omega (\sigma_{\text{c}} + \sigma_{\text{f}} + \sigma_{\text{m}}) \Phi \mathrm{d}V \mathrm{d}t \mathrm{d}E \mathrm{d}\Omega \end{aligned} \tag{2.292}$$

These relationships show how $\tau_{\text{r}}^{\text{TL}}$ is related to the definition of τ_{r} in Eq. (2.278).

2.8.2.4 Other Lifetime Estimators

In addition to the collision, absorption, and track length estimators of the prompt removal lifetime τ_{r}, The MCNP code provides the escape, capture (n,0n), and fission prompt lifespans and lifetimes for all KCODE problems having a sufficient number of settle cycles. Further, the "average time of" printed in the problem summary table is related to the lifespans, and track-length estimates of many lifetimes can be computed using the $1/v$ tally multiplier option on the FM card for track-length tallies.

In KCODE problems, the MCNP code calculates the lifespan of escape l_{e}, capture (n,0n) l_{c}, fission l_{f}, and removal l_{r} as

$$l_{\text{e}} = \frac{\sum W_{\text{e}} T_{\text{e}}}{\sum W_{\text{e}}}, \tag{2.293a}$$

$$l_{\text{c}} = \frac{\sum W_{\text{c}} T_{\text{c}}}{\sum W_{\text{c}}}, \tag{2.293b}$$

$$l_{\text{f}} = \frac{\sum W_{\text{f}} T_{\text{f}}}{\sum W_{\text{f}}}, \tag{2.293c}$$

$$l_{\text{r}} = \frac{\sum W_{\text{e}} T_{\text{e}} + \sum W_{\text{c}} T_{\text{c}} + \sum W_{\text{f}} T_{\text{f}}}{\sum W_{\text{e}} + \sum W_{\text{c}} + \sum W_{\text{f}}}. \tag{2.293d}$$

These sums are taken over all the active histories in the calculation. If KC8 = 0 on the KCODE card, then the sums are over both active and inactive cycle histories, but KC8 = 1, the default, is assumed for the remainder of this discussion. The capture (n,0n) and fission contributions are accumulated at each collision with a nuclide, so these are absorption estimates. Thus,

$$l_{\text{r}} \approx \tau_{\text{r}}^{\text{A}}. \tag{2.294}$$

The difference is that $\tau_{\text{r}}^{\text{A}}$ is the average of the $\tau_{\text{r}}^{\text{A}}$ for each cycle and l_{r} is the average over all histories. $l_{\text{r}} = \tau_{\text{r}}^{\text{A}}$ if there is precisely one active cycle, but then neither $\tau_{\text{r}}^{\text{A}}$ nor l_{r} is printed out because there are too few cycles. The cycle average $\tau_{\text{r}}^{\text{A}}$ does not precisely equal the history-average l_{r} because they are ratios.

l_{e} and l_{c} are the "average time to" escape and capture (n,0n) that is printed in the problem summary table for all neutron and photon problems.

$(1/N)\sum W_{\text{e}}$, $(1/N)\sum W_{\text{c}}$, and $(1/N)\sum W_{\text{f}}$ are the weight lost to escape, capture (n,0n), and fission in the problem summary table.

The "fractions" F_x printed out below the lifespan in the KCODE summary table are, for $x = \mathrm{e}, \mathrm{c}, \mathrm{f}, \mathrm{r}$,

$$F_x = \frac{W_x}{\sum W_\mathrm{e} + \sum W_\mathrm{c} + \sum W_\mathrm{f}}. \tag{2.295}$$

The prompt lifetimes [172] for the various reactions τ_x are then

$$\tau_x = \frac{\tau_\mathrm{r}}{F_x} = \frac{\int_V \int_0^\infty \frac{\Phi}{v} \mathrm{d}V\mathrm{d}t}{\rho_\mathrm{a} \int_V \int_0^\infty \sigma_x \Phi \mathrm{d}V\mathrm{d}t}. \tag{2.296}$$

Both $\tau_\mathrm{r}^\mathrm{A}$ and the covariance-weighted combined estimator $\tau_\mathrm{r}^{(\mathrm{C/A/T})}$ are used. Note again that the slight differences between similar quantities are because l_x and F_x are averaged over all active histories whereas $\tau_\mathrm{r}^\mathrm{A}$ and $\tau_\mathrm{r}^{(\mathrm{C/A/T})}$ are averaged within each active cycle, and then the final values are the averages of the cycle values, i.e., history averages vs. batch averages.

The prompt removal lifetime can also be calculated using the F4 track-length tally with the $1/v$ multiplier option on the FM card and using the volume divided by the average source weight W_s as the multiplicative constant. The standard track length tally is then converted from

$$F4 = \int \Phi \mathrm{d}t \tag{2.297}$$

to

$$F4 = \frac{V}{W_s} \int \frac{\Phi}{v} \mathrm{d}t. \tag{2.298}$$

Remember to multiply by volume, either by setting the FM card constant to the volume or overriding the F4 volume divide by using segment divisors of unity on the SD card. W_s should be unity for KCODE calculations. The only difference between $\tau_\mathrm{r}^\mathrm{TL}$ and the modified F4 tally will be any variations from unity in W_s and the error estimation, which will be batch-averaged for $\tau_\mathrm{r}^\mathrm{TL}$ and history-averaged for the F4 tally.

Lifetimes for all other processes also can be estimated by using the FM multiplier to calculate reaction rates as well (the numerator and denominator are separate tallies that must be divided by the user—see the examples in Chapter 10):

$$\tau_x^\mathrm{TL} = \frac{(1/v \text{ multiplier})}{\text{reaction rate multiplier}} = \frac{\int_V \int_0^\infty \frac{\Phi}{v} \mathrm{d}V\mathrm{d}t}{\rho_\mathrm{a} \int_V \int_0^\infty \sigma_x \Phi \mathrm{d}V\mathrm{d}t}. \tag{2.299}$$

Note that the lifetimes are inversely additive as

$$\frac{1}{\tau_\mathrm{r}} = \frac{1}{\tau_\mathrm{e}} + \frac{1}{\tau_\mathrm{c}} + \frac{1}{\tau_\mathrm{f}}. \tag{2.300}$$

2.8.2.5 Combined k_eff and τ_r Estimators

The MCNP code provides a number of combined k_eff and τ_r estimators that are combinations of the three individual k_eff and τ_r estimators using two at a time or all three. The combined k_eff and τ_r values are computed by using a maximum likelihood estimate, as outlined by Halperin [173] and discussed further by Urbatsch [165]. This technique, which is a generalization of the inverse variance weighting for uncorrelated estimators, produces the maximum likelihood estimate for the combined average k_eff and τ_r, which, for multivariate normality, is the almost-minimum variance estimate. It is "almost" because the covariance matrix is not known exactly and must be estimated. The three-combined k_eff and τ_r estimators are the best final estimates from an MCNP calculation [165].

This method of combining estimators can exhibit one feature that is disconcerting: sometimes (usually with highly positively correlated estimators) the combined estimate will lie outside the interval defined by the

two or three individual average estimates. Statisticians at Los Alamos have shown [165] that this is the best estimate to use for a final k_{eff} and τ_{r} value. Reference [165] shows the results of one study of 500 samples from three highly positively correlated normal distributions, all with a mean of zero. In 319 samples, all three estimators fell on the same side of the expected value. This type of behavior occurs with high positive correlation because if one estimator is above or below the expected value, the others have a good probability of being on the same side of the expected value. The advantage of the three-combined estimator is that the Halperin algorithm correctly predicts that the true value will lie outside of the range.

2.8.2.6 Error Estimation and Estimator Combination

After the first I_c inactive cycles, during which the fission source spatial distribution is allowed to come into spatial equilibrium, the MCNP code begins to accumulate the estimates of k_{eff} and τ_{r} with those estimates from previous active (after the inactive) cycles. The relative error R of each quantity is estimated in the usual way as

$$R = \frac{1}{\overline{x}}\sqrt{\frac{\overline{x^2} - \overline{x}^2}{M-1}}, \tag{2.301}$$

where M is the number of active cycles,

$$\overline{x} = \frac{1}{M}\sum_{m=1}^{M} x_m, \tag{2.302a}$$

$$\overline{x^2} = \frac{1}{M}\sum_{m=1}^{M} x_m^2, \tag{2.302b}$$

and x_m is a quantity such as k_{eff} from active cycle m. This assumes that the cycle-to-cycle estimates of each k_{eff} are uncorrelated. This assumption generally is good for k_{eff}, but not for the eigenfunction (fluxes) of optically large systems [174].

The MCNP code also combines the three estimators in all possible ways and determines the covariance and correlations. The simple average of two estimators is defined as $x^{ij} = (1/2)\left(x^i + x^j\right)$, where, for example, x^i may be the collision estimator $k_{\text{eff}}^{\text{C}}$ and x^j may be the absorption estimator $k_{\text{eff}}^{\text{A}}$.

The "combined average" of two estimators is weighted by the covariances as

$$x^{ij} = x^i - \frac{\left(x^i - x^j\right)(C_{ii} - C_{ij})}{(C_{ii} + C_{jj} - 2C_{ij})} = \frac{(C_{jj} - C_{ij})x^i + (C_{ii} - C_{ij})x^j}{(C_{ii} + C_{jj} - 2C_{ij})}, \tag{2.303}$$

where the covariance C_{ij} is

$$C_{ij} = \frac{1}{M}\sum_{m=1}^{M} x_m^i x_m^j - \left(\frac{1}{M}\sum_{m=1}^{M} x_m^i\right)\left(\frac{1}{M}\sum_{m=1}^{M} x_m^j\right). \tag{2.304}$$

Note that $C_{ii} = \overline{x^2} - \overline{x}^2$ for estimator i.

The "correlation" between two estimators is a function of their covariances and is given by

$$\text{correlation} = \frac{C_{ij}}{\sqrt{|C_{ii}C_{jj}|}}. \tag{2.305}$$

The correlation will be between positive one (perfect positive correlation) and minus one (perfect anti or negative correlation). If the correlation is one, no new information has been gained by the second estimator. If the correlation is zero, the two estimators appear statistically independent and the combined estimated

standard deviation should be significantly less than either. If the correlation is negative one, even more information is available because the second estimator will tend to be low, relative to the expected value, when the first estimator is high and vice versa. Even larger improvements in the combined standard deviation should occur.

The combined average estimator (k_{eff} or τ_{r}) and the estimated standard deviation of all three estimators are based on the method of Halperin [173] and is much more complicated than the two-combination case. The improvements to the standard deviation of the three-combined estimator will depend on the magnitude and sign of the correlations as discussed above. The details and analysis of this method are given in [165].

For many problems, all three estimators are positively correlated. The correlation will depend on what variance reduction (for example, implicit or analog capture) is used. Occasionally, the absorption estimator may be only weakly correlated with either the collision or track length estimator. It is possible for the absorption estimator to be significantly anticorrelated with the other two estimators for some fast reactor compositions and large thermal systems. Except in the most heterogeneous systems, the collision and track length estimators are likely to be strongly positively correlated.

There may be a negative bias [171] in the estimated standard deviation of k_{eff} for systems where the locations of fission sites in one generation are correlated with the locations of fission sites in successive generations. The statistical methods used in the MCNP code for estimating standard deviations in k_{eff} calculations do not account for the effects of intergenerational correlation, leading to underprediction of standard deviations. These systems are typically large with small neutron leakage. The magnitude of this effect can be estimated by batching the cycle k_{eff} values in batch sizes much greater than one cycle [171], which the MCNP code provides automatically. For problems where there is a reason to suspect the results, a more accurate calculation of this effect can be done by making several independent calculations of the same problem (using different random number sequences) and observing the variance of the collection of independent k_{eff} values. The larger the number of independent calculations that can be made, the better the distribution of k_{eff} values can be assessed.

2.8.2.7 Creating and Interpreting k_{eff} Confidence Intervals

The result of a Monte Carlo criticality calculation (or any other type of Monte Carlo calculation) is a confidence interval. For criticality, this means that the result is not just k_{eff}, but k_{eff} plus and minus some number of estimated standard deviations to form a confidence interval (based on the Central Limit Theorem) in which the true answer is expected to lie a certain fraction of the time. The number of standard deviations used (for example, from a Student's t Table) determines the fraction of the time that the confidence interval will include the true answer, for a selected confidence level. For example, a valid 99% confidence interval should include the true result 99% of the time. There is always some probability (in this example, 1%) that the true result will lie outside of the confidence interval. To reduce this probability to an acceptable level, either the confidence interval must be increased according to the desired Student's t percentile, or more histories need to be run to get a smaller estimated standard deviation.

The MCNP code uses three different estimators for k_{eff}. The advantages of each estimator vary with the problem: no one estimator will be the best for all problems. All estimators and their estimated standard deviations are valid under the assumption that they are unbiased and consistent, therefore representative of the true parameters of the population. This statement has been validated empirically [165] for all MCNP estimators for small dominance ratios. The batched k_{eff} results table should be used to estimate if the calculated batch-size-of-one k_{eff} standard deviation appears to be adequate.

The confidence interval based on the three-statistically-combined k_{eff} estimator is the recommended result to use for all final k_{eff} confidence interval quotations because all of the available information has been used in the final result. This estimator often has a lower estimated standard deviation than any of the three individual estimators and therefore provides the smallest valid confidence

interval as well. The final estimated k_{eff} value, estimated standard deviation, and the estimated 68%, 95%, and 99% confidence intervals (using the correct number of degrees of freedom) are presented in the box on the k_{eff} results summary page of the output. If other confidence intervals are wanted, they can be formed from the estimated standard deviation of k_{eff}. At least 30 active cycles need to be run for the final k_{eff} results box to appear. Thirty cycles are required so that there are enough degrees of freedom to form confidence intervals using the well-known estimated standard deviation multipliers. When constructing a confidence interval using any single k_{eff} estimator, its standard deviation, and a Student's t Table, there are $I_t - I_c - 1$ degrees of freedom. For the two- and three-combined k_{eff} estimators, there are $I_t - I_c - 2$ and $I_t - I_c - 3$ degrees of freedom, respectively.

All of the k_{eff} estimators and combinations by two or three are provided in the MCNP code so that the user can make an alternate choice of confidence interval if desired. Based on statistical studies, using the individual k_{eff} estimator with the smallest estimated standard deviation is not recommended. Its use can lead to confidence intervals that do not include the true result the correct fraction of the time [165]. The studies have shown that the standard deviation of the three-combined k_{eff} estimator provides the correct coverage rates, assuming that the estimated standard deviations in the individual k_{eff} estimators are accurate. This accuracy can be verified by checking the batched k_{eff} results table. When significant anti-correlations occur among the estimators, the resultant much smaller estimated standard deviation of the three-combined average has been verified [165] by analyzing a number of independent criticality calculations.

2.8.2.8 Analysis to Assess the Validity of a Criticality Calculation

The two most important requirements for producing a valid criticality calculation for a specified geometry are sampling all of the fissionable material well and ensuring that the fundamental spatial mode was achieved before and maintained during the active k_{eff} cycles. The MCNP code has checks to assess the fulfillment of both of these conditions.

The MCNP code verifies that at least one fission source point was generated in each cell containing fissionable material. A WARNING message is printed on the k_{eff} results summary page that includes a list of cells that did not have any particles entering, and/or no collisions, and/or no fission source points. For repeated structure geometries, a source point in any one cell that is repeated will satisfy this test. For example, assume a problem with a cylinder and a cube that are both filled with the same universe, namely a sphere of uranium and the space outside the sphere. If a source point is placed in the sphere inside the cylinder but not in the sphere inside the cube, the test will be satisfied.

One basic assumption that is made for a good criticality calculation is that the normal spatial mode for the fission source has been achieved after I_c cycles were skipped. The MCNP code attempts to assess this condition in several ways. The estimated combined k_{eff} and its estimated standard deviation for the first and second active cycle halves of the problem are compared. A WARNING message is issued if either the difference of the two values of combined col/abs/track-length k_{eff} does not appear to be zero or the ratio of the larger-to-the-smaller estimated standard deviations of the two col/abs/ track-length k_{eff} is larger than expected. Failure of either or both checks implies that the two active halves of the problem do not appear to be the same and the output from the calculation should be inspected carefully.

The MCNP code checks to determine which number of cycles skipped produces the minimum estimated standard deviation for the combined k_{eff} estimator. If this number is larger than I_c, it may indicate that not enough inactive cycles were skipped. The table of combined k_{eff}-by-number-of-cycles skipped should be examined to determine if enough inactive cycles were skipped.

It is assumed that N is large enough so that the collection of active cycle k_{eff} estimates for each estimator will be normally distributed if the fundamental spatial mode has been achieved in I_c cycles and maintained for the rest of the calculation. To test this assumption, the MCNP code performs normality checks [175, 176] on each of the three k_{eff} estimator cycle data at the 95% and 99% confidence levels. A WARNING message is

issued if an individual k_{eff} data set does not appear to be normally distributed at the 99% confidence level. This condition will happen to good data about 1% of the time. Unless there is a high positive correlation among the three estimators, it is expected to be rare that all three k_{eff} estimators will not appear normally distributed at the 99% confidence level when the normal spatial mode has been achieved and maintained. When the condition that all three sets of k_{eff} estimators do not appear to be normal at the 99% confidence level occurs, the box with the final k_{eff} will not be printed. The final confidence interval results are available elsewhere in the output. Examine the calculation carefully to see if the normal mode was achieved before the active cycles began. The normality checks are also made for the batched-k_{eff} and k_{eff}-by-cycles-skipped tables so that normality behavior can be studied by batch size and I_c.

These normality checks test the assumption that the individual cycle k_{eff} values behave in the assumed way. Even if the underlying individual cycle k_{eff} values are not normally distributed, the three average k_{eff} values and the combined k_{eff} estimator will be normally distributed if the conditions required by the Central Limit Theorem are met for the average. If required, this assumption can be tested by making several independent calculations to verify empirically that the collection of the average k_{eff} values appear to be normally distributed with the same population variance as estimated by the MCNP code.

The MCNP code tests for a monotonic trend of the three-combined k_{eff} estimator over the last ten active cycles. This type of behavior is not expected in a well-converged solution for k_{eff} and could indicate a problem with achieving or maintaining the normal spatial mode. A WARNING message is printed if such a monotonic trend is observed.

To assist users in assessing the convergence of the fission source spatial distribution, the MCNP code computes a quantity called the Shannon entropy of the fission source distribution, H_{src} [169, 170]. The Shannon entropy is a well-known concept from information theory and provides a single number for each cycle to help characterize convergence of the fission source distribution. It has been found that the Shannon entropy converges to a single steady-state value as the source distribution approaches stationarity. Line plots of Shannon entropy vs. cycle are easier to interpret and assess than are 2-D or 3-D plots of the source distribution vs. cycle.

To compute H_{src}, it is necessary to superimpose a 3-D grid on a problem encompassing all of the fissionable regions, and then to tally the number of fission sites in a cycle that fall into each of the grid boxes. These tallies may then be used to form a discretized estimate of the source distribution, $\{P_J, J = 1, N_{\text{s}}\}$, where N_{s} is the number of grid boxes in the superimposed mesh, and P_J is (the number of source sites in Jth grid box)/(total number of source sites). Then, the Shannon entropy of the discretized source distribution for that cycle is given by

$$H_{\text{src}} = -\sum_{J=1}^{N_{\text{s}}} P_J \cdot \ln_2(P_J). \tag{2.306}$$

H_{src} varies between 0 for a point distribution to $\ln_2(N_{\text{s}})$ for a uniform distribution. Also note that as P_J approaches 0, $P_J \ln_2(P_J)$ approaches 0. The MCNP code prints H_{src} for each cycle of a KCODE calculation. Plots of H_{src} vs. cycle can also be obtained during or after a calculation, using the z option and requesting plots for "kcode 6." The user may specify a particular grid to use in determining H_{src} using the HSRC input card. If the HSRC card is provided, users should specify a small number of grid boxes (e.g., 5–10 in each of the x, y, z directions), chosen according to the symmetry of the problem and layout of the fuel regions. If the HSRC card is not provided, the MCNP code will automatically determine a grid that encloses all of the fission sites for the cycle. The number of grid boxes will be determined by dividing the number of histories per cycle by 20, and then finding the nearest integer for each direction that will produce this number of equal-sized grid boxes, although not fewer than $4 \times 4 \times 4$ will be used.

Upon completion of the problem, the MCNP code will compute the average value of H_{src} for the last half of the active cycles, as well as the (estimated population) standard deviation. The MCNP code will then report the first cycle found (active or inactive) where H_{src} falls within one standard deviation of its average for the last half of the cycles, along with a recommendation that at least that many cycles should be inactive.

Plots of H_{src} vs. cycle should be examined to further verify that the number of inactive cycles is adequate for fission source convergence.

⚠ Caution

When running criticality calculations with the MCNP code, it is essential that users examine the convergence of both k_{eff} and the fission source distribution (using Shannon entropy). If either k_{eff} or the fission source distribution is not converged prior to starting the active cycles, then results from the calculations will not be correct.

2.8.2.9 Normalization of Standard Tallies in a Criticality Calculation

Track length fluxes, surface currents, surface fluxes, heating and detectors—all the standard MCNP tallies—can be made during a criticality calculation. The tallies are for one fission neutron generation. Biases may exist in these criticality results, but appear to be smaller than statistical uncertainties [171]. These tallied quantities are accumulated only after the I_c inactive cycles are finished. The tally normalization is per active source weight w, where $w = N \cdot (I_t - I_c)$, and N is the nominal source size (from the KCODE card); It is the total number of cycles in the problem; and I_c is the number of inactive cycles (from KCODE card). The number w is appropriately adjusted if the last cycle is only partially completed. If the tally normalization flag (on the KCODE card) is turned on, the tally normalization is the actual number of starting particles during the active cycles rather than the nominal weight above. Bear in mind, however, that the source particle weights are all set to $W = N/M$ so that the source normalization is based upon the nominal source size N for each cycle.

An MCNP tally in a criticality calculation is for one fission neutron being born in the system at the start of a cycle. The tally results must be scaled either by the total number of neutrons in a burst or by the neutron birth rate to produce, respectively, either the total result or the result per unit time of the source. The scaling factor is entered on the FM card.

The statistical errors that are calculated for the tallies assume that all the neutron histories are independent. They are not independent because of the cycle-to-cycle correlations that become more significant for large or loosely coupled systems. For some very large systems, the estimated standard deviation for a tally that involves only a portion of the problem has been observed to be underestimated by a factor of five or more [pages 42–44 of 174]. This value also is a function of the size of the tally region. In the [174] slab reactor example, the entire problem (that is, k_{eff}) standard deviation was not underestimated at all. An MCNP study [177] of the FFTF fast reactor indicates that 90% coverage rates for flux tallies are good, but that 2 out of 300 tallies were beyond four estimated standard deviations. Independent runs can be made to study the real eigenfunction distribution (that is, tallies) and the estimated standard deviations for difficult criticality calculations. This method is the only way to determine accurately these confidence intervals for large or loosely coupled problems where intergeneration correlation is significant.

2.8.2.10 Neutron Tallies and the MCNP Net Multiplication Factor

The MCNP net multiplication factor M printed out on the problem summary page differs from the k_{eff} from the criticality code. We will examine a simple model to illustrate the approximate relationship between these quantities and compare the tallies between standard and criticality calculations.

Assume we run a standard MCNP calculation using a fixed neutron source distribution identical in space and energy to the source distribution obtained from the solution of an eigenvalue problem with $k_{\text{eff}} < 1$. Each generation will have the same space and energy distribution as the source. The contribution to an estimate of any quantity from one generation is reduced by a factor of k_{eff} from the contribution in the

preceding generation. The estimate E_k of a tally quantity obtained in a criticality eigenvalue calculation is the contribution for one generation produced by a unit source of fission neutrons. An estimate for a standard MCNP fixed source calculation, E_s, is the sum of contributions for all generations starting from a unit source,

$$E_s = E_k + k_{\text{eff}} E_k + k_{\text{eff}}^2 E_k + k_{\text{eff}}^3 E_k + \cdots = \frac{E_k}{1 - k_{\text{eff}}}. \tag{2.307}$$

Note that $1/(1 - k_{\text{eff}})$ is the true system multiplication, often called the subcritical multiplication factor. The above result depends on our assumptions about the unit fission source used in the standard MCNP run. Usually, E_s will vary considerably from the above result, depending on the difference between the fixed source and the eigenmode source generated in the eigenvalue problem. E_s will be a fairly good estimate if the fixed source is a distributed source roughly approximating the eigenmode source. Tallies from a criticality calculation are appropriate only for a critical system and the tally results can be scaled to a desired fission neutron source (power) level or total neutron pulse strength.

In a fixed-source MCNP problem, the net multiplication M is defined to be unity plus the gain G_{f} in neutrons from fission plus the gain G_x from nonfission multiplicative reactions. Using neutron weight balance (creation equals loss),

$$M = 1 + G_{\text{f}} + G_x = W_{\text{e}} + W_{\text{c}}, \tag{2.308}$$

where W_{e} is the weight of neutrons escaped per source neutron and W_{c} is the weight of neutrons captured per source neutron. In a criticality calculation, fission is treated as an absorptive process; the corresponding relationship for the net multiplication is then

$$M^o = 1 + G_x^o = W_{\text{e}}^o + W_{\text{c}}^o + W_{\text{f}}^o, \tag{2.309}$$

where the superscript o designates results from the criticality calculation and W_{f}^o is the weight of neutrons causing fission per source neutron. Because k_{eff} is the number of fission neutrons produced in a generation per source neutron, we can also write

$$k_{\text{eff}} = \overline{\nu} W_{\text{f}}^o, \tag{2.310}$$

where $\overline{\nu}$ is the average number of neutrons emitted per fission for the entire problem. Making the same assumptions as above for the fixed source used in the standard MCNP calculation and using Eqs. (2.307), (2.308), and (2.309), we obtain

$$M = W_{\text{e}} + W_{\text{c}} = \frac{W_{\text{e}}^o + W_{\text{c}}^o}{1 - k_{\text{eff}}} = \frac{M^o - W_{\text{f}}^o}{1 - k_{\text{eff}}}, \tag{2.311}$$

or, by using (2.309) and (2.310),

$$M = \frac{M^o - \frac{k_{\text{eff}}}{\overline{\nu}}}{1 - k_{\text{eff}}} = \frac{1 - \frac{k_{\text{eff}}}{\overline{\nu}} + G_x^o}{1 - k_{\text{eff}}}. \tag{2.312}$$

Often, the nonfission multiplicative reactions $G_x^o \ll 1$. This implies that k_{eff} can be **approximated** by $k_{\text{eff}}^{\text{FS}}$ (from an appropriate fixed source calculation) as

$$k_{\text{eff}} \approx k_{\text{eff}}^{\text{FS}} = \frac{M - 1}{M - \frac{1}{\overline{\nu}}} \tag{2.313}$$

when the two fission neutron source distributions are nearly the same. The average value of $\overline{\nu}$ in a problem can be calculated by dividing the fission neutrons gained by the fission neutrons lost as given in the totals of the neutron weight balance for physical events. Note, however, that the above estimate is subject to the same limitations as described in Eq. (2.307).

2.8.3 Recommendations for Making a Good Criticality Calculation

2.8.3.1 Problem Setup

As with any calculation, the geometry must be adequately and correctly specified to represent the true physical situation. Plot the geometry and check cells, materials, and masses for correctness. Specify the

appropriate nuclear data, including $S(\alpha,\beta)$ thermal data, at the correct material temperatures. Ensure that initial fission source points exist in every cell that contains fissionable material. Try running short problems with both analog and implicit capture (see the PHYS:N card) to improve the figure of merit for the combined k_{eff} and any tallies being made. Follow the tips for good calculations listed at the end of Chapter 1.

2.8.3.2 Number of Neutrons per Cycle and Number of Cycles

Criticality calculations can suffer from two potential problems. The first is the failure to sufficiently converge the spatial distribution of the fission source from its initial guess to a distribution fluctuating around the fundamental eigenmode solution. It is recommended that the user make an initial calculation with a relatively small number of source particles per generation (perhaps 500 or 1000) and generously allow a large enough number of cycles so that the eigenvalue appears to be fluctuating about a constant value. The user should examine the results and continue the calculation if any trends in the eigenvalue are noticeable. The SRCTP file from the last k_{eff} cycle of the initial calculation can then be used as the source for the final production calculation to be made with a larger number of histories per cycle.

This convergence procedure can be extended for very slowly convergent problems—typically large, thermal, low-leakage systems, where a convergence calculation might be made with 500 or 1000 histories per cycle. Then a second convergence calculation would be made with 1000 histories per cycle, using the SRCTP file from the first run as an initial fission source guess. If the results from the second calculation appear satisfactory, then a final calculation might be made using 5000 or 10000 particles per cycle with the SRCTP file from the second calculation as an initial fission source guess. In the final calculation, only a few cycles should need to be skipped. The bottom line is this: skip enough cycles so that the fundamental spatial mode is achieved.

The second potential problem arises from the fact that the criticality algorithm produces a very small negative bias in the estimated eigenvalue. The bias depends upon $1/N$, where N is the number of source particles per generation. Thus, it is desirable to make N as large as possible. Any value of $N > 500$ should be sufficient to reduce the bias to a small level. The eigenvalue bias Δk_{eff} has been shown [171] to be

$$-\Delta k_{\text{eff}} = \frac{I_t - I_c}{2k_{\text{eff}}}\left(\sigma^2_{k_{\text{eff}}} - \sigma^2_{\text{approx}}\right), \tag{2.314}$$

where

$\sigma_{k_{\text{eff}}}$	is the true standard deviation for the final k_{eff},
σ_{approx}	is the approximate standard deviation computed assuming the individual k_{eff} values are statistically independent, and
$\sigma^2_{k_{\text{eff}}} > \sigma^2_{\text{approx}}$.	

The standard deviations are computed at the end of the problem. Because the σ^2s decrease as $1/(I_t - I_c)$, Δk_{eff} is independent of the number of active cycles. Recall that Δk_{eff} is proportional to $1/N$, the number of neutrons per k_{eff} cycle.

Eq. (2.314) can be written [171] as the following inequality:

$$\frac{|\Delta k_{\text{eff}}|}{\sigma_{k_{\text{eff}}}} < \frac{(I_t - I_c)\sigma_{k_{\text{eff}}}}{2k_{\text{eff}}}. \tag{2.315}$$

This inequality is useful for determining an upper limit to the number of active cycles that should be used for a calculation without having Δk_{eff} dominate $\sigma_{k_{\text{eff}}}$. If $\sigma_{k_{\text{eff}}}/k_{\text{eff}}$ is 0.0010, which is a reasonable value for

criticality calculations, and $I_t - I_c$ is 1000, then $k_{\text{eff}}/\sigma_{k_{\text{eff}}} < 0.5$ and Δk_{eff} will not dominate the k_{eff} confidence interval. If $\sigma_{k_{\text{eff}}}$ is reasonably well approximated by the MCNP code's estimated standard deviation, this ratio will be much less than 0.5.

The total running time for the active cycles is proportional to $N \cdot (I_t - I_c)$, and the standard deviation in the estimated eigenvalue is proportional to $1/\sqrt{N \cdot (I_t - I_c)}$. From the results of the convergence calculation, the total number of histories needed to achieve the desired standard deviation can be estimated.

It is recommended that 200 to 1000 active cycles be used. This large number of cycles will provide large batch sizes of k_{eff} cycles (for example, 40 batches of 10 cycles each for 400 active cycles) to compare estimated standard deviations with those obtained for a batch size of one k_{eff} cycle. For example, for 400 active cycles, 40 batches of 10 k_{eff} values are created and analyzed for a new average k_{eff} and a new estimated standard deviation. The behavior of the average k_{eff} by a larger number of cycles can also be observed to ensure a good normal spatial mode. Fewer than 30 active cycles is not recommended because trends in the average k_{eff} may not have enough cycles to develop.

2.8.3.3 Analysis of Criticality Problem Results

The goal of the calculation is to produce a k_{eff} confidence interval that includes the true result the desired fraction of the time. Check all WARNING messages. Understand their significance to the calculation. Study the results of the checks that the MCNP code makes that were described in §2.8.2.8.

The criticality problem output contains a lot of useful information. Study it to make sure that:

1. the problem terminated properly;
2. enough cycles were skipped to ensure that the normal spatial mode for fission sources was achieved;
3. all cells with fissionable material were sampled;
4. the average combined k_{eff} appears to be varying randomly about the average value for the active cycles;
5. the average combined k_{eff}-by-cycles-skipped does not exhibit a trend during the latter stages of the calculation;
6. the confidence intervals for the batched (with at least 30 batch values) combined k_{eff} do not differ significantly from the final result;
7. the impact of having the largest of each of the three k_{eff} estimators occurring on the next cycle is not too great on the final confidence interval; and
8. the combined k_{eff} figure of merit should be stable.

The combined k_{eff} figure of merit should be reasonably stable, but not as stable as a tally figure of merit because the number of histories for each cycle is not exactly the same, and the combined k_{eff} relative error may experience some changes because of changes in the estimated covariance matrix for the three individual estimators.

Plots (using the z option) can be made of the three individual and average k_{eff} estimators by cycle, as well as the three-estimator-combined k_{eff}. Use these plots to better understand the results.

If there is concern about a calculation, the k_{eff}-by-cycles-skipped table presents the results that would be obtained in the final result box for differing numbers of cycles skipped. This information can provide insight into fission source spatial convergence, normality of the k_{eff} data sets, and changes in the 95% and 99%

confidence intervals. If concern persists, a problem could be run that tallies the track-length estimator k_{eff} using an F*4*:n or TMESH tally and an FM card using the -6 and -7 reaction multipliers (see §6.4.3.1 for an example). In the most drastic cases, several independent calculations can be made and the variance of the k_{eff} values (and any other tallies) could be computed from the individual values.

If a conservative (too large) k_{eff} confidence interval is desired, the results from the largest k_{eff} occurring on the next cycle table can be used. This situation could occur with a maximum probability of $1/(I_t - I_c)$ for highly positively correlated k_{eff} values to $1/(I_t - I_c)^3$ for no correlation.

Finally, keep in mind the discussion in §2.8.2.9. For large systems with a dominance ratio close to one, the estimated standard deviations for tallies could be much smaller than the true standard deviation. The cycle-to-cycle correlations in the fission sources are not taken into account, especially for any tallies that are not made over the entire problem. The only way to obtain the correct statistical errors in this situation is to run a series of independent problems using different random number sequences and analyze the sampled tally results to estimate the statistical uncertainties.

2.9 Volumes and Areas

The particle flux in Monte Carlo transport problems often is estimated as the track length per unit volume or the number of particles crossing a surface per unit area. Therefore, knowing the volumes and surface areas [178] of the geometric regions in a Monte Carlo problem is essential. Knowing volumes is useful in calculating the masses and densities of cells and thus in calculating volumetric or mass heating. Furthermore, calculation of the mass of a geometry is frequently a good check on the accuracy of the geometry setup when the mass is known by other means.

Calculating volumes and surface areas in modern Monte Carlo transport codes is nontrivial. The MCNP code allows the construction of cells from unions and/or intersections of regions defined by an arbitrary combination of second-degree surfaces, toroidal fourth-degree surfaces, or both. These surfaces can have different orientations or be segmented for tallying purposes. The cells they form can even consist of several disjoint subcells. Cells can be constructed from quadrilateral or hexagonal lattices or can be embedded in repeated structures universes. Although such generality greatly increases the flexibility of the MCNP code, computing cell volumes and surface areas understandably requires increasingly elaborate computational methods.

The MCNP code automatically calculates volumes and areas of polyhedral cells and of cells or surfaces generated by surfaces of revolution about any axis, even a skew axis. If a tally is segmented, the segment volumes or areas are computed. For nonrotationally symmetric or nonpolyhedral cells, a stochastic volume and surface area method that uses ray tracing is available [§2.9.3].

2.9.1 Rotationally Symmetric Volumes and Areas

The procedure for computing volumes and surface areas of rotationally symmetric bodies follows:

1. Determine the common axis of symmetry of the cell [178]. If there is none and if the cell is not a polyhedron, the MCNP code cannot compute the volume (except stochastically) and the area of each bounding surface cannot be computed on the side of the asymmetric cell.
2. Convert the bounding surfaces to q-form:

$$ar^2 + br + cs^2 + ds + e = 0, \tag{2.316}$$

where s is the axis of rotational symmetry in the r-s coordinate system. All MCNP surfaces except tori are quadratic surfaces and therefore can be put into q-form.

3. Determine all intersections of the bounding surfaces with each other in the r-s coordinate system. This procedure generally requires the solution of a quartic equation [43]. For spheres, ellipses, and tori, extra intersection points are added so that these surfaces are not infinite. The list of intersections are put in order of increasing s-coordinate. If no intersection is found, the surface is infinite; its volume and area on one side cannot be computed.

4. Integrate over each bounding surface segment between intersections:

$$V = \pi \int r^2 \mathrm{d}s \tag{2.317a}$$

for volumes and

$$A = 2\pi \int r \sqrt{1 + \left(\frac{\mathrm{d}r}{\mathrm{d}s}\right)^2}\, \mathrm{d}s \tag{2.317b}$$

for surface areas.

A bounding surface segment lies between two intersections that bound the cell of interest.

A numerical integration is required for the area of a toroidal surface; all other integrals are directly solved by integration formulas. The sense of a bounding surface to a cell determines the sign of V. The area of each surface is determined cell-by-cell twice, once for each side of the surface. An area will be calculated unless bounded on both sides by asymmetric or infinite cells.

2.9.2 Polyhedron Volumes and Areas

A polyhedron is a body bounded only by planes that can have an arbitrary orientation. The procedure for calculating the volumes and surface areas of polyhedra is as follows:

1. For each facet side (planar surface), determine the intersections (r_i, s_i) of the other bounding planes in the r-s coordinate system. The r-s coordinate system is redefined for each facet to be an arbitrary coordinate system in the plane of the facet.

2. Determine the area of the facet:

$$a = \frac{1}{2} \sum (s_{i+1} - s_i)(r_{i+1} - r_i), \tag{2.318}$$

and the coordinates of its centroid, $(r_\mathrm{c}, s_\mathrm{c})$:

$$r_\mathrm{c} = \frac{1}{6a} \sum (s_{i+1} - s_i)\left(r_{i+1}^2 + r_{i+1} r_i + r_i^2\right), \tag{2.319}$$

$$s_\mathrm{c} = \frac{1}{6a} \sum (r_{i+1} - r_i)\left(s_{i+1}^2 + s_{i+1} s_i + s_i^2\right). \tag{2.320}$$

The sums are over all bounding edges of the facet where i and $i+1$ are the ends of the bounding edge such that, in going from i to $i+1$, the facet is on the right side. As with rotationally symmetric cells, the area of a surface is determined cell-by-cell twice, once for each side. The area of a surface on one side is the sum over all facets on that side.

3. The volume of a polyhedron is computed by using an arbitrary reference plane. Prisms are projected from each facet normal to the reference plane, and the volume of each prism is $V = \mathrm{d}a \cos\theta$ where

d	is the distance from reference plane to facet centroid;
a	is the facet area; and
θ	is the angle between the external normal of the facet and the positive normal of the reference plane.

The sum of the prism volumes is the polyhedron cell volume.

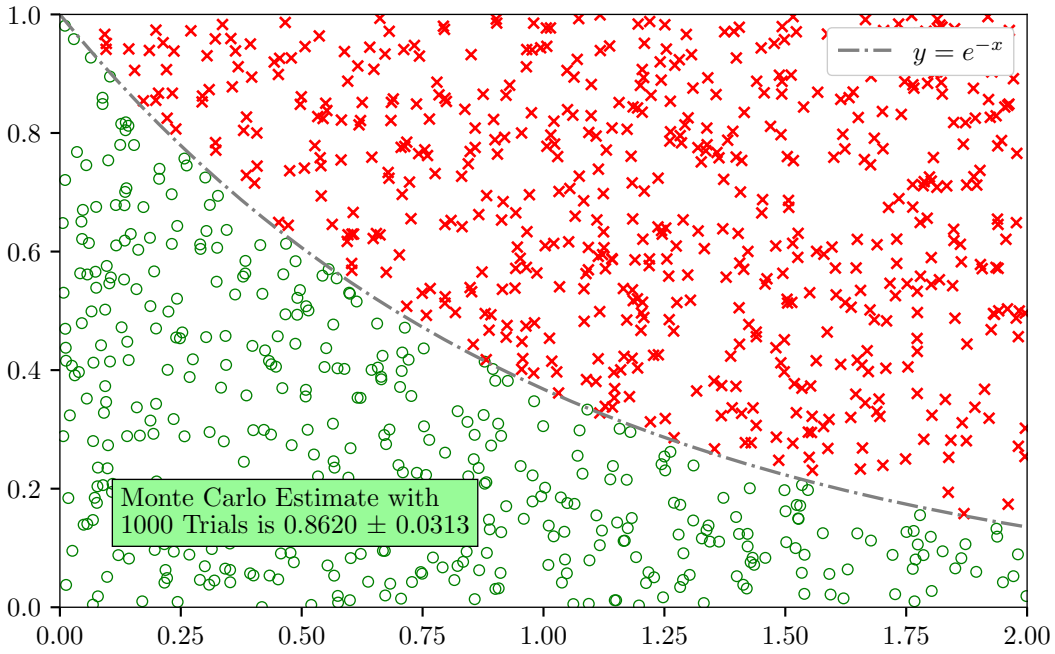


Figure 2.27: Visualization of the estimate of $\int_0^2 \exp(-x)\mathrm{d}x$ using a rejection-sampling Monte Carlo method.

2.9.3 Stochastic Volume and Area Calculation

The MCNP code cannot calculate the volumes and areas of asymmetric, non-polyhedral, or infinite cells. Also, in some cases, the volume and area calculation can fail because of round-off errors. For these cases a stochastic estimation is possible by ray tracing. The theory is described here while the application is given in §5.5.1.1.

The basic idea is to use the Monte Carlo method to solve area and volume integrals and leverage the MCNP code to apply the method.

Where the form of the surface flux and volume flux tallies are given in Equation 2.175, the solution to the tallies (integrals) can be used to calculate the unknown area or volume.

Consider a simple 1-D example: finding the solution to $\int_0^2 \exp(-x)\mathrm{d}x$.

One way to calculate this solution is similar to the classic method of calculating the value of π with a circle inscribed in a square. In this method, two random numbers are selected for the sampled location in the enveloping region, and a comparison is made to see if the location is within the area under the curve of the function. The estimate of the integral would then be the known area of the superimposed rectangle multiplied by the number of positions that are under the curve divided by the total samples. A visualization of this method, as well as the convergence to the true value of $\int_0^2 \exp(-x)\mathrm{d}x = 1 + \sinh(2) - \cosh(2) \approx 0.8646647$ is shown in Figure 2.27. An immediate issue is seen in Figure 2.28 which shows this method hunting for the true value and displaying significant error until a substantial number of samples have been taken (Figure 2.16).

A better method, using sampling on a single random variable, is to sample a PDF over a domain of integration. For any given function, there are a number of PDFs that can be defined to sample over. Again, considering

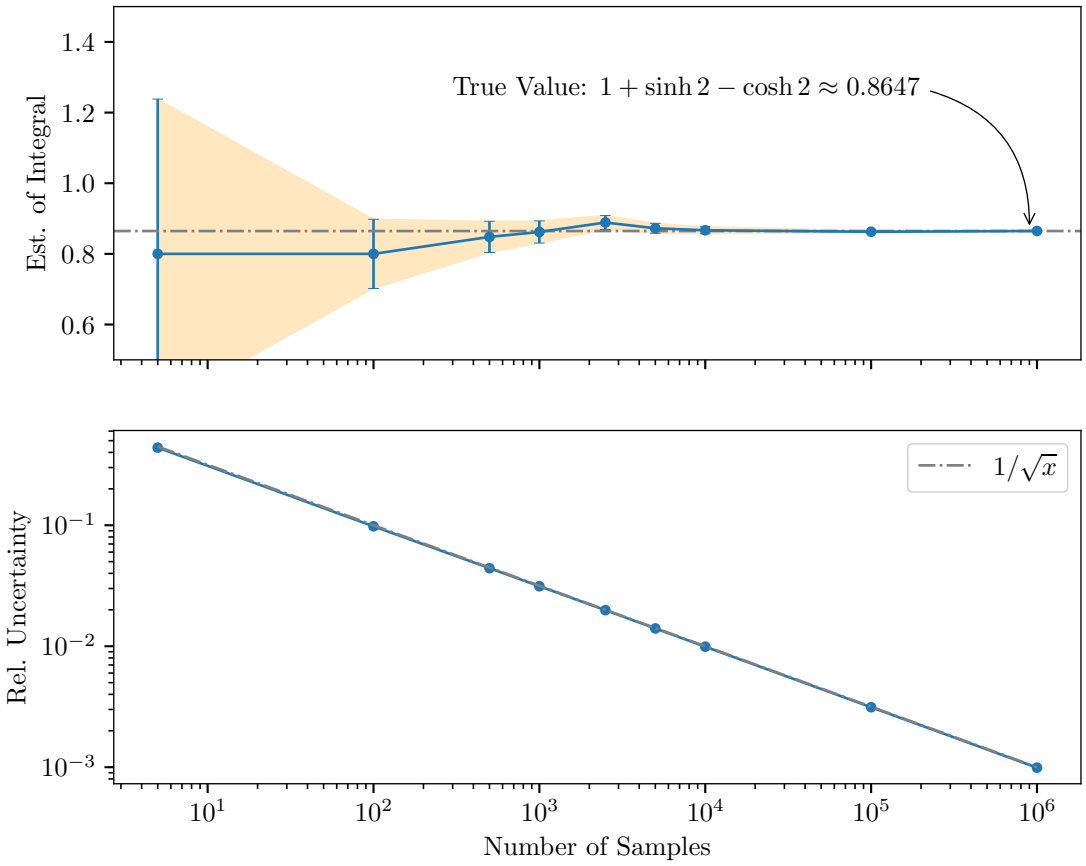


Figure 2.28: Convergence of the rejection-sampling estimate of the variance of the mean of $\int_0^2 \exp(-x)\mathrm{d}x$.

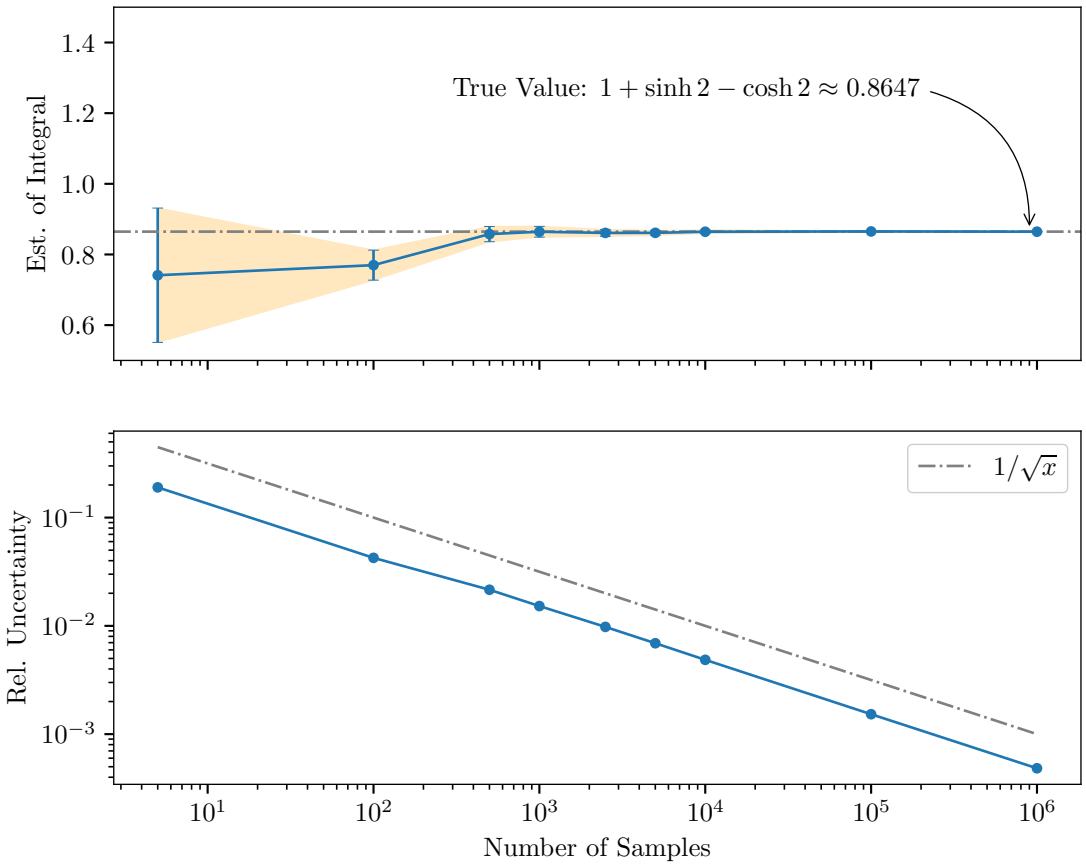


Figure 2.29: Estimate of $\int_0^2 \exp(-x)\mathrm{d}x$ with uniform sampling ($q(x) = 1/2$) .

the integral $\int_0^2 \exp(-x)\mathrm{d}x$, the simplest sampling PDF over the domain of integration is a uniform sampling over the domain

$$q(x) = \begin{cases} (b-a)^{-1} = 0.5, & 0 \leq x \leq 2 \\ 0, & \text{otherwise} \end{cases}. \tag{2.321}$$

This equation is piecewise to satisfy the condition that the integral of the PDF is 1 from $-\infty$ to ∞.

The sampling CDF is then

$$Q(x) = \int_0^x q(x')\mathrm{d}x' = \frac{x}{2}. \tag{2.322}$$

Then, inverting the CDF to obtain samples using a random number between 0 and 1 gives

$$\xi = Q(x) \rightarrow \xi = x/2 \rightarrow x = 2\xi. \tag{2.323}$$

The scoring moments are

$$s^m = \left[\frac{f(x_i)}{q(x_i)}\right]^m. \tag{2.324}$$

This importance sampling preserves the expected value regardless of the PDF selected. As shown in Figures 2.29 and 2.30, multiple functions can be chosen to sample from directly. Some functions are better than others, as shown by the convergence to the true value of the well-biased PDF

$$q(x) = 1 - x/2. \tag{2.325}$$

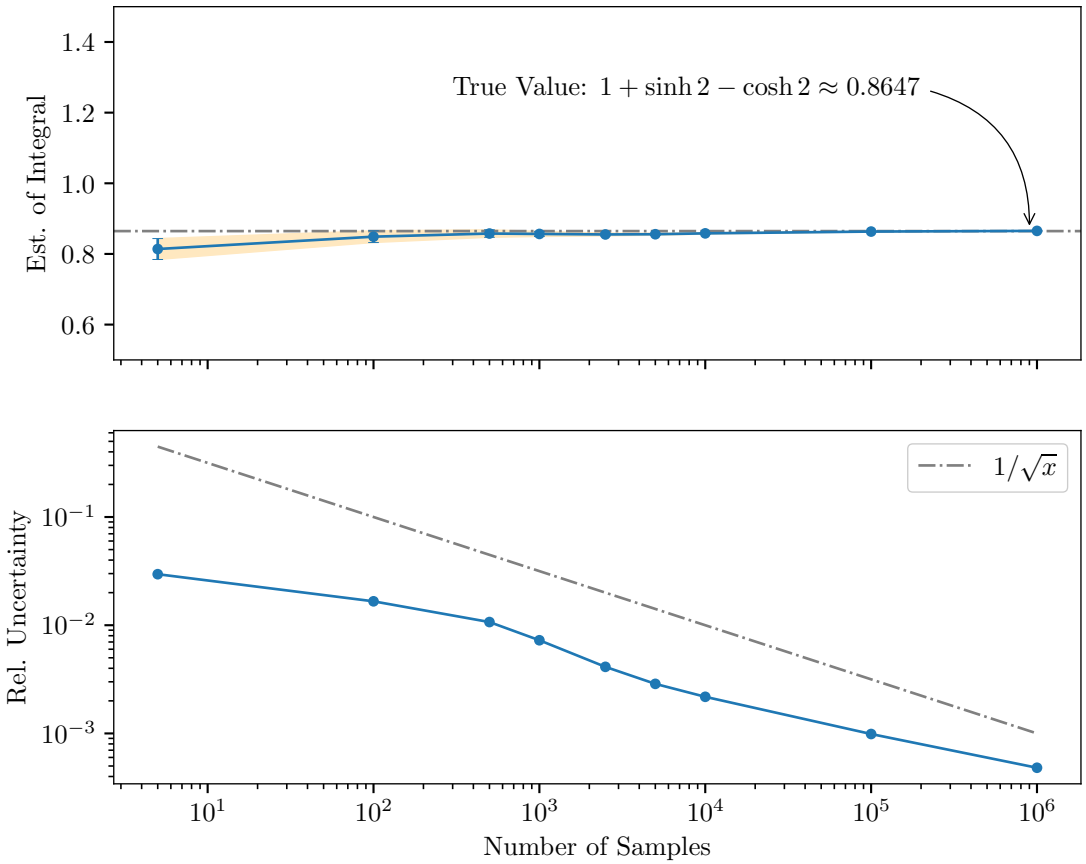


Figure 2.30: Estimate of $\int_0^2 \exp(-x)\mathrm{d}x$ with good ($q(x) = 1 - \frac{x}{2}$) biasing.

When applied to the calculation of arbitrary volumes, the function that defines the volume may not be known. Thus, selecting an effective biasing function can be difficult. However, as described in §5.5.1.1, using an inwardly directed spherical source with a biased cosine distribution is considered the best source setup [179]. While a simple inwardly directed spherical source with rays coming off the surface tangent to the point they are born (`DIR = -1`) may be used, some shapes may benefit from sampling a non-tangent angle which allows more of the unknown volume to be interrogated. Higher track lengths are important to reduce variance in the calculation of this volume. The `WGT` entry on the `SDEF` card should be πr^2, where r is the radius of the sphere, to account for the spherical surface area that defines the source. This is similar to multiplying the area of the bounding box in the naive example described previously.

2.10 Plotter

The MCNP plotter draws cross-sectional views of the problem geometry according to commands entered by the user. See Chapter 6 for the command vocabulary and examples of use. The pictures can be drawn on the screen of a terminal or to a postscript file as directed by the user. The pictures are drawn in a square viewport on the graphics device. The mapping between the viewport and the portion of the problem space to be plotted, called the window, is user-defined. A plane in problem space, the plot plane, is defined by specifying an origin $\mathbf{r}_0$ and two perpendicular basis vectors $\mathbf{a}$ and $\mathbf{b}$. The size of the window in the plot plane is defined by specifying two extents. The picture appears in the viewport with the origin at the center, the first basis vector pointing to the right and the second basis vector pointing up. The width of the picture is twice the first extent and the height is twice the second extent. If the extents are unequal, the picture is distorted. The central task of the plotter is to plot curves representing the intersections of the surfaces of the geometry with the plot plane within the window.

All plotted curves are conics, defined here to include straight lines. The intersection of a plane with any MCNP surface that is not a torus is always a conic. A torus is plotted only if the plot plane contains the torus axis or is perpendicular to it, in which case the intersection curves are conics. The first step in plotting the curves is to find equations for them, starting from the equations for the surfaces of the problem. Equations are needed in two forms for each curve: a quadratic equation and a pair of parametric equations. The quadratic equations are needed to solve for the intersections of the curves. The parametric equations are needed for defining the points on the portions of the curves that are actually plotted.

The equation of a conic is

$$As^2 + 2Hst + Bt^2 + 2Gs + 2Ft + C = 0, \tag{2.326}$$

where s and t are coordinates in the plot plane. They are related to problem coordinates (x, y, z) by

$$\mathbf{r} = \mathbf{r}_0 + s\mathbf{a} + t\mathbf{b} \tag{2.327}$$

or in matrix form

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ x_0 & a_x & b_x \\ y_0 & a_y & b_y \\ z_0 & a_z & b_z \end{bmatrix} \begin{bmatrix} 1 \\ s \\ t \end{bmatrix} \tag{2.328a}$$

$$= PL \begin{bmatrix} 1 \\ s \\ t \end{bmatrix}. \tag{2.328b}$$

Table 2.11: Useful Conic Parametric Equations

Type	Variable		Equation
Straight Line	s	$=$	$C_1 + C_2 p$
	t	$=$	$C_4 + C_5 p$
Parabola	s	$=$	$C_1 + C_2 p + C_3 p^2$
	t	$=$	$C_4 + C_5 p + C_6 p^2$
Ellipse	s	$=$	$C_1 + C_2 \sin p + C_3 \cos p$
	t	$=$	$C_4 + C_5 \sin p + C_6 \cos p$
Hyperbola	s	$=$	$C_1 + C_2 \sinh p + C_3 \cosh p$
	t	$=$	$C_4 + C_5 \sinh p + C_6 \cosh p$

In matrix form the conic equation is

$$0 = \begin{bmatrix} 1 & s & t \end{bmatrix} \begin{bmatrix} C & G & F \\ G & A & H \\ F & H & B \end{bmatrix} \begin{bmatrix} 1 \\ s \\ t \end{bmatrix} \tag{2.329a}$$

$$\begin{bmatrix} 1 & s & t \end{bmatrix} QM \begin{bmatrix} 1 \\ s \\ t \end{bmatrix}. \tag{2.329b}$$

Thus, finding the equation of a curve to be plotted is a matter of finding the QM matrix, given the PL matrix and the coefficients of the surface.

Any surface in the MCNP code, except for tori, can be readily written as

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fzx + Gx + Hy + Jz + K = 0, \tag{2.330}$$

or in matrix form as

$$0 = \begin{bmatrix} 1 & x & y & z \end{bmatrix} \begin{bmatrix} K & G/2 & H/2 & J/2 \\ G/2 & A & D/2 & F/2 \\ H/2 & D/2 & B & E/2 \\ J/2 & F/2 & E/2 & C \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} \tag{2.331a}$$

$$= \begin{bmatrix} 1 & x & y & z \end{bmatrix} AM \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix}. \tag{2.331b}$$

The transpose of the transformation between (s, t) and (x, y, z) is

$$\begin{bmatrix} 1 & x & y & z \end{bmatrix} = \begin{bmatrix} 1 & s & t \end{bmatrix} PT^T, \tag{2.332}$$

where PL^T is the transpose of the PL matrix. Substitution in the surface equation gives

$$\begin{bmatrix} 1 & s & t \end{bmatrix} PT^T \, AM \, PL \begin{bmatrix} 1 \\ s \\ t \end{bmatrix} = 0. \tag{2.333}$$

Therefore, $QM = PL^T \, AM \, PL$.

A convenient set of parametric equations for conics is given in Table 2.11.

The type of a conic is determined by examination of the conic invariants [180], which are simple functions of the elements of QM. Some of the surfaces produce two curves, such as the two branches of a hyperbola or

two straight lines. A separate set of parametric coefficients, C_1 through C_6, is needed for each curve in such cases. The parametric coefficients are found by transforming QM into yet another coordinate system where most of its elements are zero. The parametric coefficients are then simple functions [180] of the remaining elements. Finally, the coefficients are transformed from that coordinate system back to the (s, t) system.

For a torus that can be plotted, the curves are either a pair of identical ellipses or a pair of concentric circles. The parametric coefficients are readily calculated from the surface coefficients and the elements of QM are simple functions of the parametric coefficients.

The next step is to reject all curves that lie entirely outside the window by finding the intersections of each curve with the straight line segments that bound the window, taking into account the possibility that an ellipse may lie entirely inside the window.

The remaining curves are plotted one at a time. The intersections of the current curve, with all of the other remaining curves and with the boundaries of the window, are found by solving the simultaneous equations

$$\begin{bmatrix}1 & s & t\end{bmatrix} QM_i \begin{bmatrix}1\\ s\\ t\end{bmatrix} = 0, \tag{2.334}$$

where $i = 1$ is the current curve and $i = 2$ is one of the other curves. This process generally requires finding the roots of a quartic. False roots and roots outside the window are rejected and the value of the parameter p for each remaining intersection is found. The intersections then are arranged in order of increasing values of p.

Each segment of the curve—the portion of the curve between two adjacent intersections—is examined to see whether and how it should be plotted. A point near the center of the segment is transformed back to the (x, y, z) coordinate system. All cells immediately adjacent to the surface at that point are found. If there is exactly one cell on each side of the surface and those cells are the same, the segment is not plotted. If there is exactly one cell on each side and those cells are different, the segment is plotted as a solid line. If anything else is found, the segment is plotted as a dotted line, which indicates either that there is an error in the problem geometry or that some other surface of the problem also intersects the plot plane along the segment.

If a curve to be plotted is not a straight line, it is plotted as a sequence of short straight lines between selected points on the curve. The points are selected according to the criterion that the middle of the line drawn between points must not lie farther from the nearest point on the true curve than the nominal resolution of the picture. The maximum resolution is fixed at 1/5000 of a side of the viewport.

2.11 Random Numbers

Like any other Monte Carlo program, the MCNP code uses a sequence of random numbers to sample from probability distributions. There are two types of random number generators in the code: linear congruential generators and SFC64. Their structure and behavior are different from each other.

2.11.1 Linear Congruential Generators

The MCNP code has traditionally used the linear congruential scheme of Lehmer [34], though the mechanics of implementation have been modified for portability to different computer platforms. A random sequence of integers I_n is generated by

$$I_{n+1} = G \cdot I_n + C \mod 2^M, \; n = 0, 1, \ldots, \tag{2.335}$$

where G is the random number multiplier, I_0 is the initial random seed, C is an additive constant, and M-bit integers and M-bit floating point mantissas are assumed. The random number is then

$$R_n = 2^{-M} I_n. \tag{2.336}$$

The starting random number for history k is

$$I_0^{(k)} = G^{kS} I_0 + C\big(G^{kS} - 1\big)/(G-1) \mod 2^M, \tag{2.337}$$

where S is the random number stride, that is, the number of random numbers allocated to each single history. This initial random number expression is evaluated very efficiently using a fast skip-ahead algorithm [181]. Successive random numbers for history k are then

$$I_n^{(k)} = G \cdot I_n^{(k)} + C \mod 2^M. \tag{2.338}$$

The default values of G, M, I_0, S, and C, which can be changed with the RAND card, are

$$G = 5^{19} = 19,073,486,328,125 \tag{2.339a}$$
$$M = 48 \tag{2.339b}$$
$$I_0 = 1 \tag{2.339c}$$
$$S = 152,917 \tag{2.339d}$$
$$C = 0 \tag{2.339e}$$

The values of G, M, and C are controlled by the GEN keyword. Generator 1 uses the above values, generators 2–4 use 63 bit integers and a nonzero C, and generators 5–7 use 63 bit integers and a zero C [182]. I_0 is given by SEED, and S is given by STRIDE.

The period P of the MCNP algorithm using the default parameters is $P = 2^{46} \approx 7.04 \times 10^{13}$, and $P = 2^{63} \approx 9.2 \times 10^{18}$ for the extended random number parameters.

The MCNP code prints a WARNING and counts the number of histories for which the stride S is exceeded. The MCNP code also prints a WARNING if the period P is exceeded. Exceeding the stride or the period does not result in wrong answers but may result in an underestimate of the variance. However, because the random numbers are typically used for different purposes, the MCNP code seems insensitive to overrunning either the stride or the period [183] but poor behavior has been observed for select problems [184].

Sometimes users wish to know how much of the variation between problems is purely statistical and the variance is insufficient to provide this information. In correlated sampling [§2.7.2.19] and criticality problems, the variances can be underestimated because of correlation between histories. In this case, rerun the problems with a different random number sequence, either by starting with a new random number or by changing the random number stride or multiplier on the RAND card. The MCNP code checks for and does not allow invalid choices, such as an even numbered initial random number that, after a few random numbers, would result in all subsequent random numbers being zero.

2.11.2 SFC64

The SFC64 version 4 generator, which stands for "Small, Fast, Counting, 64 bit output," is a random invertible mapping generator written by Chris Doty-Humphrey [185]. It is a 256 bit state generator that uses bit level operations in its structure to advance its state:

$$t_n = a_n + b_n + n \tag{2.340a}$$
$$a_{n+1} = b_n \oplus (b_n \gg 11) \tag{2.340b}$$
$$b_{n+1} = c_n + (c_n \ll 3) \tag{2.340c}$$
$$c_{n+1} = \text{rotate\_left}(c, 24) + t_n \tag{2.340d}$$

where $\oplus$ is bitwise XOR, $\gg$ is right shift, $\ll$ is left shift, and rotate_left(a, b) rotates the bits of a over b positions to the left with overflow being placed on the right. All arithmetic is performed on 64 bit unsigned integers. The resulting integer t_n is converted to a floating point value ξ that spans $(0, 1)$ exclusive.

This generator has a few key advantages in comparison to the LCGs [186]. First, due to the use of the counter n within its construction, each initial (a_0, b_0, c_0) configuration will generate an independent sequence of at least 2^{64} values. As such, one does not need to skip through the sequence to achieve independence. Second, the generator produces significantly higher quality bits, with no known test at the time of this writing indicating correlation between bits of the sequence.

As used within the MCNP code, the value a_0 is the user-provided `SEED` value on the `RAND` card. The upper 32 bits of b_0 is a "stream type" identifier. For example, 1 corresponds to the generator used for particle transport, and 4 corresponds to the generator used for tallies that need random samples. The lower 32 bits of b_0 are reserved. c_0 is the history index. Then, prior to use, the generator is iterated 18 times to thoroughly shuffle the state.

When used in this fashion, each history is provided an independent sequence of at least length 2^{64}, effectively preventing random number reuse. For this reason, sequence overrun does not need to be (and is not) tracked with SFC64 in the same manner as stride overrun needs to be tracked for LCGs. In addition, changing the `SEED` value (even from 1 to 2) changes the stream for every single particle.

2.12 Perturbations

The evaluation of response or tally sensitivities to cross-section data involves finding the ratio of the change in a tally to the infinitesimal change in the data, as given by the Taylor series expansion. In deterministic methods, this ratio is approximated by performing two calculations, one with the original data and one with the perturbed data. This approach is useful even when the magnitude of the perturbation becomes very small. In Monte Carlo methods, however, this approach fails as the magnitude of the perturbation becomes small because of the uncertainty associated with the response. For this reason, the differential operator technique was developed.

The differential operator perturbation technique as applied in the Monte Carlo method was introduced by Olhoeft [187] in the early 1960s. Nearly a decade after its introduction, this technique was applied to geometric perturbations by Takahashi [188]. A decade later, the method was generalized for perturbations in cross-section data by Hall [189, 190] and later Rief [191]. A rudimentary implementation into the MCNP code followed shortly thereafter [192]. With an enhancement of the user interface and the addition of second order effects, this implementation has evolved into a standard MCNP feature.

2.12.1 Derivation of the Operator

In the differential operator approach, a change in the Monte Carlo response c, due to changes in a related data set (represented by the parameter v), is given by a Taylor series expansion

$$\Delta c = \frac{\mathrm{d}c}{\mathrm{d}v} \cdot \Delta v + \frac{1}{2!} \cdot \frac{\mathrm{d}^2 c}{\mathrm{d}v^2} \cdot \Delta v^2 + \cdots + \frac{1}{n!} \cdot \frac{\mathrm{d}^n c}{\mathrm{d}v^n} \cdot \Delta v^n + \cdots , \tag{2.341}$$

where the nth-order coefficient is

$$u_n = \frac{1}{n!} \cdot \frac{\mathrm{d}^n c}{\mathrm{d}v^n}. \tag{2.342}$$

This can be written as

$$u_n = \frac{1}{n!} \sum_{b \in B} \sum_{h \in H} x_b^n(h) \left(\frac{\partial^n c}{\partial x_b^n(h)} \right), \tag{2.343}$$

for the data set

$$x_b(h) = K_b(h) \cdot e^v \; ; \; b \in B, \; h \in H, \tag{2.344}$$

where $K_b(h)$ is some constant, B represents a set of macroscopic cross sections, and H represents a set of energies or an energy interval.

For a track-based response estimator

$$c = \sum_j t_j q_j, \tag{2.345}$$

where t_j is the response estimator and q_j is the probability of path segment j (path segment j is comprised of segment $j-1$ plus the current track). This gives

$$u_n = \frac{1}{n!} \sum_j \left[\sum_{b \in B} \sum_{h \in H} x_b^n(h) \left(\frac{\partial^n}{\partial x_b^n(h)} (t_j q_j) \right) \right], \tag{2.346}$$

or

$$u_n = \frac{1}{n!} \sum_j \gamma_{nj} t_j q_j, \tag{2.347}$$

where

$$\gamma_{nj} = \sum_{b \in B} \sum_{h \in H} x_b^n(h) \left(\frac{\partial^n}{\partial x_b^n(h)} (t_j q_j) \right) \left(\frac{1}{t_j q_j} \right). \tag{2.348}$$

With some manipulations presented in [193, 194], the path segment estimator $\gamma_{nj} t_j$ can be converted to a particle history estimator of the form

$$u_n = \sum_i V_{ni} p_i, \tag{2.349}$$

where p_i is the probability of the ith history and V_{ni} is the nth-order coefficient estimator for history i, given by

$$V_{ni} \equiv \frac{1}{n!} \sum_{j'} \gamma_{nj'} t_{j'}. \tag{2.350}$$

Note that this sum involves only those path segments j' in particle history i. The Monte Carlo expected value of u_n becomes

$$\langle u_n \rangle = \frac{1}{N} \sum_i V_{ni} = \frac{1}{Nn!} \sum_i \left(\sum_{j'} \gamma_{nj'} t_{j'} \right), \tag{2.351}$$

for a sample of N particle histories.

The probability of path segment j is the product of the track probabilities,

$$q_j = \prod_{k=0}^{m} r_k, \tag{2.352}$$

where r_k is the probability of track k and segment j contains $m+1$ tracks. If the kth track starts with a neutron undergoing reaction type "a" at energy E' and is scattered from angle θ' to angle θ and E, continues for a length λ_k, and collides, then

$$r_k = \left(\frac{x_a(E')}{x_T(E')} \right) P_a(E' \to E; \theta' \to \theta) \mathrm{d}E \mathrm{d}\theta [\exp(-x_T(E)\lambda_k)] x_T(E) \mathrm{d}\lambda_k, \tag{2.353}$$

where $x_a(E')$ is the macroscopic reaction cross section at energy E', $x_T(E')$ is the total cross section at energy E', and $P_a(E' \to E; \theta' \to \theta)\mathrm{d}E\mathrm{d}\theta$ is the probability distribution function in phase space of the emerging neutron. If the track starts with a collision and ends in a boundary crossing, then

$$r_k = \left(\frac{x_a(E')}{x_T(E')}\right) P_a(E' \to E; \theta' \to \theta)\mathrm{d}E\mathrm{d}\theta[\exp(-x_T(E)\lambda_k)]. \tag{2.354}$$

If the track starts with a boundary crossing and ends with a collision,

$$r_k = [\exp(-x_T(E)\lambda_k)]x_T(E). \tag{2.355}$$

And finally, if the track starts and ends with boundary crossings

$$r_k = \exp(-x_T(E)\lambda_k). \tag{2.356}$$

2.12.1.1 First Order

For a first-order perturbation, the differential operator becomes

$$\gamma_{1j'} \equiv \sum_{b\in B}\sum_{h\in H} x_b(h)\left(\frac{\partial}{\partial x_b(h)}(t_{j'}q_{j'})\right)\left(\frac{1}{t_{j'}q_{j'}}\right) \tag{2.357a}$$

$$= \sum_{b\in B}\sum_{h\in H}\left(\frac{x_b(h)}{q_{j'}}\frac{\partial q_{j'}}{\partial x_b(h)} + \frac{x_b(h)}{t_{j'}}\frac{\partial t_{j'}}{\partial x_b(h)}\right) \tag{2.357b}$$

whereas,

$$\frac{1}{q_{j'}}\frac{\partial q_{j'}}{\partial x_b(h)} = \sum_{k=0}^{m}\frac{1}{r_k}\frac{\partial r_k}{\partial x_b(h)}. \tag{2.358}$$

Then

$$\gamma_{1j'} = \sum_{k=0}^{m}\beta_{j'k} + R_{1,j'}, \tag{2.359}$$

where

$$\beta_{j'k} \equiv \sum_{b\in B}\sum_{h\in H}\left(\frac{x_b(h)}{r_k}\right)\left(\frac{\partial r_k}{\partial x_b(h)}\right) \tag{2.360a}$$

$$= \sum_{b\in B}\sum_{h\in H}\left(\delta_{hE'}\delta_{ba} - \frac{\delta_{hE'}x_b(E')}{x_T(E')} - \delta_{hE}x_b(E)\lambda_k + \frac{\delta_{hE}x_b(E)}{x_T(E)}\right) \tag{2.360b}$$

for a track segment k that starts with a particle undergoing reaction type "a" at energy E' and is scattered to energy E and collides after a distance λ_k. Note that $\delta_{hE'}$and δ_{ba} are unity if $h = E$ and $b = a$; otherwise they vanish. For other types of tracks (for which the various expressions for r_k were given in the previous section), that is, collision to boundary, boundary to collision, and boundary to boundary, derivatives of r_k can be taken leading to one or more of these four terms for $\beta_{j'k}$.

The second term of $\gamma_{1j'}$ is

$$R_{1j'} = \sum_{b\in B}\sum_{h\in H}\frac{x_b(h)}{t_{j'}}\frac{\partial t_{j'}}{\partial x_b(h)}, \tag{2.361}$$

where the tally response is a linear function of some combination of reaction cross sections, or

$$t_{j'} = \lambda_k\sum_{c\in C}x_c(E), \tag{2.362}$$

where c is an element of the tally cross sections, $c \in C$, and may be an element of the perturbed cross sections, $c \in B$. Then,

$$R_{1j'} = \sum_{b\in B}\sum_{h\in H} \frac{x_b(h)}{\left(\sum_{c\in C} x_c(h)\right)} \frac{\partial}{\partial x_b(h)} \left(\sum_{c\in C} x_c(h)\right) \tag{2.363a}$$

$$= \frac{\sum_{b\in B}\sum_{h\in H} x_c(E)}{\sum_{c\in C} x_c(E)}. \tag{2.363b}$$

$R_{1j'}$ is the fraction of the reaction rate tally involved in the perturbation. If none of the nuclides participating in the tally is involved in the perturbation, then $R_{1j'} = 0$, which is always the case for F1, F2, and F4 tallies without FM cards. For F4 tallies with an FM card, if the FM card multiplicative constant is positive (no flag to multiply by atom density) it is assumed that the FM tally cross sections are unaffected by the perturbation and $R_{1j'} = 0$. For KCODE k_{eff} track length estimates, F6 and F7 heating tallies, and F4 tallies with FM cards with negative multipliers (multiply by atom density to get macroscopic cross sections), if the tally cross section is affected by the perturbation, then $R_{1j'} > 0$. For k_{eff} and F6 and F7 tallies in perturbed cells where all nuclides are perturbed, generally $R_{1j'} = 1$.

Finally, the expected value of the first-order coefficient is

$$\langle u_1 \rangle = \frac{1}{N} \sum_i \left[\sum_{j'} \left(\sum_{k=0}^{m} \beta_{j'k} + R_{1j'} \right) t_{j'} \right]. \tag{2.364}$$

2.12.1.2 Second Order

For a second-order perturbation, the differential operator becomes

$$\gamma_{2j'} \equiv \sum_{b\in B}\sum_{h\in H} x_b^2(h) \left(\frac{\partial^2}{\partial x_b^2(h)} (t_{j'} q_{j'}) \right) \left(\frac{1}{t_{j'} q_{j'}} \right). \tag{2.365}$$

Whereas $t_{j'}$ is a linear function of $x_b(h)$, then

$$\gamma_{2j'} = \sum_{b\in B}\sum_{h\in H} \frac{x_b^2(h)}{t_{j'} q_{j'}} \left(t_{j'} \frac{\partial^2 q_{j'}}{\partial x_b^2(h)} + 2 \frac{\partial q_{j'}}{\partial x_b(h)} \frac{\partial t_{j'}}{\partial x_b(h)} + q_{j'} \frac{\partial^2 t_{j'}}{\partial x_b^2(h)} \right) \tag{2.366}$$

Further,

$$\frac{\partial^2 t_{j'}}{\partial x_b^2(h)} = 0 \tag{2.367}$$

and by taking first and second derivatives of the r_k terms of $q_{j'}$ as for the first-order perturbation,

$$\gamma_{2j'} = \sum_{k=0}^{m} \left(\alpha_{j'k} - \beta_{j'k}^2 \right) - R_{1j'}^2 + \left(\sum_{k=0}^{m} \beta_{j'k} + R_{1j'} \right)^2, \tag{2.368}$$

where

$$\alpha_{j'k} = \sum_{b\in B}\sum_{h\in H} \left[\frac{2\delta_{hE'} x_b^2(E')}{x_T^2(E')} - \frac{2\delta_{hE'}\delta_{ba} x_b(E')}{x_T(E')} + \delta_{hE} x_b^2 \lambda_k^2 - \frac{2\delta_{hE} x_b^2(E) \lambda_k}{x_T(E)} \right.$$
$$\left. + 2 \left(\delta_{hE'}\delta_{ba} - \frac{x_b(E')\delta_{hE'}}{x_T(E')} \right) \left(\frac{x_b(E)\delta_{hE}}{x_T(E)} - \lambda_k \delta_{hE} x_b(E) \right) \right]. \tag{2.369}$$

The expected value of the second-order coefficient is

$$\langle u_2 \rangle = \frac{1}{2N} \sum_i \left[\sum_{j'} \left(\sum_{k=0}^{m} (\alpha_{j'k} - \beta_{j'k}^2) - R_{ij'}^2 + \left(\sum_{k=0}^{m} \beta_{j'k} + R_{1j'} \right)^2 \right) t_{j'} \right], \tag{2.370}$$

where $\beta_{j'k}$ and $\alpha_{j'k}$ are given by one or more terms as described above for track k and $R_{1j'}$ is again the fraction of the perturbation with nuclides participating in the tally.

2.12.1.3 Implementation in the MCNP Code

The total perturbation printed in the MCNP output file is

$$\langle \Delta c \rangle = \frac{1}{N} \sum_i \sum_{j'} \Delta c_{j'}. \tag{2.371}$$

For each history i and path j',

$$\Delta c_{j'} = \frac{\mathrm{d}c_{j'}}{\mathrm{d}v} \cdot \Delta v + \frac{1}{2} \cdot \frac{\mathrm{d}^2 c_{j'}}{\mathrm{d}v^2} \cdot \Delta v^2. \tag{2.372}$$

Let the first-order perturbation with $R_{1j'} = 0$ be

$$P_{1j'} = \sum_{j'} \left(\sum_{k=0}^{m} \beta_{j'k} \right) t_{j'}, \tag{2.373}$$

and let the second-order perturbation with $R_{1j'} = 0$ be

$$P_{2j'} = \sum_{j'} \left[\sum_{k=0}^{m} (\alpha_{j'k} - \beta_{j'k}^2) \right] t_{j'}. \tag{2.374}$$

Then the Taylor series expansion for $R_{1j'} = 0$ is

$$\Delta c_{j'} = \left[P_{1j'} \Delta v + \frac{1}{2} (P_{2j'} + P_{1j'}^2) \Delta v^2 \right] t_{j'}. \tag{2.375}$$

If $R_{1j'} \neq 0$ then

$$\Delta c_{j'} = \left[(P_{1j'} + R_{1j'}) \Delta v + \frac{1}{2} \left(P_{2j'} - R_{1j'}^2 + (P_{1j'} + R_{1j'})^2 \right) \Delta v^2 \right] t_{j'} \tag{2.376a}$$

$$= \left[P_{1j'} \Delta v + \frac{1}{2} (P_{2j'} + P_{1j'}^2) \Delta v^2 + R_{1j'} \Delta v + P_{1j'} R_{1j'} \Delta v^2 \right] t_{j'}. \tag{2.376b}$$

That is, the $R_{1j'} \neq 0$ case is just a correction to the $R_{1j'} = 0$ case.

In the MCNP code, $P_{1j'}$ and $P_{2j'}$ are accumulated along every track length through a perturbed cell. All perturbed tallies are multiplied by

$$P_{1j'} \Delta v + \frac{1}{2} (P_{2j'} + P_{1j'}^2) \Delta v^2, \tag{2.377}$$

and then if $R_{1j'} \neq 0$ the tally is further corrected by

$$R_{1j'} \Delta v + P_{1j'} R_{1j'} \Delta v^2.$$

$R_{1j'}$ is the fraction of the reaction rate tally involved in the perturbation. $R_{1j'} = 0$ for F1, F2, F4 tallies without FM cards, and F4 tallies with FM cards with positive multiplicative constants.

2.12.2 Limitations

Although it is always a high priority to minimize the limitations of any MCNP feature, the perturbation technique has the limitations:

1. A fatal error is generated if a PERT card attempts to unvoid a region. The simple solution is to include the material in the unperturbed problem and void the region of interest with the PERT card [Appendix B of 195].
2. A fatal error is generated if a PERT card attempts to alter a material composition in such a way as to introduce a new nuclide. The solution is to set up the unperturbed problem with a mixture of both materials and introduce PERT cards to remove each [Appendix B of 195].
3. The track length estimate of k_{eff} in KCODE criticality calculations assumes the fundamental eigenfunction (fission distribution) is unchanged in the perturbed configuration.
4. DXTRAN, point detector tallies, and pulse height tallies are not currently compatible with the PERT card.
5. While there is no limit to the number of perturbations, they should be kept to a minimum, as each perturbation can degrade performance by 10–20%.
6. Use caution in selecting the multiplicative constant and reaction number on FM cards used with F4 tallies in perturbation problems.
7. The METHOD keyword can indicate if a perturbation is so large that higher than second-order terms are needed to prevent inaccurate tallies.
8. If a perturbation changes the relative abundances of nuclides (MAT keyword) it is assumed that the perturbation contribution from each nuclide is independent (that is, second-order differential cross terms are neglected).

2.12.3 Accuracy

Analyzing the first- and second-order perturbation results presented in [196] leads to the following rules of thumb. The first-order perturbation estimator typically provides sufficient accuracy for response or tally changes that are less than 5%. The default first- plus second-order estimator offers acceptable accuracy for response changes that are less than 20–30%. This upper bound depends on the behavior of the response as a function of the perturbed parameter. The magnitude of the second-order estimator is a good measure of the range of applicability. If this magnitude exceeds 30% of the first-order estimator, it is likely that higher-order terms are needed for an accurate prediction. The METHOD keyword on the PERT card allows one to tally the second-order term separate from the first [§5.10.1].

The MCNP perturbation capability assumes that changes in the relative concentrations or densities of the nuclides in a material are independent and neglects the cross-differential terms in the second-order perturbation term when changing two or more cross sections at once. In some cases there will be a large FALSE second-order perturbation term. Reference [196] provides more discussion and a method for calculating the cross terms.

The MCNP perturbation capability has been shown to be inaccurate for some large but very localized perturbations in criticality problems. An alternative implementation that only requires post-processing standard MCNP tallies has been shown to be much more accurate in some cases [197].

Part II

MCNP User Manual

Chapter 3

Introduction to MCNP Usage

This part of the MCNP manual, Part II, provides comprehensive documentation for using the MCNP code, version 6.3.1. This part includes description of ASCII input file commands (often referred to as input cards), geometry specifications, and tally plotting details. In addition to this manual, classes providing detailed instruction for using MCNP6 are held on a regular basis (see http://mcnp.lanl.gov).

The remainder of Chapter 3 presents an overview of MCNP6 applications and provides a basic primer for MCNP usage with a sample problem. There are certain limitations in code usage that the user must be made aware of; these items are listed in §3.4.5. A general description of the MCNP6 input structure can be found in Chapter 4, while Chapter 5 provides detailed descriptions of each of the available input parameters. Chapter 6 contains basic geometry, cross-section, and tally plotting instructions. Numerous examples, both simple and complex, are presented in Chapter 10 as part of Part III.

3.1 MCNP6 Versatility

Application areas for the code among the thousands of MCNP users worldwide are quite broad and constantly developing. Examples include the following:

- Reactor design
- Nuclear criticality safety
- Nuclear safeguards
- Medical physics, especially proton and neutron therapy
- Design of accelerator spallation targets, particularly for neutron scattering facilities
- Investigations for accelerator isotope production and destruction programs, including the transmutation of nuclear waste
- Research into accelerator-driven energy sources
- Accelerator based imaging technology such as neutron and proton radiography
- Detection technology using charged particles via active interrogation
- Design of shielding in accelerator facilities
- Activation of accelerator components and surrounding groundwater and air
- High-energy dosimetry and neutron detection

- Investigations of cosmic-ray radiation backgrounds and shielding for high altitude aircraft and spacecraft
- Single-event upset in semiconductors from cosmic rays in spacecraft or from the neutron component on the earth's surface
- Analysis of cosmo-chemistry experiments, such as Mars Odyssey
- Charged-particle propulsion concepts for spaceflight
- Investigation of fully coupled neutron and charged-particle transport for lower-energy applications
- Transmutation, activation, and burnup in reactor and other systems
- Nuclear material detection
- Design of neutrino experiments

For all of these applications, the structure of user specification of the problem and selection of input options are similar. The chapters in the remainder of Part II provide general guidance for creation of MCNP input and specific examples of particular problems. The guidance should be sufficient for a user to simulate radiation transport problems in their area of interest.

3.2 MCNP6 Input for Sample Problem

The MCNP6 code is driven primarily by the main input file, with the default name `inp`, that contains a user-specified description of the problem. Users must specify the geometric properties of the problem of interest, initial particle state, relevant physics parameters, and desired tallies.

Problem input consists of a series of structured text commands, where each command is referred to as a card. We present here only the subset of cards required to run the simple fixed source demonstration problem. All input cards are described in Chapter 5.

The MCNP6 code uses consistent units for each dimensional quantity. The units used in the sample problem that follows are length in centimeters (cm), energy in MeV, mass density in grams per cubic centimeter (g/cm^3), and atomic density in atoms/barn-cm. Additional standard MCNP6 units are provided in Chapter 4. The basic constants used in the MCNP6 code are printed in an optional `PRINT` Table 98 in the output file via the `PRINT` card.

The simple sample problem is illustrated in Figure 3.1. We wish to start 14-MeV neutrons isotropically as a point source in the center of the small sphere of oxygen that is embedded in a cube of carbon. A small sphere of iron is also embedded in the carbon. The carbon is a cube 10 cm on each side; the spheres have a 0.5-cm radius and are centered between the front and back faces of the cube. We wish to calculate the total and energy-dependent flux in increments of 1 MeV from 1 to 14 MeV, where bin 1 will be the tally from 0 to 1 MeV

1. on the surface of the iron sphere, and
2. averaged in the iron sphere volume.

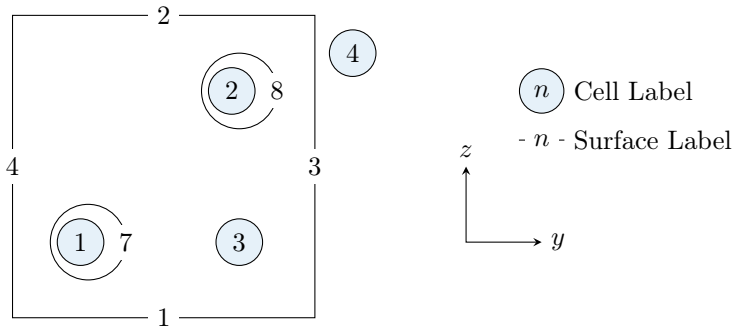


Figure 3.1: A 0.5-cm-radius sphere of oxygen (Cell 1) and a 0.5-cm-radius sphere of iron (Cell 2) embedded in a carbon cube (Cell 3) with a side dimension of 10 cm. Cell 4 represents the "outside world".

3.2.1 Introduction to Geometry Specification

As depicted in Figure 3.1, this geometry has four cells (volumetric regions) and eight two-dimensional surfaces—six planes and two spheres. Circled numbers indicates cell numbers and surface numbers are written next to the appropriate surfaces. Surface 5 comes out from the page in the $+x$ direction and surface 6 goes back into the page in the $-x$ direction.

The cell cards for the geometry of our problem are set up using knowledge of the sense of a surface and the union and intersection operators (see §3.2.3). To simplify this step, assume the cells are void for now. The following cards describe cells 1 and 2:

```
1 0  -7
2 0  -8
```

where the first entry on each of these cell cards is the cell number, the second entry is the material number, with "0" indicating a void, and the third number provides cell surface information. In this sample problem, the negative signs denote the regions inside (the negative sense of) surfaces 7 and 8.

Cell 3 is everything in the problem universe above surface 1 intersected with everything below surface 2, intersected with everything to the left of surface 3, and so forth for the remaining three surfaces. The region in common to all six surfaces is the cube, but we also need to exclude the two spheres by intersecting everything outside surface 7 and outside surface 8. By using a blank space to denote the intersection of two regions of space, the card entries required to describe cell 3 are

```
3 0  1 -2 -3 4 -5 6 7 8
```

Cell 4 requires the use of the union operator, which is denoted by a colon (:) between two surfaces. Cell 4 is the referred to as the "outside world" and is defined as everything in the universe below surface 1, plus everything above surface 2, plus everything to the right of surface 3, and so forth. The cell card for cell 4 is

```
4 0  -1 : 2 : 3 : -4 : 5 : -6
```

Cell 4, the outside world, would usually have zero importance (not denoted here). More guidance on cell importance is provided in §3.2.5.2.

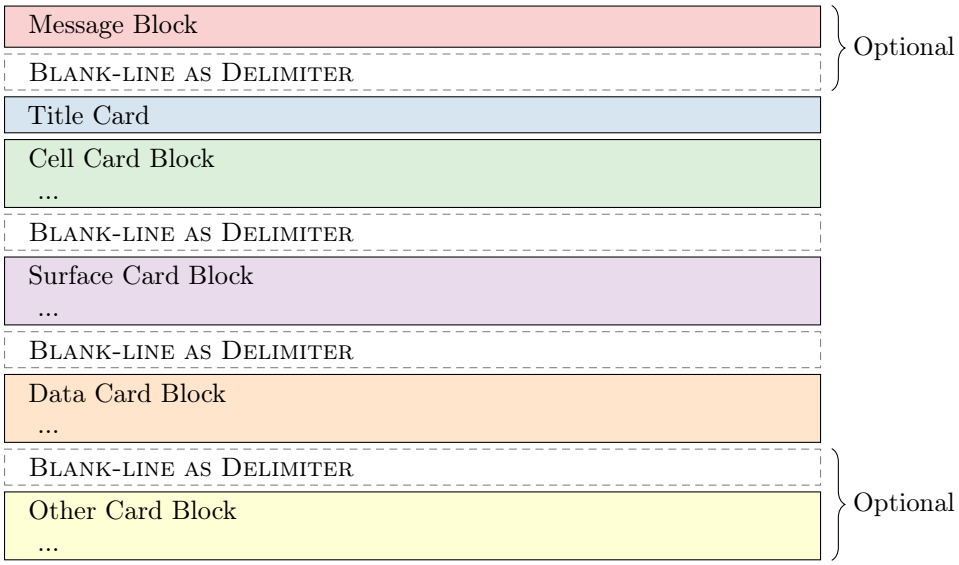


Figure 3.2: MCNP Input File Format

3.2.2 The MCNP Input File

An MCNP6 input file has the form shown in Fig. 3.2.

The input file consists of a series of ASCII text-based, structured commands, referred to herein as cards. Each card consists of a series of keywords and data entries, separated by one or more blank spaces, and a card always starts on a new line. A collection of multiple cards is referred to as a block. The MCNP input consists of one optional and three required blocks. A *single* blank line is used as a delimiter between blocks and as an optional input-file terminator. Care must be taken to not include additional blank lines between blocks, as they will lead to input being ignored and misleading fatal errors.

All input lines are limited to 128 columns. Alphabetic characters can be upper, lower, or mixed case. Unprintable characters or those outside of base ASCII found in an input line are converted to blank spaces. Windows new-line characters are correctly processed in the MCNP code, version 6.2 or newer. MacOS 9-style new-line characters (`CR`) are not supported. A `$` (dollar sign) terminates data entry on a line. Anything on the line that follows the `$` is interpreted as a comment and ignored by the code.

Tab characters in the input file are converted to one or more blank spaces, such that the character following the tab will be positioned at the next tab stop. Tab stops are set every eight characters, i.e., 9, 17, 25, etc. The limit of input lines to 128 columns applies after tabs are expanded into blank spaces. It is recommended that blank spaces be used instead of tab characters for this reason.

Comment cards can be used anywhere in the `inp` file after the problem title card and before the optional blank terminator card. Comment lines must have a `c` somewhere in columns 1–5 followed by at least one space and can be a total of 128 columns long.

Cell, surface, and data cards must all begin within the first five columns. Entries are separated by one or more blank spaces. Numbers can be integer or floating point; the MCNP code makes the appropriate conversion. A few entries on some cards are allowed to be 8-byte integers, i.e., integers larger than 2.147 billion but less than $\sim 10^{19}$. These entries are noted in their respective card description in Chapter 5. A data entry item, e.g., `imp:n` or `1.1e2`, must be completed on one line.

Blank spaces filling the first five columns indicate a continuation of the data from the last named card. An `&` (ampersand) ending a line indicates data will continue on the following card, where data on the continuation card can be in columns 1–128.

The optional message block, discussed in detail in §4.4.1, is used to change file names and specify running options such as a continue-run. On most systems these options and files may alternatively be specified with an execution line (see §3.3.2). If both an execution line and a message block are present and there is a conflict, the execution line entries supersede the message block entries. The blank line delimiter signals the end of the message block.

The first card in the file after the optional message block is the required problem title card. If there is no message block, this must be the first card in the INP file. It is limited to one 128-column line and is used as a title in various places in the MCNP6 output. It can contain any information you desire but usually contains information describing the particular problem. Immediately following the title card are three unlabeled blocks of cards: the cell, surface, and data card blocks, separated by single blank lines.

Input file checking and error handling is described in §4.7.

3.2.3 Cell Cards

After the title card, the next required block is the cell cards. When populating cell cards, the cell number is the first entry and must begin in the first five columns. The next entry is the cell material number, which is arbitrarily assigned by the user. The corresponding material is described on a material (`M`) card with the same material number [§3.2.5.5]. If the cell is a void, a zero is entered for the material number. The cell and material numbers cannot exceed eight digits each. Following the material number is the cell material density. A positive entry is interpreted as atom density in units of 10^{24} atoms/cm^3. A negative entry is interpreted as mass density in units of g/cm^3. There is no density entry for a void cell. After the material density, a complete specification of the geometry of the cell follows. This specification includes a list of the signed surfaces bounding the cell where the sign denotes the sense of the regions defined by the surfaces. The regions are combined with the Boolean intersection and union operators. A space indicates an intersection and a colon indicates a union.

Optionally, after the geometry description, cell parameters can be entered. The form for cell parameters is `KEYWORD=value`. The following line illustrates a cell card format:

```
1 1 -0.0014 -7 IMP:N=1
```

Cell 1 contains material 1 with density 0.0014 g/cm^3, is bounded only by surface 7, and has a neutron importance of 1. If cell 1 were a void, the cell card would be

```
1 0         -7 IMP:N=1
```

The complete cell input for this problem (with two comment cards) is

```
c cell cards for sample problem
1 1 -0.0014 -7
2 2 -7.86 -8
3 3 -1.60 1 -2 -3 4 -5 6 7 8
4 0 -1:2:3:-4:5:-6
c end of cell cards for sample problem
```

The blank line at the end of the card block terminates the cell-card section of the MCNP input file. A complete explanation of the cell card input is found in §5.2.

Table 3.1: Surface Equations for Sample Problem

Mnemonic	Equation	Card Entries
px	$x - D = 0$	D
py	$y - D = 0$	D
pz	$z - D = 0$	D
s	$(x - \overline{x})^2 + (y - \overline{y})^2 + (z - \overline{z})^2 - R^2 = 0$	$\overline{x}\ \overline{y}\ \overline{z}\ R$

3.2.4 Surface Cards

When populating surface cards the surface number is the first entry. The surface number must begin in columns 1–5 and not exceed eight digits. The surface number is followed by an alphabetic mnemonic entry that indicates the surface type. The surface type is followed by the numerical coefficients of the equation of the surface, in the required order. This simplified description enables us to proceed with the sample problem. For a full description of the surface card see §5.3.1.

Our problem uses planes normal to the x-, y-, and z-axes and two general spheres. The respective mnemonics are `px`, `py`, `pz`, and `s`. Table 3.1 shows the equations that determine the sense of the surface for the cell cards and the entries required for the surface cards. A complete list of available surface equations is contained in Table 5.1.

For the planes defining the cube, D is the point where the plane intersects the axis. If we place the origin in the center of the 10-cm cube shown in Figure 3.1, the planes will be at $x = -5$, $x = 5$, etc. The two spheres are not centered at the origin or on an axis, so we must give the x, y, z-coordinates of their center as well as their radii, R. The complete surface card input for this problem is shown below. A blank line terminates the surface card portion of the input.

```
c Beginning of surfaces for cube
1 pz -5
2 pz  5
3 py  5
4 py -5
5 px  5
6 px -5
c End of cube surfaces
7 s 0 -4 -2.5 0.5 $ oxygen sphere
8 s 0  4  4   0.5 $ iron sphere
```

3.2.5 Data Cards

The remaining data input for MCNP6 follows the second blank card delimiter (or third blank card if there is a message block). The data cards block is where users define additional properties about the simulation of particle histories. For each data card, the card name is the first entry and must begin in the first five columns. The required entries follow, separated by one or more blank spaces.

Some data cards require a particle designator that indicates the corresponding particle type of that card. The particle designator consists of the symbol : (colon) and the alphabetic particle symbol (see Table 4.3) immediately following the name of the card. For example, to enter neutron importance, use an `IMP:n` card; enter photon importance on an `IMP:n` card; enter positive pion importance on an `IMP:/` card, etc.

No data card can be used more than once with the same mnemonic, that is, M1 and M2 are acceptable, but two M1 cards are not allowed. Defaults have been set for cards in some categories. The sample problem will use cards in the following categories:

1. physics (MODE)
2. cell and surface parameters (IMP:n)
3. source specification (SDEF)
4. tally specification (F, E)
5. material specification (M)
6. problem termination (NPS)

A complete description of the data cards can be found in §5.4–§5.12.

3.2.5.1 MODE Card

The MODE card consists of the mnemonic `mode` followed by a list of particles (separated by spaces) to be transported. If the `mode` card is omitted, MODE N is assumed (i.e., neutron transport only).

By default, MODE N P does not account for photo-neutrons, but does account for neutron-induced photons. Photonuclear particle production can be turned on through an option on the PHYS:p card [§5.7.2]. Photon production cross sections do not exist for all nuclides. If they are not available for a MODE n p problem, MCNP6 will print out warning messages.

`mode p` or `mode n p` problems generate bremsstrahlung photons with a thick-target bremsstrahlung approximation. This approximation can be turned off with the PHYS:e card.

The sample problem is a neutron-only problem, so the MODE card can be omitted because MODE n is the default.

3.2.5.2 Cell and Surface Parameter Cards

Data related to individual cells can be entered either on the cell card or in the data-card block of the input file. Data related to individual surfaces can only be entered using the data card format. If entered on a card in the data block section, entries must be listed in the same order as the associated cell (or surface) cards that appear earlier in the `inp` file. The number of entries on a cell or surface data card must equal the number of cells or surfaces in the problem, otherwise MCNP6 prints out a warning or fatal error message. In the case of a warning, MCNP6 allows the problem to continue, but assumes that the value of the parameter for each cell or surface is zero. Cell parameters also can be defined on cell cards using the `KEYWORD=`*`value`* format.

⚠ Caution

If a cell parameter is specified on any cell card, that cell parameter must be specified only on cell cards and cannot be present in the data card section.

The IMP:n card is used to specify relative cell importance in the sample problem. There are four cells in the sample problem, so the IMP:n card would have four entries. The IMP:n card is used (a) for terminating the particle's history if the importance is zero and (b) for geometry splitting and Russian roulette to help particles move more easily to important regions of the geometry. An IMP:n card for the sample problem is

Table 3.2: Source Specification Card Entry Summary

Parameter		Entry	Default Entry if Unspecified
pos	=	$x\ y\ z$	0 0 0
cel	=	cell number	Cell containing sampled location
erg	=	energy	14 MeV
wgt	=	statistical weight	1
tme	=	time	0
par	=	source particle type	Lowest-numbered source particle

```
IMP:n 1 1 1 0
```

A listing of available cell parameter cards appears in §5.2. Examples include importance cards (IMP:$\mathscr{P}$) and weight-window cards (WWE:$\mathscr{P}$, WWNi:$\mathscr{P}$), etc. Each problem requires some method of specifying relative cell importance. Most of the other cell parameters are used to specify optional variance reduction techniques. The only surface parameter card is AREA.

3.2.5.3 Source Specification Cards

A source definition (SDEF) card is one of four available methods of defining the properties of starting particles. Section 3.2.5.3 has a complete discussion of source specification. The SDEF card defines the basic source parameters, some of which are given in Table 3.2.

The cel entry is only required if cells are used to restrict the domain of sampled particles; otherwise the code will determine the cell number of starting particles automatically. The default starting direction for source particles is isotropic.

For the example problem, a fully specified source card is

```
sdef pos=0.0 -4.0 -2.5 erg=14 wgt=1.0 tme=0 par=n
```

Neutrons will start at the center of the oxygen sphere $(0, -4, -2.5)$, in cell 1 (as determined by the code), with an energy of 14 MeV, and with weight of 1.0 at time 0. All these source parameters except the starting position are the default values, so the most concise source card is

```
sdef pos=0.0 -4.0 -2.5
```

If all the default conditions applied to the problem, only the mnemonic SDEF would have been required.

3.2.5.4 Tally Specification Cards

The tally cards are used to specify what you want to learn from the Monte Carlo calculation, such as, current across a surface, flux at a point, etc. You request this information with one or more tally cards. Tally specification cards are not required, but if none are supplied, no tallies will be printed when the problem

Table 3.3: Tally Specification Card Entry Summary

Tally Mnemonic	Description
F1:$\mathscr{P}$	Surface current
F2:$\mathscr{P}$	Surface flux
F4:$\mathscr{P}$	Track length estimate of cell flux
F5a:N or F5a:P	Flux at a point (point detector)
F6:$\mathscr{P}$	Track length estimate of energy deposition
F7:n	Track length estimate of fission energy deposition
F8:$\mathscr{P}$	Energy distribution of pulses created in a detector

is run and a warning message is issued. Many of the tally specification cards describe tally "bins." A few examples are energy (E), time (T), and cosine (C) bin cards.

MCNP6 provides seven different standard tally types, all normalized to be per starting particle. Some tallies in criticality calculations are normalized differently. A summary of the tally types is given in Table 3.3, Chapter 2 discusses tallies more completely, and §3.2.5.4 lists all the tally cards and fully describes each one.

The tallies are identified by tally type and particle type. Tallies are given the numbers 1, 2, 4, 5, 6, 7, 8, or increments of 10 thereof, and are given a particle designator indicating the particle type to be tallied. You may practically have as many of any basic tally as you need, each with different energy bins, flagging, or any other desired specification. The tally designations `f4:n`, `f14:n`, `f104:n`, and `f234:n` are all legitimate neutron cell-flux type tallies, as indicated by the last digit of 4; these tallies could all be for the same cell(s) but with different energy or multiplier bins, for example. Similarly `f5:p`, `f15:p`, and `f305:p` are all photon point detector tallies. Having both an `f1:n` card and an `f1:p` card in the same `inp` file is not allowed. Limitations on tally numbers and other quantities is given in Table 4.2.

For our sample problem we will use F cards (tally type) and E cards (tally energy).

3.2.5.4.1 Tally (F) Cards

The sample problem has a surface flux tally and a track length cell flux tally. Thus, the tally cards for the sample problem shown in Figure 3.1 are

```
f2:n   8   $ flux across surface 8
f4:n   2   $ track length in cell 2
```

Printed out with each tally result is the uncertainty of the tally corresponding to one estimated standard deviation. Results are not reliable until they become stable as a function of the number of histories run. Much information is provided for a specified bin of each tally in the tally fluctuation charts at the end of the output file to help determine tally stability. The user is strongly encouraged to look at this information carefully, with more detail provided in §2.6.9.2.3.

3.2.5.4.2 Tally Energy (E) Cards

We wish to calculate neutron flux in increments of 1 MeV from 1 to 14 MeV. Another tally specification card in the sample input file establishes these energy bins.

The entries on the En card are the upper bounds in MeV of the energy bins for tally n. The entries must be given in order of increasing magnitude. If a particle has an energy greater than the last entry, it will not be tallied, and a warning is issued. The MCNP code automatically provides the total over all specified energy bins unless inhibited by putting the symbol `nt` as the last entry on the selected En card.

The following cards will create energy bins for the sample problem:

```
e2 1 2 3 4 5 6 7 8 9 10 11 12 13 14
e4 1 12i 14
```

If no En card exists for tally n, a single bin over all energy will be used. To change this default, an E0 (zero) card can be used to set up a default energy bin structure for all tallies. A specific En card will override the default structure for tally n. We could replace the E2 and E4 cards with one E0 card for the sample problem, thus setting up identical bins for both tallies.

3.2.5.5 Materials Specification

The cards in this section specify both the isotopic composition of the materials and the cross-section evaluations to be used in the cells. For a comprehensive discussion of materials specification, see §5.6.

3.2.5.5.1 Material (M) Card

The following card is used to specify a material for all cells containing material m, where m cannot exceed five digits:

Mm identifier$_1$ *fraction*$_1$ identifier$_2$ *fraction*$_2$ ...

The `m` on a material card corresponds to a material number on a cell card [§3.2.3]. The consecutive pairs of entries on the material card consist of the target identifier of the constituent element or nuclide followed by the atomic fraction (or weight fraction if entered as a negative number) of that element or nuclide, until all the elements and nuclides needed to define the material have been listed. These entries are further described as

1. Nuclide Identifier. This can take any of the possible forms listed in §1.2.3. If a suffix is provided, the code will find the entry with the same Z, A, S, library identifier, and physics identifier as the one input here. This means that U-238.80c can be used to load 92238.80c. If a suffix is not provided, the first Z, A, S match will be used. If atomic data is necessary for the simulation, such as for photoatomic or electron physics, A will be set to zero before searching through the **xsdir**.

   In multi-particle simulations, such as MODE n p, only one suffix can be specified per identifier inline. To control multiple libraries, one can use either the `xlib=` keyword options on the M card to set the value for the entire material, or specify the full identifier on the corresponding MX card.

2. Nuclide Fraction. The nuclide fractions may be normalized to 1 or left unnormalized. For example, if the material is H_2O, the fractions can be entered as 0.667 and 0.333, or as 2 and 1 for H and O, respectively. If the fractions are entered with negative signs, they are weight fractions; otherwise they are atomic fractions. Weight fractions and atomic fractions cannot be mixed on the same M*m* card.

Appropriate material cards for the sample problem are

```
m1   O-16  1      $ oxygen 16
m2   Fe-54 0.0585 $ Iron
     Fe-56 0.9175
     Fe-57 0.0212
     Fe-58 0.0028
m3   C-12  1      $ carbon 12
```

3.2.5.6 Problem Termination

Problem termination cards are used to specify parameters for some of the ways to terminate execution of MCNP6. The full list of available cards and a complete discussion of problem cutoffs is found in §5.13.1. For our problem we will use only the history cutoff (NPS) card. The card name NPS is followed by a numeric entry (npp) that specifies the number of histories to transport. MCNP6 will terminate after *npp* histories unless it has terminated earlier for some other reason.

3.2.6 Sample Problem Input File

The entire input deck for the sample problem is given in Listing 3.1. Recall that the input text can be upper case, lower case, or mixed case.

Listing 3.1: Complete MCNP Input for Sample Problem

```
Sample problem input deck
c Cell cards
1 1 -0.0014  -7
2 2 -7.86    -8
3 3 -1.60     1 -2 -3  4 -5  6  7  8
4 0          -1:2:3:-4:5:-6

c Surface cards
1 pz -5              $ bounding plane (-z)
2 pz  5              $ bounding plane (+z)
3 py  5              $ bounding plane (+y)
4 py -5              $ bounding plane (-y)
5 px  5              $ bounding plane (+x)
6 px -5              $ bounding plane (-x)
7 s   0 -4 -2.5  0.5 $ oxygen sphere
8 s   0  4  4.5  0.5 $ iron sphere

m1  8016 1 $ oxygen 16
m2 26000 1 $ natural iron
m3  6000 1 $ carbon
sdef pos=0 -4 -2.5
f2:n 8 $ flux across surface 8
f4:n 2 $ track length in cell 2
e0 1 12i 14
imp:n 1 1 1 0
nps 100000
```

3.2.7 Running the Sample Problem

To run the example problem, place the input file in your current directory. Let's assume the file is called `primer.mcnp.inp`. In a terminal (or the equivalent command prompt for Windows installations), type

```
mcnp6 n=primer.mcnp.inp
```

where `n` is an abbreviation for the keyword `name`. MCNP6 will produce an output file `primer.mcnp.inpo` that you can examine at your terminal. To look at the geometry with the `plot` module using an interactive graphics terminal, type

```
mcnp6 ip n=primer.mcnp.inp
```

After the plot window appears, click anywhere in the picture to get the default plot. This plot will show an intersection of the surfaces of the problem by the plane $x = 0$ with an extent in the x direction of 100 cm on either side of the origin. If you want to do more with `plot`, see the instructions in Chapter 6. Otherwise click `end` to terminate the session.

3.2.8 Checking for Geometry Errors and the VOID Card

The MCNP6 code does extensive input checking but is not foolproof. A geometry should always be checked by looking at several different views with the geometry plotting option. Any surfaces that bound an incorrectly defined volume are indicated as an error in the plotter via red dotted lines. Additional verification of the geometry can be performed by simulating particles in a voided geometry using the `VOID` card. The `VOID` card (with no parameters) removes all materials and cross sections in a problem and sets all non-zero cell importance to unity. It is very effective for finding errors in the geometry description because many particles can be run in a short time. Flooding the geometry with many isotropic particles increases the chance of particles traversing any invalid regions of the geometry and getting lost. The other uses for the `VOID` card and its parameters are discussed in §5.6.10.

The sample input deck could include a `VOID` card while testing the geometry for errors. The source in this problem is isotropic, and the geometry is simple, so a sufficiently large value specified on the `NPS` card will find any geometry errors. Run a short simulation with the `VOID` card added and study the output to see if you are calculating what you think you are calculating. When you are satisfied that the geometry is error-free, remove the `VOID` card.

For more complicated geometries, it is helpful to surround the entire voided geometry with a sphere and define a source with an inward cosine distribution on the bounding spherical surface. The importance of the cell bounded by the source sphere must be nonzero. For more details on this procedure, see §4.8.

3.3 Executing MCNP6

This section assumes a basic knowledge of command-line interface (CLI) environments. Lines the user will type and execute are written with a `fixed-width font`. Press the Enter key after each input line.

3.3.1 The MCNP6 Runtime Environment

A successful installation of the MCNP6 code will automatically include the location of the executable binary file `mcnp6` in the `PATH` environment variable. The installation process will also specify the `DATAPATH` environment variable, which is read by the code to access nuclear data. Tabular nuclear data files are indexed by the code through the **`xsdir`** file, which is a listing (i.e., a directory) of cross-section data. The `DATAPATH` is the absolute path of the directory containing the latest **`xsdir`** file and additional data library files. If the code is compiled manually or installation errors have occurred, it may be necessary to modify these environment variables before executing the `mcnp6` binary; always take care when modifying the `PATH` variable.

3.3.2 Execution Line

The MCNP6 execution line has the following form:

```
mcnp6 KEYWORD=value ... KEYWORD=value execution_option(s) other_options
```

where each instance of `KEYWORD` is an MCNP6 default file name to which the user may assign a specific `value` (i.e., file name or path); `execution_option`(s) provides a character or string of characters that informs MCNP6 which of five execution module(s) to run; and `other_option`(s) provides the user with additional execution control. The execute line message may be up to 4096 characters long. The order of the entries on the MCNP6 execution line is irrelevant. If no changes are desired to the default names and options, no entries to the MCNP6 execution line are necessary.

The execution-line keywords (i.e., default file names), execution options, and other options are summarized in Tables 3.4, 3.5, and 3.6, respectively. Each of these execution-line inputs is detailed in the following sections.

3.3.2.1 Execution `KEYWORD=value` Entries

The entry `KEYWORD` is any of the available default MCNP6 file names. The code uses several files for input and output. User-specified file names can include full paths to the files (e.g., `/mydir/problem-x/jobs/problem_1a.inp`), but the path cannot be longer than 256 characters. In the simplest case, in which the MCNP6 execution command has no arguments, a file named `inp` must be present in the local directory; then, during problem execution, MCNP6 will create two output files: `outp` and `runtpe`. Other simulations will require additional files or generate additional output files.

The default name of any of the files in Table 3.4 can be changed on the MCNP6 execution line by entering

```
KEYWORD=newname
```

For example, if you have an input file called `mcin` and want the output file to be `mcout` and the restart file to be `mcrestart`, the appropriate execution line would read

```
mcnp6 inp=mcin outp=mcout runtpe=mcrestart
```

Only enough letters of the default name are required to identify it uniquely. For example,

```
mcnp6 i=mcin o=mcout ru=mcrestart
```

also works. If a file in your local file space has the same name as a file MCNP6 needs to create, the file is created with a different unique name by changing the last letter of the name of the new file to the next letter in the alphabet. For example, if you already have a file named `outp` in the directory, MCNP6 will create `outq`. However, if the file includes an extension, such as `.txt` or `.inp`, the last character before the extension will be checked and changed if necessary.

Sometimes it is useful for all files from one calculation to have similar names. If your input file is called `job1`, the following line

```
mcnp6 name=job1
```

will create an output file called `job1o` and a restart file called `job1r`. If these files already exist, the code will *not* overwrite them or modify the last letter, but will issue a message that `job1o` already exists and terminate.

3.3.2.2 Execution Options

MCNP6 provides users control over the execution of six distinct modules: **`imcn`**, **`plotg`**, **`xact`**, **`mcrun`**, **`mcplot`**, and **`partisn_input`**. The **`xact`** and **`mcrun`** options are ignored when they are combined with **`the partisn_input`** option. A description of these modules, including a one-letter mnemonic assigned to each, appears in Table 3.5.

Given no other instructions, MCNP6 will process the input (`i`), process the cross-section data (`x`), and then perform the particle transport (`r`). Thus, the default execution input is `ixr`. Entering the proper mnemonic on the execution line controls the execution of the modules. If more than one operation is desired, combine the single characters (in any order) to form a string. To look for input errors only, specify `i`; to debug a geometry by plotting, use `ip`; to plot cross-section data, enter `ixz`; to plot tally results from the `runtpe` or `mctal` files, specify `z`; and to create a LNK3DNT geometry file for use in PARTISN, specify `m` on the execution line as the `execution_option`.

After a calculation has been run, the print file `outp` can be examined with an editor on the computer. Numerous important messages about the problem execution and statistical quality of the results are displayed at the terminal; these messages are repeated in the `outp` file.

The `other_option` entries add additional flexibility when running MCNP6 executables and are shown in Table 3.6.

3.3.2.3 Parallel Execution

To take advantage of multi-core computer architecture, MCNP6 provides two parallel models: task-based threading using the OpenMP model and distributed processing supported through the use of the MPI model.

The simplest parallel execution of the code is to use theading with the **`tasks`** option on the command line as follows:

```
mcnp6 i=input tasks <n>
```

MCNP6 will create <*n*> processes running on separate CPUs.

Because adding OpenMP capability requires extensive modifications to the source code, some MCNP features cannot run in parallel using threading. These features include:

- `DBCN` event logging,
- CINDER for delayed neutrons and/or photons using `ACT`,
- LLNL photofission multiplicity,
- `FMULT` CGMF, FREYA, or LLNL fission multiplicity models,
- `SSR` surface source write,
- legacy (i.e., non-HDF5) `PTRAC`,
- `TMESH` tallies,
- `HISTP` file generation,
- model physics for missing cross section data, and
- any particle other than neutrons, photons, or electrons.

If any of these features are used in the problem, MCNP6 will print a warning message and run the problem with only one thread.

To use MPI parallelization, a separate MPI software package must be installed that is compatible with the provided MCNP6 MPI binaries. These MPI packages and associated MCNP6 executable names are:

- Linux
  - OpenMPI: mcnp6.ompi
  - MPICH: mcnp6.mpich
- macOS
  - OpenMPI: mcnp6.mpi
- Windows
  - Microsoft-MPI: mcnp6.mpi.exe

Generally, the MPI execution line using `mpiexec` program will look like:

```
mpiexec -n <m> mcnp6.mpi i=input ...
```

where **`mcnp6.mpi`** refers to the appropriate MCNP6 MPI executable name and where <*m*> is the total number of MPI processes, including the manager, and <*m*>-1 worker processes that will track the particles. On other systems, or for other MPI implementations, the syntax of the MPI command may differ. The MCNP6 MPI executables include OpenMP, allowing each MPI worker to utilize additional threads; this is accomplished by setting the `tasks` option as follows:

```
mpiexec -n <m> mcnp6.mpi i=input tasks <n>
```

where *<m>*-1×*<n>* processes are used to track particles. The syntax required to allocate enough resources for the threading varies by system. To utilize an MPI-compiled executable in sequential or threads-only mode (for example, in early testing of a problem or for plotting geometry), run **mcnp6.mpi** without **mpiexec** -n *<m>*.

Table 3.4: MCNP6 Execution Line Inputs—File Names

Keyword† (Default File Name)	Description‡
COM	File from which plot commands will be read.
COMOUT	File to which all plot requests are written.
DUMN1 and **DUMN2**	Command-line-specified file names for FILES card.
HISTP	Command-line-specified HISTP (history tape) file name.
INP	User-supplied input file name. This is the name of the file that contains the problem input specification and must be present as a local file.
KSENTAL	Name of ASCII results file for KCODE sensitivity profiles.
LINKIN	Name of LNK3DNT file to input.
LINKOUT	Name of LNK3DNT-format geometry file created by MCNP6.
MCTAL	Tally results file (ASCII).
MDATA	TMESH mesh tally data (unformatted binary).
MESHTAL	FMESH tally output file (ASCII).
NAME	User-supplied input file name. Will automatically generate **OUTP**, **RUNTPE**, **MDATA** files with the user-supplied name appended with a **o**, **r**, and **d**, respectively. Other generated output files will have unique corresponding single character extentions, followed by any file-type extensions.
OUTP	File name to which results are written. This file may be viewed and/or printed. Created by MCNP6 during problem execution.
PARTINP	PARTISN input file for MCNP6 to output.
PLOTM	Name of graphics metafile.
PTRAC	Name (without extension) of output file containing user-filtered particle events.
RSSA	Name of file from which surface and volume source particles are read.
RUNTPE	Name of file containing binary start/restart data. Created by MCNP6 during initial problem execution and modified by the code during continued problem execution.
SRCTP	Name of file containing fission source data for a KCODE calculation.
WSSA	Name of file to which surface and volume source particles are recorded.
WWINP	Name of weight-window generator input file containing either cell- or mesh-based lower weight-window bounds.
WWONE	Name of weight-window generator output file containing cell- or mesh-based time- and/or energy-integrated weight windows.
WWOUT	Name of weight-window generator output file containing either cell- or mesh-based lower weight-window bounds.
XSDIR	Name of cross-section directory (XSDIR) file. Note: The default name for the XSDIR file is version specific, e.g., **xsdir_mcnp6.3** for MCNP6.3.

† Requires only enough letters of the default name to identify it uniquely.
‡ File names are limited to a maximum of 256 characters. File names may also include directory paths.

Table 3.5: MCNP6 Execution Line Inputs—Mode Options

Option†	Description
`-v`	Print build info to screen, if used all other command line input is disregarded. Input file is not needed for this option.
`i`	Execute module IMCN to process the input file.
`m`	Execute module PARTISN_INPUT to create LNK3DNT-format geometry file.
`p`	Execute module PLOTG to plot geometry.
`r`	Execute module MCRUN to perform the particle transport.
`x`	Execute module XACT to process the cross-section data.
`z`	Execute module MCPLOT to plot tally results or cross-section data.

† Default is `ixr`.

Table 3.6: MCNP6 Execution Line Inputs—Other Options

Option	Description
`BALANCE`	Provides load balancing when used with MPI. Note: When using multiprocessing with `KCODE` runs, the option for load balancing is not available.
`CN` *m*	Continue a run, starting with the *m*th dump and writing the dumps immediately after the fixed part of the **`RUNTPE`**, rather than at the end.
`C` *m*	Continue a previous run starting with the *m*th dump. If *m* is omitted, the last dump is used.
`DBUG` *n*	Write debug information every *n* particles. Note: An 8-byte integer is allowed.
`DEV-TEST`	Delete time-dependent quantities from files for SQA testing. See 3rd entry on the `PRDMP` card.
`EOL`	Add after all other MCNP6 keywords to distinguish MCNP6 keywords from directives added by MPICH. Only needed if the MPICH implementation of MPI is used.
`FATAL`	Transport particles and calculate volumes even if fatal errors are found. Not applicable to UM calculations.
`NOTEK`	Indicates that the terminal has no graphics capability. `PLOT` output is in **`PLOTM.PS`**. Equivalent to `TERM=0`.
`PRINT`	Create the full output file; equivalent to `PRINT` card in the input file.
`TASKS` *n*	Invokes OpenMP threading on shared memory systems. The parameter *n* is the number of threads to be used. This keyword may be used in conjunction with MPI on a hybrid system.

3.3.3 Interrupts

For non-MPI versions, MCNP6 allows four types of interactive interrupts while it is running:

Ctrl + c , Enter	MCNP6 status (Default if no entry is provided)
Ctrl + c , s	MCNP6 status. The Ctrl + c , s interrupt causes MCNP6 to print the computer time used so far, the number of particles run so far, and the number of collisions. If the code is processing in the IMCN module, it prints the input line being processed and if in the XACT module, it prints the cross section being processed.
Ctrl + c , m	Invoke MCPLOT to create interactive plots of tallies or to further invoke PLOT, to plot the geometry
Ctrl + c , q	Quit MCNP6 gracefully after current history. The Ctrl + c , q interrupt has no effect until MCRUN, the particle transport section of the code, is executed. After particle transport simulation has commenced <ctrl-c>q causes the code to stop after the current particle history, to terminate gracefully, and to produce final output and RUNTPE files.
Ctrl + c , k	Kill MCNP6 immediately. The Ctrl + c , k interrupt kills MCNP6 immediately, without normal termination. If Ctrl + c , k fails, enter Ctrl + c three or more times in a row.

Batch calculations, run in sequential or multiprocessing mode, may be interrupted and stopped with the creation of a file in the directory where the calculation was started. The name of the file must be "STOP*inp*" where *inp* is the name of the original input file that initiated the run. On a computer system that is case sensitive (e.g., Linux), the "stop" must be in lower case and *"INP"* must match the case of the input file name. The contents of this file are meaningless. Once this file is created, MCNP6 will terminate the calculation during the next output rendezvous (see 5th entry on PRDMP card) as if a Ctrl + c , q interrupt had been issued.

Caution

If one uses the Ctrl + c , q interrupt during a KCODE multiple-processor MPI calculation in Linux, MCNP6 does not finish writing the OUTP file before the code exits. This failure appears to be an MPI error in the MPI_FINALIZE call, where the last processor kills all worker and manager processes. Also, the Ctrl + c interrupt does not function properly when using the MPI executable on Windows systems.

On some computer systems, MPI versions, even when run sequentially, do not allow the interactive interrupts because the MPI daemon catches the signal and aborts the MCNP6 run.

3.4 Tips for Correct and Efficient Problems

Provided in this section are checklists of helpful advice that applies to three phases of your calculation: defining and setting up the problem, preparing for the long computer runs that you may require, and executing the runs that will give you results. A fourth checklist is provided for KCODE calculations. These checklists should be periodically revisited as you simulate more complicated problems using the techniques described in the remainder of Part II.

3.4.1 Problem Setup

1. Draw a picture of your geometry to help you with geometry setup.
2. Always plot the geometry to see if it is defined correctly and that it is what was intended.

3. Model the geometry and source distribution in enough detail as needed for accurate particle tracking.
4. Use simple cells.
5. Use the simplest surfaces that solve the problem, including macrobodies.
6. Avoid excessive use of the complement operator, #.
7. Do not set up all the geometry at one time.
8. Put commonly used cards in a separate file and add them to your input file via the READ card.
9. Pre-calculate and compare MCNP6-calculated mass, cell volumes, and surface areas.
10. Use the VOID card when checking the geometry.
11. Look at print tables 10, 110, and 170 to check the source.
12. Check your source with a mesh tally.
13. Be aware of physics approximations, problem cutoffs, and default cross sections.
14. Cross-section sets matter! Check the listing of datasets in the output file.
15. Use separate tallies for the fluctuation chart.
16. Use the most conservative variance-reduction techniques.
17. Do not use too many variance-reduction techniques.
18. Balance user time with computer time.
19. Study *all* warning messages.
20. Generate the best output (consider always using the PRINT card).
21. Recheck the INP file (materials, densities, masses, sources, etc.).
22. Remember that garbage into MCNP6 equals garbage out of MCNP6.

3.4.2 Preproduction

1. Do *not* use MCNP6 as a black box. Become familiar with the theory and methods.
2. Run some short calculations.
3. Examine the outputs carefully.
4. Study the summary tables.
5. Study the statistical checks on tally quality and the sources of variance.
6. Study the trends of the figures of merit and variance of the variance.
7. Consider the collisions per source particle.
8. Examine the track populations by cell.
9. Scan the mean-free-path column.
10. Check detector diagnostic tables.
11. Understand large tally contributions (with event logs).
12. Strive to reduce the number of unimportant tracks.
13. Check secondary particle production.
14. Compare a "back-of-the-envelope calculation" to MCNP6 results.

3.4.3 Production

1. Save RUNTPE file for expanded output printing, continue-run, and tally plotting.
2. Limit the size of the RUNTPE file with the PRDMP card.
3. Look at figure of merit stability.
4. Make sure answers seem reasonable.
5. Examine and understand the 10 statistical checks provided by MCNP6.
6. Form valid confidence intervals.
7. Make continue-runs if necessary.
8. See if stable errors decrease by $1/\sqrt{N}$.
9. Remember, accuracy is only as good as the nuclear data, modeling, MCNP6 sampling approximations, etc.
10. Adequately sample all cells.

3.4.4 Criticality

1. Determine how many inactive cycles are needed by using the MCNP6 plotter to examine the behavior of k_{eff} and the Shannon entropy of the source distribution with cycle number.
2. Run a large number of histories per cycle. For production runs, at least 10000 neutrons per cycle are recommended. More neutrons per cycle are better.
3. Examine the behavior of k_{eff} with cycle number and continue calculations if trends are noticed.
4. Use at least 100 cycles after source convergence.
5. After a production run, use the MCNP6 plotter again to examine the behavior of k_{eff} and the Shannon entropy of the source distribution with cycle number. Ensure that a sufficient number of inactive cycles were used so that k_{eff} and the source distribution are both properly converged.

3.4.5 Warnings and Limitations

All computer simulation codes must be validated for specific uses, and the needs of one project may not overlap completely with the needs of other projects. It is the responsibility of the user to ensure that his or her needs are adequately identified and that benchmarking activities are performed to ascertain how accurately the code will perform. The benchmarking done by code developers for the MCNP6 sponsors may or may not be adequate for a user's particular requirements. We make our benchmarking efforts public as they are completed, but the user must also develop a rigorous benchmarking program for their own application. Such benchmarking efforts by the user also ensures that the user understands how to use MCNP6 for their application.

The following warnings and known bugs apply to the energies and particles beyond MCNP4C [198]:

1. Perturbation methods used in MCNP4C have not been extended yet to the non-tabular models present in MCNP6. MCNP6 gives a fatal error if it is run for problems that invoke the perturbation capabilities above the MCNP4C energy range or beyond the MCNP4C particle set.

2. KCODE criticality calculations work only with available actinide nuclear data libraries and have not been extended to the model energy regions of the code.

3. Charged-particle reaction products are not generated for some neutron reactions below 20 MeV in the LA150N library. In calculating total particle-production cross sections, the library processing routines include only those reactions for which complete angular and energy information is given for secondary products. Most 150-MeV evaluations are built "on top" of existing ENDF and JENDL evaluations which typically go to 20 MeV. Although the 150-MeV evaluations do include the detailed secondary information in the 20–150-MeV range, the <20-MeV data typically do not. Therefore secondary production is generally ignored when processing interactions in that energy range. Table 4.3 lists the actual secondary particle-production thresholds in LA150N. Fixing this situation is non-trivial, and involves a re-evaluation of the low-energy data. Improved libraries will be issued, but on an isotope-by-isotope basis.

4. Beware of the results of an F6:P tally in small cells when running a photon or photon/electron problem. Photon heating numbers include the energy deposited by electrons generated during photon collisions, but assume that the electron energy is deposited locally. In a cell where the majority of the electrons lose all of their energy before exiting that cell, this is a good approximation. However, if the cell is thin and/or a large number of electrons are created near the cell boundary, these electrons could carry significant energy into the neighboring cell. For this situation, the F6:P tally for the cell in which the electrons were created would be too large. The user is encouraged to consider use of the F6:E tally instead, which provides an accurate tally of electron energy deposition within a cell.

5. The FLUKA [199] physics module that was in MCNPX is not present in MCNP6. We recommend using the Los Alamos Quark-Gluon String Model (LAQGSM03.03) [200–216] for very high-energy calculations.

6. Specifying different densities for the same material produces a warning. MCNP6 performs a material density correction for charged-particle energy deposition that is not a strict linear function. MCNP6 searches through all cells, finds the first one with the material of interest, and uses the associated material density to determine the correction factor for all cells using that material. For MCNP4C applications the effect is typically small; therefore this is an adequate procedure. For MCNP6 applications that utilize more charged particles and a greatly expanded energy range, this formerly "small" correction becomes increasingly important, and the usual way of handling it is not sufficient. A suggested practice in such instances is to specify a unique material identifier for each density.

7. "Next-event estimators," i.e., point and ring detectors, DXTRAN, and radiography tallies, use an assumption of isotropic scatter for contributions from collisions within the model regime. These estimators require the angular distribution data for particles produced in an interaction to predict the "next event." Information on these distributions is available in tabular form in the libraries; however, this information is not available in the required form from physics models used to produce secondary particles above the tabular region.

8. A numerical problem occurs in the straggling routines with densities less than about 10^{-9} g/cm^3 for heavier charged particles and with densities less than about 10^{-15} g/cm^3 for electrons. Users should avoid such low densities.

Chapter 4

Description of MCNP6 Input

The MCNP code processes several input files during execution. These processed files are either provided as part of the code distribution, generated during MCNP code execution, or supplied by the user. This chapter focuses on the user-supplied input file. The input file is the primary way that users interact with the MCNP code. The input file contains information about the problem including the geometry specification, the description of materials and selection of cross-section evaluations, the location and characteristics of the source, the type of answers or tallies desired, and other user-controlled properties of the simulation. The remainder of this chapter provides details on the structure, requirements, and processing of the input file.

The input file can have two forms: one for an initial calculation and one for a restarted calculation. The user specifies problem parameters using various ASCII text-based input cards. All input cards are described in Chapter 5. The user will provide only a small subset of all available input cards in a given problem. All features of MCNP6 should be used with knowledge and caution. Read and understand the relevant sections of the manual before using them.

Throughout the input file, alphabetic characters can be upper, lower, or mixed case. Table 4.2 summarizes some of the numerical limitations within the code on user-specified numeric labels for various features. Table 4.1 summarizes some additional limitations for certain input cards. For particular problems, a user may need to increase the size of the dimensioned arrays associated with some of these parameters by altering the source code and recompiling. Although not limited by the code, in some cases a large input may produce an MCNP simulation that exceeds the available system memory for a particular compute resource. Total storage requirements in such cases can be significantly reduced by turning off model physics for neutron problems (see `PHYS`:N card).

Table 4.1: MCNP Code Option Limitations

Category	Number Allowed
Maximum quantity of `TMESH` tallies	20
Maximum quantity of transformations	999
Maximum levels for nested geometry	20
Maximum length of the list for any single cell	9,999
Maximum length of file names	256 characters
Maximum file name path length, including file name	256 characters
Maximum file path lengths for `EMBED` card	80 characters
Maximum quantity of different tallies	9,999
Maximum quantity of detector tallies	10,000
Quantity of DXTRAN spheres per particle type	10
Quantity of URAN universes	unlimited
Entries on `IDUM` and `RDUM` cards	2,000
Maximum length of an input line (card)	128 characters

Table 4.2: MCNP Numerical Limitations on Permitted Values for Card Labels

Category	Number Allowed
Cell numbers	1–99,999,999
Surface numbers	1–99,999,999
Material numbers	1–99,999,999
Universe numbers	0–99,999,999
Surface numbers for transformations	1–999
Cell numbers for transformations	1–999
Tally numbers	1–99,999,999
Perturbation numbers	1–99,999,999
Source distribution numbers	1–999
"Card numbers"	1–99,999,999

4.1 MCNP Units

The units of measurement used throughout MCNP6 are the following:

- length in centimeters,
- energy in MeV,
- time in shakes (10^{-8} s),
- temperature in MeV (kT),
- atomic density in atoms/barn-cm,
- mass density in g/cm^3,
- cross sections in barns (10^{-24} cm^2),
- heating numbers in MeV/collision, and
- atomic weight ratio based on a neutron mass of 1.008664967 amu (as compared to the current NIST value of 1.008664915 amu [217]). In these units, Avogadro's number is $0.59703109 \times 10^{24}$ per neutron mass in amu. This corresponds to a value of Avogadro's number of 6.02204×10^{23} per mole (as compared to the current NIST value of 6.02214×10^{23} per mole [217, 218]). These details are used by the MCNP code for computing atom densities from mass densities, so these details may become important when modifying existing **xsdir** files or specifying the XS card.

All numerical entries specified by the user will be processed by MCNP6 using the above units, for the physical parameter corresponding to the user entry. See PRINT Table 98 for other physical constants used by MCNP6. Certain outputs have different units, which are noted as appropriate.

4.2 Initiate Calculation

This form of the input file is used to set up a Monte Carlo particle transport problem (describe geometry, materials, tallies, etc.). Upon execution, the MCNP code will process the input and simulate the specified problem, using additional information from either the message block or the execution line. See §3.3.2.2 for more detail on executing the initial calculation input file. The initial calculation input file has the form shown in Fig. 4.1.

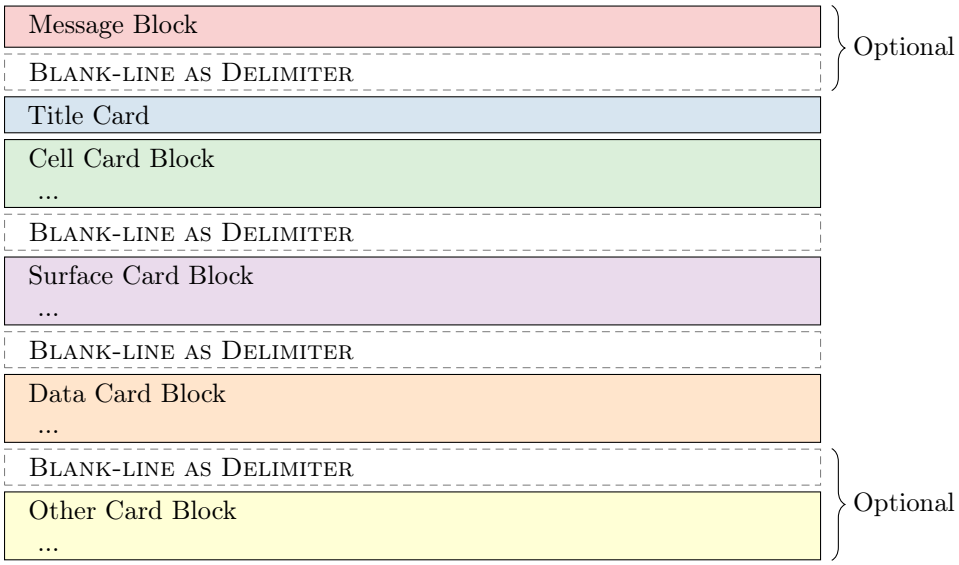


Figure 4.1: MCNP Initial-calculation Input File Format

In an MCNP6 initial calculation input file, an optional message block with its blank line delimiter is followed by a required title card. After the title card appears, three card blocks follow, each separated by a *single* blank line. These three blocks provide, respectively,

1. cell descriptions,
2. surface descriptions, and
3. data about everything else in the problem (materials, source, tallies, etc.).

MCNP6 interprets a blank line as the end of the preceding information block. A final (optional) blank line at the end of the data block signals the end of the input file. With a valid set of cards, MCNP6 will run with or without the blank line terminator. However, when MCNP6 encounters the blank line terminator, MCNP6 will stop reading the input file even if additional lines exist in the file. This region following the blank line terminator can be used by the user for problem documentation or to retain cards not used in the current run.

4.3 Restarted Calculations

This section describes the form of the input file used to restart a previously terminated calculation where it left off. Restarted calculations can also be used to reconstruct the output of a previous calculation. During MCNP execution of an input file used in an initial calculation, MCNP6 will generate an HDF5 restart file with default name **`runtpe.h5`**. See §D.2 for more details on the restart file and §D.1 for information on binary HDF5 files in general. This self-contained restart file contains all information necessary to restart the initial calculation from the beginning. In addition, the problem results at various stages of the calculation are recorded in a series of dumps that can be used to restart (i.e., continue) a simulation. For example, a simulation run for two hours may be restarted and executed for additional time. Generally, for a restart file to be readable by the MCNP code during a restarted calculation, its closure must have been complete without an unexpected crash, error, or file corruption during the execution of the initial (or previous) calculation.

There are two ways to restart a calculation, which differ depending on whether previous dumps are preserved or not. The MCNP6 execution line (or message-block execution information) must contain either a `C` or `CN`

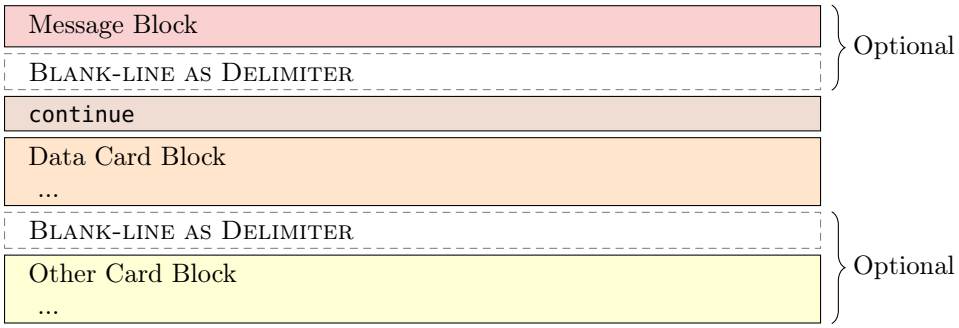


Figure 4.2: MCNP Restart-calculation Input File Format

entry to restart a calculation. The restarted calculation will start with the last dump on the specified restart file by default. Alternatively, it will start execution with the dump numbered *m* if either `C` *m* *or* `CN` *m* is specified, where *m* is an integer corresponding to the "dump no." generated by a previous MCNP6 execution. The available dumps for restarting calculations are sequentially numbered HDF groups listed in the group **`variable`** of the restart file. See the PRDMP card for a discussion of the selection of the dump frequency.

The MCNP6 code checks the validity of the requested dump's results by ensuring the simulation that generated the dump did not terminate unexpectedly from an error (e.g., too many lost particles or a "bad trouble" error). These unexpected terminations are more likely to leave the code and results in an invalid state during MPI execution. If such an error is detected, the restarted calculation will stop immediately with a corresponding message. If the error can be corrected through the restarted calculation's input file, then the restarted calculation can be executed with a dump number earlier in the simulation to get results. If not, the original error must be corrected and the initial calculation repeated from the beginning.

Unlike other execution forms, restarted calculations use the same restart file. When the **`C`** option is specified on the MCNP6 execution line, the dumps produced during the restarted calculation are appended to the dumps from which the calculation restarted. However, by specifying the **`CN`** option instead of the **`C`** option, all previous dumps are removed and new dumps are appended. These new dumps overwrite the old dumps, providing a way for the user to prevent unmanageable growth of restart files for particularly large simulations. Restart file growth also can be controlled by the ndmp entry on the PRDMP card. For both execution options, only the **`variable`** group is changed during the restarted calculation, so all fixed problem data is preserved between code executions.

⚠ Caution

A restarted calculation with the **`CN`** option will overwrite the dump that began the restarted calculation. Thus, it is recommended to use the **`C`** option unless disk space is an issue, particularly if the dump may be revisited, e.g., to compare results with different card changes in the restarted calculation's input file.

In addition to the **`C`** or **`CN`** option on the MCNP6 execution line, users can specify the restart file name and an optional restart-calculation input file. Specifying the file name to be used by MCNP6 is detailed in §3.3.2.2. The optional restart-calculation input file must have the word `CONTINUE` as the first entry on the first line (title card), or after the optional message block and its blank line delimiter. This file has the form shown in Fig. 4.2.

The data cards allowed in the restart-calculation input file are a subset of the data cards available for an initial-calculation input file. The allowed restart-calculation data cards include FQ, DD, NPS, CTME, IDUM, RDUM, PRDMP, LOST, DBCN, PRINT, KCODE, MPLOT, MESH, TALNP, ZA, ZB, ZC, FMESH, RAND, STOP, ZD, and EMBED.

Additionally, the number of threads *n* specified on the execution line (`tasks n`) may be changed between calculations.

If none of the above items is to be changed the restart-calculation input file is not required; only the restart file and the `C` option on the MCNP6 execution line are necessary. For example, with a properly closed **runtpe.h5** file in the current directory the command line sequence **mcnp6 c** will pick up the calculation where it stopped and continue until another stopping condition is reached.

If the initial calculation producing the restart file was stopped because of particle cutoff (`NPS` card), the value of *npp* on the `NPS` card must be increased for a restarted calculation via a restart-calculation input file. The parameter *npp* represents the cumulative histories to be run, i.e., it includes the summation of the initial-calculation and restart-calculation histories. Contrarily, the *tme* parameter on the `CTME` card in a restarted calculation is the number of minutes *more* to run, not the cumulative time. To run more `KCODE` cycles, only the fourth entry on the `KCODE` card, *kct*, must be changed. Like *npp*, *kct* refers to total cycles to be run, including previous ones.

In a restarted calculation, a negative number entered for *npp* on the `NPS` card produces a print output file at the time of the requested dump. No more histories will be run. This can be useful when the printed output has been lost or you want to alter the content of the output with the `PRINT` or `FQ` cards.

Restarted calculations do not produce identical results to initial-runs for delayed particle calculations (see the `ACT` card).

Note that restart files *are not* compatible from one version of MCNP6 to the next. Therefore, a restarted calculation should use the same code version as that which created the restart file. Unless explicitly noted during code execution, restart files can be read independent of whether MPI and/or task parallelism is used during the initial or restarted calculations. Be aware that files from the initial calculation will be overwritten during the restarted calculation if the `FILES` card was present in the initial calculation's input file; see the `FILES` card for more details.

4.4 Card Format

Most text input is entered in a horizontal format as a series of cards, with individual data entries that are separated by one or more spaces. Blank lines are used as delimiters between input blocks and as a terminator after the data block to indicate the end of the input file. The remainder of this section describes the structure and options for text entries in the various blocks of the input file.

4.4.1 Message Block

The optional message block allows the user to provide MCNP6 with additional execution information. It is also a convenient way to avoid retyping an often-repeated message. Both initial- and restart-calculation input files can contain a message block that replaces or supplements the MCNP6 execution line information. If used, the message block is located before the problem title card in the input file. The message block starts with the string, **MESSAGE:**. The message block ends with a blank line delimiter before the title card. The commands may continue over multiple lines of the message block, up to the blank line delimiter. The **$** symbol (which indicates an end-of-line comment follows) and the **&** symbol (which indicates that the information continues on the next line) are permitted in message blocks and act as end-of-line markers. The syntax and components of the message are the same as for the regular execution line entries. Any file name substitution, program module execution option, or keyword entry on the execution line takes precedence over conflicting information in the message block. Renaming of the input file default file name, i.e., `INP` = `filename`, is not a recognized entry in the message block.

For example, assume the MCNP6 execution line required to run the input file, **sphere.i**, and to assign user-designated names to the output files is

```
mcnp6 i=sphere.i o=sphere.o r=sphere.r mctal=sphere.m
```

The following simplified execution line

```
mcnp6 i=sphere_msgblock.i
```

can provide the same file name assignments through the message block feature:

```
message: o=sphere.o r= sphere.r mctal= sphere.m

Title: bare uranium sphere
```

The three lines above show the message block and the input-file title card separated by a blank line delimiter for an input file **sphere_msgblock.i**.

4.4.2 Problem Title Card

The first card in the file after the optional message block is the required problem title card. If there is no message block, this must be the first card in the INP file. It is used as a title in various places in the MCNP6 output. It can contain any information the user desires (or it can be left blank), but it typically contains information describing the particular problem. Note that a blank line elsewhere is used as a delimiter or as a terminator.

4.4.3 Comment Cards

General comment cards can be used anywhere in the input file after the problem title card and before the last blank terminator card. These comment cards are used, and encouraged, in input files to provide additional explanation, details, and organization to the often cryptic input commands. Comment cards must have a `c` anywhere in columns 1–5 followed by at least one blank. General comment cards are printed only with the input file listing and not anywhere else in the MCNP6 output file. Additionally, a comment can be added to any input card: a `$` (dollar sign) terminates data entry on a card and anything that follows the `$` is interpreted as a comment. One exception is that you cannot enter a comment card or a `$` terminator within a `TMESH` tally definition.

Specific comment cards are provided for tallies (the `FC`n card) and for sources (the `SC`n card). User-provided text on these cards are printed in the output as a tally title and as a heading for a source probability distribution, *n*, respectively.

4.4.4 Auxiliary Input File Capability

Subsections of the input file may be inserted using the `READ` card. The text of these insertions will be expanded in the output file unless disabled by the `NOECHO` keyword on the `READ` card.

4.4.5 Cell, Surface, and Data Cards

Detailed specifications for the cell, surface, and data card blocks are provided in Chapter 5. A general description of the structure of the cards in each of these blocks is provided in §3.2.2. Although a horizontal input format for cards is most commonly used, a vertical format option permitted by MCNP6 for certain data block cards. The vertical format is particularly useful for some cell parameters and source distributions. Both formats are described in the sections that follow.

4.4.5.1 Data Card Horizontal Input Format

Like cell and surface cards, data cards all must begin within the first five columns. The card name or number and particle designator are followed by data entries separated by one or more blanks. An individual entry cannot be split between two lines. There can be only one card of any given type for a given particle designation [§4.5]). Integers must be entered where integer input is required. Other numerical data can be entered as integer or floating point and will be read properly by MCNP6. In fact, non-integer numerical data can be entered in any form acceptable to a Fortran E-edit descriptor.

MCNP6 allows five shortcuts to facilitate data input:

1. nR means *repeat* the immediately preceding entry on the card n times. For example, `2 4R` is the same as `2 2 2 2 2`.
2. nI means *insert* n linear interpolates between the entries immediately preceding and following this feature. For example, `1.5 2I 3.0` on a card is the same as `1.5 2.0 2.5 3.0`. In the construct X nI Y, if X and Y are integers, and if Y-X is an exact multiple of $n+1$, then correct integer interpolates will be created. Otherwise, only real interpolates will be created, but Y will be stored directly in all cases. In the above example, the 2.0 value may not be exact, but in the example `1 4I 6`, all interpolates are exact and the entry is equivalent to `1 2 3 4 5 6`.
3. xM means *multiply* the previous entry on the card by the value x. For example, `1 1 2M 2M 2M 2M 4M 2M 2M` is equivalent to `1 1 2 4 8 16 64 128 256`.
4. nJ means *jump* over the entry and take the default value. As an example, the following two cards are identical in their effect:

```
DD 0.1 1000
```

```
DD J 1000
```

   `J J J` is also equivalent to `3J`. Also using this shortcut, you can jump to a particular entry on a card without having to specify explicitly previous items on the card. This feature is convenient if you know you want to use a default value but cannot remember it. Another example of this capability is `DBCN` 2J 10 15.
5. nLOG or, equivalently, nILOG means insert n (base-10) logarithmic interpolates between the entries immediately preceding and following this feature. For example, `0.01 2ILOG 10` is equivalent to `0.01 0.1 1 10`. In the construct X nILOG Y, X and Y must be non-zero and have the same sign otherwise a fatal error is produced.

These features apply to both integer and floating-point quantities. If n (an integer) is omitted in the constructs nR, nI, nLOG, nILOG, and nJ, then n is assumed to be 1. If x (integer or floating point) is omitted in xM, it is a fatal error. The rules for dealing with adjacent special input items are as follows:

1. nR must be preceded by a number or by an item created by R or M.
2. nI, nLOG, and nILOG must be preceded by a number or by an item created by R or M, and must be followed by a number. The preceding number cannot be 0.0 for nLOG or nILOG.
3. xM must be preceded by a number or by an item created by R or M.
4. nJ may be preceded by anything except I and may begin the card input list.

Several examples follow:

- `1 3M 2R` is equivalent to `1 3 3 3`
- `1 3M I 4` is equivalent to `1 3 3.5 4`
- `1 3M 3M` is equivalent to `1 3 9`
- `1 2R 2I 2.5` is equivalent to `1 1 1 1.5 2.0 2.5`
- `1 R 2M` is equivalent to `1 1 2`
- `1 R R` is equivalent to `1 1 1`
- `1 2I 4 3M` is equivalent to `1 2 3 4 12`
- `1 2I 4 2I 10` is equivalent to `1 2 3 4 6 8 10`
- `3J 4R` is illegal
- `1 4I 3M` is illegal
- `1 4I J` is illegal

4.4.5.2 Data Card Vertical Input Format

Column input is particularly useful for cell parameters and source distributions. Cell importance or volumes strung out on horizontal input lines are not very readable and often lead to errors when users add or delete cells. In vertical format, all the cell parameters for one cell can be on a single line, labeled with the name of the cell. If a cell is deleted, the user deletes just one line of cell parameters instead of hunting for the data item that belongs to the cell in each of several multi-line cell-parameter cards. For source distributions, corresponding `SI`, `SP`, and `SB` values are side by side. Source options, other than defaults, are on the next line and must all be entered explicitly. The `&` continuation symbol is not needed and is ignored if it is present.

In column format, card names are put side by side on one input line and the data values are listed in columns under the card names. To indicate that vertical input format is being used, a `#` is put somewhere in columns 1–5 on the line with the card names. The card names must be all cell parameters, all surface parameters, or all something else. If a card name appears on a `#` card, there must not be a regular horizontal card by that name in the same input file. If there are more entries on data value lines than card names on the `#` line, the first data entry is assumed to be a cell or surface number. If any cell names are entered, all must be entered. If cell names are entered, the cells do not have to be in the same order as they are in the cell cards block. If cell names are omitted, the default order is the order of the cells in the cell card block. The same rules apply

to surface parameters, but because we presently have only one surface parameter (AREA), column input of surface parameters is less useful.

There can be more than one block of column data in an input file. Typically, there would be one block for cell parameters and one for each source distribution. If a lot of cell parameter options are being used, additional blocks of column data would be needed.

We strongly suggest keeping columns reasonably neat for user readability. The column format is intended for input data that naturally fit into columns of equal length, but less tidy data are not prohibited. If a longer column is to the right of a shorter column, the shorter column must be filled with enough J entries to eliminate any ambiguity about which columns the data items are in.

Special syntax items (R, M, I, LOG, ILOG, and J) are not as appropriate in column format as they are on horizontal lines, but they are not prohibited. They are, of course, interpreted vertically instead of horizontally. Multiple special syntax items, such as 9R, are not allowed if cell or surface names are present.

The form of a column input block is:

$$\begin{array}{cccccc} \texttt{\#} & S_1 & S_2 & \cdots & S_m \\ k_1 & d_{11} & d_{12} & & d_{1m} \\ k_1 & d_{21} & d_{22} & & d_{2m} \\ \\ \vdots & & \ddots & & \vdots \\ \\ k_N & d_{N1} & d_{N2} & \cdots & d_{Nm} \end{array}$$

The following rules apply:

1. The # is somewhere in columns 1–5.
2. Each line can be only 128 columns wide.
3. Each column, S_i through d_{li}, where l may be less than N, represents a regular input card.
4. The S_i must be valid MCNP6 card names. They must be all cell parameters, all surface parameters, or all something else.
5. d_{li} through d_{Ni} must be valid entries for an S_i card, except that $d_{l+1,i}$ through d_{Ni} may be some Js possibly followed by some blanks.
6. If d_{ji} is non-blank, $d_{j,i-1}$ must also be non-blank. A J may be used if necessary to make $d_{j,i-1}$ non-blank.
7. The S_i must not appear anywhere else in the input file.
8. The k_j are optional integers. If any are non-blank, all must be non-blank.
9. If the S_i are cell parameter card names, the k_j, if present, must be valid cell names. The same is true with surface parameters.
10. If the k_j is present, the d_{ji} must not be multiple special syntax items, such as 9R or 9M.

4.4.6 Continuation Lines

If the first five columns of a card are blank, the entries on the card are interpreted as a continuation of the data from the last named card. The user also can continue data on the following card by ending the line with an `&` (ampersand) preceded by at least one blank space. In this case, the data on the continuation card can be anywhere from column 1 until the end of the line [Table 4.1]. As before, completely blank cards are reserved as delimiters between the cell, surface, and data-card blocks in the input file.

4.5 Particle Designators

Numerous input cards require a particle designator to distinguish between input data for tracked particles. Refer to the pertinent card information for instructions. The particle designator consists of a colon (:) followed by the particle symbol(s) immediately after the name of the card. These particle designations are presented in Table 4.3.

Details:

1. "DOP" indicates decayed on production.
2. The "`#`" symbol represents all possible heavy-ion types—in other words, any ion that is not one of the four light ions available in MCNP6. A list of heavy ions available for transport is provided in Appendix C.
3. Positrons are treated identical to electrons in transport (outside magnetic field effects), and need to be used with the electron transport option (particle `e`) enabled. Refer to the `MODE` card to see the exceptions for positron labels.

At least one blank must follow the particle designator. For example, `IMP:N` signifies neutron importance values follow while `IMP:P` signifies photon importance values follow. To specify the same value for more than one kind of particle, a single card can be used instead of several. For example, if the designation `IMP:E,P,N 1 1 0` appears on a cell card, the electron importance for that cell is 1, the photon importance is 1, and the neutron importance is 0. With a tally card, the particle designator follows the card name including tally number. For example, `*F5:N` indicates a neutron point-detector energy tally. In the heating tally case, two particle designators may appear. The syntax `F6:N,P` indicates the combined heating tally for both neutrons and photons.

4.6 Default Values

Many MCNP6 input parameters have default values that are described with the associated card definitions. Therefore, you do not always have to specify explicitly every input parameter every time if the defaults match your needs. If an input card is left out, the default values for all parameters on the card are used. However, if you want to change a particular default parameter on a card where others precede that parameter, you have to specify the others or use the *nJ* jump feature to jump over the parameters for which you still want the defaults. For example, the input `CUT:p 3J -0.10` is a convenient way to use the defaults for the first three parameters on the photon cutoff card but change the fourth.

Table 4.3: MCNP6 Particle Identifiers and Parameters

ipt	Name	Symbol	Mass [219] (MeV)	Low Energy Cutoff (MeV)	Default Cutoff (MeV)	Mean Lifetime [219] (s) In MCNP6	Mean Lifetime [219] (s) Actual (if different)
1	neutron (n)	N	939.56563	0.0	0.0	Stable	887.0
2	photon (γ)	P	0.0	10^{-6}	10^{-3}	10^{29}	
3	electron (e^-)	E	0.511008	10^{-5}	10^{-3}	10^{29}	
4	negative muon (μ^-)	|	105.658389	10^{-3}	0.11261	2.19703×10^{-6}	
5	anti neutron ($\overline{n}$)	Q	939.56563	0.0	0.0	Stable	887.0
6	electron neutrino (ν_e)	U	0.0	0.0	0.0	10^{29}	
7	muon neutrino (ν_m)	V	0.0	0.0	0.0	Stable	
8	positron (e^+) [3]	F	0.511008	10^{-3}	10^{-3}	10^{29}	
9	proton (p^+)	H	938.27231	10^{-3}	1.0	10^{29}	
10	lambda baryon (Λ^0)	L	1115.684	10^{-3}	1.0	DOP [1]	2.632×10^{-10}
11	positive sigma baryon (Σ^+)	+	1189.37	10^{-3}	1.26760	DOP [1]	7.99×10^{-11}
12	negative sigma baryon (Σ^-)	-	1197.436	10^{-3}	1.26760	DOP [1]	1.479×10^{-10}
13	cascade; xi baryon (Ξ^0)	X	1314.9	10^{-3}	1.0	DOP [1]	2.9×10^{-10}
14	negative cascade; negative xi baryon (Ξ^-)	Y	1321.32	10^{-3}	1.40820	DOP [1]	1.639×10^{-10}
15	omega baryon (Ω^-)	O	1672.45	10^{-3}	1.78250	DOP [1]	8.22×10^{-11}
16	positive muon (μ^+)	!	105.658389	10^{-3}	0.11261	2.19703×10^{-6}	
17	anti electron neutrino ($\overline{\nu}_e$)	<	0.0	10^{-3}	0.0	10^{29}	
18	anti muon neutrino ($\overline{\nu}_m$)	>	0.0	0.0	0.0	Stable	
19	anti proton ($\overline{p}$)	G	938.27231	10^{-3}	1.0	10^{29}	
20	positive pion (π^+)	/	139.56995	10^{-3}	0.14875	2.603×10^{-8}	
21	neutral pion (π^0)	Z	134.9764	0.0	0.0	8.4×10^{-17}	
22	positive kaon (K^+)	K	493.677	10^{-3}	0.52614	1.2371×10^{-8}	
23	kaon, short (K^0_S)	%	497.672	0.0	0.0	0.8926×10^{-10}	
24	kaon, long (K^0_L)	^	497.672	0.0	0.0	5.17×10^{-8}	
25	anti lambda baryon ($\overline{\Lambda^0}$)	B	1115.684	10^{-3}	1.0	DOP [1]	2.632×10^{-10}
26	anti positive sigma baryon ($\overline{\Sigma^+}$)	_	1189.37	10^{-3}	1.26760	DOP [1]	7.99×10^{-11}
27	anti negative sigma baryon ($\overline{\Sigma^-}$)	~	1197.436	10^{-3}	1.26760	DOP [1]	1.479×10^{-10}
28	anti cascade; anti neutral xi baryon ($\overline{\Xi^0}$)	C	1314.9	10^{-3}	1.0	DOP [1]	2.9×10^{-10}
29	positive cascade; positive xi baryon (Ξ^+)	W	1321.32	10^{-3}	1.40820	DOP [1]	1.639×10^{-10}
30	anti omega ($\overline{\Omega^-}$)	@	1672.45	10^{-3}	1.78250	DOP [1]	8.22×10^{-11}
31	deuteron (d)	D	1875.627	10^{-3}	2.0	10^{29}	
32	triton (t)	T	2808.951	10^{-3}	3.0	12.3 years	
33	helion (^{3}He)	S	2808.421	10^{-3}	3.0	10^{29}	
34	alpha particle (α)	A	3727.418	10^{-3}	4.0	10^{29}	
35	negative pion (π^-)	*	139.56995	10^{-3}	0.14875	2.603×10^{-8}	
36	negative kaon (K^-)	?	493.677	10^{-3}	0.52614	1.2371×10^{-8}	
37	heavy ions [2]	#	varies	10^{-3}	5.0	10^{29}	

4.7 Input Error Messages

MCNP6 makes extensive checks of the input file for user errors. If the user violates a basic constraint of the input specification, a fatal error message is printed, both at the terminal and in the OUTP file. If a fatal input error is detected, MCNP6 will terminate before running any particles. The first fatal error is real; subsequent error messages may or may not be real because of the nature of the first fatal message. The `FATAL` option on the MCNP6 execution line instructs MCNP6 to ignore fatal errors and run particles, but the user should be extremely cautious when doing this. The `FATAL` does not apply to UM calculations [Chapter 8].

Most MCNP6 error messages are either warnings or comments that are not fatal. Warnings are intended to inform the user about unconventional input parameters or running conditions and may need further attention. Comments relay useful additional information to the user. The user should not ignore these messages but should understand their significance before making important calculations.

In addition to fatal, warning, comment, and deprecation messages, MCNP6 issues `BAD TROUBLE` messages immediately before any impending catastrophe, such as a divide by zero, which would otherwise cause the program to "crash." MCNP6 terminates as soon as the `BAD TROUBLE` message is issued. User input errors in the INP file are the most common reason for issuing a `BAD TROUBLE` message. These error messages indicate what corrective action is required.

Other output messages that may be encountered describe IEEE exception warnings after a calculation has finished. These usually indicate that exceptional arithmetic was performed during the calculation relative to [220] (e.g., invalid operations such as 0.0/0.0, division by zero, overflow, underflow, or inexact calculations that cannot be represented with infinite precision such as 2.0/3.0 or log(1.1)).

4.8 Geometry Errors

There is one important kind of input error that MCNP6 will not detect while processing data from the INP file. MCNP6 cannot detect overlapping cells or gaps between cells until a particle track actually gets lost. Even then the precise nature of the error may remain unclear. However, there is much that you can and should do to check your geometry before starting a long computer run.

Use the geometry-plotting feature of MCNP6 to look at the system from several directions and at various scales. Be sure that what you see is what you intend. Any gaps or overlaps in the geometry will probably show up as dashed lines. The intersection of a surface with the plot plane is drawn as a dashed line if there is not exactly one cell on each side of the surface at each point. Dashed lines can also appear if the plot plane happens to coincide with a plane of the problem, there are any cookie-cutter cells in the source, or there are DXTRAN spheres in the problem.

One way to check your geometry is to set up and run a short problem in which your system is flooded with particle tracks from an external source. The changes required in the INP file to perform this test follow:

1. Add a `VOID` card to override some of the other specifications in the problem and make all the cells voids, turn heating tallies into flux tallies, and turn off any tally multiplication (`FM`) cards.
2. Add another cell and a large spherical surface to the problem such that the surface surrounds the system and the old outside world cell is split by the new surface into two cells: the space between the system and the new surface, which is the new cell, and the space outside the new surface, which is now the outside world cell. Be sure that the new cell has non-zero importance. Actually, it is best to make all non-zero importance equal. If the system is infinite in one or two dimensions, use one or more planes instead of a sphere.

3. Replace the source specifications by an inward directed surface source to flood the geometry with particles. To do this, you can use the command

```
SDEF SUR=m NRM=-1
```

where m is the number of the new spherical surface added in Step 2. If the new surface is a plane, you must specify the portion to be used by means of `POS` and `RAD` or possibly `X`, `Y`, and `Z` source distributions.

Because there are no collisions, a short calculation will generate a great many tracks through your system. If there are any geometry errors, they should cause some of the particles to get lost.

When a particle first gets lost, whether in a special run with the `VOID` card or in a regular production run, the history is rerun to produce some special output on the `OUTP` file. Event-log printing is turned on during the rerun. The event log will show all surface crossings and will tell you the path the particle took to the bad spot in the geometry. When the particle again gets lost, a description of the situation at that point is printed. You can usually deduce the cause of the lost particle from this output. It is not possible to rerun lost particles in a multitasking run.

If the cause of the lost particle is still obscure, try plotting the geometry with the origin of the plot at the point where the particle got lost and with the horizontal axis of the plot plane along the direction the particle was moving. You might also consider turning `COLOR OFF` using the interactive geometry plotter. A wire drawing then appears, reducing the complexity of the visual representation caused by the color. The cause of the trouble is likely to appear as a dashed line somewhere in the plot or as some discrepancy between the plot and your idea of what it should look like.

Chapter 5

Input Cards

The MCNP input file contains entries that are commonly referred to as cards. Cards are usually structured to take a list of numbers or keyword-value pairs. This chapter describes each of the MCNP6 input cards. The overall file format is discussed in Chapter 4.

5.1 Geometry Specification Card Introduction

The geometry of MCNP6 treats an arbitrary three-dimensional configuration of user-defined materials in geometric cells bounded by first- and second-degree surfaces and fourth-degree elliptical tori. See Table 5.1. The cells are defined by the intersections, unions, and complements of the regions bounded by the surfaces. Surfaces are defined by supplying coefficients to the analytic surface equations or, for certain types of surfaces, known points on the surfaces. MCNP6 also provides a "macrobody" capability, where basic shapes such as spheres, boxes, cylinders, etc., may be combined using Boolean operators.

Each surface divides all space into two regions, one with positive sense with respect to the surface and the other with negative sense. Define $S = f(x, y, z) = 0$ as the equation of a surface in the problem. For any set of points (x, y, z) if $S = 0$, the points are on the surface; if S is negative, the points are said to have a negative sense with respect to that surface, and if S is positive, the points have a positive sense. The expression for a surface is the left side of the equation for the surface in Table 5.1. For the sphere, cylinder, cone, and torus, this definition is identical to defining the sense to be positive outside the figure. With planes normal to axes (`PX`, `PY`, or `PZ`), the definition gives positive sense for points with x, y, or z values exceeding the intercept of the plane. For the `P`, `SQ`, and `GQ` surfaces, the user supplies all of the coefficients for the expression and thus can determine the sense of the surface at will. This is different from the other cases where the sense, though arbitrary, is uniquely determined by the form of the expression. Therefore, in a surface transformation (see the `TR`*n* card) a `PX`, `PY`, or `PZ` surface will sometimes be replaced by a `P` surface just to prevent the sense of the surface from getting reversed.

The geometry of each cell is described on a cell card by a list of operators and signed surfaces that bound the cell. If the sense is positive, the "+" sign can be omitted. This geometry description defines the cell to be the intersection, union, and/or complement of the listed regions. The intersection operator in MCNP6 is implicit; it is simply the blank space between two signed surface numbers on the cell card. The union operator, signified by a colon (`:`), allows concave corners in cells and also cells that are completely disjoint. Because the intersection and union operators are binary Boolean operators, their use follows Boolean algebra methodology; unions and intersections can be used in combination in any cell description. Spaces on either side of the union operator are irrelevant, but a space without the colon signifies an intersection.

The complement operator, signified by the `#` symbol, provides no new capability over the intersection and union operators. It is just a shorthand cell-specifying method that implicitly uses the intersection and union operators. The complement operator can be thought of as standing for "not in." The notation `#`*n*, where *n* is a previously defined cell number, means that the description of the current cell is the complement of the

description of cell *n*. In other words, a number immediately after a complement operator, without parentheses, is interpreted as a cell number and is shorthand for the geometry specification of that cell number. The notation `#(...)`, where `(...)` is usually a list of surfaces describing another cell, means to complement the portion of the cell description in parentheses. Note that the symbol "#" is also used to denote heavy ions; however, the meaning of the symbol in the input file is obvious from how and where it is applied.

The default order of operations is complement first, intersection second, and unions third. There is no right-to-left ordering. Parentheses can be used to clarify operations and in some cases are required to force a certain order of operations. Innermost parentheses are cleared first. Spaces are optional on either side of a parenthesis. A parenthesis is equivalent to a space and signifies an intersection. Parentheses and operator symbols also function as delimiters; where they are present, blank delimiters are not necessary.

5.2 Cell Cards

Recommended precautions when creating cell definitions include the following:

1. Avoid excessively complicated cells. A problem geometry constructed of numerous simple cells runs faster than the same problem described using fewer, more complicated cells.
2. Avoid ineffective use of the complement operator, which can cause unneeded surfaces to be added to the geometry description of a cell. Extra surfaces make the problem run more slowly and may destroy the necessary conditions for volume and area calculations. See the example in §10.1.1.14.
3. Always use the geometry-plotting feature of MCNP6 to check the geometry of a problem [§6.2].
4. Flood the system with particles from an outside source to find errors in the geometry [§4.8].
5. If you add or remove cells, remember to change all the cell parameter cards accordingly. The difficulty of this can be reduced if the vertical format is used to specify values on the cell parameter cards [§4.4.5.2]. Alternatively, define cell-parameter values directly on cell cards and eliminate cell parameter cards entirely.

Form 1: *j m d geom params*

Form 2: *j* LIKE *n* BUT *list* (1)

j — Cell number assigned by the user.

Restriction: $1 \le j \le 99,999,999$

Restriction: If the cell is affected by a TRCL transformation, then *j* must be in the range $1 \le j \le 999$.

m — Material number if the cell is not a void.

Restriction: $1 \le m \le 99,999,999$

- $m > 0$ — the cell contains material m, which is specified on the M*m* card located in the data card section of the MCNP input file.
- $m = 0$ — the cell is a void.

d — Cell material density. This parameter is absent if the cell is a void.

- $d > 0$ — interpret the value as the atomic density in units of 10^{24} atoms/cm^3 (i.e., atoms/b-cm).

$d < 0$	interpret the value as the mass density in units of g/cm^3.

geom — Specification of the geometry of the cell. This specification consists of signed surface numbers and Boolean operators that specify how the regions bounded by the surfaces are to be combined. Boolean operators include the following:

<space>	indicates intersection,
:	indicates union; and
#	indicates complement.

params — Optional specification of cell parameters by entries in the *KEYWORD* = *value* form. Allowed keywords include IMP, VOL, PWT, EXT, FCL, WWN, DXC, NONU, PD, TMP, U, TRCL, LAT, FILL, ELPT, COSY, BFLCL, and UNC ((2),(3)).

n — Number of another cell.

Restriction: Cell card for cell *n* must appear in the MCNP input file before the cell card for cell *j*.

list — Set of *KEYWORD* = *value* specifications that define the attributes that differ between cells *n* and *j*. Allowed keywords include MAT (material number) and RHO (density) as well as the cell parameter keywords IMP, VOL, PWT, EXT, FCL, WWN, DXC, NONU, PD, TMP, U, TRCL, LAT, FILL, ELPT, COSY, BFLCL, and UNC .

Details:

(1) The LIKE *n* BUT feature is very useful in problems with many repeated structures. Cell j inherits from cell *n* the values of all attributes that are not specified in the list. The cell card for cell n must be before the cell card for cell j in the MCNP input file. The LIKE *n* BUT feature uses keywords for the cell material number and density. The mnemonics are MAT and RHO, respectively. These two keywords are only allowed following the LIKE *n* BUT construct, and may not appear in a normal cell description. Any other keyword name that appears after the BUT is a cell parameter and, therefore, must appear on cell cards only, not on any cards in the data block of the MCNP input file.

(2) Cell parameters may be defined on cell cards instead of in the data card section of the MCNP input file. If a cell parameter is entered on any cell card, a cell-parameter card with that name cannot be present, nor can the mnemonic appear on any vertical-format input card. It is permitted for some cell parameters to be specified on cell cards, while other subsets are specified in the data section. The format for cell parameters defined on cell cards is *KEYWORD* = *value(s)*, where the allowed keywords are IMP, VOL, PWT, EXT, FCL, WWN, DXC, NONU, PD, TMP, U, TRCL, LAT, FILL, ELPT, COSY, BFLCL, and UNC, with particle designators where necessary. The cell-parameter cards associated with the repeated structures capability, U, LAT, and FILL, may be placed either on the cell cards or in the data card section of the MCNP input file (see the U, LAT, and FILL cards).

(3) TMP and WWN data can be entered on cell cards in two ways. The *KEYWORD* = *value* form (TMP1=*value* TMP2=*value* ...) can be used or a special syntax is available where the single keyword TMP is followed by all the temperatures of the cell in an order corresponding to the times on the THTME card. The form for the WWN keyword is analogous: WWN1:*n* = *value* or WWN:*n* followed by all the lower weight bounds for the energy intervals of the cell.

Example 1

```
3  0    -1    2   -4          $ definition of cell 3
5  0    #3                    $ equivalent to each of the next 2 lines
```

or

```
5  0   #(-1  2  -4)
```

or

```
5  0   (+1 : -2 : +4)
```

Cell 3 is defined as the region in space with negative sense with respect to surface 1, positive sense with respect to surface 2, and negative sense with respect to surface 4. Cell 5 is the region of space not including cell 3. In the second and third lines of the example, it is specified using the complement operator; in the fourth line, the same region is specified using the union operator.

Example 2

```
2   3  -3.7     -1   IMP:N=2   IMP:P=4
3   LIKE 2 BUT       IMP:N=10  TRCL=1
```

This second example says that cell 3 is the same as cell 2 in every respect except that cell 3 has a different location (`TRCL = 1`) and a different neutron importance. The material in cell 3, the density, and the definition are the same as cell 2 and the photon importance is the same.

Example 3

```
10  16  -4.2   1   -2    3   IMP:N=4  IMP:P=8  EXT:N=-0.4X
```

This says that cell 10 is to be filled with material 16 at a density of 4.2 g/cm^3. The cell consists of the intersections of the regions on the positive side of surface 1, the negative side of surface 2, and the positive side of surface 3. The neutron importance in cell 10 is 4 and the photon importance is 8. Neutrons in cell 10 are subject to an exponential transform in the $-x$ direction with stretching parameter 0.4.

5.3 Surface Cards

5.3.1 Surfaces Cards, Defined by Equations

The available surface types, equations, mnemonics, and the order of the card entries are given in Table 5.1. To specify a surface by this method, find the surface in Table 5.1 and determine the coefficients for the equation. The information is entered on the surface card according to the following format:

Form: *j n A list*

j	Surface number assigned by the user. Restriction: $1 \leq j \leq 99,999,999$ Restriction: If the surface is affected by a TR transformation, then *j* must be in the range $1 \leq j \leq 999$ [§5.5.3 and §5.5.4].
**j*	Reflecting surface number. A particle track that hits a reflecting surface is reflected specularly (1)
+j	White boundary surface number. A particle hitting a white boundary is reflected with a cosine distribution relative to the surface normal (1)
n	Transformation number. If *n* is absent then no coordinate transformation is specified.
$n > 0$	the value specifies a transformation number *n* of a TR*n* card.
$n < 0$	the value specifies that surface *j* is periodic with surface *n* (2).
A	Equation mnemonic from Table 5.1 that specifies the type of surface.
list	One to ten numerical entries, as required to define the selected surface.

In addition, using the X, Y, Z, and P mnemonics a surface can be defined based on points [§5.3.2 and §5.3.3]. Finally, macrobodies can be used to conveniently define surfaces [§5.3.4].

Details:

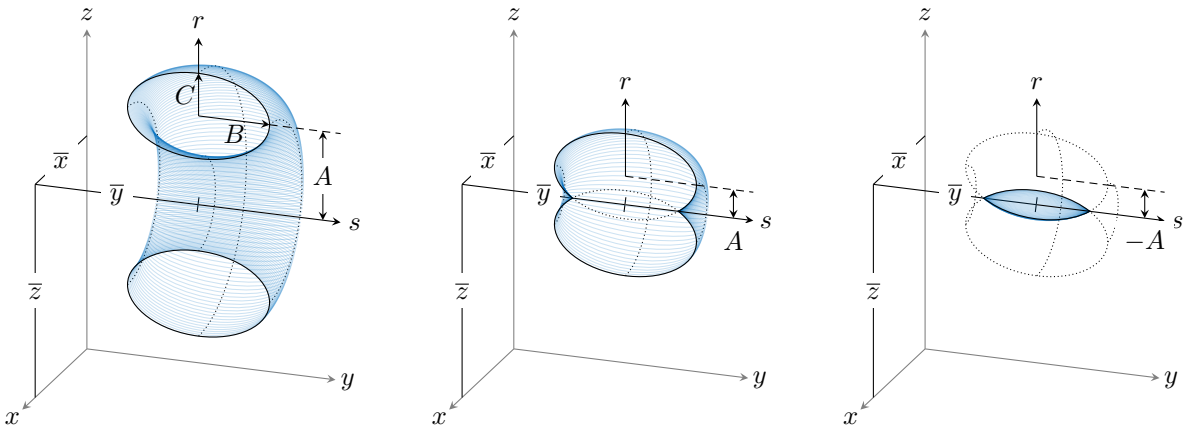
1. Detectors and DXTRAN (next-event estimators) usually should not be used in problems that have reflecting surfaces or white boundaries. Also, tallies in problems with reflecting surfaces will need to be normalized differently as discussed in §2.2.3.3 and §2.5.6.4.2.

2. If periodic boundaries are specified (i.e., $n < 0$) such that surface *j* is periodic with surface *n*, the following restrictions apply:
   (a) Surfaces *j* and *n* must be planes.
   (b) No surface transformation is allowed for the periodic planes.
   (c) The periodic cell(s) can be infinite or bounded by planes on the top and bottom that can be reflecting or white, but cannot be periodic.
   (d) Periodic planes can bound only other periodic planes or top and bottom planes.
   (e) A single zero-importance cell must be on one side of each periodic plane.
   (f) All periodic planes must have a common rotational vector normal to the geometry top and bottom.
   (g) Next-event estimators such as detectors and DXTRAN should not be used.

3. The quadratic equation for a cone describes a cone of two sheets—one sheet is a cone of positive slope, and the other has a negative slope. MCNP6 provides the option to select either of the two sheets. The +1 or the -1 entry on the cone surface card causes the one sheet cone treatment to be used (and is only used for single-sheet cones). If the sign of the entry is positive, the specified sheet is the one that extends to infinity in the positive direction of the coordinate axis to which the cone axis is parallel. The converse is true for a negative entry. A cell whose description contains a two-sheeted cone may require an additional

Table 5.1: MCNP6 Surface Cards

Type	Mnemonic	Description	Equation	Card Entries
Plane	P	General	$Ax + By + Cz - D = 0$	A, B, C, D
	PX	Normal to x axis	$x - D = 0$	D
	PY	Normal to y axis	$y - D = 0$	D
	PZ	Normal to z axis	$z - D = 0$	D
Sphere	SO	Centered at Origin	$x^2 + y^2 + z^2 - R = 0$	R
	S	General	$(x-\overline{x})^2 + (y-\overline{y})^2 + (z-\overline{z})^2 - R^2 = 0$	$\overline{x}, \overline{y}, \overline{z}, R$
	SX	Centered on x axis	$(x-\overline{x})^2 + y^2 + z^2 - R^2 = 0$	$\overline{x}, R$
	SY	Centered on y axis	$x^2 + (y-\overline{y})^2 + z^2 - R^2 = 0$	$\overline{y}, R$
	SZ	Centered on z axis	$x^2 + y^2 + (z-\overline{z})^2 - R^2 = 0$	$\overline{z}, R$
Cylinder	C/X	Parallel to x axis	$(y-\overline{y})^2 + (z-\overline{z})^2 - R^2 = 0$	$\overline{y}, \overline{z}, R$
	C/Y	Parallel to y axis	$(x-\overline{x})^2 + (z-\overline{z})^2 - R^2 = 0$	$\overline{x}, \overline{z}, R$
	C/Z	Parallel to z axis	$(x-\overline{x})^2 + (y-\overline{y})^2 - R^2 = 0$	$\overline{x}, \overline{y}, R$
	CX	On x axis	$y^2 + z^2 - R^2 = 0$	R
	CY	On y axis	$x^2 + z^2 - R^2 = 0$	R
	CZ	On z axis	$x^2 + y^2 - R^2 = 0$	R
Cone (3, 4)	K/X	Parallel to x axis	$\sqrt{(y-\overline{y}^2) + (z-\overline{z}^2)} - t(x-\overline{x}) = 0$	$\overline{x}, \overline{y}, \overline{z}, t^2, \pm 1$
	K/Y	Parallel to y axis	$\sqrt{(x-\overline{x}^2) + (z-\overline{z}^2)} - t(y-\overline{y}) = 0$	$\overline{x}, \overline{y}, \overline{z}, t^2, \pm 1$
	K/Z	Parallel to z axis	$\sqrt{(x-\overline{x}^2) + (y-\overline{y}^2)} - t(z-\overline{z}) = 0$	$\overline{x}, \overline{y}, \overline{z}, t^2, \pm 1$
	KX	On x axis	$\sqrt{y^2 + z^2} - t(x-\overline{x}) = 0$	$\overline{x}, t^2, \pm 1$
	KY	On y axis	$\sqrt{x^2 + z^2} - t(y-\overline{y}) = 0$	$\overline{y}, t^2, \pm 1$
	KZ	On z axis	$\sqrt{x^2 + y^2} - t(z-\overline{z}) = 0$	$\overline{z}, t^2, \pm 1$
Ellipsoid Hyperboloid Paraboloid	SQ	Axes parallel to x, y, or z axis	$A(x-\overline{x})^2 + B(y-\overline{y})^2 + C(z-\overline{z})^2$ $+2D(x-\overline{x}) + 2E(y-\overline{y})$ $+2F(z-\overline{z}) + G = 0$	A, B, C, D, E, F, G, $\overline{x}$, $\overline{y}$, $\overline{z}$
Cylinder Cone Ellipsoid Hyperboloid Paraboloid	GQ	Axes not parallel to x, y, or z axis	$Ax^2 + By^2 + Cz^2 + Dxy + Eyz$ $+Fzx + Gx + Hy + Jz + K = 0$	A, B, C, D, E, F, G, H, J, K
Torus (5)	TX	Axis parallel to x, y, or z axis	$\frac{(x-\overline{x})^2}{B} + \frac{\left(\sqrt{(y-\overline{y})^2 + (z-\overline{z})^2} - A\right)^2}{C^2} - 1 = 0$	$\overline{x}, \overline{y}, \overline{z}, A, B, C$
	TY		$\frac{(y-\overline{y})^2}{B} + \frac{\left(\sqrt{(x-\overline{x})^2 + (z-\overline{z})^2} - A\right)^2}{C^2} - 1 = 0$	$\overline{x}, \overline{y}, \overline{z}, A, B, C$
	TZ		$\frac{(z-\overline{z})^2}{B} + \frac{\left(\sqrt{(x-\overline{x})^2 + (y-\overline{y})^2} - A\right)^2}{C^2} - 1 = 0$	$\overline{x}, \overline{y}, \overline{z}, A, B, C$



(a) Ring Torus

(b) Degenerate Torus, $0 < A < C$

(c) Degenerate Torus, $A < 0 < C$

Figure 5.1: Elliptical Tori

surface specification to help distinguish between the two sheets. This ambiguity surface helps to eliminate any ambiguities as to which region of space is included in the cell.

④ The value t^2 entered to define the angle of the surface relative to the axis of the cone is $t^2 = \tan^2(\theta) = (r/h)^2$ where θ is the angle in radians and r is the radius of the cone at distance h along the axis of the cone from its apex. The relationship among these parameters is shown in Fig. 5.2.

⑤ The TX, TY, and TZ mnemonics represent elliptical tori (fourth degree surfaces) rotationally symmetric about axes parallel to the x, y, and z axes, respectively. A TY torus is illustrated in Fig. 5.1a. Note that the input parameters $\overline{x}, \overline{y}, \overline{z}, A, B, C$ specify the ellipse

$$\frac{s^2}{B^2} + \frac{(r-A)^2}{C^2} = 1 \tag{5.1}$$

rotated about the s axis in the (r, s) cylindrical coordinate system (Fig. 5.1a) whose origin is at in the (x, y, z) system. In the case of a TY torus,

$$s = (y - \overline{y}) \tag{5.2}$$

and

$$r = \sqrt{(x-\overline{x})^2 + (z-\overline{z})^2}. \tag{5.3}$$

A torus is degenerate if $|A| < C$ where $0 < A < C$ produces the outer surface (Fig. 5.1b), and $-C < A < 0$ produces the inner surface (Fig. 5.1c).

Coordinate transformations for tori are limited to those in which each axis of the auxiliary coordinate system is parallel to an axis of the main system.

⚠ Caution

MCNP6 may incorrectly compute the internal volume of tori that exhibit a large ratio of major to minor axes. A warning message is printed when the ratio of the major to minor axes exceeds 2000.

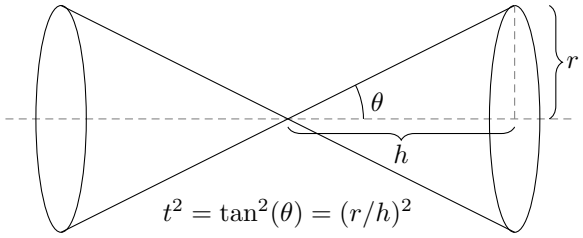


Figure 5.2: Cone Parameters

5.3.1.1 Example 1

```
1     PY  3
```

Surface 1 describes a plane normal to the y axis at $y = 3$ with positive sense for all points with $y > 3$.

5.3.1.2 Example 2

```
3     K/Y  0   0   2   0.25   1
```

Surface 3 is a cone whose vertex is at $(x, y, z) = (0, 0, 2)$ and whose axis is parallel to the y axis. The cone has height $h = 100$ and radius $r = 50$ leading to $t^2 = \tan^2(\theta) = (r/h)^2 = 0.25$ (where θ is the angle between the surface and the cone's axis) and only the positive (right hand) sheet of the cone is used. Points outside the cone have a positive sense.

5.3.1.3 Example 3

```
11     GQ  1  0.25  0.75   0   -0.866
           0  -12   -2   3.464   39
```

This is a cylinder of radius 1 cm whose axis is in a plane normal to the x axis at $x = 6$, displaced 2 cm from the x axis and rotated 30 degrees about the x axis off the y axis toward the z axis. The sense is positive for points outside the cylinder. Such a cylinder would be much easier to specify by first defining it in an auxiliary coordinate system where it is symmetric about a coordinate axis and then using the TR*n* card to define the relationship between the basic and auxiliary coordinate systems. The input would then be

```
11  7  CX  1

*TR7    6  1  -1.732   0  30  60
```

5.3.2 Axisymmetric Surfaces Defined by Points

Surface cards of the type X, Y, Z, and P can be used to describe surfaces by coordinate points rather than by equation coefficients as in the previous section. The surfaces described by these cards must be symmetric about the x, y, or z axis, respectively, and, if the surface consists of more than one sheet, the specified coordinate points must all be on the same sheet.

Each of the coordinate pairs defines a geometric point on the surface. On the Y card, for example, the entries may be *j* Y *y1 r1 y2 r2* where $ri = \sqrt{xi^2 + zi}$ and *yi* is the coordinate of point i. If one coordinate pair is used, a plane (PX, PY, or PZ) is defined. If two coordinate pairs are used, a linear surface (PX, PY, PZ, CX, CY, CZ, KX, KY, or KZ) is defined. If three coordinate pairs are used, a quadratic surface (PX, PY, PZ, SO, SX, SY, SZ, CX, CY, CZ, KX, KY, KZ, or SQ) is defined. Note that planes and linear surfaces are degenerate quadratic surfaces, which is why they are listed multiple times.

When a cone is specified by two points, a cone of only one sheet is generated.

The senses of these surfaces (except SQ) are determined by the code to be identical to the senses one would obtain by specifying the surface by equations. For SQ, the sense is defined so that points sufficiently far from the axis of symmetry have positive sense. Note that this is different from the equation-defined SQ, where the user could choose the sense freely through the sign of the coefficient G.

Form: *j n A list*

j	Surface number assigned by the user. Restriction: $1 \leq j \leq 99,999,999$ Restriction: If the surface is affected by a TR transformation or in a repeated structure, then *j* must be in the range $1 \leq j \leq 999$ [§5.5.3 and §5.5.4].	
n	Transformation number. If *n* is absent then no coordinate transformation is specified.	
	$n > 0$	the value specifies a transformation number *n* of a TR*n* card.
	$n < 0$	the value specifies that surface *j* is periodic with surface *n*. (See Note 2.)
A	The letter X, Y, or Z.	
list	One to three coordinate pairs.	

5.3.2.1 Example 1

```
12   X   7  5   3  2   4  3
```

This input describes a surface symmetric about the x axis, which passes through the three (x, r) points $(7, 5)$, $(3, 2)$, and $(4, 3)$. This surface is a hyperboloid of two sheets, converted in MCNP6 to its equivalent

```
12   SQ   -0.083333333 1 1 0 0 0 68.52083 -26.5 0 0
```

5.3.2.2 Example 2

```
12   Y   1  2   1  3   3  4
```

These data describe two parallel planes at $y = 1$ and $y = 3$ and is a fatal error because the requirement that all points be on the same sheet is not met.

5.3.2.3 Example 3

```
12   Y   3  0   4  1   5  0
```

This input describes a 1-cm-radius sphere with center at $(x, y, z) = (0, 4, 0)$.

5.3.2.4 Example 4

```
12   Z   1  0   2  1   3  4
```

This surface is rejected because the points are on two different sheets of the hyperboloid

$$x^2 + y^2 - 7z^2 + 20z - 13 = 0. \tag{5.4}$$

However, the surface

```
12   Z   2  1   3  4   5  9.380832
```

which has the same surface equation as above is accepted because all coordinates lie on a single surface: the right sheet of the hyperboloid.

5.3.2.5 Example 5

Listing 5.1: example_axisym_surf.mcnp.inp.txt

```
example 5
1   0      1 -2   3
2   0      #1

1   Y     -3  2   2  1          $ cone
2   Y      2  3   3  3   4  2   $ ellipsoid
3   Y      2  1   3  1   4  2   $ hyperboloid

sdef
imp:n 1 0
nps 1
```

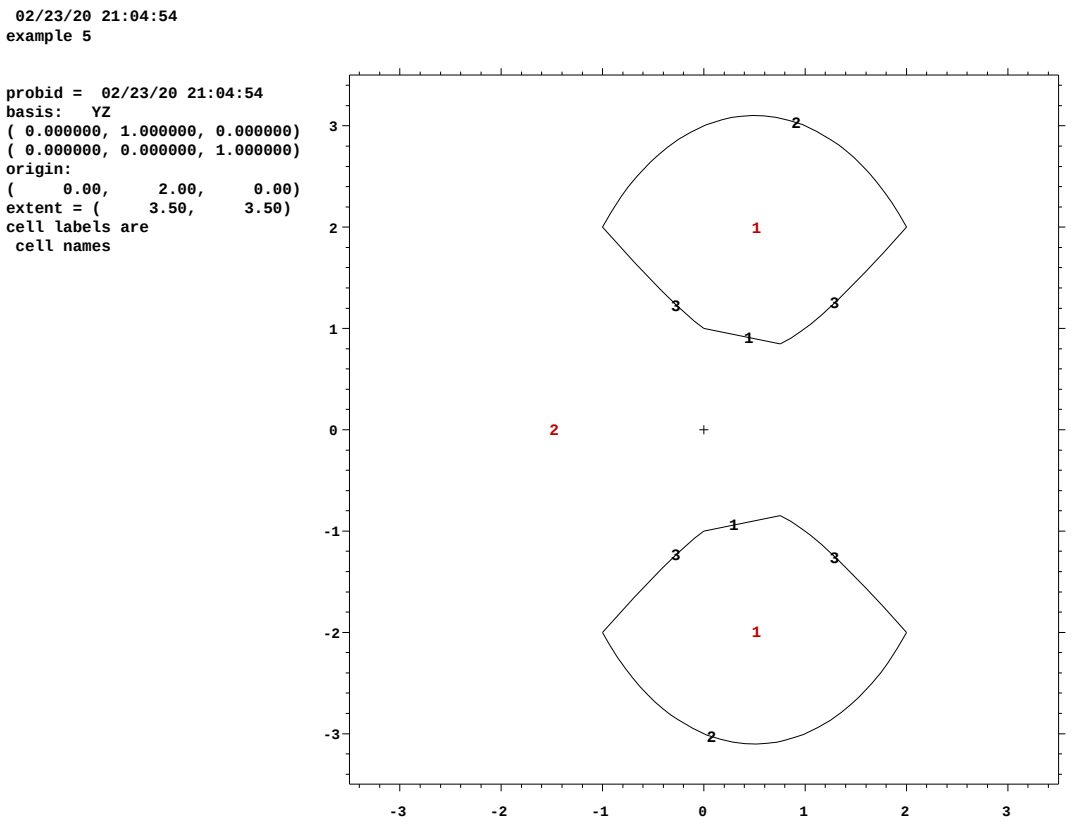


Figure 5.3: A geometry plot of Cell 1 of Example 5.

The final example in Listing 5.1 defines a cell bounded by a cone, hyperboloid, and an ellipsoid. The three surfaces define the donut-like cell that is symmetric about the y axis. A cross section of this cell is seen in Fig. 5.3. To plot this view, use the interactive plotter command input file in Listing 5.2.

Listing 5.2: example_axisym_surf.mcnp.comin.txt

```
label 1 1 cel or 0 2 0 ex 3.5 scale 1
```

One surface goes through the points $(-3, 2)$ and $(2, 1)$. The second surface goes through $(2, 3)$, $(3, 3)$, and $(4, 2)$. The last surface is defined by the points $(2, 1)$, $(3, 1)$, and $(4, 2)$. These coordinate points are in the form (y, r). Using these cards, MCNP6 indicates that surface 1 is a cone of one sheet, surface 2 is an ellipsoid, and surface 3 is a hyperboloid of one sheet. The equation coefficients for the standard surface equations are printed out for the various surfaces when the PRINT input card or execution option is used. For example, an SQ surface defining surface 3 is

```
3 SQ 1 -1.5 1 0 0 0 -0.625 0 2.5 0
```

5.3.3 General Plane Defined by Three Points

If there are four entries on a surface card with a P mnemonic, they are assumed to be the general plane equation coefficients as in Table 5.1. If there are more than four entries, they give the coordinates of three points lying in the desired plane. The code uses the coordinate points to determine the required surface coefficients to produce the plane

$$Ax + By + Cz - D = 0. \tag{5.5}$$

The sense of the plane is determined by requiring the origin to have negative sense. If the plane passes through the origin ($D = 0$), the point $(0, 0, \infty)$ has positive sense. If this fails ($D = C = 0$), the point $(0, \infty, 0)$ has positive sense. If this fails ($D = C = B = 0$), the point $(\infty, 0, 0)$ has positive sense. If this fails, the three points lie in a line and a fatal error is issued.

Form: *j n P x1 y1 z1 x2 y2 z2 x3 y3 z3*

j	Surface number assigned by the user. Restriction: $1 \le j \le 99,999,999$ Restriction: $1 \le j \le 999$ if *j* is the surface number of a repeated structure or if surface *j* defines a surface transformed with TR.
n	If *n* is absent, then no coordinate transformation is specified. $n > 0$ specifies transformation number *n* of a TR*n* card. $n < 0$ specifies surface *j* is periodic with surface *n*.
P	Mnemonic that indicates this is a planar surface [Table 5.1].
xi yi zi	Coordinates of three points that define the plane.

5.3.4 Surfaces Defined by Macrobodies

Using a combinatorial-geometry-like macrobody capability is an alternative method of defining cells and surfaces. The combinatorial geometry bodies available are similar to those in the Integrated Tiger Series

(ITS) [57] codes. The macrobodies can be mixed with the standard cells and surfaces. The macrobody surface is decomposed internally by MCNP6 into surface equations and the facets are assigned individual numbers according to a predetermined sequence. The assigned numbers are the number selected by the user followed by a decimal point and $1, 2, \ldots$. The facets can be used for tallying, tally segmentation, other cell definitions, SDEF sources, etc. They cannot be used on the surface source read and write cards (SSR/SSW), the surface flagging card (SF), non-HDF5 PTRAC files, or MCTAL files.

The space inside a macrobody has a negative sense with respect to the macrobody surface and all its facets. The space outside a body has a positive sense. The sense of a facet is the sense assigned to it by the macrobody "master" cell and the facet retains that assigned sense if it appears in other cell descriptions and must be properly annotated. More information regarding facets is provided in §5.3.4.11.

5.3.4.1 BOX: Arbitrarily Oriented Orthogonal Box

BOX *vx vy vz a1x a1y a1z a2x a2y a2z a3x a3y a3z*

vx vy vz	The (x, y, z) coordinates of a corner of the box.
a1x a1y a1z	Vector of first side from the specified corner coordinates.
a2x a2y a2z	Vector of second side from the specified corner coordinates.
a3x a3y a3z	Vector of third side from the specified corner coordinates.

Details:

1. All corner angles are 90°.
2. If (*a3x*, *a3y*, *a3z*) is not specified, the box will be infinite along the vector normal to the plane specified by (*a1x*, *a1y*, *a1z*) and (*a2x*, *a2y*, *a2z*).

An example $1 \times 2 \times 3$-cm box, centered about $(0, 0, 0)$ with sides normal to the x, y, and z axes, is given in Listing 5.3.

Listing 5.3: example_macrobodies.mcnp.inp.txt

```
1010 box  -0.5 -1 -1.5    1 0 0    0 2 0    0 0 3
```

5.3.4.2 RPP: Rectangular Parallelepiped

RPP *xmin xmax ymin ymax zmin zmax*

xmin xmax	Termini of box sides normal to the x axis.
ymin ymax	Termini of box sides normal to the y axis.
zmin zmax	Termini of box sides normal to the z axis.

Details:

1. RPP surfaces will only be normal to the x, y, and z axes.
2. The x, y, and z values are relative to the origin.
3. If *xmin* = *xmax*, *ymin* = *ymax*, or *zmin* = *zmax*, the rectangular parallelepiped will be infinite in that dimension. Only one dimension may be infinite at a time.

An example $1 \times 2 \times 3$-cm rectangular parallelepiped, centered about $(1.5, 0, 0)$ with sides normal to the x, y, and z axes, is given in Listing 5.4. This RPP specification is comparable (other than absolute position) to the BOX example in Listing 5.3.

Listing 5.4: example_macrobodies.mcnp.inp.txt

```
1020 rpp   1 2    -1 1    -1.5 1.5
```

5.3.4.3 SPH: Sphere

SPH *vx vy vz r*

vx vy vz	The (x, y, z) coordinates of the center of the sphere.
r	Radius of sphere.

An example 20-cm radius sphere, centered about $(0, 0, 0)$, is given in Listing 5.5.

Listing 5.5: example_macrobodies.mcnp.inp.txt

```
1030 sph   0 0 0    20
```

5.3.4.4 RCC: Right Circular Cylinder

RCC *vx vy vz h1 h2 h3 r*

vx vy vz	The (x, y, z) coordinates at the center of the base for the right circular cylinder.
h1 h2 h3	Right circular cylinder axis vector, which provides both the orientation and the height of the cylinder.
r	Radius of cylinder.

An example 0.5-cm radius right-circular cylinder aligned parallel to the z axis, centered about $(3, 0, 0)$ and with a length of 3 cm, is given in Listing 5.6.

Listing 5.6: example_macrobodies.mcnp.inp.txt

```
1040 rcc   3.0 0 -1.5    0 0 3    0.50
```

5.3.4.5 RHP or HEX: Right Hexagonal Prism

RHP *vx vy vz h1 h2 h3 r1 r2 r3 s1 s2 s3 t1 t2 t3*
or
HEX *vx vy vz h1 h2 h3 r1 r2 r3 s1 s2 s3 t1 t2 t3*

vx vy vz	The (x, y, z) coordinates at the center of the bottom of the hexagonal prism.
h1 h2 h3	Vector from the bottom to the top of the hexagonal prism. For a z hex with height h, $(h1, h2, h3) = (0, 0, h)$.
r1 r2 r3	Vector from the axis to the center of the first facet. For a pitch $2p$ facet normal to y axis, $(r1, r2, r3) = (0, p, 0)$.
s1 s2 s3	Vector to center of the second facet. This is optional for a regular hexagon but required for an irregular hexagon.
t1 t2 t3	Vector to center of the third facet. This is optional for a regular hexagon but required for an irregular hexagon.

Details:

1. The right-hexagonal prism in the MCNP code differs from the ITS-ACCEPT [57] format.
2. One can make an infinite right-hexagonal prism by setting the length of the vector $(h1, h2, h3)$ greater than or equal to 10^6 cm. Surfaces 7 and 8 in Table 5.2 will then not be created.

An example regular right-hexagonal prism using the RHP keyword aligned parallel to the z axis, centered about $(4.5, 0, 0)$ with a length of 3 cm and first-facet offset of 0.5-cm normal to the y axis, is given in Listing 5.7. An example regular right hexagonal prism using the HEX keyword aligned parallel to the z axis, centered about $(6, 0, 0)$ with a length of 3 cm and first-facet offset of 0.5-cm normal to the x axis, is also given. Two examples of irregular hexagons are also given.

Listing 5.7: example_macrobodies.mcnp.inp.txt

```
1050 rhp   4.5 0 -1.5    0 0 3    0 0.5 0
1051 hex   6.0 0 -1.5    0 0 3    0.5 0 0
1052 hex   6.0 2 -1.5    0 0 3    0.5 0 0   0.4 0.69282 0.0  -0.4 0.69282 0.0
1053 hex   6.0 4 -1.5    0 0 3    0.5 0 0   0.4 0.69282 0.0  -0.5 0.85    0.0
```

5.3.4.6 REC: Right Elliptical Cylinder

REC *vx vy vz h1 h2 h3 v1x v1y v1z v2x v2y v2z*

vx vy vz	The (x, y, z) coordinates of the cylinder bottom.
h1 h2 h3	Cylinder axis height vector.
v1x v1y v1z	Ellipse minor axis vector, which is normal to $(h1, h2, h3)$.
v2x v2y v2z	Ellipse major axis vector, which is orthogonal to vectors $(h1, h2, h3)$ and $(v1x, v1y, v1z)$.

Details:

1. If there are 10 entries instead of 12, the 10th entry is the minor axis radius, where the direction is determined from the cross product of (*h1*, *h2*, *h3*) and (*v1x*, *v1y*, *v1z*).

An example right-elliptical cylinder aligned parallel to the z axis, centered about $(7.5, 0, 0)$ with an axial length of 3 cm and 0.25- and 0.5-cm minor and major axes along the x and y axes, respectively, is given in Listing 5.8.

Listing 5.8: example_macrobodies.mcnp.inp.txt

```
1060 rec   7.5 0 -1.5    0 0 3    0.25 0 0   0 0.5 0
```

5.3.4.7 TRC: Truncated Right-angle Cone

TRC *vx vy vz h1 h2 h3 r1 r2*

vx vy vz	The (x, y, z) coordinates of the cone bottom.
h1 h2 h3	Cone axis height vector.
r1	Radius of lower cone base.
r2	Radius of upper cone base.

An example truncated cone aligned parallel to the z axis, with its base centered at $(9, -1, 0)$ with a length of 2 cm, lower-base radius of 0.15 cm, and upper-base radius of 0.5 cm, is given in Listing 5.9. An example truncated cone aligned parallel to the z axis, with its base centered at $(10.5, -1, 0)$ with a length of 2 cm, lower-base radius of 0.5 cm, and upper-base radius of 0.15 cm, is also given.

Listing 5.9: example_macrobodies.mcnp.inp.txt

```
1070 trc   9.0 -1 0    0 2 0    0.15   0.5
1071 trc  10.5 -1 0    0 2 0    0.5    0.15
```

5.3.4.8 ELL: Ellipsoid

ELL *v1x v1y v1z v2x v2y v2z r*

v1x v1y v1z	Coordinates determined by sign of *r*: $r > 0$ the coordinates of the first focus. $r < 0$ the coordinates of the center of the ellipsoid.
v2x v2y v2z	Coordinates determined by sign of *r*: $r > 0$ the coordinates of the second focus. $r < 0$ major axis vector (vector from the center of the ellipsoid through a focus to the vertex, where the length is equal to the major radius).
r	Radius based on sign:

$r > 0$ major radius length.

$r < 0$ minor radius length.

Details:

1. The major and minor radii are half the lengths of the major and minor axes, respectively.
2. The ellipsoid macrobody is a surface of revolution about the major axis, but the major radius may be smaller than the minor radius.

An example ellipsoid aligned parallel to the y axis, centered about $(12, 0, 0)$ with a major-axis distance of 2 cm and 0.5-cm minor radius length, is given in Listing 5.10. An example ellipsoid aligned parallel to the y axis, centered about $(13.5, 0, 0)$ with a distance of 1 cm between foci and 0.75-cm major radius length, is also given.

Listing 5.10: example_macrobodies.mcnp.inp.txt

```
1080 ell  12.0    0 0    0      1 0    -0.5
1081 ell  13.5 -0.5 0    13.5 0.5 0    0.75
```

5.3.4.9 WED: Wedge

WED *vx vy vz v1x v1y v1z v2x v2y v2z v3x v3y v3z*

vx vy vz	The (x, y, z) coordinates of wedge vertex.
v1x v1y v1z	Vector of first side of triangular base.
v2x v2y v2z	Vector of second side of triangular base.
v3x v3y v3z	Height vector.

Details:

1. A right-angle wedge has a right triangle for a base defined by $(v1x, v1y, v1z)$ and $(v2x, v2y, v2z)$ and a height $(v3x, v3y, v3z)$.
2. The vectors $(v1x, v1y, v1z)$, $(v2x, v2y, v2z)$, and $(v3x, v3y, v3z)$ are orthogonal to each other.

An example right-angle wedge aligned parallel to the z axis, about $(15, 0, 0)$ with an axial length of 3 cm, a first-side length of 2 cm and a second-side length of 0.5-cm, is given in Listing 5.11.

Listing 5.11: example_macrobodies.mcnp.inp.txt

```
1090 wed  14.75 -1 -1.5    0 2 0    0.5 0 0    0 0 3
```

5.3.4.10 ARB: Arbitrary Polyhedron

There must be eight triplets of entries input for the ARB to describe the (x, y, z) coordinates of the corners, although some may not be used (just use triplets of zeros). These are followed by six more entries, n_i, which follow a prescribed convention: each entry is a four-digit integer that defines a side of the ARB in terms of the corners for the side.

For example, the entry 1278 would define this plane surface to be bounded by the first, second, seventh, and eighth triplets (or equivalently, corners). Because three points are sufficient to determine the plane, only the first, second, and seventh corners would be used in this example to determine the plane. The distance from the plane to the fourth corner (corner 8 in the example) is determined by MCNP6. If the absolute value of this distance is greater than 10^{-6} cm, an error message is given and the distance is printed in the MCNP output file along with the (x, y, z) that would lie on the plane. If the fourth digit is zero, the fourth point is ignored. For a four-sided ARB, four non-zero four-digit integers (last digit is zero for four-sided since there are only three corners for each side) are required to define the sides. For a five-sided ARB, five non-zero four-digit integers are required, and six non-zero four-digit integers are required for a six-sided ARB. Since there must be 30 entries altogether for an ARB (or MCNP6 gives an error message), the last two integers are zero for the four-sided ARB and the last integer is zero for a five-sided ARB.

ARB *ax ay az bx by bz ... n1 n2 n3 n4 n5 n6*

ax ay az	The (x, y, z) coordinates of first corner of the polyhedron.
bx by bz	The (x, y, z) coordinates of second corner of the polyhedron.
cx cy cz	The (x, y, z) coordinates of third corner of the polyhedron.
dx dy dz	The (x, y, z) coordinates of fourth corner of the polyhedron.
ex ey ez	The (x, y, z) coordinates of fifth corner of the polyhedron.
fx fy fz	The (x, y, z) coordinates of sixth corner of the polyhedron.
gx gy gz	The (x, y, z) coordinates of seventh corner of the polyhedron.
hx hy hz	The (x, y, z) coordinates of eighth corner of the polyhedron.
ni	Four-digit numbers describing a side of the polyhedron in terms of its corresponding corners. For example, *n1* = 1278 is a plane/side bounded by corners 1, 2, 7, and 8 (points *a*, *b*, *g*, and *h*).

Details:

(1) There must be eight (x, y, z) triplets to describe the eight corners of the polyhedron.

An example $1 \times 2 \times 3$-cm rectangular parallelepiped, centered about $(16, 0, 0)$ with sides normal to the x, y, and z axes, is given in Listing 5.12. This ARB specification is equivalent to the BOX and RPP examples in Listings 5.3 and 5.4, respectively.

Listing 5.12: example_macrobodies.mcnp.inp.txt

```
1100 arb  15.5 -1 -1    16.5 -1 -1    16.5  1 -1    15.5  1 -1
          15.5 -1  1    16.5 -1  1    16.5  1  1    15.5  1  1
          1458 2367     1256 3478     1234 5678
```

Table 5.2: Macrobody Facet Descriptions

Macrobody Type	Facet Number	Facet Description
BOX	1	Plane normal to end of (*a1x*, *a1y*, *a1z*)
	2	Plane normal to beginning of (*a1x*, *a1y*, *a1z*)
	3	Plane normal to end of (*a2x*, *a2y*, *a2z*)
	4	Plane normal to beginning of (*a2x*, *a2y*, *a2z*)
	5	Plane normal to end of (*a3x*, *a3y*, *a3z*)
	6	Plane normal to beginning of (*a3x*, *a3y*, *a3z*)
RPP	1	Plane *xmax*
	2	Plane *xmin*
	3	Plane *ymax*
	4	Plane *ymin*
	5	Plane *zmax*
	6	Plane *zmin*
SPH		Treated as a regular surface so no facet
RCC	1	Cylindrical surface of radius *r*
	2	Plane normal to end of (*h1*, *h2*, *h3*)
	3	Plane normal to beginning of (*h1*, *h2*, *h3*)
RHP or HEX	1	Plane normal to end of (*r1*, *r2*, *r3*)
	2	Plane opposite facet 1
	3	Plane normal to end of (*s1*, *s2*, *s3*)
	4	Plane opposite facet 3
	5	Plane normal to end of (*t1*, *t2*, *t3*)
	6	Plane opposite facet 5
	7	Plane normal to end of (*h1*, *h2*, *h3*)
	8	Plane normal to beginning of (*h1*, *h2*, *h3*)
REC	1	Elliptical cylinder
	2	Plane normal to end of (*h1*, *h2*, *h3*)
	3	Plane normal to beginning of (*h1*, *h2*, *h3*)
TRC	1	Conical surface
	2	Plane normal to end of (*h1*, *h2*, *h3*)
	3	Plane normal to beginning of (*h1*, *h2*, *h3*)
ELL		Treated as a regular surface so no facet
WED	1	Slant plane including top and bottom hypotenuses
	2	Plane including vectors (*v2x*, *v2y*, *v2z*) and (*v3x*, *v3y*, *v3z*)
	3	Plane including vectors (*v1x*, *v1y*, *v1z*) and (*v3x*, *v3y*, *v3z*)
	4	Plane including vectors (*v1x*, *v1y*, *v1z*) and (*v2x*, *v2y*, *v2z*) at end of (*v3x*, *v3y*, *v3z*) (top triangle)
	5	Plane including vectors (*v1x*, *v1y*, *v1z*) and (*v2x*, *v2y*, *v2z*) at beginning of (*v3x*, *v3y*, *v3z*) (bottom triangle)
ARB	1	Plane defined by corners *n1*
	2	Plane defined by corners *n2*
	3	Plane defined by corners *n3*
	4	Plane defined by corners *n4*
	5	Plane defined by corners *n5*
	6	Plane defined by corners *n6*

5.3.4.11 Macrobody Facets

The facets of the macrobodies are numbered sequentially and can be used on other MCNP6 cards. BOX and RPP can be infinite in a dimension, in which case those two facets are skipped and the numbers of the remaining facets are decreased by two. RHP can be infinite in the axial dimension in which case facets 7 and 8 do not exist. Facet numbering can be displayed graphically with MBODY = OFF in the geometry plotter. The order of the facet numbering presented by macrobody type, is provided in Table 5.2.

5.3.4.11.1 Example 1

The following input file describes five cells and illustrates a combination of the various body and cell/surface descriptions. In Fig. 5.4, surface numbers are in given within the planes they define and cell numbers are given within circles. Note that the cell and surface numbers do not have to start with 1 or be consecutive.

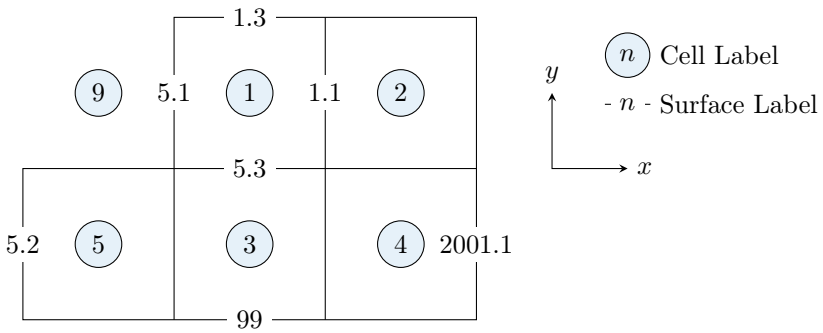


Figure 5.4: Macrobody Geometry Example

```
3  0 -1.2    -1.1  1.4 -1.5 -1.6 99
4  0  1.1 -2001.1 -5.3 -5.5 -5.6 -5.4
5  0  -5
1  0  -1
2  like 1 but trcl = (2 0 0)
9  0 (-5.1:1.3:2001.1:-99:5.5:5.6) #5

5  rpp  -2 0 -2 0 -1 1
1  rpp   0 2  0 2 -1 1
99  py  -2
```

Alternative descriptions for cell 3:

```
3  0  5.1 -1.1 -5.3 -5.5 -5.6 99
3  0  5.1 -1.1  1.4 -5.5 -5.6 -5.4
3  0 -1.2 -1.1 -5.3 -5.5 -5.6 -5.4
```

5.4 Data Card Introduction

All MCNP6 input cards other than those for cells [§5.2] and surfaces [§5.3] are entered after the blank card delimiter following the surface card block. The card name must begin within the first five columns.

Only the cards listed in §4.3 are allowed in a restart-calculation input file. No data card can be used more than once with the same number or particle type designations. For example, `M1` and `M2`are acceptable, as are `CUT:N` and `CUT:P`, but two `M1` cards or two `CUT:N` cards are disallowed. Note that when data cards accept keyword-value pairs, the equals sign (=) between the keyword and the value is optional.

5.5 Geometry-focused Data Cards

5.5.1 VOL: Cell Volume

Volumes or masses of cells are required for some tallies. MCNP6 calculates the volumes of all cells that are rotationally symmetric (generated by surfaces of revolution) about any axis, even a skew axis. It will also

calculate the volumes of polyhedral cells. As a by-product of the volume calculation, areas and masses are also calculated. These volumes, areas, and masses can be printed in the MCNP output file by using the PRINT card. The user can enter values on the VOL card for the volume of any cell and these values, instead of the calculated values, will be used for tally purposes. If a cell volume required for a tally cannot be calculated and is not entered on the VOL or SDn cards, a fatal error message is printed.

Cell-card Form: VOL *x*
or
Data-card Form: VOL [NO] *x1 x2 ... xJ*

x	Volume of cell.
xj	Volume of cell j where $j = 1, 2, ..., J$ where J is equal to the number of cells in the problem (1).
NO	Optional, no volumes or areas are calculated (2).

Default: Use MCNP6-calculated volumes. MCNP6 attempts to calculate the volume of all cells unless the NO keyword appears on the VOL card.

Use: Use if required cell volumes are not properly calculated. Provides an alternative way to enter volumes required by tallies. Normally the SD*n* card is used. The VOL card can be used only for cell volumes; the SD*n* card can be used for cell and segment volumes or masses.

Details:

1. If the number of entries does not equal the number of cells in the problem, it is a fatal error. Use the jump (*n*J) feature to skip over cells for which you do not want to enter values.
2. When the NO entry appears on the VOL card, MCNP6 bypasses the volume calculation altogether. The *xj* entries following NO are optional. If present, *xj* entries are the volume values the code will use. If no value is entered for a cell on the VOL card, the calculated volume is used.

5.5.1.1 Stochastic Volume and Area Calculation

MCNP6 cannot calculate the volumes and areas of asymmetric, non-polyhedral, or infinite cells. Also, in some cases, the volume and area calculation can fail because of round-off errors. For these cases, when neither MCNP6 nor the user can calculate the volume or area, a stochastic estimation is possible by ray tracing. The procedure follows:

1. Void out all materials in the problem by inserting a VOID card into the data card portion of the input.
2. Set all nonzero importance to one and all positive weight windows to zero.
3. Use a planar source with a source weight equal to the surface area to flood the geometry with particles. This setup will cause the particle flux throughout the geometry to statistically approach unity.
   (a) Alternatively, it has been suggested that the best way to do a stochastic volume estimation is to use an inward-directed, biased cosine source on a spherical surface with weight equal to πr^2 [179], where this weight scales the track-length tally to convert it from a mean chord length estimate to a volume estimate.

4. Use the track-length tally (F4) to estimate volumes and the surface flux tally (F2) to estimate areas. The cell flux tally is inversely proportional to cell volume. Thus in cells whose volumes are known, the unit flux will result in a tally of unity and, in cells whose volumes are uncalculated, the unit flux will result in a tally of volumes. Similarly, the surface flux tally is inversely proportional to area so that the unit flux will result in a tally of unity wherever the area is known and a tally of area wherever it is unknown.
5. For any tally volume or area that MCNP6 cannot calculate, use the AREA, VOL, or SD card to assign a value of 1.0 to the area(s) and/or volume(s) of the surface(s) or cell(s) of interest.

An example of a stochastic volume calculation is given in Listing 5.13 where the FC cards indicate the expected values and the TF cards are used to ensure that TFC values correspond to the volume indicated in the FC card.

Listing 5.13: example_tally_universe_expansion_stochastic_volume.mcnp.inp.txt

```
sdef sur = 400 nrm = -1 dir = d1 wgt = 30790.7495978336 $ wgt = pi * (r = 99) ** 2
si1 0 1
sp1 -21 1
sb1 -21 2
c
c This block of tallies demonstrates stochastic volume sampling.
c
fc114 Hexagon: 6928 cm^3
fc124 Cylinder: 39270 cm^3
fc134 Cylinder - Two Hexagons: 25414 cm^3
fc144 Box: 216000 cm^3
fc154 Box - Cylinder: 176730 cm^3
fc164 Inner Sphere: 4064379 cm^3
fc174 Inner Sphere - Two Boxes: 3632379 cm^3
c
f114:n (1000 < 2000 < 3000)  (1100 < 2000 < 3000)
f124:n ((1000 1100 1200) < 2000 < 3000)
f134:n (1200 < 2000 < 3000)
f144:n ((1000 1100 1200) < 2000 < 3000) (2100 < 3000) t
f154:n (2100 < 3000)
f164:n ((1000 1100 1200) < 2000 < 3000) ((1000 1100 1200) < 2000 < 3100)
       (2100 < 3000) (2100 < 3100) 4000 t
f174:n 4000
c
sd114 1 1
sd124 1
sd134 1
sd144 1 1 1
sd154 1
sd164 1 1 1 1 1 1
sd174 1
c
tf144 3 7j $ f d u s m c e t
tf164 6 7j $ f d u s m c e t
```

5.5.2 AREA: Surface Area

MCNP6 calculates the area of surfaces as a by-product of the volume calculation. If the volume of all cells on either side of the surface can be calculated, the area of the surface will be calculated. Otherwise, the area calculation will fail.

Data-card Form: AREA *x1 x2 ... xJ*

xj	Area of surface j where $j = 1, 2, ..., J$ where J is equal to the number of cells in the problem (1,2).

Default: MCNP6 attempts to calculate the area of all surfaces.

Use: Use if required surface areas for F2 tallies are not properly calculated. Provides an alternative way to enter areas required by tallies. Normally the SD*n* card is used. The AREA card can be used only for areas of whole surfaces; the SD*n* card can be used for area of surface segments as well as whole surfaces.

Details:

1. If the number of entries does not equal the number of surfaces in the problem, it is a fatal error. Use the jump (*n*J) feature to skip over surfaces for which you do not want to enter values.
2. If no value is entered for a surface on the AREA card, the calculated area, if any, is used. A fatal error occurs if an area is required for tallying purposes and is not available either from the MCNP6 calculation or from and AREA or SD*n* card.

5.5.3 TR: Coordinate Transformation

Coordinate transformations in MCNP6 can be used to simplify the geometric description of surfaces. They also may be used to relate the coordinate system of a surface source problem to the coordinate system of the problem that wrote the surface source file and to position universes within container cells. See the surface source SSR card or universe U card. Periodic boundary surfaces cannot have surface transformations.

To use a transformation to simplify the description of a surface, choose an auxiliary coordinate system in which the description of the surface is easy, include a transformation number n on the surface card, and specify the transformation on a TR*n* card. See Section 10.1.2 for an example showing how much easier it is to specify a skewed cylinder this way than as a GQ surface. Often a whole cluster of cells will have a common natural coordinate system. All of their surfaces can be described in that system and then translated and/or rotated to a new system by a single TR*n* card.

Data-card Form: TR*n o1 o2 o3 xx' yx' zx' xy' yy' zy' xz' yz' zz' m*

n	Number assigned to the transformation.
	Restriction: $1 \le n \le 999$ for surface transformations; no limit for cell transformations using TRCL= *n*
o1 o2 o3	Displacement vector of the transformation. DEFAULT: $(0, 0, 0)$

The default rotation matrix is

$$\begin{bmatrix} xx' & yx' & zx' \\ xy' & yy' & zy' \\ xz' & yz' & zz' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.6}$$

See also 1, 2, 5.

m	Displacement vector origin	
	$m = 1$	the displacement vector is the location of the origin of the auxiliary coordinate system, defined in the main system. (DEFAULT)
	$m = -1$	the displacement vector is the location of the origin of the main coordinate system, defined in the auxiliary system.

Default: TR*n* 0 0 0 1 0 0 0 1 0 0 0 1 1

Use: Optional. Convenient for many geometries. TR cards used in a surface definition must have numbers $1 \leq n \leq 999$. TR card used for cell transformations via TRCL= *n* can have any number.

Reminder: When a transformation is applied to a cell, MCNP6 generates a set of new unique surface numbers based on the original surface numbers. The number of the generated surface is equal to the number of original surface plus 1000 times the number of the cell. This formula creates generated surface numbers that are predictable and can be used on other cell cards and on tally cards. This method, however, limits cell numbers to no more than 6 digits and the original surface numbers to no more than 3 digits.

Details:

1. If the symbol *TR is used, the rotation matrix entries are angles in degrees instead of cosines of the angles.
2. The rotation matrix entries specify the relationship between the directions of the axes of the two coordinate systems. For example, the value of *xx'* is the cosine of the angle (or, if the optional asterisk is used, the angle in degrees ranging from 0° to 180°) between the x axis of the main coordinate system and the x' axis of the auxiliary coordinate system. Similarly, *yx'* is the cosine of the angle between the y axis of the main coordinate system and the x' axis of the auxiliary system.
3. The meaning of the rotation matrix entries do not depend on the value of *m*. It is usually not necessary to enter all of the elements of the matrix. The following patterns are acceptable:
   (a) All nine elements. Required if one of the systems is right-handed and the other is left-handed.
   (b) Two of the three vectors either way in the matrix (6 values). MCNP6 will create the third vector by cross product.
   (c) One vector each way in the matrix (5 values). The component in common must be less than 1. MCNP6 will fill out the matrix by the Eulerian angles scheme.
   (d) One vector (3 values). MCNP6 will create the other two vectors in some arbitrary way. Appropriate when the auxiliary coordinate system is being used to describe a set of surfaces that are all surfaces of rotation about a common skew axis.
   (e) None. MCNP6 will create the identity matrix. Appropriate when the transformation is a pure translation.
4. A vector consists of the three elements in either a row or a column in the matrix. In all cases, MCNP6 cleans up any small non-orthogonality and normalizes the matrix. In this process, exact vectors like $(1, 0, 0)$ are left unchanged. A warning message is issued if the non-orthogonality is more than about 0.001 radian.

(5) A cone of one sheet can be rotated only from being on or parallel to one coordinate axis to being on or parallel to another coordinate axis (multiples of 90°). A cone of one sheet can have any origin displacement vector appropriate to the problem. A cone of two sheets can be transformed anywhere. A cone of two sheets with an ambiguity surface in the cell description to cut the two-sheet cone in half (so that the cell appears as one sheet) can be transformed. The ambiguity surface must have the same transformation number as the cone of two sheets. Ambiguity surfaces are described in §2.2.3.2.

5.5.3.1 Example 1

```
  17  4  RCC   0 0 0   0 12 0   5
*TR4           20 0 0  45 -45 90  135 45 90  90 90 0
```

In this example, surface 17 is transformed via transformation 4 causing it to be displaced to $(x, y, z) = (20, 0, 0)$ and rotated 45° counter-clockwise with respect to x and y. If the rotational matrix is left incomplete, MCNP6 will calculate what it should be, but completeness is the only way to be sure you get what you want and get error messages if you are wrong.

5.5.3.2 Example 2

```
11 4 PX    5
TR4         7 0.9 1.3  0 1 0  0 0 1  1 0 0
```

Surface 11 is set up in an auxiliary coordinate system that is related to the main coordinate system by transformation number 4. MCNP6 will produce coefficients in the main coordinate system as if surface 11 had been entered as

```
11 P 0 -1.0 4.1
```

It will not produce

```
11 PY 4.1
```

This surface, represented by PY as shown in the line above, has the wrong sense. More examples of the transformation capability appear in §10.1.2.

5.5.4 TRCL: Cell Coordinate Transformation

A cell transformation (TRCL cell parameter) may be applied to a cell using either of two formats. The first TRCL entry form is an integer that is interpreted as the number of a TR*n* card that contains transformation information for all of the surfaces defining the cell. The associated TR*n* card is located in the data card section of the MCNP input file. An alternate entry form for TRCL allows the actual transformation to be entered on the cell card following the TRCL mnemonic, enclosed by parentheses. If the actual transformation is entered, all the rules applying to the TR card are valid.

Although a cell transformation can be applied to a standard cell, the utility of the TRCL parameter becomes most evident when applied to repeated structures [§5.5.5]. Assume your analysis model contains several cells identical in size and shape but located at multiple places in the geometry. You can describe the surfaces that describe these cells once and then use the TRCL keyword to position the identical cells in various locations and/or orientations. The TRCL feature is especially valuable when these cells are filled with the same universe. If the surfaces of these filled cells and the surfaces of the cells belonging to the universe that fills them are all described in the same auxiliary coordinate system, then a single transformation will completely define the interior of all these filled cells. That is, the cells of the universe will inherit the transformation of the cells they fill.

Cell-card Form: TRCL = *n*
or
Cell-card Form: TRCL = (*o1 o2 o3 xx' yx' zx' xy' yy' zy' xz' yz' zz' m*)

n	Number of the transformation corresponding to a TR*n* card in the data section of the input file. DEFAULT: $n = 0$
o1 o2 o3	Displacement vector of the transformation. DEFAULT: $(0, 0, 0)$

The default rotation matrix is

$$\begin{bmatrix} xx' & yx' & zx' \\ xy' & yy' & zy' \\ xz' & yz' & zz' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.7}$$

See the description of the rotation matrix in §5.5.3, Detail (2) and §5.5.4, Detail (2).

m	Displacement vector origin	
	$m = 1$	the displacement vector is the location of the origin of the auxiliary coordinate system, defined in the main system. (DEFAULT)
	$m = -1$	the displacement vector is the location of the origin of the main coordinate system, defined in the auxiliary system.

Default: No transformation if TRCL card is absent. This is equivalent to

```
TRCL=0
```

or

```
TRCL   0 0 0  1 0 0  0 1 0  0 0 1  1
```

Use: Optional. Convenient for many geometries. Use with the LIKE BUT cell description. To transform a standard cell description it is recommended that the TR parameter associated with the surface cards be used.

Details:

1. Coordinate transformations using TRCL can be applied only to cells with surface numbers < 1000. When a transformation is applied to a cell, MCNP6 generates a set of new unique surface numbers based on the original surface numbers. The number of the generated surface is equal to the number of the original surface plus 1000 times the number of the cell. This formula creates generated surface numbers that are predictable and can be used on other cell cards and on tally cards. This method, however, limits cell numbers to 6 digits and original surface numbers to no more than three digits. These generated surfaces are only the bounding surfaces of the transformed cell, not the surfaces of any universe that fills it. MCNP6 requires only one full description of each universe, no matter how many times that universe is referenced in the problem
2. If the symbol *TRCL is used, the rotation matrix entries are angles in degrees instead of cosines of the angles.
3. The displacement vector, *o1 o2 o3*, rotation matrix, *xx' yx' zx' xy' yy' zy' xz' yz' zz'*, and displacement vector origin, *m*, must be enclosed in parentheses for the second form.

5.5.4.1 Example 1

```
1    0     -17        $ rcc can
21   like 1 but *trcl=(20 0 0  45 -45 90  135 45 90  90 90 0)
```

Cell 21 is like cell 1 but is translated to $(x, y, z) = (20, 0, 0)$ and rotated 45° counter-clockwise with respect to x and y. If the rotational matrix is left incomplete, MCNP6 will calculate what it should be, but completeness is the only way to be sure you get what you want and get error messages if you are wrong.

5.5.5 Repeated Structures

The primary goal of the repeated-structures capability is to make it possible to describe only once the cells and surfaces of any structure that appears more than once in a geometry. Obvious examples of geometry models constructed from repeated structures include a reactor core with dozens of nearly identical fuel modules or a room containing complicated but nearly identical objects arranged in an irregular order. Although the repeated-structures feature reduces input and memory use, problems will not run any faster than with any other description. Examples of the use of repeated structures cards appear in §10.1.3.

The repeated structures capability extends the concept of an MCNP6 cell. Four cards are used exclusively to define repeated-structure features of a geometry: universe (U); fill (FILL); lattice (LAT); and, for stochastic geometries, URAN. Additionally, the cell transformation keyword (TRCL) is a supportive companion to the repeated-structures capability and the LIKE *n* BUT cell-description construct provides a convenient way to create multiple cells with similar attributes.

The user can specify that a cell is to be filled with something called a universe. The U card identifies the universe, if any, to which a cell belongs, and the FILL card specifies with which universe a cell is to be filled. A universe is either a lattice (LAT card) or a user-specified collection of cells. A single universe, described only once, can be designated to fill each of any number of cells in the geometry. Some or all of the cells in a universe may themselves be filled with universes. To use the repeated-structures capability effectively, keep in mind the following information:

Details:

1. Cell parameters (IMP, VOL, PWT, EXT, FCL, WWN, DXC, NONU, PD, TMP, U, TRCL, LAT, FILL, ELPT, COSY, BFLCL, and UNC) can be defined on cell cards.
2. LIKE *n* BUT is a shorthand method to describe easily one cell as equivalent to another except for a limited list of attributes [§5.2].
3. The universe (U) card specifies to what universe a cell belongs.
4. The fill (FILL) card specifies with which universe a cell is to be filled.
5. The cell transformation (TRCL) keyword allows the user to define only once the surfaces that bound several cells identical in size and shape but located at different places in the geometry. The TRCL keyword follows the transformation rules established for the surface transformation (TR) card.
6. The lattice (LAT) card defines an infinite array of hexahedra or hexagonal prisms. Lattice cell indexing is determined by the user-specified order of the surfaces that describe the $[0, 0, 0]$ lattice cell.
7. A general source description can be defined in a repeated structures part of the geometry. Surface source surfaces must be regular MCNP6 surfaces, not surfaces associated with a repeated structures part of the geometry. No check is made that this requirement is met.
8. An importance assigned to a cell that is in a universe is interpreted as a multiplier of the importance of the filled cell. Weight-window lower bounds are not multipliers. Mesh-based weight windows (MESH card) automatically address this issue.

5.5.5.1 U: Universe Keyword

Think of a universe as either a lattice cell or a collection of ordinary cells that you want to treat in a common manner. For example, perhaps this collection of cells appears multiple times in your model, but in varying orientations. You can use the repeated structures capability to simplify the setup of your model.

By assigning a non-zero universe number to one or more cells, the user creates a geometry unit that can be manipulated or referenced as a group. This assignment is accomplished by using the universe (U) card or, equivalently, the U cell-parameter keyword. If a cell lacks a universe assignment or is assigned to universe zero (U = 0), then the cell does not belong to any universe and is a member of the real world. By using the FILL card, a cell can be filled with a collection of cells that are assigned to the same universe. Note that the cells of a universe may be geometricly finite or infinite, but they must fill all of the space inside any cell that the universe is specified to fill.

One way to think about the connection between a filled cell and the filling universe is that the filled cell is a "window" that looks into a second level, like a window in a wall provides a view of the outdoors. Cells in the second level can be infinite because they will be (virtually) truncated when they bump into or intersect the surfaces of the window. The second level can have its own origin, in a primed coordinate system, unrelated to the upper level origin. However, if the filled cell and filling universe have all their surfaces in the same coordinate system, one TRCL parameter defines the coordinate system of both filled and filling cells.

A cell in a universe can be filled by another universe, in which case a third level is introduced. Up to 20 levels are permitted, more than most problems will need. The nomenclature chosen to address these hierarchical levels uses the following convention: the highest to lowest level is in inverse order to the associated numerical value. That is, the highest level is level zero (also known as the real world), lower is level one, lower still is level two, etc.

Every cell in the problem is either part of the real world (universe level 0) or part of some universe, but the surfaces of a problem are less restricted. A single planar surface can be used to describe cells in more than one universe. Coincident surfaces cannot be reflecting or periodic, source surfaces, or tally surfaces. Materials are normally put into the cells of the lowest level universe, not in the higher level, but there is an exception in the case of a lattice.

Cell-card Form: `U` = *n*
or
Data-card Form: `U` *n1 n2 ... nJ*

n	Arbitrary universe number (integer) to which cell is assigned. DEFAULT: `U` = 0, the "real world" universe (1). Restriction: $0 \leq n \leq 99,999,999$
nj	Universe numbers assigned to each cell of the problem in the same order as the cells appear in the cell card section. Note: when provided in the form of a data card, there must be an entry (which can be 0) for each cell in the problem. The jump feature (*nJ*) can be used for cells not assigned a universe number. Restriction: $0 \leq nj \leq 99,999,999$

Default: Lack of a `U` card or a zero entry means that the cell does not belong to any universe. Instead the cell is part of what is termed the "real world."

Use: Required for repeated structures.

Details:

1 A problem will run faster by preceding the `U` card entry with a minus sign for any cell that is not truncated by the boundary of any higher-level cell. The minus sign indicates that calculating distances to boundary in higher-level cells can be omitted.

Caution

Use this capability with extreme caution. MCNP6 cannot detect errors in this feature because the logic that enables detection is omitted by the presence of the negative universe. Extremely wrong answers can be quietly calculated. Plot several views of the geometry and/or run with the `VOID` card to check for errors.

5.5.5.1.1 Example 1

Planar surfaces of a filled cell and those in a filling universe can be coincident as shown in Listing 5.14.

Listing 5.14: example_universe_1.mcnp.inp.txt

```
1    0     1 -2 -3  4 -5  6         fill=1
2    0    -7  1 -3  8         u=1   fill=2  lat=1
3    0   -11                 u=-2
4    0    11                 u=2
```

```
 5    0    -1:2:3:-4:5:-6

 1    px     0
 2    px    50
 3    py    10
 4    py   -10
 5    pz     5
 6    pz    -5
 7    px    10
 8    py     0
10    py    10
11    s      5  5  0  4
```

In other words, the cells of a universe can fit exactly into the filled cell. This example illustrates this feature. Represented is a $50 \times 20 \times 10$-cm box filled with a lattice of $10 \times 10 \times 10$-cm cubes, each of which is filled with a sphere. Cell 1 is filled with cell 2, which is designated universe 1. Cell 2 is filled with cells 3 and 4 (universe 2). It is also a square lattice cell [§5.5.5.2]. Cell 3 is designated universe -2 indicating it is fully enclosed by surface 11.

The minus universe number of cell 3 indicates that calculating distances to boundary in higher level cells can be omitted. Cell 3 is a finite cell and is not truncated by any other cell. Cell 4 cannot have a negative universe number because it is an infinite region that is truncated by cell 2. This negative notation can increase computational efficiency.

The example in Listing 5.14 can be described with macrobodies as shown in Listing 5.15.

Listing 5.15: example_universe_1_macrobody.mcnp.inp.txt

```
 1    0   -20          fill=1
 2    0   -30     u=1  fill=2  lat=1
 3    0   -11     u=-2
 4    0    11     u=2
 5    0    20

20    rpp   0   50  -10   10   -5    5
30    rpp   0   10    0   10    0    0
11    s     5    5    0    4
```

5.5.5.2 LAT: Lattice

Two different lattice-element shapes can be specified in MCNP6: hexahedra (LAT = 1), solids with six faces, and hexagonal prisms (LAT = 2), solids with eight faces. A non-zero entry on the LAT card indicates that the corresponding cell is the $[0, 0, 0]$ element of a lattice. The hexahedra need not be rectangular and the hexagonal prisms need not be regular, but the lattices made out of them must fill space exactly. In other words, opposite sides have to be identical and parallel. A hexahedral lattice cell may be infinite in one or two of its dimensions. A hexagonal prism lattice cell may be infinite in the direction along the length of the prism. The cross-sectional shape of a lattice element must be convex. It does not matter whether the lattice is left-handed or right-handed. A lattice must be the only thing in its universe. The real world (universe level 0) itself can be a lattice. If a particle leaves the last cell of a real-world, limited-extent lattice (see the FILL card for how the extent of a lattice can be limited), the particle escapes and is killed.

The cell description of a lattice cell not only provides the standard MCNP6 cell description for the base lattice element, but, through the order of the surface specification for the lattice cell, it identifies which lattice element lies beyond each surface. The first two surfaces listed on the cell card define the direction of the first

lattice index; the third and fourth surfaces listed in the cell description define the direction of the second lattice element, etc. This concept is further explained in the following discussion.

After designing your lattice, decide which element you want to be the $[0, 0, 0]$ element and in which directions you want the three lattice indices to increase. In the case of a hexagonal prism lattice you have two constraints: the first and second indices must increase across adjacent surfaces and the third index must increase in one or the other direction along the length of the prism. Enter the bounding surfaces of the $[0, 0, 0]$ element on the cell card in the appropriate order, in accordance with the following conventions. For a hexahedral lattice cell, beyond the first surface listed is the $[1, 0, 0]$ element, beyond the second surface listed is the $[-1, 0, 0]$ element. Similarly, the $[0, 1, 0]$, $[0, -1, 0]$, $[0, 0, 1]$, and $[0, 0, -1]$ lattice elements are beyond the 3rd, 4th, 5th, and 6th surfaces in that order. This method provides the order of arrangement of the lattice to the code so that when you specify element $[7, 9, 3]$, the code knows to which element you are referring.

For a hexagonal prism lattice cell, on the opposite side of the first surface listed is element $[1, 0, 0]$, opposite the second listed surface is $[-1, 0, 0]$, followed by the lattice elements $[0, 1, 0]$, $[0, -1, 0]$, $[-1, 1, 0]$, $[1, -1, 0]$, $[0, 0, 1]$, and $[0, 0, -1]$, which are opposite the 3rd through 8th surfaces defining the hexagonal prism $[0, 0, 0]$ lattice cell. These last two surfaces must be the base surfaces of the prism. You can use the MCNP6 geometry plotter to label the lattice cells with their indices. This provides an easy way to verify the lattice index arrangement. The example in §10.1.3.7 illustrates a hexagonal prism lattice cell.

Each cell containing a lattice, whether specified using a `LAT` keyword or a `LAT` data card, must have an associated `FILL` keyword.

Cell-card Form: `LAT` = *n*
or
Data-card Form: `LAT` *n1 n2 ... nJ*

n	Lattice type.
	n = 1 — the cell describes a rectangular (square) lattice comprised of hexahedra with six faces.
	n = 2 — the cell describes a hexagonal (triangular) lattice comprised of hexagonal prisms with eight faces.
nj	Lattice type assigned to each cell of the problem in the same order as the cells appear in the cell card section
	nj = 1 — the cell describes a rectangular (square) lattice comprised of hexahedra with six faces.
	nj = 2 — the cell describes a hexagonal (triangular) lattice comprised of hexagonal prisms with eight faces.
	Note: when provided in the form of a data card, there must be an entry for each of the cells in the problem. The jump feature (*nJ*) can be used to pass over cells that are not lattice cells.

Use: Used to define an infinite array of hexahedra or hexagonal prisms. A non-zero entry on the `LAT` card means that the corresponding cell is the $[0, 0, 0]$ element of a lattice. The order of specification of the surfaces of a lattice cell identifies which lattice element lies beyond each surface. Required for lattices.

5.5.5.2.1 Example 1

This example is the same as in Listing 5.15. Cell 2 is the base $[0, 0, 0]$ element of a square lattice described by surface 30, a right parallelepiped with *xmin* = 0, *xmax* = 10, *ymin* = 0, *ymax* = 10, and infinite in the z

direction. It is filled with Universe 2 (cells 3 and 4) and it is assigned to universe 1, which fills and is bounded by cell 1 (an RPP with *xmin* $= 0$, *xmax* $= 50$, *ymin* $= -10$, *ymax* $= 10$, *zmin* $= -5$, and *zmax* $= 5$). In this case the lattice elements $[i, j, k]$ would be $[0:4, -1:0, 0:0]$.

5.5.5.3 FILL: Fill

A nonzero entry on the FILL card indicates the number of the universe that fills the corresponding cell. The same number on the U card identifies the cells making up the filling universe. If the filled cell is a lattice, the FILL specification can be either a single entry or an array. If it is a single entry, every cell of the lattice is filled by the same universe. If it is an array, the portion of the lattice covered by the array is filled and the rest of the lattice does not exist. It is possible to fill various elements of the lattice with different universes.

5.5.5.3.1 Lattice Indexing

The array specification for a cell filled by a lattice has three-dimension array declarations followed by the array values themselves. The dimension declarations define the ranges of the three lattice indices. They are in the same form as in Fortran, but both lower and upper bounds must be explicitly stated with positive, negative, or zero integers, separated by a colon. The indices identify each lattice element's location with respect to the $[0, 0, 0]$ element. The LAT card describes how the specified order of the surfaces of the lattice-element cell $[0, 0, 0]$ determines the ordering of the lattice elements. The numerical range of the indices depends on where in the lattice the $[0, 0, 0]$ element is located. For example, $-5:5$, $0:10$, and $-10:0$ all define a range of 11 elements.

The array values follow the dimension declarations. Each element in the array corresponds to an element in the lattice. Only those elements of the lattice that correspond to elements in the array actually exist. The value of each array element is the number of the universe that is to fill the corresponding lattice element. A real world (level zero) lattice, by default, is universe zero and can only be universe zero.

Cell-card Form: FILL $=$ *n* (*q*)
or
Cell-card Form: FILL $=$ *n* (*o1 o2 o3 xx' yx' zx' xy' yy' zy' xz' yz' zz' m*)
or
Cell-card Form: FILL $i_1{:}i_2$ $j_1{:}j_2$ $k_1{:}k_2$ $n_{1,1,1}$ $n_{2,1,1}$ ... n_{i_1,j_1,k_1} ... n_{i_2,j_2,k_2}
or
Data-card Form: FILL n_1 n_2 ... n_n ... $n_{I\times J\times K}$

n	Arbitrary number (integer) of the universe with which cell is to be filled. If the filled cell is a lattice, every cell of the lattice is filled by the same universe. DEFAULT: FILL=0
q	Optional transformation number of a TR*q* surface transformation card, enclosed in parentheses.
o1 o2 o3	Optional displacement vector of the transformation. DEFAULT: $(0, 0, 0)$

The default rotation matrix is

$$\begin{bmatrix} xx' & yx' & zx' \\ xy' & yy' & zy' \\ xz' & yz' & zz' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.8}$$

See the description of the rotation matrix in §5.5.3, Detail (2) and §5.5.4, Detail (2).

m	Displacement vector origin. See the description of the displacement vector in §5.5.3.
i_1:i_2 j_1:j_2 k_1:k_2	Lattice element parameters for the upper and lower bounds in the i, j, and k directions (for fully specified fill).
$n_{i,j,k}$	Number of the universe with which to fill each existing lattice element (for fully specified fill). Each element in the array corresponds to an element in the lattice. The portion of the lattice covered by the array is filled and the rest of the lattice does not exist ((3)).
n_n	Number of the universe with which each cell is to be filled in the same order as the cells appear in the cell card section ((3)). Note: when provided in the form of a data card, there must be an entry for each of the cells in the problem. The jump feature (*n*J) can be used for cells not assigned a universe number.

Default: FILL = 0

Use: Required for repeated structures.

Details:

(1) As with a single entry FILL specification, any FILL entry for a fully specified FILL card optionally may be followed by, in parentheses, either a transformation number or the transformation itself. This transformation is between the coordinate systems of the filled cell and the filling universe, with the universe considered to be in the auxiliary coordinate system. If no transformation is specified, the universe inherits the transformation, if any, of the filled cell.

(2) A *FILL may be used if the rotation matrix entries are angles in degrees rather than cosines. In the data card section of the MCNP input file you cannot have both a FILL and a *FILL entry. If you want to enter some angles by degrees and some angles by cosines, all FILL and *FILL data must be placed on the cell cards of the MCNP input file.

(3) There are two *nj* values that can be used in the lattice array that have special meanings. A zero in the level-zero (real world) lattice means that the lattice element does not exist, making it possible, in effect, to specify a non-rectangular array. If the array value is the same as the number of the universe of the lattice, that element is not filled with any universe but with the material specified on the cell card for the lattice cell. Therefore, using the universe number of a real world lattice as an *nj* value to fill that element with the cell material is not possible.

(4) The displacement vector, *o1 o2 o3*, rotation matrix, *xx' yx' zx' xy' yy' zy' xz' yz' zz'*, and displacement vector origin, *m*, must be enclosed in parentheses for the second form.

5.5.5.3.2 Example 1

```
FILL=0:2 1:2 0:1    4 4 2   $ i=0,1,2 for j=1 & k=0
                    0 4 0   $ i=0,1,2 for j=2 & k=0
                    0 3 3   $ i=0,1,2 for j=1 & k=1
                    4 4 0   $ i=0,1,2 for j=2 & k=1
```

Only eight elements of this lattice exist. Elements $[0, 1, 0]$, $[1, 1, 0]$, $[1, 2, 0]$, $[0, 2, 1]$ and $[1, 2, 1]$ are filled with universe 4. Element $[2, 1, 0]$ is filled with universe 2. Elements $[1, 1, 1]$ and $[2, 1, 1]$ are filled with universe 3.

5.5.5.4 URAN: Stochastic Geometry for HTGRs

The URAN card provides a limited means of modeling stochastic geometry in MCNP6 for both fixed-source and eigenvalue problems. It is primarily intended for modeling the randomly located fuel kernels in high-temperature gas-cooled reactor (HTGR) geometries.

⚠ Caution

Although this feature may have other possible applications, users should proceed carefully and perform their own verification calculations to ensure that the feature adequately represents the physical problem they are modeling.

MCNP6 has been used frequently to model HTGRs with explicit geometric representation of fuel compacts or pebbles, including the microscopic fuel kernels within them [221, 222]. Each fuel kernel typically has a spherical (≈ 0.5 mm diameter) uranium oxycarbide region surrounded by layers of graphite, pyrolytic graphite, and silicon carbide. Modular HTGRs contain cylindrical fuel compacts filled with randomly located fuel kernels in a graphite matrix. Pebble bed HTGRs contain spherical fuel pebbles filled with randomly located fuel kernels in a graphite matrix.

Modeling these geometries in multigroup deterministic codes requires the implementation of shielding factors to account for double heterogeneities (i.e., fuel kernels and fuel particles). Monte Carlo codes that permit hierarchical geometry models, such as MCNP6 with its embedded lattices and universes, can explicitly model the pebble bed double heterogeneities. The random locations of fuel kernels within each fuel compact or pebble are typically modeled in MCNP6 using a regular lattice arrangement, ignoring any randomness.

To provide a limited form of randomness to the locations of fuel kernels in HTGR models, the URAN card may be used to flag selected universes in a lattice as stochastic. This feature provides an additional, random transformation to the geometry each time a neutron enters the lattice element. That is, when a neutron enters a lattice element containing an embedded universe flagged as stochastic, the universe coordinates are transformed randomly according to

$$x = x + \delta_x(2\xi_1 - 1), \tag{5.9a}$$
$$y = y + \delta_y(2\xi_2 - 1), \tag{5.9b}$$
$$z = z + \delta_z(2\xi_3 - 1), \tag{5.9c}$$

where ξ_1, ξ_2, ξ_3 are random numbers uniformly distributed on $(0, 1)$, and δ_x, δ_y, δ_z are user-defined parameters supplied on the URAN card. Different translation parameters can be declared for different levels of the geometry, and the random translations are performed only when entering lattice elements containing universes that the user declares as stochastic on the URAN card. To preserve mass and packing fractions, the translation parameters should be chosen such that fuel kernels or other objects are not displaced beyond the edges of the enclosing cell or lattice element.

In addition to the random translation applied to a neutron entering a stochastic universe, special treatment is needed to save the fission sites in an eigenvalue calculation. When a fission occurs and the site parameters are saved in the fission bank, the current values of the random translation parameters must be saved along with the normal fission-site data. In the next cycle of the calculation, these saved translation parameters are used for the neutron starting at that fission site. This ensures that the flight continues from the same stochastic realization in effect when the site was saved.

This stochastic geometry treatment has been verified for several realistic HTGR problems [221, 222].

Data-card Form: `URAN n1 d1x d1y d1z n2 d2x d2y d2z ... nJ dJx dJy dJz`

nj	Universe number to which to apply stochastic transformation. Only applied when used to fill a lattice element.
djx	Maximum translation in the $\pm x$ direction for stochastic transformation applied to universe *nj*.
djy	Maximum translation in the $\pm y$ direction for stochastic transformation applied to universe *nj*.
djz	Maximum translation in the $\pm z$ direction for stochastic transformation applied to universe *nj*.

Default: None.

Use: To model random nature of HTGR or similar geometries.

Caution

There is no stochastic geometry plotting capability associated with the URAN card. Users should be extremely cautious in supplying information using the URAN card because MCNP6 has no means of checking whether the supplied parameters properly represent the physical model being simulated.

5.5.6 Hybrid Geometries: Structured and Unstructured Meshes

A geometry mesh from an external file can be embedded into an MCNP constructive solid geometry model using the "universe" construct from the repeated structures capability. The embedded geometry mesh may be structured or unstructured.

A structured mesh, such as that defined by the geometry/materials block of the PARTISN discrete ordinates (S_N) code [223], can be embedded into an MCNP geometry. The resultant hybrid geometry can then be used for the MCNP calculations. The only structured-mesh geometry file format that the MCNP code can process is LNK3DNT, a binary file format used by the PARTISN code. The MCNP code can import a geometry description from a LNK3DNT file for continuous-energy neutron particle transport. It is also possible to convert an MCNP constructive solid geometry to a structured mesh geometry. This capability samples an MCNP constructive solid geometry, creates a homogenized regular mesh of materials in 1-D (r), 2-D (r, z), or 3-D (x, y, z) or (r, z, θ), and writes a structured mesh model in the file formatted as LNK3DNT.

Unstructured meshes, such as those created by the Abaqus/CAE code (https://www.3ds.com/products-services/simulia/products/abaqus), also can be embedded in a hybrid arrangement. Many other computer-aided engineering (CAE) or mesh generation tools have the ability to generate a mesh from a solid model that can be converted to the Abaqus format. The MCNP code can process meshes that consist of four-, five-, or

six-sided finite elements, with linear, bi-linear, or quadratic faces. Starting with the MCNP code, version 6.3, the code can process a mesh formatted as an Abaqus input file or an HDF5 mesh input file; see §D.6 for an HDF5 mesh input file format.

Before creating or incorporating structured or unstructured meshes, it is highly recommended that the interested user become familiar with Chapter 8 and external references, as appropriate [224–226]. This section is not meant to provide information regarding PARTISN or Abaqus file formats nor provide a primer on how to run or interact with these codes.

5.5.6.1 Creation of a Structured Mesh Geometry File

LNK3DNT-format files, used by the LANL PARTISN code, can be created from an MCNP input file [226]. Two cards are required to accomplish this task: the `MESH` card and the `DAWWG` card. It is a known issue that some complex constructive solid geometries or unstructured mesh geometries in MCNP input files cannot be converted into correct LNK3DNT format files; that is PARTISN may not be able to read some LNK3DNT files created by the MCNP code.

5.5.6.2 MESH: Superimposed Importance Mesh for Mesh-Based Weight-Window Generator

The `MESH` card specifies the layout and orientation of the geometry to be generated with respect to the cell-based coordinate system. For this application, the supported coordinate-system options include `XYZ` (x, y, z), `CYL` in either 2-D (r, z) or 3-D (r, z, θ), and `SPH` (r). The optional `ORG`, `AXS`, and `VEC` keywords of the `MESH` card can be used to align the mesh with the MCNP geometry from which the discrete-ordinates geometry will be generated.

The geometry orientation is not transferred to the resultant LNK3DNT file. Regarding the `MESH` orientation parameters, the generated homogenized geometry will have a geometric center at $(0, 0, 0)$ and will be aligned with the global MCNP Cartesian coordinate system. For cylindrical geometries, the defaults are that the cylinder axis is aligned with the positive z axis and the azimuthal plane ($\theta = 0$) is aligned with the positive x axis.

5.5.6.3 DM: Target Aliases for Deterministic Materials

When generating a PARTISN input using the `DAWWG` card, the nuclide identifiers in the `matls` input array in Block IV need to be set. By default, the code will convert the identifier to the ZAID format and remove the suffix (e.g., `U-238.80c` would be written as `92238`). The `DM` card allows the user to override this nuclide mapping on a material-by-material basis.

Data-card Form: `DM`*n original_target_1 new_target_1 original_target_2 new_target_2 ...*

n	The material to which this card applies. If *n* = `0`, this card will apply to all materials.
original_target_j	Any target identifier of the forms in §1.2.3. All targets with the same Z, A, and S will be overridden with the matching *new_target_j*. The suffix, if present, will be ignored.
new_target_j	The replacement target identifier for `matls` in Block IV in the PARTISN input. The MCNP code will read and write identifiers up to 8 characters in length.

Default: The target identifiers on the material card will be used with the suffix removed.

5.5.6.4 DAWWG: Deterministic Adjoint Weight-window Generator

The DAWWG card specifies the number of points to sample in each element of the mesh. This sampling is used to estimate the volume fraction of different materials within each element. From this information a homogenized material definition with its associated density can be generated for each element. DAWWG may also be used to pass information directly to PARTISN.

Caution

The DAWWG card must appear after the MESH card in the input file's data section.

Data-card Form: DAWWG keyword = *values(s)*

POINTS = *n*	Randomly sample the material within each element of the defined mesh using n sample points in each coordinate direction for each mesh element, where n is an integer. This sampling is used to estimate material volume fractions and thereby estimate the composition of each geometry element. (DEFAULT: POINTS = 1) Required. For example: if $n = 10$, then $10^3 = 1000$ points will be sampled in each geometry mesh element.
XSEC = *name*	Declares that cross-section library *name* will be passed to the discrete-ordinates code for weight-window generation. This information is not explicitly used in the generation of the mesh. (Required)[a]
TALLY = *i*	[b]
BLOCK = *k*	Passes values from an MCNP input file to a PARTISN input file. (Optional) Multiple BLOCK entries are permitted. The allowed keywords and generally suitable for each BLOCK are provided in Tables 5.3, 5.4, 5.5, and 5.6. If

BLOCK = *1*	then pass values of the listed keywords to the dimension and controls block of the PARTISN input file [Table 5.3].
BLOCK = *3*	then pass values of the listed keywords to the nuclear data type and options block of the PARTISN input file [Table 5.4].
BLOCK = *5*	then pass values of the listed keywords to the solver input block of the PARTISN input file [Table 5.5].
BLOCK = *6*	then pass values of the listed keywords to the edit controls block of the PARTISN input file [Table 5.6].

[a] This keyword will be implemented at a later date.
[b] This keyword will be implemented at a later date.

5.5.6.4.1 Example 1

Listing 5.16: example_dawwg.mcnp.inp.txt

```
Converts MCNP cube geometry to LNK3DNT format
1   1  -18.7     -1    imp:n=1
2   0             1    imp:n=0
```

Table 5.3: PARTISN Block 1: Dimension and Control Defaults Suitable for the DAWWG Card

Block Keyword	Type	Default	Description
NGROUP	Integer	30	Number of energy groups
ISN	Integer	8	S_N order
NISO	Integer	0	Number of isotopes
MT	Integer	1	Number of materials
IQUAD	Integer	6	Quadrature (1–9)
FMMIX	Integer	1	1 means read composition from LNK3DNT file
NOSOLV	Integer	0	1 means suppress solver module
NOEDIT	Integer	0	1 means suppress edit module
NOGEOD	Integer	0	1 means suppress writing GEODST file
NOMIX	Integer	0	1 means suppress writing mixing files
NOASG	Integer	0	1 means suppress writing ASGMAT file
NOMACR	Integer	0	1 means suppress writing MACRXS file
NOSLNP	Integer	0	1 means suppress writing SOLINP file
NOEDTT	Integer	0	1 means suppress writing EDITIT file
NOADJM	Integer	0	1 means suppress writing ADJMAC file

Table 5.4: PARTISN Block 3: Nuclear Data Type and Options Defaults Suitable for the DAWWG Card

Block Keyword	Type	Default	Description
LIB	Text	ndilib	Form of the cross-section data file
LIBNAME	Text	mendf5	Cross-section file name
FISSNEUT	Integer	0	Fission neutron flag
LNG	Integer	0	Number of the last neutron group
BALXS	Integer	0	Cross-section balance control (-1,0,1)
NTICHI	Integer	0	MENDF fission fraction to use

Table 5.5: PARTISN Block 5: Solver Defaults Suitable for the DAWWG Card

Block Keyword	Type	Default	Description
IEVT	Integer	1	Calculation type (0–4)
ISCT	Integer	3	Legendre order
ITH	Integer	0	Direct (0) or adjoint (1) calculation
TRCOR	Text	diag	
IBL	Integer	0	Left boundary condition
IBR	Integer	0	Right boundary condition
IBT	Integer	0	Top boundary condition
IBB	Integer	0	Bottom boundary condition
IBFRNT	Integer	0	Front boundary condition
IBBACK	Integer	0	Back boundary condition
EPSI	Real	0.0001	Convergence precision
OITM	Integer	20	Maximum outer iteration count
NOSIGF	Integer	0	1=inhibit fission multiplication
SRCACC	Text	DSA	Transport acceleration (DSA, TSA, NO)
DIFFSOL	Text	mg	Diffusion operator solver
TSASN	Integer	0	S_N order for low order TSA sweeps
TSAEPSI	Real	0.0	Convergence criteria for TSA sweeps
TSAITS	Integer	0	Maximum TSA iteration count
TSABETA	Real	0.0	Scattering cross-section reduction for TSA
PTCONV	Integer	0	1=Special criticality convergence scheme
NORM	Real	1.0	
XSECTP	Integer	0	Cross-section print flag (0, 1, 2)
FISSRP	Integer	1	1 means print fission source rate
SOURCP	Integer	0	Source print flag (0, 1, 2, 3)
ANGP	Integer	0	1 means print angular flux
BALP	Integer	0	1 means print coarse-mesh balance tables
RAFLUX	Integer	0	1 means prepare angular flux file
RMFLUX	Integer	0	1 means prepare flux moments file
AVATAR	Integer	0	1 means prepare special XMFLUXA file
ASLEFT	Integer	i	*i* means right-going flux at plane i (0: none)
ASRITE	Integer	i	*i* means left-going flux at plane i (0: none)
ASBOTT	Integer	j	*j* means top-going flux at plane j (0: none)
ASTOP	Integer	j	*j* means bottom-going flux at plane j (0: none)
ASFRNT	Integer	k	*k* means back-going flux at plane k (0: none)
ASBACK	Integer	k	*k* means front-going flux at plane k (0: none)

Table 5.6: PARTISN Block 6: Edit Control Defaults Suitable for the DAWWG Card

Block Keyword	Type	Default	Description
MASSED	Integer	1	1 enables mass edits
PTED	Integer	0	1 enables edits by fine mesh
ZNED	Integer	0	1 enables edits by (edit) zone
RZFLUX	Integer	0	1 means write a-flux file
RZMFLUX	Integer	0	1 means write b-flux file
EDOUTF	Integer	3	ASCII output files control [-3:3]
BYVOLP	Integer	0	1 means printed point reaction rates scaled by mesh volume
AJED	Integer	0	Regular (0) and Adjoint (1) edit
FLUXONE	Integer	0	1 means flux override

```
4
5  1  rpp    -10 10  -10 10  -10 10
6
7  kcode     5000  1.0  50  250
8  ksrc      0.0 0.0 0.0
9  m1        92235.69c   1.0
10 dm1 92235 92235.50
11 prdmp    j    275
12 mesh  geom  xyz
13       ref      0.0    -0.0    -0.0
14       origin -10.000 -10.000 -10.000
15       imesh  10
16       iints   2
17       jmesh  10
18       jints   2
19       kmesh  10
20       kints   2
21 f4:n 1
22 dawwg points=10
23       block=1 ngroup=16 isn=16 iquad=4
24       block=3 libname=mendf5 lib=ndilib
25       block=5 trcor=diag srcacc=dsa diffsol=mg isct=2
26       block=6 massed=1 edoutf=3
```

The MCNP input file in Listing 5.16 creates a LNK3DNT-format mesh file of a solid, one-material cube with density 18.7 g/cm^3. Each edge of the cube is 20 cm long ($-10 < x, y, z < 10$ cm) and has two mesh segments along each cardinal direction (i.e., mesh boundaries at $x, y, z = -10, 0, 10$ cm). In each element of the mesh, 1000 points will be randomly sampled to estimate the material composition in each element. The `DAWWG` card is used to pass additional options directly to the PARTISN discrete ordinates code. These additional options are not used by the MCNP code.

5.5.6.4.2 Example: Creation of Structured Mesh File

After an MCNP input file has been created to generate the structured mesh file, execute the code using the `M` execution option:

```
1 mcnp6 M I=MYINP LINKOUT=MYLNK
```

The LNK3DNT file created by the MCNP code is named **`LINKOUT`** by default but can be changed via file name assignment on the execution line. If a file with the name **`LINKOUT`** already exists, the MCNP code will adhere to the usual rules for selecting a file name. In the example, the MCNP input file is **`MYINP`** and the LNK3DNT output file is **`MYLNK`**. When invoked in this manner, the code will process the input file, generate the **`LINKOUT`** file, and exit.

5.5.6.5 Mesh Importation and Specification of an Embedded Geometry

Structured and/or unstructured mesh geometries may be embedded into the MCNP code by using the repeated structure's universe (`U`) and fill (`FILL`) constructs. When used in the MCNP code, an embedded geometry must be assigned to an MCNP universe. To ensure that this universe will completely fill the cell in which it is placed, a background cell that represents the infinite region surrounding the embedded geometry

must also be assigned to the same universe. The embedded mesh geometry must not be clipped by the fill cell into which it is placed and no other universe or cell may be contained within it. This combination of the mesh geometry that is to be embedded and its surrounding infinite cell is called a "mesh universe." Similar to other repeated structures, a mesh universe may be placed in multiple locations within an MCNP geometry; also, more than one unique unstructured mesh universe may be embedded into an MCNP geometry, but only one LNK3DNT mesh may be embedded.

In the MCNP code, space is defined by a collection of cells using surfaces, lattices, universes, fills, etc. Building on this cell-geometry concept, incorporation of an embedded geometry into the code requires defining cells to represent the embedded geometry and its composition. There are three special MCNP cell categories for specifying the use of an embedded geometry: pseudo-cells [§8.2], background cells [§8.2], and fill cells [§5.5.5.3]. The treatment of these cells differs depending on the type of embedded mesh; this concept is explained in great detail below. Each embedded mesh universe consists of one or more pseudo-cells and a single background cell.

Pseudo-cells are defined in the MCNP cell block by a null surface—with a surface number of "0". These pseudo-cells are used to communicate normal MCNP cell properties from the external mesh to the MCNP code. The cell-card format for pseudo-cells exhibits the following properties that differ from those of regular MCNP cells:

- Pseudo-cells have a single null-surface entry (i.e., 0) instead of a list of signed surfaces.
- Pseudo-cells are assigned to a universe (e.g., `U` = 10 or `U` = `E`10 where the `E` prefix to the universe is optional and signifies a mesh universe).
- The universe number of the pseudo-cell must match the n specified on its associated `EMBED`*n* card.
- Pseudo-cells cannot be filled by another universe or lattice (i.e., a pseudo-cell cannot have a `FILL` or `LAT` entry).

Pseudo-cell cards may contain material, density, importance, and other cell properties. All cell-card fields that typically are required by regular MCNP cells also are required by pseudo-cells. How the user should think about these cells differs slightly depending on the type of embedded mesh. As will be seen in the next section, the pseudo-cells are connected to an embedded geometry through the `MATCELL` entries on the associated `EMBED` card.

For the PARTISN mesh, a pseudo-cell cell must exist for each material defined in an embedded geometry mesh input file. If void elements exist, a separate pseudo-cell is required to represent them. Pure elements, i.e., those that contain only one material or void, belong to only one pseudo-cell. Multi-material elements, however, belong to each pseudo-cell that is associated with a material contained in the element. Material density is assigned to each mesh element within the external mesh geometry file. The materials within the LNK3DNT geometry file must be numbered using non-negative integers, where 0 is for a void. The material information in the external mesh file is transferred to the MCNP code through the pseudo-cells and the `MATCELL` keyword of `EMBED` card; each unique material in the external mesh file must have an associated pseudo-cell, where the first and second entries of the `MATCELL` keyword are respectively material numbers read from a LNK3DNT file and the pseudo-cell numbers in the MCNP cell block.

For the Abaqus unstructured mesh, the unstructured mesh library that handles all of the details of unstructured meshes for the MCNP code establishes pseudo-cells (unique combinations of materials and element sets) from the information contained in the Abaqus input file. The reader is referred to Chapter 8 for details on this topic. A pseudo-cell cross-reference table is printed to the MCNP output file. Among other things, this table lists the known pseudo-cells and the associated material assignments that are expected. For each pseudo-cell entry in this table, there must be a pair of values for the `MATCELL` keyword on the `EMBED` card. The first and second entries of the `MATCELL` keyword are the unstructured mesh pseudo-cell numbers processed from the

mesh input file and the pseudo-cell numbers defined in the MCNP cell block. The first entries of the MATCELL keyword must start from 1 and increase by 1. This MATCELL association connects the MCNP pseudo-cells with the unstructured mesh pseudo-cells and is essential to establish the material properties for the unstructured mesh pseudo-cells because the only required material property in the mesh file is the material numbers. It is up to the user to ensure that this information is input correctly through the MATCELL entries on the associated EMBED card. For large and complex geometries, this can be a tedious process. To help with this input setup, users should take advantage of the unstructured mesh pre-processor program: **um_pre_op** [227], which is now deprecated [DEP-53422].

Mesh element assignment to each MCNP pseudo-cell is determined by the data as it appears in the external geometry mesh file. For a given geometry mesh, if the element sets of the mesh are built in a different order, then the mapping to the associated MCNP pseudo-cells will also be different. The user is directed to the previous list of references to decipher the cross-reference mapping data provided in the external geometry mesh file.

To make a mesh universe [§8.2] infinite in extent, it must have a background cell that consists of all space outside the associated embedded geometry. The background cell is connected to the mesh universe through the BACKGROUND entry on the EMBED card. The format of the background cell in MCNP cell-card definitions is the same as pseudo-cells.

The third cell type required to embed a mesh geometry into a MCNP input is the fill cell. The fill cell is a regular MCNP cell into which the mesh universe is placed. Mesh surfaces should not be coincident with fill-cell surfaces, otherwise lost particles may result. The cell card for a cell filled with a mesh universe has the following properties that differ from those of other MCNP fill cells:

- A fill cell cannot be a pseudo-cell (i.e., it must be defined using bounding surfaces and not use a null, 0, surface).
- Fill cells have a fill entry of the form FILL = E*n* or FILL = *n*, where *n* is the MCNP universe number of the mesh universe. This universe number, *n*, also appears as the *n* on the EMBED*n* card. The FILL entry may have a transformation (TR) to permit realignment of the embedded geometry within the fill cell. Recall that LNK3DNT geometries are always defined relative to the MCNP origin.
- Fill cells cannot have a lattice entry.

Eight MCNP data cards are available to support mesh importation. The EMBED card is required for both structured and unstructured mesh importation. Seven additional cards (EMBEE, EMBEB, EMBEM, EMBTB, EMBTM, EMBDE, and EMBDF) are optional and only valid for unstructured meshes. These seven cards provide for elemental edits, i.e., results accumulated on the unstructured mesh. These mesh results, along with a generic description of the unstructured mesh model, can be output to a special elemental-edit output (EEOUT) file [§8.4]. Not all tally features are duplicated with the unstructured mesh elemental edit capability, hence the name edit instead of tally. If traditional MCNP statistical analysis is desired for the results, the user must set up a tally for the appropriate pseudo-cell(s).

Be aware that there are additional limitations when using an unstructured mesh in an MCNP geometry. Consult Chapter 8 for an up-to-date list and more detailed guidance on using the UM feature.

5.5.6.6 EMBED: Embedded Geometry Specification

One card, EMBED, is required for embedding a mesh geometry into an MCNP input file. For each unique embedded geometry used in an MCNP input file, there must exist an associated EMBED card.

Deprecation Notice **DEP-53361**

The BACKGROUND pseudo-cell listed on the EMBED must be unique from all MATCELL pseudo-cell entries. This is noted in (2), but has not been strictly enforced allowing embedded mesh BACKGROUND cell and MATCELL cells to list the same pseudo-cell. This flexibility will be removed in a future version of the MCNP code to strictly require a unique BACKGROUND cell.

It is recommended to add a unique cell in the MCNP cell block that corresponds only to the BACKGROUND pseudo-cell (not included within the MATCELL pseudo-cell listing(s)).

Data-card Form: EMBED*n* keyword = *values(s)*

n	The universe number assigned to the embedded mesh, which must match a universe number specified on the pseudo-cell cards. ((1)).
BACKGROUND = *c*	Cell number of the background pseudo-cell. (Required)
MATCELL = *m1 c1 m2 c2 ...*	Integer material-cell pairs. (Required) For the structured mesh, there is one pair for each material (including void), where *mi* values are the embedded mesh material numbers and *ci* is the pseudo-cell number associated with *mi*. If the material is void, *mi* = 0 ((2), (3)). *mi* values must be in a LNK3DNT file. For the unstructured mesh there is one pair associating each pseudo-cell in the unstructured mesh (*mi*) file with one pseudo-cell in the MCNP cell block (*ci*). Pseudo-cells in the unstructured mesh are numbered sequentially, beginning with 1. A void, *mi* = 0, is not used by the unstructured mesh. See the pseudo-cell cross-reference table in the MCNP output file for the expected pairings or use the utility program to assist with problem setup. *mi* must start from 1 and increase by 1.
MESHGEO = *format*	Format specification of the embedded mesh input file. Required.

MESHGEO = LNK3DNT	the embedded (structured) geometry file is in LNK3DNT format.
MESHGEO = ABAQUS	the embedded (unstructured) geometry file is in Abaqus format [§8.7].
MESHGEO = MCNPUM	the Abaqus format has been converted to the format used internal to the MCNP code, now deprecated [DEP-53424].
MESHGEO = HDF5	the embedded (unstructured) geometry file is in an HDF5 file [§D.6].

MGEOIN = *filename*	Name of the input file containing the mesh description. (Required)
MEEOUT = *filename*	Name of the (legacy) EEOUT results file [DEP-53294] to write; if not specified, no legacy EEOUT file is written (Unstructured mesh only)
MEEIN = *filename*	Name of the (legacy) EEOUT results file [DEP-53294] to read; required for a restarted calculation if not using HDF5-based information (see HDF5FILE entry). Must not be the same as MEEOUT. Must not be present if the HDF5FILE entry is present.
CALC_VOLS = *value*	Select the volume calculation (Structured mesh only; Optional), if

`CALC_VOLS` = `YES`	calculate the inferred geometry cell volumes and masses.
`CALC_VOLS` = `NO`	do not calculate the inferred geometry cell volumes and masses.

`DENTYPE` = *value* Select the density type in a LNK3DNT file (Structured mesh only; Required), if

`DENTYPE` = `MASS`	densities in a LNK3DNT file are mass densities.
`DENTYPE` = `ATOM`	densities in a LNK3DNT file are atom densities.

`DEBUG` = *value* If `DEBUG` = `ECHOMESH`, write the embedded geometry parameters to the OUTP file. (No other values are supported at this time; Structured mesh only; Optional)

`ELEMENTCHK` = *value* Toggle UM elemental quality metric calculation and reporting (Unstructured mesh only; Optional), if

`ELEMENTCHK` = `YES`	calculate and report elemental quality metrics during UM input processing. (DEFAULT)
`ELEMENTCHK` = `NO`	do not assess elemental quality metrics during UM input processing.

`FILETYPE` = *type* File type for the elemental edit output file. (Unstructured mesh only; Optional) If

`FILETYPE` = `ASCII`	then write the elemental edit output file in ASCII format. (DEFAULT)
`FILETYPE` = `BINARY`	then write the elemental edit output file as a binary file.

`GMVFILE` = *filename* Name of the GMV output file.(Unstructured mesh only; Optional) General Mesh View (GMV) is an external program written to support mesh geometry visualization [228]. This option is now deprecated [DEP-53519].

`HDF5FILE` = *filename* File name for the HDF5-based elemental edit output file and prefix for the associated XDMF file. For example, specifying `HDF5FILE=eeout.h5` will produce two files: the binary HDF5 file `eeout.h5` and the ASCII XML file `eeout.h5.xdmf`. If this option is present, restart information is also written (currently to the `/restart/unstructured_mesh` group of the MCNP runtape file) and used (from the same location), as applicable. When the MCNP code is executed with an input option only, the binary HDF5 file and the ASCII XML file are created where these files contain only the mesh model.

`LENGTH` = *f* A multiplicative conversion factor to centimeters for all mesh dimensions in the input and output files (DEFAULT: $f = 1$) (Unstructured mesh only; Optional)

`MCNPUMFILE` = *filename*
Name of the MCNPUM output file. This option is deprecated [DEP-53424].

`OVERLAP` = *key value* Model to treat overlapping parts (4). First entry should be one of the following:

- `EXIT` (default when `OVERLAP` is not provided),
- `ENTRY`

- AVERAGE.

Treatments for individual pseudo-cells can be specified by following the initial entry with a second parameter and a list of valid pseudo-cell numbers (from the MATCELL entry).

Details:

1. This universe number must match those specified on the related inferred-cell cards.
2. The MATCELL keyword entries must have one m_1-c_1 pair for each structured material or unstructured mesh pseudo-cell in the embedded mesh. A unique pseudo-cell must exist in the MCNP cell block for each unique structured mesh material or unstructured mesh pseudo-cell [DEP-53361]. If there are void elements in the embedded structured mesh geometry, there must also be a MATCELL entry pair for material 0 and an associated pseudo-cell. The assigned BACKGROUND cell must be a unique pseudo-cell. A warning is issued if there are pseudo-cells that are not listed on the BACKGROUND or MATCELL entries. It is a fatal error if a structured mesh material or unstructured mesh pseudo-cell appears in the external mesh file that is not mapped to a pseudo-cell.
3. In the case of a structured mesh, while pseudo-cells associated with non-void embedded geometry mesh elements must have a specified density, MCNP6 uses the element-specific densities stored in the external mesh file for transport (e.g., cross-section lookup) and plotting. The density for the pseudo-cell on the cell card should be considered a reference density.
4. The OVERLAP treatments are currently ignored and the EXIT treatment is always applied.

5.5.6.7 EMBEE: Embedded Elemental Edits Control

If no EMBEE card is present, a cumulative elemental track-length fluence edit is created, without statistical uncertainties, for each particle on the MODE card.

Data-card Form: EMBEE*n*:$\mathscr{P}$ keyword = *values(s)*

n	Elemental edit number ending in 4, 6, or 7. These values follow the F4, F6, and F7 tally convention.
$\mathscr{P}$	Particle designator. Restriction: Only n or p or n,p or individual charged particles.
EMBED = *value*	Embedded mesh universe number. Must correspond to a valid EMBED card and mesh universe number. (Required)
COMMENT = *value*	Edit comment to appear in the legacy EEOUT file [DEP-53294] or the HDF5 EEOUT file; limited to 128 lower case alphanumeric characters. Similar functionality as FC card for tallies. (Optional)
ENERGY = *value*	Multiplicative conversion factor from MeV/g for all energy-related output. (DEFAULT: *value* = 1) (Optional)
ERRORS = *value*	Record relative fractional uncertainty to the legacy EEOUT file [DEP-53294] or HDF5 EEOUT file. NO or YES (default). (Optional)

TIME = *value*	Multiplicative conversion factor from shakes for all time-related output. (DEFAULT: *value* = 1) (Optional)
ATOM = *value*	Flag to multiply by atom density; NO (default) or YES. (Optional)
FACTOR = *value*	Multiplicative constant; default: 1.0; equivalent in concept to |C| on the FM card. (Optional)
LIST = *values*	Reaction list where this is the sum and/or product of ENDF or special reaction numbers. Limited to 1 reaction list as with FMESH tallies. Parentheses can be used but are ignored by the code. (Optional)
MAT = *value*	Material number identified on an M*n* card. Can be a dummy material or 0 (default). If the value is 0, use the cell material. (Optional)
MTYPE = *type*	Multiplier type (Optional). Acceptable character input values follow:

flux	Normal volume flux calculations. Same interpretation as FMESH tally *type* = flux. (default)
isotopic	Isotopic calculation. UM equivalent to the FMESH isotopic mesh tallies that require an +FM card.
population	Population calculation. Same as an F4 tally with an FM card where $k = -2$ in the multiplier set.
reaction	Reaction calculation that requires the LIST parameter. This MTYPE with the LIST parameters is equivalent to an FMESH tally with a single multiplier set specified and its accompanying FM card.
source	Accumulate source point locations. Same interpretation as FMESH tally *type* = source.
tracks	Tracks calculation. Same as an F4 tally with an FM card where $k = -1$ in the multiplier set.

5.5.6.8 EMBEB: Embedded Elemental Edit Energy Bin Boundaries

Data-card Form: EMBEB*n e1 e2 ... eK*

n	The elemental edit number from EMBEE card; $n = 0$ is not valid
ek	Upper energy (MeV) of the kth bin. List must be monotonically increasing. (DEFAULT: One energy bin with boundary set to the maximum energy limit for the particle type.)

5.5.6.9 EMBEM: Embedded Elemental Edit Energy Bin Multipliers

Data-card Form: EMBEM*n m1 m2 ... mK*

n	The elemental edit number from EMBEE card; $n = 0$ is not valid
mk	Multiplier for the kth energy bin. (DEFAULT: $mk = 1$)

5.5.6.10 EMBTB: Embedded Elemental Edit Time Bin Boundaries

Data-card Form: EMBTB*n t1 t2 ... tK*

n	The elemental edit number from EMBEE card; $n = 0$ is not valid
tk	Upper time (in shakes, where 1 shake $= 10^{-8}$ s) for the tth time bin. List must be monotonically increasing. (DEFAULT: One time bin with boundary set to the maximum time limit for the particle type.)

5.5.6.11 EMBTM: Embedded Elemental Edits Time Bin Multipliers

Data-card Form: EMBTM*n m1 m2 ... mK*

n	The elemental edit number from EMBEE card; $n = 0$ is not valid
mk	Multiplier for the kth time bin. (DEFAULT: $mk = 1$)

5.5.6.12 EMBDE: Embedded Elemental Edit Dose Energy Bin Boundaries

To avoid confusion and maintain the separability of edits from tallies, separate response-function-specification cards are available to implement response functions on UM edits. These are similar to the standard DE/DF cards, but there are no built in functions associated with these cards at this time.

Data-card Form: EMBDE*n e1 e2 ... eK*

n	The elemental edit number from EMBEE card; $n = 0$ is not valid
ek	Upper energy (MeV) for the kth energy bin. List must be monotonically increasing. (DEFAULT: one energy bin with boundary set to the maximum energy limit for the particle type.)

5.5.6.13 EMBDF: Embedded Elemental Edits Dose Function Bin Multipliers

Data-card Form: EMBDF*n m1 m2 ... mK*

n	The elemental edit number from EMBEE card; $n = 0$ is not valid
mk	Multiplier for the kth energy bin. (DEFAULT: $mk = 1$)

5.5.6.13.1 Example 1

A LNK3DNT file can be generated with the MCNP input file in Listing 5.17 and then used by the MCNP input file in Listing 5.18.

Listing 5.17: example_structured_mesh_generate_1.mcnp.inp.txt

```
Generate LNK3DNT file
1 1 -18.7 -1 imp:n=1
2 0        1 imp:n=0

1 rpp -10 10  -10 10  -10 10

kcode 5000 1.0 50 250
ksrc -5.0  0.0  0.0     5.0  0.0  0.0
      0.0 -5.0  0.0     0.0  5.0  0.0
      0.0  0.0 -5.0     0.0  0.0  5.0
totnu no
m1        92235.69c 1.0
dm1 92235 92235.50
prdmp    j    275
mesh  geom=xyz ref=0.0 -0.0 -0.0 origin=-10.0 -10.0 -10.0
      imesh=10 iints=3
      jmesh=10 jints=4
      kmesh=10 kints=5
dawwg points=10
      block=1 ngroup=16 isn=16 iquad=4
      block=3 libname=mendf5 lib=ndilib
      block=5 trcor=diag srcacc=dsa diffsol=mg isct=2
      block=6 massed=1 edoutf=3
rand gen=2 seed=12345
print
```

Listing 5.18: example_structured_mesh_read_1.mcnp.inp.txt

```
Read LNK3DNT file
11 1 -18.7  0    u=e10 imp:n=1
12 0        0    u=e10 imp:n=1 $ background
20       0 -1 fill=e10 imp:n=1
c
21       0  1          imp:n=0

1  s 0.0 0.0 0.0   15

kcode 5000 1.0 50 250
ksrc -5.1  0.1  0.1     5.1  0.1  0.1
      0.1 -5.1  0.1     0.1  5.1  0.1
      0.1  0.1 -5.1     0.1  0.1  5.1
totnu no
m1 92235.69c 1.0
```

```
m2  1001.60c 1.0
prdmp j 275
print
rand gen=2 seed=12345
embed10 meshgeo=lnk3dnt mgeoin=linkout debug=echomesh
        calc_vols=yes background=12 matcell= 1 11
```

5.5.6.13.2 Example 2

Similar to the last example's geometry, an unstructured mesh using first-order hexahedra can be defined and used in the MCNP input file shown in Listing 5.19. The Abaqus-formatted mesh input file is electronically attached to this document as **example_unstructured_mesh.abaq.inp.txt**.

Listing 5.19: example_unstructured_mesh.mcnp.inp.txt

```
Example unstructured mesh
11 1 -18.7     0     u=2 imp:n=1
12 0           0     u=2 imp:n=1 $ background
20 0         -100 fill=2 imp:n=1
21 0          100        imp:n=0

100  so 25.0

kcode 5000 1.0 50 250
ksrc -5.1  0.1  0.1     5.1  0.1  0.1
      0.1 -5.1  0.1     0.1  5.1  0.1
      0.1  0.1 -5.1     0.1  0.1  5.1
totnu no
m1 92235.69c 1.0
print
rand gen=2 seed=12345
embed2 meshgeo=abaqus mgeoin=example_unstructured_mesh.abaq.inp.txt
       meeout=example_unstructured_mesh.eeout
       hdf5file=example_unstructured_mesh.eeout.h5
       background= 12 matcell= 1 11
```

The EMBED card is allowed in a restarted embedded unstructured mesh problem. This allows for the previous elemental edit output file to be read in as the elemental edit input file (now deprecated functionality) or from the MCNP runtape file and for a new name to be assigned to the newly created elemental edit output file, as shown in Listing 5.20 and is executed with the command (depending on the runtape name): `mcnp6 c r= runtpe.h5 i= example_unstructured_mesh_continue.mcnp.inp.txt`.

Listing 5.20: example_unstructured_mesh_continue.mcnp.inp.txt

```
continue
c
kcode 5000 1.0 50 300
embed2 meshgeo=abaqus mgeoin=example_unstructured_mesh.abaq.inp.txt
       meeout=example_unstructured_mesh_continue.eeout
       hdf5file=example_unstructured_mesh_continue.eeout.h5
       background= 12 matcell= 1 11
```

5.6 Material-focused Data Cards

Material cards within the MCNP code serve two purposes. The first is to specify the composition of the materials. The second is to select the data libraries to use during simulation. The M card is used to create a material and define its composition. It also has a few options to define material-specific properties to use during transport. The MT card allows one to attach a neutron thermal scattering $S(\alpha, \beta)$ table to a material. Finally, the MX card allows one to alter targets and tables on a per-physics basis.

5.6.1 M: Material Specification

The material card allows a user to input the composition of a given material. The main input consists of pairs of targets and concentrations. Targets can be specified using target identifiers [§1.2.2], or using table identifiers [§1.2.3]. Concentrations can be specified by either mass or weight fraction. Additionally, there are key-value pairs on this card that set specific material properties.

⚠ Caution

M card keywords may appear anywhere among the target-fraction pairs, but must not separate a pair.

The M card also doubles as a mechanism for selecting data tables. The MCNP code loads data using a process described in §2.3.1. Each target listed on the M card is added to the data search list. The necessary physics identifiers are inferred from the MODE and PHYS cards. The library identifiers can be input on the M card in three ways. The first is by using a full table identifier instead of a target identifier. This library identifier will only be used for the corresponding physics. Second, to select libraries for each physics, the *x*LIB (NLIB, PLIB, PNLIB, ELIB, HLIB, ALIB, SLIB TLIB, DLIB) key-value options can be used. When set, the value will be used for every target in the material. Third, a special material card, M0, can be used to set the default for all materials.

The methods one can use to select the library, in decreasing priority, are:

1. Setting a full table identifier on the MX card for a given physics.
2. Setting a full table identifier on an M card.
3. Setting the *x*LIB key-value options on an M card.
4. Setting the *x*LIB key-value options on an M0 card.

The *x*LIB options are the only way to specify which data libraries to use for the nuclides that are implicitly added to the simulation through transmutation during BURN calculations. Without *x*LIB, the first library listed in the **xsdir** is always used for those nuclides.

It is recommended to always specify compositions by nuclide or isomer rather than by element (in which A is set to zero) for two reasons. First, in simulations with nuclear interactions, inputting elemental identifiers results in using older data because current practices are to provide only isotopic data files, so elemental data files are generally out of date and less accurate. Second, atom densities are computed using the elemental atomic mass listed in the **xsdir** file. For elements in which there exists a natural composition, this value is computed using that composition. For elements for which no natural composition exists, the elemental mass is specified, but no pedigree is available. This can result in unexpected behavior.

Data-card Form: `Mm z1 f1 z2 f2 ... zK fK keyword = values(s)`

m	Arbitrary material number; same as material number, *m*, on a cell card [§5.2]. When $m = 0$, keyword entries on that card are applied to all other M cards. Restriction: $0 \leq m \leq 99,999,999$.
zk	Either a target identifier [§1.2.2] or a table identifier [§1.2.3] for constituent k. All forms of target identifier are supported.
fk	Fraction of the kth constituent in the material (1, 2), where *fk* > 0 — indicates that the value is interpreted as an atomic fraction and *fk* < 0 — indicates that the value is interpreted as the weight fraction. Atomic and weight fractions may not both appear on a single M card.
GAS = *value*	Optional, flag for density-effect correction to electron stopping power. If GAS = 0 — the code calculates a density-effect correction appropriate for material in the condensed (solid or liquid) state (DEFAULT), or GAS = 1 — the code calculates a density-effect correction appropriate for material in the gaseous state.
ESTEP = *n*	Causes the number of electron sub-steps per energy step to be increased to *n* for the material. If *n* is smaller than the built-in default found for this material, the entry is ignored. Both the default value and the ESTEP value actually used are available in PRINT Table 85 of the output file. (DEFAULT: internally set)
HSTEP = *n*	Causes the number of proton or other charged-particle sub-steps (exclusive of electrons, but including heavy ions) per energy step to be increased to *n* for the material. If ESTEP is specified and HSTEP is not, then the ESTEP value is used for HSTEP. Both the default value and the HSTEP value actually used are available in PRINT table 85 of the output file. (DEFAULT: internally set)
NLIB = *x*	Changes the default neutron table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)
PLIB = *x*	Changes the default photoatomic table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)
PNLIB = *x*	Changes the default photonuclear table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)
ELIB = *x*	Changes the default electron table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)
HLIB = *x*	Changes the default proton table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)
ALIB = *x*	Changes the default alpha table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)
SLIB = *x*	Changes the default helion table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)

TLIB = *x*	Changes the default triton table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)
DLIB = *x*	Changes the default deuteron table identifier to the string *x* (DEFAULT: blank string, which selects the first matching entry in the **xsdir** file)
COND = *value*	Sets conduction state of a material only for the EL03 electron-transport evaluation. If

COND < 0	then the material is a non-conductor.
COND $= 0$	then the material is a non-conductor if there is at least one non-conducting component; otherwise it is a conductor (DEFAULT)
COND > 0	then the material is a conductor if there is at least one conducting component.

REFI = *A*	Constant refractive index
REFI = *A B C D*	Cauchy coefficients (units are micrometers) for refractive index that are used to calculate $$n(\lambda) = A + \frac{B}{\lambda^2} + \frac{C}{\lambda^4} + \frac{D}{\lambda^6}.$$
REFS = *B1 C1 B2 C2 B3 C3*	Sellmeier coefficients for refractive index that are used to calculate $$n^2(\lambda) = 1 + \frac{B1 \cdot \lambda^2}{\lambda^2 - C1} + \frac{B2 \cdot \lambda^2}{\lambda^2 - C2} + \frac{B3 \cdot \lambda^2}{\lambda^2 - C3}.$$

Use: Required if you want materials in cells. Recall that an equals sign (=) following a keyword, such as the *x*LIB keywords, is optional. Inclusion of the decimal point or the physics identifier in the library *x* designation (e.g., .70c as opposed to 70) is permitted, but not required.

Details:

1. The nuclide fractions can be normalized to 1.0 or left unnormalized, in which case the code performs the normalization.
2. The code uses the atomic weight ratio values from the transport table to convert mass fractions to atom fractions. To avoid this conversion and therefore ensure the most accurate material representation, it is recommended that atom fractions be specified.

5.6.1.1 Example 1

```
M1   C-12.00c 1   O-16.12p 2   NLIB=80c  PNLIB=24u  PLIB=14p
```

In this example, there are two components to this material, one part ^{12}C and two parts ^{16}O by atom fraction. For ^{12}C, the neutron library identifier `00` is explicitly set. For ^{16}O, the photoatomic library identifier `12` is

Table 5.7: M Card Example 1 Table Listing

Physics	^{12}C	^{16}O
Neutron	6012.00c	8016.80c
Photoatomic	6000.14p	8000.12p
Photonuclear	6012.24u	8016.24u
Electron	First `e` dataset with $Z = 6$, $A = 0$	First `e` dataset with $Z = 8$, $A = 0$
Proton	First `h` dataset with $Z = 6$, $A = 12$, $S = 0$	First `h` dataset with $Z = 8$, $A = 16$, $S = 0$

explicitly set. The neutron default library identifier is set to `80`, the photonuclear default library identifier is set to 24, and the photoatomic default library identifier is set to `14`.

Given a simulation with neutron, photon, electron, and proton physics, the tables shown in Table 5.7 will be loaded. There are several notes here. First, most tables are historically identified in ZAID format [§1.2.2]. The MCNP code converts between target identifier formats to load 6012 for C-12 and 8016 for O-16. Second, for atomic data, A is automatically set to zero. Photoatomic and electron data search for C-0 and O-0 instead of C-12 and O-16. Third, the library identifier attached to the target identifier overrides the *`xLIB`* options. This occurs for ^{12}C neutron data and ^{16}O photoatomic data. Finally, as neither electrons nor protons have their library specified, the first value in the **`xsdir`** will be used based on the listed rules.

5.6.1.2 Example 2

```
M1  NLIB=50D    H-1  2    O-16.50C  1   C-12  1
```

This material consists of three isotopes. Hydrogen (H-1) and carbon (C-12) are not fully specified and will use the default neutron table that has been defined by the `NLIB` entry to be 50d, the discrete-reaction library. Oxygen (O-16.50c) is fully specified and will use the continuous-energy library 8016.50c. The same default override hierarchy applies to proton, photonuclear, photon, and electron specifications.

5.6.1.3 Example 3

```
M1    Ag-110m  1
```

In this example, the material consists of ^{110m}Ag. In older files, this may be represented by the ZAID 47510, where $Z = 47$, $A = 110$, and the identifier is incremented by $300 + 100S = 400$ to indicate the first metastable state.

5.6.1.4 Example 4

```
M1  H-1 2   O-16 1  REFI=1.3199
```

Water with a constant refractive index of 1.3199.

5.6.1.5 Example 5

```
M1 H-1 2 O-16 1 REFC=1.3119 6.878e-2 1.132e-3 1.11e-4
```

Water with a refractive index specified by coefficients for a 4th-order Cauchy expression. The coefficients are in units of micrometers.

5.6.1.6 Example 6

```
M1 Si-28 1 O-16 2
     REFS = 1.0396 6e-3 0.2318 2.0018e-2 1.0104 1.0356e2
```

Approximate borosilicate crown glass (cf. [material 156 of 229]) with a refractive index specified by coefficients for Sellmeier's equation. Sellmeier coefficients are applied directly; they are not squared.

5.6.2 MT: $S(\alpha, \beta)$ Thermal Neutron Scattering

It is possible to treat nuclides within a material as if they were a molecule or crystalline solid by applying $S(\alpha, \beta)$ neutron thermal scattering data. This data, described in more detail in §2.3.6, replaces portions of the neutron physics at low energies, below approximately 10 eV depending on library, with those altered by molecular binding forces.

The MT card is specified for each material as needed. The input is a list of $S(\alpha, \beta)$ table identifiers [§1.2.3]. $S(\alpha, \beta)$ table identifiers are slightly different from typical identifiers, in that the target is an arbitrary string due to the many possible molecules or crystals that can be represented. Modern libraries use a "target-molecule" format for the target identifiers. As an example, `H-H2O` would alter the hydrogen physics to correspond to molecular interactions in water. The physics identifier is `t` for this type of data.

The MT card will override the physics of all nuclides marked as a target. In the case of `H-H2O`, all hydrogen tables will be altered to include water binding energy. It is not possible for two tables to alter a single target or to alter only a fraction of a material. If multiple tables for a given target are loaded, the MT0 card can be used to match a specific $S(\alpha, \beta)$ table to a specific target table such as when stochastic temperature mixing is used.

Data-card Form: MT*m* *sabid1* *sabid2* ... *sabidK*

m	Material identifier, same as *m* on the corresponding material (M*m*) card.
sabidk	$S(\alpha, \beta)$ identifier [§1.2.3] corresponding to a particular target on the M*m* card. $S(\alpha, \beta)$ contributions to detectors (F5 tallies) and DXTRAN spheres (DXT card) are approximate.

Default: None.

Use: Essential for problems with thermal neutron scattering.

5.6.2.1 Example 1

```
M1    H-1  2    O-16  1  $ light water
MT1   H-H2O.40t
```

In this example, all hydrogen in material 1 will be altered with the `H-H2O.40t` table.

5.6.2.2 Example 2

```
M8    C-12  1             $ graphite
MT8   GRPH.47t
```

In this example, all carbon in material 8 will be altered with the `GRPH.47t` table.

5.6.2.3 Example 3

When a particle is within the energy regime at which the $S(\alpha, \beta)$ treatment applies, the specification

```
M1   H-1 2  O-16 1  Be-9 1e-3  $ light water w/ small amt of Be
MT1  H-H2O.40t BE-MET.40t
```

will substitute the light-water $S(\alpha, \beta)$ library for the hydrogen and the beryllium metal library for the beryllium.

However, the specification

```
M1   Be-9 2  O-16 1       $ Be oxide
MT1  BE-MET.40t BE-BEO.40t
```

will not work as desired because both libraries will try to substitute for the beryllium in the problem. Only the first $S(\alpha, \beta)$ specification (for `BE-MET.40t`) will be used.

5.6.2.4 MT0 Card: $S(\alpha, \beta)$ Special Treatment for Specific Isotopes

Some MCNP input files make use of an old, traditional method for dealing with material temperatures called "stochastic mixing." When a material is used in a cell that has a temperature in-between the temperatures used by NJOY in producing the ACE files, users can approximate the temperature effects on cross-section data by including both a "hot" version and a "cold" version of the ACE data used for each isotope in the material. For example, if there are ACE files available at 293.6K and 600K, and a cell has a temperature of 446.8K (halfway between the available ACE files), then an approximate way to model the material is to include each isotope twice - once using a target at the lower ACE file temperature and once using a target for the ACE file at the higher temperature, with 50% of the atom or weight fractions used for each of the

bounding targets. During transport, the MCNP code will select the "hot" target half the time and the "cold" target the other half, stochastically, so that the average for the mixture matches the cell temperature. This approach is approximate, because it is not interpolation based on physics, just a stochastic mixing of the bounding data.

The use of stochastic mixing for ordinary cross-section data is routine and straightforward. For the $S(\alpha,\beta)$ ACE datasets, however, there is the complication that the "hot" $S(\alpha,\beta)$ data must be specifically associated with the "hot" identifier and the "cold" $S(\alpha,\beta)$ data must be specifically associated with the "cold" identifier. To make the correct association of $S(\alpha,\beta)$ data tables to nuclear data tables, the MT0 card can be used to fully specify the $S(\alpha,\beta)$ assignments.

The format of the MT0 card is:

Data-card Form: MT0 *sabid1 identifier_1 ... sabidK identifier_k*

sabidk	$S(\alpha,\beta)$ dataset identifier [§1.2.3]. The library identifier and physics identifier must be explicitly included.
identifier_k	material target identifier. The library identifier and physics identifier must be explicitly included.

Default: None.

Use: Essential for problems with thermal neutron scattering where $S(\alpha,\beta)$ datasets and nuclear datasets are specified at more than 1 temperature for a single material.

The entries in MT0 consist of pairs of $S(\alpha,\beta)$ dataset identifiers and nuclear data identifiers. Whenever the $S(\alpha,\beta)$ dataset is requested on an MT card, it will only be used to alter the physics for the data in the corresponding nuclear data library. Libraries listed on this card must also be fully specified on their corresponding M and MT cards. Users should check PRINT Table 102 to verify the correct assignments in the materials.

5.6.2.5 Example 4

A material is at 446.8K, halfway between ACE files available at 293.6K and 600K. Water may be represented in the following approximate manner using stochastic mixing:

```
mt0    h-h2o.40t  H-1.00c            $ S(a,b)-table matching at 293.6K
       h-h2o.54t  H-1.01c            $ S(a,b)-table matching at 600K
c
m100   H-1.00c 1.0   O-16.00c 0.5    $ at 293.6K
       H-1.01c 1.0   O-16.01c 0.5    $ at 600K
mt100  h-h2o.40t  h-h2o.54t
```

Each nuclide is included twice, with half the required fraction for each, and both $S(\alpha,\beta)$ datasets are included on the MT100 card. The pairs of identifiers from the MT0 card are used to provide the proper assignments for the material. The h-h2o.40t data is associated with the H-1.00c nuclide, and the h-h2o.54t data is associated with the H-1.01c nuclide. PRINT Table 102 is used to verify that these assignments were made.

5.6.3 MX: Material Card Nuclide Substitution

The MCNP nuclide substitution capability [230] enables mixing of physics models and data tables for individual isotopes. Different nuclides can be substituted for different particle types. For example, natural carbon and calcium can be used for neutrons, whereas ^{12}C and ^{40}Ca can be used for protons and photonuclear reactions.

Above tabular data limits, models are automatically called in the MCNP code. The model to be used depends on values set on the LCA card. The physics models can be disabled by the MPHYS card. The sole exception is photonuclear interactions, for which CEM03.03 [201, 207, 208, 212, 216] is always used regardless of whether CEM03.03 is used for other particles (see the LCA card). Using the term MODEL on an MX card will substitute model physics for the entire energy range of a particle—no tabular data will be used. This option should be carefully considered before use. The parameter 0 on an MX card eliminates all interaction physics, whether model or table-based. This makes sense in the case of photonuclear interactions on hydrogen, which do not exist in nature, but should be avoided for other cases.

Data-card Form: MX*m*:$\mathscr{P}$ *z1 z2 ... zK*

m	Material identifier, same as *m* on the corresponding material (M*m*) card. The MX*m* card must appear after its associated (M*m*) material card.
$\mathscr{P}$	Particle designator [Table 4.3]; allowed values are neutron (n), photonuclear (p), proton (h), deuteron (d), triton (t), hellion (s), and alpha (a).
zk	Varies behavior such that if
zk = *table_identifier*	[§1.2.3], then substitute the specified library for the *k*th nuclide identifier on the M card. All target formats are supported.
zk = *target_identifier*	[§1.2.2] then substitute the *k*th nuclide on the M card with the nuclide listed.
zk = MODEL	then substitute model physics for the *k*th nuclide on the M card. A mixture of models and tabular data may be specified for nuclides on a single M card.
zk = 0	for a photonuclear substitution card (MX*m* : P), then omit photonuclear reactions for *zk*. This option is only available for photonuclear particles.
	No substitutions are allowed for photoatomic (P) and electron (E) data because these data depend only on Z and are not isotope-specific.

Use: The MX*m* card enables nuclide substitution for different particle types. The nuclide replacement capability is particularly useful for photonuclear and proton calculations when few data tables are available. Libraries are used when available and models are used otherwise.

5.6.3.1 Example 1

```
MODE      n  h  p
M3        H-2   1  H-3   1  C-12 1  Ca-40 1
```

```
MX3:N     j        MODEL    C-0      Ca-0
MX3:H     MODEL    H-1      j        j
MX3:P     C-12     0        j        j
```

In this example, models will be used for neutrons on tritium and protons on deuterium. Natural libraries will be used for neutron interactions on carbon and calcium. A model will be used for proton interactions for deuterium, and protons on tritium will substitute the hydrogen cross section. For photonuclear, ^{12}C substitutes for deuterium and the cross section for tritium interactions will be set to 0.0.

5.6.3.2 Example 2

```
m1         O-16   1.0
          Pb-206  10.0
            nlib=.60c
            hlib=.24h
           pnlib=.24u
mx1:h   j  Fe-56.70h
mx1:n   j Ra-223.70c
mx1:p   j Pu-239.70u
```

For ^{16}O of material 1, the MCNP code will use the neutron, proton, and photonuclear cross-section data files, `O-16.60c`, `O-16.24h` and `O-16.24u`, respectively. For ^{206}Pb of material 1, the `MX` cards specify that data file `Ra-223.70c` will be substituted for `Pb-206.60c`, `Fe-56.70h` for `Pb-206.24h`, and `Pu-239.70u` for `Pb-206.24u`.

5.6.4 MPN: Photonuclear Nuclide Selector

Deprecation Notice **DEP-53483**

This feature has been replaced by the material card nuclide substitution (`MX`) capability. To control the selection of photonuclear nuclide data, use the `MX` card.

5.6.5 OTFDB: On-the-fly Doppler Broadening

The MCNP code has a capability for on-the-fly (OTF) Doppler broadening of neutron cross sections. Background, theory and methodology, and implementation details are provided in several references [231–234].

To use the OTF Doppler broadening, data tables with temperature-fitting coefficients must first be prepared using the **`fit_otf`** code. This code is included in the MCNP distribution in the **`MCNP_CODE/Utilities/FIT_OTF`** directory. Input specifications and examples for running **`fit_otf`** are available in §E.3. Running the **`fit_otf`** code will produce a file of OTF coefficients in either a binary or a text file format. These files have names of the form

- Binary: **`otf_92235.70c.binary`**, **`otf_8016.70c.binary`**, etc.
- Text: **`otf_92235.70c.txt`**, **`otf_8016.70c.txt`**, etc.

The identifier (with suffix) that is part of the file name refers to the original identifier for the base dataset used as input to **fit_otf** (not necessarily to the identifier used in an MCNP input file). The files generated by **fit_otf** for various nuclides should be placed in the DATAPATH directory or in the working directory. Alternatively, symbolic links to the files could be placed in the DATAPATH directory, with the actual files located elsewhere.

The OTFDB card is used to provide the MCNP code with the list of OTF data files that should be used in a calculation. The table identifier portion of the file names should be supplied, including the suffix.

Data-card Form: OTFDB *z1 z2 ... zK*

zk	are the identifiers for OTF Doppler broadening data tables. These identifiers may use any format listed in [§1.2.3], so long as they match the OTF database filename.

In the MCNP input processing, identifiers specified on the material input (M) cards are matched with available identifiers from the OTFDB card. This is done by decomposing the Z, A, and S of both the M card input and the OTFDB input. If they match, the OTFDB data replaces the low-energy portion of the corresponding dataset. Library identifiers are ignored. The OTFDB data is broadened to the value provided on the TMP card for each cell that contains the affected material. The TMP value must be higher than the minimum temperature of the library for all cells, including void cells and cells with zero importance.

5.6.5.1 Example 1

```
1 OTFDB     U-235.70c   O-16.70c
```

This input loads the files **otf_U-235.70c.txt** and **otf_O-16.70c.txt** or their binary equivalent. For all ^{235}U and ^{16}O in the problem, the loaded OTF data will be used to provide temperature dependence.

5.6.6 TOTNU: Total Fission

Caution

Former MCNP5 users need to be aware that the default behavior of this card has changed to total $\overline{\nu}$.

Data-card Form: TOTNU *value*

value	If	
	value = blank	then use total $\overline{\nu}$, which samples both prompt and delayed fission neutrons, for all fissionable nuclides for which prompt and delayed values are available.
	value = NO	then use only prompt $\overline{\nu}$ for all fissionable nuclides for which prompt values are available.

Default: If the TOTNU card is absent or if a TOTNU card is present but has no entry after it (*value* is not specified), total $\overline{\nu}$, which samples both prompt and delayed fission neutrons, is used for all fissionable nuclides

for which prompt and delayed values are available. Thus, the TOTNU card is not needed unless only prompt $\overline{\nu}$ is desired.

Use: Needed to specify use of only prompt $\overline{\nu}$. A TOTNU card with NO as the *value* causes prompt $\overline{\nu}$ to be used for all fissionable nuclides for which prompt values are available, ignoring delayed neutrons from fission.

5.6.7 NONU: Disable Fission

The NONU card provides the ability to disable fission in a cell. The fission is then treated as simple capture and is accounted for on the loss side of the problem summary as the "Loss to fission" entry. The NONU card is not allowed in a restarted calculation.

Cell-card Form: NONU = *a*
or
Data-card Form: NONU *a1 a2 ... aJ*

a	If	
	$a = 0$	then fission in cell treated as capture; gammas produced.
	$a = 1$	then fission in cell treated as real; gammas produced.
	$a = 2$	then fission in cell treated as capture; gammas not produced (1).
	blank	then fission in the cells is treated like capture; gammas produced (i.e., $a = 0$).
aj	Number of entries equals the number of cells unless no entry appears. If	
	$aj = 0$	then fission in cell treated as capture; gammas produced.
	$aj = 1$	then fission in cell treated as real; gammas produced.
	$aj = 2$	then fission in cell treated as capture; gammas not produced (1).
	blank	then fission in the cells is treated like capture; gammas produced (i.e., $aj = 0$).

Default: If the NONU card is absent, fission is treated as real fission ($aj = 1$). If the card is present but without entries, fission is treated as capture with gammas produced ($aj = 0$).

Use: Needed with SSR for fissioning neutron problems only. When fission is already modeled in the source, such as SSR, it should not be duplicated in transport and should be turned off with NONU. Use $aj = 2$.

Details:

(1) Note 1: An *aj* value of 2 treats fission as capture and, in addition, no fission gamma rays are produced. This option should be used with KCODE fission source problems written to surface source files. Suppressing the creation of new fission neutrons and photons is necessary because they are already accounted for in

the source. Consider a problem with a fixed source in a multiplying medium. For example, an operating reactor power distribution could be specified as a function of position in the core either by an SDEF source description or by writing the fission source from a KCODE calculation to a WSSA file with a CEL option on an SSW card. Without the ability to turn off fission, the non-KCODE calculation would be impossible to run because of the criticality of the system and because fission neutrons have already been accounted for. Using the NONU card in the non-KCODE mode allows this problem to run correctly by treating fission as simple capture.

5.6.8 AWTAB: Atomic Weight

Entries on this card override the existing atomic weight ratios as contained in both the **xsdir** file and the cross-section tables. The AWTAB card is needed when atomic weights are not available in an **xsdir** file.

Data-card Form: AWTAB *z1 a1 z2 a2 ... zK aK*

zk	Nuclide or element identifier [§1.2.2].
ak	Atomic weight ratios.

Default: If the AWTAB card is absent, the atomic weight ratios from the **xsdir** file and cross-section tables are used.

Use: Discouraged. Occasionally useful when XS card introduces rare isotopes.

⚠ Caution

Using atomic weight ratios different from the ones in the cross-section tables in a neutron problem can lead to negative neutron energies that will cause the problem to terminate prematurely.

5.6.9 XS: Cross-Section File

The XS*n* card can be used to load cross-section evaluations not listed in the **xsdir** file. The XS*n* cards can be used in addition to the **xsdir** file. Each XS*n* card describes one cross-section table. The entries for the XS*n* card are identical to those that appear in the default cross-section directory file (i.e., **xsdir_mcnp6.3**, see Appendix B), except that the "+" is not used for continuation.

Data-card Form: XS*n z1 a1 z2 a2 ...*

n	Arbitrary cross-section identification number. Restriction: $1 \leq n \leq 99,999,999$.
zk	Full table identifier [§1.2.3] used on the M material card. All target formats are allowed.
ak	Atomic weight ratio associated with nuclide k.
...	Remaining **xsdir** file entries for the user-provided cross-section table as described in Appendix B.

Use: Add an **xsdir**-type entry for nuclides not represented in the **xsdir** file.

5.6.10 VOID: Material Void

Data-card Form: `VOID` *c1 c2 ...*

cj	The list of cells to treat as void.

Default: Use problem materials.

Use: Debugging geometry and calculating volumes stochastically.

Details:

1. When the `VOID` card is blank, the material number and density is set to zero for all cells, `FM` cards are turned off, heating tallies are turned into flux tallies, and, if there is no `NPS` card, the effect of an `NPS` 100000 card is created. If there is a **`TALLYX`** subroutine, it may need to be changed, too.
2. Entries on the `VOID` card selectively set the material number and density to zero for the specified cells. Can be used to check whether the presence of some object in your geometry makes a significant difference in the results.

5.6.11 MGOPT: Multigroup Adjoint Transport Option

"`J`" is not an acceptable value for any of the `MGOPT` card parameters. Further, *mcal* and *igm* must be specified.

Data-card Form: `MGOPT` *mcal igm iplt isb icw fnw rim*

mcal	Setting *mcal* to `F` specifies a forward problem and setting *mcal* to `A` specifies an adjoint problem (2).
igm	The total number of energy groups for all kinds of particles in the problem. A negative total indicates a special electron-photon problem (3).
iplt	Indicator of how weight windows are to be used. If
	iplt = 0 — then `IMP` values set cell importance. Weight windows, if any, are ignored for cell importance splitting and Russian roulette. (DEFAULT)
	iplt = 1 — then weight windows must be provided and are transformed into energy-dependent cell importance. A zero weight-window lower bound produces an importance equal to the lowest non-zero importance for that energy group.
	iplt = 2 — then weight windows do what they normally do.
isb	Controls adjoint biasing for adjoint problems; valid only for *mcal* is `A`. If
	isb = 0 — then collisions are biased by infinite-medium fluxes. (DEFAULT)
	isb = 1 — then collisions are biased by functions derived from weight windows, which must be supplied.

	isb = 2	then collisions are not biased.
icw	Name of the reference cell for generated weight windows. If	
	icw = 0	then weight windows are not generated. (DEFAULT)
	icw ≠ 0	then volumes must be supplied or calculated for all cells of non-zero importance.
fnw	Normalization value for generated weight windows. The value of the weight-window lower bound in the most important energy group in cell *icw* is set to *fnw*. (DEFAULT: *fnw* = 1)	
rim	Compression limit for generated weight windows. Before generated weight windows are printed out, the weight windows in each group separately are checked to see that the ratio of the highest to the lowest is less than *rim*. If not, they are compressed. (DEFAULT: *rim* = 1000)	

Use: Required for neutron multigroup calculations.

Details:

1. Presently, the standard MCNP6 multigroup neutron cross sections are given in 30 groups and photons are given in 12 groups. Thus, an existing continuous-energy input file can be converted to a multi-group input file simply by adding one of the following cards:

```
MGOPT F 30 $ MODE N
MGOPT F 42 $ MODE N P
MGOPT F 12 $ MODE P
```

2. An input file for an adjoint problem can have both an IMP card and weight-window cards (*iplt* = 0 and *isb* = 1). The entries on the weight-window cards are not weight windows in the normal sense but biasing functions. If *iplt* = 1, the values on a weight-window card become energy-dependent cell importance.

3. A negative *igm* value allows a single cross-section table to include data for more than one sort of particle. This feature applies currently to electron/photon multigroup calculations only. A problem with 50 electron groups followed by 30 photon groups in one table would have *igm* = −80. Also, all tables must have the same group structure. A negative *igm* value will use the energy variable on the source or tally card as groups index unless it is associated with a distribution. For an energy distribution on the source card, there should be *igm* increasing integer entries for each group on the SI card. On a tally energy card, if there are fewer than *igm* entries, they will be taken as energies in MeV; otherwise, the bins will be according to group index. The particles can be separated in tallies by using the PTT keyword on the FT*n* tally special treatment card.

5.6.12 DRXS: Discrete-Reaction Cross Section

If the necessary discrete data are available, nuclides listed on the optional DRXS card are given a discrete energy treatment instead of the regular fully continuous-energy cross-section treatment.

All discrete reaction libraries are based on a 262-energy-group structure. Groups below 1 eV make the discrete treatment appropriate for thermal neutron problems near room temperature. All discrete reaction libraries have photon production data given in expanded format.

Data-card Form: DRXS *z1 z2 ... zK*

zk	is a target identifier [§1.2.2]. All target formats are allowed.

Default: Continuous-energy cross-section treatment if DRXS is absent. If the DRXS card is present but has no entries after the mnemonic, discrete cross sections will be used for every nuclide, if available.

Use: Discouraged. Applies only to neutron cross sections. It is not recommended that this card be used unless you are transporting neutrons in an energy region where resonances and hence self-shielding are of little importance. If the problem under consideration meets this criterion, using the DRXS card can reduce computer storage requirements and enhance timesharing.

Details:

1. Use of these discrete cross sections will not result in the calculation being what is commonly referred to as a multigroup Monte Carlo calculation because the only change is that the cross sections are represented in a histogram form rather than a continuous-energy form. The angular treatment used for scattering, energy sampling after scattering, etc., is performed using identical procedures and data as in the continuous-energy treatment. The user wanting to make a truly multigroup Monte Carlo calculation should use the MGOPT card multigroup capability.

5.7 Physics-focused Data Cards

The data provided in this section describe the physics options that can be selected.

5.7.1 MODE: Problem Type

The MODE card is used to specify the list of all particles that will be transported. The selection of valid particle-identifier values can be found in the "Symbol" column of Table 4.3, with the exception of positrons (f), which are invalid on the MODE card. The listed individual particle identifiers must be space-delimited. The ordering of particles listed does not matter.

In addition to the particle designators in Table 4.3, anti-particles may be designated by placing a "-" in front of the particle identifier. For example, MODE h -h, MODE h g, and MODE g -g are all valid ways to specify both proton (h) and anti-proton (g).

Data-card Form: MODE $\mathscr{P}_1$ $\mathscr{P}_2$ ... $\mathscr{P}_K$

$\mathscr{P}_k$	Particle designator.

Default: If the MODE card is omitted, MODE n is assumed.

Use: Optional for neutron-only simulations. Required for any other non-neutron or mixed-particle simulations.

Details:

(1) The # symbol represents all possible heavy ions. Although the # is generic to all heavy ions, the identities of different heavy ions are tracked by their appropriate Z (charge) and A (mass number). The user cannot choose to transport any particular heavy ion; however, the user may specify individual ions as source particles (see `par` keyword on the SDEF card) and may request tallies for specific ions (see *res* option on the FT special treatment tally card, §5.9.18.12).

(2) If heavy ions (#) are specified on the MODE card, any residuals produced from any model physics will be transported even if the source particle is not a heavy ion.

5.7.2 PHYS: Particle Physics Options

5.7.2.1 Neutrons (PHYS:n)

Caution

The PHYS:n data card entries are different for MCNP6 than they were for MCNP5 or MCNPX. In particular, the MCNPX PHYS:n 5th entry (*tabl*) has been replaced with the MCNP6 8th entry (*cutn*); the fission multiplicity setting on the PHYS:n card (*fism* for MCNPX and *fisnu* for MCNP5) has been moved to the FMULT card.

Data-card Form: `PHYS:n` *emax emcnf iunr* `J J J` *coilf cutn ngam* `J J` *i_int_model i_els_model*

emax	Upper limit for neutron energy and memory reduction control (DEFAULT: *emax* = 100 MeV). If *emax* < *cutn*, all model physics is eliminated, thus reducing memory requirements ((1), (2), (3)).
emcnf	Analog energy limit (DEFAULT: *emcnf* = 0 MeV). If E is the energy of the neutron and $E <$ *emcnf*, then perform analog capture. $E >$ *emcnf*, then perform implicit capture.
iunr	Controls unresolved resonance range probability table treatment when data tables are available. If *iunr* = 0, then probability table treatment is on (DEFAULT). *iunr* = 1, then probability table treatment is off.
`J`	Unused placeholder. Be sure to put the `J` in the keyword string ((4)).
`J`	Unused; fatal error if a value appears. Be sure to put the `J` in the keyword string. In the MCNPX code, the 5th parameter of the PHYS:n card is *tabl*. As of MCNP6, *tabl* has been moved to the 8th entry and is now *cutn*.
`J`	Unused; fatal error if a value appears. Be sure to put the `J` in the keyword string. In the MCNP5 code, the 5th entry on the PHYS:n card is *fisnu* and the MCNPX code's 6th entry on the PHYS:n card is *fism*. As of MCNP6, these have been moved to the FMULT card.

coilf = n.m — Light-ion and heavy-ion recoil and Neutron Capture Ion Algorithm (NCIA) control (see §5.7.2.2). In this format, n is an integer and m is a specified fractional value. If

n = 0, 1, 2, 4 and 0 < m ≤ 1,	then m is the number of light ions (protons, deuterons, tritons, ^{3}He, and alphas) per incident neutron to be created at each neutron elastic scatter event with light nuclei H, D, T, ^{3}He, and ^{4}He in the tabular physics regime. Heavy ions are also created if they are specified on the MODE card. Outside of the tabular physics regime, recoil production is performed by the model physics and is not affected by this setting. See the *recl* option on the PHYS:h card (§5.7.2.5) to enable this for all light-ion projectiles.
n = 3, 5,	then m = 0 and light-ion recoil is turned off.
n = 2, 3,	then NCIA is active only when the production of NCIA ions, shown in Table 5.8, is not modeled with the nuclear data tables.
n = 4, 5,	then NCIA is active and the nuclear data tables for production of NCIA ions are not used.

Using the above set of criteria, valid *coilf* entries include:

coilf = 0	then light-ion recoil is off; NCIA is off (DEFAULT).
0.001 < *coilf* < 1.001	then light–ion recoil makes *coilf* ions from elastic scatter.
1.001 < *coilf* < 2.001	then light-ion recoil makes *coilf* − 1 ions from elastic scatter; NCIA ions from neutron capture. Table data ion production will be used if possible.
coilf = 3	then light-ion recoil is off; NCIA ions from neutron capture. Table data ion production will be used if possible.
3.001 < *coilf* < 4.001	then light-ion recoil makes *coilf* − 3 ions from elastic scatter; NCIA ions from neutron capture. NCIA will be used even if table data are available.
coilf = 5	then light-ion recoil is off; NCIA ions from neutron capture. NCIA will be used even if table data are available.

cutn — Controls table-based physics cutoff and memory reduction. If

cutn ≥ 0,	use physics models for energies above *cutn* and data tables for those energies below *cutn*, if available (otherwise use models).
cutn = −1,	then mix and match. When tables are available, use them up to their upper limit for each nuclide, then use the physics models above that limit. See

		MX card for mixing and matching isotopic-specific data table and model physics usage (DEFAULT).
	cutn > *emax*,	save memory by eliminating all model physics arrays.
ngam	Controls secondary photon production (5). If	
	ngam = 0,	no photons are produced.
	ngam = 1,	photons are produced using the ACE nuclear data tables (DEFAULT).
	ngam = 2,	photons are produced using the Cascading Gamma Multiplicity (CGM) model [235].
J	Unused placeholder. Be sure to put the J in the keyword string.	
J	Unused placeholder. Be sure to put the J in the keyword string.	
i_int_model	Controls treatment of nuclear interactions. If	
	i_int_model = −1,	no interactions. Equivalent to setting the inelastic cross section to zero.
	i_int_model = 0,	process all interactions (DEFAULT).
	i_int_model = 1,	no secondaries, inelastic collisions treated as weight reduction.
	i_int_model = 2,	no secondaries, inelastic collisions treated as removal.
i_els_model	Controls treatment of nuclear elastic scattering (6). If	
	i_els_model = −1,	no elastic scattering (i.e., treat as pseudo collision)
	i_els_model = 0,	elastic scattering by the Prael/Liu/Striganov model [236] (DEFAULT).

Default: `PHYS:n 100 0 0 J J J 0 -1 J J J 0 0`

Use: Optional to modify default neutron table and model physics treatments. Additional neutron fission multiplicity options available on the FMULT card.

Limitations: Restarted calculations are not supported for delayed-neutron calculations that use model physics.

Details:

1. Memory allocation can be reduced significantly for MODE n p e problems that do not invoke the photonuclear (PHYS:p *ispn* = 0) option. By setting the neutron table/model cutoff energy, *cutn*, greater than the maximum neutron energy, *emax*, physics models are disabled, storage requirements for secondary particles are greatly reduced, and, consequently, the amount of memory that must be allocated to several MCNP6 arrays is decreased. The reduction of memory usage is helpful particularly for burnup problems. Use of this memory reduction option (i.e., setting *emax* < *cutn*) is confirmed by the following MCNP output file message: "Memory reduction option specified, physics models disabled.".
2. The parameter *emax* must be higher than the highest energy in the problem or the physics is wrong. For problems with energies above 100 MeV, *emax* should be chosen carefully; the default is appropriate for problems with energies below 100 MeV.

Table 5.8: NCIA Reactions

Isotope	Reaction(s)
^{3}He	^{3}He(n,h)t; n(^{3}He,d)d
^{6}Li	n(^{6}Li,t)α
^{10}B	n(^{10}B,α)^{7}Li

3. Neutron data above *emax* are expunged, as are neutron data below the lower energy cutoff, which is entered via the 2nd entry on the CUT:n card. If a neutron is born at an energy greater than *emax*, that neutron is rejected and the event (such as fission) is resampled until an energy below *emax* is obtained.

4. The *dnb* parameter for delayed neutron biasing, which previously held this position, has been removed. The ACT card can be used to set delayed neutron parameters.

5. Correlated neutron and gamma emission is provided by CGM (unlike use of the ACE libraries) [235], although execution times will increase. CGM/CGMF currently treats neutron interactions with targets of $Z > 9$, except elastic scatter which continues to be treated by ACE libraries.

6. Elastic scattering will be ignored if nuclear interactions are turned off.

5.7.2.1.1 Example: 1

The configuration shown in Listing 5.21 forces all neutrons transported to perform analog capture and a recoil ion will be created at each elastic scatter event.

Listing 5.21: example_phys_cut_nh.mcnp.inp.txt

```
phys:n 100  100  0  3J  1
```

5.7.2.2 Light Ion Recoil Physics and the Neutron Capture Ion Algorithm (NCIA) Discussion

Light ion recoil physics accounts for the ionization potential and uses the proper two-body kinematics (with neutron free-gas thermal treatment if appropriate) to bank recoil particles with the proper energy and angle. The input card MODE n h d t s a is required to produce and transport the proton (h), deuteron (d), triton (t), helion (s), and alpha (a) light ions. Heavy-ion recoils are produced if # is on the MODE card. The particle-specific low-energy cutoff can be set with the 2nd option, *e*, on the CUT:$\mathscr{P}$ card. For the $\mathscr{P}$ ions given on the MODE card, it is recommended to adjust the low-energy cutoff such that recoil ions produced are not killed by energy cutoff. See Table 4.3 for the default low-energy cutoffs for each particle type.

If activated by the 7th entry, *coilf*, on the PHYS:n card, the optional NCIA performs neutron capture in ^{3}He, ^{6}Li, and ^{10}B to produce protons, tritons, deuterons, and/or alphas according to Table 5.8.

The diagnostic indicating that NCIA has been used appears in PRINT Table 100.

Unlike most secondary particle production in the table physics region, NCIA particles are correlated. However, if $0.001 < \textit{coilf} < 1.001$, then one light ion is created by the data library and the other by NCIA; the correlation between the two particles is lost. If both particles are produced by the library, no correlation exists, either. Thus, $3.001 < \textit{coilf} < 5$ is recommended so that when NCIA data are available, table data are not used.

When performing heating calculations, the user must exercise caution. Because neutron energy deposition is physically mediated in most cases by the secondary particle emission, NCIA may be inconsistent for heating calculations. Neutron heating is done with kerma factors (heating numbers), whereas heating from the charged secondaries is done at collisions. For +F6 tallies and type 3 TMESH mesh tallies, the charged-ion heating is subtracted from the neutron heating and thus is counted only once. For F6:n and F6:h,d,t,a tallies, the heating is counted once for each particle type. If heating tallies are done in cells where charged ions are produced, energy may be double-counted in F6:$\mathscr{P}$ tallies. See §2.5.3 for further details.

5.7.2.3 Photons (PHYS:P)

⚠ Caution

Former MCNPX users need to be aware that the default behavior of the PHYS:p *nodop* option has changed. Photon Doppler broadening is now on by default (*nodop* = 0).

Data-card Form: `PHYS:p` *`emcpf ides nocoh ispn nodop`* `J` *`fism`*

emcpf	Upper energy limit for detailed photon physics treatment; photons with energy greater than *emcpf* will be tracked using the simple physics treatment (DEFAULT: *emcpf* = 100 MeV) (1).
ides	Controls generation of electrons by photons in `MODE p e` problems or, in photon-only problems, controls generation of bremsstrahlung photons with the thick-target bremsstrahlung model (2). If
ides = 0,	then generation is on (DEFAULT).
ides = 1,	then generation is off.
nocoh	Controls coherent (Thomson) scattering. If
nocoh = 0,	then coherent scattering is turned on (DEFAULT).
nocoh = 1,	then coherent scattering is turned off (3).
ispn	Controls photonuclear particle production (4). If
ispn = −1,	then photonuclear particle production is analog. One photon interaction per collision is sampled.
ispn = 0,	then photonuclear particle production is turned off (DEFAULT).
ispn = 1,	then photonuclear particle production is biased. The bias causes a photonuclear event at each photoatomic event.
nodop	Controls photon Doppler energy broadening (5). If
nodop = 0,	then Doppler energy broadening is turned on (DEFAULT).
nodop = 1,	then Doppler energy broadening is turned off.
`J`	Unused. Be sure to put the `J` in the keyword string. (6)
fism	Controls selection of photo-fission method (7). If
fism = 0,	sample photo-fission from ACE libraries (no photo-fission prompt gammas) (DEFAULT).

fism = 1,	sample photo-fission from the LLNL fission library [237]. Requires photonuclear physics (*ispn* ≠ 0).

Default: PHYS:p 100 0 0 0 0 J 0

Use: Optional.

Limitations: Restarted calculations are not supported for delayed-gamma calculations.

Details:

1. If *emax* on the PHYS:e card is less than *emcpf* on the PHYS:p card, MCNP6 will internally reset *empcf* to be equal to *emax*.
   - If *wc1* = 0 on the CUT:p card, analog capture is used in the energy region above *emcpf*. Otherwise capture is simulated by weight reduction with Russian roulette handling low-weight particles. Photons with energy less than *emcpf* will be treated with the more detailed physics that always includes analog capture. For a detailed discussion of the simple and detailed photon physics treatments, see §2.4.4.1 and §2.4.4.2, respectively.
   - The simple physics treatment, intended primarily for higher energy photons, considers the following physical processes: photoelectric effect without fluorescence, Compton scattering, and pair production. The highly forward peaked coherent Thomson scattering is ignored. In the detailed physics treatment, photoelectric absorption can result in fluorescent emission, the Thomson and Klein-Nishina differential cross sections are modified by appropriate form factors [238] and Compton profiles taking electron binding effects into account, and coherent scattering is included.
2. To turn off the production of secondary electrons generated by photons, the switch *ides* can be set, either on the PHYS:p or on the PHYS:e card. If either of these cards sets *ides* = 1, photons will not produce electrons, even if *ides* = 0 is set on the other. In a photon-only problem, turning off secondary electrons causes the thick-target bremsstrahlung model to be bypassed. This option should be exercised only with great care because it alters the physics of the electron-photon cascade and will give erroneously low photon results when bremsstrahlung and electron transport are significant.
3. When *nocoh* = 1, the cross section for coherent scattering will be set to zero. This approximation can be useful in problems with bad point detector variances.
4. Photonuclear physics models enable (γ, n) and other photonuclear reactions when photonuclear data tables are unavailable. When some photonuclear data tables are available, MCNP6 will mix and match, using tables when available and physics models when no tables are available. Consider using an MX:p card to override this default behavior.
5. When photon Doppler broadening is turned on (*nodop* = 0), there is no effect unless photon Doppler broadening momentum profile data are available in the photon library. These data are available in the **MCPLIB03** and later photon libraries.
6. The *dgb* parameter for delayed photon biasing, which previously held this position, has been removed. The ACT card can be used to set delayed gamma parameters.
7. When *fism* = 1, photo-fission secondaries are sampled only when a photo-fission event occurs (unlike *fism* = 0). This enables coincidence counting of photo-fission secondaries. The LLNL fission library for photo-fission is the only way to produce prompt photo-fission gammas; these gammas are correlated with the photo-fission neutrons with appropriate multiplicities. When *fism* = 1 on the PHYS:p card, photonuclear physics must be turned on (*ispn* ≠ 0) and the LLNL fission library should be used also for neutrons (*method* = 5 on the FMULT card).

5.7.2.4 Electrons (PHYS:E)

Electron-specific physics settings are controlled with this card. However, because of the close relationship between photons and electrons during transport, also consider using the `PHYS`:`p` card in case adequate control is unavailable on the `PHYS`:`e` card alone.

Data-card Form: PHYS:e *emax ides iphot ibad istrg bnum xnum rnok enum numb i_mcs_model mode_electron_elastic* J *efac electron_method_boundary ckvnum*

emax Upper limit for electron energy (DEFAULT: *emax* on PHYS:n card or 100 MeV if no PHYS:n card) (1).

ides Controls production of electrons by photons in MODE e problems or, in photon-only problems, controls generation of bremsstrahlung photons with the thick-target bremsstrahlung model. If

- *ides* = 0, then electron production by photons is turned on (DEFAULT).
- *ides* = 1, then electron production by photons is turned off.

iphot Controls production of photons by electrons. If

- *iphot* = 0, then photon production by electrons is turned on (DEFAULT).
- *iphot* = 1, then photon production by electrons is turn off.

ibad Controls bremsstrahlung angular distribution method. If

- *ibad* = 0, perform full bremsstrahlung tabular angular distribution (DEFAULT).
- *ibad* = 1, perform simple bremsstrahlung angular distribution approximation (2).

istrg Controls electron continuous-energy slowing down ("straggling") treatment. If

- *istrg* = 0, use sampled value straggling method to compute electron energy loss at each collision (DEFAULT).
- *istrg* = 1, use expected-value straggling method to compute electron energy loss at each collision.

bnum Controls production of bremsstrahlung photons created along electron sub steps. If

- *bnum* = 0, bremsstrahlung photons will not be produced.
- *bnum* > 0, produce *bnum* times the analog number of bremsstrahlung photons. Radiative energy loss uses the bremsstrahlung energy of the first sampled photon (DEFAULT: *bnum* = 1).

The specification *bnum* < 0 is only applicable is using the **EL03** electron-transport cross section library. Produce |*bnum*| times the number of analog photons. Radiative energy loss uses the average energy of all the bremsstrahlung photons sampled.

xnum Controls sampling of electron-induced x-rays produced along electron sub steps. If

xnum = 0,	x-ray photons will not be produced by electrons.
xnum > 0,	produce *xnum* times the analog number of electron-induced x-rays (DEFAULT: *xnum* = 1).

rnok Controls creation of knock-on electrons produced in electron interactions. If

rnok = 0,	knock-on electrons will not be produced.
rnok > 0,	produce *rnok* times the analog number of knock-on electrons (DEFAULT: *rnok* = 1).

e_bias_num Controls generation of photon-induced secondary electrons (3). If

e_bias_num = 0,	photon-induced secondary electrons will not be produced.
e_bias_num > 0,	produce *e_bias_num* times the analog number of photon-induced secondary electrons (DEFAULT: *e_bias_num* = 1).

numb Controls bremsstrahlung production on each electron sub step (4). If

numb = 0,	analog bremsstrahlung production (DEFAULT).
numb > 0,	produce bremsstrahlung on each sub step.

i_mcs_model Controls the choice of Coulomb scattering model. If

i_mcs_model = −1,	turn off angular deflection.
i_mcs_model = 0,	select the standard Goudsmit-Saunderson angular deflection method (DEFAULT).

mode_electron_elastic
Controls choice of electron elastic cross section. Single-event transport only. If

mode_electron_elastic = 0,	large-angle elastic scattering is used (cosine > 10^{-6}) (DEFAULT).
mode_electron_elastic = 2,	total elastic cross section (large-angle + in-peak) is used.

J Unused placeholder. Be sure to put the J in the keyword string.

efac Controls stopping power energy spacing (5). Restriction: $0.8 \leq$ *efac* ≤ 0.99 (DEFAULT: *efac* = 0.917).

electron_method_boundary
Controls the start of single-event transport (6, 7). *electron_method_boundary* is the energy (in MeV) above which MCNP6 transports electrons by the condensed-history algorithms and below which the single-event method is used (DEFAULT: *electron_method_boundary* = 10^{-3}).

ckvnum Scales Cerenkov photon emission from a particular particle by a fractional amount with the photons emitted at higher weight (8). Allowed values are $0 \leq$ *ckvnum* < 1); values of 10^{-3}–10^{-2} are recommended. *ckvnum* = 0 turns off Cerenkov emission (DEFAULT: 0).

Default: `PHYS:e 100 0 0 0 0 1 1 1 1 0 0 0 J 0.917 0.001 0`

Use: Optional.

Caution

The use of the switches (or of zero values for the biasing parameters) to turn off various processes goes beyond biasing and actually changes the physics of the simulation. Therefore such actions should be taken with extreme care. These options are provided primarily for purposes of debugging, code development, and special-purpose studies of the cascade transport process.

Details:

1. The parameter *emax* should be set to the highest electron energy encountered in your problem.
2. Point detectors and DXTRAN spheres use the simple bremsstrahlung angular distribution approximation. Always use (*ibad* = 1).
3. The specification *enum* = 0 differs from *ides* = 1. If *enum* = 0, pair production is totally turned off. If *ides* = 1, the pair production-produced annihilation photons are still produced.
4. Only a real event, i.e., one that has been sampled to have a bremsstrahlung interaction, causes energy loss. The weights of the bremsstrahlung photons are multiplied by the probability of interaction in a substep. If two or more photons are produced in a real event, the weight of the second or more photons is the unadjusted value because there is no Poisson sampling, except for real events.
5. When *efac* is specified, the energy spacing for multiple-scattering tables (stopping power, range, etc.) is determined by

$$E_{n+1} = E_n \times F \tag{5.10}$$

where E_1 is the highest energy and

$$f = \left(\frac{1}{2}\right)^{1/D} \tag{5.11}$$

where

$$D = \text{real}\left(\frac{\text{nint}\left(\ln\left(\frac{1}{2}\right)\right)}{\ln(\textit{efac})}\right). \tag{5.12}$$

This means that on average, the energy of the particle will decrease by a factor of two in D energy steps and that a larger value of *efac* results in more points in the multiple-scattering tables. The default value, *efac* = 0.917, leads to the traditional choice of eight energy steps for a factor-of-two energy loss.

6. To invoke the single-event electron-transport method, the problem must have access to photon data, even if the user is not interested in the photon transport. Therefore, the `MODE` card must included the specification for both photons (`p`) and electrons (`e`). Access to the **EPRDATA14** library data is required to transport electrons below 1 keV. This library, which is denoted by the cross-section identifier ".14p," is not the default in the **xsdir_mcnp6.3** cross-section directory file provided with the MCNP code, version 6.3; therefore the **EPRDATA14** library may need to be requested on the material cards explicitly. This low-energy (< 1 keV) data are only for zero-temperature atomic targets, so temperature, condensed state, and molecular effects are not yet treated for electrons in this regime.
7. The energy boundary that defines the switch to single-event transport should never be lower than 1 keV, because condensed-history methods rapidly collapse below this traditional lower limit.
8. Cerenkov photon production requires that a refractive index is specified on the `M` card.

5.7.2.4.0.1 Example 1

The configuration shown in Listing 5.22 causes the energy-boundary switch to the single-event electron-transport method to occur at 10 keV.

Listing 5.22: example_phys_e.mcnp.inp.txt

```
phys:e 100. 13j 0.01
```

5.7.2.5 Protons (PHYS:H)

Proton-specific physics settings are controlled with this card.

Data-card Form: `PHYS:h emax ean tabl J istrg J recl J J J i_mcs_model i_int_model i_els_model efac J ckvnum drp`

emax	Upper proton energy limit (DEFAULT: *emax* on PHYS:n card or 100 MeV if no PHYS:n card) (1).
ean	Analog energy limit (DEFAULT: *ean* = 0 MeV). If E is the energy of the proton and
	$E <$ *ean*, then perform analog capture.
	$E >$ *ean*, then perform implicit capture.
tabl	Table-based physics cutoff. If
	tabl $= -1$, then mix and match. When tables are available, use them up to their upper limit for each nuclide, then use the physics models above this limit (DEFAULT).
	tabl ≥ 0, use physics models for energies $E >$ *tabl*) and data tables otherwise, if available (otherwise use models).
J	Unused placeholder. Be sure to put the J in the keyword string.
istrg	Controls charged-particle straggling. If
	istrg $= 0$, use Vavilov model for charged-particle straggling (DEFAULT).
	istrg $= 1$, use continuous slowing-down approximation for charged-particle straggling.
J	Unused placeholder. Be sure to put the J in the keyword string.
recl	Recoil production control for light-ion tabular physics (2). If
	recl $= 0$, then no recoil ions are produced in tabular physics (DEFAULT).
	$0 <$ *recl* ≤ 1, *recl* is the number of recoil ions to be created at each light-ion elastic scatter event in tabular physics regimes. Recoil production can include protons, deuterons, tritons, ^{3}He, alphas, and heavy ions if they are present on the MODE card. Outside

	of the tabular physics regime, recoil production is performed by the model physics and is not affected by this setting. Note that this option enables the feature for all light-ion projectiles, regardless of whether protons are on the MODE card or not. See *coilf* on PHYS:n for the option to enable this for neutron projectiles.
J	Unused placeholders. Be sure to put the J in the keyword string.
J	Unused placeholders. Be sure to put the J in the keyword string.
J	Unused placeholders. Be sure to put the J in the keyword string.
i_mcs_model	Controls the choice of Coulomb scattering model. If
	i_mcs_model $= -1$, turn off angular deflection.
	i_mcs_model $= 0$, use FermiLab angular deflection model with Vavilov straggling (DEFAULT).
	i_mcs_model $= 1$, use Gaussian angular deflection model with Vavilov straggling.
	i_mcs_model $= 2$, use FermiLab coupled energy/angle MCS model.
i_int_model	Controls treatment of nuclear interactions. If
	i_int_model $= -1$, no interactions. This is equivalent to setting the inelastic cross section to zero.
	i_int_model $= 0$, process all interactions (DEFAULT).
	i_int_model $= 1$, no secondaries, inelastic collisions treated as weight reduction.
	i_int_model $= 2$, no secondaries, inelastic collisions treated as removal.
i_els_model	Controls treatment of nuclear elastic scattering (3). If
	i_els_model $= -1$, no elastic scattering (i.e., treat as pseudo collision).
	i_els_model $= 0$, elastic scattering by Prael/Liu/Striganov model[236] (DEFAULT).
efac	Controls stopping power energy spacing (4). Restriction: $0.8 \leq$ *efac* ≤ 0.99. (DEFAULT: *efac* $= 0.917$)
J	Unused placeholder. Be sure to put the J in the keyword string.
ckvnum	Scales Cerenkov photon emission from a particular particle by a fractional amount with the photons emitted at higher weight (6). Allowed values are $0 \leq$ *ckvnum* < 1); values of 10^{-3}–10^{-2} are recommended. *ckvnum* $= 0$ turns off Cerenkov emission (DEFAULT: 0).
drp	Lower energy delta-ray cutoff (5) If
	drp $= -1$, turn on delta-ray production and use the default energy cutoff (0.020 MeV).
	drp $= 0$, turn off delta-ray production (DEFAULT).
	drp > 0, turn on delta-ray production and set the cutoff to *drp* MeV, valid for charged particles only.

Default: PHYS:h 100 0 -1 J 0 J 0 J J J 0 0 0 0.917 0 0

Use: Optional

Details:

1. If *emax* on the PHYS:e card is less than *emax* on the PHYS:h card, the MCNP code will internally set the PHYS:h *emax* to the PHYS:e *emax*. The parameter *emax* must be higher than the highest energy in the problem or the physics is wrong. For problems with energies above 100 MeV, *emax* should be chosen carefully; the default is appropriate for problems with energies below 100 MeV.
2. Light ion recoil physics accounts for the ionization potential and uses the proper two-body kinematics (with neutron free-gas thermal treatment if appropriate) to bank recoil particles with the proper energy and angle. The particle-specific low-energy cutoff can be set with the 2nd option, *e*, on the CUT:$\mathscr{P}$ card. For the $\mathscr{P}$ ions given on the MODE card, it is recommended to adjust the low-energy cutoff such that recoil ions produced are not killed by energy cutoff. See Table 4.3 for the default low-energy cutoffs for each particle type. Note that protons colliding with hydrogen to produce more protons can produce an overwhelming number of protons. Therefore, caution is required, and *recl* < 1 may be needed. This capability is the same for incident neutrons as controlled by the *coilf* keyword on the PHYS:h card.
3. Elastic scattering will be ignored if nuclear interactions are turned off.
4. When *efac* is specified, the energy spacing for multiple-scattering tables (stopping power, range, etc.) is determined by
$$E_{n+1} = E_n \times F \tag{5.13}$$
where E_1 is the highest energy and
$$f = \left(\frac{1}{2}\right)^{1/D} \tag{5.14}$$
where
$$D = \text{real}\left(\frac{\text{nint}\left(\ln\left(\frac{1}{2}\right)\right)}{\ln(\textit{efac})}\right). \tag{5.15}$$
This means that on average, the energy of the particle will decrease by a factor of two in D energy steps and that a larger value of *efac* results in more points in the multiple-scattering tables. The default value, *efac* $= 0.917$, leads to the traditional choice of eight energy steps for a factor-of-two energy loss.
5. Delta-ray production is according to the formulation by B. Rossi [239]. The E^{-2} differential spectrum is truncated by the *drp* parameter, which should be greater than 1 keV, with a default value of 20 keV and a maximum of 1.022 MeV. To increase execution speed, this parameter should be set as large as possible, while retaining important effects to tallies of interest.
6. Cerenkov photon production requires that a refractive index is specified on the M card.

5.7.2.5.1 Example 1

The configuration shown in Listing 5.23 forces all protons transported to perform analog capture and a recoil ion to be created at each elastic scatter event.

Listing 5.23: example_phys_cut_nh.mcnp.inp.txt

```
phys:h 100  100 -1  3J  1
```

5.7.2.6 Other Particles (PHYS:$\mathscr{P}$)

Physics settings for all other particle types are controlled with this card.

Data-card Form: PHYS:$\mathscr{P}$ *emax* J J J *istrg* J *xmunum xmugam* J J *i_mcs_model i_int_model i_els_model efac* J *ckvnum drp*

$\mathscr{P}$	Particles designators other than n, p, e, and h (1).
emax	Upper energy limit (DEFAULT: *emax* on PHYS:n card or 100 MeV if no PHYS:n card) (2).
J	Unused placeholders. Be sure to put the J in the keyword string.
J	Unused placeholders. Be sure to put the J in the keyword string.
J	Unused placeholders. Be sure to put the J in the keyword string.
istrg	Controls charged-particle straggling. If
	istrg = 0, use Vavilov model with an energy correction addressing stopping powers (DEFAULT).
	istrg = 1, use continuous slowing-down ionization model.
J	Unused placeholder. Be sure to put the J in the keyword string.
xmunum	Controls the selection of muonic x-ray data. Restriction: Only valid for muons (PHYS:|). This PHYS card 7th entry has other meanings for $\mathscr{P}$ = n, p, e, and h and is ignored for other particles. If
	xmunum = −1, use only x-ray literature data.
	xmunum = 1, emit all x-rays including data from literature and from the MUON/RURP code package [240] (DEFAULT).
xmugam	Probability for emitting k-shell photon (DEFAULT: *xmugam* = 0.65). Restriction: Only valid for muons (PHYS:| card) . This PHYS card 8th entry has other meanings for $\mathscr{P}$ = n, p, e, and h and is ignored for other particles.
J	Unused placeholders. Be sure to put the J in the keyword string.
J	Unused placeholders. Be sure to put the J in the keyword string.
i_mcs_model	Controls the choice of Coulomb scattering model. Restriction: Valid for charged particles only. If
	i_mcs_model = −1, turn off angular deflection.
	i_mcs_model = 0, use FermiLab angular deflection model with Vavilov straggling (DEFAULT).
	i_mcs_model = 1, use Gaussian angular deflection model with Vavilov straggling.
	i_mcs_model = 2, use FermiLab coupled energy/angle MCS model.
i_int_model	Controls treatment of nuclear interactions. If
	i_int_model = −1, no interactions. This is equivalent to setting the inelastic cross section to zero.

	i_int_model = 0,	process all interactions (DEFAULT).
	i_int_model = 1,	no secondaries, inelastic collisions treated as weight reduction.
	i_int_model = 2,	no secondaries, inelastic collisions treated as removal.
i_els_model	Controls treatment of nuclear elastic scattering (3) If	
	i_els_model = −1,	no elastic scattering (i.e., treat as pseudo collision).
	i_els_model = 0,	elastic scattering by Prael/Liu/Striganov model [236] (DEFAULT).
efac	Controls stopping power energy spacing (4). Restriction: $0.8 \leq$ *efac* ≤ 0.99; valid for charged particles only (DEFAULT: *efac* = 0.917).	
J	Unused placeholder. Be sure to put the J in the keyword string.	
ckvnum	Scales Cerenkov photon emission from a particular particle by a fractional amount with the photons emitted at higher weight (6). Allowed values are $0 \leq$ *ckvnum* < 1); values of 10^{-3}–10^{-2} are recommended. *ckvnum* = 0 turns off Cerenkov emission (DEFAULT: 0).	
drp	Lower energy delta-ray cutoff (5) If	
	drp = −1,	turn on delta-ray production and use the default energy cutoff (0.020 MeV).
	drp = 0,	turn off delta-ray production (DEFAULT).
	drp > 0,	turn on delta-ray production and set the cutoff to *drp* MeV. Valid for charged particles only.

Default: PHYS:$\mathscr{P}$ 100 3J 0 5J 0 0 0 0.917 J 0 0

Default: PHYS:| 100 3J 0 J 1 0.65 2J 0 0 0 0.917 J 0 0

Use: Optional.

Details:

(1) If *emax* on the PHYS:e card is less than *emax* on the PHYS:$\mathscr{P}$ card, MCNP6 will internally set the PHYS:$\mathscr{P}$ *emax* to the PHYS:e *emax*. Although heavy ions (#) may be designated, there is no heavy ion recoil for proton elastic scattering events.

(2) The parameter *emax* must be higher than the highest energy in the problem or the physics is wrong. For problems with energies above 100 MeV, *emax* should be chosen carefully; the default is appropriate for problems with energies below 100 MeV.

(3) Elastic scattering will be ignored if nuclear interactions are turned off.

(4) When *efac* is specified, the energy spacing for multiple-scattering tables (stopping power, range, etc.) is determined by

$$E_{n+1} = E_n \times F \tag{5.16}$$

where E_1 is the highest energy and

$$f = \left(\frac{1}{2}\right)^{1/D} \tag{5.17}$$

where

$$D = \text{real}\left(\frac{\text{nint}\left(\ln\left(\frac{1}{2}\right)\right)}{\ln(\textit{efac})}\right). \tag{5.18}$$

This means that on average, the energy of the particle will decrease by a factor of two in D energy steps and that a larger value of *efac* results in more points in the multiple-scattering tables. The default value, *efac* = 0.917, leads to the traditional choice of eight energy steps for a factor-of-two energy loss.

(5) Delta-ray production is according to the formulation by B. Rossi [239]. The E^{-2} differential spectrum is truncated by the *drp* parameter, which should be greater than 1 keV, with a default value of 20 keV and a maximum of 1.022 MeV. To increase execution speed, this parameter should be set as large as possible, while retaining important effects to tallies of interest.

(6) Cerenkov photon production requires that a refractive index is specified on the M card.

5.7.3 ACT: Activation Control Card

Available delayed particles are: neutrons, gammas, betas, alphas, and positrons. Delayed-neutron emission can be calculated using library (dn = library) or model (dn = model) treatments. The library treatment uses ACE data and produces delayed neutrons only for fission. The model treatment uses data from the **delay_library_v5.dat** library and produces delayed neutrons for fission and, if requested, activation. Delayed-gamma emission is calculated by line emission data (dg = lines), from ENDF/B-VII.1 data contained in **cindergl.dat** and augmented by model data contained in **delay_library_v5.dat**, or only model data (dg = mg). Delayed betas, alphas, and positrons are sampled solely from **delay_library_v5.dat** data.

The **delay_library_v5.dat** delayed-particle library provides unique delayed neutron, gamma, beta, alpha, and positron spectra to be sampled for each radionuclide. Delayed neutron spectra are sampled from 750 bins ranging from 0–7.5 MeV for 298 nuclides. Delayed gamma spectra are sampled from 500 bins ranging from 0–10 MeV for 1865 nuclides ((3), (4)). Delayed beta spectra are sampled from 100 bins ranging from 0–10 MeV for 1891 nuclides. Delayed positron spectra are sampled from 100 bins ranging from 0–10 MeV for 531 nuclides. Delayed alpha spectra are sampled from 100 bins ranging from 0–10 MeV for 248 nuclides. A warning is issued when no delayed particle data is available and a nuclide with a non-zero delayed-particle probability is sampled.

Delayed-gamma emission is limited to fixed source (SDEF) problems.

Data-card Form: ACT KEYWORD = value(s)

fission	Type of delayed particle(s) to be produced from residuals created by fission. If
fission = none,	create no delayed particles from fission events.
fission = n, p, e, f, a,	create delayed neutrons (n), delayed gammas (p), delayed beta particles (e), delayed positron particles (f), and/or delayed alphas (a) from fission events. Only those listed will be created (DEFAULT: fission = n).

`fission` = `all`,	create all delayed particles from fission events.

`nonfiss` Type of delayed particle(s) to be produced by simple multi-particle reaction activation (i.e., non-fission) events. If

`nonfiss` = `none`,	create no delayed particles from non-fission events (DEFAULT).
`nonfiss` = `n`, `p`, `e`, `f`, `a`,	create delayed neutrons (`n`), delayed gammas (`p`), delayed beta particles (`e`), delayed positron particles (`f`), and/or delayed alphas (`a`) from non-fission events. Only those listed will be created.
`nonfiss` = `all`,	create all delayed particles from non-fission events.

`dn` Delayed neutron data source. If

`dn` = `model`,	production of delayed neutrons uses models only (1).
`dn` = `library`,	production of delayed neutrons uses libraries only (DEFAULT).
`dn` = `both`,	production of delayed neutrons uses models when libraries are missing.
`dn` = `prompt`,	treat prompt and delayed neutrons as prompt.

`dg` Delayed gamma data source (2). If

`dg` = `lines`,	sample delayed gammas using models based on line-emission data contained in **`cindergl.dat`**, augmented by data in the latest **`delay_library_v[n].dat`**.
`dg` = `mg`,	sample delayed gammas using models based on 25-group emission data (3).
`dg` = `none`,	do not create delayed gammas (DEFAULT).

`thresh` = *f* The fraction of highest-amplitude discrete delayed-gamma lines, *f*, that will be retained (4) (DEFAULT: `thresh` = 0.95).

`dnbias` = *n* Produce up to *n* delayed neutrons per interaction (DEFAULT: analog calculation). Restriction: $1 \leq n \leq 10$; `dnbias` is disallowed in `KCODE` calculations.

`nap` = *m* The integer number *m* of activation products for which cumulative distribution functions will be calculated once and stored for reuse. The *m* most frequently accessed distribution functions are dynamically updated during execution. The `nap` keyword is applicable to `ACT` `nonfiss` problems using line data only (DEFAULT: `nap` = 10).

`dneb` = *w1*, *e1*, *w2*, *e2*, . . . , *wK*, *eK*
Delayed neutron energy biasing parameters where

wk	is the weight for the *k*th energy bin, and
ek	is the upper energy for the *k*th energy bin (initial lower bin bound of 0 assumed).

Energies within a bin are sampled evenly; probability of sampling from within a bin is based upon *wk* (5).

`dgeb` = *w1*, *e1*, *w2*, *e2*, . . . , *wK*, *eK*
Delayed photon energy biasing where

wk	is the weight for the *k*th energy bin, and
ek	is the upper energy for the *k*th energy bin (initial lower bin bound of 0 assumed).

Energies within a bin are sampled evenly; probability of sampling from within a bin is based upon *wk* (5).

`pecut` = *e*
Delayed-gamma energy cutoff (MeV). Gamma lines below `pecut` will be expunged (DEFAULT: `pecut` = 0).

`hlcut` = *t*
Spontaneous-decay half-life threshold (seconds). Decay chains are truncated when a daughter half-life exceeds `hlcut`. Delayed-particle production from this and subsequent daughters is omitted (DEFAULT: `hlcut` = 0, i.e., no truncation of decay chains).

`sample`
Flag for correlated or uncorrelated. If

`sample` = `correlate`,	treat as correlated.
`sample` = `nonfiss_cor`,	treat as uncorrelated.

Details:

1. Delayed-particle emission is currently integrated over 10^{10} seconds with 99 time steps; however, the user should consider increasing the stability half-life parameter (10th entry on the DBCN card) when emission from long-lived radionuclides is important. Increasing this parameter results in an increase in the time integration to 10^{19} seconds with 234 time steps.
2. The `fission` keyword enables delayed-particle emission from the decay of radioactive fission products created by neutron- or photon-induced fission treated by ACE libraries or any fission event treated by model physics. The `nonfiss` keyword enables delayed-particle emission from the decay of radioactive residuals created by neutron and photon interactions treated by ACE libraries or any nuclear interaction treated by model physics. Most neutron ACE libraries contain the necessary secondary-production cross sections needed to determine radioactive residuals, however few ACE photonuclear libraries currently contain this data. Thus, users should consider the use of photonuclear model physics (see the MX card) or obtain updated ACE photonuclear libraries in which secondary reactions are not lumped into `MT` = 5. Proton ACE library interactions also suffer from this issue.
3. Bin-wise emission (`dg` = `mg`) is preferred when individual line-amplitude detail is not important. This option is significantly faster and the emission spectra will converge more quickly than line emission mode (i.e., `dg` = `lines`). Line emission augmented with bin-wise emission (`dg` = `lines`) is useful for studies that require high fidelity, detailed-amplitude emission spectra. This option is significantly slower and can require the execution of large numbers of histories to suitably converge low probability delayed-gamma emission lines.
4. Set `thresh` = 1.0 to retain all lines in the **`cindergl.dat`** file.

Caution

For some problems (e.g., fission), the calculation with `thresh` = 1.0 will either run slowly or exceed memory limits and fail.

(5) A weight of zero should not be set for any of the energy bins with `dgeb` or `dneb`. Instead, a small value (e.g., 0.001) is recommended [241].

5.7.4 Physics Cutoffs

5.7.4.1 CUT: Time, Energy, and Weight Cutoffs

Data-card Form: `CUT:`$\mathscr{P}$ *t e wc1 wc2 swtm*

$\mathscr{P}$	Particle designator.
t	Time cutoff in shakes, 1 shake = 10^{-8} s ((1), (2)).
e	Lower energy cutoff in MeV ((1), (3)).
wc1, *wc2*	Weight cutoffs. If weight goes below *wc2* roulette is played to restore weight to *wc1*. Negative entries scale weight cutoff relative to the minimum source weight of a particle ((4)). Setting *wc1* = *wc2* = 0 invokes analog capture ((5), (6)).
swtm	Minimum source weight ((7), (8)).

Neutron default: *t* is very large, *e* = 0.0 MeV, *wc1* = −0.50, *wc2* = −0.25, *swtm* is the minimum source weight if the general source is used.

Photon default: *t* is the neutron cutoff time, *e* = 0.001 MeV, *wc1* = −0.50, *wc2* = −0.25, *swtm* is the minimum source weight if the general source is used; if there are pulse-height tallies, *wc1* = *wc2* = 0, unless forced collisions are also used; if pulse-height tallies exist with forced collisions, the default values are *wc1* = −0.50 and *wc2* = −0.25.

Electron default: *t* is the neutron cutoff time, *e* = 0.001 MeV, *wc1* = 0, *wc2* = 0, *swtm* is the minimum source weight if the general source is used; if there are pulse-height tallies, *wc1* = *wc2* = 0, unless forced collisions are also used; if pulse-height tallies exist with forced collisions, the default values are *wc1* = −0.50 and *wc2* = −0.25.

With the exception of photon energy and electron/positron energy (see §5.7.4.2 and §5.7.4.3), the default energy cutoff values for all particles appear in Table 4.3. All other particle time and weight default cutoffs are the same as for electrons.

Use: Optional, as needed. Analog capture is highly recommended when using weight windows and for many other applications.

Details:

(1) If a particle's time exceeds the *t* specified for that particle, it is killed. Although MCNP6 is time dependent, particle decay is not considered. Any particle with energy lower than the *e* specified for that particle is killed.

(2) The default (and maximum) emission time for delayed particle emission is 10^{10} s. By using the `CUT` card(s), the maximum emission time becomes (1) the particle's time cutoff if time cutoff is specified or (2) the minimum of time cutoff if multiple time cutoffs are provided.

③ For adjoint (MGOPT) problems, *e* is the upper energy cutoff, not the lower energy cutoff.

④ For non-analog capture, if a particle's weight w falls below *wc2* times the ratio R of the source cell importance to the current cell importance, then with probability

$$\frac{w}{wc1 \times R}$$

the particle survives and is assigned $w = wc1 \times R$. If negative values are entered for the weight cutoffs, the values $|wc1|w_s$ and $|wc2|w_s$ will be used for *wc1* and *wc2*, respectively, where w_s is the minimum starting weight assigned to a source particle from an MCNP6 general source. Scaling the weight cutoff to the minimum source weight of a particle prevents source particles from being immediately killed by falling below the cutoff. These negative entries are recommended over positive entries for most problems. If only *wc1* is specified, then $wc2 = 0.5wc1$.

⑤ If *wc1* is set to zero, capture is treated explicitly by analog rather than implicitly by reducing the particles' weight according to the capture probability. If *ean* or *emcnf* = *emax* on the PHYS:$\mathscr{P}$ card (i.e., applies to neutrons or protons), analog capture is used regardless of the value of *wc1* except for particles leaving a DXTRAN sphere.

⑥ To generate delayed particles from non-fissioning isotopes, *wc1* must be set to zero on both the photon and neutron CUT:$\mathscr{P}$ cards so that analog capture is invoked.

⑦ When the source is biased in any way, there will be a fluctuation in starting source weights. By playing the weight cutoff game relative to the minimum source weight, the weight cutoff in each cell is the same regardless of starting source weight. Note that if the source weight can go to zero, the minimum source weight is set to 10^{-10} times the value of the *wgt* parameter on the SDEF card.

⑧ The parameter *swtm* can be used to make the weight cutoffs relative to the minimum starting weight of a source particle for a user source, as is done automatically for the general source. The entry will, in general, be the minimum starting weight of all source particles, including the effects of energy and direction biasing. The entry is also effective for the general source. Then *swtm* is multiplied by the *wgt* entry on the SDEF card, but is unaffected by any directional or energy biasing. This entry is ignored for a KCODE calculation.

5.7.4.2 Additional Photon Cutoff Notes

The CUT:p weight cutoffs are analogous to those on the CUT:n card except that they are used only for energies above the *emcpf* entry on the PHYS:p card. If *wc1* = 0, analog capture is specified for photons of energy greater than *emcpf*. For energies below *emcpf*, analog capture is the only choice with one exception: photons leaving a DXTRAN sphere. Their weight is always checked against the CUT:p weight cutoff upon exiting. If only *wc1* is specified, then $wc2 = 0.5wc1$.

In a coupled neutron/photon problem, the photon weight cutoffs are the same as the neutron weight cutoffs unless overridden by a CUT:p card.

In a coupled neutron/photon problem, photons are generated before the neutron weight cutoff game is played.

Although the default photon energy cutoff is 1 keV, a user may explicitly specify a lower cutoff down to 1 eV. The required photoatomic cross sections from ENDF/B-VI, release 8, are included in the data library **EPRDATA14** (Electron-Photon-Relaxation DATA). The tables in this library are presented in a newly developed ACE format specifically designed for use with MCNP6. They cannot be used correctly with the earlier codes MCNP5 or MCNPX. The proper tables can be requested on material cards using the library identifier ".14p". Users are cautioned that at very low energies, molecular and other effects become important for scattering and absorption, and these more complex effects are not yet included in the photon transport methods. Also,

note that although electron transport has been extended down to 10 eV, electron energies have not been extended as low as photon energies.

MCNP6 allows only analog capture below 0.001 MeV. Because the photoelectric cross section is virtually 100% of the total cross section below that energy for all isotopes, tracks will be quickly captured and terminated.

5.7.4.3 Additional Electron/Positron Cutoff Notes

Positron physics in MCNP6 is identical to electron physics, except for tracking directions in magnetic fields and consideration of positron annihilation. Whereas electrons below the energy cutoff are terminated, positrons below the energy cutoff produce annihilation photons. The positrons have a positive charge and may be tallied using the FT card *elc* option [§5.9.18.8]. Electron transport, which has a default cutoff of 1 keV, may be explicitly specified down to 10 eV.

To transport electrons at energies below 1 keV, the **EPRDATA14** library is required. As with low-energy photon transport, the proper tables can be requested with the library identifier ".14p". Also as with photons, the same cautions regarding temperature, molecular, solid-state, and other low-energy phenomena apply to low-energy electrons.

For very low-energy electrons, a physics-based practical difficulty can arise: the lack of energy-loss-inducing processes. Although bremsstrahlung is still present, it is completely dominated by electron elastic scattering, which results in no energy loss. Electro-ionization, an important energy-loss channel, vanishes below the binding energy of the least-bound shell given in the data. Whether that event occurs above or below 10 eV is element-dependent. Excitation, another energy-loss process, also can vanish at some energy above 10 eV, depending on the element. Consequently, there can be a small energy range just above 10 eV in which the electron can no longer lose energy and only experiences a large number of elastic scatterings. Coupled with the very short step sizes that characterize electron transport at low energies, the effect is that the transport suddenly grinds nearly to a halt because an electron has become trapped, taking a huge number of small steps with little or no opportunity to lose energy. Such an electron is very close to the energy cutoff, but cannot get there because it is spending all its time in elastic scatter. Preliminary practical experience indicates that setting the electron cutoff no lower than about 12 eV may be sufficient to avoid this occasional effect. Again note that the low-energy cross-section data are only for cold atomic targets, and that potential future treatments of molecular and other low-energy physics will significantly alter this discussion.

5.7.4.4 Example 1

The configuration shown in Listing 5.24 causes the neutron energy cutoff to be increased from zero to 0.99999 eV, the proton energy cutoff to be lowered to zero, and the weight cutoff roulette parameters set to zero to force analog proton capture. Note that increasing the neutron energy cutoff should only be done in circumstances where fissile material is not involved in the simulation. In general, it is not recommended.

Listing 5.24: example_phys_cut_nh.mcnp.inp.txt

```
cut:n  J  0.99999e-6
cut:h  J  0  0  0
```

5.7.4.5 Example 2

The configuration shown in Listing 5.25 lowers the photon and electron energy cutoffs to 1 eV and 10 eV, respectively. This allows for the single event electron treatment to be used for the electron transport from the *electron_method_boundary* energy specified on the PHYS:e card down to 10 eV.

Listing 5.25: example_phys_e.mcnp.inp.txt

```
cut:p j 1e-6
cut:e j 1e-5
```

5.7.4.6 ELPT: Cell-by-Cell Energy Cutoff

Cell-card Form: ELPT:$\mathscr{P}$ *x*
or
Data-card Form: ELPT:$\mathscr{P}$ *x1 x2 ... xJ*

$\mathscr{P}$	Particle designator.
x	Lower energy cutoff of cell.
xj	Lower energy cutoff of cell j. Number of entries, J, equals number of cells in problem (1).

Default: Use cutoff parameters from CUT:$\mathscr{P}$

Use: Optional. For cell-dependent energy cutoff.

Details:

1 A separate lower energy cutoff can be specified for each cell in the problem. The higher of either the value on the ELPT:$\mathscr{P}$ card or the global value *e* on the CUT:$\mathscr{P}$ card applies.

5.7.5 TMP: Free-Gas Thermal Temperature

The TMP cards provide the MCNP code with the time-dependent thermal cell temperatures that are necessary for the free-gas thermal treatment of low-energy neutron transport. This treatment becomes important when the neutron energy is less than about four times the temperature of heavy nuclei or less than about 400 times the temperature of light nuclei. Thus, the TMP cards should be used when parts of the problem are not at room temperature and neutrons are transported with energies within a factor of 400 from the thermal temperature.

The TMP card has two effects. The first is that the value of TMP is used during neutron collision kinematics to properly compute the outgoing energy spectrum due to a free-gas moving target. A simple constant cross-section approximation is used in normal operation. Using the DBRC card will additionally remove this constant approximation and is recommended if the necessary 0 K libraries are available.

The second involves an adjustment to the neutron elastic scattering cross section itself. If the cell card temperatures all match and are the same as the nuclear data temperature, no adjustment is made. If the cell card temperatures all match but do not match the nuclear data temperature, the elastic scattering cross section is re-broadened using an approximate method to the cell temperatures. If the cell card temperatures do not match, the elastic cross section is un-broadened to 0 K and broadened on-the-fly to the cell temperature using this same approximation.

The approximation used makes two assumptions: the elastic scattering cross section at 0 K is constant and all other cross sections at 0 K are proportional to $1/v$. From this, the elastic scattering cross section is divided

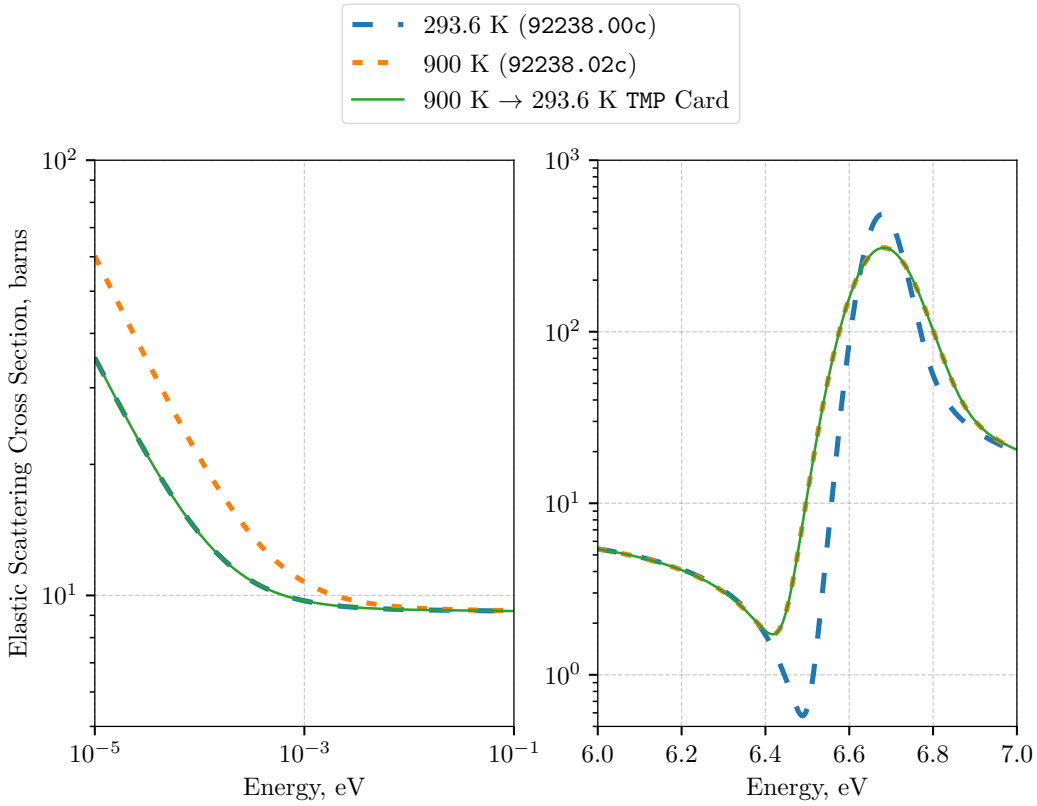


Figure 5.5: Application of TMP card on 900 K ^{238}U data to adjust temperature to 293.6 K.

by the broadened constant value at the old temperature and multiplied by the broadened constant value at the new temperature. Because nuclear data follows these assumptions at the asymptotic 0 eV limit, this is generally accurate at energies below the influence of the lowest-energy resonance. However, this assumption does not handle resonances whatsoever. The effect of this algorithm on 900 K ^{238}U data can be seen in Fig. 5.5. On the left, the de-broadened 900 K data closely follows the 293.6 K value at low energies. On the right, however, the 6.67 eV resonance is unchanged and still follows the 900 K value.

For maximum accuracy, whenever the entire geometry is not room temperature, it is recommended to set the TMP card on all cells that contain a nuclear data library to the temperature of that nuclear data library. Without TMP cards, the scattering cross section will be adjusted to room temperature using the approximation mentioned above.

Cell-card Form: *TMPn t*
or
Data-card Form: TMP*n tn1 tn2 ... tnJ*
or
Data-card Form: TMP *t1 t2 ... tJ*

n	Index of time on the thermal time (THTME) card. Restriction: $n \leq 99$.
t	Temperature of cell at time index *n*, in MeV (1, 2).
tnj	Temperature of cell *j* at time index *n*, in MeV. Number of entries equals number of cells in the problem (1, 2).
tj	Temperature of cell *j* at all times, in MeV. Number of entries equals number

of cells in the problem (no THTME is card is present; (2)).

Default: $tnj = 2.53 \times 10^{-8}$ MeV, room temperature, for all cells of the problem.

Use: Optional. Required when THTME card is used. Needed for low-energy neutron transport at other than room temperature. A fatal error occurs if a zero temperature is specified for a non-void cell.

Details:

(1) Cell thermal temperatures at times between two entries are determined by linear interpolation. Times before the first time value or after the last time value use the thermal temperature(s) at the nearest time entry.

(2) The thermal temperature of a cell is denoted by kT in units of MeV. The conversions in Table 5.9 may be convenient.

5.7.6 THTME: Thermal Times

The THTME card specifies the times at which the thermal temperatures on the TMP*n* cards are provided. For example, the temperatures on the TMP1 card are at *t1* on the THTME card; the temperatures on the TMP2 card are at time *t2* on the THTME card, etc. The times must be monotonically increasing. For each entry on the THTME card, there must be a TMP*n* card.

Data-card Form: *THTME t1 t2 ... tj*

tj	Time in shakes (10^{-8} s) at which thermal temperatures are specified on the TMP*j* card(s). Number of entries is equal to the total number of thermal times specified. Restriction: $j \leq 99$.

Default: Zero; temperature is not time dependent.

Use: Optional. Use with TMP card(s).

5.7.7 DBRC: Doppler Broadening Resonance Correction

A Doppler broadening resonance correction (DBRC) treatment is implemented to address known deficiencies in the free-gas scattering model [242, 243]. Modifications to the free-gas scattering treatment that account for non-constant scattering cross sections have been proposed and tested in previous versions of the MCNP

Table 5.9: Temperature Conversion Factors

Unit of T	kT (MeV)
K	$T \times 8.617 \times 10^{-11}$
°C	$(T + 273.15) \times 8.617 \times 10^{-11}$
°R	$T \times 4.787 \times 10^{-11}$
°F	$(T + 459.67) \times 4.787 \times 10^{-11}$

code [244–246]. With availability of 0-K nuclear cross sections that are needed to apply the DBRC treatment, the previously tested treatments are available through the DBRC data card.

To use the DBRC treatment, data tables with preprocessed energy and scattering cross section pairs at 0 K are prepared using the **dbrc_make_lib** code (see §E.1). This code is included in the MCNP6 distribution in the **MCNP_CODE/Utilities/DBRC_LIB** directory. Both the **DBRC_endf71.txt** and **DBRC_endf80.txt** files distributed are installed within the **MCNP_DATA** directory are products of the **dbrc_make_lib** code based on the 0-K scattering data from the ENDF/B-VII.1 or ENDF/B-VIII.0 nuclear data libraries, respectively. Further information on the DBRC code, data files, implementation and testing is available in a separate report [247].

The DBRC input card provides user control over the DBRC treatment. If the DBRC card is not present among the data cards, then the traditional free-gas scattering treatment is used, with free-gas scattering for ^{1}H at all energies and free-gas scattering at energies below 400 kT (10.12 eV at room temperature) for all other nuclides (for energies higher than the range of $S(\alpha, \beta)$ [§2.3.6] data if used for a nuclide).

If the DBRC card is present among the data cards, then the following keyword-value options are available:

Data-card Form: DBRC KEYWORD = *values(s)*

endf = *nn*	library identifier for selecting scattering data at 0 K. • Default data available include *nn* = 71 or *nn* = 80, representing ENDF/B-VII.1 or ENDF/B-VIII.0 scattering data, respectively. • The **DBRC_endf***nn***.txt** file in the **DATAPATH** contains all of the preprocessed 0-K scattering data for a given library. • This entry is required if one or more nuclides are listed in the *iso_list*. • Note that 0-K data cannot be mixed between libraries.
emax = *eee*	the upper energy limit for applying DBRC for all nuclides except ^{1}H, in units of MeV. • The default is 2.1×10^{-4} MeV. • If *eee* is specified and the *iso_list* is not present, then conventional free-gas scattering is performed for all nuclides up to *eee*, rather than the traditional 400 kT limit. • *eee* must be less than or equal to the **DBRC_endf***nn***.txt** datafile upper limit, currently 2.5×10^{-4} MeV.
isos = *iso_list*	list of one or more target identifiers [§1.2.2]. All target formats are allowed. • The DBRC treatment applies to every isotope listed in all materials.

Default: Use traditional free-gas scattering treatment.

Use: Optional.

5.7.7.1 Example 1

The data card in Listing 5.26 will apply the DBRC treatment to ^{238}U, using the ENDF/B-VII.1 0-K scattering cross sections, and a default energy cutoff of 2.1×10^{-4} MeV.

Listing 5.26: example_dbrc_1.mcnp.inp.txt

```
dbrc  endf=71  emax=2.10e-4  isos=U-238
m1    U-238.83c  1
```

5.7.7.2 Example 2

The data card in Listing 5.27 will apply the DBRC treatment to ^{234}U, ^{235}U, ^{236}U, and ^{238}U, using the ENDF/B-VIII.0 0-K scattering cross sections, and an energy cutoff of 2.3×10^{-4} MeV.

Listing 5.27: example_dbrc_2.mcnp.inp.txt

```
dbrc  endf=80  emax=2.3e-4  isos=U-234 U-235 U-236 U-238
m1    U-234.00c  0.011150
      U-235.00c  0.97694
      U-236.00c  0.0019919
      U-238.00c  0.0099250
m2      H-1.00c  0.66667
       O-16.00c  0.33320
       O-17.00c  1.3333e-4
mt2    h-h2o
```

5.7.8 Model Physics and Physics Models

5.7.8.1 MPHYS: Model Physics Control

The use of physics models is controlled with the MPHYS card. When isotopes that are missing cross-section libraries in a problem or when reactions exceed a library's maximum energy, MCNP6's behavior can change whether physics models are being used or not.

Data-card Form: MPHYS *toggle*

toggle	Control to enable or disable model physics. If the value of toggle is
	on, then model physics are enabled (default for particles other than n, p, e).
	off, indicates that model physics are disabled (default for n, p, e).

Default: All MODE n p e problems (and subsets) run with physics models off (MPHYS off) by default. Any particle on the MODE card other than n, p, or e will automatically activate the use of physics models (MPHYS on).

Use: To disable the use of physics models, set MPHYS off. To enable the use of physics models, set MPHYS on or include the MPHYS card with no entries.

5.7.8.2 Physics Models Options

Five cards (LCA, LCB, LCC, LEA, and LEB) control physics parameters for the Bertini [248, 249], ISABEL [250, 251], CEM03.03 and LAQGSM03.03 [200–216], and INCL4 [252] with ABLA [253, 254] options. All of the input values on the five cards have defaults, which will be taken in the absence of the cards, or with the use of the J input option.

These MCNP6 input cards provide the user control of physics options. A summary of the cards follows. The options controlling the Bertini and ISABEL physics modules are taken from [255]. The user is referred to that document for further information.

Table 5.10: Permissible Model Physics Combinations

	LCA 3rd entry (*iexisa*)	LCA 9th entry (*icem*)	LEA 7th entry (*ievap*)
Bertini/Dresner	1	0	0
ISABEL/Dresner	2	0	0
Bertini/ABLA	1	0	2
ISABEL/ABLA	2	0	2
CEM03.03	N/A	1	N/A
INCL4/Dresner	0	2	0
INCL4/ABLA	0	2	2

Table 5.10 shows how different combinations of physics models are possible using the 3rd and 9th entries on the LCA card, *iexisa* and *icem*, and the 7th entry on the LEA card, *ievap*. The CEM03.03 model contains an intranuclear cascade model and evaporation/fission models; therefore, the *iexisa* and *ievap* options are not applicable when *icem* = 1.

Caution

Combinations of options for the physics models should be chosen with careful consideration. Although many combinations are allowed, inappropriate choices can lead to incorrect results.

5.7.8.2.1 LCA

The LCA card is used to select the Bertini, ISABEL, CEM03.03, or INCL4 model, as well as to set certain parameters used in Bertini and ISABEL. CEM03.03 is a self-contained package with no user-adjustable options presently defined.

Data-card Form: LCA *ielas ipreq iexisa ichoic jcoul nexite npidk noact icem ilaq nevtype*

ielas Controls elastic scattering. If

- *ielas* = 0, then no nucleon elastic scattering.
- *ielas* = 1, then elastic scattering for neutrons only.
- *ielas* = 2, then elastic scattering for neutrons and protons (DEFAULT).

ipreq Controls pre-equilibrium model [256] for Bertini and ISABEL (1). If

- *ipreq* = 0, no pre-equilibrium model will be used.
- *ipreq* = 1, use pre-equilibrium model after intranuclear cascade (DEFAULT).
- *ipreq* = 2 and *iexisa* = 0, select *ipreq* = 1 and *ipreq* = 3 randomly, with an energy-dependent probability that goes to *ipreq* = 3 at low energies and to *ipreq* = 1 at high incident energies. If *iexisa* ≠ 0, defaults to *ipreq* = 1.
- *ipreq* = 3 and *iexisa* = 0, use pre-equilibrium model instead of the intranuclear cascade. If *iexisa* ≠ 0, defaults to *ipreq* = 1.

iexisa	Controls model choice (2, 3). If
iexisa = 0,	do not use ISABEL intranuclear cascade (INC) model for any particle (DEFAULT if *icem* = 2, which specifies the INCL4 model).
iexisa = 1,	use Bertini model for nucleons and pions and ISABEL model for other particle types (DEFAULT).
iexisa = 2,	use ISABEL model for all incident particle types.
ichoic	Four integers (*ijkl*) that control ISABEL intranuclear cascade model (DEFAULT: *ichoic* = 0023). If
i = 0,	use partial Pauli blocking (DEFAULT).
i = 1,	use total Pauli blocking.
i = −2,	do not use Pauli blocking (not recommended).
j = 0,	no interaction between particles already excited above the Fermi sea (DEFAULT).
j > 0,	*j* is the number of time steps to elapse between such "CAS-CAS" interactions.
k = 0,	use Meyer's density prescription with 8 steps.
k = 1,	use original (isobar) density prescription with 8 steps.
k = 2,	use Krappe's folded-Yukawa prescription for radial density in 16 steps, with a local density approximation to the Thomas-Fermi distribution for the (sharp cutoff) momentum distribution (DEFAULT).
k = 3,	the choice is the same as *k* = 0 but using the larger nuclear radius of the Bertini model.
k = 4,	the choice is the same as *k* = 1 but using the larger nuclear radius of the Bertini model.
k = 5,	the choice is the same as *k* = 2 but using the larger nuclear radius of the Bertini model.
l = 1,	perform reflection and refraction at the nuclear surface, but no escape cutoff for isobars.
l = 2,	perform reflection and refraction at the nuclear surface, with escape cutoff for isobars.
l = 3,	perform no reflection or refraction, with escape cutoff for isobars (DEFAULT).
l = 4,	the choice is the same as *l* = 1 but using a 25-MeV potential well for pions.
l = 5,	the choice is the same as *l* = 2 but using a 25-MeV potential well for pions.
l = 6,	the choice is the same as *l* = 3 but using a 25-MeV potential well for pions.
jcoul	Controls Coulomb barrier for incident charged particles. If

	jcoul = 1,	the Coulomb barrier is on (DEFAULT).
	jcoul = 0,	the Coulomb barrier is off.
nexite	Subtract nuclear recoil energy to get excitation energy. If	
	nexite = 1,	this feature is on (DEFAULT).
	nexite = 0,	this feature is off.
npidk	Controls pion termination treatment. If	
	npidk = 0,	force π^- to interact by nuclear capture (INC) when cutoff is reached (DEFAULT).
	npidk = 1,	force π^- to terminate by decay at the pion cutoff energy (4).
noact	Particle transport options. If	
	noact = −2,	source particles immediately collide; all progeny escape. In other words, all secondary particles produced are transported with no interactions and no decay. Used to compute and tally double-differential cross sections and residual nuclei with an F1 or F8 tally in conjunction with the FT *res* option (5).
	noact = −1,	nuclear interactions of source particles only; transport and slowing down are off.
	noact = 0,	turn off all non-elastic reactions.
	noact = 1,	perform normal transport (DEFAULT).
	noact = 2,	attenuation mode; transport primary source particles without non-elastic reactions.
icem	Choose alternative physics model. If	
	icem = 0,	use the Bertini or ISABEL model determined by the *iexisa* parameter.
	icem = 1,	use the CEM03.03 model (DEFAULT) (6).
	icem = 2,	use INCL4 model (7). Default evaporation model is ABLA; see *ievap* on LEA card.
ilaq	Choose light ion and nucleon physics modules (7). If	
	ilaq = 0,	use LAQGSM03.03 to handle all heavy-ion interactions as well as all light-ion interactions above 940 MeV/nucleon. ISABEL will handle light-ion interactions below this energy. Use LAQGSM03.03 for proton and neutron interactions above the energy cutoff specified by parameters *flenb1* and *flenb2* on the LCB card (DEFAULT).
	ilaq = 1,	use LAQGSM03.03 to handle all heavy-ion interactions as well as all light-ion interactions.
nevtype	Choose number of evaporation particles modeled by GEM2 (DEFAULT: *nevtype* = 66). If *nevtype* = N, evaporation modeling is limited to the lightest N particles. *nevtype* has a minimum value of 6, which includes n, p, d, t, ^{3}He, and ^{4}He). Values below this will be raised to 6 (8).	

Use: CEM03.03 and LAQGSM03.03 are highly recommended (LCA 8J 1 1); *noact* is very useful for examining single reactions, i.e., interactions with nuclei without transport.

Details:

1. CEM03.03 and LAQGSM03.03 use their own pre-equilibrium model [200, 204, 216] all the time. INCL uses no pre-equilibrium model.
2. The antinucleons and kaons are unaffected by the choice of physics models. They always choose ISABEL below the *flenb5* energy and LAQGSM03.03 above the *flenb6* energy (see LCB card). At energies intermediate to these two, a weighted random choice is made between the two models.
3. The ISABEL INC model requires a much greater execution time. In addition, incident particle energies must be less than 1 GeV per nucleon for light ions (at higher energies, the LAQGSM03.03 model is automatically invoked).
4. The capture probability for any isotope in a material is proportional to the product of the number fraction and the charge of the isotope. However, capture on ^{1}H leads to decay rather than interaction.
5. If *noact* $= -2$ on the LCA card, table physics will be used whenever possible to get the differential data actually used in a given problem. To get the differential data with models only, table data can be turned off by setting the *cutn* parameter on the PHYS:n card and the *tabl* parameter on the PHYS:h card.
6. CEM03.03 allows neutrons, protons, pions, and photons to initiate nuclear reactions. We recommend when possible using CEM03.03 for target-nuclei energies up to about 5 GeV for reactions induced by nucleons and pions on heavy nuclei-targets, up to about 1.2 GeV for photonuclear reactions, and up to about 1 GeV for reactions on light nuclei-targets.
   - Although results from CEM03.03 are expected to be more reliable in these energy regions, CEM03.03 is expected also to work quite reliably for all target-nuclei at energies up to about 5 GeV. CEM03.03 consists of an IntraNuclear Cascade (INC) model [216, 257, 258], followed by its own pre-equilibrium model [200, 204, 216] and an evaporation model (see details in [216] and references therein).
   - Possible fission events are initiated in the equilibrium stage for compound nuclei with a charge number $Z > 65$. The evaporation/fission is handled by a modification of the Generalized Evaporation/Fission Model (GEM2) [259], which is an extension to an earlier evaporation model [260] and fission model [261] as described in [262]. Fission fragments undergo an evaporation stage that depends on their excitation energy. When the mass number of excited nuclei produced after INC, as well as after and during the pre-equilibrium and evaporation/fission stages of reactions, $A < 13$, CEM03.03 uses the Fermi break-up model to calculate the following de-excitation, instead of using the pre-equilibrium and/or the evaporation/fission models. After the last stage of a reaction calculated by CEM03.03 (usually, the evaporation), a de-excitation of the residual nucleus follows in MCNP6 (but not in CEM03.03 when used as a stand-alone code [216]), generating gammas with the PHT code adopted from LAHET [255].
7. By default, light ions (d, t, ^{3}He, ^{4}He) are handled by ISABEL below 940 MeV/nucleon and LAQGSM03.03 above 940 MeV/nucleon. Specifying *ilaq* $= 1$ will send them to LAQGSM03.03 at all energies. Specifying *icem* $= 2$ will instead send them to INCL for all energies.
8. By default, GEM2 models the evaporation of 66 types of particles (up to ^{28}Mg). As heavier nuclei often have negligible fission/evaporation probabilities, specifying *nevtype* $= N$, limits evaporation modeling to the lightest N particles. A minimum number of 6 particle types (n, p, d, t, ^{3}He, and ^{4}He) is needed, and will default to 6 when *nevtype* < 6. It is recommended that users of CEM03.03 and LAQGSM03.01 use a *nevtype* value of 66 only when evaporation of fragments heavier than ^{4}He are desired; otherwise the value of 6 is recommended to save computational performance.

5.7.8.2.2 LCB

The LCB card controls which physics module is used for particle interactions depending on the kinetic energy of the particle.

Data-card Form: LCB *flenb1 flenb2 flenb3 flenb4 flenb5 flenb6 ctofe flim0*

flenb1 Kinetic energy (DEFAULT: *flenb1* = 3500 MeV). For nucleons, the CEM/Bertini/INCL INC model will be used below this value (1, 2). See the LCA *icem* parameter for choice of INC model.

flenb2 Kinetic energy (DEFAULT: *flenb2* = 3500 MeV). For nucleons, the LAQGSM03.03 high-energy generator will be used above this value (1, 2). See the LCA *ilaq* parameter for choice of high-energy model.

flenb3 Kinetic energy (DEFAULT: *flenb3* = 2500 MeV). For pions, the CEM/Bertini/INCL INC model will be used below this value (2, 3). See the LCA *icem* parameter for choice of INC model.

flenb4 Kinetic energy (DEFAULT: *flenb4* = 2500 MeV). For pions, the LAQGSM03.03 high-energy generator will be used above this value (2, 3). See the LCA *ilaq* parameter for choice of high-energy model.

flenb5 Kinetic energy (DEFAULT: *flenb5* = 800 MeV). The ISABEL INC model will be used below this value (2).

flenb6 Kinetic energy (DEFAULT: *flenb6* = 800 MeV). An appropriate model will be used above this value (2). For

iexisa = 2,	*flenb5* and *flenb6* apply to all particle types.
iexisa = 1,	*flenb5* and *flenb6* apply to all particles except nucleons and pions.
iexisa = 0,	*flenb5* and *flenb6* are immaterial.

See §5.7.8.2.2.1 for further explanation.

ctofe The cutoff kinetic energy (MeV) for particle escape during the INC when using the Bertini model. The cutoff energy prevents low-energy nucleons from escaping the nucleus during the INC; for protons, the actual cutoff is the maximum of *ctofe* and a Coulomb barrier. If

ctofe ≥ 0,	*ctofe* will be used as the cutoff energy.
ctofe < 0,	a random cutoff energy, uniformly distributed from zero to twice the mean binding energy of a nucleon will be sampled for each projectile-target interaction and separately for neutrons and protons. In this case the Coulomb barrier for protons is also randomized (DEFAULT: *ctofe* $= -1.0$).

For the ISABEL INC, the randomized cutoff energy is always used.

flim0 The maximum correction allowed for mass-energy balancing in the cascade stage, used with *nobalc* = 1 on the LEA card. If

flim0 > 0,	kinetic energies of secondary particles will be reduced by no more than a fraction of *flim0* in attempting to obtain a non-negative excitation of the residual nucleus and a consistent mass-energy balance. A cascade will be resampled if the correction exceeds *flim0*.
flim0 $= 0$,	no correction will be attempted and a cascade will be resampled if a negative excitation is produced.
flim0 < 0,	for incident energy, E, the maximum correction is 0.02 for $E > 250$ MeV, 0.05 for $E < 100$ MeV, and is set equal to $5/E$ between those limits (DEFAULT: *flim0* $= -1.0$).

Details:

1. For nucleons, the Bertini model switches to a scaling procedure above 3.495 GeV, wherein results are scaled from an interaction at 3.495 GeV. Although both models will execute to arbitrarily high energies, a plausible upper limit for the Bertini scaling law is 10 GeV.
2. The interaction model selected is sampled uniformly, but weighted by proximity to the energy bound, between *flenb1* and *flenb2*, *flenb3* and *flenb4*, or *flenb5* and *flenb6*, as appropriate.
3. For pions, the Bertini model switches to the scaling law method above 2.495 GeV.

5.7.8.2.2.1 Example 1

The configuration shown on the LCB card in Listing 5.28 changes the default energy-boundary switches and the ranges for stochastic model selection sampling for all nucleon and pion interactions.

Listing 5.28: example_mphys_lcab.mcnp.inp.txt

```
lca   2j 2 4j -2 0
lcb   3000  3000  2000  2000  1000  1000
```

For *iexisa* = 1, the default model on the LCA card, nucleons will switch to the Bertini model from the LAQGSM03.03 model below *flenb1* = 3 GeV, and pions will switch below *flenb3* = 2 GeV. Kaons and anti-nucleons will switch to the ISABEL model from the LAQGSM03.03 model below 1 GeV. Muons have no nuclear interactions.

For *iexisa* = 2 , selected by the 3rd entry on the LCA card in Listing 5.28, nucleons and pions will switch to the ISABEL model below *flenb5* = 1 GeV. Note that the upper energy threshold in the ISABEL version used by MCNP6 is 1 GeV/nucleon. No interactions are allowed at energies greater than this value.

5.7.8.2.3 LCC

The LCC card specifies control parameters for the INCL4 model and the ABLA fission-evaporation model. INCL4 is invoked by setting the 9th LCA card entry, *icem* = 2, and ABLA is invoked by setting the 7th LEA card entry, *ievap* = 2.

Data-card Form: `LCC` *`stincl v0incl xfoisaincl npaulincl nosurfincl J J ecutincl ebankincl ebankabla`*

`stincl`	Rescaling factor of the cascade duration (DEFAULT: *`stincl`* = 1.0).
`v0incl`	Potential depth (DEFAULT: *`v0incl`* = 45 MeV).
`xfoisaincl`	Controls the maximum impact parameter for Pauli blocking, $rmaxws = r_0 + $ *`xfoisaincl`* $\times a$, where r_0 is the radius of the nucleus and a is the diffuseness (DEFAULT: *`xfoisaincl`* = 8.0).
`npaulincl`	Controls the Pauli blocking parameter. If
	`npaulincl` = 1, use Pauli strict blocking.
	`npaulincl` = 0, use Pauli statistic blocking (DEFAULT).
	`npaulincl` = −1, no Pauli blocking.
`nosurfincl`	Controls the diffuse nuclear surface based on Wood-Saxon density. If
	`nosurfincl` = −2, use Wood-Saxon density and INCL4 stopping time (DEFAULT).
	`nosurfincl` = −1, use Wood-Saxon density and stopping time with impact dependence.
	`nosurfincl` = 0, use Wood-Saxon density and stopping time without impact dependence.
	`nosurfincl` = 1, use sharp surface.
`J`	Unused placeholder. Be sure to put the `J` in the keyword string.
`J`	Unused placeholder. Be sure to put the `J` in the keyword string.
`ecutincl`	Use Bertini model below this energy (DEFAULT: *`ecutincl`* = 0).
`ebankincl`	Write no INCL bank particles below this energy (DEFAULT: *`ebankincl`* = 0).
`ebankabla`	Write no ABLA bank particles below this energy (DEFAULT: *`ebankabla`* = 0).

5.7.8.2.4 LEA

The `LEA` card controls evaporation, Fermi-breakup, level-density parameters, and fission models. These are external to the particular intranuclear cascade/pre-equilibrium model chosen (Bertini, ISABEL, or INCL), and may be used with any of these choices (except CEM03.03 and LAQGSM03.03).

Data-card Form: `LEA` *`ipht icc nobalc nobale ifbrk ilvden ievap nofis`*

`ipht`	Control generation of de-excitation photons. If
	`ipht` = 0, generation of de-excitation photons is off.
	`ipht` = 1, generation of de-excitation photons is on (DEFAULT).
`icc`	Defines the level of physics to be applied for the LAHET-PHT [255, 263]

photon physics. If

icc = 0,	use the continuum model.
icc = 1,	use the Troubetzkoy (E1) model.
icc = 2,	use the intermediate model (hybrid between *icc* = 1 and *icc* = 2).
icc = 3,	use the spin-dependent model.
icc = 4,	use the full model with experimental branching ratios (DEFAULT).

nobalc Controls mass-energy balancing in the cascade stage (see [255] for historical information). A forced energy balance may distort the intent of any intranuclear cascade model. Energy balancing for the intranuclear cascade is controlled by the input parameter *flim0* on the LCB card. If

nobalc = 0,	use mass-energy balancing in the cascade phase.
nobalc = 1,	turn off mass-energy balancing in the cascade phase (DEFAULT).

nobale Controls mass-energy balancing in evaporation stage (see [255] for historical information). If

nobale = 0,	use mass-energy balancing in the evaporation stage (DEFAULT).
nobale = 1,	turn off mass-energy balancing in the evaporation stage.

ifbrk Controls Fermi-breakup model nuclide range. If

ifbrk = 1,	use Fermi-breakup model for atomic mass number $A \leq 13$ and for $14 \leq A \leq 20$ with excitation below 44 MeV (DEFAULT).
ifbrk = 0,	use Fermi-breakup model only for atomic mass number $A \leq 5$.

ilvden Controls level-density model. If

ilvden = −1,	use original HETC level-density formulation. See the LEB card for details on parameter inputs.
ilvden = 0,	use Gilbert-Cameron-Cook-Ignatyuk level-density model [256] (DEFAULT).
ilvden = 1,	use the Julich level-density parameterization as a function of mass number [264].

ievap Controls evaporation and fission models (1). If

ievap = 0,	use the RAL fission model [265].
ievap = −1,	use the ABLA evaporation model with its built-in fission model when *icem* = 2 , and use the RAL fission model [265] for all other cases (see *icem* on the LCA card) (DEFAULT).
ievap = 1,	use the ORNL fission model [266]. The ORNL model allows fission only for isotopes with atomic number $Z \geq 91$.

	ievap = 2,	use the ABLA evaporation model with its built-in fission model.
nofis	Controls fission. If	
	nofis = 1,	allow fission (DEFAULT).
	nofis = 0,	suppress fission.

Details:

1. Bertini and ISABEL invoke the Dresner evaporation model with Rutherford Appleton Laboratory (RAL) fission by default. The fission model can be switched to the ORNL model using the *ievap* option on the LEA card. The evaporation model can be switched from Dresner to ABLA (with its built-in fission model) by setting *ievap* = 2.

5.7.8.2.5 LEB

The LEB card controls level-density input options for the original HETC implementation, *ilvden* = −1 on the LEA card.

Data-card Form: LEB *yzere bzere yzero bzero*

yzere	The $Y0$ parameter in the level-density formula for atomic number $Z \leq 70$ (DEFAULT: *yzere* = 1.5). Zero or negative is an error condition. If the atomic number of the target nucleus is	
	$Z \leq 70$,	the *bzere* and *yzere* parameters are used to compute level densities. The default values are those used in LAHET before installation of the ORNL fission model.
	$Z \geq 71$,	the *bzero* and *yzero* parameters are used to compute level densities for the target nucleus and fission fragments.
bzere	The $B0$ parameter level-density formula for atomic number $Z \leq 70$ (DEFAULT: *bzere* = 8.0). Zero or negative is an error condition; see *yzere* above.	
yzero	The $Y0$ parameter in the level-density formula for atomic number $Z \geq 71$ and all fission fragments (DEFAULT: *yzero* = 1.5). Zero or negative is an error condition; See *yzere* above.	
bzero	The $B0$ parameter in the level-density formula for atomic number $Z \geq 71$ and all fission fragments (DEFAULT: *bzero* = 10.0 for *ievap* = 0 and for *ievap* = 1 on the LEA card). Zero and negative is an error condition; see *yzere* above.	

5.7.9 FMULT: Fission Multiplicity Constants and Physics Models

For neutron-induced fission, the average value of neutron multiplicity, $\overline{\nu}$, is available in nuclear data libraries and is a function of the incident neutron energy. Historically, the neutron-induced fission multiplicity probability distribution, $P(\nu)$, is unavailable in nuclear data libraries. Additionally, the nuclear data libraries that contain the projectile-target reaction data for neutron-induced fission reactions do not include spontaneous fission decay data. To simulate the individual neutron emissions from a spontaneous fission source, the combination of the `par` = `sf` particle type on the `SDEF` card and the multiplicity constants and physics model options on the `FMULT` card are needed. Additionally, when using the `FMULT` card, the neutron-induced fission multiplicity is simulated regardless of whether a spontaneous fission source is present.

Data-card Form: `FMULT` *target_identifier* `KEYWORD` = `value(s)`

target_identifier	Target identifier [§1.2.2] (1). All formats supported. At this time, any values set for non-metastable nuclides also apply to the corresponding metastable nuclide, and a metastable input is converted to a non-metastable input.
`sfnu`	If
	`sfnu` = `x`, where `x` is a single value, then `x` is the average neutron multiplicity, $\overline{\nu}$, used for sampling spontaneous fission multiplicity from a Gaussian distribution with width `w` (2).
	`sfnu` = `x0 x1 ... x`N, where multiple values are provided, then the `x`N values form the ordinates of a discrete cumulative probability distribution of spontaneous fission multiplicity, $P(\nu = n)$ for $n = 0 \ldots N$ (2). A maximum of ten values may be specified, simulating between 0 and 9 neutrons emitted in a single spontaneous fission event.
`width` = `w`	Gaussian width full-width at half maximum (FWHM) for sampling $P(\nu)$ for both spontaneous and neutron-induced fission. This value is ignored for spontaneous fission when `sfnu` is specified as a cumulative probability distribution (2).
`sfyield` = `y`	Spontaneous fission yield (n/s-g). Required for selecting the spontaneously fissioning nuclide when more than one is present in a material (2).
`watt` = `a b`	Watt energy spectrum parameters `a` and `b` (see Eq. (5.27)) for spontaneous fission neutron energy sampling (2).
`method` = `m`	Use to select the Gaussian sampling algorithm method or model physics option (3). If
	`method` = 0, use the MCNP5 sine/cosine sampling method (DEFAULT, see Table 5.11).
	`method` = 1, use the Lestone moment-fitting method [73]; this is MCNPX polar sampling with 0.5 added to the result.
	`method` = 3, use the Ensslin/Santi/Beddingfield/Mayo method [267, 268]; this is MCNPX polar sampling with a random number between 0 and 1 added to the result.

	`method` = 5,	use the LLNL fission library for neutron-induced, spontaneous, and photonuclear (if `ispn` ≠ 0 and `fism` = 1 on the PHYS: `p` card) fission [237] (4, 5). Restriction: `method` = 5 cannot be used with delayed neutron biasing (`dnbias` on ACT card).
	`method` = 6,	use the FREYA fission model for neutron-induced and spontaneous fission [269] (4, 5, 6). Restriction: `method` = 6 cannot be used with delayed neutron biasing (`dnbias` on ACT card).
	`method` = 7,	use the CGMF fission model for neutron-induced and spontaneous fission [270] (4, 5, 6). Restriction: `method` = 7 cannot be used with delayed neutron biasing (`dnbias` on ACT card).
	Note: All other values for `m` are unused.	
`data` = `d`	Use to select data for isotope multiplicities (3). If	
	`data` = 0,	use bounded integer fission sampling (DEFAULT).
	`data` = 1,	use Lestone re-evaluated Gaussian width by isotope for multiplicities.
	`data` = 2,	use original Terrell Gaussian widths by isotope for multiplicities [72].
	`data` = 3,	use Ensslin/Santi/Beddingfield/Mayo.
`shift` = `s`	Designate method to modify the sampled ν to preserve the average multiplicity, $\overline{\nu}$ (3). If	
	`shift` = 0,	use the MCNP5 treatment, which assumes an integer number of neutrons per fission. For example, if $\overline{\nu} = 2.7$, then the number of neutrons will be two 30% of the time and three 70% of the time (DEFAULT).
	`shift` = 1,	use the MCNPX-style adjustment method that uses a re-evaluated Gaussian width to sample fission neutron multiplicities for all fissionable isotopes.
	`shift` = 2,	sample the Gaussian distribution and preserve the average multiplicity by increasing the $\overline{\nu}$ threshold.
	`shift` = 3,	sample the Gaussian distribution without correction. This will overpredict $\overline{\nu}$.
	`shift` = 4,	use the MCNP4C integer sampling method in the presence of spontaneous fission.

Defaults shown correspond to the condition that no `method`, `data`, or `shift` keywords are specified. If any of these keywords appear, the code will automatically assign values for the unspecified keywords. The default assignments otherwise are `method` = 3, `data` = 3, and `shift` = 1.

Use: Optional. Enables users to override or add additional fission multiplicity data.

Table 5.11: Mapping from MCNP5 and MCNPX PHYS:n Options to MCNP6 FMULT Options (3)

MCNPX PHYS:n *fism* (6th entry)	MCNP6 FMULT method	data	shift
0	0	0	0
-1, 1	3	3	1
2	3	3	2
3	3	3	3
4	3	3	4
5	5	N/A	N/A
MCNP5 PHYS:n *fisnu* (5th entry)	**MCNP6 FMULT method**	**data**	**shift**
0	0	0	0
1	0	1	0
2	0	2	0

Details:

(1) When overriding the default values for the sfnu, width, sfyield or watt keywords, the *target_identifier* option must be specified. Defaults exist for the most common fissioning nuclei; these defaults are provided in PRINT Table 38 of the MCNP output [267, 271–279]. To have a spontaneous fission source for nuclides without default values (zero values in PRINT Table 38), a FMULT data card is required. Without a *target_identifier* option specified, only the fissioning nuclei that are missing default values will inherit the specified keyword values. method, shift, and data keywords are not isotope specific while the rest of the FMULT keywords are isotope specific; therefore, *target_identifier* is optional.

(2) The sfnu, sfyield, and watt keywords are only applicable to spontaneous fission isotopes. For neutron-induced fission, the average fission neutron multiplicity, $\overline{\nu}$, and the fission spectrum, χ, come from the nuclear data libraries at the energy of the incident neutron causing fission. The width keyword value is required for the neutron-induced fission isotope even if a cumulative distribution on the sfnu keyword is specified for spontaneous fission. The spontaneous fission yield (sfyield) must be specified if more than one spontaneous fission source nuclide occurs.

(3) The specific method, shift, and data parameter combinations listed in Table 5.11 are the only ones assured to work correctly. Other combinations are possible but have not been tested. While the method, shift, and data keywords may be specified on multiple FMULT cards in the input deck, only the last instance of each keyword determines the algorithms and data used for multiplicity sampling. When specifying method = 5, 6, or 7, both shift and data keywords are not applicable and should not be used.

(4) The LLNL fission library, the FREYA fission model, and the CGMF fission model are the only ways in MCNP6 to produce correlated prompt fission photons with multiplicities for spontaneous fission and low-energy neutron-induced fission events. For all other method values, no spontaneous fission photons are produced, and the neutron-induced photon production comes from the nuclear data libraries where fission and non-fission photons produced at a collision may not be distinguishable depending on the incident energy of the neutron and the library in use. Delayed fission photons are independent of the selected method and are controlled by the ACT card.

(5) If the LLNL fission library, the FREYA fission model, or the CGMF fission model is used, then only the spontaneous fission yield (sfyield) is used for nuclides in the respective model. For fissioning nuclides not in the LLNL fission library, the FMULT parameters are used.

(6) If either the FREYA or CGMF fission model is used, and the fission nuclide is unavailable, the LLNL fission library is used to emit correlated neutrons and photons.

5.7.9.1 Multiplicity Parameters Default Values

The spontaneous fission multiplicity constants in PRINT Table 38, shown in Listing 5.29, are the default values [272] of the multiplicity parameters in MCNP6. These constants are printed with three digits of precision, but they are represented in the MCNP source code with seven digits of precision.

Data actually used within the simulation are denoted by an $*$, shown in Listing 5.29. If any data are overridden by FMULT user input, the user data replaces the default data shown in PRINT Table 38. If the LLNL fission library, FREYA, or CGMF methods are selected, additional informational messages can be seen in the output file below PRINT Table 38.

Fission Watt spectra parameters and fission yields are not available for the following nuclides: ^{246}Pu, ^{246}Cm, ^{248}Cm, ^{246}Cf, ^{250}Cf, ^{254}Cf, ^{257}Fm, and ^{252}No.

Listing 5.29: Default Fission Multiplicity Constants in Print Table 38

```
1fission multiplicity data.                                                                                print table 38

 Gaussian widths from Lestone, LA-UR-05-0288 (2005)

    zaid     width     watt1     watt2     yield   sfnu

   90232  1.069754  0.800000  4.000000  6.00E-08  2.140
   92232  1.069754  0.892204  3.722780  1.30E+00  1.710
   92233  1.070000  0.854803  4.032100  8.60E-04  1.760
   92234  1.084430  0.771241  4.924490  5.02E-03  1.810
 * 92235  1.088000  0.774713  4.852310  2.99E-04  1.860
   92236  1.099106  0.735166  5.357460  5.49E-03  1.910
 * 92238  1.116000  0.648318  6.810570  1.36E-02  0.048  0.297  0.722  0.950  0.993  1.000  1.000  1.000  1.000  1.000
   93237  1.106444  0.833438  4.241470  1.14E-04  2.050
   94236  1.099106  0.000000  0.000000  0.00E+00  0.080  0.293  0.670  0.905  0.980  1.000  1.000  1.000  1.000  1.000
   94238  1.113782  0.847833  4.169330  2.59E+03  0.056  0.267  0.647  0.869  0.974  1.000  1.000  1.000  1.000  1.000
   94239  1.140000  0.885247  3.802690  2.18E-02  2.160
   94240  1.128458  0.794930  4.689270  1.02E+03  0.063  0.295  0.628  0.881  0.980  0.998  1.000  1.000  1.000  1.000
   94241  1.150000  0.842472  4.151500  5.00E-02  2.250
   94242  1.143134  0.819150  4.366680  1.72E+03  0.068  0.297  0.631  0.879  0.979  0.997  1.000  1.000  1.000  1.000
   95241  1.135796  0.933020  3.461950  1.18E+00  3.220
   96242  1.143134  0.887353  3.891760  2.10E+07  0.021  0.168  0.495  0.822  0.959  0.996  0.999  1.000  1.000  1.000
   96244  1.157810  0.902523  3.720330  1.08E+07  0.015  0.131  0.431  0.764  0.948  0.991  1.000  1.000  1.000  1.000
   96246  1.172486  0.000000  0.000000  0.00E+00  0.015  0.091  0.354  0.699  0.917  0.993  1.000  1.000  1.000  1.000
   96248  1.187162  0.000000  0.000000  0.00E+00  0.007  0.066  0.287  0.638  0.892  0.982  0.998  1.000  1.000  1.000
   97249  1.194500  0.891281  3.794050  1.00E+05  3.400
   98246  1.172486  0.000000  0.000000  0.00E+00  0.001  0.114  0.349  0.623  0.844  0.970  1.000  1.000  1.000  1.000
   98250  1.201838  0.000000  0.000000  0.00E+00  0.004  0.040  0.208  0.502  0.801  0.946  0.993  0.997  1.000  1.000
   98252  1.216514  1.180000  1.034190  2.34E+12  0.002  0.028  0.153  0.427  0.733  0.918  0.984  0.998  1.000  1.000
   98254  1.231190  0.000000  0.000000  0.00E+00  0.000  0.019  0.132  0.396  0.714  0.908  0.983  0.998  1.000  1.000
  100257  1.253204  0.000000  0.000000  0.00E+00  0.021  0.073  0.190  0.390  0.652  0.853  0.959  0.993  1.000  1.000
  102252  1.216514  0.000000  0.000000  0.00E+00  0.057  0.115  0.207  0.351  0.534  0.717  0.863  0.959  0.997  1.000

 * = used in problem.
```

5.7.9.2 Example 1

Listing 5.30: example_fmult_1.mcnp.inp.txt

```
fmult method=0 data=1 shift=0
```

This input card only specifies the method, data, and shift keywords, relying on the default values for the fission multiplicity and spectrum constants given in PRINT Table 38 in Listing 5.29.

5.7.9.3 Example 2

Listing 5.31: example_fmult_2.mcnp.inp.txt

```
sdef  par=sf
fmult Pu-239  width=1.16   watt=0.885247 3.8026   sfyield=0.0218   sfnu=2.1
fmult Cf-252  width=1.207  watt=1.18     1.03419  sfyield=2.34e12
      sfnu=0.002 0.028 0.155 0.428 0.732 0.917 0.983 0.998 1.0
```

These input cards specify a spontaneous fission source on the SDEF card and user-specified values for the width, watt, sfnu, and sfyield keywords. PRINT Table 38 includes all of the user-specified constants given on the FMULT cards and the remaining default values not overridden by the user input. PRINT Tables 117 and 115 in the output file include information about the spontaneous and/or induced fission multiplicity sampling that occurred in the simulation, including the moments of the sampled ν values, and a summary of all fission neutron multiplicity, respectively.

5.7.9.4 Example 3

Listing 5.32: example_fmult_3.mcnp.inp.txt

```
m123     Fm-257  1
awtab    Fm-257  254.88653438
mx123:n  Cf-252
c
sdef  par=sf
fmult Fm-257  watt=1.4 2.0  sfyield=5E11
```

Nuclear cross-section tables for transporting ^{246}Cf, ^{254}Cf, ^{257}Fm, and ^{252}No are not generally available. To model spontaneous fission from these nuclides, it is necessary to do the transport either with model physics or by substituting cross sections. Physics models are not recommended at low energies. To make a nuclide substitution, the AWTAB and MX cards must be used. The AWTAB card provides the atomic weight ratio for ^{257}Fm, which may not be available depending on the available data libraries installed. The MX123:n card in this example substitutes ^{252}Cf, for which there are neutron cross-section data, for the corresponding nuclide ^{257}Fm on the M123 material card.

5.7.10 TROPT: Transport Options

The TROPT card allows the user to modify the default options for modeling how particle interactions occur. Typically these options are useful for diagnosing the importance of certain physical processes, and for generating tabulated double-differential cross sections when using physics models. The PHYS card parameters for electrons and positrons are not set or modified by the TROPT card entries.

Data-card Form: `TROPT KEYWORD = value(s)`

`mcscat`	Controls multiple Coulomb scattering. If
	`mcscat = off`, multiple coulomb scattering is disabled; no angular deflection occurs.
	`mcscat = fnal1`, (DEFAULT).
	`mcscat = gaussian`
	`mcscat = fnal2`, then treats `eloss = strag1` as `eloss = csda` (recommended).
`eloss`	Controls slowing down energy losses. If
	`eloss = off`, no energy loss occurs during slowing down.
	`eloss = strag1`, CSDA is used with straggling (DEFAULT).
	`eloss = csda`, Energy loss modeled using only CSDA.
`nreact`	Controls nuclear reactions. If
	`nreact = off`, no nuclear reactions occur.
	`nreact = on`, nuclear reactions allowed (DEFAULT).
	`nreact = atten`, attenuation is turned on and absorption weighting occurs at collision.
	`nreact = remove`, the incident particle is killed.
`nescat`	Controls nuclear elastic scattering. This keyword has no effect if `nreact = off`. If
	`nescat = off`, acts as a delta-scatter for the elastic process in a transport calculation. For a `genxs` calculation, sets the elastic scattering cross section to zero.
	`nescat = on`, (DEFAULT).
`genxs`	Enables the generation of double-differential particle production cross sections and residual nucleus production cross sections from the high-energy nuclear interaction models. See §5.7.10.1. If the
	`genxs` keyword is absent, standard transport occurs.
	`genxs = filename` is present, but no `filename` is specified, read the edit input from a file named **`inxc`**. If `filename` is specified, read the edit input from a file named `filename`.

5.7.10.1 Application of the genxs Option

The genxs option allows the application of high-energy nuclear interaction models in a cross-section generation mode without particle transport. A source may be specified inside a medium; each history will consist only of the interaction of the source particle at the source energy with the components of the medium. The tallied outcome from the event consists of the energies and direction cosines of the secondary particles and the recoil nuclei. In typical applications, the material composition will be a single isotope; however, averaged results may be obtained for a natural multi-isotopic element or a complex composition. A genxs calculation is independent of the material density specification.

The genxs option requires two input files: the standard MCNP6 input file and an accompanying auxiliary **inxc** file. To invoke the genxs cross-section-generating option, specify genxs or genxs = filename on the TROPT card. The content and format of the edited output are determined by the content of the auxiliary input file associated with the genxs option. If genxs is specified on the TROPT card without a user-provided file name, by default the output tally edit information will be read from a file named **inxc**. If a file name is provided with the genxs keyword, the output tally edit information will be read from the user-specified filename. In either case, the absence of the required file will produce a fatal error. A description of the **inxc** file structure can be found in §D.9 with examples in §5.7.10.1.1 and §5.7.10.1.2.

To calculate inelastic secondary particle production only, turn off the elastic scattering by setting nescat = off on the TROPT card. Isotopic elastic scattering cross sections will be set to zero and the total cross section will equal the nonelastic cross section. All histories will sample the nonelastic interaction model. Note that this applies only to the genxs option; in a transport calculation, nescat = off implies a delta-scatter for the elastic process.

To examine only elastic scattering, use nreact = atten on the TROPT card. All histories will sample the elastic scattering model and produce results for the scattered projectile and the recoil nucleus.

In the output data for a multi-isotopic composition, quoted cross sections are a weighted average of the isotopic cross sections, weighted by the input atom fractions. Thus, the cross-section output represents average cross sections per atom in the composition. Results will reflect the variance introduced by sampling for the target isotope.

Energy- and angle-integrated results are provided as yield as well as cross section. The term "yield" might be better defined as "multiplicity". The nonelastic yield for a given particle type is the number of secondary particles of that type produced per nonelastic event. The elastic yield is per elastic event and is always unity. The *iyield* input option in the **inxc** file (see §D.9) allows single- and double-differential results to be provided as yield rather than cross section, with the above normalization.

5.7.10.1.1 Example 1

In the cross-section generation (genxs) calculation shown in Listing 5.33, 23.08-GeV protons impinge upon natural tungsten.

Listing 5.33: example_genxs_1.mcnp.inp.txt

```
Test problem: RECOIL2
1 1 -16.654 -1 2 -3
2 0         -4 (1:-2:3)
3 0          4

1 cz    4.0
2 pz   -1.0
```

```
3 pz    1.0
4 so   50.0

m1    74180 0.001300  74182 0.263000  74183 0.143000
      74184 0.306700  74186 0.286000
sdef  erg = 23080 par = h dir = 1 pos = 0 0 0 vec 0 0 1
imp:h   1 1 0
phys:h  23080
mode    h
print   40 110 95
nps     10000000
prdmp   2j -1
tropt   genxs nreact atten
```

Note that the `genxs` keyword of the `TROPT` card of the MCNP6 input file does not specify a user-supplied file name; therefore, MCNP6 expects an auxiliary input file named **`inxc`** (Listing 5.34) to be available.

Listing 5.34: example_genxs_1.inxc.inp.txt

```
Test problem: RECOIL2
5,1,1/
Elastic scattering edit
0,-200,1/
2.0/                          ! 200 bin boundaries, 2 deg to 0 deg
-1/                           ! elastic scattered projectile
Elastic scattering energy edit
125,0,1/
23079,23079.01/               ! 125 10-keV bins above 23.079 GeV
-1/                           ! elastic scattered projectile
Elastic recoil angle edit
0,102,1/
0.0,0.02/                     ! 101 boundaries mu=0 to 0.02&1.0
-2/                           ! elastic recoil nucleus
Elastic recoil energy edit
125,0,1/
0.01/                         ! 125 10-keV bins below 1.25 MeV
-2/                           ! elastic recoil nucleus
Elastic recoil momentum edit
150,0,1,,1/
5/                            ! 150 5-MeV/c bins below 750 MeV/c
-2/                           ! elastic recoil nucleus
```

Five cross-section edit cases plus the residual nucleus edit are specified. A **`mctal`** file is written for plotting. Because only elastic scattering occurs, all the cases are chosen to be single-differential cross sections only (i.e., *nerg* $= 0$ or *nang* $= 0$ in the **`incx`** input file):

1. $\mathrm{d}\sigma/\mathrm{d}\Omega$ for the projectile, binned by degrees;
2. $\mathrm{d}\sigma/\mathrm{d}E$ for the projectile, binned by energy;
3. $\mathrm{d}\sigma/\mathrm{d}\Omega$ for the recoil nuclei, binned by cosine;
4. $\mathrm{d}\sigma/\mathrm{d}E$ for the recoil nuclei, binned by energy; and
5. $\mathrm{d}\sigma/\mathrm{d}p$ for the recoil nucleus, binned by momentum.

Listing 5.35: Resultant MCNP6 Output File Excerpt

```
1 Distribution of residual nuclei:

                         Cross Section (b)
     Z = 74  all A     1.11594E+00 0.0000
             A = 180   1.40752E-03 0.0089
             A = 182   2.91615E-01 0.0005
             A = 183   1.59094E-01 0.0008
             A = 184   3.42476E-01 0.0005
             A = 186   3.21350E-01 0.0005

Summary by charge number:

  Z    Cross Section (b)     Mean Recoil (MeV)
 74   1.11594E+00 0.0000    1.30329E-02 0.0008

Summary by mass number:

  A     Cross Section (b)    Mean Recoil (MeV)
180    1.40752E-03 0.0089   1.39919E-02 0.0233
182    2.91615E-01 0.0005   1.31164E-02 0.0016
183    1.59094E-01 0.0008   1.30618E-02 0.0021
184    3.42476E-01 0.0005   1.30351E-02 0.0015
186    3.21350E-01 0.0005   1.29360E-02 0.0015

Mean weight of residual nuclei per event    3.97430E-01 0.0000
Number of residual nuclei outside of table range:      0
```

Because the computation is for only elastic scattering from a composition (natural element), the cross section shown for production of a particular residual nucleus is just $f_i\sigma_i^e$ per atom in the element and the cross section for any residual with charge number $Z = 74$ is

$$\sum_{i=1,\ldots,5} f_i\sigma_i^e,$$

i.e., the average elastic cross section per atom in the composition. Because the attenuation weighting option (`nreact = atten`) was used and every event is an elastic event, the quantity "mean weight of residual nuclei per event" equals the ratio of the mean elastic cross section to the mean total cross section for the element.

5.7.10.1.2 Example 2

In this example, the yields (i.e., production cross sections) of products from a thin ^{238}U target bombarded by 1-GeV protons are calculated. The `SDEF` card of the MCNP input file (Listing 5.36) defines a 1000-MeV proton beam source pointed in the direction of the z axis.

Listing 5.36: example_genxs_2.mcnp.inp.txt

```
MCNP6 test: p + U238 by CEM03.03 at 1 GeV, nevtype=66
1 1 1.0   -1  2  -3
2 0       -4  (1:-2:3)
3 0        4

1  cz    4.0
2  pz   -1.0
3  pz    1.0
4  so   50.0
```

```
sdef    erg=1000  par=H  dir=1  pos=0 0 0  vec 0 0 1
imp:h   1 1 0
phys:h  1000
m1      92238 1.0
mode    h
LCA     8j 1     $ use CEM03.03
tropt   genxs inxc01  nreact on  nescat off
print   40 110 95
nps     1000000
prdmp   2j -1
```

This beam bombards ^{238}U, which fills a cylinder with a 4-cm radius oriented on the z axis from $z = -1$ to $z = 1$ cm. The provided LCA card parameters select the CEM03.03 event generator for this calculation. The card indicates a genxs problem with an auxiliary input file named **inxc01** (Listing 5.37).

Listing 5.37: example_genxs_2.inxc.inp.txt

```
MCNP6 test: p + U238 at 1 GeV for TR applications
1 1 1 /
Cross Section Edit
56 0 9 /
5. 10. 15. 20. 25. 30. 35. 40. 45. 50. 55. 60. 65. 70. 75. 80.
85. 90. 95. 100. 120. /
1 5 6 7 8 21 22 23 24 /
```

We will calculate only inelastic secondary particle production (nreact = on) and we turn off the elastic scattering (nescat = off).

The input parameters of the **inxc** file indicate that one double-differential cross-section edit is requested, the results are to be written to the **mctal** file, and a residual nuclei edit is desired. The fourth card of the input file specifies angle-integrated energy spectra with 56 energy bin boundaries for nine particle types. The 56 energy bin boundaries are defined on the 5th card (on multiple lines) using a combination of user-provided values (the first 21 values) and code-generated values (the final 25 values). The nine particle types to tally are defined on the final card of the **incx** input file using flag values to specify neutron, proton, π^+, π^-, π^0, deuteron, triton, ^{3}He, and ^{4}He.

Listing 5.38: Resulting MCNP6 Output File Excerpt

```
1 Distribution of residual nuclei:

                        Cross Section (b)
     Z =  1  all A     3.72774E+00 0.0010
              A =   2   2.64148E+00 0.0011
              A =   3   1.08626E+00 0.0016
     Z =  2  all A     1.56336E+00 0.0015
              A =   3   2.09100E-01 0.0032
              A =   4   1.34273E+00 0.0016
              A =   6   1.12506E-02 0.0135
              A =   8   2.76052E-04 0.0861
     Z =  3  all A     2.60266E-02 0.0090
              A =   6   8.27338E-03 0.0158
              A =   7   1.33507E-02 0.0125
              A =   8   3.58050E-03 0.0239
              A =   9   8.22021E-04 0.0499
     Z =  4  all A     8.39607E-03 0.0157
              A =   7   9.16083E-04 0.0473
              A =   9   3.31262E-03 0.0249
```

```
            A =  10   3.75840E-03 0.0233
            A =  11   3.76249E-04 0.0737
            A =  12   3.27173E-05 0.2500
    Z =  5  all A     3.64593E-03 0.0238
            A =   8   1.02241E-05 0.4472
            A =  10   1.14919E-03 0.0422
            A =  11   1.39253E-03 0.0384
            A =  12   9.44711E-04 0.0465
            A =  13   1.49273E-04 0.1170
    Z =  6  all A     2.80551E-03 0.0270
            A =  10   1.63586E-05 0.3536
            A =  11   1.22690E-04 0.1291
            A =  13   8.40425E-04 0.0493
            A =  14   8.26111E-04 0.0497
            A =  15   2.18797E-04 0.0967
            A =  16   2.86276E-05 0.2673
    Z =  7  all A     1.13897E-03 0.0424
            A =  12   2.04483E-06 1.0000
            A =  13   8.17932E-06 0.5000
            A =  14   2.37200E-04 0.0928
            A =  15   5.25521E-04 0.0624
            A =  16   1.32914E-04 0.1240
            A =  17   2.33110E-04 0.0937
.
.
.
    Z = 35  all A     7.15465E-02 0.0053
            A =  73   1.43138E-05 0.3780
            A =  74   3.29217E-04 0.0788
            A =  75   1.24530E-03 0.0405
            A =  76   2.92206E-03 0.0264
            A =  77   5.38608E-03 0.0195
            A =  78   6.02611E-03 0.0184
            A =  79   9.08926E-03 0.0150
            A =  80   7.63744E-03 0.0163
            A =  81   8.99316E-03 0.0150
            A =  82   6.54959E-03 0.0176
            A =  83   7.09147E-03 0.0170
            A =  84   4.52930E-03 0.0212
            A =  85   4.75627E-03 0.0207
            A =  86   2.40881E-03 0.0291
            A =  87   2.05096E-03 0.0316
            A =  88   1.10421E-03 0.0430
            A =  89   7.66811E-04 0.0516
            A =  90   3.35352E-04 0.0781
            A =  91   2.04483E-04 0.1000
            A =  92   6.33897E-05 0.1796
            A =  93   2.45379E-05 0.2887
            A =  94   1.63586E-05 0.3536
            A =  95   2.04483E-06 1.0000
    Z = 36  all A     9.17208E-02 0.0046
```

Of particular interest is the production of ^{87}Br and ^{88}Br, primary delayed neutron emitters with relatively long half-lives of 55.60 and 16.29 s, respectively. From this portion of the output, we see that the cross section for the production of ^{87}Br is equal to 2.05096×10^{-03} b ($\pm 3.16\%$) and that of ^{88}Br is 1.10421×10^{-03} b ($\pm 4.30\%$). We also see in the output file four isotopes of lithium, including ^{9}Li, and cross sections for the production of ^{17}N and ^{16}C. These three isotopes are also important delayed neutron emitters, although their half-lives are only 0.178, 4.173, and 0.747 s, respectively.

5.7.11 UNC: Uncollided Secondaries

The historical definition of an uncollided particle in MCNP6 is any particle that has not undergone a collision since its creation, whether as a source particle or as a secondary particle. This definition, in which secondary particles are created as uncollided particles, makes separation of the contribution to a tally from the direct source and contribution from secondary particles difficult. Identification of the uncollided components is particularly useful for users who employ track-length tallies in radiography applications instead of next-event estimators.

The UNC card allows the user to control if secondaries are born as uncollided or collided particles. When created as collided particles, secondaries inherit the number of collisions of their parent particle. If a particle inherits the number of collisions of the parent, then the number of collisions is always greater than or equal to one.

Cell-card Form: UNC:$\mathscr{P}$ u
or
Data-card Form: UNC:$\mathscr{P}$ $u1$ $u2$ ... uJ

$\mathscr{P}$	Particle designator.	
u	If	
	$u = 0$,	then secondaries are considered to be collided for the cell.
	$u = 1$,	then secondaries are considered uncollided for the cell (DEFAULT).
uj	Number of entries equals number of cells in the problem, J. If	
	$uj = 0$,	then secondaries are considered to be collided for the cell.
	$uj = 1$,	then secondaries are considered uncollided for the cell (DEFAULT).

Default: $uj = 1$, secondaries are considered uncollided for cell j.

Use: Optional. Useful for separating the contribution resulting from uncollided source particles from that of secondaries that do not collide after their creation.

5.7.12 Magnetic Field Tracking

MCNP6 provides two methods to simulate magnetic field effects on charged particles [280, 281]. The first method utilizes transfer maps produced by the beam dynamics simulation and analysis code COSY INFINITY [282]. This method is fast and accurate; however, its use is limited to void cells only (i.e., in a vacuum) and to ensembles of particles with a fairly small energy spread. The second method, magnetic field particle ray tracing, is based on an algorithm adopted from the MARS [283–285] transport code. This method can be applied to both void and material cells and is valid over a very large range of particle energies. In addition, for the magnetic field ray tracing method, MCNP6 includes an option that simulates third-order aberrations for quadrupole magnets caused by fringe-field effects by providing edge kicks for particles entering and exiting the magnet faces. This latter feature is especially important for proper particle transport through proton radiography beam lines and magnetic lenses.

5.7.12.1 Magnetic Field Transfer Map

COSY INFINITY is a beam optics code that utilizes numerical integration and differential algebraic techniques to generate transfer maps based on a Taylor series expansion of a particle's canonical variables [286]. These transfer maps represent the functional relation between the phase-space coordinates of a particle that has passed through a region with a magnetic field and its phase-space coordinates before entering the field region. In the transfer map approach to particle transport, the actual trajectories that the particles follow through the field region do not appear explicitly; in applying precomputed maps, charged particles are transported from an initial location to a final location in one step by applying the transfer maps to the initial phase space coordinates.

Although the COSY map method provides a fast and accurate method for transporting charged particles in magnetic fields, the transfer map method has several limitations:

1. Map methods can only be used in void regions.
2. The COSY maps are limited to only one particle type.
3. The Taylor expansions used in applying the maps have a finite volume of convergence in phase space. The convergence volume has a very complicated shape in five dimensions $(x, y, \mathrm{d}x, \mathrm{d}y, p)$, requiring that the shape of the phase-space volume and the order of the Taylor series needed in order to get a given accuracy in final particle position is not easily predicted in practice and can be checked only by particle tracking.

   For example, a map to fifth order in energy deviation might be applied with good accuracy to particles with energies within 10% of the reference energy, but not to those with a 50% deviation. In other words, COSY maps are specific to particle momentum; therefore, a particle with significantly different energy or mass than what was used to create the map will not be transported correctly.

5.7.12.1.1 COSYP: Magnetic Field Transfer Map Parameters

The `COSYP` card is used to define the parameters associated with external COSY map files and how they may be generally applied within a problem. No information about the magnetic fields is written to the output file.

Data-card Form: `COSYP prefix` *`axsh axsv emap1 emap2 ... emapK`*

`prefix`	The COSY map file prefix number is required. The map files must reside in the current working directory.
`axsh`	Horizontal axis orientation. If
	`axsh` $= 1$, the x axis is the horizontal axis (DEFAULT).
	`axsh` $= 2$, the y axis is the horizontal axis.
	`axsh` $= 3$, the z axis is the horizontal axis.
`axsv`	Vertical axis orientation. If
	`axsv` $= 1$, the x axis is the vertical axis.
	`axsv` $= 2$, the y axis is the vertical axis (DEFAULT).
	`axsv` $= 3$, the z axis is the vertical axis.
`emapk`	Set *`emapk`* $= e_k$, where e_k is the operating beam energy of the kth map assigned (DEFAULT is the energy of the kth COSY map).

Use: Optional. Use with COSY maps.

5.7.12.1.2 COSY: Magnetic Field Assignments

The COSY card is used with the COSYP card to assign the COSY maps to specific cells in the problem geometry.

Cell-card Form: `COSY = m`
or
Data-card Form: `COSY m1 m2 ... mJ`

`m`	Assign COSY map `m` to the current cell.
`mj`	Set cell `j` to COSY map `mj`.

Use: Use with COSY maps.

Default: No map is assigned to the cell.

5.7.12.1.3 COSYP and COSY Example 1

```
cosyp   57 2 1 23070 11r
cosy    3j 1 j 2 j 3 j 4 10j 5 j 5 j 6 j 6
```

In this example, the COSYP card defines the COSY map file parameters. Each COSY map file name is prefixed with 57. The horizontal axis is the y axis, and the vertical axis is the x axis. The operating energy for all twelve maps assigned is 23,070 MeV. Field maps are assigned to twelve cells specified on the COSY card. Table 5.12 lists the map assignments. The COSY map files **571**, **572**, **573**, **574**, **575**, and **576** must be in the working directory.

5.7.12.2 Magnetic Field Particle Ray Tracing

To overcome the limitations of transfer maps, MCNP6 has implemented direct magnetic field tracking utilizing numerical integration methods. These routines were adopted from the MARS high-energy particle transport code. Tracking in a void and material is performed by a higher-order numerical integration algorithm, with a maximum step size controlled by the user. Within a step, the trajectory is approximated by a segment of the helical trajectory corresponding to a constant field equal to the field at the midpoint of the step, i.e., the field variation within the step is neglected. A solution of a 3-D equation of trajectory in such a field provides the new direction cosines and new particle coordinates at the end of the step. With appropriate parameters, this algorithm provides extremely high accuracy of tracking.

Table 5.12: Example COSY Map Assignment

Map Number	COSY Map File Name	Cell Numbers
1	**571**	4
2	**572**	6
3	**573**	8
4	**574**	10
5	**575**	21 and 23
6	**576**	25 and 27

For quadrupole fields, MCNP6 includes a model to include the effect of the magnet fringe fields. This can be approximated by applying hard-edge kicks to the particle as it enters and leaves the magnetic field cell. An option for edge kicks has been implemented for the quadrupole magnetic field model. For a particle traveling along the z axis, the following equations describe the position and momentum jumps applied to a particle as it enters the upstream fringe field of a quadrupole [287]:

$$\delta x = \frac{Gp}{q}\left[\frac{x^3}{12} + \frac{xy^2}{4}\right], \tag{5.19}$$

$$\delta t_x = \frac{Gp}{q}\left[\frac{xy}{2}t_y - \frac{x^2+y^2}{4}t_x\right], \tag{5.20}$$

$$\delta y = -\frac{Gp}{q}\left[\frac{y^3}{12} + \frac{x^2y}{4}\right], \tag{5.21}$$

$$\delta t_y = -\frac{Gp}{q}\left[\frac{xy}{2}t_x - \frac{x^2+y^2}{4}\right]t_y. \tag{5.22}$$

In these equations, t_x and t_y are the direction cosines of the momentum vector. The quantity G is the quadrupole gradient (in T/m) and p/q is the particle rigidity (in T-m). In order to conserve energy, t_z is also recalculated using the formula

$$t_z = \sqrt{1 - t_x^2 - t_y^2}. \tag{5.23}$$

For particles passing through the downstream fringe field of a quadrupole, the equations are the same, except that Gp/q is replaced everywhere by $-Gp/q$.

Summary of known limitations of the magnetic field particle ray tracing method:

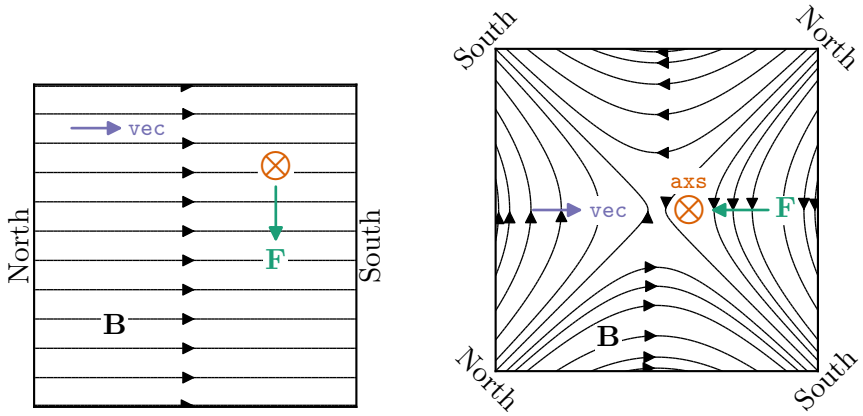
1. A particle can get lost, especially for complicated geometries and lattice cells.
2. In rare cases, MCNP6 could hang in an infinite loop.

5.7.12.2.1 BFLD: Magnetic Field Definition

The magnetic field tracking option is accessed by use of the magnetic field definition card, `BFLD`. The MCNP code can model dipole and quadrupole fields such as those shown in Fig. 5.6, where the quadrupole fields can also include fringe-field edge kicks.

Data-card Form: `BFLDn type KEYWORD = value(s)`

`n`	The magnetic field identification number.
`type`	The magnetic field polarity is required. If
`type = const`,	magnetic field is a dipole field.
`type = quad`,	magnetic field is a quadrupole field.
`type = quadff`,	magnetic field is a quadrupole field with fringe-field edge kicks.
`field` $= f$	Required for each of the `type`s. If
`type = const`,	$f = B$, the magnetic field strength (Tesla).
`type = quad` or `type = quadff`,	$f = B/l$, the magnetic field gradient (Tesla/cm).



(a) Dipole (b) Quadrupole

Figure 5.6: Supported magnetic field types with the associated `vec` and `axs` vectors necessary to represent the field shown. The symbol ⊗ represents a vector (i.e., a particle) traveling into the page, **B** the magnetic field, and **F** the force on the particle.

`vec` $= (u_f, v_f, w_f)$	Optional for each of the `types` (DEFAULT: `vec` $= 1, 0, 0$). If
	`type` $=$ `const`, (u_f, v_f, w_f) is the direction of the magnetic field.
	`type` $=$ `quad` or `type` $=$ `quadff`, (u_f, v_f, w_f) is the direction of a focusing quadrupole.
`axs` $= (u_q, v_q, w_q)$,	The direction cosines of the quadrupole beam axis, which do not need to be normalized. Only applies to quadrupole fields (DEFAULT: `axs` $= 0, 0, 1$).
`refpnt` $= (x, y, z)$,	A point anywhere on the quadrupole beam axis. Only applies to quadrupole fields (DEFAULT: `refpnt` $= 0, 0, 0$).
`mxdeflc` $= a$,	Maximum deflection angle per step size (mrad) (DEFAULT: `mxdeflc` $=$ `10`).
`maxstep` $= ss$,	Maximum step size (cm) (DEFAULT: `maxstep` $=$ `100`).
`ffedges` $= s_1\ s_2\ \ldots\ s_J$,	List of surface numbers that fringe-field edge kicks are to be applied. Only applies to quadrupole fields with fringe-field kicks (`type` $=$ `quadff`).

Use: Optional. If the `type` parameter of the `BFLD` card is not provided, a fatal error occurs.

5.7.12.2.2 BFLCL: Magnetic Field Cell Assignment

The `BFLCL` card is used with the `BFLD` card(s) to assign the magnetic fields to specific cells in the problem geometry.

Cell-card Form: `BFLCL = m`
or
Data-card Form: `BFLCL m1 m2 ... mJ`

`m`	Assign magnetic field `m` to the current cell.
`mj`	Set cell `j` to magnetic field `mj`.

Default: No magnetic field is assigned to the cell.

5.7.12.2.3 BFLD and BFLCL Example 1

```
bfld1   CONST  FIELD .03  VEC 0 1 0
bflcl   2j  1
```

A constant magnetic field of strength 0.03 Tesla is applied to cell 3. The field is in the $+y$ direction.

5.7.12.2.4 BFLD and BFLCL Example 2

```
bfld2  QUADFF  FIELD 0.195  FFEDGES = 31 2i 34
bflcl  31j 2 0 2
```

A quadruple magnet field of gradient 0.195 T/cm is assigned to cells 32 and 34. Fringe-field edge kicks are applied to surfaces 31, 32, 33, and 34.

5.7.12.2.5 BFLD and BFLCL Example 3

```
bfld3   QUAD  FIELD 0.116
        VEC 0.5 0.5 0.707
        AXS 0.85 -0.14 -0.5
        REFPNT 40 30 100
        MXDEFLC 10 MAXSTEP=1
bflcl   101j 3 0 3 7j 3 0 3
```

A quadrupole magnetic field of gradient 0.116 T/cm is assigned to cells 102, 104, 112, and 114. The axis of the quadrupole is along the vector $(0.85, -0.14, -0.5)$, and the focusing direction is along the vector $(0.5, 0.5, 0.707)$. The maximum step size is 1 cm, and the maximum angular deflection is 10 mrads.

5.7.13 FIELD: Gravitational Field

The FIELD card was historically an undocumented feature in earlier MCNP releases that allowed the user to model planetary gravitational effects on neutrons, which results in their orbiting the planet. The theory and equations used for this feature are documented in a paper by Feldman, et al. [288]. This can be an important effect for low-energy (e.g., thermal) neutron detectors aboard orbiting spacecraft.

Caution

The gravitational field capability is not tested, has known issues, and is being considered for deprecation for future versions of the MCNP6 code if there is no user interest in this capability. People interested in this capability should send an email to mcnp_help@lanl.gov.

Data-card Form: `FIELD KEYWORD=value(s)`

`gcut` $= e$	Gravitational binding energy (eV) threshold. Below this energy, the gravitational field treatment is applied (DEFAULT: `gcut` $= 0.653$).
`gpar` $= p$	Particle type (limited to neutrons) (DEFAULT: `gpar` $= 1$).
`grad` $= r$	Radius of planetary object (km) (DEFAULT: `grad` $= 6,371$).
`gsur` $= s$	Required list of one or more surface numbers. When the `gpar` particle crosses the surface, it is assumed to orbit the planetary object and reenter the `gsur` surface at a later time, unless terminated via decay.

Default: Applied to neutrons (`gpar` $= 1$) with respect to the Earth's radius (`grad` $= 6,371$), and below Earth's gravitational binding energy (`gcut` $= 0.653$). The list of one or more surfaces (`gsur`) is required.

Use: Optional.

Details:

1. The orbiting particle reenters at the same location as they exit, thus this is really a partial reflection at surface `gsur`. Even though the particle would reenter at a different location, it is assumed that there is a uniform distribution of such reentering particles.

5.7.13.1 Example

The input card shown in Listing 5.39 contains the default keyword-value options applied to neutrons as they cross surface `gsur` $= 2$.

Listing 5.39: example_field.mcnp.inp.txt

```
field gcut=0.653 gpar=1 grad=6371.0 gsur=2
```

5.8 Source Specification-focused Data Cards

Every MCNP problem has one of four sources:

1. General source (SDEF card),
2. Surface source (SSR card),
3. Criticality source (KCODE card), or
4. User-supplied source.

All can use source distribution functions, specified on SI, SP, SB, and DS cards.

5.8.1 SDEF: General Source Definition

The specification of a source variable has one of the following three forms:

1. A scalar or vector, in which a single, explicit value is given for the specified variable (e.g., CEL = 1 or POS = 0 0 6).
2. A distribution number, *n*, prefixed by a D, in which the specified source variable may have multiple values that will be sampled from distribution SI*n*. For example, CEL = D1 indicates that multiple cell numbers will appear on the SI1 card and will be sampled using probabilities entered on the associated SP1 card.
3. The name of another variable prefixed by an F, followed by a distribution number prefixed by a D. (For example., POS = FCEL = D1 indicates that the position specification will depend on the cell(s) specified on the SI1 card.) Only one level of dependence is allowed. Each distribution may be used for only one source variable. None of the position-related keywords (i.e., CEL, SUR, RAD, AXS, EXT, X, Y, Z, and CCC) can be a dependent distribution of POS.

The above scheme translates into three levels of source description. The first level exists when a source variable has an explicit or default value (for example, a single energy) or a default distribution (for example, an isotropic angular distribution). The second level occurs when a source variable is given by a probability distribution. This level requires the SI and/or SP cards. The third level occurs when a variable depends on another variable. This level requires the DS card.

The MCNP code samples the source variables in an order set up according to the needs of the particular problem. Each dependent variable must be sampled after the variable it depends on has been sampled. If the value of one variable influences the default value of another variable or the way it is sampled, as SUR influences DIR, they need to be sampled in the right order. The scheme used in the MCNP code to set up the order of sampling is complicated and may not always work. If it fails, a message will be printed. The fix in such instances may be to use explicit values or distributions instead of depending on defaults.

The source variables SUR, VEC, NRM, and DIR are used to determine the initial direction of source-particle flight. The direction of flight is sampled with respect to the reference vector VEC, which can itself be sampled from a distribution. The polar angle is the sampled value of the variable DIR. The azimuthal angle is sampled uniformly in the range from 0° to 360°. If VEC and DIR are not specified for a volume distribution of position (SUR = 0), an isotropic distribution of direction is produced by default. If VEC is not specified for a distribution on a surface (SUR = 0), the vector normal to the surface, with the sign determined by the

sign of NRM, is used by default. If DIR is not specified for a distribution on a surface, the cosine distribution ($p(\texttt{DIR}) = 2 \times \texttt{DIR}$, $0 < \texttt{DIR} < 1$) is used by default. A biased distribution of DIR can be used to make more source particles start in a direction toward the tallying regions of the geometry. The exponential distribution function (-31, Table 5.15) is usually most appropriate for this.

The source variables SUR, POS, RAD, EXT, AXS, X, Y, Z, and CCC are used in various combinations to determine the coordinates (x, y, z) of the starting positions of the source particles. With them you can specify three different kinds of volume distributions and three different kinds of distributions on surfaces. Degenerate versions of those distributions provide line and point sources. More elaborate distributions can be approximated by combining several simple distributions, using the S option of the SI and DS cards.

A description of each SDEF keyword appears below. Following the card description and its notes are more detailed discussions of volume and surface source specification. Examples of the general source follow discussion of the SI, SP, SB, and DS cards.

Data-card Form: SDEF KEYWORD=value(s)

CEL	Cell number. DEFAULT: Determined from the position of the particle, and possibly the direction of the flight of the particle if the position is on a surface of a cell.
SUR	Surface number (1). DEFAULT: SUR = 0, which indicates a cell (volume) source. Always required when source points lie on the boundary (surface) of a cell.
ERG	Kinetic energy (MeV). (16) DEFAULT: ERG = 14
TME	Time (shakes) (2). DEFAULT: TME = 0
DIR	μ, the cosine of the angle between VEC and the particle's direction of flight. Azimuthal angle is always sampled uniformly in 0° to 360° (3). Defaults are volume source: μ is sampled uniformly in -1 to 1, i.e., the source is isotropic. surface source: $p(\mu) = 2\mu$ in 0 to 1 , i.e., cosine distribution.)
VEC	Reference vector for DIR in vector notation. DEFAULT for volume source: Required unless source is isotropic. DEFAULT for surface source: Vector normal to the surface with sign determined by NRM.
NRM	Sign of the surface normal. DEFAULT: NRM = +1
POS	Reference point for position sampling in vector notation. DEFAULT: POS = 0,0,0
RAD	Radial distance of the position from POS or AXS. DEFAULT: RAD = 0
EXT	For a volume source is the distance from POS along AXS. For a surface source is the cosine of angle from AXS. DEFAULT: EXT = 0
AXS	Reference vector for EXT and RAD in vector notation. DEFAULT: No direction.
X	x coordinate of position. DEFAULT: X = 0
Y	y coordinate of position. DEFAULT: Y = 0
Z	z coordinate of position. DEFAULT: Z = 0

CCC	Cookie-cutter cell number. (4, 5) DEFAULT: no cookie-cutter cell.
ARA	Area of surface. Required only for direct contributions to point detectors from plane surface source. DEFAULT: none
WGT	Particle weight (input as explicit value only). DEFAULT: WGT = 1
TR	Source particle transformation number (TR = *n*) or distribution of transformations (TR = D*n*). Corresponding TR card(s) is required. (6, 7) DEFAULT: none.
EFF	Rejection efficiency criterion for position sampling (input as explicit value only). (8) DEFAULT: EFF = 0.01
PAR	Source particle type(s) by symbol or number (e.g., PAR = H or PAR = 9). (15) For a complete list of particle types, see Table 4.3. Use a distribution for sampling multiple particle types. To specify a particular heavy ion as a source particle, set PAR to a target identifier [§1.2.2]. All formats supported. Metastable states are ignored. To sample cosmic particles, if

PAR = [-]CR	the source is a combination of all cosmic particles
PAR = [-]CH or PAR = C1001	the source contains cosmic protons only
PAR = [-]CA or PAR = C2004	the source contains cosmic alphas only
PAR = [-]C7014	the source contains cosmic nitrogen only
PAR = [-]C14028	the source contains cosmic silicon only
PAR = [-]C26056	the source contains cosmic iron only

When the negative sign is omitted from these options, the SDEF WGT keyword (i.e., source normalization) is set to the integral 2π flux obtained from the Castagnoli and Lal analytic equation ([289], as corrected by [290]). If the cosmic particle designator is preceded by a negative sign, then the particle weight normalization only considers the SDEF WGT information provided by the user.

To sample background particles (9), if

PAR = [-]BG	the background is a combination of all background particles (currently limited to neutrons and gammas)
PAR = [-]BN	the background contains neutrons only
PAR = [-]BP	the background contains gammas only

If the negative sign is omitted, the SDEF WGT keyword (i.e., source normalization) is multiplied by values contained in the **BACKGROUND.dat** file. If the negative sign is included, then the source normalization is taken only from the SDEF WGT keyword.

To sample spontaneous fission [§5.8.1.7] if

PAR = SF	normalize summary and tally information by the number of spontaneous-fission neutrons.

PAR = -SF	normalize summary and tally information by the number of histories (generally, the number of spontaneous fissions).

To sample spontaneous neutrons:

PAR = SN	decay neutrons will be created based on the relative activities of the unstable isotopes in the material(s) located at the source location(s).

To sample spontaneous photons:

PAR = SP	decay gammas will be created based on the relative activities of the unstable isotopes in the material(s) located at the source location(s).

To sample spontaneous betas:

PAR = SB	decay betas will be created based on the relative activities of the unstable isotopes in the material(s) located at the source location(s).

To sample spontaneous positrons:

PAR = ST	decay betas will be created based on the relative activities of the unstable isotopes in the material(s) located at the source location(s).

To sample spontaneous alphas:

PAR = SA	decay alphas will be created based on the relative activities of the unstable isotopes in the material(s) located at the source location(s).

To sample all-particle spontaneous decay:

PAR = SD	all decay particles (SN, SP, SB, SA, ST) will be created based on the relative activities of the unstable isotopes in the material(s) located at the source location(s). Decay particle types that are missing from the MODE card will be omitted (with a related warning message)

To specify the decay particles from a particular heavy ion as the source, set PAR to a target identifier corresponding to the ion, and set the energy of the ion to zero (ERG = 0.0). Requires that heavy ions (#) be specified on the MODE card. DEFAULT: If no MODE card, PAR = N. DEFAULT: If MODE card in INP file, lowest **IPT** number or symbol represented on MODE card.

DAT *m d y*

Date to use for cosmic-ray (PAR = CR, CH, CA) and background (PAR = BG, BN, BP) sources ((10)):

m	An integer value representing the month of the year ($1 \leq m \leq 12$)
d	An integer value representing the day of the month ($1 \leq d \leq 31$)
y	A 4-digit integer representing the year

`LOC` *lat lng alt*	Location of cosmic particle source ((11)):
	lat — latitude ($-90 \leq$ *lat* ≤ 90, relative to equator; negative values are south of the equator and positive values are north of the equator)
	lng — longitude ($-180 \leq$ *lat* ≤ 180, relative to Greenwich, UK; negative values are west longitude and positive values are east longitude)
	alt — altitude in km of cosmic particles when `PAR` = `CR`, `CH`, `CA` (DEFAULT: *alt* = 65.0 km), or elevation in km of the background source when `PAR` = `BG`, `BN`, `BP` (no default).
`BEM` *exn eyn bml*	Beam emittance parameters ((12)):
	exn — normalized beam emittance parameter, ε_{nx}, for phase-plane coordinates x, x'(π-cm-radians)
	eyn — normalized beam emittance parameter, ε_{ny}, for phase-plane coordinates y, y' (π-cm-radians)
	bml — distance from the aperture to the spot, L (cm)
	DEFAULT: none
`BAP` *ba1 ba2 u*	Beam aperture parameters ((13)):
	ba1 — beam aperture half-width in the x transverse direction, x_0 (cm)
	ba2 — beam aperture half-width in the y transverse direction, y_0 (cm)
	u — unused, but must be set to an arbitrary value
	DEFAULT: none

Default: Isotropic point source at position = $(0, 0, 0)$, time = 0, energy = 14 MeV, and particle weight = 1.

Use: Required for problems using the general source. Optional for problems using the criticality source. Reminder: an equals sign (=) following a keyword is optional.

Details:

(1) If the source location is on any surface (including "extended" surfaces of macrobodies) used to describe the cell that contains that source, the `SUR` keyword must be used. A source can lie on an extended surface used to describe any other cell of the problem.

(2) Emitted source decay gammas are assumed to arise from instantaneous activity of a large pool of decaying isotopes; time behavior is defined by the `TME` keyword. If an isotope emits multiple gamma lines, the emissions will not necessarily be correlated. Isotopes with half-lives longer than 10^{18} seconds ($\sim 3.17 \times 10^{10}$ years) are treated as stable. When the decay gammas of a heavy ion are specified as the source, `PRINT` Table 110 of the output file will list the sampled heavy-ion isotopes but not the created gamma lines.

3 Discrete values of DIR are allowed. DIR = 1 gives a mono-directional source in the direction of VEC. This is sometimes useful as an approximation to an actual source that is at a large distance from the geometry of the problem. In most cases discrete values of DIR will prevent direct contributions to point detectors from being scored. The direct contribution will be scored only if the source is on a plane surface, is sampled uniformly in area within a circle (using RAD sampled from SP -21 1), VEC is perpendicular to the surface (the default), and DIR = 1. A cookie-cutter cell is allowed and a value of ARA is necessary. Discrete values of DIR with DXTRAN are generally wrong because $p(\mu) = 0.5$ is assumed.

4 Cookie-cutter rejection is available for both cell and surface sources. If CCC is present, the sampled position is accepted if it is within cell CCC and is resampled if it is not. It is recommended that cookie-cutter cells be bounded by surfaces used for no other purpose in the problem and that the cookie-cutter cell cards appear at the end of the list of cell cards. Also, keep the cookie-cutter cell as simple as possible. For example, for a surface source, the intersection of the cookie-cutter cell with the source surface is what matters. For a plane surface source, an infinitely long cell of uniform cross section bounded by planes and cylinders is usually adequate.

5 **Caution**

The combination of either CEL or CCC rejection with biased sampling of the position is nearly always an unfair game. If the user employs this combination, they must ensure that the game is fair; the MCNP code cannot detect this error. To accomplish this, the source weight needs to be multiplied by the ratio of the biased acceptance probability to the unbiased acceptance probability (discussed in [291]).

6 A general transformation of the generated source may be specified with a single transformation TR = *n* or with a distribution of transformations TR = D*m*. In either case, all SDEF parameters relating to particle position or direction are interpreted as being in an auxiliary coordinate system in which the source specification is simpler. A general transformation is applied to a source particle after its coordinates and direction cosines have been determined in the auxiliary coordinate system. Particle coordinates are modified by both rotation and translation, while direction cosines are modified only by rotation. The source after transformation is treated as a volume source (i.e., surface number not defined); the cell for the source particle is determined after transformation. (SUR and CEL are used only in the initial generation procedure). To avoid the possibility of lost particles, do not place the transformed source exactly on a surface of the physical geometry. With the form TR = *n*, a transformation card TR*n* must be specified. With the form TR = D*m*, in addition to the TR cards, the user must provide SI*m*, SP*m*, and possibly SB*m* cards. If a distribution of transformations is specified, the option parameter on the corresponding SI*m* card must be L. The option parameter on the SP*m* and SB*m* cards may be blank, D, or C.

7 Sources may be translated to different locations with the TR option. For example, the source transformation capability allows the user to rotate the direction of an accelerator beam or move the entire beam of particles in space. In addition, this capability is useful for setting up the source as an accelerator beam and then using the translation as a distribution to repeat the accelerator source at different locations and orientations. The TR option can be dependent on other source variables. For example, the particle type can depend on the translated source location:

```
SDEF    CEL=FTR=D3   PAR=FTR=D1   TR=D2
```

or the translated source location can be a dependent distribution function of cell:

```
SDEF    CEL=D2       TR=FCEL=D5
```

(8) The efficiency criterion EFF applies to both CCC and CEL rejection. If in any source cell or cookie-cutter cell the acceptance rate is too low (the default value of EFF is 0.01), the problem is terminated for inefficiency. To increase efficiency, the user is encouraged to revise the source description. If a source efficiency lower than 0.01 is unavoidable, specify a lower value for EFF.

(9) The BG, BN, and BP options require that the user:

(a) properly normalize the source in a spherical volume (WGT = sphere surface area/3.0), cylindrical volume (WGT = cylindrical surface area/3.4), cube volume (WGT = cube surface area/3.7), spherical surface (WGT = πr^2), or some other enclosed surface (WGT set to a central cell tally that has unit flux);

(b) use the appropriate SDEF keywords to specify an isotropic uniform spatial distribution within these volumes or a cosine-weighted uniform distribution on any enclosing surface;

(c) ensure that the background source volume is large compared to the geometry of interest (i.e., a radius or diameter that is 10 times that of the interior geometry); and

(d) ensure that the **BACKGROUND.dat** file is in the local directory or in the **DATAPATH** directory. When the "-" sign is omitted from these options, the WGT normalization will be further modified by the neutron and/or gamma flux normalization provided in the **BACKGROUND.dat** file, as well as being multiplied by the neutron/cosmic-photon elevation scaling factor [292]. The elevation scaling is only performed when the LOC elevation (3rd entry) differs from that of the selected **BACKGROUND.dat** grid-point elevation. This scaling will be omitted when the LOC elevation is specified as "-1" or when the grid-point location is over seawater. These background source options require use of the LOC keyword and the sampling of this source ignores any specification for the ERG keyword. The LOC keyword identifies the normalization and energy spectrum to be sampled from the **BACKGROUND.dat** file.

(10) The DAT keyword is used with the PAR = BG, BN, BP option to scale the background fluxes from the date specified in the **BACKGROUND.dat** file to the date specified by the DAT keyword. It can be used with the PAR = BG, BN, BP option when the cosmic source is intended for use within the Earth's atmosphere (in which case solar modulation effects are included). When the keyword DAT specifies a date between 1936 and 2014, linear interpolation of the yearly solar modulation values determines the appropriate modulation. Specified dates prior to 1936 or after 2014 use a sine-wave fit to approximate the solar modulation based on the measured data available for 1936–2014.

(11) The keyword LOC is used only the PAR = CR, CH, CA, BG, BN, or BP options and should be specified when the cosmic or background source is intended for use within the Earth's atmosphere. Omission of the LOC keyword with the PAR = CR, CH, or CA option provides a cosmic source appropriate for interplanetary analysis. The 47th entry on the DBCN card can be used to switch between the default Clem formulation [293] and the Lal formation [289]. When the LOC keyword is used, the **SKYMAP.dat** file must be available in the local directory or in the **DATAPATH** directory. The sky map data file contains rigidity data on an approximate 5° latitude resolution (non-uniform spacing) and 20° longitude resolution (uniform spacing). Based on the LOC keyword, the algorithm uses a closest-match approach, first finding the closest longitude match followed by the closest latitude match. Fractional values are allowed after the LOC keyword—however, these are converted to the nearest integer degree for comparison to the sky map data. The PAR = CR, CH, CA option will automatically include heavy ions if they are included on the MODE card, unless the Lal source is specified.

(12) The PAR = SD, SN, SP, SB, ST, SA, and target identifier (with ERG = 0) options require time integration of daughter production at each level within a decay chain. This is facilitated by setting all decay constants to unity and uniformly spacing all time bins within 20 s (or ≈ 20 decay levels), which will include all decay particle production within most decay chains (i.e., an equilibrium production). The user can adjust the integration time, and thus the number of decay levels, by modifying the 55th entry on the DBCN card. For example, setting this to ≈ 1.0 s will result in daughter production and related decay particle production from just the precursor. If long-lived radionuclides are included (half life $< 1.5768 \times 10^{16}$ s) the user must increase the 10th entry on the DBCN card to obtain related decay particle production. This will also increase the fidelity of the time integration by increasing the number of time steps from 99 to 234.

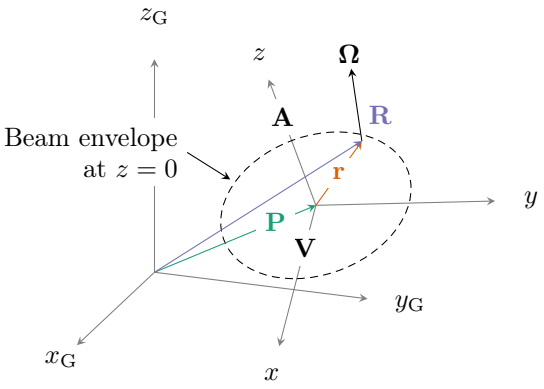


Figure 5.7: Locating and aiming a beam in the MCNP code involves a transformation from local (x, y, z) to global (x_G, y_G, z_G) coordinates. The beam aperture is located in the local-coordinate x, y plane at the entrance to the drift region ($z = 0$) at position **P** ("POS"). The beam envelope is aligned in the direction **A** ("AXS") parallel to the $+z$ local-coordinate direction with the azimuthal orientation given by **V** ("VEC"). Particle emission is in the direction **Ω** at the local-coordinate position **r** (the global position **R**) as determined by beam parameters and Monte Carlo sampling.

13 In a multiple-source-particle problem, the "energy per source particle" given in the summary tables is normalized to the source particle weight for each source particle type. If the particle type is not a source particle (listed on the MODE card, but not on SDEF), then the "energy per source particle" is normalized to the source particle weight of the lowest particle type.

14 To simplify the description of the beam parameters BEM and BAP, the beam is referenced to the z axis and the aperture is described as if it lies in the x, y plane. Other SDEF keywords, namely POS, AXS, and VEC, are employed to describe the location and orientation of the beam. These three keywords specify the center of the aperture, the beam direction, and the azimuthal orientation of the beam, respectively. The Fig. 5.7 caption explains further the keyword relationships.

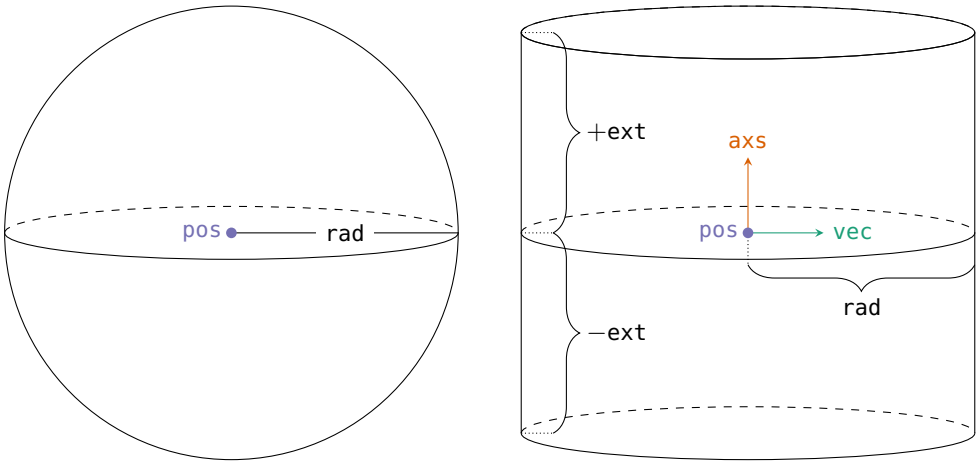
15 In the MCNP code, version 5, the specification of PAR = 4 would result in a positron. However, in the MCNP code, version 6, PAR = 4 indicates a negative muon.

16 If there is a negative igm on the MGOPT card, which indicates a special electron-photon multigroup problem, ERG on the SDEF card is interpreted as an energy group number, which is an integer.

5.8.1.1 Volume Source Specification

The three volume distributions are Cartesian, spherical, and cylindrical. A volume distribution can be used in combination with the CEL or CCC keywords to sample uniformly throughout the interior of a cell. A Cartesian, spherical, or cylindrical region that completely contains a cell is specified and is sampled uniformly in volume. If the sampled point is found to be inside the cell, it is accepted. Otherwise it is rejected and another point is sampled. If you use this technique, you must make sure that the sampling region really does contain every part of the cell because the MCNP code has no way of checking for this. Cookie-cutter (CCC) rejection can be used instead of or in combination with CEL rejection.

A Cartesian volume distribution is specified with the keywords X, Y, and Z. A degenerate case of the Cartesian distribution, in which the three variables are constants, defines a point source. A single point source can be specified easily by providing values of X, Y, and Z on the SDEF card. If several source points need to be specified, it is usually easier to use a degenerate spherical distribution for each point. Other degenerate cases of the Cartesian distribution are a line source and a rectangular plane source.



(a) Sphere (b) Cylinder

Figure 5.8: Volumetric Sampling Source-parameter Arrangements

A Cartesian distribution is an efficient shape for the CEL rejection technique when the cell is approximately rectangular. It is much better than a cylindrical distribution when the cell is a long thin slab. The Cartesian distribution is limited in that the faces can only be perpendicular to the coordinate axes.

A spherical volume distribution is specified with the keywords POS and RAD as shown in Fig. 5.8a. The keywords X, Y, Z, and AXS must not be specified or the distribution will be assumed to be Cartesian or cylindrical. The sampled value of the vector POS defines the center of the sphere. The sampled value of RAD defines the distance from the center of the sphere to the position of the particle. The position is then sampled uniformly on the surface of the sphere of radius RAD. Uniform sampling in volume is obtained if the distribution of RAD is a power law with $a = 2$, which is the default case. If RAD is not specified, the default is zero. This is useful because it specifies a point source at the position POS. A distribution for POS, with an L on the SI card, is the easiest way to specify a set of point sources in a problem.

A common use of the spherical volume distribution is to sample uniformly in the volume between two concentric spherical surfaces. The two radii are specified on the SI card for RAD and the effect of an SP*n* -21 2 card is obtained by default.

A cylindrical volume distribution is specified with the keywords POS, AXS, RAD, and EXT as shown in Fig. 5.8b. The axis of the cylinder passes through the point POS in the direction AXS. The position of the particles is sampled uniformly on a circle whose radius is the sampled value of RAD, centered on the axis of the cylinder. The circle lies in a plane perpendicular to AXS at a distance from POS which is the sampled value of EXT. A useful degenerate case is EXT = 0, which provides a source with circular symmetry on a plane (i.e., a thin disk source).

A common use of the cylindrical distribution is to sample uniformly in volume within a cylindrical shell. The distances of the ends of the cylinder from POS are entered on the SI*n* card for EXT and the inner and outer radii are entered on the SI*n* card for RAD. Uniform sampling between the two values of EXT and power law sampling between the two values of RAD, with $a = 1$ which gives sampling uniform in volume, are provided by default.

The reason for using the $a = 2$ and $a = 1$ as the power-law parameters on the radial SP cards for spheres and cylinders, respectively, leading to quadratic and linear radial sampling is because of the need to sample the radial position proportional to differential volume. That is, for a sphere, the volume is defined as $V = 4\pi r^3/3$ so $\mathrm{d}V/\mathrm{d}r = 4\pi r^2 \propto r^2$. Similarly, for cylinders, the volume is $V = \pi r^2 h$ so $\mathrm{d}V/\mathrm{d}r = 2\pi r h \propto r$, where the sampling along the extent of the cylinder based on its height, h, is constant (i.e., $\mathrm{d}V/\mathrm{d}h = \pi r^2$, which

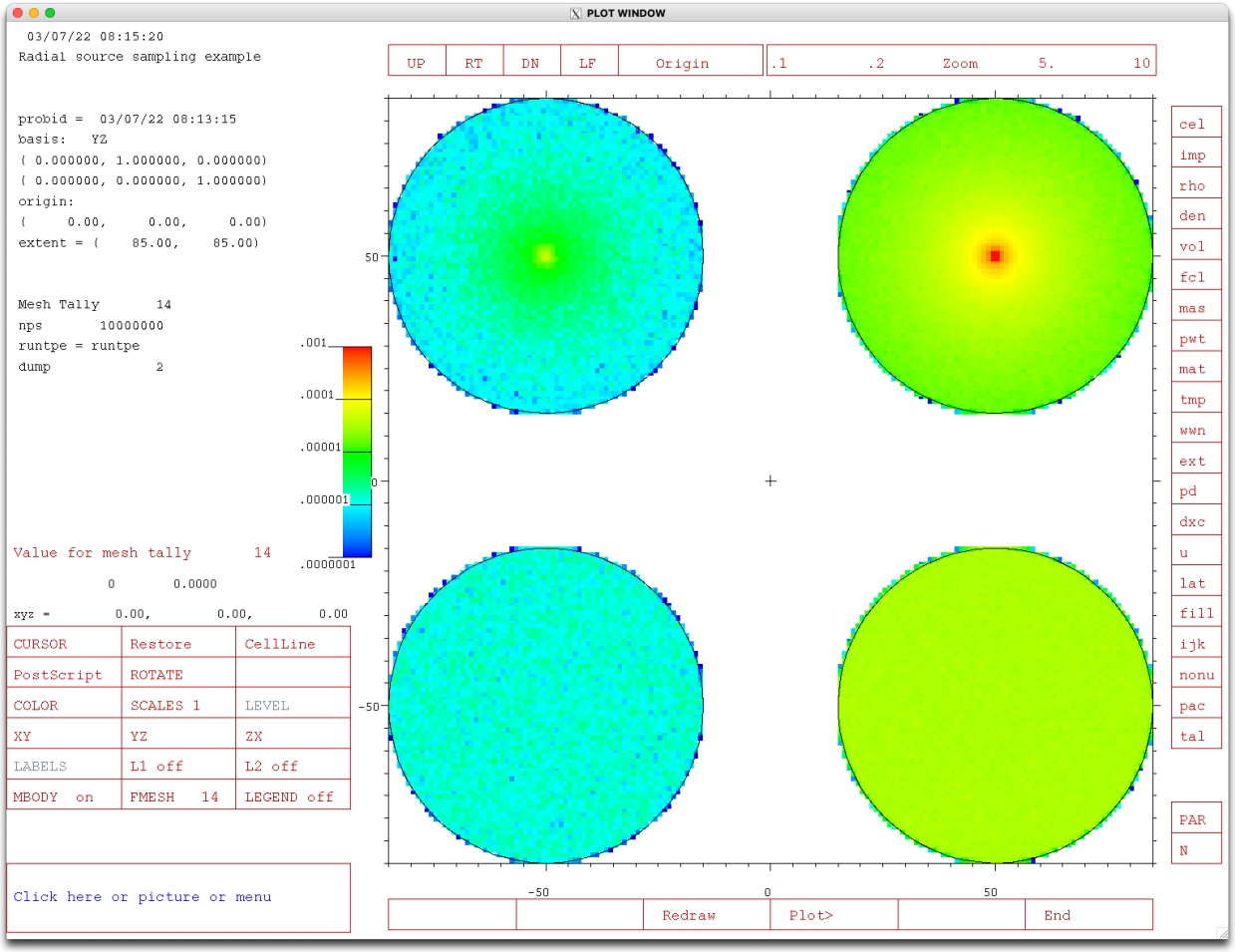


Figure 5.9: Volumetrically nonuniform (top) versus volumetrically uniform (bottom) radial sampling for spheres (left) and cylinders (right).

is height invariant). The effect of incorrectly and correctly specifying radial sampling is demonstrated in Fig. 5.9, which is generated from the MCNP input file shown in Listing 5.40. Note that the shading in the volumes in the (volumetrically uniform and generally correctly specified) lower half of Fig. 5.9 is constant when neglecting the noise that arises from uneven source sampling versus the uneven radial profiles visible in the (generally incorrectly specified) top half.

Listing 5.40: example_radial_source_sampling.mcnp.inp.txt

```
Radial source sampling example
c Most calculations using spherical and cylindrical sources have a
c volumetrically uniform source, which requires uniform volumetric sampling within
c the source volume. Using nonuniform volumetric sampling unintentionally will
c result in incorrect source sampling.  This input file demonstrates both
c approaches.
1000  0  -100                  imp:n=1 $ Sphere, Volumetrically Nonuniform - TYPICALLY WRONG
2000  0  -200                  imp:n=1 $ Sphere, Volumetrically Uniform
3000  0  -300                  imp:n=1 $ Cylinder, Volumetrically Nonuniform - TYPICALLY WRONG
4000  0  -400                  imp:n=1 $ Cylinder, Volumetrically Uniform
5000  0   100 200 300 400 -500  imp:n=1 $
9999  0   500                  imp:n=0 $ Graveyard

100 s     0 -50  50 35
```

```
200 s     0 -50 -50 35
300 rcc -0.5 50  50    1    0   0 35
400 rcc -0.5 50 -50    1    0   0 35
500 rpp -100 100 -100 100 -100 100

mode n
c
sdef cel = d1 pos = fcel = d2 rad = fcel = d3 axs = fcel = d4 ext = fcel = d5
si1 l  1000 2000 3000 4000 $ cel
sp1    0.25 0.25 0.25 0.25
ds2 s    20   21   22   23 $ pos
ds3 s    30   31   32   33 $ rad
ds4 s    40   41   42   43 $ axs
ds5 s    50   51   52   53 $ ext
c
si20 l 0 -50   50
sp20            1
si21 l 0 -50  -50
sp21            1
si22 l 0  50   50
sp22            1
si23 l 0  50  -50
sp23            1
c
si30 h 0 35
sp30 -21 1  $ Sphere, Linear Radial Sampling, Volumetrically Nonuniform - TYPICALLY WRONG
si31 h 0 35
sp31 -21 2  $ Sphere, Quadratic Radial Sampling, Volumetrically Uniform
si32 h 0 35
sp32 -21 0  $ Cylinder, Constant Radial Sampling, Volumetrically Nonuniform - TYPICALLY WRONG
si33 h 0 35
sp33 -21 1  $ Cylinder, Linear Radial Sampling, Volumetrically Uniform
c
si40 l 0 0 0
sp40       1
si41 l 0 0 0
sp41       1
si42 l 1 0 0
sp42       1
si43 l 1 0 0
sp43       1
c
si50 l 0
sp50   1
si51 l 0
sp51   1
si52 h -0.5 0.5
sp52      0   1
si53 h -0.5 0.5
sp53      0   1
c
fmesh14:n geom=xyz origin=-35  -85 -85 imesh=35  iints=70
                                       jmesh=85  jints=170
                                       kmesh=85  kints=170
                                       out=none  type=source
rand gen=2 seed=12345
print
nps 1e7
```

Caution

Never position any kind of degenerate volume distribution so that it lies on a defined surface of the problem geometry. Even a bounding surface that extends into the interior of a cell can cause trouble. If possible, use one of the surface distributions instead. Else, move to a position a small distance from the surface. This positioning will make no detectable difference in the answers, but will prevent particles from getting lost.

5.8.1.2 Surface Source Specification

The value of the keyword `SUR` is non-zero for a distribution on a surface. The shape of the surface can be a spheroid, sphere, cylinder, or plane. A spheroid is an ellipse revolved around one of its axes. If `X`, `Y`, and `Z` are specified, their sampled values determine the position. The user must in this case make sure that the point really is on the surface because the MCNP code does not check. If `X`, `Y`, and `Z` are not specified, the position is sampled on the surface `SUR`. With the exception of a spherical surface, the `SUR` keyword does not automatically provide source points on the listed surface. The user must still use the `X`, `Y`, `Z`, `POS`, `AXS`, `RAD`, and `EXT` keywords to ensure the source points actually lay on the prescribed surface. For a surface source, sampling using `CEL` rejection is not an option; however, cookie-cutter rejection can be used.

If the value of `SUR` is the name of a spheroidal surface, the position of the particle is sampled uniformly in area on the surface. A spheroid for this purpose must have its axis parallel to one of the coordinate axes. Although there is no provision for easy non-uniform or biased sampling on a spheroidal surface, a distribution of cookie-cutter cells could be used to produce a crude non-uniform distribution of position.

If the value of `SUR` is the name of a spherical surface, the position of the particle is sampled on that surface. A spherical surface source does not have to be on a cell-bounding problem surface. If the vector `AXS` is not specified, the position is sampled uniformly in area on the surface. If `AXS` is specified, the sampled value of `EXT` is used for the cosine of the angle between the direction `AXS` and the vector from the center of the sphere to the position point. The azimuthal angle is sampled uniformly in the range from 0° to 360°. A non-uniform distribution of position, in polar angle only, is available through a non-uniform distribution of `EXT`. A biased distribution of `EXT` can be used to start more particles from the side of the sphere nearest the tallying regions of the geometry. The exponential distribution function (`-31`, Table 5.15) usually is the most appropriate way to specify this behavior. The keyword `DIR` may be specified without `VEC`, allowing `VEC` to default to the outward surface normal.

Cylindrical surface sources must be specified as degenerate volume sources. For a cylindrical surface source, the cylindrical surface can be, but does not have to be, a cell-bounding problem surface specified by the keyword `SUR`. If the cylindrical surface is a problem surface, then the surface number must be specified on the `SDEF` card with the `SUR` keyword. The default of `VEC` is the surface normal. If both `DIR` and `VEC` are specified, then particle directions are relative to `VEC` rather than to the cylindrical surface normal. `DIR` may be specified without `VEC`, causing `VEC` to default to the outward surface normal.

If the value of `SUR` is the name of a plane, the position is sampled on that plane. The sampled value of `POS` must be a point on the plane. The user must make sure that `POS` really is on the plane because the MCNP code does not check. The sampled position of the particle is at a distance from `POS` equal to the sampled value of `RAD`. The position is sampled uniformly on the circle of radius `RAD` centered on `POS`. Uniform sampling in area is obtained if the distribution of `RAD` is a power law with $a = 1$, which is the default in this case.

5.8.1.3 Unstructured Mesh Source Specification

The `VOLUMER` option is a new option for the `POS` parameter on the `SDEF` card. More information about describing volume sources for the MCNP unstructured mesh calculation can be found in the source keyword

discussion in §8.3.1.1 for a mesh model formatted as an Abaqus input file and in the source group discussion in §D.6.2.4for a mesh model formatted as an HDF5 file. The VOLUMER value is for unstructured mesh volume source(s) so that x, y, z may be sampled from the volume source description. Note that the last character 'R' stands for sampling by rejection. This section describes how the user can select among multiple volume sources defined in the pseudo-cells.

First, if volume sources have been defined in the mesh model and you do not wish to sample from them, don't use the VOLUMER value anywhere in describing the source on the SDEF card. A fatal error is thrown if the VOLUMER value is used in describing the source on the SDEF card but the volume sources are not defined in the mesh model, or if the VOLUMER value is associated with a pseudo-cell that does not use source elements.

Second, to sample uniformly over all volume source regions defined in a model, simply set the POS parameter to VOLUMER:

```
SDEF POS=VOLUMER
```

Next, if the volume sources appear in different pseudo-cells and you desire to sample non-uniformly among the pseudo-cells, use a dependent distribution where POS is a function of CEL. Only uniform sampling within a cell is possible:

```
SDEF CEL=D1 POS=FCEL=D2
SI1 L 101 103
SP1 0.4 0.6
DS2 L VOLUMER VOLUMER
```

In this example, the MCNP code will first select proportionally from cells 101 (40%) and 103 (60%). With the cell selected, the code will select uniformly over that cell proportional to each element's volume to find an element from which it will uniformly sample a position.

Finally, it is possible to combine volume sources with point sources (and other legacy source descriptions) with a dependent distribution of distributions.

```
SDEF CEL=D1 POS=FCEL=D2
C
SI1 L 101 102 103
SP2   0.4 0.2 0.4
C
DS2 S 4 5 6
C
SI4 L VOLUMER
SP4   1
C
SI5 L .1 .2 .3
SP5    1
C
SI6 L VOLUMER
SP6   1
```

As before, the cell is selected first, then the position from the appropriate distribution. In this example, the point source is selected 20% of the time.

5.8.1.4 Guidance: Defining Embedded Source Distribution Information

Source distributions may be embedded within each other to describe accelerator micro-pulses and other phenomena. The format to specify an embedded source is

```
SDEF TME=( D11 < D12 < D13 )
```

or, for distributions of distributions, the following form may be used:

```
SDEF   TME=D41
SI41   S  51  ( D11 < D12 < D13 ) 52
```

In both cases, distributions 11, 12, 13 are all for the same variable, time. Distribution 11 covers a small time range that is repeated as often as needed to fill exactly the larger time range of distribution 12. Similarly, distribution 12 is repeated as often as needed to fill exactly the even larger time range of distribution 13. See [§5.8.6.21] for an example.

Note that the parentheses are optional and that the designator "D" on the SI card with "S" option is also optional. Thus

```
SDEF   TME=( D11 < D12 < D13 )
```

and

```
SDEF   TME=  D11 < D12 < D13
```

are equivalent.

Also,

```
SI41   S  51  ( D11 < D12 < D13 ) 52
```

and

```
SI41   S  51    D11 < D12 < D13   52
```

and

```
SI41   S  51  (  11 <  12 <  13 ) 52
```

and

```
SI41   S  51     11 <  12 <  13   52
```

are all equivalent.

The embedded distributions must start at zero or a fatal error message is issued. For (D11 < D12 < D13) the lowest value on the SI11 and SI12 cards must be zero. The embedding distribution, D13, can have any range.

Caution

The **-21** entry on an SP card for a power-law distribution cannot be used with the embedded distributions; it will lead to incorrect results.

The embedded distributions should fit within each other (nearly) exactly. If they do not there the fatal error message, "embedded distribution *nn* has improper range" is issued and the distribution will spill into the next bin and have a strange normalization for values in its last bin.

Only continuous source distributions such as ERG, TME, X, Y, Z, DIR, RAD and EXT may use embedded distributions.

5.8.1.5 Guidance: Defining a Source in a Repeated Structures or Lattice Geometry

Hint: Carefully study PRINT Table 110 in the MCNP output file to ensure that the proper source path and position are being sampled when repeated structures are used in a source description.

When the source is specified in a repeated structure part of the geometry, the CEL parameter on the SDEF card must have a value that is a path, enclosed in parentheses, from level n to level 0 (i.e., the highest level), where n is not necessarily the bottom level:

$$\texttt{CEL} = (\texttt{c}_n < \texttt{c}_{n-1} < \cdots < \texttt{c}_0) \tag{5.24}$$

In this specification $\texttt{c}_i$ is either zero or a cell in the universe that fills cell $\texttt{c}_{i-1}$, or is Dm for a distribution of cells in the repeated structure case. A distribution of cells (i.e., Dm) is not valid for a lattice; however, a range of lattice elements may be specified. Cell designator $\texttt{c}_i$ can have a minus sign, but Dm cannot. This is discussed below. If $\texttt{c}_i = 0$, the cell at that level is searched for. If $\texttt{c}_i$ is one specific element in a lattice, it is indicated as

$$\cdots < \texttt{c}_i[\texttt{i j k}] < \cdots$$

The coordinate system for position and direction sampling (PDS) is the coordinate system of the first negative or zero $\texttt{c}_i$ in the source path starting from the right and proceeding left. Each entry in the source path represents a geometry level, where level zero is the last specified source path entry, level one is the second entry to the left, and so forth. Level zero is above level one and level two is below level one. The PDS level is the level associated with the PDS cell or PDS coordinate system. All levels above the PDS level must be included in the source path. Levels below the PDS level need not be specified, and when given, may include one or more zero entries. When the path has no negative or zero entry, the default PDS level is the first (i.e., lowest) entry in the source path.

Position rejection is done in cells at all levels where $\texttt{c}_i \neq 0$, but if any $\texttt{c}_i$ has a negative universe number on its cell card and is at or above the PDS level, higher level cells are not checked.

Table 5.13: Cell Path versus PDS Level

CEL Source Path	Cell of PDS Level	PDS Level
(5 < 6 < 7 < 8)	5	3
(6 < -7 < 8)	7	1
(0 < 4 < 0 < -6 < 7 < 8)	6	2
(0 < 6[0 0 0] < -7[1 0 0] < 8)	7	1
(0 < 6[0 0 0] < 7[1 0 0] < 8)	Will be determined	3

Table 5.14: Cell Path versus Accepted/rejected Lattice Elements

CEL Source Path	Accepted	Rejected
7	All elements	None
(0 < 7)	All elements	None
(8 < 7)	[1 0 0]	[0 0 0], [2 0 0]
(10 < 7)	[2 0 0]	[0 0 0], [1 0 0]

Table 5.13 illustrates the concept of the PDS level.

A range of lattice indices may be specified to produce a uniform sampling among those lattice elements. The ability to sample source points from a range of lattice indices requires the use of a fully specified FILL card for the listed lattice cell. The sampling is accomplished using rejection on all possible lattice elements. Note that the SDEF keyword EFF may need to be decreased to accommodate sampling of a small portion of a large lattice. A lattice cell without indices results in uniform sampling in all elements if a fully specified FILL card is provided. Uniform sampling is applied to lattice cell entries in the source path that lack an explicit lattice index and that are at or above the PDS level. Lattice cells not defined by the expanded FILL card must include an explicit lattice index when at or above the PDS level. Rejection of automatically sampled lattice elements depends on the entry before the lattice cell number in the source path.

Assume the following cell descriptions where cell 7 is a 3-element lattice defined using the following data entries:

```
lat=1 u=1 fill=0:2 0:0 0:0  1 2 3
```

Cells 8 and 9 are members of universe 2, and cells 10 and 11 are members of universe 3.

Cell 7 is a lattice with three existing elements: [0 0 0], which is filled by itself [u=1]; [1 0 0], which is filled by cells 8 and 9 [u=2]; and [2 0 0], which is filled by cells 10 and 11 [u=3]. Table 5.14 show which elements are accepted and which are rejected.

The sampling efficiency for cell 7 in the MCNP output file will reflect the element rejections. Lattice cell entries that lack an explicit lattice index and are below the PDS level are not sampled. Instead, the appropriate lattice element is determined by the input source position.

Lattice element sampling is independent from position sampling. First a lattice element is chosen, then a position is chosen. If the sampled position is not in the sampled lattice element, the position is resampled until it is in the specified source path and in the lattice element chosen or until an efficiency error occurs. The lattice elements will not be resampled to accommodate the sampled position. Lattice element rejection is done only as described above.

Using the previous description of lattice cell 7, add that cell 6 is filled by cell 7. The source path becomes (0 < 7 < 6). Three elements of the lattice exist (fill = 0 : 2 0 : 0 0 : 0) but element [0 0 0] now is cut off

by cell 6. Lattice element [0 0 0] still will be sampled one-third of the time. The first time element [0 0 0] is sampled a fatal error will occur because the sampled position, no matter what it is, will be rejected because element [0 0 0] does not exist.

⚠ Caution

Implement automatic lattice sampling carefully and ensure that all of the lattice elements specified on the expanded FILL card really do exist.

Note that the format of the CEL source path is the same as for tally cards. See [§5.9.1.5] for more information about specifying the path for repeated structures or lattices for tallies.

5.8.1.6 Shorthand: Specifying Multiple Cell Paths for Repeated Structures or Lattices

The source cell path input format also allows a shorthand notation for one source cell path to represent a number of source paths, similar to the way that one "tally 4" path sequence enclosed in parentheses can represent a number of separate tallies. For example, the input source path (5 < 7 8 9 10 11 < 1) is interpreted by the MCNP code as the following five paths: (5 < 7 < 1), (5 < 8 < 1), (5 < 9 < 1), (5 < 10 < 1), and (5 < 11 < 1). The sequence order of these paths is determined from left to right in the original input master path. Similarly, single or multiple lattice indexes within the square brackets of path (5 < 3[...] < 2) can have the following four optional input forms for the $[i, j, k]$ index data for lattice cell element(s) with the FILL array defined on the cell 3 card:

i	Indicates the ith lattice element of cell 3 as defined by the FILL array using only one count index; e.g., $i = 1$ is the first element.
$i\ j\ k$	Indicates a lattice element from the FILL array using the three indexes.
$i_1{:}i_2\ \ j_1{:}j_2\ \ k_1{:}k_2$	Indicates a range of one or more lattice elements, where the ":" and last entry of any of the three pairs can be omitted if that lattice element does not vary.
$\mathrm{U} = m$	Specifies all of the lattice elements that have universe "m".

For the third specification form listed above, the MCNP code will create "n" source paths, where

$$n = (i_2 - i_1 + 1) \times (j_2 - j_1 + 1) \times (k_2 - k_1 + 1), \tag{5.25}$$

with the order of these n paths being the order of the indexes changing from left to right with the left index varying most rapidly. For the fourth specification, the n source paths are the number of lattice elements with universe m, where the order of the source paths is the order in the FILL matrix for cell 3. Since the SP card must specify the corresponding probabilities, this sequence order may be important. This sequence of the split paths is shown in the "cell" column of PRINT Table 10 of the MCNP output file.

When more than one cell (or lattice cell) is specified on more than one level in the source input path, the MCNP code splits into multiple paths with the variation most rapid from the left. However, the first level (level n) and the last level (level 0) entered in the source input path can only have one entry. The path in this new format must always be enclosed in parentheses, but there must not be any inner parentheses in the path.

5.8.1.7 Spontaneous Fission Sources: Physics and Tally Normalization

Eighteen nuclides are available for a spontaneous fission source (PAR = SF): ^{232}Th, ^{232}U, ^{233}U, ^{234}U, ^{235}U, ^{236}U, ^{238}U, ^{237}Np, ^{238}Pu, ^{239}Pu, ^{240}Pu, ^{241}Pu, ^{242}Pu, ^{241}Am, ^{242}Cm, ^{244}Cm, ^{249}Bk, and ^{252}Cf.

If more than one spontaneous-fission nuclide is present in a source cell, the fissioning nuclide will be chosen proportionately to the product of its atom fraction and the spontaneous-fission yield for each nuclide. If no spontaneous-fission nuclide is found in a specified source cell, the code exits with a "BAD TROUBLE" error: "spontaneous fission impossible."

The number of spontaneous-fission neutrons then is sampled. The spontaneous-fission multiplicity data of Santi [272] and references cited by him are used by default. Alternatively, the LLNL FREYA or CGMF fission model can be used (see the FMULT card for more details). The energies are sampled from a Watt spectrum with appropriate spontaneous-fission parameters for the selected nuclide. Only the first spontaneous-fission neutron from each history is printed. If the spontaneous fission samples a multiplicity of zero—that is, no neutrons for a given spontaneous fission—then the history is omitted from the first 50 history lists of PRINT Table 110. The number of source particles is the number of spontaneous-fission neutrons, which will be $\overline{\nu}$ times the requested number of source histories on the NPS card.

The spontaneous fission source is different from most other SDEF fixed sources. Let

- N be the number of source-particle histories run in the problem,
- W be the average source particle weight, and
- $\overline{\nu}$ be the average number of spontaneous fission neutrons per fission.

For most other fixed-source (SDEF) problems,

- N is the summary table source tracks,
- W is the summary table source weight, and
- summary tables and tallies are normalized by N.

For the spontaneous fission source, SDEF PAR = SF,

- summary table source tracks = $\overline{\nu} \cdot N$,
- summary table source weight = W, and
- summary tables and tallies are normalized by $\overline{\nu} \cdot N$, the number of spontaneous fission neutrons.

For the spontaneous fission source, SDEF PAR = -SF,

- summary table source tracks = $\overline{\nu} \cdot N$
- summary table source weight = $\overline{\nu} \cdot W$, and
- summary tables and tallies are normalized by N, the number of spontaneous fissions.

5.8.2 SI: Source Information

Data-card Form: *SIn* option $i_1 \ldots i_K$

n	Distribution number from corresponding distribution number on SDEF card. Restriction: $1 \leq n \leq 999$
option	Determines how the *i* values are interpreted. If

option = H or absent	*i* values are monotonically increasing histogram bin upper boundaries (scalar only) (1). (DEFAULT)
option = L	*i* values are discrete source variable values (e.g., cell numbers or energies of photon spectrum lines).
option = A	*i* values are points where a probability density is defined. Entries must be monotonically increasing, with the lowest and highest values defining the range of the variable (2).
option = S	*i* values are distribution numbers (3).

$i_1 \ldots i_K$ Source variables or distribution numbers

Default: SI*n* H $i_1 \ldots i_K$

Details:

1. The H option is an integral, bin-wise method for describing a source distribution. It is integral in the sense that the fundamental differential distribution (e.g., particles/MeV for energy) must be integrated over an interval and its integration value placed on the SP card, corresponding to the upper bin value listed on the SI card. For example, if an energy differential distribution is integrated from E_1 to E_2 (and these are the first two entries (i_1 and i_2) on the SI card), then the integration value over this interval is listed as the 2nd entry (p_1) on the corresponding SP card.

2. When the A option is used, the entries on the SI card are values of the source variable at which the probability density is defined. The A option is a differential, point-wise method for describing a source distribution. The fundamental differential distribution is placed directly on the SP card, in a point-wise fashion. For each point listed on the SI card, the corresponding value of the differential distribution is listed on the SP card. For example, if an energy differential distribution has a value of V_1 at E_1 and V_2 at E_2, then the SI entries i_1 and i_2 become E_1, E_2 and the SP entries p_1 and p_2 become V_1 and V_2. Typically, the first entry on the SP card would not be zero (although it can be). To sample this description of a source variable, the code must integrate the point-wise distribution and formulate an integral, bin-wise cumulative distribution for sampling (i.e., basically do what the user had to do when using the H option). To accomplish this, the code uses a corrected trapezoidal (i.e., linear) integration scheme, along with linear interpolation for intra-bin sampling. While this integration scheme is fairly accurate, users are encouraged to increase the number of points on their SI/SP cards and note effects to tallies to ensure this linear integration scheme is adequate for their specified differential distribution. Included in PRINT Table 10 are the integral, bin-wise cumulative distribution that will be used when sampling the associated source variable and the renormalized input differential distribution.

3. The S option on the SI card allows sampling among distributions, one of which is chosen for further sampling. This feature makes it unnecessary to fold distributions together and is essential if some of the distributions are discrete and others are linearly interpolated. The distributions listed on an SI card with the S option can themselves have the S option. Each distribution number on the SI card can be prefixed with a D, or the D can be omitted. If a distribution number is zero, the default value for the variable is

used. A distribution can appear in more than one place with an S option, but a distribution cannot be used for more than one source variable.

5.8.3 SP: Source Probability

Data-card Form: *SP*n option $p_1 \ldots p_K$
or
Data-card Form: *SP*n -f *a b*

n	Distribution number from corresponding distribution number on SDEF and SI cards. Restriction: $1 \leq n \leq 999$
option	Determines how the *p* values are interpreted (1). If
option absent	it is the same as D for an H or L distribution on the SI card or probability density for an A distribution on the SI card (2).
option = D	*p* values are bin probabilities for an H or L distribution on the SI card (3, 4). (DEFAULT)
option = C	*p* values are cumulative bin probabilities for an H or L distribution on the SI card (5, 6).
option = V	*p* values are for cell distributions; probability is proportional to cell volume ($\times p_i$ if p_i are present) (5).
option = W	*p* values are intensities for a mix of particle sources. Negative *p* values corresponding to SF or SP sources indicate cell numbers, the volumes of which will be used for the computation of the intensity (6).
$p_1 \ldots p_K$	Source variable probabilities. Restriction: Must be zero for 1st histogram bin
-f	Designator (negative number) for a built-in function.
a b	Parameters for the built-in function (Refer to Table 5.15).

Default: SP*n* D $p_1 \ldots p_K$

The first form of the SP card, where the first entry is positive or non-numeric, indicates that it and its SI card define a probability distribution function. The entries on the SI card are either values of the source variable or, when the S option is used, distribution numbers. The entries on the SP card are probabilities that correspond to the entries on the SI card.

The second form of the SP card, where the first entry is negative, indicates that a built-in analytic function is to be used to generate a continuous probability density function for the source variable. Built-in functions can be used only for scalar variables.

Details:

(1) Probabilities on the SP card need not be normalized.

Table 5.15: Special Source Probability Functions

Keyword	Function ID and Input Parameters	Description
ERG	-2 a	Maxwell fission spectrum
ERG	-3 a b	Watt fission spectrum
ERG	-4 a b	Gaussian fusion spectrum
ERG	-5 a	Evaporation spectrum
ERG	-6 a b	Muir velocity Gaussian fusion spectrum
TME	-7 a	Exponential decay
DIR, RAD, or EXT	-21 a	Power law: $p(x) = c\|x\|^a$
DIR or EXT	-31 a	Exponential: $p(\mu) = c\exp(a\mu)$
TME or X, Y, Z	-41 a b	Gaussian distribution of time, t, or of position coordinates (x, y, z) (for beam sources)

(2) When the A option is used on the SI card, the numerical entries on the associated SP card are values of the probability density corresponding to the values of the variable on the SI card. This set of SI and SP values creates a curve from which intermediate values are linearly interpolated. The first and last entries on the SP card will typically be zero, but non-zero values are allowed.

(3) When the H option is used on the SI card, the first numerical entry on the corresponding SP card must be zero and the following entries are bin probabilities or cumulative bin probabilities, depending on whether the D or C option on the SP card is selected. The variable is sampled by first sampling a bin according to the bin probabilities and then sampling uniformly within the chosen bin.

(4) When the L option is used on the SI card, the entries on the associated SP card are either probabilities of those discrete values or cumulative probabilities, depending on whether the D or C option is selected.

(5) The V option on the SP card is a special case used only when the source variable is CEL. This option is useful when the cell volume is a factor in the probability of particle emission. If the MCNP code cannot calculate the volume of such a cell and the volume is not provided on a VOL card, a fatal error results.

(6) The W option of the SP card allows the user to specify intensities for a mix of particle sources. The intensities will be normalized, as is done for all MCNP source distributions; however the factor used to renormalize the intensities will be applied to the source weight to give the tallies the correct magnitude. The SPn W distribution specification can only be applied to particle distributions.

5.8.3.1 Description of Built-in Probability Density and Bias Functions

The special (i.e., built-in) source probability functions are summarized in Table 5.15 and described in detail next.

f = -2 Maxwell fission energy spectrum:

$$p(E) = CE^{1/2}\exp(-E/a), \tag{5.26}$$

where a is temperature in MeV.

Default: $a = 1.2895$ MeV

f = -3 Watt fission energy spectrum:

$$p(E) = C\exp(-E/a)\sinh\left(\sqrt{bE}\right). \tag{5.27}$$

See Listing 5.29 for the default parameters used with the FMULT card for spontaneous fission.

Default: $a = 0.965$ MeV, $b = 2.29$ MeV^{-1}.

f = -4 Gaussian fusion energy spectrum:

$$p(E) = C \exp\left[-((E-b)/a)^2\right], \tag{5.28}$$

where a is the width in MeV and b is the average energy in MeV. Width here is defined as the ΔE above b where the value of the exponential is equal to e^{-1}. If $a < 0$, it is interpreted as a temperature in MeV and b must also be negative. If $b = -1$, the D-T fusion energy is calculated and used for b. If $b = -2$, the D-D fusion energy is calculated and used for b. Note that a is not the full-width-at-half-maximum (FWHM) but is related to it by FWHM $= 2a\sqrt{\ln(2)}$.

Default: $a = -0.01$ MeV, $b = -1$ (DT fusion at 10 keV).

f = -5 Evaporation energy spectrum:

$$p(E) = CE \exp(-E/a). \tag{5.29}$$

Default: $a = 1.2895$ MeV.

f = -6 Muir velocity Gaussian fusion energy spectrum:

$$p(E) = C \exp\left[-\left(\left(E^{1/2} - b^{1/2}\right)/a\right)^2\right], \tag{5.30}$$

where a is the width in MeV$^{1/2}$, and b is the energy in MeV corresponding to the average speed. Width here is defined as the change in velocity above the average velocity $b^{1/2}$, where the value of the exponential is equal to e^{-1}. To get a spectrum somewhat comparable to f = -4, the width can be determined by $a = \sqrt{b + a_4} - b^{1/2}$, where a_4 is the width used with the Gaussian fusion energy spectrum. If $a < 0$, it is interpreted as a temperature in MeV. If $b = -1$, the D-T fusion energy is calculated and used for b. If $b = -2$, the D-D fusion energy is calculated and used for b.

Default: $a = -0.01$ MeV, $b = -1$ (D-T fusion at 10 keV).

f = -7 Exponential decay:

$$p(t) = \alpha_0 (1/2)^{t/a}. \tag{5.31}$$

Allows the creation of a source with an exponential decay shape. The activity at TME = 0 is given by α_0. The parameter a is the half-life in shakes.

Default: $a = 1$.

f = -21 Power law:

$$p(x) = c|x|^a. \tag{5.32}$$

The default depends on the variable. For DIR, $a = 1$. For RAD, $a = 2$, unless AXS is defined or SUR $\neq 0$, in which case $a = 1$. For EXT, $a = 0$.

f = -31 Exponential:

$$p(\mu) = c \exp(a\mu). \tag{5.33}$$

Default: $a = 0$.

`f = -41` Gaussian distribution of time t or position coordinates x, y, z:

$$p(t) = c \exp\left[-(1.6651092(t-b)/a)^2\right], \tag{5.34}$$

where a is the width at half maximum and b is the mean. For time, a and b are in shakes, while for position variables, the units are centimeters. Note: This distribution may be written in normal form as

$$p(t) = c \exp\left[-(t-b)^2/2\sigma^2\right]. \tag{5.35}$$

The FWHM is thus $a = \sqrt{8 \ln 2}\sigma$.

Default: a = no default, $b = 0$.

The built-in functions can be used only for the variables shown in Table 5.15. Any of the built-in functions can be used on SP cards, but only -21 and -31 can be used on SB cards. If a function is used on an SB card, only that same function can be used on the corresponding SP card. The combination of a regular table on the SI and SP cards with a function on the SB card is not allowed.

A built-in function on an SP card can be biased or truncated or both by a table on SI and SB cards. The biasing affects only the probabilities of the bins, not the shape of the function within each bin. If it is biased, the function is approximated within each bin by n equally probable groups such that the product of n and the number of bins is as large as possible but not over 300. Unless the function is -21 or -31, the weight of the source particle is adjusted to compensate for truncation of the function by the entries on the SI card.

Special defaults are available for distributions that use built-in functions:

- If SB *f* is present and SP *f* is not, an SP *f* with default input parameters is, in effect, provided by the MCNP code.
- If only an SI card is present for RAD or EXT, an SP -21 with default input parameters is, in effect, provided.
- If only SP -21 or SP -31 is present for DIR or EXT, an SI 0 1 for -21, or SI -1 1 for -31, is, in effect, provided.
- If SI *x* and SP -21 are present for RAD, the SI is treated as if it were SI 0 *x*.
- If SI *x* and SP -21 or SP -31 are present for EXT, the SI is treated as if it were SI -*x* *x*.

5.8.4 SB: Source Bias

The SB card is used to provide a probability distribution for sampling that is different from the true probability distribution on the SP card. Its purpose is to bias the sampling of its source variable to improve the convergence rate of the problem. The weight of each source particle is adjusted to compensate for the bias. All rules that apply to the first form of the SP card apply to the SB card.

Data-card Form: SBn option $b_1 \ldots b_K$
or
Data-card Form: SBn -f *a b*

n	same as for the SP card.
option	same as for the SP card.
$b_1 \ldots b_K$	source-variable-biased probabilities.
-f	same as for the SP card, except that the only values allowed for *-f* are -21 and -31.
a b	same as for the SP card.

Default: SB*n* D $b_1 \ldots b_K$

5.8.5 DS: Dependent Source Distribution

The DS card is used instead of the SI card for a variable that depends on another source variable, as indicated on the SDEF card. No SP or SB card is used. The MCNP code first determines the value of the independent variable as usual by sampling the probability function of the independent variable. Then the value of the dependent variable is determined according to the form of the DS card.

The first form of the DS card has several possibilities. If the SI card of the independent variable has a histogram distribution of m bins ($m+1$ entries) and the DS card has the blank or H option, the DS card must have $m+1$ entries to specify m bins. The first entry need not be zero. If the sampled value of the independent variable is $i_k + [f(i_{k+1} - i_k)]$, then the value of the dependent variable is $j_k + [f(j_{k+1} - j_k)]$, where the terms in f are used only for continuous distributions. The interpolation factor f always exists whether or not it is needed for the independent distribution.

The second form of the DS card specifies the T option. When the T option is selected, the sampled value of the independent variable is sought among the i_k, and if a match is found, the independent variable gets the value j_k. If no match is found, the dependent variable gets its default value. The purpose of the T option is to shorten the input when a dependent variable should usually get the default value.

When the Q option is used on a DS card, as it is in the third form, the v_k define a set of bins for the independent variable. The sampled value of the independent variable is compared with the v_k, starting with v_1, and if the sampled value is less than or equal to v_k, the distribution s_k is sampled for the value of the dependent variable. The value of v_k must be greater than or equal to any possible value of the independent variable. If a distribution number s_k is zero, the default value for the variable is used. The Q option is the only form of the DS card that can be used when the distribution of the independent variable is a built-in function.

Data-card Form: DS*n option* $j_1 \ldots j_K$
or
Data-card Form: DS*n* T i_1 j_1 ... i_K j_K
or
Data-card Form: DS*n* Q v_1 s_1 ... v_K s_K

n	Distribution number. Restriction: $1 \le n \le 999$
option	Determines how the *j* values are interpreted. If
	option = H or absent source variable values in continuous distribution, for

		scalar variables only. (DEFAULT)
	option = L	discrete source variable values follow. (1)
	option = S	distribution numbers follow. (1, 2)
$b_1 \ldots b_K$	source-variable-biased probabilities.	
T	Values of the dependent variable (j_k) follow values of the independent variable (i_k), which must be a discrete scalar variable.	
i_k	Values of the independent variable.	
j_k	Values of the dependent variable.	
Q	Distribution numbers (s_k) follow values of the independent variable (v_k), which must be a scalar variable. (3)	
v_k	Monotonically increasing set of values of the independent variable.	
s_k	Distribution numbers for the dependent variable.	

Default: DS*n* H $j_1 \ldots j_K$

Details:

1. If the L or S option is used on the DS card, m entries are required to specify m discrete values (for all options on the independent variable except H). See 2 for an independent variable that is represented by a histogram. It is not necessary for the distributions of the independent and dependent variables to be both discrete or both continuous. All combinations work correctly.
2. If the S option is used on the DS card and the independent variable has a histogram defined by $m + 1$ SI entries, then m numbers must appear on the DS card. Recall that the first bin of a histogram distribution must have an SP value of 0.0. The code will assume that the first independent histogram bin is ignored. A fatal error will result if a dependent source value is assigned to the first histogram bin.
3. The DS Q option does not support using cells, surfaces, or transforms as the independent variable, as the sort order of these variables is not maintained internally in the code. The T option is more appropriate for these variables.

5.8.6 Examples of the General Source Card and Distribution Cards

5.8.6.1 Example 1

```
SDEF
```

This card specifies a 14-MeV isotropic point source at position $(0, 0, 0)$ at time 0 with weight 1 (all defaults).

5.8.6.2 Example 2

```
SDEF   ERG=D1  POS=x y z  WGT=w
SI1    H   e1  e2 ... ek
SP1    D    0  p2 ... pk
SB1    D    0  b2 ... bk
```

This is a point isotropic source at (x, y, z) with a biased histogram energy distribution and average source particle weight w. The starting cell is not specified. The MCNP code will determine it from the value of (x, y, z).

5.8.6.3 Example 3

```
SDEF   SUR=m  AXS=i j k  EXT=D6
SB6    -31   1.5
```

This is a source on surface `m`. The presence of `AXS` and `EXT` implies that surface `m` is a sphere because `AXS` and `EXT` are not otherwise used together for sources on a surface. By default, the particles are emitted in a cosine distribution. They are emitted outward if the positive normal to the sphere is outward, which it is for all the spherical surface types but might not be if the sphere is specified as type `SQ`. The position on the surface is biased toward the direction (i, j, k) by an exponential bias (specified by `-31`). Table 2.10 shows the effect of the biasing parameter on the maximum and minimum source particle weights and the cumulative probability distribution. By default, the MCNP code provides the effect of two cards: `SI6 -1 1` and `SP6 -31 0`.

5.8.6.4 Example 4

```
SDEF   SUR=999   NRM=-1   DIR=D1   WGT=1.13097e6
SB1    -21 2
void
f4:n   1 2 3 4
vol    1 5r
imp:n  1.0 4r  0.0
```

These data cards illustrate how an inward-directed (`NRM = -1`), biased cosine source on a spherical surface can be used to stochastically calculate the volume of MCNP cells. All materials are voided in the problem (`VOID` card) and all non-zero importance are set to 1 (`IMP:N` card). In this example, the surface source is placed on the surface of a 600-cm-radius sphere (`SUR = 999`) that surrounds the cells of interest and the source weight (`WGT`) is set to 1.13097×10^6 cm^2 (πr^2). All volumes are forced to unity (`VOL` card). Type 2 and type 4 flux tallies will provide estimates of the areas and volumes of the cells, respectively. By default, the MCNP code provides the effect of two cards: `SI1 0 1` and `SP1 -21 1`. The directional bias by the `SB1` card causes higher track density toward the center of the sphere, where presumably the cells of greatest interest lie, than it would be if the unbiased cosine distribution were used. This bias, incidentally, provides a zero-variance estimate of the (known) volume of the sphere 999.

5.8.6.5 Example 5

```
SDEF   CEL=D3  POS=0 6 0  EXT=D1  RAD=D2  AXS= 0 1 0
SI3    L (1<10[0 0 0]<11) (1<10[1 0 0]<11) (1<10[2 0 0]<11)
         (1<10[0 1 0]<11) (1<10[1 1 0]<11) (1<10[2 1 0]<11)
```

The SDEF card creates a cylindrical volume source oriented along the y axis with radius specified by the SI2 source information and SP2 source probability cards and extent given by SI1 and SP1. This CEL source specification for repeated-structures geometries is consistent with the repeated-structures tally format. The old-style format (listing cells in the opposite order separated by ":") is no longer recognized and will produce a fatal error.

5.8.6.6 Example 6

```
SDEF  POS=0 0 0  RAD=1  EXT=D1  AXS=1 0 0  SUR=5
```

```
SDEF  POS=0 0 0  RAD=1  EXT=D1  AXS=1 0 0  SUR=5  DIR=D2
```

```
SDEF  POS=0 0 0  RAD=1  EXT=D1  AXS=1 0 0  DIR=D2
```

The first SDEF card specifies a cylindrical source on surface 5 with default cosine distribution relative to the surface normal. The second SDEF card specifies a cylindrical source on surface 5 with a specified angular distribution that is relative to the cylindrical surface normal. The third SDEF source specification is similar except that a degenerate volume source is used to specify the cylindrical surface source (i.e., omitting the SUR keyword) with a specified angular distribution relative to the surface normal.

5.8.6.7 Example 7

```
SDEF  DIR=1  VEC=0 0 1  X=D1  Y=D2  Z=0  CCC=99  TR=1
SP1 -41  fx  0
SP2 -41  fy  0
TR1 x0 y0 z0   cos(theta) -sin(theta) 0   sin(theta) cos(theta) 0   0 0 1
```

The SDEF card sets up an initial beam of particles traveling along the z axis (DIR = 1, VEC = 0 0 1). Information on the x and y coordinates of particle position is detailed in the two SP cards. The z coordinate is left unchanged. The first entry on the SP cards is -41, indicating sampling from a built-in Gaussian distribution. The second SP card entry is the full width half maximum (FWHM) of the Gaussian in either the x or y direction. This value must be computed for the x and y axes by the user as follows: $f_x = \sqrt{8 \ln 2}a \approx 2.35482a$ and $f_y = \sqrt{8 \ln 2}b \approx 2.35482b$, where a and b are the standard deviations of the Gaussian in the x and y directions, respectively. More details are provided in [§10.3.2]. The third entry represents the centroid of the Gaussian in either the x or y direction. It is recommended the user input zero for this third entry and handle any transformations of the source with a TR card. The specification of the cookie-cutter cell 99 for source rejection prevents the beam Gaussian from extending infinitely. The TR card performs a rotation of the major axis of the source distribution. Other beam examples appear in [§10.3.2].

5.8.6.8 Example 8

```
SDEF   ERG=D1  POS=x y z  CEL=m  RAD=D2
       EXT=D3  AXS=i j k
SP1   -3
SI2   r1  r2
SI3   l
```

This source is distributed uniformly in volume throughout cell `m`, which presumably approximates a cylinder. The cell is enclosed by a sampling volume centered at (x, y, z). The axis of the sampling volume is the line through (x, y, z) in the direction (i, j, k). The inner and outer radii of the sampling volume are `r1` and `r2`, and it extends along (i, j, k) for a distance from (x, y, z). The user has to make sure that the sampling volume totally encloses cell `m`. The energies of the source particles are sampled from the Watt fission spectrum using the default values of the two parameters, making it a Cranberg spectrum. By default, the MCNP code interprets `SI3 l` as if it was actually `SI3 -l +l` and provides the effect of two cards: `SP2 -21 1` and `SP3 -21 0`.

5.8.6.9 Example 9

```
SDEF    SUR=m  POS=x y z  RAD=D1  DIR=1  CCC=n
SI1     r
```

This is a mono-directional source emitted from surface `m` in the direction of the positive normal to the surface. The presence of `POS` and `RAD` implies that surface `m` is a plane because `POS` and `RAD` are not otherwise used together for sources on a surface. The position is sampled uniformly in area on the surface within radius `r` of point (x, y, z). The user must make sure that point (x, y, z) actually lies on surface `m`. The sampled position is rejected and resampled if it is not within cookie-cutter cell `n`. The starting cell is found from the position and the direction of the particle. By default, the MCNP code interprets `SI1 r` as if it were actually `SI1 0 r` and provides the effect of card `SP1 -21 1`.

5.8.6.10 Example 10

```
SDEF   PAR=SF  CEL=D1  POS=D2  RAD=FPOS=D3
```

This is a spontaneous-fission source in which source points will be started from within defined spheres (`POS`, `RAD`) and limited to fission cells by `CEL`. Each sampled source point will be a spontaneous-fission site (`PAR` = `SF`) producing the appropriate number of spontaneous-fission neutrons per fission at the appropriate energy with isotropic direction.

5.8.6.11 Example 11

```
SDEF    PAR=D1
SI1 L   1  9  Li-6  Fe-56  U-238
SP1     1  1   0.1    0.3    0.5
```

Five different source particles are sampled in this example: neutrons; protons; and the three heavy ions: ^{6}Li, ^{56}Fe, and ^{238}U. The relative sampling frequency is given by the probability parameters on the `SP1` card.

5.8.6.12 An Aside on PAR = Dn

Note the following when using a distribution specification for the SDEF PAR keyword:

1. The characters L, A, H, S, Q, and T are reserved as SI and DS card options. L means discrete source variables, S means distribution numbers, etc. If the first entry on the SI or DS card is L, A, H, S, Q, or T, it will be interpreted as a distribution option. To list source particles types L, A, H, S, Q, or T, either the corresponding particle numbers (10, 34, 9, 33, 5, 32) must be used or L, A, H, S, Q, or T must appear as the second or later particle type. Generally, it is best to specify the discrete source variable option; therefore, L will be the first entry, followed by the particle types. A second L will be interpreted correctly as particle type L. For example,

```
SI99  L  -H  N  L  Q  F  T  S
```

2. Antiparticles may be designated, as usual, with negative entries:

```
SI77  L  -E  N  -H
```

3. Either characters (N, P, E, H, D, T, S, A, etc.) or numbers (1, 2, 3, 9, 31, 32, 33, 34, etc.) may be used. For example,

```
SI98  L  -H  3  -32  N
```

4. Spontaneous fission may be used as a particle type. For example,

```
SI87  L  SF  N
```

5. Particle types may be listed multiple times to give them different energy distributions, angular distributions, etc., in different parts of the problem. For example:

```
SI23  L  N  n  1  n  N
```

6. Heavy ions may be specified using the appropriate target identifier for individual ions. Multiple heavy ions may be specified for the source using a distribution. Dependent distributions can be used to specify different energies for different heavy ions. Heavy ion particle energy should be input as total energy, not energy/nucleon.

7. Tallies are normalized by dividing the total source weight by the number of source histories. Note that weight (WGT on the SDEF card) cannot be a source distribution (either independent or dependent). The weight of particles in the summary tables is controlled by the SI, SP, SB, and DS cards for the particle distribution. This normalization procedure is described in [§5.8.6.13].

5.8.6.13 Example 12

```
SDEF   PAR=D1  POS=FPAR=D2  ERG=FPAR=D3
SI1   L  H N
SP1      2 1
SB1      1 2
DS2   L  0 0 0  15 0 0
DS3   L  2 3
```

```
SDEF   PAR=FPOS=D2  POS=D1  ERG=FPOS=D3
SI1   L  0 0 0  15 0 0
SP1      2 1
SB1      1 2
DS2   L  h n
DS3   L  2 3
```

The first source definition above defines the source particle type, PAR, as the independent variable; while in the second source definition, the source particles specified by PAR depend on the source positions (POS). Both approaches result in the same source distributions.

The total source weight is WGT = 1.0 by default. From the SP1 card, the weight of the neutrons that are produced is 1/3 and the weight of protons that are produced is 2/3. From the SB1 card, the total number of neutron tracks is $2/3 \times N$ for neutrons and $1/3 \times N$ for protons (where N is the number of source histories actually run). The energy per source particle is normalized to the source particle weight for each source particle type. If the particle type is not a source particle (e.g., photons in the above problem), then the energy per source particle is normalized to the source particle weight of the lowest particle type. In this example, photon source energy would be normalized in the photon creation-and-loss table by 1/3, which is the weight of the source neutrons produced.

5.8.6.14 Example 13

Listing 5.41: example_source_fpos_ds_1.mcnp.inp.txt

```
sdef pos = d1 erg = fpos = d2
si1  l  -51 0 0    51 0 0
sp1         0.3       0.7
ds2  s        3         4
si3  h  2 10 14
sp3  d  0  1  2
sp4    -3   0.965   2.29
```

The example shown in Listing 5.41 is a point isotropic source in two locations, shown by two (x, y, z)s on the SI1 card. The code will determine the starting cell. With probability 0.3 the first location will be picked, and with probability 0.7 the second location will be chosen. Each location has a different energy spectrum pointed to by the DS2 card. All other needed source variables will use their default values.

5.8.6.15 Example 14

```
SDEF   DIR=1  VEC=0 0 1  X=D1  Y=0  Z=-2 TR=1
SI1    0.0  0.5
SP1    0.0  1.0
TR1    0.5 0.5 0.0   0.4 0.3 0.0   -0.3 0.4 0.0
```

This example generates a source uniform on a straight line from $(x, y, z) = (0.5, 0.5, -2.0)$ to $(x, y, z) = (0.9, 0.8, -2.0)$ in the $+z$ direction. In the auxiliary coordinate system, the source is easily created as uniform from $(0.0, 0.0, -2.0)$ to $(0.5, 0.0, -2.0)$ and then transformed.

5.8.6.16 Example 15

```
SDEF      TR=D1
SI1  L    1    3    5
SP1  D    1.0  1.0  1.0
SB1  C    0.2  0.5  1.0
```

In this example, a distribution of transformations is specified using TR = D1 on the SDEF card. Three transformations are assigned: TR1, TR3, and TR5. The L option on the SI card is required so that the MCNP code interprets the values as discrete transformation numbers. The option on the SP and SB cards may be blank, D, or C. For this problem, the transformations are equally probable, but are biased to sample TR1 20% of the time, TR3 30% of the time, and TR5 50% of the time.

5.8.6.17 Example 16

```
SDEF   TME=D1
SP1    -7   2e8          $ 2e8 shakes=2 seconds
```

The source shape will be represented by exponential decay with a half-life of 2 s.

5.8.6.18 Example 17

```
999  0      -999                    $ cookie cutter cell CCC
...
999  SQ     25 100 0 0 0 0 -4 0 0 0  $ surface for cell CCC
...
SDEF   DIR=1  VEC=0 0 1  X=D1  Y=D2  Z=0  CCC=999  TR=D3
SP1    -41  0.470964  0
SP2    -41  0.235482  0
SI3   L  11 22 33
SP3       1  2  3
SB3       1  1  1
TR11    0 0 -2   1 0 0   0 1 0   0 0 1
TR22   -2 0  0   0 1 0   0 0 1   1 0 0
TR33    0 -2 0  0.707107 0 0.707107  0.707107 0 -0.707107  0 1 0
```

In this example, the source particle coordinates are generated in an auxiliary coordinate system in the $(x', y', 0)$ plane around the origin with a Gaussian profile (FWHM = 0.470964) in the x' coordinate and a Gaussian profile (FWHM = 0.235482) in the y' coordinate. The beam is truncated by "cookie cutter cell" `CCC`, which restricts the source to an ellipse corresponding to two standard deviations of the Gaussian distributions in the x' and y' coordinates. The subsequent application of the transformation `TR` = `D3` results in three intersecting beams with the following characteristics:

- Beam 1 is centered at $(0, 0, -2)$ with the major axis of the beam distribution along the x axis, emitted in the $+z$ direction, with relative intensity 1;
- Beam 2 is centered at $(-2, 0, 0)$ with the major axis of the beam distribution along the y axis, emitted in the $+x$ direction, with relative intensity 2; and
- Beam 3 is centered at $(0, -2, 0)$ with the major axis of the beam distribution along the line $x = z$, emitted in the $+y$ direction, with relative intensity 3.

5.8.6.19 Example 18

```
m1  1001   1
    8016   1
    7016   1e-4          $ Unstable isotope N-16
    25054  1e-2          $ Unstable isotope Mn-54
c
sdef par=sp pos= 0 0 0   $ Location of material 1
ACT   DG=LINES
```

The source is defined as decay gammas from the unstable isotopes ^{16}N and ^{54}Mn in material 1, which is the material located at the user-provided source position coordinates $(0, 0, 0)$. The two unstable isotopes will be sampled based on their relative activities within material 1. The default time (`TME` = `0`) is assumed.

5.8.6.20 Example 19

```
mode   p #
sdef   par=N-16  erg=0  pos= 0 0 0
ACT    DG=LINES
```

Setting the source particle to the heavy ion ^{16}N (`PAR` = `N-16`) and specifying the energy of the ion as zero (`ERG` = `0`) defines the source as the decay gammas of ^{16}N. The heavy ions will not be transported. Notice that the heavy-ion symbol, `#`, appears on the `MODE` card.

5.8.6.21 Example 20

Listing 5.42: example_embedded_dist.mcnp.inp.txt

```
adapted from mcnpx_extended/test27a/inp07/inp07.inp
1   0     -1 imp:n=1
2   0      1 imp:n=0

```

```
1   so     0.001

sdef   erg 1 cel=1 tme=d41
si41   S  52<51 (D31<32<d33) 61
sp41     .1 .8 .1
si51   A -26 -16
sp51       0   1
si52   H   0 1 2
sp52       0 1 0
si61   A  32  40
sp61       1   0
si31       0 1 2
sp31       0 1 0
si32       0  16
sp32     -41 8 8
si33     -16  32
sp33       0   1
f1:n 1
t0 -30 79i 50
nps 1e7
```

The example given in Listing 5.42 illustrates how embedded distributions can reside within distributions of distributions (`D41`), and can use built-in functions (`D32` uses a Gaussian centered at $t = 8$ with FWHM $= 8$) and interpolated distributions (`D51` and `D61` use the `SI` `A` option). Distribution `D52` is embedded in distribution `D51`; distribution `D31` is embedded in distribution `D32`, which is embedded in distribution `D33`. A tally plot of this embedded distribution appears in Fig. 5.10. The tally plot is created with the MCNP interactive plotter command input file given in Listing 5.43.

Listing 5.43: example_embedded_dist.mcnp.comin.txt

```
tfc m
free t
end
end
```

5.8.6.22 Example 21

```
sdef     cel=d1
si1  L  (4<2[-1:1 -2:2 -3:3]<1)
sp1      1  104r
```

This source definition creates source particles in a subset of a lattice using ranges specified for the lattice elements. The lattice must have been defined using a fully specified `FILL` card.

5.8.6.23 Example 22

```
si1  L    n     p     h
sp1  W   3e9   5e9   2e9
```

The source shown here mixes contributions from three source particles and samples them according to their relative magnitudes (neutrons 30%, photons 50%, and protons 20%). The weight assigned to each particle will be the sum of the non-normalized values, $3 \times 10^9 + 5 \times 10^9 + 2 \times 10^9 = 1 \times 10^{10}$.

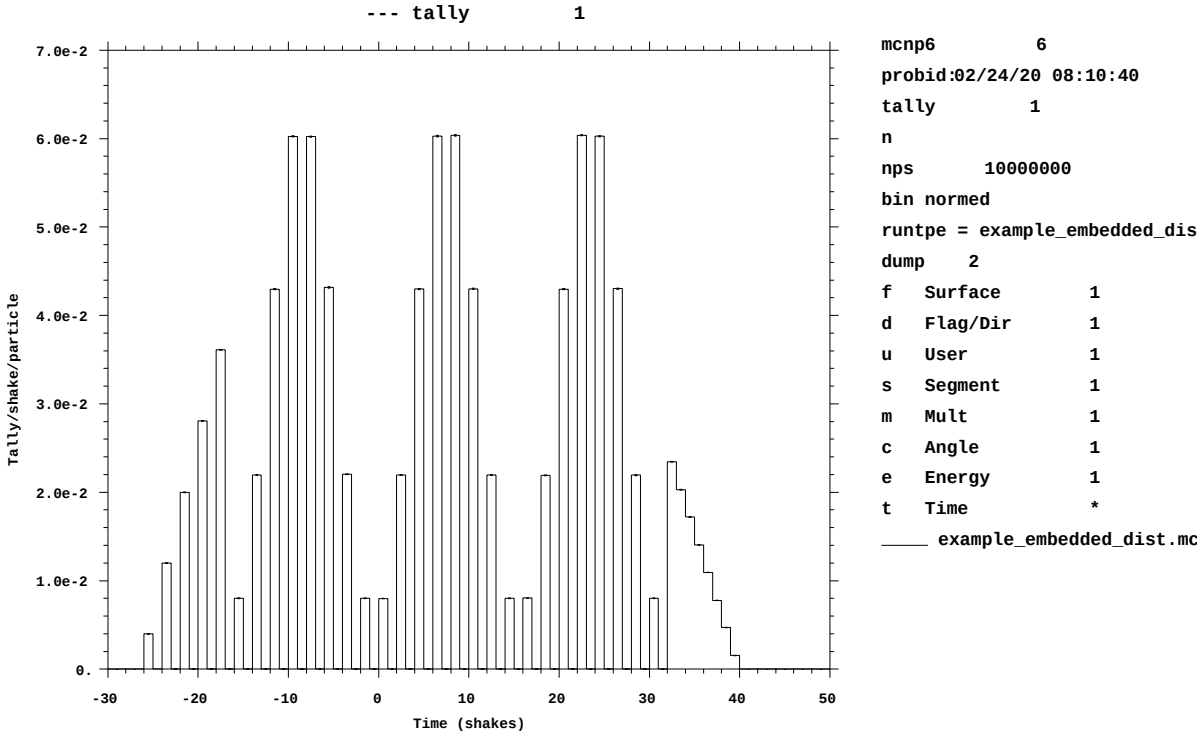


Figure 5.10: MCPLOT plot of tally from −30 to 50 shakes.

5.8.6.24 Example 23

```
si1  L   sp    sf      n
sp1  W  -10   -15      3e9
```

The spontaneous photon source will look to cell 10 and use the material and volume to calculate the overall activity that will be substituted into the SP1 distribution. Correspondingly, the SF source will look to the material and volume in cell 15 for the intensity of the spontaneous fission source. Note that a +SF normalizes per spontaneous fission neutron, a -SF normalizes per spontaneous fission. The neutron source is unchanged. After the overall activity is computed, the source distribution normalization will be done as described above and the weight adjustment value passed into the weight parameter.

5.8.6.25 Example 24

```
sdef  par=d1 wgt=264
si1   L   sf
sp1   w  -35
```

If the cell specified in the SP*n* W option is a lattice cell, then the code may not know the correct volume for this cell. If the user does not wish to correct the volume using the VOL card or cell keyword, a WGT keyword can be used with the source as a multiplicative factor. In this example, the spontaneous fission source is weighted by the activity from cell 35, which has been duplicated 264 times in the geometry. The final source weight will be the activity from cell 35 multiplied by 264.

5.8.7 SC: Source Comment

Data-card Form: *SC*n comment

n	the distribution number such that $1 \leq n \leq 999$
comment	user-supplied text describing the source.

Default: No comment.

Details:

1. The comment is printed as part of the header of distribution *n* in the source distribution table and in the source distribution frequency table. The & continuation symbol is considered to be part of the comment, not a continuation command.

5.8.8 SSW: Surface Source Write

This card is used to write a surface source file or KCODE fission volume source file for use in a subsequent MCNP calculation. Include enough geometry beyond the specified surfaces to account for albedo effects.

During execution, surface source information is written to the scratch file **WXXA**. Upon normal completion, **WXXA** becomes **WSSA**. If the calculation terminates abnormally, the **WXXA** file will appear instead of **WSSA** and must be saved along with the runtape file. The calculation must be continued for at least one more history. At the subsequent normal termination, **WXXA** disappears and the correct surface source file **WSSA** is properly written.

Data-card Form: SSW s_1 s_2 (c_1 ... c_J) s_3 ... s_K keyword = *value(s)*...

s_k	Problem surface number, with the appropriate sign to indicate sense of inward or outward particle direction, for which particle-crossing information is to be written to the surface source file **WSSA**. Macrobody facets are allowed.
c_j	Problem cell number. A positive entry denotes a cell the particle is entering. A negative entry specifies a cell that particle is leaving. This option provides a means to include only a portion of tracks crossing a certain surface (1, 2).
SYM = *value*	Symmetry option flag. If
	SYM = 0 — no symmetry assumed. (DEFAULT)
	SYM = 1 — spherical symmetry assumed. The list of problem surface numbers must contain only one surface and it must be a sphere (3).
	SYM = 2 — write particles to a surface bidirectionally. Otherwise, only particles going out of a positive surface and into a negative surface are recorded.
PTY = $\mathscr{P}_1$ $\mathscr{P}_2$ ...	Controls tracks to record. If PTY is absent, record all tracks for all particle types. (DEFAULT) Each $\mathscr{P}_i$ entry is a particle type selected from those listed in Table 4.3.
CEL = cf_1 cf_2 ...	List of names of all the cells from which KCODE fission source neutrons are to be written, active cycles only (4, 5).

Default: SYM = 0; no PTY keyword (record tracks for all particle types)

Use: Optional.

Details:

1. The SSW card allows a list of one or more cell names, positive or negative, after any of the surface names. The list of cell names must be enclosed in parentheses. If the list of cells is absent, any track that crosses the surface in the "correct direction" (as specified by the positive or negative sign on the surface number) will be recorded. If the list of cells is present, a track will be recorded if it crosses the surface in the correct direction and is either entering a cell in the list whose entry is positive or leaving a cell in the list whose entry is negative.

(2) Problem cell numbers, c_j, cannot include chain information; i.e., all cells listed must be at the lowest level. Lattice cells should not be listed because in most cases other cells are filled into a lattice cell. In the rare case that a lattice cell is filled with itself, simply list the lattice cell without any reference to a specific element.

(3) If the SYM = 1 option is used, the geometry inside the surface must be spherically symmetric and the materials must be symmetric. This symmetric situation only occurs rarely and it is the responsibility of the user to determine whether SYM = 1 is appropriate for the problem. If the SYM = 1 option is invoked, fewer words per particle need to be written to the surface source file and certain biasing options become available when reading the surface source file. The SYM = 1 option cannot be used if CEL is specified.

(4) Fission volume sources from a KCODE calculation can be written from active cycles only. The fission neutrons and prompt photons can then be transported in a subsequent calculation using the SSR surface source read fixed-source capability. In a KCODE criticality calculation the fission neutron sources and prompt photons produced from fission during each cycle are written to the **WSSA** surface source file if the SSW card has the CEL keyword followed by the names of all the cells from which fission source neutrons are to be written. Particles crossing specified surfaces can also be written by specifying s_k.

(5) Fission neutrons and photons written to the surface source file in a KCODE calculation can be used as a volume-distributed source in a subsequent calculation. A NONU card should be used so that fission neutrons and photons are not counted twice. Generally a TOTNU card is not required. Total $\overline{\nu}$ is the default for both KCODE and non-KCODE sources. Prompt $\overline{\nu}$ may be invoked by specifying TOTNU NO.

5.8.8.1 Example 1

```
SSW    4  -7  19 (45  -46)  16  -83 (49)
```

A track that crosses surface 19 in the correct direction will be recorded only if it is either entering cell 45 or leaving cell 46. A track that crosses surface 83 in the correct direction will be recorded only if it is entering cell 49. A track that crosses surface 4, 7, or 16 in the correct direction will be recorded regardless of what cells it happens to be leaving or entering.

5.8.8.2 Example 2

```
SSW    1   2 (3  4)   CEL 8 9
```

A track that crosses surface 2 in the correct direction will be recorded only if it enters cell 3 or 4. A track crossing surface 1 in the correct direction always will be recorded. Particles created from fission events in cells 8 and 9 will be recorded.

5.8.9 SSR: Surface Source Read

This card is used to read a surface source file or KCODE fission volume source file that was created in a previous MCNP calculation. The file **WSSA** must have previously been created using the SSW card; the file must be renamed to **RSSA** before it can be read by the SSR feature.

The number of particle histories reported in the output file for an SSR calculation is related to the number written to the **WSSA** file during the SSW procedure, so that proper normalization is preserved. However, a user may specify a different value on the NPS card in the SSR input file than that used in the initial SSW calculation. If the value of the *npp* parameter of the NPS card is smaller than that used in the initial calculation, an appropriate ratio of tracks will be rejected. If the *npp* value is larger than that of the initial calculation, an appropriate duplication of tracks will be sampled. For example, if the SSW calculation used an *npp* value of 100 and the SSR calculation uses an *npp* of 200, then every track is duplicated, each with a different random number seed and each with half the original weight. Note that a larger value of *npp* on the SSR calculation will indeed lower the tally errors until the weight variance contained on the **RSSA** file dominates. Therefore, a user should maximize the number of tracks on the **RSSA** file. Because the *npp* value can readjust particle weights as described above, some variance reduction parameters (e.g., weight-window bounds) may need to be renormalized for SSR applications.

The problem summary tables for a surface source problem represent the weights of the particles read from the **RSSA** file, not the weights in the original problem that wrote the surface source. To understand the resultant `Problem Summary Tables` for an SSR problem, consider the following example with two calculations in sequence, first:

```
MODE N E
SSW         $ neutrons and electrons written to WSSA file
```

followed by

```
MODE N P E
SSR         $ no photons available on RSSA to read
```

The weight creation and loss columns for all particles are normalized by the number of histories run in the problem. For this example, the neutron and electron average energies are determined by normalizing by the respective starting source weights from the **RSSA** file. Because no photons were available to be read, the photon summary table average energies will be normalized by the first particle source weight from **RSSA** in the problem, where neutrons have first priority (as in this example), then photons, then electrons, etc.

For the general SSR problem, one or more particle types will have source weights. The average energies in a particle Problem Summary Table are obtained in the following order: 1) if source particles are read from the **RSSA** file, then the average energies are determined by normalizing by the starting source weight; else 2) the first particle type with source weight will be used for obtaining average summary table energies.

Any variance-reduction technique that requires the input of normalized weight parameters (e.g., weight-window bounds, negative entries on the DD card, etc.) may need to be renormalized for SSW/SSR applications. Consider the following observations and comments:

1. In general, weight-window bounds generated in a SSW calculation are not useful in the SSR calculation, unless the tally identified on the WWG card of the SSW calculation is the same as that desired for the SSR calculation and plenty of tracks contributed to that tally during the SSW calculation.
2. A window generated in an SSR calculation will likely have to be renormalized in subsequent runs that use those windows, unless the value on the NPS card remains unchanged. If the value on the NPS card is changed, the `WGT` keyword on the SSR card can be used to renormalize the source weights to ensure weights are within the window in the source region. Whenever the `WGT` keyword is used in this fashion, tallies must be properly normalized by using this value on the SD card or the inverse of this value as a multiplier on the FM card.

Data-card Form: `SSR keyword=value(s)...`

`OLD` = s_1 s_2 ... s_K	List of K problem surface numbers that are a subset of the surfaces on the SSW card that created the file **WSSA**, now called **RSSA**. Negative entries are not allowed as filtering is not available based on crossing direction. A positive value (as on the SSW card) simply means to accept all tracks that have crossed that surface regardless of direction. (DEFAULT: All surfaces in the original calculation.) Restriction: Macrobody surfaces are allowed.
`CEL` = c_1 c_2 ... c_K	List of K cells numbers that represent a subset of the cells on the SSW card that created the file **WSSA**, now called **RSSA**. This subset specifies which fission cells to accept of those from the KCODE calculation that wrote the **RSSA** file (1), (DEFAULT: All cells in the original calculation.)
`NEW` = sa_1 sa_2 ... sa_K sb_1 sb_2 ... sb_K sm_1 sm_2 ... sm_K	Problem surface numbers on which the surface source is to start particles in this run. The K entries may be repeated to start the surface source in multiple (m) transformed locations. In other words, for $m = 1$, each particle written from surface s_k in the `OLD` list will start on surface $s1_k$. For $m = 2$, each particle written on surface s_k in the `OLD` list will start on surface $s2_k$, etc. See the `TR` keyword below. (DEFAULT: Surfaces in the `OLD` list.)
`PTY` = $\mathscr{P}_1$ $\mathscr{P}_2$ ...	A blank-delimited list of particle types for which the tracks are to be read. If the `PTY` keyword is absent, read all tracks for all particle types in the problem (2, 3). (DEFAULT: `PTY` absent.)
`COL`	Collision option flag. If

`COL` = `-1`	start from the surface source file only those particles that came directly from the source without a collision.
`COL` = `1`	start from the surface source file only those particles that had collisions before crossing the recording surface.
`COL` = `0`	start particles without regard to collisions. (DEFAULT)

`WGT`	Each particle weight is multiplied by the constant `WGT` as it is accepted for transport. (DEFAULT: `WGT` = `1`)
`TR` = n or `TR` = `D`n	Transformation number, n. Track positions and velocities are transformed from the auxiliary coordinate system (the coordinate system of the problem that wrote the surface source file) into the coordinate system of the current problem, using the transformation on the TR card, which must be present in the MCNP input file of the current problem (4). Distribution number, `D`n, where $1 \leq n \leq 999$. Distribution number for a set of SI, SP, and SB cards. If the surface source is transformed into several locations, the SI card lists the transformation numbers and the SP and SB cards give the probabilities and bias of each transformation, respectively (5). (DEFAULT: no transformation)
`PSC` = c	A non-negative constant that is used in an approximation to the PSC evaluation for the probability of the surface source emitting a particle into a specified angle relative to the surface normal (6).

The following four keywords are used only with spherically symmetric surface sources, that is, sources generates with SYM = 1 on the SSW card.

AXS = *u v w*	Direction cosines that define an axis through the center of the surface sphere in the auxiliary (original) coordinate system. This is the reference vector for EXT. (DEFAULT: No axis)
EXT = D*n*	Distribution number ($1 \leq n \leq 999$) (SI, SP, and SB cards) that will bias the sampling of the cosine of the angle between the direction AXS and the vector from the center of the sphere to the starting point on the sphere surface. (DEFAULT: No position bias)
POA = *c*	Particles with a polar angle cosine relative to the source surface normal that falls between 1 and *c* will be accepted for transport. All others are disregarded and no weight adjustment is made. (DEFAULT: POA = 0)
BCW = *r zb ze*	All particles with acceptable polar angles relative to the surface normal are started so that they will pass through a cylindrical window of radius r, starting at *zb* from the center of the source sphere and ending at *ze* from the center. The axis of the cylinder is parallel to the z axis of the auxiliary (original) coordinate system and contains the center of the source sphere. The weight of each source particle is adjusted to compensate for this biasing of position and direction. (DEFAULT: No cylindrical window) Restriction: $0 < zb < ze$

Use: Required for surface source problems.

Details:

1. Problem cell numbers, c_k, cannot include chain information; i.e., all cells listed must be at the lowest level. When a source point is kept for transport, the code determines the cell(s) for all higher levels in the geometry, based on its absolute location (i.e., (x, y, z) position).
2. By default, all particle types defined with the MODE card are read from the **RSSA** file if available. Particle types not defined with the MODE card are rejected without weight adjustment. Particle types can be selected from the **RSSA** file using the PTY keyword.
3. When heavy ions are specified in the problem, the charge and mass for each heavy ion are stored in the surface source file, **WSSA**, and will be read back to reconstruct the proper source distribution.
4. For each surface s_k in the OLD list, a corresponding surface $s1_k$ must appear in the NEW list such that TR*n* transforms the coordinates of a particle written from s_k to be on surface $s1_k$ in the current problem. However, if the surfaces $s1_k$ are "dummy" surfaces not used in constructing the real geometry, then the transformed source will effectively be treated as a volume source not specifically defined to be on any surface.
5. If NEW is present with multiple ($m > 1$) transformed locations, then the distribution must specify exactly m transformations that properly represent the relationship of the $m \times K$ surfaces on the NEW list to the K surfaces on the OLD list. Otherwise, the NEW specification is ignored (if present) and the application of TR = D*n* is analogous to its use on the SDEF card. The source after transformation is treated as a volume source (surface number not defined); the cell for the source particle is determined after transformation. It may be wise not to place the transformed source exactly on a surface of the physical geometry (to avoid lost particles in some cases).

6 An exact treatment of point detectors or DXTRAN spheres with a surface source is not possible because the $p(\cos\theta)$ values required for the source contribution are not readily available. To use detectors or DXTRAN with a surface source, an approximate $p(\cos\theta)$ must be specified on the SSR card. The most common azimuthally symmetric approximation for an angular emission probability density function is given by

$$p(\cos\theta) = C_c(\cos\theta)^c\,,\ c \geq 0. \tag{5.36}$$

The PSC value entered is c, the power to which $p(\cos\theta)$ is raised. C_c is a normalization constant calculated in the MCNP code and θ is the angle between the direction vector to the point detector and the surface normal at the point where the particle is to be started. Because surface crossings are recorded in only one direction specified on the SSW card, the limits on $\mu = \cos(\theta)$ are always between 1 and 0. A PSC entry of zero specifies an isotropic angular distribution on the surface. An entry of 1 specifies a cosine angular distribution that produces an isotropic angular flux on the surface. In the case of a 1-D spherical surface source of radius R, a cosine distribution is adequate if the point detector or DXTRAN sphere is more than $4R$ away from the source.

Caution

Remember that the value entered for PSC is only an approximation. If the point detector or DXTRAN sphere is close to the source sphere and the approximation is poor, the answers will be wrong.

5.8.9.1 Example 1

Original calculation:

```
SSW   1   2   3
```

Current calculation:

```
SSR   OLD  3  2   NEW  6  7  12  13   TR D5   COL 1
SI5  L   4     5
SP5    0.4   0.6
SB5    0.3   0.7
```

Particles starting on surface 1 in the original calculation will not be started in the current calculation because surface 1 is absent from the list of OLD surface numbers. Particles recorded on surface 2 in the original calculation will be started on surfaces 7 and 13, and particles recorded on surface 3 in the original calculation will be started on surfaces 6 and 12, as prescribed by the mapping from the OLD to the NEW surface numbers. The COL keyword causes only particles that crossed surfaces 2 and 3 in the original problem after having undergone collisions to be started in the current problem. The TR entry indicates that distribution function 5 describes the required surface transformations. According to the SI5 card, surfaces 6 and 7 are related to surfaces 3 and 2, respectively, by transformation TR4; surfaces 12 and 13 are related to 3 and 2 by TR5. The physical probability of starting on surfaces 6 and 7 is 40% according to the SP5 card, and the physical probability of starting on surfaces 12 and 13 is 60%. The SB5 card causes the particles from surfaces 3 and 2 to be started on surfaces 6 and 7 30% of the time with weight multiplier 4/3 and to be started on surfaces 12 and 13 70% of the time with weight multiplier 6/7.

5.8.9.2 Example 2

Original calculation:

```
SSW      3   SYM 1
```

Current calculation:

```
SSR     AXS  0 0 1   EXT D99
SI99     -1    0.5    1
SP99    0.75     1
SB99    0.5    0.5
```

All particles written to surface 3 in the original problem will be started on surface 3 in the new problem, which must be exactly the same because no OLD, NEW, COL, or TR keywords are present. Because this is a spherically symmetric problem, indicated by the SYM 1 flag in the original calculation, the position on the sphere can be biased. It is biased in the z direction with a cone bias described by distribution 99.

5.8.9.3 Example 3

Original calculation:

```
SSW     2  4  6
```

Current calculation:

```
SSR     OLD 2  TR=D1  WGT 6.0
SI1  L  11 22 33
SP1      1  2  3
SB1      1  1  1
TR11     0  0 -3  1 0 0  0 1 0  0 0 1
TR22    -3  0  0  0 1 0  0 0 1  1 0 0
TR33     0 -3  0  0.707 0 0.707  0.707 0 -0.707  0 1 0
```

All particles written from surface 2 in the original problem will be accepted; those written from surfaces 4 and 6 will be rejected. The distribution D1 will be sampled for each accepted particle and one of the transformations TR11, TR22, or TR33 will be applied. In this case, the particle current across surface 2 in the original problem will be applied as three intersecting beams in the x, y, and z directions. The relative intensities are $2:3:1$ respectively, but the sampling rate is the same in all three directions through use of the SB card.

5.8.10 KCODE: Criticality Source

The KCODE card specifies the MCNP criticality source that is used for determining k_{eff}. The criticality source uses total fission $\overline{\nu}$ values unless overridden by a TOTNU NO card and applies only to neutron problems.

In a MODE N P problem, secondary photon production from neutrons is turned off during inactive cycles. SSW particles are not written during inactive cycles.

Fission sites for each cycle are those points generated by the previous cycle. For the initial cycle, fission sites can come from an **SRCTP** file from a similar geometry, from a KSRC card, or from a volume distribution specified by an SDEF card.

Since the mid-2000s, there have been many detailed studies on the theory and practice of performing Monte Carlo criticality calculations. These studies have resulted in a set of "best practices" for performing KCODE calculations with the MCNP code. Best practices are discussed in [294–300] and in older documents including [295–300]. To summarize these reports:

Convergence of the fission source shape should be assessed with plots of the Shannon entropy vs. cycle. To avoid bias from the renormalization of the fission source each cycle, it is very strongly recommended that at least 10,000 neutrons/cycle should be specified on the KCODE card, with even larger numbers for large reactor problems. The initial guess for the source distribution (via KSRC, **SRCTP**, or SDEF) should be a reasonable representation covering the fissionable regions of a problem.

Data-card Form: `KCODE nsrck rkk ikz kct msrk knrm mrkp kc8`

nsrck	Number of source histories per cycle (1). (DEFAULT: *nsrck* = 1000)
rkk	Initial guess for k_{eff} (2). (DEFAULT: *rkk* = 1.0)
ikz	Number of cycles to be skipped before beginning tally accumulation. (DEFAULT: *ikz* = 30)
kct	Total number of cycles to be done. If *kct* = 0, never terminate on the number of cycles, but terminate on time [§3.2.5.6]. (DEFAULT: *kct* = ikz + 100)
msrk	Number of source points for which to allocate storage (3). (DEFAULT: *msrk* = maximum of 4500 or 2 × *nsrck*)
knrm	Controls normalization of tallies. If
	knrm = 0 normalize tallies by weight. (DEFAULT)
	knrm = 1 normalize tallies by number of histories.
mrkp	Maximum number of cycle values on **MCTAL** or **RUNTPE** files. (DEFAULT: *mrkp* = 6500)
kc8	Controls the number of cycles over which summary and tally information are averaged. If
	kc8 = 0 average over all cycles. (4)
	kc8 = 1 average over active cycles only. (DEFAULT)

Use: Required for criticality calculations.

Details:

(1) The default approach is to allow the histories per cycle to fluctuate around this value from generation to generation. If any tallies are performed with batch statistics (such as FMESH with the tally = batch option), the number of source particles in the fission bank will be resampled at each generation to precisely

nsrck particles to ensure the validity of the statistics. This will change the random number sequence but will yield statistically equivalent results.

2. If in the first cycle the source being generated overruns the current source, the initial guess (*rkk*) is probably too low. The code then proceeds to print a comment, continues without writing a new source, calculates k'_{eff}, reads the initial source back in, and begins the problem using k'_{eff} instead of *rkk*. If the generated source again overruns the current source after the first cycle, the calculation terminates and either a better initial guess (*rkk*) or more source space (*msrk*) should be specified on the next try.

3. If an **SRCTP** file with a larger value of *msrk* is read for the initial source, the larger value is used.

4. Setting the parameter *kc8* to zero causes tallies and summary table information to be for both active and inactive cycles and should not be used. Setting *kc8* = 0 also results in strange **MCTAL** file normalization, as these are normalized by active cycles.

5.8.11 KSRC: Criticality Source Points

This card contains up to *nsrck* (x, y, z) triplets that are locations of initial source points for a KCODE calculation. At least one point must be in a cell containing fissile material and points must be away from cell boundaries. It is not necessary to input all *nsrck* coordinate points. The MCNP code will start approximately *nsrck*/(number of points) particles at each point. Usually one point in each fissile region is adequate, because the MCNP code will quickly calculate and use the new fission source distribution. The energy of each particle in the initial source is sampled from a Watt fission spectrum hardwired into the MCNP code, with $a = 0.965$ MeV, $b = 2.29$ MeV^{-1}.

A **SRCTP** file from a previous criticality calculation can be used instead of a KSRC card. If the current problem has a lot in common with the previous problem, using the **SRCTP** file may save some computer time. Even if the problems are quite different, the **SRCTP** file may still be usable if some of the points in the **SRCTP** file are in cells containing fissile material in the current problem. Points in void or zero importance cells will be deleted. The number of particles actually started at each point will be such as to produce approximately *nsrck* initial source particles.

An SDEF card also can be used to sample initial source points in fissile material regions. The SDEF card parameters applicable to volume sampling can be used: CEL, POS, RAD, EXT, AXS, X, Y, Z; and CCC, ERG, and EFF. If a uniform volume distribution is chosen, the early values of k_{eff} will likely be low because too many particles are put near where they can escape, just the opposite of the usual situation with the KSRC card. Do not change the default value of WGT for a KCODE calculation.

Data-card Form: KSRC x_1 y_1 z_1 x_2 y_2 z_2 ... x_K y_K z_K

x_k y_k z_k	the locations of the initial source points.

Default: None. If this card is absent, an **SRCTP** source file or SDEF card must be supplied to provide initial source points for a criticality calculation.

Use: Optional card for use with criticality calculations.

5.8.12 KOPTS: Criticality Calculations Options

By invoking options on the KOPTS card, a number of features can be enabled. These mainly cluster into point-kinetics calculations and fission matrix acceleration.

For the point-kinetics parameters, the MCNP code can calculate the following parameters for criticality: the neutron generation time (Λ), the effective delayed neutron fraction (β_{eff}), and Rossi-α. The MCNP code computes the point-kinetics parameters in a forward calculation with only the existing random walks by breaking the active cycles of a KCODE calculation into sequential blocks of fission generations. For best results of the KOPTS card, the system should be as near critical ($k_{\text{eff}} = 1$) as possible.

For the fission matrix acceleration, k-eigenvalue problems can be accelerated by computing the eigenvalues of the fission matrix and weighting the inactive cycle source distribution by these eigenvalues. This can substantially improve the rate of convergence on a wide variety of problems [301]. In addition, it can be used to determine if a k-eigenvalue problem is poorly converged using a variety of statistical comparisons.

The fission matrix requires a mesh in order to operate. In addition, the statistical tests require the Shannon entropy mesh to match as well. There are three ways to give the MCNP code a mesh for this purpose. The first is to explicitly specify one on the HSRC card. The second is to set the FMATNX, FMATNY, and FMATNZ values, in which the extent of the bounding box is computed from the end-of-first-batch fission source distribution and subdivided using these values. Finally, if neither the HSRC card or the FMATN∗ values are specified, the MCNP code will generate a mesh by finding the bounding box from the end-of-first-batch fission source distribution and subdividing it by the fission-to-fission mean free path in each direction. In all cases, if source particles are found outside this mesh, it will be automatically expanded.

Data-card Form: KOPTS keyword = *value(s)...*

BLOCKSIZE = *ncy*	Controls the number of cycles in every outer iteration. Number of cycles, *ncy*, in blocks for adjoint weighting (1, 2, 3). (DEFAULT: *ncy* = 10) Restriction: $n \geq 2$	
KINETICS = *value*	If	
	KINETICS = YES	calculate point-kinetics parameters.
	KINETICS = NO	do not calculate point-kinetics parameters. (DEFAULT)
PRECURSOR = *value*	If	
	PRECURSOR = YES	calculate detailed precursor information.
	PRECURSOR = NO	do not calculate detailed precursor information (4). (DEFAULT)
KSENTAL = *fileopt*	Select format of sensitivity profiles output file, **KSENTAL** (5). If	
	KSENTAL = MCTAL	write the sensitivity profiles in a **MCTAL**-like file, from which the profiles may be plotted using **MCPLOT** (6).
	no format is specified	print no file. (DEFAULT)
FMAT = *value*		
	FMAT = YES	compute the fission matrix. Statistical tests on the convergence of the fission matrix, the convergence of the Shannon entropy, and the distributions of both relative to each other will be performed and reported to the user. This will include information on whether or not the statistical tests indicate that the problem is converged, as well as possible undersampling issues. For those interested in analyzing the resulting fission matrix, it is available

as a 0-indexed compressed-sparse-row (CSR) matrix in the results section of the runtape (see §D.5).

`FMAT` = `NO` — do not compute the fission matrix. (DEFAULT)

`FMATCONVRG` = *value*

`FMATCONVRG` = `YES` — the `ikz` option of `KCODE` will be ignored and instead, the statistical tests described above will be used to determine when to enable active cycles.

`FMATCONVRG` = `NO` — do not use the fission matrix to determine convergence. (DEFAULT)

`FMATACCEL` = *value*

`FMATACCEL` = `YES` — the eigenvalues of the fission matrix will be used to weight the importances of the fission sources in the problem during inactive cycles. This can be used to converge difficult problems more quickly. This option operates better with more particles per batch, as the matrix fills out quicker and has lower variance.

`FMATACCEL` = `NO` — do not use the fission matrix to determine convergence. (DEFAULT)

`FMATSRC` = *value*

`FMATSRC` = `YES` or `AUTO` — sample source particles uniformly from within the fission matrix mesh. `KSRC` and `SDEF` are ignored. Requires a mesh to be explicitly input on an `HSRC` card.

`FMATSRC` = `NO` — do not automatically generate a source. (DEFAULT)

`FMATSKIP` = *fmat_skip* — Skips this many batches before accumulating fission matrix tallies. (DEFAULT = 1)

`FMATNCYC` = *fmat_ncyc* — Batches between fission matrix solves. Larger values allow more fission matrix tallies to occur, improving stability, whereas smaller values can allow a problem to converge faster when used in acceleration. (DEFAULT = 10)

`FMATSPACE` = *fmat_space* — Initial number of nonzero elements to allocate the fission matrix to store. If exceeded, the MCNP code will dynamically reallocate the arrays. (DEFAULT = 100000000)

`FMATNX` = *fmat_nx* — Ignored with an `HSRC` card. If not zero, compute the x-axis mesh spacing by subdividing the end-of-first-batch fission source extent by this number. (DEFAULT = 0)

`FMATNY` = *fmat_ny* — Ignored with an `HSRC` card. If not zero, compute the y-axis mesh spacing by subdividing the end-of-first-batch fission source extent by this number. (DEFAULT = 0)

FMATNZ = *fmat_nz*	Ignored with an HSRC card. If not zero, compute the z-axis mesh spacing by subdividing the end-of-first-batch fission source extent by this number. (DEFAULT = 0)

Details:

1. Specification of BLOCKSIZE without setting KINETICS = YES is allowed, but the MCNP code will try to do adjoint weighting without tallying anything.
2. The default block size of 10 cycles produces results with sufficient accuracy for most problems of interest. Using fewer cycles per block introduces greater bias from truncation, but provides a more statistically efficient calculation. Larger blocks are more accurate, but the accuracy gained for larger block sizes is often small relative to the increased computer time required to preserve the statistical precision. Users are encouraged to check whether the selected block size is sufficient for their application by running a larger block size and comparing the results. For small, leakage-dominated systems, the block size can often be reduced to 5.
3. Because sensitivity coefficients (see the KSEN card) are adjoint weighted, they theoretically require infinitely many cycles before a tally may be performed. In practice, the default BLOCKSIZE value of 10 generations is usually more than sufficient to get accurate results.
4. If PRECURSOR = YES, then KINETICS must be set to YES.
5. The KSENTAL keyword requires there be at least one KSEN card specified in the MCNP input file.
6. The **MCTAL** format of the sensitivity profiles is much like the standard **MCTAL** file except that the symbols for bins have different meanings: F = cells (with 0 denoting all cells); D = unused; U = unused; S = isotopes; M = reaction numbers; C = cosine bins; E = energy bins; and T = incident energy bins (for fission χ or scattering laws). The tally plotter, **MCPLOT**, may be loaded to plot these results. The results should be normalized to be per unit lethargy with the "LETHARGY" option and plotted on a semi log-x scale for visually accurate area plots [§6.5].

5.8.12.1 Example 1

```
KOPTS   BLOCKSIZE=15  KINETICS=YES  PRECURSOR=YES
```

Both standard kinetics parameters and detailed precursor information are requested. Because the BLOCKSIZE value is not the default, we assume the user determined from empirical studies that 15 generations per block are needed for the application.

5.8.12.2 Example 2

```
KOPTS   FMAT=YES FMATCONVRG=YES
```

This will compute the fission matrix and use it to determine when to enable active cycles. The fission matrix mesh will be determined either from an HSRC card if present or from the first batch if not present.

5.8.12.3 Example 3

```
KOPTS   FMAT=YES FMATCONVRG=YES FMATACCEL=YES
```

This is identical to the previous example, except the fission matrix is used to accelerate the convergence of the problem.

5.8.12.4 Example 4

```
KOPTS   FMAT=YES FMATNX=5 FMATNY=5 FMATNZ=5
```

If there is no HSRC card, the fission matrix will be computed on a mesh that is initially $5 \times 5 \times 5$. The extent of the mesh is computed from the fission source distribution at the end of the first batch. The mesh will be extended as necessary throughout simulation.

5.8.13 HSRC: Mesh for Shannon Entropy of Fission Source Distribution

To assist users in assessing the convergence of the fission source distribution, the MCNP code computes a quantity called the Shannon entropy of the fission source distribution, H_{src}. To compute H_{src}, it is necessary to superimpose a 3-D grid on a problem encompassing all of the fissionable regions, and then to tally the number of fission sites in a cycle that fall into each of the grid boxes. The user may specify a particular grid to use in determining H_{src} by means of the HSRC input card. If the HSRC card is provided, users should use a small number of grid boxes (e.g., 5–10 in each of the x, y, and z directions), chosen according to the symmetry of the problem and layout of the fuel regions.

If the HSRC card is not provided, the MCNP code will automatically generate a mesh for use with Shannon entropy. The fission matrix capability on the KOPTS card will generate a mesh as described there that is usable for both the fission matrix and for entropy calculations. If the fission matrix capability is not enabled, the number of grid boxes will be determined by dividing the number of histories per cycle by 20 and then finding the nearest integer for each direction that will produce this number of equal-sized grid boxes, although not fewer than $4 \times 4 \times 4$ will be used. If the grid is automatically determined by the MCNP code, it will be expanded as necessary if fission source sites for a cycle fall outside of the grid. The grid size will not be reduced. If the grid is provided by the user using the HSRC card, then the MCNP code will issue warning messages either if 90% of the grid-boxes have zero scores for a cycle or if 25% of the fission source is located outside of the grid. Either of these messages is an indication that the user-supplied grid was poorly chosen for computing H_{src}. While H_{src} may not be computed reliably, there is no effect on k_{eff} or other tallies.

Data-card Form: HSRC n_x $x_{\min}$ $x_{\max}$ n_y $y_{\min}$ $y_{\max}$ n_z $z_{\min}$ $z_{\max}$

n_x	Number of mesh intervals in x direction, $n_x > 0$.
$x_{\min}$	Minimum x value for mesh.
$x_{\max}$	Maximum x value for mesh.
n_y	Number of mesh intervals in y direction, $n_y > 0$.
$y_{\min}$	Minimum y value for mesh.

y_{max}	Maximum y value for mesh.
n_z	Number of mesh intervals in z direction, $n_z > 0$.
z_{min}	Minimum z value for mesh.
z_{max}	Maximum z value for mesh.

Default: None. If this card is absent, if fewer than nine entries are supplied, or if $n_x \times n_y \times n_z \leq 0$, the MCNP code will automatically create a mesh that encloses all of the fission source sites in a cycle. This automatic mesh will be expanded if necessary on later cycles. The minimum number of mesh cells for the automatic mesh is $4 \times 4 \times 4$. If the HSRC card is supplied, one or more intervals may be specified for each of the x, y, and z directions.

Use: Optional card to specify the mesh for computing Shannon entropy of the fission source distribution in criticality calculations.

5.8.14 BURN: Depletion/Burnup (KCODE Problems Only)

Requirement: The **CINDER.dat** library file contains decay, fission yield, and 63-group cross-section data not calculated by the MCNP code. This library file must be present and accessible by the MCNP code for the burnup capability to work properly. To be accessible, the **CINDER.dat** file must reside in either the working directory or the **DATAPATH**.

MCNP depletion is a linked process involving steady-state flux calculations in the MCNP code and nuclide depletion calculations in CINDER90. The MCNP code runs a steady-state calculation to determine the system eigenvalue, 63-group fluxes, energy-integrated reaction rates, fission multiplicity (ν), and recoverable energy per fission (Q values). CINDER90 then takes those MCNP-generated values and performs the depletion calculation to generate new atom densities for the next time step. The MCNP code takes these new atom densities and generates another set of fluxes and reaction rates. The process repeats itself until after the final time step specified by the user.

Steady-state particle transport in the MCNP code includes only those isotopes listed on the material cards, selected from a fission product tier (presented in Table 5.16), or produced by the isotope generator algorithm. This algorithm captures only the daughter reactions and a few other residual reactions of the isotopes specified on the materials card; not the entire isotope decay chain. These daughter products are depicted in Fig. 5.11, which provides the relative locations of the products of various nuclear processes on the Chart of the Nuclides. To track the buildup of additional decay-chain isotopes in the transport calculation, the code adds the isotopes must be listed on the material (M) card. If decay-chain isotopes of interest are not initially present, the user must add these nuclides to the material card (M) with low atomic/weight fraction values (f_i) (e.g., 10^{-36}).

Table 5.16: Fission Product Content Within Each Burnup Tier

Tier 1	Tier 2	Tier 3
		^{69}Ga ^{71}Ga
		^{70}Ge ^{72}Ge ^{73}Ge ^{74}Ge ^{76}Ge
	^{74}As ^{75}As	^{74}As ^{75}As
		^{74}Se ^{76}Se ^{77}Se ^{78}Se ^{79}Se ^{80}Se ^{82}Se
	^{79}Br ^{81}Br	^{79}Br ^{81}Br
	^{78}Kr ^{80}Kr ^{82}Kr ^{83}Kr ^{84}Kr ^{86}Kr	^{78}Kr ^{80}Kr ^{82}Kr ^{83}Kr ^{84}Kr ^{85}Kr ^{86}Kr

continued on next page...

Table 5.16, continued

Tier 1	Tier 2	Tier 3
	^{85}Rb ^{87}Rb	^{85}Rb ^{86}Rb ^{87}Rb
		^{84}Sr ^{86}Sr ^{87}Sr ^{88}Sr ^{89}Sr ^{90}Sr
	^{89}Y	^{89}Y ^{90}Y ^{91}Y
^{93}Zr	^{90}Zr ^{91}Zr ^{92}Zr ^{93}Zr ^{94}Zr ^{96}Zr	^{90}Zr ^{91}Zr ^{92}Zr ^{93}Zr ^{94}Zr ^{95}Zr ^{96}Zr
	^{93}Nb	^{93}Nb ^{94}Nb ^{95}Nb
^{95}Mo	^{95}Mo	^{92}Mo ^{94}Mo ^{95}Mo ^{96}Mo ^{97}Mo ^{98}Mo ^{99}Mo ^{100}Mo
^{99}Tc	^{99}Tc	^{99}Tc
^{101}Ru	^{101}Ru ^{103}Ru	^{96}Ru ^{98}Ru ^{99}Ru ^{100}Ru ^{101}Ru ^{102}Ru ^{103}Ru ^{104}Ru ^{105}Ru ^{106}Ru
	^{103}Rh	^{103}Rh ^{105}Rh
	^{102}Pd ^{104}Pd ^{105}Pd ^{106}Pd ^{108}Pd ^{110}Pd	^{102}Pd ^{104}Pd ^{105}Pd ^{106}Pd ^{107}Pd ^{108}Pd ^{110}Pd
	^{107}Ag ^{109}Ag	^{107}Ag ^{109}Ag ^{111}Ag
	^{106}Cd ^{108}Cd ^{110}Cd ^{111}Cd ^{112}Cd ^{113}Cd	^{106}Cd ^{108}Cd ^{110}Cd ^{111}Cd ^{112}Cd ^{113}Cd ^{114}Cd ^{116}Cd
		^{113}In ^{115}In
	^{120}Sn	^{112}Sn ^{113}Sn ^{114}Sn ^{115}Sn ^{116}Sn ^{117}Sn ^{118}Sn ^{119}Sn ^{120}Sn ^{122}Sn ^{123}Sn ^{124}Sn ^{125}Sn ^{126}Sn
		^{121}Sb ^{123}Sb ^{124}Sb ^{125}Sb ^{126}Sb
		^{120}Te ^{122}Te ^{123}Te ^{124}Te ^{125}Te ^{126}Te ^{128}Te ^{130}Te ^{132}Te
	^{127}I ^{129}I ^{135}I	^{127}I ^{129}I ^{130}I ^{131}I ^{135}I
^{131}Xe ^{134}Xe	^{124}Xe ^{126}Xe ^{128}Xe ^{129}Xe ^{130}Xe ^{131}Xe ^{132}Xe ^{134}Xe ^{135}Xe ^{136}Xe	^{123}Xe ^{124}Xe ^{126}Xe ^{129}Xe ^{130}Xe ^{131}Xe ^{132}Xe ^{133}Xe ^{134}Xe ^{135}Xe ^{136}Xe
^{133}Cs ^{137}Cs	^{133}Cs ^{134}Cs ^{135}Cs ^{136}Cs ^{137}Cs	^{133}Cs ^{134}Cs ^{135}Cs ^{136}Cs ^{137}Cs
^{138}Ba	^{138}Ba	^{130}Ba ^{132}Ba ^{133}Ba ^{134}Ba ^{135}Ba ^{136}Ba ^{137}Ba ^{138}Ba ^{140}Ba
		^{138}La ^{139}La ^{140}La
		^{136}Ce ^{138}Ce ^{139}Ce ^{140}Ce ^{141}Ce ^{142}Ce ^{143}Ce ^{144}Ce
^{141}Pr	^{141}Pr	^{141}Pr ^{142}Pr ^{143}Pr
^{143}Nd ^{145}Nd	^{143}Nd ^{145}Nd ^{147}Nd ^{148}Nd	^{142}Nd ^{143}Nd ^{144}Nd ^{145}Nd ^{146}Nd ^{147}Nd ^{148}Nd ^{150}Nd
	^{147}Pm ^{148}Pm ^{149}Pm	^{147}Pm ^{148}Pm ^{149}Pm ^{151}Pm
	^{147}Sm ^{149}Sm ^{150}Sm ^{151}Sm ^{152}Sm	^{144}Sm ^{147}Sm ^{148}Sm ^{149}Sm ^{150}Sm ^{151}Sm ^{152}Sm ^{153}Sm ^{154}Sm
	^{151}Eu	^{151}Eu ^{152}Eu ^{153}Eu ^{154}Eu ^{155}Eu ^{156}Eu ^{157}Eu
	^{152}Gd ^{154}Gd ^{155}Gd ^{156}Gd ^{157}Gd ^{158}Gd ^{160}Gd	^{152}Gd ^{153}Gd ^{154}Gd ^{155}Gd ^{156}Gd ^{157}Gd ^{158}Gd ^{160}Gd
		^{159}Tb ^{160}Tb
		^{156}Dy ^{158}Dy ^{160}Dy ^{161}Dy ^{162}Dy ^{163}Dy ^{164}Dy
	^{165}Ho	^{165}Ho
		^{162}Er ^{164}Er ^{166}Er ^{167}Er ^{168}Er ^{170}Er

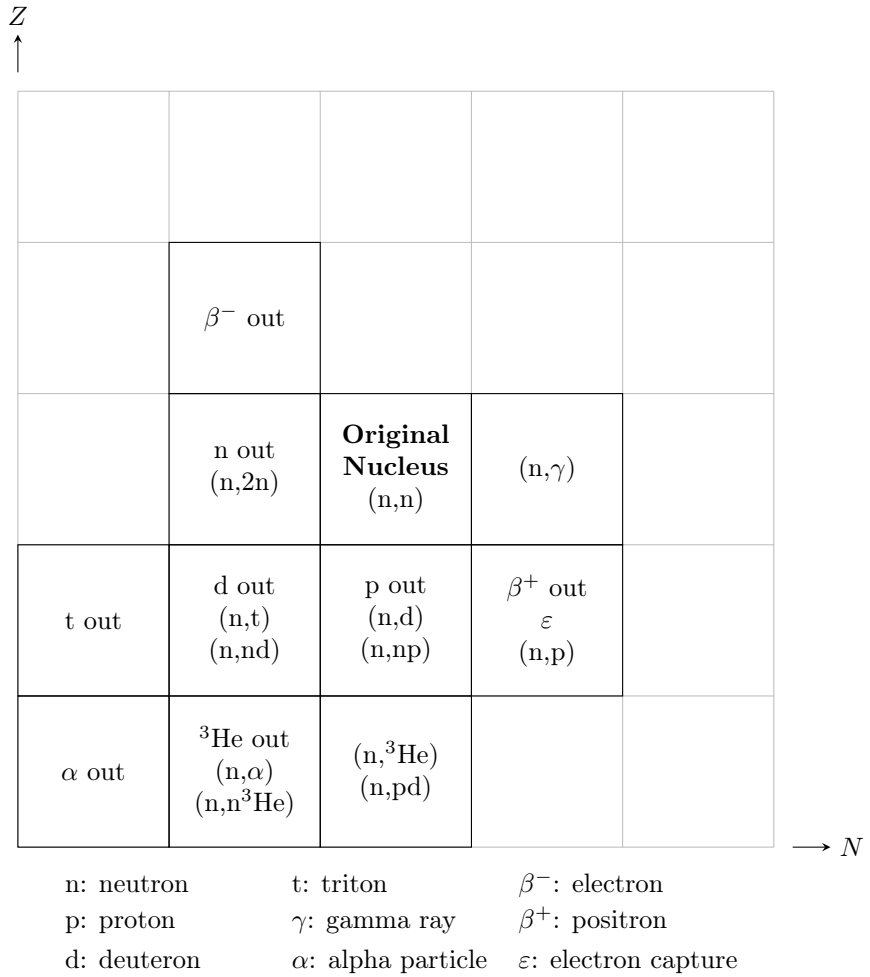


Figure 5.11: Nuclides selected for inclusion by the Isotope Generator Algorithm

When the information is not specified by the MCNP code, CINDER90 uses inherent intrinsic cross-section and decay data to track the time-dependent reactions of 3400 nuclides. The MCNP code can only track energy-integrated reaction-rate information for isotopes containing transport cross sections. For isotopes not containing transport cross-section information, the MCNP code calculates a 63-group flux that is sent to CINDER90. This flux data then is matched with a 63-group cross-section set inherent within CINDER90 to generate 63-group reaction rates. These resultant reaction rates are then energy integrated to determine the total reactions occurring.

Burnup is given in units of gigawatt days (GWD) per metric tons of uranium (MTU), where MTU is the sum of masses of isotopes containing $\geq$90 protons.

Data-card Form: `BURN keyword = value(s)...`

`TIME` = t_1 t_2 ...
Incremental time duration t_i for each successive burn step. Time unit is days. (1) (DEFAULT: A single one-day time step)

`PFRAC` = f_1 f_2 ...
Fraction f_i of total system power to be applied to the corresponding time step t_i. A power fraction of zero will perform decay without computing the corresponding reaction rates. Caution: If the number of entries in the `PFRAC` keyword is less than in the `TIME` keyword, the missing entries will be given a power fraction of zero. (DEFAULT: $f_i = 1$ for all t_i)

`POWER` = *pwr*
Total recoverable fission system power in MW. (DEFAULT: *pwr* = 1)

`MAT` = m_1 m_2 ...
List of materials to participate in the burnup calculation. Each ID corresponds to the ID on an `M` card. Positive m_i will be transmuted during the simulation and used to compute heating for power normalization. Negative m_i will be used only to compute heating for power normalization and will not be transmuted. (2, 9)

`OMIT` = m_1 n_1 `[omitted targets]`$_1$ m_2 n_2 `[omitted targets]`$_2$ ...
For material m_i, omit n_i targets [§1.2.2] as listed in `[omitted targets]`$_i$ from the transport calculation. All formats are supported. This is primarily used to remove nuclides that are part of the decay chain but do not have cross sections available. If m_i is −1, then this applies to all materials listed in the `MAT` keyword.

`AFMIN` = af_1 af_2
Atom fraction controls.

af_1	is the atom fraction below which nuclides are not tracked. If a nuclide atom fraction goes below this limit, the atom fraction is set to zero. (DEFAULT: $af_1 = 10^{-10}$)
af_2	is the transmutation chain convergence criteria used in CINDER90. (DEFAULT: $af_2 = 10^{-10}$)

`BOPT` = b_1 b_2 b_3
Burnup options.

b_1	is the Q value multiplier. (DEFAULT: $b_1 = 1$,)
b_2	is used to control the ordering and content of the output. It is the additive result of two integer values: $b_2 = I_1 + I_2$. The first value, I_1, selects among three tiers (see Table 5.16) of fission product content:

If $I_1 = 0$, include only Tier 1 fission products (DEFAULT).

If $I_1 = 10$, include Tier 2 fission products, which is more comprehensive than Tier 1.

If $I_1 = 20$, include Tier 3 fission products, which is more comprehensive than Tier 2. Tier 3 includes all fission products in the ENDF/B-VII.0 library that have CINDER90 yield information.

The second value I_2 selects among four ordering options:

If $I_2 = 1$, sort output inventory by decreasing mass (DEFAULT).

If $I_2 = 2$, sort output inventory by decreasing total activity.

If $I_2 = 3$, sort output inventory by decreasing specific activity.

If $I_2 = 4$, sort output inventory by increasing Z, followed by A, followed by S.

The sign of b_2 controls when to print. If positive, the output will be printed at the end of the calculation (DEFAULT). If negative, it will be printed at the end of each burn step.

b_3 allows the user to allow or disallow the use of high energy physics models.

If $b_3 = -1$, a fatal error will be printed if tabular data is unavailable for any nuclide (DEFAULT).

If $b_3 = 0$, the atom fraction of any data using a model is set to zero.

If $b_3 = 1$, use cross section models for nuclides not containing tabular data and then allow CINDER90 to calculate the 1-group cross section for these nuclides by convolving a 63-group flux tally with the CINDER90 63-group cross section data.

`MATVOL` $= v_1\ v_2\ \ldots$

Used to provide the volume of all cells containing a burn material in a repeated structure or lattice geometry (3). Each v_i entry is the volume of all cells containing burn material m_i. If `MATVOL` is used, all materials m_i must have a corresponding volume v_i.

`MATMOD` $= n_t\ ts_1$ `[material list]`$_1\ ts_2$ `[material list]`$_2\ \ldots$

Allows a user to make adjustments to the material abundances as a function of time. The input for this option is nested 3 layers deep. The three layers are time, material, and nuclide.

`MATMOD` $= n_t\ ts_1$ `[material list]`$_1\ ts_2$ `[material list]`$_2\ \ldots$

n_t The number of time steps in which abundance changes are specified.

$ts_j{=}i$ The ordinate of the time step, corresponding to the `TIME` keyword. If positive, the new abundances are used at times t_i and $t_{i+1/2}$. If negative, the new abundances are used at t_i and t_{i+1}. The value at $t_{i+1/2}$ is linearly interpolated.

`[material list]`$_1 = n\ m_1$ `[material info]`$_1\ m_2$ `[material info]`$_2\ \ldots$

n_m The number of materials to adjust during this time step.

m_i The material to adjust.

`[material info]`$_1 = n_z$ `target`$_1\ c_1$ `target`$_2\ c_2\ \ldots$

n_z The number of targets to adjust.

$\texttt{target}_i$	The target identifier [§1.2.2] to change the abundance of. All target formats are supported.
c_i	The new abundance of the target. If it is positive, it is interpreted as atom fractions or atom densities. If negative, it is interpreted as weight fractions or gram densities.

Note that if a nuclide is not listed, its abundance is not changed. As a result, all nuclides must be listed to reset a material. `SWAPB` is more practical for inserting fresh fuel into a problem.

`SWAPB` $= n_t$ ts_1 `[universe list]`$_1$ ts_2 `[universe list]`$_2$ ...

Allows a user to swap the contents of universes at the end of a given time step ts. This is useful for moving fresh fuel in and swapping assemblies during a refuel, for example (7, 8). Like `MATMOD`, this is a nested input, with layers of time and universe.

`SWAPB` $= n_t$ ts_1 `[universe list]`$_1$ ts_2 `[universe list]`$_2$ ...

n_t	The number of time steps in which universe fill changes are specified.
ts_j=i	The ordinate of the time step to make changes, corresponding to the `TIME` keyword. Changes are only made after a corrector step.

`[universe list]`$_1 = n_u$ u_1 `[fill spec]`$_1$ u_2 `[fill spec]`$_2$ ...

n_u	The number of universes to adjust during this time step.
u_i	The universe to adjust.
`[fill spec]`$_i$	The revised, fully specified, `FILL` card input, listing the universe numbers for each cell of the finite lattice, but omitting the range specification (6).

`NOSTATS` If present, this option will disable the computation and output of statistical parameters for reaction rates during depletion. In addition, only a single reaction-rate array is created per MPI rank, as opposed to the default duplication of the arrays for each OpenMP task. This will generally reduce performance but substantially reduce memory usage on large depletion problems. Without statistical information, users should take care to ensure their problem is converged. One should also consider using this option in concert with `DISABLE` `NUCLIDE_ACTIVITY_TABLE` for further memory reduction.

This feature will be eliminated in the next public release of the code in lieu of transparent and less-memory-intensive approaches, but it is provided here as a stop-gap measure in case memory overconsumption is encountered.

Use: The depletion/burnup capability is limited to criticality (`KCODE`) problems.

Details:

1. Burning with large time steps that encounter large flux-shape changes during the time step will lead to inaccurate calculations. Use time steps small enough to capture the flux-shape change accurately over time.

(2) For negative material numbers, m_i, specified on the MAT keyword, the recoverable energy per fission and neutrons per fission are computed for use in the power normalization procedure and the calculation of fission power fractions. A fatal error results if every material number is negative.

(3) To compute correctly isotopic masses and fluxes for burn materials, the volume of these materials must be either calculated by the MCNP code or provided by the user (on the VOL card or MATVOL keyword). For lattices or repeated structures, the MCNP code calculates the volume of each cell, but does not account for multiple occurrences of cell volumes. Therefore, if cells containing a burn material are repeated, then the volume calculated by the MCNP code will not represent the total volume of burn material and the user must provide the correct information on the MATVOL keyword.

(4) When using the MATMOD keyword, if ts_j is negative at t_i and the abundances of any of the altered isotopes at t_{i+1} is equal to the abundance set at t_i, then the abundances of the altered isotopes will be set to the value at t_i for t_i, $t_{i+1/2}$ and t_{i+1}. At $t_{i+3/2}$, the isotopes will undergo a normal depletion and the abundances will not be set to the value at t_{i+1}.

(5) When using the MATMOD keyword of the BURN card, if a burn material is set to have an abundance change at t_1, then the atom density of that isotope at $t_{1/2}$ is set to the initial value specified at t_0. This is only set for the initial midpoint time step; the rest of the calculation will follow the procedure described for the ts_j parameter.

(6) The ability to "swap" or redefine universes is limited to universes of the same level. The universe need not be actively in the geometry (i.e. the universe may be truncated out of view by the bounding surfaces and still be able to be swapped). Also, you cannot swap a universe that does not pre-exist in the geometry.

(7) At the beginning of the simulation, the code inserts a tiny quantity (an atom fraction of 10^{-37}) of nuclides the code expects to need transport reaction rates for into the transport material. These reaction rates are then used during the depletion process to compute transmutation as these nuclides are generated, which generally improves the initial predictor step accuracy. When one uses the SWAPB option to move material into the geometry after the first time step, the AFMIN option will truncate nuclide abundances below af_1, which by default will remove these nuclides from the material and prevent initial computation of these reaction rates. One can set af_1 to 10^{-38} or make the first step after fuel shuffling particularly short if this accuracy loss is a concern.

(8) As one shuffles material in the geometry, one must always ensure the volume of each material being irradiated is constant and corresponds to the val (individual cells) or MATVAL (lattices of cells) values. Otherwise, the normalization of tallies will be performed incorrectly. One can remove and insert a material throughout a simulation using SWAPB, but all of the material must be removed or inserted at the same time.

(9) Burnup is performed on a per-material basis. As a result, if one material is in multiple locations in the geometry, burnup will be performed based on the average irradiation of all locations. It is recommended to have unique materials whenever the neutron flux is expected to vary from location to location, even if the initial configuration is the same. For example, in a reactor with fuel rods, the following list is sorted in increasing accuracy: all fuel is given by one material, each assembly has a unique fuel material, each fuel rod has a unique material, each axial and radial discretization of the fuel rod has a unique material. This applies to all material being burned. Having many unique regions, however, increases the variance per region and the memory requirements for the simulation.

(10) Energy deposition in BURN is computed as $1.111 b_1 Q \Sigma_f \phi$. The factor 1.111 is a default estimate of the ratio of total recoverable energy from all reactions to the prompt fission energy and comes from [page 98 of 302]. This additionally includes delayed photon, delayed beta, and capture photons. This factor is not perfectly applicable to all problems due to compositional and spectral effects. One can adjust the value b_1 as necessary to help correct for these effects, but one should note that the factor 1.111 is still included.

The value of Q is hardcoded for 22 fissionable nuclides and includes fission fragment recoil, prompt neutron, and prompt photon energies. The fissionable nuclides with Q values are ^{232}Th, ^{233}Pa, ^{233}U through ^{240}U, ^{237}Np, ^{238}Pu through ^{243}Pu, ^{241}Am through ^{243}Am, ^{242}Cm, and ^{244}Cm. The Q values used

by the MCNP code are shown in PRINT Table 98 and come from a variety of sources such as ENDF/B-VI, ENDF/B-VII, JEFF 3.1, and expert evaluation (only in the case of those isotopes with no other source of data). The incoming neutron energy spectrum used for Q is thermal.

5.8.14.1 Example 1

Listing 5.44 gives an infinite pincell example, where materials 10, 30, and 40 are burned at 5 kW for 100 days and then 70 more days. Only material 10 contains fissionable actinides; therefore materials 30 and 40 experience transmutation only. The 2nd entry on the BOPT keyword sets the ordering of the output and selection of the fission product tier. Because the 1st digit of the 2nd entry is "1", the 2nd fission product tier will be used. Because the 2nd digit of the 2nd entry is "2", the order of the output isotope inventory will be based on high to low total activity. Because the 2nd BOPT keyword is negative, output will be given at the end of each burn step. Isotope inventories will be given for each individual burn material as well as the sum over all burn materials.

The *nlib* = 80c option is used to ensure that all neutron transport cross sections, including those brought in by the burnup process, are from one library. Since ENDF/B-VII.1 does not include all of the nuclides the code tries to add to transport, the OMIT option removes 14 nuclides that are not available.

Listing 5.44: example_burn_1.mcnp.inp.txt

```
BURN  TIME = 100 70
      MAT = 10 30 40
      POWER = 0.005
      PFRAC = 1.0 1.0
      BOPT = 1.0 -12 1
      OMIT = -1 14 C-12  C-13  C-14  N-16  O-18  F-18  Ne-20
                   Y-87  Y-88  Zr-89 Nb-91 Nb-92 Mo-91 Mo-93
M10   O-16   2.0
      U-235  0.0455
      U-238  0.9545
      nlib=80c
M20   He-4         -1.0
      nlib=80c
M30   Zr-90        -1.0
      nlib=80c
M40   H-1          4.7716e-2
      O-16         2.3858e-2
      B-10         3.6346e-6
      B-11         1.6226e-5
      nlib=80c
MT40  lwtr.20t
```

5.8.14.2 Example 2

Listing 5.45 is identical to Listing 5.44 but with three exceptions. First, after running at full power, a single decay step of 365 days is performed. Second, the second entry of BOPT is changed to "-22", indicating that Tier 3 fission products will be used instead of Tier 2. Finally, from material 4, ^{1}H and ^{16}O have been omitted so that only boron is transmuted.

Listing 5.45: example_burn_2.mcnp.inp.txt

```
BURN  TIME = 100 70 365
      MAT = 10 30 40
```

```
      POWER = 0.005
      PFRAC = 1.0 1.0 0.0
      BOPT = 1.0 -22 1
      OMIT = -1 14 C-12  C-13  C-14  N-16  O-18  F-18  Ne-20
                   Y-87  Y-88  Zr-89 Nb-91 Nb-92 Mo-91 Mo-93
```

5.8.14.3 Example 3

Listing 5.46 has three changes relative to Listing 5.44. First, the time steps have been changed to 15, 30, and 30 days. Second, the `AFMIN` keyword is used to set the minimum nuclide density to 10^{-20} and the convergence criteria to 10^{-12}. Third, the `MATMOD` keyword is used to adjust the boron abundance at time step 2. Here, ^{10}B is updated with a 6×10^{-5} atom fraction and ^{11}B is updated to a 24×10^{-5} atom fraction. All other materials and nuclides are untouched. Note that although material 40 has unnormalized nuclide fractions, the input to `MATMOD` is a relative fraction.

Listing 5.46: example_burn_3.mcnp.inp.txt

```
BURN  TIME = 15 30 30
      MAT = 10 30 40
      POWER = 0.005
      PFRAC = 1.0 1.0 1.0
      BOPT = 1.0 -12 1
      AFMIN = 1e-20 1e-12
      OMIT = -1 14 C-12  C-13  C-14  N-16  O-18  F-18  Ne-20
                   Y-87  Y-88  Zr-89 Nb-91 Nb-92 Mo-91 Mo-93
      MATMOD = 1
                 2 1
                   40 2 B-10 0.00006
```

5.8.14.4 Example 4

Listing 5.47 gives a 3×3 lattice of fuel pins where universes 1, 2, and 3 each contain a fuel rod, differing only in the material in the fuel. The fuel used is material 10, 20, and 30 respectively. Initially, only universe 1 and 2 are in the problem, in a checkerboard pattern with universe 1 in the top left corner. Material 3 is not present in the problem and is not initially being irradiated.

Using the `SWAPB` keyword at the end of time step 2, universe 2 is removed from the geometry and replaced with the fresh fuel in universe 3. Material 20 is now no longer being irradiated and decay calculations will be performed. At the end of time step 4, universe 3 is removed and replaced with universe 2 again. Material 30 is now no longer irradiated and will now decay.

The `MATVOL` keyword is required in this example order to provide the correct volumes to CINDER90. If `MATVOL` is missing, the *`vol`* option on the cell will be used, regardless of how many times that cell appears in the geometry. One must ensure the quantity of each material is constant as noted in (8).

Listing 5.47: example_burn_4.mcnp.inp.txt

```
BURN  TIME = 10 10 10 10 10
      MAT = 10 20 30
      MATVOL = 50.2655 40.2124 40.2124
      POWER = 0.030
      BOPT = 1.0 -12 1
      OMIT = -1 14 C-12  C-13  C-14  N-16  O-18  F-18  Ne-20
```

Table 5.17: Source Variables Required for each Source Particle

Code Source Variable	Variable Description
pbl%r%erg	the energy of the particle (MeV)
pbl%r%tme	the time when the particle started (shakes)
pbl%r%x, **pbl%r%y**, and **pbl%r%z**	the position of the particle
pbl%r%u, **pbl%r%v**, and **pbl%r%w**	the direction of the flight of the particle
pbl%i%ipt	the type of particle
pbl%r%wgt	the statistical weight of the particle
pbl%i%icl	the cell where the particle started
pbl%i%jsu	the surface where the particle started, or zero if the starting point is not on any surface

```
                Y-87  Y-88  Zr-89 Nb-91 Nb-92 Mo-91 Mo-93
     SWAPB = 2
             2 1
               4 1 3 1
                 3 1 3
                 1 3 1
             4 1
               4 1 2 1
                 2 1 2
```

5.8.15 Subroutines SOURCE and SRCDX

Users may write their own source subroutines to bypass the standard source capabilities. If no SDEF, SSR, or KCODE card is provided in the MCNP input file, then the MCNP code will look for a subroutine called **SOURCE**. This subroutine must be supplied by the user. In addition, if there are detectors or DXTRAN, the MCNP code also will require a **SRCDX** routine. [§10.3.4] contains an example of a **SOURCE** subroutine and [§10.3.5] discusses the **SRCDX** subroutine. The parameters that must be specified within the **SOURCE** subroutine are listed and defined in Table 5.17. Prior to calling subroutine **SOURCE**, isotropic direction cosines (u, v, w) (**pbl%r%u**, **pbl%r%v**, and **pbl%r%w**) are calculated and need not be specified if an isotropic distribution is desired.

Note that additional variables may have to be defined if there are point detectors or DXTRAN spheres in the problem. Also, **pbl%r%erg** has a different meaning in a special case. If there is a negative `igm` on the MGOPT card, which indicates a special electron-photon multigroup problem, `ERG` on the SDEF card is interpreted as an energy group number, an integer.

The SI, SP, and SB cards also can be used with the **SOURCE** subroutine, although modifications to other parts of the MCNP code may be required for proper initialization and to set up storage. A random number generator **RANG()** is available for use by **SOURCE** for generating random numbers between 0 and 1. Up to 200 numerical entries can be entered on each of the IDUM and RDUM cards for use by **SOURCE**. The IDUM entries must be integers and the RDUM entries floating point numbers.

If you are using a detector or DXTRAN and your source has an anisotropic angular distribution, you will need to supply an **SRCDX** subroutine to specify PSCs (i.e., probability of the surface source emitting a particle into a specified angle relative to the surface normal) for each detector or DXTRAN sphere.

There are unused variables stored in the particle bank that are reserved for the user. These are called **SPARE(M), M=1, MSPARE**, where **MSPARE** = 7. Depending on the application, you may need to reset them to 0 in **SOURCE** for each history; the MCNP code does not reset them.

5.8.15.1 Example 1

The **source.F90** subroutine given in Listing 5.48 is used to represent the SDEF card:

```
SDEF par=n erg=8 vec=1 0 0 dir 1 wgt=1 tme=0 pos 2 3 4
```

Listing 5.48: example_source_subroutine.f90.txt

```
subroutine source
  ! dummy subroutine.  Aborts job if source subroutine is
  ! missing. If nsr==USER_DEFINED_SOURCE, a subroutine
  ! source must be furnished by the user.

  ! At entrance, a random set of direction cosines, pbl%r%u,
  ! pbl%r%v, pbl%r%w has been defined

  ! .. Use Statements ..
  use mcnp_interfaces_mod, only : expirx
  use mcnp_debug
  use pblcom

  implicit none

  pbl%i%ipt = 1
  pbl%i%jsu = 0
  pbl%i%icl = 1
  pbl%r%x = 2
  pbl%r%y = 3
  pbl%r%z = 4
  pbl%r%u = 1
  pbl%r%v = 0
  pbl%r%w = 0
  pbl%r%erg = 8
  pbl%r%tme = 0
  pbl%r%wgt = 1

  ! call expirx(0,'source','you need a source subroutine.')
  return
end subroutine source
```

It is assumed that the source is in cell 6, which is the 1st cell number listed in the input. The **expirx** call must be commented out; otherwise the compiled source will still result in a message to the terminal of "bad trouble in subroutine source of mcrun you need a source subroutine."

5.9 Tally Specification-focused Data Cards

Tally cards are used to specify what type of information the user wants to gain from the Monte Carlo calculation. Options include such tallies as current across a surface, flux at a point, heating in a region, etc. This information is requested by the user by using a combination of cards described in this section. To obtain tally results, only the F card is required; the other tally cards provide various optional features.

The n associated with the tally-type specification is a user-chosen tally number $n \leq 99999999$; choices of n are discussed in the following section. When a choice of n is made for a particular tally type, any other input card used to refine that tally description (such as En for energy bins) is given the same value of n by the user.

Table 5.18: Tally Designators & Units

Mnemonic	Tally Description	Fn units	*Fn units
F1:$\mathscr{P}$	Current integrated over a surface	particles	MeV
F2:$\mathscr{P}$	Fluence averaged over a surface	particles/cm^2	MeV/cm^2
F4:$\mathscr{P}$	Fluence averaged over a cell	particles/cm^2	MeV/cm^2
F5a:$\mathscr{P}$	Fluence at a point or ring detector	particles/cm^2	MeV/cm^2
FIP5:$\mathscr{P}$	Array of point detectors for pinhole fluence image	particles/cm^2	MeV/cm^2
FIR5:$\mathscr{P}$	Array of point detectors for planar radiograph fluence image	particles/cm^2	MeV/cm^2
FIC5:$\mathscr{P}$	Array of point detectors for cylindrical radiograph fluence image	particles/cm^2	MeV/cm^2
F6:$\mathscr{P}$	Energy deposition averaged over a cell	MeV/g	jerks/g
+F6	Collision heating	MeV/g	N/A
F7:$\mathscr{P}$	Fission energy deposition averaged over a cell	MeV/g	jerks/g
F8:$\mathscr{P}$	Energy distribution of pulses created in a detector by radiation	pulses	MeV
+F8:$\mathscr{P}$	Charge deposition	charge	N/A

Much of the information on these cards is used to describe tally "bins," or subdivisions, of the tally space into discrete and contiguous increments such as cosine, energy, or time. Usually when the user subdivides a tally into bins, MCNP6 also can provide the total tally summed over appropriate bins (such as over all energy bins). Absence of any bin specification card results in one unbounded bin rather than one bin with a default bound. No information is printed about the limits on the unbounded bin.

If there are reflecting surfaces or periodic boundaries in the problem, the user may have to normalize the tallies in some special way. This can be done by setting the weight of the source particles or by using the FM or SD cards.

Printed with each tally bin is the relative error of the tally corresponding to one standard deviation. These errors cannot be believed reliable (hence neither can the tally itself) unless the error is fairly low. Results with errors greater than 50% are useless, those with errors between 20% and 50% can be believed to within a factor of a few, those with errors between 10% and 20% are questionable, and results with errors less than 10% are generally (but not always) reliable, except for point detectors. Detector results are generally reliable if their associated relative errors are below 5%. The tally fluctuation charts at the end of the output file base their results on the information from one specified bin of every tally. See the TF card. This bin also is used for the weight-window generator and is subject to ten statistical checks for tally convergence, including calculation of the variance of the variance (VOV). The VOV can be printed for all bins in a tally by using the DBCN card. A tally is considered to be converged with high confidence only when it passes all ten statistical checks.

5.9.1 F: Standard Tallies

MCNP6 offers an array of standard tallies to the user. These include particle current, particle flux (across a surface, in a cell, at a detector point), energy deposition, collision heating, fission energy deposition, pulse height, and charge deposition. All tallies are normalized to be per source particle unless a different normalization has been specified with the WGT keyword on the SDEF card, changed by the user with a **TALLYX** subroutine, or normalized by weight in a criticality (KCODE) calculation.

The tallies are identified by tally type and particle type as follows. Tallies are given the numbers 1, 2, 4, 5, 6, 7, 8 or increments of 10 thereof, and are given a particle designator $\mathscr{P}$, where $\mathscr{P}$ is chosen from Table 4.3. Thus you may have as many of any basic tally as you need, each with different energy bins, or flagging bins, or anything else. The designations F4:n, F14:n, F104:n, and F234:n are all legitimate neutron cell flux tallies; they could all be for the same cell(s) but with different energy or multiplier bins, for example. Similarly F5:p,

F15:p, and *F305:p are all photon point detector tallies. Having both an F1:n card and an F1:p card in the same MCNP input file is not allowed. The tally number may not exceed 99,999,999.

Several tally types allow multiple particles. For example, an energy deposition tally for both neutrons and gammas, F6:n,p, may be specified. In the case of collision heating, +F6 always applies to all particles in a problem; therefore this tally has no particle designator. For pulse-height tallies photons/electrons are a special case: F8:p,e is the same as F8:p and F8:e. Also, F8 tallies may have particle combinations such as F8:n,h.

Tally types 2, 4, and 5 are described as fluence tallies with the associated units. However, depending on the source units, these may also be fluence rate (i.e., flux) tallies with units of particles/$(\mathrm{cm}^2 \cdot \mathrm{s})$.

Tally types 1, 2, 4, and 5 are normally weight tallies; however, if the F card is flagged with an asterisk (for example, *F1:n), energy times weight will be tallied. The asterisk flagging also can be used on tally types 6 and 7 to change the units from MeV/g to jerks/g. No asterisk can be used in combination with the + on the +F8 or +F8 tallies. The asterisk on a tally type 8 converts from a pulse-height tally to an energy deposition tally. All of the units are shown in the Table 5.18.

Tally type 8 has many options. The standard F8 tally is a pulse-height tally and the energy bins are no longer the energies of scoring events, but rather the energy balance of all events in a history. In conjunction with the FT*8* card, the F8 tally can be an anti-coincidence pulse-height tally, a neutron coincidence capture tally, or a residual nuclei production tally. When flagged with an asterisk, *F8 becomes an energy deposition tally. In addition, F8 can be flagged with a plus (+) to convert it from an energy deposition tally (flagged with an asterisk) to a charge deposition tally. The +F8 tally is the negative particle weight for electrons and the positive weight for positrons. The +F8:e tally can be checked against an F1:e type surface tally with the FT*1*:e ELC option to tally charge.

Only the F2 surface flux tally requires the surface area. The area calculated is the total area of the surface that may bound several cells, not a portion of the surface that bounds only a particular cell. An exception to this statement occurs if one uses a repeated structures format to describe the tally bin [§5.9.1.5]. If you need only the segment of a surface, you might segment the full surface with the FS card and use the SD card to enter the appropriate values. You can also redefine the geometry as another solution to the problem. Similarly, tally types 4, 6, and 7 require the cell volume, which can be automatically calculated or supplied by the user via the VOL or SD cards. The limit on the total number of detectors and different tallies is given in Table 4.1. Note that a single type 5 tally may create more than one detector.

For any tally, if the tally label of the surface or cells in a given bin exceeds eleven characters, including spaces, an alphabetical or numerical designator is defined for printing convenience. The MCNP6-supplied designator will be printed with the tally output, e.g., "G is (1 2 3 4 5 6)". This labeling scheme is usually required for tallies over the union of a long list of surfaces or cells or with repeated structure tallies.

5.9.1.1 Surface and Cell Tallies (Tally Types 1, 2, 4, 6, and 7)

Simple Data-card Form: Fn:$\mathscr{P}$ *s1* ... *sK*
or
General Data-card Form: Fn:$\mathscr{P}$ *s1* (*s2* ... *s3*) (*s4* ... *s5*) *s6 s7* ... [T]

n	Tally number. Restriction: $n \leq 99999999$
$\mathscr{P}$	Particle designator (1).
sk	Problem number of surface or cell for tallying (2).
T	Total over specified surfaces for F1 tallies; average over specified surfaces or

cells for F2, F4, F6, and F7 tallies. (Optional) (3)

Use: In the simple form above, MCNP6 creates K surface or cell bins for the requested tally, listing the results separately for each surface or cell. In the more general form, a bin is created for each surface or cell listed separately and for each collection of surfaces or cells enclosed within a set of parentheses. Entries within parentheses also can appear separately or in other combinations. Parentheses indicate that the tally is for the union of the items within the parentheses. For unnormalized tallies (tally type 1), the union of tallies is a sum, but for normalized tallies (types 2, 4, 6, and 7), the union results in an average. See §5.9.1.5 for an explanation of the repeated structure and lattice tally format.

Details:

1. Tally type 7 allows $\mathscr{P}$ = n only.
2. Only surfaces that define cell boundaries and that are listed in a cell card description can be used on F1 and F2 tallies.
3. The symbol T entered on surface or cell F cards is shorthand for a region that is the union of all of the other entries on the card. A tally is made for the individual entries on the F card plus the union of all the entries. The entry is optional.
4. Surface flux tallies require an approximation when counting grazing contributions, that is, for contributions where the dot product of the particle direction and the surface normal are between −0.001 and 0.001 (the current default, new in MCNP6.2). The grazing angle cutoff can be reset using the 24th entry on the DBCN card; i.e., "DBCN 23J 0.1" changes the grazing angle cutoff from the MCNP6.2 value of ±0.001 to the historic MCNP value of ±0.1.

5.9.1.1.1 Aside: Surface Flux Tally (F2)

For particles grazing the surface, $1/|\mu|$ (where μ is the cosine of the angle that the particle track makes with the surface normal) is very large and the MCNP code approximates the surface flux estimator in order to satisfy the requirement of one central limit theorem. An unmodified surface flux estimator has an infinite variance, and thus confidence intervals could not be formed via the central limit theorem, because the central limit theorem requires a finite variance. For this reason, the MCNP code sets $|\mu| = 0.0005$ when $|\mu| < 0.001$; because of this approximation, the F2 tally is not an exact estimate of the surface flux. The grazing angle cutoff cosine can be changed using the 24th entry on the DBCN card.

While the numeric values may vary, this is the standard approximation used in Monte Carlo codes. This approximation is accurate when the angular flux is isotropic or linearly anisotropic with respect to μ on the surface and the limits of the flux integral with respect to μ are symmetric. However, these assumptions may become invalid on external surfaces or in other cases of one-way surface crossings; when exactly tangent crossing is not possible because of the geometry of the problem; or when cosine bins are used. Users should be especially careful in these cases. More details may be found in [303, 304].

5.9.1.1.2 Aside: Energy Deposition Tally (F6)

The energy deposition tallies in the MCNP code are fairly complicated, and require some explanation in order to ensure the correct result is extracted. See §2.5.3 for further details, as well as some rules of thumb for how to ensure the accuracy of your simulation.

5.9.1.1.3 Example 1

```
F2:N    1  3  6  T
```

This card specifies four neutron flux tallies, one across each of the surfaces 1, 3, and 6 and one which is the average of the flux across all three of the surfaces.

5.9.1.1.4 Example 2

```
F1:P  (1  2) (3  4  5) 6
```

This card provides three photon current tallies, one for the sum over surfaces 1 and 2; one for the sum over surfaces 3, 4, and 5; and one for surface 6 alone.

5.9.1.1.5 Example 3

```
F371:N    (1  2  3) (1  4) T
```

This card provides three neutron current tallies, one for the sum over surfaces 1, 2, and 3; one for the sum over surfaces 1 and 4; and one for the sum over surfaces 1, 2, 3, and 4. The point of this example is that the `T` bin is not confused by the repetition of surface 1.

5.9.1.1.6 Example 4

```
+F6 2
```

This card produces energy deposition (MeV/g) from all particles averaged over cell 2. This will include heating values and/or $\mathrm{d}E/\mathrm{d}x$ energy from particles undergoing library interactions (e.g., neutrons, photons, electrons, protons) and $\mathrm{d}E/\mathrm{d}x$, recoil, and non-tracked secondary particle energy from all model interactions.

5.9.1.2 Detector Tallies (Tally Type 5)

Point detectors, ring detectors, and radiography tallies use an assumption of isotropic scatter for contributions from collisions within the model regime (i.e., generally $E > 150$ MeV). These estimators require the angular distribution data for particles produced in an interaction to predict the "next event." Information on these distributions is available in tabular form in the libraries; however, this information is not available in the required form from physics models used to produce secondary particles above the tabular region. The limit on the number of detectors is given in Table 4.1.

The user is encouraged to read about detectors before implementing them because they are susceptible to unreliable results if used improperly. Here are a few hints:

1. Remember that contributions to a detector are not made through a region of zero importance.
2. Ring (rather than point) detectors should be used in all problems with axial symmetry.
3. Flux image detectors should be located in a void because the constant flux neighborhood *ro* is not used. Such a neighborhood would have to enclose the entire image grid.
4. A detector located right on a surface will probably cause trouble.
5. Detectors and DXTRAN can be used in problems with the $S(\alpha, \beta)$ thermal treatment, but the $S(\alpha, \beta)$ contributions are approximate [77].
6. Detectors used with reflecting, white, or periodic surfaces give wrong answers.
7. Consider using the PD*n* and DD*n* cards.

5.9.1.2.1 Point Detectors

Data-card Form: F*n*:$\mathscr{P}$ *x1 y1 z1 ro1 ... xK yK zK roK* [ND]

n	User-supplied tally number ending in the numeral 5. Restriction: $n \leq 99999999$
$\mathscr{P}$	Particle designator: Restriction: n for neutrons or p for photons only.
xk yk zk	Coordinates of the kth detector point (2).
rok	Radius of the sphere of exclusion for the kth detector where a positive entry is interpreted as centimeters and a negative entry is interpreted as mean free paths. A negative entry is illegal in a void (3).
ND	Optional keyword to inhibit the separate printing of the direct contribution for that detector tally (4).

5.9.1.2.2 Ring Detectors

Data-card Form: F*na*:$\mathscr{P}$ *ao1 r1 ro1 ... aoK rK roK* [ND]

n	User-supplied tally number ending in the numeral 5. Restriction: $n \leq 99999999$
a	The letter x, y, or z, which indicates the axis of the ring.
$\mathscr{P}$	Particle designator: Restriction: n for neutrons or p for photons only.
aok	Distance along axis "*a*" where the ring plane of the kth detector intersects the axis (2).
rk	Radius of the ring of the kth detector in centimeters.
rok	Same meaning as for point detectors, but describes a sphere about the point selected on the kth ring detector (3).

ND	Optional keyword to inhibit the separate printing of the direct contribution for that detector tally (④).

Default: None.

Details:

① For more than one detector with the same *n* or *na* designation, sets of the input parameters (quadruplets for F*n* or triplets for F*na*) are simply continued on the same F*n* or F*na* card.

② If more than one detector of the same type (an F5:n and an F15:n, for example) are at the same location, the time-consuming contribution calculation upon collision is made only once and not independently for each detector. Thus it is inexpensive to add more than one detector (each with a different response function, for example) at the same location.

③ The radius of the sphere of exclusion, ±*rok*, should be about 1/8 to 1/2 mean free path for particles of average energy at the sphere and zero in a void. Supplying *rok* in terms of mean free path will increase the variance and is not recommended unless you have no idea how to specify it in centimeters.

⚠ Caution

The exclusion sphere should not encompass more than one material. MCNP6 cannot verify this and the consequences may be wrong answers.

④ The printout for detectors is normally in two parts: (1) the total of all contributions to the detector (as a function of any defined bins such as energy) and (2) the direct (or un-collided) contribution to the detector from the source. The direct contribution is always included in the total of all contributions. Adding the symbol ND at the end of a type 5 detector tally card inhibits the separate printing of the direct contribution for that tally. In coupled neutron/photon problems, the direct contribution in photon tallies is from photons created at neutron collisions.

5.9.1.3 The Radiography Tally

MCNP6 can generate simulated radiography images as one would expect to see from an x-ray or pinhole projection of an object containing the particle source. This allows the recording of both the direct (source) image as well as that due to background (scatter). This tool is an invaluable aid to the problem of image enhancement, or extracting the source image from a background of clutter. MCNP6 includes two types of image capability; the pinhole image projection and the transmitted image projection.

The radiography capability is based on point detector techniques, and is extensively described in [305, 306]. In essence, the radiography focal plane grid is an array of point detectors.

5.9.1.3.1 FIP: Pinhole Image Projection

Deprecation Notice **DEP-53484**

The PI card formerly used by MCNPX for pinhole image projection is replaced by the FIP card. The input format is identical.

FIP establishes a flux image through a pinhole to a planar grid. In the pinhole image projection case, a point is defined in space that acts much like the hole in a pinhole camera and is used to focus an image onto a grid which acts much like the photographic film. The pinhole is actually a point detector and is used to define the direction cosines of the contribution that is to be made to the grid. The pinhole position relative to the grid is also used to define the element of the grid into which this contribution is scored. Once the direction is established, a ray-trace contribution is made to the grid bin with attenuation being determined for the material regions along that path. The source need not be within the object being imaged, nor does it need to produce the same type of particles that the detector grid has been programmed to score. The grid and pinhole will image either source or scattered events produced within the object (see NOTRN card) for either photons or neutrons. These event-type contributions can be binned within the grid tallies by binning as source only, total, or by using special binning relative to the number of collisions contributing cells, etc. Steps to define the image grid for a pinhole image are provided later in this section.

Data-card Form: FIP*n*:$\mathscr{P}$ *x1 y1 z1 r0 x2 y2 z2 f1 f2 f3*

n	Tally number, tally type 5. Restriction: $n \leq 99999999$
$\mathscr{P}$	Particle designator: Restriction: n for neutrons or p for photons only.
x1 y1 z1	The coordinates of the pinhole center.
r0	Always 0 (zero) for this application. Note: neither the pinhole nor the grid should be located within a highly scattering media.
x2 y2 z2	The reference coordinates (center of object) that establish the reference direction cosines for the normal to the detector grid. This direction is defined as being from $(x2, y2, z2)$ to the pinhole at $(x1, y1, z1)$.
f1	If
	$f1 > 0$: this value is the radius of a cylindrical collimator, centered on and parallel to the reference direction, which establishes a radial field of view through the object and surrounding materials and onto the image grid.
	$f1 = 0$: the value of the radius is "large." (DEFAULT)
f2	The radius of the pinhole perpendicular to the reference direction. If
	$f2 = 0$: this represents a perfect pinhole.
	$f2 > 0$: the point within the pinhole through which the particle flux contribution will pass is picked randomly (i.e., uniformly in area) for each source and scatter event. This simulates a less-than-perfect pinhole.
f3	The distance from the pinhole at $(x1, y1, z1)$ to the center of the detector grid along the direction established from $(x2, y2, z2)$ to $(x1, y1, z1)$. The image grid is perpendicular to this reference vector.

Details:

1. Only one pinhole image tally per FIP card is allowed. The point detector Russian roulette game is not used with the FIP tally. Consider use of the NOTRN card for only direct contributions and the TALNP card

to reduce the size of the MCNP output file for large-image grids. The image grid should not be in a scattering material because the point detector average flux neighborhood is not used for flux image tallies.

5.9.1.3.2 FIR and FIC Transmitted Image Projection

Deprecation Notice DEP-53482

The TIR and TIC cards formerly used by MCNPX for pinhole image projection are replaced by the FIR and FIC cards, respectively. The input format is identical.

FIR establishes a flux image on a rectangular radiograph planar grid, and FIC establishes a flux image on a cylindrical radiograph grid.

In the transmitted image projection case, the grid acts like a film pack in an x-ray type image, or transmitted image projection. In both cases, for every source or scatter event a ray-trace contribution is made to every bin in the detector grid. This eliminates statistical fluctuations across the grid that would occur if the grid location of the contribution from each event were to be picked randomly, as would be the case if one used a DXTRAN sphere and a segmented surface tally. For each event, source or scatter, the direction to each of the grid points is determined, and an attenuated ray-trace contribution is made. As in pinhole image projection, there are no restrictions as to location or type of source used. These tallies automatically bin in a source-only and a total contribution. Steps to define the image grid for transmitted images are provided later in this section.

When this type of detector is being used in a problem, if a contribution is required from a source or scatter event, an attenuated contribution is made to each and every detector grid bin. Because for some types of source distributions very few histories are required to image the direct or source contributions, an additional entry has been added to the NPS card to eliminate unwanted duplication of information from the source.

Rectangular-grid Data-card Form: FIR*n*:$\mathscr{P}$ *x1 y1 z1 r0 x2 y2 z2 f1 f2 f3*
or
Cylindrical-grid Data-card Form: FIC*n*:$\mathscr{P}$ *x1 y1 z1 r0 x2 y2 z2 f1 f2 f3*

n	Tally number, tally type 5. Restriction: $n \leq 99999999$
$\mathscr{P}$	Particle designator: Restriction: n for neutrons or p for photons only.
x1 y1 z1	The coordinates of the center of the detector flux image grid, the extent and spacing of which are defined by the entries on the tally segment (FS) and cosine (C) cards. In the planar rectangular grid case, this point defines the center of the grid. In the cylindrical grid case, this point defines the center of the cylinder on which the grid is established.
r0	Always 0 (zero) in this application. Do not locate the image grid in a scattering material.
x2 y2 z2	The reference coordinates (center of object) that establish the reference direction cosines for the outward normal to the detector grid plane, as from $(x2, y2, z2)$ to $(x1, y1, z1)$. This direction is used as the outward normal to the detector grid plane for the FIR case, and as the centerline of the cylinder for the FIC case.
f1	If

	f1 > 0	only the scattered contributions will be scored. (See Note 2.)
	f1 = 0	both the direct (source) and scattered contributions will be scored at the detector grid.
f2	Radial field of view.	
	Planar grid case: Radial restriction relative to the center of the grid to define a radial field of view on the grid for contributions to be made. If *f2* = 0, no radial restriction exists. (DEFAULT)	
	Cylindrical grid case: Radius of the cylindrical surface of the image grid. If *f2* = 0, it is a fatal error.	
f3	If	
	f3 = 0	all flux contributions are directed to the center of each grid bin.
	f3 = 1	contributions are made with a random offset from the center of the image grid bin. This offset remains fixed and is used as the offset for contributions to all of the grid bins for this event.

Details:

1. Only one flux image detector is allowed on each FIC or FIR card. The point detector Russian roulette game is not used with FIC or FIR tallies. Consider use of the NOTRN card for only direct contributions, the second entry on the NPS card for limiting the direct FIR contributions, and the TALNP card to reduce size of the MCNP output file for large-image grids.
2. The scattered contributions can often be made on a much coarser image grid because there is much less structure to the scattered image. Use *f1* = −1 in this case. The NOTRN card can be used to obtain only the direct image with *f1* = 0.

5.9.1.3.3 Defining an FIP, FIR, or FIC Image Grid Using Space and Cosine Segmenting Cards

The grid plane is in the two-dimensional (s, t) coordinate system where the s and t axes are orthogonal to the reference direction. The s and t dimensions are established from entries on tally segment (FS*n*) and cosine (C*n*) cards, where the tally number *n* matches the flux image tally number.

In the case of FIP and FIR, the image-plane rectangular grid dimensions are defined by setting the first entry on the FS*n* and C*n* cards to the lower limit (in centimeters) of the first image bin for the s axis and t axis, respectively. The other entries on the FS*n* and C*n* cards set the upper limit of each of the bins. These limits are set relative to the intersection of the reference direction and the grid plane.

In the cylindrical (FIC) grid case, the entries on the FS*n* card are the distances along the symmetry axis of the cylinder from (*x1*, *y1*, *z1*), and the entries on the C*n* card are the angles in degrees as measured counterclockwise from the positive t axis.

The relationship of the s axis, t axis, and reference direction for the planar image grid is calculated by MCNP6 and follows the right-hand rule. Since the orientation of the s axis and the t axis is dependent on the

reference direction in the geometry coordinate system, the MCNP6 tally output should be examined to see the direction cosines of these two planar image grid axes. These limits should be defined taking into account any image size change at the grid caused by magnification. The image grid should not be in a scattering material because the point detector average flux neighborhood is not used for flux image tallies.

There is no limit to the number of image grid bins that can be defined by `FS`*n* and `C`*n*. However, it is easy to define a tally with a huge number of point detectors. For example, a 1000×1000 grid is the equivalent of 1-million point detectors, which could take a long time to run. Fatal errors will result if the `FS`*n* and `C`*n* card bin specifications are not each monotonically increasing. The default tally fluctuation chart bin is the last `FS`*n* and `C`*n* bin in the total (direct plus scattered) detector tally. `FS` `0` and `C` `0` cards for these image tallies are not allowed. The `T` (total) and `C` (cumulative) options for the `FS`*n* and `C`*n*cards are not available for flux image tallies.

The directions of the t axis and s axis of the grid are set up such that if the reference direction (the outward normal to the grid plane) is not parallel to the z axis of the geometry, the t axis of the grid is defined by the intersection of the grid plane and plane formed by the z axis and the point where the reference direction would intersect the grid plane. If the reference direction is parallel to the z axis of the geometry, then the t axis of the grid is defined to be parallel to the y axis of the geometry. The s axis of the grid is defined as the cross product of a unit vector in the "t" direction and a unit vector in the reference direction. If the reference direction is not parallel to the z axis, MCNP6 calculates the orthogonal axes. The s and t image axes direction cosines are printed in the MCNP output file.

Example 1

```
FSn   -20.   99i   15.
Cn    -25.   99i   10.
```

These two cards set up a 100×100 grid that extends from -20 cm to 15 cm along the s axis, from -25 cm to 10 cm along the t axis, and has 10,000 equal sized bins. These bins need not be equal in size nor do they need to be symmetric about the reference direction.

5.9.1.3.4 Reading or Plotting the Radiography Tally Output

Pinhole and radiography tallies can be plotted directly in the MCNP6 tally plotter from the runtape or **`mctal`** files. To create a 2-D contour plot of the s and t axes enter `FREE SC`. The MCNPTools **`mctal2rad`** utility [307] can also plot radiograph tallies. In addition, the **`gridconv`** utility [Appendix E.4] can format radiograph tallies results for external graphics programs.

5.9.1.4 Pulse-height Tally (Tally Type 8)

The pulse-height tally is a radical departure from other MCNP6 tallies. All other tallies are estimates of macroscopic variables, such as flux, whose values are determined by very large numbers of microscopic events. The pulse-height tally records the energy or charge deposited in a cell by each source particle and its secondary particles. For other tallies it is not necessary to model microscopic events realistically as long as the expectation values of macroscopic variables are correct. For the pulse-height tally, microscopic events must be modeled much more realistically.

The departures from microscopic realism in MCNP6 are everywhere. The number, energies, and directions of the secondary neutrons and photons from a neutron collision are sampled without any correlation between

the particles and with no regard for the conservation of energy. The fluctuations in the energy loss rate of an electron are not correlated with the production of knock-on electrons and x-rays. The variance-reduction schemes in MCNP6 distort the natural random walk process in various ways; nevertheless, they give correct results for macroscopic tallies when appropriate weighting factors are used.

Problems that give correct pulse height tallies are severely limited. For example, the pulse-height tally does not work well with neutrons because of the non-analog nature of neutron transport that departs from microscopic realism at every turn. One can have a neutron source in a MODE n p or n p e problem, but only the photons and electrons can be tallied on the F8 card. The F8 tally can be used effectively in photon problems. Electron problems may give correct results as long as the tally cells are thick enough for the errors in the energy loss rate to average out. Combining F8 tallies with a photonuclear bias is a fatal error. MCNP6 tries to detect conditions in a problem that would invalidate pulse height tallies, but it is not able to catch all of them. The user must ascertain that his problem does not violate the necessary conditions for obtaining correct answers.

Scoring the pulse-height tally is done at the end of each history. In the absence of variance reduction, the scoring is reasonably easy to describe. For example, consider a unit weight source and an F8 tally in cell 7. Suppose that on a given particle history that there are K entries into cell 7 and L departures from cell 7. The tally energy associated with an F8 tally is the kinetic energy of the particle plus 1.022016 MeV if it is a positron. Particles can enter cell 7 either by crossing a boundary into cell 7 or entering cell 7 as a source event. Particles depart cell 7 either by capture in cell 7 or by crossing a boundary out of cell 7. Let E_i be the ith tally energy of a particle entering cell 7 and let D_j be the jth tally energy of a particle departing cell 7. The total energy deposited in cell 7 is

$$T = \sum_{i=1}^{K} E_i - \sum_{j=1}^{L} D_j. \tag{5.37}$$

Suppose the pulse height bins are specified on the E8 card as:

```
E8   T1 T2 T3 T4 T5
```

If $\mathsf{T}_{m-1} < \mathsf{T} < \mathsf{T}_m$, then MCNP6 will post a unit tally in the mth bin. If the problem is analog but the source weight is w_s, then w_s would be posted in the mth bin. If there is an asterisk on the F8 card (i.e., *F8), then MCNP6 tallies $w_s T$ in the mth bin. If there is a plus on the F8 card (i.e., +F8), then MCNP6 posts the net charge change times the w_s into the mth bin. That is, an entering electron or a departing positron constitutes a charge change of -1, whereas a departing electron or an entering positron constitutes a charge change of $+1$.

The scoring details are more complex with pulse-height tally variance reduction [66].

One common application of the F8 tally is simulation of the energy distribution of pulses created in a detector by radiation. The union of tallies produces a tally sum, not an average. Cell, user, and energy bin cards are allowed. Flagging and multiplier bins are not allowed. Segment, time, and cosine bins are permitted with certain FT options. Use of the dose energy (DE) and dose function (DF) cards is also disallowed with the F8 tally.

The energy bins in the F8 pulse-height tally are different from those of all other tallies. Rather than tally the particle energy at the time of scoring, the number of pulses depositing energy within the bins are tallied. That is, the meaning of the energy bins of a pulse-height tally is the energy deposited in a cell bin by all the physically associated tracks of a history. Care must be taken when selecting energy bins for a pulse-height tally. It is recommended that a zero bin and an epsilon bin be included such as

```
E8     0   1E-5   1E-3   1E-1   ...
```

The zero bin will catch non-analog knock-on electron negative scores. The epsilon (10^{-5}) bin will catch scores from particles that travel through the cell without depositing energy.

With the FT8 special tally treatments card the F8 tally can become an anti-coincidence pulse-height tally (FT8 PHL) or a different kind of tally altogether. For example, FT8 CAP is a neutron coincidence capture tally, and FT8 RES tallies the residual nuclides from physics-model evaporation and fission models. These variations have special rules regarding possible variance reduction, time bins, and other issues.

5.9.1.4.1 Pulse-height Tally Variance Reduction

Variance reduction for F8 tallies is implemented for electrons and photons; however, not a lot of experience exists to guide the user. Experience suggests that weight windows be used instead of geometry splitting for F8 tallies. Many of the variance-reduction techniques that were designed for lowering the variance on other tally types may be used with the F8 tally. Allowed variance reduction techniques include

- Splitting/roulette (IMP card)
- Implicit capture and weight cutoff (CUT card)
- Weight window (WWN card)
- Forced collisions (FCL card)
- Exponential transform (EXT card)
- DXTRAN (DXT card)
- Weight roulette on DXTRAN particle (DD card)
- DXTRAN cell probabilities (DXC card)
- Source biasing (SB card)
- Energy splitting (ESPLT card)
- Time splitting (TSPLT card)

The roulette associated with splitting/roulette (IMP card) and weight windows (WWN card) may be less useful than it is for non-F8 tallies; roulette may be turned off by setting the keyword RR = off on the VAR card. Although implicit capture and weight cutoff have been implemented, in most cases these games are turned off by default if an F8 tally is in the problem. An exception occurs if forced collisions also are used in the problem.

Note that the weight-window generator was designed for non-F8 tallies; the generator should not be used for F8 tallies. The generator estimates the importance of a single particle at a phase-space point $\mathbf{p}$. The generator cannot estimate the importance of a collection of K particles at phase-space points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k$. To see what is involved with making a generator work with F8, see [308]. Instead, a useful weight window often can be generated using a tally such as an F4 tally in the same cell as the F8 tally.

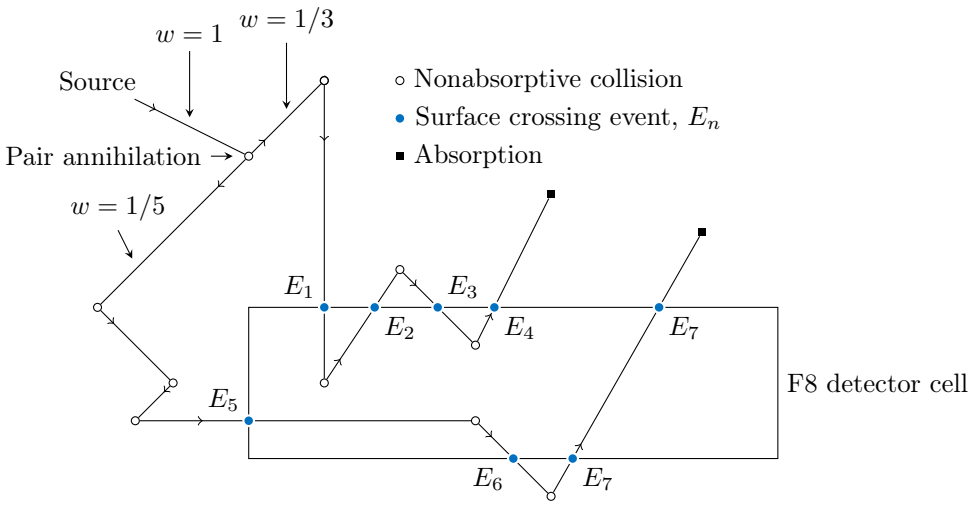


Figure 5.12: Example of exponential transform applied to both branches of a pair annihilation event.

5.9.1.4.2 Pulse-height Tally Variance Reduction: Discussion Using Examples

The MCNP6 pulse height variance-reduction theory is described in detail in [65, 66]. Two simple examples are given in this manual to give the reader an idea of how MCNP6 does variance reduction with pulse height tallies. The essential idea is that MCNP6's deconvolution method reconstructs physically possible random walks and assigns an appropriate tally weight based on how much the variance reduction has distorted the frequency of obtaining the walks. For example, if a random walk has been made twice as likely to occur in the simulation as it would naturally, then this random walk will be assigned at weight of 1/2 so that the expected tallies are preserved.

Let's suppose, as depicted in Fig. 5.12, that there is a pair annihilation event and an exponential transform is applied to both 0.511-MeV branches. Assume this is the only variance reduction used. Because the exponential transform samples a non-analog density, there will be a weight multiplication to account for this. The left branch has a track weight of 1/5 indicating that the left branch's random walk was made 5 times more likely to occur as it would have without applying the exponential transform. Similarly, the right branch has a track weight of 1/3 indicating that the right branch's random walk was made 3 times more likely to occur as it would have without applying the exponential transform. Assuming that none of the E_i is in the same bin, the tally for the total current into the cell is tallied as

- 1/5 in the energy bin around E_5
- 1/5 in the energy bin around E_7
- 1/3 in the energy bin around E_1
- 1/3 in the energy bin around E_3

and the total current leaving the cell is tallied as

- 1/5 in the energy bin around E_6
- 1/5 in the energy bin around E_7

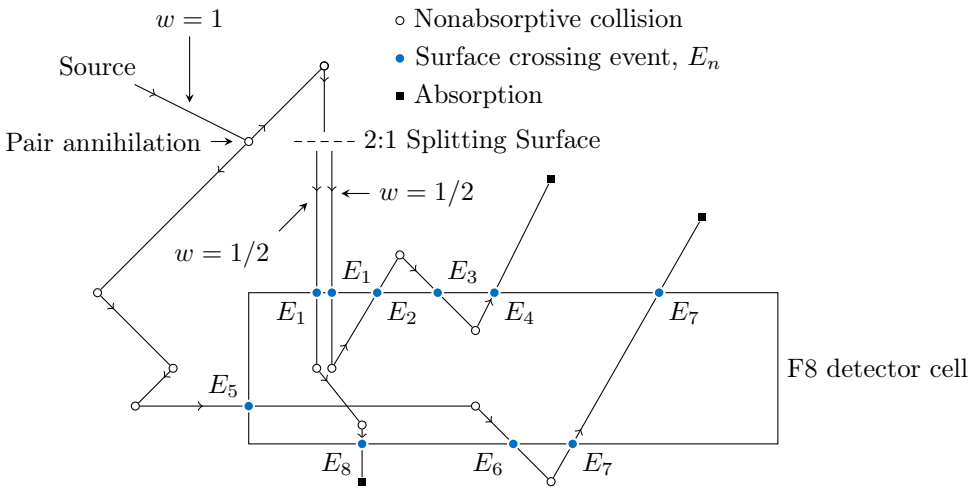


Figure 5.13: F8 Tally Splitting example.

- 1/3 in the energy bin around E_2
- 1/3 in the energy bin around E_4

By contrast, as explained below, the F8 tally for this history is (1/5)(1/3) in the energy bin around $E_5 - E_6 + E_7 - E_7 + E_1 - E_2 + E_3 - E_4$. Note that the F8 tally depends on the energy deposited collectively by both branches of the pair annihilation event. If the history above had been sampled without the exponential transforms, then the F8 tally would have been 1 in the energy bin around $E_5 - E_6 + E_7 - E_7 + E_1 - E_2 + E_3 - E_4$.

Note that the physical walk contributes $E_5 - E_6 + E_7 - E_7 + E_1 - E_2 + E_3 - E_4$ regardless of how often the walk is sampled. With the variance reduction applied here, the particular walk sampled occurred $5 \times 3 = 15$ times as often as it would in an analog calculation. Thus, the F8 tally credits the physical energy bin with a weight factor of 1/15, correcting for the fact that the annihilation pair has been made 15 times as likely to execute the walk that contributes $E_5 - E_6 + E_7 - E_7 + E_1 - E_2 + E_3 - E_4$ as it should have. Note that it is a physical collection of particles that now carries the tally modification weight because it is the physical collection that tallies to the F8 tally rather than just the individual tracks as with other tallies in MCNP6.

The second example, illustrated by Fig. 5.13, considers a 2:1 splitting event and no other variance reduction methods. Note that splitting is a mathematical artifice; only one physical particle exists after crossing the splitting surface. What the splitting does is give two (usually) different samples of the random walk after crossing the splitting surface. Both of these random walks do not physically occur at the same time. If the left split branch occurs then the right split branch does not and vice versa. Because the splitting represents a doubling of the sampling frequency for either branch, the branches are each assigned a weight of 1/2. The energy bins associated with taking the left split branch or the right split branch are, respectively, $E_5 - E_6 + E_7 - E_7 + E_1 - E_8$ or $E_5 - E_6 + E_7 - E_7 + E_1 - E_2 + E_3 - E_4$. The pulse-height tally is thus 1/2 in the energy bin around $E_5 - E_6 + E_7 - E_7 + E_1 - E_8$ and 1/2 in the energy bin around $E_5 - E_6 + E_7 - E_7 + E_1 - E_2 + E_3 - E_4$.

Simple Data-card Form: `Fn:`$\mathscr{P}$ *s1 ... sK*
or
General Data-card Form: `Fn:`$\mathscr{P}$ *s1* (*s2 ... s3*) (*s4 ... s5*) *s6 s7 ...* `T`

n	User-supplied tally number, ending in the numeral 8. Restriction: $n \leq 99999999$

$\mathscr{P}$	Particle designator. Standard F8 tallies support only "p,e" (if only one of these is specified, it is expanded to include both). Other particle types should only be specified with the FT PHL or CAP options (2).
sk	Problem number of cell for tallying, or T for the total across all listed cells.
T	Provide average of tally over specified cells. (Optional)

Details:

1. An asterisk on the F8 card converts the tally from a pulse-height tally to an energy deposition tally. A plus on the F8 card converts the tally from a pulse-height tally to a charge deposition tally in units of charge. Energy binning is not recommended with the +F8 tally.
2. Both photons and electrons will be tallied if present, even if only e or only p is on the F8 card. In other words, F8:p, F8:e, and F8:p,e are all equivalent tallies.

5.9.1.4.3 Example 1

```
F8:E     1
```

or

```
F1:E      2
FT1 ELC   1
C1       0 1
```

The +F8 charge deposition tally can be checked against an electron F1:e surface tally with the FT ELC option if the volume of the +F8 is exactly enclosed by the surfaces on the F1:e card. For example, if cell 1 is enclosed by spherical surface 2, then the above tallies give the same result provided the two F1 current tally bins (in minus out) are properly subtracted.

5.9.1.5 Repeated Structures Tallies (Tally Types 1, 2, 4, 6, 7, and 8)

Simple Data-card Form: F*n*:$\mathscr{P}$ *s1* ... *sK*
or
General Data-card Form: F*n*:$\mathscr{P}$ *s1* (*s2* ... *s3*) ((*s4 s5*) < (*c1 c2*[*i1* ... *i2*]) < U = #(*c3 c4 c5*)) ... T

n	Tally number. Restriction: $n \leq 99999999$
$\mathscr{P}$	Particle designator.
sk	Problem number of a surface or cell for tallying or U = #.
ck	Problem number of a cell filled with a universe or U = #.

T	Average or total over specified surfaces or cells, depending on type of tally. (Optional)
U = #	Problem number of a universe used on a FILL card.
ik	Index data for a lattice cell element, with three possible formats (always in brackets). If

ik = *i1*	then *ik* indicates the 1st lattice element of the given cell, as defined by the FILL array.
ik = *i1*:*i2* *i3*:*i4* *i5*:*i6*	then *ik* indicates a range of one or more lattice elements. Use the same format as on the FILL card.
ik = *i1* *i2* *i3*,*i4* *i5* *i6*	then *ik* indicates individual lattice elements [*i1*, *i2*, *i3*], [*i4*, *i5*, *i6*], etc.

See LAT and FILL cards for an explanation of the indices.

Use: Consider using the SPDTL card.

In the simple repeated-structure tally form, MCNP6 creates k surface or cell bins for the requested tally, listing the results separately for each surface or cell. In the more general form, a bin is created for each surface or cell listed separately and for each collection of surfaces or cells enclosed within a set of parentheses. A tally bin can involve a single tally level or multiple tally levels. Tallies involving repeated structure and lattice geometries can use either form.

Some operators and nomenclature need to be introduced before the explanation of repeated structures and lattice tallies. The left arrow or less than symbol < is used to identify surfaces or cells within levels of repeated structures. See §5.5.5.1 for an explanation of geometry levels. A tally bin that includes one or more left arrows implies multiple levels, called a chain. Multiple entries enclosed by parentheses at any level of a tally chain indicate the union of the items. Brackets [...] immediately following a filled lattice cell identify one or more elements of that lattice.

5.9.1.5.1 Multiple Bin Format

In addition to multiple levels, multiple entries can be used in each level of the tally chain resulting in multiple output bins. Within the parentheses required around the tally bin chain, other sets of parentheses can be used to indicate the union of cells as in a simple tally description, resulting in fewer output tally bins. For example,

$$((s4\ s5) < (c1\ c2[i1\ \ldots\ i2]) < (c3\ c4\ c5))$$

results in one output tally bin and will be the union of the tally in *s4* plus *s5* that fill *c1* or *c2* elements [*i1* ... *i2*] and when *c1* or *c2* fills cells *c3*, *c4*, or *c5*. Removing the first and third inner parentheses, i.e.,

$$(s4\ s5 < (c1\ c2[i1\ \ldots\ i2]) < c3\ c4\ c5)$$

results in the creation of $2 \times 1 \times 3 = 6$ bins as follows:

$$(s4 < (c1\ c2[i1\ \ldots\ i2]) < c3),$$
$$(s4 < (c1\ c2[i1\ \ldots\ i2]) < c4),$$
$$(s4 < (c1\ c2[i1\ \ldots\ i2]) < c5),$$
$$(s5 < (c1\ c2[i1\ \ldots\ i2]) < c3),$$
$$(s5 < (c1\ c2[i1\ \ldots\ i2]) < c4),$$
$$(s5 < (c1\ c2[i1\ \ldots\ i2]) < c5).$$

The repeated structure/lattice input tally bin format with levels that have multiple entries automatically creates multiple output tally bins. The total number of bins generated is the product of the number of entries at each level. If parentheses enclose all entries at a level, the number of entries at that level is one and results in the union of all those entries. For unnormalized tallies (types 1, 8), the union is a sum. For normalized tallies (types 2, 4, 6, 7), the union is an average. A symbol `T` on the tally line creates an additional tally bin that is the union or total of all the other tally bins.

5.9.1.5.2 Brackets

Brackets [...] enclose index data for lattice cell elements. Brackets make it possible to tally on a cell or surface only when it is within the specified lattice elements. The brackets must immediately follow a filled lattice cell. Listing a lattice cell without brackets will produce a tally when the tally cell or surface is in any element of the lattice, provided the tally cell or surface fills an entry at all other levels in the chain. The use of brackets is limited to levels after the first "<" symbol in the tally specification.

To tally within lattice elements of a real world (level zero) lattice cell, use the special syntax that follows. Cell 3 contains material 1 and is bounded by four surfaces. The `F4` card specifies a tally only in lattice element $[0, 0, 0]$. This syntax is required because brackets can only follow a $<$ symbol:

```
3     1   -1.0   -1234   lat=1
.
.
.
F4:N      (3 < 3 [0 0 0])
```

5.9.1.5.3 Universe Format

The universe format, `U = #`, is a shorthand method of including all cells and lattice elements filled by universe `#`. The example shown in Listing 5.49 demonstrates this universe-expansion capability with the tally definition and comments representing the expanded form.

Listing 5.49: example_tally_universe_expansion_stochastic_volume.mcnp.inp.txt

```
f214:n ((1000 1100) < 2000 < 3000)        $ ((1000 1100) < 2000 < 3000)
f224:n ((1000 1100) < 2000 < 3100)        $ ((1000 1100) < 2000 < 3100)
f234:n ((1000 1100) < (u = 1) < 3000)     $ ((1000 1100) < (2000) < 3000))
f244:n ((1000 1100) < (u = 1) < 3100)     $ ((1000 1100) < (2000) < 3100))
f254:n ((1000 1100) < 2000 < (u = 2))     $ ((1000 1100) < 2000 < (3000 3100))
f264:n ((1000 1100) < (u = 1) < (u = 2))  $ ((1000 1100) < (2000) < (3000 3100))
f274:n ((1000 1100) < (u = 1) < u = 2)    $ ((1000 1100) < (2000) < 3000 3100)
f284:n ((1000 1100) < u = 1 < (u = 2))    $ ((1000 1100) < 2000 < (3000 3100))
```

```
f294:n ((1000 1100) < u = 1 < u = 2)      $ ((1000 1100) < 2000 < 3000 3100)
f204:n ((1000 1100) < u = 2)              $ ((1000 1100) < 3000 3100)
f314:n ( 1000 1100 < u = 1 < u = 2)       $ ( 1000 1100  < 2000 < 3000 3100)
```

Note that inner parentheses are used to indicate unions of cells in the expansion, which can lead to fewer bins than otherwise. Accordingly, in complex geometries, the `U = #` format should be used sparingly, especially with the multiple bin format. If 100 cells are filled by universe 1 and 10 cells are filled by universe 2 (which contains universe 1), then 1000 tally bins will be created if unions are not used.

Further, note that if contained cells in interstitial universes (not the highest or lowest level) are defined at varying levels of universe nesting in the same calculation, this universe expansion may be expanded incorrectly. As such, one should verify that the expansion has been performed correctly and to provide explicit input if it is found to be incorrect. A way to induce this incorrect behavior is to create a third box as cell 3200 below the current two boxes using the input in Listing 5.49 and to fill it with universe 1 (skipping universe 2). In this example, because universe 1 is now contained in both universes 2 and 3, an incorrect universe expansion may appear in the MCNP output file as `cell (1000 1100<2000 3200<3000)` (expanded from `f034:n ((1000 1100) < (u = 1) < 3000)`) or similar where the new cell 3200 is contained in cells at the same universe nesting level (cells 3000 and/or 3100), which is incorrect.

5.9.1.5.4 Example

This example shown in Listing 5.50 runs significantly faster with MCNP6 than with MCNP4C.

Listing 5.50: example_repeated_structure_tally.mcnp.inp.txt

```
21x21x21 void lattice of spheres
11 0 -31    u=1 imp:p=1
12 0  31    u=1 imp:p=1
16 0 -32    u=2 imp:p=1 lat=1 fill= -10:10 -10:10 -10:10 1 9260R
17 0 -33 fill=2 imp:p=1
18 0  33        imp:p=0

31 so 0.5
32 rpp  -1  1   -1  1   -1  1
33 rpp -21 21  -21 21  -21 21

mode p
sdef
f4:p (11<16[-10:10 -10:10 -10:10]<17)
print
nps    10000
```

Larger lattices and nested lattices offer even more dramatic speedups.

5.9.1.5.5 Use of SD Card for Repeated Structures Tallies

MCNP6 may be unable to calculate required volumes or areas for tallies involving repeated-structure and lattice geometries. For example, a universe can be repeated a different number of times in different cells and the code has no way to determine this.

There are two distinct options for entries on the `SD` card relating to repeated structures and they cannot be mixed within a single tally.

The first option is to enter a value for each first level entry on the related F card. If the entry on the F card is the union of cells, the SD card value will be the volume of the union of the cells. An example of this is shown in Listing 5.51.

Listing 5.51: example_tally_universe_expansion_stochastic_volume.mcnp.inp.txt

```
f114:n (1000 < 2000 < 3000)  (1100 < 2000 < 3000)
f124:n ((1000 1100 1200) < 2000 < 3000)
f134:n (1200 < 2000 < 3000)
f144:n ((1000 1100 1200) < 2000 < 3000) (2100 < 3000) t
f154:n (2100 < 3000)
f164:n ((1000 1100 1200) < 2000 < 3000) ((1000 1100 1200) < 2000 < 3100)
       (2100 < 3000) (2100 < 3100) 4000 t
f174:n 4000
c
sd114 1 1
sd124 1
sd134 1
sd144 1 1 1
sd154 1
sd164 1 1 1 1 1 1
sd174 1
```

The second option is to enter a value for each bin generated by the F card. An example of this is shown in Listing 5.52 (which correspond to the tallies shown in Listing 5.13).

Listing 5.52: example_tally_universe_expansion_stochastic_volume.mcnp.inp.txt

```
sd214 1
sd224 1
sd234 1
sd244 1
sd254 1
sd264 1
sd274 1
sd284 1
sd294 1
sd204 1
sd314 1 1 1 1
```

5.9.2 FC: Tally Comment

Data-card Form: FC*n info*

n	Tally number. Restriction: $n \leq 99999999$
info	Provides title for tally in MCNP output and **MCTAL** files (1).

Default: No comment.

Use: Encouraged, especially when using a modified or non-standard tally.

Details:

(1) The FC card can be continued only by blanks in columns 1–5 on succeeding lines. The & continuation symbol is considered part of the comment and not recognized as a continuation command. Like other cards, the line-length limit given in Table 4.1 applies.

5.9.3 E: Tally Energy Bins

This card is used to assign energy-bin boundaries for tallies to accumulate results into. If it is not present, the results are accumulated into a single "total" bin regardless of the particle energy.

Data-card Form: E*n e1* ... *eK* [NT] [C]

n	Tally number. Restriction: $n \leq 99999999$
ek	Upper bound (in MeV) of the kth energy bin for tally *n* ((2)).
NT	Optional notation at the end of the input line to inhibit the automatic total over all specified energy bins.
C	Optional notation at the end of the input line to cause the bin values to be cumulative and the last energy bin to be the total over all energy bins.

Default: If the E card is absent, there will be one bin over all energies unless this default has been changed by an E0 card.

Use: Required if EM card is used.

Details:

(1) An E0 card can be used to set up a default energy-bin structure for all tallies. A specific E*n* card will override the default structure for tally *n*.

(2) The energies on the E card must be entered in the order of increasing magnitude. If a particle has energy greater than the last entry, it is not tallied and a warning is issued. A comment is printed if the last energy bin is greater than the upper limit specified on the PHYS card.

5.9.3.1 Example 1

```
1 E11    0.1  1  20
```

This card will separate an F11 current tally into four energy bins:

1. from the lower energy cutoff to 0.1 MeV,
2. from 0.1 to 1.0 MeV,
3. from 1.0 to 20.0 MeV, and
4. a total over all energy.

5.9.4 T: Tally Time Bins

This card is used to assign time-bin boundaries for tallies to accumulate results into. If it is not present, the results are accumulated into a single "total" bin regardless of the particle time.

Data-card Form: `Tn t1 ... tK [NT] [C]`
or
Data-card Form: `Tn keyword = values(s)`

n	Tally number. Restriction: $n \leq 99999999$
tk	Upper bound (in shakes) of the kth time bin for tally *n* (2).
NT	Optional notation at the end of the input line to inhibit the automatic total over all specified time bins.
C	Optional notation at the end of the input line to cause the bin values to be cumulative and the last time bin to be the total over all time.
CBEG = *value*	Reference starting time in shakes (sh) (DEFAULT: CBEG = 0)
CFRQ = *value*	Frequency of cycling in 1/sh or 1/time width
COFI = *value*	Dead time interval in shakes
CONI = *value*	Alive time interval in shakes
CSUB = *value*	Number of subdivisions to use within alive time (DEFAULT: CSUB = 1)
CEND = *value*	Reference ending time in shakes (optional)

Default: If the T card is absent, there will be one bin over all times unless this default has been changed by a T0 card; CBEG = 0; CSUB = 1.

Use: Required if TM card is used. Consider FQ card.

Caution

One shake is equal to 10^{-8} seconds, which is equal to 10 nanoseconds.

Details:

(1) A T0 card can be used to set up a default time-bin structure for all tallies. A specific T*n* card will override the default structure for tally *n*.

(2) For the first form of the tally-time card, the times on the T card must be entered in the order of increasing magnitude. If a particle has a time greater than the last entry, it is not be tallied and a warning is issued. A comment is printed if the last time bin is greater than the time cutoff specified on the CUT card. For point detector tallies, time bins can exceed the time cutoff so that particles will score at detectors remote from the main body of the system. Setting the time cutoff lower than the last time bin will inhibit unproductive transport of slow neutrons in the system and will increase the efficiency of the problem.

(3) The the second form of the tally-time card, keyword entries allow for automatic creation of cyclic time bins. The standard time entries and keyword entries are mutually exclusive within a given T card. If CEND is specified, all cyclic time bins are generated for the first cycle and these are repeated out to the CEND time. Keyword entries can be in any order.

5.9.4.1 Example 1

```
T2      -1   1   1.0+37   NT
```

This will separate an F2 flux surface tally into three time bins:

1. from -∞ to -1.0 shake,
2. from -1.0 shake to 1.0 shake, and
3. from 1.0 shake to 10^{37} shakes, which is effectively infinity.

No total bin will be printed in this example.

5.9.4.2 Example 2

```
T1   CBEG=0.0  CFRQ=1000e-8  COFI=0.000005e8  CONI=0.0005e8  CSUB=5
```

This example specifies a reference starting time of 0 sh with a frequency of 1000 Hz (10^{-5} sh^{-1}). The dead time of 5 μs (COFI) results in a time bin from 0–500 sh that includes missed tally scores during the dead time. The alive time of 0.5 ms (CONI), with the specified five subdivisions (CSUB), results in five time bins equally spaced from 500–50500 sh. A final time bin from 50500–100000 sh will be provided for tally scores made after the alive time. Note that using the "e8" and "e-8" form shown here makes it easy to express the entries in seconds and Hertz rather than using the native unit of shakes.

5.9.5 C: Tally Cosine Bins (Tally Type 1 and 2)

This card is used to assign cosine-bin boundaries for surface tallies to accumulate results into. If it is not present, the results are accumulated into a single "total" bin regardless of the direction that the particle has relative to the tally surface when it crosses the surface.

Data-card Form: C*n c1* ... *cK* [T] [C]
or
Data-card Form: ∗C*n φ1* ... *φK* [T] [C]

n	Tally number. Restriction: $n \leq 99999999$
ck	Upper cosine limit of the kth angular bin for surface current or flux tally *n* (See Notes 3 and 4.). Restrictions: $c1 > -1$ and $cK = 1$, where *cK* is the entry for the last bin
φk	Upper angular limit of the kth angular bin for surface current or flux tally *n* (See Notes 3 and 4.). Restrictions: $\varphi 1 < 180$ and $\varphi K = 0$, where φK is the entry for the last bin
T	Optional notation at the end of the input line to provide the total over all specified angular bins.

C	Optional notation at the end of the input line to cause the bin values to be cumulative and the last angular bin to be the total over all angles.

Default: If the C card is absent, there will be one bin over all angles unless this default has been changed by a C0 card.

Use: For use with tally types 1 and 2 only. Required if CM card is used. Consider FQ card.

Details:

1. A C0 card can be used to set up a default angular bin structure for all tallies. A specific C*n* card will override the default structure for tally *n*. The selection of a single cosine bin for an F1 tally gives the total and not the net current crossing a surface.
2. The asterisk (∗) on the C*n* card causes the entries to be interpreted as degrees.
3. Whether entered as degrees or cosines, the entries on the C card must be such that the cosine is monotonically increasing, beginning with the cosine of the largest angle less than 180° to the normal and ending with the normal (i.e., $\cos\theta = 1$). A lower cosine bound of -1 is set in the code and should not be entered on the card.
4. The angular limits described by the C card are defined with respect to the positive normal to the surface at the particle point of entry. An FT card with an FRV *v1 v2 v3* option can be used to make the cosine bins relative to the vector $(u, v, w) = (v1, v2, v3)$. The positive normal to the surface is always in the direction of a cell that has positive sense with respect to that surface.
5. Due to the grazing angle approximation made for F2 tallies, some cosine bins may be inaccurate and the code prints a warning when a C*n* is used with an F2 tally. Details on this approximation can be found in §2.5.2.2.

5.9.5.1 Example 1

```
C1    -0.866  -0.5   0   0.5  0.866  1
```

```
*C1      150   120  90    60     30  0
```

Either card will tally currents within the following angular limits

1. 180° to 150°,
2. 150° to 120°,
3. 120° to 90°,
4. 90° to 60°,
5. 60° to 30°, and
6. 30° to 0° with respect to the positive normal.

No total will be provided.

5.9.5.2 Example 2

As an example of the relationship between a surface normal and sense for the C1 card, consider a source at the origin of a coordinate system and a plane (PY) intersecting the $+y$ axis. An entry of 0 and 1 on the C1 card will tally all source particles transmitted through the plane in the 0 to 1 cosine bin (0° to 90°) and all particles scattered back across the plane in the −1 to 0 cosine bin (90° to 180°). A plane (PY) intersecting the $-y$ axis will result in a tally of all source particles transmitted through the second plane in the −1 to 0 bin (90° to 180°) and all particles scattered back across the plane in the 0 to 1 bin (0° to 90°). Note that the positive normal direction for both planes is the same, the $+y$ axis.

5.9.6 FQ: Print Hierarchy

This card can be used to change the order in which the output is printed for the tallies. For a given tally, the default order is changed by entering a different ordering of the letters, space delimited.

Data-card Form: FQ*n a1 ... a8*

n	Tally number. Restriction: $n \leq 99999999$
ak	Letters representing all eight possible types of tally bins:$1 \leq k \leq 8$ (2).

F	cell, surface, or detector bins
D	direct or flagged bins
U	user bins
S	segment bins
M	multiplier bins
C	cosine bins
E	energy bins
T	time bins

Default: Order as given above. The tally will be printed in the output file in blocks of time (rows) and energy (columns). Any other bins in a tally will be listed vertically down the output page.

Use: Highly recommended. Prints tallies in more easily readable blocks in the output file without affecting answers.

Details:

1. An FQ0 card can be used to change the default order for all tallies. A specific FQ card will then override that order for tally *n*.
2. A subset of the letters can be used, in which case MCNP6 places them at the end of the FQ card and precedes them with the unspecified letters in the default order. The first letter is for the outermost loop of the nest in the tally printout coding. The last two sets of bins make a table—the next to last set goes vertically, and the last set of bins goes horizontally in the table. Default order is a table in E and T; any other bins in a tally will be listed vertically down the output page.

5.9.6.1 Example 1

```
FQ4    E   S   M
```

The output file printout will be tables with multiplier bins across the top, segments listed vertically, and these segment-multiplier blocks printed for each energy.

5.9.7 FM: Tally Multiplier

The FM card basically multiplies any tallied quantity (flux, current) by any cross section to give nearly all reaction rates plus heating, criticality, etc. That is, the FM card is used to calculation any quantity of the form

$$C \int \varphi(E) R_m(E) \mathrm{d}E, \tag{5.38}$$

where $\varphi(E)$ is the energy-dependent fluence (particles/cm^2) and $R_m(E)$ is an operator of additive and/or multiplicative response functions from the MCNP6 cross-section libraries or specially designated quantities. Note that some MCNP6 cross-section-library reaction numbers (R) are different from ENDF/B (MT) reaction numbers. The constant C is any arbitrary scalar quantity that can be used for normalization. The material number *m* must appear on an M*m* card, but need not be used in a geometric cell of the problem.

Data-card Form: FM*n* (*bin set 1*) (*bin set 2*) ... [T] [C]

n	Tally number. Restriction: $n \leq 99999999$
(*bin set k*)	Represents ((*multiplier set 1*) (*multiplier set 2*) ... (*attenuator set*)), where *attenuator set* = *c* -1 *m1 px1 m2 px2* ... (1) and *multiplier set i* = *c m* (*reaction list 1*) (*reaction list 2*) ... and special multiplier set i = c k.
c	Multiplicative constant (2).
-1	Flag indicating attenuator rather than multiplier set.
m	Material number identified on an M*m* card. May be set to 0 for FMESH tallies to use the cross sections of the material the particle is in.
px	Density times thickness of attenuating material; interpreted as atom density if positive, and mass density if negative.
k	Special multiplier option. If

$k = -1$	the tally is multiplied by 1/weight and the tally is the number of tracks (or collisions for the F5 tally).
$k = -2$	the tally is multiplied by 1/velocity and the tally is the neutron population integrated over time, or the prompt removal lifetime.

	$k = -3$	the tally will be multiplied by the microscopic cross section of the first interaction. This option can be used with the LCA NOACT $= -2$ option to convert multiplicities into secondary production cross sections with units of barns (3).
(*reaction list i*)	Sums and products of ENDF or special reaction numbers (4).	
T	Optional notation at the end of the input line to provide the total over all specified bins. If absent, a total over all bins is not provided.	
C	Optional notation at the end of the input line to cause the bin values to be cumulative and the last bin to be the total over all bins.	

Use: Optional. Use the attenuators only when they are thin. When used with tally types 6 and 7, only the multiplicative constant can be specified. Disallowed for tally type 8. When used with mesh tallies, only one multiplier set and reaction list per mesh tally is permitted. If $m = 0$ for a multiplier set, the reaction cross sections for the material in which the particle is traveling are used (for FMESH tallies only).

Details:

1. An attenuator set of the form *c* -1 *m px* includes one layer and allows the tally to be modified by the factor $\exp(-\sigma_t p_x)$ representing an exponential line-of-sight attenuator. This capability makes it possible to have attenuators without actually modeling them in the problem geometry.

   Caution

   The assumption is made that the attenuator is thin, so that simple exponential attenuation without buildup from scattering is valid.

   The attenuator set can include more than one layer, in which case the factor is $\exp(-\sigma_1 p_1 - \sigma_2 p_2)$. The attenuator set can also be part of a bin set, for example,

   ((*c1 m1 R1*) (*c2 m2 R2*) (*c3* -1 *m3 px3*))

   in which case the attenuation factor is applied to every bin created by the multiplier sets. Note that both the inner and the outer parentheses are required for this application.

2. If the *c* entry is negative (for type 4 tally only), *c* is replaced by $|c|$ times the atom density of the cell where the tally is made.

3. The special multiplier option with $k = -3$ works for all incident particle types except electrons; however, for charged particles, caution should be exercised because for some charged particles maximum cross sections are used instead of actual cross sections.

4. A reaction list consists of one or more reaction numbers delimited by spaces, colons, and/or pound symbols (#). A space between reaction numbers means multiply the reactions. A colon means to add the reactions and a pound symbol means to subtract the reactions. The hierarchy of operation is multiply first and then add or subtract. One bin is created for each reaction list. No parentheses are allowed within the reaction list.

   The reaction cross sections are microscopic (with units of barns) and not macroscopic. Therefore, if the constant *c* is the atomic density (in atoms/barn-cm), the results will include the normalization "per cm^3."

   Any number of ENDF/B (MT) or special (R) reactions can be used in a multiplier set as long as they are present in the MCNP6 cross-section libraries, or in special libraries of dosimetry data. If neither a material number nor any reactions are given, the tally simply is multiplied by the constant *c*.

5.9.7.1 Use of Parentheses

1. If a given multiplier set contains only one reaction list, the parentheses surrounding the reaction list can be omitted. Parentheses within a reaction list are forbidden.
2. If a given bin set consists of more than a single multiplier or attenuator set, each multiplier or attenuator set must be surrounded by parentheses, and the combination must also be surrounded by parentheses.
3. If the FM card consists only of a single bin set, and that bin set consists only of a single multiplier or attenuator bin, surrounding parentheses can be omitted.

5.9.7.2 Special Reaction Numbers

In addition to the standard ENDF reaction numbers (e.g., MT=1, 2, and 16, representing σ_{t}, σ_{el}, and $\sigma_{\mathrm{n,2n}}$, respectively from the ENDF-6 manual(s) [45, 309, 310]), Table 5.19 lists the non-standard special R numbers that may be used.

Table 5.19: ENDF/B Special Reaction Numbers, R

Reaction Type	R	Microscopic Cross-Section Description
Neutron	-1	Total cross section without thermal
	-2	Absorption cross section
	-3	Elastic cross section without thermal
	-4	Average neutron heating number (MeV/collision)
	-5	Gamma-ray production cross section , barns
	-6	Total fission cross section
	-7	Fission $\overline{\nu}$, prompt or total
	-8	Fission Q (MeV/fission)
	-9	Fission $\overline{\nu}$, delayed
Many Nuclides	-4	Average heating numbers (MeV/collision)
	-5	Gamma-ray production cross section, barns
	-7	Fission $\overline{\nu}$ (prompt or total)
	-8	Fission Q (MeV/fission)
Photoatomic	-1	Incoherent scattering cross section
	-2	Coherent scattering cross section
	-3	Photoelectric cross section, with fluorescence
	-4	Pair production cross section
	-5	Total cross section
	-6	Average photon heating number
Proton (1)	± 1	Total cross section
	± 2	Non-elastic cross section
	± 3	Elastic cross section
	± 4	Average proton heating number
Photonuclear (2)	1	Total cross section
	2	Non-elastic cross section
	3	Elastic cross section
	4	Average photonuclear heating number
Multigroup Neutron & Photon	-1	Total cross section
	-2	Fission cross section
	-3	Fission $\overline{\nu}$ data
	-4	Fission χ data
	-5	Absorption cross section

continued on next page...

Table 5.19, continued

Reaction Type	R	Microscopic Cross-Section Description
	-6	Stopping powers
	-7	Momentum transfers
Electrons	1	de/dx electron collision stopping power
	2	de/dx electron radiative stopping power
	3	de/dx total electron stopping power
	4	Electron range
	5	Electron radiation yield
	6	Relativistic β^2
	7	Stopping power density correction
	8	Ratio of rad/col stopping powers
	9	Drange
	10	dyield
	11	MG array values
	12	QAV array values
	13	EAR array values

Details:

1. On the **LA150H** proton library, the only available reaction (beyond $\pm 1, 2, 3, 4$) is MT=5 and its multiplicities, 1005, 9005, 31005, etc. The multiplicity reaction numbers are specified by adding 1000 times the secondary particle number to the reaction number. For interaction reaction MT=5, the multiplicities are 1005 for neutrons, 9005 for protons, 31005 for deuterons, etc. The proton multiplicity, MT=9001, 9004, 9005, etc., is generally available, along with the total cross section and heating number, MT=1, MT=4.
2. The principal photonuclear cross sections are the following: 1=total, 2=non-elastic, 3=elastic, 4=heating, and > 4=various reactions such as 18=(γ,f). The photonuclear yields (multiplicities) for various secondary particles are specified by adding 1000 times the secondary particle number to the reaction number. For example, 31001 is the total yield of deuterons (particle type D=31), 34001 is the total yield of alphas (particle type a=34), and 1018 is the total number of neutrons (particle type n = 1) from fission.

The total and elastic cross sections, MT=1 and MT=2, are adjusted for temperature dependence. All other reactions are interpolated directly from the library data tables. Note that for tritium production, the R number differs from one nuclide to another. Note also that tally types 6 and 7 already include reactions, so the FMn card makes little sense for $n = 6$ or 7. Generally only the constant-multiplier feature should be used for these tally types.

Photon production reactions are characterized by multiple MT numbers because more than one photon can be produced by a particular neutron reaction that is itself specified by a single MT number. Each of these photons is produced with an individual energy-dependent cross section. For example, MT 102 (radiative capture) might produce 40 photons, each with its own cross section, angular distribution, and energy distribution. Accordingly, 40 MT numbers are needed to represent the data; the MT numbers are 102001, 102002, ..., 102040.

Photonuclear and proton cross sections may be used in tally multipliers on the FM card, however the applicability of the tally is limited to the upper energy included in the related cross-section library.

In perturbed problems, the PERT card keyword RXN can affect the cross sections used with the FM card tally multipliers. If a tally in a cell is dependent on a cross section that is perturbed, then $R_{ij'} \neq 0$ and a correction is made to the $R_{1j'} = 0$ case. For this required $R_{1j'}$ correction to be made, the user must ensure

that the R reactions on the FM card are the same as the RXN reactions on the PERT card and that the FM card multiplicative constant c is negative, indicating multiplication by the atom density to get macroscopic cross sections. For example, if $R = -6$ for fission on the FM card, you should not use RXN = 18 for fission on the PERT card. If $c > 0$, the cross sections are not macroscopic; it is assumed that there is no tally dependence on a perturbed cross section, $R_{1j'} = 0$, and no correction is made. The same correction is automatically made for the F6 tally and the KCODE k_{eff} calculation, and for an F7 tally if the perturbation reaction is fission because these three tallies all have implicit associated FM cards.

It is always wise to plot the desired cross sections first to see if they are available with the expected reaction numbers in the data library. The tally multipliers treat the data the same as the data are treated in transport: the cross section at the lowest energy is extended down to $E = 0$ for protons with reaction identifier MT< 0; the cross section at the highest energy of the table is extended to $E = \infty$ for proton interaction cross sections with MT< 0; and for photonuclear interaction cross sections, MT< 1000. These extrapolations can be seen in the cross-section plots. Examples below include total energy deposition (Example 3), track length criticality estimate (Example 4), total energy deposited for materials not present in geometry (Example 5), and lifetime calculation/reaction rates (Example 6).

5.9.7.3 Example 1

Case 1:

```
FMn c m r1 r2 : r3
```

Case 2:

```
FMn c m r1 r2 : r1 r3
```

Case 3:

```
FMn c m r1 (r2 : r3)
```

These cases reiterate that parentheses cannot be used for algebraic hierarchy within a reaction list. The first case represents one reaction list (i.e., one bin) calling for reaction *r3* to be added to the product of reactions *r1* and *r2*. The second case produces a single bin with the product of reaction *r1* with the sum of reactions *r2* and *r3*. The third case creates two bins, the first of which is reaction *r1* alone; the second is the sum of *r2* and *r3*, without reference to *r1*.

5.9.7.4 Example 2

Case 1:

```
F2:N 1 2 3 4
FM2  (c1) (c2) (c3) (c4) T
```

Case 2:

```
F12:N 1 2 3 4
FM12  c1
```

Case 3:

```
F22:N (1 2 3) 4 T
FM22  (c1) (c2) (c3) (c4)
```

These three cases illustrate the syntax when only the constant-multiplier feature is used. All parentheses are required in these examples. Tally F2 creates 20 bins: the flux across each of surfaces 1, 2, 3, and 4 with each multiplied by each constant *c1*, *c2*, *c3*, *c4*, and the sum of the four constants. Tally F12 creates 4 bins: the flux across each of surfaces 1, 2, 3, and 4 with each multiplied by the constant *c1*. Tally F22 creates 12 bins: the flux across surface 1 plus surface 2 plus surface 3, the flux across surface 4, and the flux across all four surfaces with each multiplied by each constant *c1*, *c2*, *c3*, and *c4*. An FQ card with an entry of F M or M F would print these bins of the tallies in an easy-to-read table rather than vertically down the output file.

5.9.7.5 Example 3 (Total Energy Deposition)

```
F4:P    1
FM4    -1 2 -5 -6
SD4     1
F6:P    1
SD6     1
```

Multiplying the photon flux by volume (SD4 1) times the atom density (−1) for material 2 times the photon total cross section (−5) times the photon heating number (−6) is the same as the F6:p photon heating tally multiplied by mass (SD6 1), namely the total energy deposition in cell 1. Note that positive photon reaction numbers are photonuclear reactions. Note also that the SD card replaces the normal divisor (volume for F4 and mass for F6) with new values (both 1 in this example). By overriding the MCNP6-computed cell volume and mass with values of 1, you effectively multiply the unmodified F4 and F6 tallies by the volume and mass, respectively, yielding the score for the entire cell.

5.9.7.6 Example 4 (Track Length Criticality Estimate)

```
F4:n    1
FM4    -1 3 -6 -7
SD4     1
```

Multiplying the neutron flux by volume (SD4 1) times the atom density (−1) for material 3 times the fission multiplicity, $\overline{\nu}$ (−7), times the fission cross section (−6) gives the track-length estimate of criticality for cell 1.

5.9.7.7 Example 5 (Total Energy Deposited for Materials Not Present in Geometry)

Using MCNP6 tallies, there are two ways to obtain the energy deposited in a material in terms of rads (1 rad = 100 ergs/g). When the actual material of interest is present in the MCNP6 model, the simplest way is to use the heating tally with units MeV/g in conjunction with $c = 1.602 \times 10^{-8}$ on the companion FM card, where $c = (1.602 \times 10^{-6}$ ergs/MeV)/(100 ergs/g). When the material is not present in the model, rads can be obtained from type 1, 2, 4, and 5 tallies by using an FM card where c is equal to the factor above times $N_0\eta \times 10^{-24}/A$, where N_0 is Avogadro's number, η is the number of atoms per molecule, and $A$is the atomic weight of the material of interest. This value of c equals ρ_a/ρ_g as discussed in §2.5.4.1. The implicit assumption when the material is not present is that it does not affect the radiation transport significantly. In the reaction list on the FM card, you must enter `-4 1` for neutron heating and `-5 -6` for photon heating. For both F4 and F6, if a heating number from the data library is negative, it is set to zero by the code.

5.9.7.8 Example 6 (Lifetime Calculation/Reaction Rates)

```
F4:N   1
SD4    1
FM4    (-1 1 16:17) $ bin 1 = (n,xn) reaction rate
       (-1 1 -2)    $ bin 2 = capture (n,0n) reaction rate
       (-1 1 -6)    $ bin 3 = fission reaction rate
       (1 -2)       $ bin 4 = prompt removal lifetime=flux/velocity
M1     92235  -94.73   92238  -5.27
```

This F4 neutron flux tally from a Godiva criticality problem is multiplied by four FM bins and will generate four separate tally quantities. The user can divide bin 4 by bins 1, 2, and 3 to obtain the (n,xn) lifetime, the (n,0n) lifetime, and the (n,f) lifetime, respectively. The FM4 card entries are:

$c = -1$	multiply by atomic density of material 1
$m = 1$	material number on material card
$r1 = 16{:}17$	reaction number for (n,2n) cross section plus reaction number for (n,3n) cross section
$r2 = -2$	reaction number for capture cross section
$r3 = -6$	reaction number for total fission cross section
$r4 = 1 \quad -2$	prompt removal lifetime = flux/velocity = time integral of population

5.9.8 DE and DF: Dose Energy and Dose Function

Caution

Due to copyright concerns the built-in flux-to-dose conversion factors have been removed. They are available in Appendix F.1 formatted as MCNP input for DE/DF cards.

This feature allows you to enter a point wise response function (such as flux-to-dose conversion factors) as a function of energy to modify a regular tally, or apply a built-in conversion/response function.

Data-card Form either:

DE*n* *a* *e1* ... *eK*
and
DF*n* *b* *f1* ... *fK*
or
DF*n* IU = *value* FAC = *value* IC = *value*

n	Tally number (3). Restriction: $n \leq 99999999$
ek	The kth energy value (in MeV) (1).
fk	The value of the dose function corresponding to *ek* (1, 2).
a	Interpolation method for energy table (4, 5). If
	a = LOG logarithmic interpolation. (DEFAULT)
	a = LIN linear interpolation.
b	Interpolation method for dose function table (5). If
	b = LOG logarithmic interpolation. (DEFAULT)
	b = LIN linear interpolation.
IC = *value*	Apply a response function. If
	IC = 99 ICRP-60 equivalent dose function (neutrons) or dose equivalent (charged particles) for energy deposition tallies (6, 7).
	IC = name detector response function listed in Table 5.21. (DEFAULT: None)

The following keywords can only be used with IC = 99.

IU = *value*	Controls units. If
	IU = 1 US units (rem/h/source particle).
	IU = 2 International units (Sv/h/source particle). (DEFAULT)
FAC = *value*	Normalization factor for dose. If
	FAC = −3 Use ICRP-60 dose conversion factors for energy deposition tallies. (DEFAULT) (6, 7).
	FAC > 0 User-supplied normalization factor (7).

Default: If *a* or *b* is missing, LOG interpolation is used.

Use: Optional. Tally comment card recommended.

Details:

1 When both the DE and DF cards provide a user-specified dose table, they must have the same number of numerical entries. The DE card entries must increase monotonically. Particle energies outside the energy

range defined on these cards use either the highest or the lowest value, as appropriate.

(2) In addition to allowing user-supplied response functions, the dose conversion capability provides several built-in response functions. These are invoked by omitting the DE card and using keywords on the DF card.

(3) If *n* is zero on the DE and DF cards, the function will be applied to all tallies that do not have DE and DF cards specifically associated with them.

(4) By default MCNP6 uses logarithmic-logarithmic interpolation between the points rather than a histogram function as is done for the EM card. The energy points specified on the DE card do not have to equal the tally energy bins specified with the E card for the F tally.

(5) LIN or LOG can be chosen independently for either table. Thus any combination of interpolation (logarithmic-logarithmic, linear-linear, linear-logarithmic, or logarithmic-linear) is possible. The default logarithmic-logarithmic interpolation is generally appropriate for the ANSI/ANS flux-to-dose rate conversion factors; kermas for air, water, and tissue; and energy absorption coefficients.

(6) The IC = 99 and FAC = −3 keyword options apply dose conversion factors recommended in ICRP-60 [311] to energy deposition tallies. For neutrons, radiation weighting factors, w_{R}, are used to convert absorbed dose to ambient dose equivalent. Theses factors are calculated as

$$w_{\mathrm{R}} = 5 + 17e^{-(\ln(2E))^2/6}. \tag{5.39}$$

Charged particle energy deposition tallies use quality factors, Q, to calculate dose equivalent. The stopping power, $S(E,p)$ of the charged particles are used to calculate the quality factors based on the following formula:.

$$Q_{\text{ICRP-60}}(S(E,p)) = \begin{cases} 1 & 0 < S(E,p) \leq 10 \\ 0.32S(E,p) - 2.2 & 10 < S(E,p) \leq 100 \\ \dfrac{300}{\sqrt{S(E,p)}} & 100 > S(E,p) \end{cases}, \tag{5.40}$$

where the stopping power is in keV/μm.

(7) If FAC > 0 and IC = 99 , the tally results will be in absorbed dose (rad or Sv, depending the value of IU) /h/source particle, provided that the source strength is weighted by source particles/sec.

5.9.8.1 Example 1

Listing 5.53: example_de-df.cards.inp.txt

```
fc5 Point detector tally modified by an arbitrary user-supplied response function
f5:p 0. 0. 5. 1.
de5      0.01   0.1   0.2  0.5  1.0
df5 lin  0.062  0.533 1.03 2.54 4.6
```

In this example, a point detector tally is modified by a user-supplied dose function using logarithmic interpolation on the energy table and linear interpolation on the dose function table.

5.9.8.2 Example 2

Listing 5.54: example_de-df.cards.inp.txt

```
fc6 Helium-4 (alpha) dose equivalent (Sv)
f6:a 77
df6 IC=99 IU=2 FAC=-3
```

In this example, the ICRP-60 dose function is used to calculate the alpha particle dose equivalent in cell 77 in units of Sv/h/source particle. Note that the source strength must be weighted by source particles/sec.

5.9.8.3 Example 3

Listing 5.55: example_de-df.cards.inp.txt

```
fc26 Helium-3 detector response for tritium.
f26:t 6
df26 IC=he3-1
```

This example applies the He-3 detector response function to a tritium energy deposition tally.

5.9.9 EM: Energy Multiplier

The EM*n* card can be used with any tally (specified by *n*) to scale the usual current, flux, etc. by a response function. There should be one entry for each energy entry on the corresponding E*n* card. When a tally is being recorded within a certain energy bin, the regular contribution is multiplied by the entry on the EM*n* card corresponding to that bin. For example, a dose rate can be tallied with the appropriate response function entries. Tallies can also be changed to be per unit energy if the entries are $1/\Delta E$ for each bin, where ΔE is the width of the corresponding energy bin. Note that this card modifies the tally by an energy-dependent function that has the form of a histogram and not a continuous function. It also requires the tally to have as many energy bins as there are histograms on the EM*n* card. If neither of these two effects is desired, see the DE and DF cards.

Data-card Form: EM*n m1 ... mK*

n	Tally number. Restriction: $n \leq 99999999$
mk	Multiplier to be applied to the kth energy bin.

Default: None.

Use: Requires E card. Tally comment recommended.

Details:

1. A set of energy multipliers can be specified on an EM0 card that will be used for all tallies for which there is not a specific EM card.

5.9.10 TM: Time Multiplier

The TM card can be used with any tally to scale the usual current, flux, etc. by a response function. There should be one entry for each time entry on the corresponding T card. Note that this card modifies the tally by a time-dependent function that has the form of a histogram and not a continuous function. For example, tallies can be changed to be per unit time if the entries are $1/\Delta T$ for each bin, where ΔT is the width or the corresponding time bin.

Data-card Form: TM*n m1 ... mK*	
n	Tally number. Restriction: $n \leq 99999999$
mk	Multiplier to be applied to the kth time bin.

Default: None.

Use: Requires T card. Tally comment recommended.

Details:

1. A set of time multipliers can be specified on a TM0 card that will be used for all tallies for which there is not a specific TM card.

5.9.11 CM: Cosine Multiplier (tally types 1 and 2 only)

The CM card can be used with an F1 or F2 tally to scale the usual current by a response function. There should be one entry for each cosine entry on the corresponding C card. Note that this card modifies the tally by an angular-dependent function that has the form of a histogram and not a continuous function. For example, To get the directionally dependent F1 tally results to be per steradian, the kth entry on the CM1 card is $1/[2\pi(\cos\theta_i - \cos\theta_{i-1})]$ where θ_0 is 180°.

Data-card Form: CM*n m1 ... mK*	
n	Tally number. Restriction: $n \leq 99999999$
mk	Multiplier to be applied to the kth cosine bin.

Default: None.

Use: Tally types 1 and 2. Requires C*n* card. Tally comment recommended.

Details:

1. A set of cosine multipliers can be specified on a CM0 card that will be used for all F1 or F2 tallies for which there is not a specific CM card.

5.9.12 CF: Cell Flagging (Tally Types 1, 2, 4, 6, 7)

Particle tracks can be "flagged" when they leave designated cells and the contributions of these flagged tracks to a tally are listed separately in addition to the normal total tally. This method can determine the tally contribution from tracks that have passed through an area of interest.

The cell flag is turned on only upon leaving a cell. A source particle born in a flagged cell does not turn the flag on until it leaves the cell.

The flagged tallies are the contribution to the tally made by a particle or its progeny that ever had its cell flag set by leaving the flagged cell or cells designated on the CF card. For example, a flagged photon tally can be scored in by either a photon leaving the flagged cell or a neutron leaving a flagged cell, which leads to a photon that is tallied.

Caution

A particle that is killed on a surface will have its surface flag set but not have its cell flag.

Both a CF and an SF card can be used for the same tally. The tally is flagged if the track leaves one or more of the specified cells or crosses one or more of the surfaces. Only one flagged output for a tally is produced from the combined CF and SF card use.

Data-card Form: CF*n c1 ... cK*

n	Tally number. Restriction: $n \leq 99999999$
ck	Problem cell numbers whose tally contributions are to be flagged. A negative cell number requires that a collision occurs in that cell in order for the flag to be set upon exit from the cell.

Default: None.

Use: Not with detector (F5) tallies, DXTRAN (DXT) spheres, or pulse-height (F8) tallies; instead consider the FT card with the ICD keyword. Consider FQ card.

5.9.12.1 Example 1

```
F4:N    6   10   13
CF4     3   4
```

In this example the flag is turned on when a neutron leaves cell 3 or 4. The print of Tally 4 is doubled. The first print is the total track length flux tally in cells 6, 10, and 13. The second print is the tally in these cells for only those neutrons that have left cell 3 or 4 at some time before making their contribution to the cell 6, 10, or 13 tally.

5.9.13 SF: Surface Flagging (Tally Types 1, 2, 4, 6, 7)

Particle tracks can be "flagged" when they cross designated surfaces and the contributions of these flagged tracks to a tally are listed separately in addition to the normal total tally. This method can determine the tally contribution from tracks that have crossed a surface of interest.

The flagged tallies are the contribution to the tally made by a particle or its progeny that ever had its surface flag set by crossing the flagged surface or surfaces designated on the SF card. For example, a flagged photon tally can be scored in by either a photon crossing the flagged surface or a neutron crossing the flagged surface, which leads to a photon that is tallied.

Both a CF and an SF card can be used for the same tally. The tally is flagged if the track leaves one or more of the specified cells or crosses one or more of the surfaces. Only one flagged output for a tally is produced from the combined CF and SF card use.

Data-card Form: SF*n s1 ... sK*	
n	Tally number. Restriction: $n \leq 99999999$
sk	Problem surface numbers whose tally contributions are to be flagged.

Default: None.

Use: Not with detector (F5) tallies or DXTRAN (DXT) spheres; instead consider the FT card with the ICD keyword. Not with pulse-height (F8) tallies. Consider FQ card.

5.9.13.1 Example 1

```
F4:N    6   10
SF4     30
```

In this example, the flag is turned on when a neutron leaves surface 30. The print of Tally 4 is doubled. The first print is the total track length flux tally in cells 6 and 10. The second print is the tally in these cells for only those neutrons that have crossed surface 30 at some time before making their contribution to cells 6 or 10.

5.9.14 FS: Tally Segment (Tally Types 1, 2, 4, 6, 7)

This card allows you to subdivide a cell or a surface of the problem geometry into segments for tallying purposes without having to specify extra cells just for tallying. The segmenting surfaces specified on the FS card are listed with the regular problem surfaces, but they need not be part of the actual geometry and hence do not complicate the cell/surface relationships. The cell or surface to be segmented, however, must be part of the problem geometry.

Data-card Form: FS*n s1 ... sK* [T] [C]	
n	Tally number. Restriction: $n \leq 99999999$
sk	Signed problem number of a segmenting surface (1).
T	Optional notation at the end of the input line to require the automatic total over all bins. If absent, a total over all bins is not provided.
C	Optional notation at the end of the input line to cause the bin values to be cumulative and the last time bin to be the total over all bins.

Default: No segmenting.

Use: Not with detector (F5) tallies or DXTRAN (DXT) spheres. Not with pulse-height (F8) tallies. May require SD card. Consider FQ card.

Details:

1. If K surfaces are entered on the FS*n* card, $K+1$ surface or volume segments (and tally bins) are created. If the symbol T is on the FS*n* card, there will be an additional total bin. Tally *n* is subdivided into $K+1$ segment bins according to the order and sense of the segmenting surfaces listed on the FS*n* card as follows:

Bin #1	The portion of tally *n* with the same sense with respect to surface *s1* as the sign given to *s1*;
Bin #2	The portion of tally *n* with the same sense with respect to surface *s2* as the sign given to *s2*, but excluding that already scored in a previously listed segment.
Bin K	The portion of tally *n* with the same sense with respect to surface *sK* as the sign given to *sK*, but excluding that already scored in a previously listed segment.
Bin $K+1$	The remaining portion of tally *n* not yet tallied, i.e., everything else.
Bin $K+2$	The total tally for the entire surface or cell if T is present on the FS*n* card.

If the symbol T is absent from the FS*n* card, MCNP6 calculates the tally only for each segment (including the $K+1$ "everything else" segment). If multiple entries are on the F*n* card, each cell or surface in the tally is segmented according to the above rules. For tally types 1 or 2, the segmenting surfaces divide a problem surface into segments for the current or flux tallies. For tally types 4, 6, or 7, the segmenting surfaces divide a problem cell into segments. For normalized tallies, the segment areas (for type 2), volumes (for type 4), or masses (for types 6 and 7) may have to be provided. See the discussion under the SD*n* card.

5.9.14.1 Example 1

```
F2:N   1
FS2   -3  -4
```

This example subdivides surface 1 into three sections and calculates the neutron flux across each of them. There are three prints for the F2 tally:

1. the flux across that part of surface 1 that has negative sense with respect to surface 3;
2. the flux across that part of surface 1 that has negative sense with respect to surface 4, but that has not already been scored (and so must have positive sense with respect to surface 3); and
3. everything else (that is, the flux across surface 1 with positive sense with respect to both surfaces 3 and 4).

It is possible to get a zero score in some tally segments if the segmenting surfaces and their senses are not properly specified. In this example, if all tallies that are positive with respect to surface 3 are also all positive with respect to surface 4, the third segment bin will have no scores.

5.9.14.2 Example 2

```
F2:N     1
FS2     -3    4
```

The order and sense of the surfaces on the FS2 card are important. This example produces the same numbers as does the example in §5.9.14.1 but changes the order of the printed flux. Bins two and three are interchanged.

5.9.14.3 Example 3

```
F1:N     1  2  T
FS1     -3     T
```

This example produces three current tallies:

1. across surface 1,
2. across surface 2, and
3. the sum across surfaces 1 and 2.

Each tally will be subdivided into three parts:

1. that with a negative sense with respect to surface 3,
2. that with a positive sense with respect to surface 3, and
3. a total independent of surface 3.

5.9.15 SD: Segment Divisor (Tally Types 1, 2, 4, 6, 7)

For segmented cell volumes or surface areas defined by the FS card that are not automatically calculated by MCNP6, the user can provide volumes (tally type 4), areas (tally type 2), or masses (tally types 6 and 7) on this segment divisor card to be used by tally *n*. Tally type 1 (the current tally) is not normally divided by anything, but with the SD*1* card the user can introduce any desired divisor, for example, area to tally surface current density. This card is similar to the VOL and AREA cards but is used for specific tallies, whereas VOL and AREA used for the entire problem geometry.

Data-card Form: SD*n* (*d11 d12 ... d1M*) (*d21 d22 ... d2M*) ... (*dK1 dK2 ... dKM*)

n	Tally number. Restriction: $n \leq 99999999$
K	Number of cells or surfaces on F card, including T if present.
M	Number of segmenting bins on the FS card, including the remainder segment, and the total segment if FS has a T.
dkm	Area, volume, or mass of kth segment of the mth surface or cell bin for tally *n*.

Use: Not with detector (F5) tallies or DXTRAN (DXT) spheres. The parentheses are optional. May be required with FS card. Can be used without FS card.

MCNP6 uses the following hierarchy for determining the volume, area, or mass:

For cell or surface without segmenting (tally types 2, 4, 6, and 7):

- non-zero entry on SD card

- non-zero entry on VOL or AREA card
- volume, area, or mass calculated by MCNP6
- fatal error

For cell or surface with segmenting (tally types 2, 4, 6, and 7):

- non-zero entry on SD card
- volume, area, or mass calculated by MCNP6
- fatal error

For surface in a type 1 tally:

- non-zero entry on SD card
- no divisor

5.9.15.1 Example 1

```
F4:N    1  2  3  T
SD4     1  1  1  1
```

Note that the SD card can be used to define tally divisors even if the tally is not segmented. In this example the tally calculates the flux in the three cells plus the union of the three cells. The VOL card can be used to set the volume divisor of the three cells (to unity, for example), but it cannot do anything about the divisor for the union. Its divisor is the sum of the volumes (whether MCNP6-calculated or user-entered) of the three cells. However, the divisors for all four of the cell bins can be set to unity by means of the SD card. These entries override entries on the VOL and AREA cards. See §5.9.1.5 for use with repeated structure tallies.

5.9.16 FU: Special Tally or TALLYX Input

This card is used with a user-supplied tally modification subroutine **TALLYX** and some cases of the FT card. If the FU card has no input parameters, **TALLYX** will be called but no user bins will be created. The k entries on the FU card serve three purposes:

1. each entry establishes a separate user tally bin for tally n,
2. each entry can be used as an input parameter for **TALLYX** to define the user bin it establishes, and
3. the entries appear in the output as labels for the user bins.

Data-card Form: FU*n* [*x1* ... *xK*] [NT] [C]	
n	Tally number. Restriction: $n \leq 99999999$
xk	Input parameter establishing user bin k.
NT	Optional entry to inhibit MCNP6 from automatically providing the total over all specified bins.
C	Optional entry that causes the bin values to be cumulative.

Default: If the FU card is absent, subroutine **TALLYX** is not called.

Use: Used with a user-supplied **TALLYX** subroutine or FT card.

5.9.16.1 Programming Hint

iptal(3,1,tally_p_thread%ital) is the pointer to the location in the **tds** array of the word preceding the location of the data entries from the FU card. Thus if the FU card has the form shown above,

```
tds(L+1) = x1
tds(L+2) = x2
.
.
.
tds(L+k) = xk
```

where

$$\mathtt{L} = \mathbf{iptal(3,1,tally\_p\_thread\%ital)} \tag{5.41}$$

$$k = \mathbf{iptal(3,4,tally\_p\_thread\%ital)} - 1 = \mathbf{iptal(3,3,tally\_p\_thread\%ital)} - 1 \tag{5.42}$$

$$n = \mathbf{jptal(1,tally\_p\_thread\%ital)} \tag{5.43}$$

and **tally_p_thread%ital** is the program number of the tally.

MCNP6 automatically provides the total over all specified user bins. The total can be inhibited for a tally by putting the symbol NT at the end of the FU card, which changes the variables such that:

$$k = \mathbf{iptal(3,4,tally\_p\_thread\%ital)} - 1 = \mathbf{iptal(3,3,tally\_p\_thread\%ital)} \tag{5.44}$$

The symbol C at the end of the FU card causes the bin values to be cumulative, in which case

$$\mathbf{iptal(3,3,tally\_p\_thread\%ital)} = \mathbf{iptal(3,4,tally\_p\_thread\%ital)} \tag{5.45}$$

$$\mathbf{iptal(3,6,tally\_p\_thread\%ital)} = 1 \tag{5.46}$$

The discussion of the **iptal** and **jptal** arrays in the MCNP5 Volume III: Developer's Guide [312] and the following description of **TALLYX** may be useful.

5.9.17 TALLYX: User-supplied Tally Subroutine

TALLYX is called whenever a tally with an associated FU card but no FT card is scored. It is called for tally *n* only if an FU card is in the MCNP input file.

5.9.17.1 Programming Hint

The locations of the calls to **TALLYX** are such that **TALLYX** is the very last thing to modify a score before it is posted in the tally. **TALLYX** calls can be initiated by more than one FU*n* card for different values of *n*; a branch must be constructed inside the subroutine based on which tally F*n* is calling **TALLYX**, where *n* = **jptal(1,tally_p_thread%ital)**. **TALLYX** has the form shown in Listing 5.56.

Listing 5.56: example_default_tallyx.f90.txt

```
subroutine tallyx(t,ib)
  ! dummy for user-supplied tallyx subroutine.
  ! t is the input and output tally score value.
  ! ib controls scoring.  see the user's manual.

  ! .. Use Statements ..
  use mcnp_params, only : dknd, zero
  use fixcom, only : jtlx
  use errprn_mod, only : errprn
  use mcnp_debug

  implicit none

  ! .. Scalar Arguments ..
  real(dknd), intent(inout) :: t
  integer,    intent(inout) :: ib

  ! print a warning the first time this dummy tallyx is called.
  if(jtlx == 0)call errprn(jtlx,0,zero,zero,' ',' ',&
    & 'a tallyx subroutine is ordinarily needed with fu cards.')
  return
end subroutine tallyx
```

The quantity **t** (first argument of **TALLYX**) that is scored in a standard tally can be multiplied or replaced by anything. The modified score **t** is then put into one of the k user bins established by the FU card.

In **TALLYX(t,ib)** the second argument **ib** is defined to allow for more than one pass through **TALLYX** per tally score. By default, **ib** = 0, which means make one pass through the MCNP6 coding where user bin tally scores are posted. If the user sets **ib** < 0 in **TALLYX**, no score will be made. If the user sets **ib** > 0, passes through the user bin loop including **TALLYX** will be made until **ib** is reset to zero. This scheme allows for tally modification and posting in more than one user bin. The variable **tally_p_thread%ibu** is the variable designating the particular user bin established by the FU card. Its value is 1 before the first pass through the user bin loop. The indices of the current user, segment, cosine, energy, and time bins (**tally_p_thread%ibu**, **tally_p_thread%ibs**, **tally_p_thread%ibc**, **tally_p_thread%ibe**, and **tally_p_thread%ibt**, respectively) and the flag **tally_p_thread%jbd** that indicates flagged- or direct-versus-not are in the module **TSKCOM** for optional modification by **TALLYX**. Note that the index of the multiplier bin is not available and cannot be modified. The variable **tally_p_thread%ntx** is from the module **TSKCOM**. It is set equal to **nx** just before the **CALL TALLYX** in **TALLYD**, **TALLY**, and **TALPH**. The variable **nx** is set to unity just before the start of the user bins loop and is incremented after the **CALL TALLYX**, so **tally_p_thread%ntx** contains the number of the **TALLYX** call. An example of using **tally_p_thread%ntx** to tally in every user bin before leaving the user bin loop is shown in Listing 5.57.

Listing 5.57: example_tallyx_ubin_score.f90.txt

```
subroutine tallyx(t, ib)
  use mcnp_params
  use mcnp_global
  use mcnp_debug
```

```
  use tskcom, only: tally_p_thread
  use basic_tally, only: iptal

  implicit none

  ! .. Scalar Arguments ..
  real(dknd), intent(inout) :: t
  integer, intent(inout) :: ib

  t = 1d0 ! Whatever you want.
  tally_p_thread%ibu = tally_p_thread%ntx
  ib = 1
  if (tally_p_thread%ntx > iptal(3, 4, tally_p_thread%ital) - 1) ib = 0
  return
end subroutine tallyx
```

If **tally_p_thread%ibu** is out of range, no score is made and a count of out-of-range scores is incremented. If excessive loops through **TALLYX** are made, MCNP6 assumes **ib** has been incorrectly set and terminates the calculation with a **BAD TROUBLE** error (excessive means greater than the product of the numbers of bins of all kinds in the tally). Several examples of the FU card and **TALLYX** appear in Section 10.2.8. The procedure for implementing a **TALLYX** subroutine is the same as for the user-provided **SOURCE** subroutine.

5.9.18 FT: Special Treatments for Tallies

Data-card Form: FT*n id1 p11 p12 p13 ... idK pK1 pK2 pK3 ...*

n	Tally number. Restriction: $n \leq 99999999$
idk	The alphabetic keyword identifier for a special treatment (see Table 5.20)
pkj	Input parameters for the special treatment specified by *idk*: either a number, a parenthesis, or a colon (1).

Default: If the FT card is absent, there is no special treatment for tally *n*.

Use: Optional; as needed.

Details:

1. The syntax and meaning of the *pkj* is different for each *idk*. A special treatment may cause a set of user bins or possibly a set of some other kind of bins to be created. The information in the *pkj* allows the number and kind of those bins to be inferred easily. More than one special treatment can be specified by a given tally except for combinations of INC, ICD, SCD, PTT, PHL, RES, TAG, and FFT. Only one of these special treatments can be used by a tally at one time because all require user bins making them mutually exclusive.
2. Some FT treatments require an FU card; treatments that require or allow an FU card are not compatible with each other.
3. The SPM, FNS, and LCS treatments are incompatible with other FT options.

Descriptions of the available special treatments follow with an explanation of the allowed parameters for each.

Table 5.20: Special Treatment for Tallies Card (FT)

Keyword	§	Description
FRV	5.9.18.1	Fixed arbitrary reference direction for tally 1 or 2 cosine binning.
GEB	5.9.18.2	Gaussian energy broadening.
TMC	5.9.18.3	Time convolution.
INC	5.9.18.4	Identify the number of collisions (2).
ICD	5.9.18.5	Identify the cell from which each detector score is made (2).
SCX	5.9.18.6	Identify the sampled index of a specified source distribution.
SCD	5.9.18.7	Identify which of the specified source distributions was used (2).
ELC	5.9.18.8	Electron current tally.
PTT	5.9.18.9	Put different multigroup particle types in different user bins (2).
PHL	5.9.18.10	Pulse-height light tally with anticoincidence (2).
CAP	5.9.18.11	Coincidence capture.
RES	5.9.18.12	Heavy-ion and residual isotopes (2).
TAG	5.9.18.13	Tally tagging (2).
LET	5.9.18.14	Linear energy transfer. Energies as stopping powers.
ROC	5.9.18.15	Receiver Operator Characteristic (ROC) curve generation.
PDS	5.9.18.16	Point detector sampling.
FFT	5.9.18.17	First fission tally (2).
COM	5.9.18.18	Compton image tally.
SPM	5.9.18.19	Scatter probability matrix (3).
MGC	5.9.18.20	Flux weighted multigroup cross sections.
FNS	5.9.18.21	Induced fission neutron spectra (3).
LCS	5.9.18.22	Legendre coefficients for scatter reactions (3).

5.9.18.1 FRV *v1 v2 v3*

The *vi* are the (x, y, z) components of vector $\mathbf{v}$, not necessarily normalized. If the FRV special treatment is in effect for a type 1 or 2 tally, the direction $\mathbf{v}$ is used in place of the vector normal to the surface as the reference direction for getting the cosine for binning.

5.9.18.2 GEB *a b c*

The parameters specify the full width at half maximum (FWHM) of the observed energy broadening in a physical radiation detector,

$$\text{FWHM} = a + b\sqrt{E + cE^2}, \tag{5.47}$$

where E is the energy of the particle. The units of *a*, *b*, and *c* are MeV, $\text{MeV}^{1/2}$, and MeV^{-1}, respectively. The energy actually scored is sampled from a Gaussian with that FWHM.

5.9.18.3 TMC *a b*

All particles should be started at time zero. The tally scores are made as if the source was actually a square pulse starting at time *a* and ending at time *b*.

5.9.18.4 INC

No parameters follow the INC keyword but an FU card is required. Its bin boundaries are the number of collisions that have occurred in the track. The user can control if secondary particles are considered

un-collided (default) or collided at their creation with use of the UNC card. If the INC special treatment is in effect, the call to **TALLYX** that the presence of the FU card would normally trigger does not occur. Instead **ibu** is set by calling **JBIN** with the number of collisions as the argument. To capture all particles, the last FU bin value should be a very large number.

5.9.18.5 ICD

No parameters follow the keyword ICD but an FU card is required. Its bins are the names of some or all of the cells in the problem. If the cell from which a detector score is about to be made is not in the list on the FU card, the score is not made. The result is that the detector tally is subdivided into bins according to which cell had the source or collision resulting in the detector score. **TALLYX** is not called. The selection of the user bin is done in **TALLYD**.

5.9.18.6 SCX *k*

The parameter *k* is the name of one of the source distributions and is the *k* that appears on the SI*k* card. One user bin is created for each bin of source distribution *k* plus a total bin. The scores for tally *n* are then binned according to which bin of source distribution *k* the source particle came from. The score of the total bin is the score you would see for tally *n* without the special treatment, if source distribution *k* is not a dependent distribution.

⚠ Caution

For a dependent distribution, the score in the total bin is the subtotal portion of the score from dependent distribution *k*.

5.9.18.7 SCD

No parameters follow the keyword SCD but an FU card is required. Its bins are a list of source distribution numbers from SI*k* cards. The scores for tally *n* are then binned according to which distribution listed on the FU card was sampled. This feature might be used to identify which of several source nuclides emitted the source particle. In this case, the source distributions listed on the FU card would presumably be energy distributions. Each energy distribution is the correct energy distribution for some nuclide known to the user and the probability of that distribution being sampled from is proportional to the activity of that nuclide in the source. The user might want to include an FC card that tells to what nuclide each energy distribution number corresponds.

⚠ Caution

If more than one of the source distributions listed on the FU card is used for a given history, only the first one used will score.

5.9.18.8 ELC *c*

The single parameter *c* of ELC specifies how the charge of a particle is to affect the scoring of an F1 tally. Normally, an F1 tally gives particle current without regard for the charge of the particles. Additionally, this treatment can create separate bins for particles and antiparticles. There are three possible values for *c*:

$c = 1$	to cause negatively charged particles to make negative scores,
$c = 2$	to put charged particles and antiparticles into separate user bins, and
$c = 3$	for the effect of both $c = 1$ and $c = 2$.

If $c = 2$ or 3, three user bins (e.g., positrons, electrons, and total) are created.

5.9.18.9 PTT

No parameters follow the keyword PTT but an FU card is required. Its bins are a list of atomic weights in units of MeV of particles masquerading as neutrons in a multigroup data library. The scores for tally *n* are then binned according to the particle type as differentiated from the masses in the multigroup data library. For example, 0.511 0 would be for electrons and photons masquerading as neutrons.

5.9.18.10 PHL

The PHL keyword has the form:

PHL [*N ta1 ba1 ta2 ba2 ... taN baN*] [*det1*]
[*M tb1 bb1 tb2 bb2 ... tbM bbM*] [*det2*]
[*J tc1 bc1 tc2 bc2 ... tcJ bcJ*] [*det3*]
[*K td1 bd1 td2 bd2 ... tdK bdK*] [*det4*] [0] [TDEP *tg tt*]

The PHL option models a pulse-height tally with anti coincidence. This option allows the F8 tally to be based on energy/light deposition in up to four regions as specified via F6 tallies. Requires an FU card.

The parameters for keyword PHL are the following:

N	is the number of F6 tallies for the first detector region,
taN baN	are the pairings of tally number and F-bin number (see 5.9.19) for the *N* F6 tallies of the first detector region,
det1	is an optional detector descriptor chosen from Table 5.21 for the first detector region,
M	is the number of F6 tallies for the second detector region,
tbM bbM	are the pairings of tally number and F-bin number for the *M* F6 tallies of the second detector region,
det2	is an optional detector descriptor chosen from Table 5.21 for the second detector region,
J	is the number of F6 tallies for the third detector region,
tcJ bcJ	are the pairings of tally number and F-bin number for the *J* F6 tallies of the third detector region,
det3	is an optional detector descriptor chosen from Table 5.21 for the third detector region,
K	is the number of F6 tallies for the fourth detector region,

Table 5.21: Detector Descriptors for the FT PHL Option

Detector Type	Detector Name	Primary Particle Type(s)	Response Parameter	Default Value	Notes
^{3}He	HE3-1	Proton, Triton, Helion	Multiplication	100	42.3 eV/ion pair
BF_3	BF3-1	Alpha, Lithium	Multiplication	100	36.0 eV/ion pair
Li Glass	LIG-1	Triton, Alpha	Quenching Factor	5.0×10^{-4} cm/MeV	Generic value
LiI	LII-1	Triton, Alpha	Quenching Factor	5.0×10^{-4} cm/MeV	Generic value
ZnS+LiF	ZNS-1	Triton, Alpha	Quenching Factor	5.0×10^{-4} cm/MeV	Generic value
NaI	NAI-1	Electron	Quenching Factor	3.4×10^{-4} cm/MeV	[313]
BGO	BGO-1	Electron	Quenching Factor	6.5×10^{-4} cm/MeV	[314]
CsI	CSI-1	Electron	Quenching Factor	1.5×10^{-4} cm/MeV	[314]
BC-400	BC4-1	Electron	Quenching Factor	4.6×10^{-3} cm/MeV	[315]
HPGe	HPG-1	Electron	Gain	1.0	3.0 eV/ion pair

tdK bdK	are the pairings of tally number and F-bin number for the *K* F6 tallies of the fourth detector region,
det4	is an optional detector descriptor chosen from Table 5.21 for the fourth detector region,
0	a zero entry terminates input for PHL detectors entries and allows for other FT options to follow,
TDEP *tg tt*	is a keyword option that specifies a tally that will be used as a trigger for the related T8 card. The first optional TDEP entry (*tg*) specifies the trigger tally number and the second optional TDEP entry (*tt*) specifies an energy threshold (MeV).

The F-bin descriptor specified after each tally, *bai* or *bbi*, may be "0", indicating that the referenced tally includes a lattice description of multiple lattice elements. When this option is specified, all tallies within that PHL region must also include the "0" descriptor for *bai* or *bbi* and all tallies must be over the same lattice cell and elements. When this option is used in both PHL regions, the related F8 F-bins are modified, with an appropriate warning message, to include $J \times K$ bins, where J is the number of lattice elements included in PHL Region 1 and K is the number of lattice elements included in PHL Region 2. The output of Tally 8 will include coincidence results for all $J \times K$ bins, along with appropriate cell labels (e.g., $1[0\,1\,1] + 2[0\,0\,0]$, which is the combination of lattice cell 1, element [0 1 1], with lattice cell 2, element [0 0 0]). This special F-bin descriptor is typically used with the FT COM option to create a Compton image of a radiation source.

When a detector descriptor is specified, built-in particle-dependent response functions are automatically applied to all listed tallies (e.g., *ta1*, *ta2*, ... for *det1*). For photon detectors, these include material-dependent electron response functions (light output, current, etc.). For neutron detectors, these include material-dependent electron or light-ion response functions. Additional details on references regarding these parameters can be found in the source code (**Source/src/fluence_to_dose.F90**).

The gas detectors are treated by multiplying the charged-particle energy deposition by the inverse of the gas work function (see Details in §5.9.18) and the detector E-field multiplication (response parameter). The units for this response function are pico-Coulombs (pC) per source particle, thus this detector response is further multiplied by the electron charge per ion pair (1.6×10^{-7} pC). The user can override the default multiplication by appending an underscore and a multiplication value to the detector name (e.g., HE3-1_25.0).

The scintillation detectors are treated by Birks's Law [316], which is generally in good agreement with measured data for $Z < 6$ and particle energies less than ≈ 50 MeV/amu. The stopping powers used in Birks's Law are the total stopping powers calculated by the MCNP code for each particle type. The units for this response function are 1-MeVee photons per source particle (an MeVee is MeV electron-equivalent). For absolute visible light photons per source particle, one must multiply this response by the number of visible light photons produced by a 1-MeV electron (which is typically given by the detector manufacturer or can be found in the literature). The default Birks's quenching factors (QF) are given in Table 5.21, however

the user may override these values by appending an underscore and a QF value to the detector name (e.g., LIG-1_2.5e-3).

The semi-conductor detector is treated similarly to a gas detector, except the work function is typically much lower (≈ 3 eV/ion pair) and the multiplication is replaced by the gain. The units for this response are also pC per source particle. The user can override the default gain by appending an underscore and new gain value to the detector name (e.g., HPG-1_2.5).

When *M* is non-zero, indicating the use of two or more detector regions, an FU card is required for the F8 tally. The entries on the FU card are presented in units of MeV (unless modified by DE/DF cards associated with the specified F6 tallies) and must increase monotonically. Similarly, if *J* or *K* is non-zero, the energy bins must be specified with C (tally cosine) and FS (tally segment) cards, respectively. The particle type indicated on the F8 tally does not matter because this tally allows a combination of light output from various particle types. If *baN* is zero, then the number of cell bins on the F8 card must match that on the corresponding *taN* tally card, which is useful for a lattice pulse-height PHL tally.

The TDEP keyword allows the T8 values to be relative to the first contribution to any FT8 PHL tally. Invoking TDEP allows pulse distributions from different histories to be relative to the same start time rather than distributed in absolute time with significant variation based on when a particle reaches the detector. TDEP can also be followed by one or two entries, where the first entry (*tg*) is a tally number and the second (*tt*) is an energy threshold. If an energy threshold value is provided, the reference time on the T8 card is whenever the specified tally has a value greater than the specified threshold. The specified tally can be the same number as the F8 tally, in which case TDEP depends on the sum of the PHL F6 tallies, or it can be a single F6 tally that is specified in any region of the FT PHL option. If the tally number has a format of 8.3 (e.g., F8.3), for example, then the trigger tally is the sum of all F6 tallies that are specified for region 3 of the FT PHL option. The default TDEP tally number is the corresponding F8 tally number and the default energy threshold is 0 MeV.

5.9.18.10.1 Time-dependent F8 Tallies Using the Pulse-height Light (PHL) Option

The T (time bin) card is allowed with pulse-height tallies (F8), but only when used in conjunction with the FT PHL option. In this case, the time-dependent energy deposition is taken from the associated F6 tally(s). If the time entries on the F8 card do not match those provided for the various F6 tallies, a fatal error is issued. If the associated F6 tallies do not have T cards, then one matching the F8 tally will be created automatically.

5.9.18.10.2 Example 1

Case 1

```
F8:N      5
FT8   PHL  1  6  1  0
      GEB  a  b  c
E8        1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0
F6:E      5
DE6   LIN  1.0  1.5   2.0   2.5   3.0   3.5  10.0
DF6   LIN  1.0  0.99  0.98  0.97  0.96  0.95  0.92
```

Case 2

```
F8:N 5
FT8   PHL  1  6  1  1  16  1 0
      GEB  a  b  c
E8         1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0
FU8        1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5
F6:E       5
DE6   LIN  1.0  1.5   2.0   2.5   3.0   3.5  10.0
DF6   LIN  1.0  0.99  0.98  0.97  0.96  0.95  0.92
F16:E      6
DE16  LIN  1.0  1.5   2.0   2.5   3.0   3.5  10.0
DF16  LIN  1.0  0.99  0.98  0.97  0.96  0.95  0.92
```

In both cases, the F6 tallies convert energy deposition to equivalent light (units of MeV, photons, or MeVee, depending on the units of the associated DF card). SD cards are not required with the F6 tallies because these divisors renormalize only the printed output for the F6 tallies and not the values stored in the tally arrays (thus, the F8 tally will result in the same value, regardless of whether the F6 tally has an SD card). The DE/DF conversion is based on the incident particle energy, and the values on the DF card should be the $\mathrm{d}L/\mathrm{d}E$ for that incident particle energy. Thus, the F6 tally will multiply the $\mathrm{d}L/\mathrm{d}E$ values by the energy deposition to give the light output (ΔL) summed over each track. Also, no energy bins exist for the F6 tallies. The F8 tally uses the total light output. Energy bins (E6 card) can be added, but the F8 tally will use the value from the total bin. Similarly, for other bins associated with the F6 tally, in each case, the tally fluctuation chart bin is used to extract the value for the F8 tally (see the TF card to alter this). The FT GEB cards are used to perform Gaussian broadening on these tally values; however, this is done only at the end of the particle history to determine the light output value used in the pulse-height tally.

In Case 1, the electron light output from only one region (cell 5) is used to subdivide the pulse-height tally. In this case, a pulse of 1 (input source weight) is put into the first E8 bin when the light output in cell 5 is < 1 MeV. It is placed in the second E8 bin when the light output is between 1 and 2 MeV, etc. A zero F6 tally will result in no F8 tally.

In Case 2, the light output from two regions (cells 5 and 6) is used to subdivide the pulse-height tally. This case is useful for coincidence/anti-coincidence applications. A pulse of 1 (input source weight) is put into the second E8 bin and into the second FU8 bin when the light output in cell 5 is $0 < L < 1.0$ MeV and the light output in cell 6 is $0 < L < 1.5$ MeV. This pulse is put into the second E8 bin and into the third FU8 bin when the light output in cell 5 is $0 < L < 1.0$ MeV and the light output in cell 6 is between 1.5 and 2.5 MeV. A zero light output in both cells will result in no F8 tally. A zero light output in cell 5 (tally 6) with a non-zero light output in cell 6 (tally 16) will result in a pulse in the first E8 bin and the corresponding FU8 bin. Similarly, for a zero light output in cell 6 and a non-zero light output in cell 5, a pulse will be put into the first FU8 bin and the corresponding E8 bin. Note that the E8 and FU8 bins do not have to be the same and typically would not be unless the detector regions were of similar material and size. Separate F6 tallies (as in Case 2, F6 and F16) are needed only when the two regions have different light conversion functions. If the two regions are of the same material, then a single F6 tally can be used as follows:

5.9.18.10.3 Example 2

```
F8:N 5
FT8   PHL  2  6  1  6  2  0
E8         1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0
FU8        1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5
F6:E       5  6
DE6   LIN  1.0  1.5  2.0  2.5  3.0  3.5  10.0
```

```
DF6   LIN  1.0  1.1  1.2  1.3  1.4  1.5  1.6
```

In this example, the light output from the two regions (cells 5 and 6), which are included on the same F6 tally, is used to subdivide the pulse-height tally.

Currently, it is not important what cell is listed on the F8 card because this tally is made only at the end of a particle history and depends only on the tally results of the listed F6 tallies. Having multiple cells listed on the F8 card is meaningful only when the F-bin parameter (i.e., *baN* or *bbN*) of the FT PHL option is zero, indicating a lattice grid of detector regions. Otherwise, each additional F8 cell bin simply will be a duplicate of the first cell bin.

5.9.18.10.4 Example 3

```
F6:H    1
F16:T   1
F8:h,t  1
FT8     PHL  2 6 1 16 1  0
T8      10 20 30 40 50 60 70 80 90 100 1e37
```

In this example, the proton (F6) and triton (F16) energy depositions in cell 1 are combined into a pulse-height tally (F8) using the FT PHL option. The time-dependent behavior of these pulses is segregated into 11 time bins: 0–10 shakes, 10–20 shakes, etc. To obtain the time-dependent pulse shape, time-dependent energy depositions are obtained from the tallies identified by the PHL option. To accomplish this, the 11 specified T8 bins are applied to the associated F6 and F16 tallies with the automatic creation of matching T6 and T16 cards. A warning message is generated when these cards are created.

5.9.18.11 CAP [-*mc*] [-*mo*] i1 i2 [GATE *td tw*] [EDEP *tg tt*]

The FT8 capture tally scores the number of captures in specified combinations of nuclides at the end of each history. Time gating with pre-delay and gate width treatments is optional [317]. It is particularly useful for neutron coincidence detectors. In addition, captures may be written to an auxiliary output file, **PTRAC**. Section 5.13.6 describes the **PTRAC** capture file.

The FT8 CAP option converts the pulse-height tally to a neutron capture tally. Variance reduction is no longer allowed, time bins are allowed (unlike other F8 tallies), cosine bins are used to store capture frequencies and moments, and PRINT table 118 is created in the MCNP output file.

The parameters for keyword CAP are described as follows:

mc	is the optional maximum number of captures (Default is 21),
mo	is the optional maximum number of moments (Default is 12), and
in	are the capture nuclide identifiers [§1.2.2]. All formats are supported.

In addition, the time-gate keyword GATE may appear with its parameters, *td* and *tw*, where

td	is the pre-delay time and

tw	is the gate width;

and the energy deposition keyword EDEP may appear with its parameters, *tg* and *tt*, where

tg	is the trigger tally number and
tt	is the trigger tally threshold (MeV) (Default is 0.0).

The EDEP keyword specifies to record a capture whenever tally *tg* produces an energy deposition greater than *tt*. Tally *tg* can be any F6 or F8 tally, but is usually the related F8 tally of the FT CAP option (which is the default).

5.9.18.11.1 Example 1

```
F8:N       2  (5  6)  7  T
FT8   CAP  Li-6   B-10
T8         1   7LOG   1E8
```

In this example, captures and moments are tallied in cells 2, 7, in the combination of 5 and 6 and in the total over cells 2, 5, 6, 7. The captures by either ^{6}Li or ^{10}B are tallied. Results are tabulated in time bins at 1, 10, 100, 1000, 10^4, 10^5, 10^6, 10^7, and 10^8 shakes—that is, in the range of 10 nanoseconds to 1 second.

5.9.18.11.2 Example 2

```
F8:N          4
FT8 CAP    He-3   GATE  0.5  0.4
```

In this example, ^{3}He captures and moments are tallied in cell 4. There is a time gate with a pre-delay treatment of 0.5 shakes (5×10^{-9} seconds) and a width of 0.4 shakes (4×10^{-9} seconds).

5.9.18.11.3 Example 3

```
*F8:H,T   999
F18:N      999
FT18       CAP  EDEP  8  0.001
```

In this example, a capture is scored in Tally 18 whenever there is an F8 tally that exceeds 0.001 MeV.

The addition of the pre-delay and time gate width changes the capture tally scoring. When a neutron is captured at time t_0 in the specified cell by the specified nuclide, the gate is "turned on." If the pre-delay duration is t_1 and the gate width is t_2, then all captures between $t_0 + t_1$ and $t_0 + t_1 + t_2$ are counted. For a history with no captures, no events are scored. With one capture, zero events are scored. With two captures,

the first turns on the time gate at time t_0 and scores 0; the second will score one event if it is captured between $t_0 + t_1$ and $t_0 + t_1 + t_2$, or score another 0 if outside the gate.

Caution

Coincidence counting of capture multiplicities and moments requires analog capture: `CUT:n 2J 0 0`. Calculations must be totally analog with no variance reduction. Fission multiplicity also is required: `PHYS:n J 100 3J -1`. An `FT8 CAP` tally in an input file will automatically set analog capture, fission multiplicity, and exit with error messages if variance reduction is used.

The capture tallies may be written to a **PTRAC** file for further analysis by auxiliary codes. See §5.13.6 on the **PTRAC** card extensions.

5.9.18.12 RES [*z1 z2*] or RES [*za1 za2* ...]

The interaction of high-energy particles with target nuclei causes the production of many residual nuclei. The generated residual nuclei can be recorded to an `F8` tally if used with an `FT8 RES` special treatment option. The residuals are recorded at each physics model interaction as well as each neutron library interaction. The residual data can be accumulated for the entire geometry (when no cells are listed) or for specific cells listed on the `F8:#` card. A specific list of targets may also be requested on the `FT RES` card. Requires an `FU` card.

The `FT8 RES` capability can also be used with type 1, 2, 4, and 6 heavy-ion tallies (`Fn:#`) to segregate the score into bins according to the heavy ion that produced the score.

By default, the `FT RES` card with no entries causes the corresponding tally to create a user bin for each of the 2200+ possible residual nucleus ion types. A range of bins may be selected by specifying lower and upper proton numbers, z_1 and z_2, which correspond to a range of possible Z values. If z_1 and z_2 are specified and a residual is generated with a higher or lower Z, the residual will not be scored in the tally. To specify an explicit list of heavy ions to be tallied, provide target identifiers [§1.2.2] after the `RES` keyword. All formats are supported. Specifying an elemental identifier, such as Fe-0 for iron, will include all nuclides of that element into a single bin. Metastable residuals will be included with non-metastable counterparts. A metastable input is not allowed.

When used with the `F8:#` tally, the `FT RES` card yields a list of residual nuclides produced by all neutron-induced reactions and model reactions of all incident particle types (photon and proton library reactions do not yet produce residuals). The residual tallies can be obtained either with or without the emission of delayed neutrons and/or delayed gammas. Residual tallies can be obtained for analog or non-analog (implicit capture) neutron transport. The residuals are just the residuals of the nuclear reactions and not their decay products.

For models that include light-ion recoil and the neutron capture ion algorithm (NCIA) (activated using the 7th entry on the `PHYS:n` card), reaction residuals are included in the `FT8 RES` tally. In most instances, reaction residuals are determined using the ENDF reaction specifications for simple-multi-particle reactions. In rare instances, e.g., neutron bombardment of ^{6}Li(n,t)α, the ENDF reaction specifications can result in only light-ion production. In such cases, the heaviest light-ion residual is selected.

5.9.18.12.1 Example 1

```
F4:#     6
FT4 RES  O-16  Ca-40  Fe-0  U-238
```

This combination of tally cards creates a track-length tally in cell 6 and then creates three user bins for the nuclides ^{16}O, ^{20}Ca, and ^{238}U. It also creates one bin for all iron nuclides.

5.9.18.12.2 Example 2

```
F8:#   1  100  T
FT8    RES  25  27
```

The entries on the F8 tally card are cell numbers for which residuals are to be tallied. In this example, residual tallies are requested for cell 1, cell 100, and for cells 1 and 100 combined. The entries on the FT8 RES card specify the range of possible Z values for which to tally the residuals. Here, residuals with atomic numbers between (and including) $Z = 25$ and $Z = 27$ will be scored.

5.9.18.12.3 Example 3

```
F8:#   1  100  T
FT8    RES  Mn-54 Mn-55 Mn-56  Fe-55 Fe-56 Fe-57  Co-56 Co-57 Co-58
```

The entries on the F8 tally card are cell numbers for which residuals are to be tallied. In this example, residual tallies are requested for cell 1, cell 100, and for cells 1 and 100 combined. The entries on the FT8 RES card specify a list of isotopes for scoring residuals. Production for specific isotopes of manganese, iron, and cobalt will be included for this F8 tally.

The FT8 RES capability is particularly useful with the eighth LCA card entry, *noact*. When *noact* $= -2$ on the LCA card, the source particle immediately collides in the source material. All subsequent daughter particles then are transported without further collision, as if in a vacuum. The F8 tally with an FT8 RES special tally treatment is then simply the distribution of nuclides resulting from a single collision.

For additional information involving fission multiplicity see the discussion presented in §5.7.9. More capture tally information and examples appear in §10.2.5.5 and §10.2.5.6. To inspect a residual nuclei tally example, see §10.2.5.8.

5.9.18.13 TAG *a*

Tally tagging allows the user to separate a tally into components based on how and where the scoring particle was produced. This feature is available for both standard (F1, F2, F4, F6, F7) and detector (F5) tallies. Requires an FU card.

The single required parameter *a* of the keyword TAG specifies how scatter is to be treated (i.e., whether the creation tag on a particle should be retained or a separate scatter tag be invoked). More specifically, if

$a = 1$	particles undergoing elastic scattering will lose their tag and bremsstrahlung and annihilation photons will be included in the "scatter" bin (i.e., FU "0" bin);
$a = 2$	particles undergoing elastic scattering will lose their tag, but bremsstrahlung and annihilation photons will be segregated (see appropriate FU bins below);
$a = 3$	particles undergoing elastic scattering will retain their production tag. If a particle has multiple production events, the tag will be for the last production event. For example, a neutron undergoing fission followed by (n,2n) would have the (n,2n) tag. If a particle undergoes an elastic scatter, its previous tag is retained (i.e., no need for FU "0" bin);

$a = 4$	same conditions as $a = 3$ except Compton photoatomic interactions retain their tag. Neutron interactions behave identically as $a = 3$.

Binning specifications for the tagged tally must be provided on the FU special tally card. Each *bini* entry on the card requests three distinct pieces of tagging information:

1. a cell of interest where particles are produced;
2. a target nuclide from which the particle is emitted; and
3. a reaction MT identifier, or, in the case of spallation, a residual nuclide of interest, or a special designator (see below).

The format on the FU card when used in association with the tagging treatment is FU*n bin1 bin2 ... binN* [NT] where each tagging *bini* has the form *CCCCCZZAAA.RRRRR* and

CCCCC	represents a user cell number. Note: leading zeros are not required.
ZZAAA	represents a 5-digit target identifier [§1.2.2] in the ZAID form for a target nuclide where *ZZ* is the atomic number and *AAA* is the atomic mass number. Note: *ZZ* is limited to two characters, therefore nuclides with $Z > 99$ cannot be tagged.
RRRRR	specifies a reaction identifier for library interactions or a residual nuclide *ZZAAA* identifier for high-energy model interactions or a special designator. Some MCNP reaction numbers (*RRRRR*) of special designators have different meanings from ENDF reaction (MT) numbers.

By default, a total over all specified bins is provided for the FU special tally; add the NT parameter after the last specified bin to suppress this total. A list of special cases for the *CCCCCZZAAA.RRRRR* FU card entries appears later in this section.

If cell tagging is not desired, the *CCCCC* portion of the tag should be omitted or, alternatively, set to "00000". In either case, tally contributions will be accumulated for all cells for that FU bin, provided the *ZZAAA.RRRRR* portion of the tag is satisfied. In the case of particle production from electrons, which are material based (not nuclide specific), the *CCCCC* input should be used to identify the cell and the *ZZAAA* input should be set to "00000". The suffix *RRRRR* refers to a standard or special ENDF reaction (MT) number for library interactions. For example, "00102" stipulates (n,γ) or, in the case of high-energy model interactions, *RRRRR* refers to a residual nuclide *ZZAAA* identifier (e.g., "06012" for ^{12}C).

In general, a zero input for any portion of the tag results in the sum of all contributions related to the entry. For example, the tag "0000092000.00000" will collect all tally contributions for which any isotope of uranium ($Z = 92$) produced the particle making the tally. However, the tag "0000000000.00000" is reserved for elastic-scattered particles. Note that each tally contribution is made only to the first FU bin that satisfies the tag description (i.e., those that have not already been tallied). If no appropriate FU bin is found, the tally contribution is not made; however a special "everything else" *bini* (i.e., "1e10") can be specified to collect any portion of the tally that falls into no other bin. When the "everything else" bin is used, then the user is assured that the "user-bin total" bin will reproduce the original tally as if the FTn TAG option had not been used.

Special designations for *CCCCCZZAAA*:

-0000000001 or -1	source particle tag for all cells

−*CCCCC*00001	source (i.e., un-collided) particle tag for cell *CCCCC*
0000000000 or 0	elastic-scattered particle tag
10000000000 or 1e10	everything else tag

Photon tally special designations for *ZZAAA.RRRRR*:

00000.00001	bremsstrahlung from electrons
*ZZ*000.00003	fluorescence from nuclide *ZZ*
00000.00003	K x-rays from electrons
00000.00004	annihilation photons from electron-positron interactions
*ZZ*000.00005	Compton photons from nuclide *ZZ*
ZZAAA.00006	muonic x-rays from nuclide ZZAAA
00000.00007	Cerenkov photons

Electron tally special designations for *ZZAAA.RRRRR*:

*ZZ*000.00001	photoelectric from nuclide *ZZ*
*ZZ*000.00003	Compton recoil from nuclide *ZZ*
*ZZ*000.00004	pair production from nuclide *ZZ*
*ZZ*000.00005	Auger electron from nuclide *ZZ*
00000.00005	Auger electron from electrons
00000.00006	knock-on electrons

Neutron and photon tally special designations for *ZZAAA.RRRRR*:

ZZAAA.99999	delayed particles from fission of *ZZAAA*

The *RRRRR* reaction tag also includes all the MT reactions listed for neutrons, but selecting *RRRRR* is complicated by the inconsistencies of ENDF and other table data evaluations. For fission, *RRRRR* = 18 will not always catch all fission reactions. For example, in ^{239}Pu *RRRRR* = 18, but in ^{240}Pu *RRRRR* = 19, 20, 21 and all three must be listed to catch ^{240}Pu fission. Likewise, *RRRRR* = 16 only tags (n,2n) reactions; *RRRRR* = 17 must be used to get (n,3n) reactions. And then there are the exceptions. For example, photons from fission in ^{235}U have the tag *RRRRR* = 3, which is inelastic, and is also the tag of photons created by (n,xn).

5.9.18.13.1 Example 1

```
F1:N   10
FT1    TAG  1
FU1    0000092235.00016  0000092235.00000  1e10
```

If an (n,2n) neutron that is produced from an interaction with ^{235}U contributes to the `F1` tally, then its contribution will be included only in the first `FU` bin even though its tag also will satisfy the criteria for the 2nd `FU` bin. Thus, the order of the `FU` bin tags is important for segregating the tally. Note that neutrons produced by some other reaction with ^{235}U will be placed in the 2nd `FU` bin and neutrons produced by reactions with other target nuclides will be placed in the last ("everything else") bin. The sum of these three bins should preserve the value of the original `F1`:n tally.

5.9.18.13.2 Example 2

```
F1:P  1
FT1   TAG  1
FU1   0.0  01001.00102  01001.00000
           26056.00102  26056.00051  26056.00052
           26056.24052  26056.26053  26056.26054  26056.26055
           26056.00000
```

All elastic-scattered photons (i.e., coherent) will be put into the `FU` "0.0" bin. All capture gammas from ^{1}H will be put into the `01001.00102` bin; all remaining gammas from ^{1}H interactions will be put into the `01001.00000` bin. All capture gammas from ^{56}Fe will go into the `26056.00102` bin; all (n,n′) 1st level gammas will go into the `26056.00051` bin; all (n,n′) 2nd level gammas will go into the `26056.00052` bin; all de-excitation gammas from the spallation of ^{56}Fe into ^{52}Cr will go into the `26056.24052` bin; etc. All remaining gammas produced from ^{56}Fe interactions will go into the `26056.00000` bin.

5.9.18.13.3 Example 3

```
F5:P  0 0 0 1
FT5   TAG  3
FU5   -1.0  0000106012.00005  0000106012.00000
            0000026056.00102  0000026056.00000
            0000000000.00051
           10000000000.00000
```

In this case, all collided photons will retain their original creation tag. All source photons will go into the −1.0 bin. All Compton photons from ^{12}C in cell 1 will be put into the 2nd bin; all remaining photons produced from interactions with ^{12}C in cell 1 will go into the 3rd bin. All capture gammas from ^{56}Fe will go into the 4th bin; all remaining photons/gammas produced from interactions with ^{56}Fe will go into the 5th bin. All (n,n′) 1st level gammas will be put into the 6th bin, and all remaining photons/gammas that were not included in any of the previous bins will be placed in the last bin.

5.9.18.14 LET

The linear energy transfer (LET) special tally option allows track length tallies to record flux as a function of stopping power instead of energy. When the FTn LET option is specified, the values provided in the energy bins are interpreted as stopping power values with units of MeV/cm. This option can only be applied to charged particle tallies.

5.9.18.14.1 Example 1

```
fc4     Proton flux LET
f4:h    77
e4      1e-2  99ilog  6e4
ft4     LET
```

This example is a tally that records the proton flux in cell 77 for a LET tally. The tally results are recorded in 100 bins of stopping power from 0.01 to 60000 MeV/cm.

5.9.18.15 ROC *nhb* [*m*]

The ROC special tally option separates tallies into two components, signal and noise. During a calculation, the signal and noise tally values are saved for each specified batch of histories. These distributions of tally values are formed into signal and noise probability distribution functions (PDFs). Integration of the signal PDF (labeled as the Probability of Detection, PD) is plotted as a function of the integral of the noise PDF (labeled as the Probability of False Alarm, PFA), resulting in the printed Receiver-Operator Characteristic (ROC) curve. A table of the PDF values is provided in PRINT table 163 of the MCNP output file.

To specify the "signal" portion of a tally, use entries 1–8 on an associated TF card; to specify the "noise" portion, use TF entries 9–16. The ROC keyword parameter *nhb* sets the number of histories per batch. This parameter sets the 5th entry (the tally fluctuation chart frequency) on the PRDMP card. The *nhb* value should represent the total number of source particles emitted over the time interval of interest. The *npp* value on the NPS card should be set to a multiple of *nhb*; the *npp* value will then be used to determine the number of sampled batches. We recommend that *npp* should be 50–100 times the value of *nhb*. The optional parameter m specifies the maximum number of batches that will be kept and analyzed. The default value is 100. We recommend *m* be greater than 50 and perhaps two times the number of batches planned, even considering possible restarted calculations. This value cannot be increased in a restarted-calculation input file. If there are multiple tallies with ROC entries, the maximum *m* value is used. The WGT keyword on the SDEF card should be set to the default value of unity.

5.9.18.15.1 Example 1

```
f1:n   1
t1     1e8  1e37
tf1    j j j j j j j 2  j j j j j j j 1
ft1    ROC  1000
```

In this example, tally F1 scores the current of neutrons crossing surface 1. This tally is divided into two time bins, neutrons arriving before 1 second (i.e., 10^8 shakes) and those arriving after 1 second (i.e., 10^{37} shakes). The TF card associates the second time bin as the "signal" and the first as the "noise." The signal and noise currents are accumulated for each batch of 1000 particle histories. The resulting tally values are formed into signal and noise PDFs that are integrated and plotted in PRINT table 163 as a ROC curve for this tally.

Another ROC curve example is provided in §10.2.5.9.

5.9.18.16 PDS *c*

This pre-collision estimator augments the post-collision next-event estimator that has historically been used for point flux estimation in MCNP6. The pre-collision next-event estimator includes the contribution of all possible reactions before the collision isotope and resulting reaction are sampled. This procedure has the advantage of providing an improved expected estimate per collision, but with a significant increase in computational costs per collision. This improved sampling technique removes the requirement to suppress coherent scattering for photon transport problems that include photon next-event estimators [§2.4.4.2.5]. The sampling of all possible scattering reactions generally provides an increase in the Figure of Merit (FOM) for most photon problems. This increase in the FOM can be significant when the contribution to a photon next-event estimator is primarily from forward scattering. For most neutron problems there is not typically a large increase in the FOM. However, for both photons and neutrons the pre-collision next-event estimator increases the convergence rate as measured by the time to pass MCNP6's ten statistical checks.

The single parameter, *c*, specifies how the sampling of the collision is performed for the next-event estimator. If

$c = -1$	Next-event estimator sampling is performed post-collision; only a single reaction and isotope is sampled (historic MCNP4 and MCNP5 behavior)
$c = 0$	Same as $c = -1$ (DEFAULT)
$c = 1$	Next-event estimator sampling is performed using post-collision sampling of the collision isotope and pre-collision sampling of all reaction channels. (Recommended for photons.)
$c = 2$	Next-event estimator sampling is performed using pre-collision sampling of all collision isotopes and pre-collision sampling of all reaction channels.

Recommendation: Using either PDS 1 or PDS 2 allows the user to perform next-event estimator tallies with photon coherent scattering enabled.

For neutron next-event estimator tallies the user should perform scoping calculations with PDS = -1, 1, and 2. The user should check the 10 statistical tests of the three runs to assess which parameter provides the best compromise between convergence and FOM. Using a pre-collision estimator for neutrons will typically reduce the computational time needed to pass the 10 statistical checks but result in a lower FOM.

5.9.18.16.1 Example 1

```
f5p   100.0 50.0 25.0 0.0  $ post-collision next-event estimator
ft5   pds -1               $ F5 tally is post-collision
c
f15p  100.0 50.0 25.0 0.0  $ pre-collision next-event estimator
ft15  pds 1                $ F15 tally is pre-collision
```

In this example the pre-collision next-event estimator and the post-collision next-event estimator are compared for a photon tally located at $x = 100$ cm, $y = 50$ cm, and $z = 25$ cm.

5.9.18.17 FFT [*LKJI*]

A single parameter may follow the FFT keyword and an FU card is required. The optional *LKJI* parameter toggles on/off the first-fission treatment for the various physics packages, as explained below. The related FU card segregates the tally into contributions according to which fission occurred first. FU entries must be target identifiers [§1.2.2] in ZAID form of fissionable nuclides. Additionally, an FU entry of "0" should be included to score all contributions that are not associated with any other FU bin, an FU entry of "16" will score (n,xn) reactions instead of fission if they occur before any fission, and an entry of "18" will score first fissions from any nuclide that is not listed on the FU card. The bins may be entered in any order. The *LKJI* parameter combines four binary toggles that specify which physics packages should be included with the FFT treatment, such that:

L = 0/1	Omit/include neutron and photon-induced fissions treated by model physics
K = 0/1	Omit/include neutron spontaneous fissions (PAR = SF source particles)
J = 0/1	Omit/include photon-induced fissions treated by library physics
I = 0/1	Omit/include neutron-induced fissions treated by library physics ($E <\sim 20$ MeV)

The default value for *LKJI* is 0001, which is equivalent to 1 and is the default action if *LKJI* is omitted. To turn on the full FFT treatment, one would specify *LKJI* = 1111.

Cell flagging (CF card) and surface tally segmenting (FS card) have somewhat different meanings when the FFT special tally treatment is used. Unlike the standard tally segmenting, in which the segment identifies where the score is made, FFT tally segmenting identifies where the first fission occurs. Unlike the standard tally flagging, which flags cells through which the track has passed before scoring, FFT cell flagging flags cells in which the first fission occurred. Cell flagging and surface segmenting work for cells and surfaces at the lowest level so when FFT is specified, these lowest-level cells/surfaces will be the location of the first fission. If a CF card is used with the FFT option on any tally, then the use of a CF card without the FFT option is prohibited on any other tally, and the related CF card is ignored and a warning is issued.

5.9.18.17.1 Example 1

```
FT1   FFT
FU1   92238  0  16  18  94241  92235  94239
```

If an (n,xn) reaction occurs before a fission, then the 16 bin records a score. If the particle has its first fission in a listed nuclide (U-235, U-238, Pu-239, Pu-241), then that nuclide bin records a score provided 16 has no score. If the first fission is not a listed nuclide, the 18 bin records a score provided 16 has no score. The 0 bin records a score if no other bin has a score.

5.9.18.18 COM *t a*

The FT8 COM tally option produces a Compton image stored in an associated FIR radiography tally *t* using algorithm *a* (optional, currently there is one algorithm so $a = 1$). The Compton image is formed from a FT8 PHL specification of dual-region coincidences of planar lattice tallies. At the end of each particle history, Compton/photoelectric energy deposition in the front/back of these dual-panel detectors is used to create a circular "image" of the incident photon on a specified image plane. The FT8 PHL enhancement is used to obtain coincidences of front-panel energy deposition with back-panel energy deposition, on a voxel-by-voxel (or element-by-element) basis. For example, if the front-panel detector consists of a 5×5 lattice and the back-panel detector consists of a 10×10 lattice, then the FT8 PHL option produces coincident pulses for $25 \times 100 = 2500$ voxel combinations. The Compton electron energy deposition scored in a front-panel voxel (E_f) is correlated to the photoelectric energy deposition in a back-panel voxel (E_b), via the Compton equation, to produce the Compton angle of scatter and thus determine a conical angle of incidence. The form of the Compton equation that is used to obtain the conical angle of incidence is

$$\cos(\theta) = 1 - m_e \left[\frac{1}{E_b} - \frac{1}{E_f + E_b} \right], \tag{5.48}$$

where m_e is taken as 0.511 MeV.

Restrictions on E_f and E_b include: (1) $E_f < E_b$, and (2) $E_f > E_{ft}$ and $E_b < E_{bt}$, where E_{ft} and E_{bt} are threshold energies set by the user on the corresponding E8 and FU8 cards. The first of these is required to formulate a backward conical image (and helps ensure a Compton/photoelectric reaction occurred), while the latter is needed to reduce image clutter from voxel leakage (electron escape, bremsstrahlung, etc.). The FT8 COM processing algorithm is currently quite simple in that it takes the center-point of the front-panel voxel and that of the back-panel voxel to form a line which is then intersected with the image plane (at point **P**). Using the equation above, a radial distance from point **P** is determined and scores are made to various grid elements intersected by the circle about **P** (see Fig. 5.14). A simple algorithm is used, based on the size of the grid elements, to determine the number of sample points to score around the circle. A pulse of the source weight is scored in each image-plane grid element that overlaps a circular sample point.

An associated FIR radiography tally will be used to set up the image grid, with corresponding tally segment (FS card) and cosine (C card) bins.

The COM option is allowed only on F8 tallies and must be used with a corresponding dual-region PHL option. The tallies specified with the PHL option must involve multi-element lattices and use the special F-bin descriptor of "0". While the lattices in the two regions can differ in size and number of elements, tallies specified within a region must tally over the same lattice cell and elements (but can include contributions from different particle types). This feature fully supports repeated-structures geometries.

5.9.18.18.1 Example 1

The example shown in Listing 5.58 has a 2-MeV isotropic photon source located at $(-5, 3, 3)$, which is approximately 4 cm from two $1 \times 5 \times 5$ silicon panels, with the back panel 3 cm behind the front with the front panel centered at $(-1, 0, 0)$. This arrangement is shown in Fig. 5.15.

The silicon voxels are $2 \times 2 \times 2$ cm, making the panels $2 \times 10 \times 10$ cm overall in size. The image plane is coincident with the source location, so it is also approximately 4 cm from the front panel detector. The size of the image plane is 20 cm in each direction, with 10 grid elements along these s and t axes [§5.9.1.3.3]. The energy thresholds are set to 0.2 MeV on the E8 and FU8 cards. The TF8 card uses the 2nd user and energy bins for the TFC, and it is the values in these bins that are used in solving the Compton image equation, Eq. (5.48).

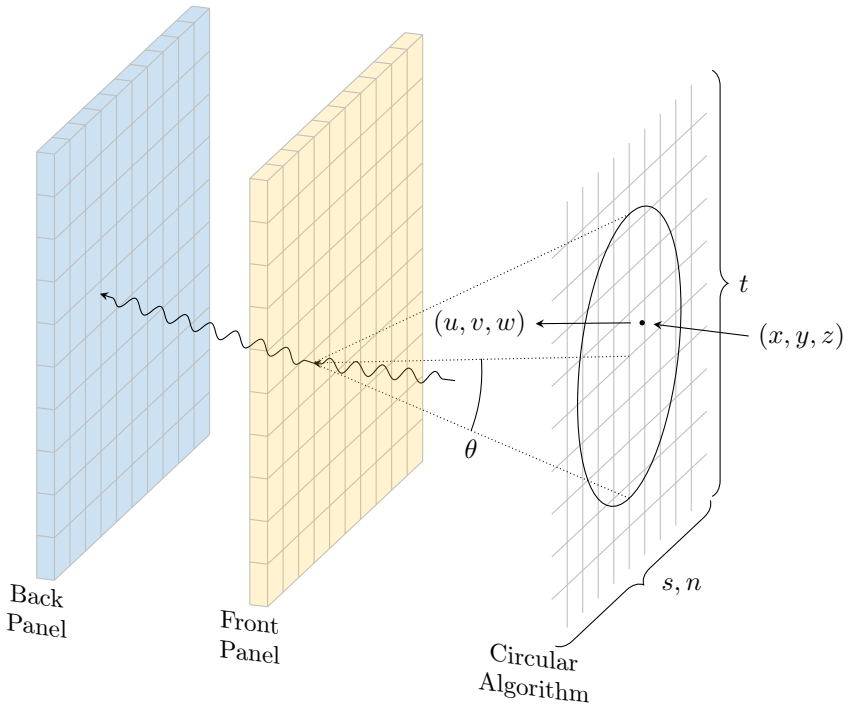


Figure 5.14: Diagram of a Compton imaging detector, along with a circular sample on the image plane.

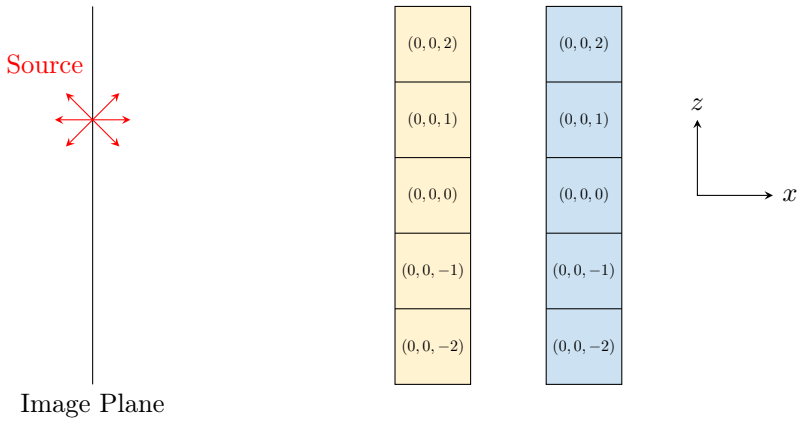


Figure 5.15: Geometry plot of Compton Imaging Tally Example, showing lattice indices for the front and back detector panels.

Listing 5.58: example_compton_img.mcnp.inp.txt

```
2-MeV photons into Si grid
1 1 -2.3 -1 lat=1 u=1       imp:p=1
            fill=0:0 -2:2 -2:2 1 24r
2 1 -2.3 -2 lat=1 u=2       imp:p=1
            fill=0:0 -2:2 -2:2 2 24r
3 0      -3    fill=1       imp:p=1
4 0      -4    fill=2       imp:p=1
5 0      -5 4 3             imp:p=1
6 0       5                 imp:p=0

1 rpp  -1 1  -1 1  -1 1
2 rpp   4 6  -1 1  -1 1
3 rpp  -1 1  -5 5  -5 5
4 rpp   4 6  -5 5  -5 5
5 so 100

mode p e
sdef par=p pos=-5 3 3 erg=2
m1  14028 1
phys:e  2j 1   $ Turn off bremsstrahlung
cut:p,e 2j 0 0 $ Analog capture
c
fir5:p -5 0 0  0  0 0 0  1 1 1
fs5 -10 9i 10
c5  -10 9i 10
c
f16:e (1<1[0:0 -2:2 -2:2]<3)
f26:e (2<2[0:0 -2:2 -2:2]<4)
c
f8:e  1
ft8    PHL 1 16 0    $ Region 1
           1 26 0    $ Region 2
           0
       COM 5 1
e8   0.2 100 NT
fu8  0.2 100 NT
tf8  j j 2 j j j 2 j
print
rand gen=2 seed=12345
nps 1e6
prdmp 2j 1
```

Figure 5.16 presents the corresponding Compton image, which can be produced using the commands shown in Listing 5.59.

Listing 5.59: example_compton_img.mcnp.comin.txt

```
tally 5 free sc contour noline
file
end
end
```

5.9.18.19 SPM *na*

The SPM special tally option generates collision exit energy-angle scatter probability matrices (SPM) averaged over particle interactions within each incident FU energy bin. The FU bins are automatically created from,

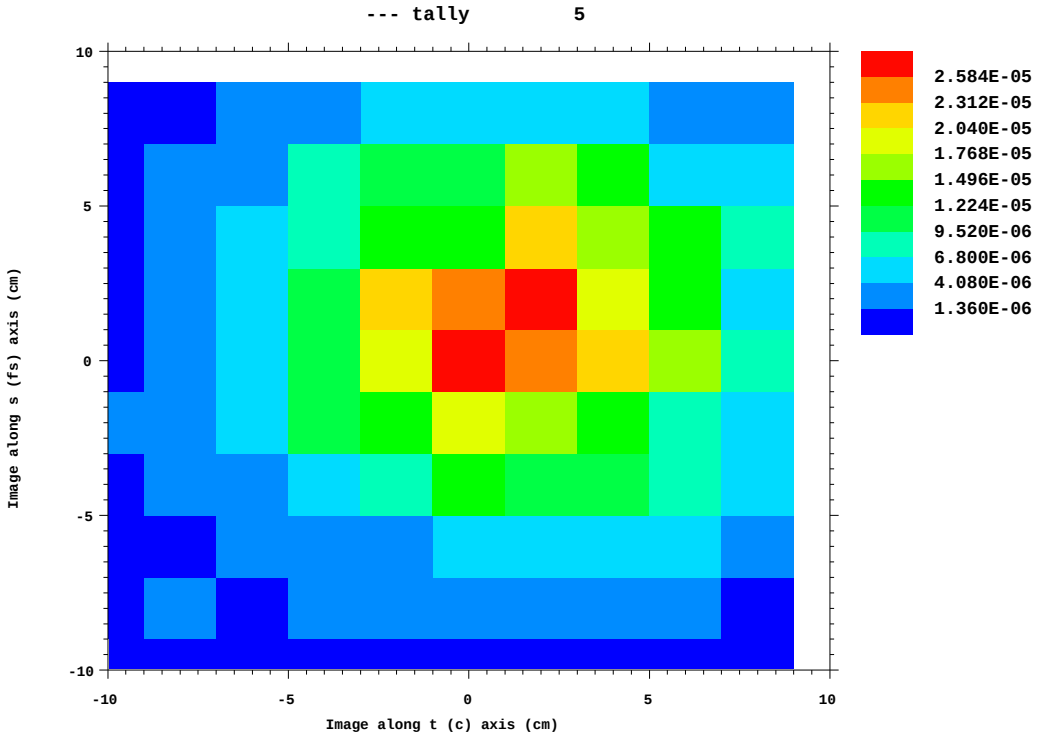


Figure 5.16: Compton image for Compton Imaging Tally Example, using a 20×20-cm image plane with 10×10 grid elements.

Table 5.22: Description of the Multiplier Bins for the MGC FT Option.

Bin #	Units	Values
1	$n/(cm^2 \cdot s)$	Flux (used as a divisor for the other bins)
2	sh/cm	Inverse velocity
3	barns	Total cross section
4	barns	Absorption cross section
5	barns	Fission cross section
6	barns	Total or prompt fission production cross section
7	barns	Delayed fission production cross section
8	barns	Fission heat production cross section
9	barns	Capture cross section (Absorption + Fission)
10	barns	Scatter cross section [Total − (Absorption + Fission)]

and are identical to, the bins specified on a related E card. This tally option can only be used with F4 tallies. The required *na* entry specifies the number of uniform cosine bins that will be generated on the related C card (from −1 to 1). This option requires analog transport, and the SPMs are tallied for each cell listed on the related F4 card. Fission neutrons are omitted from the SPM, however, subsequent collisions after fission are included. The SPMs are normalized to the number of collisions contributing to that exit energy-angle SPM, thus the sum of the E bins and C bins of each FU bin equals 1. Consider using the FQ card with "E C" to get exit cosine bins listed horizontally in the MCNP output file and exit energy bins listed vertically down the MCNP output file for each SPM. SPMs are independent of the source specification, although a poor choice in the source energy distribution may result in poor SPM convergence or no SPM results for some incident energies.

5.9.18.20 MGC *fg*

The MGC special tally option automatically generates ten FM multiplier bins that tally the flux and nine flux-weighted quantities useful to multigroup transport, using the energy bin structure specified on a related E card. This tally option can only be used with F4 tallies. The optional *fg* entry specifies microscopic cross-section units (barns) by default (i.e., when unspecified or set to "0") or macroscopic units (1/cm) when non-zero. For the tally results, the flux-weighted quantities are divided by the flux values to produce multigroup cross sections or other transport parameters. These FM bins are tallied for each cell listed on the related F4 card, and a full description of each bin is provided, as usual, in PRINT Table 30 of the MCNP output file, along with a condensed description in the tally output tables.

The ten multiplier bins are listed in Table 5.22.

5.9.18.21 FNS *nt*

The FNS special tally option generates fission neutron spectra (FNS) averaged over neutron induced fissions within each incident FU energy bin. The FU user bins are automatically created from, and are identical to, the bins specified on a related E card. This tally option can only be used with F4 tallies. The optional nt entry specifies the number of uniform half-life bins (from 100 to 10^{11} sh) that will be generated on the related T card for prompt (1st bin) and delayed (remaining *nt* bins) fission neutrons. A value of $nt = 6$ results in a prompt bin (100 sh) and the standard six ENDF delayed half-life bins (with midpoints of 0.179×10^8, 0.496×10^8, 2.230×10^8, 6.000×10^8, 21.840×10^8, and 54.510×10^8 sh). If *nt* is not specified, a T card must be used to list the prompt and delayed bin boundaries. This tally option requires analog transport, and the FNS are tallied for each cell listed on the related F4 card. For fixed-source problems, libraries and/or models can be specified to generate the delayed neutrons (see ACT card), while criticality problems only use libraries for delayed neutron production.

5.9.18.22 LCS *lo*

The LCS special tally option generates Legendre coefficients for exit energy-angle scatter probabilities over collisions within each incident FU energy bin. The FU user bins are automatically created from, and are identical to, the bins specified on a related E card. This tally option can only be used with F4 tallies. The required *lo* entry specifies the maximum Legendre order and thus the number of coefficients that will be generated on the associated C card. Included in this tally are the scatter bins' normalization factors (i.e., fraction of collisions contributing to the related coefficients). The C bins are labeled as 0.00 (normalization factor), 1.00 (1st coeff.), 2.00 (2nd coeff.), etc. These coefficients can be used in a Legendre polynomial expansion to mathematically estimate, subject to truncation error, the scatter distributions produced by the FT SPM option. This tally option requires analog transport, and the Legendre coefficients are tallied for each cell listed on the related F4 card. Fissions are not included as a scatter event.

5.9.19 TF: Tally Fluctuation

This card specifies the bin for which the tally fluctuation chart statistical information is calculated and the weight-window generator results are optimized. In addition, two separate tally bins can be specified to distinguish the "signal" vs. "noise" portions of a tally for ROC curve generation (see special tally treatment FT ROC in §5.9.18.15).

The TF card allows you to change the default bin for a given tally and specify for which tally bin the chart and all the statistical analysis output will be printed. The set of eight entries on a TF card correspond (in order) to the list of bin indices for the eight dimensions of the tally bins array. The order is fixed and not affected by an FQ card.

A helpful mnemonic suggested by Dr. Kris Ogren to remember the default bin ordering is Fred Died Under Some Mysterious Circumstances Editing Tallies—thanks Kris!

Data-card Form 1: TF*n if id iu is im ic ie it*
or
Data-card Form 2: TF*n if* 1 *id* 1 *iu* 1 *is* 1 *im* 1 *ic* 1 *ie* 1 *it* 1 *if* 2 *id* 2 *iu* 2 *is* 2 *im* 2 *ic* 2 *ie* 2 *it* 2

n	Tally number. Restriction: $n \leq 99999999$
if	The bin number of the cell, surface, or detector bin (F-bin) on F card. (DEFAULT:$if = 1$, first bin)
id	The bin number of the total, flagged, or un-collided flux (D-bin). (DEFAULT: $id = 1$, total flux)
iu	The bin number of the user bin (U-bin). (DEFAULT: iu = last bin) (2)
is	The bin number of the segment bin (S-bin). (DEFAULT: is = last bin)
im	The bin number of the multiplier bin on FM card (M-bin). DEFAULT: $im = 1$, first bin)
ic	The bin number of the cosine bin (C-bin). (DEFAULT: ic = last bin)
ie	The bin number of the energy bin (E-bin). (DEFAULT: ie = last bin)
it	The bin number of the time bin (T-bin). (DEFAULT: it = last bin)

Use: Whenever a particular tally bin is more important than the default bin. Particularly useful in conjunction with the weight-window generator. Also used to specify signal versus noise components of a ROC curve.

Details:

1. The second input format is used only with the FT*n* ROC tally option. In this case, the first 8 entries represent bins associated with the signal component while the second 8 entries identify the noise component [§5.9.18.15]. To support ROC curve generation, the entry format allows multiple bins to be specified for each entry: a single bin (e.g., 10), a range of bins (e.g., 10–12), a list of bins (e.g., 10, 11, 12), or a combination of these formats (e.g., 10–12,13,14). However, only the first bin listed in each entry is used for generating the TFC output, weight-window generation, and statistical analysis.
2. You may find the J feature useful to jump over last entries. Remember that totals are calculated for energy, time, and user bins (unless inhibited by using NT), so that last for eight energy bins is 9. If one segmenting surface divides a cell or surface into two segments, last in that case is 2, unless T is used on the FS card, in which case last is 3. If there are no user bins or cosine bins, for example, last is 1 for each; last is never less than 1.

5.9.19.1 The Tally Fluctuation Chart

At the end of the output, one chart for each tally is printed to give an indication of tally fluctuations; that is, how well the tally has converged. The tally mean, relative error, variance of the variance, Pareto slope [§2.6.8.7], and figure of merit (FOM = $R^{-2}T^{-1}$), where R is the relative error printed with the tally and T is computer time in minutes) are printed as functions of the number of histories run. The FOM should be roughly constant. The TF*n* card determines for which bin in tally *n* the fluctuations are printed. It also determines which tally bin is optimized by the weight-window generator (WWE or WWT and WWG or WWGT cards).

The mean printed in a chart will correspond to some number in the regular tally print. If you have more than one surface listed on an F2 card, for example, the default chart will be for the first surface only; charts can be obtained for all surfaces by having a separate tally for each surface.

5.9.19.2 Example 1

Suppose an F2 tally has four surface entries, is segmented into two segments (the segment plus everything else) by one segmenting surface, and has eight energy bins. By default one chart will be produced for the first surface listed, for the part outside the segment, and totaled over energy. If we wish a chart for the fifth energy bin of the third surface in the first segment, we would use

```
TF2   3  2J  1  2J  5
```

5.9.19.3 Example 2

```
TF2   3  2J  1  2J  5  J
```

In this example, statistics will be calculated based on the 3rd surface, 1st segment, and 5th energy bin provided in tally 2. Without this card, statistics will be performed on the 1st surface, 1st segment, and the total of all energy bins.

5.9.19.4 Example 3

```
TF1    2 j 10-12,13,88 1-2 j 7,8,9 1-99 2   j 1 j j 2 j j j
```

Note that spaces are not allowed within a comma-delimited list of bins and/or bin ranges. Spaces continue to be used to delimit the eight bin-type entries. The first eight entries specify the bins that constitute the signal component of the tally, while the second eight entries specify the bins that constitute the noise component of the tally.

5.9.20 NOTRN: Direct-only Neutral-particle Point Detector Contributions

Data-card Form: `NOTRN`

Default: None.

Use: This option works with point-detector tallies as well as pinhole or transmitted image tallies. If the `NOTRN` card appears in the MCNP input file, no transport of the neutral particle source particles takes place, and only the direct neutral particle source contributions are made to the detectors and the detector grid. This is especially useful for checking the problem setup or doing a fast calculation to generate the direct source image. A `NOTRN` card is not allowed in a restarted calculation.

5.10 Tally Perturbations and Reactivity Sensitivities

MCNP6 offers two flavors of perturbation theory, one based on the differential operator (`PERT` card) and two others based on adjoint weighting (`KPERT` and `KSEN` cards). Both methods offer advantages and disadvantages. The differential operator technique is based on a Taylor series expansion and works very well for generalized responses in fixed-source problems. In eigenvalue problems, however, the differential operator methodology may produce inaccurate results because the MCNP6 implementation does not account for the perturbation of the fission source distribution. The adjoint-based methodology implicitly captures the perturbation in the fission source; however, it is only capable of finding the change in reactivity resulting from perturbations in cross sections and not other responses.

Should a user desire estimates of changes in reactivity for reactor physics applications or sensitivity coefficients to the k-eigenvalue, then the adjoint-based methodology is appropriate. An important limitation of the adjoint-based methods as implemented in MCNP6 is that they do not consider perturbations that may arise from scattering laws or from fission emission spectra; this limitation has been shown to lead to spurious results. For perturbations where the dominant effects are from absorption or the scattering is mostly isotopic, the results tend to agree well with those from direct cross-section substitutions and from the adjoint-methodology code TSUNAMI-3D [318], which employs multigroup cross-section data rather than continuous-energy data.

5.10.1 PERT: Tally Perturbations via Differential Operator

This card allows perturbations in cell material density, composition, or reaction cross-section data. The perturbation analysis uses the first and second order differential operator technique. Perturbation estimates

are made without actually changing the input material specifications. Multiple perturbations can be applied in the same run, each specified by a separate PERT card.

There is no limit to the number of perturbations because dynamic memory is used for perturbation storage. The entire tally output is repeated for each perturbation, giving the estimated difference in the tally, or this difference can be added to the unperturbed tally (see the METHOD keyword). For this reason, the number of tallies and perturbations should be kept to a minimum. However, an entire parameter study can be done with just two PERT cards [319]. A track length estimate of perturbations to k_{eff} is automatically estimated and printed for KCODE problems.

Data-card Form: PERTn:$\mathscr{P}$ *KEYWORD=value(s)...*

n	Unique, user-selected, arbitrary perturbation number. Restriction: $0 < n \leq 99999999$
$\mathscr{P}$	Particle designator. Only three options allowed: neutron (N); photon (P); or combined neutron-photon (N,P). Not available for other particles.
CELL $= c_1\, c_2 \ldots c_K$	Comma or space delimited list of cells, $c_1 \ldots c_K$, to which to apply the perturbation. Required.
MAT $= m$	Single material number, m (corresponding to an Mm card), with which to fill all cells listed in CELL keyword. Use MAT only if the perturbation changes the material from one cell material to another. Use with caution especially if more than one nuclide in the material is changed. New nuclides cannot be added in the new material card. Must have a corresponding M card (1).
RHO $= r$	Single value of perturbed density of cells listed after the CELL keyword (2). If
RHO > 0	the perturbed density is given in units of atoms/b-cm.
RHO < 0	the perturbed density is given in units of g/cm^3.
METHOD $= j$	Controls tally printing and specifies the number of terms to include in the perturbation estimate (3). If
METHOD $= +1$	perform 1st and 2nd order perturbation calculation and print the difference in the unperturbed tally. (DEFAULT)
METHOD $= -1$	perform 1st and 2nd order perturbation calculation and print the perturbed tally.
METHOD $= +2$	perform 1st order perturbation calculation only and print the difference in the unperturbed tally.
METHOD $= -2$	perform 1st order perturbation calculation only and print the perturbed tally.
METHOD $= +3$	perform 2nd order perturbation calculation only and print the difference in the unperturbed tally.
METHOD $= -3$	perform 2nd order perturbation calculation only and print the perturbed tally.
ERG $= e_{\text{LB}}\, e_{\text{UB}}$	Two entries, e_{LB} and e_{UB}, that provide the lower and upper bounds of the energy range to which the perturbations are to be applied (4). (DEFAULT: all energies)

RXN $= r_1\ r_2 \ldots$	ENDF/B reaction number(s) that identify one or more specific reaction cross sections to perturb (5). (DEFAULT: RXN $= 1$ for neutrons and multigroup, RXN $= -5$ for photons.) Restriction: RXN reaction numbers must be identical to FM card reaction numbers.

Default: METHOD=+1; ERG=all energies; RXN=1 for neutrons and multigroup, RXN=-5 for photons.

Use: Optional. The CELL keyword, which identifies one or more perturbed problem cells, is required. Additionally, either the MAT or RHO keyword must be specified.

Details:

1. Composition changes can only be made through the use of the MAT keyword. If the RHO keyword is omitted, the MAT keyword is required. Certain composition changes (discussed in §5.10.1.1) are prohibited.
2. If the MAT keyword is absent, the RHO keyword is required.
3. The ability to produce first- and second-order Taylor series expansion terms separately enables the user to determine the significance of including the second-order estimator for subsequent runs. If the second-order results are a significant fraction (20%–30%) of the total, then higher order (or other) terms are necessary to predict accurately the change in the unperturbed tally. In such cases, the magnitude of the perturbation should be reduced to satisfy this condition. Typically, this technique is accurate to within a few percent for up to 30% changes in the unperturbed tally. It is strongly recommended that the magnitude of the second order term be determined before the user continues with this capability. Classical first-order sensitivity analysis requires only the first-order term, METHOD $= +2$; in this case, the relative magnitude of the second-order term is irrelevant.
4. The ERG keyword is usually used with the RXN keyword to perturb a specific cross section over a particular energy range.
5. The RXN keyword allows the user to perturb a single reaction cross section of a single nuclide in a material, all reaction types of a single nuclide, a single reaction for all nuclides in a material, and a set of cross sections for all nuclides in a material. Relevant non-standard special R numbers, listed in Table 5.19, can be used. Those that are irrelevant and therefore cannot be used are -4, -5, -7, and -8 for neutrons; -6 for photons; and -3, -4, -6, and -7 for multigroup problems. If these irrelevant R numbers are used, the following fatal error will be printed: "`fatal error. reaction # illegal in perturbation #.`"

   RXN reaction numbers must be consistent with FM card reaction numbers if the perturbation affects the tally cross section. The specification RXN $= -6$ is most efficient for fission, although MT=18, MT=19, or MT=-2 (multigroup) also work for k_{eff} and F7 tallies.

5.10.1.1 PERT Card Limitations/Cautions

1. The perturbation method is limited to the 1st and 2nd order terms of a Taylor series expansion. Examine the 1st and 2nd order terms separately for large ($> 30\%$) perturbations to determine the significance of the 2nd order terms. If 2nd order terms are a significant fraction (20%–30%) of the total perturbation, inaccurate tallies can result (3). A warning message is generated.
2. Nuclide fraction changes (MAT keyword) are assumed to be independent and, consequently, differential cross terms are ignored. Stated another way, when multiple isotopes are perturbed at once, the perturbation estimate is the sum of the independent nuclide perturbations and does not include the 2nd-order differential term. Therefore, it is very important to change only one isotope density in each PERT card, or to change all isotope densities the same relative amount [319].

3. FM tallies in perturbed cells can be wrong. Surface tallies and tallies in perturbed cells are safe. A warning message is generated.
4. Detector (F5) and pulse-height tallies (F8) are not compatible with the PERT card. (i.e., give zero perturbation).
5. DXTRAN (DXT) is not compatible with the PERT card. A fatal error message is generated.
6. You cannot un-void a region. That is, if you take a region originally specified as void and put in a material in that region with the perturbation technique, a fatal error message is generated. However, you can specify a region as containing a material and use the PERT card to make it void by setting RHO=0.
7. You cannot introduce a new nuclide into a material composition. A fatal error message is generated. However, you can set up the problem with a mixture of all nuclides of interest and use PERT cards to remove one or more nuclides.
8. Although there is no limit to the number of perturbations, each perturbation may increase running time by 10%–20%, though this value depends on the complexity of the problem and the PERT card(s).
9. Some perturbations (those with small changes) converge slowly.
10. The track length estimate of k_{eff} in criticality calculations assumes the fundamental eigenvector (fission distribution) is unchanged in the perturbed configuration. This approximation can lead to serious errors [197]. For the effect of a perturbation on k_{eff}, use the KPERT card.
11. Use caution when selecting the multiplicative constant and reaction number on FM cards used with F4 tallies in perturbation problems. The track length correction term $R_{1j'}$ is made only if the multiplicative constant on the FM card is negative (indicating macroscopic cross sections with multiplication by the atom density of the cell). If the multiplicative constant on the FM card is positive, it is assumed that any FM card cross sections are independent of the perturbed cross sections. If there is a reaction (RXN) specified on the PERT card, the track length correction term R_{1j} is set only if the exact same reaction is specified on the FM card. For example, an entry of RXN=2 (elastic cross section) on the PERT card is not equivalent to the special elastic reaction -3 on the FM card. The user should either enter 2 as the reaction of the FM card and RXN=2 on the PERT card or -3 on FM and -3 on PERT.
12. Limited to N and/or P problems.

5.10.1.2 Example 1

```
PERT1:N,P    CELL=1   RHO=0.03
```

This perturbation specifies a density change to 0.03 atoms/b-cm in cell 1. This change is applied to both neutron and photon interactions.

5.10.1.3 Example 2

```
PERT3:N,P   CELL=1 10i 12   RHO=0   METHOD=-1
```

This perturbation makes cells 1 through 12 void for both neutrons and photons. The estimated changes will be added to the unperturbed tallies.

5.10.1.4 Example 3

```
60  13  -2.34   105 -106  -74   73  $ mat 13 at 2.34 g/cm3
.
.
.
M13    1001 -0.2  8016 -0.2  13027 -0.2  26000 -0.2  29000 -0.2
M15    1001 -0.2  8016 -0.2  13027 -0.2  26000 -0.2  29000 -0.4
PERT1:P   CELL=60  MAT=15  RHO=-2.808  RXN=51 9i 61,91  ERG=1,20
        METHOD=2
PERT2:P   CELL=60          RHO=-4.68   RXN=2   METHOD=2
```

This example illustrates first-order sensitivity analysis. The first PERT card generates the first-order Taylor series terms Δc_1 for changes in tallies caused by a $p = 100\%$ increase in the copper cross section (ENDF/B reaction types 51–61 and 91) above 1 MeV. To effect a $p\%$ change for a specific isotope, set up a perturbed material mimicking the original material, except multiply the composition fraction of the perturbed isotope by $1 + p$ (-0.2 to -0.4). The density of the perturbed material is the density of the original material (2.34 g/cm^3) multiplied by the ratio of the sum of the weight fractions of the perturbed material (1.2) to the sum of the weight fractions of the unperturbed material (1.0), or RHO $= (-2.34$ g/cm^3 $\times 1.2/1.0) = -2.808$ g/cm^3. This change must be made to RHO to maintain the other nuclides in their original amounts. Otherwise, after the MCNP code normalizes the M15 card and multiplies the constituent weight fractions by the unperturbed material density, the density of all of the constituents would be perturbed, which is not the intent. When the MCNP code normalizes the M15 card and multiplies the constituent weight fractions by the correctly modified material density, the density of the unperturbed isotopes will be unchanged, but the density of the perturbed isotope will be changed by a factor $1 + p$, as intended.

The first-order sensitivity of response c is calculated in post processing using $S = \Delta c_1/(c_0)p$, and p is arbitrary [319].

The second PERT card (PERT2:p) gives the first-order Taylor series terms Δc_1 for changes in tallies caused by a 100% increase in the elastic (RXN=2) cross section of material 13. RHO$= -2.34$ g/cm^3 $\times 2 = -4.68$ g/cm^3.

5.10.1.5 Example 4

```
M4 6000.60C 0.5 6000.50C 0.5
M6 6000.60C 1
M8 6000.50C 1
PERT1:N CELL=3 MAT=6 METHOD=-1
PERT2:N CELL=3 MAT=8 METHOD=-1
```

The perturbation capability can be used to determine the difference between one cross-section evaluation and another. The difference between these perturbation tallies will give an estimate of the effect of using different cross-section evaluations.

5.10.1.6 Example 5

```
1 1 0.05 -1 2 -3 $ mat 1 at 0.05 x 10 atoms/cm
.
.
.
M1 1001 0.1 8016 0.2 92235 0.7
M9 1001 0.1 8016 0.22 92235 0.7
F14:N 1
FM14 -1 1 -6 -7 $ keff estimator for cell 1
PERT1:N CELL=1 MAT=9 RHO=0.051 METHOD=1
PERT2:N CELL=1 MAT=9 RHO=0.051 METHOD=-1
```

These perturbations involve a 10% increase in the oxygen atom fraction of material 1 (`RHO` $= 0.05 \times (1.02/1.0) = 0.051$). The effect of this perturbation on tally 14, which is a track-length estimate of k_{eff}, will be provided as a difference (`PERT`1) as well as with this change added to the unperturbed estimate of k_{eff} (`PERT`2). Note: if the `RHO` keyword is omitted from the `PERT` cards, the ^{235}U composition will be perturbed, which can produce invalid results [§5.10.1.1].

5.10.1.7 Example 6

The MCNP6 perturbation capability assumes that changes in the relative abundances of the nuclides in a material are independent and neglects the cross-differential terms in the second-order perturbation term when changing two or more cross sections at once. In the case illustrated below there will be a large false second-order perturbation term.

```
M1 6000.50c 0.5 6012.50c 0.5
M2 6000.50c 0.9 6012.50c 0.1
PERT1:N CELL 1 MAT 2
```

The perturbation should be zero because 6000.50c is exactly the same as 6012.50c, making materials `M`1 and `M`2 identical. In fact, the first-order term will be zero (`METHOD=2`, correct) but the second-order term will be wrong because of the differential cross term.

5.10.1.8 Example 7

There is no problem if all the nuclides have the same density change (`RHO` option but no `MAT` option). There is also no cross term problem if only one nuclide has a density change, for example:

```
cell 1 material 1 density rho=3.0
.
.
.
M1 1001 2 8016 1
M2 1001 2 8016 2
PERT1:N CELL 1 MAT 2 RHO=4.0
```

The cell density times the normalized atom fraction of 1001 is unchanged ($3 \times 2/3 = 4 \times 2/4$) and only the density of 8016 is changed (from $3 \times 1/3$ to $4 \times 2/4$). However, there will be a second-order cross-differential

term that is neglected when the cell density times nuclide fraction changes for more than one nuclide in a perturbed material. Therefore, if the MAT keyword is used for a perturbation, the first- and second-order terms should be examined. If the second-order perturbation term is small relative to the first-order term (METHOD=3 and METHOD=2), then generally the differential cross term is small and the perturbed tally can be accepted with confidence.

5.10.2 KPERT: Reactivity Perturbations via Adjoint Weighting

The adjoint-weighted perturbation methodology invoked by the KPERT card was designed to investigate changes in k_{eff} as a result of material substitution. While this method, in theory, allows for more general perturbations, it introduces an approximation in the handling of scattering laws that can lead to large and unacceptable deviations in scattering sensitivities. Additionally, the user interface was designed with material substitution with mind; using it for sensitivity coefficient calculations may be cumbersome for some users. For sensitivity coefficient calculations, see the KSEN card. Multiple KPERT cards are permitted in a single input file.

Data-card Form: KPERT*n KEYWORD=value(s)...*

n	Unique, user-selected, arbitrary perturbation number. *n* has the same limits as regular tallies. See Table 4.2.
CELL = $c_1\, c_2 \ldots c_K$	Comma or space delimited list of cells, $c_1 \ldots c_K$, to which to apply the perturbation. Required.
MAT = $m_1\, m_2 \ldots m_K$	List of materials that are to be substituted in each of the perturbed cells listed in the CELL keyword. Each cell must be associated with exactly one material number and each unique material identifier number must have an associated M card (1).
RHO = $r_1\, r_2 \ldots r_K$	List of densities corresponding to each of the perturbed cells listed in the CELL keyword. Each cell specified on the CELL keyword must be associated with exactly one density value specified on the RHO keyword (2). If
$r_k > 0$	the perturbed density is given in units of atoms/b-cm^2.
$r_k < 0$	the perturbed density is given in units of g/cm^3.
ISO = $z_1\, z_2 \ldots z_K$	List of target identifiers [§1.2.2] or table identifiers [§1.2.3] that the perturbation impacts. All formats supported. The list applies to all cells in the CELL list (3). (DEFAULT: all isotopes assumed affected)
RXN = $rx_1\, rx_2 \ldots rx_K$	List of MT or special reaction numbers that the perturbation impacts. The list applies to all cells in the CELL list (4). Table 5.23 provides a list of acceptable entries. (DEFAULT: all reactions assumed affected)
ERG = $e_1\, e_2 \ldots e_K$	List of energies (MeV), in ascending order, over which to apply the perturbation (5). (DEFAULT: all energies)
LINEAR=*value*	Provides the ability to force an unperturbed fission source, yielding a linear equation to estimate the change in reactivity that arises from a change in cross sections. Many applications, such as the calculation of sensitivity coefficients demand the use of linear-perturbation theory in which the denominator is unperturbed. If
LINEAR = NO	do not use the perturbed fission source in the denominator.

LINEAR = YES	use the perturbed fission source in the denominator. (DEFAULT)

Default: ISO=all isotopes; RXN=all reactions; ERG=all energies; LINEAR=NO

Use: Optional. The CELL keyword, which identifies one or more perturbed problem cells, is required. Additionally, either the MAT or RHO keyword must be specified.

Details:

1. If the RHO keyword is absent, the MAT keyword is required. Use the MAT keyword, for example, to test the effect of changing the enrichment of a particular set of cells.
2. If the MAT keyword is absent, the RHO keyword is required. This keyword allows the user to perform density perturbations. RHO may be used in addition to the MAT keyword to perturb both the material and the density of the cells specified in the CELL keyword list.
3. The ISO keyword is useful for testing the effect of individual nuclides.
4. The RXN keyword is useful for testing the effect of individual reactions.
5. The ERG keyword is similar to energy binning with tallies, except that there is no implied lower bound of 0 MeV.

5.10.2.1 Example 1

```
KPERT5    CELL=1 4  MAT=2 2  RHO=-19.1 -19.1
```

This perturbation takes whatever materials are in cells 1 and 4 and makes them both material 2 with a mass density of 19.1 g/cm^3.

5.10.2.2 Example 2

```
KPERT98    CELL=10  RHO=-18.6  RXN=18
```

This perturbation looks at the effect on the fission reaction (MT=18) when the mass density of the material in cell 10 is changed to 18.6 g/cm^3.

5.10.2.3 Example 3

```
KPERT1    CELL=22 26  MAT=92 92  ISO=U-238.70c RXN=51 39i 91
          ERG=0 2 5 20  LINEAR=YES
```

This perturbation judges the impact of ^{238}U inelastic scattering in cells 22 and 26 by a change to material 92. The perturbation is further broken down by energy, with regions of less than 2 MeV, between 2 and 5 MeV, and between 5 and 20 MeV. The perturbation is also linear.

5.10.3 KSEN: k_{eff} Sensitivity Coefficients via Adjoint Weighting

The KSEN card [320, 321] provides the ability to compute sensitivity coefficients of the effective multiplication k (i.e., k_{eff}) for nuclear data. These types of calculations are useful for code validation and the development of benchmark suites applicable to specific sets of applications, for the design of critical (integral) experiments, and for uncertainty quantification. This computation is done in a KCODE calculation using the KSEN card; fixed-source problems are not appropriate for KSEN. Multiple KSEN cards are permitted in a single input file.

MCNP6 has the ability to subdivide sensitivities by spatial zone. These can be done either as a collection of cells or materials. The keywords on the KSEN card to do this are CELL = *c1* (*c2 c3*) ... and MAT = *m1* (*m2 m3*) ... Each entry defines a spatial zone, and like with tally specifications, cells or materials may be grouped by parentheses. Duplicate cells or materials are allowed. A KSEN card may not have both the CELL and MAT keywords; doing this both ways requires multiple instances of KSEN.

The methods employed are based upon linear-perturbation theory using adjoint weighting, the same as those used by TSUNAMI-3D for this purpose [318]. The adjoint weighting is performed in a single forward calculation using the Iterated Fission Probability method. The capability is specifically designed for use in continuous-energy calculations, and while it is possible to use this option in multigroup calculations, MCNP6 does not compute the effect of the cross-section self-shielding on the sensitivity coefficients.

Data-card Form: KSEN*n* sen KEYWORD = *value(s)* ...

n	Unique, user-selected, arbitrary perturbation number. *n* has the same limits as regular tallies. See Table 4.2. Values greater than 999999 may result in asterisks in the outp file.
sen	Type of sensitivity. If sen = XS — a cross-section or nuclear data sensitivity is specified. This is the only kind of sensitivity supported at this time.
ISO = *z1 z2 ... zK*	List of target identifiers [§1.2.2] or table identifiers [§1.2.3] for which sensitivities are desired. All formats supported. (DEFAULT: all data tables in the problem)
RXN = *rx1 rx2 ... rxK*	List of reaction MT numbers or special reaction numbers. Table 5.23 provides a list of acceptable entries. [DEFAULT: total cross section without $S(\alpha, \beta)$]
MT = *rx1 rx2 ... rxK*	Same behavior as RXN.
ERG = *e1 e2 ... eK*	List of energy bin boundaries, in ascending order, over which to provide the sensitivities. For cross sections and fission ν, the energies are taken to be those entering the collision (incident energy). For secondary distributions of fission χ and scattering laws, the energies are taken to be energies exiting the collision. If used, a minimum of two entries are required to establish at least one lower and upper boundary (①). (DEFAULT: all energies)
EIN = *e1 e2 ... eK*	Specifies a range of incident energy bins (①). Only used for fission-χ (-1018) or scattering-law (-1002 or -1004) sensitivities. (DEFAULT: all energies)
LEGENDRE	The LEGENDRE keyword is followed by a single integer (> 0) stating the order of Legendre moments to calculate sensitivities for (e.g., "LEGENDRE = 3" would give the k_{eff} sensitivity to the P_1, P_2, and P_3 Legendre scattering moments) any scattering law sensitivity. If present calculates the scattering law sensitivity to Legendre moments instead of as a function of cosine

	binning. Note that to do this, the MCNP code needs a background cosine grid that may be provided by the user with the COS keyword. If this is not provided, a default cosine grid of 200 equally spaced cosine bins from −1 to 1 is used.
COS	Specifies a range of direction-change cosines for the scattering events. Only used for scattering law (−1002 or −1004) sensitivities. (DEFAULT: all angles)
CONSTRAIN	Only used for fission-χ (−1018) or scattering-law (−1002 or −1004) sensitivities. If
	CONSTRAIN = NO — do not renormalize the energy (or cosine) sensitivity distribution.
	CONSTRAIN = YES — renormalize the energy (or cosine) sensitivity distribution (2). (DEFAULT)
CELL	List of cell numbers of the problem for spatial zoning. Each entry defines a spatial zone and multiple cells may be grouped into a single spatial zone with parentheses. Duplicate cells are allowed.
MAT	Like the CELL keyword except material numbers are used as opposed to cell numbers. Zones are defined to encompass all cells containing that material.

Default: ISO=all isotopes in the problem; RXN=total cross section without $S(\alpha,\beta)$; ERG=all energies; EIN=all energies, COS=all angles; CONSTRAIN = YES

Use: Optional. If the KSEN card is used, the KOPTS card is recommended.

Details:

1. Unlike tallies, there is no implied zero lower-energy-bin boundary.
2. Increasing a distribution in one region of energy (or cosine) space needs to be offset by decreases elsewhere to preserve the condition that the distribution be normalized to a constant value, typically one. For most applications, users should use the default, i.e., renormalize the sensitivities. Full normalization [322] is applied.
3. For cross sections and fission ν, the energies listed on ERG are taken to be those entering the collision, whereas for secondary distributions of fission χ and scattering laws they are taken to be energies exiting the collision.

5.10.3.1 Example 1

```
KSEN3  XS
```

Default behavior. Gives the total cross-section sensitivities (integrated over all energies) to all isotopes and $S(\alpha,\beta)$ laws in the problem.

Table 5.23: Allowed Reaction Numbers for KSEN with Continuous-energy Physics

Nuclear Data	MT Number	Special Reaction Number
Total	1	—
Total and $S(\alpha,\beta)$	—	−1
Capture	—	−2
Elastic	2	—
Total Inelastic	4	—
Elastic and $S(\alpha,\beta)$	—	−3
Total Fission	18	−6
First-Chance Fission	19	—
Second-Chance Fission	20	—
Third-Chance Fission	21	—
Fourth-Chance Fission	38	—
Total Fission ν	452	−7
Prompt Fission ν	456	—
Delayed Fission ν	455	—
(n,2nd)	11	—
(n,2n)	16	—
(n,3n)	17	—
(n, nα)	22	—
(n, n3α)	23	—
(n, 2nα)	24	—
(n, np)	28	—
(n, n2α)	29	—
(n, 2n2α)	30	—
(n,nd)	32	—
(n,nt)	33	—
(n, n^{3}He)	34	—
(n,nd2α)	35	—
(n,nt2α)	36	—
(n,4n)	37	—
(n, 2np)	41	—
(n, 3np)	42	—
(n, n2p)	44	—
(n, npα)	45	—
(n,γ)	102	—
(n,p)	103	—
(n,d)	104	—
(n,t)	105	—
(n,^{3}He)	106	—
(n,α)	107	—
Inelastic Levels (1–40)	51, 52, . . . , 90	—
Inelastic Continuum	91	—
Total Fission χ	—	−1018
Prompt Fission χ	—	−1456
Delayed Fission χ	—	−1455
Total Scatter Law	—	−1001
Elastic Scatter Law	—	−1002
Inelastic Scatter Law	—	−1004

5.10.3.2 Example 2

```
KSEN14  XS  ISO=U-235.70c U-238.70c  MT=-1 2 4 -6
```

Gives total, elastic, inelastic, and fission cross-section sensitivities for ^{235}U and ^{238}U.

5.10.3.3 Example 3

```
KSEN8  XS  ISO=H-1.70c lwtr.10t  MT=2 4  ERG=0.0 0.625e-6 0.1 20
```

Gives ^{1}H elastic scattering and the light-water $S(\alpha, \beta)$ inelastic scattering kernel sensitivities as a function of energy with bins between 0 and 0.625 eV, 0.625 eV to 100 keV, and 100 keV to 20 MeV.

5.10.3.4 Example 4

```
KSEN99  XS   ISO=Pu-239.70c  MT=-1018  ERG=0 0.1 1.0 2.0 5.0 10.0 20.0
             EIN=0 2.5 8.0 20.0  CONSTRAIN=NO
```

Gives ^{239}Pu fission-χ sensitivities as a function of outgoing and incident energy. The incident energy bins are 0 to 2.5 MeV, 2.5 to 8 MeV, and 8 to 20 MeV. For each of these, a fission-χ sensitivity is given for the six energy bins specified by the `ERG` keyword. The sensitivity is also not renormalized, which is normally discouraged.

5.10.3.5 Example 5

```
KSEN8016  XS  ISO=O-16.70c  MT=-1002  ERG=0 19i 20  COS=-1 0 1
```

Gives ^{16}O elastic scattering law sensitivities for 1-MeV (outgoing) energy bins from 0 to 20 MeV. Each outgoing energy bin is subdivided into two cosine bins for forward and back scattering. The sensitivity includes neutrons scattering at all possible incident energies.

5.10.3.6 Example 6

```
KSEN101  XS  CELL=10 20 (10 20)  ERG=SCALE-238
```

Gives total cross section sensitivities for all isotopes in the problem with an energy binning defined by SCALE's 238-group library. Three energy-resolved sensitivity profiles are given: one for cell 10, another for cell 20, and a third for both (the sum of the sensitivities for cells 10 and 20).

5.10.3.7 Example 7

```
KSEN101  XS  RXN=-1002 -1004  LEGENDRE=5  ISO=Fe-56.70c  MAT=20
```

Gives the sensitivities for the first five Legendre moments of elastic and inelastic scattering of ^{56}Fe, but only for cells with material 20. The default cosine grid of 200 equally spaced intervals from -1 to 1 is used for computing the Legendre moment sensitivities because the COS keyword is not specified.

5.10.3.8 Additional Discussion

Other options may be controlled by use of the KOPTS card, which contains various options for KCODE calculations. The two options are BLOCKSIZE, which controls the number of cycles in every outer iteration, and KSENTAL, which controls output printing of a results file for sensitivity profiles. The format for these is as follows: KOPTS BLOCKSIZE = *NCY* KSENTAL = *FILEOPT*.

The *NCY* argument denotes the number of cycles. A greater number leads to better accuracy of the answer, but the results will be less statistically resolved. The default is 10 cycles, which has been shown to be conservative for almost all cases and still preserves a reasonable about of statistical precision. For small, leakage dominated systems, this can often be reduced to 5.

The *FILEOPT* argument gives a file format for printing the sensitivity profiles. The default is to print no file. In MCNP6, two file formats are available: MCTAL and TSUNAMI-B. The MCTAL format has the MCNP code print the sensitivity profiles in a special file called **ksental** which is similar to a **mctal** file for tallies, and can be plotted by MCPLOT. The TSUNAMI-B format is defined in [Table 6.5.A.2 of 323]. The concepts used by the MCNP and SCALE codes are not necessarily compatible depending on the sensitivity profile options in either code, so the TSUNAMI-B format may not be able to capture everything that the MCNP code can compute. A description of the formats are given below.

An example illustrating these concepts:

```
KOPTS   BLOCKSIZE = 5   KSENTAL = TSUNAMI-B
```

By default the MCNP code prints the sensitivity profiles to the output file. These are located below "the box" with the k results with the heading "nuclear data sensitivity profiles". The ordering of results changes depending upon the requested information. Regardless, the sensitivities are presented as the sensitivity result (integrated over an energy bin) and its associated relative uncertainty. Note that because sensitivities may either be positive or negative, those near zero may have a very large (greater than one) relative uncertainty, but the absolute uncertainty may be quite small.

If no energy bins are requested, then the sensitivities will be presented as:

```
ZAID    REACTION    SENS    REL UNC
```

The ZAID is the first 12 characters of the full table identifier. If energy bins are requested, then the sensitivities will be presented as a function of energy for each isotope and reaction:

```
ELOW    EHIGH     SENS    REL UNC
```

Here *ELOW* and *EHIGH* denote the energy bin boundaries. These energy-resolved results may be plotted for visualization in various plotting programs (Gnuplot, Microsoft Excel, etc.). When doing so, it is usually recommended to plot the profiles per unit lethargy (divide each sensitivity by the logarithm of the ratio of *EHIGH* to *ELOW*) on a semi-log x axis. Doing so makes it visually accurate in that areas under curves are visually representative of magnitudes of sensitivity coefficients integrated over energy ranges.

If incident energy grids for secondary distributions are requested, then an energy-resolved profile in the above format is given for each incident energy bin. For cosine bins, if an ERG parameter is specified, then additional grids in the above format is given for each cosine bin. If no ERG parameter is specified, but COS bins are, then the following results are given for all outgoing energies:

```
CLOW    CHIGH     SENS    REL UNC
```

Here *CLOW* and *CHIGH* are the lower and upper cosine bounds.

If the KOPTS option *KSENTAL* = *MCTAL*, results will be output in a special MCTAL-formatted file called **ksental**. This MCTAL file format is very much like the standard MCTAL file except that the symbols for bins have different meanings. These are:

F	spatial zones as cells or materials (0 denoting all cells)
D	unused
U	unused
S	isotopes
M	reaction MTs
C	cosine bins
E	energy bins
T	incident energy bins (for fission χ or scattering laws)

The MCNP tally plotter, MCPLOT may be loaded to plot these results. Again, the results should be normalized to be per unit lethargy with the "lethargy" option and plotted on a semi-log x axis for visually accurate area plots.

If the KOPTS option *KSENTAL* = *TSUNAMI-B*, results will be output in TSUNAMI-B format. The TSUNAMI-B format is given in the n[Table 6.5.A.2 of 323]. Because the SCALE and MCNP6 sensitivity capabilities are different, not all concepts in each code perfectly translate. In writing the TSUNAMI-B file format, the MCNP code will do the following:

- Multiple energy grids, which is possible in the MCNP code by multiple uses of the KSEN card, are not supported by TSUNAMI-B. To handle this, each instance of the KSEN card is listed in the file one after the other. For use in SCALE plotting tools, these will need to be split into multiple files.
- Unlike the MCNP code, energy units in SCALE are in eV, not MeV. The TSUNAMI-B format gives the energies in eV.
- The concept of a unit is not defined in the MCNP code, and the portion of the header that reports a unit number will give a 0 if no spatial zoning is involved, 1 if the zoning is by cell, and 2 if it is by material. The entry that follows (normally the region within the unit) is an enumeration of each spatial zone (the first zone has a "1", the second a "2", and so on).

- The MCNP code may not be able to compute the volume of a region. In this case, the MCNP code prints zero to the TSUNAMI-B file.
- In the place where TSUNAMI-B reports the number of uses of the region, the MCNP code reports the number of spatial zones on this instance of KSEN.
- For fission-χ sensitivities, the ones reported are automatically summed over all incident energy grids as the TSUNAMI-B format does not support this.
- The TSUNAMI-B format does not support scattering laws, so these are omitted.

5.11 Superimposed Mesh Tallies

MCNP6 offers two different mesh tallies to the user. The TMESH tally was developed for the MCNPX code, while the FMESH tally was developed for MCNP5. Although similar, each method is characterized by its own syntax, card format, and output files. The user is encouraged to read about both methods and choose the one that is most appropriate for his or her problem.

5.11.1 TMESH: Superimposed Mesh Tally A

The TMESH tally is a method of graphically displaying particle flux, dose, or other quantities on a rectangular, cylindrical, or spherical grid overlaid on top of the standard problem geometry. Particles are tracked through the independent mesh as part of the regular transport problem. The contents of each mesh cell are written to the **RUNTPE** file and can be plotted with the MCNP6 geometry plotter superimposed over a plot of the problem geometry. The TMESH tally data are also written to the **MCTAL** file and can be plotted with the MCNP6 tally plotter, MCPLOT.

Further, the TMESH tally data are written to the **mdata** file at the end of each initial or restarted calculation. The **gridconv** utility [Appendix E.4] can convert the **mdata** file into a number of standard formats suitable for reading by various graphical analysis packages.

Four different mesh-tally types are provided by TMESH, depending on the information the user wishes to view:

Type 1	Track-Averaged Mesh Tally [§5.11.1.2]
Type 2	Source Mesh Tally [§5.11.1.3]
Type 3	Energy Deposition Mesh Tally [§5.11.1.4]
Type 4	DXTRAN Mesh Tally [§5.11.1.5]

Each of the four types has its own associated keywords and input values.

Examples involving the superimposed geometry TMESH tally are available in §6.4.3.

5.11.1.1 Setting Up the TMESH Tally in the MCNP Input File

All of the input for TMESH tallies must be in a dedicated set of cards in the MCNP input file data-card block. This set must start with a card containing the word TMESH in the first five columns and end with a card containing the word ENDMD in the first five columns. For each requested mesh tally (a maximum of 20 TMESH tallies are permitted), a minimum of four cards must exist between the TMESH and ENDMD cards: an RMESH, CMESH, or SMESH control card, and CORA, CORB, and CORC cards. Optional cards within the mesh-tally block include ERGSH and MSHMF. An FM tally multiplier card may be specified only for Type 1 TMESH mesh tallies; however, if an FM card is associated with a Type 1 mesh tally, it must appear outside of the TMESH/ENDMD card block. Each of these cards is described in the discussion that follows.

The basic structure of the desired mesh as well as what quantities are to be stored to the mesh tally are determined by a mesh control card (RMESH, CMESH, or SMESH). The general form of the control cards follow:

RMESH*n*:$\mathscr{P}$ *KEYWORD=value(s)* ...

CMESH*n*:$\mathscr{P}$ *KEYWORD=value(s)* ...

SMESH*n*:$\mathscr{P}$ *KEYWORD=value(s)* ...

where

RMESH	specifies a rectangular mesh;
CMESH	specifies a cylindrical mesh;
SMESH	specifies a spherical mesh;
n	is a user-defined mesh-tally number for which the last digit of n , defines the type (1, 2, 3, or 4) of mesh tally and, consequently, the type of information to be stored in the mesh; and
$\mathscr{P}$	is the particle type to be tallied—this parameter may or may not be required, depending on the mesh-tally type;
KEYWORD	options vary depending on the mesh-tally type.

The notation *X*MESH will be used in subsequent sections to indicate any of the three mesh geometries. Input keywords for the four mesh-tally types are described in sections that follow. Note that the chosen mesh-tally number must be different from all other tallies in the problem. For example, an F1:N tally will conflict with a RMESH1:N tally.

In addition to the *X*MESH control card, the following set of cards provides details about the TMESH mesh characteristics and must be present for each requested mesh tally:

CORA*n* $corra_{n,1}$ $corra_{n,2}$ ...

CORB*n* $corrb_{n,1}$ $corrb_{n,2}$ ...

CORC*n* $corrc_{n,1}$ $corrc_{n,2}$ ...

where *n* is the same user-defined mesh-tally number as that on the associated *X*MESH control card. The mesh tally number must end in 1, 2, 3, or 4 corresponding to the mesh tally type. The entries on the CORA, CORB, and CORC cards describe a mesh in three coordinate directions as defined by the mesh type (rectangular,

cylindrical, or spherical), prior to any transformation. Each tally type supports an optional TRANS keyword to allow the application of a coordinate transformation to the mesh.

To describe a rectangular mesh, the entries on the CORA card represent planes perpendicular to the x axis, CORB entries are planes perpendicular to the y axis, and CORC entries are planes perpendicular to the z axis. Bins do not have to be equally spaced.

To describe a cylindrical mesh, the middle coordinate, CORB, is the untransformed z axis, which is the symmetry axis of the cylinder, with radial meshes defined on the CORA input line. The first smallest radius must be equal to zero. The values following CORB define planes perpendicular to the untransformed z axis. The values following CORC are positive angles relative to a counter-clockwise rotation about the untransformed z axis. These angles, in degrees, are measured from the positive x axis and must have at least one entry of 360, which is also required to be the last entry. The lower limit of zero degrees is implicit and never appears on the CORC card.

For spherical meshes, scoring will happen within a spherical volume, and can also be further defined to fall within a conical section defined by a polar angle (relative to the $+z$ axis) and azimuthal angle. The CORA card entries are sphere radii; inner and outer radii are required. The CORB entries define the polar angle meshing in which the polar angle ranges from 0 to 180 degrees, the 1st bin must be greater than 0 degrees, and the last bin must be 180. The CORC entries are the same as in the cylindrical case, with the 1st bin greater than 0 degrees and the last bin equal to 360. It is helpful in setting up spherical problems to think of the longitude-latitude coordinates on a globe.

The "I" data-input notation [§4.4.5.1] is allowed, enabling a large number of regularly spaced mesh points to be defined with a minimum of entries on the coordinate lines. All of the coordinate entries must be monotonically increasing for the tally mesh features to work properly, but do not need to be equally spaced. It should be noted that the size of these meshes scales with the product of the number of entries for the three coordinates. Machine memory could become a problem for very large meshes with fine spacing.

Additional cards that can be used with TMESH mesh tallies include the following:

ERGSH*n* e_1 e_2

MSHMF*m* e_1 f_1 e_2 f_2 ... e_K f_K

FM*n* ...

where positive values on the ERGSH card, e_1 and e_2, are the lower and upper energy limits for information to be stored to mesh tally *n*. On the other hand, negative values of e_1 and e_2 represent lower and upper time limits (in shakes) for information to be stored to mesh tally *n*. The default is to consider all energies and all times. The value of *m* on the MSHMF card does not refer to a corresponding mesh tally; instead, *m* is an arbitrary user-assigned value between 1 and 9. The entries on the MSHMF card, e_k and f_k, are pairs of energies and the corresponding response functions; as many pairs as needed can be designated. Use of the FM card is limited to TMESH Type 1 mesh tallies [§5.11.1.2] and the card must not appear inside the TMESH card block.

Note that the type 1 (particle track) and type 3 (energy deposition) mesh tallies work with heavy ions although there is no capability to separate out contributions from particular heavy ion species.

5.11.1.2 Track-averaged TMESH Mesh Tally (Type 1)

Caution

Due to copyright concerns the `DOSE` keyword has been deactivated and the built-in flux-to-dose conversion factors removed from the source code. They are available in Appendix F.1 formatted as MCNP input for `DE`/`DF` cards. Use the `MFACT` keyword and the `MSHMF` card to add a flux-to-dose conversion response function.

The first `TMESH` mesh type scores track-averaged data such as flux, tracks, population and energy deposition. The `MSHMF` card can be used to apply a response function.

Data-card Form: *XMESHn:𝒫 KEYWORD=value(s)...*

n	Type 1 mesh-tally type identifier. Restriction: $n = 1, 11, 21, \ldots$
$\mathscr{P}$	the particle type(s).
TRAKS	If TRAKS appears on the input line, tally the number of tracks through each mesh volume. No values accompany the keyword.
FLUX	If FLUX appears on the input line, then the average fluence is particle weight times track length divided by volume in units of number/cm^2. If the source is considered to be steady state in particles per second, then the value becomes flux in number/cm^2/second. No values accompany the keyword. (DEFAULT)
POPUL	If POPUL appears on the input line, tally the population (i.e., weight times the track length) in each volume.
PEDEP	If PEDEP appears on the input line, scores the average energy deposition per unit volume (MeV/cm^3/history) for the particle type $\mathscr{P}$. In contrast to the 3rd type of mesh tally, energy deposition can be obtained in this option for any particular particle. This option allows one to score the equivalent of an F6:$\mathscr{P}$ heating tally for the particle type $\mathscr{P}$. Note, the mesh is independent of problem geometry, and a mesh cell may cover regions of several different masses. Therefore the normalization of the PEDEP option is per mesh cell volume, not per unit mass.
MFACT	Can have from one to four numerical entries following it. The value of the first entry, *m*, is an arbitrary number that refers to an energy-dependent response function given on an MSHMF*m* card. If $m = -1$, then it is followed by a single value that is used as a constant multiplier. (No default) The second entry is 1 for linear interpolation and 2 for logarithmic interpolation. (DEFAULT is 1) If the third entry is 0, the response is a function of the current particle energy; if the third entry is 1, the response is a function of the energy deposited (only valid with the PEDEP option). (DEFAULT is 0) The fourth entry is a constant multiplier and is the only floating-point entry allowed. (DEFAULT is 1.0) If any of the last three entries are used, the entries preceding it must be present so that the order of the entries is preserved. Only one MFACT keyword may be used per tally.

TRANS	Must be followed by a single reference to a TR card number that can be used to translate and/or rotate the entire mesh. Only one TR card reference is permitted with each card (1).

Default: None

Details:

1 If a TR card is used with a TMESH tally, it must appear outside of the mesh data block between the TMESH and ENDMD cards.

It is possible to use the FM tally multiplier card to calculate reaction rates in a type 1 mesh tally if both of the following criteria hold:

1. the FM card must not appear within the mesh data block between the TMESH and ENDMD cards; and
2. if the multiplier involves a MT reaction identifier, the FM card must be included in an equivalent F4 tally specification.

5.11.1.3 Source TMESH Mesh Tally (Type 2)

The second type of mesh tally scores source-point data, in which the weight of the source particles $\mathscr{P}_1$, $\mathscr{P}_2$, $\mathscr{P}_3$, ..., $\mathscr{P}_n$ are scored in mesh arrays $1, 2, 3, ..., n$. A separate mesh tally grid will be produced for each particle chosen.

The usefulness of this method involves locating the source of particles entering a certain volume, or crossing a certain surface. The user asks the question, "If particles of a certain type are present, where did they originally come from?" In shielding problems, the user can then try to shield the particles at their source.

This mesh tally is normalized as number of particles per SDEF source particle.

Data-card Form: *XMESHn* $\mathscr{P}_1$ $\mathscr{P}_2$ ... *KEYWORD=value(s)...*

n	Type 2 mesh-tally type identifier. Restriction: $n = 2, 12, 22, \ldots$
$\mathscr{P}_k$	Particle designators, i.e., n, p, e, etc. (See Table 4.3) Restriction: $k \leq 10$ Source particles are considered to be those that come directly from the source defined by the user and those new particles created during nuclear interactions. One should be aware that storage requirements can get very large, very fast, depending on the dimensions of the mesh, because a separate histogram is created for each particle chosen. If there are no entries on this card, the information for neutrons is scored by default.
TRANS	Must be followed by a single reference to a TR card number that can be used to translate and/or rotate the entire mesh. Only one TR card reference is permitted with each card.

5.11.1.4 Energy Deposition TMESH Mesh Tally (Type 3)

The third type of mesh tally scores energy deposition data in which the energy deposited per unit volume from all particles is included. This can be due to the slowing of a charged particle, the recoil of a nucleus, energy deposited locally for particles born but not tracked, etc. The results are similar to the scoring of an +F6 tally.

Note that in MCNP6 the option to track energy deposition from one type of particle alone in a problem is included in the first mesh tally type. See the PEDEP keyword in §5.11.1.2. The energy deposition mesh tally described here gives results for all particles tracked in the problem, and has no option to specify a particular particle.

Because the mesh is independent of problem geometry, a mesh cell may cover regions of several different masses. Therefore the normalization of the output is per unit volume (MeV/cm^3/source particle), not per unit mass.

Data-card Form: *XMESHn KEYWORD=value(s)...*

n	Type 3 mesh-tally type identifier. Restriction: $n = 3, 13, 23, \ldots$
TOTAL	If TOTAL appears on the input line, score energy deposited from any source. No values accompany the keyword. (DEFAULT)
DE/DX	If DE/DX appears on the input line, score ionization from charged particles. No values accompany the keyword.
RECOL	If RECOL appears on the input line, score energy transferred to recoil nuclei above tabular limits. No values accompany the keyword.
TLEST	If TLEST appears on the input line, score track length folded with tabular heating numbers. No values accompany the keyword.
EDLCT	If EDLCT appears on the input line, score non-tracked particles assumed to deposit energy locally. This allows the user to ascertain the potential error in the problem caused by allowing energy from non-tracked particles to be deposited locally. This can be a serious problem in neglecting the tracking of high-energy photons or electrons. No values accompany the keyword.
MFACT	Can have from one to four numerical entries following it. The value of the first entry, *m*, is an arbitrary number that refers to an energy-dependent response function given on an MSHMF*m* card. If $m = -1$, then it is followed by a single value that is used as a constant multiplier. (No default) The second entry is 1 for linear interpolation, and 2 for logarithmic interpolation. (DEFAULT is 1) If the third entry is 0, the response is a function of the current particle energy; if the third entry is 1, the response is a function of the energy deposited. (DEFAULT is 0) The fourth entry is a constant multiplier and is the only floating-point entry allowed (DEFAULT is 1.0). If any of the last three entries are used, the entries preceding it must be present so that the order of the entries is preserved. Only one MFACT keyword may be used per tally.

TRANS	Must be followed by a single reference to a TR card number that can be used to translate and/or rotate the entire mesh. Only one TR card reference is permitted with each card.

5.11.1.5 DXTRAN TMESH Mesh Tally (Type 4)

The fourth type of mesh tally scores the tracks contributing to all point detectors defined in the input file for the $\mathscr{P}$ particle type. If this card is preceded by an asterisk (∗), tracks contributing to DXTRAN spheres [§5.12.10] are recorded. Obviously, a point detector or DXTRAN sphere must already be defined in the problem, and the tally will record tracks corresponding to all such defined items in the problem. The user should limit the geometric boundaries of the grid to focus on a specific detector or DXTRAN sphere in order to prevent confusion with multiple detectors (although the convergence of the particle tracks should help in the interpretation). This tally is an analytical tool useful in determining the behavior of detectors and how they may be effectively placed in the problem.

Data-card Form: *XMESHn:*$\mathscr{P}$ *KEYWORD=value(s)...*

n	Type 4 mesh-tally type identifier. Restriction: $n = 4, 14, 24, \ldots$
$\mathscr{P}$	the particle type [neutron (N) or photon (P)].
TRANS	Must be followed by a single reference to a TR card number that can be used to translate and/or rotate the entire mesh. Only one TR card reference is permitted with each card.

5.11.1.6 Processing the TMESH Mesh Tally Results

The values of the coordinates, the tally quantity within each mesh bin, and the relative errors are all written by MCNP6 to the **runtpe** file, the optional **mctal** file, and an unformatted binary file named **mdata**.

The mesh tallies may be plotted with the MCNP6 geometry plotter either during the course of a calculation (by placing an MPLOT card in the input file or by using the TTY interrupt capability to invoke MCPLOT) or after a calculation using the **runtpe** file and the MCNP6 geometry plotter. These plots are superimposed over 2-D views of the problem geometry. Note that the geometry plotter must be accessed via the tally plotter. For example,

```
MCNP6 Z
MCPLOT>RUNTPE=<filename>
MCPLOT>PLOT
PLOT>py 4 ex 40 or 0 4 0 la 0 1 tal12 color on la 0 0 con 0 100 %
```

After the PLOT command, the MCNP6 interactive geometry plotter appears. If the Plot> button (bottom center) is clicked, then the above command after the PLOT> prompt can be entered. Alternatively, the mesh tally superimposed on the geometry can be viewed by clicking buttons (tal, etc.) of the interactive tally plot. Note that the command tal12 has no space between tal and 12 and that the cell labels (la 0 1 tal12) must be turned on to set the color (color on) and then be turned off (la 0 0).

The second mesh tally processing option is to use the MCNP6 tally plotter (MCPLOT) after a calculation with the optional **mctal** file (see PRDMP card). For example,

```
MCNP6 Z
MCPLOT>RMCTAL=<filename>
tal 12 free ik
```

Note that there is a space between `tal` and `12` and that the mesh tally dimensionality $[i, j, k]$ corresponding to `CORA`, `CORB`, and `CORC`) must be specified.

The third mesh tally processing option is to post-process the **`mdata`** (or **`mctal`**) file with the **`gridconv`** utility [Appendix E.4] and then use an external graphics package.

5.11.2 FMESH: Superimposed Mesh Tally B

The `FMESH` card allows the user to define a mesh tally superimposed over the problem geometry. Tally results are either written to a separate output file or can be accessed from the runtape file via the XDMF output file. By default, the mesh tally calculates the track-length estimate of the particle flux averaged over a mesh cell in units of particles/cm^2. If an asterisk precedes the `FMESH` card, energy times particle weight will be tallied in units of MeV/cm^2. Other mesh-tally types include source points, partial current, and isotopic reaction rate tallies.

`FMESH` mesh tallies can be used in combination with the `DE`/`DF`, `FC`, `FM`, `SF`, `CF` and `TR` cards. With the surface (`SF`) and cell (`CF`) flagging cards, only one mesh tally, the flagged tally, is created. A separate mesh tally is needed for unflagged tally results.

Deprecation Notice **DEP-53292**

Except for `none` and `xdmf`, all output formats for the `FMESH` are deprecated.

Consistent with prior and current behavior, mesh tallies specified as output type `none` will only be written to the runtape file for the purpose of restarting the calculation and/or for use within the interactive plotter.

Mesh tallies specified as output type `xdmf` will create a separate XDMF [324, 325] file, named **`meshtal.xdmf`** by default. This file contains metadata which is then used to access the mesh tally data and associated attributes from the runtape file. This file permits direct and hierarchical access to the mesh tally results in the runtape with a variety of programming languages and also straightforward 3-D visualization with third-party software such as ParaView [326] and VisIt [327].

Note that this option will also create a new HDF5 group on the runtape file, **`/results/mesh_tally`**, which is used by the XDMF file to access the mesh tally data. For more details, see D.4.

Data-card Form: `FMESH`*n*:$\mathscr{P}$ `keyword = value(s)...`

`n`	Tally number ending with `4` or `01`
$\mathscr{P}$	A single particle designator.
`geom`	Mesh geometry, either Cartesian (`XYZ` or `REC`) or cylindrical coordinates (`RZT` or `CYL`). (DEFAULT: `geom = XYZ`)
`origin`	Coordinates `(x,y,z)` of the origin of the mesh in terms of the MCNP cell geometry (DEFAULT: `origin = 0.0, 0.0, 0.0`) (1). If

`geom` = `XYZ`	the origin corresponds to the bottom, left, behind of a rectangular mesh.
`geom` = `RZT`	the origin corresponds to the bottom center of a cylindrical mesh.

`axs` Vector giving the direction of the axis of the cylindrical mesh (2). (DEFAULT: `axs` = `0.0, 0.0, 1.0`)

`vec` Vector defining, along with `AXS`, the plane for (2). (DEFAULT: `vec` = `1.0, 0.0, 0.0`)

`imesh` Locations of the coarse mesh points in the x direction for rectangular geometry or in the r direction for cylindrical geometry (3). (DEFAULT: none)

`iints` Number of fine mesh points within each corresponding coarse mesh in the x direction for rectangular geometry or in the r direction for cylindrical geometry. (DEFAULT: `iints` = `1`)

`jmesh` Locations of the coarse mesh points in the y direction for rectangular geometry or in the z direction for cylindrical geometry (3). (DEFAULT: none)

`jints` Number of fine mesh points within each corresponding coarse mesh in the y direction for rectangular geometry or in the z direction for cylindrical geometry. (DEFAULT: `jints` = `1`)

`kmesh` Locations of the coarse mesh points in the z direction for rectangular geometry or in the θ direction (in revolutions) for cylindrical geometry (3). (DEFAULT: none)

`kints` Number of fine mesh points within each corresponding coarse mesh in the z direction for rectangular geometry or in the θ direction for cylindrical geometry. (DEFAULT: `kints` = `1`)

`emesh` Values of the coarse mesh points in energy in MeV. (DEFAULT: `emesh` = `0.0`, $E_{\mathscr{P}_{\max}}$)

`eints` Number of fine mesh points within each corresponding coarse mesh in energy. (DEFAULT: `eints` = `1`)

`enorm` Energy normalization. (DEFAULT: `enorm` = `no`) If

`enorm` = `no`	then the tally results are not divided by energy bin width.
`enorm` = `yes`	then the tally results are per unit energy (MeV^{-1}).

`tmesh` Values of the coarse mesh points in time in shakes (4). (DEFAULT: `tmesh` = $-\infty$, $T_{\max}$)

`tints` Number of fine mesh points within each corresponding coarse mesh in time. (DEFAULT: `tints` = `1`)

`tnorm` Time normalization. (DEFAULT: `tnorm` = `no`) If

`tnorm` = `no`	then the tally results are not divided by time bin width.

	`tnorm` = `yes`	then provides the tally results per shake (sh^{-1}).
`factor`	Multiplicative factor for each mesh. (DEFAULT: `factor` = `1.`)	
`out`	Output format, either in column format or as a series of 2-D matrices. If	
	`out` = `col`	a columnar output format is provided, listing the coordinates of the center of the bin, the tally results, and associated relative error. (DEFAULT)
	`out` = `colsci`	the same as `out` = `col` but with all values formatted using scientific notation.
	`out` = `cf`	a columnar output format is provided, listing the coordinates of the center of the bin, the tally results, and associated relative error. In addition, the volume and the tally results multiplied by the volume are also printed.
	`out` = `cfsci`	the same as `out` = `cf` but with all values formatted using scientific notation.
	`out` = `ij` or `ik` or `jk`	tally results are printed as a series of two 2-D matrices, with $I = x$ or r, $J = y$ or z, and $K = z$ or θ, depending on the coordinate system chosen. The first matrix contains the tally results, and the second matrix the relative errors. The rows and columns are labeled by the mid-points of the corresponding mesh bins. These pairs of matrices will be printed for each mesh bin in the third coordinate.
	`out` = `none`	no **`meshtal`** file is printed (but the information is still written to the runtape file (5)).
	`out` = `xdmf`	a **`meshtal.xdmf`** file is created that can be used to interrogate tally information that is additionally written to the runtape file. Because of other formats being deprecated [DEP-53292], one cannot combine XDMF output with another type of output except `none` in the same input file. (6, 7).
`tr`	Number of the transformation to be applied to the mesh (8). (DEFAULT: none)	
`inc low high`	Defines a range of collisions that will contribute to the `FMESH` tally. (DEFAULT: `low` = `0`; `high` = `infinite`) For a particle track undergoing `n` collisions, the track will contribute to the `FMESH` tally if `LOW` $\leq$ `n` $\leq$ `HIGH`. The specification of `LOW` is optional. If only `LOW` is specified, then a particle track undergoing exactly `n` collisions will contribute to the `FMESH` tally if `n` = `LOW`. The keyword entry `INFINITE` can be used for `HIGH` to represent an infinite number of collisions.	
`type`	Allows users to specify the quantity being tallied. If	
	`type` = `flux`	volumetric track-length fluxes are tallied. See `F4` type tally. (DEFAULT)
	`type` = `source`	source points are tallied.

`kclear`	Used in KCODE calculations for generating visualizations of cycle wise quantities. Zeros out the mesh tally every `n` KCODE cycles, where `n` is specified with `kclear` = `n`. If `n` is non-zero, then tallies are accumulated both during active and inactive cycles; consequently, the output should only be used for visualizations and not as actual results. If `n` is zero, then the mesh tally behaves normally and mesh tally results are never cleared. (DEFAULT: `kclear` = `0`)
`tally`	Tallying algorithm. See §5.11.2.4 for more details.

`tally` = `hist`	basic history statistics-based algorithm.
`tally` = `fast_hist`	high efficiency history statistics-based algorithm. Corresponds with the approach from version 6.2 and prior releases of the MCNP code. (DEFAULT)
`tally` = `batch`	batch statistics-based algorithm.
`tally` = `rma_batch`	batch statistics-based algorithm using remote memory access. Requires MPI and compatible software and hardware.

Use: Optional

Details:

1. The location of the nth coarse mesh in the u direction ($r_{u,n}$ in what follows) is given in terms of the most positive surface in the u direction. For a rectangular mesh, the coarse mesh locations $(r_{x,n}, r_{y,n}, r_{z,n})$, are given as planes perpendicular to the x, y, and z axis, respectively, in the MCNP cell geometry coordinate system. Thus the `origin` point (x_0, y_0, z_0) is the most negative point of the mesh tally. For a cylindrical mesh, `origin` defines the bottom center point of the mesh. The z coordinate is then measured from the cylindrical mesh origin. For both types of geometry, the lowest energy value is 0 MeV. The coarse mesh locations and energy values must increase monotonically (beginning with the `origin` point). The fine meshes are evenly distributed within the nth coarse mesh in the u direction.

2. For a cylindrical mesh, the `axs` and `vec` vectors need not be orthogonal but they must not be parallel; the one half-plane that contains them and the `origin` point will define $\theta = 0$. The `axs` vector will remain fixed. The length of the `axs` or `vec` vectors must not be zero. The z coordinate is specified in the cylinder geometry coordinate system. The θ coarse mesh locations are given in revolutions and the last one must be 1.

3. At least one coarse mesh per coordinate direction must be specified using `imesh`, `jmesh`, and `kmesh` keywords. The code uses a default value of 1 fine mesh per coarse mesh if the `iints`, `jints`, or `kints` keywords are omitted. If the `iints`, `jints`, or `kints` keywords are present, the number of entries must match the number of entries on the `imesh`, `jmesh`, and `kmesh` keywords, respectively. Entries on the `iints`, `jints`, and `kints` keywords must be greater than zero.

4. Because the lower time bound is minus infinity, users are encouraged to specify the first bin as a dummy bin with the smallest time of interest (usually zero shakes). The user should then ignore the first time bin when plotting.

5. If the FMESH card is present in a restarted calculation, only the `out` keyword is permitted.

6. Appendix D.4 describes how to use the **`meshtal.xdmf`** file to plot mesh tally results with the third-party 3-D visualization software ParaView [326]. It also describes the new FMESH tally HDF5 hierarchy on the runtape file.

(7) Both the runtape file and the XDMF output are affected by the PIO card.

(8) Any FMESH mesh can be transformed using the tr keyword followed by a transformation number. The transformation is defined on the associated TR card.

(9) As with the F card, a unique number is assigned to each FMESH tally relative to F4 tallies. Since only track-length mesh tallies are typical, the mesh tally number must end with a 4, and it must not be identical to any number that is used to identify an F4 tally. The track length is computed over the mesh tally cells, and is normalized to be per starting particle, except in KCODE criticality calculations, in which results are usually normalized by the active cycle weight [§2.8.2.9].

5.11.2.0.1 Example 1

Listing 5.60: example_fmesh.mcnp.inp.txt

```
FC4   Cylindrical FMESH tally example
FMESH4:n GEOM=CYL   ORIGIN= -100 0 0
         IMESH=   5  10   IINTS=  5 2
         JMESH= 100 200   JINTS= 10 5
         KMESH= 0.5   1   KINTS= 1 2
         AXS= 1 0 0  VEC=0 1 0
```

This example describes a cylindrical mesh tally along the x axis, with base at $x = -100$ and $\theta = 0$ along the $+y$ axis. The tally is divided into five bins from $r = 0$ to $r = 5$, two bins from $r = 5$ to $r = 10$, ten bins from $z = 0$ to 100, five bins from $z = 100$ to $z = 200$, one bin from $\theta = 0°$ to $\theta = 180°$, and two bins from $\theta = 180°$ to 360°.

5.11.2.1 Special Cases of the FM Tally Multiplier Used in Conjunction with the FMESH Mesh Tally

5.11.2.1.1 Default Materials

When the FMESH capability is associated with a tally multiplier (FM) card, then the material number specified on the FM card determines the cross sections that are used to calculate the mesh bin values. That is, the cross sections of the specified material are used for the entire mesh, even if the mesh covers several different materials. If instead, a "0" is entered as the material number on the FM card, then MCNP will use the reaction data of the material through which the particle travels to calculate the bin values. Thus, material-dependent quantities that are computed with the use of the FM card (e.g., neutron heating) can be calculated using mesh tallies that cover more than one material.

5.11.2.1.2 Example 1

Listing 5.61: example_fmesh.mcnp.inp.txt

```
FC14  FMESH tally energy deposition (in MeV/cm^3)
FMESH14:n GEOM=xyz    ORIGIN= -5 -5 -5
          IMESH=  5   IINTS=100
          JMESH=  5   JINTS=  1
          KMESH=  5   KINTS=100
FM14    -1  0  1  -4
```

This example describes an energy-deposition rectangular-mesh tally covering a $10 \times 10 \times 10$ cm box centered on the origin. The tally is divided into 100 bins in both the x and z directions and one bin in the y direction. The FM4 card specifies the energy deposition for all materials in the mesh.

Caution

The use of parentheses anywhere on an FMESH FM card can cause the code to exit with a fatal error.

5.11.2.2 Isotopic Reaction Rate Tallies

Individual isotopic reaction rates can be obtained throughout the mesh tally geometry. To invoke this capability, define a new dummy material card containing only the isotope(s) of interest. This dummy material card should be specified in exactly the same way as a standard M card; however, the dummy material number should not appear in the problem geometry—instead the material number is used exclusively by the FM card. Note that for these dummy materials, the isotopic densities are not used by MCNP but values must be provided as placeholders. Instead, the required isotopic atom fractions will be extracted from the appropriate material data used during transport.

To specify a reaction-rate mesh tally based on isotopic fractions, place a "+" symbol in front of the FM card associated with the mesh tally. The rest of the FM card is set up the regular way, with a multiplicative constant followed by the dummy material number and the ENDF reaction numbers of interest.

When calculating the mesh tally, MCNP will multiply the particle flux times the cross sections for the isotopes defined on the material card. This value is then multiplied by the atom fraction of the dummy material isotope(s) which are present in the material in which the particle is traveling to calculate the isotopic reaction rate. Depending on the units of the source, the units of the results will be (number of reactions)$\cdot$barn$^{-1}\cdot$cm^{-1}, or (number of reactions) $\cdot$ barn^{-1} $\cdot$ cm^{-1} $\cdot$ shake^{-1}.

Recall that placing a minus sign in front of the multiplicative constant will multiply the results by the atom density of the cell. Therefore using c $= -1$ will return units of (number of reactions) $\cdot$ cm^{-3}, or (number of reactions) $\cdot$ cm^{-3} $\cdot$ shake^{-1} for the specific isotopes.

See §10.2.3 for examples of isotopic reaction rate mesh tallies

5.11.2.3 Special Case of Leakage Tallies using the FMESH Mesh Tally

As a very special and limited extension of the FMESH mesh tallies, FMESH cards with a tally number ending in the two numerals 01 can be used to obtain the outgoing leakage (or outgoing partial current) across each of the 6 faces of each mesh element.

Details:

1. Outgoing partial current FMESH tallies defined in the problem input must end in "01".
2. Specifying FMESH901:n in the problem input will result in the following FMESH tallies being created internally:
   - FMESH911:n: outgoing partial current in the $+x$ or $+r$ direction
   - FMESH921:n: outgoing partial current in the $-x$ or $-r$ direction

- FMESH931:n: outgoing partial current in the $+y$ or $+z$ direction
- FMESH941:n: outgoing partial current in the $-y$ or $-z$ direction
- FMESH951:n: outgoing partial current in the $+z$ or $+\theta$ direction
- FMESH961:n: outgoing partial current in the $-z$ or $-\theta$ direction

(3) All 6 of the partial current tallies will have the same specifications that are supplied for FMESH901:n. The specific tally FMESH901:n will not actually be stored or be available for referencing with FM, DE/DF, SC, or SF cards.

(4) These tallies are not divided by volume or area. They produce the total particle weight crossing each surface of a mesh cell in the outward direction, normalized to be per unit source particle.

(5) The incoming partial currents to a mesh element can be obtained from the outgoing partial currents of neighboring elements. Since the partial current FMESH tally only tallies outward currents for each mesh element, it is necessary to specify the mesh to include a "halo" of inactive elements surrounding the active problem domain in order to properly capture incoming current at the boundary of the problem domain.

(6) Use of the "*" prefix, as in *FMESH901:n, is permitted. This will result in the FMESH partial current tallies providing the energy crossing each mesh element surface in the outward direction.

(7) The tally modifier cards, such as FM, DE/DF, CF, or SF may be used with the partial current tallies (although most modifiers don't make physical sense). To do so, however, the tally number for only the first of the created FMESH tallies must be used. For example, if FMESH901:n is specified, then tally number 911 should be used on any FM, DE/DF, CF, or SF cards. Those cards will also be applied to the FMESH912:n, ..., FMESH916:n partial current mesh tallies. Note that, FM modifiers that are typically used with flux tallies may not be appropriate with partial currents.

(8) The partial current tallies can be plotted, using the tally numbers for the 6 created mesh tallies. For example, in the plotter one can specify "**fmesh 911**" for the example above.

(9) The tallies appear in the standard format in the **meshtal** file, with the 6 names of tallies created, e.g., FMESH911, and can be combined using merge_meshtal.

The FMESH partial current tallies are deliberately limited in scope and usage. They are provided primarily so that users can obtain a complete particle balance for individual mesh elements. That is, using FMESH4 tallies for particle production and particle capture, the FMESH01 tallies provide the leakage across mesh element surfaces.

5.11.2.4 FMESH Mesh Tally Algorithms

The tallying system provides four algorithms, History (tally = hist), Fast History (tally = fast_hist), Batch (tally = batch), and Batch RMA (tally = rma_batch). Each algorithm has various tradeoffs.

To summarize:

- All methods will give the same mean when the same tally events occur (for KCODE, bear in mind (1)).
- Methods that use batch statistics will have a higher variance of the error estimate than history statistics. This gets worse with fewer batches. See §2.6.11. One should use at least 100 batches, preferably more, for this reason.
- For KCODE, batch statistics can yield slightly more conservative error estimates as compared to history, as they properly handle correlation between histories in a given generation. Correlation between generations is not yet handled.

Table 5.24: Approximate Peak FMESH Memory Usage

Algorithm	Peak Node Memory Usage, Bytes
`hist`	$16T_{\text{size}}n_{\text{rank}} + 8T_{\text{size}}n_{\text{rank}}n_{\text{threads}}$
`fast_hist`	$16T_{\text{size}}n_{\text{rank}} + 8.8T_{\text{size}}n_{\text{rank}}n_{\text{threads}}$
`batch`	$16T_{\text{size}} + 8T_{\text{size}}n_{\text{rank}}$
`rma_batch`	$24T_{\text{size}}/n_{\text{nodes}}$

- For simulation speed, generally `hist` $\ll$ `fast_hist` $<$ `batch`. For `rma_batch`, it depends on the size of the problem and the MPI library.
- For memory usage, generally `batch_rma` $<$ `batch` $\ll$ `hist` $<$ `fast_hist`. An approximate memory usage estimate can be found in Table 5.24. T_{size} is the number of tally regions, n_{rank} is the number of MPI processes per node, n_{nodes} is the number of nodes, and $n_{\text{threads}}$is the number of threads per MPI process.
- As shown in that table, all except `rma_batch` use memory more efficiently when one maximizes the number of threads (via `tasks`, see § 3.3.2.3) and minimizes the number of MPI processes.

In general, if you do not know which to pick, `fast_hist` is a safe choice and is default for that reason. `batch` will perform better and use less memory, at the cost of the quality of the statistics. `rma_batch` is ideal for use with extremely large tallies ($> 10^8$ regions) that are beyond the reach of the other algorithms. `hist` is only useful for very small tallies for performance reasons.

5.11.2.4.1 Fast History Tally Algorithm (`tally` = `fast_hist`)

This algorithm, which is the default, acts as a drop-in replacement to the algorithm used in versions of the MCNP code prior to 6.3. Relative to the previous version of the algorithm, it should run faster and use less memory in most circumstances. The approach used to compute statistical parameters is identical to the previous version as well, so answers should be identical within numerical roundoff.

5.11.2.4.2 History Tally Algorithm (`tally` = `hist`)

This algorithm removes an optimization used in the Fast History algorithm that speeds up problems in which a single particle history touches a small fraction of tally regions. As a result, it should be slower on all but the smallest problems. Statistics are computed quite differently in this mode. First, statistical moments are computed using a more numerically stable approach described in [328]. Second, the number of degrees of freedom used for the standard error is always the number of active histories simulated minus one. This will result in slight changes to the standard error. The mean will match Fast History within roundoff.

5.11.2.4.3 Batch Tally Algorithm (`tally` = `batch`)

This algorithm switches from history to batch statistics, in order to take advantage of the performance benefits. When running a fixed source problem, batch size is set by the *n_per_batch* option on the NPS card. The batch count is then *npp*/*n_per_batch*. For KCODE problems, the batch size is *nsrck*, and the batch count is given by the active generations. In KCODE problems, when a tally that uses batch statistics is detected, the fission bank will be resampled to always be a fixed size. This will change results when compared to not resampling, but the solutions are equivalent within statistics. One can compare algorithms by running both within the same simulation.

5.11.2.4.4 Remote Memory Access Batch Tally Algorithm (`tally` = `rma_batch`)

This algorithm is only available if the MCNP code is built with MPI. Otherwise, the Batch Tally Algorithm is used. This algorithm is mathematically identical to the Batch Tally Algorithm and will give identical results, but the two results arrays and the worker array are uniformly distributed across all processes and nodes in the problem. This means that running the problem with more nodes of a cluster will increase the maximum problem size.

While this algorithm allows for the largest possible tallies of all the methods, it comes with some caveats. The first is that it will generally be slower than the Batch algorithm. Second, performance is very sensitive to the MPI library and cluster interconnect. See the build guide [329] for more details. Third, if the **`MPI_THREAD_MULTIPLE`** build option is disabled (which is default, visible by running `mcnp6.mpi -v`), it is best to run simulations only with MPI (no `tasks`) for performance reasons.

5.11.3 SPDTL: Lattice Tally Speed Enhancement

The SPDTL card allows the user to force or prevent the use of the lattice speed tally enhancement [330]. This feature allows the user to run a short test case with and without the enhancements to verify they are appropriate by comparing the tally results of the two runs.

Data-card Form: `SPDTL` *value*

value	Toggles whether to force or prohibit lattice speed tally enhancement. If
value = `FORCE`	Force the use of the lattice speed tally enhancement feature. No values accompany the keyword (2).
value = `OFF`	Prevent the use of the lattice speed tally enhancement feature. No values accompany the keyword.

Default: Lattice speed tally enhancement is enabled by default if strict criteria are met.

Use: Optional.

Details:

1. Only one keyword may be specified for SPDTL.
2. Using SPDTL `FORCE` also causes comments to be printed about lattice speed tally enhancement conflicts with other cards.

5.11.3.1 Conditions Required for Lattice Speed Tally Enhancements

The lattice speed tally enhancements greatly reduce the runtime of certain problems, namely large lattices used for voxel phantoms. This enhancement will only work under certain conditions, which MCNP6 will try to detect. If any of the following criteria are not met, then the lattice speed tally enhancement will not be used unless the SPDTL `FORCE` card is used. Using the SPDTL `FORCE` card to run the lattice speed tally

enhancement is discouraged, since it may result in a program crash, tally values that are all zeros, or silent wrong answers.

Criteria that must be met for MCNP6 to automatically (and appropriately) use the lattice speed tally enhancement include the following:

1. A hexagonal lattice must be present in the geometry.
2. All F4 tallies contain a hexahedral lattice.
3. None of the following cards are used: DXT, DXC, F1, F2, *F4, F6, F7, F8, +F8, PERT, WWG, WWGE, WWGT.
4. None of the following cards are used to modify an F4 tally: FT, E, EM, T, TM, CF, SF, FS, C.
5. All F4 tallies have an associated FM4 card that contains only a single digit multiplier.
6. All F4 tallies have associated DE/DF cards.

The following criteria are not checked by MCNP6. The user must verify that the input deck meets these criteria:

1. Nested lattices are not tallied over.
2. The entries for a cell's FILL card do not include that cell's own universe number.
3. The full lattice index range is given on every lattice on each F4 tally card.

For more information, see [330].

5.12 Variance Reduction-focused Data Cards

Many of these variance-reduction cards require knowledge of both the Monte Carlo method and the particular variance reduction technique employed. Section 1.2.7 and its references are a good place to start learning more about these topics.

Only two variance reduction games in MCNP6 are enabled by default: implicit capture/weight cutoff and Russian roulette for point detectors and DXTRAN spheres. All other variance reduction games must be applied explicitly and therefore are considered optional. In spite of this statement, the code does require that either (1) the IMP card be present in the data-card section of the MCNP input file (or, equivalently, an IMP parameter be specified on each cell card) or (2) weight windows be supplied through WWN cards (or, equivalently, read from a **WWINP** file). Otherwise, a fatal error will occur during the input-checking process.

Some variance reduction cards (e.g., IMP) in the data section require the number of entries to equal the number of cells or surfaces in the problem; otherwise, a fatal error results. For other cards (e.g., EXT) no fatal error results if the number of entries does not equal the number of cells or surfaces, but a warning may be issued. The order of the cells or surfaces on these cards correspond in order to the cell or surface cards that appear in the MCNP input file. The *n*R repeat or *n*J jump features may help in supplying the desired values. Note that the *n*J feature relies on the presence of a default value. Users should refer to the individual cards to learn about their defaults.

5.12.1 IMP: Cell Importance

A cell's importance is used

1. to terminate the particle's history when a particle enters a cell with importance zero,
2. for playing geometry splitting and Russian roulette as a means to control the particle population upon entering a cell, and
3. for scaling the cutoffs in the weight cutoff game. An importance assigned to a cell that is in a universe is interpreted as a multiplier of the importance of the filled cell.

Caution

The splitting behavior that takes place as particles enter and exit UM pseudocells as a result of defining varying pseudocell importances for adjacent pseudocells may lead to potentially silent wrong answers with UM geometry or, more clearly, seemingly unrelated issues such as the code reporting negative emission energy following certain collisions. Rather than using cell-based importances, it is recommended to use cell-based weight windows and to set *mwhere* $= -1$ on the WWP card to avoid such issues, which arise because of particle-banking behavior as particles enter and exit UM pseudocells.

Cell-card Form: IMP:$\mathscr{P}$=x
or
Data-card Form: IMP:$\mathscr{P}$ x_1 x_2 ... x_K

$\mathscr{P}$	Any particle symbol from Table 4.3. May also be a list of particle symbols separated by commas as long as the importance are the same for the desired importance are the same for the different listed particle types.
x	Cell importance. One entry must appear on each cell card for each particle type that has non-default values.
x_k	Importance of cell k. Number of entries must equal number of cells in the problem.

Default: Default IMP values are variable and depend on the presence or absence of other cards as illustrated below. For this reason, it is highly recommended that the user explicitly specify IMP values for all particle types or verify from PRINT Table 60 the values used in the calculation are those intended.

If no WWN card is present: IMP values are explicitly required for one of the requested particle types on the MODE card, otherwise a fatal error occurs. Additionally, if (1) one particle is explicitly assigned IMP values and (2) IMP values are not supplied for the other particle types, then the default IMP values for the remaining particles are 1 where the explicitly assigned importance are greater than 0 and 0 where the explicitly assigned importance are 0.

If a WWN card is present: IMP values are not required when using cell-based weight windows. However, one set of IMP values is required when using mesh-based weight windows. The default IMP values for the particle(s) on the WWN card are set to 1 where the weight-window lower bounds are not -1, otherwise they are set to 0. IMP values for all other particles not having a WWN card are set to 0. If IMP values are explicitly provided along with the WWN card(s), they are retained and IMP values not explicitly provided for any other particles are set to 1 where the explicitly set IMP values are not 0; otherwise, they are set to 0.

If a cell importance is set to 0 for any particle, all particle importance for that cell will be set to 0 (default implicit value) unless specified otherwise. However, if the *nJ* feature is used to specify importance, the values

jumped over are given a default importance value of 0. Particles entering a cell with an importance value of 0 are immediately terminated as are contributions to detectors and DXTRAN spheres. The outside world cell (surrounding the geometry of interest) should be such a cell; problems without such a cell will experience lost particles.

Use: Use IMP when weight windows are not desired. See details in the default discussion above.

Different particle types can be split differently by having separate IMP:$\mathscr{P}$ cards. When using the data card entry format, it is a fatal error if the number of entries on any IMP:$\mathscr{P}$ card is not equal to the number of cells in the problem. Similarly, if an IMP:$\mathscr{P}$ parameter appears on one cell card, a fatal error occurs if a comparable entry does not appear on all cell cards. The *n*R repeat and *n*M multiply features are especially useful with this card in the data-card section. Be careful when using these shorthand notations together: R does not duplicate the M, but rather the value that the M notation creates.

A track will neither be split nor rouletted when it enters a void cell even if the importance ratio of the adjacent cells would normally call for a split or roulette. However, the importance of the non-void cell that a particle exits is remembered and splitting or Russian roulette will be played when the particle next enters a non-void cell. As an example of the benefit of not splitting into a void, consider a long cylindrical void (or pipe) surrounded by a material like concrete where the importance are decreasing radially away from the pipe. Considerable computer time can be wasted by tracks bouncing back and forth across the pipe and doing nothing but splitting, then immediately undergoing roulette. Splitting into a void increases the time per history but has no counterbalancing effect on the expected history variance. Thus, the figure of merit (FOM) is reduced by the increased time per history.

If a superimposed weight-window mesh is used, the IMP card is required. Cell importance are only used for the weight cutoff game in zero-window meshes.

Caution

The splitting behavior that takes place as particles enter and exit UM pseudocells as a result of defining varying pseudocell importances for adjacent pseudocells may lead to potentially silent wrong answers with UM geometry or, more clearly, seemingly unrelated issues such as the code reporting negative emission energy following certain collisions. If one must use weight windows with UM, it is recommended that *mwhere* $= -1$ be set to avoid such issues, which arise because of particle-banking behavior as particles enter and exit UM pseudocells.

5.12.1.1 Example 1

```
IMP:N    1  2  2M  0  1  20R
```

The neutron importance of cell 1 is 1, cell 2 is 2, cell 3 is 4, and cell 4 is 0. The importance for cells 5 through 25 are 1. A track will be split 2 for 1 going from cell 2 into cell 3, each new track having half the weight of the original track before splitting. A track moving in the opposite direction will be terminated in half the cases (that is, with probability 0.5), but it will be followed in the remaining cases with twice the weight.

5.12.2 VAR: Variance Reduction Control

The VAR card is used to control variance-reduction methods across several variance-reduction techniques. In particular, it allows the roulette game for weight windows and cell/energy/time importance to be turned off. Turning off roulette can be helpful for F8 tallies using variance reduction (1).

Data-card Form: VAR *keyword=value*

RR	controls rouletting game for weight windows and cell/energy/time importances. if
RR = ON	the roulette game is turned on.
RR = OFF	the roulette game is turned off.

Default: No modifications of variance reduction methods.

Use: Optional

Details:

1. For a pulse-height tally (that uses the de-branching method), Russian rouletting a particle produces zero tallies for all collections of particles that include the rouletted particle. This procedure results in no bias, but adds computational effort. In this circumstance, roulette is contraindicated.

5.12.3 Weight-window Cards

Weight windows can be either cell-based or mesh-based. Mesh-based windows eliminate the need to subdivide geometries finely enough for importance functions.

Weight windows provide an alternative means to importance (IMP values), energy splitting (ESPLT cards), and time splitting (TSPLT cards) for specifying space-, energy-, and time-dependent importance functions. The advantages of weight windows are that they

1. provide an importance function in space, time, space-energy, space-time, or space-energy-time;
2. attempt to control particle weights;
3. are more compatible with other variance-reduction features such as the exponential transform (EXT card);
4. can be applied at surface crossings, collisions, or both;
5. control the severity of splitting or Russian roulette;
6. can be turned off in selected space, time, or energy regions; and
7. can be automatically generated by the weight-window generator.

The disadvantages are that

1. weight windows are not as straightforward as importance and
2. when the source weight is changed, the weight windows may have to be renormalized (see the 7th entry on the WWP card).

The novice weight-window user is strongly advised to read §2.7.2.12.

In repeated structures, an additional difference between cell importance and weight windows exists. For cell importance (IMP card), an importance in a cell that is in a universe is interpreted as a multiplier of the importance of the filled cell [§5.12.1] and action (i.e., splitting or roulette) is taken based on the ratio of importance. The weight-window bounds are absolute bounds, not multipliers. The lower window bound in cell j and energy bin k is unaffected by the repeated structures. Mesh based windows are recommended for use with repeated structures.

A cell-based weight-window lower bound of a cell that is in a universe is interpreted as a multiplier of the weight-window lower bound of the filled cell.

5.12.3.1 WWE: Weight-window Energies (or Times)

The WWE card defines the energy (or time) intervals for which weight-window bounds will be specified on the WWN card. The minimum energy is not entered on the WWE card, but is defined to be zero. Similarly, the minimum time is $-\infty$. Whether energy or time is specified is determined by the 6th entry on the WWP card. For time-dependent weight windows, the WWT card is now recommended, but times are allowed on the WWE card to preserve backward compatibility.

Data-card Form: WWE: $\mathscr{P}$ e_1 e_2 ... e_K

$\mathscr{P}$	Particle designator.
e_k	Upper energy (or time) bound of kth window (1). Restriction: $1 \le k \le 99$
e_{k-1}	Lower energy (or time) bound of kth window.

Default: If the WWE card is omitted and weight windows are used, one energy (or time) interval is established corresponding to the energy (or time) limits of the problem.

Use: Optional. Use only with WWN card. See the WWGE card for use with the weight-window generator.

Details:

1 Parameter e_k accepts time entries to allow backward compatibility. See WWT card for time-dependent weight windows.

5.12.3.2 WWT: Weight-window Times

The WWT card defines the time intervals in shakes for which weight-window bounds will be specified on the WWN card. The minimum time is not entered on the WWT card, but is defined to be $-\infty$.

Data-card Form: WWT: $\mathscr{P}$ t_1 t_2 ... t_K

$\mathscr{P}$	Particle designator.
t_k	Upper time bound of kth window. Restriction: $1 \le k \le 99$
t_{k-1}	Lower time bound of kth window.

Default: One weight-window time interval.

Use: Optional. Use only with WWN card. See WWGT card for use with the weight-window generator.

5.12.3.3 WWN: Cell-based Weight-window Lower Bounds

The WWN card specifies the lower weight bound of the space-, time-, and energy-dependent weight windows in cells. It must be used with the WWP card and, if the weight windows are energy and/or time dependent, with the WWE and/or WWT card. For a particular particle type, both IMP and WWN cards should not be used with one exception: mesh-based weight windows require the presence of IMP cards (see the IMP card default value discussion). The weight-window game turns off the IMP card game unless the weight-window phase-space region has a lower bound of 0—then the weight cutoff game, which uses the IMP values to scale the cutoff values, is played.

In terms of the weight window, particle weight bounds are always absolute and not relative; the user must explicitly account for weight changes from any other variance reduction techniques such as source biasing. The user must specify one lower weight bound per cell per energy per time interval. There must be no holes in the specification; that is, if WWN*i* is specified, WWN*k* for $1 < k < i$ must also be specified.

Cell-card Form: WWN*i*:$\mathscr{P}$=w_i
or
Data-card Form: WWN*i*:$\mathscr{P}$ w_{i1} w_{i2} ... w_{IJ}

$\mathscr{P}$	Particle designator.
i	energy or time index.
w_i	If
	$w_i > 0$: the value is the lower weight bound in the cell for energy (or time) interval $e_{i-1} < e < e_i$, where $e_0 = 0$, or time interval $t_{i-1} < t < t_i$, where $t_0 = -\infty$. If no WWE or WWT card is included in the MCNP input file, then $i = 1$ (1).
	$w_i = 0$: then no weight-window game is played (2).
	$w_i = -1$: then any particle entering the cell is killed (equivalent to zero importance) (3).
w_{ij}	The number of entries must equal the number of cells in the problem. If
	$w_{ij} > 0$: the value is the lower weight bound in cell j for energy (or time) interval $e_{i-1} < e < e_i$, where $e_0 = 0$, or time interval $t_{i-1} < t < t_i$, where $t_0 = -\infty$. If no WWE or WWT card is included in MCNP input file, then $i = 1$ (1).
	$w_{ij} = 0$: then no weight-window game is played (2).
	$w_{ij} = -1$: then any particle entering cell j is killed (equivalent to zero importance) (3).

Default: None.

Use: Either cell importance [§5.12.1] or weight windows must be supplied to MCNP6.

Details:

1. If $w_{ij} > 0$, particles entering or colliding in the cell are split or rouletted based on the conditions setup by the WWP card parameters.
2. If $w_{ij} = 0$, the weight-window game is turned off in cell j for energy or time bin i and the weight cutoff game is turned on with a 1-for-2 roulette limit. Sometimes it is useful to specify the weight cutoffs on the CUT card as the lowest permissible weights desired in the problem. Otherwise, too many particles entering cells with $w_{ij} = 0$ may be killed by the weight cutoff. Usually, the 1-for-2 roulette limitation is sufficient to use the default weight cutoffs, but caution is needed and the problem output file should be examined carefully. The capability to turn the weight-window game off in various phase-space regions is useful when these regions cannot be characterized by a single importance function or set of weight-window bounds.
3. Caution should be exercised when one energy (or time) group out of many groups is set to -1. If the intent is to kill only low-energy particles, this may be okay; otherwise, it may be better to set all groups to -1.

5.12.3.4 Example 1

```
WWE:N e1 e2 e3
WWN1:N w11 w12 w13 w14
WWN2:N w21 w22 w23 w24
WWN3:N w31 w32 w33 w34
```

These cards define three energy intervals and the weight-window bounds for a four-cell neutron problem.

5.12.3.5 Example 2

```
WWN1:P w11 w12 w13
```

This card, without an accompanying WWE card, defines an energy- or time-independent photon weight window for a three-cell problem.

5.12.3.6 WWP: Weight-window Parameters

The WWP card contains parameters that control various aspects of the weight-window game.

Caution

The default *mwhere* treatment for weight windows has been observed to lead to potentially silent wrong answers with UM geometry or, more clearly, seemingly unrelated issues such as the code reporting negative emission energy following certain collisions. If one must use weight windows with UM, it is recommended that *mwhere* $= -1$ be set to avoid such issues, which arise because of particle-banking behavior as particles enter and exit UM pseudocells.

Data-card Form: WWP:$\mathscr{P}$ *wupn wsurvn mxspln mwhere switchn mtime wnorm etsplt wu nfmp*

$\mathscr{P}$	Particle designator.
wupn	Multiplier to define the weight window upper limit. If the particle weight goes above *wupn* times the lower weight bound, the particle will be split. Restriction: *wupn* ≥ 2 (DEFAULT: *wupn* $= 5$)
wsurvn	Multiplier to define the maximum Russian roulette survival weight within the window. If the particle survives the Russian roulette game, its weight becomes $\min(\textit{wsurvn} \times w_L, w \times \textit{mxspln})$, where w_L is the lower-weight bound and w is the original weight. Restriction: $1 <$ *wsurvn* $<$ *wupn*. DEFAULT: *wsurvn* $= 0.6 \times$ *wupn*, which defaults to three times the lower bound.
mxspln	Maximum number of integer splits. No particle will ever be split more than *mxspln*-for-one or be rouletted more harshly than one-in-*mxspln*. Restriction: *mxspln* > 1 (DEFAULT: *mxspln* $= 5$).
mwhere	Controls where to check a particle's weight (5). If
mwhere $= -1$	check the weight at collisions only.
mwhere $= 0$	check the weight at surface crossings and collisions. (DEFAULT)
mwhere $= 1$	check the weight at surface crossings only.
switchn	Controls where to get the lower weight-window bounds. If
switchn < 0	get the lower weight-window bounds from an external **WWINP** file containing either cell- or mesh-based lower weight-window bounds. Requires an IMP card (1).
switchn $= 0$	get the lower weight-window bounds from WWN*i* cards present in the MCNP input file (2). (DEFAULT)
switchn > 0	set the lower weight-window bounds equal to *switchn* divided by the cell importance from the IMP card (3).
mtime	Controls treatment of WWE card. This parameter remains to allow backward compatibility. See WWT card for time-dependent weight windows. If
mtime $= 0$	energy-dependent windows are provided on the WWE card. (DEFAULT)
mtime $= 1$	time-dependent windows are provided on the WWE card.
wnorm	Weight-window normalization factor. If *wnorm* > 0, *wnorm* is a multiplicative constant for all lower weight-window bounds on WWN*i*:$\mathscr{P}$ cards or values in the **WWINP** file. Applies to particle type $\mathscr{P}$ specified by this WWP card. (DEFAULT: *wnorm* $= 1$)
etsplt	Energy- and time-splitting control. If
etsplt $= 0$	then any entries on the ESPLT and TSPLT cards are used solely to scale the weight window. (DEFAULT)

	etsplt = 1	then any entries on the ESPLT and TSPLT cards are used to split/roulette particles as well as scale the weight windows.
wu	Limits the maximum lower weight-window bound for any particle, energy, or time to *wu*. If *wu* = 0, there is no limit (4). (DEFAULT: *wu* = 0)	
nmfp	Number of mean-free paths to travel before checking mesh-based weight windows for neutron and photon problems only (5). (DEFAULT: *nmfp* = 1)	

Default: *wupn* = 5; *wsurvn* = 3; *mxspln* = 5; *mwhere* = 0; *switchn* = 0; *mtime* = 0; *wnorm* = 1.0; *etsplt* = 0; *wu* = 0; *nmfp* = 1

Use: Weight windows are required unless importance are used.

Details:

1. If *switchn* < 0, an external **WWINP** file with either cell- or mesh-based lower weight-window bounds must exist and an IMP card is required. The **WWINP** file is a weight-window generator output file, either **WWOUT** or **WWONE**, that has been renamed in the local file space or equivalenced on the execution line using WWINP = *filename*. The different formats of the **WWINP** file will indicate to the code whether the weight windows are cell or mesh based. For mesh-based weight windows, the mesh geometry will also be read from the **WWINP** file [Appendix A].
2. If *switchn* is zero, the lower weight-window bounds must be specified with the WWN cards present in the MCNP input file.
3. An energy-independent weight window can be specified using existing importance from the IMP card and setting the fifth entry (*switchn*) on the WWP card to a positive constant C. If this option is selected, the lower weight bounds for the cells become C/I, where I is the cell importance. A suggested value for C is one in which source particles start within the weight window, such as 0.25 times the source weight. If that is not possible, the window is probably too narrow or the source should be re-specified. Having *switchn* > 0 and also having WWN*i* cards is a fatal error.
4. Unreasonably high weight-window bounds can be generated if (1) tracks that pass through a cell score only rarely or score very low, or (2) adjoint Monte Carlo is used. When weight windows with very high bounds are used in a subsequent run, the ultra-high windows will roulette nearly all particles in those phase-space regions. This results in no future estimate in these regions by the weight-window generator and potentially biased results. Use the 9th entry, *wu*, to limit the maximum lower weight-window bound. A good value of *wu* is often 1–10 times the maximum source weight.
5. Weight window processing is always performed during source emission (though source particles should start with weights consistent with the local window to avoid immediate splitting or rouletting). The *mwhere* parameter controls weight-window processing during collision or surface-crossing events that take place during particle transport. Similarly, the *nmfp* parameter controls weight-window processing following free flight for a specified number of mean-free paths when using mesh-based weight windows as long as no other event has taken place (setting *nmfp* to a high number effectively disables this behavior).

5.12.4 Stochastic Weight-window Generator Cards

The weight-window generator estimates the importance of the space-energy-time regions of phase space specified by the user. The space-energy-time weight-window lower bounds are then calculated inversely proportional to the importance.

The cell-based generator estimates the average importance of a phase-space cell. Inadequately sized (i.e., large) geometry cells often lead to inappropriate weight windows because of a large variation in the importance inside the cell. An appropriate user action is to refine the cell definitions or use the mesh-based weight window. Inadequate geometry specification for weight-window purposes also results when there are large importance differences between adjacent cells. Fortunately, the code provides information about whether the geometry specification is adequate for sampling purposes by printing to the MCNP output file a list of neighboring cells that differ by a factor of 4 or more. If geometries are inadequately subdivided by the geometry cells, mesh-based weight windows should be used.

The user is advised to become familiar with weight windows [§2.7.2.12], before trying to use the weight-window generator.

5.12.4.1 WWG: Weight-window Generation

The WWG card allows the code to generate an importance function for a user-specified tally (input parameter i_t). Because the weight-window bounds are estimated quantities, they should be well converged or else they can cause more harm than good. When well converged, they can improve efficiency substantially. Note that the number of histories per minute is often lower in the more efficient problem because more time is spent sampling important regions of the problem phase space. Moreover, in many cases, a window using the adjoint function will not be too far from optimal.

For the cell-based weight-window generator, the code creates WWE and WWNi cards that are printed, evaluated, and summarized in the MCNP output file and written to the weight-window generator output file **WWOUT**.

For the mesh-based weight-window generator, the code writes the weight-window lower bounds and a mesh description only to the **WWOUT** file. The format of the mesh-based **WWOUT** file is provided in Appendix A.

In either case, the generated weight-window information can be easily used in subsequent runs using *switchn* < 0 on the WWP card. For many problems, the weight-window generator results are superior to anything an experienced user can guess and then input on an IMP card. To generate energy- and/or time-dependent weight windows, use the WWGE and/or WWGT cards.

Data-card Form: WWG i_t i_c w_g J J J J i_e

i_t	Problem tally number (n of the Fn card). The particular tally bin for which the weight-window generator is optimized is defined by the TFn card (1). No default value; must be specified if the WWG card is present.
i_c	Invokes cell- or mesh-based weight-window generator. If
$i_c > 0$	then invoke the cell-based weight-window generator with i_c as the reference cell (typically a source cell).
$i_c = 0$	then invoke the mesh-based weight-window generator. The MESH card is then required (2). (DEFAULT: $i_c = 0$)
w_g	Value of the generated lower weight-window bound for cell i_c or for the reference mesh. See MESH card (3). If $w_g = 0$, then the lower bound will be half the average source weight. (DEFAULT: $w_g = 0$)
J J J J	Unused placeholders.
i_e	Toggles energy- or time-dependent weight windows. This parameter remains to allow backward compatibility. See WWGT card for time-dependent weight windows. If

$i_e = 0$	then interpret WWGE card entries as energy bins. (DEFAULT: $i_e = 0$)
$i_e = 1$	then interpret WWGE card entries as time bins.

Default: No weight-window values are generated.

Use: Optional.

Details:

1. Weight-window generation relies on scores being made by the primary source particle to or near (so secondary particles can score) the reference tally, i_t. The primary source particle is typically specified by PAR on the SDEF card. If PAR is a distribution or unspecified, then the primary source particle is the particle with the lowest number on the MODE card.
2. For mesh-based weight windows, a reference point (REF) is required instead of a cell number. See the MESH card.
3. The value w_g of the lower weight-window bound for reference cell i_c or reference mesh location is chosen so that the source weight will start within the weight window, when possible. The reference cell i_c is often chosen as the source cell and the reference mesh location is often chosen in or near the source cell.

5.12.4.2 WWGE: Weight-window Generation Energies (or Times)

If the WWGE card is present, energy- (or time-) dependent weight windows are generated and written to the **WWOUT** file and, for cell-based weight windows, to the MCNP output file. In addition, single-group energy- (or time-) independent weight windows are written to a separate output file, **WWONE**. Energy- (or time-) independent weight windows are sometimes useful for trouble-shooting the energy- (or time-) dependent weight windows on the **WWOUT** file. The **WWONE** file format is the same as that of the **WWOUT** file [Appendix A]. The selection of energy- or time-dependent weight windows is made with the 8th entry on the WWG card.

Data-card Form: WWGE:$\mathscr{P}$ e_1 e_2 ... e_K

$\mathscr{P}$	Particle designator.
e_k	Upper energy (or time (1)) bound for weight-window group to be generated, $e_{k+1} > e_k$. Restriction: $k \leq 15$.

Default: If the WWGE card is omitted and the weight window is used, a single energy (time) interval will be established corresponding to the energy (time) limits of the problem being run. If the card is present but has no entries, ten energy (time) bins will be generated with energies (times) of $e_k = 10^{k-8}$ MeV (or shakes), for $i = 1, 2, \ldots, 10$. Both the single energy (time) and the energy- (time-) dependent windows are generated.

Use: Optional.

Details:

1. Although the WWGE card will accept time bins so to be compatible with previous versions of the MCNP code, it is recommended that the user use the WWGT card for time-dependent weight-window generation.

5.12.4.3 WWGT: Weight-window Generation Times

If the WWGT card is present, time-dependent weight windows are generated and written to the **WWOUT** file and, for cell-based weight windows, to the MCNP output file. In addition, single-group time-independent weight windows are written to a separate output file, **WWONE**. Time-independent weight windows are sometimes useful for trouble-shooting the time-dependent weight windows on the **WWOUT** file. The **WWONE** file format is the same as that of the **WWOUT** file [Appendix A].

Data-card Form: WWGT:$\mathscr{P}$ t_1 t_2 ... t_K

$\mathscr{P}$	Particle designator.
t_k	Upper time bound for weight-window group to be generated, $t_{k+1} > t_k$. Units are shakes. Restriction: $k \leq 15$.

Default: If the WWGT card is omitted and the weight window is used, a single time interval will be established corresponding to the time limits of the problem being run. If the card is present but has no entries, ten time bins will be generated with times of $t_k = 10^{k-8}$ shakes, for $i = 1, 2, \ldots, 10$. Both the single time and the time-dependent windows are generated.

Use: Optional.

5.12.4.4 Example 1

```
WWG       111   45  0.25
WWGE:p    1  100
WWGT:p    1  100  1.e20
```

The cell-based windows generated from the above cards would look like:

```
WWP:p 5 3 5
WWE:p 1 100
WWT:p 1 100 1.e20
WWN1:p w1 w2 w3 ... $ energy 1 time 1
WWN2:p w1 w2 w3 ... $ energy 2 time 1
WWN3:p w1 w2 w3 ... $ energy 1 time 2
WWN4:p w1 w2 w3 ... $ energy 2 time 2
WWN5:p w1 w2 w3 ... $ energy 1 time 3
WWN6:p w1 w2 w3 ... $ energy 2 time 3
```

This example generates a 2-energy group, 3-time group weight window. In particular, the WWG card would generate weight windows to optimize tally 111. The lowest weight-window bound in any energy-time bin group in cell 45 (the reference cell) would be 0.25. The WWGE and WWGT cards would generate two energy bins and three time bins for photons.

5.12.4.5 MESH: Superimposed Importance Mesh for Mesh-Based Weight-Window Generator

Data-card Form: `MESH keyword=value(s)...`

`GEOM`	Controls mesh geometry type. (DEFAULT: `GEOM` = `XYZ`). If
	`GEOM` = `XYZ` or `GEOM` = `REC`
	mesh geometry is Cartesian.
	`GEOM` = `RZT` or `GEOM` = `CYL`
	mesh geometry is cylindrical.
	`GEOM` = `RPT` or `GEOM` = `SPH`
	mesh geometry is spherical.

`REF`	x, y, and z coordinates of the reference point; used to create the normalization constant for the mesh-based weight-window generator. (DEFAULT: none) Restriction: Required.
`ORIGIN`	x, y, and z coordinates, in MCNP6 cell geometry, of the origin (bottom, left, rear for rectangular; bottom center for cylindrical; center for spherical) of the superimposed mesh. (DEFAULT: `ORIGIN` = 0., 0., 0.,)
`AXS`	Vector giving the direction of the (polar) axis of the cylindrical (1) or spherical mesh (DEFAULT: `AXS` = 0., 0., 1.,)
`VEC`	Vector defining, in conjunction with `AXS`, the plane for $\theta = 0$. For spherical geometry, `VEC` must be orthogonal to ϕ. (DEFAULT: `VEC` = 1., 0., 0.,)
`IMESH`	Locations of the coarse meshes in the x direction for rectangular geometry or in the r direction for cylindrical or spherical geometry (2), (3), (4). (DEFAULT: none)
`IINTS`	Number of fine meshes within corresponding coarse meshes in the x direction for rectangular geometry or in the r direction for cylindrical or spherical geometry (6), (7). (DEFAULT: `IINTS` = 1 fine mesh in each coarse mesh)
`JMESH`	Locations of the coarse meshes in the y direction for rectangular geometry, in the z direction for cylindrical geometry, or the ϕ polar angle bounds for spherical geometry (2), (3), (4), (5). (DEFAULT: none)
`JINTS`	Number of fine meshes within corresponding coarse meshes in the y direction for rectangular geometry, in the z direction for cylindrical geometry, or in the ϕ direction for spherical geometry (6), (7). (DEFAULT: `JINTS` = 1 fine mesh in each coarse mesh)
`KMESH`	Locations of the coarse meshes in the z direction for rectangular geometry or in the θ direction for cylindrical or spherical geometry (2), (3), (4), (5). (DEFAULT: none)
`KINTS`	Number of fine meshes within corresponding coarse meshes in the z direction for rectangular geometry or in the θ direction for cylindrical or spherical geometry (6), (7). (DEFAULT: `KINTS` = 1fine mesh in each coarse mesh)

Use: Required to generate mesh-based weight windows; not required to use without weight-window generation. This card is also used to generate a structured discrete-ordinates-style geometry file.

Details:

1. For a cylindrical mesh, the AXS and VEC vectors need not be orthogonal but they must not be parallel; the one half-plane that contains them and the ORIGIN point will define $\theta = 0$. The AXS vector will remain fixed. The length of the AXS or VEC vectors must not be zero.
2. For both the cylindrical and spherical meshes, the lower radial and angular mesh bounds (r, ϕ, θ) are implicitly zero.
3. The location of the nth coarse mesh in the u direction (ru_n in what follows) is given in terms of the most positive surface in the u direction. For a rectangular mesh, the coarse mesh locations (rx_n, ry_n, rz_n) are given as planes perpendicular to the x , y , and z axes, respectively, in the MCNP6 cell coordinate system; thus, the ORIGIN point (x_0, y_0, z_0) is the most negative point of the mesh. For a cylindrical mesh, ORIGIN (r_0, z_0, θ_0) corresponds to the bottom center point and, for a spherical mesh, ORIGIN (r_0, ϕ_0, θ_0) corresponds the sphere center. The coarse mesh locations must increase monotonically.
4. In the XYZ (REC) mesh, the IMESH, JMESH, and KMESH are the actual (x, y, z) coordinates. In the RZT (CYL) mesh, IMESH (radius) and JMESH (height) are relative to ORIGIN and KMESH (θ) is relative to VEC. In the RPT (SPH) mesh, IMESH (radius) is relative to ORIGIN, JMESH (ϕ) is relative to AXS, and KMESH (θ) is relative to VEC.
5. Polar and azimuthal angles may be specified in revolutions ($0 \le \phi \le 0.5$ and $0 \le \theta \le 1$), radians, or degrees. MCNP6 recognizes the appropriate units by looking for 0.5, 3.14, or 180 for the last spherical geometry JMESH entry and for 1, 6.28, or 360 for the last spherical or cylindrical KMESH entry.
6. The fine meshes are evenly distributed within the nth coarse mesh in the u direction. The mesh in which the reference point lies becomes the reference mesh cell for the mesh-based weight-window generator; this reference mesh cell is analogous to the reference cell used by the cell-based weight-window generator. The mesh cell containing the REF point will have its (over energy) weight-window lower bound equal to the third entry on the WWG card.
7. The code uses a default value of 1 fine mesh per coarse mesh if IINTS, JINTS, or KINTS keywords are omitted. If IINTS, JINTS, or KINTS keywords are present, the number of entries must match the number of entries on the IMESH, JMESH, and KMESH keywords, respectively. Entries on the IINTS, JINTS, and KINTS keywords must be greater than zero.

5.12.4.5.1 Using an Existing Superimposed Mesh

A second method of providing a superimposed mesh is to use one that already exists, written either to the **WWOUT** file or to the **WWONE** file. To implement this method, use the WWG card with $i_c = 0$ in conjunction with the MESH card where the only keyword is REF. The reference point must be within the superimposed mesh and must be provided because there is no reference point in either **WWOUT** or **WWONE**. If the mesh-based weight-window generator is invoked by this method, MCNP6 expects to read a file called **WWINP**. The **WWINP** file is a weight-window generator output file, either **WWOUT** or **WWONE**, that has been renamed in the local file space or equivalenced on the execution line using **WWINP=*filename*** [Appendix A].

It is not necessary to use mesh-based weight windows from the **WWINP** file in order to use the mesh from that file. Furthermore, previously generated mesh-based weight windows can be used (WWP card with *switchn* < 0 and **WWINP** file in mesh format) while the mesh-based weight-window generator is simultaneously generating weight windows for a different mesh (input on the MESH card). However, it is not possible to read mesh-based weight windows from one file and a weight-window generation mesh from a different file.

5.12.4.5.2 Hints and Guidelines Regarding Superimposed Mesh Creation

The superimposed mesh should fully cover the problem geometry; i.e., the outer boundaries of the mesh should lie outside the outer boundaries of the geometry, rather than being coincident with them. This requirement guarantees that particles remain within the weight-window mesh. A line or surface source should not be made coincident with a mesh surface. A point source should never be coincident with the intersection of mesh surfaces. In particular, a line or point source should never lie on the axis of a cylindrical mesh. These guidelines also apply to the WWG reference point specified using the REF keyword.

If a particle does escape the weight-window mesh, the code prints a warning message giving the coordinate direction and surface number (in that direction) from which the particle escaped. The code prints the total number of particles escaping the mesh (if any) after the tally fluctuation charts in the standard output file. If a track starts outside the mesh, the code prints a warning message giving the coordinate direction that was missed and which side of the mesh the particle started on. The code prints the total number of particles starting outside the mesh (if any) after the tally fluctuation charts in the standard output file.

Specifying $i_c = 0$ on the WWG card with no MESH card is a fatal error. If AXS or VEC keywords are present and the mesh is rectangular, a warning message is printed and the keyword is ignored. If there are fatal errors and the FATAL option is on, weight-window generation is disabled.

5.12.4.5.3 Example 1

```
MESH   GEOM=CYL  REF=1e-6 1e-7 0  ORIGIN=1  2  3
       IMESH   2.55 66.34
       IINTS   2 15  $ 2 fine bins from 0 to 2.55, 15 from 2.55 to 66.34
       JMESH   33.1  42.1  53.4  139.7
       JINTS   6     3     4     13
       KMESH   0.5     1
       KINTS   5     5
```

5.12.4.5.4 Example 2

```
MESH   GEOM=REC  REF=1e-6 1e-7 0  ORIGIN=-66.34 -38.11 -60
       IMESH   -16.5  3.8  53.66
       IINTS    10   3    8    $ 10 fine bins from -66.34 to -16.5, etc.
```

5.12.4.5.5 Example 3

```
MESH   GEOM sph  ORIGIN 7 -9 -12  REF -23 39 -10  AXS 0.4 -0.5 0.2
       VEC   0.1 -0.2 -0.7
       IMESH     60.                  IINTS  3
       JMESH    0.1   0.35   0.5      JINTS  1 1 1
       KMESH    0.2   0.85   1        KINTS  1 1 1
```

In this example a spherical mesh is located at ORIGIN = 7 − 9 − 12. The reference location in the (x, y, z) coordinate system of the problem is at REF = −23 39 −10. The weight-window generator lower weight-window bound will be W for whatever mesh cell contains this location, where W is half the source weight by default or whatever is the 3rd entry on the WWG weight-window generator card. The polar (ϕ) axis of the spherical mesh (as in latitude on the globe) is AXS = 0.4 −0.5 0.2, which MCNP6 will normalize to a unit vector. The azimuthal planes (as in longitude on a globe, or cylindrical mesh theta bins) are measured relative to the azimuthal vector, theta (θ), VEC = 0.1 −0.2 −0.7. VEC will also be renormalized by MCNP6 and must be orthogonal to ϕ. The radial mesh bins have three interpolates between 0 and 60—that is, the mesh bounds are at 0, 20, 30, and 60 cm. The polar angles (ϕ) are at 0.1, 0.35, and 0.5 revolutions from the AXS vector. The azimuthal angles (θ) are at 0.2, 0.85, and 1 revolutions from the VEC vector. Note that $0 \le \phi \le 0.5$ and $0 \le \theta \le 1$ are always required.

Examples that show how to plot superimposed weight-window meshes are given in §6.4.6.

5.12.5 ESPLT: Energy Splitting and Roulette

The ESPLT card allows problem-wide splitting and Russian roulette of particles in energy, similar to how the IMP card allows splitting and Russian roulette as a function of geometry for continuous-energy calculations. The changes to a particle's weight caused by the ESPLT card will create compensating weight adjustments to the weight cutoff and weight-window values.

Data-card Form: ESPLT:$\mathscr{P}$ r_1 e_1 ... r_K e_K

$\mathscr{P}$	Particle designator.
r_k	Provides splitting/roulette ratios at each energy boundary, e_k, for decreasing energy. The meanings of the r_k differ depending on whether or not there is a weight window present for the particle type $\mathscr{P}$. These splitting/roulette ratios are internally converted in the code to an absolute importance function with an $r_0 = 1$ inserted to set the importance to unity for energies greater than the maximum of the e_k. Restriction: $1 \le k \le 20$

When weight windows are not used, and if the energy of a particle of type $\mathscr{P}$ falls below e_k (decreasing energy), then when

$r_k > 1$	r_k is the number of tracks into which a particle will be split.
$0 < r_k < 1$	r_k is the probability of Russian roulette.
$r_k = 1$	there is no action.

If the energy of a particle of type $\mathscr{P}$ increases in energy above e_k, then when

$1/r_k > 1$	$1/r_k$ is the number of tracks into which a particle will be split.
$0 < 1/r_k < 1$	$1/r_k$ is the probability of Russian roulette.

Exception: if the first $r_1 < 0$, then no game is played on energy increases.

When weight windows are specified, then the energy splitting is accomplished solely with the weight windows (1). The r_k in this case are energy importance modifications to the weight window. If

- the energy of a particle of type $\mathscr{P}$ falls below e_k, then the existing weight windows will be adjusted by dividing the windows by r_k.

	• the energy of a particle of type $\mathscr{P}$ increases above e_k, then the weight windows are multiplied by r_k. • more than one energy boundary is crossed, the windows are adjusted by the product of the r_k values.
e_k	Energy (MeV) at which particles are to undergo splitting or Russian roulette. Values must be monotonically increasing. Restriction: $1 \leq k \leq 20$

Default: Energy splitting will not occur for a given particle type unless this card is defined.

⚠ Caution

The ESPLT card was originally designed with the intent that it be used with a time-dependent weight window. The ESPLT card is not recommended for an energy-dependent weight window as these two cards may interfere with one another (see the discussion in the table above). However, the code does not prevent the user from invoking both at the same time. Instead of a single-range weight window and an ESPLT card, consider using an energy-dependent weight window.

Details:

1. If the eighth entry on the WWP card is 1 (0 is the default), then in addition to the weight-window adjustment, the particle will be explicitly split or rouletted upon crossing e_i, just as is the case without a weight window. It is anticipated that the default will be appropriate for almost all problems.

5.12.5.1 Additional Information Regarding ESPLT

The entries on the ESPLT card consist of pairs of energy-importance ratio parameters, r_k and e_k, with a maximum of twenty pairs allowed. A warning message is issued if the e_k are not monotonically increasing. The value of r_k can be non-integer and also can be between 0 and 1. For an energy decrease below an e_k with an associated r_k greater than 1, particle splitting will occur. For a value of r_k between 0 and 1, r_k becomes the survival probability in the Russian roulette game. For an energy increase above an e_k with an associated $1/r_k$ greater than 1, particle splitting will occur. For a value of $1/r_k$ between 0 and 1, $1/r_k$ becomes the survival probability in the Russian roulette game. If a particle's energy becomes less than e_k, the specified splitting or roulette is sampled. If more than one energy boundary is passed during a particle trajectory, the product of the r_k values is used to determine the outcome.

If the particle's energy falls below e_k, the specified splitting or roulette always occurs. If the particle's energy increases above e_k, the inverse game is normally played (unless r_1 has been specified as less than zero). For example, suppose roulette is specified at 0.1 MeV with a survival probability of 0.5; if a particle's energy increases above 0.1 MeV, then it is split 2-for-1.

A neutron's energy may increase by fission or from thermal up-scattering. There are cases when it may not be desirable to have the splitting or roulette game played on energy increases (particularly in a fission-dominated problem). If $r_1 < 0$, then splitting or roulette will be played only for energy decreases and not for energy increases.

5.12.5.2 Example 1

```
ESPLT:N    2  0.1   2  0.01   0.25  0.001
```

This example specifies a 2-for-1 split when the neutron energy falls below 0.1 MeV, another 2-for-1 split when the energy falls below 0.01 MeV, and Russian roulette when the energy falls below 0.001 MeV with a 25% chance of surviving. Thus, a neutron that enters a collision at 0.5 MeV and exits at 0.005 MeV will be split 4-to-1.

5.12.5.3 Example 2

```
ESPLT:N    2  0.1   2  0.01   0.25  0.001
WWP:N      5  3  5  0  0  0  J  J
```

This example divides the weight windows by 2 when the energy falls below 0.1 MeV, divides by 2 again when the energy falls below 0.01 MeV, and divides by 0.25 when the energy falls below 0.001 MeV.

5.12.5.4 Example 3

```
ESPLT:N    2  0.1   2  0.01   0.25  0.001
WWP:N      5  3  5  0  0  0  J  1
```

This example is similar to §5.12.5.3 except that the eighth entry on the WWP card (etsplt) is set to 1. Consequently, in addition to the weight-window adjustment, the particle will be explicitly split or rouletted upon crossing e_k. For this example, the weight windows will be divided by 2 when the energy falls below 0.1 MeV, divided by 2 again when the energy falls below 0.01 MeV, and divided by 0.25 when the energy falls below 0.001 MeV. In addition, a 2-for-1 split will occur when the neutron energy falls below 0.1 MeV, another 2-for-1 split will happen when the neutron energy falls below 0.01 MeV, and Russian roulette with a survival probability of 0.25 will be played when the neutron energy falls below 0.001 MeV.

5.12.6 TSPLT: Time Splitting and Roulette

The TSPLT card allows problem-wide splitting and Russian roulette of particles in time, like the IMP card allows splitting and Russian roulette as a function of geometry. The TSPLT card can be used in all problems except multigroup problems. The changes to a particle's weight caused by the TSPLT card will create compensating weight adjustments to the weight cutoff and weight-window values.

Data-card Form: TSPLT:$\mathscr{P}$ r_1 t_1 ... r_K t_K

$\mathscr{P}$	Particle designator (1).
r_k	Provides splitting/roulette ratios at each time boundary, t_k, for increasing time. These splitting/roulette ratios are internally converted in the code to an absolute importance function with an $r_0 = 1$ inserted to set the

	importance to unity for times less than the minimum of the r_k. Restriction: $1 \leq k \leq 20$ When weight windows are not used, and if $r_k > 1$ — r_k is the number of tracks into which a particle will be split. $0 < r_k < 1$ — r_k is the probability of Russian roulette. $r_k = 1$ — there is no action. When weight windows are specified, then the time splitting/roulette is accomplished solely with the weight windows (2). The r_k in this case are time importance modifications to the weight window. If • the particle crosses t_k, the existing weight windows will be adjusted by dividing the windows by r_k. • more than one time boundary is crossed, the windows are divided by the product of the r_k values.
t_k	Time (shakes) at which particles are to undergo splitting or Russian roulette. Values must be monotonically increasing. Restriction: $1 \leq k \leq 20$

Default: Omission of this card means that time splitting will not take place for those particles for which the card is omitted.

Use: Optional. Cannot be used in multigroup calculations.

Caution

The TSPLT card is intended to be used with an energy-dependent weight window. The TSPLT card is not recommended for a time-dependent weight window as these two cards may interfere with one another. However, the code does not prevent the user from invoking both at the same time. Instead of a single-range weight window and a TSPLT card, consider using a time-dependent weight window.

Details:

1. Normally in a coupled mode problem (e.g., MODE N P), if particle type $\mathscr{P}$ is important late in time, then all particles producing particle type $\mathscr{P}$ will also be important late in time. For these reasons, it is suggested that the user have a TSPLT card for each relevant particle type. Thus in a MODE N P problem, if a TSPLT:P card is specified then a TSPLT:N card would normally be specified as well.
2. If the eighth entry on the WWP card is 1 (0 is the default), then in addition to the weight-window adjustment, the particle will be explicitly time-split or rouletted upon crossing t_k, just as is the case without a weight window. It is anticipated that the default will be appropriate for almost all problems.

5.12.6.1 Additional Information Regarding TSPLT

The entries on the TSPLT card consist of pairs of time-importance ratio parameters, r_k and t_k, with a maximum of twenty pairs allowed. A warning message is issued if the t_k are not monotonically increasing.

The value of r_k can be non-integer and also can be between 0 and 1. For an r_k greater than 1, particle splitting will occur. For a value of r_k between 0 and 1, r_k becomes the survival probability in the Russian roulette game. If a particle's time becomes greater than t_k, the specified splitting or roulette is sampled. If more than one time boundary is passed during a particle trajectory, the product of the r_k values is used to determine the outcome. The t_k are in units of shakes.

5.12.6.2 Example 1

```
TSPLT:N 2 100 2 1000 0.2 10000
```

This example specifies a 2-for-1 split when the neutron time exceeds 100 shakes, another 2-for-1 split when the time exceeds 1000 shakes, and Russian roulette with a survival probability of 0.2 when the time exceeds 10000 shakes. A neutron that crosses both 1000 and 10000 shakes will have a survival probability of 0.4.

5.12.6.3 Example 2

```
TSPLT:N 2 100 2 1000 0.2 10000
WWP:N 5 3 5 0 0 0 J J
```

This example divides the weight windows by 2 when the neutron time exceeds 100 shakes, divides by 2 again when the time exceeds 1000 shakes, and divides by 0.2 when the time exceeds 10000 shakes. Thus the weight window will be divided by a factor of 4 for a particle whose time at the start of the transport step was 90 shakes and whose time at the end of the transport step was 1010 shakes.

5.12.6.4 Example 3

```
TSPLT:N 2 100 2 1000 0.2 10000
WWP:N 5 3 5 0 0 0 J 1
```

This example is similar to §5.12.6.3 except that the eighth entry on the WWP card (`etsplt`) is set to 1. Consequently, in addition to the weight-window adjustment, the particle will be explicitly split or rouletted when it exceeds t_k. For this example the weight windows will be divided by 2 when the neutron time exceeds 100 shakes, divided by 2 again when the time exceeds 1000 shakes, and divided by 0.2 when the time exceeds 10000 shakes. In addition, this example specifies a 2-for-1 split when the neutron time exceeds 100 shakes, another 2-for-1 split when the time exceeds 1000 shakes, and a Russian roulette survival probability of 0.2 when the time exceeds 10000 shakes.

5.12.7 EXT: Exponential Transform

The exponential transform method [§2.7.2.13] stretches the path length between collisions in a preferred direction by adjusting the total cross section.

Cell-Card Form: EXT:$\mathscr{P}$ a
or
Data-card Form: EXT:$\mathscr{P}$ a_1 a_2 ...

$\mathscr{P}$	Particle designator.
a	Each entry a is of the form a=QVm, where Q is the stretching parameter and Vm defines the stretching direction for the cell [Table 5.25].
ak	Each entry ak is of the form ak=QVm, where Q is the stretching parameter and Vm defines the stretching direction for the cell k [Table 5.25]. The number of entries need not equal the number of cells in the problem, but a warning message is printed if they are not equal.

Default: No transform, ak= 0.

Use: Optional. Use cautiously. Weight windows are strongly recommended. A warning message is given if weight windows are not present when the exponential transform is used. The exponential transform should not be used in the same cell as forced collisions or without good weight control. The transform works best when the particle flux has an exponential attenuation, such as in problems with highly absorbing media.

5.12.7.1 The Stretching Parameter

The exponential transform method stretches the path length between collisions in a preferred direction by adjusting the total cross section as follows:

$$\Sigma_t^* = \Sigma_t(1 - p\mu), \tag{5.49}$$

where

Σ_t^*	is the artificially adjusted total cross section,
Σ_t	is the true total cross section,
p	is the stretching parameter, and
μ	is the cosine of the angle between the particle direction and the stretching direction.

The stretching parameter, p, can be specified by the stretching entry, Q, in three ways. If

$Q = 0$	$p = 0$ and the exponential transform is not used.
$Q = p$	$0 < p < 1$ and a constant stretching parameter is specified.
$Q = S$	$p = \Sigma_c/\Sigma_t$ where Σ_c is the capture cross section (as defined by nuclear engineers).

Letting $p = \Sigma_c/\Sigma_t$ can be used for implicit capture along a flight path, as described in §2.4.3.4.3 and §2.7.2.14.

The stretching direction is defined by the Vm part of each ak entry on the EXT card with three available options:

Table 5.25: Exponential Transform Stretching Parameter

Cell	*ak*	*Q*	*Vm*	Stretching Parameter	Stretching Direction
3	0.7V2	0.7	V2	$p = 0.7$	Toward point $(1, 1, 1)$
4	S	S		$p = \Sigma_c/\Sigma_t$	Particle direction
5	-SV2	S	-V2	$p = \Sigma_c/\Sigma_t$	Away from point $(1, 1, 1)$
6	-0.6V9	0.6	-V9	$p = 0.6$	Away from origin
8	0.5V9	0.5	V9	$p = 0.5$	Toward origin
9	SZ	S	Z	$p = \Sigma_c/\Sigma_t$	Along $+z$ axis
10	-0.4X	0.4	-X	$p = 0.4$	Along $-x$ axis

1. If the *Vm* part of the *ak* entry is omitted (i.e., *ak* $= 0$ for a given cell), then the stretching is in the particle direction ($\mu = 1$). This is not recommended unless implicit capture along a flight path is desired, in which case *ak* $= S$ should be used so that the distance to scatter rather than the distance to collision is sampled.

2. The stretching direction may be specified as *Vm*, where *m* is a unique integer that is associated with the vector entry provided on the VECT card. The stretching direction is defined as the line from the collision point to the point (x_m, y_m, z_m), where (x_m, y_m, z_m) is provided on the VECT card. The direction cosine μ is now the cosine of the angle between the particle direction and the line drawn from the collision point to point (x_m, y_m, z_m). The sign of *ak* governs whether stretching is toward or away from (x_m, y_m, z_m).

3. The stretching direction may also be specified as *Vm* = X or Y or Z, so the direction cosine μ is the cosine of the angle between the particle direction and the x , y, or z axis, respectively. The sign of *ak* governs whether stretching is toward or away from the x , y, or z axis.

5.12.7.2 Example 1

```
EXT:N    0  0  0.7V2  S  -SV2  -0.6V9  0  0.5V9  SZ  -0.4X
VECT     V9  0  0  0    V2  1  1  1
```

The 10 entries are for the 10 cells in this problem. Path length stretching is not turned on for photons or for cells 1, 2, and 7. Table 5.25 is a summary of path length stretching in the other cells.

5.12.8 VECT: Vector Input

The entries on the VECT card are quadruplets that may define any number of vectors for either the exponential transform or user patches. See the EXT card for a usage example.

Data-card Form: VECT Vm x_m y_m z_m ... Vn x_n y_n z_n

m,n	Any number to uniquely identify vectors Vm, Vn, ...
x_m y_m z_m	Coordinate triplets to define vector Vm.

Default: None.

Use: Optional.

5.12.9 FCL: Forced Collision

The FCL card controls the forcing of neutron or photon collisions in each cell. This is particularly useful for generating contributions to point detectors or DXTRAN spheres. The weight-window game at surfaces is not played when entering forced-collision cells.

Because the forced-collision variance reduction method can produce several low-weight particles, the weight cutoff game is turned on by default when using pulse-height tallies and forced collisions together. Any of the default settings can be overridden by explicitly setting the weight cutoffs on the CUT card.

Cell-card Form: FCL:$\mathscr{P}$=x
or
Data-card Form: FCL:$\mathscr{P}$ x_1 x_2 ...

$\mathscr{P}$	Particle designator. Restriction: Only neutrons (N) and photons (P) are permitted.
x	Forced-collision control for cell (1), (2), (3). Restriction: $-1 \le x \le 1$. If
	$x > 0$: forced collision applies to particles entering the cell and to those surviving weight cutoff/weight-window games in the cell (4).
	$x < 0$: forced collision applies only to particles entering the cell (5).
	$x = 0$: no forced collision in the cell. (DEFAULT)
x_k	Forced-collision control for cell k (1), (2), (3). Restriction: $-1 \le x \le 1$. If
	$x_k > 0$: forced collision applies to particles entering the cell k and to those surviving weight cutoff/weight-window games in the cell (4).
	$x_k < 0$: forced collision applies only to particles entering the cell k (5).
	$x_k = 0$: no forced collision in the cell k. (DEFAULT)
	The number of entries need not equal the number of cells in the problem, but a warning message is printed if they are not equal.

Default: $x_k = 0$, no forced collisions.

Use: Optional. Exercise caution.

Details:

(1) If $x_k \neq 0$, all particles entering cell k are split into collided and un-collided parts with the appropriate weight adjustment. If $|x_k| < 0$, Russian roulette is played on the collided parts with survival probability $|x_k|$ to keep the number of collided histories from getting too large. Fractional x_k entries, rather than values of -1 or 1, are recommended if a number of forced-collision cells are adjacent to each other.

(2) When cell-based weight-window bounds bracket the typical weight entering the cell, choose $x_k > 0$. When cell-based weight-window bounds bracket the weight typical of forced-collision particles, choose $x_k < 0$. For mesh-based windows, $x_k > 0$ usually is recommended. When using importance, $x_k > 0$ because $x_k < 0$ turns off the weight cutoff game.

(3) Let $x_k = 1$ or -1 unless a number of forced collision cells are adjacent to each other or the number of forced collision particles produced is higher than desired. Then fractional values are usually needed.

(4) If $x_k > 0$, the forced collision process applies both to particles entering cell k and to the collided particles surviving the weight cutoff or weight-window games. Particles will continue to be split into un-collided and (with probability $|x_k|$) collided parts until killed by either weight cutoff or weight windows.

(5) If $x_k < 0$, the forced collision process applies only to particles entering the cell. After the forced collision, the weight cutoff is ignored and all subsequent collisions are handled in the usual analog manner. Weight windows are not ignored and are applied after contributions are made to detectors and DXTRAN spheres.

5.12.10 DXT: DXTRAN Sphere

DXTRAN (which stands for deterministic transport) spheres are used to improve the particle sampling in a given region of phase-space, a type of angle biasing, or, conversely, to block high-weight particles from reaching a given region. Primarily, the DXT card specifies the spheres needed to define a spherical phase-space region and the special weight-cutoff game that applies inside the spheres, depending upon the presence or absence of other variance reduction games specified in the problem. See §2.7.2.18 for more details about this method.

Data-card Form: DXT:$\mathscr{P}$ *x1 y1 z1 ri1 ro1 x2 y2 z2 ri2 ro2 ... dwc1 dwc2 dpwt*

$\mathscr{P}$	Particle designator. Restriction: Only neutrons (n) and photons (p) are permitted.
xk yk zk	Coordinates of the point at the center of the kth pair of spheres ((1), (2)). Restriction: $k \leq 10$.
rik	Radius of the kth inner sphere in centimeters. The inner sphere is only used to bias placement of the DXTRAN particles on the outer sphere by modifying the probability density function into a two-step histogram [§2.7.2.18]. All particles start on the outer sphere ((3)). Restriction: $k \leq 10$.
rok	Radius of the kth outer sphere in centimeters. Restriction: $k \leq 10$.
dwc1	Upper weight cutoff in the spheres for the DXTRAN weight-cutoff game inside the sphere. (DEFAULT: *dwc1* $= 0$)
dwc2	Lower weight cutoff in the spheres for the DXTRAN weight-cutoff game inside the sphere. (DEFAULT: *dwc2* $= 0$)
dpwt	Minimum photon weight. Entered on DXT:n card only ((4)). (DEFAULT: *dpwt* $= 0$)

Defaults: Zero for *dwc1*, *dwc2*, and *dpwt*. No defaults for locations or radii.

Use: Optional. Consider using DXC:$\mathscr{P}$ or DD cards when using DXT.

Details:

(1) There can be up to 10 sets of positions and radii per particle type. There is only one set of *dwc1* and *dwc2* entries for each particle type. The *dwc* pair is entered after conclusion of the other data and (with DXT:n) before the one value of *dpwt*. The weight cutoffs apply to DXTRAN particle tracks inside the outer radii

and have default values of zero. The DXTRAN photon weight cutoffs have no effect unless the simple physics is used, with one exception: upon leaving the sphere, track weights (regardless of what physics is used) are checked against the cutoffs of the CUT:p card. The DXTRAN weight cutoffs *dwc1* and *dwc2* are ignored when mesh-based weight windows are used, but are active for cell-based weight windows because the weight-window game is turned off inside the spheres.

2. DXTRAN spheres can be nested inside one another [331]. The allowed nesting is reasonably general: more than one DXTRAN sphere may be nested inside a larger DXTRAN sphere and the centers of the nested DXTRAN spheres need not be concentric. Also, the spherical surfaces must not intersect. This nesting mitigates weight fluctuation problems as the particles approach the region(s) of interest.

3. When the DXTRAN method is used as a means to produce a higher particle population near a tally, the inner radius *ri* should be at least as large as the tally region. The purpose of the inner sphere is for biasing placement of DXTRAN particles on the outer sphere; there is no problem making the two radii the same.

4. The minimum photon weight limit *dpwt* on the DXT:n card parallels almost exactly the minimum photon weight entries on the PWT card. One slight difference is that in Russian roulette during photon production inside DXTRAN spheres, the factor for relating current cell importance to source cell importance is not applied. Thus, the user must have some knowledge of the weight distribution of the DXTRAN particles (from a short calculation with the DD card, for example) inside the DXTRAN sphere, so the lower weight limit for photon production may be specified intelligently. As in the case of the PWT entries, a negative entry will make the minimum photon weight relative to the source particle starting weight. The default value is zero, which means photon production will occur at each neutron DXTRAN particle collision in a material with non-zero photon production cross section inside the DXTRAN sphere.

5.12.10.1 Additional Information Regarding DXTRAN Spheres

One use of DXTRAN is to improve the particle sample in the vicinity of a tally. It should not be misconstrued as a tally itself, such as a point detector; it is used in conjunction with tallies as a variance reduction technique. DXTRAN spheres must not overlap. The spheres should normally cover the tally region if possible. Specifying a tally cell or surface partly inside and partly outside a DXTRAN sphere usually will make the mean of the tally erratic and the variance huge.

The technique is most effective when the geometry inside the spheres is very simple and can be costly if the inside geometry is complicated, involving several surfaces. However, the nested DXTRAN treatment should alleviate some of these complicated geometry issues. The inner sphere is intended to surround the region of interest. The outer sphere should surround neighboring regions that may scatter into the region of interest. In MCNP6, the relative importance of the two regions is five. That is, the probability density for scattering toward the inner sphere region is five times as high as the probability density for scattering between the inner and outer spheres. This position biasing is only one of several factors that affect the weights of the DXTRAN particles.

All collisions producing neutrons and photons contribute to DXTRAN, including model physics interactions. When the secondary neutron/photon angular scattering distribution function is unknown, isotropic scattering, which may be a poor approximation, is assumed. Although the extension to higher energies often is approximate, a tally with an appropriate energy structure can provide the user with insight regarding the contributions at these energies. This approximation is superior to neglecting charged-particle and high-energy neutron collisions altogether.

As mentioned above, DXTRAN uses an assumption of isotropic scatter for contributions from collisions within the model regime. These estimators require the angular distribution data for particles produced in an interaction to predict scattering toward the sphere(s). Information on these distributions is available in

tabular form in the libraries; however, this information is not available in the required form from physics models used to produce secondary particles above the tabular region.

DXTRAN can be used in a problem with the $S(\alpha, \beta)$ thermal treatment [77], but contributions to the DXTRAN spheres are approximate. DXTRAN should not be used with reflecting surfaces, white boundaries, or periodic boundaries [§2.5.6.4.2]. DXTRAN can be used with mono-direction sources, but the user should understand that no contributions from sources occur unless the source is directed at the DXTRAN sphere.

DXTRAN spheres can be used around point detectors (F5 tallies), but the combination may be very sensitive to reliable sampling.

If more than one set of DXTRAN spheres is used in the same problem, they can "talk" to each other in the sense that collisions of DXTRAN particles in one set of spheres cause contributions to another set of spheres. The contributions to the second set have, in general, extremely low weights but can be numerous with an associated large increase in computer time. In this case the DXTRAN weight cutoffs probably will be required to kill the very-low-weight particles, provided mesh-based weight windows are not used. The DD card can give you an indication of the weight distribution of DXTRAN particles.

Remember that the DD card roulette game is on by default and the reference weight is a moving average for the first *dmmp* histories unless this Russian roulette game is turned off or a fixed level is input (as a $-k_i$ on the DD card). It is highly recommended that the user make a short calculation to establish a value to input; a value that is 10% of the average contributed weight to the sphere is a good place to start. See the DD card input requirements about more details regarding the number of histories used to find the average contribution. If the user were to rely on the default behavior, then running a single history after the first *dmmp* histories (perhaps for the sake of debugging or dealing with a lost particle) will not yield the same result as before.

5.12.11 DD: Detector Diagnostics

The DD card (1) can speed up calculations significantly by using a Russian roulette game to limit small contributions that are less than some fraction *k* of the average contribution per history to detectors or DXTRAN spheres and (2) can provide more information about the origin of large contributions or the lack of a sufficient number of collisions close to the detector or DXTRAN sphere. The information provided about large contributions can be useful for setting cell importance or source-biasing parameters.

The DD card eliminates tracks to DXTRAN spheres, and contributions to detectors.

Data-card Form: DD*n* k_1 m_1 k_2 m_2 ...

n	Control flag. If *n* ends in the number 5, then *n* is the tally number for a specific detector tally to which the card applies. If
	$n = 0$ or is blank, then diagnostic parameters apply to all detector tallies and DXTRAN spheres unless overridden with a separate DD*n* card.
	$n = 1$ provide detector diagnostics for neutron DXTRAN spheres.
	$n = 2$ provide detector diagnostics for photon DXTRAN spheres.
k_k	Criterion for playing Russian roulette for DXTRAN sphere k or detector k in tally *n*. Let A_k be the average score per history for either the sphere or detector (1). If

	$k_k < 0$	DXTRAN or detector scores greater than $\|k_k\|$ will always be made and contributions less than $\|k_k\|$ are subject to Russian roulette; or
	$0 < k_k \leq 1$	all DXTRAN sphere or detector contributions are made for the first *dmmp* histories. Thereafter, any contribution to the detector or sphere greater than $k_k A_k$ will always be made, but any contribution less than $k_k A_k$ is subject to Russian roulette (2); or
	$k_k = 0$	no Russian roulette is played on small DXTRAN or detector scores.
m_k	Criterion for printing diagnostics for large contributions for DXTRAN sphere k or detector k of tally n. Let A_k be the average score per history for either the sphere or detector (1). If	
	$m_k = 0$	no diagnostic print.
	$m_k > 0$ and $k_k > 0$	then no diagnostic print made for the first *dmmp* histories. Thereafter, the first 100 contributions larger than $m_k A_k$ will be printed (2).
	$m_k > 0$ and $k_k < 0$	then the first 100 contributions larger than $m_k \|k_k\|$ will be printed.

Default: If k_k is not specified on a DD*n* card, k_k on the DD card is used. If that is not specified, $k_k = 0.1$ is used. A similar sequence of defaults defines m_k, with a final default of $m_k = 1000$.

Use: Optional. Remember that Russian roulette will be played for detectors and DXTRAN unless specifically turned off by use of the DD card. The value of $k_k = 0.5$ is suitable for most problems; the non-zero default value, $k_k = 0.1$, means that the game is always played unless explicitly turned off by the user. Consider also using the PD or DXC cards.

Details:

1. The average contribution per history, A, to a particular DXTRAN sphere or detector is calculated from all contributions to the detector or sphere made by particle histories until the first tally fluctuation chart (TFC) interval is reached (see the *dmmp* entry on the PRDMP card). The default is 1000 particles per interval for fixed-source problems or one KCODE cycle. The average is then updated at all subsequent tally fluctuation chart intervals.
2. Remember that when k_k is positive, the Russian roulette game is played on the basis of the estimated average contribution per history, A_k. Because the estimate improves from time to time, the game is based on different values for different histories. This can make debugging a problem more complicated, and the variance estimate does not quite obey the Central Limit Theorem.

A procedure worth considering is to determine the average contribution per history in a preliminary calculation and then use some fraction of the negative of this value in subsequent longer runs. The Russian roulette game is played without regard to particle time or energy; thus time and energy bins for which the ultimate tally is small may lose a disproportionate share of scores by the roulette game.

Table 5.26: DD Card Example k_k, m_k Parameters

	k_k	m_k
sphere 1	-1.1×10^{25}	3000
sphere 2	0.15	2000
sphere 3	0.2	3000
sphere 4	0.2	100
detector 1	0.4	10
detector 2	0.15	2000

5.12.11.1 Example 1

```
DXT:N    x1  y1  z1   ri1  ro1
         x2  y2  z2   ri2  ro2
         x3  y3  z3   ri3  ro3
DXT:P    x4  y4  z4   ri4  ro4
F15X:P   a1  r1  R1
         a2  r2  R2
DD        0.2   100  0.15  2000
DD1   -1.1E25  3000     J     J    J  3000
DD15      0.4   10
```

This input results in the following interpretation for the DD parameters for the detectors and DXTRAN spheres shown in Table 5.26.

5.12.12 PD: Detector Contribution

The PD card reduces the number of contributions to point detector tallies (F5) from selected cells that are relatively unimportant to a given detector, thus saving computing time. At each collision in cell j, the point detector tallies are made with probability $0 \leq p_j \leq 1$; that is, a Russian roulette game is played in which the survival probability is p_j to determine if the contribution should take place. When the contribution survives the roulette game, the tally is then increased by the factor $1/p_j$ to obtain unbiased results for all cells except those where $p_j = 0$. This enables the user to decrease the problem runtime by setting $p_j < 1$ for cells many mean free paths from the detectors. It also selectively eliminates detector contributions from cells by setting the p_j values for those cells to zero. This card is analogous to the DXC card, but is used for contributions to point detector tallies (F5).

Caution

Cells should generally never be assigned a value of $p_j = 0$, because this will always prevent contribution from a cell to the associated point detector(s). Unless it can be guaranteed that contributions from the given cell to the detector are impossible, it is recommended to assign a small value to have infrequent but high-weight contributions (by virtue of weight increase following rouletting survival) to give the MCNP code the opportunity to sample important rare events that will manifest as notable increases in the point detector tally's variance and variance of the variance

Cell-card Form: PD*n*=*p*
or
Data-card Form: PD*n p1 p2* ...

n	Tally number (1). Restriction: $n \leq 9999$
p	Probability of contribution to detector *n* from cell. (DEFAULT: $p = 1$)
pj	Probability of contribution to detector *n* from cell k. (DEFAULT: $pj = 1$) Number of entries is equal to the number of cells in the problem.

Default: $pj = 1$

Use: Optional. Consider also using the DD card.

Details:

1 A default set of probabilities can be established for all tallies by use of a PD0 card. These default values will be overridden for a specific tally *n* by values entered on a PD card, where *n* must end in 5 to apply to a particular point detector.

5.12.13 DXC: DXTRAN Contribution

The DXC card is analogous to the PD card for detector contributions except it is used for contributions to DXTRAN spheres.

Cell-card Form: DXC*n*:$\mathscr{P}$=*p*
or
Data-card Form: DXC*n*:$\mathscr{P}$ *p1 p2* ...

n	Which DXTRAN sphere the DXC card applies to. If $n = 0$ or absent, the DXC card applies to all the DXTRAN spheres in the problem. (DEFAULT: $n = 0$)
$\mathscr{P}$	Particle designator. Restriction: Only allowed particles are neutrons (N) and photons (P).
p	Probability of contribution to DXTRAN sphere *n* from cell. (DEFAULT: $p = 1$)
pk	Probability of contribution to DXTRAN sphere *n* from cell k. (DEFAULT: $pk = 1$) Number of entries is equal to the number of cells in the problem.

Use: Optional. Consider also using the DD card.

5.12.14 BBREM: Bremsstrahlung Biasing

The bremsstrahlung process generates many low-energy photons, but the higher-energy photons are often of more interest. One way to generate more high-energy photon tracks is to bias each sampling of a bremsstrahlung photon toward a larger fraction of the available electron energy. Use the BBREM card to specify this biasing toward higher-energy photons.

Data-card Form: BBREM b_1 b_2 ... b_{49} m_1 m_2 ... m_K	
b_1	Any positive value (currently unused).
b_2 ... b_{49}	Bias factors for the bremsstrahlung energy spectrum.
m_1 ... m_K	List of K materials for which the biasing is invoked.

Default: None.

Use: Optional.

5.12.14.1 Example 1

```
BBREM     1.  1.  46I  10.  888  999
```

This specification will create a gradually increasing enhancement (from the lowest to the highest fraction of the electron energy available to a given event) of the probability that the sampled bremsstrahlung photon will carry a particular fraction of the electron energy. This biasing would apply to each instance of the sampling of a bremsstrahlung photon in materials 888 and 999. The sampling in other materials would remain unbiased. The bias factors are normalized by the code in a manner that depends both on material and on electron energy, so that although the ratios of the photon weight adjustments among the different groups are known, the actual number of photons produced in any group is not easily predictable. For the EL03 treatment, there are more than 49 relative photon energy ratios so the lower energy bins have a linear interpolation between b_1 and b_2 for their values.

In most problems the above prescription will increase the total number of bremsstrahlung photons produced because there will be more photon tracks generated at higher energies. The secondary electrons created by these photons will tend to have higher energies as well, and will therefore be able to create more bremsstrahlung tracks than they would at lower energies. This increase in the population of the electron-photon cascade will make the problem run more slowly. The benefits of better sampling of the high-energy domain must be balanced against this increase in run time.

5.12.15 PIKMT: Photon-production Biasing

For several classes of coupled neutron-photon calculations, the desired result is the intensity of a small subset of the entire photon energy spectrum. Two examples are discrete-energy (line) photons and the high-energy tail of a continuum spectrum. In such cases, it may be beneficial to bias the spectrum of neutron-induced photons to produce only those that are of interest.

⚠ Caution

Use of the PIKMT card can cause non-zero probability events to be excluded completely, resulting in a biasing game that may not be fair. While neutron tallies will be unaffected (within statistics), the only reliable photon tallies will be those with energy bins immediately around the energies of the discrete photons produced.

To use this feature, users will likely need information about the MT identifiers of the reactions that produce discrete energy photons. The user is encouraged to consult [Appendix B of 45] for a list of all MT identifiers

and look through [Chapters 12 and 13 of 45] (i.e., Files 12 and 13) for a better understanding of ENDF neutron-induced photon production.

This photon-production biasing feature is also useful for biasing the neutron-induced photon spectrum to produce very high-energy photons (for example, $E_\gamma \geq 10$ MeV). Without biasing, these high-energy photons are produced very infrequently; therefore, it is difficult to extract reliable statistical information about them. An energy cutoff can be used to terminate a track when it falls below the energy range of interest.

Data-card Form: PIKMT $target\_identifier_1$ $ipik_1$ $mt_{1,1}$ $pmt_{1,1}$ ... $mt_{1,ipik_1}$ $pmt_{1,ipik_1}$ ... $target\_identifier_K$ $ipik_K$ $mt_{K,1}$ $pmt_{K,1}$ ... $mt_{K,ipik_k}$ $pmt_{K,ipik_k}$

$target\_identifier_k$	Target identifier [§1.2.2] of the kth entry. All formats are allowed.
$ipik_k$	Controls the biasing for $target\_identifier_k$. If
	$ipik_k = 0$: no photon-production biasing is done for $target\_identifier_k$. That is, photons from $target\_identifier_k$ are produced with the normal sampling technique.
	$ipik_k = -1$: no photons are produced from $target\_identifier_k$.
	$ipik_k > 0$: photon-production is biased for $target\_identifier_k$. The value of $ipik_k$ is the number of partial photon-production reactions to be sampled.
$mt_{k,l}$	MT reaction identifiers for the partial photon-production reactions to be sampled. Note: only required for $target_identifier_k$s that have $ipik_k > 0$, then must provide appropriate $mt_{k,l}$ and $pmt_{k,l}$ pairs.
$pmt_{k,l}$	Controls, to a certain extent, the frequency with which the specified MT reactions are sampled. Note: only required for $target_identifier_k$s that have $ipik_k > 0$, then must provide appropriate $mt_{k,l}$ and $pmt_{k,l}$ pairs (1).

Default: If the PIKMT card is absent, no biasing of neutron-induced photons occurs. If the PIKMT card is present, any target identifier not listed has a default value of $ipik_k = -1$, and no photons are produced for these unlisted target identifiers.

Use: Only useful for biasing photon production. Only available for neutron libraries.

Details:

1. Entries on the *mt* and *pmt* pairs need not be normalized. For a target identifier with a positive value of *ipik*, any reaction that is not identified with its *mt* on the PIKMT card will not be sampled.

5.12.15.1 Example 1

```
PIKMT    Fe-56     1  102001  1            N-14  0
         Cu-63     2    3001  2  3002  1
          O-16    -1
```

This example results in normal sampling of all photon-production reactions for ^{14}N. All photons from neutron collisions with ^{56}Fe are from the reaction with MT identifier 102001. Two photon-production reactions with ^{63}Cu are allowed. Because of the *pmt* parameters, the reaction with MT identifier 3001 is sampled twice as frequently relative to the reaction with MT identifier 3002 than otherwise would be the case. No photons are produced from ^{16}O or from any other isotopes in the problem that are not listed on the PIKMT card.

5.12.16 SPABI: Secondary Particle Biasing

Secondary particle biasing allows the user to adjust the number and weight of secondary particles produced at the time of their creation. Multiple SPABI cards for different secondary particles are allowed.

Data-card Form: SPABI:$\mathscr{P}1$ $\mathscr{P}2$ e_1 s_1 e_2 s_2 ...

$\mathscr{P}1$	Secondary particle designator [Table 4.3].
$\mathscr{P}2$	List of primary particles to be considered. For example, NPHE represents reactions of neutrons, photons, protons, and electrons. No spaces are allowed. If all particles are to be considered, the entry should be ALL.
e_k	The kth upper energy bin limit of secondary particles. The lower bin limit is considered to be zero.
s_k	Splitting/rouletting control. If
	$s_k > 1$ use splitting for secondary particles in the kth bin.
	$0 \le s_k \le 1$ use roulette if for secondary particles in the kth bin.

5.12.16.1 Example 1

```
SPABI:N   NHE   1  0.1  5  1  10  2  20  4
```

This example specifies that neutron secondaries produced by neutron, proton, and electron primaries will be biased in the following manner: below 1 MeV, the secondary neutrons will be rouletted by a factor of 0.1. At energies, 1 to 5 MeV, no biasing is performed. At energies from 5 to 10 MeV, the secondary neutrons will be split 2-for-1, and from 10 to 20 MeV, the secondary neutrons will be split 4-for-1 (with a corresponding reduction in particle weights).

5.12.17 PWT: Photon Weight Control

The PWT card is used in MODE N P or MODE N P E problems. Its purpose is to control the number and weight of prompt neutron-induced photons produced at neutron collisions. Use the ACT card to control the number and weight of delayed photons.

Cell-card Form: PWT=w
or
Data-card Form: PWT w_1 w_2 ...

w Relative threshold weight of photons produced at neutron collisions in cell (1, 2). If

$w > 0$	only neutron-induced photons with weights greater than $w \times I_s/I_k$ are produced, where I_s and I_k are the neutron importance of the source and collision cells, respectively. Russian roulette is played to determine if a neutron-induced photon with weight below this value survives.
$w < 0$	only neutron-induced photons with weights greater than $-w \times w_s \times I_s/I_k$ are produced, where w_s is the starting weight of the neutron for the history being followed, and I_s and I_k are the neutron importance of the source and collision cells, respectively. Russian roulette is played to determine if a neutron-induced photon with weight below this value survives.
$w = 0$	exactly one photon will be generated at each neutron collision in the cell, provided that photon production is possible.
$w =$ -1.0e6	photon production in the cell is turned off.

w_k Relative threshold weight of photons produced at neutron collisions in cell k (1, 2). Number of entries is equal to number of cells in the problem. If

$w_k > 0$	only neutron-induced photons with weights greater than $w_k \times I_s/I_k$ are produced, where I_s and I_k are the neutron importance of the source and collision cells, respectively. Russian roulette is played to determine if a neutron-induced photon with weight below this value survives.
$w_k < 0$	only neutron-induced photons with weights greater than $-w_k \times w_s \times I_s/I_k$ are produced, where w_s is the starting weight of the neutron for the history being followed, and I_s and I_k are the neutron importance of the source and collision cells, respectively. Russian roulette is played to determine if a neutron-induced photon with weight below this value survives.
$w_k = 0$	exactly one photon will be generated at each neutron collision in cell k, provided that photon production is possible.
$w_k =$ -1.0e6	photon production in cell k is turned off.

Default: $w_k = -1$ if neutrons and photons appear on the MODE card.

Use: Recommended for MODE N P and MODE N P E problems without weight windows.

Details:

(1) For problems using photon cell importance (IMP:P), rather than photon weight windows (WWN*n*:P), a good first guess for PWT card entries is either the default value, $w_k = -1$, or set w_k in every cell to the average source weight.

(2) For problems with photon weight windows (i.e., WWP:P exists), the PWT card is ignored and the correct numbers of photons are produced within the weight windows.

The PWT card controls the production of neutron-induced photons by comparing the total weight of photons produced with a relative threshold weight specified on the PWT card. This threshold weight is relative to the neutron cell importance and, if $w_k < 0$, to the source neutron weight. If more neutron-induced photons are desired, the absolute value of w_k should be lowered to reduce the weight and therefore increase the number of photons. If fewer neutron-induced photons are desired, the absolute value of w_k should be increased.

5.13 Problem Termination, Output Control, and Miscellaneous Cards

5.13.1 Problem Termination

Six normal ways to terminate an MCNP6 calculation are:

1. the NPS card,
2. the CTME card,
3. the STOP card,
4. the calculation time limit,
5. the end of a surface source file, and
6. the number of cycles on a KCODE card.

If more than one termination condition is in effect, then the one encountered first will control termination of the MCNP6 calculation.

5.13.1.1 NPS: History Cutoff

Terminates the MCNP6 calculation after a requested number of histories have been transported, unless the calculation is terminated earlier for some other reason (such as the computer time cutoff, CTME).

Data-card Form: NPS *npp npsmg n_per_batch*

npp	The total number of histories to be run in the problem. An 8-byte integer is permitted for *npp* (1,2).
npsmg	Number of histories for which direct source contributions are to be made to the pixels of an FIR radiography image grid. An 8-byte integer is permitted for *npsmg* [§2.5.6.3.1]. (3)

n_per_batch	The number of histories to be used per batch for tallies that use batch statistics. If no tallies use batch statistics, this has no effect. If *npp* is not evenly divided by *n_per_batch*, *npp* will be increased to the nearest value that is. *npsmg* is not adjusted at this time, as the FIR tally does not yet support batch statistics. An 8-byte integer is permitted for *n_per_batch*.

Default: Infinite.

Use: As needed to terminate the calculation. In a criticality calculation, the NPS card has no meaning and a warning error message is issued if it is used. Highly recommended for use in multiprocessor computations.

Details:

1. In a restart calculation, the NPS input values are the total number of particles including those calculations before the current restarted calculation; they are cumulative. A negative *npp* entry means to print an output file at the time of the last history run and then stop.
2. In a surface source problem, either more or fewer than all of the particle histories on the **RSSA** surface source file will be run, depending on the value of *npp* entered on the NPS card. Let N_1 represent the number of original histories. If *npp* $< N_1$, Russian roulette with weight adjustment will be played with each history in the file using a survival probability of *npp*$/N_1$. If *npp* $> N_1$, the histories will be split *npp*$/N_1$-to-1 with the fractional part is taken care of by sampling. This can be done equally well for non-spherical sources by cell importance splitting. With a spherical source, each multiple occurrence of the history is sampled for a different starting location on the source sphere, possibly improving the spatial statistics of the results. In either case, the use of the NPS card will not provide additional information about the original source distributions or the transport to the recording surface crossing.
3. When the number of source histories exceeds *npsmg*, the time-consuming process of determining the attenuation of the FIR direct contribution is avoided by adding the average of the previous direct contributions into each of the appropriate tally bins. Depending on the computer time required to calculate the direct image in a particular problem, *npsmg* can save from a few seconds to upward of ten minutes per history in some cases. For example, a mono-energetic isotropic point source or a mono-energetic mono-directional surface source requires only one history to determine completely the direct image. For this case, *npsmg* $= 1$ is adequate.

5.13.1.2 CTME: Computer Time Cutoff

Allows the user to specify minutes of computer time after which MCNP6 will terminate the calculation. MCNP6 checks the computer time remaining in a running problem and will terminate the calculation itself, allowing enough time to wrap up and terminate gracefully.

Data-card Form: `CTME` *tme*

tme	maximum amount of computer time (in minutes) to be spent in the Monte Carlo calculation (1, 2).

Default: Infinite.

Use: As needed.

Details:

(1) For a restarted calculation, the time on the CTME card is the time relative to the start of the restarted calculation; it is not cumulative.

(2) For multiprocessor calculations, it is highly recommended that the NPS card be used to limit the run time.

5.13.1.3 STOP: Precision Cutoff

Enables termination of calculations based on the number of particle histories run, the computer time expended, or the desired precision for a specified tally.

Data-card Form: `STOP keyword=value(s)...`

`NPS npp [npsmg]`	Stop calculation after *npp* particle histories ((1)). See the NPS card.
`CTME tme`	Stop calculation after *tme* minutes of computer time ((2)). See the CTME card.
`Fk e`	Stop calculation when the tally fluctuation chart of tally *k* has a relative error less than *e*. ((3))

Use: If values for any (or all) of the keywords are supplied, MCNP6 will terminate the problem at the first met criteria.

Details:

(1) For radiography problems, a second NPS keyword entry, *npsmg*, may be provided to specify how many histories are used for direct radiography tally contributions.

(2) For multitasking calculations, CTME will be checked only at rendezvous points, where all tasks rendezvous for tally fluctuations and other activities.

(3) The tally precision stop will be checked only at rendezvous points for the tally bin of the tally fluctuation charts. Thus, the calculation usually will proceed for a short time after the desired error is achieved. See TF card.

5.13.1.3.1 Example 1

```
1 STOP   F111   0.05
```

MCNP6 will stop at the first rendezvous for which the relative error of the tally bin for the tally fluctuation chart of tally F111 is less than 0.05. MCNP6 may stop at error=0.048 or other value slightly less than 0.05.

5.13.2 RAND: Random Number Generation

Specifies the type of random number generator as well as the seed, starting history number, and stride (if applicable).

Data-card Form: RAND keyword = value(s)...

GEN	Type of random number generator to be used by MCNP6 (3). If
GEN = 1	then use MCNP6 Lehmer 48-bit Linear Congruential Generator (LCG), which has a period of 7.0×10^{13} numbers. (DEFAULT)
GEN = 2	then use L'Ecuyer 63-bit LCG number 1, which has a period of 9.2×10^{18} numbers.
GEN = 3	then use L'Ecuyer 63-bit LCG number 2, which has a period of 9.2×10^{18} numbers.
GEN = 4	then use L'Ecuyer 63-bit LCG number 3, which has a period of 9.2×10^{18} numbers.
GEN = 5	then use L'Ecuyer 63-bit LCG number 4, which has a period of 2.3×10^{18} numbers.
GEN = 6	then use L'Ecuyer 63-bit LCG number 5, which has a period of 2.3×10^{18} numbers.
GEN = 7	then use L'Ecuyer 63-bit LCG number 6, which has a period of 2.3×10^{18} numbers.
GEN = 8	then use SFC64, which can provide $2^{192} \approx 6.2 \times 10^{57}$ histories an independent random sequence of length $2^{64} \approx 1.8 \times 10^{19}$ (5). This generator is recommended for all use cases.
SEED	Random number generator seed for starting the transport of the first particle history in a run. (DEFAULT: SEED = 19073486328125) Restriction: For generators 1–7, must end with an odd digit. Note: An 8-byte integer is permitted for keyword SEED (i.e., up to 18 digits). (4)
STRIDE	Number of random numbers between source particles. This option does not affect generator 8. Note: An 8-byte integer is permitted for keyword STRIDE. (DEFAULT: STRIDE = 152917)
HIST	If HIST = *n*, then causes the starting random number of the problem to be that which would normally start the *n*th history. That is, causes the *n*th history to be the first history of a problem for debugging purposes. Setting HIST = *n* can also be used to select a random number sequence different from that in an identical problem to compare statistical convergence. Note: An 8-byte integer is permitted for keyword HIST. (4) (DEFAULT: HIST = 1)

Details:

1. RAND entries must be used instead of obsolete DBCN entries 1, 8, 13, and 14.
2. The RAND card may be used to change the problem random number seed in a restarted calculation. This capability provides a work-around for avoiding a troublesome particle history. This procedure is not recommended, but is permitted. Be aware that repeatability is very difficult to achieve if this feature is used.
3. If the period is exceeded, random numbers will be reused, but the random number sequence used for subsequent histories will differ from the random number sequence used for previous histories. There

should be no impact on results. Generator 8 provides the longest period. For generators 1–7, decreasing the stride will reduce the chances of exceeding the period, but may cause reuse of random numbers if the stride is exceeded. Generally, strides of 4000 or so are reasonable for criticality problems, while the default stride or greater may be needed for problems with heavy variance reduction.

(4) The ith source particle always starts with the same random number; this correlated source sampling enables faster evaluation of small problem differences where the problems have identical source distributions. Caution:

⚠ Caution

When trying to duplicate a particle history by setting the starting random number with either SEED or HIST, the random number sequence may be altered by a default Russian roulette game on contributions to detectors or DXTRAN spheres. If a problem has detectors or DXTRAN, the only ways to reproduce histories with SEED or HIST are a) to turn off the Russian roulette game on the DD card by setting $k_i = 0$; b) to play the roulette game with a fixed criterion by setting $k_i < 0$ on the DD card; or c) to reproduce a particle history that occurs before the first TFC interval.

(5) The SFC64 generator operates differently than the other generator options. In the LCGs, there is a single stream within which each history is started at some equally spaced point based on the STRIDE option. In SFC64, the initial condition for each history is a function of both the history identifier and the user-input SEED. This function is designed such that changing the SEED will change the initial condition for every history to an independent one. Each initial condition provides an independent sequence of length 2^{64}. As a result, random number reuse is functionally impossible and there are no pairs of SEEDs that can result in correlated simulations [§2.11.2].

5.13.3 PRINT: Printed Output Tables

Allows selective printing or suppression of optional output tables.

Data-card Form: PRINT x_1 x_2 ...

x_i	List of optional and default table numbers to be included in or excluded from the output file. If there are no entries for x_i, the full output print is provided. If	
	$x_i > 0$	for all i, the tables specified by each positive x_i are provided in addition to the "basic" output tables.
	$x_i < 0$	for any i, the full output applicable to the problem is printed with the exception of those tables identified by negative x_i values.

Default: Absence of a PRINT card in the MCNP input file or a PRINT option on the MCNP6 execution line [§3.3.2] will result in a reduced output print comprised only of the tables in Table 5.27 marked "basic," "default," and "shorten."

Use: Optional. To get all optional PRINT tables applicable to your problem use the PRINT card without entries in the MCNP input file or the PRINT option on the execution line. The execution line takes precedence over the input card. Entries on the PRINT card can be in any order; however, no entries may follow the PRINT option on the MCNP6 execution line.

Table 5.27: MCNP6 Output Tables

#	Type	Table Description	§
10	optional	Source coefficients and distribution.	5.13.3.4
20	optional	Weight-window information.	
30	optional	Tally description.	
32	optional	Mesh tally description.	
35	optional	Coincident detectors.	
38	optional	Fission multiplicity data; controlled by Table 30.	
40	optional	Material composition.	
41	default	LAHET physics options.	
43	unavailable	LAHET elastic cross sections.	
44	optional	Activities of the materials in an SDEF PAR = SP problem.	
50	optional	Cell volumes and masses, surface areas.	5.13.3.6
60	basic	Cell importance.	5.13.3.7
62	basic	Forced collision and exponential transform.	
70	optional	Surface coefficients.	5.13.3.8
72	basic	Cell temperatures.	5.13.3.9
80	optional	ESPLT/TSPLT importance ratios.	
85	optional	Continuous energy calculation: Charged-particle stopping powers and straggling.Multigroup calculation: flux values for biasing adjoint calculations.	5.13.3.10
86	optional	Electron bremsstrahlung and secondary production.	5.13.3.11
87	optional	Secondary heavy ion stopping powers and straggling (1).	
90	optional	KCODE source data.	
95	default	GENXS [§5.7.10.1] tally input.	
98	optional	Physical constant and compile options.	5.13.3.12
100	basic	Cross-section tables.	5.13.3.13
101	basic	Particle and energy limits.	
102	optional	Assignment of $S(\alpha, \beta)$ data to nuclides.	
110	optional	Initial phase-space values for first 50 histories	5.13.3.14
115	default	Fission multiplicity summary.	
117	default	Spontaneous and induced fission multiplicities and moments.	
118	default	Neutron captures, moments, and multiplicity distributions.	
120	optional	Analysis of the quality of your importance function.	
126	basic	Particle activity in each cell.	5.13.3.15
128	optional	Universe map.	
130	optional	Neutron/photon/electron weight balance.	5.13.3.16
140	optional	Neutron/photon nuclide activity (2).	5.13.3.17
150	optional	DXTRAN diagnostics.	
160	default	TFC bin tally analysis.	
161	default	tally empirical history-score probability density function plot.	5.13.3.18
162	default	tally empirical history-score cumulative density function plots.	5.13.3.19
163	optional	Receiver-Operator Characterization (ROC) curve data.	
170	optional	Source distribution frequency tables, surface source.	
175	shorten	Estimated k_{eff} results by cycle.	5.13.3.20
178	optional	Estimated k_{eff} results by batch size.	
179	optional	ASCII plot of estimated collision/absorption/track-length k_{eff} one-standard-deviation interval versus cycle number.	
190	basic	Weight-window generator summary.	5.13.3.21

continued on next page...

Table 5.27, continued

#	Type	Table Description	§
198	optional	Weight windows from multigroup fluxes.	
199	optional	Weight-window diagnostics table.	5.13.3.22
200	basic	Weight-window-generated values.	5.13.3.23
210	default	Burnup summary table.	
220	default	Burnup summary table summed over all materials.	

Details:

1. Table 87 will not be printed unless `85` and `87` are specified explicitly on the `PRINT` card.
2. The `DISABLE` card will suppress this table even if it is listed on the `PRINT` card.
3. The LNK3DNT embedded geometry information `PRINT` table has placeholder number "XX." This deficiency has been logged in the MCNP issue tracking system for resolution.

The following output will be printed automatically, as applicable:

- a listing of the input file,
- the problem summary of particle creation and loss,
- `KCODE` cycle summaries,
- tallies,
- tally fluctuation charts,
- the tables listed in Table 5.27 as basic, and
- the tables listed in Table 5.27 as default, provided they are not turned off explicitly with the `PRINT` card.

In an MCNP6 output file, a table number appears in the upper right-hand corner of each table, providing a convenient pattern when scanning the output file with an editor. The pattern is "`print table n`," where *n*, the table number, is always preceded by one space and is a two- or three-digit number. The table numbers, titles, and type are summarized in Table 5.27. "Basic" tables cannot be controlled by the `PRINT` card. "Default" tables are automatically printed but can be turned off. "Optional" tables with can be turned off and on with the `PRINT` card or option.

With two exceptions, the `PRINT` control can be used in a restarted calculation to recover all or any applicable `PRINT` tables, even if they were not requested in the original calculation. The following example restarted-calculation input file:

```
continue
nps -1
print
prdmp 2j 1
```

will create the output file for the initial calculation starting with the Problem Summary (located after `PRINT table 110`) and will also create a **`mctal`** file using the `PRDMP` card. However, this input may not always be sufficient; additional information such as an `EMBED` card may be needed for calculations that use embedded mesh geometries. Moreover, note that

- PRINT table 128 can never be printed if it was not requested in the original calculation and
- PRINT table 140 can never be printed if it was disabled in the original calculation using the DISABLE card.

Several of the output tables listed in Table 5.27 have additional restrictions:

1. If you turn off table 160, then tables 161 and 162 will not appear either. If table 160 is printed, they will all be printed. All three tables are automatically printed if there is no PRINT card or if there is a blank PRINT card. If a PRINT card has a positive entry, tables 160, 161, and 162 will not appear unless table 160 is explicitly requested. If the entry is negative, they will appear unless table 160 is explicitly turned off.
2. Table 175 cannot be turned off completely, but the output can be greatly shortened to every 100 cycles plus the last five cycles. The specification PRINT -175 and PRINT 110 both will produce the short version of table 175.
3. Tables 128 and 140 have unique storage behavior. If table 128 is not turned on in an initial calculation, it cannot be turned on in a subsequent restarted calculation because the (often large) storage arrays have not been set up. If table 140 is disabled in the initial calculation using the DISABLE card, it cannot be turned on in a subsequent restarted calculation. The information in the other tables is always stored, whether or not it is printed. A warning will be printed in a repeated structures problem if the universe map/lattice activity table (table 128) is not requested in the initial calculation. Similarly, a comment will be printed if table 140 is disabled in the initial calculation.
4. PRINT Table 87 does not appear as a result of the standard "default" convention because stopping powers for all 100 elements for each material results in large output files. To output Table 87, the user must specify 85 and 87 explicitly on the PRINT card.

5.13.3.1 Example 1

```
PRINT    110   40   150
```

The output file will contain the "basic" tables plus tables 40, 110, 150, and the shortened version of 175, but not 55, 117, 118, 160, 161, 162, 210, 220 (the "default" tables).

5.13.3.2 Example 2

```
PRINT     -170   -70   -110
```

The output file will contain all the "basic" tables, all the "default" tables, the long version of Table 175, and all the optional tables applicable to your problem, except Tables 70, 110, and 170.

5.13.3.3 Example 3

```
PRINT    -1   87
```

Prints all output including Table 87.

5.13.3.4 Table 10

All source variables defined explicitly or by default are printed. The order of sampling of the source variables is also printed, which is important for source variables that are dependent on other source variables.

5.13.3.5 Table 43

Output of this table is controlled from within the LAHET Code System but never enabled.

5.13.3.6 Table 50

A cell can be composed of physically separate regions or pieces joined with the union operator. Improperly defined cells can be composed unintentionally of more than one piece (for example, a surface is extended unknowingly and forms a cell). If a cell is composed of more than one piece, a warning message is given and one should verify that the number of pieces is correct.

If the mass or volume of a geometry or parts of it are known, one should compare the known volume or mass with what the MCNP code calculates to verify the correctness of the geometry. The volumes, masses, and/or surface areas that the MCNP code cannot calculate (but supplies a placeholder value such as zero) do not affect the totals. Cell volumes that are not calculated by the MCNP code can be entered on the `VOL` card. Areas that are not calculated by the MCNP code can be entered on the `AREA` card.

5.13.3.7 Table 60

This table summarizes cell properties. It includes values given in `PRINT` Table 50 on density, volume, mass, and number of pieces. However, it also includes the materials assigned to each cell (and whether they are modified by an $S(\alpha, \beta)$ treatment shown as an `s` after the material number) and the importance(s) assigned to the cell.

5.13.3.8 Table 70

The entries in this table are the surface coefficients used by the MCNP code [§1.3.3.1] and are not necessarily the entries on the surface cards.

5.13.3.9 Table 72

The cell temperature can be controlled using `TMP` or else a default value is used (2.53×10^{-8} MeV).

5.13.3.10 Table 85

The "Density Effect Data" table contains the material data necessary to correct the stopping power term for the polarization of the media. If a fast electron passes through an equal areal density (mass density multiplied by length) of two materials, it will lose more energy in a sparse (rather than dense) material. This effect is small for heavier particles, but for electrons with relativistic velocities transversing dielectric media, it can be significant. For 1-MeV electrons in water, this correction can be as large as 5%.

Next, the electron range and straggling table for each material is listed. Electron energies are listed in in ascending order and gives the respective stopping powers due to collision and radiation and the range of the electron in the material. Radiation yield is the fraction of the electron's kinetic energy that is converted into bremsstrahlung energy.

5.13.3.11 Table 86

The "electron secondary production for..." table contains a list of electron energies in ascending order and gives the respective stopping powers due to collision and radiation and the range of the secondary electron created in the electron in the material.

5.13.3.12 Table 98

The physical constants and units used in the MCNP code are listed here.

The compilation options are also listed. Knowing how the code was compiled is useful if it is slow, runs out of space, does not plot (usually because the plot option is wrong for the computer being used or run-time libraries for plotting are located somewhere other than expected), or cannot find the data libraries (usually because of an incorrect **DATAPATH** environment variable).

5.13.3.13 Table 100

The cross-section table list shows the nuclear and atomic data used in the problem. For example, a **c** appended to the neutron data indicates continuous energy, a **d** indicates discrete reaction data, a **p** indicates photon data, and an **e** indicates electron data. A listing of data classes is given in Table B.1.

Warnings are printed in `MODE n p` problems if the photon production cross sections are unavailable or are in the less accurate equiprobable bin format. Note that electron data may be loaded even though electrons are not transported because the data may be used for the thick-target bremsstrahlung model.

Any cross sections outside the energy range of the problem as specified by the `PHYS` and `CUT` cards are deleted.

5.13.3.14 Table 110

This table gives starting information about the first 50 source particles. The columns are as follows:

nps	the history number for the source particle,
x,**y**,**z**	the initial position of the source particle in (x, y, z) coordinates,

cell	the cell ID of the region of space that the particle is started in or directed into,
surf	the surface the particle started on, if any,
u,**v**,**w**	the initial direction of the source particle as $\mathbf{\Omega} \equiv \mathbf{u}\mathbf{i} + \mathbf{v}\mathbf{j} + \mathbf{w}\mathbf{k}$,
energy	the initial energy of the particle in MeV,
weight	the initial statistical weight of the particle, and
time	the initial (physical) time of the particle in shakes.

5.13.3.15 Table 126

This table provides cell-by-cell information on particle behaviors.

The tracks entering a cell refers to all tracks entering a cell, including source particles. If a track leaves a cell and later reenters that same cell, it is counted again. Tracks entering does not include particles from the bank (from variance-reduction events at collisions or physical events at collisions).

The population in a cell is the number of tracks entering a cell plus source particles plus particles from the bank (from variance-reduction or physical events at collisions). Population does not include reentrant tracks. Comparing tracks entering and population for a given cell can indicate the amount of back scattering in the problem. An often successful approach for choosing importances is to select them so that population is kept roughly constant in all cells between the source and tally regions. Information, carried by histories through phase space, cannot be regained once lost.

The number of collisions in a cell is important for detectors and anything involving collision rate. A lack of collisions may indicate a need to force them. This quantity is not normalized by cell volume. In some problems, most of the computer time is spent modeling collisions. Cells with excessive numbers of collisions are possibly oversampled in this regard. This often happens when many thermal neutrons diffuse and contribute little of significance to the problem solution. In such cases, energy-dependent weight windows are most effective, followed by energy roulette, exponential transform, analog capture, time cutoff, and/or energy cutoff. Note that the last two methods may introduce a bias into the problem. Subdividing the cell into smaller cells with different importances is also effective.

The number of collisions times the weight of the particles having the collisions is an indication of how important the collisions were.

The number-weighted energy in a cell is computed as

$$\frac{\int \mathrm{d}E \int \mathrm{d}t\, E \cdot n(E,t)}{\int \mathrm{d}E \int \mathrm{d}t\, n(E,t)} = \frac{\sum E \cdot w \cdot (T_l/v)}{\sum w \cdot (T_l/v)}, \tag{5.50}$$

where E is energy, t is time, $n = T_l/v$ is the number density of the particles, T_l is distance to next event (i.e., the track length), v is particle velocity, and w is particle statistical weight. The number-weighted energy can be useful to understand what energy is dominating a cell, and if low and unlikely to contribute significantly later, whether an energy-controlling variance-reduction technique may be useful.

Similarly, the flux-weighted energy in a cell is computed as

$$\frac{\int \mathrm{d}E \int \mathrm{d}t\, E \cdot \phi(E,t)}{\int \mathrm{d}E \int \mathrm{d}t\, \phi(E,t)} = \frac{\sum E \cdot w \cdot T_l}{\sum w \cdot T_l}. \tag{5.51}$$

It is difficult, and perhaps meaningless, to determine an average energy in this way because a large spectrum involving several orders of magnitude is frequently encountered leading to the problem of representing this

spectrum by one number. That is why an average energy has been calculated using two methods. If the number-weighted energy is significantly lower than the flux-weighted energy, it indicates a large number of low-energy particles with large track time (large T_l/v). As the energy cutoff is raised, these two weighted energies agree more closely. Note that the two average energies are identical for constant velocity photons.

The relative average track weight is computed as

$$\frac{I_c \sum w \cdot T_l}{\sum T_l}, \tag{5.52}$$

where I_c is the importance of the cell under consideration. By making the average track weight relative to the cell importance, the weight reduction from importance splitting is removed. For most problems with consistently assigned cell importances and a source-cell importance of one, the average track weight is constant from cell to cell and significant deviations indicate a poor importance function. With weight windows, the average track weight should be within the weight window bounds.

The average flux-weighted track mean-free path is computed as

$$\frac{\int \mathrm{d}E\, \phi(E)/\Sigma_t(E)}{\int \mathrm{d}E\, \phi(E)} = \frac{\sum w \cdot T_l/\Sigma_t}{\sum w \cdot T_l}, \tag{5.53}$$

where Σ_t is the total macroscopic cross section for the material in the cell under consideration. The mean-free path is strongly dependent upon energy, so this average mean-free path may be meaningless. However, an approach for developing cell-based importances is that importances should double approximately every mean free path. This is usually a poor rule, but it is sometimes better than nothing. The average-track mean-free path is thus useful for making (potentially poor) guesses at cell importances. It is also useful for determining the fictitious sphere surrounding point detectors [§5.9.1.2], the outer radius of DXTRAN spheres [§5.12.10], exponential transform stretching parameters [§5.12.7], the necessity of forced collisions [§5.12.9], etc. Occasionally this quantity may even provide physical insight into the problem.

5.13.3.16 Table 130

The tables listed show all possible ways a particle's weight may be changed in each cell based on external, variance-reduction, and physical events. In addition to itemizing what is happening to particles and where, this information can be useful in debugging a problem.

Note that there may be apparent weight discrepancy between `PRINT` Table 126 and 130, but this is because Table 126 concerns tracks while Table 130 concerns histories. Furthermore, in Table 126 the weight is relative, whereas it is absolute in Table 130. If the average track weight is multiplied by the tracks entering cell and then divided by both the number of source particles and the importance ratio, the two weights are in close agreement. The overall value of $\overline{\nu}$ in a problem with fissionable material can be obtained by taking the ratio of fission neutrons to fission loss in Table 130

5.13.3.17 Table 140

The activity of each nuclide per source particle in each cell can tell one how important various nuclides, such as trace elements, are to the problem and may aid in selecting cross-section libraries when memory is limited.

Neutron-induced photon production is also listed for `MODE n p` problems. Totals are also given for activity per source particle summed over all cells in the problem and photoatomic nuclide activity per source particle summed over all cells in the problem, as applicable.

If applicable, another set of information is provided for how many photons were produced in each cell and the energy spectrum of the photons averaged over the problem. Because photons are produced only at neutron collisions in certain calculations, there is a correlation between the number of collisions in a cell, the PWT card, and this output. The earlier output showing the photon activity for the problem includes isotope-dependent neutron-induced photon-production information.

5.13.3.18 Table 161

This plot is the unnormalized empirical history score probability density for the tally fluctuation chart bin of the tally under consideration. The probability density is the number of history scores (horizontal) plotted against the value of the score (vertical). The nonzero mean is denoted by the horizontal line of **m**s. If a problem has been undersampled, this plot will often show "holes," or unsampled regions of the PDF for relatively high scores. If the slope is less than 10, this plot will also show a curve of **s**s, which represent the Pareto curve fit to the PDF at the high scores. This allows the user to visually compare the fit of the high-weight tally scores to the calculated distribution.

5.13.3.19 Table 162

The first plot is the cumulative number of scores in the tally fluctuation chart bin of tally under consideration. It is the cumulative version of PRINT Table 161; i.e., the cumulative probability density function. The ordinate and abscissa values are printed in the left-hand columns. A plot then follows that gives the unnormalized cumulative scores in the tally fluctuation chart bin.

5.13.3.20 Table 175

The a minimal set of output is produced that indicates how k_{eff} iteration is proceeding by cycle if full or optional output of this table or PRINT Table 179 is not selected. If selected, then a detailed listing showing seven different estimates of k_{eff} and removal lifetimes are given versus cycle.

5.13.3.21 Table 190

This table lists the lower weight-window bounds generated by the stochastic weight-window generator [§2.7.2.12.2] using the WWG card.

5.13.3.22 Table 199

This table gives a listing of tracks by cell in constant-scaled weight windows relative to the cell-wise weight-window lower bound value. The intent of this table is to communicate the number of tracks above the local weight window, and by how much, to inform whether the weight window should be scaled up or otherwise adapted to reduce the frequency of particle splitting.

Note that this table is only produced if specifically requested using the PRINT card, e.g., as PRINT 199.

5.13.3.23 Table 200

This table provides weight-window input cards (e.g., WWP and WWN cards) that can, with some text editing, be used instead of the IMP cards in subsequent calculations.

Necessary edits may include replacing zero-valued weight-window lower bounds with a good guess and/or adjusting weight-window lower bounds that differing significantly from those in neighboring cells. Output is also given that indicates weight-window lower-bound ratios between adjacent cells to simplify the aforementioned edits.

5.13.4 TALNP: Negate Printing of Tallies

Controls the printing of bin prints for specified tally numbers.

Data-card Form: `TALNP` t_1 t_2 ...

t_i	List of tally numbers to be excluded from output file (1). If there are no t_i entries, then turn off the bin prints for all tallies in the problem. If there are t_i entries, then turn off the bin prints for the tally numbers that are listed.

Default: If card is present without entries, then no bin prints are provided for tallies. If card is absent, bin prints are provided for all tallies.

Details:

1 If, after the calculation is completed, one would like to see these numbers, the printing of the bin values can be restored with the TALNP card in an MCNP input file used in a restarted calculation. To accomplish this, the tally numbers t_i must be entered on the TALNP card as negative numbers. A single entry of zero in a restarted-calculation input file restores the prints of all tally bins.

5.13.5 PRDMP: Print and Dump Cycle

The PRDMP card allows the user to control the interval at which tallies are printed to the MCNP output file and information is dumped to the runtape file.

For this card, many options have dual meanings, one for a purely history-based simulation, and one for a batch (NPS with *n_per_batch* set and a batch-based tally enabled) or cycle (KCODE) based simulation.

Data-card Form: `PRDMP ndp ndm mct ndmp dmmp`

ndp	Increment for printing tallies. An 8-byte integer is permitted for *ndp*. If
ndp > 0	the problem summary and tallies are printed to the output file after every *ndp* histories (or cycles for a KCODE or batch-based problem) (1).

	ndp < 0	the problem summary and tallies are printed to the output file after every *ndp* minutes of computer time.
ndm	Increment for dumping to the runtape file. An 8-byte integer is permitted for *ndm*.	
	ndm > 0	a dump is written to the runtape file after every *ndm* histories (or cycles for a KCODE or batch-based problem) (1).
	ndm < 0	a dump is written to the runtape file after every *ndm* minutes of computer time.
mct	Controls printing of **MCTAL** file (2). If	
	mct > 0	write **MCTAL** file at problem completion.
	mct $= 0$	do not write **MCTAL** file.
	mct $= -1$	**MCTAL** file is written at problem completion, but references to code name, version number, problem ID, figure of merit, and anything else having to do with running time are omitted from **MCTAL** and MCNP output file. This ensures tracking runs (using identical random walks) yield identical **MCTAL** and MCNP output files.
	mct $= -2$	**MCTAL** file is written at problem completion, but additional prints in MCNP output file are turned off to assist in comparing multitasking output.
ndmp	Maximum number of dumps on the runtape file (3).	
dmmp	Controls how frequently tally fluctuation chart (TFC) entries and rendezvous occur (4, 5). An 8-byte integer is permitted for *dmmp*. If	
	dmmp < 0	write charts every 1000 particles for non-KCODE, non-batch problems or every $\lvert dmmp \rvert$ cycles for KCODE or batch-based problems.
	dmmp $= 0$	write charts every 1000 particles or, if multiprocessing, 10 times total during the calculation.
	dmmp > 0	write charts every *dmmp* particles for non-KCODE, non-batch problems or every $\lvert dmmp \rvert$ cycles for KCODE or batch-based problems.

Default: Print only after the calculation has successfully ended. Dump to the runtape every 60 minutes and at the end of the problem. Do not write a **MCTAL** file. Write all dumps to the runtape file. Write charts and rendezvous for fixed-source problems every 1000 particles or, if multiprocessing, 10 times total during the calculation (dmmp=0); for KCODE or batch-based problems, write charts and rendezvous at the end of each cycle.

Use: Recommended, especially for complex problems. For multiprocessor problems, it is recommended that the *ndp*, *ndm*, and *dmmp* entries be provided in number of histories.

Details:

1. If *ndp* or *ndm* is set to time in a parallel calculation, it will be time used by one processor, approximately elapsed wall time. The scheduled print or dump will be delayed to the next rendezvous or cycle to assure consistent data. For parallel (i.e., multiprocessor) calculations, is highly recommended that the *ndp* and *ndm* values be set in terms of particles or cycles, instead of minutes,

2. The **MCTAL** file is an ASCII file of tallies that can be subsequently plotted with the MCNP6 MCPLOT capability. The **MCTAL** file is also a convenient way to store tally information in a format that is stable for use in the user's own auxiliary programs. For example, if the user is on a system that cannot use the MCNP6 MCPLOT option, the **MCTAL** file can be manipulated into whatever format is required by the user's own local plotting algorithms.

3. Using the parameter *ndmp*, the PRDMP card allows the user to control the size of the runtape file. The runtape file will contain the last *ndmp* dumps that were written. For example, if *ndmp*=4, after dump 20 is written only dumps 17, 18, 19, and 20 will be on the runtape file. In all cases, the fixed data and cross-section data at the front of the runtape file are preserved.

4. The fifth entry *dmmp* has several possible meanings. For sequential non-KCODE non-batch MCNP6, a value of $dmmp \leq 0$ results in TFC entries every 1000 particles initially. This value doubles to 2000 after 20 TFC entries. A positive value of *dmmp* produces TFC entries every *dmmp* particles initially. For non-KCODE non-batch distributed memory multiprocessing, $dmmp < 0$ produces TFC entries and task rendezvous every 1000 particles initially, the same as does the sequential version. The default value, $dmmp = 0$, produces ten TFC entries and task rendezvous, rounded to the nearest 1000 particles, based on other cutoffs such as NPS, CTME, etc. This selection optimizes speedup in conjunction with TFC entries. If detectors/DXTRAN are used with default Russian roulette criteria (DD card default), the $dmmp = 0$ entry is changed by MCNP6 to $dmmp < 0$, ensuring tracking with the sequential version (i.e., TFC entries and rendezvous every 1000 particles). As with the sequential non-KCODE non-batch version, $dmmp > 0$ produces TFC entries and task rendezvous every *dmmp* particles, even with detectors/DXTRAN with default Russian roulette criteria. Setting *dmmp* to a large positive number minimizes communication time and maximizes speedup. However, the TFC may not have many entries, possibly only one, if $dmmp = npp$.

5. The rendezvous frequency of a multiprocessor calculation is the minimum interval of parameters or *ndp*, *ndm*, and *dmmp*.

5.13.6 PTRAC: Particle Track Output

The PTRAC card generates an output file of user-filtered particle events referred to as a particle track file. Adding a FILE = HDF5 entry will produce an HDF5 output file [Appendix D.3], but the raw binary format (FILE = BIN) is the default for backward compatibility in this release [DEP-53382]. The HDF5 output file is easier to post process and allows for parallel execution with MPI, tasks, or both, as detailed in §5.13.6.1. The default file name **ptrac** (or **ptrac.h5**) can be changed on the execution line or within the message block.

Use of one or more card keywords will reduce the particle track file size significantly. The card keywords are organized into three categories: output-control keywords, event-filter keywords, and history-filters keywords. The output-control keywords provide user control of the output file and I/O. The event-filter keywords filter particle events on an event-by-event basis, i.e., only events that meet all event-filter criteria are written to the output file. The history-filter keywords will filter all particle events for a particular history. That is, if the entire history meets the filter criteria, all filtered events for that history are written to file. Restarted calculations that utilize the PTRAC feature will not change the results in the original particle track output file and cannot be used to generate additional PTRAC card results, but other aspects of the simulation will be completed. Simulations with unique random number seeds (5.13.2) can be used to generate separate files that can be processed as unique histories if required.

The output formats for PTRAC and event logs limit the printing of cell, surface, and material numbers to a maximum of five characters. Users intending to use the PTRAC card with the ASCII format or event logs should avoid the use of cell, surface, or material numbers greater than 99,999.

Deprecation Notice **DEP-53382**

The FILE keyword options ASC, AOV, BIN, and BOV for the PTRAC card are deprecated, and the HDF5 option should be preferred. The MAX, BUFFER, and WRITE keywords are also deprecated, as they are unused by the HDF5 option. The ICL and JSU event filter options are deprecated, as they are replaced with the simpler SUR and CEL event filter options for HDF5 outputs.

Deprecation Notice **DEP-53383**

The COINC keyword and the EVENT=CAP options for the PTRAC card are deprecated. There is no current plan to support these features with the HDF5 format in future releases. People interested in continuing to use these features should send an email to mcnp_help@lanl.gov.

Data-card Form: `PTRAC keyword=value(s)...`

Output Control Keywords

`BUFFER`	Except for `FILE` = `HDF5`, determines the amount of storage available for filtered events *within each history*. Single integer entry. (DEFAULT: `BUFFER` = 100) Restriction: `BUFFER` > 0.
`FILE`	Controls file type [DEP-53382]. If

`FILE` = `HDF5`	generates an HDF5 output file (recommended).
`FILE` = `ASC`	generates an ASCII output file.
`FILE` = `BIN`	generates a binary output file. (DEFAULT)
`FILE` = `AOV`	generates an ASCII output file by overwriting an existing ASCII particle track file to a named pipe on UNIX systems. Requires a particle track file to exist prior to execution.
`FILE` = `BOV`	generates a binary output file by overwriting an existing binary particle track file to a named pipe on UNIX systems. Requires a particle track file to exist prior to execution.

`FLUSHNPS`	Controls write frequency for HDF5 output file type. Single integer entry. Restriction: `FLUSHNPS` > 0. For non-`KCODE` simulations (3), events will be written to the HDF5 particle track file at least every `FLUSHNPS` histories. See §5.13.6.2 for guidance.
`MAX`	Sets the maximum number of events to write to the particle track file. If `FILE` = `HDF5`, this entry is ignored. Single integer entry. (DEFAULT: `MAX` = `10000`) Restrictions: `MAX` ≠ 0. The value of |`MAX`| will be truncated to $2^{31} - 1$ if it is larger than $2^{31} - 1$. If

`MAX` > 0	write `MAX` events to the particle track file.
`MAX` < 0	MCNP6 is terminated when |`MAX`| events have been written to the particle track file.

MEPH	Determines the maximum number of events per history to write to the particle track file. Single integer entry. (DEFAULT: write all events) Restriction: MEPH > 0
WRITE	Controls what particle parameters are written to the particle track file. (1) If FILE = HDF5, all parameters are always written. Otherwise, if
WRITE = POS	write only the (x, y, z) location of the particle with related cell and material numbers. (DEFAULT)
WRITE = ALL	write the (x, y, z) location of the particle with related cell and material numbers and the (u, v, w) direction cosines, as well as particle energy, weight, and time.
COINC	Activates a particle track file format specifically for coincidence tally scoring [DEP-53383]. Used in conjunction with TALLY keyword. (4) If
COINC = COL	a full printing of all specified tally scores is produced, even if the tally scores were zero. The output is column-based. (DEFAULT)
COINC = LIN	tally score pairs are printed for non-zero scores only.

Event Filter Keywords

EVENT	Specifies the type of events written to the particle track file. Up to six mnemonic entries can be specified. (DEFAULT: write all events) If
EVENT = SRC	write initial source events.
EVENT = BNK	write bank events. These include secondary sources (e.g., photons produced by neutrons, as well as particles created by variance-reduction techniques such as DXTRAN and energy splitting). See Appendix D.3 for a complete list and more details on bank events.
EVENT = SUR	write surface events.
EVENT = COL	write collision events.
EVENT = TER	write termination events.
EVENT = CAP	write coincident capture events [DEP-53383]. (5)
FILTER	Specifies additional MCNP6 variables for filtering each event (2). The parameter values consist of one or two numerical entries and a variable mnemonic that corresponds to a variable in the PBL derived structure or other related quantities. See Table 5.28 for available mnemonics. A single numerical entry requires an exact value; two numerical entries represent a range. When a range is specified, the first entry must be less than or equal to the second. Multiple sets of numerical entries and mnemonics are allowed. (DEFAULT: no additional filtering) Examples:

- FILTER=2,ICL writes only those events that occur in cell 2.

- `FILTER=0,10,X` writes only those events in which the particle's x coordinate is between 0 and 10 cm.
- `FILTER=0.0,10.0,X 0,1,U 1.0,2,ERG` writes only those events in which the particle's x coordinate is between 0 and 10 cm and the particle's x-axis cosine is between 0 and 1 and the particle's energy is between 1 and 2 MeV.

`TYPE`	Filters events based on one or more particle types. (DEFAULT: Write events for all particles.) May specify filtering of a single particle or multiple particles, where `<pl`$_i$`>` is a particle identifier specified in Table 4.3: TYPE=`<pl`$_1$`>`,`<pl`$_2$`>`, ...

History Filter Keywords

`NPS`	Sets the range of particle histories for which events will be output. A single value produces filtered events only for the specified history. Two entries indicate a range and will produce filtered events for all histories within that range. The first entry must be less than or equal to the second. (DEFAULT: Events for all histories) Note: An 8-byte integer is permitted for keyword `NPS`. Restriction: `NPS` > 0. Examples:

- `NPS=10` write events only for particle number 10.
- `NPS=10,20` writes events for particles 10 through 20.

`CELL`	List of cell numbers to be used for filtering histories. If any track enters a listed cell(s), all filtered events for the history are written to the particle track file. Note: Number of entries is unlimited Restriction: `CELL` > 0. (DEFAULT: No filtering based on cell entrance.) Example:

- `CELL=1,2` writes all filtered events for those histories that enter cell 1 or 2.

`SURFACE`	List of surface numbers to be used for filtering histories. If any track crosses a listed surface(s), all filtered events for the history are written to the particle track file. Note: Number of entries is unlimited. Restriction: `SURFACE` > 0. (DEFAULT: No filtering based on surface crossing.)
`TALLY`	List of tally numbers to be used for filtering histories. If any track contributes to the TFC bin of listed tallies, all filtered events for the history are written to the particle track file. (See the `TF`*n* card for specification of the TFC bin for tally *n*.) Note: A negative `TALLY` entry indicates that the corresponding `VALUE` entry is a multiplier rather than an absolute value. Number of entries is unlimited. Restriction: `TALLY` $\neq 0$. Only positive `VALUE` entries are allowed when `FILE` = `HDF5`. (DEFAULT: No filtering based on tally contribution.) Example:

- `TALLY=4` writes all filtered events for those histories that contribute to tally 4. (See `VALUE` keyword for control of filter criteria.)

VALUE	Specifies the tally cutoff above which history events will be written. The number of entries must equal the number of entries on the TALLY keyword. A negative TALLY value indicates that the corresponding VALUE entry is a multiplier (6). (DEFAULT: VALUE = 0.0 for each tally associated with the TALLY keyword). Examples:

- TALLY=4 VALUE=2.0 writes all filtered events of any history that contributes 2.0 or more to the TFC bin of tally 4.
- TALLY=-4 VALUE=2.0 writes all filtered events of any history that contributes more than $2.0 \times T_a$ to tally 4, where T_a is the average tally of the TFC bin. The values for T_a are updated every *dmmp* histories (see the PRDMP card).
- TALLY=4 VALUE=0.0 writes all filtered events of every history that scores to tally 4.

Default: Using the PTRAC card without any keywords causes all particle events to be written to the particle track output file. Caution: If all particle events are written to the particle track file, an extremely large file likely will be created unless NPS is small.

Use: Optional.

Details:

1. For the HDF5 particle track file there is a single format, so the WRITE keyword is ignored.
2. For FILE = BIN and FILE = ASC, event-based filters specified with the FILTER keyword are not applied to BNK events. For FILE =HDF5, BNK events are subject to filters listed on the FILTER entry.
3. For FILE = HDF5 and KCODE simulations, the FLUSHNPS keyword is not required. Data is written to the particle track file at the end of each cycle.
4. The COINC feature only supports the TALLY and VALUE keywords as filter options. When used in conjunction with the COINC keyword, TALLY entries must be positive. The existing VALUE keyword can be used to set threshold scores for the tallies itemized on the TALLY keyword. All scores below that threshold are treated as zero. The COINC keyword will force ASCII file output format (FILE = ASC or FILE = AOV).
5. For EVENT =CAP, most of the standard PTRAC capabilities are bypassed (for speed) and the data written to each line (or record) of the particle track file are very different from the usual event data. For binary files, the entries on each PTRAC line include the particle history number ("NPS"), the time from source event to analog capture in any FT8 CAP tally ("Time"), and the cell number in which the analog capture occurred ("Cell"). Additionally, for ASCII files, a fourth column, "Source," provides the source particle number of a given history. The 5th column of output provides the target identifier of the spontaneous fission nuclide. A value of 0 appears if the source is not spontaneous fission, i.e., PAR =SF. The 6th column contains the target identifier [§1.2.2] of the first induced fission; the 7th, that of the second induced fission; the 8th, that of the third induced fission; and the 9th, that of the last fission before capture, either induced or spontaneous.
6. Filtering based on the T_a values will occur only when they become non-zero. Thus, when using a multiplier, PTRAC events may not be written for several thousand particles, or at all, if scores are seldom or never made to the TFC bin of the specified tally. In most cases, it is best to enter an absolute value.

Table 5.28: Mnemonic Values for the FILTER Keyword

Mnemonic	MCNP6 Variable	Description
X	pbl%r%x	x coordinate of particle position (cm)
Y	pbl%r%y	y coordinate of particle position (cm)
Z	pbl%r%z	z coordinate of particle position (cm)
U	pbl%r%u	Particle x axis direction cosine
V	pbl%r%v	Particle y axis direction cosine
W	pbl%r%w	Particle z axis direction cosine
ERG	pbl%r%erg	Particle energy (MeV)
WGT	pbl%r%wgt	Particle weight
TME	pbl%r%tme	Time at the particle position (shakes)
VEL	pbl%r%vel	Speed of the particle (cm/shake)
IMP1	pbl%r%fiml(1)	Neutron cell importance
IMP2	pbl%r%fiml(2)	Photon cell importance
IMP3	pbl%r%fiml(3)	Electron cell importance
SPARE1	pbl%r%spare(1)	Spare banked variable
SPARE2	pbl%r%spare(2)	Spare banked variable
SPARE3	pbl%r%spare(3)	Spare banked variable
ICL	pbl%i%icl	Problem number of current cell (7)
CEL	ncl(pbl%i%icl)	User specified number of current cell (7)
JSU	pbl%i%jsu	Problem number of current surface (7)
SUR	nsf(pbl%i%icl) + 0.1kfq	User specified number of current surface or macrobody (7)
IDX	pbl%i%idx	Number of current DXTRAN sphere
NCP	pbl%i%ncp	Count of collisions for current branch
LEV	pbl%i%lev	Geometry level of particle location
III	pbl%i%iii	1st lattice index of particle location
JJJ	pbl%i%jjj	2nd lattice index of particle location
KKK	pbl%i%kkk	3rd lattice index of particle location

(7) The filter options for ICL and JSU are the program numbers stored by the MCNP6 program and are numbered by the order the cells and surface appear in the input file, respectively; these options are not allowed if FILE = HDF5. The options for CEL and SUR are the numbers specified by the user in the input file; these options are only allowed with FILE = HDF5.

The particle track file will contain the heavy ion particles and their track information, but not individual heavy ion identities.

5.13.6.1 Using PTRAC with Parallel Execution

The HDF5 particle track format can be used with MPI parallelism and shared-memory parallelism (i.e., the **tasks** command line option), both independently and combined. The use of the PTRAC card with multiple MPI processes requires an MPI-parallel HDF5 installation. Check the MCNP6 build documentation for recommended MPI implementations and versions to avoid past bugs in the parallel writing features. The **tasks** option can be used with any MCNP installation and does not require an MPI installation.

For optimal performance, the MPI-parallel HDF5 library is intended to be used on a distributed parallel file system, e.g., a Lustre file system. Local file systems typically provide acceptable performance as well.

Caution

Some network file systems (e.g., NFS) are not compatible with parallel access and MPI-parallel PTRAC simulations may take a long time to write and close the file. When using multiple MPI worker processes, it is recommended to run a simulation with a small value of NPS and ensure that the simulation is completed and the file written successfully. If parallel writing is unsuccessful or inefficient for the available file systems, then the tasks option can be used to achieve parallel performance, which does not require an MPI-parallel HDF5 distribution or parallel file system.

When a simulation is performed with tasks greater than 1, there is no guarantee of the order histories will be written to the output file; events within a history are always ordered correctly. Because histories are statistically independent, the history ordering only concerns users interested in a particular NPS value or in reproducing the exact history order of an equivalent serial simulation. To reproduce the order of a serial simulation, the entries in the **RecordLog** dataset should be sorted by NPS (i.e., the unique history identifier) with the order of events within each history preserved [Appendix D.3].

5.13.6.2 Guidance for the FLUSHNPS keyword

An important difference from FILE=BIN and FILE=ASC is that HDF5 PTRAC simulations will buffer event data into memory across multiple histories and only periodically write to a file, which improves performance. Users must specify how often the data is written through the FLUSHNPS keyword if FILE = HDF5. The buffered data will be written to the output file and buffers emptied at least every FLUSHNPS histories. If the value is set too low, then the cost of writing and (optionally) MPI communication becomes expensive. However, if the value is set too high, memory access can become slower and the simulation may crash if it runs out of memory. If the simulation crashes due to an out of memory error, it will print a message to the terminal that an instance of a 'std::bad_alloc' error has occurred, followed by additional errors resulting from the simulation crashing.

To help choose a value for FLUSHNPS, after the first write to the particle track file during the simulation, an estimate of the peak memory usage for *all* MPI processes and shared-memory tasks is printed to the screen. Data is written to the particle track file at every rendezvous, including those that can be controlled

by the PRDMP card, so the peak memory estimate may be written at a value of NPS that is less than the FLUSHNPS value. For typical simulations and computer systems currently available, a peak memory usage of 1000 MB between writes provides a sufficient balance of performance and safety. As a general rule, try setting a value of FLUSHNPS=1E05 and running a short test simulation. If the printed peak memory usage is less than 1000 MB, then the FLUSHNPS number is sufficiently low for typical simulations. Scaling the FLUSHNPS value to use 1000 MB or larger between writes may increase performance. When in doubt, prefer a smaller value of FLUSHNPS to avoid 'std::bad_alloc' errors later in the simulation.

An optimal value of FLUSHNPS depends on many parameters, including the problem (in particular, the amount of memory used by material and geometry data), the number of filters active, and the volatile memory available on the system. As an example of fine-tuning performance, a simulation with EVENT = SRC can set a value of FLUSHNPS = 5E06 and use less than 1000 MB between writes. For a neutron simulation of a bare 5 cm sphere of 2.25 g/cm^3 graphite that writes all events from all histories, a value of FLUSHNPS = 1E06 will use about 1000 MB between writes. Some problems that write all events with extreme amounts of particle splitting, multiplication, surface crossings, or collisions, will require setting a much lower value, such as FLUSHNPS = 1000 or lower to avoid crashing. In such cases, it is preferable to try to add additional filtering to reduce the amount of memory used when possible. The memory usage by particle track buffers between writes is mostly independent of the number of MPI ranks and shared-memory **tasks**, for a particular value of FLUSHNPS. However, using less MPI ranks than available computational cores on a system reduces the memory used by geometry and cross section data, which can allow for larger values of FLUSHNPS and more efficient particle track simulations.

5.13.6.3 Example 1

```
PTRAC FILE=HDF5 EVENT=SUR,COL TYPE=N,P FLUSHNPS=1E05
```

This input line will generate an HDF5 particle track output file that contains all surface and collision events that occurred during the simulation, for photons and electrons. Data is flushed to the output file at least every 100,000 histories.

5.13.6.4 Example 2

```
PTRAC  FILTER=8,9,ERG  EVENT=SUR  NPS=1,50  TYPE=E  CELL=3,4
```

When multiple keywords are entered on the PTRAC card, the filter criteria for each keyword must be satisfied to obtain an output event. This input line will write only surface crossing events for 8–9-MeV electrons generated by histories 1–50 that have entered cells 3 or 4.

5.13.7 MPLOT: Plot Tally While Problem is Running

The MPLOT card specifies a plot of intermediate tally results that is to be produced periodically during the calculation.

Data-card Form: MPLOT *keyword=value(s)...*

The entries on the MPLOT card are MCPLOT commands [§6.2.4.1] for one picture.

Default: None.

Use: Optional. The specification of 8-byte integer values is allowed for FREQ.

During the calculation, as determined by the FREQ *n* keyword entry, MCRUN will call MCPLOT to display the current status of one or more of the tallies in the problem. If a FREQ *n* command is not included on the MPLOT card, *n* will be set to 5000. The following MCPLOT commands cannot appear on the MPLOT card: RMCTAL, RUNTPE, DUMP, and END. All of the commands on the MPLOT card are executed for each displayed picture, so coplots of more than one bin or tally are possible. No output is sent to a **COMOUT** file. MCPLOT will not take plot requests from the terminal; it returns to MCRUN after each plot is displayed. See §6.3.3 for a complete list of MCPLOT commands available.

A second way to plot intermediate tally results is to use the TTY interrupt Ctrl + c MCPLOT or Ctrl + c , m that allows interactive plotting during the calculation. At the end of the history that is running when the interrupt occurs, MCRUN will call MCPLOT, which will take plot requests from the terminal. No output is sent to the **COMOUT** file. The following commands can not be used: RMCTAL, RUNTPE, DUMP, and END.

5.13.8 HISTP: Create LAHET-compatible Files

The results of particle transport, and medium- and high-energy physics interactions from within the LAHET Code System are available through the LAHET-compatible **histp** files using the HISTP card in the MCNP input file. This card controls the writing of information to an external **histp** file useful for analysis by the **HTAPE3X** program distributed with MCNPX or the **HTAPE6** source code distributed with the MCNP6.2 source code.

Caution

The **HTAPE6** utility code is no longer supported and is thus no longer distributed with current or future versions of the MCNP6 code. Furthermore, the HISTP card will be marked for deprecation in a future version of the MCNP6 code once a suitable alternative capability is available that provides equivalent history information to that encapsulated within the **histp** file. People interested in continuing to use either the MCNP-generated **histp** files or the **HTAPE6** utility, or would like to provide input to the development team on the future direction of the replacement capabilities, should contact the development team by sending an email to mcnp_help@lanl.gov.

Data-card Form: HISTP *icl*1 ... *iclK*

iclK	List of cell numbers. Only events occurring within these cells will be written to the **histp** file. If no *iclK* values are provided, all events will be written. Negative values are unused.

Default: All events in all cells written to the **histp** file.

Use: Optional.

Limitations:

- No heavy ion transport information is written to the **histp** file aside from the usual recoils from which the heavy ions are started.
- Writing **histp** files during multiprocessing is unavailable.

5.13.8.0.1 Example

Listing 5.62: example_histp.mcnp.inp.txt

```
histp  11
```

The **histp** file will contain only events within cell 11.

5.13.9 PIO: Enable Parallel IO

Data-card Form: PIO *value*

value	If	
	value = blank or ON,	the code is built with parallel HDF5 support, and is run with MPI, features that have parallel IO support will use it.
	value = NO,	then parallel IO is disabled. (DEFAULT)

Default: NO. This option is not saved in the runtape. It must be specifically enabled for all runs, continue or otherwise.

Use: Certain components of the code (such as FMESH with the XDMF output format) can write results using parallel HDF5. This can be extremely useful on parallel file systems when writing very large results arrays. Enabling this feature will perform input and output in parallel in these circumstances.

This feature is not enabled by default as not all file systems will benefit from parallel IO. Some file systems will even cause the MCNP code to lock up if parallel IO is used. In testing, NFS partitions would often cause this. As a result, one should test file systems with short simulations to see if this will work and provide a benefit to a simulation before running a large problem.

5.13.10 READ: Auxiliary Input File and Encryption

The MCNP6 READ card enables

1. the reading of parts of the input file from other (auxiliary) files,
2. the suppression of the printing of the auxiliary input files to shorten output files and protect proprietary information, and
3. the encryption of auxiliary input files to protect proprietary information.

Unlike most MCNP6 input cards, there may be as many READ cards and auxiliary input files as desired. The READ card may appear anywhere after the title card of an MCNP6 input file but not in the middle of a card continuation. READ cards may appear in auxiliary files, allowing the nesting of READ cards to multiple levels. The encryption capability may be applied to any or all of the READ levels. There is no limit to the number of nested levels.

The encryption capability can be used to protect proprietary designs of tools and other systems modeled with MCNP6. The encryption capability is localized in subroutine ENCRYPT. The MCNP6 scheme is very simple; therefore, it protects nothing. To protect input, the subroutine should be modified to a more sophisticated scheme known only to those producing the data and only executable MCNP6 versions should be provided to users of the encrypted files.

Form: READ *KEYWORD=value(s) ...*

FILE=*filename*	Causes input from the file filename to be inserted after the READ card in the MCNP6 input deck.
NOECHO	Suppresses printing in the output file of the input cards that follow the READ card.
ECHO	Resumes echoing in the output file of the input after a NOECHO keyword was given in a previous READ card. Echoing also will resume when the next READ card is encountered without the NOECHO keyword. (DEFAULT)
DECODE=*password*	Allows reading of an encrypted file. When DECODE is invoked, the encrypted input file is not echoed, and many default print tables are turned off (and cannot be turned back on) to protect the data in the encrypted file.
ENCODE=*password*	Allows the writing of an encrypted file.

5.13.10.1 Example 1

```
READ FILE=filename NOECHO
```

Because the echoing of the input cards also is resumed when an "end of file" is encountered, this example causes the input from the auxiliary file, *filename*, to be suppressed. After the file *filename* is read, input transfers back to the input file that contains the READ card and printing is no longer suppressed.

5.13.10.2 Example 2

```
READ DECODE password FILE=filename
```

This example causes the reading of the encrypted file, *filename*.

5.13.10.3 Example 3

```
READ ENCODE password FILE=filename
```

This example causes an encrypted file, *filename*, to be written.

5.13.11 DBCN: Debug Information

Caution

Former MCNPX users need to be aware that several DBCN inputs may be required to invoke MCNPX default behavior. In particular, please see DBCN parameters x_{38}, x_{39}, and x_{60}.

The entries on this card are used primarily for debugging problems and the code itself. The first 12 entries can be changed in a restarted calculation, which is useful for diagnosing troubles that occur late in a long-running problem.

⚠ Caution

The `DBCN` card is intended for MCNP developers. It should be applied with extreme caution and a thorough understanding of the side effects.

Data-card Form: `DBCN` x_1 x_2 ...

x_1	Obsolete - do not use. The random number used for starting the transport of the first particle history in a run. Required: Use the RAND card with the SEED keyword. (1.)
x_2	Debug print interval. Print out information about every x_2^{th} particle. The information consists of a) the particle history number, b) the total number of collisions, c) the total number of random numbers generated, and d) the random number at the beginning of the history. This information is printed at the beginning of the history and is preceded by the letters DBCN in the output to aid in a pattern search. (DEFAULT: x_2=0)
x_3,x_4	History number limits for event-log printing. Event-log printing is done for histories x_3 through x_4, inclusively. The information includes a step-by-step account of each history, such as where and how a particle is born, which surface it crosses and which cell it enters, what happens to it in a cell, etc. (Note: The output formats for event logs limit the printing of cell, surface, and material numbers to a maximum of five characters (i.e., identifying numbers $\leq$99,999). See x_{11}. (DEFAULT: x_3=0 and x_4=0)
x_5	Maximum number of events per history in the event log. (DEFAULT: x_5=600)
x_6	Detector/DXTRAN underflow limit. (See Note 3.) (DEFAULT: x_6=80.0) Restriction: $50 \leq x_6 \leq 200$ If the attenuation factor, λ, to the detector or DXTRAN sphere is $>x_6$, then the score is terminated as "underflow in transmission."
x_7	Useful only to MCNP6 code developers.
	x_7=0 — no print from the volume and surface area calculations is produced. (DEFAULT)
	$x_7 \neq 0$ — a detailed print from the volume and surface area calculations is produced.
x_8	Obsolete - do not use. Causes the starting random number of the problem to be that which would normally start the x_8^{th} history. That is, causes the x_8^{th} history to be the first history of a problem for debugging purposes; can also be used to select a random number sequence different from that in an identical problem to compare statistical convergence. Required: Use the RAND card with the HIST keyword. (See Note 1.)
x_9	Defines the distance allowed between coincident repeated-structures surfaces for them still to be considered coincident. (DEFAULT: 10^{-4}) A value of 10^{-30} reproduces the earlier treatment where coincident repeated structure surfaces were not allowed. The parameter x_9 should not have to be changed

unless geometries have dimensions greater than 10^5 or unless surfaces at different levels are intended to be closer than 10^{-4}.

x_{10} Specifies the half-life threshold for stable nuclides (DEFAULT: 1.5768×10^{16} s)

x_{11} If

x_{11}=0,	collision events are not printed in event logs for lost particles. (DEFAULT)
$x_{11}\neq$0,	the collision lines in the lost-particle event log are printed.

x_{12} Expected number of random numbers for this calculation. Entering x_{12} will cause the last line of the output file to print x_{12} and the actual number of random numbers used so that a quick comparison can be made to see if two problems tracked each other. DEFAULT: x_{12}=0, i.e., test ignored)

x_{13} Obsolete - do not use. Random number stride. Required: Use the RAND card with the STRIDE keyword. (1.)

x_{14} Obsolete - do not use. Random number multiplier. Required: Use the RAND card with the GEN keyword. (1.)

x_{15} If

x_{15}=0,	the usual selection of statistical quantities is printed (DEFAULT)
$x_{15}\neq$0,	the shifted confidence interval and the variance of the variance for all tally bins are printed. An extra line of tally output is created for each tally that contains non-zero information. The shifted confidence interval center is followed by the estimated VOV. If the tally mean and relative error (RE) are all zeros, the VOV line is not printed because it is all zero also. Changing x_{15} from non-zero to zero in a restarted calculation will cause the VOV information not to be printed. The parameter x_{15} cannot be changed from zero to non-zero in a restarted calculation.

x_{16} Scale the history score grid for the accumulation of the empirical f(x) in print table 161 and 162. MCNP6 uses a logarithmically spaced history score grid in print table 161 for f(x), producing a straight line for f(x) on a log-log plot for $1/x^n$ behavior, covering 60 decades of unnormalized tally magnitudes from 1E-30 to 1E30. This range can be multiplied by the x_{16} entry when the range is not sufficient. A negative entry means that negative history scores will be accrued in the score grid f(-x) and the absolute value of x_{16} will be used as the score grid multiplier. Positive history scores will then be lumped into the lowest bin with this option. This scaling can be done only in the original problem, not in a restarted calculation. (DEFAULT: x_{16}=1.0)

x_{17} If

x_{17}=0,	use default angular treatment for partial sub steps to generation sites of secondary particles. This treatment accounts for the probability of the delta function first, then interpolates in the cosine of the deflection angle. It does not preserve the plane in which the deflection angle will lie at the end of the full sub step. (DEFAULT)
x_{17}>0,	use alternate angular treatment for secondary generation. The cosine of the electron angle is interpolated and the end-of-sub step plane is preserved, but the changing probability of the delta function along the sub step is ignored. This option is preserved for further testing of angular algorithms because results have been known to be sensitive to these details.
x_{17}<0,	use MCNP4A treatment of electron angles at secondary generation sites.

x_{18} Controls the energy-indexing algorithm for electron transport related to bin interpolation.

If x_{18}=0,	use "MCNP-style" energy-indexing algorithm; also called the "bin-centered" treatment. (Used by MCNPX.)
If x_{18}=1,	use Integrated Tiger Series (ITS)-style energy-indexing algorithm; also called the "nearest group boundary" treatment.
If x_{18}=2,	use detailed Landau straggling sampling logic, also called the "energy- and step-specific" treatment. Required for single-event electron transport. (DEFAULT)

x_{19} In use by MCNP6 developer(s) to study quadratic polynomial interpolation. [DEFAULT (x_{19}=0) provides current model.]

x_{20} Unused.

x_{21} Unused.

x_{22} Unused.

x_{23} If

x_{23}=0,	use pulse-height tally variance reduction trees if variance reduction is present, otherwise do not use PHT VR trees. (DEFAULT)
x_{23}=1,	force pulse-height tally variance reduction trees whether they are needed or not.
x_{23}=-1,	do no use pulse-height tally variance reduction trees.

x_{24} Controls grazing contribution cut-off for surface flux tallies. If

x_{24}=0,	$\lvert mu_{cut} \rvert$ = 0.001. (DEFAULT)

	$x_{24}\neq 0$,	$\lvert mu_{cut}\rvert = x_{24}$
x_{25}	Unused.	
x_{26}	Unused.	
x_{27}	Controls antiparticle promotion. If	
	$x_{27}=0$,	do not promote antiparticles. (DEFAULT)
	$x_{27}=1$,	promote antiparticles (affects MODE card and certain tallies); lumps particle and antiparticle pairs under one particle type. (Used in MCNPX.) (Certain restrictions may apply.)
x_{28}	Bank size. (DEFAULTs vary by application: $x_{28}=2048$ for most fixed-source problems, $x_{28}=128$ for criticality problems, $x_{28}=16384$ for high-energy problems)	
x_{29}	Unused.	
x_{30}	Unused.	
x_{31}	Unused.	
x_{32}	If	
	$x_{32}=0$,	normal GENXS behavior. (DEFAULT)
	$x_{32}\neq 0$,	use internal bremsstrahlung spectrum generation with CEM and LAQGSM models for GENXS.
x_{33}	If	
	$x_{33}=0$,	do not apply an additional interpolation/smoothing method to stopping powers for heavy ions. (DEFAULT)
	$x_{33}\neq 0$,	apply an additional interpolation/smoothing method to stopping powers for heavy ions.
x_{34}	Used to reproduce a bug in μ^--induced gammas. [DEFAULT ($x_{34}=0$) is to use the corrected code.]	
x_{35}	If	
	$x_{35}=0$,	causes slight (arbitrary) spreading of nuclear excitation during μ^- capture. (DEFAULT)
	$x_{35}\neq 0$,	turns off slight (arbitrary) spreading of nuclear excitation during μ^- capture.
x_{36}	If	
	$x_{36}=0$,	use user-provided data for μ^--induced gamma rays, if available. (DEFAULT)
	$x_{36}\neq 0$,	use older data (literature or MUON/RURP) previously hard-coded in MCNPX.
x_{37}	Set minimum of internal bremsstrahlung spectrum for CEM and LAQGSM in GENXS when $x_{32}\neq 0$. (DEFAULT: $x_{37}=30$ MeV)	

x_{38}	If
	x_{38}=0, use Barashenkov/Polanski data file BARPOL2001.dat. (DEFAULT)
	$x_{38}\neq$0, use older BARPOL.dat data file from 1996.
x_{39}	Controls the default S(α,β) smoothing behavior, which was present in MCNPX but not in MCNP5. If
	x_{39}=0, use default S(α,β) sampling treatment, as in MCNP5 (DEFAULT).
	$x_{39}\neq$0, use MacFarlane/Little sampling, as in MCNPX.
x_{40}	Controls writing of MCPLIB and **xsdir** lines
x_{41}	Controls printing printing photon/electron data
x_{42}	If
	x_{42}=0, use default method for model cross sections. (DEFAULT)
	$x_{42}>$0, use original MCNPX model cross-section method.
	$x_{42}<$0, use earlier MCNP6 method (MARS coding).
x_{43}	Control photon form-factor interpolation. If
	x_{43}=0, use linear form-factor interpolation. (Used by MCNPX.)
	x_{43}=2, use best method for form-factor interpolation. (DEFAULT) Currently the best method is logarithmic inversion or log-log.
x_{44}	For developers: to study coherent scattering in isolation. (DEFAULT: x_{44}=0, all processes)
x_{45}	If
	x_{45}=0, use MCNP6 elastic scattering method. (DEFAULT)
	$x_{45}\neq$0, use earlier MCNPX elastic scattering method.
x_{46}	If
	x_{46}=0, use default CEM-to-LAQGSM photonuclear energy boundary.
	x46>0, set x_{46} as CEM-to-LAQGSM energy boundary.
x_{47}	If
	x_{47}=0, use CLEM model for cosmic-ray spectra.(DEFAULT)
	$x_{47}\neq$0, use Lal model for cosmic-ray spectra.
x_{48}	If
	x_{48}=0, allow MCNP6 to forbid threading when not suitable. (DEFAULT)

	$x_{48}\neq 0$,	insist on threading if requested.
x_{49}	If	
	$x_{49}=0$,	perform normal input checking. (DEFAULT)
	$x_{49}>0$,	expert user option to skip some lattice input checking for very large problems to save time in initialization.
x_{50}	Modifies printing the tally fluctuation chart (TFC). Controls printing of the relative fractional uncertainty (i.e., the "error") and the VOV. If	
	$x_{50}=0$,	do traditional printing of tally fluctuation charts. (DEFAULT)
	$x_{50}=1$,	provide the relative fractional uncertainty and VOV in scientific notation and decrease the printing of three side-by-side TFCs to two side-by-side TFCs.
	$x_{50}=2$,	is the same as $x_{50}=1$ but also prints more digits in the mean column.
x_{51}	Used to turn off all photon-induced fluorescence. (Default is to have photon-induced fluorescence active.)	
x_{52}	Used to turn off Compton-induced relaxation. Applies to fluorescence and Auger electrons. (Default is have Compton-induced relaxation active.)	
	Set	$x_{52}=1$, to invoke MCNPX functionality in emission of Auger electrons.
x_{53}	If	
	$x_{53}=0$,	use new ENDF photoelectric relaxation data, if available. (DEFAULT)
	$x_{53}\neq 0$,	use traditional photoelectric fluorescence; i.e., use limited pre-ENDF/B VI.8 treatment. Applies to fluorescence and Auger electrons.
x_{54}	Controls sampling method for ENDF Law 9. If	
	$x_{54}=0$,	use traditional sampling for first 10^8 tries but then use new, improved sampling method. (DEFAULT)
	$x_{54}\neq 1$,	use new, improved sampling method.
x_{55}	Spontaneous decay integration time. Default is 20 s which includes ≈ 20 decay levels, or ≈ 1 s per decay level. Complex decay chains may require an increase in this parameter [§5.8.1, (12)].	
x_{56}	Unused.	
x_{57}	Unused.	
x_{58}	Unused.	
x_{59}	Unused.	
x_{60}	If	

	x_{60}=0,	print number of calls to each high-energy model. DEFAULT
	x_{60}≠0,	also include successes for each model.
x_{61}	Models of knock-on electron angles. (DEFAULT=0)	
x_{62}	Used to debug single-event electrons excitation energy loss. (DEFAULT=0)	
x_{63}	Unused.	
x_{64}	To debug single-event electrons angular deflection for knock-on electrons. (DEFAULT=0)	
x_{65}	To debug single-event ionization and treat deflection for incident particles. (DEFAULT=0)	
x_{66}	To control single-event bremsstrahlung photon angles. (DEFAULT=0)	
x_{67}	Controls number of particle histories (NPS) for first calculation of the average contribution per history for point detectors and DXTRAN spheres for Russian roulette game. If	
	x_{67}=0,	use TFC value of NPS for first calculation of detector or DXTRAN average contribution. (DEFAULT)
	x_{67}>0,	use the first x_{67} particles to determine the average contribution per history for point detectors and DXTRAN spheres for Russian roulette game.
x_{68}	Unused.	
x_{69}	Used to increase the LJA array size, which stores the surfaces bounding the cells. Only needed when a fatal error occurs and the MCNP code advises the user to "Set dbcn(69) to increase mlja > [...]". dbcn(69) sets `mlja`, which controls the size of the LJA array.	
x_{70}	Debug choice of some interaction models. (DEFAULT=0)	
x_{71}	If	
	x_{71}=0,	allow model photonuclear capability. (DEFAULT)
	x_{71}≠0,	prohibit model photonuclear capability.
x_{72}	If	
	x_{72}=0,	explicit log-log interpolation in ELXS_MOD. (DEFAULT)
	x_{72}≠0,	random linear interpolation.
x_{73}	Unused.	
x_{74}	Unused.	
x_{75}	If x_{75}≠0, print extra info for F-matrix calculations.	
x_{76}	If x_{76}≠0, print array storage info after setup.	
x_{77}	If x_{77}≠0, specify number of bins for hash-based cross-section searches. DEFAULT is 8192.	

x_{78}	For developers: 0 for old 6.1 $S(\alpha,\beta)$ method, 1 for new.
x_{79}	If x_{79}=0, use MT=101 for PTRAC absorption and MT=18 for fission. $x_{79}\neq$0, use MT=2 for absorption and fission.
x_{80}	Unused.
x_{81}	0 uses linear interpolation of electron elastic scatter and 1 uses log-log interpolation within a data table.
x_{82}	0 uses linear interpolation of electron elastic scatter and 1 uses log-log interpolation between data tables.
x_{83}	0 uses linear interpolation for electron partial x-s and 1 uses log-log interpolation.
x_{84}	0 uses linear interpolation for electron bremsstrahlung energy and 1 uses log-log interpolation within a data table.
x_{85}	0 uses linear interpolation for electron bremsstrahlung energy and 1 uses log-log interpolation between data tables.
x_{86}	0 uses linear interpolation for electron excitation energy and 1 uses log-log interpolation.
x_{87}	0 uses linear interpolation for electron knock-on energy and 1 uses log-log interpolation within a table.
x_{88}	0 uses linear interpolation for electron knock-on energy and 1 uses log-log interpolation between tables.
x_{89}	0 uses linear interpolation for electron ionization x-s and 1 uses log-log interpolation.
x_{90}	If $x_{90}\neq$0, set maximum number of terms for the Goudsmit-Saunderson distribution (3). DEFAULT is 240. If DBCN(90) $<$ 240, the number of terms for the Goudsmit-Saunderson distribution will be set to DEFAULT due to the limitation in the data.
x_{91}	If $x_{91}>$0, set the minimum ROC curve count value to x_{91}.
x_{92}	If $x_{92}>$0, set the maximum ROC curve count value to x_{92}.
x_{93}	Unused.
x_{94}	Unused.
x_{95}	Unused.
x_{96}	Unused.
x_{97}	Unused.
x_{98}	Unused.
x_{99}	Unused.
x_{100}	0 uses new coincident-surface method and 1 uses old method.

Use: Optional. All DBCN parameters allow 8-byte entries.

Details:

1. Settings for the random-number-generator parameters are now accomplished using the RAND card. The DBCN entries 1, 8, 13, and 14 were used long ago, but are no longer permitted. Setting these entries on the DBCN only (and not the RAND card) is a fatal error.
2. The contributions neglected because of underflow are typically insignificant to the final answer. However, in some cases, the underflow contribution is significant and necessary. When DXTRAN spheres for point detectors are used to get tally contributions for generating weight windows, sometimes these underflow contributions cannot be neglected. If DXTRAN or detector underflow is significant in the calculation, generally there are serious problems, such as not sampling enough collisions near the detector. Changing the underflow limit should be done only with extreme caution.
3. Setting the number of terms for the Goudsmit-Saunderson distribution can stabilize the underlying angular deflection distributions used in transport, yielding improved simulation results [332]. However, the increase in the number of terms for the Goudsmit-Sauderson distribution is only valid for electron energies greater than 0.256 MeV. For electrons with energies less than, the number of terms for the Goudsmit-Sauderson distribution is set to DEFAULT.

5.13.12 LOST: Lost Particle Control

The LOST card allows the user to increase the number of lost particles the code will allow before terminating.

Data-card Form: LOST *lost1 lost2*

lost1	Number of particles that can be lost before the calculation terminates with BAD TROUBLE. (DEFAULT: *lost1*=10)
lost2	Maximum number of debug prints that will be made for lost particles. (DEFAULT: *lost2*=10)

Defaults: 10 lost particles and 10 debug prints.

Use: Discouraged. Losing more than 10 particles is rarely justifiable.

The word "lost" means that a particle gets to an ill-defined section of the geometry and does not know where to go next. This card should be used cautiously: the user should know why the particles are being lost and the number lost should be statistically insignificant out of the total sample. Even if only one of many particles gets lost, there could be something seriously wrong with the geometry specification. Geometry plots in the area where the particles are being lost can be extremely useful in isolating the reason that particles are being lost.

5.13.13 IDUM: Integer Array

The IDUM integer array is in the **MCNP_DEBUG.F90** module and is available to the users. IDUM is included in the dumps on the restart file and therefore can be used for any purpose, including accumulating information over the entire course of a problem through several restarted calculations. The array is declared as Fortran **integer(4)** type, so it provides 32 bits of precision.

Data-card Form: IDUM i_1 i_2 ... i_K	
i_k	Any user-assigned integer value where $1 \leq k \leq K = 2000$.

Default: All array values zero.

Use: Useful only in user-modified versions of MCNP6.

Details:

1. Up to 2000 entries can be provided to fill the IDUM array with integer numbers. If floating-point numbers are entered, they will be truncated and converted to integers.

5.13.14 RDUM: Floating-Point Array

The RDUM floating-point array is in the **MCNP_DEBUG.F90** module and is available to the users. RDUM is included in the dumps on the restart file and therefore can be used for any purpose, including accumulating information over the entire course of a problem through several restarted calculations. The array is declared as Fortran **real(8)** type, so it provides 64 bits of precision.

Data-card Form: RDUM r_1 r_2 ... r_K	
r_k	Any user-assigned floating-point value where $1 \leq k \leq K = 2000$.

Default: All array values zero.

Use: Useful only in user-modified versions of MCNP6.

Details:

1. Up to 2000 entries can be provided to fill the RDUM array with floating-point (real) numbers.

5.13.15 ZA, ZB, ZC, and ZD: Developers Card Placeholders

The ZA, ZB, ZC, and ZD cards are made available to advanced user-developers who wish to construct their own input cards in MCNP6. Similar to the use of IDUM and RDUM, source code that is modified by users to create a modified version of MCNP6 no longer carries the extensive validation and verification the original LANL-created source and executables do. Users must perform their own verification and validation to ensure their modifications have not had adverse effects on existing capabilities.

5.13.16 FILES: File Creation

Data-card Form: FILES *unit_no filename access form record_length*

unit_no	Recommendation: *unit_no* > 100. (DEFAULT: none)
filename	Name of the file. (DEFAULT: none)
access	Options are SEQUENTIAL or DIRECT access. (DEFAULT: SEQUENTIAL)
form	Options are FORMATTED or UNFORMATTED. (DEFAULT: FORMATTED if SEQUENTIAL access has been specified, UNFORMATTED if DIRECT access has been specified.)
record_length	Record length in direct access file. (DEFAULT: not required if SEQUENTIAL access has been specified, no default if DIRECT access has been specified.)

Use: When a user-modified version of MCNP6 needs files whose characteristics may vary from calculation to calculation. Not allowed in restarted calculations.

Details:

1. If this card is present, the first two entries are required and must not conflict with existing MCNP6 units and files. Setting *unit_no* greater than 100 and less than 1,000 will likely prevent any conflicts with MCNP6 unit numbers during input reading and output writing. A file unit conflict may occur if the user-defined file is both accessed during particle transport, and the sum of 60 and the number of parallel execution threads requested by the user (e.g. *tasks* on the command line) are equal to the user-specified file unit number.
2. The words SEQUENTIAL, DIRECT, FORMATTED, and UNFORMATTED can be abbreviated. The maximum number of files allowed is six, unless the second dimension of the **KUFIL** array in **FIXCOM.F90** is increased and the **UFILES.F90** subroutine is updated appropriately.

Caution

The names of any user files in a restarted calculation will be the same as in the initial calculation. The names are not automatically sequenced if a file of the same name already exists; therefore, a second output file from a restarted calculation will overwrite and replace the content of an existing file of the same name. If you are using the FILES card for an input file and restart the calculation, you will have to provide the coding for keeping track of the record number and then positioning the correct starting location on the file when you continue or MCNP6 will start reading the file at the beginning.

5.13.16.1 Example 1

```
FILES    21  ANDY  S  F  0    22  MIKE  D  U  512
```

5.13.16.2 Example 2

```
FILES    17  DUMN1
MCNP6  INP=TEST3  DUMN1=POST3
```

If the file name is **DUMN1** or **DUMN2**, the user can optionally use the execution line message to designate a file whose name might be different from run to run, for instance in a restarted calculation.

5.13.17 DISABLE: Disable MCNP Features

The DISABLE card allows a user to deliberately disable certain features of the MCNP code that otherwise run by default. This can be useful for problems approaching the resource limitations of the computer running it.

Form: DISABLE *[options]*

NUCLIDE_ACTIVITY_TABLE
: If this option is present, this disables the computation of PRINT Table 140 completely. Subsequent restart runs will be unable to print this table out. This option is useful to reduce the memory usage of problems with very large numbers of materials and nuclides per material.

Chapter 6

MCNP Geometry and Tally Plotting

MCNP6 has two plotting capabilities. The first, **PLOT**, is used to plot two-dimensional slices of the problem geometry specified in the INP file. The user can perform interactive geometry plotting in two ways: either "point-and-click" mode or "command-prompt" mode. In addition, generation of plot files can be done in batch mode using a command file. The second plotting capability, **MCPLOT**, plots tally results produced by MCNP6 and cross-section data used by MCNP6. Mesh tallies may be plotted either in **MCPLOT** from `mctal` files or superimposed over geometry plots in **PLOT** from `runtpe` files.

Section 6.1 addresses system issues external to MCNP6 related to graphics. Section 6.2 discusses how to invoke the **PLOT** features, whereas §(6.3) discusses the **MCPLOT** features. An explanation of each set of input commands is given. Lines the user will type are shown in `typewriter` font. The Enter key must be pressed after each input line. Although in this section plot options and keywords are shown in UPPER CASE, they are case insensitive.

6.1 System Graphics Information

X Windows is the only graphics system supported by MCNP6. This graphics library is device-independent in general and gives considerable flexibility in processing graphical output.

The X-window graphics library (http://www.x.org) allows the user to send/receive graphics output to/from remote hosts as long as the window manager on the display device supports the X protocol [e.g., OpenLook window manager, MOTIF window manager, Cygwin (PC Windows), etc.]. Before running MCNP6, perform the following steps to use these capabilities. Note that these steps use UNIX C-shell commands. Other shells may require different syntax.

1. On the host that will execute MCNP6, enter **setenv DISPLAY** *displayhost:0* where *displayhost* is the name of the host that will receive the graphics. If the *displayhost* is the same as the execution host (*executehost*), set **DISPLAY** to `localhost:0` or just `:0`.
2. If the two hosts are different, in a CONSOLE window of the display host enter **xhost** *executehost* where *executehost* is the name of the host that will execute MCNP6.

With the **setenv** or the **xhost** command, the host IP address can be used in place of the host name. For example, **setenv DISPLAY** *128.10.3.1:0*. This option is useful when one remote system does not recognize the host name of another.

Note to LANL Users: On some systems, including the Los Alamos Integrated Computing Network (ICN) and other LANL local area networks, use of the **xhost** command is strongly discouraged. This is because it creates a security problem. In place of using **xhost**, the secure shell (SSH) can be used to log into remote

hosts and provide X Windows forwarding. This is considered to be more secure, and it handles setting the **DISPLAY** variable for the user. If SSH is used, do not manually set **DISPLAY** as this will interfere with the secure forwarding. On local systems (where *displayhost* and *executehost* are the same), this warning does not apply.

6.2 The Geometry Plotter, PLOT

The geometry plotter is used to plot two-dimensional slices of a problem geometry specified in the INP file. This feature of MCNP6 is invaluable for debugging geometries. You should first verify your geometry model with the MCNP6 geometry plotter before running the transport part of MCNP6, especially with a complicated geometry where it is easy to make mistakes. The time that is required to plot the geometry model is small compared with the potential time lost working with an erroneous geometry.

6.2.1 PLOT Input and Execute Line Options

To plot geometries with MCNP6, enter the following command:

MCNP6 IP INP=*filename KEYWORD=value(s)*

where IP stands for "initiate and plot." The allowed keywords are:

NOTEK	Suppress plotting at the terminal and send all plots to the graphics metafile, PLOTM. The NOTEK keyword is used for production and batch situations or when the user's terminal has no graphics capability.
COM=*filename*	Use file *filename* as the source of plot requests. When an end-of-file (EOF) is read, control is transferred to the terminal. In a production or batch situation, end the file with an END command to prevent transfer of control. Never end the COM file with a blank line. If COM is absent, the terminal is used as the source of plot requests.
PLOTM=*filename*	Name the graphics metafile *filename*. The default name is **PLOTM**. For some systems this metafile is a standard postscript file and is named **plotm.ps**. Unique names for the output file, PLOTM, will be chosen by MCNP6 to avoid overwriting existing files.
COMOUT=*filename*	Write all plot requests to file *filename*. The default name is **comout**. PLOT writes the COMOUT file in order to give the user the opportunity to do the same plotting at some later time, using all or part of the old COMOUT file as the COM file in the second run. Unique names for the output file, COMOUT, will be chosen by MCNP6 to avoid overwriting existing files.

The most common method of plotting is with an interactive graphics terminal. First, MCNP6 will read the input file and perform the normal checks for consistency, then the interactive point-and-click geometry plotting window will appear in its own window.

When X Windows is in use, the plot window supports a variety of interactive features that assist the user in manipulating the plot. The interactive options are discussed after the discussion of the command-line plot options.

When names are defaulted, unique names for the output files, PLOTM and COMOUT, will be chosen by MCNP6 to avoid overwriting existing files. Unique names are created by changing the last letter of the default name

until the next available name is found. For example, if the file **plotm.ps** already exists, MCNP6 tries the name **plotn.ps**, etc., until it finds an available name.

MCNP6 can be run in a batch environment without much difficulty, but the user interaction with the plotter is significantly reduced. When not using an interactive graphics terminal, use the NOTEK option on the MCNP6 execution line or set TERM=0 along with other PLOT commands when first prompted by PLOT. Setting NOTEK will prevent a blank window from appearing prior to the first PLOT command being entered. In systems with no X Windows support, using NOTEK will prevent the MCNP6 code from returning an error that there is no display available. In the interactive mode, plots can be sent to the graphics metafile with the FILE keyword. See the keyword description in §6.2.4 for a complete explanation. The **plotm.ps** file is a postscript file that can be sent to a postscript printer. Every view plotted will be put in a postscript file called plot?.ps where ? begins at M and goes to the next letter in the alphabet if **plotm.ps** exists.

6.2.2 Geometry Plotting Basic Concepts

Before describing the individual plotting commands, it is helpful to understand some basic mechanics of two-dimensional (2-D) plotting. To obtain a 2-D slice of a geometry, one must decide where the slice should be taken and how much of the slice should be viewed in the plotting window. The slice is a 2-D plane that may be arbitrarily oriented in space; therefore, the first problem is to decide the plane position and orientation.

In an orthogonal three-dimensional coordinate system the three axes are perpendicular to each other. An orthogonal axis system is defined with a set of BASIS vectors on the 2-D plane used to slice the geometry to determine the plot orientation. The first BASIS vector is in the horizontal direction of the screen. The second BASIS vector is the vertical direction on the screen. The surface normal for the plane being viewed is perpendicular to the two BASIS vectors and directed out of the screen towards the viewer.

For example, the BASIS vectors that define a view of the x-y plane (or "down" the z-axis) are 1, 0, 0 and 0, 1, 0. The x-axis view can be mirrored by changing the vectors to: -1, 0, 0 and 0, 1, 0. This would cause the x-axis values to decrease from left to right while the y-axis values increase from bottom to top as before. A complete mirror on both axes can be obtained by setting the basis to -1, 0, 0 and 0, -1, 0. Usage of BASIS and other commands referenced in this section are discussed in §6.2.4.1.4.

The default BASIS vectors define views of the y-z, z-x, and x-y planes, which are generally sufficient for viewing geometry. However, if required, the flexibility of the BASIS command can be used to examine the geometry along any desired slice. Arbitrarily oriented basis vectors are defined with component magnitudes that need not be normalized (e.g. 0, 1, 1 is functionally equivalent to 0, 2, 2). If an angle between the vector and an axis is known, the sine or cosine of the angle can be used to determine the magnitudes. A few decimal places of precision will often suffice.

The center of the view plane may be set with the ORIGIN command. For example, on a y-z plot, the x coordinate of ORIGIN sets the "depth" that the slice is viewed from and the y and z coordinates translate the view in that slice. Because planes are infinite and only a finite area can be displayed at any given time, the extent of the cross-sectional plane displayed can be specified with the EXTENT command. For instance, on a y-z plot at an ORIGIN of x1, y1, z1, the y-z plane is viewed at a depth of x = x1, and it is centered at y1and z1. If EXTENT y2 z2 is entered, the plot displayed would have a horizontal extent from y1 - y2 to y1 + y2 and a vertical extent of z1 - z2 to z1 + z2. Thus EXTENT may be used to zoom the view of the plot slice in or out.

All the plot parameters for the MCNP6 plotter have defaults. In command-line mode, respond to the first MCNP6 prompt with Enter to obtain a default plot; in the interactive mode, click on the plot area of the interactive screen. The default geometry plot is a PX plane centered at 0, 0, 0 with an extent of $-100 < y < 100$ and $-100 < z < 100$. The y-axis will be the horizontal axis of the plot, and the z-axis will be the vertical axis. Surface labels are printed. In command-prompt mode, this default is the equivalent of entering the following command line:

```
ORIGIN 0 0 0 EXTENT 100 100 BASIS 0 1 0 0 0 1 LABEL 1 0
```

By manipulating selected plot parameters, any arbitrary 2-D plot can be obtained. Most parameters remain set until they are explicitly changed either by the same command with new values or by a conflicting command.

⚠ Caution

Placing the plot plane exactly on a surface of the geometry is not a good idea.

For example, if the input geometry has a `PX` plane at `x = 0`, that plane coincides with the default plot plane. Several results can occur:

1. Some portion of the geometry may be displayed in dotted lines, which usually indicates a geometry error (even if there is none in this case).
2. Some portion of the geometry may simply not show up at all.
3. Very infrequently the code may crash with an error.

To prevent all these unpleasantries, move the plot plane some tiny amount away from surfaces. The terminal will show a warning when the plot plane is coplanar with a geometry plane.

6.2.3 Interactive Geometry Plotting in Point-and-click Mode

The geometry plotter supports interactive point-and-click plotting for all systems with X Windows graphics [§6.1]. The plot area is active at all times when the interactive plotter is enabled. However, it is not active when the command-line interface is in use (e.g., requested via the `Plot>` button) except for text commands that need mouse input from the plot window such as the `LOCATE` command [§6.2.4.1]. This command requires a mouse click in the plot area to provide the intended terminal output. Figure 6.1 shows an example geometry plot window with the interactive controls outlined. The controls are separated into left, right, top, and bottom menus. The plot area itself is also active. An explanation of the point-and-click commands in each control menu follows.

6.2.3.1 Top Menu: Translate and Zoom Functions

`UP RT DN LF`	When clicked, these buttons move the plot frame one full window upwards (`UP`), to the right (`RT`), downwards (`DN`), or to the left (`LF`).
`ORIGIN`	After clicking, the user can click in the plot geometry to set the origin and center the view at the clicked point.
`.1 .2 Zoom 5. 10.`	The zoom command requires a minimum of two mouse clicks: The first click on the zoom scale selects a discrete zoom factor between 0.1 and 10 for the current plot. The selected scale factor is displayed above the "Click here or picture or menu" box in the lower-left of the plot window. A second mouse click on the same scale factor will zoom at that factor centered at the current plot origin.

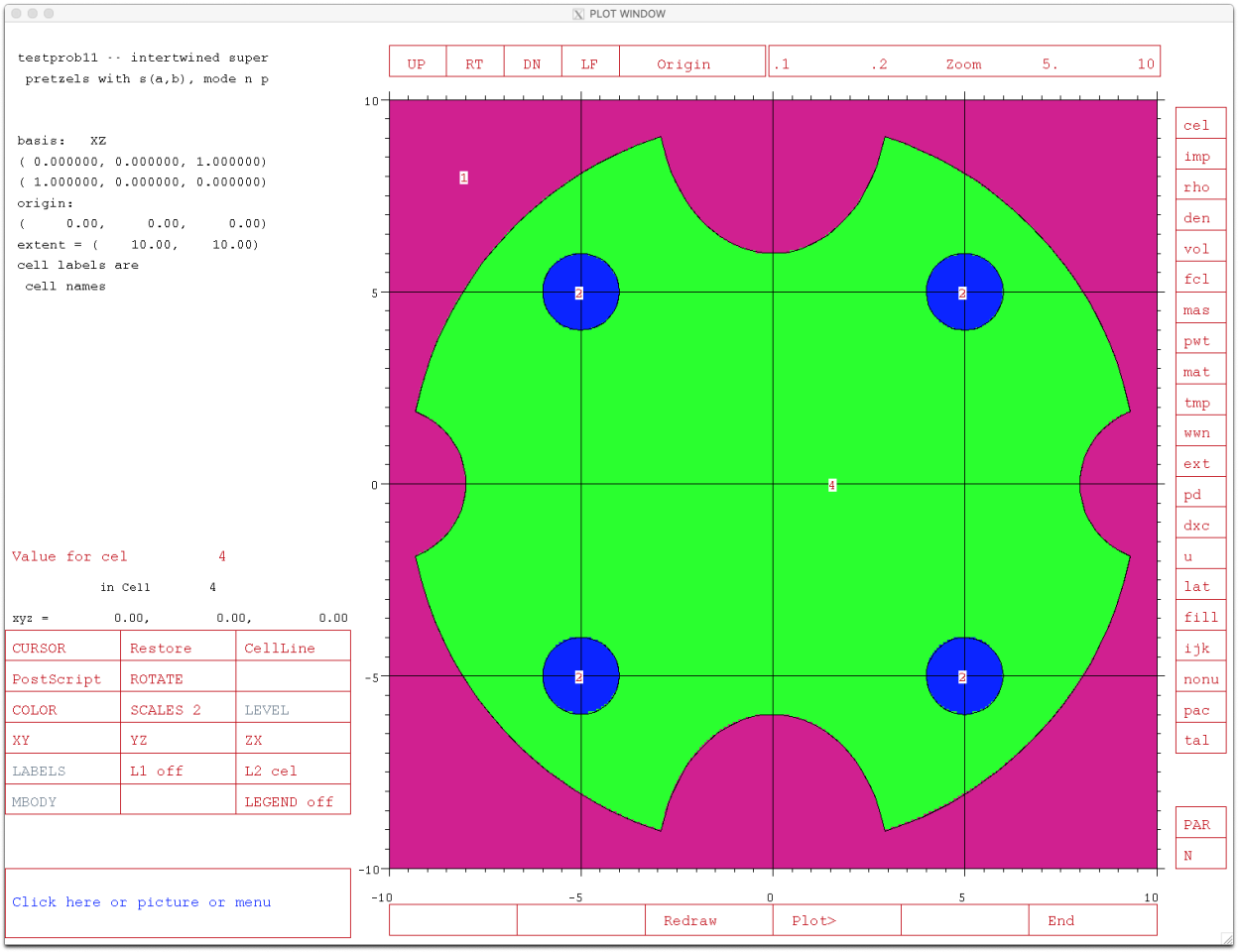


Figure 6.1: Geometry Plot Window Interactive Plotting Controls

If the second mouse click is on a different scale factor, it then counts as a new "first click."

If the second click occurs in the plot geometry, the origin is set to that point and the zoom occurs about this new origin.

Hint: To effectively cancel a zoom command, click the `Zoom` label (which corresponds to a 1× zoom) on the scale twice.

6.2.3.2 Left Menu: What is Plotted and How

`(Hidden button)` A "hidden" button resides in the upper left quadrant of the plot window and triggers a redraw. If your plot window appears blank when exposed, click in the upper left of the screen to refresh it. On some systems, the entire plot window may appear blank if resized or minimized and then restored. Just click this hidden button to redraw the window if this happens. This button is equivalent to the `Redraw` button [§6.2.3.4], but is easier to find when the window is blank.

`Value for` *`var`* While not a control, this area immediately above the lower left control box provides useful information:

By default, *`var`* is "mat" and provides information on the material number under the cursor. The line under the `Value for` *`var`* always lists the current cell the cursor is in. Without any other button clicked, one can click through the geometry and see the cell number and value attached to *`var`* change. The *`var`* can take any of the parameters listed on the right menu bar [§6.2.3.3]. This provides an easy way to query parameters like density or cell importance. The last line before the control box gives the coordinates of the current cursor location.

`CURSOR` Clicking this button activates the cursor-region selector; the cursor changes shape to appear like the upper left corner of a box. Click in the plot window at a point representing the upper left spatial boundary of the desired plot. The cursor will change shape again; now click the lower right position of the desired plot. The plot will be redrawn using the new boundaries and keep a 1:1 aspect ratio. This is equivalent to an `EXTENT` command and an `ORIGIN` command.

`RESTORE` This button restores the view to the previous "frame." For example, if `CURSOR` is used to zoom on a portion of the geometry from a full view, `Restore` can be used to return to the full view without having to enter `ORIGIN` and `EXTENT` commands. It is a single level "undo" button for the plotted geometry.

`CellLine` Toggles among available line modes:

- `CellLine` Plot constructive solid geometry cells, outlined in black. (DEFAULT)
- `No Lines` Plot cells not outlined in black.
- `WW MESH` Plot weight-window superimposed mesh without cell outlines.
- `WW+Cell` Plot weight-window superimposed mesh and cell outlines.
- `WWG MESH` Plot weight-window generator mesh.
- `WWG+Cell` Plot weight-window generator mesh and cell outlines.
- `MeshTaly` Plot `TMESH` mesh tally boundaries.
- `MT+Cell` Plot `TMESH` mesh tally boundaries and cell outlines.

The `CellLine` and `No Lines` options are always available. `WW MESH` and `WW+Cell` are available only when the `WWP` card calls for using a superimposed weight-window mesh (5th entry negative) and a *`WWINP`* file is provided. `WWG MESH` and `WWG+Cell` are available only when a `MESH` card appears in the input and when the `WWG` card requests superimposed mesh generation (2nd entry is 0). `MeshTaly` and `MT+Cell` are available only when a `TMESH` mesh tally has been requested.

Note: After a line mode is chosen, click `REDRAW` from the bottom menu to force the selected mesh and/or cell lines to be drawn on the plot.

`PostScript`

When clicked, writes the *next* plot to a postscript file (default name, `plotm.ps`). The image in the postscript file is of far higher quality than a screenshot of the plot window because it is a vector graphic. To create a postscript image of the current plot, activate the `PostScript` button and then click `Redraw` from the bottom menu. After one plot is written to the postscript file, the `PostScript` button is reset.

The function of the `PostScript` button is equivalent to the non-interactive command `FILE` with no argument; i.e., only the next plot is written to the file.

`ROTATE`

Toggles between two modes: `ROTATE on` and `ROTATE off`.

`ROTATE on` interchanges the first two basis vectors, resulting in a 90° rotation of the plot around the "off-basis" axis (e.g., the z-axis in an x-y plot). Whether the rotation is clockwise or counter-clockwise depends on the initial basis vectors.

Note: After a rotation mode is chosen, click `Redraw` from the bottom menu to force the plot to be redrawn with new orientation.

`COLOR` *`var`*

Toggles colors on and off.

Color shading of geometry plots may be on a variety of cell parameters.

By default, *`var`* registers the cell parameter `mat`, which indicates that plot colors are assigned to materials. By toggling the `COLOR` button, all color can be turned off, presenting only a line drawing after a redraw of the plot (which can substantially improve plotter speed).

Alternatively, the cell parameter on which the plot color scheme is based can typically be changed to any parameter in the right margin control menu appropriate to the problem [§6.2.3.3]. To change the parameter, click a cell parameter from the right menu, click the `COLOR` button to turn off color, then click `COLOR` again to reactivate it. The new selected cell parameter will now register as *`var`*.

For example, to color by cell density, one would click the `den` parameter on the right menu, then click `COLOR` so "`COLOR off`" appears. Clicking the "`COLOR off`" button again will show "`COLOR den`". A click of the `Redraw` button will display the new colors.

Note: Any changes in plot colors require a redraw of the plot via the `Redraw` button.

`SCALES`

Toggles among three scale modes on each click:

- If `SCALES` is set to 0, no scale is provided on the plot (DEFAULT).
- If `SCALES` is set to 1, dimensional scales for both horizontal and vertical axes are provided; and
- If `SCALES` is set to 2, dimensional scales for both horizontal and vertical axes with an associated grid are provided.

The values drawn on the axes are the distance from the origin of the current plot, i.e., they go from *`-EXTENT`* to *`+EXTENT`* in the two directions.

Note: after the scale mode is chosen, click `Redraw` from the bottom menu to force the scales to be drawn on the plot.

LEVEL	Toggles through universe levels in repeated-structure geometry. If there are no sublevels, then the LEVEL button is not active. The button label identifies the level to be plotted if levels are present in the input. Requires that the Redraw command from the bottom menu be clicked to create the revised plot.
XY, YZ, ZX	Alter plot perspective to corresponding planar combinations: • The XY command sets the basis to (1 0 0 0 1 0); • The YZ command sets the basis to (0 1 0 0 0 1) (DEFAULT); and • The ZX command sets the basis to (0 0 1 1 0 0). In all cases, the origin is unchanged.
LABELS, L1, L2	Controls the status of surface and cell labels. If L1 is set to *sur*, then surface labels are displayed (DEFAULT). If L1 is set to *off*, then surface labels are not displayed. If L2 is set to *off*, then cell labels are not displayed (DEFAULT). If L2 is set to *var*, then the cell parameter, *var*, is the cell label. To change the cell parameter, click one from the right menu, then click the L2 button to change the label type to the new selection. A change in state of any LABELS parameter requires a redraw of the plot to update the view.
MBODY	Toggles labeling of macrobody facets. If MBODY is on and if L1 of the LABEL command is set to *sur*, then general macrobody surface numbers are displayed (DEFAULT). If MBODY is off and if L1 is set to *sur*, then macrobody facet numbers for each macrobody surface are displayed. A change in state of MBODY requires the plot be redrawn to update the surface labels.
FMESH	Cycle through mesh tallies. Does not change plot layout. Only present if FMESH tallies exist in the input. A change in state of FMESH requires that Redraw be clicked from the bottom menu to display the revised plot.
LEGEND	When activated, displays a contour plot legend for a mesh tally. The legend will display the association of the color key to the numerical values in the plot.
Click here or picture or menu	Clicking in this area changes the button to show "Enter Data>" and requires the user to enter a plot command. A list of commands is in §6.2.4.1. Up to 29 characters representing one or more plot commands can be entered. Pressing Enter accepts the command string. If the command line is terminated with an "&", the "Enter Data>" prompt remains and another command (or a continuation of a long command) can be entered. A line that does not end with an "&" sends the command(s) and triggers a redraw when Enter is pressed. An example of the usefulness of this feature is entering a specific origin to center the plot at. Where clicking the Origin button in the top menu and clicking the plot is convenient, if more precision is required (such as tracking geometry errors leading to lost particles), a user should click the Click here... box and type something like "origin 10.725 -2.663 1.004". Note: for extended access to the command-line interface, use the Plot> option in the bottom menu to pass control to the terminal window.

6.2.3.3 Right Menu: Parameter Choices for Labels, Colors, etc.

The right menu is used to set the variables used for for cell labels and geometry coloring. After clicking a button in the right menu, the left menu `L2` or `COLOR` button must be set to the new parameter. A redraw must be triggered before the plot is updated with the new labels and/or colors. Some right-menu options work for both colors and labels such as `cel`. Other options only work as labels.

`cel`	Cell labels/colors will be cell numbers.
`imp`	Cell labels will be importance by particle type.
`rho`	Cell labels/colors will be atom densities ($\text{barn}^{-1} \cdot \text{cm}^{-1}$).
`den`	Cell labels/colors will be mass densities (g/cm^3).
`vol`	Cell labels will be volumes (calculated or user-supplied, cm^3).
`fcl`	Cell labels will be forced-collision fraction by particle type.
`mas`	Cell labels will be masses (g).
`pwt`	Cell labels will be photon production weights.
`mat`	Cell labels/colors will be material numbers (DEFAULT for `COLOR` variable, *`var`*).
`tmp`	Cell labels/colors will be temperature (MeV) for time index 1: `TMP`1.
`wwn`	Cell labels/colors will be weight windows for energy or time index *`N`* (or combined energy-time index *`N`*): `WWN`*`N`*:$\mathscr{P}$. Note: When combining time and energy binning, the index *`N`* varies as follows: For 3 time bins and 3 energy bins, the index, *`N`*, would map to the following sequence: (T1, E1), (T1, E2), (T1, E3), (T2, E1), (T2, E2), (T2, E3), (T3, E1), (T3, E2), (T3, E3).
`ext`	Cell labels will be exponential transform stretching parameter by particle type.
`pd`	Cell labels will be detector contribution frequency fraction by particle type.
`dxc`	Cell labels will be DXTRAN contribution frequency fraction.
`u`	Cell labels will be universe numbers.
`lat`	Cell labels will be the enclosed lattice type.
`fill`	Cell labels will be filling universe identification numbers.
`ijk`	Cell labels will be lattice indices.
`nonu`	Cell labels will be fission turnoffs.
`pac`	When the interactive plotter is called from `MCPLOT`, cell labels will be values of columns in `PRINT` Table 126. Use the `par` and `N` buttons to toggle particle and column respectively. The columns shown by this button are:

`pac1`:$\mathscr{P}$	The labels are the "tracks entering" column.
`pac2`:$\mathscr{P}$	The labels are the "population" column.
`pac3`:$\mathscr{P}$	The labels are the "collisions" column.
`pac4`:$\mathscr{P}$	The labels are the "collisions * weight (per history)" column.
`pac5`:$\mathscr{P}$	The labels are the "number weighted energy" column.

pac6:$\mathscr{P}$	The labels are the "flux weighted energy" column.
pac7:$\mathscr{P}$	The labels are the "average track weight (relative)" column.
pac8:$\mathscr{P}$	The labels are the "average track mfp (cm)" column.

tal	Used for plotting TMESH tally results when PLOT is called from MCPLOT. See §6.4.3 for usage.
par	Selects particle type for cell quantities that have particle-specific values (e.g., imp:$\mathscr{P}$). Click the par button then another button in the right menu to toggle the particle type. Prior to redrawing the plot, the associated LABELS or COLOR button should be clicked twice to toggle from the current particle to "off" then back to the desired parameter with the updated particle type.
N	Selects a numerical index for cell quantities or mesh-based weight-window bins that have indexed values. Updating the index for a displayed parameter follows the same process described in the description of par. Example: WWN3:P would provide photon weight windows in the 3rd energy group and be selected by clicking wwn, par, and N. Note: For both the par and N buttons, clicking these with a relevant parameter selected will show the change in particle or index in the "Value for *var*" information box. This is useful for keeping track of the index or particle currently selected.

6.2.3.4 Bottom Menu: Commands

Redraw	Triggers a redraw of the plot.
Plot>	Passes control to the command-line window enabling traditional plot commands to be entered. Once in the command-line mode, control can be returned to the interactive plotter with the command INTERACT. Note: For brief text commands, use the Click here ... button to type up to 29-character text commands.
End	Terminates the plot session and exit PLOT.

6.2.4 Interactive Geometry Plotting in Command-prompt Mode

Invoking the geometry plotter through the command-line interface offers more flexibility for combining commands when compared to the point-and-click interactive plotter. Command-line-interface entry of commands can be invoked in two ways:

1. The non-interactive plotter can be called with the NOTEK keyword as described in §6.2.1 and results will be viewable in the plot?.ps file. The user can open the X Window plotter after execution with the PLOT command TERM 1.
2. The interactive plotter is started and the user clicks the Plot> button in the bottom menu of the interactive window (§6.2.3.4). This transfers command entry to the terminal window with the results of the command visible in the interactive window. In this mode the user can still use the interactive plotter buttons. For more information on this interactive environment, see §6.2.3. The user may return to the point-and-click interactive mode by entering the command INTERACT at the terminal prompt.

A plot request consists of a sequence of commands terminated by pressing the Enter key. A command consists of a keyword that is usually followed by some parameters. A plot request line cannot have more than 128 characters on a single line, but lines can be continued by typing an `&` (ampersand) before pressing the Enter key. However, each keyword and its parameter(s) must be complete on one line. The `&` character can be used in the input `COM` file [§6.2.7] as well as at the `PLOT` prompt. Keywords and parameters are blank-delimited with commas and equal signs interpreted as blanks. Numbers can be entered in free-form format and do not require a decimal point for floating-point data. Keywords and parameters remain in effect until they are explicitly changed. The commands `OPTIONS`, `HELP`, and `?` display a complete list of keywords.

Keywords can be abbreviated by shortening them to any degree as long as they are not ambiguous and are spelled correctly. If a shortened keyword is ambiguous, the entire command string will be rejected and the terminal will warn that an ambiguous command was used. An example of an ambiguous keyword would be "`O`". It is unclear if `O` refers to `ORIGIN` or `OPTION`, thus another character is required to differentiate it. Parameters following keywords can not be abbreviated.

6.2.4.1 `PLOT` Commands

6.2.4.1.1 Device-control Commands

Normally `PLOT` draws plots to a system's X Window display. By using the following commands, the user can specify that plots not be drawn to the display and/or that they be sent to a graphics metafile or PostScript file for processing later by a graphics utility program.

`TERM` *n*	Output device type is specified by *n*. *n*=`0` for a terminal with no graphics forwarding capability (for a system without the X Window System). No plots are drawn to a display window, and all plots are sent to the graphics metafile. `TERM 0` is equivalent to putting `NOTEK` on MCNP6's execution line [§6.2.1]. *n*=`1` restores the plotting window on the next plot request.
`FILE` *aa*	Send or do not send plots to the graphics metafile `PLOTM.PS` according to the value of the parameter *aa*. The graphics metafile is not created until the first `FILE` command is entered. `FILE` has no effect in the `NOTEK` or `TERM 0` cases. The allowed values of *aa* are the following:

aa is blank	Only the current plot is sent to the graphics metafile.
aa=`ALL`	The current plot and all subsequent plots are sent to the metafile until another `FILE` command is entered.
aa=`NONE`	The current plot is not sent to the metafile nor are any subsequent plots until another `FILE` command is entered.

`VIEWPORT` *aa*	Make the viewport rectangular or square according to the value of *aa*. This option does not affect the appearance of the plot. It only determines whether the area around the plot is padded for a legend, scales, and interactive controls. If *aa*=`RECT`, allow space beside the plot for a legend and around the plot for scales. (DEFAULT) If *aa*=`SQUARE`, the legend area, the legend, and scales are omitted. Note: Use of the `SQUARE` option disables the interactive-window plotter capability.

6.2.4.1.2 General Commands

`&`	Continue reading commands for the current plot from the next input line. The `&` must be the last character on the line. The `&` command must not break another command and its parameters onto two lines; instead, it is used to continue long user command strings on new lines.
`INTERACT`	Return to the interactive point-and-click geometry plotter interface. This command is used to return from the terminal-command interface when the `Plot>` button is clicked or the command `PLOT` is entered in the "`Click here or ...`" box while in the interactive plotter.
`RETURN`	If `MCPLOT` was called during investigation of geometry with `PLOT` (via the `ip` execution option), control returns to `PLOT`. Otherwise `RETURN` has no effect.
`MCPLOT`	Call the `MCPLOT` tally and cross-section plotter [§6.3]. For tally results, a `RUNTAPE` file or `MCTAL` must be read [§6.3.1.1 and §6.3.3.4].
`PAUSE` *n*	Can be used on any line of a plot command file that is specified with the execute `COM=`*filename* option [§6.2.1]. Holds each view for *n* seconds. If no *n* value is provided, each view remains until Enter is pressed. When absent, the commands specified in the command file will run sequentially until the end of the command file is reached at which point control returns to the terminal.
`END`	Terminate execution of `PLOT`. Closes any open X Windows and returns the terminal from the `PLOT` prompt to a standard system shell prompt.

6.2.4.1.3 Inquiry Commands

The following commands print information to the terminal.

`OPTIONS` or `?` or `HELP`	Display a list of the `PLOT` commands and available colors.
`STATUS`	Prints to the terminal the current values of the plotting parameters such as the `EXTENT`, `BASIS`, and `ORIGIN`.
`LOCATE`	Present the graphics cursor and prepare to receive cursor input from the user. This command is available only if the system has graphics (X Windows [§6.1]) capability. After entering this command, left-click on the plot window. The cursor icon changes to a "+". Move this cursor to a point in the picture and left-click again. The x, y, and z coordinates of the point are displayed. The `LOCATE` command should be the only command on the input line.

6.2.4.1.4 Plot Commands

Plot commands define the values of the parameters used in drawing the next plot. Parameters entered for one plot remain in effect for subsequent plots until they are overridden, either by the same command with new values or by a conflicting command.

BASIS *x1 y1 z1 x2 y2 z2*

Orient the plot so that the direction (x_1, y_1, z_1) points to the right and the direction (x_2, y_2, z_2) points up. The default values are **0 1 0 0 0 1**, causing the y axis to point to the right and the z axis to point up.

The two vectors of **BASIS** do not have to be normalized, but they should be orthogonal. If the two vectors are not orthogonal, MCNP6 chooses an arbitrary second vector that is orthogonal to the first vector. MCNP6 will ignore the command if parallel or zero-length vectors are entered.

ORIGIN *vx vy vz*

Position the plot so that the origin, which is in the middle of the geometry slice, is at the point (v_x, v_y, v_z). The default values are **0 0 0**. The **BASIS** vectors are relative to this point.

EXTENT *eh [ev]*

Set the scale of the plot so that the horizontal distance from the origin to either side of the plot is *eh* and the vertical distance from the origin to the top or bottom is *ev*. The *ev* parameter is optional, and if omitted, it is set equal to *eh*. If *ev* is set and not equal to *eh*, the plot aspect ratio will be distorted. The default values are 100 and 100, creating a viewport of the geometry covering 200×200 cm.

PX *vx*

Plot a cross section of the geometry in the plane normal to the x axis at a distance *vx* from the origin. This command is a shortcut equivalent of "**BASIS 0 1 0 0 0 1 ORIGIN** *vx* **vy vz**" where **vy** and **vz** are the current values of **vy** and **vz**.

PY *vy*

Plot a cross section of the geometry in the plane normal to the y axis at a distance *vy* from the origin.

PZ *vz*

Plot a cross section of the geometry in the plane normal to the z axis at a distance *vz* from the origin.

LABEL *slabel [clabel [par]]*

Put labels of size *slabel* on the surfaces and, optionally, labels of size *clabel* in the cells. The parameter, *par*, following *clabel* is further optional and defaults to **MAT**. The sizes specified by *slabel* and *clabel* are relative to 0.01 times the height of the view window. If *slabel* or *clabel* is zero, that kind of label will be omitted. The allowed range of sizes for the labels is [0.2–100].

The default is **LABEL 1 0**. The possible values of *par* follow, where :$\mathscr{P}$ indicates the particle type.

CEL	Cell labels will be cell numbers.
***IMP:*$\mathscr{P}$**	Cell labels will be cell importances for particle type $\mathscr{P}$.
RHO	Cell labels will be atom densities ($\text{barn}^{-1} \cdot \text{cm}^{-1}$).
DEN	Cell labels will be mass density (g/cm^3).
VOL	Cell labels will be volume (calculated or user-supplied, cm^3).
FCL:$\mathscr{P}$	Cell labels will be forced-collision fraction (from **FCL**:$\mathscr{P}$) for particle type $\mathscr{P}$.
MAS	Cell labels will be masses (g).
PWT	Cell labels will be photon production weights.
MAT	Cell labels will be material number (DEFAULT).
TMP*[n]*	Cell labels will be temperature (MeV) at time *n* (specified on the **TMP** and **THTME** cards). The *n* is optional and defaults to 1.

WWN*[n]*:$\mathscr{P}$	Cell labels will be weight windows for time or energy index *n* (or combined time-energy index *n*) for particle type $\mathscr{P}$. The *n* is optional and defaults to 1. Note: When combining time and energy binning, the index *n* varies as follows: For 3 time bins (T) and 3 energy bins (E), the index, *n*, would map to the following sequence: (T1, E1), (T1, E2), (T1, E3), (T2, E1), (T2, E2), (T2, E3), (T3, E1), (T3, E2), (T3, E3).
EXT:$\mathscr{P}$	Cell labels will be exponential transform stretching parameter for particle type $\mathscr{P}$.
PD*n*	Cell labels will be detector contribution frequency fraction to tally *n*.
DXC:$\mathscr{P}$	Cell labels will be DXTRAN contribution frequency fraction.
U	Cell labels will be universe numbers.
LAT	Cell labels will be the enclosed lattice type.
FILL	Cell labels will be filling universe identification numbers.
IJK	Cell labels will be lattice indices.
NONU	Cell labels will be fission behavior toggles.
PAC*[n]*:$\mathscr{P}$	When PLOT is called from MCPLOT, cell labels will be values of columns in PRINT Table 126 [§5.13.3.15]. The *n* is optional and defaults to 1. Allowed values of *n* are:

PAC1:$\mathscr{P}$	The labels are the "tracks entering" column of Table 126.
PAC2:$\mathscr{P}$	The labels are the "population" column of Table 126.
PAC3:$\mathscr{P}$	The labels are the "collisions" column.
PAC4:$\mathscr{P}$	The labels are the "collisions * weight (per history)" column.
PAC5:$\mathscr{P}$	The labels are the "number weighted energy" column.
PAC6:$\mathscr{P}$	The labels are the "flux weighted energy" column.
PAC7:$\mathscr{P}$	The labels are the "average track weight (relative)" column.
PAC8:$\mathscr{P}$	The labels are the "average track mfp (cm)" column.

⚠ Caution

In the command-prompt PLOT mode, the PD*n* option for the LABEL command will show all zeros on the cell labels even if the user has specified values on a PD card. If these labels are desired, then use the preview plotter cell-label capability discussed in §7.3.2.1.

LEVEL *n*	Plot only the *n*th level of a repeated structure geometry. A negative entry (DEFAULT) plots the geometry at all levels. If user-supplied *n* is greater than the

number of levels in the geometry, all levels are plotted as if a negative entry was supplied.

Note: $n \leq 20$.

`MBODY` *state*

Where *state* can be:

`ON`	Display only the macrobody surface number. (DEFAULT)
`OFF`	Display the macrobody surface facet numbers.

`MESH` *n*

Controls plotting of cell lines, and/or the weight-window or weight-window generator superimposed mesh.

Always Available:

n= 0 (No Lines)	Plot cells not outlined in black.
n= 1 (CellLine)	Plot constructive solid geometry cells, outlined in black. (DEFAULT)

Available when appropriate cards are present:

n= 2 (WW MESH)	Plot weight-window superimposed mesh without cell outlines.
n= 3 (WW+Cell)	Plot weight-window superimposed mesh and cell outlines.
n= 4 (WWG MESH)	Plot weight-window generator mesh.
n= 5 (WWG+Cell)	Plot weight-window generator mesh and cell outlines.
n= 6 (MeshTaly)	Plot `TMESH` mesh tally boundaries (RMESH, CORA, etc., required).
n= 7 (MT+Cell)	Plot `TMESH` mesh tally boundaries + CellLine

The `CellLine` and `No Lines` options are always available. `WW MESH` and `WW+Cell` are available only when the `WWP` card calls for using a superimposed weight-window mesh (5th entry negative) and a *WWINP* file is provided on the MCNP6 execution line. `WWG MESH` and `WWG+Cell` are available only when a `MESH` card appears in the input and when the `WWG` card requests superimposed mesh generation (2nd entry is 0). Similarly, `MeshTaly` and `MT+Cell` are available only when a `TMESH` mesh tally has been requested.

Depending on the combination of cards in an input, a user may have a `WWP` card, no mesh generation cards, and a `TMESH` card. In this case, `MESH` *n* for *n*= `0` to `3` will behave as described, but *n*= `4` and `5` would be the `MeshTaly` and `MT+Cell` options above and anything above *n*= `5`would default to `No Lines`. Other behavior can occur with different combinations of input cards. The user is encouraged to experiment and arrive at an understanding for an input-by-input basis.

`FMESH` *n*

Plot `FMESH` mesh tally *n*. `FMESH off` will turn off the mesh tally plotter.

Changes the layout of the plot depending on the type of mesh tally:

For rectangular meshes, the horizontal axis is in the direction of the dimension with the greatest number of bins, and the vertical axis is in the direction of the dimension with the second greatest number of bins.

For cylindrical meshes, the horizontal axis is along the axis of the cylinder and the vertical axis is along the $\theta = 0$ plane. The center of the plot in both cases is at the center of the mesh.

Note: To keep the original layout, use the `FMESH` button of the interactive plotter instead.

SCALES *n*

Put scales, or scales and a grid, on the plot. Scales and grids are incompatible with VIEWPORT SQUARE.

Note: Scales are centered at the current plot origin and go to plus or minus EXTENT in both directions.

n= 0	Neither scales nor a grid are displayed. (DEFAULT)
n= 1	Display scales on the edges of the viewport.
n= 2	Display scales on the edges of the viewport and overlay an associated grid.

CONTOUR *cmin cmax cint*

The parameters *cmin*, *cmax*, and *cint* are the minimum, maximum, and interpolation scheme, respectively. All 3 arguments are required: *cmin*, *cmax*, and *cint*. If this is not satisfied, the plotter hangs indefinitely and must be killed. The CONTOUR command is valid for TMESH mesh tallies only, for FMESH, see §6.3.3.11. The CONTOUR command usage and syntax is different in **MCPLOT** [§6.3.3.10].

The expected form of both *cmin* and *cmax* changes between *cint* options (1):

cint=% or PCT	If either the % symbol or the PCT keyword is used, *cmin* and *cmax* are percentages between the minimum and maximum values of the TMESH tally results. Values between *cmin* and *cmax* are linearly interpolated across 10 values. Restriction: $0 \leq cmin < cmax \leq 100$
cint=LIN	Behaves similarly to % or PCT, but values specified for *cmin* and *cmax* are actual tally values instead of percents. Restriction: $cmin < cmax$
cint=LOG	Values of *cmin* and *cmax* are tally values that are logarithmically interpolated between. Can result in a smoother color-map than the other two options, especially when there is a large range in data. Restriction: $cmin < cmax$

Special usage:

CONTOUR OFF	After using the CONTOUR command as described above, revert to a default view with CONTOUR OFF. This is the same as CONTOUR MIN_TMESH MAX_TMESH LOG. Valid for TMESH mesh tallies.

COLOR *n*

Turn color on or off, set the resolution, or select the physical property for color shading.

n=ON	Turn color on. (DEFAULT)
n=OFF	Turn color off.
$50 \leq n \leq 5000$	Set the color resolution to *n*. A larger value increases resolution (which can better represent color shading along curved interfaces) and drawing time.
n=BY *aa*	Select the physical property to use for geometry shading. Allowed *aa* options for COLOR BY include:

aa=MAT	Cell colors will be cell materials. (DEFAULT)
aa=DEN	Cell colors will be mass density (g/cm^3).

aa=RHO	Cell colors will be atom density ($\text{barn}^{-1} \cdot \text{cm}^{-1}$).
aa=TMP	Cell colors will be temperature (MeV).
aa=CEL	Cell colors will be cell numbers.
aa=IMP:$\mathscr{P}$	Cell colors will be cell importances for particle type $\mathscr{P}$.
aa=GRADIENT,	use a continuous gradient of 256 colors to show the color values.
aa=SOLID,	use a solid color to represent a range of cell values.

When DEN, RHO, TMP , or IMP:$\mathscr{P}$ is used, the geometry will be shaded using the color GRADIENT mode. Linear interpolation between the minimum non-zero value and the maximum value is used to select the color. If shading by cell importance and if the minimum and maximum importance varies enough, then logarithmic interpolation is used. A color bar legend of the shades will be drawn in the left margin. The legend is labeled with the property name and the minimum and maximum values. See Fig. 6.1 for an example of coloring by mass density (DEN). Coloring by material (MAT) or cell (CEL) does not invoke the color bar legend.

SHADE *m1 value1 m2 value2 m3 value3 ...*

Sets the color of material number *m1* to *value1* and so on. This command is only valid when COLOR BY MAT is active (the default with "COLOR ON").

Legal entries for *valueN* are either an integer from 1–64 or one of the color names that are displayed with the HELP (or ? or OPTIONS) command. The integers map to the color names by row first, then column (top to bottom, left to right). For example, SHADE 1000 4 and SHADE 1000 green both set material 1000 to green. See Fig. 6.2 for a list of colors. Note: color names are case-sensitive.

Details:

① For all valid combinations of *cmin*, *cmax*, and *cint*, a description of the TMESH tally's minimum and maximum values are given to the right of the text "**contour plot values:**". Similarly, the range of the values that are covered by colors in the plotter are shown to the right of the "**colors:**" text. The interpolation scheme and number of histories follows. This is helpful to query useful minimums and maximums of *cmin* and *cmax* respectively.

6.2.4.1.5 View Manipulation Commands

View manipulation commands redefine the origin, bases, and extent relative to the current view origin, bases, and extent. The new origin, bases, and extent will be used for all subsequent plots until they are again redefined, either by view manipulation commands or by plot commands such as ORIGIN. The view manipulation commands are usually used to zoom in on some feature of the plot.

Figure 6.2: Available MCNP Plotter Colors for SHADE

CENTER *dh dv*	Change the origin of the plot by the amount *dh* in the horizontal direction and by the amount *dv* in the vertical direction. This command is usually used to define the center of a portion of the current plot that the user wants to enlarge.
FACTOR *f*	Enlarge the plot by the factor $1/f$. The parameter *f* must be greater than 10^{-6}.
THETA *th*	Rotate the plot counterclockwise by the angle *th*, in degrees. Negative values rotate the plot clockwise.
CURSOR	Present the graphics cursor and prepare to receive cursor input from the user. This command is available only if the system has graphics (X Windows [§6.1]) capability. After entering this command, left-click on the plot window. The cursor changes shape to appear like the upper left corner of a box. Click in the plot window at a point representing the upper left spatial boundary of the desired plot. The cursor will change shape again; now click the lower right position of the desired plot. The plot will be redrawn using the new boundaries and keep a 1:1 aspect ratio. This is equivalent to an EXTENT command and an ORIGIN command.
RESTORE	Restore the origin and extent to the values they had before the most recent CURSOR command. The RESTORE command should be the only command on the input line. It cannot be used to undo the effects of the CENTER, FACTOR, and THETA commands.

6.2.5 Plotting Embedded-mesh Geometries

The MCNP6 plotter supports color-shaded plotting of the materials, mass density, or atom density of an imported embedded mesh. For these cases, the values from the external mesh geometry file (typically a LNK3DNT or Abaqus-style file) are used; these values may vary element to element.

When the geometry is plotted with COLOR BY DEN (mass density) or COLOR BY RHO (atom density), each mesh element is shown in one solid color. The element net value is plotted, i.e., the net mass density or net number density of the element. The color distribution is set by the minima and maxima. These net values are also the values reported for plot queries when DEN or RHO is selected from the right-hand-side interactive menu [§6.2.3.3].

If MAT is selected from the right-hand-side interactive menu, clicking on a spot containing multiple materials will randomly select which material to report. Repeatedly clicking on such a spot may show different materials on different clicks. Void elements in the mesh are not shaded (i.e., shown as white) on material plots.

6.2.6 Geometry Debugging

Surfaces appearing on a plot as red dashed lines usually indicate that the geometry is improperly defined. A geometry error can arise when a region has been defined in more than one cell or a particular region has never been defined. These geometry errors must be corrected.

Dashed or incomplete lines also can occur because the plot plane is coincident with a plot surface. In this case, the terminal will issue a warning. The plot plane should be moved so it is not coincident with any geometry surface.

Dashed lines may also indicate a cookie cutter cell (red dashes) or a DXTRAN sphere (blue dashes). These are not errors.

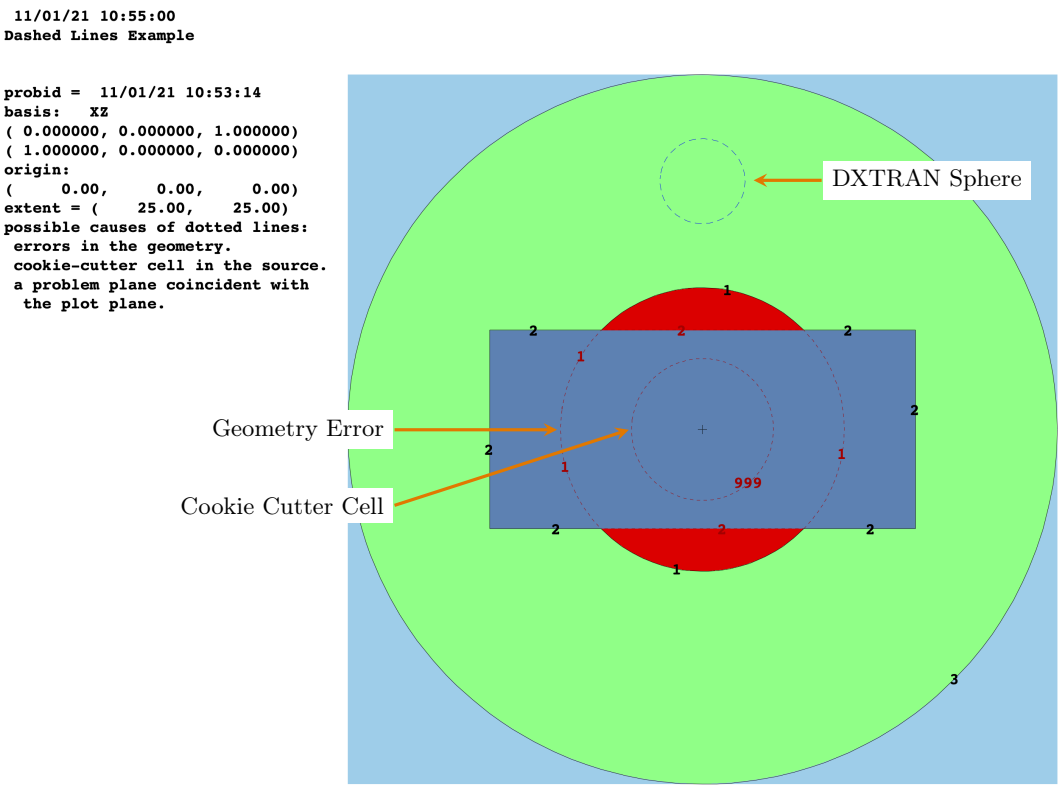


Figure 6.3: Different types of Dashed Lines

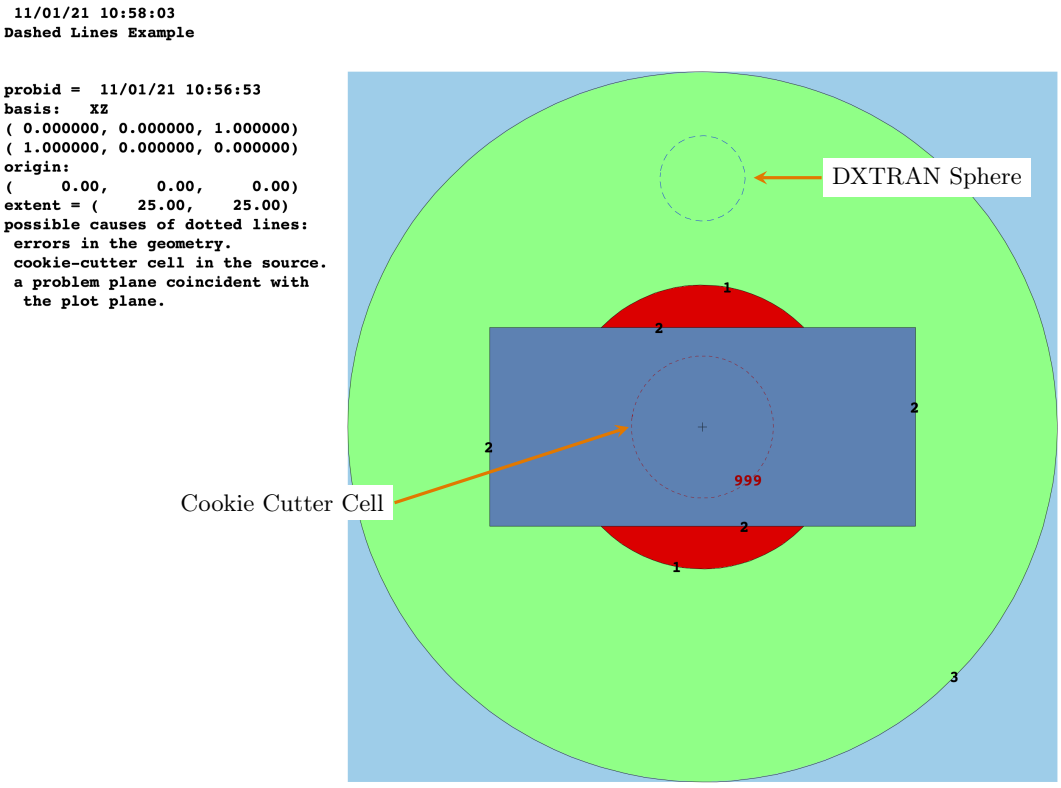


Figure 6.4: Dashed Lines with no Geometry Errors

In Figure 6.3 and 6.4, both geometry errors and cookie cutter cells are represented with the same red dashed line. Thus, the reason for dashed lines on an MCNP6 geometry plot must be understood before running a problem.

When checking a geometry model, errors may not appear on the two-dimensional slice chosen, but one or more particles may get lost in tracking. To find the modeling error, use the coordinates and trajectory of the particle when it got lost (listed in the output file). Entering the particle coordinates as the `ORIGIN` and the particle trajectory as the first `BASIS` vector with any other non-colinear vector as the second `BASIS` vector will result in the plotter centered on the point the particle was lost with the horizontal direction of the plot consistent with the direction the particle was moving.

⚠ Caution

In some cases, particles may be lost in a lattice but entering the `ORIGIN` and `BASIS` as described above will not display any broken geometry lines. In these cases, the user has likely not fully specified the geometry universes filling the lattice. Just as the top level universe must have all regions of space defined, the sub universes should be fully defined. If the user is still losing particles, surfaces of filling universes should be made non-coincident with surfaces on the filled lattice cells.

6.2.7 Geometry Plotting in Batch Mode

Although MCNP6 can be run in a batch environment, user interaction with the plotter is significantly reduced. Rather than entering commands manually in this environment, it is recommended to use the **`NOTEK`** option on the MCNP6 execution line and read a command file with the **`COM`** option. Every view plotted will be put in a local graphics file. See §6.2.1 for more information on **`NOTEK`** and the **`COM`** execute options.

6.3 The Tally and Cross-Section Plotter, MCPLOT

Tally results and cross-section data are plotted by MCNP6 through the **`MCPLOT`** module. It can draw ordinary two-dimensional x-y plots and contour or color-filled tally plots of three-dimensional data. **`MCPLOT`** supports a wide variety of plot options including plotting data linearly or logarithmically, manipulation of the axes limits, and data coplotting. Tally plots can be created from tally data that exists within a `MCTAL` or `RUNTPE` file. However, when plotting from a `MCTAL` file, not all options are available because not all the information is available in that format.

In addition to plotting tally results, **`MCPLOT`** plots cross-section data specified in an `INP` file. Either individual nuclides or the complete material composed of constituent nuclei properly weighted by atomic fraction may be plotted. The data plotted reflect adjustments to the cross sections made by MCNP6 such as energy cutoffs, neutron cross-section temperatures, $S(\alpha, \beta)$ treatment, the summation of photon reactions to provide a total photon cross section, simple physics treatment for photon data, electron stopping powers, and more. Cross-section plots cannot be made from a `RUNTPE` file.

Final tally results can be plotted after particle transport has finished. The temporary status of one or more tallies can also be displayed during the calculation as transport is ongoing. After transport is finished, **`MCPLOT`** is invoked by typing a **`Z`** on the MCNP6 execution line, and reading an existing `RUNTPE` or `MCTAL` file. The `RUNTPE` file may be read as an execution line option or after invoking the tally plotter. The `MCTAL` file may only be read after invoking the tally plotter.

`MCNP6 Z RUNTPE=`*`filename`*

or

MCNP6 Z then type runtpe=*filename* at the MCPLOT> prompt.

or, for a MCTAL file:

MCNP6 Z then type rmctal=*filename* at the MCPLOT> prompt.

To superimpose a mesh tally with problem geometries, initiate **MCPLOT** using one of the execute lines above and then enter the geometry plotter using the PLOT command. A RUNTPE file must be read to obtain the mesh tally data.

There are two ways to request that a plot be produced periodically during the run: use an MPLOT card in the INP file or use the TTY interrupt feature [§3.3.3]. Note: The TTY interrupt capability is not always possible during parallel computations, particularly when using MPI parallelization.

The TTY interrupt, Ctrl + c , m , causes MCNP6 to pause at the end of the history that is running when the interrupt occurs and allows plots to be made by calling **MCPLOT**. During run-time plotting, no output is sent to the COMOUT file. In addition, the following commands cannot be used after invoking **MCPLOT** with an interrupt: RMCTAL, RUNTPE, and DUMP. The END or RETURN commands are used to exit **MCPLOT** and return MCNP6 to transport mode. Cross-section data cannot be displayed after a TTY interrupt or by use of the MPLOT card.

Mesh tally, radiography tally, and lattice tally results can be displayed as color contour plots. Mesh tallies can also be plotted superimposed over problem geometries. All of these plots are done in MCNP6 without the need of auxiliary post-processing codes.

MCPLOT can make tally plots on a machine different from the one on which the problem was run by using the MCTAL file. When the INP file has a PRDMP card with a non-zero third entry, a MCTAL file is created at the end of the run. The MCTAL file is an ASCII file that contains all the tally data in the last RUNTPE dump. When the MCTAL file is created, its name can be specified in the execute line using the following format:

MCNP6 I=*inpfile* **MCTAL=***filename*

If the MCTAL option is omitted, the default *filename* is a unique name based on MCTAL: First MCTAL, then MCTAM, then MCTAN and so on.

The **MCPLOT** HELP command provides an alphabetized columnar listing of options [333]. Below the listing of commands are instructions describing how to:

1. invoke a listing of all HELP commands with an explanation of their function and use syntax (HELP ALL),
2. provide a listing of function and syntax for a single command (HELP *command*),
3. request an overview of the **MCPLOT** capability (HELP OVERVIEW), and
4. summarize input and execution-line options (HELP EXECUTE).

See §6.4 for examples of using **MCPLOT**.

6.3.1 Execution Line Options Related to MCPLOT Initiation

To run only **MCPLOT** and plot tallies upon termination of the calculation by MCNP6, enter the following command:

MCNP6 Z *KEYWORD[=value(s)]*

where **Z** invokes **MCPLOT**. Cross-section data cannot be plotted by this method. The allowed keywords are:

NOTEK	Suppress plotting at the terminal and send all plots to the graphics metafile, PLOTM. The NOTEK keyword is used for production and batch situations or when the user's terminal has no graphics capability.
COM=*filename*	Use file *filename* as the source of plot requests. When an end-of-file (EOF) is read, control is transferred to the terminal. In a production or batch situation, end the file with an END command to prevent transfer of control. Never end the COM file with a blank line. If COM is absent, the terminal is used as the source of plot requests.
RUNTPE=*filename*	Read file *filename* as the source of MCNP6 tally data. The default file name is **runtpe.h5**. If the default restart file does not exist, the user will be prompted at the MCPLOT> prompt to read a restart file with the RUNTPE command.
PLOTM=*filename*	Name the graphics metafile *filename*. The default metafile is a standard postscript file and named **plotm.ps**. In the absence of a specified metafile name, MCNP6 increments the last character until it runs out of unique names: **plotm.ps**, **plotn.ps**, **ploto.ps**, . . . , **plotl.ps**. Unique names for the output file, PLOTM, will be chosen by MCNP6 to avoid overwriting existing files.
COMOUT=*filename*	Write all plot requests to file *filename*. The default name is **comout**. PLOT writes the COMOUT file in order to give the user the opportunity to do the same plotting at some later time, using all or part of the old COMOUT file as the COM file in the second run. In the absence of a specified COMOUT filename, MCNP6 increments the last character until it runs out of unique names: **comout**, **comouu**, **comouv**, . . . , **comous**.

To run transport, plot cross-section data, and tallies in one line, use the execution line:

MCNP6 INP=*filename* **IXRZ** *KEYWORD[=value(s)]*

This causes MCNP6 to run the problem specified in *filename*, following which the prompt MCPLOT> appears for **MCPLOT** commands. At this point, both cross-section data and tallies can be plotted.

Cross-section data cannot be plotted after a TTY interrupt or by use of the MPLOT card.

To plot only cross-section data, use the execute line command:

MCNP6 INP=*filename* **IXZ** *KEYWORD[=value(s)]*

The problem cross sections are read in, but no transport occurs. When using this method to plot cross sections, the following commands cannot be used: BAR, CONTOUR, DUMP, FREQ, HIST, PLOT, RETURN, RMCTAL,RUNTPE, SPLINE, WASH, and WMCTAL.

6.3.1.1 MCPLOT Basic Concepts

Plot requests are entered from the terminal or they can be read from a file. A plot is requested on the terminal by entering a sequence of plot commands at the `MCPLOT>` prompt. The request is terminated with the Enter key. Commands consist of keywords usually followed by some parameters, either space or comma delimited. Command keywords, but not parameters, can be abbreviated to any degree not resulting in ambiguity, but they must be correctly spelled. The maximum line length is 128 characters. If the line is terminated with the `&` character, the command string may be continued on the next line.

Note that the `&` character may only break a string between subsequent commands and not between the values entered for a command. For example:

```
xlim 1e-8 1e-4 &
ylim 1e-5 1e-1
```

is valid, while

```
xlim 1e-8 &
1e-4 ylim 1e-5 1e-1
```

is invalid.

Termination of a line with the `COPLOT` command will wait to draw the plot until the next string of commands is terminated with the Enter key. Only those commands marked with a dagger (†) in the list presented in §6.3.3 can be used after the first `COPLOT` command in a plot request because the others affect the framework of the plot or are for contour or 3-D plots only.

When MCNP6 is run with just **`z`** as the execute line option (**`mcnp6 z`**), the code will attempt to locate and read the file **`runtpe.h5`**. If this file is present and has more than one energy bin in a tally, a default plot is obtained by pressing the Enter key once `MCPLOT>` prompt is displayed. This default is a lin-log histogram plot of the lowest numbered tally in tally/MeV against energy, with error bars and suitable labels. If any of these default requirements are unsatisfied a message to that degree will be printed to the terminal window.

In this Section, the term "current plot" means the plot that is being defined by the commands currently being typed in, which might not be the plot that is showing on the screen.

6.3.2 Plot Types Available in MCNP6

6.3.2.1 2-D Plot

The origin of coordinates for two-dimensional **`MCPLOT`** plots is at the lower-left corner of the picture. The horizontal axis is called the x-axis. It is the axis of the independent variable such as user bin, cell number, or energy. The vertical axis is called the y-axis. It is the axis of the dependent variable such as flux, current, or dose. Each axis can independently be either linear or logarithmic.

6.3.2.2 Contour Plot

Similarly, the origin of coordinates for **MCPLOT** contour plots is at the lower-left corner of the picture. The horizontal axis is called the x-axis. It is the axis of the first of the two independent variables. The vertical axis is called the y-axis. It is the axis of the second independent variable. The contours represent the values of the dependent variable. For contour plots, only linear axes are available. Each contour is drawn in a different color depending on its value with respect to the z-value extrema. Extensions to the `FREE` and `CONTOUR` commands allow for shaded contour plots of tally and mesh data.

For additional examples involving contour plots see §6.4.2 and §6.4.3.

6.3.2.3 Color-wash Plot

This plot option is similar to contour plotting, but instead of drawing contours of $z(x, y)$ data, each tally bin is filled with a color selected by the tally value in the bin. The axis conventions are the same as in contour plotting. This option is selected with the command `WASH`. If two free variables have been selected with the `FREE` command, a color-filled plot is drawn. This is a useful option for radiography tallies. The color index is selected by linear interpolation between the z-minimum and the z-maximum values.

6.3.3 Tally Plot Commands Grouped by Function

A dagger (†) indicates a command can be used after the first `COPLOT` command in a plot request.

6.3.3.1 Device-control Commands

Normally **MCPLOT** draws plots to a system's X Window display. By using the following commands, the user can specify that plots not be drawn to the display and/or that they be sent to a graphics metafile or PostScript file for processing later by a graphics utility program.

`TERM` *n*	Output device type is specified by *n*. *n*=`0` for a terminal with no graphics forwarding capability (for a system without the X Window System). No plots are drawn to a display window, and all plots are sent to the graphics metafile. `TERM 0` is equivalent to putting `NOTEK` on MCNP6's execution line [§6.3.1]. *n*=`1` restores the plotting window on the next plot request.
`FILE` *aa*	Send or do not send plots to the graphics metafile `PLOTM.PS` according to the value of the parameter *aa*. The graphics metafile is not created until the first `FILE` command is entered. `FILE` has no effect in the `NOTEK` or `TERM 0` cases. The allowed values of *aa* are the following:

aa is blank	Only the current plot is sent to the graphics metafile.
aa=`ALL`	The current plot and all subsequent plots are sent to the metafile until another `FILE` command is entered.
aa=`NONE`	The current plot is not sent to the metafile nor are any subsequent plots until another `FILE` command is entered.

6.3.3.2 General Commands

`&`	Continue reading commands for the current plot from the next input line. The `&` must be the last character on the line. The `&` command must not break another command and its parameters onto two lines; instead, it is used to continue long user command strings on new lines.†
`COPLOT`	Plot a curve according to the commands entered so far and keep the plot open for co-plotting one or more additional curves. `COPLOT` is effective for 2-D plots only. If `COPLOT` is the last command on a line, it functions as if it were followed by an `&`. Only the commands followed by a dagger (†) in this section are valid to enter following `COPLOT`.
`FREQ` *n*	Use with the `MPLOT` card; has no effect when **MCPLOT** is called through the **Z** execution option. Specifies the interval between calls to **MCPLOT** to be every *n* histories. In a `KCODE` calculation, the interval is every *n* cycles. If *n* is negative, the interval is in CPU minutes. If *n*= 0, **MCPLOT** is not called while MCNP6 is running histories. Note: An 8-byte integer is allowed for *n*. (DEFAULT: *n*= 0)
`RETURN`	If **MCPLOT** was called by MCNP6 while running histories or by **PLOT** while doing geometry plotting, control returns to the calling subroutine. Otherwise `RETURN` has no effect.
`PLOT`	Call or return to the **PLOT** geometry plotter. This cannot be done when plotting from a `MCTAL` file.
`PAUSE` *[n]*	Can be used on any line of a plot command file that is specified with the execute `COM=`*filename* option [§6.3.1]. Holds each view for *n* seconds. If no *n* value is provided, each view remains until Enter is pressed. When absent, the commands specified in the command file will run sequentially until the end of the command file is reached at which point control returns to the terminal.
`END`	Terminate execution of `PLOT`. Closes any open X Windows and returns the terminal from the `PLOT` prompt to a standard system shell prompt.†

6.3.3.3 Inquiry Commands

When one of these commands is encountered, the requested display is made and then **MCPLOT** waits for the user to enter another line, which can be just pressing the Enter key, before resuming. The same thing will happen if **MCPLOT** sends any kind of warning or comment to the user as it prepares the data for a plot.

`OPTIONS` or `?` or `HELP`	`OPTIONS` or `?` may be interchanged in:†	
	`HELP` *[COMMAND]*	Display a list of available **MCPLOT** commands or the help text of the specified **MCPLOT** command. While this can be convenient for a quick reminder of the usage of the command, this Manual should be referenced as some of the help text provided by this option is out of date.
	`HELP OVERVIEW`	Display a description of the **MCPLOT** module akin to the introduction of this Section (§6.3).
	`HELP EXECUTE`	Display help text for **MCPLOT** input and execution-line options.

Caution

The HELP EXECUTE text incorrectly implies that RMCTAL is an execution line option in MCNP6.

Caution

While the HELP *[COMMAND]* functionality can be useful for quick syntax checks, some of the help-text is out of date or incorrect. The user should primarily refer to this manual for instruction on using the tally plotter module. Please email mcnp_help@lanl.gov if there is a discrepancy in the functionality of a command.

STATUS	Display the current values of the plotting parameters including the name of the file being plotted from, tally number, bin information, and more.†
PRINTAL	Display the available tally numbers in the current RUNTPE or MCTAL file.†
IPTAL	Display the IPTAL array for the current tally. The command prints how many bins are in each dimension of the current 8-dimensional tally. This helps remind the user of how the tally is setup and may eliminate the need to reference the input file. †
PRINTPTS *[filename]*	Display the x-y coordinates and the relative error of the points in the current plot. PRINTPTS is not available for co-plots, contour plots, color-wash plots, or 3-D plots. Print to the terminal (default behavior) or to the file named *filename* (optional, if specified).

6.3.3.4 File Manipulation Commands

RUNTPE *filename [n]*	Read dump *n* from RUNTPE file *filename*. If the parameter *n* is omitted, the last dump in the file is read.†
DUMP *n*	Read dump *n* of the current RUNTPE file.†
WMCTAL *filename*	Write the tally data in the current RUNTPE dump to MCTAL file *filename*.†
RMCTAL *filename*	Read tally data from MCTAL file *filename*.†

6.3.3.5 Parameter-setting Commands

Parameters entered for one curve or plot remain in effect for subsequent curves and plots (including co-plots) until they are either reset to their default values with the RESET command or are overridden, either by the same command with new values, by a conflicting command, or by the FREE command that resets many parameters. There are two exceptions: FACTOR and LABEL are effective for the current curve only. An example of a conflicting command is BAR, which turns off HIST, PLINEAR, and SPLINE.

TALLY *n*	Define tally *n* as the current tally.† The parameter *n* is the tally designation on the F card in the INP file of the problem represented by the current RUNTPE or MCTAL file. The default is the first tally in the problem: which is the lowest numbered neutron tally or, if there are no neutron tallies, the lowest numbered photon tally or, if there are no neutron or photon tallies, the lowest numbered electron tally.

PERT *n*	Plot a perturbation associated with the current tally, where *n* corresponds to a PERT*n* card.† The command PERT 0 will reset PERT *n*.
LETHARGY	Divide tally bin by lethargy bin width for log energy abscissa. Produces visually accurate area plots for a 2-D logarithmic energy abscissa (FREE E). A lethargy-normalized plot is equivalent to plotting $e \cdot f(e)$. Note: LOGLIN or LOGLOG must be specified and NONORM must not be invoked. See §6.5.
NONORM	Suppress bin normalization. The default in a 2-D plot is to divide the tallies by the bin widths if the independent variable is cosine, energy, or time. Bin structure is described in the description of the MCTAL file [§6.3.4]. Bin normalization is not done in 3-D, contour, or color-wash plots.
FACTOR *a f [s]*	Multiply the data for axis *a* by the factor *f* (restriction: $f > 0$) and then add the term *s*.† The parameter *a* is a cartesian axis: X, Y, or Z. The parameter *s* is optional and defaults to 0. The value given by FACTOR affects only the current curve or plot.
RESET *aa*	Reset the parameters of command *aa* to their default values.† The parameter *aa* can be a parameter-setting command or ALL. If *aa* is ALL, the parameters of all parameter-setting commands are reset to their default values. After a COPLOT command, only ALL or any of the parameter-setting commands that are marked with a † in this list may be reset. Resetting ALL while COPLOT is in effect causes the next plot to be an initial plot of the most recently read RUNTPE file.

6.3.3.6 Titling Commands

The use of quotation marks is required for character strings that have whitespace within them.

TITLE *n* "*aa*"	Use *aa* as line *n* of the main title at the top of the plot. The allowed values of *n* are 1 and 2. The maximum length of *aa* is 40 characters. The default is the comment on the FC card for the current tally, if any. Otherwise it is the name of the current RUNTPE or MCTAL file plus the name of the tally. KCODE plots have their own special default title.
BELOW	Put the title below the plot instead of above it. The keyword BELOW has no effect on 3-D plots.
SUBTITLE *x y "aa"*	Write subtitle *aa* at location *x*, *y*, which can be anywhere on the plot including outside the plot as long as it is within the limits of the X Window. The values of *x* and *y* are x- and y-axis values. The maximum length of *aa* is 40 characters.
XTITLE *"aa"*	Use *aa* as the title for the x-axis. The default is the name of the variable represented by the x-axis. The maximum length of *aa* is 40 characters.

YTITLE *"aa"*	Use *aa* as the title for the y-axis. The default is the name of the variable represented by the y-axis. The maximum length of *aa* is 40 characters.
ZTITLE *"aa"*	Use *aa* as the title for the z-axis in 3-D plots. The default is the name of the variable represented by the z-axis. The maximum length of *aa* is 40 characters.
LABEL *"aa"*	Use *aa* as the label for the current curve.† The label is printed in the lower right of the plot window beside a sample of the line style used to plot the curve. The maximum length of *aa* is 10 characters. The value of LABEL reverts to its default value, blank, after the current curve is plotted. If LABEL is blank, the name of the RUNTPE or MCTAL file being plotted is printed as the label for the curve.
FONT *ax title*	Use to adjust font size for plot axes (*ax*) and title(s) (*title*). Allowable values for the parameters are dependent on the user's system, but cannot exceed 100% ($ax, title = 1$). Default: FONT 0.4375 0.6667. Example: FONT 0.3 0.7 sets the axis labels to 30% and the title to 70% of their maximum. Example: FONT j 0.5 uses previously specified value for the axis font (or default) and sets the title to 50% of maximum.

6.3.3.7 Plot-Variable Control Commands

Tallies in MCNP6 are binned according to the values of eight independent variables:

F	Tally bins on an F tally (cell, surfaces, or detector),
D	Total vs. direct or flagged vs. unflagged contributions (see CF, SF),
U	User-defined bins. For example, FT TAG bins (§5.9.18.13),
S	Segment bins on an FS card,
M	Multiplier bins from an FM card,
C	Cosine bins from a C card,
E	Energy bins from an E card,
T	Time bins from a T card.

Note: Other cards may affect binning of the eight dimensions. The reader should reference §5.9 for more information.

Because only one or two of those variables can be used as independent variables in any one plot, one or two of the eight independent variables have to be designated as free variables, and the rest become fixed variables. Fixed values (bin numbers) are defaulted for all fixed variables, but may be explicitly overridden. The default value for each fixed variable is the total bin, if present; otherwise the first is used.

FREE *x[y] [nXm] [ALL|NOALL]*	Use variable *x* (*y* blank) or variables *x* and *y* as the independent variable or variables in the plot. Valid values for *x* and *y* are the tally bin indices F, D, U, S, M, C, E, T, I, J, and K, where I, J, and K refer to lattice or mesh indices. If only *x* is specified, 2-D

plots are made. If both *x* and *y* are specified, contour, color-wash, or 3-D plots are made, depending on whether 3-D is in effect. The default value of *x* is E, and gives a 2-D plot in which the independent variable is energy.

The *nXm* ("*n* by *m*") entry specifies the number of bins associated with the I and J lattice indices. Only valid when *x* =I, J, or K or when *xy* is a combination of of those indexes.

The ALL entry specifies that the minimum and maximum contour range should be taken from all the tally bins. Only valid when *x* =I, J, or K or when *xy* is a combination of of those indexes. Omitting this parameter results in the default minimum and maximum contour range, which includes only those tally values contained in the specified 2-D plot.

The NOALL entry specifies that the minimum and maximum contour range should be taken only from those of the FIXED command slice. (DEFAULT)

The FREE command resets XTITLE, YTITLE, ZTITLE, XLIMS, YLIMS, HIST, BAR, and PLINEAR to their defaults.

For more information regarding usage of the FREE command, see §6.3.3.12.

FIXED *q n*

Set *n* as the bin number for fixed variable *q*.† The symbols that can be used for *q*, are F, D, U, S, M, C, E, T, I, J, and K, where I, J, and K refer to lattice or mesh indices.

Restriction: Only the J and K indices are allowed for a 1-D IJK plot and only the K index is allowed for a 2-D IJK contour plot.

SET *f d u s m c e t*

Define which variables are free and define the bin numbers of the fixed variables.

SET effectively executes the FREE and several FIXED commands in one compact command.

The value of each parameter can be either a bin number (the corresponding variable is then a fixed variable) or an asterisk (∗) (the corresponding variable is then a free variable). If there is only one ∗, 2-D plots are made. If there are two, contour plots are made. SET performs the same resetting of parameters that FREE does.

TFC *info*

Plot the tally fluctuation chart of the current tally. Unless otherwise noted, the independent variable is *nps*, the number of source histories.

Allowed values of *info* include the following:

M	Mean*
E	Relative fractional uncertainty*
F	Figure of merit* (See §2.6.5)
L	201 largest tallies vs x (Unnormalized tally density vs x; the PDF of the tally. See §5.13.3.18.)
N	Cumulative number of scores in TFC bin under consideration (Cumulative $f(x)$ vs x; the CDF of the tally. See §5.13.3.19.)
P	TFC bin PDF probability $f(x)$ vs x (NONORM for number frequency vs x.)
S	Slope of the Pareto fit for high tallies as a function of *nps*
T	Cumulative tally fraction of $f(x)$ vs x
V	Variance of the variance as a function of *nps*
1-8	1st to 8th moments of $x^{1-8} \cdot f(x)$ vs x (NONORM for $x^{1-8} \cdot \Delta x \cdot f(x)$ vs x.)

1c-8c	1st to 8th cumulative moments of $x^{1-8} \cdot f(x)$ vs x

* These data are available when plotting from a MCTAL file.

KCODE *i*

The independent variable is the KCODE cycle. The individual estimator plots start with cycle one. The average col/abs/track-length plots start with the fourth active cycle.†

Plot k_{eff} or removal lifetime according to the value of *i*:

1	k_{eff} (collision)
2	k_{eff} (absorption)
3	k_{eff} (track)
4	Depends on value of FMAT on the KOPTS card:
	FMAT=no — Prompt removal lifetime (collision).
	FMAT=yes — The k_{eff} of the fission matrix solution.
5	Depends on value of FMAT on the KOPTS card:
	FMAT=no — Prompt removal lifetime (absorption).
	FMAT=yes — Shannon entropy of the fission matrix solution.
6	Shannon entropy of fission source distribution. Can be plotted only from a runtape file, not from a MCTAL file.
11–15	The quantity corresponding to $i - 10$, averaged over the cycles so far in the problem.
16	Average collision/absorption/track-length k_{eff} and one estimated standard deviation.
17	Average collision/absorption/track-length k_{eff} and one estimated standard deviation by cycle skipped. Cannot plot fewer than 10 active cycles.
18	Average collision/absorption/track-length k_{eff} figure of merit
19	Average collision/absorption/track-length k_{eff} relative fractional uncertainty.

6.3.3.8 Cross-section Plotting Commands

The cross section plotter is initiated with the **IXZ** option on the execution line:

```
mcnp6 ixz i=inputfile
```

Cross section plots cannot be made from a runtape or MCTAL file.

XS *m*

Plot a cross section according to the value of *m*. Where *m* is one of:†

M*n*	A material card in the input file for material *n*. For example: XS M15 for the total cross-section. The available materials will be listed if a material is requested that does not exist in the INP file.

`Z`	A table identifier [§1.2.3]. Example: `XS 92235.00C`. The full identifier with extension must be provided. Only the tables requested in the input file may be plotted. The available tables will be listed if a table is requested that does not exist in the input file.
`?`	Print out a cross-section plotting primer.

`MT` *n*
Plot reaction n of material or nuclide specified by *`XS`* `m`.†

The default is the total cross section. The available reaction numbers in the data file will be listed if a reaction number that is invalid or doesn't exist is entered (e.g., 999)

`PAR` $\mathscr{P}$
Plot the data for particle type $\mathscr{P}$, of material `Mn`.†

Restriction: $\mathscr{P}$ can only be `N`, `P`, `E`, `H`, `D`, `T`, `S`, or `A`

The default is neutrons for `XS` = `Mn`. For `XS` = *z*, the particle type is determined from the data library type. For example, `XS 92235.00c` defines `PAR` = `N`

Must be first entry on the line.

6.3.3.9 2-D Plotting Commands

`LINLIN`
Use linear x-axis and linear y-axis. (DEFAULT for tally contour plots)

`LINLOG`
Use linear x-axis and logarithmic y-axis. (DEFAULT for all except tally contour plots)

`LOGLIN`
Use logarithmic x-axis and linear y-axis.

`LOGLOG`
Use logarithmic x-axis and logarithmic y-axis.

`XLIMS` *`min max [nsteps]`*
Define the lower limit: *`min`*, upper limit: *`max`*, and (optionally) number of subdivisions: *`nsteps`*, on the x-axis.

The parameter *`nsteps`* is optional for a linear axis and is ineffective for a logarithmic axis. In the absence of any specification by the user, the values of *`min`*, *`max`*, and *`nsteps`* are determined automatically.

`YLIMS` *`min max [nsteps]`*
Define the lower limit: *`min`*, upper limit: *`max`*, and (optionally) number of subdivisions: *`nsteps`*, on the y-axis.

The parameter *`nsteps`* is optional for a linear axis and is ineffective for a logarithmic axis. In the absence of any specification by the user, the values of *`min`*, *`max`*, and *`nsteps`* are determined automatically.

`SCALES` *n*
Put scales on the plots according to the value of *`n`*:

`0`	No scales on the edges and no grid.
`1`	Scales on the edges (DEFAULT).
`2`	Scales on the edges and a grid on the plot.

`HIST`
Make histogram plots.†

This is the default if the independent variable is cosine, energy, or time.

PLINEAR	Make piecewise, linear plots.† This is the default if the independent variable is not cosine, energy, or time.
BAR	Make bar plots.†
NOERRBAR	Suppress default error bars.†
THICK *x*	Set the thickness of the plot curves to the value *x*.† The legal values are $x \geq 0.01$ and the default value of *n* is 0.02.
THIN	Set the thickness of the plot curves to the legal minimum of 0.01.†
LEGEND *[x] [y]*	Include or omit the legend according to the values of optional parameters *x* and *y*. If neither *x* nor *y* is specified, put the legend in its normal place at the right side of the plot window. (DEFAULT) If $x = 0$ and *y* is blank, omit the legend. If both *x* and *y* defined, for 2-D plots only, put most of the legend (restart file, tally bin information, etc) in the default location, but place the line labels at location x, y, where the values of *x* and *y* are based on the units and values of the x and y-axes.

6.3.3.10 Contour-plot Commands

The CONTOUR command can be used to examine 3-D data such as plots of TMESH tallies, or from a FREE command calling out two free tally dimensions. The general form of the contour command is:

CONTOUR *cmin cmax cstep [scheme]*	The parameters *cmin*, *cmax*, and *cstep* are the minimum, maximum, and step values for contours, respectively. The optional *scheme* parameter is the interpolation scheme between *cmin* and *cmax*. The *cstep* parameter varies based on scheme. The default is CONTOUR 5 95 10 %.

Options for the scheme parameter are:

scheme = % or PCT	The first two parameters (*cmin*, *cmax*) are interpreted as percentages of the range between the minimum and maximum value of the dependent variable. The *cstep* parameter is the stride between drawn levels and dictates how many levels exist between *cmin* and *cmax*. Example: CONTOUR 5 95 10 % sets the lower contour level at 5% of the range of the tally and the upper level at 95% with a stride of 10% between drawn contours (5%, 15%, 25%, ..., 95%).
scheme = LIN	The first two parameters (*cmin*, *cmax*) are actual values of the tally. The *cstep* parameter is the stride in terms of the actual values of the tally. Example: CONTOUR 1e-4 1e-1 1e-2 LIN sets the lower contour level at 1×10^{-4} and the upper level at 1×10^{-1} with steps of 1×10^{-2} for a total of 11 values: 1×10^{-4}, 1.01×10^{-2}, 2.01×10^{-2}, 3.01×10^{-2}, ..., 2×10^{-2}
scheme = LOG	The first two parameters (*cmin*, *cmax*) are actual values of the tally and the *cstep* parameter sets the number of logarithmically spaced values to draw contours between. Values of *cmin* and *cmax* are defaulted internally when the user first requests CONTOUR LOG. The default *cstep* is 10. Example: CONTOUR 1e-4 1e-1 4 LOG draws contours between 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, and 1×10^{-1}.

Special uses of CONTOUR:

CONTOUR [ALL|NOALL]		
	ALL	Specifies that the minimum and maximum contour range should be taken from all of the tally bins
	NOALL	Sets the minimum and maximum contour range from the bins in the current plot (DEFAULT)
CONTOUR [LINE|NOLINE]		
	LINE	Draws lines at each contour level (DEFAULT)
	NOLINE	Does not draw lines at contour levels
CONTOUR [COLOR|NOCOLOR]		
	COLOR	The contour plot is colored between the contour values (DEFAULT)
	NOCOLOR	The contour plot is uncolored and just displays lines at the contour levels
WASH *aa*	Set or unset $z(x, y)$ plotting to use color-wash instead of contours. The parameter *aa* can be one of two values:	
	aa = ON	Turn on color-wash plotting for two free variables
	aa = OFF	Turn off color-wash plotting and return to contour plotting for two free variables. (DEFAULT)

Any value for *aa* other than ON is equivalent to OFF.

6.3.3.11 FMESH Mesh Tally Plot Commands

MCNP6 uses the geometry plotter to display the results of the FMESH mesh tally.

The default view depends on the geometry of the mesh tally. For rectangular mesh tallies, the horizontal axis is in the direction of the dimension with the most number of bins, and the vertical axis is in the direction of the dimension with the second most number of bins. For cylindrical mesh tallies, the horizontal axis is along the axis of the cylinder and the vertical axis is along the $\theta = 0$ plane.

The center of the plot in both cases is at the center of the mesh. Different views are obtained by using the MCNP6 geometry plotter commands [§6.2.3 and §6.2.4]. Exiting the mesh tally plotter will return control to the tally plotter interface.

Note: there are two ways to change the FMESH tallies that are plotted. One way is to use the FMESH button of the interactive plotter command panel [§6.2.3.2]. This will change the mesh tally that is drawn, but not the plot attributes (BASIS, EXTENT, and ORIGIN). In other words, using the button will not center the view on the center of the new mesh with the horizontal and vertical axes described above. The second method is to enter the FMESH *n* command in the command box. This will reset the plot layout to the default for that particular mesh tally.

The only FMESH tally plotting related command accessible directly from **MCPLOT** is the FMESH *n* command. The others are input in the command box in the interactive geometry plotter after a mesh tally is drawn.

FMESH *n*	Plot FMESH tally *n*.
FMRELERR *[n]*	Plot the relative errors of the current FMESH tally if the parameter *n* is not provided. The tally number *n* does not need to match the current FMESH tally number, so the plot will show the relative error for the requested tally.
ZLEV *scale [n1 n2]*	Controls the visualization of the FMESH tally results. The parameters *ni* are optional and set the range of the scale:

scale = LOG,	the tally data scaling is set to logarithmic. (DEFAULT)
scale = LIN,	the tally data scaling is set to linear.
n1,	the lower limit of the tally scale.
n2,	the upper limit of the tally scale.

If only n1 is provided, n2 is defaulted to the maximum value of the plot. If neither n1 nor n2 are provided, the range is reset. The default value of scale or the last value of scale is stored by **MCPLOT** and the form of ZLEV can simply be:

```
zlev n1 n2
```

The ZLEV command can also be used to set discrete values to plot without a gradient. The form of this usage is:

```
zlev n1 n2 n3 ... ni
```

To get discrete values, there must be at least 3 values of ni provided.

EBIN *n*	Plot energy bin *n* of the current FMESH tally. The total energy bin is the last bin of the tally (e.g., if there are three energy bins in an FMESH tally, the total energy bin can be requested with EBIN 4).
TBIN *n*	Plot time bin *n* of the current FMESH tally. Similar to the energy bins, the total time bin is the last bin of the tally.

6.3.3.12 Additional Guidance When Using the FREE Command

The FREE command, described in §6.3.3.7, can be used to plot TMESH or lattice tallies with the I, J, and K parameters.

For TMESH tallies, the I, J, and K parameters of the FREE command refer to the CORA, CORB, and CORC mesh-tally dimensions.

For lattice tallies, the I, J, and K parameters of the FREE command refer to i, j, and k lattice indices.

For radiography tallies, the command FREE S C is used to make a contour plot of the radiograph's s and t axes.

For lattice tallies that are not fully specified, the *nXm* dimensions must be provided (e.g. "FREE IJ 5X3"). Mesh and radiography tallies are always specified fully, so [nXm] is never required for them.

One-dimensional mesh, radiography, and lattice tallies may be specified by giving the free dimension of the FREE command and fixing the other two dimensions:

```
FREE I FIXED J=10 FIXED K=12
```

6.3.4 MCTAL Files

A MCTAL file contains the tally data of one dump of a RUNTAPE file. It can be written by the **MCRUN** module of MCNP6 or by the **MCPLOT** module, by other codes, or even by hand in order to send data to **MCPLOT** for co-plotting with MCNP6 tally data. Data from TMESH mesh tallies are written to the MCTAL file; however, data from FMESH mesh tallies are not.

As written by MCNP6, a MCTAL file has the format described in §6.3.4.1, but only as much of it as is essential to contain the information of real substance is necessary. Furthermore the numerical items do not need to be in the columns implied by the formats as long as they are in the right order and are blank delimited. For example, to give **MCPLOT** a table of some value and the associated error versus energy, the user might write a file as simple as the following (note: the file is case-sensitive):

```
e    7
    .2   .4   .7   1   3   8   12
vals
     4.00E-5   .022   5.78E-4   .054   3.70E-5   .079   1.22E-5   .122
     7.60E-6   .187   2.20E-6   .245   9.10E-7   .307
```

If more than one independent variable is desired, other lines such as a t line followed by a list of time values would be needed and the table of tally/error values would need to be expanded. If more than one table of tally/error values is wanted, the file would have to include an ntal line followed by a list of arbitrarily chosen tally numbers, a tally line, and lines to describe all of the pertinent independent variables would have to be added for each table.

The order of expansion of the value/error table depends on the independent variables. For example, with two time bins and n energy bins, the order of the values would be: val(e_1, t_1), val(e_1, t_2), val(e_2, t_1), val(e_2, t_2), ..., val(e_n, t_1), val(e_n, t_2).

When the limits on permitted cell and surface numbers were expanded to 99,999,999, the formatting of MCTAL files was modified to accommodate these values. The cell and surface numbers for the problem are first checked to see if any are greater than 99,999. If not, then the traditional formatting is used when writing the MCTAL file. If there are numbers greater than 99,999, then (I10) format is used instead for these integers written to the MCTAL file. Although the MCTAL file is defined to be free-format, some simple user-written utility programs that read the MCTAL file may expect fixed format. If such user-written programs cannot be modified to handle the larger integers, then users should be careful to use only numbers less than 99,999 for cell and surface numbers in their input files.

6.3.4.1 Form of the MCTAL File

The (case sensitive) form of the MCTAL file as written by MCNP6 with the PRDMP card is as follows:

6.3.4.1.1 Header Information

code_name ver probid knod nps rnr (A8,3x,A5,A19,I11,1x,I20,1x,I15) (or (A8,3x,A5,A19,I5,1x,I15,1x,I15) if knod and nps are smaller), where

code_name	is the name of the code, MCNP6.

ver	is the version, e.g., 6.3.
probid	is the date and time when the problem was run and, if it is available, the designator of the machine that was used.
knod	is the dump number.
nps	is the number of histories that were run.
rnr	is the number of pseudorandom numbers that were used.

One blank followed by columns 1–79 of the problem identification (1x,A79) line, which is the first line in the problem's input file.

`ntal n npert m` (A5,I11,A6,I11) (or (A5,I5,A6,I5) if `n` and `m` are smaller), where

n	is the number of tallies in the problem.
m	is the number of perturbations in the problem.

List of the tally numbers, on as many lines as necessary. (16I5[1])

6.3.4.1.2 Tally Information

If there are tallies in the problem, the following information is written for each tally:

`tally m i j k` (A5,I4[2],T20,I21,2I5) where

m	is the problem name of the tally, one of the numbers in the list after the `ntal` line.
i	If $i > 0$, then i is the particle type: 1=N, 2=P, 3=N+P, 4=E, 5=N+E, 6=P+E, 7=N+P+E, where N=neutron, P=photon, E=electron. If $i < 0$, then i is the number of particle types and the next `MCTAL` line will list which particles are used by the tally.
j	is the type of detector tally with values

0	none,
1	point,
2	ring,
3	pinhole radiograph (`FIP`),
4	transmitted image radiograph (rectangular grid, `FIR`),
5	transmitted image radiograph (cylindrical grid, `FIC`)

k	is tally modifier information with values

0	none
1	for * modifier
2	for + modifier

[1] This field is increased for big problems (cells or surfaces > 99,999).
[2] This field is increased for big problems (cells or surfaces > 99,999).

List of 37 entries indicating which particle types are used by the tally. (40I2). This is only present if particle type value (i) above is negative. Entries follow the order of particles listed in Table 4.3 and are 1 if the particle is present in the tally and 0 if it is not.

The FC card lines, if any, each starting with 5 blanks. (5x,A75)

f n (A2,1x,I7) where

n	is the number of cell, surface, or detector bins. For repeated-structures tallies, the F-bins have an i, j, k index which goes like: (i_1, j_1, k_1), (i_2, j_1, k_1), ..., (i_n, j_1, k_1), (i_1, j_2, k_1), (i_2, j_2, k_1), ..., (i_n, j_2, k_1),..., (i_1, j_m, k_2), (i_2, j_m, k_2), ..., (i_n, j_m, k_2), etc. This order is reflected in the vals section of the MCTAL file.

List of the cell or surface numbers, on as many lines as necessary. (11I7[3]) If a cell or surface bin is made up of several cells or surfaces, a zero is written. This list is omitted if the tally is a detector tally.

d n (A2,1x,I7) where

n	is the number of total vs. direct or flagged vs. unflagged bins. For detectors, n = 2 unless there is an *ND* on the F5 card; for cell and surface tallies, $n = 1$ unless there is an SF or CF card.

u n or ut n or uc n (A2,1x,I7) where

n	is the number of user bins, including the total bin if there is one. If there is only one unbounded bin, n = 0 instead of 1. If there is a total bin, the line begins with "ut". If there is cumulative binning, the line begins with "uc". These conventions concerning a single unbounded bin and the total bin also apply to the s, m, c, e, and t lines below.

s n or st n or sc n (A2,1x,I7) where

n	is the number of segment bins. If the tally is a radiograph tally, then a list of the bin boundaries will be printed.

m n or mt n or mc n (A2,1x,,I7) where

n	is the number of multiplier bins.

c n f or ct n f or cc n f (A2,1x,I7,I4) where

n	is the number of cosine bins.

[3] This field is increased for big problems (cells or surfaces > 99,999).

f	is an integer flag. If f = 0 or is absent, the cosine values in the following list are histogram bin upper boundaries. Otherwise they are the discrete points where the tally values ought to be plotted and the values are not divided by the bin widths (see NONORM, §6.3.3.5). The e and t entries have similar flags. This integer flag is not written by MCNP6 but can be added to hand-written MCTAL files.

List of cosine values, on as many lines as necessary. (6ES13.5)

e n f or et n f or ec n f (A2,1x,I7,I4) where

n	is the number of energy bins.

List of energy values, on as many lines as necessary. (6ES13.5)

t n f or tt n f or tc n f (A2,1x,I7,I4) where

n	is the number of time bins.

List of time values, on as many lines as necessary. (6ES13.5)

vals values (A4,4(ES13.5,F7.4)) or vals pert values (A10,4(ES13.5,F7.4)) where

values	is a list of tally value-error data pairs, on as many lines as necessary. The order of the values is that of a 9-dimensional Fortran array if it were dimensioned (2,NT,NE,...,NF) where NT is the number of time bins, NE is the number of energy bins, ..., and NF is the number of cell, surface, or detector bins. In other words, time bins are under energy bins, which are under cosine bins, ..., which are under the cell, surface, or detector bins. Values printed in this list are exactly the same as those in the problem's OUTP file.

tfc n jtf (A4,I4,8(1x,I7)) (or (A4,I11,8(1x,I11)) if bins are large), where

n	is the number of sets of tally fluctuation data.
jtf	is a list of 8 numbers, the bin indexes of the tally fluctuation chart bin (see TF card)

nps mean error fom (1x,I14,3ES13.5) (or (1x,I20,3ES13.5) for large problems), where

nps	is the histories run at the time of the TFC dump
mean	is the mean of the tally
error	is the tally error
fom	is the tally figure of merit

This is the end of the information written for each standard tally.

6.3.4.1.3 Superimposed Mesh Tally Type A Information

If a problem contains a TMESH tally, the ntal entry in the general tally information will reflect the numbers of any TMESH tallies, and the following information is written:

tally nugd -j -j8 (A5,3I5), where

nugd	is the mesh tally number
j	is the number of particles in the mesh tally
j8	is the mesh type:

1	rectangular
2	cylindrical
3	spherical

List of 37 entries indicating which particle types are used by the tally. (40I2). Entries follow the order of particles listed in Table 4.3 and are 1 if the particle is present in the tally and 0 if it is not.

f mxgc 0 ng1 ng2 ng3 (A2,I8,4I5), where

mxgc	is the total number of spatial bins (or "voxels") in the TMESH tally.
ng1	is the number of bins on the CORA card
ng2	is the number of bins on the CORB card
ng3	is the number of bins on the CORC card

List of CORA bin values on as many lines as necessary (6ES13.5).

List of CORB bin values on as many lines as necessary (6ES13.5).

List of CORC bin values on as many lines as necessary (6ES13.5).

d 1 (A2,I8)

u 1 (A2,I8)

s mxgv (A2,I8), where

mxgv	is the number of S bins on the tally from different keywords. For example, mxgv = 1for "RMESH1:$\mathscr{P}$ FLUX" and mxgv = 3 for "RMESH3 TOTAL DE/DX RECOL"

m 1 (A2,I8)

c 1 (A2,I8)

e 1 (A2,I8)

t 1 (A2,I8)

vals values (4(ES13.5,f7.4)), where

values	is a list of tally value-error data pairs, on as many lines as necessary. The ordering is similar to that of the standard tally ordering but with fewer bins. In a RMESH tally with 3 bins on the CORA card (x), 2 bins on the CORB card (y), and 2 bins on the CORC card (z), the ordering is as follows: (x_1, y_1, z_1), (x_2, y_1, z_1), (x_3, y_1, z_1), (x_1, y_2, z_1), (x_2, y_2, z_1), (x_3, y_2, z_1), (x_1, y_1, z_2), (x_2, y_1, z_2),..., (x_3, y_2, z_2). In the event that there are multiple S-bins, the CORA, CORB, and CORC coordinates for each S-bin are grouped. In other words, the CORA bins are under the CORB bins, which are under the CORC bins, which are under the S-bins.

6.3.4.1.4 KCODE Information

If a MCTAL file is written during a KCODE problem, the following information is included:

kcode kcz ikz l (A5,3I5) (or (A5, 3I10) if kcz or ikz $>$ 9999), where

kcz	is the number of recorded KCODE cycles.
ikz	is the number of settle cycles.
l	is the number of variables provided for each cycle (set in code to 19).

List of KCODE quantities on as many lines as necessary (5ES12.6). The quantities are: $k_{\text{eff.}}$(collision), $k_{\text{eff.}}$(absorption), $k_{\text{eff.}}$(track length), prompt removal lifetime (collision), prompt removal lifetime (absorption), average collision $k_{\text{eff.}}$, average collision $k_{\text{eff.}}$ standard deviation, average absorption $k_{\text{eff.}}$, average absorption $k_{\text{eff.}}$ standard deviation, average track length $k_{\text{eff.}}$, average track length $k_{\text{eff.}}$ standard deviation, average col/abs/trk-len $k_{\text{eff.}}$, average col/abs/trk-len $k_{\text{eff.}}$ standard deviation, average col/abs/trk-len $k_{\text{eff.}}$ by cycles skipped, average col/abs/trk-len $k_{\text{eff.}}$ by cycles skipped standard deviation, prompt removal lifetime (col/abs/trk-len), prompt removal lifetime (col/abs/trk-len) standard deviation, number of histories used in each cycle, col/abs/trk-len $k_{\text{eff.}}$figure of merit (1).

Details:

(1) The col/abs/trk-len $k_{\text{eff.}}$figure of merit is only printed if the *mct* option on the PRDMP card is equal to 1. If the MCTAL file is written with *mct* $= -1$or -2, this value will always be zero.

6.4 Tally Plotting Examples

6.4.1 Example of Use of COPLOT

Assume all parameter-setting commands [§6.3.3.5] have been previously defined. The following input line will put two curves on one plot:

```
RUNTPE A.h5 COPLOT RUNTPE B.h5
```

The first curve will display tally data from RUNTPE file "A.h5" and the second curve will display tally data from the RUNTPE file "B.h5" for the same tally number. Unless reset, MCPLOT will continue to read from B.h5 for subsequent plot commands.

If there is a current RUNTPE file read, the following commands can be entered:

```
XLIMS min max TALLY 11 COPLOT RMCTAL AUX TALLY 41 &
COPLOT RUNTPE A.h5 TALLY 1
```

These commands change the upper and lower limits of the x-axis to max and min, respectively and plots tally 11 from the current RUNTPE. The first COPLOT entry requests the plotter to draw a second curve (tally 41) from the MCTAL file named "AUX". The third curve is requested on the next line (note the & at the end of the first line). This final curve is tally 1 from The RUNTPE file named "A.h5". Any subsequent plot requests will use data from RUNTPE A.h5 unless reset.

The command

```
TALLY 24 NONORM FILE COPLOT TALLY 44
```

will output tally 24 and tally 44 to the graphics metafile.

6.4.2 Radiography Tally Contour Plot Example

Tally output may be plotted as 2-D color contours from either MCTAL or RUNTPE files. For example, a radiography tally with s and t-axes specified on FS and C cards can be plotted with the **MCNP6 Z** execute option, as described below.

The following example is a radiograph of a 4-cm-radius, 1-cm-thick ^{238}U disc with an embedded 4-mm-void sphere and skew-oriented 1-cm × 1-cm × 8-mm box. The input file is given in Listing 6.1.

Listing 6.1: example_radiograph_tally.mcnp.inp.txt

```
Radiography Tally
1 5 -25.0 -1  4 5 imp:p=1
2 0        1 -2   imp:p=1
3 0           2   imp:p=0
4 0          -4   imp:p=1
5 0          -5   imp:p=1

1 RCC     0 0 0 0 0 1 4
2 RPP   -100 100  -100 100  -100 100
4 SPH    3 0  0.5  0.4
5 BOX   -1 1 0.1  0.6 0.8 0  -0.8 0.6 0  0 0 0.8

mode p
nps     100 5
sdef    pos=0 0 -20 axs=0 0 1 rad=d1 ext=0 vec=0 0 1 dir=d2 erg=6
si1      0 0.1
sp1    -21   1
si2     -1   1
sp2    -31   1
m5     92238 1
```

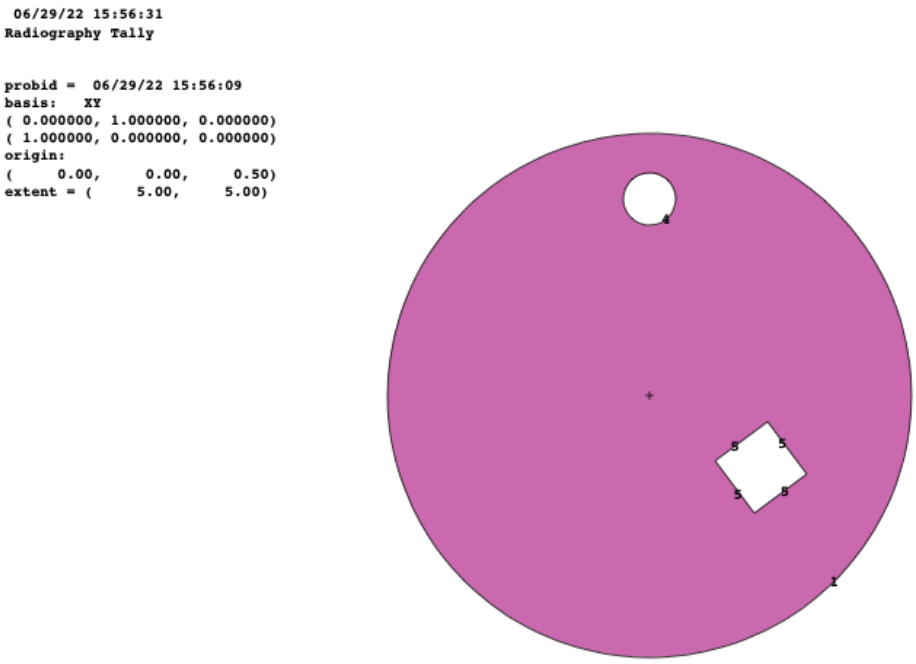


Figure 6.5: Geometry plot of radiograph example.

```
print
prdmp    2j 1
fir5:p   0 0 10 0 0 0 -100 0 100 0
fs5    -10. 99i 10.
c5     -10. 99i 10.
```

The x-y plot of this geometry is given in Fig. 6.5. To get the contour plot, first run the input, then type the following MCNP6 execution line command:

```
MCNP6 Z RUNTPE=filename
```

The contour plots may also be read from a `MCTAL` file instead of the `RUNTPE` file [§6.3 introduction]. When the code presents the `MCPLOT>` prompt, enter the two tally dimensions corresponding to the horizontal and vertical axes of the radiography plot with the `FREE` command [§6.3.3.7]. For example to see the radiography tally in Listing 6.1, which has image bins defined by the "S" tally dimension (from the `FS` entries) and the "C" tally dimension (from the `C` entries), one would simply enter "`FREE SC`" at the `MCPLOT>` prompt. The plot axes are then oriented according to the right-hand rule between the reference direction on the `FIR` entry and the global coordinates. For more information on how this orientation is determined, see Section 5.9.1.3.3.

The results are plotted in Figure 6.6. The embedded sphere and box are seen plainly in the disc.

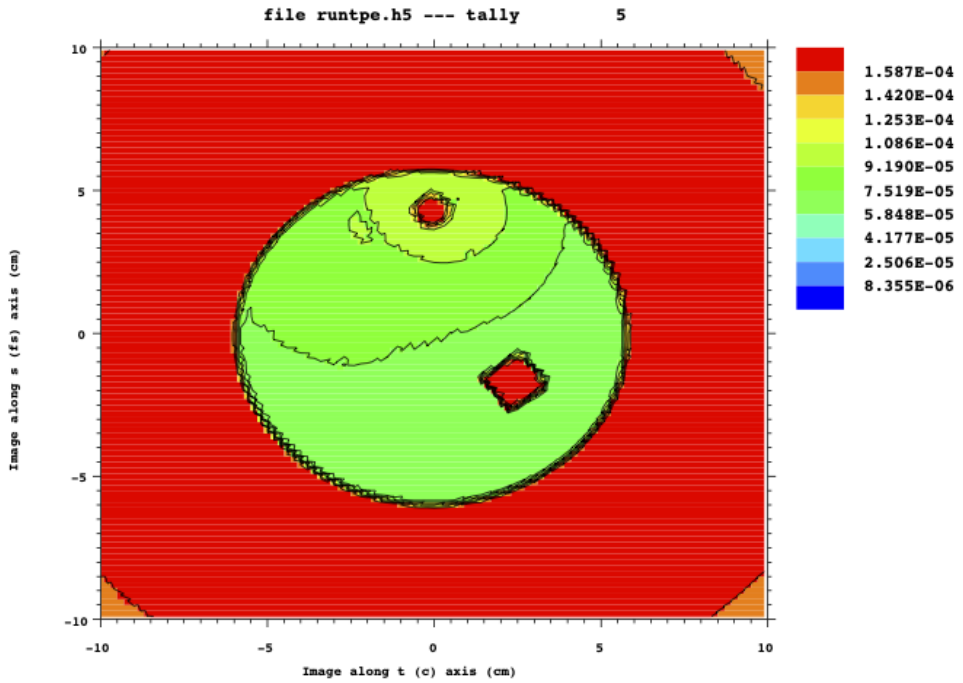


Figure 6.6: Scattered photon radiographic image of ^{238}U disc.

Recall that the possible tally dimensions are FDUSMCET [§5.9.19, §6.3.3.7].

Thus, if the radiography tally has other bins such as energy or time bins, the plot for these bins can be examined with the "FIXED" command [§6.3.3.7] following the "FREE SC" command.

6.4.3 TMESH Mesh Tally Plot Examples

TMESH mesh tallies may be plotted either in the MCNP6 tally plotter (MCPLOT) from MCTAL files or superimposed over geometry plots in the geometry plotter (PLOT) from RUNTPE files.

6.4.3.1 MCPLOT TMESH Mesh Tally

Listing 6.2 is a critical configuration of seven identical barrels of fissionable material.

Listing 6.2: example_plotting_tmesh_tally.mcnp.inp.txt

```
cylinders containing critical fluid in macrobody hex lattice
1 1 -8.4 -1     u=1 imp:n=1
2 0      -2     u=1 imp:n=1
3 2 -2.7 -3 1 2 u=1 imp:n=1
4 3 -.001 3     u=1 imp:n=1
10 3 -.001 -6 lat=2 u=2 imp:n=1 fill=-2:2 -2:2 0:0
                                2 2 2 2 2
                                2 2 1 1 2
                                2 1 1 1 2
                                2 1 1 2 2
                                2 2 2 2 2
11 0       -8           imp:n=1 fill=2
50 0        8           imp:n=0

1 rcc 0  0 0  0 12 0  5
2 rcc 0 12 0  0 8  0  5
3 rcc 0 -1 0  0 22 0  6
6 rhp 0 -1 0  0 22 0  9 0 0
8 rcc 0 -1 0  0 22 0  30

m1    1001 5.7058e-2   8016 3.2929e-2
     92238 2.0909e-3  92235 1.0889e-4
m2   13027 1
m3   7014 .8 8016 .2
c
fc14 total keff in each element
f4:n (1<10[-2:2 -2:2 0:0]<11)
fq4 f m
sd4 1 24r
f14:n   (1<10[-1 1 0])  (1<10[0 1 0])
        (1<10[-1 0 0])  (1<10[0 0 0])  (1<10[1 0 0])
        (1<10[0 -1 0])  (1<10[1 -1 0]) t
fq14 f m
sd14 1 7r
tf14 4
fm14 (-1 1 -6 -7)
print -160
prdmp 2j 1
kcode 1000 1 10 50
```

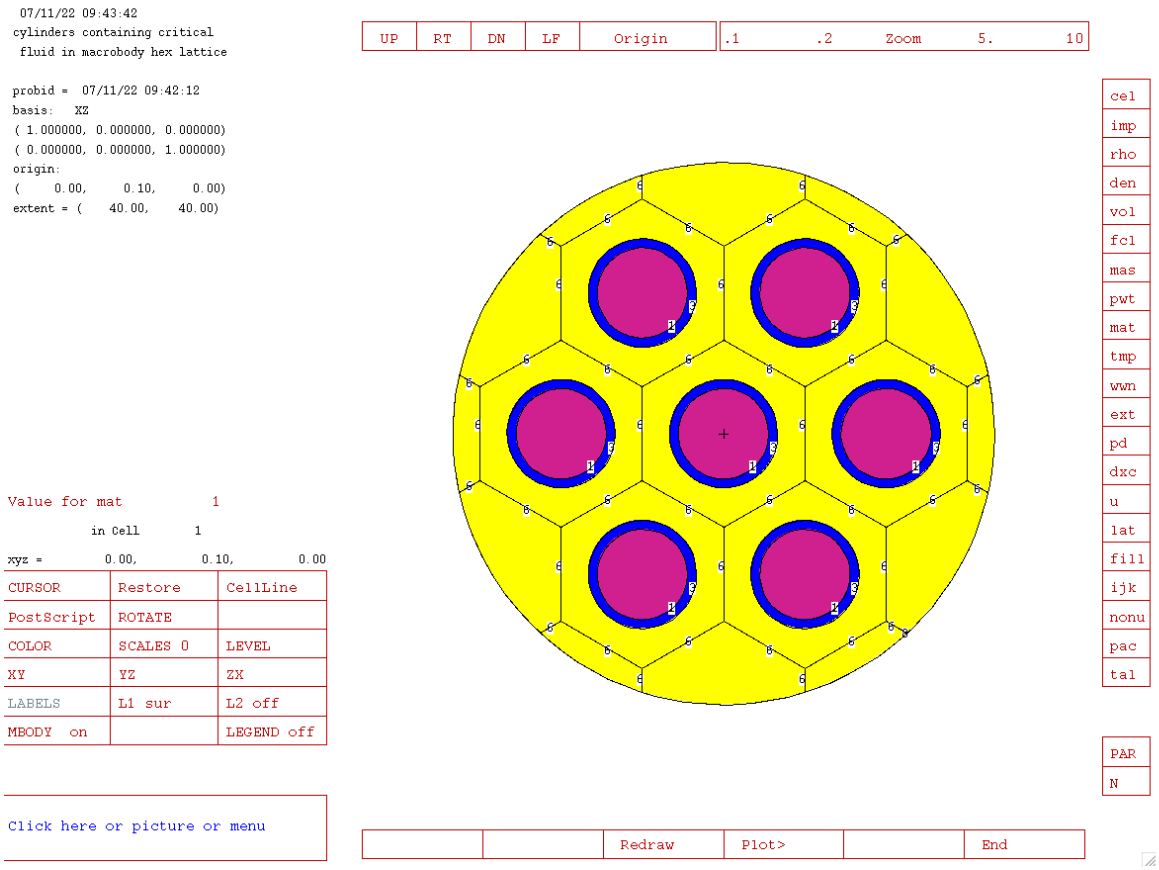


Figure 6.7: Geometry of the seven-barrel problem.

```
ksrc 0 6 0  18 6 0  -18 6 0  9 6 15  -9 6 15  9 6 -15  -9 6 -15
tmesh
 rmesh12
 cora12  -30. 99i 30.
 corb12    0. 12.
 corc12  -30. 99i 30.
endmd
```

The geometry is shown in Figure 6.7.

The mesh tally is generated from a `MCTAL` file in the **`MCPLOT`** tally plotter, called with the **`MCNP6 Z`** input line. The plot command to enter at the `MCPLOT>` prompt to read the MCTAL file and plot the `TMESH` tally is:

```
rmctal mctal_filename tal 12 free ik
```

Figure 6.8 shows the resulting `TMESH` tally of the configuration.

6.4.3.2 Superimposed Geometry Plot TMESH MESH Tally

Figure 6.9 shows the `TMESH` mesh tally results superimposed over the geometry plot. First, the `RUNTPE` file

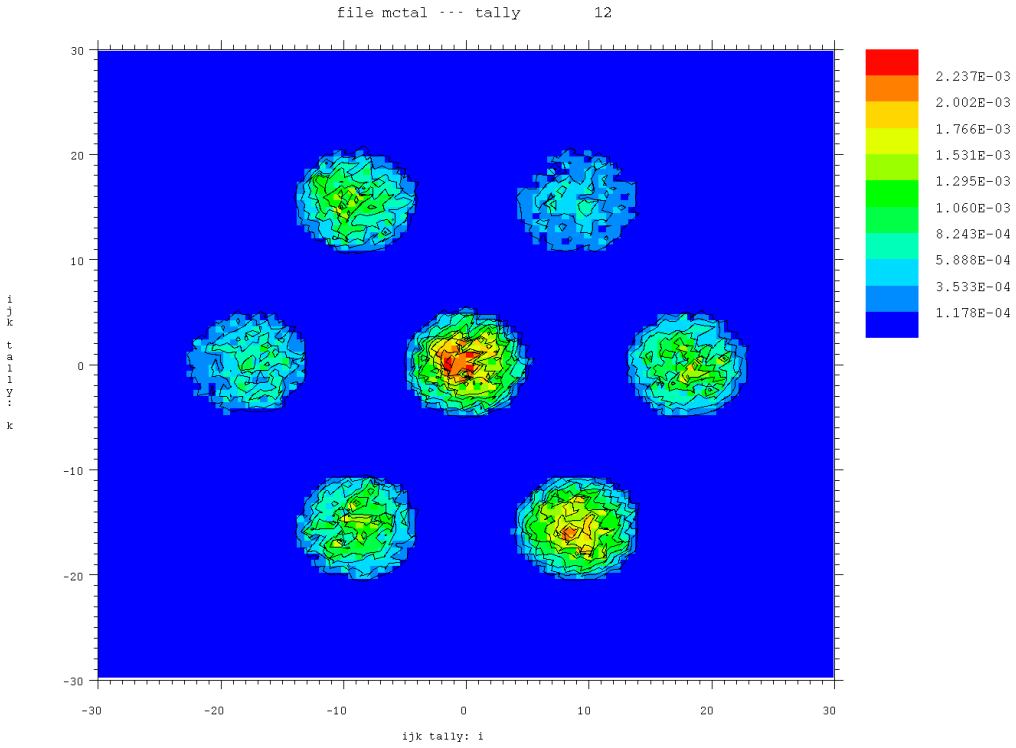


Figure 6.8: Mesh tally of barrel geometry.

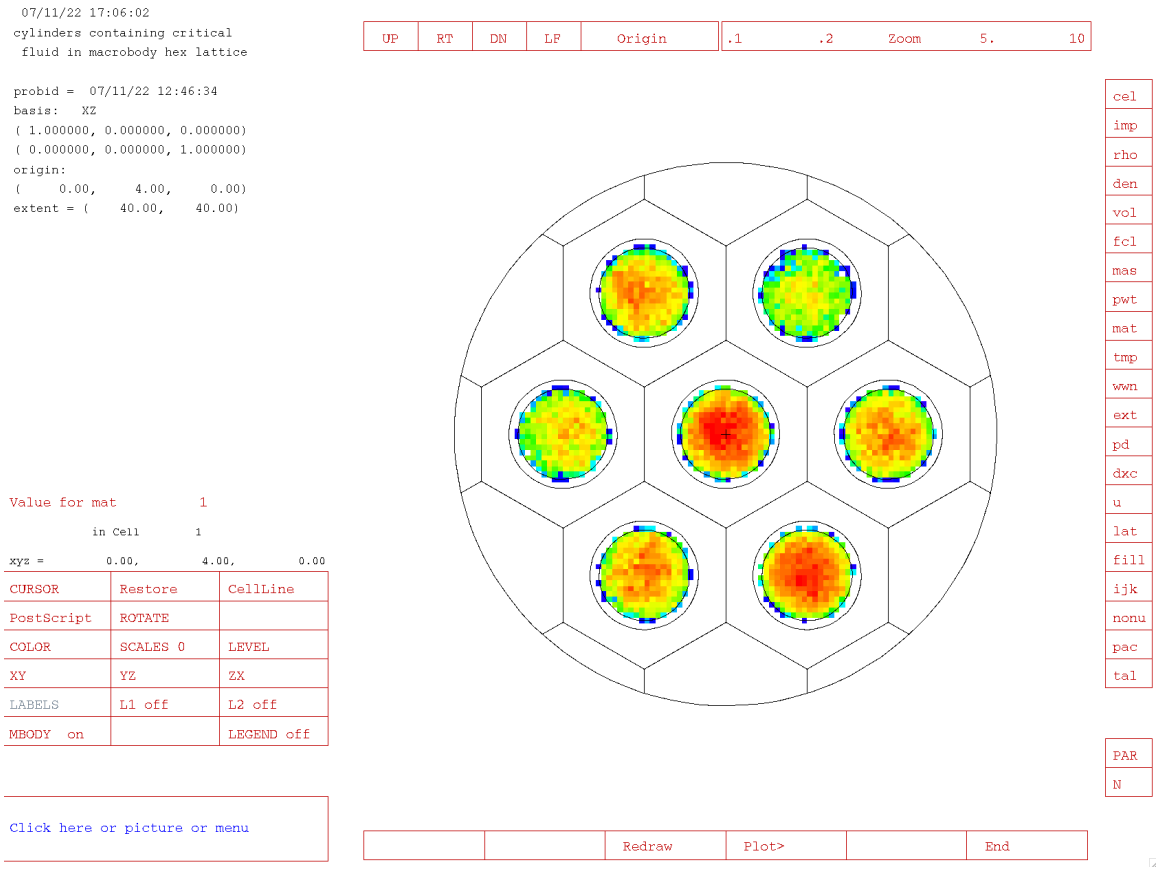


Figure 6.9: TMESH Mesh plot superimposed on geometry plot.

must be loaded using the **MCNP6 Z** option:

```
mcnp6 z r=runtpe_filename
```

Then, at the MCPLOT> prompt, the MCNP geometry plotter is invoked with the PLOT command.

Next, the MCNP geometry needs to be oriented such that the mesh tally is in view. For the geometry of Figure 6.9, click on the lower left box ("Click here or picture...") and enter the following command:

```
or 0 4 0 py 4 ex 40 la 0 0
```

To plot the TMESH tally results, make the mesh tallies the "Edit" quantity by clicking the tal button at the bottom of the right-hand control bar (§6.2.3.3):

Click tal and check if the tally number expected shows on the left-hand information pane (see entry for "Value for *var*" in §6.2.3.2). If the tally number presented in the information pane is not the mesh tally of interest, click the N button below tal to cycle available tallies.

Note: The format of the information pane for the tallies sometimes makes the tally number and tally value appear as a single digit like so: tal1114.6784-5. In these cases, the plotter is not broken.

After the "Edit" quantity is specified, the color parameter must be changed to reflect the selected tally. Change the color parameter by clicking the COLOR button in the lower left control box twice. The first click will change the text in the box to "COLOR off" and, for this example, the next click will change it to read "COLOR tal12". Once the color parameter is set, the Redraw button in the bottom-middle of the plotter must be clicked.

In addition, the actual mesh tally voxel borders can be displayed by clicking CellLine button and cycling through the options to get either "MeshTaly" (which draws mesh tally grid lines over the results) or MT+Cell (which draws mesh tally grid lines and cell surface lines over the plot).

6.4.4 MCPLOT FREE Command Examples

6.4.4.1 Example 1

Consider the input shown in Listing 6.3, which is a simple 3x3x2 lattice of water spheres surrounded by air with a monoenergetic and monodirectional photon source incident on the lattice parallel to the $i-$dimension in the top layer and bisecting the j-dimension. Thus, we would expect to see a higher tally value on the [-1:1 1 1] lattice elements (refer to §5.9.1.5 for information on lattice tally indexing).

Listing 6.3: example_mcplot_free.mcnp.inp.txt

```
Example lattice tally
10 100 -1       -1 u=1       imp:p=1
20 200 -0.00125  1 u=1       imp:p=1
30 0            -2 u=2 lat=1 imp:p=1
      fill=-1:1 -1:1 0:1
      1 17r
```

```
40 0           2 u=2          imp:p=0
50 0          -3 fill=2       imp:p=1
60 0           3 -4           imp:p=1
70 0           4              imp:p=0

1 so  0.75
2 rpp -1 1 -1 1 -1 1
3 rpp -3 3 -3 3 -1 3
4 so  100

mode p
nps 1e4
sdef par=p dir=1 vec=1 0 0 pos=-4 0 2
c
f4:p (10<30[-1:1 -1:1 0:1]<50)
c
prdmp 2j 1
print -128
m100 1001 2 8016 1
m200 7014 0.78 8016 0.22
```

First, each layer (in the k-dimension) can be examined by first calling the **MCPLOT** module with the **MCNP6 Z** execution line then entering:

```
tally 4 free ij 3x3 fixed k=1
```

which gives Figure 6.10.

The second layer can be seen with:

```
tally 4 free ij 3x3 fixed k=2
```

which gives Figure 6.11.

It can then be useful to examine a single row of lattice elements. In this case, the elements directly in the line of the source beam may be of interest. The way to do this is somewhat non-intuitive. To get a 1-dimensional plot of the three elements directly inline with the beam, the following command is entered at the `MCPLOT>` prompt:

```
free i 3x3 fixed j=2 fixed k=2
```

Which gives Figure 6.12.

The first command in the sequence, `FREE i 3x3`, preps the **MCPLOT** module to expect a `FIXED` command for the other two tally dimensions and the lattice positions of interest lie in the 3x3 plane on the second level of the lattice. The `FIXED j=2` command requests the second slice into the j-dimension and likewise the second `FIXED` command sets the k-index to the second layer. Referring to Listing 6.3 the lattice is specified as "fill= -1:1 -1:1 0:1" but it is accessed with the location of the desired position in the 1-indexed Fortran array (i.e., the middle slice of the i and j-dimensions is "2", not "0").

When considering the resulting `MCTAL` file (shown in Listing 6.4), the offset into the `MCTAL` output can be easily determined. The description of the `MCTAL` file format is in Section 6.3.4 and describes the order of the values in the file.

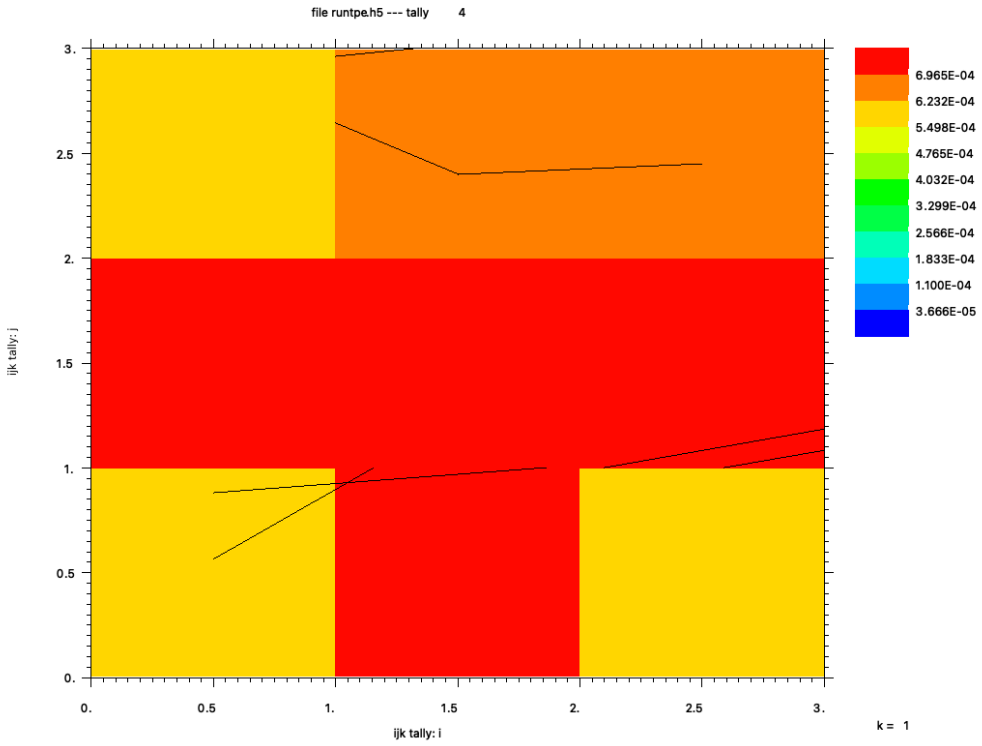


Figure 6.10: `FREE` Command Bottom Layer (`FIXED K=1`)

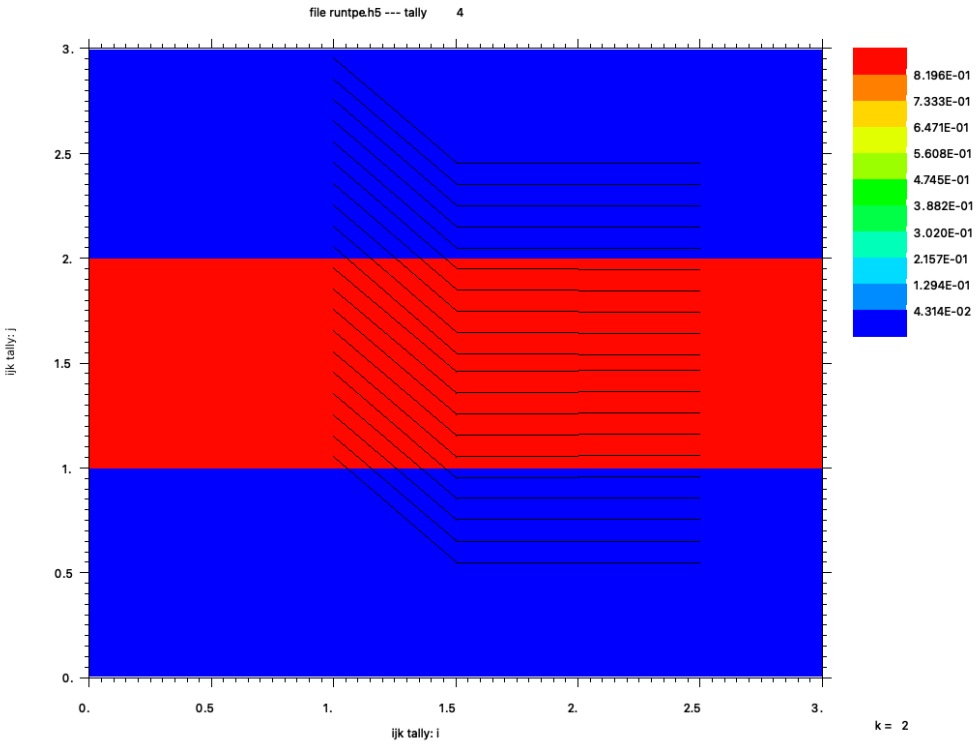


Figure 6.11: `FREE` Command Top Layer (`FIXED K=2`)

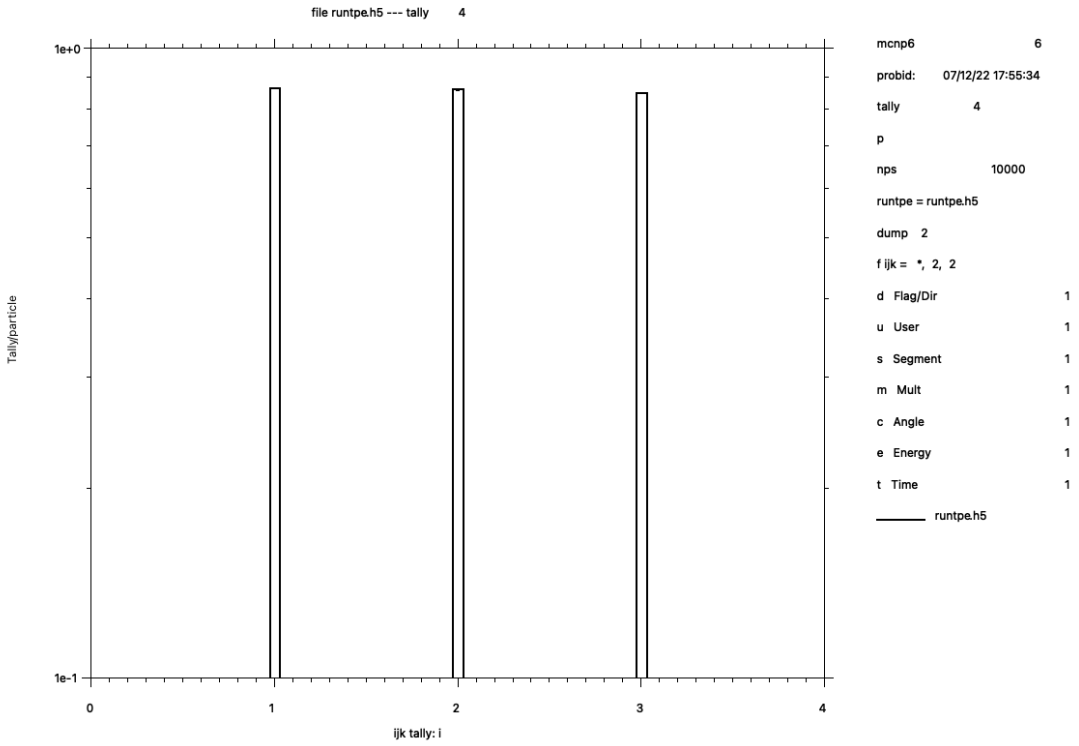


Figure 6.12: 1-Dimensional Slice of the 3x3x2 Lattice Tally

Listing 6.4: example_mcplot_free.mcnp.mctal.txt

```
mcnp6      6     07/12/22 17:55:34     2           10000          115043
 Example lattice tally
ntal     1
    4
tally    4                   -1    0    0
 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
f       18
     10     10     10     10     10     10     10     10     10     10     10
     10     10     10     10     10     10     10
d        1
u        0
s        0
m        0
c        0
e        0
t        0
vals
  6.07366E-04 0.3453  7.10948E-04 0.2987  5.62171E-04 0.3431  8.43411E-04 0.2734
  1.20640E-03 0.2235  1.29530E-03 0.2202  6.13934E-04 0.3156  6.39769E-04 0.2928
  6.65327E-04 0.3148  4.24137E-04 0.3589  1.72955E-03 0.1978  1.47042E-03 0.2014
  8.63171E-01 0.0014  8.58873E-01 0.0019  8.48629E-01 0.0023  9.79779E-04 0.2617
  1.49976E-03 0.2087  1.42213E-03 0.2137
tfc   10       1       1       1       1       1       1       1       1
          1000  0.00000E+00  0.00000E+00  0.00000E+00
          2000  4.46765E-04  9.99750E-01  3.06949E+03
          3000  5.12752E-04  7.16064E-01  3.97030E+03
          4000  5.62662E-04  5.82767E-01  4.49529E+03
          5000  7.71062E-04  4.49790E-01  5.96871E+03
          6000  6.42552E-04  4.49827E-01  5.03891E+03
          7000  5.94551E-04  4.23154E-01  4.83527E+03
          8000  5.20232E-04  4.23175E-01  4.20527E+03
          9000  6.02741E-04  3.67689E-01  4.97285E+03
         10000  6.07366E-04  3.45330E-01  5.10550E+03
```

In the case of the lattice tally in this example, there are no bins in the `FDUSMCET` array other than the `F`-bins which are indexed with their ijk locators. The values go as follows: (i_1, j_1, k_1), (i_2, j_1, k_1), (i_3, j_1, k_1), (i_1, j_2, k_1), (i_2, j_2, k_1), (i_3, j_2, k_1),..., (i_1, j_3, k_2), (i_2, j_3, k_2), (i_3, j_3, k_2). Following this ordering, it's clear that the slice of interest for this problem is the 13th to 15th value-error pairs. Thus, we expect an offset into the array of 12 value-error pairs. Examining the way the values are ordered, first we step over all the values where $k_n = k_1$to get to the start of the second layer: $(3 \times 3) \times (2 - 1) = 9$. Then, we need to step over all the values where $k_n = k_2$ and $j_m = j_1$ to get to the start of the j_2 values in the second layer: $3 \times 1 = 3$. Thus, our offset is $9 + 3 = 12$, as evident in the `MCTAL` file with the largest value-error pairs on line 21. In other words, we can consider the offset indexing of the value-error pairs to be from 0, where the values on the `FIXED` command are from 1, so one can account for the off-by-one error and calculate the offset like:

$$(i\text{-dimension} \times j\text{-dimension}) \times (\text{Fixed } k\text{-dimension} - 1) + j\text{-dimension} \times (\text{Fixed } j\text{-dimension} - 1) \quad (6.1)$$

The above equation can be generalized with an understanding of the way the values in the `MCTAL` files are presented, and the desired results to plot.

If the j and k-indices had not been specified, their default value of 1 is assumed, which results in an offset of 0.

Finally, the normalization of the values comes from the minimum and maximum value of the 3 i-bin values shown in the plot.

6.4.4.2 Example 2

The command

```
FREE IJ 10x30 ALL FIXED K=60
```

specifies a 10×30 2-D contour plot, which corresponds to a lattice tally with 10 i-bins, 30 j-bins, and at least 60 k-bins. The k-index is specified using the `FIXED` command, which sets the offset into the `F`-bins as

$$(i\text{-dimensions} \times j\text{-dimensions}) \times (\text{Fixed } k\text{-dimension} - 1) = (10 \times 30) \times 59 = 17,700 \tag{6.2}$$

In this case, the `ALL` option on the `FREE` command causes the contour range to taken from all of the `F`-bin tally values.

6.4.5 Photonuclear and Photoatomic Cross-section Plots

MCNP6 can plot photonuclear data in addition to the photoatomic data of MCNP6. A list of the special reaction numbers available in the MCNP code (in addition to the standard ENDF reaction numbers found in the ENDF-6 manual [45]) is in Table 5.19. For photonuclear reactions with $R > 4$, refer to the list of standard ENDF reaction numbers.

The photonuclear yields (multiplicities) for various secondary particles are specified by adding 1000 times the secondary particle number to the reaction number. For example, 31,001 is the total yield of deuterons (particle type D = 31), 34,001 is the total yield of alphas (particle type $\alpha = 34$), and 1018 is the total number of neutrons (particle type N = 1) from fission.

The example input in Listing 6.5 shows a basic input that photoatomic and photonuclear cross sections can be plotted from.

Listing 6.5: example_photonuclear_photoatomic_plotting.mcnp.inp.txt

```
photonuclear and photoatomic cross section plotting example
1 100 -1 -1 imp:n=1
2 0       1 imp:n=0

1 so 10

m100 82208 1 6012 1
     pnlib=24u
     plib=84p
c
c Turn on photonuclear physics
phys:p 3j -1
mode p n
sdef par=p
```

When the file is run with the **`mcnp6 ixz`** execution line, the code processes the cross sections and calls the **`MCPLOT`** module. The user is presented with the `MCPLOT>` prompt.

To find out which reactions are available for a particular nuclide or material, enter an invalid reaction number with the `MT` command, followed by the `XS` command and the nuclide or material of interest. For example, determining the allowed photonuclear reactions and available yields for carbon-12 is accomplished with:

```
mt 99 xs 6012.24u
```

The resulting output is shown in Listing 6.6.

Listing 6.6: Console Output For Requesting Photonuclear MT=99 on Carbon-12

```
 mcplot>
mt 99 xs 6012.24u
 setting particle type to p

 no data found for reaction  99

  6012.24u allowable reactions:
       1       2       3       4       5      50     600
    1001    1004    1005    1050    2001    2004    2005    9001    9004    9005
    9600   31001   31004   31005   34001   34004   34005
```

The console output shows the reaction numbers in Table 5.19 in addition to numbers 5, 50, and 600, plus a series of yields. To determine the meaning of the non-yield numbers, the ENDF-6 manual should be referenced. There, we find that 5 is a catchall for reactions not defined by other MT numbers, 50 is the production of a neutron, leaving the residual nucleus in the ground state, and 600 is the production of a proton, leaving the residual nucleus in the ground state.

We can plot the total photonuclear cross sections for the carbon, lead, and material 100 with the following command:

```
xlim 5 200 ylim 1e-5 1 mt 1 xs 6012.24u cop mt 1 xs 82208.24u cop mt 1 xs m100
```

This command first sets the x and y-axis limits to a convenient value, then coplots the total photonuclear cross sections of each table along with the total material which results in the plot shown in Figure 6.13.

Entering a bad table identifier with the `XS` command (i.e., 12345.67u), will cause MCNP6 to list the available nuclides and library suffixes which can inform the user of the availability of other data to plot. In this example, the .00c and .84p data are both available, and their cross sections, along with the total material cross section can be plotted with:

```
ylim 1e-1 1e7 mt -5 xs 6000.84p cop mt -5 xs 82000.84p cop mt -5 xs m100
```

Which results in Figure 6.14.

6.4.6 Weight-Window-generator Superimposed Mesh Plots

MCNP6 can plot the `WWG` superimposed mesh specified on the `MESH` card in an input file. In the MCNP6 interactive geometry plotter, toggle `CellLine` in the left-hand controls (§6.2.3.2) for the following options:

`CellLine`	Plot constructive solid geometry cells, outlined in black. (DEFAULT)
`No Lines`	Plot cells not outlined in black.

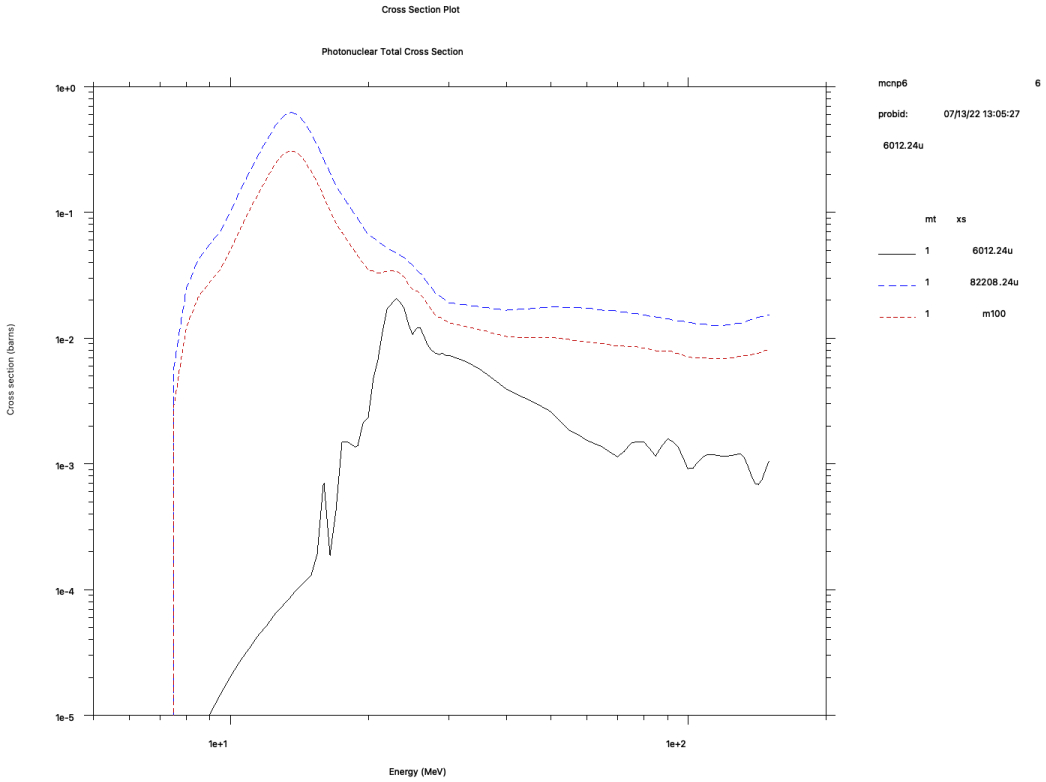


Figure 6.13: Photonuclear cross-section plot.

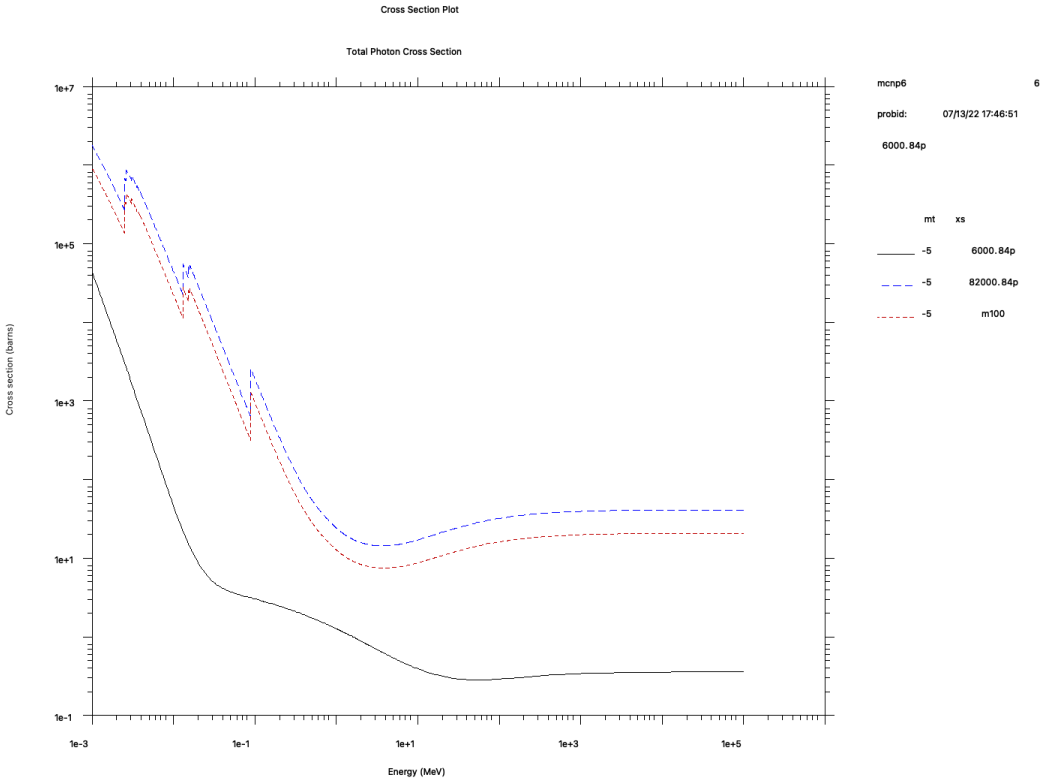


Figure 6.14: Photoatomic cross-section plot.

`WW MESH`	Plot weight-window superimposed mesh without cell outlines.
`WW+Cell`	Plot weight-window superimposed mesh and cell outlines.
`WWG MESH`	Plot weight-window generator mesh.
`WWG+Cell`	Plot weight-window generator mesh and cell outlines.
`MeshTaly`	Plot `TMESH` mesh tally boundaries.

The `CellLine` and `No Lines` options are always available. `WW MESH` and `WW+Cell` are available only when the `WWP` card calls for using a superimposed weight-window mesh (5th entry negative) and a `WWINP` file is provided. `WWG MESH` and `WWG+Cell` are available only when a `MESH` card appears in the input and when the `WWG` card requests superimposed mesh generation (2nd entry is 0). `MeshTaly` and `MT+Cell` are available only when a `TMESH` mesh tally has been requested.

6.4.6.1 Cylindrical Mesh Example

One can generate a plot using the MCNP input file in Listing 6.7 and the interactive plotter command input file in Listing 6.8 with the execution line:

```
mcnp6 i=example_cylindrical_mesh.mcnp.inp.txt com=example_cylindrical_mesh.mcnp.comin.txt ip
```

Listing 6.7: example_cylindrical_mesh.mcnp.inp.txt

```
Demonstration of WWG Plot
1 1 1.0   -1  imp:p 1
2 0        1  imp:p 0

1 rcc 0 0 0  0 10 0  5

mode p
sdef sur 1.3 vec 0 1 0 dir 1 erg 100
m1 1001 2 8016 1
nps 1000
f1:p 1.2
wwg 1 0
mesh geom=cyl origin=0 -1 0  ref=0 .1 0  axs=0 1 0  vec=1 0 0
     imesh 6   iints 7  jmesh 12  jints 7   kmesh 1   kints 3
```

Listing 6.8: example_cylindrical_mesh.mcnp.comin.txt

```
ex 10 lab 0 0 px 0 mesh 4
pause
py 5
pause
```

Or, instead of using the command file (with plot commands in command mode), the interactive plotter can be used:

```
mcnp6 i=example_cylindrical_mesh.mcnp.inp.txt ip
```

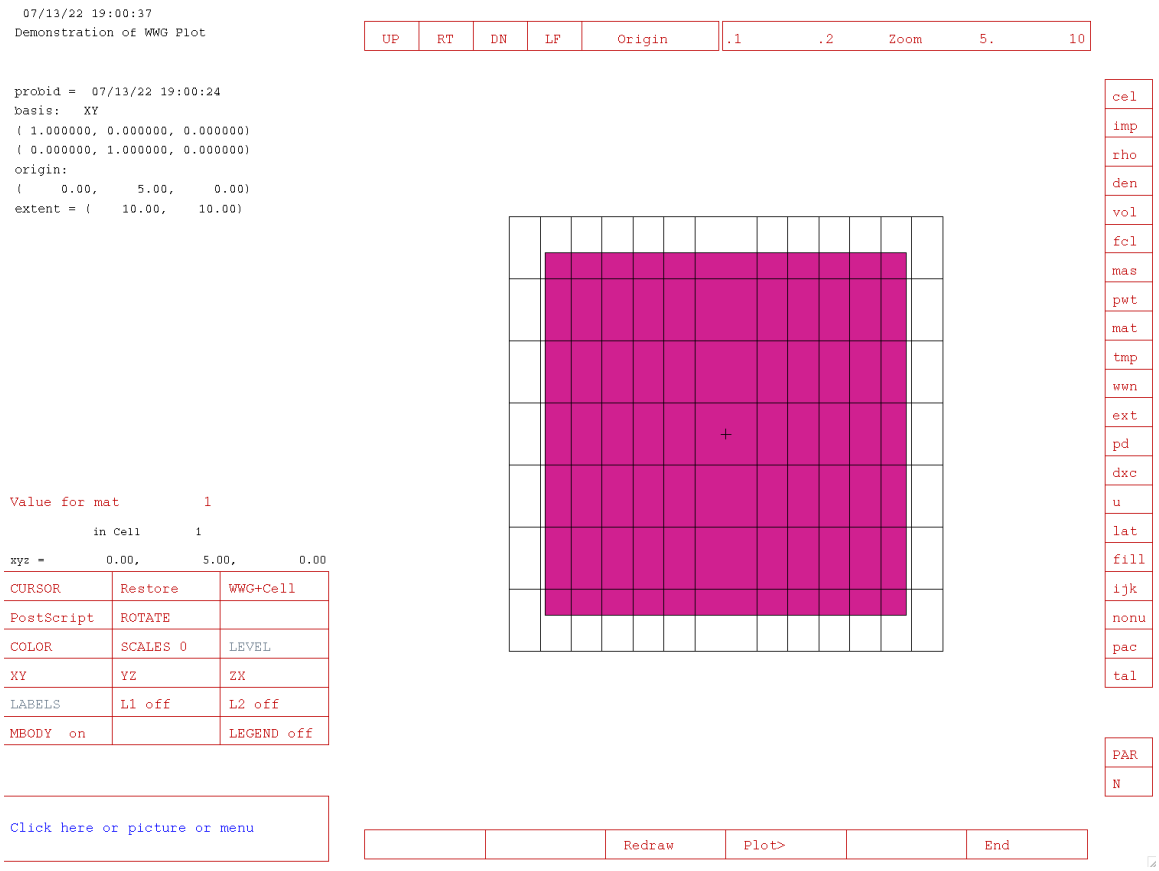


Figure 6.15: WWG mesh plot, axial view.

The superimposed mesh can displayed with the following sequence:

1. Click the `CellLine` button in the lower left-hand control box twice to get `WWG+Cell`
2. Click the `L1 sur` button once to turn off surface labels
3. Click `XY` to get the view equivalent to `px` $= 0$ (axial view, Fig. 6.15)
4. Click "10" on the `Zoom` bar twice to get a 10 times magnification at the origin
5. Click `Origin` then click in the center of the mesh to center the geometry in the plotter window
   (a) The origin information in the top left of the plotter should show approximately $(0, 5, 0)$
6. Click `ZX` to get a view approximately equivalent to `py` $= 5$ (radial view, Fig. 6.16)

6.4.6.2 Spherical Mesh Example With Weight Windows

The spherical mesh geometry may be thought of as a globe where the phi (φ) polar angles are latitude and the theta (θ) azimuthal angles are longitude. The north pole is at $\varphi = 0°$; the south pole is at $\varphi = 180°$; Greenwich (near London) is at $\theta = 0°$ and all the way around the globe at $\theta = 360°$.

The interface for geometry plots of the spherical mesh window boundaries is the same as for cylindrical mesh boundaries. An example input (Listing 6.9) and associated `WWINP` file (Listing 6.10) are given and are attached to this document.

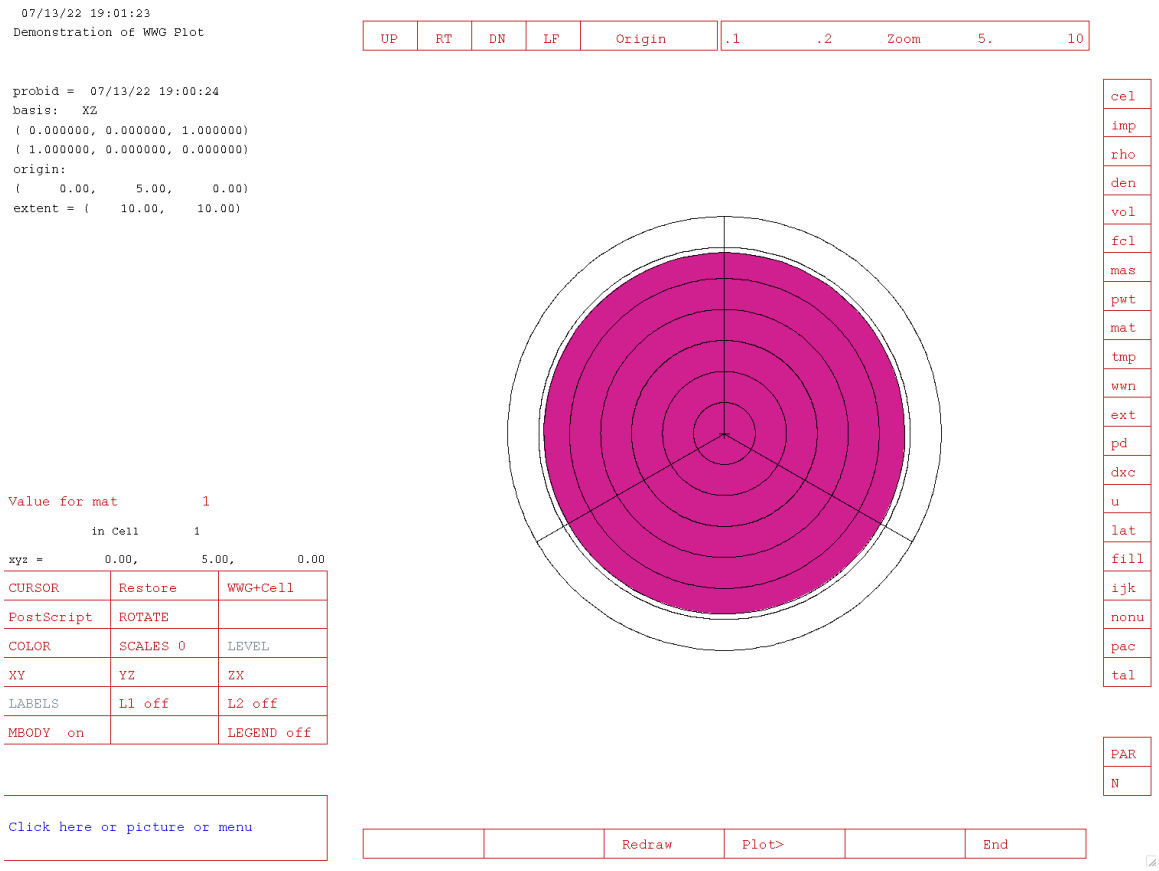


Figure 6.16: WWG mesh plot, radial view.

Listing 6.9: example_spherical_mesh.mcnp.inp.txt

```
Demonstration of Spherical WW Mesh Plot
1 1 1.0   -1   imp:p=1
2 0        1   imp:p=0

1 so 10

mode p
sdef erg=5
m1 1001 2 8016 1
nps 1000
f1:p 1
wwp:p 4j -1
c The lines below are used to generate the WWINP file with the wwp card commented out.
c wwg 1 0
c mesh geom=sph ref=0 0 0 origin=0 0 0
c      axs=0 0 1        $ Reference vector for the polar axis
c      vec=1 0 0        $ Reference vector for the azimuthal axis
c      imesh 3 7.5 10   $ Radii of the mesh, cm
c      jmesh 36 126 180 $ Polar angles (phi), implicit 0 lower bound
c      kmesh 72 306 360 $ Azimuthal angles (theta), implicit 0 lower bound
```

Listing 6.10: example_spherical_mesh.mcnp.wwinp.txt

```
         1         1         2        16                     07/14/22 12:19:24
         0         1
   3.0000       3.0000       3.0000       0.0000       0.0000       0.0000
   3.0000       3.0000       3.0000       0.0000       0.0000       10.000
   10.000       0.0000       0.0000       3.0000
   0.0000       1.0000       3.0000       1.0000       1.0000       7.5000
   1.0000       1.0000       10.000       1.0000
   0.0000       1.0000      0.10000       1.0000       1.0000      0.35000
   1.0000       1.0000      0.50000       1.0000
   0.0000       1.0000      0.20000       1.0000       1.0000      0.85000
   1.0000       1.0000       1.0000       1.0000
   100.00
  0.50000      0.32667      0.94444E-01  0.28333      0.18519      0.68571E-01
  0.59048      0.32500      0.77778E-01  0.37500      0.32500      0.11481
  0.20523      0.16637      0.62500E-01  0.25648      0.17368      0.59459E-01
  0.55833      0.35833      0.10833      0.24815      0.17976      0.90909E-01
  0.63333      0.40667      0.79167E-01
```

To see these weight windows, first run the following MCNP6 execution line:

```
mcnp6 i=example_spherical_mesh.mcnp.inp.txt wwinp=example_spherical_mesh.mcnp.wwinp.txt ip
```

This will load the familiar geometry plotter. The view in Figure 6.17, which is looking down the polar axis (in this case, the global z-axis) is achieved through the following steps:

1. Click the `CellLine` button twice so that `WW+Cell` shows.
2. Click the `L1 sur` button once so it reads `L1 off`.
3. Click the `wwn` button on the right-hand bar, then `COLOR` twice so it reads `COLOR wwn1:p`.

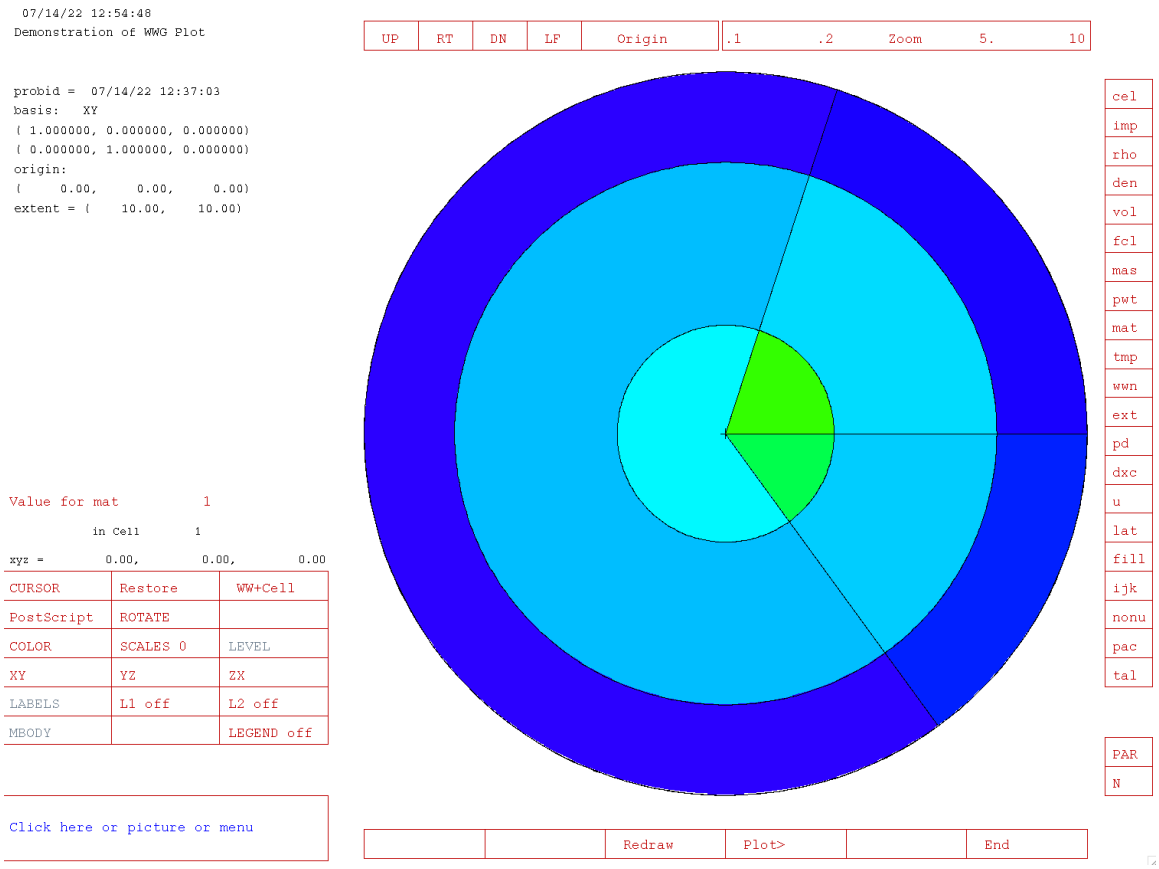


Figure 6.17: View along polar axis at origin showing azimuthal planes at KMESH = 72, 306, and 360 degrees. The azimuthal vector, VEC, is to the right (360 degree plane)

(a) If the mesh has multiple particle types, energies, etc, the indexes can be stepped through by clicking the PAR and N buttons before cycling the COLOR button. Ordering is described in the description of the wwn button in in §6.2.3.3.

4. Click the XY button which draws the plot colored by the values of the weight window mesh.

5. Click 10 in the Zoom bar twice.

The equivalent command input to the PLOT> prompt is:

```
la 0 1 wwn1:p color on la 0 0 mesh 3 ba 1 0 0 0 1 0 ex 10
```

This view is along the polar axis, which is defined by the AXS parameter on the MESH card, and shows the segmentation of the spherical mesh from the azimuthal angles (longitude) that are defined on the KMESH parameter of the MESH card.

The view in Figure 6.18 can be achieved by following the same steps listed above, but clicking the YZ button instead of the XY button.

This view orients the polar reference vector towards the top of the geometry and shows essentially a slice of the cone that is defined by the polar angles (latitude). Clicking in the lower-left box and typing "PX 1", "PX 2", "PX 3" and so on will demonstrate this conical section (see Figure 6.19). This shows the 3 bins that are defined on the JMESH parameter of the MESH card.

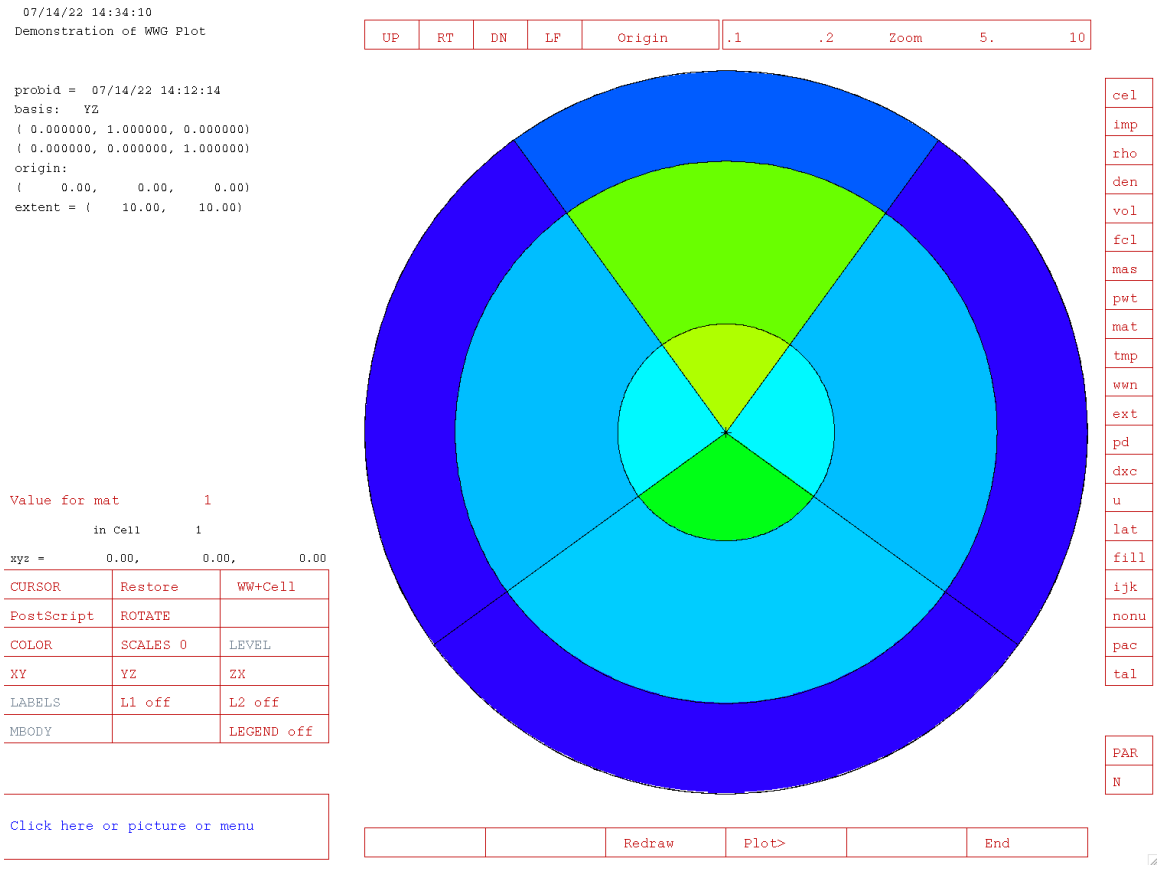


Figure 6.18: Plot view orthogonal to polar axis showing polar bins JMESH = 36 and 126 degrees. The polar axis (0 degrees) is through the center of the mesh towards the top of the plot.

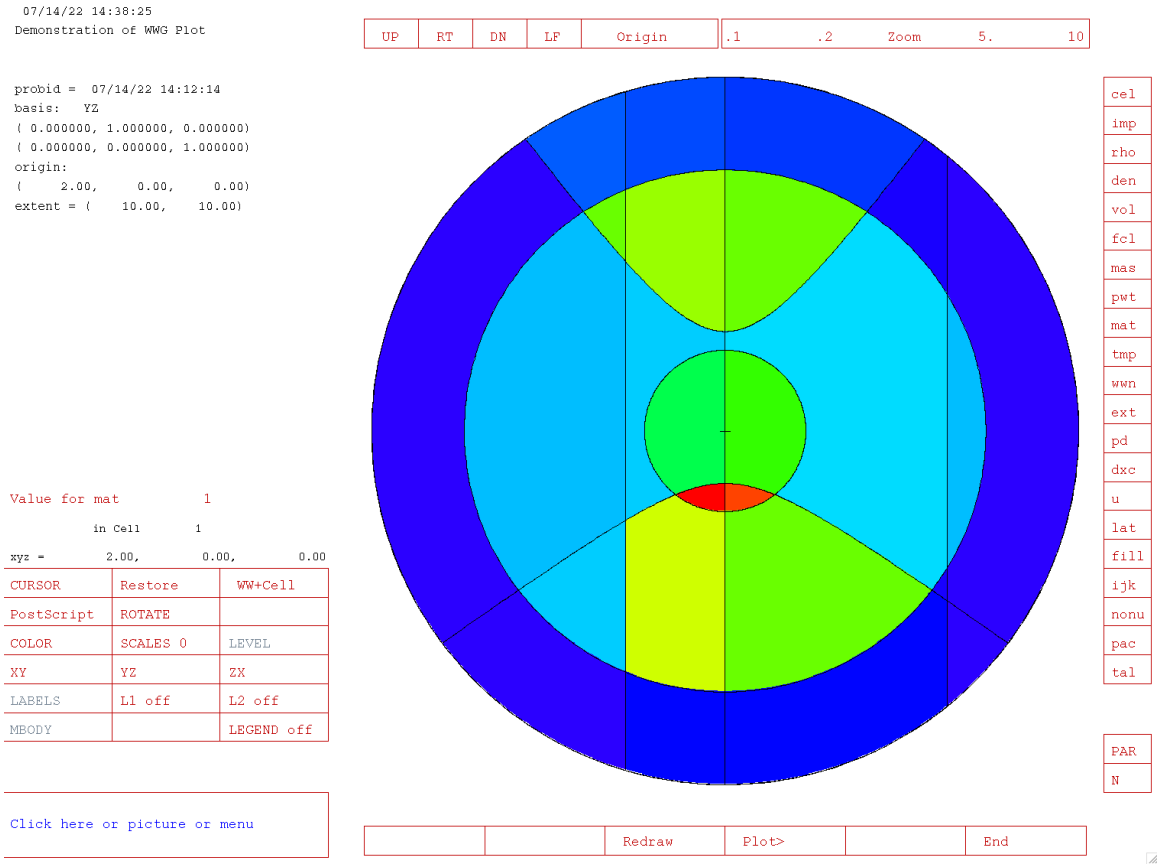


Figure 6.19: Plot view achieved with the command `PX=2`. The polar axis is towards the top of the plot. The azimuthal axis is coming out of the screen. This view shows the smooth conical sections of the polar angles and the other vertical vertical lines not through the center of the mesh are the azimuthal angles intersecting with the slice (left-hand vertical is $\theta = 306°$, right-hand vertical is $\varphi = 72°$.

6.5 Normalization of Energy-dependent Tally Plots

This section discusses two methods of normalizing an energy-dependent tally for plotting:

- dividing by the width of each energy bin, and
- dividing by the logarithmic width of each energy bin (i.e., dividing by the lethargy width).

This section also discusses how to obtain plots that provide an easy visualization for the tally results by the area under the curve. Examples of both normalizations are provided for a logarithmic energy abscissa.

6.5.1 MCNP6 Tally Values and Energy-normalized Tallies

Assume that an MCNP6 energy-dependent tally density such as flux or reaction rate has a form of $f(E)$ per unit energy. An MCNP6 tally result T_i in energy bin i from $f(E)$ is

$$T_i = \int_{E_{l_i}}^{E_{u_i}} f(E)\mathrm{d}E \quad \text{tally units}, \tag{6.3}$$

where the energy bin limits are E_{l_i} to E_{u_i}. The T_is tend to be small for small energy bins and large for larger energy bins. Thus, there is no explicit information about the density $f(E)$ in the T_is unless all energy bins have the same constant width, in which case the correct histogram shape of $f(E)$ is obtained from the T_is. The average value of $f(E)$ over the energy range ΔE_i between E_{l_i} and E_{u_i} is

$$f_i(E) = \frac{\int_{E_{l_i}}^{E_{u_i}} f(E)\mathrm{d}E}{\int_{E_{l_i}}^{E_{u_i}} \mathrm{d}E} = \frac{T_i}{E_{u_i} - E_{l_i}} \quad \frac{\text{tally units}}{\text{unit energy}}. \tag{6.4}$$

The $f_i(E)$s are the bin-wise histogram representations of the tally of $f(E)$ because they are the average values of $f(E)$ in each energy bin. Note that $f_i(E)$ is a constant between E_l and E_u.

This $E_u - E_l$ normalizing of T_i, the default for a 2-D MCNP6 energy-dependent tally plot, is generally agreed to be the proper way to display the $f_i(E)$s when the abscissa E of a 2-D plot is linear. When a `LINLIN` (linear abscissa and linear ordinate) plot of $f_i(E)$s is made with the ordinate starting at zero, the visual area under each histogram represents T_i. This type of visually correct area plot will be termed a Visually Accurate Area (VAA) plot. A VAA plot provides correct visual information about the tally by the area under the histogram.

The average energy $\overline{E}_i$ for each $f_i(E)$ bin is

$$\overline{E}_i = \frac{\int_{E_{l_i}}^{E_{u_i}} E\, f(E)\mathrm{d}E}{\int_{E_{l_i}}^{E_{u_i}} f(E)\mathrm{d}E} \approx \frac{\int_{E_{l_i}}^{E_{u_i}} E\, f_i(E)\mathrm{d}E}{\int_{E_{l_i}}^{E_{u_i}} f_i(E)\mathrm{d}E} = \frac{E_{u_i} + E_{l_i}}{2}, \tag{6.5}$$

where the $f_i(E)$ histogram approximation to $f(E)$ cancels out. $\overline{E}_i$ is used as the average energy for plotting the statistical error bars for tally bin i.

6.5.2 Definition of Neutron Lethargy

The lethargy, U, of a neutron with energy E is defined to be

$$U = \ln\left(\frac{E_0}{E}\right) = \ln(E_0) - \ln(E), \tag{6.6}$$

where E_0 is the upper neutron energy for the problem. On the average, neutrons lose a fixed fraction of their energy in each elastic collision with a specific isotope above thermal energies. The lethargy U is used in nuclear reactor analysis to assess the average logarithmic energy loss of these elastically scattered neutrons.

A neutron with energy E_0 has zero lethargy. As the neutron loses energy, its lethargy increases (hence the name "lethargy," because the neutron becomes more lethargic) and is always positive because no energy is greater than E_0. A neutron with zero energy has infinite lethargy.

For eigenvalue problems, MCNP6 calculates the Energy of the Average neutron Lethargy causing Fission (EALF):

$$\text{EALF} = \exp\left[\frac{\int \ln(E)\Phi(E)\Sigma_{\text{f}}(E)\text{d}E}{\int \Phi(E)\Sigma_{\text{f}}(E)\text{d}E}\right], \tag{6.7}$$

where $\Phi(E)$ is the neutron flux and $\Sigma_{\text{f}}(E)$ is the fission cross section. MCNP6 can plot energy-dependent tallies versus a logarithmic energy scale using lethargy for tally bin normalization.

6.5.3 Lethargy-normalized Tallies for a Logarithmic Energy Abscissa

When the abscissa E is logarithmic, the normalization of the tally scores, T_i, involves the differences in the natural logs (ln) of the energy instead of the differences in the energies. It is useful to relate the differences in the logs of the bin energies to the often used neutron lethargy U:

$$\ln(E_{u_i}) - \ln(E_{l_i}) = U_{l_i} - U_{u_i}, \tag{6.8}$$

where U_{l_i} is the lethargy at E_{l_i} and U_{u_i} is the lethargy at E_{u_i} (the $\ln(E_0)$ terms cancel: see Eq. (6.6)).

The tally T_i can be converted to an average bin i lethargy-normalized value $F_i(U)$ by

$$F_i(U) = \frac{T_i}{\ln\left(\frac{E_{u_i}}{E_{l_i}}\right)} = \frac{T_i}{U_{l_i} - U_{u_i}} \quad \frac{\text{tally units}}{\text{unit lethargy}}. \tag{6.9}$$

The $F_i(U)$s are the histogram approximation to $F(U)$ per unit lethargy. MCNP6 plots the $F_i(U)$s instead of the $f_i(E)$s for a $\ln(E)$ abscissa when the `LETHARGY` plot command is used. The $F_i(U)$s are not plotted when the energy abscissa is linear. Only the $f_i(E)$s and T_is can be plotted for a linear E abscissa. A `LOGLIN` (log abscissa and linear ordinate) plot of the $F_i(U)$s is a VAA plot because the $\ln(E)$ abscissa is linear in U and the area under the histogram is visually correct.

6.5.4 Relation of Tally Lethargy Normalizing to Tally Energy Normalizing

To determine the functional form of $F(U)$ in terms of $f(E)$, equate the U and E density function areas to produce

$$F(U)\text{d}U = -f(E)\text{d}E. \tag{6.10}$$

The negative sign is required because E decreases as U increases. Integrating the left hand side of Eq. (6.10) from U_{u_i} to U_{l_i} is equal to T_i, as is the integral of the right hand side from E_{u_i} to E_{l_i}.

The differential $\mathrm{d}U$ can be written in terms of energy E from Eq. (6.6) as

$$\mathrm{d}U = -\mathrm{d}[\ln(E)] = \frac{-\mathrm{d}E}{E}. \tag{6.11}$$

Substituting Eq. (6.11) for $\mathrm{d}U$ into Eq. (6.10) gives

$$F(U) = E\, f(E). \tag{6.12}$$

Equation (6.12) shows that $F(U)$ can be thought of as the energy E multiplied by $f(E)$. Thus, besides producing VAA `LOGLIN` plots, lethargy-normalized plots have the additional virtue of flattening the $1/E$ neutron flux shape that often occurs in neutron spectra. For an $f(E)$ that has a $1/E$ shape everywhere, $F_i(U)$ is a constant for all i (as opposed to the widely varying $f_i(E)$s), which produces a VAA plot for the $1/E$ shape. Lethargy-normalized plots remove many of the decades of $f_i(E)$ change, represent the $1/E$ portions of the spectrum as a constant, and make understanding and comparing the $F_i(U)$ results easier.

6.5.5 Average Energy for a Lethargy-normalized Tally

The lethargy-averaged energy $\langle E_i \rangle$ for energy bin i is defined as

$$\langle E_i \rangle = \frac{\int_{U_{u_i}}^{U_{l_i}} E\, F(U)\mathrm{d}U}{\int_{U_{u_i}}^{U_{l_i}} F(U)\mathrm{d}U}. \tag{6.13}$$

For the histogram approximation of $F(U)$ by $F_i(U)$, the $F_i(U)$s cancel, and changing variables using Eq. (6.11) gives

$$\langle E_i \rangle \approx \frac{\int_{U_{u_i}}^{U_{l_i}} E\mathrm{d}U}{\int_{U_{u_i}}^{U_{l_i}} \mathrm{d}U} = \frac{\int_{E_{l_i}}^{E_{u_i}} \mathrm{d}E}{\int_{E_{l_i}}^{E_{u_i}} \frac{\mathrm{d}E}{E}} = \frac{E_{u_i} - E_{l_i}}{\ln\left(\frac{E_{u_i}}{E_{l_i}}\right)}. \tag{6.14}$$

In the limit as $E_{u_i} - E_{l_i}$ becomes small, $\langle E_i \rangle \approx \overline{E}_i$ in Eq. (6.5) as expected. This average $\langle E_i \rangle$ is considered to be the centroid energy for a lethargy-normalized bin and is used in MCNP6 to plot statistical error bars, `BAR` plots, and `PLINEAR` plots, as well as printing the plotted points using the `PRINTPTS` command.

6.5.6 MCNP6 LETHARGY Command for Lethargy Normalization

Lethargy-normalized plots of energy-dependent tallies with a log energy abscissa are made with the `LETHARGY` plotting command. This command cannot be used for cross-section plots. For this command to apply, `FREE E` must be active, `LOGLIN` or `LOGLOG` axes must be used, and the `NONORM` command must not be invoked. The `LETHARGY` command cannot be used after a `COPLOT` command and can be disabled to return to energy-bin-width tally normalization for a log energy abscissa by using `RESET LETHARGY`. Switching from a logarithmic energy to a linear energy abscissa with `LETHARGY` in use will automatically change a plot of the $F_i(U)$s to the $f_i(E)$s. Switching back to the log energy abscissa will again plot the $F_i(U)$s.

If an E_0 were specified for a log-abscissa plot, a linear lethargy abscissa could be specified starting at the right with a value of zero at E_0 and linearly increasing to the left in steps of about 2.3 per energy decade decrease. MCNP6 does not label the abscissa as lethargy because of the difficult energy interpretation. The logarithmic energy decades are plotted instead to allow easier interpretation of the areas under the lethargy-normalized histogram. For this reason, there is neither a need nor a provision to specify E_0.

6.5.7 Requirements for Producing a Visually Accurate Area (VAA) Tally Plot

Consider the characteristics of a function of one variable, such as an MCNP6 2-D tally histogram plot. One important quantity for this histogram is the integral over the tally range, which is the total MCNP6 tally. Another important characteristic is the shape of this histogram that provides information about where the largest regions of the tally have occurred. The area of a tally range under the plotted curve is a measure of the contribution of each range to the total.

The area under this curve is best visualized with both the abscissa and the ordinate having a linear scale. The ordinate usually has a lower value of zero to represent correctly the curve area. A `LINLIN` plot of the $f_i(E)$s fits these criteria and therefore is a VAA plot. Often linear scales do not allow complete display of a tally, so logarithmic scales must be used. A logarithmic axis scale typically changes by decades. Each decade change on the abscissa changes a ΔE for a specified length along the abscissa by a decade. The area under the `LOGLIN` $f_i(E)$ curve is proportional to ΔE, which is not reflected in the visual area representation on the log abscissa plot. A logarithmic ordinate does not visually display the correct tally contribution under the curve because this area in the plot is proportional to the number of ordinate decades. When both axes are logarithmic, the visual interpretation in the plot of the area under the curve is further obscured.

The lethargy variable U is linear in the logarithm of the energy as defined in Eq. (6.6). U values for decreasing energy E values of E_0, $E_0/10$, and $E_0/100$ are 0, 2.3, and 4.6. Therefore, a `LOGLIN` plot of the $F_i(U)$s instead of the $f_i(E)$s satisfies the VAA plot linear scale criterion for visually examining the area under a curve. The area under each histogram $F_i(U)$ is $F_i(U) \cdot (U_{l_i} - U_{u_i})$, which is exactly the bin i tally T_i as defined in Eq. (6.9). Similarly, the area under all the $F_i(U)$s is the sum of the T_is, which is the total MCNP6 tally. A `LOGLIN` plot of the $F_i(U)$s is a VAA plot; a `LOGLIN` plot of the $f_i(E)$s is not a VAA plot.

6.5.7.1 Tally Fluctuation Chart History Score Plotting

Two-dimensional plots of a tally $F(x) = x\,f(x)$ are made by dividing the tally bin value by the width of the tally bin $x_{i+1} - x_i$. VAA plots of $F(x)$ are plots whose visual area under the curve is an accurate representation of the tally in each of the tally bins; i.e., the visual area represents $F(x)(x_{i+1} - x_i)$ for all abscissa values. A VAA plot will be produced for a "linlin0" (linear abscissa scale, linear ordinate scale starting at 0) $F(x)$ plot; i.e., the area of $F(x)$ from x_i to x_{i+1} correctly represents the bin tally value visually for all x when both the abscissa and ordinate scales are linear and the smallest ordinate value is zero. A linlin0 plot can be achieved with the following command:

```
LINLIN XLIMS 0 [max]
```

In a similar manner to the linlin0 VAA plot described above, a VAA plot of $F(x)$ on a "loglin0" (log abscissa scale, linear ordinate scale starting at 0) plot is produced if the tally bin value is divided by the difference in the logarithms of the abscissa values. The loglin0 plot can be achieved with:

```
LOGLIN XLIMS 0 [max]
```

If $y = \ln(x)$, then $G(y)$ is defined to be

$$G(y) \equiv \frac{\text{tally bin value}}{\ln(x_{i+1}) - \ln(x_i)} = \frac{\text{tally bin value}}{y_{i+1} - y_i}. \tag{6.15}$$

A loglin0 plot of $G(y)$ is a VAA plot of $F(x)$ because y is linear on a log abscissa and $G(y)(y_{i+1} - y_i)$ is the area of the tally bin.

The relation between $G(y)$ and $F(x)$ is

$$G(y)|\mathrm{d}y| = F(x)|\mathrm{d}x| \tag{6.16}$$

where

$$\mathrm{d}y = \mathrm{d}\ln(x) = \frac{\mathrm{d}x}{x}. \tag{6.17}$$

Therefore,

$$G(y) = x\,F(x). \tag{6.18}$$

This y normalizing of the tally bin value to make a loglin $G(y)$ plot is equivalent to making a loglin $x\,F(x)$ plot. Lethargy plotting of an energy-dependent tally $F(E)$, with E representing energy, is the equivalent of plotting $G(\ln(E)) = E\,F(E)$ on a loglin0 scale to produce a VAA plot for the tally $F(E)$. For more information on normalization of energy dependent tallies, see Section 6.5.

These normalizing statements can be generalized to any function $h(x)$. The VAA interpretation of a 2-D plot therefore depends on the abscissa axis scale. A linlin0 plot of $h(x)$ is a VAA $h(x)$ plot and a loglin0 $h(x)$ plot is a VAA plot for $h(x)/x$.

Two-dimensional plots from a `RUNTPE` (but not a `MCTAL`) file of the empirical history score probability density function $f(x)$ moments can be made using the Tally Fluctuation Chart (TFC) tally plot commands [§6.3.3.7]. In this case, the tally $F(x) = x\,f(x)$. From the discussion above, a loglin0 $F(x)$ plot can be interpreted as a VAA plot for $f(x)$; i.e., the area under the curve on a loglin0 scale represents where the $f(x)$ sampling has occurred. A linlin0 $F(x)$ plot is a VAA plot for the tally $F(x)$.

Based on these observations, the following statements can be made about TFC commands to create $f(x)$ moment plots for the TFC bin of a tally (without the `NONORM` option):

- $f(x)$ TFC bin plots are VAA plots when
  1. `TFC P` [$f(x)$] is on a linlin0 scale; and
  2. `TFC 1` [$x\,f(x)$] is on a loglin0 scale.
- $x\,f(x) = F(x)$ TFC bin tally plots are VAA plots when
  1. `TFC 1` [$x\,f(x) = F(x)$] is on a linlin0 scale; and
  2. `TFC 2` [$x^2\,f(x) = x\,F(x)$] is on a loglin0 scale.
- $x^n\,f(x)$ TFC bin tally moment plots are VAA plots when
  1. `TFC n` [$x^n\,f(x)$] is on a linlin0 scale; and
  2. `TFC n` [$x^{n+1}\,f(x)$] is on a loglin0 scale.

The VAA contributions to the nth $f(x)$ moment can be viewed with an $x^n\,f(x)$ linlin0 plot or an $x^{n+1}\,f(x)$ loglin0 plot. The empirical $f(x)$ slope result can be checked by viewing a `TFC n` plot. If the high-score $f(x)$ tail for a long-tailed distribution (not a finite distribution) is proportional to $1/x^n$, then the `TFC n`plot will be a statistical constant at the high x scores. A large score $f(x)$ slope of at least n exists if the high-score `TFC n` [$x^n\,f(x)$] values are decreasing. VAA $f(x)$ moment plots can be a useful tool in studying the detailed impact of variance reduction techniques on $f(x)$ (not the history sampling time as function of x) and the efficiency of a calculation.

6.5.8 Comparisons of Energy and Lethargy Tally Normalizations for a Log in Energy Abscissa

Energy-normalized and lethargy-normalized log energy abscissa tally plots for two analytic and two critical uranium assembly problems are discussed to show which are VAA plots. The two analytic $f(E)$ examples will be accurate to three significant figures in the text.

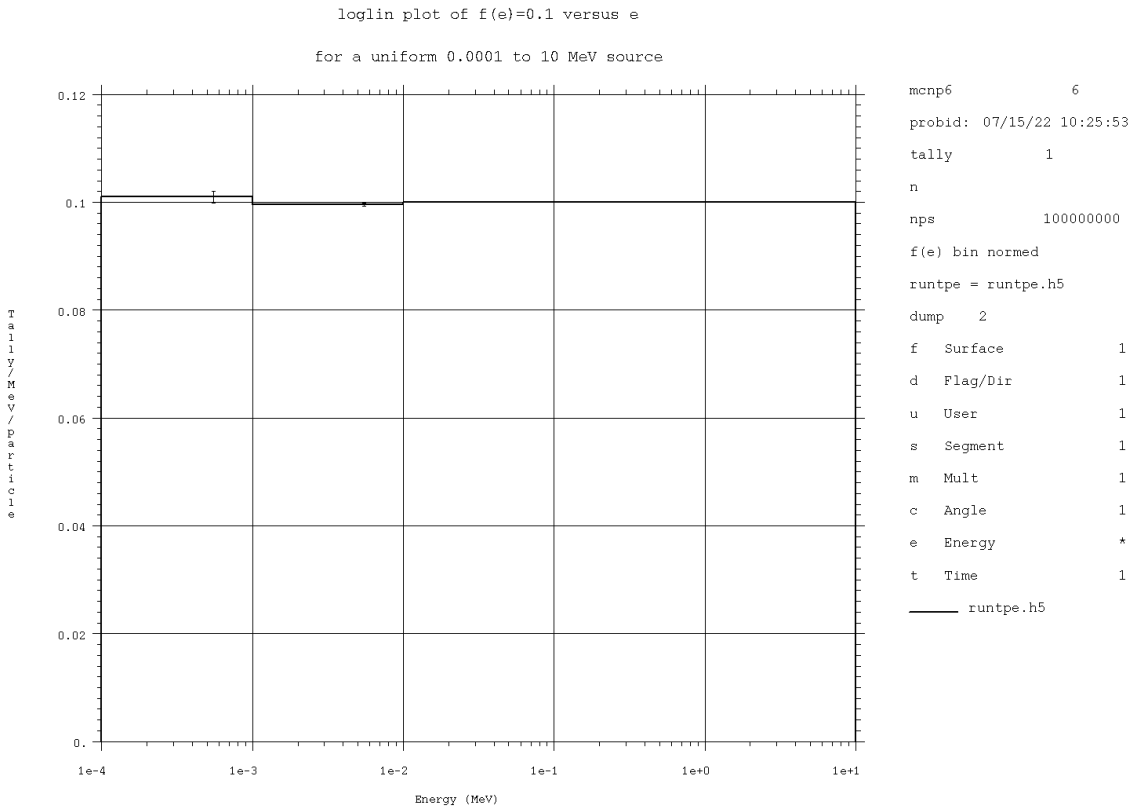


Figure 6.20: A LOGLIN plot of energy-normed $f_i(E)$ versus E for a uniformly sampled energy source between 0.0001 and 10 MeV. The expected value of all $f_i(E)$s is 0.1.

6.5.8.1 Example 1: A Constant $f(E) = 0.100$ from 0.0001–10 MeV

The first example is the tally of a uniform energy source between 0.0001 and 10 MeV. The expected value of all $f_i(E)$s is 0.100. Figure 6.20 shows a plot of the five energy-normalized $f_i(E)$s, each with an energy bin width of a decade. The tally T_i in the energy bin from 1 to 10 MeV is $0.1(10-1) = 0.9$. The next lowest energy bin tally is $0.1(1-0.1) = 0.09$. The tally bin T_is are decreasing by a decade per decade decrease in energy, but the $f_i(E)$s are a statistically constant 0.1. Visually interpreting this LOGLIN plot of the $f_i(E)$s by the area under the curve may not be useful because the energies are changing by decades along the logarithmic energy abscissa.

Figure 6.21 shows a LOGLIN lethargy-normalized plot of the corresponding five $F_i(U)$s. The "$f(u) = e\,f(e)$ bin normed" text on the right hand side of the plot is a reminder that this is a lethargy-normalized plot. The shapes of the $f_i(E)$s in Fig. 6.20 and the $F_i(U)$s in Fig. 6.21 are completely different. $F_i(U)$ for the 1 to 10 MeV tally bin is $0.9/\ln(10/1) = 0.391$. The area in the plot of this bin is $0.391\ln(10/1) = 0.900$, which is the correct T_i for this bin. The tally bin area from 0.1 to 1 MeV is $0.391\ln(1/0.1) = 0.0900$. The visual areas of each of the tally bins represent the tally for that bin because of the linear lethargy abscissa obtained by the lethargy normalization.

The LOGLIN lethargy-normalized plot in Fig. 6.21 clearly displays the relative contribution of each of the five tally bins by the area under the histogram.

Figure 6.22 shows the same plot as in Fig. 6.21, except the ordinate is now logarithmic. The five $F_i(U)$s are the same in both plots, but the visual area interpretation in Fig. 6.22 is misleading because the ordinate scale is logarithmic. Nevertheless, Fig. 6.22 is useful for assessing the behavior of the $F_i(U)$s that are small and cannot be seen in Fig. 6.21 with the linear ordinate.

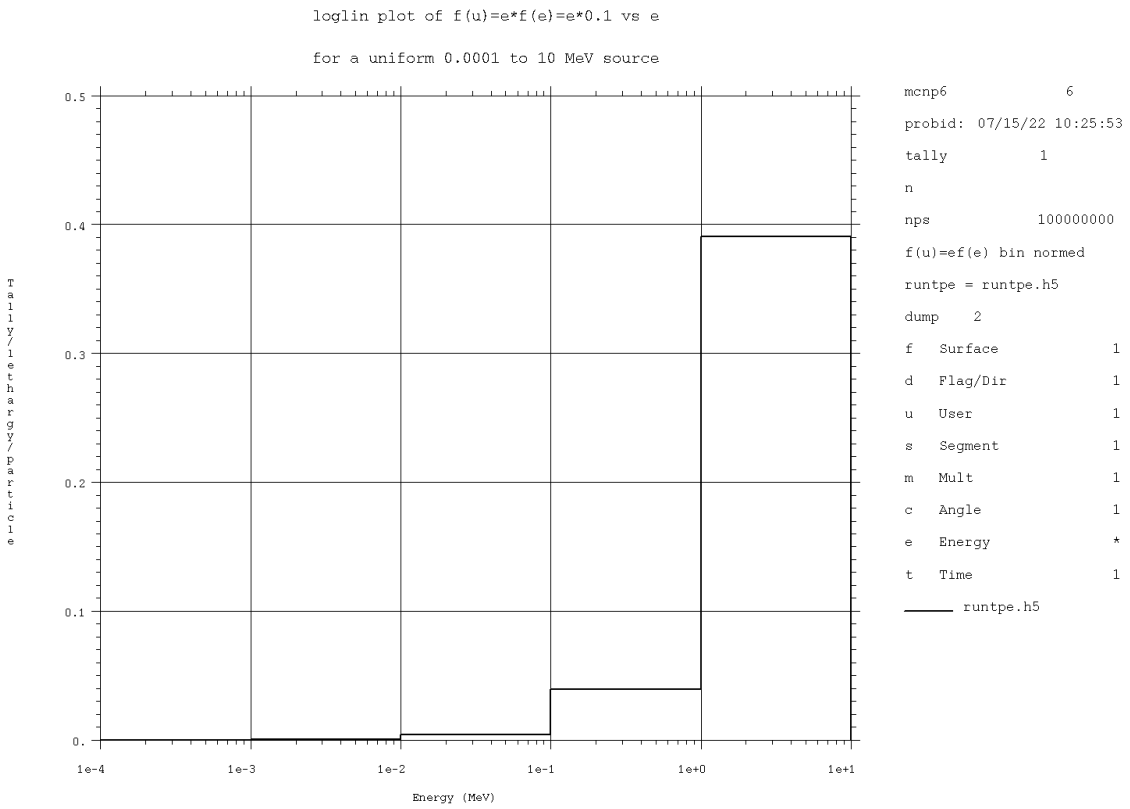


Figure 6.21: A **LOGLIN** plot of lethargy-normed energy-normed $F_i(E)$ versus E for a uniformly sampled energy source between 0.0001 and 10 MeV. The area $F_i(E \cdot \Delta U_i)$ of each tally bin is the tally value.

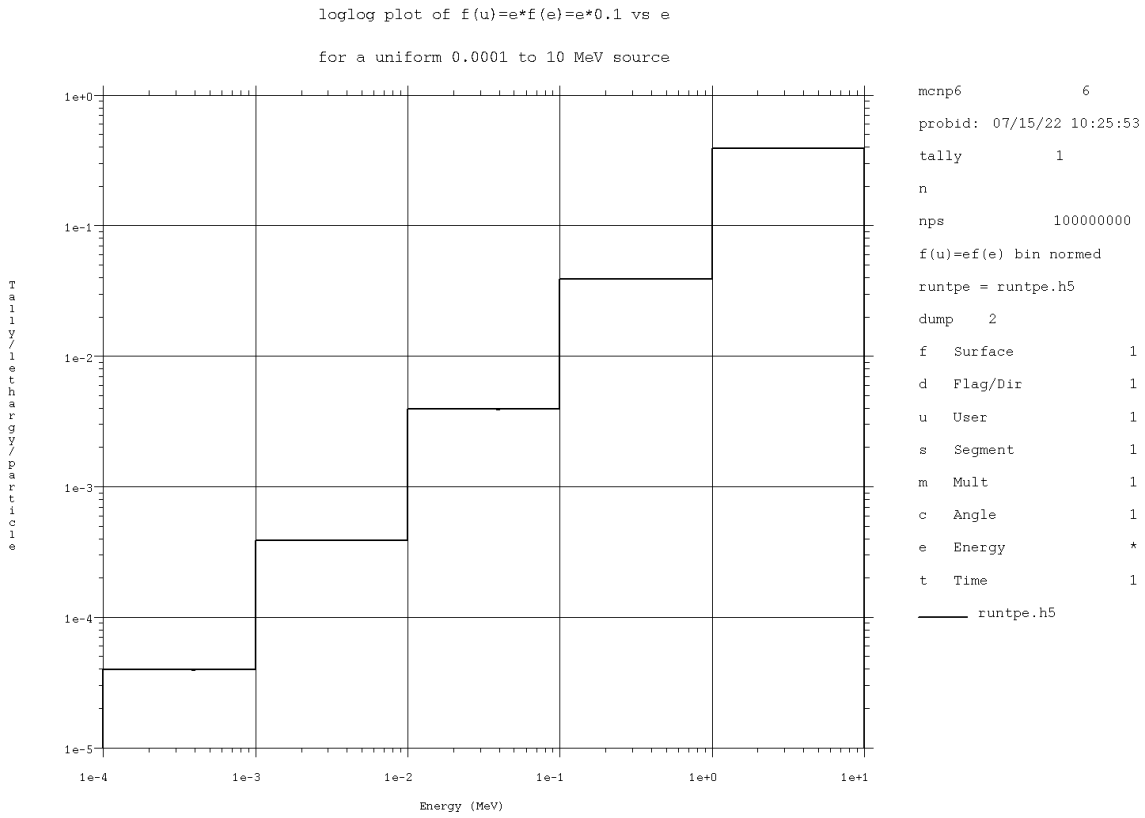


Figure 6.22: A LOGLOG plot of $F_i(U)$ versus E for a uniformly sampled energy source between 10^{-4} and 10 MeV. The smaller tallies not visible at lower energies in Fig. 6.21 can be seen here.

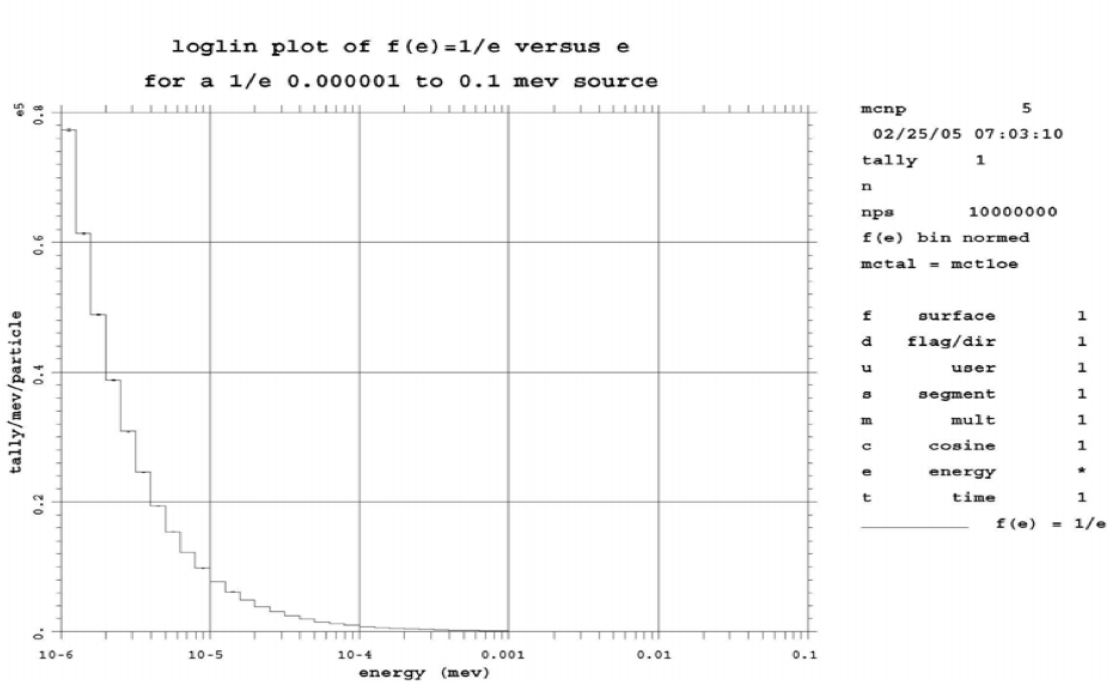


Figure 6.23: A LOGLIN plot of $f_i(E)$ versus $1/E$ energy source between 10^{-6} and 0.1 MeV. Equal lethargy bin spacing (0.23) in energy is used, so all bins contribute the same amount to the tally for the $1/E$ source.

6.5.8.2 Example 2: $f(E) = 0.087/E$ from 10^{-6}–0.1 MeV

For a second example, an equal-lethargy 50-bin tally was made of a $1/E$ energy source from 10^{-6} to 0.1 MeV. The lethargy width of each tally bin is $\ln\left(0.1/10^{-6}\right)/50 = 0.23$. All T_is have an expected value of $0.23/\ln\left(0.1/10^{-6}\right) = 0.02$ for the equal lethargy energy bins. Figures 6.23 and 6.24 show LOGLIN and LOGLOG plots of the $f_i(E)$s. Each tally bin has a relative error of 0.2%. The $f_i(E)$s have the expected $1/E$ shape of the source. The histograms in both figures decrease with increasing energy because the T_is are constant and ΔE_is are continuously increasing. Neither Fig. 6.23 nor 6.24 is a VAA plot because neither shows a meaningful visual under-the-curve area representation of the T_is for this tally.

The lethargy-normalized plot of the $F_i(U)$s in Fig. 6.25 is a VAA plot. The $F_i(U)$s are a statistical constant $(0.02/0.23 = 0.087)$ for $f(E) = 1/E$, as predicted by Eq. 6.12. Figure 6.25 shows visually that all equal lethargy widths contribute equally to the total tally, which is correct for the $1/E$ source. The integral under the curve of Fig. 6.25 is $0.087\ln\left(0.1/10^{-6}\right) = 1$, which is the source strength. Once again, the shapes of the $f_i(E)$s in Figs. 6.23 and 6.24 and the $F_i(U)$s in Fig. 6.25 are completely different.

6.5.8.3 Example 3: Neutron Fluxes and Fission Rate Spectra for Two Critical Uranium Systems

A third and more realistic example is a comparison of $f_i(E)$ and $F_i(U)$ plots for the neutron fluxes and fission rate spectra calculated by MCNP6 for two critical uranium systems:

1. a water-reflected, water-moderated array of 18×20 2.35% low-enriched uranium (LEU) UO_2 aluminum clad fuel elements [334]; and

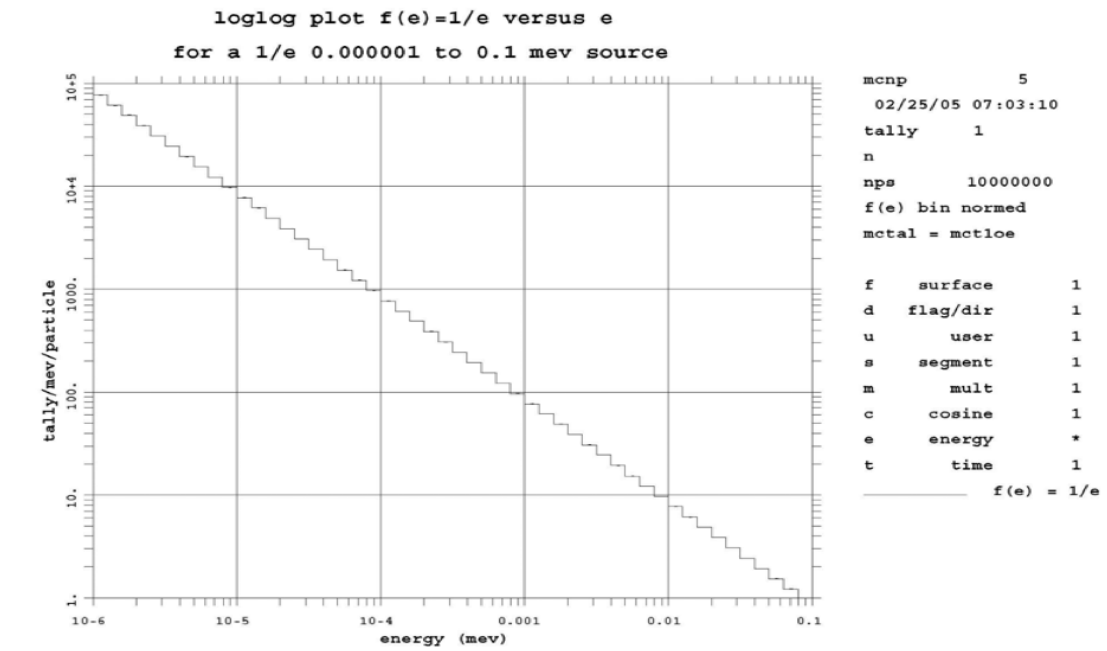


Figure 6.24: A `LOGLOG` plot of $f_i(E)$ versus $1/E$ energy source between 10^{-6} and 0.1 MeV. The $1/E$ behavior of $f_i(E)$ is evident.

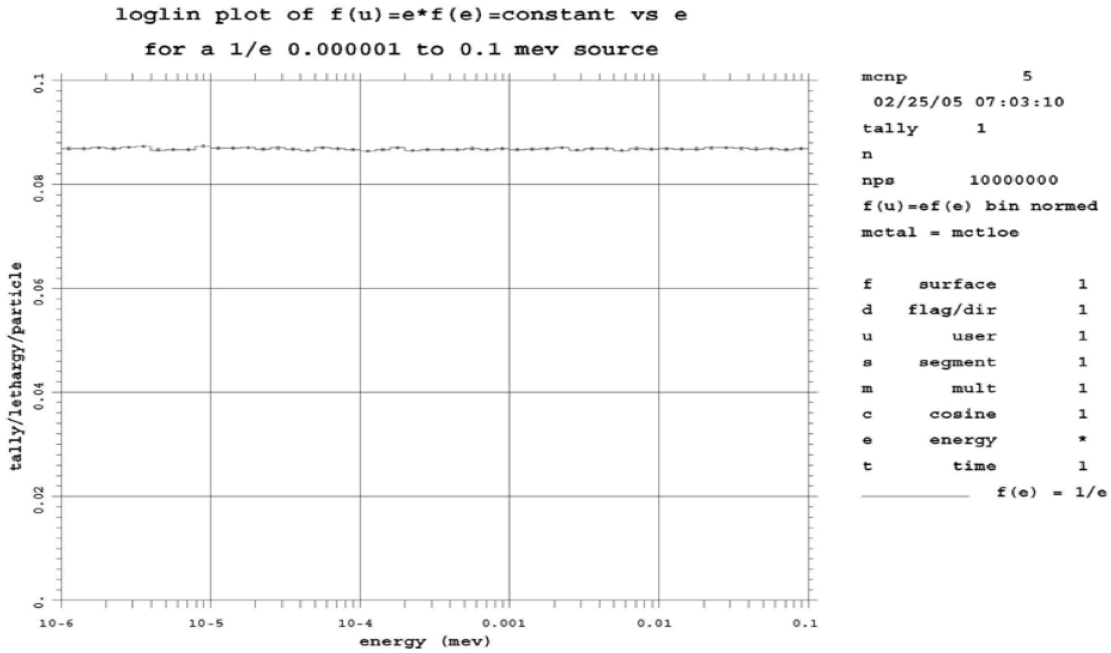


Figure 6.25: A `LOGLIN` plot of $F_i(U) = E\,f(E) = E(0.087/E) = 0.087$ versus $1/E$ energy source between 10^{-6} and 0.1 MeV. The integral of this plot is unity, which is the source strength.

Table 6.1: Percentage of Fission Rates by Incident Neutron Energy

System	Spectrum	$E < 0.0625$ eV	0.0625 eV $< E <$ 100 keV	100 keV $< E$
LEU	Thermal	91.4	4.5	4.1
HEU	Fast	0.0	5.4	94.6

2. the Godiva bare metal 93.7% highly enriched uranium (HEU) sphere [335].

The calculations were performed with pre-ENDF/B-VII uranium isotope cross sections (from Los Alamos National Laboratory Group T-16) that are identified by a ".69c." All other isotopes in the LEU system used ENDF/B-VI ".66c" cross sections with ".60t" $S(\alpha, \beta)$ data for light water and polyethylene. The calculated $k_{\text{eff.}}$ for the LEU system is 0.9968 with an estimated standard deviation of 0.0003. The HEU system $k_{\text{eff.}}$ is 0.9987 with a standard deviation of 0.0003. The calculated EALF for the LEU and HEU systems is 1.0×10^{-7} MeV and 0.82 MeV. The calculated percentages of the incident neutrons causing fission by energy range are listed in Table 6.1.

Figures 6.26 and 6.27 compare the energy-normalized and lethargy-normalized plots of the neutron flux $f_i(E)$s and $F_i(U)$s for the thermal LEU and fast HEU systems.

The areas under all curves are one. Only the HEU flux values with relative errors less than 0.1 were plotted, which is the reason this flux curve terminates abruptly. The plots of the $f_i(E)$s in Fig. 6.26 do not convey the contributions of the $f_i(E)$ flux by the area under the curve because both scales are logarithmic. The $1/E$ flux behavior for the LEU system is evident in the figure over much of the ten decades of the $f_i(E)$s. Fig. 6.26 is not a VAA plot.

Figure 6.27 is a VAA plot because the visual area under each curve accurately represents the contributions to the total flux by energy range because both axes are linear. The $1/E$ $f_i(E)$ flux behavior is characterized as the flat $F_i(U)$ range, as predicted by Eq. (6.12).

Figure 6.28 shows a `LOGLOG` plot of the fission rate $f_i(E)$s versus E for the thermal neutron spectrum LEU and fast high-energy spectrum HEU systems. Each curve is divided by the total tally over all energies so the area under each curve is unity. Figure 6.28 shows the thermal and fast fission rate shapes, but does little to convey the fission rate percentages shown in Table 6.1. Figure 6.28 is not a VAA plot.

Figure 6.29 shows a `LOGLIN` plot of the $f_i(E)$s versus E for just the LEU system. The area under this curve representation of the LEU system also does not visually agree with the results in Table 6.1: there is no curve area above 6×10^{-7} MeV (0.6 eV). This conclusion about incorrect visual areas is not surprising since the $F(U)$ and $f(E)$ shapes differ so markedly for the first two simple examples. Figure 6.29 is not a VAA plot.

Figure 6.30 shows a `LOGLIN` VAA plot of the fission rate $F_i(U)$s versus E for both systems. The area beneath both curves is one. Now the fission rate percentages occurring in each energy range become clear and visually match the results in Table 6.1. The `LOGLIN` lethargy-normalized plot in Fig. 6.30 visually conveys much more information about the fission rate characteristics as a function of energy than the plot of the $f_i(E)$s in Figs. 6.28 and 6.29.

Comparing the LEU $f_i(E)$s in Fig. 6.29 with the LEU $F_i(U)$s in Fig. 6.30 shows that the $f_i(E)$ thermal fission rate peak in Fig. 6.29 is skewed toward the lower energies. This shift is caused by the ever-increasing $1/\Delta E_i$ for decreasing energies. The visual area representation of the LEU tally is correct for $F_i(U)$ in Fig. 6.30 and incorrect for $f_i(E)$ in Fig. 6.29.

Figure 6.31 shows a `LOGLOG` plot of the fission rate $F_i(U)$s versus E. Even though the visual area under this curve is misrepresented by the log ordinate, the behavior of the smaller $F_i(U)$ values versus E becomes clearer.

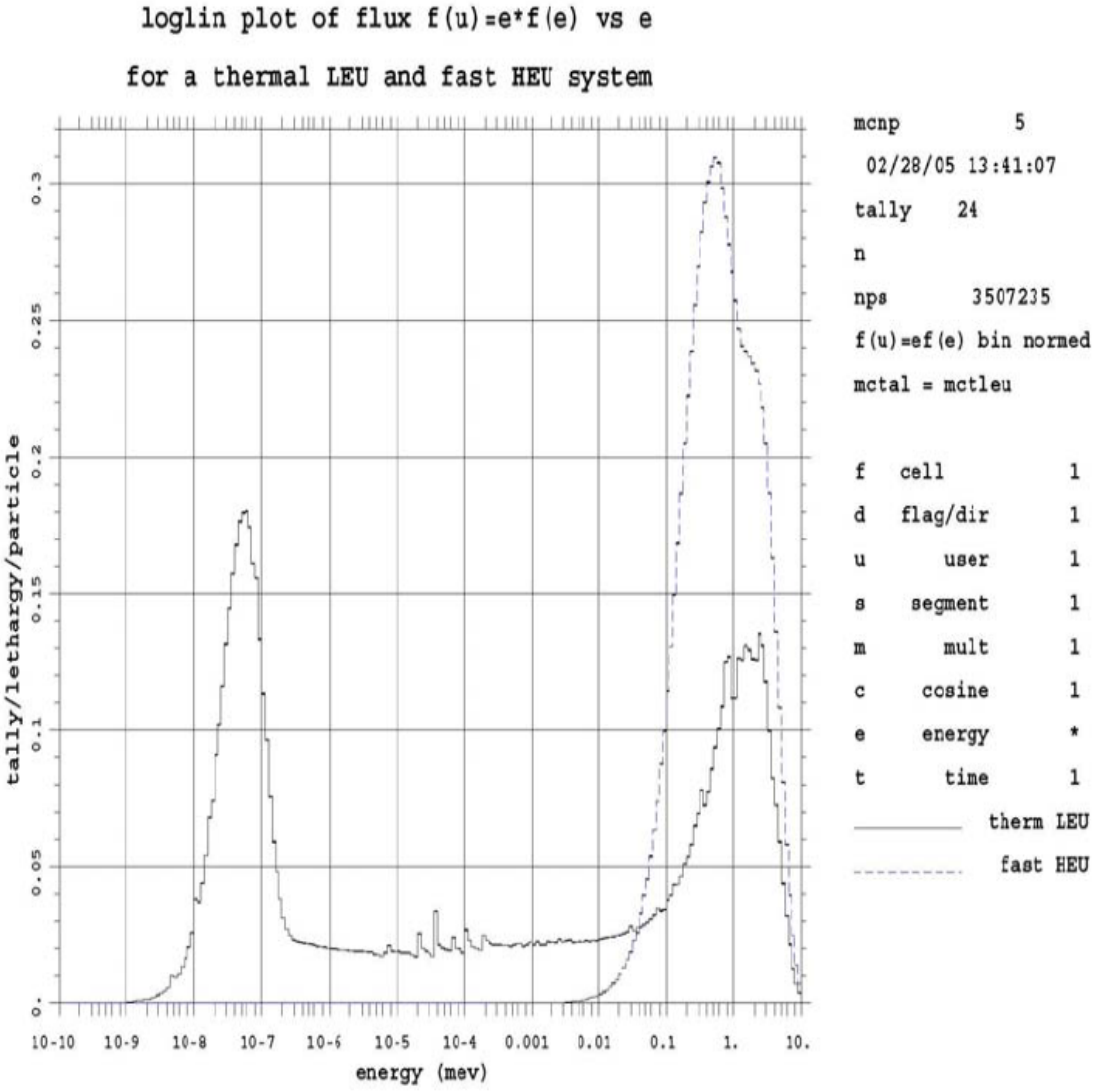


Figure 6.26: A LOGLOG plot of energy-normed neutron flux $f_i(E)$ versus E for the thermal LEU (larger curve) and fast HEU (smaller curve) systems. The area under curves is one.

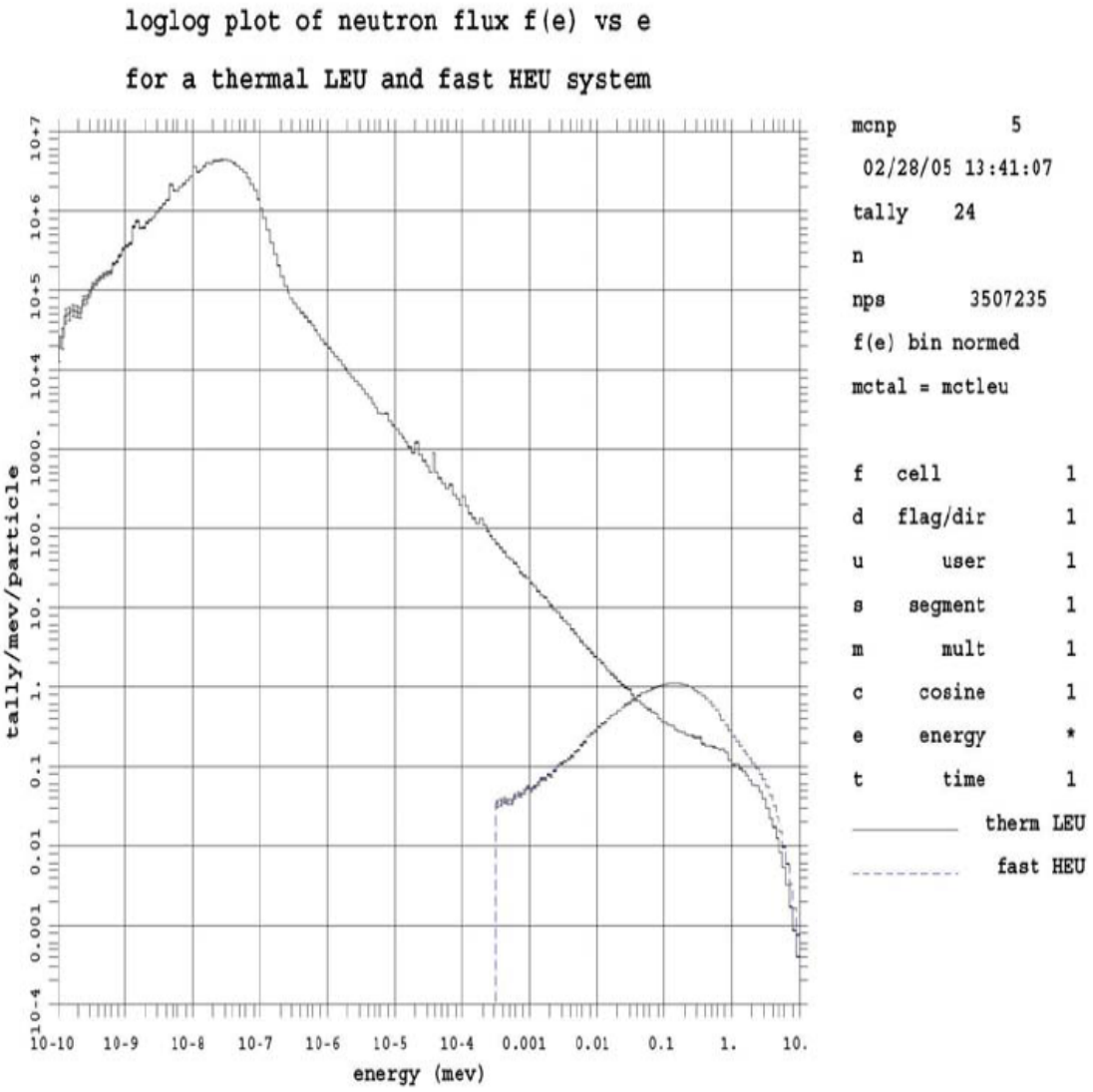


Figure 6.27: A `LOGLIN` plot of the lethargy-normed flux $F_i(U)$ versus E for the thermal LEU (smaller curve) and fast HEU (larger curve) systems. The area under curves is one.

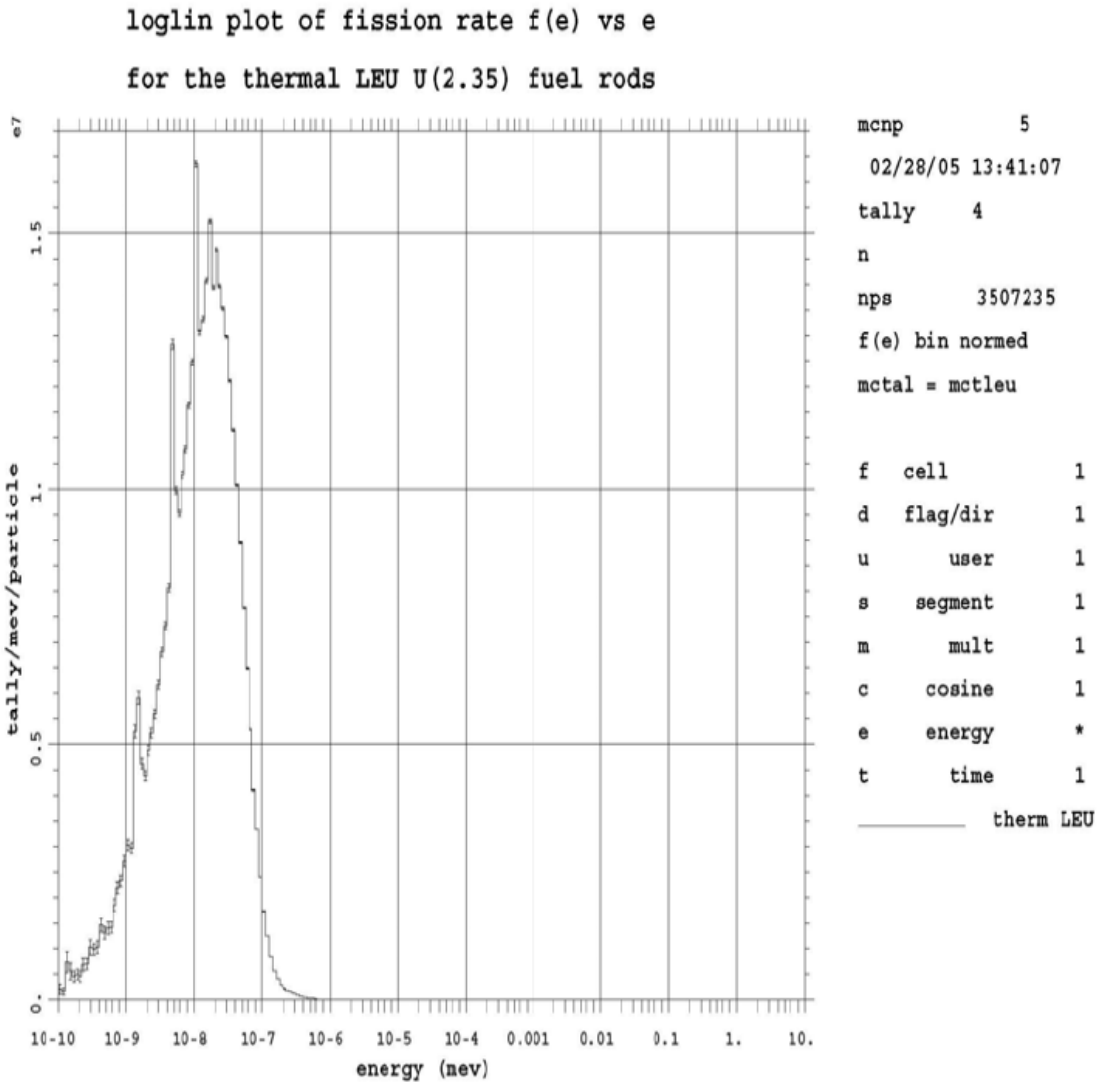


Figure 6.28: A LOGLOG plot of the fission rate $f_i(E)$ versus E for the thermal LEU (larger curve) and fast HEU (smaller curve) systems. The area under curves is one.

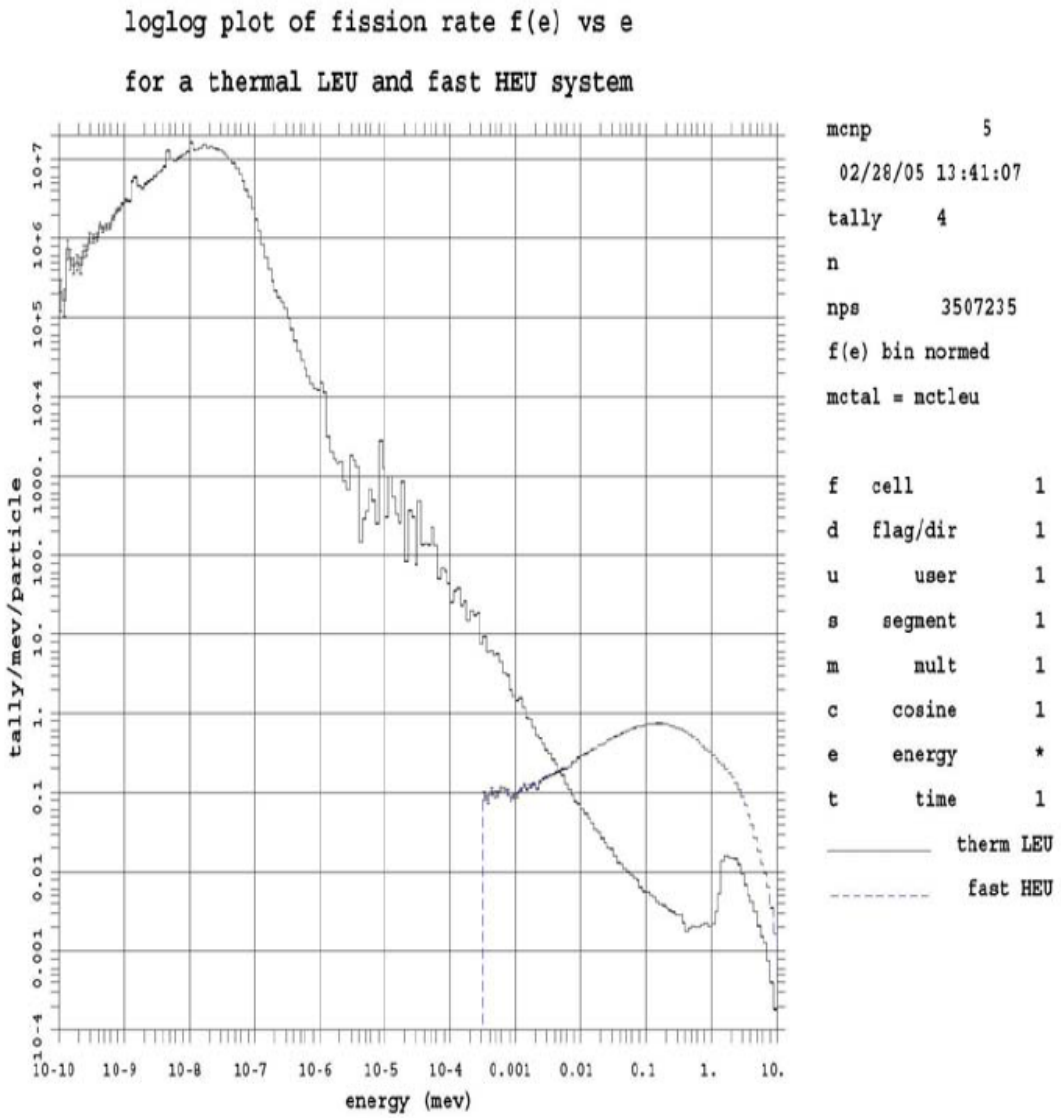


Figure 6.29: A `LOGLIN` plot of the fission rate $f_i(E)$ versus E for the thermal LEU. The area under curve is one.

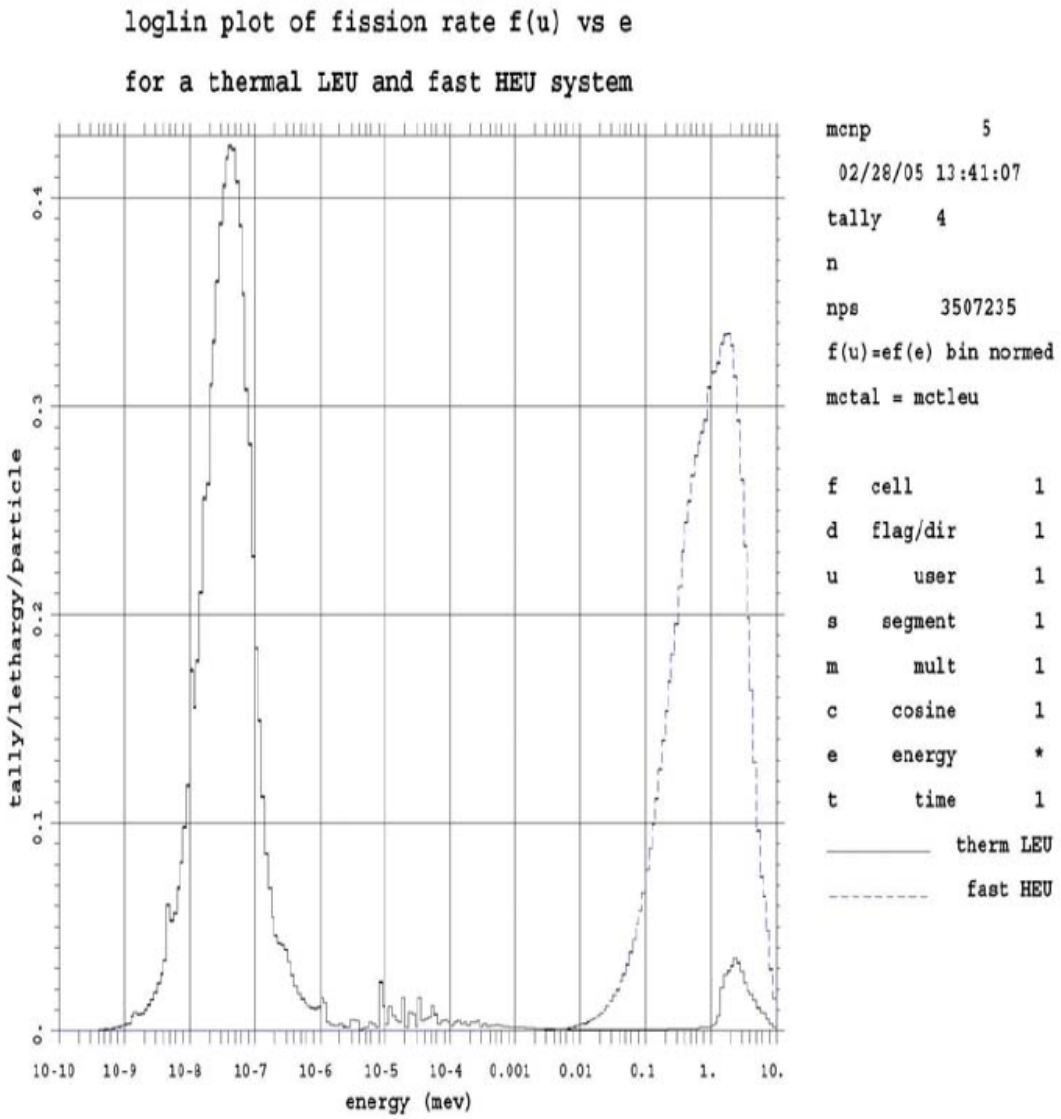


Figure 6.30: A `LOGLIN` plot of the fission rate $f_i(E)$ versus E for the thermal LEU (left curve) and fast HEU (right curve) systems. The area under curves is one.

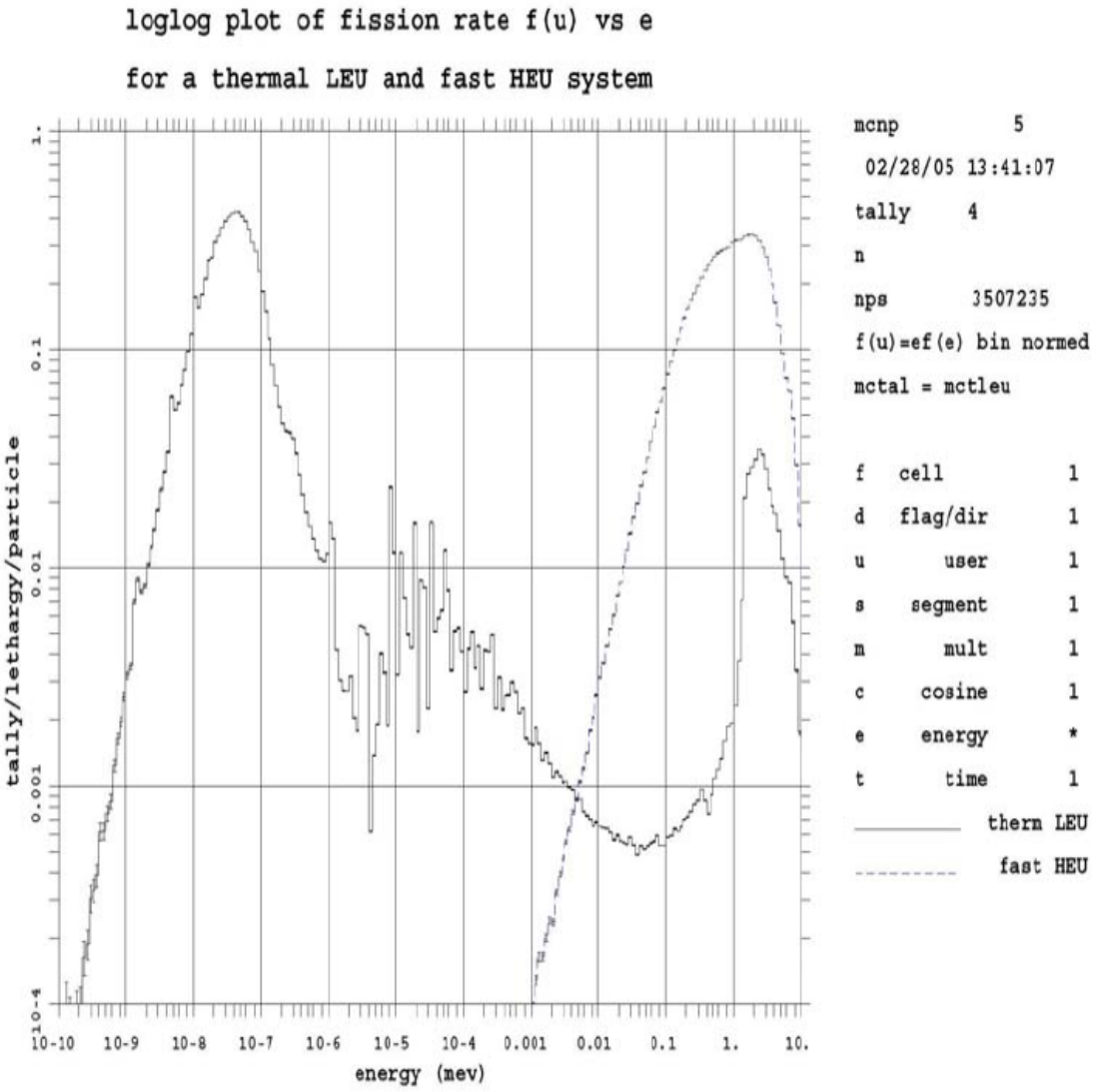


Figure 6.31: A LOGLOG plot of the fission rate $F_i(E)$ versus E for the thermal LEU (left curve) and fast HEU (right curve) systems. The area under curves is one.

6.5.9 Summary of Energy-normalized and Lethargy-normalized MCNP6 Tally Plots

Visually Accurate Area (VAA) plots allow an accurate visual assessment of the contributions made to a tally by various ranges. For a `LINLIN` plot, the energy-normalized $f_i(E)$s are VAA plots. For a `LOGLIN` plot, the `LETHARGY` command produces lethargy-normalized $F_i(U)$s that are VAA plots. All other plots, which may provide useful information about the tally, are not VAA plots. The energy location in a tally bin of the statistical error bars for energy-normalized and lethargy-normalized plots is different, as shown by Eqs. (6.5) and (6.14).

VAA plots are useful tools that allow visual assessment of the characteristics of the tally by examining the area under the curve. Equal abscissa bin spacing is not required for VAA plots. The more uniformly subdivided the abscissa intervals are, however, the easier the area visualization becomes; e.g., it may be hard to estimate the area for a narrow bin that is much higher than other bins. If the abscissa intervals are all the same length, then the shape of a plot is identical to a `NONORM` plot where the bin T_is themselves are plotted. The magnitude of the two curves will differ by the bin-width normalization. MCNP6 can create lethargy-normalized plots for $\ln(E)$ abscissas for all particle types when the `LETHARGY` plotting command is used.

The bottom-line: both the `LINLIN` energy-normalized and `LOGLIN` lethargy-normalized plots of energy-dependent tallies allow a direct tally contribution visualization by the area under the histogram.

Chapter 7

Technology Preview: Qt Based MCNP Geometry and Tally Plotting

With this version of the MCNP6 code we are bundling a preview of our next-generation Qt Framework based plotter that enables visualization of 2-D slices of the geometry. Depending on how the preview plotter (henceforth referred to as the plotter) is invoked, these slices can be overlaid with spatial tally results or display graphs of the standard tallies. It is expected that in a future release of the MCNP6 code this plotter will replace the current plotter described in the previous chapter [§6]. With this new Qt-based plotter the MCNP6 code will run as a native app any time the plotter is invoked - using X-windows on linux and the native OS infrastructure on Windows and macOS. As a result, the MCNP code can run and display graphics on the local machine with no special software install required.

It is expected that the plotter will primarily be used to verify the geometry before starting transport and to peruse the results after transport is done since this does not require using any additional tools. For deeper data exploration and advanced visualization our recommended path is to use the HDF5/XDMF capability described in §D.4.3 in conjunction with a dedicated visualization package such as ParaView [326].

In addition to the capabilities described in this chapter, the plotter can also be invoked in a mode that displays the cross sections of nuclear data loaded for the run. The invocation and the commands allowed are identical to those described elsewhere [§6.3.3.8], so the cross-section mode is not described here. Note that:

Details:

1. All keyboard input to the plotter should be followed by the Enter key for the plotter to act on them. For brevity we have omitted this repetitive piece of information in the text. Hence, when the manual directs the user to type a command in the Input Pane, it is explicitly assumed that the user will follow that command by hitting the Enter key to execute it.
2. Plotter keyboard entry is case insensitive, so `PLOT` is the same as `plot` which is the same as `pLoT`. For clarity, we have used uppercase for the text entry keywords in this chapter.
3. All graphical capabilities of the plotter can be controlled via keyboard input in the Input Pane. For large, complex geometries it is recommended to use the plotter in batch mode (`NOTEK` option, §7.1.2) with the `SAVEPDF` command to generate the image.

7.1 Viewing Geometry

The plotter renders 2-D slices of the geometry with the capability to overlay the slice with `FMESH` and `TMESH` tally results loaded from a runtape file [§D.2]. The user also has the option to graph other tally data within the runtape file. Figure 7.1 shows the interface that the plotter presents when invoked in Geometry

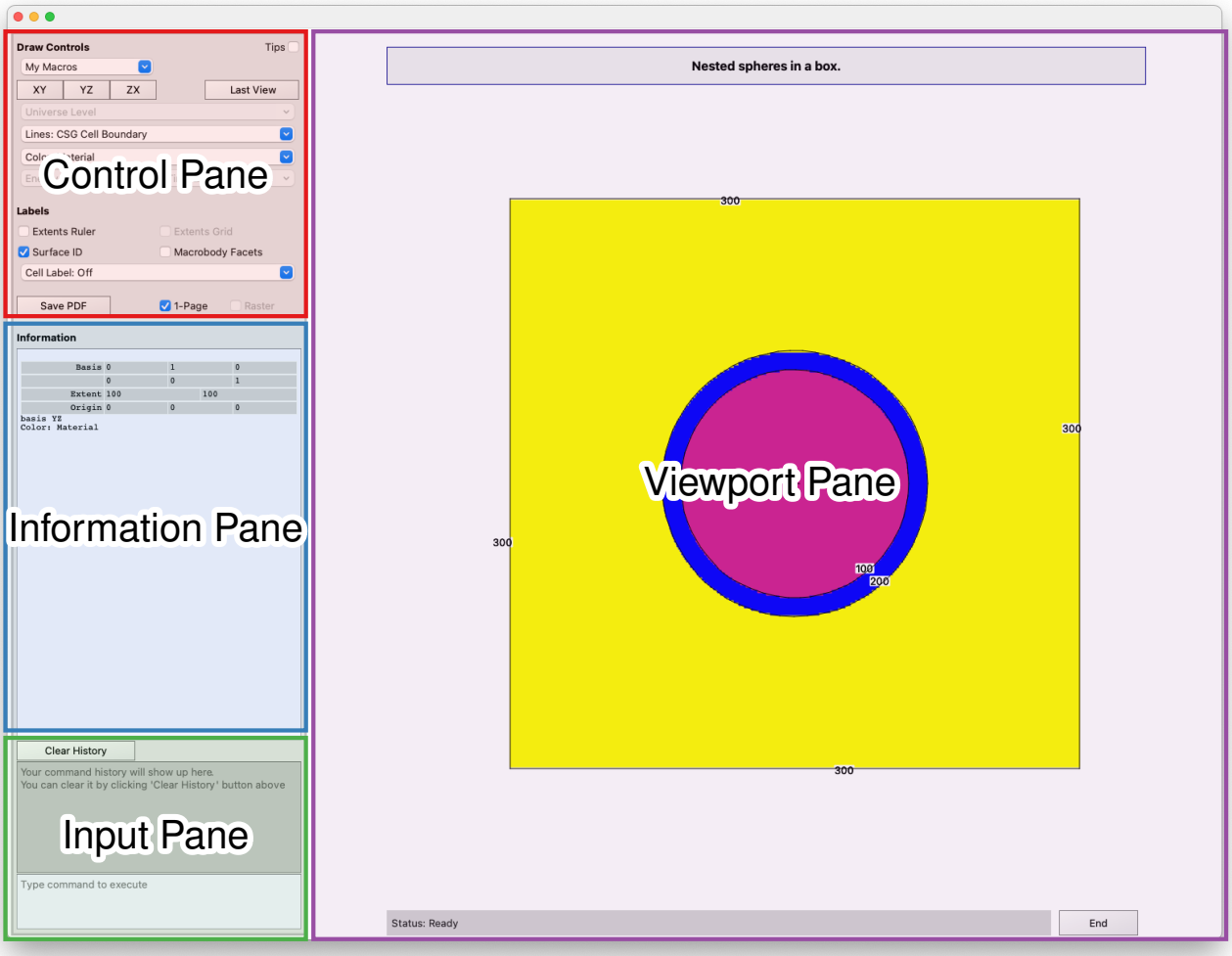


Figure 7.1: Overlay showing the four panes of the Plotter: the Viewport Pane which displays the rendered slice, the Control Pane which provides controls for manipulating the view, the Information Pane which displays the current view information and details of the cell clicked, and the Input pane where command line input can be entered. These panes are described in §7.3.

Viewing mode. The plotter interface has four distinct panes that are hi-lighted. These are the Viewport Pane [§7.3.1], the Control Pane [§7.3.2], the Information Pane [§7.3.3], and the Input Pane [§7.3.4]. The user interacts with the plotter using graphical controls present in these panes. When viewing geometries, the view can be changed using either the mouse in the Viewport Pane, the controls in the Control Pane, or keyboard input in the Input Pane. For example, once a slice has been rendered by the plotter, the Viewport Pane displays the rendered image and enables mouse-based interaction for translation, rotation, and zoom. Left-clicking in the Viewport Pane will display information about the clicked cell in the Information Pane. The Control Pane enables direct access to frequently used operations for both the view and the rendered quantities. The Information Pane displays the current view details and information on the last cell clicked. The Input Pane allows the user to input text commands to control the view rendered.

7.1.1 Geometry Specification

To plot a 2-D slice of geometry, one needs to specify the plane to be plotted, the orientation of the geometry within that plane, and the extents of the plane that are visible in the slice. The plotter enables this through use of three concepts: `BASIS` vectors, `ORIGIN` at the center of the Viewport Pane, and `EXTENT` of space visible

within the Viewport Pane. The first of these keywords, `BASIS`, defines the orientation within the Viewport Pane by specifying the horizontal and vertical directions. The normal direction is computed as the cross product of these two bases. Specifying the basis vectors orients the geometry, but does not identify where the slice is located along the normal direction. This is done by specifying the `ORIGIN`, which identifies the point in space that is at the center of the Viewport Pane irrespective of the `BASIS` selected. The combination of the `BASIS` and `ORIGIN` uniquely determines the plane to be plotted. The `EXTENT` concept limits how much of the plane is visible in the horizontal and vertical directions. Together these three concepts completely specify the slice to be rendered.

The plotter uses these three concepts to render geometry. In the current implementation the plotter does not limit the basis to orthogonal vectors but it does require that they not be collinear. If the basis vectors specified are not perpendicular to each other, then the horizontal direction is set to the first basis and the normal direction is computed using the cross product of the two basis vectors. The vertical direction is then determined from the horizontal direction and the normal. The plotter also does not constrain the extent in the horizontal and vertical direction to be the same. This enables the user to stretch the view in a given direction to tease out details in highly asymmetric structures. The basis and origin are specified in real-world coordinates whereas the extents are relative to the origin. So, for example, if the origin is specified as (x, y, z), the point at the center of the Viewport Pane will be set to this, irrespective of the basis vectors selected. The extent command is then interpreted as distance along the horizontal and vertical bases, so the point at the bottom right of the Viewport Pane would then be $(x, y, z) + e_{hor} \times (b_{1x}, b_{1y}, b_{1z}) - e_{vert} \times (b_{2x}, b_{2y}, b_{2z})$, where $\mathbf{b}_1$ is the unit vector in the horizontal direction, $\mathbf{b}_2$ is the unit vector in the vertical direction, and e_{hor} and e_{vert} are the extents in the horizontal and vertical direction respectively.

The next few sections describe how the plotter provides access to these concepts to plot different slices through the geometry defined in the input.

7.1.2 Launching The Plotter

The plotter is launched in geometry/tally mode using either an MCNP6 input file (with or without an associated runtape) or with just a runtape file by using one of the following two methods at the shell command prompt:

```
mcnp6_preview IP INP=filename [KEYWORD=value(s)]
```

or

```
mcnp6_preview Z RUNTPE=filename.h5 [KEYWORD=value(s)]
```

The first line with the `IP` option will initialize the geometry from the input file specified by the `INP` keyword, perform the normal input checks, and display the geometry. In this launch mode, if a runtape file is specified using the `RUNTPE` keyword, tallies will be read in from that file. The initial view in this mode of invocation is looking down the x axis with the z axis along the horizontal direction, the y axis along the vertical direction, origin set to $(0, 0, 0)$, and with extents of ± 100 in each direction.

The second line, invoked with the `Z` option, is useful once transport has been run and a runtape file has been generated. In this case the plotter initializes the geometry and tally data from the runtape file provided and presents a view of the first tally specified in the file. The window presented has a keyboard Input Pane at the bottom left and a view of the tally in the body.

The plotter provides the capability to switch seamlessly between these two views of the data. To switch from the geometry view to the tally view, the user types `MCPLOT` in the Input Pane, and to switch in the other direction, the user types `PLOT` in the Input Pane. Note that in Geometry View mode, if a runtape has not been specified, the `MCPLOT` command is ignored. An example of the two views displayed by launching the plotter in Geometry/Tally mode using the above two lines is shown in Fig. 7.2.

We recommend launching with the `IP` option to verify the geometry before running transport, especially with complicated geometries. The time that is required to plot and verify the geometry model is small compared with the potential time lost working with an erroneous geometry. See §7.1.5 for hints on debugging geometries.

The keywords allowed at the command line during launch are:

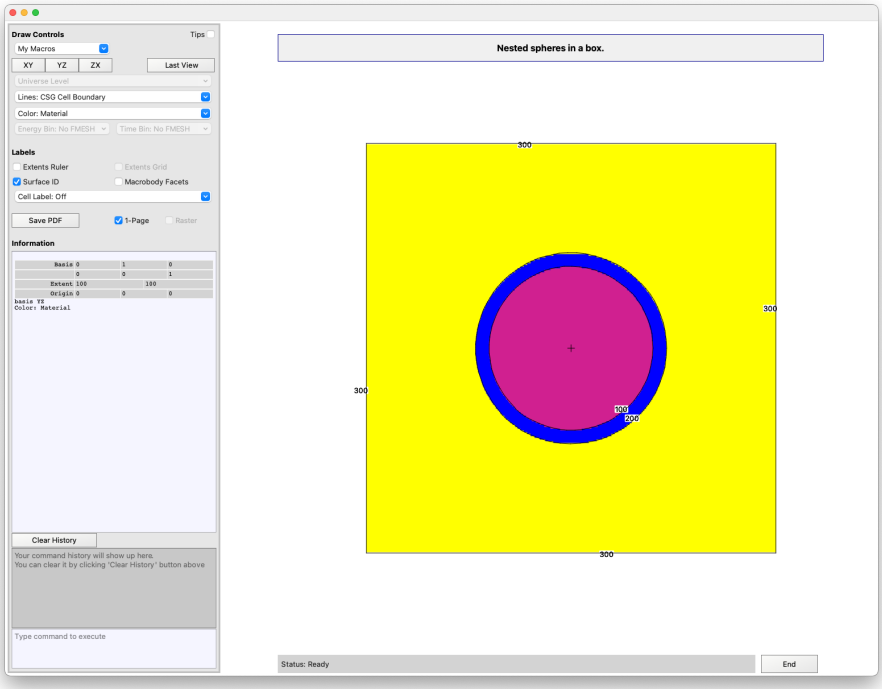
`RUNTPE`=*`filename`*	Name of the runtape that holds the tally data. When the plotter is invoked with *IP*, the user can optionally provide a runtape using the `RUNTPE` keyword either at the command line or in the Input Pane after launching. If the plotter is launched with *Z*, then this keyword is required at the command line.
`COM`=*`filename`*	Execute plotter commands within filename upon startup. When an end-of-file (EOF) is read, control is transferred to the plotter. In a production or batch situation, end the file with an `END` command to prevent transfer of control. Never end the `COM` file with a blank line. If COM is absent, the plotter starts up in interactive mode.
`PLOTM`=*`filename`*	Prefix for PDF/PS/PNG files. If this is not specified, the plotter will select a unique filename prefix to avoid overwriting existing files. When commands `SAVEPDF` or `SAVEPNG` are executed, the PDF and PNG files with the current view will be saved as **`PLOTM_#####.PDF`** or **`PLOTM_#####.PNG`** with the index incremented with every invocation. If the command `SAVEPS` is executed, the plotter will add a page to the file **`PLOTM_qt.PS`**.
`COMOUT`=*`filename`*	Write all plotter commands to file *`filename`*. The default name is **`comout`**. This output file can be used at a later time to regenerate the views of the current session by using all or part of the old COMOUT file as the COM file in the second run. Unique names for the output file, COMOUT, will be chosen by MCNP6 to avoid overwriting existing files.
`NOTEK`	Specifies off-screen rendering without displaying any windows. This is useful when running in batch mode or remotely over a slow network. The user can change the views using the keyboard commands in §7.1.6, and save the resulting view with one of `SAVEPDF`, `SAVEPNG`, or `SAVEPS` to save PDF, PNG, or PS files respectively.

To recreate Fig. 7.2a, launch the plotter with the following command

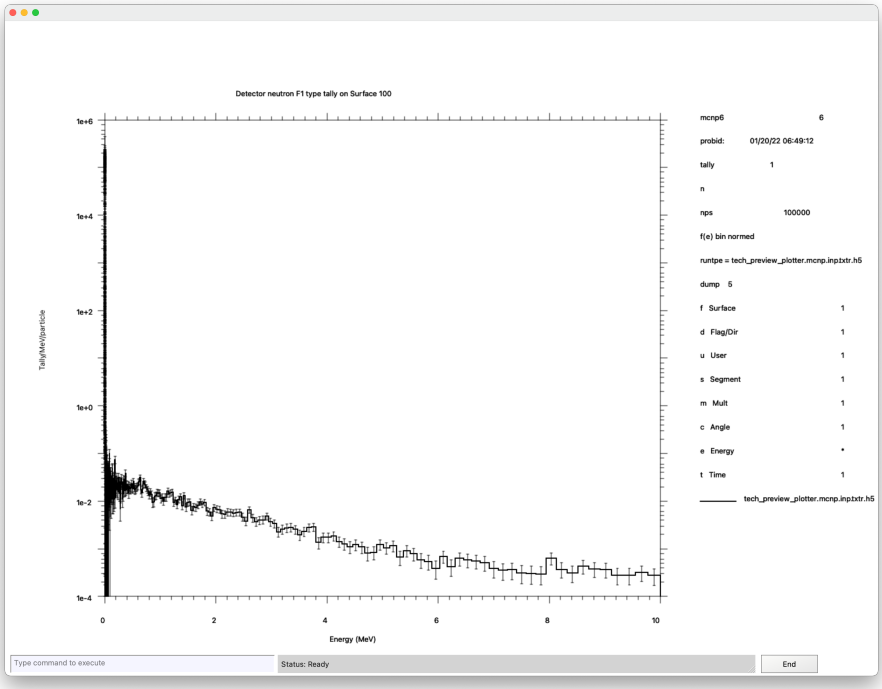
```
mcnp6_preview ip inp=tech_preview_plotter.mcnp.txt
```

The user can change the view from the default (Fig. 7.2a) by typing `BASIS` *`0 1 0 0 0 1`* `ORIGIN` *`1 1 1`* `EXTENT` *`15 15`* `LABEL` *`1 0`* in the Input Pane. This command sets the basis vectors to the y and z axes and the normal to the x axis. The origin is selected as $1, 1, 1$ and the extents of the viewport are then set to origin ± 15 in the horizontal and vertical directions. The label command turns on surface labels and turns off the cell labels. The input used here can be found in program Listing 7.1.

The user can also change the view by using the mouse in the Viewport Pane [§7.3.1], the controls in the Control Pane [§7.3.2] or by entering commands from Table §7.1.6 in the Input Pane. A command consists of a keyword typically followed by some parameters. Multiple keywords and their parameters can be entered on



(a) Initial geometry view [§7.1]



(b) Initial tally view [§7.2]

Figure 7.2: Initial views displayed when launching the plotter with either `IP` or `Z` specified at command line. In geometry mode, a default view of the geometry is provided with an x axis normal, origin at $(0, 0, 0)$, and extents of ± 100. In tally mode, the first tally from the results is displayed.

each line. Keywords and parameters are space-delimited. Keywords can be shortened to any degree as long as they are not ambiguous. Parameters following the keywords cannot be abbreviated. Numbers can be entered in free-form format and do not require a decimal point for floating point data. Keywords and parameters remain in effect until changed. Note: if a shortened, ambiguous keyword is used, the entire command line from that point on will be rejected and a message to that effect will be printed to the terminal. The most common keywords are represented in the Control Pane in the form of buttons and menus. These can be used to change the rendered slice as well. For example, to color the cells by their IDs, the user selects *Cell* from the Color menu. Similarly the user can toggle the display of surface labels by clicking the *Surface ID* checkbox, and select a label for all the cells by selecting an item from the Cell Label menu.

⚠ Caution

Placing the plot plane exactly on a surface defined by the geometry can stall the geometry engine.

If the view plane selected is coincident with a surface defined by the geometry, undefined behavior can occur, and performance can drop significantly. For example, if the input geometry has a PX plane at $x = 0$, that plane coincides with the default plot plane. When this occurs, some portion of the geometry may be displayed in dotted lines, which usually indicates a geometry error or part of the geometry may simply not show up at all. Very infrequently the code may crash with an error. To prevent all these unpleasantries, move the plot plane a tiny amount away from surfaces.

7.1.3 Saving the View

A PDF file is saved by clicking the Save PDF button or typing SAVEPDF in the Input Pane. To save PNG or PS files, the user would type SAVEPNG and SAVEPS respectively in the Input Pane. Each of the three commands SAVEPDF, SAVEPNG, and SAVEPS saves the current view. If the user enters other commands on the same line, e.g. **ORIGIN 20 1 0 SAVEPS EX 10 10**, then the origin would first be set to $(20, 1, 0)$ and then the view would be saved to the PS file. Then the extents would be changed. Although the button to save PDF files is not visible in all modes, the keyboard commands are available anytime the Input Pane is visible.

7.1.4 Plotting Embedded-mesh Geometries

The plotter supports color-shaded plotting of the materials, mass density, or number density of an imported embedded mesh. For these cases the values from the external mesh geometry file (typically a LNK3DNT or Abaqus-style file) are used; these values may vary element to element.

For mass density (den) and number density (rho) plots, each element will be shown in one solid color. The element net value is displayed, i.e., the net mass density or net number density of the element. The color distribution is set by the minima and maxima. These net values are also the values reported in the Information Pane when a cell is clicked.

For material plots, multi-material zones may appear striped as the color to plot is chosen randomly based on the material mass fraction. This means that redrawing a color-by-mat plot may give a slightly different striping. For example, in a two-material element with a 50/50 mass-fraction mix, approximately 50/50 color striping will display horizontally. If Material is selected from the Color menu, clicking on a spot containing multiple materials will randomly select which material to report. Repeated clicking on such a spot may show different materials on different clicks. Void elements in the mesh are not shaded (i.e., shown as white) on material plots.

7.1.5 Geometry Debugging

Surfaces appearing on a plot as red dashed lines usually indicate that adjoining space is improperly defined. Dashed lines caused by a geometry error can indicate space that has been defined in more than one cell or space that has never been defined. These geometry errors need to be corrected. Dashed lines also can occur because the plot plane corresponds to a bounding planar surface. The plot plane should be moved so it is not coincident with a problem surface. Dashed lines can indicate a cookie cutter cell or a DXTRAN sphere. These are not errors. The reason for the presence of dashed lines on an MCNP6 plot should be understood before running a problem. When checking a geometry model, errors may not appear on the two-dimensional slice chosen, but one or more particles will get lost in tracking. To find the modeling error, use the coordinates and trajectory of the particle when it got lost. Entering the particle coordinates as the ORIGIN and the particle trajectory as the first basis vector will result in a plot displaying the problem space.

7.1.6 Keyboard Commands In Geometry View

In addition to mouse manipulation described in the section on Navigating The Plotter [§7.3], the view can be changed using a number of keyboard commands. We list here the different keyboard commands that can be issued. All keyboard input to the plotter should be followed by the Enter key for the plotter to act on them. For brevity we have omitted this repetitive piece of information in the text. Hence, when the manual directs the user to type a command in the Input Pane, it is explicitly assumed that the user will follow that command by hitting the Enter key to execute it.

`BASIS` *`b1x b1y b1z b2x b2y b2z`*	Specify the basis vectors for the horizontal and vertical axes of the Viewport Pane as triples of $x\,y\,z$ for each of the vectors. The vectors need not be orthogonal, but must not be collinear. The plotter will normalize the vectors to unit vectors, determine an orthogonal basis and display the orthogonal basis in the Information Pane.
`CENTER` *`dh dv`*	Translate the center of the viewport by *`dh`* along the horizontal basis and *`dv`* along the vertical basis
`COLOR` *`command`*	Turns filling of cells on or off, or change coloring parameters based on *`command`*. Valid Entries for *`command`* are:

`ON`	Turn filling of cells on
`OFF`	Turns off filling of cells
`[50-5000]`	Set resolution of shape decomposition, with *`5000`* representing the highest resolution (smoothest curves) gained at the cost of performance
`BY CEL`	Color by Cell IDs
`BY DEN`	Color by mass density
`BY GRADIENT`	Use a 256 color palette to display values
`BY MAT`	Color by material (default). See also `SHADE` command
`BY RHO`	Color by atomic density
`BY SOLID`	Use a solid color to represent values
`BY TMP`	Color by temperature

`END`	Ends the current plotter session
`EXTENT` *`ex [ey]`*	Sets the extents of current view. If only *`ex`* is specified, *`ey`* is set equal to *`ex`*

EBIN *n*	Select Energy Bin *n* to display when an FMESH is active
FACTOR *f*	Scale the plot by a factor of *f* i.e. zoom by *1/f*
FLINES *0|1*	If an FMESH is active, turn on or off drawing the outlines of the FMESH cells
FMESH *ID*	If no runtape file has been read in, displays the outline of the FMESH grid with the specified *ID* defined in the input. If a runtape has been read in using either the RUNTPE command or as a command line argument, display the values of the selected Energy and Time bin tallies superimposed on the geometry. If *ID* is set to *OFF*, revert to filling the geometry by material number.
HELP	print available commands to the terminal window
LABEL *slabel [clabel [par]]*	Put labels of size *slabel* on the surfaces and, optionally, labels of size *clabel* in the cells. The parameter, *par*, following *clabel* is further optional and defaults to MAT. The sizes specified by *slabel* and *clabel* are relative to 0.01 times the height of the view window. If *slabel* or *clabel* is zero, that kind of label will be omitted. The allowed range of sizes for the labels is [0.2–100]. In case Cell labels are set to on, The cell label is selected by providing one of the following entries:

CEL	cell ID
DEN	mass density
DXC:$\mathscr{P}$	DXTRAN contribution by particle type
EXT:$\mathscr{P}$	Exponential transform by particle type
FCL:$\mathscr{P}$	Forced collision by particle type
FILL	Filling Universe
IJK	Lattice indices of repeated structure/lattice geometries
IMP:$\mathscr{P}$	Importance by particle type
LAT	Lattice type
MAS	Mass
MAT	Material number
NONU	Fission turnoff
PDn	Detector contribution (*n* = tally number)
TMPn	Temperature (*n* = index of time)
U	Universe number
WWNn:$\mathscr{P}$	Weight-window lower bound (*n* = energy or time interval) by particle type

LEVEL *n*	Plot only the n^{th} level of repeated structure geometries
MBODY *on|off*	Display macrobody surface number in addition to surface labels. Only available if surface labels are set to on.
MCPLOT	Switch the view to tally viewing. If no RUNTPE has been read in, this command does nothing.
MESH *n*	Controls plotting of the weight-window, weight-window generator, FMESH, and TMESH superimposed structured mesh. The valid values of *n* are:

0	No lines on plot
1	Geometry cell outlines

2	Weight-window mesh outlines
3	Weight-window mesh + geometry cell outlines
4	Weight-window generator mesh outlines
5	Weight-window generator mesh + geometry cell outlines
6	TMESH Tally outlines
7	TMESH Tally outlines + geometry cell outlines
8	FMESH Tally outlines
9	FMESH Tally outlines + geometry cell outlines

`MYMACROS` *add|load|save|remove|addCurrentView*
See §7.3.2.2 for details

`ORIGIN` *ox oy oz* — Sets origin for the Viewport Pane

`PX` *vx* — Set the x coordinate of the origin to *vx* and set the view to point down the x axis. This is equivalent to the command `BASIS` *010001* `ORIGIN` *vx vy vz*,where *vy* and *vz* are the current y and z coordinates of the origin.

`PY` *vy* — Set the y coordinate of the origin to *vy* and set the view to point down the y axis. This is equivalent to the command `BASIS` *001100* `ORIGIN` *vx vy vz*,where *vx* and *vz* are the current x and z coordinates of the origin.

`PZ` *vz* — Set the z coordinate of the origin to *vz* and set the view to point down the z axis. This is equivalent to the command `BASIS` *100010* `ORIGIN` *vx vy vz*,where *vx* and *vy* are the current x and y coordinates of the origin.

`RUNTPE` *filename* — Read in tallies from the specified runtape. This command can also be used to load up a runtape file from a different run with the constraint that the geometries in the two files must be identical. If the geometries do not match, the plotter will fail with a **SIGSEGV**.

`SAVEPDF` — Saves current Viewport and Information panes to a PDF file

`SAVEPNG` — Saves current Viewport and Information panes to a PNG file

`SAVEPS` — Saves current Viewport and Information panes to a legacy postscript file

`SCALES` *0|1|2* — Specify whether the extents ruler and grid are drawn on the viewport: 0 turns off ruler and grid, 1 draws the ruler, and 2 draws both ruler and grid.

`SHADE` *m1=value1 m2=value2 ...*
This command is only valid when `COLOR` *by mat* is in effect. This sets the shade for material *m1* to *value1*, and so on, where values are integers in the range [1–64]. Alternately, values could be specified as colors (e.g. red, blue, green, etc.). Color names are case sensitive. The command `options` will list available color values. Indices in the list run from top left to bottom right.

`STATUS` — Print currently selected plot options to the terminal

`TBIN` *n* — Select time bin to display when an `FMESH` is active

`THETA` θ — Rotates the plot by θ degrees counterclockwise around the center of current view.

`TMESH` *ID* — If a RUNTPE has been read in then the plotter colors are the `TMESH` cells by the tally values and superimposes them on the geometry.

7.2 Viewing Tally Results

Although visualizing the geometry is extremely useful, oftentimes one would like to take a look at the results of a run. To do this run the MCNP code with **mcnp6_preview i= tech_preview_plotter.mcnp.txt n= preview_1** which creates the runtape file **preview_1r.h5**. The results are then visualized using the command:

```
mcnp6_preview Z RUNTPE= preview_1r.h5 [keyword=value]
```

This will generate Fig. 7.2b. The *Z* argument to the plotter reads in the geometry and the results from the runtape file specified by the RUNTPE argument. In this mode one can graph the different tallies specified in the input file/runtape file. The default view shows the lowest numbered tally on a log-lin scale with energy on the x axis and tally value on the y axis. Typing PRINTAL in the Input Pane at the bottom of the window will print out the tallies available in the terminal window. For this file, there are two tallies: 1 and 21. To switch between them type the commands TALLY 1 and TALLY 21.

An alternate way to visualize the tally results is to launch with the IP option, as in §7.1, specifying a runtape file either on the command line or in the Input Pane with RUNTPE= *<filename>*, and then typing MCPLOT in the Input Pane.

Except when viewing FMESH and TMESH tallies, the Tally Viewing mode is driven by command line input in the bottom left.

7.2.1 Plotting Standard (F) Tally Results

The standard tallies in the MCNP code specified using the F cards are stored as 8-dimensional arrays. Not surprisingly, plotting 8-D objects is beyond the current capabilities of the plotter. Instead the plotter allows the user to display 2-D projections of this 8-D space. The simplest of these views displays one of the dimensions along the x axis and the corresponding tally values on the y axis. The default view is such an example. It displays the tally particles as a function of energy on a log-linear scale. The axes can be switched between log and linear by using the command *<x-mode><y-mode>* where *<x-mode>* and *<y-mode>* are either LOG or LIN. So to switch to a log-linear scale the user would type LOGLIN resulting in the image shown in Fig. 7.3. Similarly to switch to a log-log scale, the user would enter LOGLOG in the Input Pane and so on. Plotting tallies in the MCNP code is described in detail in §6.3, §6.4, and §6.5. The user is directed to those chapters for the commands available since the new plotter provides identical capabilities.

7.2.2 Viewing FMESH And TMESH Superimposed Mesh Tallies

When a runtape file is loaded, FMESH and TMESH tally results within the file can be superimposed on the geometry. Available tallies are enumerated in the Color menu. To view FMESH tally results the user must first switch to the geometry view by typing PLOT. Then the user can select the FMESH *14* menu item from the Color menu in the control pane. When an FMESH is selected, the entries *FMESH* and *FMESH + Cell Boundary* are enabled in the Lines menu enabling the user to draw the mesh and geometry outlines superimposed on the tally results. The first time an FMESH is selected, the view is auto-adjusted to change orientation and scale to show the entire mesh. For rectangular meshes, the horizontal axis is in the direction of the dimension with the greatest number of bins, and the vertical axis is in the direction of the dimension with the second greatest number of bins. For cylindrical meshes, the horizontal axis is along the axis of the cylinder and the vertical axis is along the $\theta = 0$ plane. The center of the plot in both cases is at the center of the mesh. In

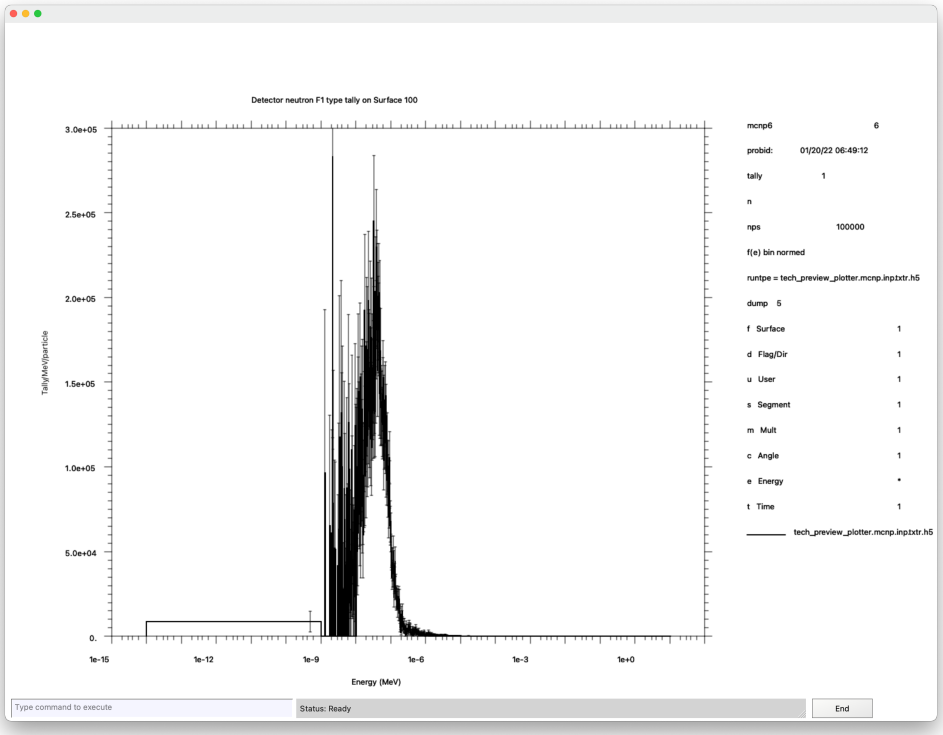


Figure 7.3: Energy tally plotted on a log-linear scale. This is achieved by typing `LOGLIN` in the Input Pane when the tally is displayed. The ability to switch the axes between log and linear mode is available any time a tally chart is displayed by entering the command `<x-mode><y-mode>` where `<x-mode>` and `<y-mode>` are one of `LOG` and `LIN` for log and linear mode respectively.

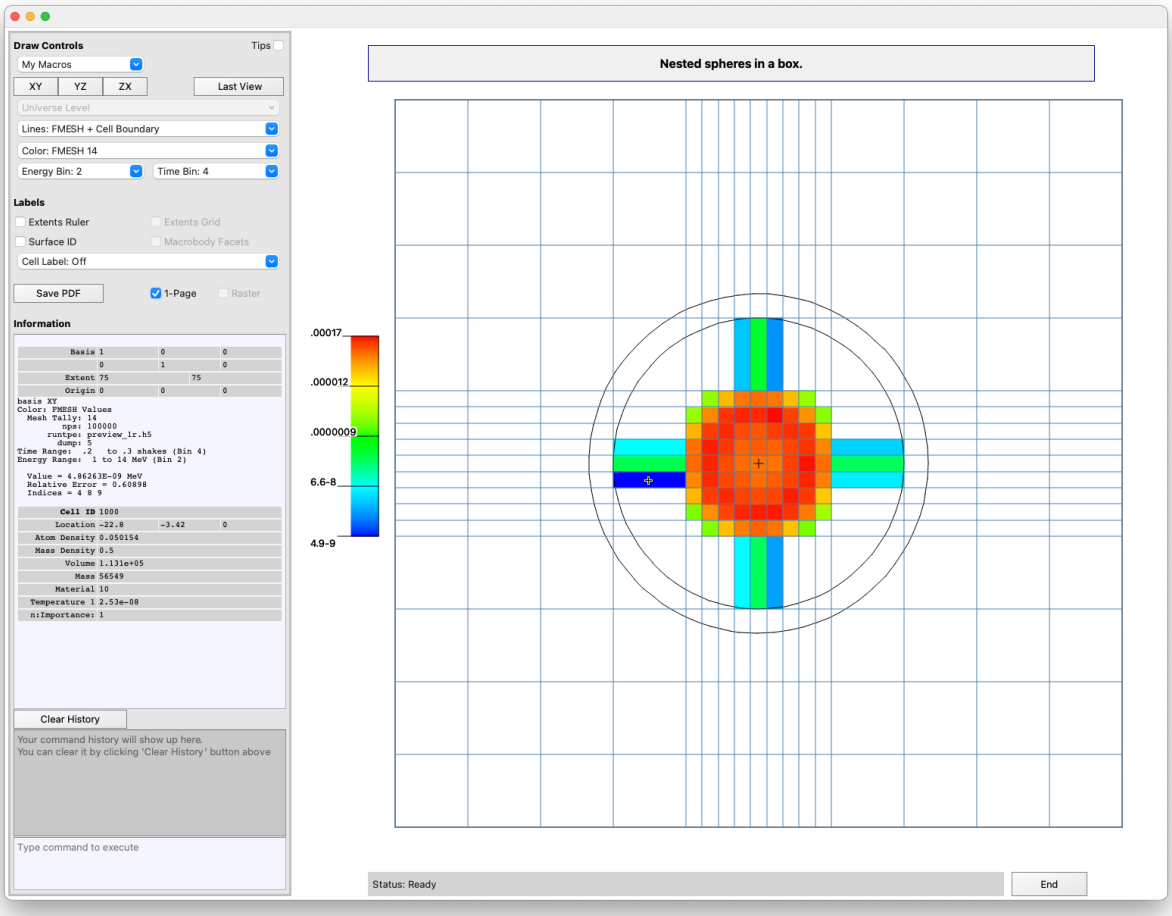


Figure 7.4: Results from a short calculation showing tallies on an FMESH. In this view, time bin 4 and energy bin 2 are selected for FMESH 14. FMESH cell $(8, 9, 9)$ has been clicked and is indicated by the cross-hair with the yellow halo in the image.

this case, the default `FMESH` view results in a cartesian mesh centered at $(0, 0, 0)$ with a dimension of 150 in each direction. To display this `FMESH`, the view is first rotated so that the z axis is the normal, with the x axis along the horizontal direction. Then the extents for the view are reset to ± 75. This is shown in 7.4.

To select the `TMESH` `111` that is defined in the input, the user would either enter `TMESH` ***111*** in the Input Pane or select ***TMESH 111*** from the `Color` menu. As with `FMESH` tallies, when a `TMESH` is selected, the entries ***Mesh Tally*** and ***Mesh Tally + Cell*** are enabled in the `Lines` menu enabling the user to draw the mesh and geometry outlines superimposed on the tally results. This is shown in 7.5.

7.3 Navigating the Plotter

7.3.1 Viewport Pane

The user interacts with the Viewport using the mouse to interrogate, translate, rotate, and zoom the scene displayed. Left-clicking on a cell in the viewport will display additional information in the Information Pane. Left-clicking the mouse and dragging the rendered image around will dynamically change the origin for the

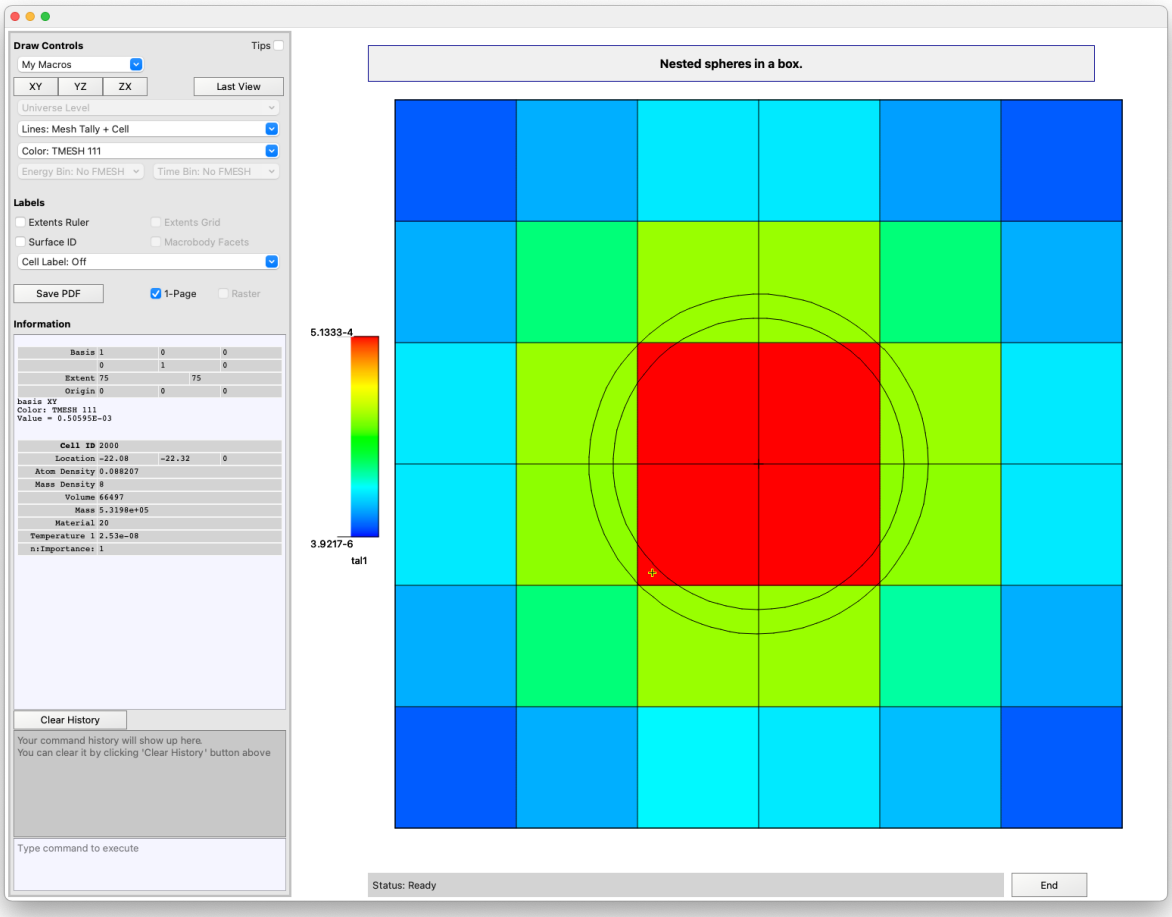


Figure 7.5: Results from a short calculation showing tallies on a TMESH. In this view, TMESH 111 is selected from the Color menu.

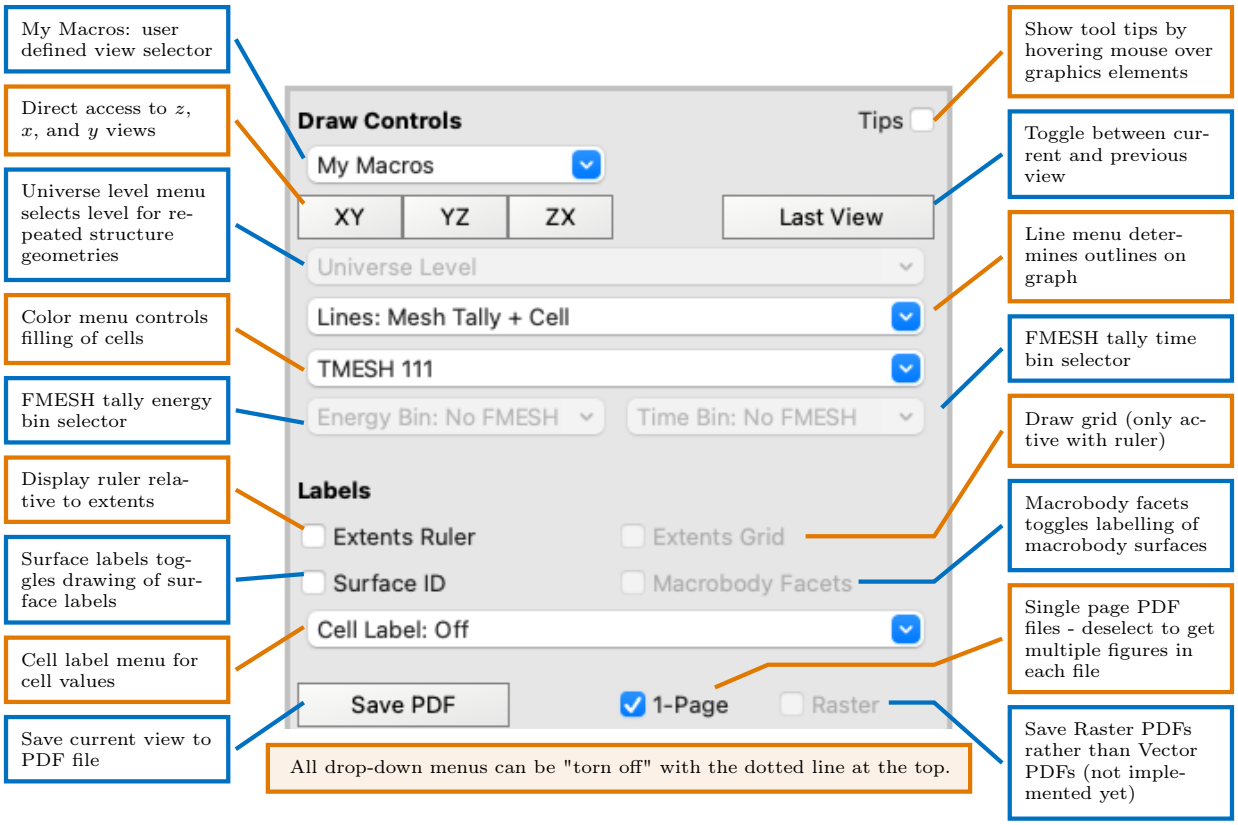


Figure 7.6: Control Pane of graphical interface annotated with functions of the different elements.

plot and translate the view. If the Shift ⇧ key pressed when the mouse button is depressed, then dragging will rotate the basis vectors around the center of the Viewport. If the Ctrl key pressed when the mouse button is depressed, dragging the mouse will zoom the rendered view around the point where the mouse was pressed proportional to the vertical displacement of the mouse from the original point. In the zoom mode the origin is dynamically shifted so as to keep the originally clicked point stationary within the Viewport.

The rendered view can also be shifted by using a combination of the Ctrl and the four arrow keys ← , → , ↑ , ↓ to translate the view left, right, up, or down respectively. Zooming around the origin is achieved by using Ctrl + / Ctrl - . Keeping the Shift ⇧ key pressed with these combinations will increase the magnitude of the change. Note that on Macintosh computers the Ctrl key will be mapped to the Command key. This is because the default key bindings on macOS use the Ctrl key to navigate different desktops.

The `EXTENT`, `ORIGIN` and `THETA` commands from the listing in §7.1.6 enable the user to emulate the results of mouse based view control. There are no keyboard equivalents for the selection of a cell by mouse click.

The `End` button at the bottom right of the Viewport Pane will exit the plotter. The status bar to the left of the `End` button provides hints regarding the state of the plotter driver when it is executing a command in the background, such as when plotting a complex geometry which can cause the interface to become non-responsive while the rendered view is being calculated.

7.3.2 Control Pane

The Control pane is the set of buttons, checkboxes, and menus shown in Fig. 7.1. An annotated view is shown in Fig. 7.6 and a description of the elements is given in §7.3.2.1. These graphical elements are enabled/disabled

depending on the features present in the input file. The contents of menus change dynamically based on input. All menus in this interface can be "torn off" making it easier to navigate auto-populated menu items that the user might want to switch between. Except for Tips, all other Control Pane actions can be emulated using keyboard input.

7.3.2.1 Control Pane Elements

The following elements are components within the Control Pane. If a command line equivalent from §7.1.6 is available, it is listed in parenthesis at the end.

`Cell label`	Controls display of cell labels on the rendered slices. Individual menu items are enabled or disabled depending on the active elements within the input file. (`LABEL`)
`Color`	Controls how the cells in the rendered slice are filled. The selections are Atomic Density, Mass Density, Cell ID, Material ID, Temperature, Importance, and available `FMESH` and `TMESH` tallies. (`COLOR`, `FMESH`, `TMESH`)
`Energy Bin`	List of energy bins in the selected `FMESH`. When an energy bin is selected, the results displayed are restricted to only that bin. When no runtape is provided the entries are grayed out, but can be inspected to ensure that the bins found are the ones expected. This menu is dynamically instantiated. (`EBIN`)
`Extents Grid`	Draws a grid within the view port. This checkbox is disabled if Extents Ruler is not checked. (`SCALES`)
`Extents Ruler`	Draws a ruler around the viewport that shows distance from the origin. The ruler goes from *`-extent`* to *`+extent`* in the x and y directions. (`SCALES`)
`Lines`	Controls display of cell outlines. The outlines available are the Constructive Solid Geometry cells, Weight Window cells, Weight Window Generator Mesh cells, `FMESH`, and `TMESH` outlines. (`MESH`)
`Macrobody Facets`	Adds macro-body facet suffixes to surface ID labels. Only active if Surface ID is checked. (`MBODY`)
`My Macros`	A User controlled menu for executing arbitrary plotter commands [§7.3.2.2]. (`MYMACROS`)
`Surface ID`	Controls display of surface ID labels on the rendered slices. (`LABEL`)
`Time Bin`	List of time bins in the selected `FMESH`. When a time bin is selected, the results displayed are restricted to only that bin. As with energy bins, the entries are grayed out and for information only when no runtape is provided. This menu is dynamically instantiated. (`TBIN`)
`Tips`	Enables/disables the display of tool tips when the mouse is hovered over other graphical elements.
`Universe Level`	Controls display of repeated structure geometries [§2.2.2]. This menu is disabled if no Universes are defined in the problem. (`LEVEL`)
`XY` / `YZ` / `ZX`	Direct access to common basis functions for views down the z, x, and y axis respectively. (`BASIS`)

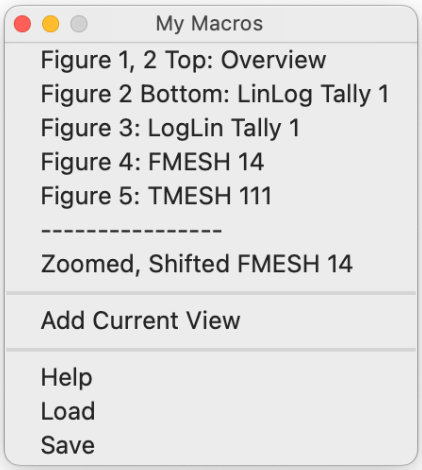


Figure 7.7: Sample `My Macros` menu created by launching the MCNP code as shown in §7.2 and loading Listing 7.2 using either the command `MYMACROS load tech_preview_plotter_mymacros.txt` or the `Load` menu item in the `My Macros` menu. The first five entries created enable the user to recreate the figures in this chapter. The sixth entry is an emulated separator with no command associated. The final entry provides a zoomed and shifted view of FMESH 14 defined in the input file.

7.3.2.2 My Macros Menu

The `My Macros` menu allows the user to create a menu to execute arbitrary graphics commands. This could, for example, be a set of commands to load up a specific view, or to switch to a given `FMESH` tally at a specific energy bin. Our expectation is that the menu will primarily be used to define a set of views that can be provided to collaborators or for checking critical regions of a given geometry so that it is easy to recreate a view across different invocations of the code. On start up, the menu has three entries: ***`Add Current View`***, ***`Help`***, and ***`Load`***. Selecting ***`Help`*** will print out help on how to use the `My Macros` menu to the terminal. If the ***`Add Current View`*** menu item is selected, it will add the current view in the viewport to the menu as *View 1*. This will also add a *Save* entry to the menu. Selecting the ***`Add Current View`*** item a second time will increment the View number. Once views are loaded in the menu, they can be saved to a file for loading at a future date by using the ***`Save`*** item. The format of the **`tech_preview_plotter_mymacros.txt`** file is very simple with each line representing a view. The first word of the line is the label that is shown in the menu. Spaces can be included in the label by using quotes to enclose the entire label. The rest of the line is the command that is executed. The saved file can be loaded during a different invocation of the MCNP code by using the ***`Load`*** item and selecting the saved file. Duplicate labels in the file will not overwrite previous entries but will result in duplicate entries in the menu.

A sample input file can be found in program Listing 7.2 with entries that will regenerate the figures in this chapter when loaded with the command `MYMACROS load tech_preview_plotter_mymacros.txt` or the `Load` menu item in the `My Macros` menu with the MCNP code invoked as shown in §7.2. The menu shown in Figure 7.7 is displayed on the screen. The first five entries in this menu enable the user to recreate the figures in this chapter. The sixth entry is an emulated separator with no command associated. The seventh entry provides a zoomed, shifted view of `FMESH14` defined in the input file. Following the user-defined entries in the menu are the default entries for adding the current view, printing help to the terminal, loading menu entries from a file or saving the current entries to a file.

Below are the keyboard equivalents of the `My Macros` menu:

`MYMACROS addCurrentView`
: Adds the current view to the `My Macros` menu

`MYMACROS add "view name" commands to execute`
: Adds *view name* to the `My Macros` menu. The first word is interpreted as the label to display in the menu and the rest of the words up to the end of the line are taken as the command to execute. If spaces are desired in the label then use double quotes as shown. No special treatment is needed for the words in the command.

`MYMACROS load filename`
: Loads macro file with predefined macros for the menu. This command will "Tear off" the `My Macros` menu. at the current mouse location.

`MYMACROS save filename`
: Saves the current views defined in the `My Macros` menu to the given filename. This file can be edited in a text editor to modify the view settings.

`MYMACROS remove "view name"`
: Removes the named view from the current menu. Use quotes to encapsulate menu entries with spaces in the name.

7.3.3 Information Pane

Left-clicking on cells in the Viewport Pane will display extended cell information in the Information Pane. A sample is shown in Fig. 7.8. Changing the view either using the mouse or keyboard commands to translate/rotate/zoom the picture will erase the current cell information. Using the `Save PDF` button will save the Viewport Pane to a PDF file and include the contents of the information pane.

The information displayed includes the following fields:

Current View
: This includes the `BASIS`, `EXTENT`, and `ORIGIN` for Viewport Pane. This information can be copied and pasted into the Input Pane for editing/reuse. If the current view is an on-axis view, then that view will be printed as **`basis YZ`**/**`basis XZ`**/**`basis XY`** as well. This information is always displayed in the information pane.

FMESH Related Fields
: If an `FMESH` is active, the next few lines provide `FMESH` related information including:

  Number of source histories (`NPS`)

  `KCODE` cycles

  Runtape file name

  Dump number

  The `FMESH` Tally ID

  Comments from the `FMESH` description

  Energy range selected

  Time range selected

  `FMESH` value for cell that is clicked

  Relative error for cell that is clicked

  Indices of cell that is clicked

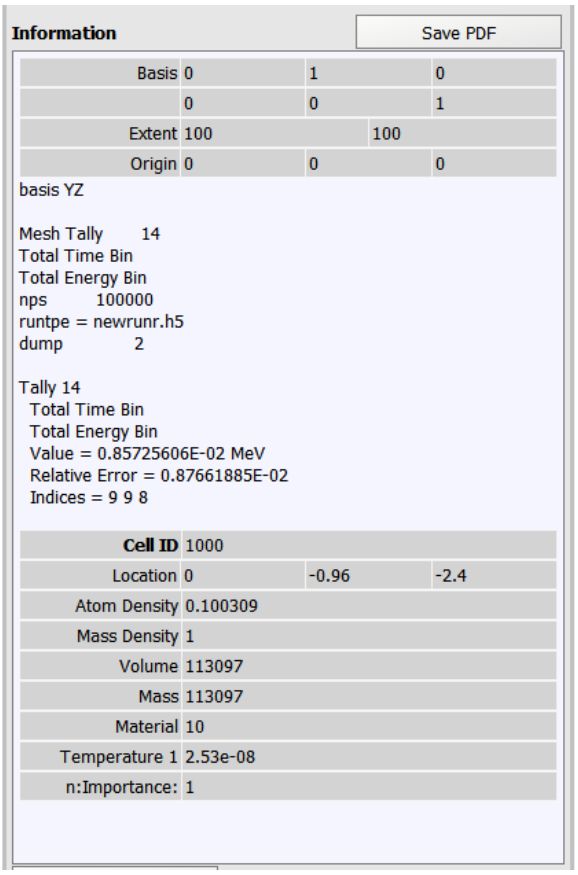


Figure 7.8: Example of information displayed when a cell is clicked in the Viewport Pane with the left mouse button. The information displayed is context sensitive and will include only fields that are defined for the current input file/runtape.

Cell Information	MCNP cell information that includes:
	Cell ID
	Coordinates of the clicked point
	Universe, Lattice and Fill Universe, and Lattice ijk for repeated data structures
	Atom density
	Mass density
	Volume
	Mass
	`PWT` value
	Material information
	Temperature(s)
	Importance for each particle type
	Forced collision values for each particle type (`FCL`)
	Weight window lower bounds for energy or time interval values by window number and particle type (`WWN`:$\mathscr{P}$)
	Detector contribution values (`PD`N)

Information that is not part of the calculation will not be displayed. For instance, if an input file does not use `WWN` values, then those values will be omitted from the clicked-cell-information.

7.3.4 Input Pane

The input pane sits at the bottom left of the main window and is shown in Fig. 7.9. It consists of three elements: An input area at the bottom, a history area in the middle, and a `Clear History` button at the top. Keyboard commands are entered in the input area at the bottom. It is not necessary to click in the input area. Any time the mouse focus is on the Viewport Pane, the Information pane or the Control Pane, the keyboard focus will be set to the input area of the Input Pane. A keyboard command is terminated by the Enter key. Once the Enter key is pressed, processing is started for executing the command just entered. As described in §7.1, a command is a keyword from §7.1.6 followed by appropriate parameters. Multiple keywords can be entered on a line, each followed by its parameters. If an error is detected in either a keyword or its parameters, then the rest of the command line from that point onwards is ignored. Once a command is entered, it appears in the *History* section of the Input Pane. Commands can be copied from the *History* section and pasted into the *Input* area for editing/execution using the arrow keys. Additionally, previously typed entries can be accessed using the ↑ and ↓ keys. Once the entry desired is displayed, the command can be edited, using the ← and → keys to move the cursor left or right if required. Once editing is complete, pressing the Enter key will execute it.

7.4 Program Listings for Generating Images

The MCNP input file used in the examples in this chapter is given in Listing 7.1. The `MYMACROS` input file that generates figures in this chapter is given in Listing 7.2.

Listing 7.1: tech_preview_plotter.mcnp.txt

```
1 Nested spheres in a box.
2 c Cell Definitions
```

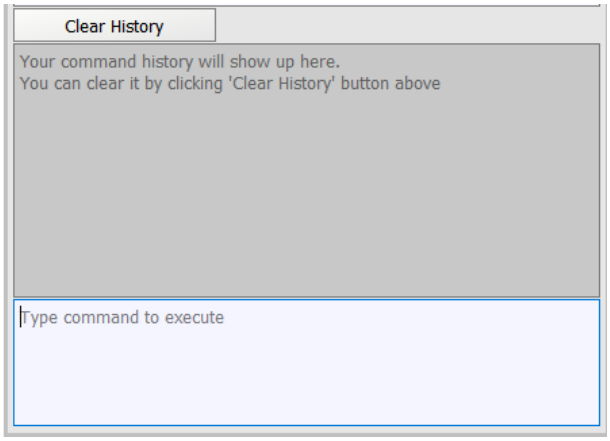


Figure 7.9: The Input Pane is where users can enter keyboard input. Entries are typed in the text box at the bottom left followed by the Enter key. Previous entries are shown in the history label and can be copied and pasted into the entry pane. Users can scroll through previous entries in the text entry pane using the ↑, ↓, ←, and → keys.

```
1000 10 -0.5     -100      imp:n=1 $ Inner sphere
2000 20 -8        100 -200 imp:n=1 $ Outer sphere
3000 30 -1.20e-3  200 -300 imp:n=1 $ Air box
9999  0           300      imp:n=0 $ Graveyard

c Surface Definitions
100 so 30
200 so 35
300 rpp -75 75 -75 75 -75 75

c Data Cards
mode n
sdef pos = 0 0 0 erg = 14  $ 14-MeV isotropic point source of neutrons at the origin
c
fc1 Detector neutron F1 type tally on Surface 100
f1:n 100    $ surface current tally
e1 1e-9 999ilog 10  $ log scale, 0.001 eV to 10 MeV
c1 -.866 -.5 0 0.6 0.866 1.0 $ cosines for cosine tally
c
fc21 Detector neutron F1 type tally on Surface 100
f21:n 200   $ surface current tally
e21 1e-9 999ilog 10 $ log scale, 0.001 eV to 10 MeV
c21 -.866 -.5 0 0.6 0.866 1.0 $ cosines for cosine tally
c
m10  1001.80c 2    8016.80c 1      $ Water, 50% density
mt10 lwtr.10
m20 26056.80c 0.97 6000.80c 0.03   $ Pseudo Carbon Steel
m30  7014.80c 0.79 8016.80c 0.21   $ Pseudo Air
c
fmesh14:n geom = xyz origin = -75 -75 -75 out = xdmf
          imesh = -60 -15 15 60 75 iints = 1 3 9 3 1
          jmesh = -60 -15 15 60 75 jints = 1 3 9 3 1
          kmesh = -60 -15 15 60 75 kints = 1 3 9 3 1
          emesh =     1 14 100
          eints =     1  1   1
          tmesh =  0  1 10 100
          tints =  1 10  9   1
```

```
fmesh24:n geom = xyz origin = -75 -75 -75 out = xdmf
          imesh = 75 iints = 5
          jmesh = 75 jints = 5
          kmesh = 75 kints = 5
          emesh =     1 14 100
          eints =     1  1   1
          tmesh =  0  1 10 100
          tints =  1 1  1   1
c
rand gen=2 seed=12345
print
prdmp j 25000
nps 1e5
tmesh
rmesh111:n
cora111 -75 5i 75
corb111 -75 5i 75
corc111 -75 5i 75
endmd
```

Listing 7.2: tech_preview_plotter_mymacros.txt

```
"Figure 1, 2 Top: Overview" plot FMESH OFF color by mat ba 1 0 0 0 1 0 or 0 0 0 ex 75 mbody off la 0 0
"Figure 2 Bottom: LinLog Tally 1" mcplot linlog tally 1
"Figure 3: LogLin Tally 1" mcplot loglin tally 1
"Figure 4: FMESH 14" plot FMESH 14 ebin 2 tbin 4 ba 1 0 0 0 1 0 or 0 0 0 ex 75 mbody off la 0 0 mesh 9
"Figure 5: TMESH 111" plot TMESH 111 ba 1 0 0 0 1 0 or 0 0 0 ex 75 mbody off la 0 0 mesh 7
"----------------"
"Zoomed, Shifted FMESH 14" plot FMESH 14 ba 0 1 0 0 0 1 or 0 7.5 0 ex 15 15 la 1 0
```

Chapter 8

Unstructured Mesh

8.1 Introduction

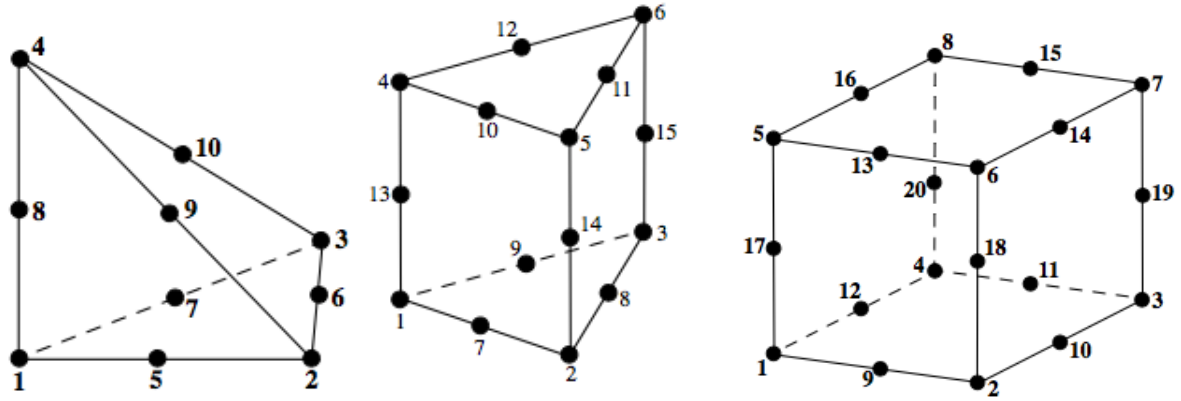
The MCNP code has a general geometry specification capability by using a constructive solid geometry (CSG) approach. This approach consists of using first- and second-order implicit surfaces (plane, sphere, etc.,) and fourth-order elliptical tori with Boolean operations to define regions of space [§1.3, §2.2]. This CSG capability has been well-tested and verified. However, it has long been recognized that as the model complexity increases, creating a CSG model is difficult, tedious, and error prone [336–338]. Consequently, innovators have taken on the task of developing a better way to construct geometries in the MCNP code. The MCNP code addresses this issue by permitting the user to embed an unstructured mesh (UM) representation of a geometry in its CSG cells to create a hybrid geometry using universe (`U`) keywords on cell cards. Particle tracking methods for CSG and UM models are different. The code implementation for the MCNP UM input processing and particle tracking is known as "REGL," which stands for Revised Extended Grid Library. This library is also called the UM library in the MCNP manual.

The UM capability was originally designed to work with an unstructured mesh created with the Abaqus/CAE [339] software and the ASCII input file that it generates. Many other mesh generation tools have the ability to generate a mesh from a solid model that can be exported as the Abaqus input file format. It is the user's responsibility to verify that these meshing tools are generating the Abaqus input file format that meets the MCNP specification; see §8.7. In addition, the information in the Abaqus input file can be converted to the MCNP code-friendly **`MCNPUM`** file type (now deprecated, [DEP-53424]). Version 6.3 of the MCNP code introduces the new capability of tracking particles on a UM model formatted as an HDF5 file; the details of the HDF5 UM mesh format can be found in §D.6.

8.2 Terminology

The MCNP code can only process an Abaqus input file organized into parts that are instanced into an assembly. An overview of an Abaqus input file format is given in §8.7. Tracking particles on a UM geometry is a combination of two fields: particle transport and finite element analysis (FEA). One of the problems of merging two capabilities that have long, independent development paths is dealing with the distinct and sometimes contradictory terminology that has evolved with each. For example, the term "cell" is often used to generically denote the smallest building block in a UM geometry. However, an MCNP cell is quite different from a UM cell, which will be referred to as a "finite element". Note that element, mesh, part, elset, instance, and assembly terminology described in this section are used in FEA and Abaqus. The purpose of this section is to introduce the terminology and more information will be discussed later in this chapter. MCNP users who do not have an FEA background should consult the information in §8.7.

elements (or finite elements)	The smallest building blocks into which the mesh geometry is broken. Nodal data (i.e., node numbers and node coordinates) are used to define



(a) Tetrahedron (b) Pentahedron (c) Hexahedron

Figure 8.1: Finite element types (second-order elements with planar faces).

an element. The number of nodes per element depends on an element type. The element types are unstructured polyhedra with 4, 5, or 6 sides or faces, Fig. 8.1. First-order elements have nodes only at the vertices. When a face has 4 nodes, all 4 nodes are not guaranteed to lie in the same plane. This face has a degree of curvature and is known as bilinear. Thus, first-order elements may have either planar or bilinear faces. First-order elements with bilinear faces have trilinear volumes.

Second-order elements have nodes at the vertices and at the midpoints between the vertices. When 4 or more nodes define a face, they are not guaranteed to lie in the same plane. With 6 or 8 nodes defining a face, the degree of curvature can be greater than with 4 nodes and the faces are known as biquadratic. Thus, second-order elements may have either planar, bilinear, or biquadratic faces. Second-order elements with biquadratic faces have triquadratic volumes.

mesh	The collection of elements comprising the entire model. The mesh geometry can be structured or unstructured and is a representation of the geometry described by the solid model.
part	A part is defined by a collection of elements and nodes. In an Abaqus input file, it is possible to further subdivide a part into multiple sets of elements. This is done by grouping the elements in a part into sets where each element set is assigned a different name.
elsets	Elsets is short for element sets. An elset is a collection of elements that is associated with a specific tag, label, or name. The MCNP code requires that each part in an Abaqus input file must have an elset with "material" and "statistic" in its name. These elsets are referred to as a "material elsets" or "statistic elsets", respectively. This term is only used for a UM model defined in an Abaqus input file.
instance	An instance is a copy of a part used in constructing an assembly. Thus, each part may be used multiple times, giving rise to multiple instances of that part. This term is only used for a UM model defined in an Abaqus input file.

(a) parts (b) assembly

Figure 8.2: Constructing an assembly from parts.

assembly	An assembly consists of one or more instances. It can be viewed as a composite object. From this assembly, the MCNP code will create a global mesh model. This term is only defined for a UM model defined in an Abaqus input file.
pseudo-cell	In the MCNP input file, a pseudo-cell is a specialized cell definition, defined with a null or zero surface, that is used to associate normal MCNP cell features with the set of elements placed in the cell (e.g., a cell for an `F4` tally). The MCNP code uses instances and associated parts in an Abaqus input file to construct pseudo-cells. The elsets with distinct statistic elset names in the part are used to form the pseudo-cells when each part is instanced to form an assembly. For an HDF5 mesh input file, a pseudo-cell is created from a cell group.
background cell	An MCNP cell defined with a null surface. It serves as the background medium into which the UM model is placed.
mesh universe	This is the MCNP universe composed of the UM geometry (i.e., pseudo-cells) and the background cell. This universe may not contain any other lower universes or cells. The UM geometry must not be clipped by the boundaries of the fill cell that define this universe. This clipping requirement is not enforced by the code at this point, but is the user's responsibility to ensure that it doesn't occur. If clipping does occur, the user will experience lost particles in these regions of phase space. Particle tracking takes place on the pseudo-cells and the background cell.

8.3 Constructing an Unstructured Mesh Geometry

The MCNP UM calculations require two input file types: an MCNP input file and one or more UM geometry input files. The MCNP code can process a UM model formatted as either an Abaqus input file, or an HDF5 mesh input file. This section focuses on how to create a UM model formatted as an Abaqus input file. The first step in creating a UM model for use in the MCNP code is to create a part or series of parts. Each part can consist of a single element set of one homogeneous material or multiple element sets of different homogeneous materials. Once each part is created and meshed, element sets must be assigned in a part. The parts are then instanced to form an assembly, Fig. 8.2. The final step is to define material names. Other mesh generation tools may promote a different workflow, but the resulting file ultimately must meet the MCNP mesh format requirements. See §8.7 for more information on an Abaqus input file format.

⚠ Caution

The mesh geometry input files used on the `EMBED` card must have a filename that is all lowercase.

8.3.1 Naming Elsets and Materials

8.3.1.1 Elset Naming Guidance

The MCNP code requires that each part must has one or more elset keyword lines; see §8.7.2.4. Each elset in a part must be tagged with a name. The MCNP code requires the elset name to be in a specific format:

```
???AAA???%ZZZ
```

where:

`AAA`	Is one of the keywords: `material`, `statistic`, `tally`, `source` (1).
`ZZZ`	The set number following an underscore, "_", or a hyphen, "-", and `ZZZ` can be from 1 to 12 digits in length.
`???`	Any other characters or strings that may be used to describe the set, but should NOT repeat any of the keywords.
`%`	Indicates either a hyphen or an underscore.

Details:

1 The keywords `statistic` and `tally` are interchangeable; use one or the other, but not both to describe the elset in a part.

If a part is made up of more than one elset, the `ZZZ` number must be unique within the part. The `ZZZ` number must be unique within the assembly for the material elsets and material names in order for the legacy EEOUT file [DEP-53294] to be fully functional with auxiliary programs such as GMV [228]. This material elset number is assigned internally to the elements by the MCNP code and is output in the **`EEOUT`** file [§D.7] for each element. For best results, the user should make each `ZZZ` of the material element sets to be the same material number that appears on the MCNP material card.

As a convenience, it is possible to construct one elset that has multiple functions by specifying more than one keyword in the elset name. The naming format is:

```
???AAA???%???BBB???%???CCC???%ZZZ
```

where `AAA`, `BBB`, and `CCC` are the keywords defined above. Examples of legal elset names are material-tally-001, material_statistic_source_002, material_001, or statistic_001. Examples of illegal elset names are material_tally_statistic_001 or HEU-5.

In each part, the `material` and `statistic` (or `tally`) elset keywords are required, and the number of statistic elsets must be greater than or equal to the number of material elsets. The MCNP code will use the statistic elset numbers in each part to construct the corresponding pseudo-cells. If a part has only one statistic elset, then all elements in the part are assigned into one pseudo-cell. If a part has two statistic elsets and one material elset, then elements in this part are divided into two pseudo-cells with the same material number. The MCNP code checks that the number of statistic elsets must be greater or equal to the number of material

elsets in each part. If this requirement is not met, a fatal error is thrown. The intended use of the `statistic` (or `tally`) keyword is to collect individual elements in the elset into a pseudo-cell for the purpose of volume tallies (`F4`, `F6`, or `F7`). Basically, the elements in the same statistic elset share the same cell-like properties; hence coining of the term pseudo-cell.

Caution

The MCNP code requires its CSG cells to be associated with only one material and this must be upheld through the UM pseudo-cells. The MCNP code does not check the material and statistic elsets to ensure this requirement. Thus it is user's responsibility to make sure that only one material is assigned to each statistic elset.

All elements in each part must be assigned to material and statistic elsets. If not, a fatal error will be thrown.

The `source` elset keyword is optional in each part. This keyword should only be used to describe a volume source region in the UM model. The MCNP code will sample the source starting position (x, y, z) uniformly over the elements associated with the volume source; multiple volume sources are permitted, but see §5.8.1.3 on how to select among various volume sources. That is, a source element is selected from the source elset(s) with a probability proportional to the fractional volume of the source element in the total source volume. The source coordinates (x, y, z) are uniformly selected by rejection sampling over the selected element. No source biasing of position within a source elset (or pseudo-cell) is permitted with this capability. All other, non-positional fixed source (`SDEF`) options should work in conjunction with this capability, but extensive testing has not been performed. Volume source elsets may be defined but will not be used unless requested on the `SDEF` card.

8.3.1.2 Material Naming Guidance

The material names are independently created and are placed outside the assembly block near the end of the Abaqus input file. The material names must have the following format

```
???%ZZZ
```

where `ZZZ` is a material number that corresponds to those numbers used in the material elset and `???` are any other characters or strings, except the keywords reserved for elset naming. The `%` indicates either a hyphen or underscore. In other words, the material name can be a description like "BoronCarbide" followed by `ZZZ`; a valid material number in the MCNP input file.

Material names appear in the pseudo-cell cross reference table, which is written to the MCNP output file after the MCNP code processes the mesh description and creates the global tracking model. This table is intended to help users understand how the pseudo-cells should be specified. When searching for material names to insert into this table, the code tries to match the material number for the pseudo-cell to the material number in the material name. If that fails, the code assumes that the material names have been entered sequentially from 1 to the maximum number of material numbers and uses the pseudo-cell material number to select one of these. If both of these rules fail to produce a defined name, a message is inserted into the table to the effect that the material name does not exist.

The material properties may be assigned within the material name block in the Abaqus input file; however, the MCNP code does not use any material property presented in the Abaqus input file for the MCNP calculations. The isotopic (or mass) ratios are defined on a standard `M` card in the MCNP input. Likewise, the material densities are defined in the MCNP input on the pseudo-cells. The density information may be added to an Abaqus input file when an Abaqus input file is preprocessed to generate an MCNP UM input file.

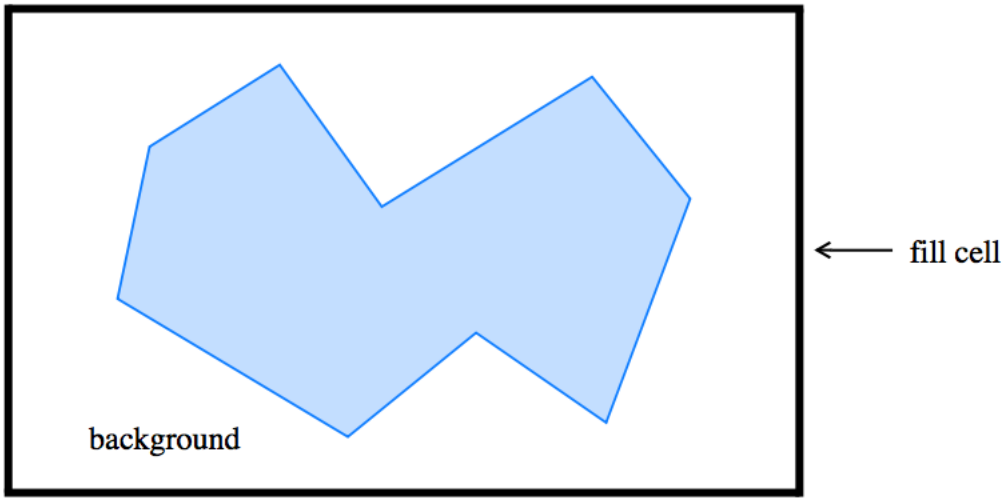


Figure 8.3: Example mesh universe with unstructured mesh.

8.3.2 Pseudo-Cell Creation

The MCNP code uses instances and statistic element sets to define the internal pseudo-cells. The pseudo-cells internally created by the code are numbered consecutively starting at 1, in the order the parts are instanced into the assembly. If part #2 is instanced ahead of part #1 in the Abaqus input file and each part has only one statistic elset, then an internal pseudo-cell #1 contains the elements in part #2 and an internal pseudo-cell #2 contains the elements in part #1. If only one part is instanced in an assembly and this part has 3 tally elsets (tally-10, tally-3, and tally-5), then an internal pseudo-cell #1 contains the elements in the tally-3 elset, an internal pseudo-cell #2 contains the elements in the tally-5 elset, and an internal pseudo-cell #3 contains the elements in tally-10 elsets. The internal pseudo-cells are matched to the pseudo-cell numbers in an MCNP input by the `MATCELL` keyword on the `EMBED` card, where the first and second entries of the `MATCELL` keyword are respectively the internal pseudo-cell numbers created from an Abaqus input file and the pseudo-cell numbers in an MCNP input file. The internal pseudo-cells are also known as (unstructured) mesh cells. The user should examine the pseudo-cell cross-reference table in the MCNP output file to make sure that the global model built by the MCNP code is the model that the user intends to study.

8.3.3 Mesh Universe

A simplified MCNP hybrid geometry arrangement with a UM geometry model embedded in the CSG model (i.e., mesh universe) is shown in Fig. 8.3. The mesh universe is everything contained within the fill cell where the fill cell's outer boundary is the heavy black rectangle. Note, "fill cell" means the traditional MCNP cell card that contains the "`FILL`" parameter and a collection of defined surfaces that crop the universe which it contains. These surfaces that define the fill cell must not intersect the pseudo-cells.

A background cell is needed to make the mesh universe infinite in extent and is the region outside of the blue unstructured mesh region in Fig. 8.3; it is cropped by the surface(s) that defines the fill cell. Specifying the background cell in the MCNP input file is a 2-step process: First, a background cell defined by the null surface must be specified in the MCNP cell block. Second, the background keyword must appear on the `EMBED` data card. The material specified for the background cell is also the material used in all gaps within the UM model.

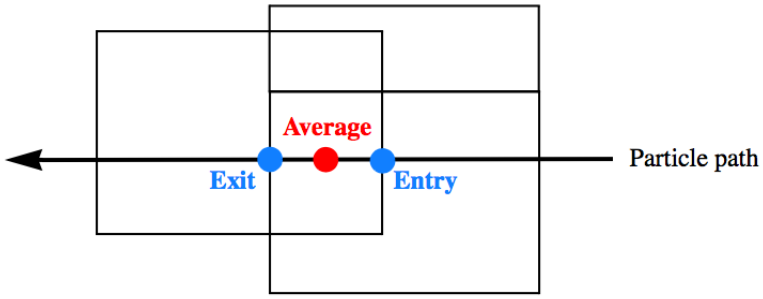


Figure 8.4: Illustration of the three critical points for the overlap models.

8.3.4 Overlaps

One of the initial requirements for the MCNP UM implementation was to permit multiple, non-contiguous, meshed parts instead of requiring one contiguous mesh. This naturally leads to the possibility of overlapping parts, particularly when two parts attempt to share a curved surface. If it is crucial to the model that the integrity of any curved surface be maintained, the user should then consider merging the two separate parts into a single part, using second-order elements, and/or refining the mesh. Significant overlapping regions are never a good idea. Users should never rely on any of the following models to correctly produce the same results as a model where the boundary between two regions is defined so that there is no overlap.

The MCNP code can accommodate a small amount of overlap in one of several ways. For the initial implementation, there was no correction for tracking through overlapping elements. A particle tracks in an element until it finds a definite transition point in phase space (i.e., another element, gap, or background cell). Of the three overlap models currently in place (see the `OVERLAP` keyword on the `EMBED` card and Fig. 8.4), the initial implementation is known as the `EXIT` model, meaning that in an overlap situation, the exit point of the overlap is used and a path-length is accumulated by ignoring an overlap region.

The second overlap model, `ENTRY`, is the one that uses the entry point of the overlap in an overlap situation and the results are accumulated accordingly. If the entry point is behind the particle's current position, the current position is used; the particle never moves backwards. The third and last overlap model is called `AVERAGE` and results in averaging the entry and exit points in an attempt to find the midpoint of the overlap in the direction the particle is tracking; the particle's path length in the overlap is then divided between the two parts instead of being assigned to one or the other.

Although the code defaults to the `EXIT` model, ultimately the choice of which model to use is left to the user. If both parts are important and the particle flux through this region is fairly isotropic, the `AVERAGE` model is probably the best choice. If the flux is somewhat more directional and one part is deemed more significant than the other, a better choice might be `ENTRY` or `EXIT`, depending on the problem. The user also has the ability to select the model to use by the part, with the decision based upon the current part in which the particle resides. For example, if the particle is currently in a part that specifies the `EXIT` model and the part into which it will travel specifies the `ENTRY` model, the `EXIT` model is used.

Note that testing has been performed with the `EXIT` model but not the other two.

8.4 Output: Elemental Edits

To obtain results at the element level, a path length estimate of the flux is accumulated as particles track from one element face to another, Fig. 8.5.

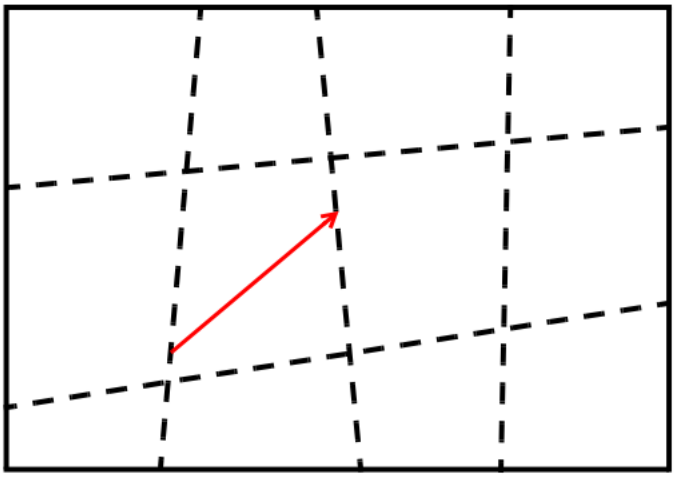


Figure 8.5: Illustration of element-to-element tracking on a 12-element part.

To differentiate the mesh results from the traditional MCNP tally treatment, those results accumulated on the unstructured mesh are referred to as "elemental edits." There is no current intention to duplicate all of the tally features with the edits. The elemental edits, along with a generic description of the unstructured mesh model, are output in a special file known as the **EEOUT** (Elemental Edit OUTput) file. See §D.7 and DEP-53294 for a description of the legacy EEOUT file and §D.6 for a description of the HDF5 EEOUT file.

At this time, relative errors are optional for the results on any element. Specifying errors can result in large **EEOUT** files. If the traditional MCNP statistical analysis (e.g., tally fluctuation chart, empirical history score pdf) is desired for the results, set up a tally for an appropriate pseudo-cell. More information on estimation of the Monte Carlo precision can be found in §2.6.4.

8.5 MCNP Geometry Plotter

Plotting of the UM geometry with the MCNP plotter is very limited. It is only possible to produce shaded plots of the pseudo-cells by material, atom density, or mass density so the user may see that the UM geometry is positioned correctly relative to the CSG cells. No cell outlines or UM lines are possible. Labels may appear but may not be correct. See Figs. 8.6–8.10 for several examples. Overlaps may make regions appear distorted, Fig. 8.9. Gaps may give rise to extended regions of the background material, Fig. 8.10.

Caution should be exercised with large UM files. While the plotter should be able to plot large UM geometries, it may take a long time to build the model. The MCNP plotter is an old technology and thus cannot be used to view a UM model in 3-dimensions. Many modern software packages can be utilized to view UM models in 3-dimensions. A UM model formatted as an Abaqus input file can be visualized by the Abaqus and Cubit codes; Cubit is a mesh generation tool kit developed by Sandia National Laboratories (`https://cubit.sandia.gov`). A UM model formatted as an HDF5 mesh file can be visualized by modern visualization software packages such as ParaView (`https://www.paraview.org`) and VisIt (`https://visit-dav.github.io/visit-website/index.html`).

8.6 Limitations and Restrictions

The UM capability is currently not fully integrated with all of the pre-existing MCNP features. This section highlights known limitations and restrictions of the MCNP UM feature.

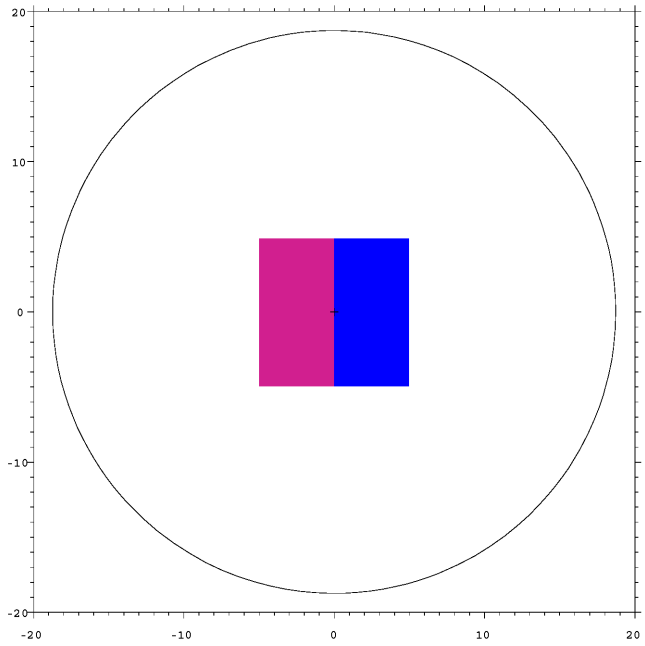


Figure 8.6: Pseudo-cells shaded by material in the mesh universe.

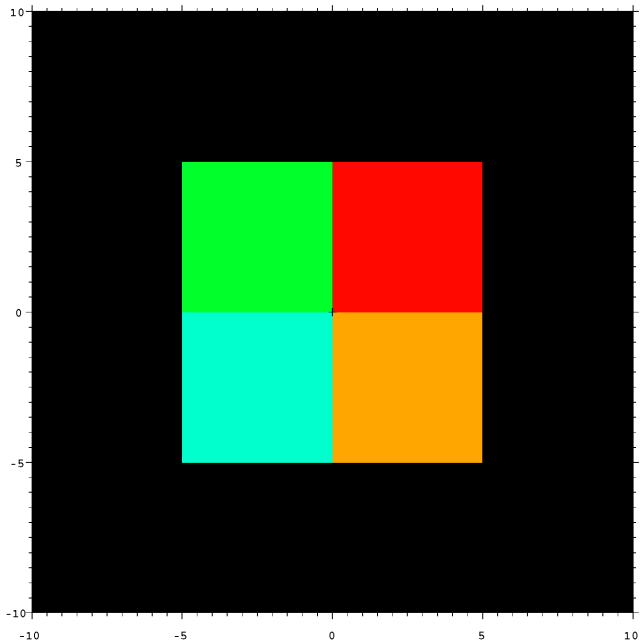


Figure 8.7: Pseudo-cells shaded by material density.

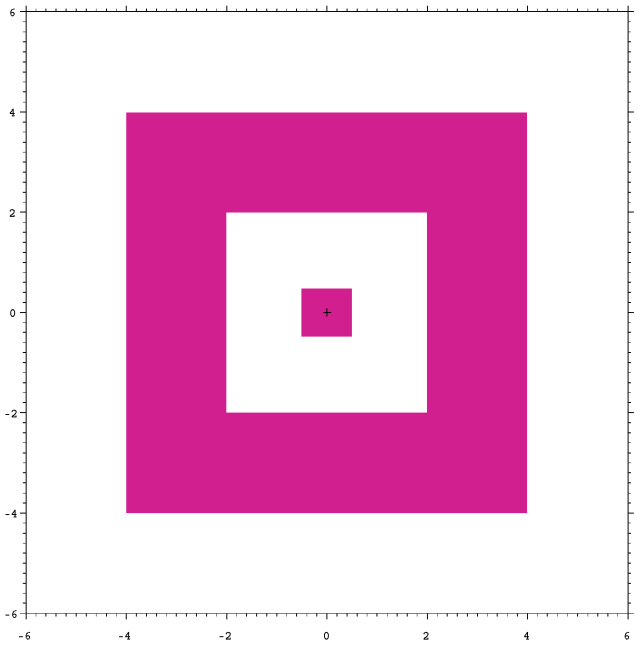


Figure 8.8: Model demonstrating correct plotting of a gap.

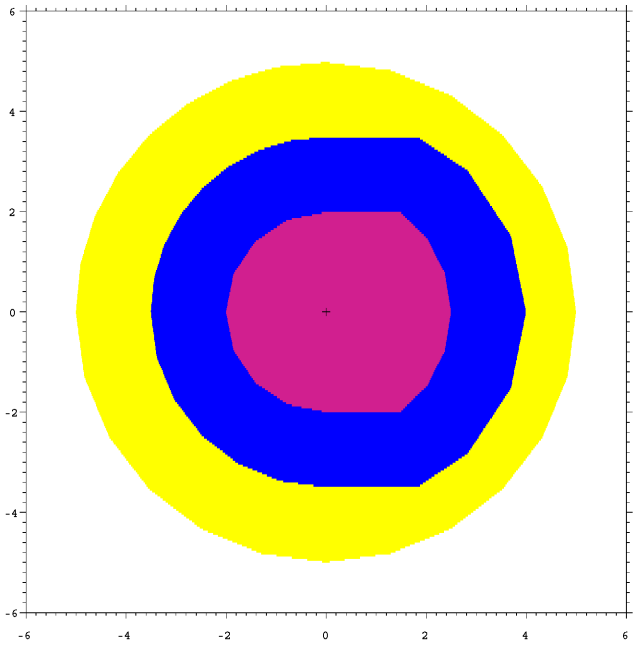


Figure 8.9: Model demonstrating overlaps.

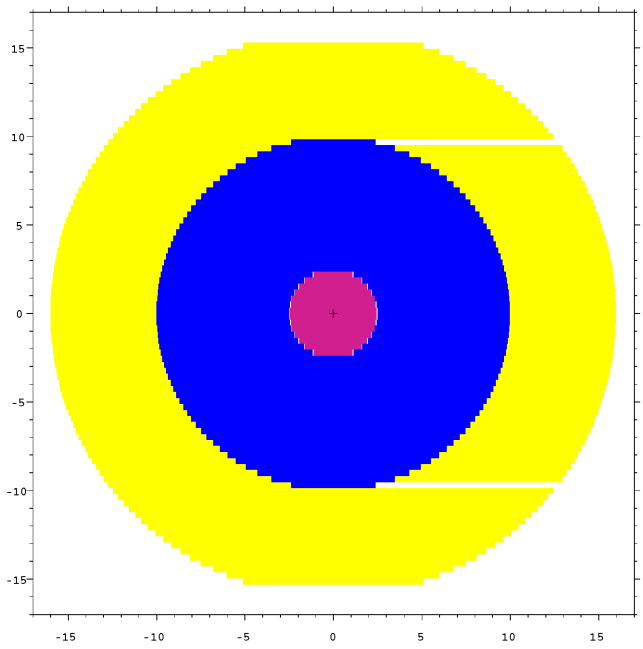


Figure 8.10: Model demonstrating gaps.

- Limited to neutrons, photons, electrons with the default physics options, protons, and charged particles heavier than protons. Testing for other particle transport problems, except neutrons and photons, is limited.
- Cannot be used with magnetic fields.
- A UM model can not be placed inside a lattice.
- A universe can not be placed within a mesh universe.
- CSG surfaces must not clip or intersect the UM model.
- The MCNP plotter may be used to plot limited aspects of the UM geometry for the purpose of seeing its position in the hybrid geometry.
- Mesh surfaces can not be used for surface sources; normal surface source reads and writes have under gone limited testing with the UM feature and are not guaranteed to work with it.
- Reflecting and periodic boundary conditions are not guaranteed to work with the pseudo-cells but should work with CSG cells/surfaces that have these conditions.
- Source particles may not be started in mesh gaps.
- Surface tallies are not permitted in the background cell and pseudo-cells, but can still be used with CSG surfaces.
- Only pentahedra and hexahedra elements may appear together in a part; otherwise a part must contain only a single mesh type.
- Overlapping parts must not be severe; any single element may not be wholly contained within another element.
- Testing for multiple UM models embedded into multiple CSG cells is very limited.
- Forced collision (FCL) variance reduction cannot be used with the UM feature.

- Testing for embedding both UM and LNK3DNT geometries in a problem is very limited.
- Splitting particles as they enter and exit pseudo-cells as a result of weight windows or pseudo-cell importances may lead to potentially silent wrong answers with a UM geometry or, more clearly, seemingly unrelated issues, such as the code reporting negative emission energy following certain collisions (see the `WWP` and `IMP` entries for more information).
- It is unknown whether a `PTRAC` file will contain all surface related information.
- Not all combinations of parameters associated with the `SDEF` card have been tested in conjunction with the UM volume sources.

8.7 Abaqus-formatted Mesh Input File

8.7.1 Creating an Abaqus Input File

The Abaqus input files needed for MCNP UM calculations must have the correct Abaqus syntax and meet the additional requirements by the MCNP code. A mesh model is a representation of a solid geometry model, and several software packages can generate a UM model formatted as an Abaqus input file. Typically, a computer aided design (CAD) software is used to construct a solid geometry model, which is then imported into a mesh generation software to prepare and mesh the solid model. Finally, the mesh model is exported into a file formatted as an Abaqus input file. When creating Abaqus input files for MCNP UM calculations, users should be aware of these two points:

1. Depending upon the purpose of the model creation and who is generating it, there may be extraneous information in the input file that could cause problems with the MCNP input file parser. What is shown in this section is the basic information that the MCNP code needs. For best results only include the data types discussed here.
2. Other meshing tools may be used to export an Abaqus input file. It is the user's responsibility to ensure that the Abaqus input files generated by other meshing tools meet the requirements outlined in this section.

8.7.2 Abaqus Input File Format

An Abaqus input file is an ASCII file that contains a series of lines. Each line in the file cannot exceed 256 characters. If required data cannot be fit on a single 256-character line, then a comma is placed at the end of the line to indicate that the next line is a continuation line. Three types of input lines are used in an Abaqus input file: comment lines, keyword lines, and data lines.

Comments	Begin with double asterisks in columns 1 and 2 (**). The comment lines are not used by the MCNP code.
Keywords	Must begin with an asterisk (*) in column 1. The keyword lines may have parameters that appear as words or phrases separated by commas. The keyword must be followed by a comma if it has parameters. The parameters in a keyword line can stand alone or have values. If a parameter has a value, an equal sign and a double quotation mark are respectively used to assign and group the value. Some keywords occur in pairs, meaning that there is a keyword that starts a block of data and another keyword that ends a block of data. Other keywords are singular in that they start a block of data and an unrelated keyword or comment ends the block. Most keyword lines require one or more data lines. If the data lines are required, they must immediately follow the keyword line.

Data lines	Are generally used to provide entry values for the associated keyword options. Data lines have no special characters preceding them and data items are separated by commas. If there is only one item on a data line, it must be followed by a comma.

The MCNP code reads and processes Abaqus input files that make use of part and assembly definitions. An Abaqus UM model is created by defining parts and then assembling instances of each part. Each part can be used (instanced) one or more times, where each instance has its own position within the assembly. Only one assembly can be defined in a model. A component defined within a part, instance, or the assembly is local to that part, instance, or the assembly. A part definition must appear outside the assembly definition. Multiple parts can be defined in a model and each part must have a unique name. An instance definition must appear within the assembly definition, where each instance must have a unique name and refer to a part name defined in the part data block. Data lines may be used to position the instance within the assembly. These positioning data lines include a translation and rotation for the instance relative to the origin of the assembly coordinate system. Other components must be categorized and fall within the proper level: part, assembly, instance, or model. Material definitions are model-level data. The part-level data definitions required by the MCNP code are node, element, and element sets. All part definition blocks must be defined before the assembly material definition blocks. The assembly material blocks may appear in any order after the part blocks. Greater detail on the Abaqus file format can be found in the Abaqus documentation released with the Abaqus software package (`https://www.3ds.com/products-services/simulia/products/abaqus/`).

The MCNP code reads and processes the following keywords and associated data lines: `*Heading`, `*Part` and `*End Part`, `*Node`, `*Element`, `*Elset`, `*Assembly` and `*End Assembly`, `*Instance` and `*End Instance`, `*Material`, and `*Density`. The "`*Heading`" keyword is optional. The sample input file in Section 8.7.2.10 is color-coded for ease of reading. The keywords of interest to the unstructured mesh parser are shown in blue. Several special tags, also of interest to the parser, are shown in red and are discussed below. The model present in this sample file is simple and consists of one part that has been instanced four times in the assembly; this is discussed in more detail in Section 8.7.2.7. Each of the keywords of interest to the unstructured mesh parser are discussed in the order that they usually appear in the sample input file. In the following, keywords are shown in mixed case, but the input parser is case-insensitive.

8.7.2.1 Part

The "`*Part`" keyword signifies the beginning of the information for a particular part. The required parameter is the name after the "`name=`" characters on the keyword line. The label of the name parameter must be unique since it will be used to refer to the part. The UM library parser retrieves everything after the equals sign up to and including 256 characters in the name. This name is used by the UM library in locating the correct part when it is instanced in the assembly. The part name is also used when the UM library outputs information about the mesh model. Do not use any of the element set keywords (§8.3.1.1) in the name of the part.

8.7.2.2 Node

The "`*Node`" keyword appears in the part-level block and signifies the beginning of the node data specific to the part. The MCNP code does not use other parameters (such as `input`, `nset`, `system`) on this keyword line. The "`*Node`" keyword must have data lines follow that specify the node numbers and their coordinates. Each data line contains four numbers: the first entry is a positive integer and the other entries are three real numbers. The positive integer is the node number and the three real numbers are the x-, y-, and z-locations of the given node.

8.7.2.3 Element

The "*Element" keyword appears in the part-level block and marks the beginning of the element connectivity data. The required parameter is "type=". Other parameters (such as elset, input, etc.) should not be in the "*Element" keyword line since they are not used by the MCNP code. The parameter value on this keyword line after "type=" is a description of the type of elements in this part. The element type codes appearing on this line that the UM library can handle are presented in Table 8.1. The MCNP code treats a continuum shell element as a linear hexahedron; it is included as a convenience for users that must rely on the SC8 element type. In the example input file in Section 8.7.2.10, the type code is presented in red-lettered characters on the "*Element" keyword line.

Table 8.1: Element Type Codes

Element Type	Type Code
First-order tetrahedra	C3D4
First-order pentahedra	C3D6
First-order hexahedra	C3D8
Second-order tetrahedra	C3D10
Second-order pentahedra	C3D15
Second-order hexahedra	C3D20
Continuum shell element	SC8

Each line following this keyword contains a variable number of integers depending upon the number of nodes that define the element. In the type code given in Table 8.1, the number of nodes for a particular element type appears as the number following the tag "D" or "SC". The first integer on the data line is the element number; the remainder are the node numbers that define the element. The exception to this is second order hexahedra, where two lines are required for each element. For these, the first line contains the element number plus 15 node numbers; the second line contains the remaining 5 node numbers and is generally indented.

The MCNP code can handle a part with two mixed element types. When two element types appear in a part, Abaqus places two "*Element" keyword sets in the "*Part" block. Currently, the UM library can only handle mixed element parts containing pentahedra and hexahedra elements or continuum shell and hexahedra elements. If tetrahedra elements are needed in the model, a tetrahedra part must not contains other element types. Other element type codes are used by the Abaqus code; it is the user's responsibility to ensure use of the type codes from Table 8.1 to specify the element types. The "*Element" keyword and data lines must be defined after the "*Node" block.

8.7.2.4 Element Set

In Abaqus parlance, element sets are referred to as "elsets" and the "*Elset" keyword signifies the beginning of the elset data. The elset mechanism permits the grouping of elements in order to assign various properties. The MCNP code uses the elset definition blocks defined in the part-level data. The elset keyword and data lines must be defined after the "*Element" block(s). The elset parameter is required for this keyword line; the parameter value after "elset=" is the name of the element set to which the elements will be assigned. The "elset=" parameter must be the first parameter on the keyword line. Each part may have more than one elset and each elset name must be unique. At least material and tally elsets (see Section 8.3.1.1) must be defined in each part. These elset names are easy to find in the example input file in Section 8.7.2.10; they are in the red-lettered characters after the "*Elset" keyword that is in a blue font. The other parameters allowed in the "*Elset" keyword line is generate. Other parameters (such as instance, unsorted, etc.) should not be used on this keyword line.

The first elset is the material elset and is required. All of the elements in a part must be assigned a material number. The name or tag for this elset must contain the word "`material`" and the material number. The material number must be the last part of the tag and it must be separated from the rest of the tag by an underscore or hyphen. In addition to the material elset tag presented in the example in Section 8.7.2.10, the following tag is also acceptable:

```
Set-my_material_uranium_02
```

Note that any number of characters can appear between the word "material" and the material number, but the total length of the line containing the keyword and the tag is limited to 256.

The second elset is the statistic (or tally) elset. This elset is also required. The name or tag for this elset must contain the word "`statistic`" or "`tally`" (but not both) and the statistic set number. The same rules and conventions apply to this elset tag as for material elsets. All elements in a statistic elset must have the same material number; there is no mixing of materials in the statistic set. The UM library will enforce this.

For each of these keyword types, the data lines following them may be one of two forms. The first of which is just an integer list of element numbers, on the order of 16 integers or fewer per line. The second form is in compact notation where the word "`generate`" appears on the "`*Elset`" keyword line and the data line consists of 3 integers. The first integer is the starting element number. The second integer is the ending element number. The third integer is the stride from the starting to the ending element numbers. For example, to specify all of the odd element numbers from 1 to 27, use the following:

```
1, 27, 2
```

The MCNP code uses these two elsets (material and statistic) and instance data to define the internal pseudo-cells that must be mapped back to the pseudo-cell cells in the MCNP input file; this mapping is done with the `MATCELL` keyword on an `EMBED` card. The MCNP code outputs a "Pseudo-Cell Cross-Reference" table that shows how the internal pseudo-cell numbers match the pseudo-cell numbers defined in the MCNP input file, the instance numbers, the part numbers, the material numbers, and the material names.

8.7.2.5 End Part

The "`*End Part`" keyword marks the end of a part's input. Another part description may follow, in which case there will be another "`*Part`" keyword to signify its beginning, or the assembly description may follow.

8.7.2.6 Assembly

The "`*Assembly`" keyword appears after all of the parts are defined. A look at the sample file shows that an assembly name appears after this keyword much like what appeared for the part. The UM library does not use the assembly name; it only uses the "`*Assembly`" keyword to determine the end of the part data

There is also an "`*End Assembly`" keyword that signifies the end of a particular assembly. Between these two keywords is the important information that the UM library needs in order to construct the mesh model from the parts.

8.7.2.7 Instance

Appearing in the assembly-level block are the "`*Instance`" keywords. The numbers that appear here correspond to the parts used to form the Assembly for each instance. Each part may be instanced many times. There are two parameters appearing on the "`*Instance`" keyword line: "`name=`" and "`part=`". The parameter value after "`name=`" is the name of the instance and, unless changed by the user in the meshing tool, is just the part name appended with an instance number. The parameter value after "`part=`" is the part name as one of those used with the "`*Part`" keyword. The "`name`" parameter must be defined before the "`part`" parameter. The UM library uses this "`part`" parameter name to match with the "`*Part`" keyword name in order to locate the right one to use.

The "`*End Instance`" keyword marks the end of the information block for a particular instance. From the example in Section 8.7.2.10, there are four instances of the same part. The last instance in this example has no additional lines between the "`*Instance`" and "`*End Instance`" keyword lines while the other three have one or two data lines present that describe the translation or rotation of the part as it was instanced into the assembly.

The first data line appearing between the keywords is the translation information. The three real values given here are the values of the translation applied in the x-, y-, and z-directions, respectively. These translation values have the same unit used to define node positions in parts.

If the part is rotated as it is instanced into the assembly, two lines appear between the instance keyword lines. The first line is the translation information as discussed previously. If there is a pure rotation the values for the three real numbers on the translation line are all zero. If there is both a translation and rotation, the translation is applied before the rotation.

There are seven real numbers that appear on the rotation line. The first six real numbers define an axis of rotation. The first three numbers are the x-, y-, and z-locations of the first point that defines the axis. The second three numbers are the x-, y-, and z-locations of the second point that defines the axis. The seventh number is the angle of rotation in degrees about the axis.

In the sample input file, the first and third instances have just a translation while the second instance has a rotation but no translation. The fourth instance is neither translated or rotated.

8.7.2.8 Material

The "`*Material`" keyword has one parameter, which is a material name. The parameter value after "`name=`" is the name of material which must end with a number. The UM library parser retrieves everything after the equals sign up to and including 256 characters in the name and extract the ending number. This ending material number is then used to match with an elset material number in a part. See Section 8.3.1 for the recommendations in naming materials. The material properties present in an Abaqus input file are not used for MCNP calculations. The MCNP code uses the material properties defined in the MCNP input file.

8.7.2.9 Density

The "`*Density`" keyword is the material sub-keyword of interest to the UM library. The data line following this keyword has one data item. This keyword and associated value are only used by the **`um_pre_op`** program, [§E.12, DEP-53422]. The "`*Density`" keyword and data line are not required by the UM library. If this keyword and data line are not in an Abaqus input file, the MCNP skeleton input file written by the **`um_pre_op`** program will be zero density (void) and the user must manually edit the densities in the MCNP input file. Mass densities used in Abaqus calculations must be positive, but mass densities in an MCNP file must be

entered as negative numbers. The **um_pre_op** writes the density values read from an Abaqus input file into an MCNP input file without adjusting the sign, and it is user's responsibility to edit the mass densities in an MCNP input if they are positive in an Abaqus input file. A negative density value is not a correct Abaqus input format, but the UM library will read the negative density value without any warning. The density values in an Abaqus input file are not used in MCNP calculations, and it is not required that the material density in the Abaqus input file be the same as the density in the MCNP input file.

8.7.2.10 Example Abaqus Input File

```
*Heading
an example of an Abaqus input file
** Job name: job_block_demo_01 Model name: Model-1
** Generated by: Abaqus/CAE 6.10-1
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Part-block_01
*Node
      1,           4.,           4.,           4.
      2,           4.,           2.,           4.
      3,           4.,           0.,           4.
      4,           4.,           4.,           2.
      5,           4.,           2.,           2.
      6,           4.,           0.,           2.
      7,           4.,           4.,           0.
      8,           4.,           2.,           0.
      9,           4.,           0.,           0.
     10,           2.,           4.,           4.
     11,           2.,           2.,           4.
     12,           2.,           0.,           4.
     13,           2.,           4.,           2.
     14,           2.,           2.,           2.
     15,           2.,           0.,           2.
     16,           2.,           4.,           0.
     17,           2.,           2.,           0.
     18,           2.,           0.,           0.
     19,           0.,           4.,           4.
     20,           0.,           2.,           4.
     21,           0.,           0.,           4.
     22,           0.,           4.,           2.
     23,           0.,           2.,           2.
     24,           0.,           0.,           2.
     25,           0.,           4.,           0.
     26,           0.,           2.,           0.
     27,           0.,           0.,           0.
*Element, type=C3D8
1, 10, 11, 14, 13,  1,  2,  5,  4
2, 11, 12, 15, 14,  2,  3,  6,  5
3, 13, 14, 17, 16,  4,  5,  8,  7
4, 14, 15, 18, 17,  5,  6,  9,  8
5, 19, 20, 23, 22, 10, 11, 14, 13
6, 20, 21, 24, 23, 11, 12, 15, 14
7, 22, 23, 26, 25, 13, 14, 17, 16
8, 23, 24, 27, 26, 14, 15, 18, 17
```

```
*Nset, nset=Set-material_01, generate
  1,  27,   1
*Elset, elset=Set-material_01, generate
 1,  8,  1
*Nset, nset=Set-statistic_01, generate
  1,  27,   1
*Elset, elset=Set-statistic_01, generate
 1,  8,  1
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Part-block_01-1, part=Part-block_01
          4.,           0.,           0.
*End Instance
**
*Instance, name=Part-block_01-2, part=Part-block_01
          0.,           0.,           0.
          0.,           0.,           0.,           0.,           1.,    0.,          90.
*End Instance
**
*Instance, name=Part-block_01-3, part=Part-block_01
          4.,           0.,          -4.
*End Instance
**
*Instance, name=Part-block_01-4, part=Part-block_01
*End Instance
**
*End Assembly
**
** MATERIALS
**
*Material, name=Material-part1_01
*Density
18.74,
```

8.8 HDF5-based Mesh Input and Output Files

In addition to the Abaqus-formatted mesh input file described in §8.7, an HDF5-formatted [340] mesh input file can be used. The format of this file is described in §D.6. The HDF5-formatted UM output can also be requested and has the format described in §D.6. Advantages provided by HDF5 include hierarchical organization, binary representation of data with compression, and many options for software and programming language interoperability.

The legacy **EEOUT** file [§D.7, DEP-53294] provides comprehensive output for MCNP UM calculations, which includes providing the ability to restart calculations. However, the **EEOUT** file uses a non-standard mesh file format, so it requires custom post-processing parsers that often rely heavily on regular expressions. Accordingly, it can be burdensome for end users to convert the **EEOUT** file into a format for downstream analysis and/or visualization. The **GMV** file [§8.9.1] also prevents easy interrogation except by select visualization applications or custom post-processing applications. Because of the **GMV** format, post-processing is usually limited to serial execution.

As such, an output file option is available that produces an HDF5 binary file containing the UM geometry and edit results and an accompanying XDMF version 2 file [324, 325] that permits direct visualization in applications such as ParaView [326] and VisIt [327]. The HDF5 file itself can be processed in parallel. In this way, the XDMF file can be immediately used to visualize UM results and the HDF5 file can be interrogated and/or manipulated to enable downstream analysis using standard HDF5 utilities, which are available in a variety of programming languages (such as with Python's **h5py** package).

When HDF5 output is enabled, data necessary for restart calculations are also written (and used, as applicable). For the MCNP code, these data are written to the **/restart/unstructured_mesh** group in the HDF5-format runtape file [§D.2]. As such, MCNP calculations can be performed using the legacy **EEOUT** output [DEP-53294] or HDF5 output for results and restart purposes. This provides a means to compare behavior and permits deprecating the legacy **EEOUT** output. Note that during comparison with ASCII **EEOUT** files, some differences are expected because the ASCII output stores only five decimal places.

To enable the HDF5 output, the `hdf5file` parameter on the `EMBED` card specifies the name for (and enables writing) a binary HDF5 file containing UM geometry structure and data as well as an accompanying XDMF version 2 file. The `hdf5file` option and the MCNP input command line option can be used to create the HDF5/XDMF files containing the mesh model.

The accompanying XDMF version 2 file is an ASCII XML file. Because of its standard format, the XDMF file format is not described here. The XDMF file does not contain the mesh model data nor the edit results. This XDMF file points at the appropriate data in the HDF5 binary file so that the mesh model and edit output can be visualized. Note that the XDMF and HDF5 files must remain together in the same directory to permit a utility reading the XDMF file to find the HDF5 file. If a Python script is developed to post-process the mesh data and edit results, then only the HDF5 file is needed since the mesh data and edits are contained in this file.

8.9 Other Files

8.9.1 GMV File

Deprecation Notice **DEP-53519**

The GMV file output capability using the `EMBED` card is deprecated. Because of the new HDF5-formatted output file that can be trivially visualized and/or post-processed, this output file format is no longer necessary. As such, the GMV keyword on the `EMBED` card is deprecated.

Often times it is beneficial to have an independent and easy to use program for mesh geometry visualization. The General Mesh Viewer, GMV, program [228] is such a program. For this reason, it is possible to generate a GMV input file (see embed card, parameter `gmvfile`). Note that if during model creation in the CAE tool, the material elsets don't have unique numbers, it will be difficult to differentiate parts in GMV. That is, if each part has one material and the number assigned to that material is the same one in all of the parts, then all elements in GMV will have the same color. Also, GMV limits material names to 8 characters.

This GMV file capability is primarily for LANL use.

8.9.2 MCNPUM File

Deprecation Notice **DEP-53424**

The MCNPUM file (as both input and output) specified on the EMBED card for unstructured mesh (UM) calculations is deprecated. Because of algorithmic improvements during UM input processing, the need for this file to accelerate that process is no longer necessary. Furthermore, the historic guidance that suggests limiting components to 30,000–50,000 elements is no longer necessary.

As such, the MCNPUMFILE keyword on the EMBED card and the mcnpum option for the MESHGEO keyword are also deprecated.

The Abaqus input file contains some basic information about the unstructured mesh, but does not contain everything that MCNP6 needs. Once MCNP6 reads this file, it uses the Abaqus data to generate other information that it needs in its tracking routines, such as nearest neighbor lists. Even with the parallel input processing, discussed elsewhere in this document, significant computer time can be required to regenerate this data and create other internal data structures for every MCNP6 calculation that uses the Abaqus unstructured mesh file.

The **MCNPUM** file-type [341] was created to contain all of the unstructured mesh data structures that MCNP6 needs, thus eliminating the need to "input process" the Abaqus input file every time the code is run, including restarted calculations. MCNP6 can generate this file (primarily after processing the Abaqus input file) by simply including the MCNPUMFILE option on the EMBED card. MCNP6 can use the **MCNPUM** file when MESHGEO=MCNPUM on the EMBED card.

The **um_convert** utility [§E.10, DEP-53421] is a highly parallelized program that can convert the Abaqus mesh input file to the **MCNPUM** file type. This file type is highly recommended when a complex geometry will be used more than once.

Part III

MCNP Primers

Chapter 9

Introduction

This part of the MCNP manual will provide a collection of topical primers similar to the MCNP Source Primer [342]. Until that time, the Examples chapter of the MCNP6.2 manual [Chapter 4 of 2] is relocated here.

Chapter 10

Examples

Instructive examples of several topics are included in this chapter. Some of the examples are simplistic while others illustrate more complex features of the MCNP6 code. They should be studied in conjunction with the theory, instructions, and previous examples provided in Chapters 3, 4, and 5 of this manual.

Following the simple geometry specification examples are related geometry examples that exercise coordinate transformations, repeated structure and lattice geometries, and embedded meshes. After the geometry-related examples are those related to tally options, including the FM, FMESH, FS, and FT cards as well as the **TALLYX** subroutine for user-defined tallies using the FU card. Next are source specification examples for the generalized source, beam sources, and a burnup case followed by example **SOURCE** and **SRCDX** subroutines for point detectors and/or DXTRAN spheres. Finally, a materials example of table and model-data mix-and-match and a physics model example complete the section.

10.1 Geometry Examples

The geometry discussions in Chapters 3 and 4 must be understood before studying the following examples. The concept of combining regions of space bounded by surfaces to make a cell must be fully appreciated; the following examples should help solidify this concept. The use of macrobodies will simplify many geometry definition situations.

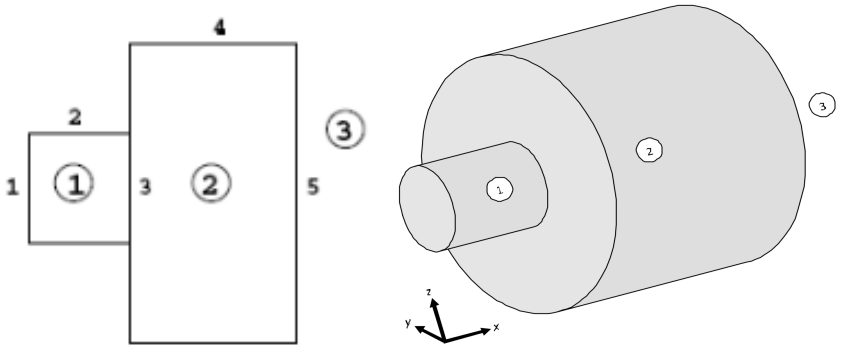
10.1.1 Geometry Specification

Several examples of the union and complement operators follow. These should help you better understand how cells are defined. In the illustrations, cell numbers will be circled; surface numbers will not be circled but will appear next to the surface they represent. For simplicity, all cells are void of material.

The next several examples become progressively more difficult and usually take advantage of what you learned in the preceding ones. Remember that unless altered by parentheses, the hierarchy of operations is that intersections are performed first and then unions.

10.1.1.1 Example 1

Figure 10.1, surfaces 2 and 4 are cylinders and the others are planes with their positive sides to the right. The figure includes a perspective view to make it clearer what is being defined. The surfaces used in this example are:



(a) Elevation view. (b) Perspective view.

Figure 10.1: Example 1 sample geometry—two Stacked Cylinders: The XZ cross section (at left) shows the three cells and defining surface indices.

```
1  PX  0  $ plane perpendicular to the X axis at x=0
2  CX  2  $ cylinder on the X axis of radius 2
3  PX  2  $ plane perpendicular to the X axis at x=2
4  CX  3  $ cylinder on the X axis of radius 3
5  PX  6  $ plane perpendicular to the X axis at x=6
```

Cells 1 and 2 are easy to specify:

```
1  0 -2 1 -3 $ inside cylinder 2, right of plane 1, left of plane 3
2  0 -4 3 -5 $ inside cylinder 4, right of plane 3, left of plane 5
```

Cell 3 is more complex: There are multiple ways it can be defined. Here are some definitions of cell 3, each of which is described in more detail:

```
3  0   (2  3): 1:4:5   $ parentheses used for clarity; not required
3  0   4: 1:5:(2  3)   $ parentheses not required
3  0   ( 1:2) (-3:4):5 $ parentheses are required for correctness
3  0   #1 #2           $ everything that is "not" cell 1 or 2
```

It may be helpful to refer to Fig. 2.3 and its explanation. Remember that a union adds regions and an intersection gives you only the areas that overlap or are common to both regions. In addition, intersections take precedence over unions. Regions can be added together more than once—or duplicated—with the union operator.

Let us arbitrarily start with the definition of cell 3 at cylindrical surface 2. The expression 2 -3 defines the following region: everything in the world outside surface 2 intersected with everything to the left of plane surface 3. This region is hatched in Fig. 10.2. Let us examine in detail how Fig. 10.2 was derived. First look at each region separately. The area with a positive sense with respect to surface 2 is shown in Fig. 10.3. It includes everything outside surface 2 extending to infinity in all directions. The area with negative sense with respect to surface 2 is undefined so far. The area with negative sense with respect to surface 3 is shown in Fig. 10.4. It includes everything to the left of surface 3 extending to infinity, or half the universe. Recall

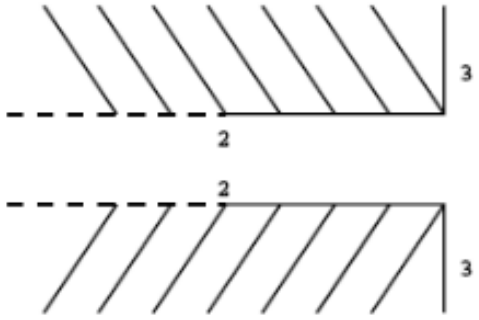


Figure 10.2: Outside (i.e., positive sense) of cylindrical surface 2 intersected with region to left (i.e., negative sense) of plane surface 3.

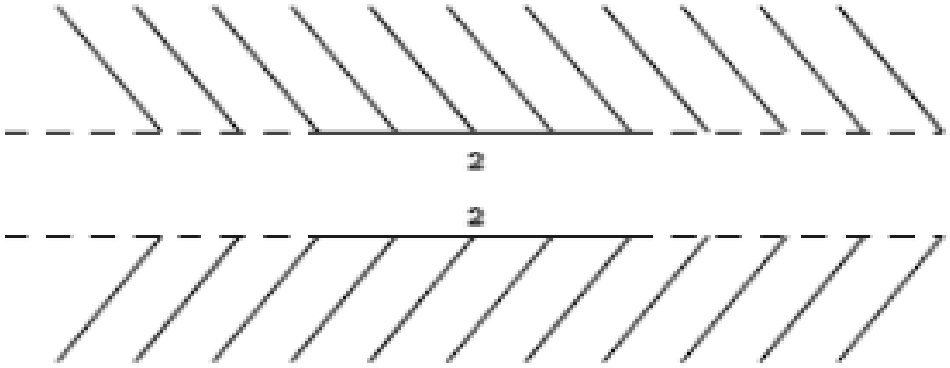


Figure 10.3: Region with positive sense with respect to cylindrical surface 2

Figure 10.4: Region with negative sense with respect to plane surface 3.

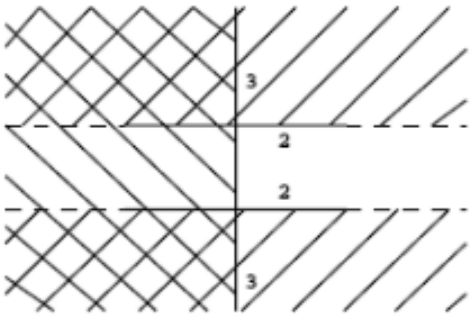


Figure 10.5: Figure 4-3 and Figure 4-4 overlaid creating a cross-hatched region that is identical to the hatched region in Figure 4-2.

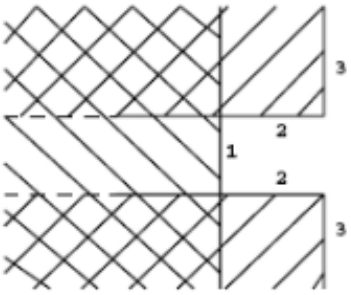


Figure 10.6: Region shown in Figure 4-2 superimposed with region negative with respect to (i.e., left of) plane surface 1.

that an intersection of two regions gives only the area common to both regions or the areas that overlap. Superimposing Fig. 10.3 and Fig. 10.4 results in Fig. 10.5. The cross-hatched regions show the space common to both regions. This is the same area hatched in Fig. 10.2.

Let us now deal with surface 1. To the quantity 2 -3 we will add everything with a negative sense with respect to plane surface 1 as indicated by the expression 2 -3:-1, or (2 -3):-1 if you prefer. First, recall that in the hierarchy of operations, intersections are performed first and then unions. Consequently, the parentheses are unnecessary in the previous expression. Second, recall that a union of two regions results in a space containing everything in the first region plus everything in the second region. This union also includes everything common to both regions. Superimposing the region shown in Fig. 10.2 and the region to the left of surface 1 results in Fig. 10.6. Our geometry now includes everything hatched plus everything crosshatched and has added part of the tunnel that is interior to cylindrical surface 2.

By the same method we will deal with cylindrical surface 4. To the quantity 2 -3:-1 we will add everything with a positive sense with respect to surface 4, written as 2 -3:-1:4. Figure 10.7 shows our new geometry. It includes everything in Fig. 10.6 plus everything outside surface 4.

Our final step is to block off the large tunnel extending to positive infinity (i.e., to the right) by adding the region with a positive sense with respect to plane surface 5 to the region shown in Fig. 10.7. The final expression that defines cell 3 of Fig. 10.1 is 2 -3:-1:4:5.

There is more than one way to define cell 3. Starting with plane surface 1, we can add the region to the left of 1 to the region outside cylindrical surface 2 or -1:2. This newly defined region is illustrated in Fig. 10.8. We wish to intersect this space with the space having a negative sense with respect to plane surface 3. Superimposing Fig. 10.8 and the region to the left of surface 3 results in Fig. 10.9. The cross-hatched area

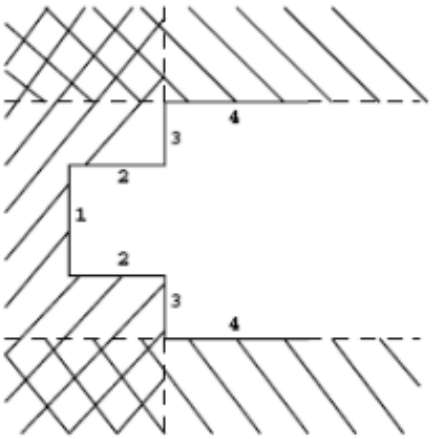


Figure 10.7: Region outside of surface 4 added to the region shown in Figure 4-6.

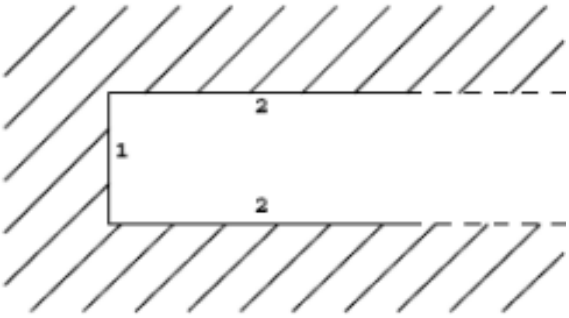


Figure 10.8: Union of regions to the left of surface 1 and outside of surface 2.

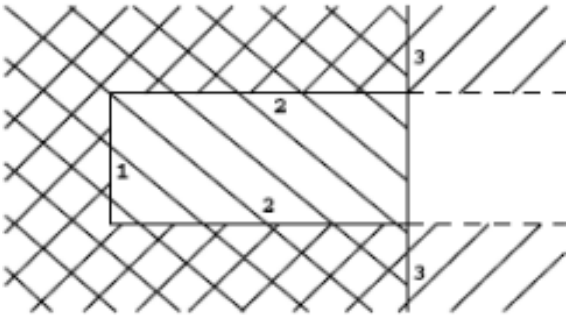


Figure 10.9: Region of Figure 4-8 superimposed with the region to the left of surface 3.

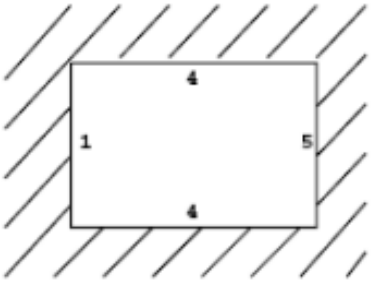


Figure 10.10: A starting point for defining cell 3.

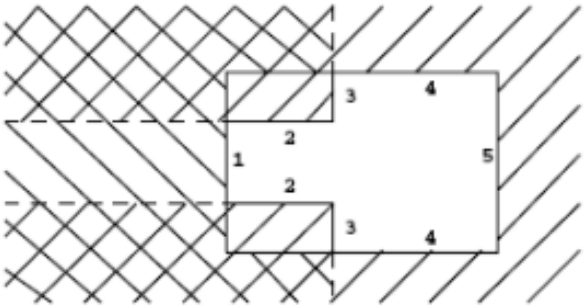


Figure 10.11: Union of the space block defined using outer boundaries of model and the left corner regions.

indicates the area common to both regions and is the result of the intersection. Note that the cross-hatched area of Fig. 10.9 is identical to the entire hatched plus crosshatched area of Fig. 10.6. Therefore, we have defined the same geometry in both figures but have used two different approaches to the problem. To ensure that the intersection of -3 is with the quantity -1:2 as we have illustrated, we must use parentheses giving the expression (-1:2) -3. Remember the order in which the operations are performed. Intersections are done before unions unless parentheses alter the order. The final expression is (-1:2) -3:4:5.

Another tactic to define cell 3 uses a somewhat different approach. Rather than defining a small region of the geometry as a starting point and adding other regions until we get the final product, we shall start by defining a block of space and adding to or subtracting from that block as necessary. We arbitrarily choose our initial block to be represented by 4: 1:5, illustrated in Fig. 10.10. Notice that the boundaries of this block are the outermost surfaces of our model: cylindrical surface 4 and planar surfaces 1 and 5.

To this block we need to add the space in the upper and lower left corners that belong to cell 3. The expression 2 -3 isolates the space we need to add. Adding 2 -3 to our original block, we have 4:-1:5:(2 -3). The parentheses are not required for correctness in this case but help to illustrate the path our reasoning has followed.

Figure 10.11 depicts the union of 2 -3 with the block of space we originally chose.

Now let us arbitrarily choose a different initial block, 4:5:-3, all the world except cell 2. From this region we need to subtract cell 1. If we intersect the region (2:-1) with (4:5:-3), as shown in Fig. 10.12, we will have introduced an undefined tunnel to the right of surface 5. To correct this error, define an area (2:-1:3) or (2:-1:5) and intersect this region with the initial block.

Yet another approach is to intersect the two regions -1:2 and -3:4, then add that to the region to the right of surface 5 by (-1:2) (-3:4):5. In the above paragraph the expression (4:5:-3) (2:-1:5) can have the common quantity 5 factored out, also resulting in (-1:2) (-3:4):5.

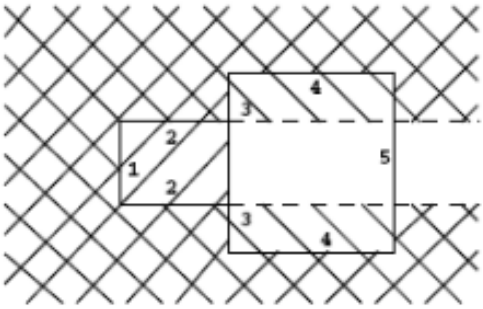


Figure 10.12: Region (2:-1) intersected with region (4:5:-3), creating an undefined region.

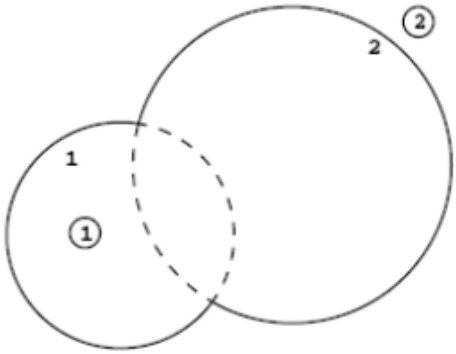


Figure 10.13: Simple two-cell model.

Finally, another approach is to forget about the reality of the geometry and to define cell 3 take the inverse (or complement) of all the cells bounding cell 3—cells 1 and 2. This says that cell 3 is the entire world excluding that which has already been defined to be in cells 1 and 2. The advantage of this method is that cells 1 and 2 are easy to specify and you do not get bogged down in details for cell 3. Cell 3 thus becomes (-1:2:3) (-3:4:5). Note that the specifications for cells 1 and 2 are reversed. Intersections become unions. Positive senses become negative. Then each piece is intersected with the other. There is a complement operator in MCNP6 that is a shorthand notation for the above expression; it is the symbol #, which can be thought of as meaning "not in." Therefore, cell 3, when specified as #1 #2, is translated as everything in the world that is not in cell 1 and not in cell 2.

10.1.1.2 Example 2

In this example (Fig. 10.13), cell 1 includes everything interior to both surfaces 1 and 2. It is simple enough that the answer is provided without explanation.

```
1  0   -1 : -2
2  0    1    2
```

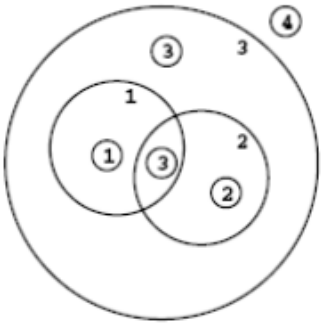


Figure 10.14: Illustration of disconnected cell 3.

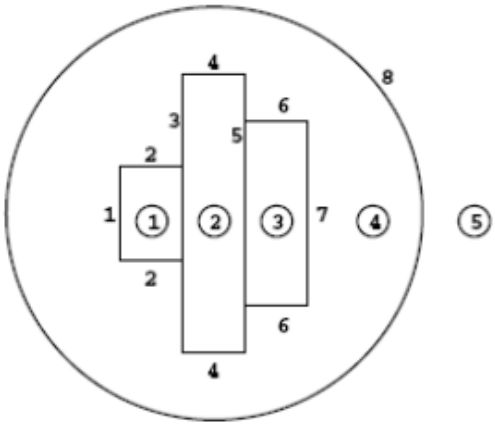


Figure 10.15: Horizontal cylinders internal to a sphere.

10.1.1.3 Example 3

In this geometry (Fig. 10.14) of four cells defined by three spheres, cell 3 is disconnected, consisting of two disjoint volumes. Cell 3 is the region inside surface 3 but outside surfaces 1 and 2 (-3 1 2) plus the region enclosed between surfaces 1 and 2 (-2 -1):

```
1  0   -1  2
2  0   -2  1
3  0   (-3 1 2):(-2 -1) $ parentheses not required
4  0    3
```

10.1.1.4 Example 4

In this example (Fig. 10.15), all vertical lines are planes with their positive sides to the right and all horizontal lines are cylinders. The surface list (with notional dimensions) is:

```
1  PX  -3
```

```
2  CX   2
3  PX  -1
4  CX   5
5  PX   1
6  CX   3.5
7  PX   3
8  SO   8
```

Cells 1, 2, and 3 are simple right-circular cylinders. Cell 4 is also simple to define with the complement operator. Cell 5 is also simple, everything in the world with a positive sense with respect to the outer sphere, surface 8.

```
1  0  1 -2 -3
2  0  3 -4 -5
3  0  5 -6 -7
4  0  #1 #2 #3 -8  $ or (-1:4:7:2 -3:5 6) -8
5  0  8            $ everything outside the outer sphere
```

Some users might try defining cell 5 simply as #4 (i.e., not cell 4). However, that would be incorrect. That syntax says cell 5 is everything in the universe not in cell 4, which includes cells 1, 2, and 3. The specification #4 #1 #2 #3 would be correct but should not be used because it is computationally inefficient. It tells MCNP6 that cell 5 is bounded by surfaces 1 through 7 in addition to surface 8. The lesson here is that extra, irrelevant surfaces in cell definitions—implicit or explicit—can cause MCNP6 to run significantly more slowly than it should because any time a particle enters a cell or has a collision in it, the intersection of the particle's trajectory with each bounding surface has to be calculated.

Specifying cell 4 exclusively with the complement operator is very convenient and computationally efficient in this case. However, it will be instructive to set up cell 4 explicitly without complements. There are many different ways to specify cell 4, The following approach should not be considered to be the way.

First consider cell 4 to be everything outside the big cylinder of surface 4 that is bounded on each end by surfaces 1 and 7. This is specified by (-1:4:7). The parentheses are not necessary but may add clarity. Now all that remains is to add the corners outside cylinders 2 and 6. The corner outside cylinder 2 is (2 -3), whereas it is (5 6) outside cylinder 6. Again the parentheses are optional. These corners are then added to what we already have outside cylinder 4 to get

```
(-1:4:7):(2 -3):(5 6)
```

The region described so far does not include cells 1, 2, or 3 but extends to infinity in all directions. This region needs to be terminated at the spherical surface 8. In other words, cell 4 is everything we have defined so far that is also common with everything inside surface 8 (that is, everything so far intersected with -8). So as a final result,

```
((-1:4:7):(2 -3):(5 6)) -8
```

The inner parentheses can be removed, but the outer ones are necessary (remember the hierarchy of operations) to give us

```
(-1:4:7:2 -3:5 6) -8
```

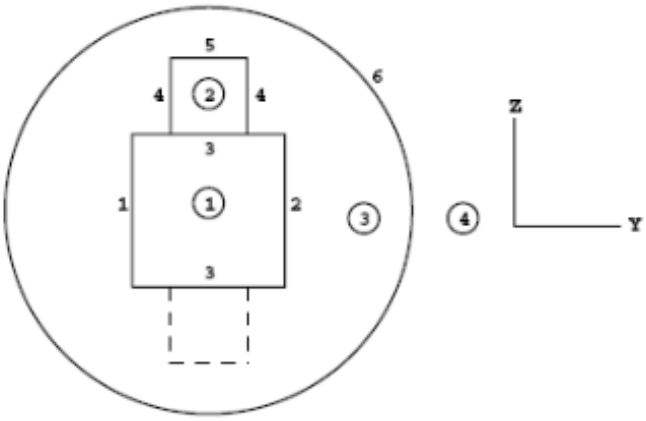


Figure 10.16: Horizontal and vertical cylinders in a sphere.

If the outer parentheses are removed, the intersection of -8 will occur only with 5 and 6, an event that is clearly incorrect.

10.1.1.5 Example 5

This example (Fig. 10.16) is similar to the previous one except that a vertical cylinder (surface 4) is added to one side of the horizontal cylinder (surface 3).

Cell 1 is (1 -3 -2), cell 3 is #1 #2 #4, and cell 4 is just 6.

Cell 2 is more than might initially meet the eye. The description of cell 2 might appear to be simply (-5 -4 3), but this definition causes two images of cell 2 to be created: one we desire above the y axis and one we do not want below the y axis. This undesired mirror image of cell 2 resides in the bottom half of cell 1 and is depicted by the dashed lines in Fig. 10.16. We need to add an ambiguity surface to keep cell 2 above the y axis. Let surface 7 be an ambiguity surface that is a plane at $z = 0$. This surface is defined in the MCNP6 input file like any other surface. Then cell 2 becomes (-5 -4 3 7) for the final result. You should convince yourself that the region above surface 7 intersected with the region defined by -5 -4 3 is cell 2. Do not even think of surface 7 as an ambiguity surface but just another surface defining some region in space. The mirror problem can also be avoided by defining cells 1 and 2 as right-circular-cylinder (RCC) macrobodies. The necessary cards for defining cells 1 and 2 as macrobodies could be, for example,

```
1  rcc   0 -2  0   0 4 0   4
2  rcc   0  0  0   0 0 7   1
```

In this case cells 1, 2 and 3 would simply be (-1), (-2 1), and (1 2 -6) respectively. Notice that to get the interface between the cylinders correct, macrobody 2 extends into cell 1 and is then truncated by the definition of cell 1.

10.1.1.6 Example 6

Figure 10.17 contains three concentric spheres with a box cut out of cell 3. Surface 8 is the front of the box and surface 9 is the back of the box. The cell cards are

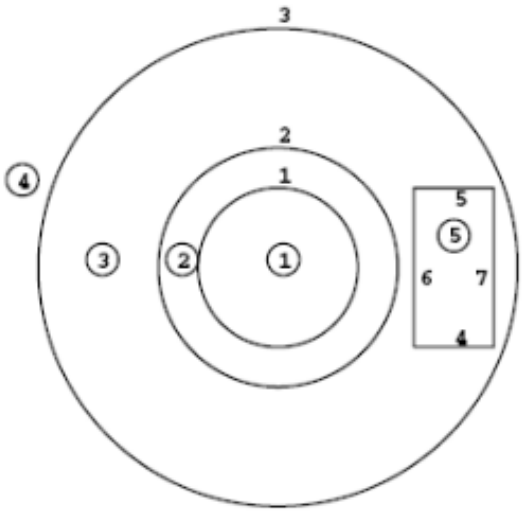


Figure 10.17: A box located within a concentric sphere.

```
1  0  -1
2  0  -2  1
3  0  -3  2 (-4:5:-6:7:8:-9) $ These parentheses are required.
4  0   3
5  0   4 -5  6 -7 -8  9
```

Cell 3 is everything inside surface 3 intersected with everything outside surface 2 but not in cell 5. Therefore, cell 3 could be written as

```
3  0   -3  2  #(4 -5 6 -7 -8 9)
```

or

```
3  0   -3  2  #5
```

or

```
3  0   -3  2 (-4:5:-6:7:8:-9)
```

Cell 5 could also be specified using a `RPP` macrobody. The correct cell and surface cards for this would be

```
5  0   -4                         $ Cell card
4 rrp   2  4    7.5  8.5    -2  2   $ Surface card
```

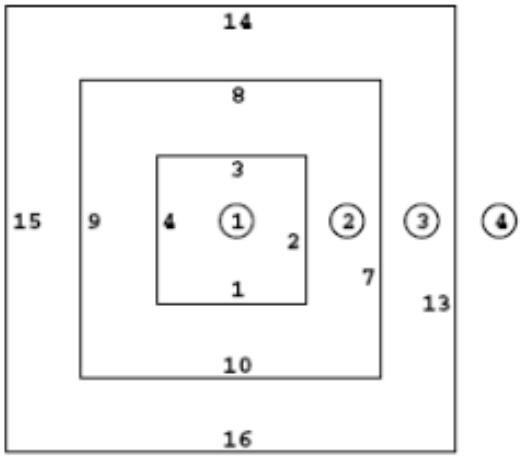


Figure 10.18: Concentric boxes.

10.1.1.7 Example 7

Figure 10.18 contains three concentric boxes, a geometry that is very challenging to set up using only intersections, easier with unions, and almost trivial with the `BOX` macrobody. Surfaces 5, 11, and 17 are the back sides of the boxes (smaller to larger, respectively); 6, 12, and 18 are the fronts:

```
1  0   -2  -3   4   1   5  -6
2  0   -7  -8   9  10  11 -12
       (2 : 3 : -4 :  -1 :  -5 :  6)
3  0  -13 -14  15  16  17 -18
       (7 : 8 : -9 : -10 : -11 : 12)
4  0   13 : 14 : -15: -16 : -17: 18
```

10.1.1.8 Example 8

Figure 10.19 contains two concentric spheres with a torus attached to cell 2 and cut out of cell 1:

```
1  0   -1  4
2  0   -2 (1 : -4)
3  0    2
```

If the torus were attached to cell 1 and cut out of cell 2, this bug-eyed geometry would be:

```
1  0   -1 : -4
2  0   -2  1  4
3  0    2
```

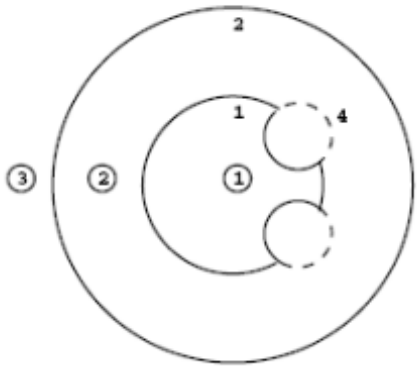


Figure 10.19: Torus attached to a concentric sphere.

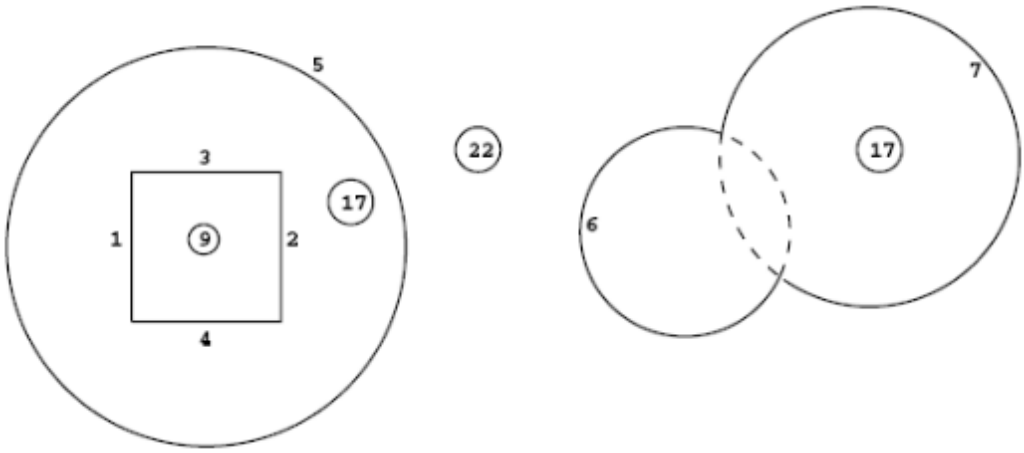


Figure 10.20: Disconnected cell.

10.1.1.9 Example 9

Notice that cell 17 is disconnected, having two pieces. Cell 9 in Fig. 10.20 is a box cut out of the left part of spherical cell 17; surface 9 is the front of the box and surface 8 is the rear. The right part of cell 17 is the space interior to spheres 6 and 7. An F4 tally in cell 17 would be the average flux in all parts of cell 17. An F2 surface tally on surface 7 would be the flux across only the solid portion of surface 7 in the figure. The cell specifications are:

```
9   0   -3 -2  4  1  8 -9
17  0   -5 (3 : -4 : -1 : 2 : 9 : -8) : -6 : -7
22  0    5  6  7
```

A variation on this problem is for the right portion of cell 17 to be the intersection of the interiors of surfaces 6 and 7 (the region bounded by the dashed lines in Fig. 10.20):

```
9   0   -3 -2  4  1  8 -9
17  0   -5 (3 : -4 : -1 : 2 : 9 : -8) : -6 -7
```

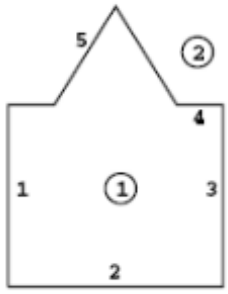


Figure 10.21: Box with an upside-down cone.

```
22  0    5 (6 : 7)
```

10.1.1.10 Example 10

Figure 10.21 contains a box with a cone sitting on top of it. Surface 6 is the front of the box and 7 is the rear. You should understand this example before going on to the next one.

```
1  0   1  2 -3 (-4 : -5) -6  7
2  0  -1 : -2 : 3 : 4  5 : 6 : -7
```

This problem could be simplified by replacing surfaces 1–6 with a `BOX` macrobody. To specify individual macrobody surfaces, the resulting cell and surface definitions must use macrobody facet notation. Typical cell and surface cards would look like

```
c cell cards
1  0   -8:(-5 8.5)
2  0   #1          $ or -8.4:-8.6:8.3:(8.5 5):8.1:-8.2

c surface cards
5  kz     8  0.25  -1
8  box   -2.5 -2.5 0  5 0 0  0 5 0  0 0 5
```

10.1.1.11 Example 11

Two views of this example appear in Figure 10.22. Surfaces 15 and 16 are cones, surface 17 is a sphere, and cell 2 is disconnected.

```
1  0    -1   2   3 (-4 : -16) 5 -6 (12 : 13 : -14)
       (10 : -9 : -11 : -7 : 8) 15
2  0   -10   9  11   7  -8  -1 : 2 -12 14 -6 -13 3
3  0   -17  (1 : -2 : -5 : 6 : -3 : -15 : 16 4)
4  0    17
```

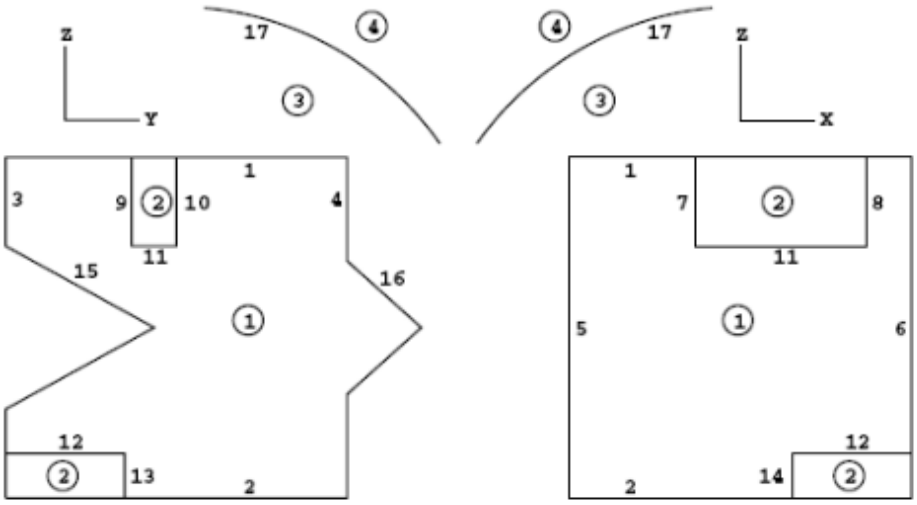


Figure 10.22: Views from two different perspectives of a complicated four-cell model.

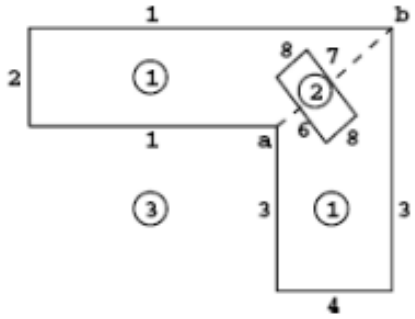


Figure 10.23: Two intersecting cylinders.

10.1.1.12 Example 12

In Figure 10.23, cell 1 consists of two cylinders joined at a 45-degree angle. Cell 2 is a disk consisting of a cylinder (surface 8) bounded by two planes. Surface 5 is a diagonal plane representing the intersection of the two cylinders. The problem is to specify the disk (cell 2) in one cell formed by the two cylinders (cell 1). A conflict arises in specifying cell 1 since, from the outside of cell 3, corner a between surfaces 1 and 3 is convex, but on the other side of the cell the same two surfaces form a concave corner at b. The dilemma is solved by composing cell 1 of two disconnected cells, each bounded by surface 5 between corners a and b. Surface 5 must be included in the list of surface cards in the MCNP6 input file. When the two parts are joined to make cell 1, surface 5 does not appear. Convince yourself by plotting it using an origin of 0 0 24 and basis vectors 0 1 1 0 -1 1. See Chapter 6 for an explanation of plotting commands.

```
1  0  (2 -1 -5 (7:8:-6)):(4 -3 5(-6:8:7))
2  0   -8 6 -7
3  0  (-2:1:5) (-4:3:-5)
```

A more efficient expression for cell 1 is

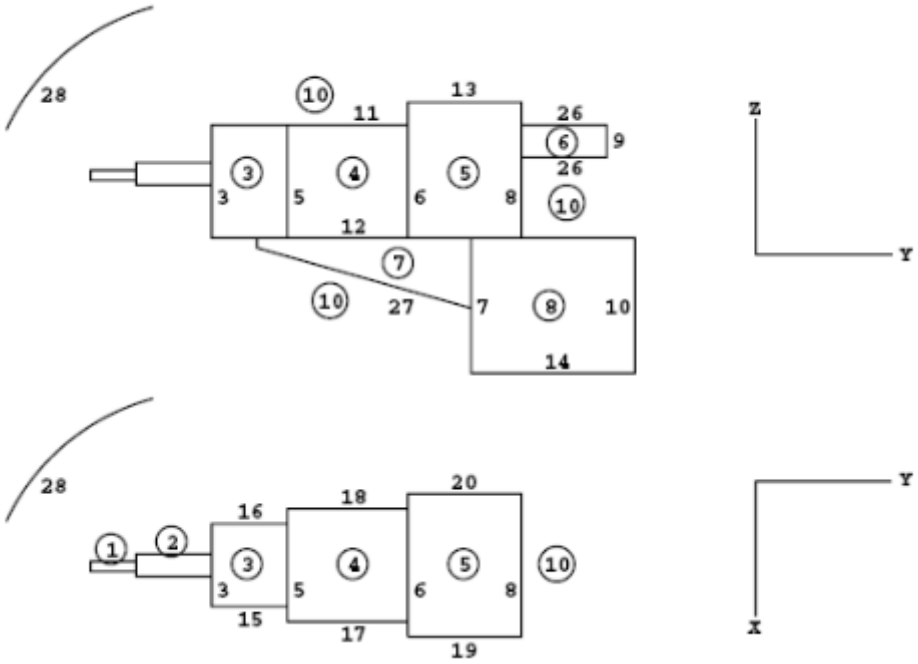


Figure 10.24: More complicated, yet straightforward to define.

```
1  0  (2 -1 -5:4 -3 5) (-6:8:7)
```

10.1.1.13 Example 13

This example (Figure 10.24) has the most complicated geometry so far, but it can be described very simply.

You can see that this example is similar to §10.1.1.1. There is just a lot more of it. It is possible to set this geometry up by any of the ways mentioned in §10.1.1.1. However, going around the outer surfaces of the cells inside cell 10 is tedious. There is a problem of visualization and also the problem of coming up with undefined tunnels going off to infinity as in §10.1.1.1.

The way to handle this geometry is by the last method in §10.1.1.1. Set up the cell/surface relations for each interior cell, then just take the complement for cell 10. For the interior cells,

```
1   0    1  -2 -23
2   0   -3  25 -24   2
3   0    3  -5  12 -15  16 -11
4   0    5  -6  12 -17  18 -11
5   0    6  -8  12 -13 -19  20
6   0    8  -9 -26
7   0  -12   4  -7 -27
8   0  -12   7 -10  14 -21  22
9   0    2  -3 -25
```

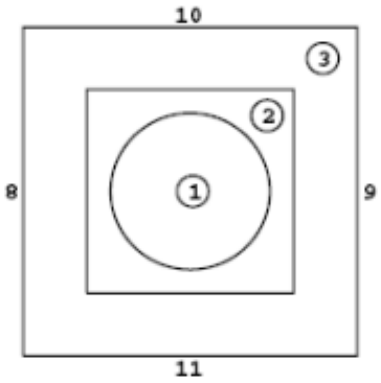


Figure 10.25: Sphere in a box in a box.

Cell 10 is surrounded by the spherical surface 28. Considering cell 10 to be everything outside cells 1 through 9 but inside surface 28, one can reverse the senses and replace all intersections with unions to produce

```
10  0  (-1:2:23) (3:-25:24:-2)
       (-3:5:-12:15:-16:11)
       (-5:6:-12:17:-18:11)
       (-6:8:-12:13:19:-20)
       (-8:9:26) (12:-4:7:27)
       (12:-7:10:-14:21:-22)
       (-2:3:25) -28
```

Note how easy cell 10 becomes when the complement operator is used:

```
10 0 #1 #2 #3 #4 #5 #6 #7 #8 #9 -28
```

Once again this example can be greatly simplified by replacing all but cell 7 with macrobodies. However the definition of cell 7 must then be changed to use the facets of the surrounding macrobodies instead of surfaces 12 and 7. The facets of macrobodies can be visualized using the `MBODY OFF` option of the geometry plotter [§6.2.4.1.4].

10.1.1.14 Example 14

Figure 10.25 illustrates some necessary conditions for volume and area calculations. The geometry has three cells, an outer cube, an inner cube, and a sphere at the center. If cell 3 is described as

```
3  0   8 -9 -10 11 -12 13 #2 #1
```

(and #1 must be included to be correct), the volume of cell 3 cannot be calculated. As described, it is not bounded by all planes so it is not a polyhedron, nor is it rotationally symmetric. If cell 3 is described by listing all 12 bounding surfaces explicitly, the volume can be calculated.

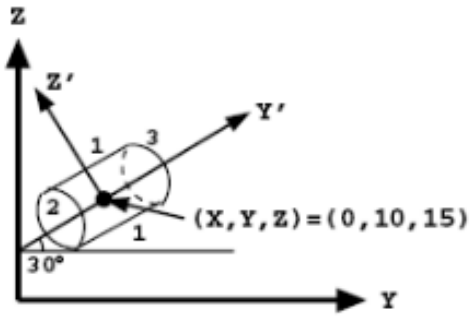


Figure 10.26: Tilted can in the y-z plane showing the main and auxiliary coordinate systems.

10.1.2 Coordinate Transformations

In most problems, the surface transformation feature of the TR card will be used with the default setting, $m = 1$. When $m = 1$ applies, most of the geometry can be set up easily in an (x, y, z) coordinate system and only a small part of the total geometry will be difficult to specify. For example, a box with sides parallel to the (x, y, z) coordinate system is simple to describe, but inside might be a tilted object consisting of a cylinder bounded by two planes. Since the axis of the cylinder is neither parallel to nor on the x, y, or z axis, a general quadratic must be used to describe the surface of the cylinder. The GQ surface card has ten entries that are usually difficult to determine. On the other hand, it is simple to specify the entries for the surface card for a cylinder centered on the y axis. Therefore, we define an auxiliary coordinate system (x', y', z') so the axis of the cylinder is one of the primed axes, y' for example. Now we will use the TR card to describe the relationship between one coordinate system and the other. The $m = 1$ specification on the TR card requires that the coordinates of a vector from the (x, y, z) origin to the (x', y', z') origin be given in terms of (x, y, z).

Only in rare instances will $m = -1$ be needed. Some unusual circumstances may require that a small item of the geometry be described in a certain system which we will call (x, y, z), and the remainder of the surfaces would be easily described in an auxiliary system (x', y', z'). The o_i displacement entries on the TR card are then the coordinates of a vector from the (x', y', z') origin to the (x, y, z) origin given in terms of the primed system.

10.1.2.1 Example 15

The following example consists of a can whose axis is in the y-z plane but tilted 30° from the y axis and whose center is at $(0, 10, 15)$ in the (x, y, z) coordinate system. The can is bounded by two planes and a cylinder, as shown in Fig. 10.26.

The surface cards that describe the can in the simple (x', y', z') system are the following:

```
1  1  CY    4
2  1  PY   -7
3  1  PY    7
```

The 1 before the surface mnemonics on the cards is the n that identifies to which TRn card these surface cards are associated. TRn card indicates the relationship of the primed coordinate system to the basic coordinate system.

We will specify the origin vector as the location of the origin of the (x', y', z') coordinate system with respect to the (x, y, z) system; therefore, $m = 1$. Since we wanted the center of the cylinder at $(0, 10, 15)$, the o_i

entries are simply `0 10 15`. If, however, we had wanted surface 2 to be located at $(x, y, z) = (0, 10, 15)$, a different set of surface cards would accomplish it. If surface 2 were at $y' = 0$ and surface 3 at $y' = 14$, the o_i entries would remain the same. The significant fact to remember about the origin vector entries is that they describe one origin with respect to the other origin. The user must locate the surfaces about the auxiliary origin so that they will be properly located in the main coordinate system.

The rotation matrix entries on the TRn card are the cosines of the angles between the axes as listed in §5.5.3. In this example, the x axis is parallel to the x' axis. Therefore, the cosine of the angle between them is 1. The angle between y and x' is 90° with a cosine of 0. The angle between z and x', and also between x and y', is 90° with a cosine of 0. The angle between y and y' is 30° with a cosine of 0.866. The angle between z and y' is 60° with a cosine of 0.5. Similarly, 90° is between x and z'; 120° is between y and z'; and 30° is between z and z'. The complete TRn card is

```
TR1   0 10 15  1 0 0  0 0.866 0.5  0 -0.5 0.866
```

An asterisk preceding TRn indicates that the rotation matrix entries are the angles given in degrees between the appropriate axes. The entries using the *TRn mnemonic become

```
*TR1   0 10 15  0 90 90  90 30 60  90 120 30
```

The default value of 1 for *m*, the thirteenth entry, has been used and is not explicitly specified.

The user need not enter values for all of the rotation matrix entries. As shown in §5.5.3, the rotation matrix may be specified in any of five patterns. Pattern 3(a) was used above, but the simplest form for this example is pattern 3(d) because all the skew surfaces are surfaces of revolution about some axis. The complete input card then becomes

```
*TR1   0 10 15  3J  90 30 60
```

10.1.2.2 Example 16

The following example illustrates another use of the TRn card. The first part of the example uses the TR card and the default *m*=1 transformation; the second part uses the TR card with *m*=−1. Both parts and transformations are used in the following input file.

Listing 10.1: example_tr.card.mcnp.inp.txt

```
EXAMPLE OF SURFACE TRANSFORMATIONS
2  0   -4   3  -5
6  0  -14 -13 : -15 41 -42
998 0  #2 #6 -999
999 0 999  $ outside world

C  Cell 2 surfaces
3   1  PX   -14
4   1  X    -14 10 0 12 14 10
5   1  PX    14
C
C  Cell 6 surfaces
```

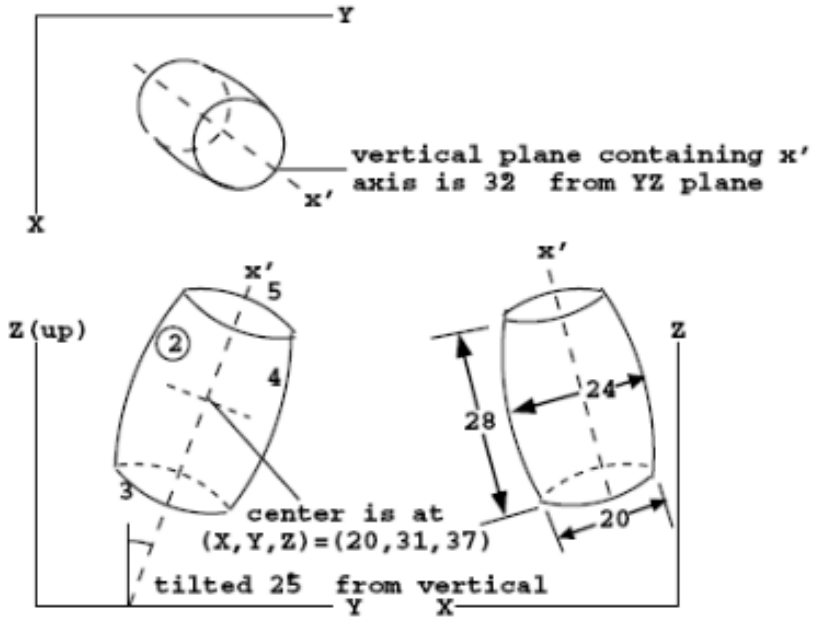


Figure 10.27: A tilted barrel as seen from three views.

```
13  2  SX   -15 70
14  2  CX    30
15  2  KY    75   1.2641975E-01
41  2  PY     0
42  2  PY    75
C
C  Surface defining outside world
999 so 500

TR1     20   31  37  0.223954  0.358401  0.906308
TR2   -250 -100 -65  0.675849  0.669131  0.309017
       J J 0.918650     J J  -0.246152    -1
C
IMP:N  1 1 1 0
SDEF
PRINT
NPS      5000
```

10.1.2.2.1 Case 1: TR and *m*=1

Cell 2 is bounded by the planar surfaces 3 and 5 and the spheroid surface 4, which is a surface of revolution about the skew axis x' in Fig. 10.27.

To get the coefficients of surfaces 3, 4, and 5, define the x' axis as shown in the drawings. Because the surfaces are surfaces of revolution about the x' axis, the orientation of the y' and z' axes does not matter. Then set up cell 2 and its surfaces with coefficients defined in the (x', y', z') coordinate system.

On the `TR`1 card, the origin vector is the location of the origin of the (x', y', z') coordinate system with respect to the main (x, y, z) system of the problem. The rotation matrix pattern 3(d) in §5.5.3 is appropriate

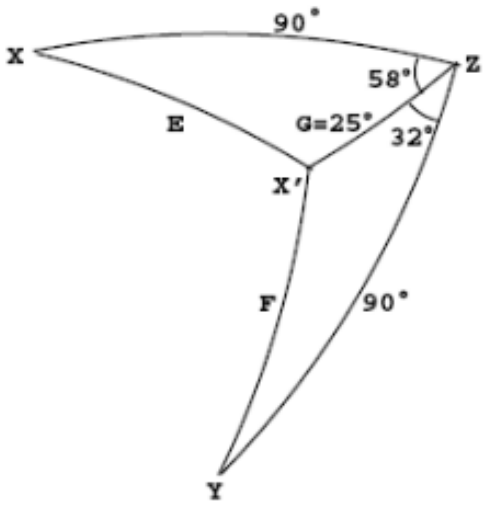


Figure 10.28: Angles between the x' axis and the main (x, y, z) coordinate system of Case 1.

since the surfaces are all surfaces of revolution about the x' axis. The components of one vector of the transformation matrix are the cosines of the angles between x' and the x, y, and z axes. They are obtained from spherical trigonometry as shown in Fig. 10.28 and by calculating

$$\cos(E) = \cos(50^\circ)\sin(25^\circ) = 0.223954, \tag{10.1}$$
$$\cos(F) = \cos(32^\circ)\sin(25^\circ) = 0.358401, \tag{10.2}$$
$$\cos(G) = \cos(25^\circ) = 0.906308. \tag{10.3}$$

10.1.2.2.2 Case 2: TR and m=−1

Cell 6 is the union of a can bounded by spherical surface 13, cylindrical surface 14, conical surface 15, and two ambiguity surfaces [§2.2.3.2] 41 and 42, which are planes. Surface 42 is required because when surface 15 is transformed into the (x, y, z) system it becomes a type GQ surface, which in this case is a cone of two sheets [Note (5) of §5.5.3]. Surfaces 13 and 14 are surfaces of revolution about one axis, and surfaces 15, 41, and 42 are surfaces of revolution about an axis perpendicular to the first axis. Both axes are skewed with respect to the (x, y, z) coordinate system of the rest of the geometry.

Define the auxiliary (x', y', z') coordinate system as shown in Fig. 10.29. Set up cell 6 with its surfaces specified in the (x', y', z') coordinate system as part of the input file and add a second transformation card, TR2.

Because the location of the origin of the (x, y, z) coordinate system is known relative to the (x', y', z') system (rather than the other way around, as in Case 1), it is necessary to use the reverse mapping. This is indicated by setting $m = -1$. In this reverse mapping, the origin vector $(-250, -100, -65)$ is the location of the origin of the (x, y, z) system with respect to the (x', y', z') system. For the components of the transformation matrix, pattern 3(c) out of the five possible choices from §5.5.3 is most convenient here. The (x, y, z) components of z' and the (x', y', z') components of z are easy to get, while the components of x and of y are not. The whole transformation matrix with the components that are obtained from Fig. 10.29 is given in Table 10.1.

The signs of the zz' and xx' components are determined by inspection of the figure.

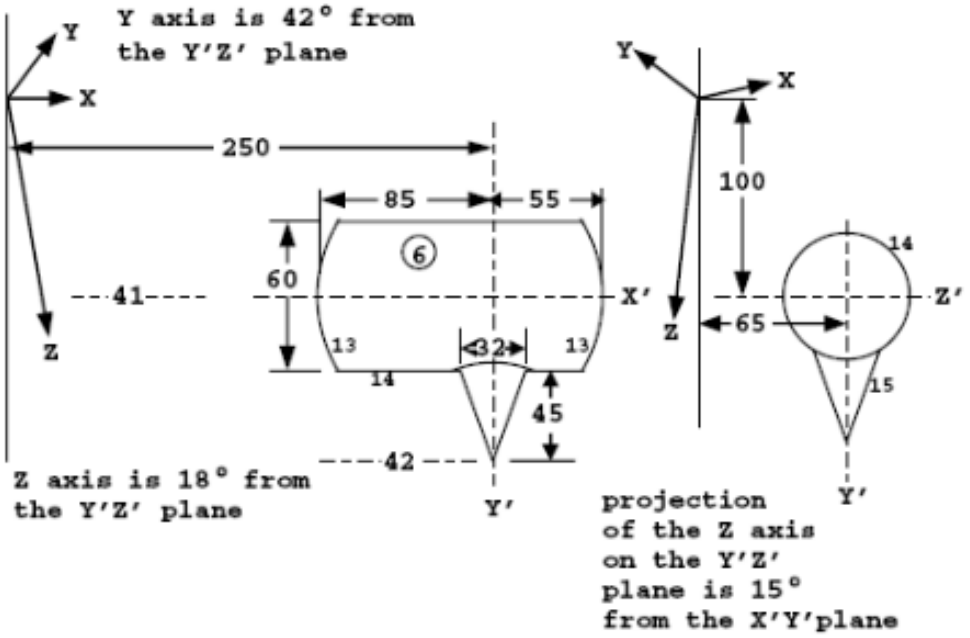


Figure 10.29: Case 2 geometry.

Table 10.1: Case 2 Transformation Matrix

	x	y	z
x'	$\sqrt{1.0 - 0.669131^2 - 0.309017^2)} = 0.675849$	$\cos(48°) = 0.669131$	$\cos(72°) = 0.309017$
y'	J	J	$\cos(15°)\cos(18°) = 0.918650$
z'	J	J	$-\sqrt{1.0 - 0.669131^2 - 0.309017^2} = 0.675849$

10.1.3 Repeated Structure and Lattice Examples

10.1.3.1 Example 17

The example shown in Listing 10.2 illustrates the use of transformations with simple repeated structures.

Listing 10.2: example_lattice_geometry_1.mcnp.inp.txt

```
simple repeated structures
1  0  -27  #2  #5                    imp:n=1
2  0    1  -2  -3   4  -5   6  fill=1 imp:n=1
3  0  -10 -11  12              u=1    imp:n=1
4  0   #3                      u=1    imp:n=1
5  like 2 but trcl=3
7  0   27                            imp:n=0

1  px   -3
2  px    3
3  py    3
4  py   -3
5  pz    4.7
6  pz   -4.7
10 cz    1
11 pz    4.5
12 pz   -4.5
27 s     3.5  3.5  0 11

sdef   pos 3.5 3.5 0
f2:n   1
*tr3   7 7 0  40 130 90  50 40 90  90 90 0
nps    10000
```

The geometry consists of a sphere enclosing two boxes that each contains a cylindrical can.

The geometric structure of this example can be displayed using the plot feature in MCNP6. Specifically, Fig. 10.30 can be obtained by launching the plotter:

```
mcnp6 ip i= example_lattice_geometry_1.mcnp.inp.txt
```

clicking the lower left hand corner of the plot window (`click here or picture or window`) and entering the following three settings:

```
b 1 0 0 0 1 0
ex 11
or 3.5 3.5 0
```

Cell 2 is filled by universe 1. Two cells are in universe 1—the cylindrical can, cell 3, and the space outside the can, cell 4. Cell 2 is defined and the `LIKE n BUT` card duplicates the structure at another location. The `TRCL` entry identifies a `TR` card that defines the displacement and rotational axis transformation for cell 5.

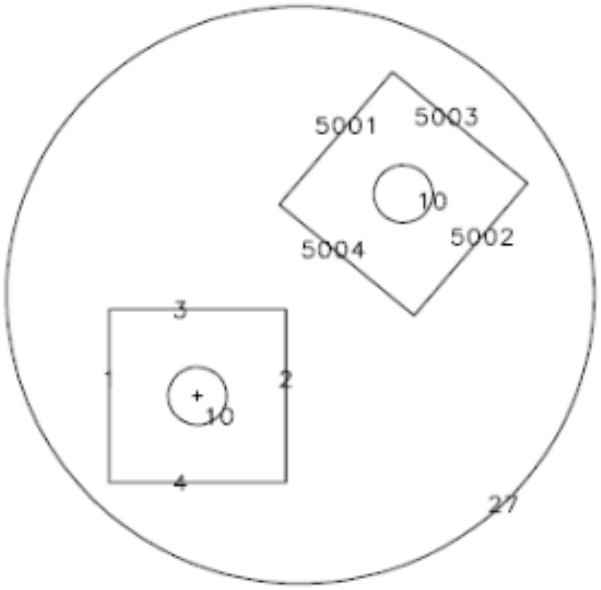


Figure 10.30: Geometry of Example 1: a sphere enclosing two boxes that each contains a cylindrical can.

10.1.3.2 Example 18

The example shown in Listing 10.3 illustrates the `LIKE n BUT` construct, the `FILL` card, the `U` card, two forms of the `TRCL` card, and a multiple source-cell definition.

Listing 10.3: example_lattice_geometry_2.mcnp.inp.txt

```
lattice example 18
1   1 -0.5   -7  #2 #3 #4 #5 #6 imp:n=1
2   0         1 -2 -3 4 5 -6    imp:n=2 trcl=2 fill=1
3   like 2 but trcl=3
4   like 2 but trcl=4
5   like 2 but trcl=5 imp:n=1
6   like 2 but trcl=6
7   0         7           imp:n=0
8   0         8 -9 -10 11 imp:n=1 trcl=(-.9 .9 0) fill=2 u=1
9   like 8 but trcl=(.9 .9 0)
10  like 8 but trcl=(.1 -.9 0)
11  2 -18    #8 #9 #10 imp:n=1 u=1
12  2 -18   -12 imp:n=1 trcl=(-.3 .3 0) u=2
13  like 12 but trcl=( .3  .3 0)
14  like 12 but trcl=( .3 -.3 0)
15  like 12 but trcl=(-.3 -.3 0)
16  1 -0.5  #12 #13 #14 #15 u=2 imp:n=1

1  px   -2
2  py    2
3  px    2
4  py   -2
5  pz   -2
6  pz    2
7  so   15
8  px   -0.7
```

```
9  py    0.7
10 px    0.7
11 py   -0.7
12 cz    0.1

sdef  erg=d1 cel=d2 rad=d5 ext=d6 axs=0 0 1 pos=d7
#      si1       sp1      sb1
        1        0        0
        3        0.22     0.05
        4        0.08     0.05
        5        0.25     0.1
        6        0.18     0.1
        7        0.07     0.2
        8        0.1      0.2
        9        0.05     0.1
       11        0.05     0.2
si2   L (12  <  8 9 10  <  2 3 4 5 6 )
        (13  <  8 9 10  <  2 3 4 5 6 )
        (14  <  8 9 10  <  2 3 4 5 6 )
        (15  <  8 9 10  <  2 3 4 5 6 )
sp2  1  59r
si5    0 0.1
sp5   -21  1
si6    -2  2
sp6     0  1
si7 L  0.3 0.3 0  0.3 -0.3 0  -0.3 0.3 0  -0.3 -0.3 0
sp7    1          1            1           1
c
m1     6000  1
m2    92235  1
c
drxs
tr2    -6  7  1.2
tr3     7  6  1.1
tr4     8 -5  1.4
*tr5   -1 -4  1    40 130 90  50 40 90  90 90 0
tr6    -9 -2  1.3
f4:n    2 3 4 5 6 12 13 14 15
e4      1 3 5 7 9 11 13
sd4     5j  1.8849555921  3r
fq      f e
cut:n   1e20  0.1
print
nps     100000
```

Cell 2 could be replaced with an `RPP` macrobody that can then be replicated and translated identically to cell 2 above.

Figure 10.31 can be displayed using the geometry plotter in command-prompt mode [§6.2.4] and entering:

```
basis 1 0 0 0 1 0  extent 21  label 0 0
```

Figure 10.31 shows five cells, numbers 2 through 6, identical except for their locations. Cell 2 is described fully and the other four are declared to be like cell 2 but in different locations. Cell 2 is defined in an auxiliary coordinate system that is centered in the cell for convenience. That coordinate system is related to the main

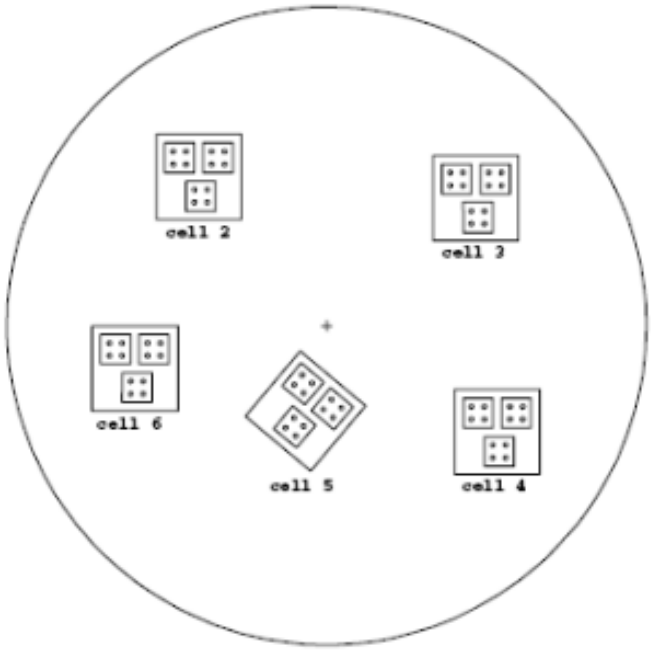


Figure 10.31: Repeated structures located at different positions and orientations.

coordinate system of the problem by transformation number 2, as declared by the TRCL=2 entry and the TR2 card. Cells 2 through 6 are all filled with universe number 1. Because no transformation is indicated for that filling, universe 1 inherits the transformation of each cell that it fills, thereby establishing its origin in the center of each of those five cells.

As shown in Fig. 10.32, universe 1 contains three infinitely long square tubes embedded in cell 11, which is unbounded. All four of these infinitely large cells are truncated by the bounding surfaces of each cell that is filled by universe 1, thus making them effectively finite. To illustrate the two possible ways of performing transformations, the transformations that define the locations of cells 8, 9 and 10 are entered directly on the cell cards after the TRCL symbol rather than indirectly through TR cards as was done for cells 2 through 6. Cells 8, 9 and 10 are each filled with universe 2, which consists of five infinite cells truncated by the boundaries of higher level cells. The simplicity and lack of repetition in this example were achieved by careful choice of the auxiliary coordinate systems at all levels. All of the location information is contained in just a few TRCL entries, some direct and some pointing to a few TR cards.

The source definition is given on the SDEF, SIn, and SPn cards. The source desired is a cylindrical volume distribution, equally probable in all the cylindrical rods. The energies are given by distribution 1. On the CEL entry, source distribution 2 includes all 60 of the cylindrical rod cells, using the shorthand method described in §5.8.1.6. A cylindrical volume distribution is specified by the source distributions on the RAD, EXT, AXS, and POS entries. The cylinder is centered about the origin, with a radius of 0.1 (SI5) and a length of 4 (SI6, from −2 to 2). The four sets of entries on the SI7 card are the origins of the four cylinders of cells 12–15. These parameters describe exactly the four cells 12–15.

10.1.3.3 Example 19

The example shown in Listing 10.4 illustrates the use of the FILL, U, and LAT cards to create an object within several cells of a lattice. A cylinder contains a square lattice and the cells in the inner 3×3 array of that lattice each contain a small cylinder.

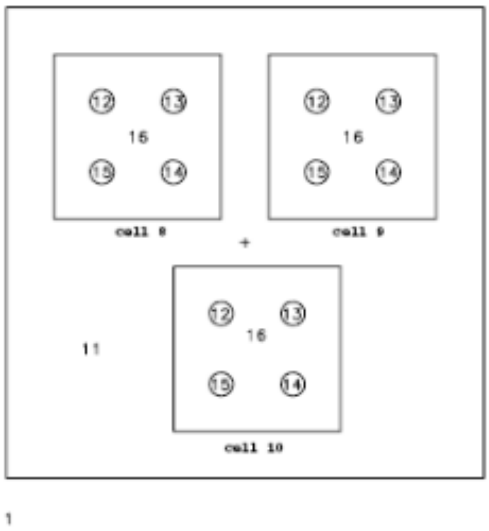


Figure 10.32: Close up of the repeated structure defined by universe 1 in Fig. 10.31.

Listing 10.4: example_lattice_geometry_3.mcnp.inp.txt

```
simple lattice
1   0    -1  fill=1 imp:n=1
2   0  -301  302 -303  304 lat=1 u=1 imp:n=1
         fill -2:2 -2:2 0:0
         1 1 1 1 1 1 2 2 2 1 1 2 2 2 1 1 2 2 2 1 1 1 1 1 1
3   0   -10  u=2 imp:n=1
4   0    #3  u=2 imp:n=1
5   0     1      imp:n=0

1   cz   45
10  cz    8
301 px   10
302 px  -10
303 py   10
304 py  -10

sdef
mode n
nps 5000
```

The resulting geometry is shown in Fig. 10.33. Cell 1 is the interior of the cylinder, and cell 5 is everything outside (all surfaces are infinite in the z direction). Cell 1 is filled by universe 1. Cell 2 is defined to be in universe 1. Surfaces 301–304 define the dimensions of the square lattice.

When filling the cells of a lattice, all visible cells, even those only partially visible, must be specified by the FILL card. In this case, the "window" created by the cylinder reveals portions of 25 cells (5×5 array). A FILL card with indices of -2 to 2 in the x and y directions will place the $[0, 0, 0]$ element at the center of the array. Universe 2, described by cells 3 and 4, is the interior and exterior, respectively, of an infinite cylinder of radius 8 cm. The cells in universe 1 not filled by universe 2 are filled by universe 1, so in effect they are filled by themselves.

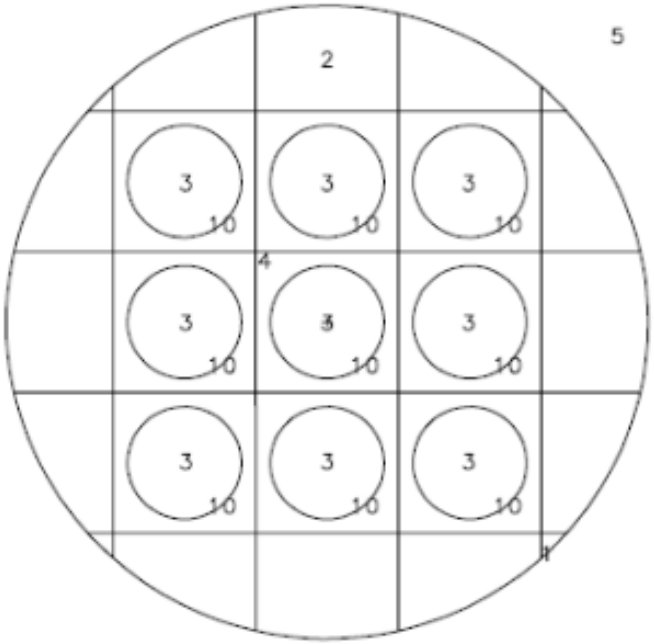


Figure 10.33: The simple lattice defined by Example 3.

10.1.3.4 Example 20

The example shown in Listing 10.5 illustrates a lattice geometry and uses the `FILL` entries followed by transformations, universes, and lattices.

Listing 10.5: example_lattice_geometry_4.mcnp.inp.txt

```
Lattice example
1  1 -0.6   -1                                        imp:n=1
c 2  0         1  2 -4                                imp:n=1
2  0          1 -2 -4  fill=1 (-6 -6.5 0)            imp:n=1
3  0          2 -3 -4 *fill=2 (-7 5 0 30  60 90 120 30 90) imp:n=1
4  0          2  3 -4 *fill=2 ( 4 8 0 15 105 90  75 15 90) imp:n=1
5  0          4                                      imp:n=0
6  0         -5   6  -7   8  -9  10 fill=3 u=1 lat=1  imp:n=1
7  0        -11  12 -13  14 -15  16 fill=5 u=2 lat=1  imp:n=1
18 3 -2.7   -18                          u=5          imp:n=1
8  2 -0.8   -17  u=3                                  imp:n=1
9  0         17  u=3                                  imp:n=1
10 0        -18  u=4                                  imp:n=1

1  sy  -5 3
2  py   0
3  px   0
4  so  15
5  px   1.5
6  px  -1.5
7  py   1
8  py  -1
9  pz   3
10 pz  -3
11 p    1 -0.5 0  1.3
```

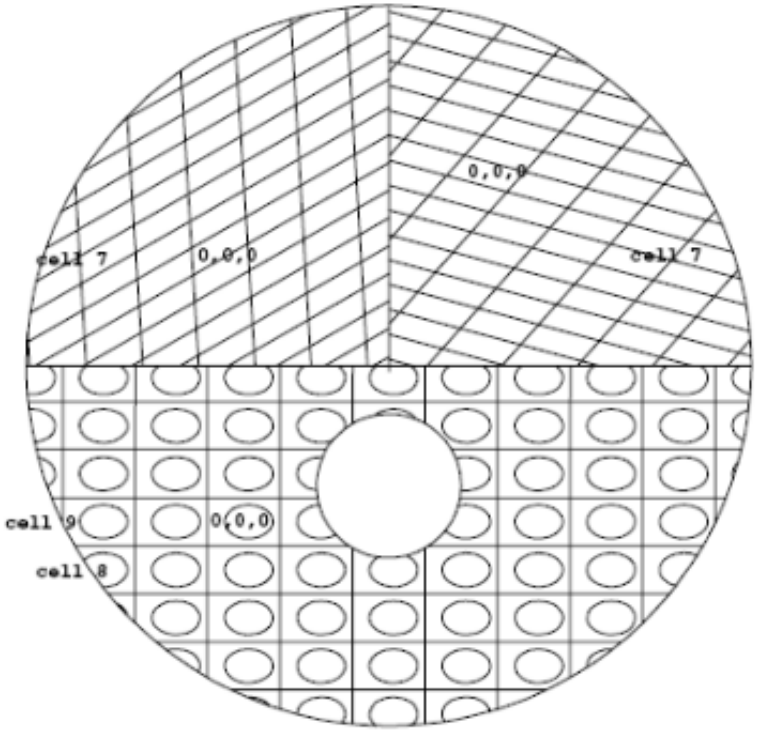


Figure 10.34: Lattices with universes and coordinate transformation.

```
12 p    1 -0.5 0 -1.3
13 py   0.5
14 py  -0.5
15 pz   3
16 pz  -3
17 sq   1  2  0  0  0  0  -1  0.2  0  0
18 so  10

sdef   pos 0 -5 0  erg d1  rad d2
si1    0 10
sp1    0  1
si2    3
sp2   -21
e0     1 2 3 4 5 6 7 8 9 10 11 12
f2:n   3
sd2    1
f4:n   8 9
sd4    1 1
m1      4009  1
m2      6000  1
m3     13027  1
nps    100000
print
dbcn     0 0 1 4
```

The geometry for this example is shown in Fig. 10.34.

Cell 2 is the bottom half of the large sphere outside the small sphere (cell 1) and is filled by universe 1. The transformation between the filled cell and the filling universe immediately follows in parentheses.

Cell 6 describes a hexahedral lattice cell (LAT = 1) and, by the order of specification of its surfaces, also describes the order of the lattice elements. The $[0, 0, 0]$ element has its center at $(-6, -6.5, 0)$, according to the transformation information on the card for cell 2. Element $[1, 0, 0]$ is beyond surface 5, element $[–1, 0, 0]$ is beyond surface 6, element $[0, 1, 0]$ is beyond surface 7, etc. Cell 6 is filled by universe 3, which consists of two cells: cell 8 inside the ellipsoid and cell 9 outside it.

Alternatively, cell 6 could have been defined using a macrobody, either RPP or BOX. When a lattice cell is defined with a macrobody, some of the lattice-element indexing is predetermined. For example, the first, third and fifth facets are used to define the direction of increasing indices. For the RPP, the second index increases in the positive y direction and the third index increases in the positive z direction. For the BOX, the order of defining the three vectors will determine the axis each index will increase in a positive direction.

Cell 3 is the top left-hand quarter of the sphere; cell 4 is the top right-hand quarter. Both are filled by universe 2. Both FILL entries are followed by a transformation. The inter-origin vector portion of the transformation is between the origin of the filled cell and the origin of the filling universe, with the universe considered to be in the auxiliary coordinate system. The $[0, 0, 0]$ lattice element is located around the auxiliary origin and the lattice elements are identified by the ordering of the surfaces describing cell 7. The skewed appearance is caused by the rotation part of the transformation.

The source is centered at $(0, –5, 0)$ (i.e., at the center of cell 1). It is a volumetric source filling cell 1, and the probability of a particle being emitted at a given radius is given by the power-law function. For RAD the exponent defaults to 2, so the probability increases as the square of the radius, resulting in a uniform volumetric distribution.

10.1.3.5 Example 21

The example in Listing 10.6 illustrates a more complicated lattice geometry and uses the FILL card followed by the array specification. It builds on the expertise from §10.1.3.4.

Listing 10.6: example_lattice_geometry_9.mcnp.inp.txt

```
Lattice Example 21
1  1 -0.6   -5 imp:n=1
2  0        -1  2 -3  4  5 -22 23 imp:n=1 fill=1
3  0         1:-2: 3:-4:22:-23    imp:n=0
4  2 -0.8   -6  7 -8  9           imp:n=1 lat=1 u=1
            fill=-2:2 -4:4 0:0 1 1 1 1 1 1 1 1 2(3) 1 1 3 1 1 1
                  1 2 3 2 1 1 1 1 1 1 1 4(2) 2 1 1 1 1 3 4(1) 1
                  1 2 3 1 1 1 1 1 1 1
5  3 -0.5  -11  10  12     imp:n=1 vol=1 u=2
6  4 -0.4   11:-10:-12     imp:n=1 vol=1 u=2
7  0       -13             imp:n=1 vol=1 u=3 fill=5
8  3 -0.5   13             imp:n=1 vol=1 u=3
9  0       -14  15 -16  17 imp:n=1 vol=1 u=5 lat=1 fill=6
10 4 -0.4  -24             imp:n=1 vol=1 u=6
11 3 -0.5  -18  19 -20  21 imp:n=1 vol=1 u=4
12 4 -0.4   18:-19: 20:-21 imp:n=1 vol=1 u=4

1  px    15
2  px   -15
3  py    15
4  py   -15
```

```
5  s      7 2.1 0  3.5
6  px     4
7  px    -5
8  py     2
9  py    -2
10 p      0.7 -0.7  0  -2.5
11 p      0.6  0.8  0   0.5
12 py    -1
13 x     -4.5 0 -0.5 1.7 3.5 0
14 px     1.6
15 px    -1.4
16 py     1
17 py    -1.2
18 px     3
19 px    -3
20 py     0.5
21 py    -0.6
22 pz     6
23 pz    -7
24 so    10

sdef   erg d1  pos 7 2 0  cel=1  rad d2
si2    3.6
si1    0 10
sp1    0  1
e4     1 3 5 7 9 11
f4:n   10
m1      4009  1
m2      6000  1
m3     13027  1
m4      1001  2   8016  1
nps    100000
dbcn   0 0 1 4
*tr1   0 0 0  10 80 90  100 10 90
*tr2   1 0 0   2 88 90   92  2 90
tr3    3 0 0
print
```

This example has three "main" cells: cell 1 is inside surface 5, cell 3 is the outside world, and cell 2 is the large square (excluding cell 1) that is filled with a lattice, some of whose elements are filled with three different universes. A schematic of the geometry is given in Fig. 10.35.

Universe 1 is a hexahedral lattice cell infinite in the z direction. Based on the FILL parameters, it can be seen that the lattice has five elements in the first direction numbered from -2 to 2, nine elements in the second direction numbered from -4 to 4, and one element in the third direction. The remaining entries on the card are the array that identifies which universe is in each element, starting in the lower left-hand corner with $[-2,-4,0]$, $[-1,-4,0]$, $[0,-4,0]$, etc. An array entry (in this case 1) that is the same as the number of the universe of the lattice itself means that element is filled by the material specified for the lattice cell. Element $[1,-3,0]$ is filled by universe 2, which is located within the element in accordance with the transformation defined on the TR3 card. Element $[-1,-2,0]$ is filled by universe 3. Cell 7, part of universe 3, is filled by universe 5, which is also a lattice. Note the use of the X parameter to describe surface 13. The quadratic surface, which is symmetric about the x axis, is defined by specifying three coordinate pairs on the surface.

The source is a volumetric source of radius 3.6 cm which is centered in and completely surrounds cell 1. The CEL keyword causes a cell rejection technique to be used to sample uniformly throughout the cell. That is, the source is sampled uniformly in volume and any points outside cell 1 are rejected. The same effect is

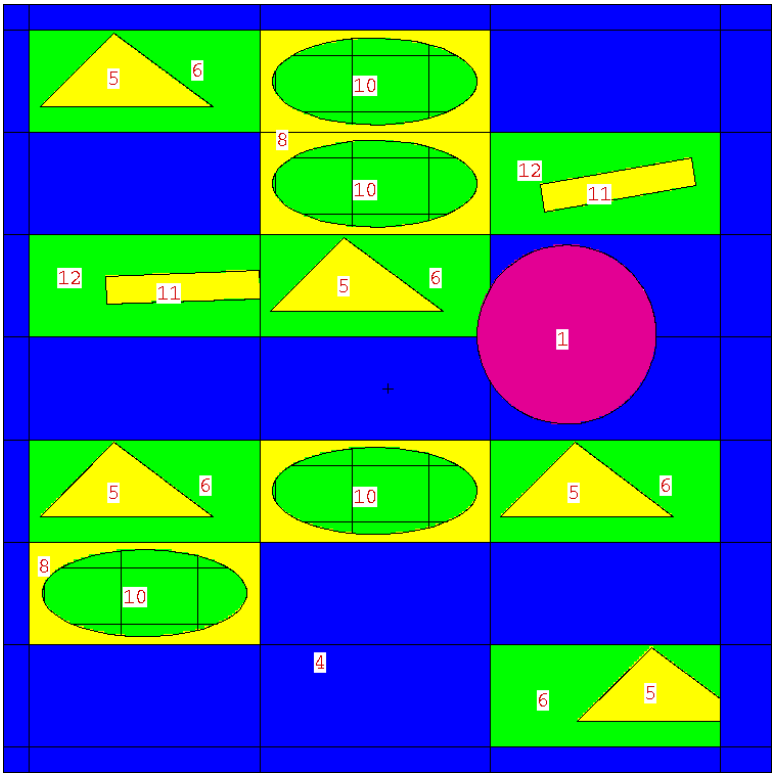


Figure 10.35: Example 21

achieved by using cookie-cutter rejection. The PRINT card results in a full output print, and the VOL card sets the volumes of all the cells to unity.

10.1.3.6 Example 22

The example shown in Listing 10.7 primarily illustrates a fairly complex source description in a lattice geometry.

Listing 10.7: example_lattice_geometry_5.mcnp.inp.txt

```
Lattice Example 22
1 0 1:-3:-4:5:6:-7 imp:n=0
2 0 -2 3 4 -5 -6 7 imp:n=1 fill=1 (-25 0 0)
3 0 -1 2 4 -5 -6 7 imp:n=1 fill=2 (0 -20 0)
4 0 -11 12 -14 13 imp:n=1 lat=1 u=1 fill=-1:1 -1:1 0:0 3 8r
5 3 -1.0 -15 2 -18 17 imp:n=1 lat=1 u=2
        fill=0:1 0:3 0:0
        4 4
        4(5 0 0) 4
        4 5
        4 4
6 1 -0.9 21:-22:-23:24 imp:n=1 u=3
7 1 -0.9 19 imp:n=1 u=4
8 2 -18 -21 22 23 -24 imp:n=1 u=3
9 1 -0.9 20(31:-32:-33:34) imp:n=1 u=5
11 2 -18 -19 imp:n=1 u=4
13 2 -18 -20 imp:n=1 u=5
```

```
15 2 -18 -31 32 33 -34 imp:n=1 u=5

1 px 50
2 px 0
3 px -50
4 py -20
5 py 20
6 pz 60
7 pz -60
11 px 8.334
12 px -8.334
13 py -6.67
14 py 6.67
15 px 25
17 py 0
18 py 10
19 c/z 10 5 3
20 c/z 10 5 3
21 px 4
22 px -4
23 py -3
24 py 3
31 px 20
32 px 16
33 py 3
34 py 6

m1 6000 0.4 8016 0.2 11023 0.2 29000 0.2
m2 92238 0.98 92235 0.02
m3 1001 1
sdef erg fcel d1 x fcel d11 y fcel d13 z fcel d15 cel d6
     rad fcel d17 ext fcel d19 pos fcel d21 axs fcel d23
ds1 s d2 d3 d4 d5
sp2 -2 1.2
sp3 -2 1.3
sp4 -2 1.4
sp5 -2 1.42
si6 s d7 d8 d9 d10
sp6 0.65 0.2 0.1 0.05
si7 l 2:4:8
sp7 1
si8 l  3:5(0 0 0):11 3:5(1 0 0):11 3:5(0 1 0):11 3:5(1 1 0):11
       3:5(0 2 0):11 3:5(0 3 0):11 3:5(1 3 0):11
sp8 1 1 1 1 1 1 1
si9 l 3:5(1 2 0):13
sp9 1
si10 l 3:5(1 2 0):15
sp10 1
ds11 s d12 0 0 d25
si12 -4 4
sp12 0 1
ds13 s d14 0 0 d26
si14 -3 3
sp14 0 1
ds15 s d16 0 0 d16
si16 -60 60
sp16 0 1
ds17 s 0 d18 d18 0
si18 0 3
```

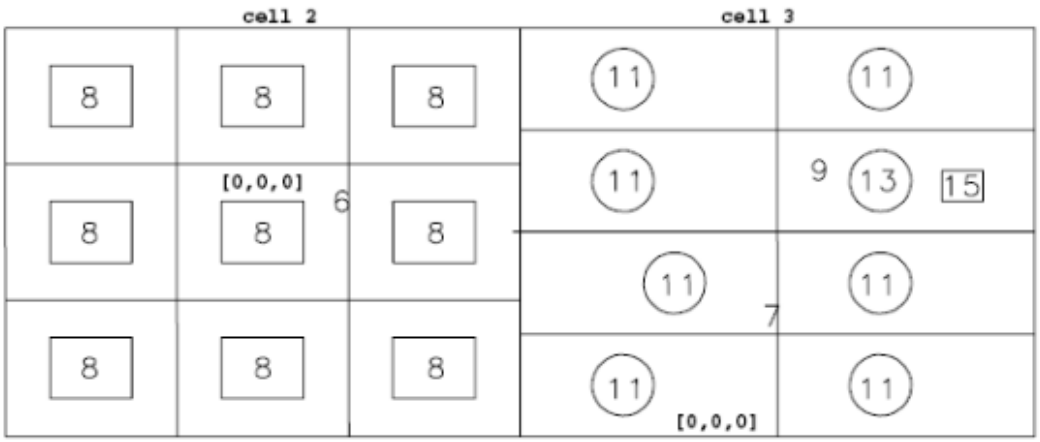


Figure 10.36: Example 22

```
sp18 -21 1
ds19 s 0 d20 d20 0
si20 -60 60
sp20 0 1
ds21 s 0 d22 d22 0
si22 l 10 5 0
sp22 1
ds23 s 0 d24 d24 0
si24 l 0 0 1
sp24 1
si25 16 20
sp25 0 1
si26 3 6
sp26 0 1
f2:n 1
e2 0.1 1 20
f6:n 2 4 6 8 3 5 7 9 11 13 15
sd6 1 1 1 1 1 1 1 1 1 1
print
nps 5000
```

The geometry consists of two “main” cells, each filled with a different lattice.

The geometry for this example is shown in Fig. 10.36.

Cell 2, the left half of Fig. 10.36, is filled with a hexahedral lattice, which is in turn filled with a universe consisting of a rectangular cell and a surrounding cell. The relationship of the origin of the filling universe, universe 1, to the filled cell, cell 2, is given by the transformation in parentheses following `FILL` = 1. The right half of Fig. 10.36, Cell 3, is filled with a different hexahedral lattice, which in turn is filled by universes 4 and 5. Lattice cells must be completely specified by an expanded `FILL` card if the lattice contains a source (cell 5) or by selecting a coordinate system of a higher level universe (`SI`7 1 -2:4:8). `PRINT` Table 110 lists the lattice elements that are being sampled.

The reader is cautioned to become familiar with the geometry before continuing with the source description that follows. In this example, a distributed volumetric source located in each of the ten boxes and eight circles (in two dimensions) is desired. The cells involved are given by distribution 6. The `S` on the `SI`6

card indicates distribution numbers will follow. The four distributions will describe the cells further. The probabilities for choosing each distribution of cells are given by the SP6 card.

The SI7 card shows the entire path from level 0 to level n for the nine boxes on the left. The expanded FILL notation is used on the cell 4 card to describe which elements of the lattice exist and which universe fills each one. All nine are filled by universe 3. The source information card SI12 indicates that x is sampled from -4 to 4; similarly, SI14 indicates that y is sampled from -3 to 3. Used together with the expanded FILL notation, source points will be sampled from all nine lattice elements. Without the expanded FILL notation, only the $[0,0,0]$ element would have source points.

Alternatively, one could use the following input cards:

```
4 0 -11 12 -14 13 imp:n=1 lat=1 u=1 fill=3
si7 l -2:4:8
si12 -46 -4
si14 -17 17
```

The minus sign in front of the second entry on the SI7 card means that the sampled position and direction will be in the coordinate system of the level preceding that entry. In this case, however, there is no preceding entry, so the position and direction will be in the coordinate system of cell 2. If a point is chosen that is not is cell 8, it is rejected and the variable is resampled.

The SI8 card describes a path from cell 3 through element $[0,0,0]$ of cell 5 to cell 11, from cell 3 through element $[1,0,0]$ of cell 5 to cell 11, and so on. Element $[1,2,0]$ is skipped and will be treated differently. The SI9 entries provide the path to cell 13, the circle in element $[1,2,0]$, while SI10 provides the path to cell 15, the box in element $[1,2,0]$. All the other source variables are given as a function of the cell and follow explanations given in §5.8.

10.1.3.7 Example 23

The example shown in Listing 10.8 illustrates a hexagonal prism lattice and shows how the order of specification of the surfaces on a cell card identifies the lattice elements beyond each surface.

Listing 10.8: example_lattice_geometry_6.mcnp.inp.txt

```
hexagonal prism lattice
1  0     -11  -19   29               fill=1           imp:n=1
2  0     -10                                    u=3 imp:n=1
3  0   -301  302 -303  305 -304  306 fill=3 lat=2 u=1 imp:n=1
4  0      11:19:-29                                 imp:n=0

11  cz   20
10  so   40
19  pz   31.75
29  pz  -31.75
301 px    1
302 px   -1
303 p     1  1.7320508076  0 2
304 p    -1  1.7320508076  0 2
305 p     1  1.7320508076 0  -2
306 p    -1  1.7320508076 0  -2

sdef
f1:n      11
nps       2000
```

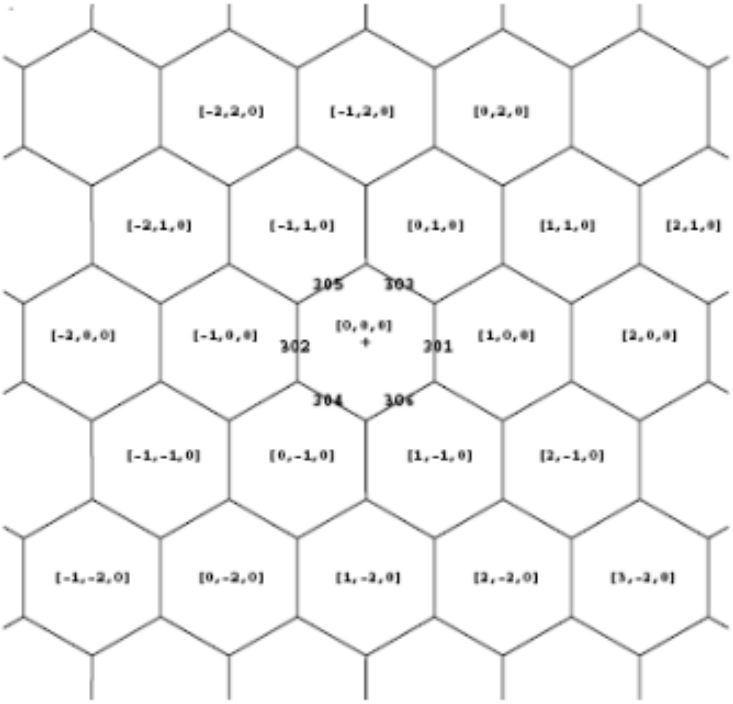


Figure 10.37: Hexagonal prism lattice.

The geometry for this example is shown in Fig. 10.37.

The $[0,0,0]$ element is the space described by the surfaces on the cell card, perhaps influenced by a TRCL entry. The user chooses where the $[0,0,0]$ element will be. The user chooses the location of the $[1,0,0]$ element—it is beyond the first surface entered on the cell card. The $[-1,0,0]$ element must be in the opposite direction from $[1,0,0]$ and must be beyond the second surface listed. The user then chooses where the $[0,1,0]$ element will be—it must be adjacent to the $[1,0,0]$ element—and that surface is listed next. The $[0,-1,0]$ element must be diagonally opposite from $[0,1,0]$ and is listed fourth. The fifth and sixth elements are defined based on the other four and must be listed in the correct order: $[-1,1,0]$ and $[1,-1,0]$. Pairs can be picked in any order, but once set the pattern must be followed. The example illustrates one pattern that could be selected and shows how the numbering of elements progresses outward from the center.

One of the most powerful uses of macrobodies is for the specification of hexagonal prisms. The example in Figure 10.37 can be simplified by using the RHP (also called HEX) macrobody as shown in Listing 10.9.

Listing 10.9: example_lattice_geometry_7.mcnp.inp.txt

```
hexagonal prism lattice using macrobodies
C Cell Cards
1  0      -2 fill=1           imp:n=1
2  0     -10              u=3 imp:n=1
3  0      -1 fill=3 lat=2 u=1 imp:n=1
4  0       2                  imp:n=0

C Surface Cards
1  rhp    0 0 -31.75  0 0 63.5  2 0 0
2  rcc    0 0 -31.75  0 0 63.5    20
10 sph    0 0 0 40

sdef
```

```
f1:n      2.1
nps       2000
```

10.1.3.8 Example 24

The example shown in Listing 10.10 demonstrates how the `LIKE n BUT` and `TRCL` cards can be used to create an array of non-identical objects within each cell of a lattice.

Listing 10.10: example_lattice_geometry_8.mcnp.inp.txt

```
Lattice/rotation example of PWR lattice
1   0        -1 -19   29      fill=1 imp:n=1
2   2        -1 -301 302 -303 304 lat=1 u=1 imp:n=1 fill=-3:3 -3:3 0:0
         1 1 1 1 1 1 1
         1 1 2 2 2 1 1
         1 2 2 2 2 2 1
         1 2 2 2 2 2 1
         1 2 2 2 2 2 1
         1 1 2 2 2 1 1
         1 1 1 1 1 1 1
3   1 -18   -10               u=2 imp:n=1
4   2 -1     #3 #5 #6 #7 #8 #9 #10 #11 #12 #13 #14 #15 #16 #17 #18
             #19 #20 #21 #22 #23 #24 #25 #26 #27 #28   imp:n=1 u=2
5  like  3 but trcl=(-6  6 0)
6  like  3 but trcl=(-3  6 0)
7  like  3 but trcl=( 0  6 0)
8  like  3 but trcl=( 3  6 0)
9  like  3 but trcl=( 6  6 0)
10 like  3 but trcl=(-6  3 0)
11 like  3 but trcl=( 0  3 0)
12 like  3 but trcl=( 6  3 0)
13 like  3 but trcl=(-6  0 0)
14 like  3 but trcl=(-3  0 0)
15 like  3 but trcl=( 3  0 0)
16 like  3 but trcl=( 6  0 0)
17 like  3 but trcl=(-6 -3 0)
18 like  3 but trcl=( 0 -3 0)
19 like  3 but trcl=( 6 -3 0)
20 like  3 but trcl=(-6 -6 0)
21 like  3 but trcl=(-3 -6 0)
22 like  3 but trcl=( 0 -6 0)
23 like  3 but trcl=( 3 -6 0)
24 like  3 but trcl=( 6 -6 0)
25 like  3 but trcl=(-3  3 0) mat=3 rho=-9
26 like 25 but trcl=( 3  3 0)
27 like 25 but trcl=(-3 -3 0)
28 like 25 but trcl=( 3 -3 0)
50  0         1:19:-29        imp:n=0

1   cz    60
10  cz     1.4
19  pz    60
29  pz   -60
301 px    10
302 px   -10
303 py    10
304 py   -10
```

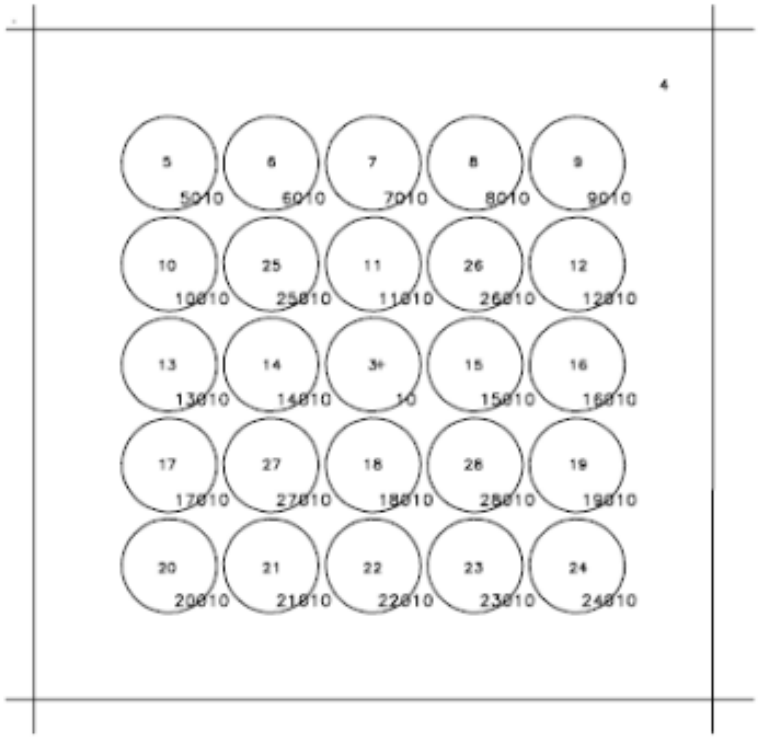


Figure 10.38: Example 24

```
48
49 kcode   1000 1 5 10
50 ksrc    0 0 0
51 m1      92235  0.02   92238  0.98
52 m2       1001  2       8016  1
53 m3      48000  1
```

A horizontal slice through this configuration is shown in Fig. 10.38.

Only one lattice element is shown in Fig. 10.38. A lattice of hexahedral subassemblies, each holding an array of 25 cylindrical rods, is contained within a cylindrical cell. Cell 1, the space inside the large cylinder, is filled with universe 1. Cell 2 is the only cell in universe 1 and is the hexahedral lattice that fills cell 1. The lattice is a $7 \times 7 \times 1$ array, indicated by the array indices on the `FILL` card, and it is filled either by universe 2 or by itself (that is, universe 1). Cell 3, a fuel rod, is in universe 2 and is the space inside the cylindrical rod. The other fuel cells, 5–24, are like cell 3 but at different x, y locations. The material in these 21 fuel cells is slightly enriched uranium. Cells 25–28 are control rods. Cell 25 is like 3 but the material is changed to cadmium, and the density and the x, y location are different. Cells 26–28 are like cell 25 but at different x, y locations. Cell 4 is also in universe 2 and is the space outside all 25 rods. To describe cell 4, each cell number is complemented. All the surfaces in Fig. 10.38 except for the center one have a new predictable surface number equal to $1000 \times$ (cell number) + (surface number). These numbers could be used in the description of cell 4 if desired.

The `KCODE` and `KSRC` cards appear because this example is a criticality calculation. The `KCODE` card specifies that there are 1000 particles per cycle, the initial guess for k_{eff} is 1, 5 cycles are skipped before the tally accumulation begins, and a total of 10 cycles will be run. The `KSRC` indicates that the neutron source for the first cycle will be a point source at the origin.

10.1.4 Embedded Meshes: Structured and Unstructured

In the following example, we first create a structured PARTISN-style geometry mesh and save it in LNK3DNT format. The cylindrical mesh consists of two materials in a checkerboard pattern that appears radially, axially, and azimuthally. After the LNK3DNT-format mesh file is created, we then embed the mesh in a new MCNP6 file.

10.1.4.1 Example 25 (Part 1)

In the example shown in Listing 10.11, the `MESH` and `DAWWG` cards specify a cylindrical geometry with diameter and length of 20 cm.

Listing 10.11: example_structured_mesh_generate_2.mcnp.inp.txt

```
Generate a LNK3DNT rzt mesh w/ multiple materials
c upper-inner
1   1   -18.7         -11  1  2  3   imp:n=1
2   2   -0.001        -11  1 -2  3   imp:n=1
3   1   -18.7         -11 -1 -2  3   imp:n=1
4   2   -0.001        -11 -1  2  3   imp:n=1
c upper-outer
6   2   -0.001     -10 11  1  2  3   imp:n=1
7   1   -18.7      -10 11  1 -2  3   imp:n=1
8   2   -0.001     -10 11 -1 -2  3   imp:n=1
9   1   -18.7      -10 11 -1  2  3   imp:n=1
c lower-inner
11  2   -0.001        -11  1  2 -3   imp:n=1
12  1   -18.7         -11  1 -2 -3   imp:n=1
13  2   -0.001        -11 -1 -2 -3   imp:n=1
14  1   -18.7         -11 -1  2 -3   imp:n=1
c lower-outer
16  1   -18.7      -10 11  1  2 -3   imp:n=1
17  2   -0.001     -10 11  1 -2 -3   imp:n=1
18  1   -18.7      -10 11 -1 -2 -3   imp:n=1
19  2   -0.001     -10 11 -1  2 -3   imp:n=1
c
c outer void
20          0           10            imp:n=0

10  rcc    0. 0. -10.  0. 0. 20.  10.  $ outer rcc
11  rcc    0  0  -10   0  0  20    5   $ inner rcc
1   py     0.0
2   px     0.0
3   pz     0.0

kcode     5000  1.0  50  250
ksrc      0.0 0.0 0.0
m1        92235.69c   1.0
m2         6012       1.0
dm1       92235 92235.50
mesh  geom cyl
      ref     0.0   0.0   0.0
      origin  0.0   0.0 -10.0  $ bottom center of cylinder
      axs     0.0   0.0   1.0
      vec     1.0   0.0   0.0
      imesh   10  $ cylinder radius
      iints    2  $ 2 radial divisions
```

```
      jmesh   20  $ axial (z) length
      jints    2  $ 2 axial divisions
      kmesh    1  $ azimuth-single rotation (0-2pi)
      kints    4  $ 4 azimuthal divisions (0, pi/2, pi, 3pi/2, 2pi)
dawwg   xsec=ndilib  points=10
```

The cylinder mesh has two radial, two axial, and four azimuthal divisions, creating a total of eight mesh elements. The materials in each of the elements alternate, creating a checkerboard-like pattern throughout the cylinder. The use of the `MESH` keywords `ORIGIN`, `AXS`, and `VEC` ensure that the mesh aligns with the geometry—the bottom center of the mesh at $(0, 0, -10)$, the cylinder oriented along the z axis, and the azimuthal plane along the positive x axis. To create the LNK3DNT file, run MCNP6 with the `M` execution-line option using Listing 10.11 as the MCNP6 input and assign the **`LINKOUT`** file the arbitrary name **`cyl.linkout`**.

10.1.4.2 Example 25 (Part 2)

Now we embed the mesh geometry into the MCNP6 input in Listing 10.12 using inferred geometry cells (one for each material in the **`cyl.linkout`** file) and one inferred background cell.

Listing 10.12: example_structured_mesh_read_2.mcnp.inp.txt

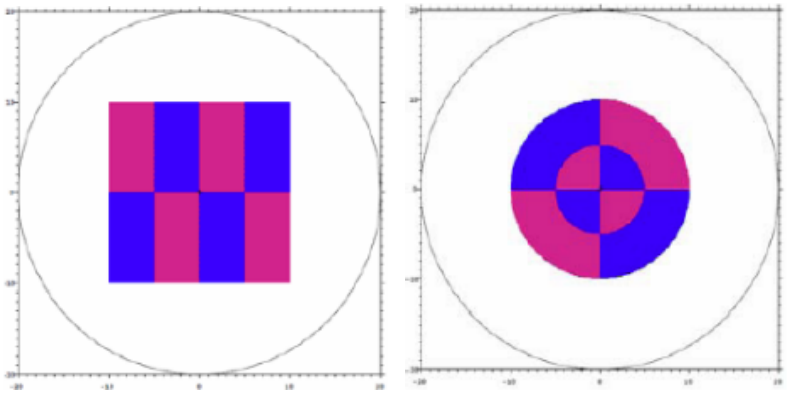
```
RZT Test of checkerboard cylinder with lnk3dnt
11  3  -18.7   0         u=e10 imp:n=1  $ inferred geometry cell
12  4  -0.001  0         u=e10 imp:n=1  $ inferred geometry cell
13  0          0         u=e10 imp:n=1  $ inferred background cell
20             0  -1  fill=e10 imp:n=1  $ embedded mesh fill cell
99  0          1               imp:n=0  $ outside world

1  so   20

kcode     500     1.0  50   100
ksrc 1 1  5 1 -1  5 -1 -1  5 -1 1  5
     5 5  5 5 -5  5 -5 -5  5 -5 5  5
     1 1 -5 1 -1 -5 -1 -1 -5 -1 1 -5
     5 5 -5 5 -5 -5 -5 -5 -5 5 -5
m3        92235.69c   1.0
m4        6012 1.0
dm1 92235 92235.50
embed10 meshgeo=lnk3dnt mgeoin=cyl.linkout debug=echomesh
        matcell= 1 11  2 12
        background=13
```

Note that inferred geometry cell 11 maps to mesh material 1, inferred geometry cell 12 maps to mesh material 2, and the inferred background cell 13 completes the embedded mesh universe by defining the space surrounding the mesh. The embedded mesh universe then fills cell 20 of the MCNP6 model. Recall that the "`e`" is optional for the `U` and `FILL` keywords to denote an embedded mesh.

Figure 10.39 shows two views of the resulting geometry with the embedded geometry shaded by material, which in this case alternates between geometry elements.

10.1.4.3 Example 25 (Part 3)

Now let's assume we want two copies of this mesh geometry embedded into our MCNP6 model, each having a different placement and orientation. We need to rotate/translate the two mesh geometry universes appropriately as we fill two distinct MCNP6 cells.



(a) y-z View (b) x-y View

Figure 10.39: Two MCNP6 geometry plots of a cylindrical (r, z, θ) embedded geometry. In each plot, the remaining axis points toward the reader.

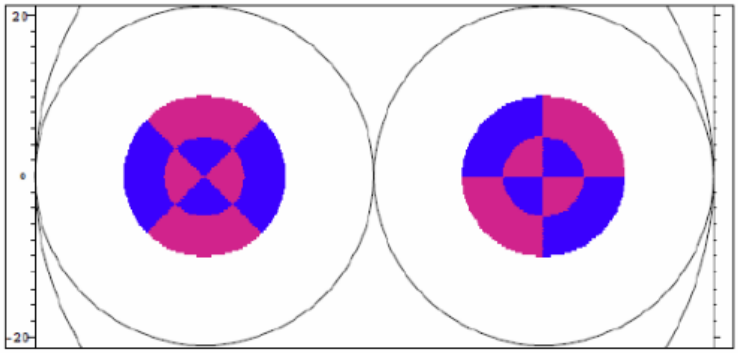


Figure 10.40: (Cropped) MCNP6 geometry plot of an embedded structured mesh placed in two unique containers. The left instance is rotated 45° in addition to being translated 20 cm in the $-x$ direction.

```
RZT Test of two checkerboard cylinders with lnk3dnt
11  3  -18.7   0     u=e10 imp:n=1  $ inferred geometry cell
12  4  -0.001  0     u=e10 imp:n=1  $ inferred geometry cell
13  0          0     u=e10 imp:n=1  $ inferred background cell
20  0         -1  fill=e10 (20 0 0) imp:n=1  $ fill cell 1
21  0         -2  fill=e10 (-20 0 0 -1 1 0 1 1 0)imp:n=1  $ fill cell 2
```

For this example, we assume that surfaces 1 and 2 are off-center spheres and the transformations shift the embedded geometry universes to be aligned with the spheres' geometric centers ($x = \pm 20$ cm). With the addition of a 45° counter-clockwise rotation applied to one of the embedded meshes, we get the geometry displayed in Figure 4-40.

10.2 Tally Examples

This section contains examples of the FM, FS, and FT tally cards, a complicated repeated structures/lattice example, and the **TALLYX** subroutine. Refer also to §5.9.1.5 for the basic repeated structure/lattice tally, and §5.9.17 for **TALLYX** before trying to understand these examples.

10.2.1 FM Card Examples (Simple Form)

10.2.1.1 Example 26

Consider input file shown in Listing 10.13.

Listing 10.13: example_tally_multiplier_1.mcnp.inp.txt

```
Tally Multiplier (FM)
10  999 -7.0   -1    imp:n=1
11   0          1 -2 imp:n=1
12   0             2 imp:n=0

1 so   5
2 so   6

sdef
f1:n      1
nps       2000
f4:n      10
fm4       0.04786   999   102
m999      92238.80c  1.0
```

The F4 neutron tally is the track length estimate of the average fluence in cell 10. Material 999 is ^{238}U with an atomic fraction of 100%.

c=0.04786	is a normalization factor (such as atom/barn-cm),
m=999	is the material number for ^{238}U as defined on the material card (with an atom density of 0.04786 atom/barn-cm), and
r_1=102	is the ENDF reaction number for radiative capture cross-section (microscopic).

The average fluence is multiplied by the microscopic (n,γ) cross section of ^{238}U (with an atomic fraction of 1.0) and then by the constant 0.04786 (atom/barn-cm). Thus the tally 4 printout will indicate the number of ^{239}U atoms/cm^3 produced as a result of (n,γ) capture with ^{238}U.

Standard F6 and F7 tallies can be duplicated by F4 tallies with appropriate FM4 cards. The FM4 card to duplicate F6 is

```
FM4    c  m   1  -4
```

where

c	is $10^{-24}\times$ number of atoms per gram,
r_1=1	is the ENDF reaction number for total cross section (barns), and
r_2=-4	is the reaction number for average heating number (MeV/collision)

and for F7 it is

```
FM4    c  m  -6  -8
```

where

c	is $10^{-24}\times$ number of atoms per gram,
r_1=-6	is the reaction number for total fission cross section (barns), and
r_2=-8	is the reaction number for fission Q (MeV/fission).

This technique applied to F2 tallies can be used to estimate the average heating over a surface rather than over a volume. It provides the surface equivalents of F6 and F7 tallies, which are not available as standard tallies in MCNP6.

10.2.1.2 Example 27

Consider the MCNP input in Listing 10.14, which contains a point detector.

Listing 10.14: example_point_detector_1.mcnp.inp.txt

```
Point Detector Tally
10  999  -1.0   -1    imp:n=1
11  1001 -5.0    1 -2 imp:n=1
12   0              2 imp:n=0

1 so   5
2 so   6

sdef
f1:n     1
nps      2000
m999     1001.80c 2   8016.80c 1
F25:N    0  0  0  0
FM25     0.00253  1001  -6  -8
M1001    92238.80c  0.9   92235.80c  0.1
```

This F25 neutron tally is the fission heating per unit volume of material 1001 at the origin. Material 1001 does not actually have to be in a cell at the origin. The FM25 card constants are:

$c = 0.00253$	atoms per barn-cm (atomic density) of material 1001,
$m = 1001$	is the material number for material being heated,
$r1 = -6$	is the reaction number for total fission cross section (barn), and
$r2 = -8$	is the reaction number for fission Q (MeV/fission).

Other frequently used FM card examples are shown in Listing 10.15.

Listing 10.15: example_fm.mcnp.inp.txt

```
f05:n 6 1 0 1                    $ Neutron heating per cm^3 of silicon with an atom
fm05 4.28836E-02 20 1 -4         $ density of 4.28836E-02 bn^-1*cm^-1 at a point detector.
c
f15x:p 6 1 1                     $ Photon heating per cm^3 of silicon with an atom
fm15 4.28836E-02 20 -5 -6        $ density of 4.28836E-02 bn^-1*cm^-1 at a ring detector.
c
f1:n 1 2 3                       $ Number of neutron **tracks** crossing surfaces 1, 2,
fm1 1 0                          $ and 3 per neutron started.
c
f35:p 6 1 0 1                    $ Number of photon collisions per source particle that
fm35 1 0                         $ contribute to the associated point detector.
c
m99 5011 1                       $ Number of (n,p) reactions in B-11 per cm^3 in cell 200.
f4:n 200                         $ Atom density assumed based on a mass density of 1 g/cm^3.
fm4 5.570369e-02 99 103
c
f104:p 200                       $ Number of pair-production reactions (mt=516) in
fm104 -1 10 516                  $ iron per cm^3 in cell 100.
c
m10   26056  1.0                 $ Pure Fe-56 for transport; density: 8 g/cm^3 assumed.
c
m20   14028  9.222300e-01        $ Natural silicon for tallying.
      14029  4.685000e-02        $ Composition from 'mattool -a 14000 1'.
      14030  3.092000e-02        $ Density: 2 g/cm^3 assumed.
```

10.2.2 FM Examples (General Form)

Remember that the hierarchy of operation is multiply first and then add, and that this hierarchy can not be superseded by the use of parentheses.

10.2.2.1 Example 28

```
F4:N   1
FM4    ( 1 (1 -4)(-2)) ( 1 1) $ where c==atomic density (atom/barn-cm)
M1     6012.10   1
```

In this example there are three different tallies, namely

1. ρ 1 1 -4
2. ρ 1 -2
3. ρ 1 1

Thus tally (1) will yield the neutron heating in MeV/cm^3 from ^{12}C in cell 11. The advantage in performing the multiplication $1 \ -4$ in tally (1) is that the correct statistics are determined for the desired product. This would not be true if tally (1) were to be done as two separate tallies and the product formed by hand after the calculation.

10.2.2.2 Example 29

In the example shown in Listing 10.16, one can obtain the total tritium production per cm^3 from natural lithium (ENDF/B-VII.1 evaluation) in cell 11.

Listing 10.16: example_tally_multiplier_2.mcnp.inp.txt

```
Tally Multiplier (FM): Total Tritium Production
10   999  -1.0     -1    imp:n=1 $ Water
11  1001  -1.       1 -2 imp:n=1 $ Li
12  0                  2 imp:n=0

1 so 5
2 so 6

sdef
nps       2000
M999     1001.80c 2       8016.80c 1
M1001    3006.80c 0.0742  3007.80c 0.9258
F4:N    11
FM4     0.04635 1001 205
```

The constant `c` on the `FM`4 card is the atomic density of natural lithium. The reaction number 205 specifies the total tritium production cross section.

Using older ENDF/B-V evaluated data such as in [343, p. 4-41], one could specify this reaction as the sum of reactions 105 and 91. Reaction 105 gives tritium production from ^{6}Li while reaction 91 in ENDF/B-V represents ^{7}Li(n,n')t, which in modern data would be the sum of reaction numbers 52 through 82.

10.2.2.3 Example 30

Suppose we have three reactions: r_1, r_2, and r_3, and we wish to add r_2 and r_3 and multiply the result by r_1. The following would not be valid: `FM`n (C m r_1 (r_2:r_3)). The correct card is: `FM`n (C m (r_1 r_2:r_1 r_3)).

10.2.3 FMESH Isotopic Reaction Rate Tally Examples

The `FMESH` card allows the user to calculate isotopic reaction rates on an arbitrary, user-specified mesh that is independent of the actual problem geometry [§5.11.2.2].

10.2.3.1 Example 31

In the input file shown in Listing 10.17, there are two cells: one composed of natural uranium and the other as depleted uranium.

Listing 10.17: example_fmesh_tally_1.mcnp.inp.txt

```
FMESH tally example
c Cells
900 100 -19.1  -1    imp:n=1 $ Natural Uranium
901 200 -19.1  -2    imp:n=1 $ Depleted Uranium
902 300 -0.001 1 2 -3 imp:n=1 $  air
903  0         3     imp:n=0 $  Void, kill n
```

```
7
8  c Surfaces
9  1 sx 4  3
10 2 sx -4 3
11 3 so 10
12
13 sdef erg=2
14 mode n
15 nps 500000
16 c
17 c Problem materials
18 c Natural Uranium
19 m100   92238  0.992745
20        92235  0.007200
21 c Hypothetical Depleted Uranium
22 m200   92238  0.9999
23        92235  0.0001
24 c Air
25 m300 7014 -0.755  8016 -0.231  18000 -0.013
26 c Dummy materials for FM mesh tallies
27 m238 92238  1.0
28 m235 92235  1.0
29 c
30 fmesh04:n  geom=xyz origin -10 -10 -10
31            imesh 10  iints 100
32            jmesh 10  jints 100
33            kmesh 10  kints 100
34            out=none
35 fmesh14:n  geom=xyz origin -10 -10 -10
36            imesh 10  iints 100
37            jmesh 10  jints 100
38            kmesh 10  kints 100
39            out=none
40 fmesh24:n  geom=xyz origin -10 -10 -10
41            imesh 10  iints 100
42            jmesh 10  jints 100
43            kmesh 10  kints 100
44            out=none
45 c Tally multipliers
46 +fm04  -1  235  -6   $ fission rate per cm3 from U235
47 +fm14  -1  238  -6   $ fission rate per cm3 from U238
48 +fm24  -1  100  -6   $ total fission rate from both U235 and U238
```

To calculate the fission rates for each isotope in both cells, a mesh tally is used. The default units of the results are the (number of fissions) $\cdot$ cm^{-3}(or cm$^{-3} \cdot$ shake^{-1}) in each mesh cell. For tally 24, material 200 could be used instead of material 100 because both materials contain the same isotopes.

10.2.3.2 Example 32

The input file shown in Listing 10.18 contains a single cell composed of concrete.

Listing 10.18: example_fmesh_tally_2.mcnp.inp.txt

```
1 FMESH tally example
2 c Cells
3 900 10 -2.5    -1    imp:n=1 $ Concrete
```

```
901 11 -7.86   -2    imp:n=1 $ Stainless Steel - 202
902 12 -0.0012 1 2 -3 imp:n=1 $  Void, transport
903  0          3    imp:n=0 $  Void, kill n

c Surfaces
1 sx 6 3
2 sx -6 3
3 so 10

sdef erg=2
mode n
nps 500000
c
c Problem materials
c Ordinary Concrete (rho = 2.35 g/cc)
m10    1001  -0.00600  8016  -0.50000 11023  -0.01700
      13027  -0.04800 14028  -0.28940 14029  -0.01518
      14030  -0.01042 19000  -0.01900 20000  -0.08300
      26054  -0.00068 26056  -0.01106 26057  -0.00026
c Stainless Steel - 202
m11    6000  -0.00075  7014  -0.00125 14000  -0.00500
      15031  -0.00030 16000  -0.00015 24000  -0.18000
      25055  -0.08750 26000  -0.67505 28000  -0.05000
m12    7014  -0.755    8016  -0.232   18000  -0.013
m20   11023   1
m21   26054   1
m22   25055   1
c
fmesh04:n  geom=xyz origin -10 -10 -10
           imesh 10  iints 50
           jmesh 10  jints 50
           kmesh 10  kints 50
           out=none
fmesh14:n  geom=xyz origin -10 -10 -10
           imesh 10  iints 50
           jmesh 10  jints 50
           kmesh 10  kints 50
           out=none
fmesh24:n  geom=xyz origin -10 -10 -10
           imesh 10  iints 50
           jmesh 10  jints 50
           kmesh 10  kints 50
           out=none
c
C 102 = (n,gamma) reaction
+fm4   -1  20  102   $ Na-24 production (not in material 11)
+fm14  -1  21  102   $ Fe-55 production (2600 in material 11)
+fm24  -1  22  102   $ Mn-56 production (not in material 10)
```

We want to calculate the production rate of ^{24}Na and ^{55}Fe in the material. The ^{23}Na and ^{54}Fe isotopes are specified on the dummy material cards because an (n,γ) reaction on these isotopes produce ^{24}Na and ^{55}Fe, respectively. The production rate is calculated by multiplying the (n,γ) reaction cross section times the atomic fraction of the isotope in material 10.

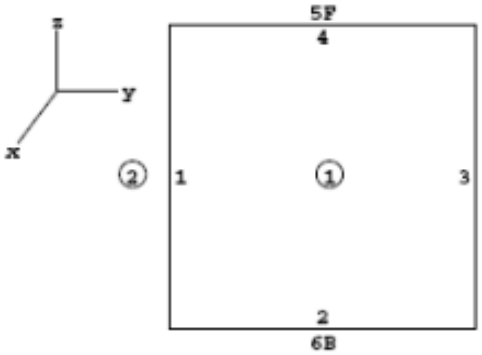


Figure 10.41: Example 33

10.2.4 FS Card Examples

The FS card allows you to subdivide your tally into geometry segments, avoiding over-specifying the problem geometry with unnecessary cells.

The entries on the FS card are the names and senses of surfaces that define how to segment any surface or cell tally.

10.2.4.1 Example 33

Consider a 1-MeV point isotropic source at the center of a 2-cm cube of carbon as shown in Listing 10.19.

Listing 10.19: example_tally_segment_1.mcnp.inp.txt

```
Tally segmenting example
1  1 -2.22   1  2 -3 -4 -5  6 imp:n=1
2  0        #1                imp:n=0

1  py    0
2  pz   -1
3  py    2
4  pz    1
5  px    1
6  px   -1

sdef    pos = 0 1 0  erg = 1
m1      6012 -1
f2:n    3
nps     100
```

We wish to calculate the flux through a 1-cm^2 window in the center of one face on the cube. The input file calculating the flux across one entire face is shown in Fig. 10.41.

The FS card retains the simple cube geometry and four more surface cards are required,

```
7  PX    0.5
8  PX   -0.5
```

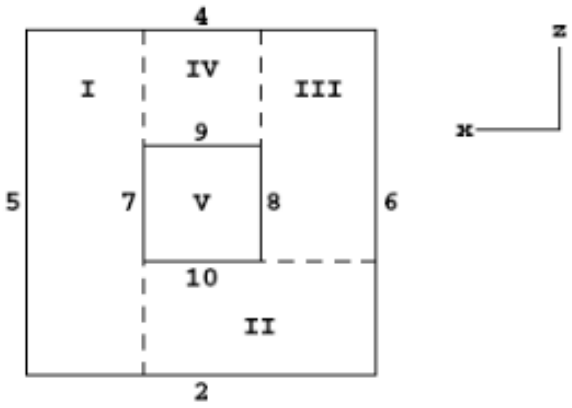


Figure 10.42: Example 33

```
9  PZ    0.5
10 PZ   -0.5

FS2     7 -10 -8  9
```

The four segmenting surface cards are listed with the other surface cards, but they are not part of the actual geometry and hence do not complicate the cell-surface relationships.

The F2 tally is subdivided into five separate tallies as shown in Fig. 10.42: 1) the first is the flux of particles crossing surface 3 but with a positive sense to surface 7; 2) the second is the remaining flux with negative sense to surface 7 crossing surface 3 but with a negative sense to surface 10; 3) the third is the remaining flux (negative sense to 7 and positive sense to 10) crossing 3 but with a negative sense to 8; 4) the remaining flux with positive sense to 9; and 5) everything else. In this example, the desired flux in the window is in the fifth sub-tally—the "everything else" portion.

The FS segmenting card could have been set up other ways. For example:

```
FS2   -10  7   9  -8
```

and

```
FS2    -8  9 -10   7
```

Each works, but the order of the sub-tallies is changed. A way to avoid the five sub-tallies and to get only the window of interest is to use the **TALLYX** subroutine [§5.9.17, §10.2.8].

10.2.4.2 Example 34

Consider a source at the center of a 10-cm-radius sphere called cell 1. We want to determine the fission heating in a segment of the sphere defined by the intersection of the 10-cm sphere, an 8-cm inner sphere, and a 20-degree cone (i.e., the angle between the axis and surface of the cone is 20°) whose vertex is at the source and is about the y axis. This is accomplished by using

```
F7:N     1
FS7     -2  -3
```

where surface 2 is the 8-cm sphere and surface 3 is the cone. This breaks the F7 tally up into three portions:

1. the heating inside the 8-cm sphere;
2. the heating outside the 8-cm sphere but within the cone—this is the desired portion; and
3. everything else, which is a 2-cm shell just inside the 10-cm sphere but outside the cone.

10.2.5 FT Examples

10.2.5.1 Example 35

Consider the following input cards.

```
F1:N    2
FT1     FRV v1 v2 v3
```

The FTn card is the special treatment for tallies card. Various tally treatments are available for certain specific tally requirements. The FTn tally with the FRV keyword used in conjunction with tally type 1 will redefine the vector normal to the tally surface. In this case, the current over surface 2 (tally type 1) uses the vector $(v1, v2, v3)$ as its reference vector for getting the cosine for binning.

10.2.5.2 Example 36

```
F5:P     4 5 6
FT5      ICD
FU5      1  3
```

In this example the photon flux at detector 5 is being tallied. However, only the contributions to the detector tally from cells 1 and 3 are of interest. The ICD keyword allows the user to create a separate bin for each cell, and only contributions from one of the specified cells are scored. The FUn card specifies the cells from which tallies are to be made, but **TALLYX** is not called.

10.2.5.3 Example 37

When keeping track of charged particle current across a surface, it is sometimes desirable to track both positive and negative score contributions, applicable in cases that include charged particles. Consider a photon source that is enclosed in a spherical shell of lead as shown in Listing 10.20.

Listing 10.20: example_tally_electron_current.mcnp.inp.txt

```
electron current example
1  1  -0.001124       -11   imp:e=1 imp:p=1
2  2 -11.0         11 -21   imp:e=1 imp:p=1
3  0               21       imp:e=0 imp:p=0

11 so    30
21 so    32

m1    6012  0.000125   7014  0.6869   8016  0.301248   18040  0.011717
m2   82000  1.
mode   p e
sdef   pos = 0. 0. 0.  erg = 2.5
f1:e   21
ft1    elc  2
f2:p   21
e2     1e-3 1e-2 0.1 0.5 1.0 1.5 2.0 2.5 C
nps    10000
```

If a surface current tally is taken over the sphere and it is desirable to tally both the positron and electron current separately, then the special treatment card option is invoked.

The input deck shown in Listing 10.20 models a sphere filled with dry air surrounded by a spherical shell of lead. The centrally located source emits 2.5-MeV photons that travel through the air into the lead shell. The F1 surface current tally has been modified with the ELC special tally option. The parameter value of 2 that follows the ELC keyword specifies that positrons and electrons be placed into separate tally user bins. Once this option has been invoked, the user can inspect the output tally bins for the respective scoring of either particle.

The F2 tally scores photon flux crossing surface 21, scored into energy bins defined on the E2 card. The C at the end of the energy bin card indicates that the bins are cumulative. For instance, the bin with an upper limit of 1 MeV would contain scores from particles that cross surface 21 with energy less than or equal to 1 MeV.

10.2.5.4 Example 38

Consider the following two point sources, each with a different energy distribution:

```
sdef   pos=d1  erg=fpos d2
si1  L   5 3 6  75 3 6
sp1      0.3    0.7
ds2  S   3 4
si3  H   2 10 14
sp3  D   0  1  2
si4  H   5 2 8
sp4  D   0 3 1
f2:n     2
ft2      scd
fu2      3 4
```

The SCD option causes tallies to be binned according to which source distribution was sampled. The FUn card is used to list the distribution numbers of interest. Thus, the tallies in this example are placed in one of

two bins, depending on which of the two sources emitted the particle. The two sources may represent two nuclides with different energy distributions. In this case use of the SCD option allows the user to determine each nuclide's contribution to the final tally.

10.2.5.5 Example 39: Capture Tallies: Interpreting Capture Tally Output

The FT8 CAP coincidence capture tally option produces both a standard tally, which is generally unreadable, and a coincidence capture table, PRINT Table 118. An example is provided to help in the interpretation of this table:

```
neutron captures, moments & multiplicity distributions. tally 8          print table 118

 cell:    999

 neutron captures on 3he
                      captures      captures    multiplicity fractions
         histories   by number     by weight    by number    by weight   error

 captures =  0     700       0   0.00000E+00  7.00000E-02  3.25400E-02  0.0364
 captures =  1    2285    2285   1.06220E-01  2.28500E-01  1.06220E-01  0.0184
 captures =  2    3223    6446   2.99647E-01  3.22300E-01  1.49823E-01  0.0145
 captures =  3    2489    7467   3.47109E-01  2.48900E-01  1.15703E-01  0.0174
 captures =  4    1022    4088   1.90033E-01  1.02200E-01  4.75084E-02  0.0296
 captures =  5     209    1045   4.85775E-02  2.09000E-02  9.71551E-03  0.0684
 captures =  6      51     306   1.42246E-02  5.10000E-03  2.37077E-03  0.1397
 captures =  7      12      84   3.90480E-03  1.20000E-03  5.57828E-04  0.2885
 captures >  7       9      73   3.39345E-03  9.00000E-04  4.18371E-04  0.3332

    total        10000   21794   1.01311E+00  1.00000E+00  4.64857E-01  0.0056

      factorial moments              by number               by weight

            3he                2.17940E+00 0.0056      1.01311E+00 0.0056
       3he(3he-1)/2!           2.01890E+00 0.0128      9.38499E-01 0.0128
    3he(3he-1)(3he-2)/3!       1.06390E+00 0.0291      4.94561E-01 0.0291
  3he(3he-1) .... (3he-3)/4!   3.93800E-01 0.0744      1.83061E-01 0.0744
  3he(3he-1) .... (3he-4)/5!   1.34100E-01 0.1636      6.23373E-02 0.1636
  3he(3he-1) .... (3he-5)/6!   4.43000E-02 0.2666      2.05932E-02 0.2666
  3he(3he-1) .... (3he-6)/7!   1.12000E-02 0.3808      5.20640E-03 0.3808
  3he(3he-1) .... (3he-7)/8!   1.70000E-03 0.5548      7.90257E-04 0.5548
```

The capture tally input for this problem was

```
F8:n      999              $ input F8 card
FT8 CAP    -8   -8   2003   $ input FT8 CAP card
```

Note that the line "captures > 7" indicates that nine histories had eight or more neutrons captured. This implies that 8 histories had $8 \times 8 = 64$ neutrons captured and 1 history had 1×9 neutrons captured, for a total of 73 neutrons captured. The table of captures evidently was too short, and the problem should have been run with FT8 CAP -9 -9 or even more captures and moments. Not specifying enough capture rows affects only the captures > 7 lines and the error estimate on the totals capture line; all other information is correct as if more captures and moments were listed.

As an interpretation of the neutron captures on 3he portion of the table, Column 1 is the number of histories according to the number of captures by the designated material (2003 =^{3}He) in the designated cell (999). This number sums to the total number of source histories for the problem, NPS 10000.

Column 2 is the number of captures by ^{3}He in cell 999=21794. Because analog capture is the default for F8 tallies, the total weight captured is also 21794.

Column 3 is the total weight captured divided by the tally normalization. In this problem, SDEF PAR = SF, and the tally normalization is the source particles = spontaneous fission neutrons = 21512. Thus, captures by weight are 21794.0/21512 = 1.01311.

Column 4 is the multiplicity fraction by number, which is Column 1 divided by the number of source histories. The total is always 1.00000.

Column 5 is the multiplicity fraction by weight, which is the weight of histories undergoing capture divided by the tally normalization. In this problem, SDEF PAR = SF and the multiplicity fraction by weight is 10000.0/21512 = 0.464857.

Note that for SDEF PAR = -SF, the tally normalization is the number of source histories = number of spontaneous fissions = 10000. Therefore, for SDEF PAR = -SF, all of the columns that are labeled by weight would be consistent with the values reported in the columns labeled by number in both the neutron capture on 3he and factorial moments sections of PRINT Table 118. For example, if SDEF PAR = -SF, then column 3 would be 21794.0/10000 = 2.17940, and column 5 would be 10000.0/10000 = 1.00000 in the neutron capture on 3he portion of the table.

The interpretation of the factorial moments portion of the table now follows.

The first moment by number is the number of captures divided by the number of source histories = 21794/10000 = 2.17940.

The first moment by weight is the total weight of capture divided by the tally normalization. In this problem, SDEF PAR = SF and the first moment by weight is 21794.0/21512 = 1.01311.

The second moment is $N \times (N-1)/2$, where N is the number of captures. In this problem,

N	$N \times (N-1)/2$		histories		
1	0	×	2285	=	0
2	1	×	3223	=	3223
3	3	×	2489	=	7467
4	6	×	1022	=	6132
5	10	×	209	=	2090
6	15	×	51	=	765
7	21	×	12	=	252
8	28	×	8	=	224
9	36	×	1	=	36
Total					20189

and the second moment by number is divided by the number of histories, 20189/10000 = 2.01890.

Because of analog capture, the second moment weight is 20189.0. The second moment by weight is divided by the tally normalization. In this problem, SDEF PAR = SF, and the second moment by weight is 20189.0/21512 = 0.938499.

The seventh moment is

$7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1/7!$	=	1	×	12	12
$8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2/7!$	=	8	×	8	64
$9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3/7!$	=	36	×	1	36
Total					112

thus, $112/10000 = 0.0112$.

The eighth moment is

$8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1/8!$	=	1	×	8	8
$9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2/8!$	=	9	×	1	9
Total					17

thus, $17/10000 = 0.0017$.

And the ninth moment is

$9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1/9!$	=	1	×	1	1

thus, $1/10000 = 0.0001$.

10.2.5.6 Example 40: Capture Tallies with Time Gating

The coincidence capture tally optionally allows specification of predelay and gate width [317] with the `GATE` keyword on the `FT8` card. The `GATE` keyword may appear anywhere after the `CAP` keyword and is part of the `CAP` command. Immediately following, the `GATE` keyword must be the predelay time and the total gate width, both in units of shakes (10^{-8} s).

The addition of the predelay and time gate width changes the capture tally scoring. When a neutron is captured at time t_0 in the specified cell by the specified nuclide (22 and ^{3}He in this example), the gate is "turned on." If the predelay is t_1 and the gate width is t_2, then all captures between $t_0 + t_1$ and $t_0 + t_1 + t_2$ are counted. For a history with no captures, no events are scored. With one capture, 0 events are scored. With two captures, the first turns on the time gate are at time t_0 and scores 0; the second will score one event if it is captured between $t_0 + t_1$ and $t_0 + t_1 + t_2$ or score another 0 if outside the gate.

Other entries after the `CAP` keyword may be placed in any order, as shown in the following examples. The negative entries change the allowed number of captures and moments (defaults 21 and 12 are changed to 40 and 40 in `F78` in this example). The list of capture nuclides may be placed anywhere after `CAP`.

Examples for three capture tallies now follow. The capture tally without gating (`F18`) is shown for reference. An infinite gate (`F38`) results in a very different `PRINT` Table 118: the number of captures is the same, but the moments are offset by one. A finite gate (`F78`) has fewer captures, as expected.

10.2.5.6.1 Case A: Capture Tally without Gate

Input:

```
f18:n 22
ft18 cap 2003
```

Output:

```
1 neutron captures, moments and multiplicity distributions.  tally  18               print table 118

 weight normalization by source histories =          20000

 cell:    22

 neutron captures on 3he
                        captures    captures       multiplicity fractions
            histories  by number   by weight     by number     by weight    error

 captures =  0  13448       0    0.00000E+00   6.72400E-01   6.72400E-01   0.0049
 captures =  1   5550    5550    2.77500E-01   2.77500E-01   2.77500E-01   0.0114
 captures =  2    588    1176    5.88000E-02   2.94000E-02   2.94000E-02   0.0406
 captures =  3    238     714    3.57000E-02   1.19000E-02   1.19000E-02   0.0644
 captures =  4     94     376    1.88000E-02   4.70000E-03   4.70000E-03   0.1029
 captures =  5     40     200    1.00000E-02   2.00000E-03   2.00000E-03   0.1580
 captures =  6     26     156    7.80000E-03   1.30000E-03   1.30000E-03   0.1960
 captures =  7      8      56    2.80000E-03   4.00000E-04   4.00000E-04   0.3535
 captures =  8      5      40    2.00000E-03   2.50000E-04   2.50000E-04   0.4472
 captures =  9      1       9    4.50000E-04   5.00000E-05   5.00000E-05   1.0000
 captures = 12      1      12    6.00000E-04   5.00000E-05   5.00000E-05   1.0000
 captures = 16      1      16    8.00000E-04   5.00000E-05   5.00000E-05   1.0000

    total       20000    8305    4.15250E-01   1.00000E+00   1.00000E+00   0.0128

      factorial moments              by number               by weight

             3he                4.15250E-01 0.0128      4.15250E-01 0.0128
        3he(3he-1)/2!           1.59300E-01 0.0651      1.59300E-01 0.0651
     3he(3he-1)(3he-2)/3!       1.47900E-01 0.2165      1.47900E-01 0.2165
  3he(3he-1) .... (3he-3)/4!    1.87750E-01 0.5063      1.87750E-01 0.5063
  3he(3he-1) .... (3he-4)/5!    2.96500E-01 0.7493      2.96500E-01 0.7493
  3he(3he-1) .... (3he-5)/6!    4.61900E-01 0.8727      4.61900E-01 0.8727
  3he(3he-1) .... (3he-6)/7!    6.15800E-01 0.9311      6.15800E-01 0.9311
  3he(3he-1) .... (3he-7)/8!    6.68950E-01 0.9626      6.68950E-01 0.9626
  3he(3he-1) .... (3he-8)/9!    5.83050E-01 0.9812      5.83050E-01 0.9812
  3he(3he-1) .... (3he-9)/10!   4.03700E-01 0.9918      4.03700E-01 0.9918
  3he(3he-1) .... (3he-10)/11!  2.19000E-01 0.9972      2.19000E-01 0.9972
  3he(3he-1) .... (3he-11)/12!  9.10500E-02 0.9994      9.10500E-02 0.9994
```

10.2.5.6.2 Case B: Infinite Gate

Input:

```
f38:n 22
ft38 cap 2003 gate 0 1e11
```

Output:

```
1 neutron captures, moments and multiplicity distributions.  tally  38               print table 118

 weight normalization by source histories =          20000

 cell:    22

 neutron captures on 3he

 time gate:  predelay =  0.0000E+00     gate width =  1.0000E+11

     pulses         occurrences  occurrences           pulse fraction
    in gate histogram by number    by weight     by number     by weight    error

 captures =  0   6552       0    0.00000E+00   3.27600E-01   3.27600E-01   0.0101
 captures =  1   1002    1002    5.01000E-02   5.01000E-02   5.01000E-02   0.0308
 captures =  2    414     828    4.14000E-02   2.07000E-02   2.07000E-02   0.0486
 captures =  3    176     528    2.64000E-02   8.80000E-03   8.80000E-03   0.0750
 captures =  4     82     328    1.64000E-02   4.10000E-03   4.10000E-03   0.1102
 captures =  5     42     210    1.05000E-02   2.10000E-03   2.10000E-03   0.1541
 captures =  6     16      96    4.80000E-03   8.00000E-04   8.00000E-04   0.2499
 captures =  7      8      56    2.80000E-03   4.00000E-04   4.00000E-04   0.3535
 captures =  8      3      24    1.20000E-03   1.50000E-04   1.50000E-04   0.5773
 captures =  9      2      18    9.00000E-04   1.00000E-04   1.00000E-04   0.7071
 captures = 10      2      20    1.00000E-03   1.00000E-04   1.00000E-04   0.7071
 captures = 11      2      22    1.10000E-03   1.00000E-04   1.00000E-04   0.7071
 captures = 12      1      12    6.00000E-04   5.00000E-05   5.00000E-05   1.0000
 captures = 13      1      13    6.50000E-04   5.00000E-05   5.00000E-05   1.0000
 captures = 14      1      14    7.00000E-04   5.00000E-05   5.00000E-05   1.0000
 captures = 15      1      15    7.50000E-04   5.00000E-05   5.00000E-05   1.0000

    total        8305    3186    1.59300E-01   4.15250E-01   4.15250E-01   0.0291

      factorial moments              by number               by weight

             n                1.59300E-01 0.0651      1.59300E-01 0.0648
         n(n-1)/2!            1.47900E-01 0.2165      1.47900E-01 0.2165
       n(n-1)(n-2)/3!         1.87750E-01 0.5063      1.87750E-01 0.5062
   n(n-1)(n-2) ... (n-3)/4!   2.96500E-01 0.7493      2.96500E-01 0.7492
   n(n-1)(n-2) ... (n-4)/5!   4.61900E-01 0.8727      4.61900E-01 0.8726
   n(n-1)(n-2) ... (n-5)/6!   6.15800E-01 0.9311      6.15800E-01 0.9311
   n(n-1)(n-2) ... (n-6)/7!   6.68950E-01 0.9626      6.68950E-01 0.9626
   n(n-1)(n-2) ... (n-7)/8!   5.83050E-01 0.9812      5.83050E-01 0.9812
   n(n-1)(n-2) ... (n-8)/9!   4.03700E-01 0.9918      4.03700E-01 0.9918
   n(n-1)(n-2) ... (n-9)/10!  2.19000E-01 0.9972      2.19000E-01 0.9972
   n(n-1)(n-2) ... (n-10)/11! 9.10500E-02 0.9994      9.10500E-02 0.9994
   n(n-1)(n-2) ... (n-11)/12! 2.80000E-02 1.0000      2.80000E-02 1.0000
```

10.2.5.7 Case C: Finite Gate

Input:

```
f78:n 22
ft78 cap gate .5 .4 -40 -40 2003
```

Output:

```
1 neutron captures, moments and multiplicity distributions.  tally  78               print table 118

 weight normalization by source histories =          20000

 cell:    22

 neutron captures on 3he

 time gate:  predelay =  5.0000E-01     gate width =  4.0000E-01

     pulses         occurrences  occurrences           pulse fraction
    in gate histogram by number    by weight     by number     by weight    error

 captures =  0   7837      0     0.00000E+00   3.91850E-01   3.91850E-01   0.0118
 captures =  1    394     394    1.97000E-02   1.97000E-02   1.97000E-02   0.0666
 captures =  2     67     134    6.70000E-03   3.35000E-03   3.35000E-03   0.1542
 captures =  3      6      18    9.00000E-04   3.00000E-04   3.00000E-04   0.4082
 captures =  4      1       4    2.00000E-04   5.00000E-05   5.00000E-05   1.0000

  total          8305     550    2.75000E-02   4.15250E-01   4.15250E-01   0.0624

     factorial moments              by number               by weight

             n                 2.75000E-02 0.0717      2.75000E-02 0.0716
          n(n-1)/2!            4.55000E-03 0.1654      4.55000E-03 0.1654
        n(n-1)(n-2)/3!         5.00000E-04 0.4690      5.00000E-04 0.4690
    n(n-1)(n-2) ... (n-3)/4!   5.00000E-05 1.0000      5.00000E-05 1.0000
```

Scratch space is needed to save capture times during the course of a history. The times are stored temporarily in the capture and moment bins of the tally. If sufficient bins are unavailable, then the number of allowed captures and moments must be increased using the negative entries after the `CAP` keyword. The message `*** warning *** dimension overflow. Some pulses not counted.` is written in `PRINT` Table 118 if the space needs to be increased.

10.2.5.8 Example 41: Residual Nuclei Tally

The input file shown in Listing 10.21 models a 1.2-GeV proton source having a single collision with ^{208}Pb.

Listing 10.21: example_residual_nuclei_tally.mcnp.inp.txt

```
Test of p(1.2GeV)+Pb(208)
1 1 -11. -1 imp:h 1
2 0        1 imp:h 0
```

.

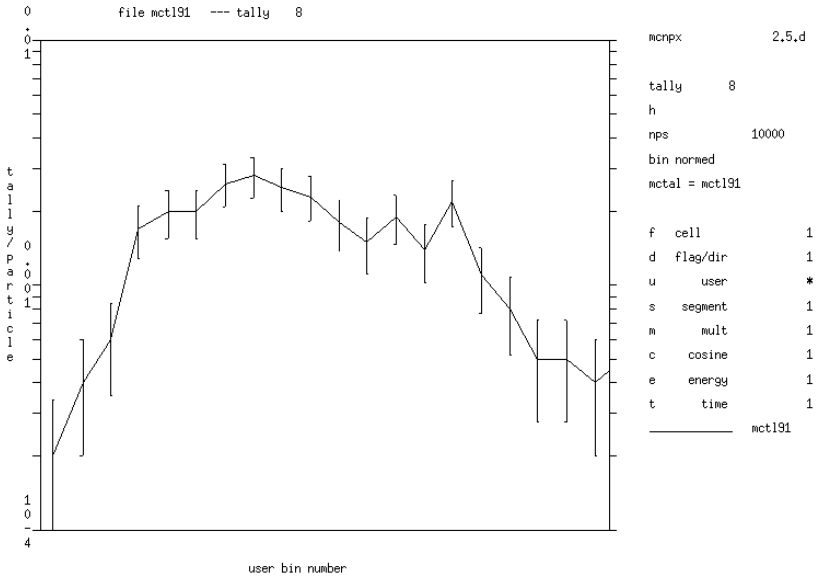


Figure 10.43: Residuals for ^{81}Tl isotopes 189 to 201 from 1.3-GeV protons on $^{208}_{82}$Pb.

```
1 so .01

mode h n
sdef par h erg=1200 vec 0 0 1 dir 1
m1 82208 1
phys:h 1300 j 0
phys:n 1300
fmult data=0
nps 10000
f8:h 1
ft8 res 1 99
fq8 u e
lca   2 1 1 23 1 1 0 -2 0
```

These data are plotted in Fig. 10.43, with MCNP6 using the tally plotter and the execute line command

```
mcnp6 z com=com91
```

where the command file, **com91**, is

```
rmctal=mctl91
tally 8 free u xlim 81189 8120 ylim .0001 .01
```

10.2.5.9 Example 42: ROC Curve Generation

The input file in Listing 10.22 models a 15-MeV photon source incident on a ^{238}U sphere (10 kg). This source is represented as a single 10 μs pulse of 10^7 photons (S_i). A $1/E$ background source is specified in the surrounding cube (200 cm each side), and the FT card PHL option is used to generate a receiver operating characteristic (ROC) curve from the signal and noise components tallied in a Ge detector for 60 s. The Ge detector is surrounded by 2 cm of Pb. The flux of the background photons was taken as 10 $\gamma/\text{cm}^2/\text{min}$. The background source strength (S_b) to produce this flux is given by $A \cdot F/3.7$, where A is the surface area of the cube and F is the flux (the factor of 3.7 comes from the shape of the cube—for a sphere this factor is 1.0). This results in $S_b = 6 \cdot 200 \cdot 200 \cdot 10/3.7$, or 648648 photons. The probability of sampling each source component becomes

$$P_i = \frac{S_i}{S_i + S_b}, \tag{10.4}$$

$$P_b = \frac{S_b}{S_i + S_b}, \tag{10.5}$$

or $P_i = 0.9391$ and $P_b = 0.0609$, as seen on the SP1 card. The *NHB* parameter of the ROC option is set to the sum of these sources, or 10648648. In this example, we ran 10 batches to formulate the signal and noise PDFs and the related ROC curve.

Listing 10.22: example_tally_roc_1.mcnp.inp.txt

```
Generate ROC curve for 15-MeV photons into U-238
1   0           -1  2  6  4  imp:n,p=1
2   0           -2           imp:n,p=1
4   1 -5.16     -3           imp:n,p=1
5   2 -19.0     -4           imp:n,p=1
6   0           -5           imp:n,p=1
7   3 -11.3     -6  3  5     imp:n,p=1
8   0            1           imp:n,p=0

1 rpp  -100 100 -100 100 -100 100
2 so   5.0
3 rcc  20 0 25 0 0 10 4.0
4 sph  20 0  0 5.0
5 rcc  20 0 20 0 0  5 4.0
6 rcc  20 0 20 0 0 17 6.0

mode p n
m1     32074.70c 1
m2     92238.70c 1
m3     82208.70c 1
mphys  on
mx2:p  model
cut:n  60e8
cut:p  60e8
phys:p j 1 j -1
act  fission=p nonfiss=p dg=mg
sdef par=p erg=d1 x=ferg d2 y=ferg d3 z=ferg d4 tme=ferg d7
     vec=1 0 0 dir=ferg d8 cel=1 wgt=1
si1 s       5       6
sp1    0.9391  0.0609
ds2 s  15  16
ds3 s  25  26
ds4 s  35  36
ds7 s  45  46
ds8 s  55  56
```

```
si15 l  5.1
sp15      1
si25 l  0.0
sp25      1
si35 l  0.0
sp35      1
si16 -100 100
sp16    0   1
si26 -100 100
sp26    0   1
si36 -100 100
sp36    0   1
si45 0 0.000010e8
sp45 0    1
si46 0 60e8
sp46 0    1
si55 l 1
sp55   1
si56 -1 1
sp56  0 1
si5 l 15.0
sp5      1
C 1/E for background source
si6 a .100 1.0  2.0  3.0   4.0   5.0   6.0   7.0   8.0   9.0   10.0
sp6   10.0 1.0  0.5  0.333 0.250 0.200 0.167 0.143 0.125 0.111 0.100
c
f4:p  2
f1:p  3.3
e1    1.0 100.0
t1    0.001e8 60e8
ft1   scx 1 roc 10648648
tf1   1 1 1 1 1 1 2 2   1 1 2 1 1 1 2 2 $ signal bins, noise bins
nps   106486480
```

The ROC output for Tally 1 is provided in PRINT Table 163, shown in Listing 10.23. The first printed plot is the ROC curve itself, plotting the noise PDF (usually referred to as the probability of false alarm) versus the signal PDF (usually referred to as the probability of detection). The data for the signal and noise PDFs is provided in the subsequent table. The jagged behavior of the ROC curve can be significantly refined by increasing the number of batches (say from 10 to 100, or by running 1064864800 particle histories).

Listing 10.23: example_tally_roc_1.mcnp.outp.txt

```
1roc curve for tally 1    10 batches, signal mean= 3.029E+01 noise mean= 2.960E+01 nps = 106486480          print table 163

 abscissa            ordinate                plot of probability of detection versus probability of false alarm - 0 to 100 percent
   noise               signal:--------10--------20--------30--------40--------50--------60--------70--------80--------90-------100
    1.000               0.000|x       |         |         |         |         |         |         |         |         |         |
    2.000               0.000|x       |         |         |         |         |         |         |         |         |         |
    3.000               0.000|x       |         |         |         |         |         |         |         |         |         |
    4.000               0.000|x       |         |         |         |         |         |         |         |         |         |
    5.000               0.000|x       |         |         |         |         |         |         |         |         |         |
    6.000               0.000|x       |         |         |         |         |         |         |         |         |         |
    7.000               0.000|x       |         |         |         |         |         |         |         |         |         |
    8.000               0.000|x       |         |         |         |         |         |         |         |         |         |
    9.000               0.000|x       |         |         |         |         |         |         |         |         |         |
   10.000              10.000|        x         |         |         |         |         |         |         |         |         |
   11.000              10.000|        x         |         |         |         |         |         |         |         |         |
   12.000              10.000|        x         |         |         |         |         |         |         |         |         |
   13.000              10.000|        x         |         |         |         |         |         |         |         |         |
   14.000              10.000|        x         |         |         |         |         |         |         |         |         |
   15.000              10.000|        x         |         |         |         |         |         |         |         |         |
   16.000              10.000|        x         |         |         |         |         |         |         |         |         |
   17.000              10.000|        x         |         |         |         |         |         |         |         |         |
   18.000              10.000|        x         |         |         |         |         |         |         |         |         |
   19.000              10.000|        x         |         |         |         |         |         |         |         |         |
   20.000              10.000|        x         |         |         |         |         |         |         |         |         |
   21.000              10.000|        x         |         |         |         |         |         |         |         |         |
   22.000              10.000|        x         |         |         |         |         |         |         |         |         |
   23.000              10.000|        x         |         |         |         |         |         |         |         |         |
   24.000              10.000|        x         |         |         |         |         |         |         |         |         |
   25.000              10.000|        x         |         |         |         |         |         |         |         |         |
   26.000              10.000|        x         |         |         |         |         |         |         |         |         |
   27.000              10.000|        x         |         |         |         |         |         |         |         |         |
   28.000              10.000|        x         |         |         |         |         |         |         |         |         |
   29.000              10.000|        x         |         |         |         |         |         |         |         |         |
   30.000              10.000|        x         |         |         |         |         |         |         |         |         |
   31.000              12.000|        | x       |         |         |         |         |         |         |         |         |
   32.000              14.000|        |   x     |         |         |         |         |         |         |         |         |
   33.000              16.000|        |     x   |         |         |         |         |         |         |         |         |
   34.000              18.000|        |       x |         |         |         |         |         |         |         |         |
   35.000              20.000|        |         x         |         |         |         |         |         |         |         |
   36.000              22.000|        |         | x       |         |         |         |         |         |         |         |
   37.000              24.000|        |         |   x     |         |         |         |         |         |         |         |
```

```
   38.000    26.000|  |  |  x  |  |  |  |  |  |  |  |
   39.000    28.000|  |  |    x|  |  |  |  |  |  |  |
   40.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   41.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   42.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   43.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   44.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   45.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   46.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   47.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   48.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   49.000    40.000|  |  |  |  x  |  |  |  |  |  |  |
   50.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   51.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   52.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   53.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   54.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   55.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   56.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   57.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   58.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   59.000    50.000|  |  |  |  |  x  |  |  |  |  |  |
   60.000    70.000|  |  |  |  |  |  |  x  |  |  |  |
   61.000    71.000|  |  |  |  |  |  |  |x  |  |  |  |
   62.000    72.000|  |  |  |  |  |  |  | x  |  |  |  |
   63.000    73.000|  |  |  |  |  |  |  |  x  |  |  |  |
   64.000    74.000|  |  |  |  |  |  |  |   x  |  |  |  |
   65.000    75.000|  |  |  |  |  |  |  |    x  |  |  |  |
   66.000    76.000|  |  |  |  |  |  |  |     x  |  |  |  |
   67.000    77.000|  |  |  |  |  |  |  |      x  |  |  |  |
   68.000    78.000|  |  |  |  |  |  |  |       x  |  |  |  |
   69.000    79.000|  |  |  |  |  |  |  |        x|  |  |  |
   70.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   71.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   72.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   73.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   74.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   75.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   76.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   77.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   78.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
   79.000    80.000|  |  |  |  |  |  |  |         x  |  |  |
```

```
  80.000           100.000|        |        |        |        |        |        |        |        |        |        x
  81.000           100.000|        |        |        |        |        |        |        |        |        |        x
  82.000           100.000|        |        |        |        |        |        |        |        |        |        x
  83.000           100.000|        |        |        |        |        |        |        |        |        |        x
  84.000           100.000|        |        |        |        |        |        |        |        |        |        x
  85.000           100.000|        |        |        |        |        |        |        |        |        |        x
  86.000           100.000|        |        |        |        |        |        |        |        |        |        x
  87.000           100.000|        |        |        |        |        |        |        |        |        |        x
  88.000           100.000|        |        |        |        |        |        |        |        |        |        x
  89.000           100.000|        |        |        |        |        |        |        |        |        |        x
  90.000           100.000|        |        |        |        |        |        |        |        |        |        x
  91.000           100.000|        |        |        |        |        |        |        |        |        |        x
  92.000           100.000|        |        |        |        |        |        |        |        |        |        x
  93.000           100.000|        |        |        |        |        |        |        |        |        |        x
  94.000           100.000|        |        |        |        |        |        |        |        |        |        x
  95.000           100.000|        |        |        |        |        |        |        |        |        |        x
  96.000           100.000|        |        |        |        |        |        |        |        |        |        x
  97.000           100.000|        |        |        |        |        |        |        |        |        |        x
  98.000           100.000|        |        |        |        |        |        |        |        |        |        x
  99.000           100.000|        |        |        |        |        |        |        |        |        |        x
                          |--------10--------20--------30--------40--------50--------60--------70--------80--------90-------100

   tally            signal                 noise
  upper bin       pdf        cum.        pdf        cum.
  1.819E+01   0.000E+00   1.000E+00   1.000E-01   1.000E+00
  1.838E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  1.857E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  1.876E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  1.895E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  1.914E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  1.933E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  1.952E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  1.971E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  1.990E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  2.009E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  2.028E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  2.047E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  2.066E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  2.085E+01   0.000E+00   1.000E+00   0.000E+00   9.000E-01
  2.104E+01   0.000E+00   1.000E+00   1.000E-01   9.000E-01
  2.123E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
  2.142E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
```

```
 2.161E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.180E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.199E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.218E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.237E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.256E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.275E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.294E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.313E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.332E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.351E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.370E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.389E+01   0.000E+00   1.000E+00   0.000E+00   8.000E-01
 2.408E+01   1.000E-01   1.000E+00   0.000E+00   8.000E-01
 2.427E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.446E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.465E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.484E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.503E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.522E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.541E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.560E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.579E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.598E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.617E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.636E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.655E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.674E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.693E+01   0.000E+00   9.000E-01   0.000E+00   8.000E-01
 2.712E+01   1.000E-01   9.000E-01   0.000E+00   8.000E-01
 2.731E+01   0.000E+00   8.000E-01   0.000E+00   8.000E-01
 2.750E+01   0.000E+00   8.000E-01   0.000E+00   8.000E-01
 2.769E+01   0.000E+00   8.000E-01   0.000E+00   8.000E-01
 2.788E+01   0.000E+00   8.000E-01   0.000E+00   8.000E-01
 2.807E+01   0.000E+00   8.000E-01   1.000E-01   8.000E-01
 2.826E+01   0.000E+00   8.000E-01   0.000E+00   7.000E-01
 2.845E+01   0.000E+00   8.000E-01   0.000E+00   7.000E-01
 2.864E+01   0.000E+00   8.000E-01   0.000E+00   7.000E-01
 2.883E+01   0.000E+00   8.000E-01   0.000E+00   7.000E-01
 2.902E+01   1.000E-01   8.000E-01   1.000E-01   7.000E-01
 2.921E+01   0.000E+00   7.000E-01   0.000E+00   6.000E-01
 2.940E+01   0.000E+00   7.000E-01   0.000E+00   6.000E-01
```

```
  2.959E+01   0.000E+00   7.000E-01   0.000E+00   6.000E-01
  2.978E+01   0.000E+00   7.000E-01   0.000E+00   6.000E-01
  2.997E+01   2.000E-01   7.000E-01   0.000E+00   6.000E-01
  3.016E+01   0.000E+00   5.000E-01   1.000E-01   6.000E-01
  3.035E+01   0.000E+00   5.000E-01   0.000E+00   5.000E-01
  3.054E+01   0.000E+00   5.000E-01   0.000E+00   5.000E-01
  3.073E+01   0.000E+00   5.000E-01   0.000E+00   5.000E-01
  3.092E+01   1.000E-01   5.000E-01   0.000E+00   5.000E-01
  3.111E+01   0.000E+00   4.000E-01   1.000E-01   5.000E-01
  3.130E+01   0.000E+00   4.000E-01   0.000E+00   4.000E-01
  3.149E+01   0.000E+00   4.000E-01   0.000E+00   4.000E-01
  3.168E+01   0.000E+00   4.000E-01   0.000E+00   4.000E-01
  3.187E+01   1.000E-01   4.000E-01   0.000E+00   4.000E-01
  3.206E+01   2.000E-01   3.000E-01   1.000E-01   4.000E-01
  3.225E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.244E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.263E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.282E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.301E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.320E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.339E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.358E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.377E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.396E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.415E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.434E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.453E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.472E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.491E+01   0.000E+00   1.000E-01   0.000E+00   3.000E-01
  3.510E+01   0.000E+00   1.000E-01   2.000E-01   3.000E-01
  3.529E+01   0.000E+00   1.000E-01   0.000E+00   1.000E-01
  3.548E+01   0.000E+00   1.000E-01   0.000E+00   1.000E-01
  3.567E+01   0.000E+00   1.000E-01   0.000E+00   1.000E-01
  3.586E+01   0.000E+00   1.000E-01   0.000E+00   1.000E-01
  3.605E+01   0.000E+00   1.000E-01   0.000E+00   1.000E-01
  3.624E+01   0.000E+00   1.000E-01   0.000E+00   1.000E-01
  3.643E+01   0.000E+00   1.000E-01   0.000E+00   1.000E-01
  3.662E+01   0.000E+00   1.000E-01   0.000E+00   1.000E-01
  3.681E+01   1.000E-01   1.000E-01   0.000E+00   1.000E-01
  3.700E+01   0.000E+00   0.000E+00   1.000E-01   1.000E-01
```

10.2.6 Repeated Structure/Lattice Tally Example

An explanation of the basic repeated structure/lattice tally format can be found in §5.9.1.5. The example shown here illustrates more complex uses.

10.2.6.1 Example 43: Repeated-structure Lattice-tally Example

An example repeated structure lattice tally with a complicated track-length tally is shown in Listing 10.24.

Listing 10.24: example_repeated_structure_tally_2.mcnp.inp.txt

```
Repeated structure lattice tally example
1  0        -1 -2   3  13  fill=4
2  0        -1 -2   3 -13  fill=1
3  0        -4  5  -6   7  u=1 lat=1
                            fill=-2:2 -2:0 0:0 1 1 3 1 1 1 3 2 3 1 3 2 3 2 3
4  0        -8  9 -10  11  u=2 fill=3 lat=1
5  0       -12             u=3
6  0        12             u=3
7  0       -14 -2   3      u=4 fill=3 trcl=(-60 40 0)
8  like 7 but trcl=(-30 40 0)
9  like 7 but trcl=(0 40 0)
10 like 7 but trcl=(30 40 0)
11 like 7 but trcl=(60 40 0)
12 0        #7 #8 #9 #10 #11 u=4
13 0        1:2:-3

1  cz   100
2  pz   100
3  pz  -100
4  px    20
5  px   -20
6  py    20
7  py   -20
8  px    10
9  px   -10
10 py    10
11 py   -10
12 cz     5
13 py    19.9
14 cz    10

sdef
f4:n   5 6 (5 6 3)                                          $ a: 3 bins
      (5<3) (5<(3[-2:2 -2:0 0:0]))                          $ b: 2 bins
      (5<(7 8 9 10 11)) (5<7 8 9 10 11<1) (5<1)             $ c: 7 bins
      ((5 6)<3[0 -1 0]) ((5 6)<3[0:0 -1:-1 0:0]) ((5 6)<3[8]) $ d: 3 bins
      (5<(4[0 0 0]3[8]))(5<4[0 0 0]<3[8])
           (3<(3[1]3[2]3[4]3[5]3[6]3[10]))                  $ e: 3 bins
      (5<u=3)                                               $ f: 12 bins
sd4    1 29r
print
nps 100
imp:n 1 11r 0
```

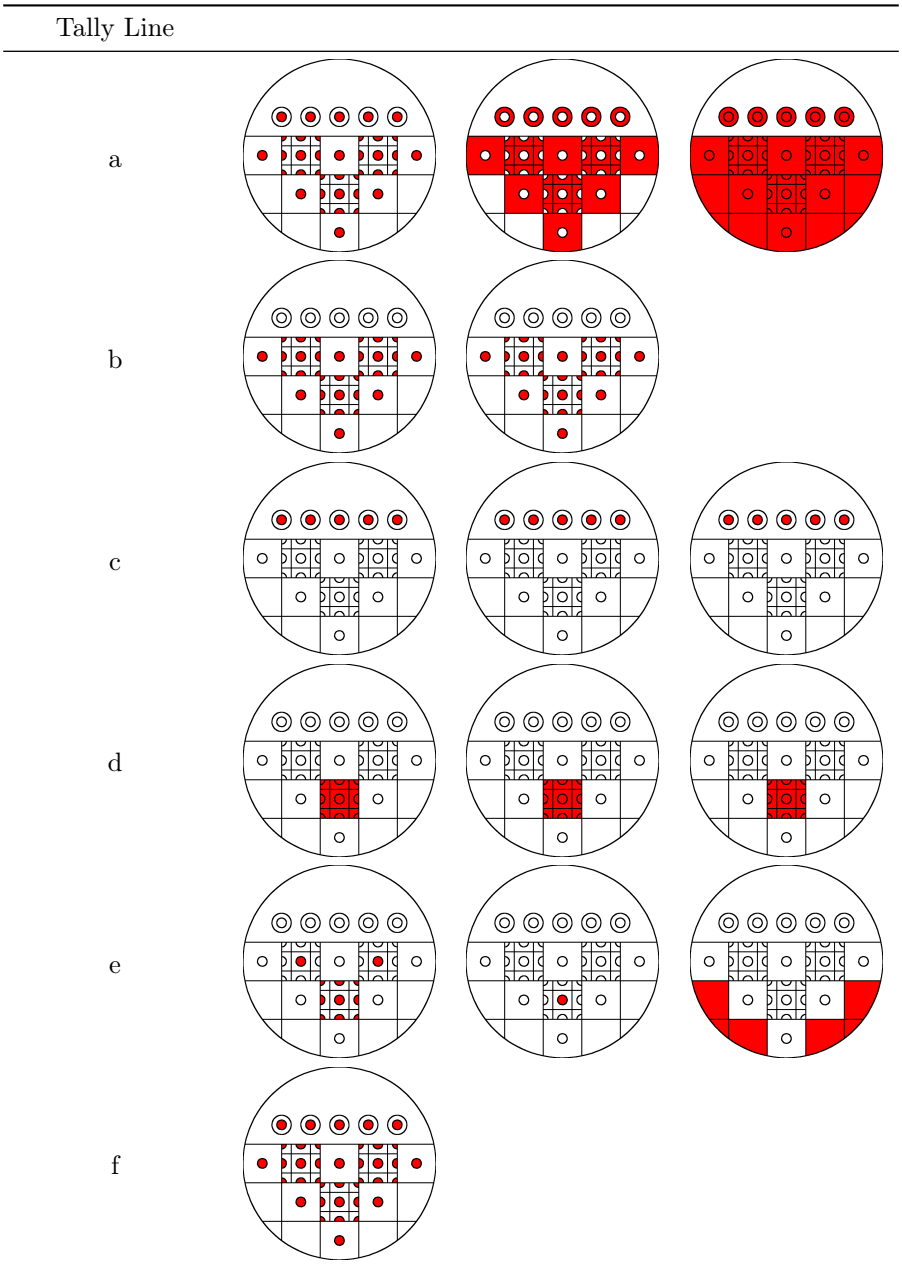


Figure 10.44: Example 33 Tally Regions by F4 Line. (a–f) indicates the tally regions for each tally line. The number of bins generated by MCNP6 is shown at the end of each tally line following the $ in-line comment symbol.

10.2.6.1.1 Tally Line 1

This first line creates three tally output bins: cell 5, cell 6, and the union of cells 5, 6, and 3, as indicated in Fig. 10.44a. Because cell 3 is filled entirely by cells 5 and 6, a tally over cell 5 plus cell 6 is the same as a tally over cell 3. If a particle is tallied in cell 5 and tallied in cell 3, it will be tallied twice in the bin (5 6 3).

Caution

A true union is performed when first level cells overlap (or fill) another cell. This is not a tally that is normally desired. If an average of cell 3 and region (5 6) outside cell 3 is desired, separate bins must be defined and properly combined using correct volume weighting.

10.2.6.1.2 Tally Line 2

These two input tally bins result in identical output tallies, as shown in Fig. 10.44b. The use of lattice index brackets that include all existing lattice elements makes the two tallies equivalent. The simpler format will execute faster.

10.2.6.1.3 Tally Line 3

This line illustrates omission of geometry levels and a single output bin versus multiple bins. All three input bins tally cell 5 within cells 7 through 11. The second bin specifies the entire path explicitly. Because the only place cell 5 exists within cell 1 is in cells 7–11, the 7–11 specification can be omitted, as in the third input bin. In the second input bin, the parentheses around cells 7–11 are omitted, creating multiple output bins. Five tally bins are produced: (5<7<1), (5<8<1), (5<9<1), (5<10<1), and (5<11<1). The sum of these five bins should equal the tally in the first and last output bins on this line. The tally regions are shown in Fig. 10.44c.

10.2.6.1.4 Tally Line 4

This line illustrates the union of multiple tally cells, (5 6), and various ways of specifying lattice index data. The three input tally bins create three output tally bins with identical values because the three different lattice descriptions refer to the same lattice element, the eighth entry on the `FILL` array. If the parentheses around (5 6) were removed, two output bins would be created for each input bin, namely $(5<3[0,-1,0])$ and $(6<3[0,-1,0])$, etc. The tally regions are shown in Fig. 10.44d.

10.2.6.1.5 Tally Line 5

This line illustrates tallies in overlapping regions in repeated structures in a lattice and a tally in lattice elements filled with themselves. Three tally output bins are produced. In the first input bin, a particle is tallied only once when it is in cell 5 and in $4[0,0,0]$ or when it is in cell 5 and in $3[0,-1,0]$. Fig. 10.44e shows all the cell 5 instances included in this tally bin. This tally is probably more useful than the overlapping regions in tally line 1. Input bin 2 demonstrates a tally for a nested lattice. A tally is made when a particle is in cell 5 and in cell 4, element $[0,0,0]$ and in cell 3, element $[0,\ -1,\ 0]$. Note that $3[0,-1,0]$ is indeed filled with cell 4 (u=2). If that were not true, a zero tally would result in this bin. The final input tally bin demonstrated a tally in lattice elements that are filled with their own universe number. This method is the only way to tally in these elements separate from the rest of cell 3.

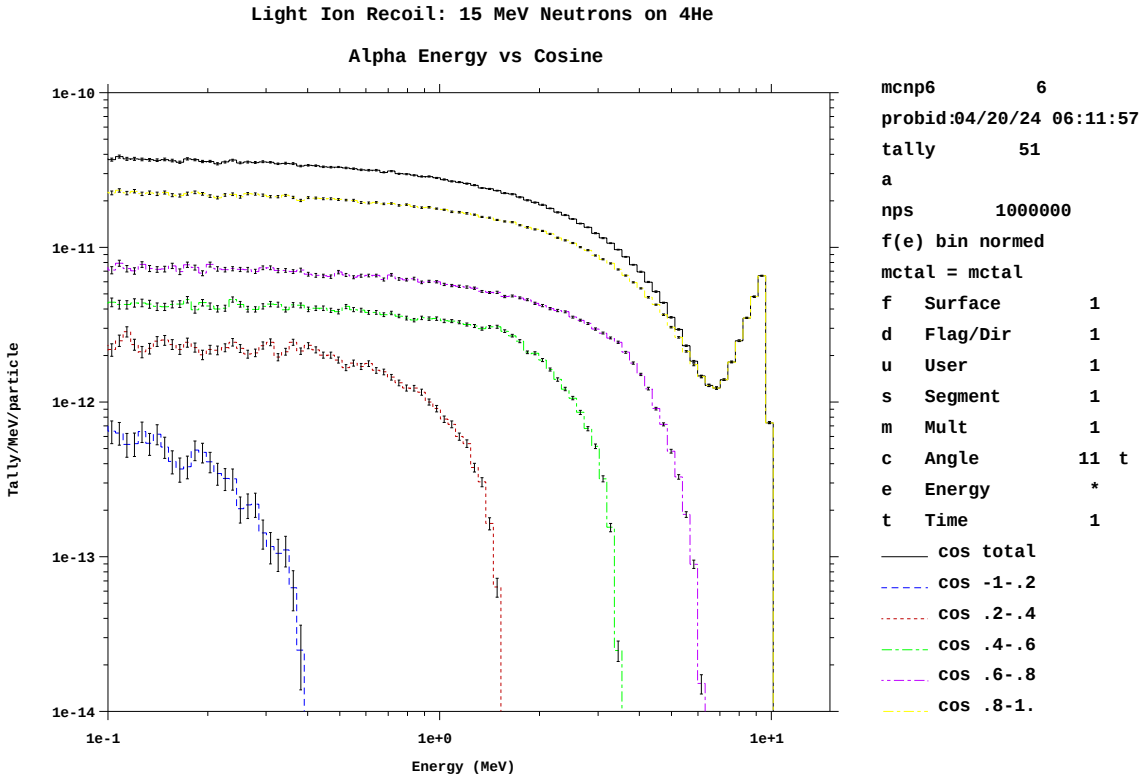


Figure 10.45: Light ion recoil.

10.2.6.1.6 Tally Line 6

This line illustrates the universe format. The single input bin includes all possible chains involving cell 5. Because u=3 is not within parentheses, the input is expanded into twelve output bins: (5<3[3], etc.). The format 3[3] indicates the third lattice element of cell 3 as entered on the cell 3 FILL array. Note that the third element is filled by universe 3, consisting of cells 5 and 6. The tally regions are shown in Fig. 10.44f.

10.2.7 Miscellaneous Tally Examples

10.2.7.1 Example 44: Light Ion Recoil (RECL)

MCNP6 can produce and track ions created by elastic recoil from neutrons or protons. Neutrons and protons undergoing elastic scatter with light nuclei (H, D, T, ^{3}He, and ^{4}He) can create ions (protons, deuterons, tritons, ^{3}He, and α) that are banked for subsequent transport.

Figure 10.45 shows the energy-angle production of alphas created from 15 MeV neutrons striking ^{4}He. Note that in the forward bin, cosine $0.8 < \mu < 1$, the α energy goes up to the theoretical maximum of 9.6 MeV. The theoretical maxima in the other cosine bins (0.8, 0.6, 0.4, and 0.2) are 6.144, 3.456, 1.536, and 0.384.

The input file for this example is shown in Listing 10.25 and the plot command input file is shown in Listing 10.26.

Listing 10.25: example_light_ion_recoil_1.mcnp.inp.txt

```
Test of light ion recoil
1  1  1e-5    -1
2  0           1

1 so      1.e-5

prdmp    2j 1
mode      n  a
imp:n,a   1  0
phys:n   6j  1
sdef     erg=15
print   -161 -162
tmp1     1e-20  0
fcl:n     1  0
m1        2004.00c  0.2
cut:a     j  0
nps       1000000
f51:a     1
e51       0.1 100log 20
c51      -0.8  8i  1  t
fq51      e  c
```

Listing 10.26: example_light_ion_recoil_1.mcnp.comin.txt

```
rmct mctal tal 51 &
xlims 0.1 15 ylims 1e-14 1e-10 loglog &
title 1 "Light Ion Recoil: 15 MeV Neutrons on 4He" &
title 2 "Alpha Energy vs Cosine" &
fix c 11 label 1 "cos total" cop fix c 6 label 2 "cos -1-.2" &
cop fix c 7 label 3 "cos .2-.4" cop fix c 8 label 4 "cos .4-.6" &
cop fix c 9 label 5 "cos .6-.8" cop fix c 10 label 6 "cos .8-1."
end
```

The MCNP commands to produce Fig. 10.45 are `mcnp6 i= example_light_ion_recoil_1.mcnp.inp.txt` followed by `mcnp6 z com= example_light_ion_recoil_1.comin.inp.txt notek`.

10.2.7.2 Inline Generation of Double Differential Cross Sections and Residual Nuclei

The double differential cross sections and distributions of residual nuclei for a single nuclear interaction thus may be calculated directly in MCNP6. Tallying of the residual nuclei is discussed in the `FT8 RES` tally description. Tallying of the differential cross section can be done with standard `F1` surface tallies, as shown in the following example. The input file shown in Listing 10.27 models a 1.2 GeV proton source having a single collision with ^{208}Pb.

Listing 10.27: example_double_diff_xs_1.mcnp.inp.txt

```
Test of p(1.2GeV)+Pb(208)
1 1 -11. -1 imp:h 1
2 0        1 imp:h 0

1 so .01

```

```
prdmp 2j 1
mode h n
sdef par h erg=1200 vec 0 0 1 dir 1
m1 82208 1
phys:h 1300 j 0
phys:n 1300
fmult data=0
nps 10000
fc1  *** neutron angle spectra tally ***
f1:n  1
ft1  frv 0 0 1
fq1 e c
*c1 167.5 9i 17.5 0 T
e1 1 50log 1300 T
lca   2 1 1 23 1 1 0 -2 0
```

Listing 10.28: example_double_diff_xs_1.mcnp.comin.txt

```
rmctal mctal
file all loglog xlim 1 1300 ylim 1e-6 1 &
fix c 13 label 1 "all angles" &
cop fix c 1 label 2 "180 deg" &
cop fix c 6 label 3 "100 deg" &
cop fix c 12 label 4 "0 deg"
end
```

The differential production for neutron production is tallied in the F1 current tally with energy and time bins. This tally is simply the neutrons that are created from the single proton collision with lead and then escape.

In Fig. 10.46, the first line (solid black) is the energy spectrum over all angles, the second (blue dashed) is the 180° output, the third (red dotted) is the 100° output, and the fourth (green broken) is the 0° output. Use of the FM -3 option for Tally 1 in this example will convert these production results into differential cross sections (units of barns).

The MCNP commands to produce Fig. 10.46 are `mcnp6 i= example_double_diff_xs_1.mcnp.inp.txt` followed by `mcnp6 z com= example_double_diff_xs_1.mcnp.comin.txt notek`.

10.2.8 TALLYX Subroutine Examples

An explanation of the **TALLYX** subroutine arguments can be found in §5.9.17. Only examples illustrating some uses of **TALLYX** will be found here.

10.2.8.1 Example 46

In §10.2.4.1, the FS*n* card is used to get the flux through a window on the face of a cube. Instead of using the FS2 card, which established five sub tallies, **TALLYX** could have been used to get only the desired window tally. Two input cards are used:

```
FU2      1
RDUM    -0.5 0.5 -0.5 0.5
```

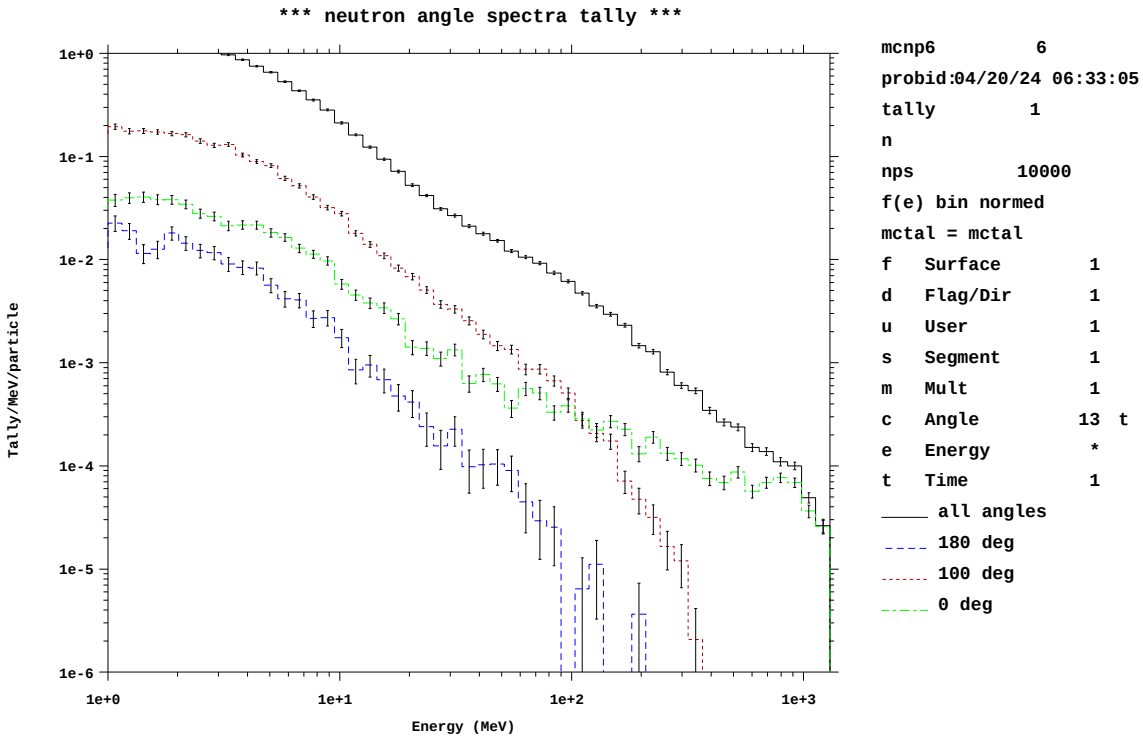


Figure 10.46: Differential production at all angles (black), 180° (blue), 100° (red), 0° (green), for 1.3 GeV protons on $^{208}_{82}$Pb.

The subroutine shown in Listing 10.29 performs the work of extracting the desired window tally. The subroutine is implemented just like a user-provided **SOURCE** subroutine by replacing the file **TALLYX.F90**. Note that **ib=0** and **tally_p_thread%ibu=1** upon entry into **TALLYX**.

Listing 10.29: example_tallyx_rdum.f90.txt

```
subroutine tallyx(t,ib)
  use mcnp_params
  use mcnp_global
  use pblcom, only: pbl
  use mcnp_debug

  implicit none

  real(dknd), intent(inout) :: t
  integer,    intent(inout) :: ib

  if( (pbl%r%x < rdum(1)) .or. (pbl%r%x > rdum(2)) ) ib=-1
  if( (pbl%r%z < rdum(3)) .or. (pbl%r%z > rdum(4)) ) ib=-1
  return
end subroutine tallyx
```

The subroutine was generalized a bit by using the RDUM input card, although the card could have been avoided by directly encoding the values of the dimensions of the window into **TALLYX**.

10.2.8.2 Example 47

Calculate the number of neutron tracks exiting cell 20 per source neutron. The input cards are

```
F4:N     20
FU4       1
SD4       1
```

and the **tallyx.f90** file is given in Listing 10.30.

Listing 10.30: example_tallyx_exiting_tracks.f90.txt

```
subroutine tallyx(t,ib)
  use mcnp_params
  use mcnp_global
  use pblcom, only: pbl
  use mcnp_debug

  implicit none

  real(dknd), intent(inout) :: t
  integer,    intent(inout) :: ib

  t=1.0_dknd
  if (pbl%r%dcs < pbl%r%dls) ib = -1
  return
end subroutine tallyx
```

The quantity `t=1.0` is scored every time a track exits cell 20. The variables used in this subroutine, **pbl%r%dcs** (the distance to collision) and **pbl%r%dls** (distance to the boundary), are available to **TALLYX** from the module PBLCOM.

10.2.8.3 Example 48

Divide the point detector scores into separate tallies (that is, user bins) depending on which of the 20 cells in a problem geometry caused the contributions. The input cards are

```
F5:N    0  0  0  0
FU5     1  18I  20
```

and **TALLYX** subroutine is shown in Listing 10.31.

Listing 10.31: example_tallyx_f5_contributing_cells.f90.txt

```
subroutine tallyx(t,ib)
  use mcnp_params
  use mcnp_global
  use tskcom, only: tally_p_thread
  use pblcom, only: pbl
  use mcnp_debug

  implicit none

  real(dknd), intent(inout) :: t
  integer,    intent(inout) :: ib

  tally_p_thread%ibu=pbl%i%icl
  return
end subroutine tallyx
```

The FU5 card establishes 20 separate user bins, one for each cell in the problem. Note the use of the "*n*I" input format [§4.4.5.1], which creates 18 linear interpolates between 1 and 20.

10.2.8.4 Example 49

Determine the quantity $\int \varphi(E) f(E) \mathrm{d}E$ in cell 14 where $f(E) = \exp(\alpha t)$. The input cards are

```
F4:N    14
FU4     alpha
```

where `alpha` is a numerical value and **TALLYX** is shown in Listing 10.32.

Listing 10.32: example_tallyx_time_response.f90.txt

```
subroutine tallyx(t,ib)
  use mcnp_params
  use mcnp_global
  use tskcom, only: tally_p_thread
  use pblcom, only: pbl
  use basic_tally, only: tds, iptal
  use mcnp_debug

  implicit none

```

```
  real(dknd), intent(inout) :: t
  integer,    intent(inout) :: ib

  t=t*exp(tds(iptal(3,1,tally_p_thread%ital)+1)*pbl%r%tme)
  return
end subroutine tallyx
```

The FU4 card establishes a single user bin, and the value of α is stored in **tds(iptal(3,1,tally_p_thread%ital)+1)** and used for the tally label.

10.2.8.5 Example 50

Tally the number of neutrons passing through cell 16 that have had 0, 1, 2, 3, or 4 collisions. The input cards are

```
F4:N    16
FU4      0  1  2  3  4
SD4      1
```

and **TALLYX** is shown in Listing 10.33.

Listing 10.33: example_tallyx_collision_bins.f90.txt

```
subroutine tallyx(t,ib)
  use mcnp_params
  use mcnp_global
  use tskcom, only: tally_p_thread
  use pblcom, only: pbl
  use mcnp_debug

  implicit none

  real(dknd), intent(inout) :: t
  integer,    intent(inout) :: ib

  tally_p_thread%ibu = pbl%i%ncp
  if(tally_p_thread%ibu > 5 ) ib=-1
  t=pbl%r%wgt
  return
end subroutine tallyx
```

If the IF statement in this **TALLYX** is omitted, a count will be made of the cases of five or more collisions, and in these cases no score will be tallied but a count will be printed of the times that the tally was unable to be made because **tally_p_thread%ibu** was a value where no bin existed.

In the five user bins, t is the number of neutrons per source neutron passing through cell 16 that have undergone 0, 1, 2, 3, or 4 collisions, respectively. Note that the FU4 card has five entries to establish the five user bins and provide labels. Note also that in this example, the neutrons are calculated so that $t = t\times$renormalization factor (which preserves the weight associated with the tracks), where in **TALLYX** subroutine Listing 10.30 the neutron tracks are calculated so that t=1. Finally, note that if pbl%i%ncp > 5 (six or more collisions) no tally is made because **ib** is set to be less than zero. If an E4 card was added, the neutrons would be tallied as a function of energy for each user bin.

10.3 Source Examples

10.3.1 General Source

Some examples of the general source are given here to illustrate the power and complexity of this feature. Refer to §5.8 for a complete explanation and other examples.

The following example is of the general source that illustrates two levels of dependency. Let us assume a duct streaming problem where the source at the duct opening has been obtained from a reactor calculation. Energies above 13.5 MeV have one angular distribution and energies below 13.5 MeV have a different angular distribution. The source has a uniform spatial distribution on a circular disk of radius 37 cm centered at (x, y, z) on planar surface 1 going into cell 2.

10.3.1.1 Example 51

```
SDEF  ERG = D1  DIR FERG D2  SUR = 1  CEL = 2
      POS = x y z  RAD D5  AXS u v w  VEC u v w
c Source Definition Card.
c In this example, AXS is needed to define a vector which
c defines the source plane of a disk source.
c In this example, POS defines the location of the center
c of the disk.
c VEC is the direction that source particles will be
c travelling once created.
c AXS and VEC can be different.
c For this duct streaming problem, they should be the same.
c
SI1  H  1E-7  1E-5 ... 13.5  14 ...    20
c Source Information 1 (SI1) corresponds to D1.
c H indicates histogram values follow.
c
SP1  D  0    10E-4 ... 10E-2 10E-1 ... 0.3
c Source Probability 1 (SP1) augments SI1.
c D indicates discrete values.
c Probability of each bin on SI1.
c The probability a source particle will be between 10E-7
c and 10E-5 MeV is 10E-4.
c
DS2  S  3     3 ...    3     4 ...     4
c Dependent Source 2 (Depends as a function of ERG).
c S indicates numbers following are themselves other
c distributions.
c In this example, if a particle has an energy in bin 10E-7 to
c 10E-5, then it will have a direction associated with source
c distribution 3.
c
SI3     0   0.2 ... 1
c Source Information 3 (Second Level)
c Default is histogram values.
c
SP3  D  0  1E-4 ... 0.1
c Source Probability 3 (Second Level).
c Probability of each bin on SI3.
```

```
c
SI4     0   0.1 ... 1
c Source Information 4 (Second Level).
c Default is histogram values.
c
SP4  D  0  1E-2 ... 0.1
c Source Probability 4 (Second Level).
c Probability of each bin on SI4.
c
SI5     37
c Source Information 5.
c Default is histogram values.
c There is one bin from 0 to 37.
c When used with the RAD keyword on the SDEF card, it indicates
c a circular distribution from 0 to 37 cm.
c
SP5     -21 1
c Source Probability 5.
c The -21 indicates a sampling scheme based on a power of the
c variable.
c In this case, the sampling is a function of radius^1,
c which results in a uniform spatial distribution over the disk.
c Since a uniform spatial distribution is the default for disk
c sources, this card is optional.
c
```

This example can be expanded by having the source in two ducts instead of one (with the same energy and angular distribution as before). The SI1, SP1, DS2, SI3, SP3, SI4, and SP4 cards remain unchanged, but the SI5 and SP5 cards are no longer valid. The SDEF card is changed as shown, and the other cards are added.

```
SDEF  ERG = D1  DIR FERG D2  SUR = D6  CEL FSUR D7
      POS FSUR D8  RAD FSUR D9  AXS FSUR D10 VEC FSUR D10
SI6  L     1          7
c Source Information 6.
c L indicates discrete values, in this case surface 1 or 7
c
SP6  D     0.6        0.4
c Source Probability 6.
c Probability of each value on SI6.
c
DS7  L     2          8
c Dependent Source 7 (Depends as a function of SUR).
c L indicates discrete value, in this case cell 2 or 8,
c depending on whether surface 1 or 7, respectively, was chosen.
c
DS8  L     x1 y1 z1    x2 y2 z2
c Dependent Source 8 (Depends as a function of SUR).
c L indicates discrete values, in this case the respective centers,
c of two disks, depending on whether surface 1 or 7 was chosen.
c
DS9  S     11         12
c Dependent Source 9 (Depends as a function of SUR).
c S indicates other distributions, in this case the respective radii,
c of two disks, depending on whether surface 1 or 7 was chosen.
c
DS10 L     u1 v1 w1    u2 v2 w2
c Dependent Source 10 (Depends as a function of SUR).
```

```
c L indicates discrete values, in this case the vectors that define a
c plane that the disk is on and the vector from which DIR is measured.
c In this streaming problem, AXS=VEC, and both depend on whether
c surface 1 or 7 was chosen.

SI11       0   37
SP11     -21    1
SI12       0   25
SP12     -21    1
c In this problem, the radius of the duct depends on which
c duct was chosen.
```

10.3.1.2 Example 52

This example is a two-source-cell problem where the material in one cell is uranium and in the other is thorium. The uranium cell has two isotopes, ^{235}U and ^{238}U, and the thorium has one, ^{232}Th. Each isotope has many photon lines from radioactive decay. The following input cards describe this source.

```
SDEF  CEL D1  ERG FCEL D2  POS FCEL D3
c
SC1 Source Cells
c Source Comment 1
c
SI1  L  1  2
c Source Information 1
c L indicates discrete values, in this case cell 1 or 2.
c The cell also determines the element in this problem.
c
SP1  D  2  1
c Source Probability 1
c Probability of each value on SI1. Here the cell with
c uranium is twice as likely as the thorium cell.
c Other distributions based on volume or decay rate,
c for example, are also possible.
c
SC2 source "spectra"
DS2  S  4  5
c Dependent Source 2 (Depends as a function of CEL).
c S indicates numbers following are themselves other
c distributions.
c In this example, if a particle starts in cell 1, then the
c ERG is defined by source distribution 4.
c
DS3  L  0 0 0  10.5 0 0
c
SC4 uranium nuclides
SI4  S  6  7
SP4  D  1  3
c Source Distribution and Probability 4.
c Here the specific uranium isotope is chosen, 238U is
c three times more likely than 235U.
c
SC5 thorium nuclide
SI5  S  8
SP5  D  1
```

```
c Source Distribution and Probability 5.
c Only one isotope of thorium is possible.
c
SC6 235U photon lines
SI6  L   1.0  2.0 $ E1 ... EI
SP6  D     1    2 $ I1 ... II
SC7 238U photon lines
SI7  L   0.1  0.2 $ E1 ... EI
SP7  D     2    1 $ I1 ... II
SC8 232Th photon lines
S
  L  0.01 0.02 $ E1 ... EI
SP8  D     1    1 $ I1 ... II
```

10.3.1.3 Example 53

```
SDEF  SUR=D1  CEL FSUR D2  ERG FSUR D6
      X FSUR D3  Y FSUR D4  Z FSUR D5
c
SI1  L   11    0
c Source Information 1
c L indicates discrete values, in this case surface 11 or 0
c (meaning the source point is not on a surface).
c
SP1      0.8   0.2
DS2  L   0     88
c Dependent Source 2 (Depends as a function of FSUR).
c L indicates discrete values, in this case cell 0,
c (meaning the point may not be within a cell), or cell 88.
c Note that with Distribution 1, the source point may either be
c on surface 11 (80% probability) or within cell 88
c (20% probability).
c
DS6  S   61    62
SP61     -3   0.98   2.2
SP62     -3   1.05   2.7
c Source Probabilities 61 and 62.
c The -3 indicates the energy is sampled from the Watt Fission
c Spectrum.
c
DS3  S    0    31
SI31     20    30
SP31      0     1
c Source Information and Probabilities for Distribution 3.
c In this case, the 0 on the DS3 card indicates that no
c distribution is given; the default variable will be selected.
c For this case, if surface 11 was selected, the variable
c POS will default to the coordinates 0 0 0.
c If surface 11 was not selected, the source point must be
c within cell 88,and the x coordinate is sampled from a single
c bin histogram with values between 20 and 30.
c Since this value corresponds to a position, the units are cm.
c
DS4  S    0    41
SI41    -17    36
```

```
SP41      0     1
DS5  S    0    51
SI51    -10    10
SP51      0     1
```

Of the particles from this source, 80% start on surface 11, and the rest start in cell 88. When a particle starts in cell 88, its position is sampled, with rejection, in the rectangular polyhedron bounded by $20 < x < 30$, $-17 < y < 36$, and $-10 < z < 10$. When a particle starts on surface 11, its cell is found from its position and direction. The energy spectrum of the particles from surface 11 is different from the energy spectrum of the particles from cell 88. A zero after the `S` option invokes the default variable value.

10.3.1.4 Example 54

The following is an example of using the `Q` option. The low-energy particles from surface `m` come out with a cosine distribution of direction, but the higher-energy particles have a more nearly radial distribution. The energy values on the `DS`2 card need not be the same as any of the e_i on the `SI`1 card.

```
SDEF  ERG=D1  DIR FERG D2  SUR=m
SI1    e1  e2 ... ek
SP1    0   p2 ... pk
DS2       Q   0.3 21   0.8 22   1.7 23   20. 24
SP21    -21  1
SP22    -21  1.1
SP23    -21  1.3
SP24    -21  1.8
```

10.3.2 Beam Sources

By implementing a general transformation on the `SDEF` card in one of two forms; `TR` = `n` or `TR` = `Dn`, a user can point a particle beam in space. In either case a general transformation is applied to a source particle after its coordinates and direction cosines have been determined using the other parameters on the `SDEF` card. Particle coordinates are modified by both rotation and translation; direction cosines are modified by rotation only. This allows the user to rotate the direction of the beam or move the entire beam of particles in space. The `TR` = `Dn` option is particularly powerful because it allows the specification of more than one beam at a time.

10.3.2.1 Example 55: A Single Beam Source

An example of specifying a Gaussian beam follows:

```
Title
c Cell cards
.
.
.
ccc  0     -nnn     $ cookie cutter cell

```

```
c Surface Cards
.
.
.
nnn  SQ   a^-2  b^-2  0  0  0  0  -c^2  0  0  0    $ cookie cutter surface

c Control Cards
SDEF     DIR=1  VEC=0 0 1  X=D1 Y=D2 Z=0  CCC=ccc  TR=n
SP1      -41  fx  0
SP2      -41  fy  0
TRn       x0 y0 z0   cos(phi) -sin(phi) 0   sin cos 0   0 0 1
```

The SDEF card sets up an initial beam of particles traveling along the z axis (DIR = 1, VEC = 0 0 1). Information on the x and y coordinates of particle position is detailed in the two SP cards. On the SDEF card, the specifications X = D1 and Y = D2 indicate that MCNP6 must look for distributions 1 and 2, here given by source probability distributions, SP1 and SP2. The z coordinate is left unchanged ($z = 0$).

Because there is no PAR option in this example, the particle generated by this source will be the one with the lowest **ipt** number in Table 4.3 (i.e., neutron).

The SP cards have three entries. The first entry is -41, which indicates sampling is to be done from a built-in Gaussian distribution. This position Gaussian distribution has the form

$$p(x,y) = \frac{\exp\left\{-\frac{1}{2}\left[\left(\frac{x'}{a}\right)^2 + \left(\frac{y'}{b}\right)^2\right]\right\}}{2\pi ab\left[1 - \exp\left(\frac{-c^2}{2}\right)\right]}. \tag{10.6}$$

The parameters a and b are the standard deviations of the Gaussian in x and y.

The second entry (fx or fy) on the SP cards is the full-width at half-maximum (FWHM) of the Gaussian in either the x or y direction. These must be computed from a and b by the user as follows:

$$f_x = \sqrt{8\ln 2}a = 2.35482a, \tag{10.7}$$

$$f_y = \sqrt{8\ln 2}b = 2.35482b. \tag{10.8}$$

The third entry on the SP cards represents the centroid of the Gaussian in either the x or y direction. We recommend that the user input 0 here, and handle any transformations of the source with a TR card. Using a non-zero value will interfere with the rejection function as specified by the "cookie cutter" option.

Note that in PRINT Table 10 in the MCNP6 output file, the definitions of a, b, and c are different from those discussed above; however, FWHM will be the same as the third entry on the SP cards. The parameter a in PRINT Table 10 differs from the parameter a above by a factor of the square root of two. This is a legacy item from the conversion of the -41 function from time to space.

The user generally does not want the beam Gaussian to extend infinitely in x and y, therefore a cookie cutter option has been included to keep the distribution to a reasonable size. CCC = ccc tells MCNP6 to look at the card labeled ccc (ccc is a user-specified cell number) to define the cutoff volume. The first entry on the ccc card is 0, which indicates a void cell. The second number, -nnn (nnn again is a user-specified number), indicates a surface card within which to accept particles. In the example, this is a SQ surface (a 2-sheet hyperboloid) that is defined as

$$\left(\frac{x'}{a}\right)^2 + \left(\frac{y'}{b}\right)^2 \leq c^2. \tag{10.9}$$

Any particle generated within this cell is accepted; any outside of the cell is rejected. Any well defined surface may be selected, and it is common to use a simple cylinder to represent the extent of a beam pipe.

In this example, a source is generated in an (x', y')-coordinate system with the distribution centered at the origin and the particles traveling in the z' direction. The particle coordinates can be modified to an (x, y)-coordinate system by translation and rotation according to the following equations, where $0 \leq \phi_L \leq \pi$:

$$x = x' \sin\phi_L - y' \cos\phi_L + x_0, \tag{10.10}$$

$$y = x' \cos\phi_L + y' \sin\phi_L + y_0. \tag{10.11}$$

Thus the angle ϕ_L is the angle of rotation of the major axis of the source distribution from the positive y direction in the laboratory coordinate system. If $\cos\phi_L = 0.0$, the angle is 90° and the major axis lies along the x axis. The TRn card in the example above implements this rotation matrix, however the user should note that ϕ_L in the TRn card is equal to $\phi_L - \pi/2$.

10.3.2.2 Example 56: Defining Multiple Beams

The opportunity to specify a probability distribution of transformations on the SDEF card allows the formation of multiple beams which differ only in orientation and intensity. This feature may have applications in radiography or in the distribution of point sources of arbitrary intensity.

The use of a distribution of transformations is invoked by specifying TR = Dn on the SDEF card. The cards SI, SP, and, optionally, SB are used as specified for the SSR card.

```
SIn   L        i1 ... ik
SPn   option   p1 ... pk
SBn   option   b1 ... bk
```

The L option on the SI card is required; input checking ensures this usage for both the SDEF and SSR applications. The "option" on the SP and SB cards may be blank, D, or C. The values i1 ... ik identify k transformations that must be supplied. The content of the SP and SB cards then follows the general MCNP6 rules.

The following example shows a case of three intersecting Gaussian parallel beams, each defined with the parameters $a = 0.2$ cm, $b = 0.1$ cm and $c = 2$ in the notation used previously [§10.3.2.1]. Each beam is normal to the plane of definition.

Beam 1	is centered at $(0, 0, -2)$. The major axis of the beam distribution is along the x axis. The beam is emitted in the $+z$ direction and has relative intensity 1.
Beam 2	is centered at $(-2, 0, 0)$. The major axis of the beam distribution is along the y axis. The beam is emitted in the $+x$ direction and has relative intensity 2.
Beam 3	is centered at $(0, -2, 0)$. The major axis of the beam distribution is along the line defined by $x = z$. The beam is emitted in the $+y$ direction and has relative intensity 3.

The card SBn is used to provide equal sampling from each of the three beams, independent of the relative intensities. The input cards are as follows:

```
Title
c Cell cards
.
.
.
999  0       -999 $ cookie cutter cell

c Surface Cards
.
.
.
999  SQ   25 100 0 0 0 0 -4 0 0 0     $ cookie cutter surface

c Control Cards
SDEF    DIR=1  VEC=0 0 1  X=D1  Y=D2  Z=0  CCC=999  TR=D3
SP1     -41  0.4709640
SP2     -41  0.23584820
SI3    L 1 2 3
SP3      1 2 3
SB3      1 1 1
TR1      0  0 -2 1     0 0     0     1  0     0 0 1
TR2     -2  0  0 0     1 0     0     0  1     1 0 0
TR3      0 -2  0 0.707 0 0.707 0.707 0 -0.707 0 1 0
```

10.3.3 Burning Multiple Materials In a Repeated Structure with Specified Abundance Changes

10.3.3.1 Example 57

In the following example, a 4×4 fuel pin array (created using repeated structures) is burned while material abundance changes are made at various time steps. Portions of the input and output files provided in this example illustrate various BURN card features:

```
burn example
1   1  6.87812e-2  -1          u=2  imp:n=1 vol=192.287 $ fuel
3   2  4.5854e-2    1 -2       u=2  imp:n=1 vol=66.43   $ clad
4   3  7.1594e-2    2          u=2  imp:n=1 vol=370.82  $ water
6   4  6.87812e-2  -1          u=3  imp:n=1 vol=192.287 $ fuel
7   5  4.5854e-2    1 -2       u=3  imp:n=1 vol=66.43   $ clad
8   6  7.1594e-2       2       u=3  imp:n=1 vol=370.82  $ water
10  0              -3  4 -5  6 u=1  imp:n=1 lat=1 fill=0:1 0:1 0:0
                                    2 3 2 3
...

...
BURN  TIME=50,10,500
      MAT=1 4
      POWER=1.0
      PFRAC=1.0 0 0.2
      OMIT= 1,8,6014,7016,8018,9018,90234,91232,95240,95244
            4,8,6014,7016,8018,9018,90234,91232,95240,95244
      BOPT= 1.0, -4
```

```
     AFMIN= 1e-32
     MATVOL= 384.57 384.57
     MATMOD= 2
             1
               1  -4  1  94238 1e-6
             2
               2  -1  2  94238 1e-6  94241 1e-6
                  -4  1  94238 1e-6
...
```

A 4×4 lattice contains universes 2 and 3, which are both repeated twice in the lattice. Universe 2 comprises cells 1, 3, and 4, where cell 1 contains material 1; universe 3 comprises cells 6, 7, and 8, where cell 6 contains material 4. The `MAT` keyword specifies that both materials 1 and 4 will be burned. The combination of the `TIME`, `POWER` and `PFRAC` keywords specify that these materials will be burned first for 50 days at 100% of 1 MW, then decayed for 10 days, and then finally burned for 500 days at 20% of 1 MW.

The `BOPT` keyword specifies that the following options will be invoked: the Q-value multiplier will be set to a value of 1.0, only Tier 1 fission products will be included, the output will be ordered by ZAID and printed at the end of each `KCODE` run, and only tabular transport cross sections will be used. Because tabular transport cross sections do not exist for every isotope that is generated, an `OMIT` keyword is required to omit these isotopes from the transport process. The transmutation of these isotopes is accounted for by sending a 63-group flux from MCNP6 to be matched to a 63-group cross-section set within CINDER90. These are energy integrated to determine a total collision rate. The `OMIT` keyword in the example omits eight isotopes from material 1 and eight isotopes from material 4. The `AFMIN` keyword states that only isotopes possessing an atom fraction below 10^{-32} will be omitted from the transport calculation.

Because there are repeated structures in the example a `MATVOL` keyword is required to calculate the track-length-estimated reaction rates in each repeated structure. Because material 1 and 4 are repeated twice and each material possesses a volume of 192.287 cm^3, `MATVOL` keyword entries of $192.287 \times 2 = 384.57$ are required for each material being burned.

A `MATMOD` keyword is used to manually change the abundance of certain isotopes at specified time steps. In this example, manual isotope abundance changes are to be completed at two time steps. At time step 1, material 4 will have the atom density of isotope 94238 changed to 10^{-6} atoms/b-cm. At time step 2, the atom densities of isotopes 94238 and 94241 in material 1 both will be revised to 10^{-6} atoms/b-cm. Also in step 2, the atom density of isotope 94238 in material 4 will be set to 10^{-6} atoms/b-cm.

`PRINT` Table 210 contains the burnup summary table:

```
1burnup summary table by material                                          print table 210


 neutronics and burnup data

 step  duration     time      power      keff      flux     ave. nu  ave. q     burnup     source
        (days)     (days)     (MW)                                             (GWd/MTU)  (nts/sec)
   0  0.000E+00  0.000E+00  1.000E+00  1.54021  7.715E+14    2.452   200.979  0.000E+00  7.616E+16
   1  5.000E+01  5.000E+01  1.000E+00  1.50987  7.945E+14    2.473   201.411  7.183E+00  7.664E+16
   2  1.000E+01  6.000E+01  0.000E+00  1.51150  0.000E+00    2.474   201.448  7.183E+00  0.000E+00
   3  5.000E+02  5.600E+02  2.000E-01  1.43413  1.699E+14    2.510   202.199  2.155E+01  1.550E+16
...
```

The burnup summary table contains information regarding the entire burn system. Each time step is listed with the corresponding time duration and actual specified time. The next six columns list the power used for

the flux normalization, k_{eff}, energy integrated system averaged flux, system averaged neutrons per fission and recoverable energy per fission, and burnup. Finally, the production rate is listed in the source column.

Since both materials 1 and 4 were burned in the example, individual burn material burnup information is also available. The available information includes: time step, time duration, actual time, fission power fraction, and individual material burnup:

```
Individual Material Burnup

 Material #:    1

 step  duration     time    power fraction    burnup
        (days)     (days)                    (GWd/MTU)
   0  0.000E+00  0.000E+00    5.015E-01      0.000E+00
   1  5.000E+01  5.000E+01    5.016E-01      7.205E+00
   2  1.000E+01  6.000E+01    5.002E-01      7.205E+00
   3  5.000E+02  5.600E+02    5.002E-01      2.158E+01

 Material #:    4

 step  duration     time    power fraction    burnup
        (days)     (days)                    (GWd/MTU)
   0  0.000E+00  0.000E+00    4.985E-01      0.000E+00
   1  5.000E+01  5.000E+01    4.984E-01      7.161E+00
   2  1.000E+01  6.000E+01    4.998E-01      7.161E+00
   3  5.000E+02  5.600E+02    4.998E-01      2.152E+01
...
```

The fission power fraction is calculated by taking the ratio of the fission power in a particular material to the sum of all burn materials. Fission power fractions are only related to fissions in burn materials as

$$\text{power fraction} = \frac{(\Phi\Sigma_{\text{f}}VQ)_i}{\sum_i (\Phi\Sigma_{\text{f}}VQ)_i}. \tag{10.12}$$

The individual material burnup is calculated by

$$\text{burnup} = \text{burnup}_{\text{prev. step}} + \frac{\text{Power Level} \times \text{Power Fraction} \times \text{Time} \times \text{PFRAC}}{\text{MTU}}. \tag{10.13}$$

The time-dependent isotope buildup/depletion is listed after the burnup summary information. The isotope buildup/depletion for each individual material is given at each time step. The information is further subdivided into actinide and non-actinide categories:

```
nuclide data are sorted by increasing zaid for material   1 volume  3.8457E+02 (cm**3)

 actinide inventory for material   1 at end of step  0, time 0.000E+00 (days), power 1.000E+00 (MW)

  no. zaid      mass     activity   spec.act.  atom den.  atom fr.   mass fr.
                (gm)       (Ci)     (Ci/gm)    (a/b-cm)
   1  90231  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   2  90232  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   3  90233  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   4  91233  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   5  92234  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   6  92235  3.441E+02  0.000E+00  0.000E+00  2.293E-03  1.000E-01  9.886E-02
```

```
...
...
 actinide inventory for material   1 at end of step  1, time 5.000E+01 (days), power 1.000E+00 (MW)

  no. zaid      mass     activity   spec.act.  atom den.  atom fr.   mass fr.
                (gm)       (Ci)     (Ci/gm)    (a/b-cm)
   1  90231  1.286E-09  6.837E-04  5.315E+05  8.718E-15  3.832E-13  3.723E-13
   2  90232  2.394E-08  2.625E-15  1.097E-07  1.616E-13  7.100E-12  6.929E-12
   3  90233  1.235E-13  4.468E-06  3.618E+07  8.298E-19  3.647E-17  3.574E-17
   4  91233  1.345E-09  2.792E-05  2.075E+04  9.039E-15  3.973E-13  3.894E-13
...
```

At the end of each subdivision there is an accumulation total of the isotope information for that subdivision. Atom and weight fractions calculations are based on the fractions of that specific subdivision.

```
...
     totals  3.455E+03  2.584E+05  7.479E+01  2.275E-02  1.000E+00  1.000E+00
...
...
 nonactinide inventory for material 1 at end of step  0, time 0.000E+00 (days), power 1.000E+00 (MW)

  no.  zaid     mass     activity   spec.act.  atom den.  atom fr.   mass fr.
                (gm)       (Ci)     (Ci/gm)    (a/b-cm)
   1   6012  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   2   6013  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   3   7014  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   4   7015  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   5   8016  4.684E+02  0.000E+00  0.000E+00  4.585E-02  1.000E+00  1.000E+00
...
```

After isotope information for each individual material is given, `PRINT` Table 220 lists the total build/up of all actinides and non-actinides from all materials combined at each of the time steps.

```
...
1burnup summary table summed over all materials                                print table 220

 nuclides with atom fractions below 1.000E-32 for a material are zeroed and deleted from print tables
     after t=0

 nuclide data are sorted by increasing zaid summed over all materials volume  7.6914E+02 (cm**3)

 actinide inventory for sum of materials at end of step  0, time 0.000E+00 (days), power 1.000E+00 (MW)

  no.  zaid      mass     activity   spec.act.  atom den.   atom fr.   mass fr.
                (gm)       (Ci)     (Ci/gm)    (a/b-cm)
   1  90231  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   2  90232  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   3  90233  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   4  91233  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   5  92234  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   6  92235  6.883E+02  0.000E+00  0.000E+00  4.585E-03  1.000E-01  9.886E-02
...
```

10.3.4 Source Subroutine

When possible, you should take advantage of the standard sources provided by the code rather than write a source subroutine. When you write your own source subroutine, you lose features such as sampling from multiple distributions, using dependent distributions, and having frequency prints for each tabular distribution. Additionally, if using next-event estimators (F5 tallies) or DXTRAN spheres, subroutine **SRCDX** is needed.

The standard sources, however, cannot handle all problems. If the general source (SDEF card), surface source (SSR), or criticality source (KCODE card) is unsuitable for a particular application, MCNP6 provides a mechanism to furnish your own source-modeling capability. The absence of SDEF, SSR, or KCODE cards causes MCNP6 to call subroutine **SOURCE**, which you must supply. Subroutine **SOURCE** specifies the coordinates, direction, weight, energy, and time of source particles as listed and defined in §5.8.15. If the value of **PBL%I%IPT** (particle type) set by **STARTP**, which calls **SOURCE**, is not satisfactory, **SOURCE** must also specify **PBL%I%IPT**. **STARTP** sets **IPT** = 1 (neutron) for MODE n, n p, and n p e; sets **IPT** = 2 (photon) for MODE p and p e; and sets **IPT** = 3 (electron) for MODE e. MCNP6 checks the user's source for consistency of cell, surface, direction, and position. If the source direction is anisotropic and there are point detectors or DXTRAN spheres, a **SRCDX** subroutine is also required [§5.8.15].

The following example of a subroutine **SOURCE** uses SIn, SPn, and SBn cards and demonstrates the use of MCNP6 subroutines **SMPSRC**, **ROTAS**, **CHKCEL**, and the function **NAMCHG**. The geometry is a 5-cm-long cylinder centered about the y axis, divided into 5 cells by PY planes at 1-cm intervals. The 1-MeV mono-energetic source is a biased isotropic distribution that is also biased along the y axis. The input distribution cards are

```
SI1   -1  0  1            $ These 3 cards
SP1    0  1  1            $ represent a biased
SB1    0  1  2            $ isotropic distribution.
SI2    0  1  2  3  4  5    $ These 3 cards
SP2    0  4  2  2  1  1    $ represent a biased
SB2    0  1  1  2  2  4    $ distribution in y.
RDUM   1                   $ cylindrical radius
IDUM   2  4  6  8  10      $ source cells
```

This problem can be run with the general source by removing the RDUM and IDUM cards and adding:

```
SDEF  ERG=1  VEC=0 1 0  AXS=0 1 0  DIR=D1  EXT=D2  RAD=D3
SI3     0   1 $ represents a covering surface of radius 1
SP3   -21   1 $ samples from the power law with k=1
```

The example source subroutine is shown in Listing 10.34, which would replace the generic subroutine **SOURCE** that is provided with the MCNP source code.

Listing 10.34: example_source_cylinder.f90.txt

```
subroutine source
  ! dummy subroutine. aborts job if source subroutine is missing.
  ! if nsr=0, subroutine source must be furnished by the user.
  ! at entrance, a random set of uuu,vvv,www has been defined. the
  ! following variables must be defined within the subroutine:
  ! pbl%r%x, pbl%r%y, pbl%r%z, pbl%r%icl, pbl%r%jsu, pbl%r%erg,
  ! pbl%r%wgt, pbl%r%tme and possibly pbl%i%ipt, pbl%r%u, pbl%r%v,
  ! pbl%r%w.
  ! subroutine srcdx may also be needed.

```

```
  use mcnp_params
  use mcnp_global
  use mcnp_interfaces_mod, only: chkcel, namchg, rotas, smpsrc
  use mcnp_debug
  use mcnp_random
  use tskcom, only: uold
  use pblcom, only: pbl

  implicit none

  real(dknd) :: a(3), c, fi, r, th

  ! smpsrc requires an array as the first argument.
  ! create dummy one dimensional array
  real(dknd) :: array(1)

  integer    :: i, ib, imax, itr, j, lev

  intrinsic cos, sin

  pbl%r%wgt=1.0_dknd

  ! rdum(1)--Radius of Source Cylinder
  ! sample radius uniform in area.

  r=rdum(1)*sqrt(rang())

  ! Y coordinate position, probability and bias are
  ! defined in distribution 2 by the SI2, SP2, SB2 cards.
  ! sample for y.
  ! IB returns the index sampled and FI the interpolated fraction.
  ! neither is used in this example.

  call smpsrc(array,2,ib,fi)
  pbl%r%y = array(1)

  ! Sample for X and Z.

  th = 2.0_dknd*pie*rang()
  pbl%r%x = -r*sin(th)
  pbl%r%z =  r*cos(th)

  ! Direction is isotropic but biased in cone along Y axis
  ! Defined as distribution 1 by the SI1, SP1, SB1 cards.
  ! Sample for cone opening C=cos(NU)
  ! Rotas samples a direction U,V,W at an angle ARCCOS(C)
  ! From the reference vector UOLD(3)
  ! and at an azimuthal angle sampled uniformly.

  call smpsrc(array,1,ib,fi)
  c = array(1)
  uold(1) = 0.0_dknd
  uold(2) = 1.0_dknd
  uold(3) = 0.0_dknd

  call rotas(c,uold,a,lev,itr)
  pbl%r%u = a(1)
  pbl%r%v = a(2)
  pbl%r%w = a(3)
```

```
70
71    ! Cell source - find starting cell.
72    ! IDUM(1) - IDUM(5) -- list of source cells on IDUM card.
73    pbl%i%jsu=0
74    j = 1
75    i = 1
76    imax = 5
77    do while ((J /= 0) .and. (i /= imax))
78      pbl%i%icl=namchg(1,idum(I))
79      call chkcel(pbl%i%icl,2,J)
80      i=i+1
81    enddo
82    if (j /= 0) call expire(1,'Source', &
83      & 'Source is not in any cells on the idum card.')
84    pbl%r%erg = 1.0_dknd
85    pbl%r%tme = 0.0_dknd
86    return
87  end subroutine source
```

10.3.5 SRCDX Subroutine

If a user has supplied a subroutine **SOURCE** that does not emit particles isotropically (uniform emission in all directions) and is using either a detector tally or DXTRAN in the calculations, then subroutine **SRCDX** must also be supplied to MCNP6. The structure of this subroutine is the same as for subroutine **SOURCE**, except that usually only a single parameter, PSC, needs to be specified for each detector or set of DXTRAN spheres. PSC as defined in **SRCDX** is used to calculate the direct contribution from the source to a point detector, to the point selected for the ring detector or DXTRAN sphere. Other parameters may also be specified in **SRCDX**. For example, if a quantity such as particle energy and/or weight is directionally dependent, its value must be specified in both subroutines **SOURCE** and **SRCDX**. When using detectors and a subroutine **SOURCE** with an anisotropic distribution, check the direct source contribution to the detectors carefully to see if it is close to the expected result.

In general, it is best to have as few directionally dependent parameters as possible in subroutine **SOURCE**. Directionally dependent parameters must also be dealt with in subroutine **SRCDX**.

The most general function for emitting a particle from the source in the laboratory system can be expressed as $p(\mu, \varphi)$, where μ is the cosine of the polar angle and φ is the azimuthal angle in the coordinate system of the problem. Most anisotropic sources are azimuthally symmetric and $p(\mu, \varphi) = p(\mu)/2\pi$. The quantity $p(\mu)$ is the probability density function for the μ variable only (i.e., $\int p(\mu)\mathrm{d}\mu = 1$, $p(\mu > 0)$). PSC is $p(\mu_0)$, where μ_0 is the cosine of the angle between the direction defining the polar angle for the source and the direction to a detector or DXTRAN sphere point in the laboratory system. MCNP6 includes the 2π in the calculation automatically. Note that $p(\mu_0)$ and hence PSC may have a value greater than unity and must be non-negative. It is valuable to point out that every source must have a cumulative distribution function based on $p(\mu, \varphi)$ from which to sample angular dependence. The probability density function $p(\mu, \varphi)$ needs only to be considered explicitly for those problems with detectors or DXTRAN.

Table 10.2 gives the equations for PSC for six continuous source probability density functions. More discussion of probability density functions is given in §2.5.6.4.6. The isotropic case is assumed in MCNP6; therefore **SRCDX** is required only for the anisotropic case.

As an example of calculating μ_0, consider a spherical surface cosine source (type 2 in Table 10.2) with several point detectors in the problem. Assume that a point on the spherical surface has been selected at which to

[1] The quantities a and b must have values such that PSC is always non-negative and finite over the range of μ_0.

Table 10.2: Continuous Source Distributions and Their Associated PSCs

Source Description	Source Distribution	PSC & Range of μ_0
Isotropic	Uniform	$\begin{cases} 0.5 & -1 \le \mu_0 \le 1 \end{cases}$
Surface Cosine	μ	$\begin{cases} 2\lvert\mu_0\rvert & 0 \le \mu_0 \le 1 \,(\text{or } -1 \le \mu_0 \le 0) \\ 0 & -1 \le \mu_0 < 0 \,(\text{or } 0 < \mu_0 \le 1) \end{cases}$
Point Cosine	$\lvert\mu\rvert$	$\begin{cases} \lvert\mu_0\rvert & -1 \le \mu_0 \le 1 \end{cases}$
Point Cosine[1]	$a + b\mu$	$\begin{cases} \frac{2(a+b\mu_0)}{2a+b} & 0 \le \mu_0 \le 1 \\ \frac{2(a+b\mu_0)}{2a-b} & -1 \le \mu_0 < 0 \\ 0 & -1 \le \mu_0 < 0 \,(\text{or } 0 < \mu_0 \le 1) \end{cases}$
Point Cosine[1]	$a + b\mu\,,\ a \ne 0$	$\begin{cases} \frac{a+b\mu_0}{2a} & -1 \le \mu_0 \le 1 \end{cases}$
Point Cosine[1]	$a + b\lvert\mu\rvert$	$\begin{cases} \frac{a+b\mu_0}{2a+b} & -1 \le \mu_0 \le 1 \end{cases}$

start a particle. The value of μ_0 for a detector is given by the scalar (or dot) product of the two directions; that is,

$$\mu_0 = uu' + vv' + ww', \tag{10.14}$$

where u, v, and w are the direction cosines of the line from the source point to the point detector location and u', v', and w' are the direction cosines for either the outward normal if the surface source is outward or the inward normal if the source is inward.

If $u = u'$, $v = v'$, and $w = w'$, then $\mu_0 = 1$, indicating that the point detector lies on the normal line. The value of PSC for the detector point is

$$\text{PSC} = \begin{cases} 2|\mu_0| & \mu_0 > 0 \,(\mu_0 < 0) \\ 0 & \mu_0 \le 0 \,(\mu_0 \ge 0) \end{cases}, \tag{10.15}$$

where the parenthetical values of μ_0 are for the inward-directed cosine distribution.

For $|\mu_0| < 0.25$ in case 2 of Table 10.2, PSC is less than 0.5, which is the value for an isotropic source. This means that source emissions for these values of $|\mu_0|$ are less probable than the isotropic case for this source distribution. The converse is also true. Note that if $|\mu_0| > 0.5$, PSC is greater than one, which is perfectly valid.

An example of a subroutine **SRCDX** for a surface outward cosine distribution is shown in Listing 10.35.

Listing 10.35: example_srcdx_outward_cosine.f90.txt

```
subroutine srcdx
  ! dummy subroutine for use with user-defined sources

  use mcnp_global
  use mcnp_params
  use tskcom, only: psc
  use pblcom, only: pbl
  use mcnp_debug

  implicit none
```

```
11
12    real(dknd) :: up, vp, wp
13
14    ! Calculate PSC for a surface (Sphere) outward cosine distribution.
15    ! Find the direction cosines for this example based on the source
16    ! point on the sphere (X,Y,Z).
17
18    up=(pbl%r%x-rdum(1))/rdum(4)
19    vp=(pbl%r%y-rdum(2))/rdum(4)
20    wp=(pbl%r%z-rdum(3))/rdum(4)
21
22    ! (RDUM(1),RDUM(2),RDUM(3)) are the coordinates of the center
23    ! of the sphere from the RDUM card. RDUM(4) is the radius.
24    ! U,V, and W have been calculated for the current point detector
25    ! in subroutine DDDET.
26
27    psc = 2.0_dknd*max(ZERO,pbl%r%u*up + pbl%r%v*vp + pbl%r%w*wp)
28    return
29  end subroutine srcdx
```

This is basically the technique that is used in MCNP6 to calculate PSC for a spherical surface source in a cosine distribution; the only difference is that MCNP6 uses the cosines of the direction from the center of the sphere that selected the source point because this is normal to the spherical surface. The primed direction cosines were calculated in Listing 10.35 to aid in illustrating this example. The direction cosines u, v, and w as defined in Eq. (10.14) have already been calculated in subroutine **DDDET** when **SRCDX** is called and are available through the **pbl** (particle) object.

For many sources, a discrete probability density function will be used. In this situation, a cumulative distribution function $P(\mu)$ is available and is defined as

$$P(\mu) = \int_{-1}^{\mu} p(\mu')\mathrm{d}\mu' \text{ and } P_{i+1} = \sum_{j=1,i} p_j \Delta\mu_j, \tag{10.16}$$

where p_j is an average value of the probability density function in the interval $\Delta\mu_j$. Thus, the probability density function is a constant p_j in the interval $\Delta\mu_j$. For this case, there are N values of P_i with $P_1 = 0$, $P_{N+1} = 1.0$ and $P_{i-1} < P_i$. Each value of P_i has an associated value of μ_i. Because PSC is the derivative of $P(\mu_0)$, then

$$\text{PSC} = \frac{P_i - P_{i-1}}{\mu_i - \mu_{i-1}}, \ \mu_{i-1} \leq \mu_0 < \mu_i. \tag{10.17}$$

This is an average PSC between μ_{i-1} and μ_i and is also an average value of $p(\mu)$ in the specified range of μ.

Frequently, the cumulative distribution function is divided into N equally probable intervals. For this case,

$$\text{PSC} = \frac{1}{N}\frac{1}{\mu_i - \mu_{i-1}}. \tag{10.18}$$

This is precisely the form used in MCNP6 for calculating contributions to the point detector for elastic scattering with $N = 32$.

An example of a subroutine **SRCDX** for a discrete probability density function is given in the example that follows. This subroutine would work with the subroutine **SOURCE** example in §10.3.4, and would calculate PSC = 1/2 for the isotropic distribution.

A biased anisotropic distribution can also be represented by

```
SIn        -1 ...  1
SPn     0  p1 ... pN
SBn     0  q1 ... qN
```

A reference vector u', v', w' for this distribution is also needed.

The subroutine **SOURCE** input cards can be modified for this case by changing the SI 1, SP 1, SB 1, and RDUM cards as follows:

```
SI1    -1  0  1    $ These 3 cards
SP1     0  2  1    $ represent a biased
SB1     0  1  2    $ anisotropic distribution.
RDUM    1  0  1  0 $ cylindrical radius and reference vector
```

SOURCE would sample this anisotropic distribution and **SRCDX** would calculate the appropriate PSC is shown in Listing 10.36.

Listing 10.36: example_srcdx_biased_anisotropic.f90.txt

```
subroutine srcdx
  ! dummy subroutine for use with user-defined sources

  use mcnp_params
  use mcnp_global
  use tskcom, only: psc
  use pblcom, only: pbl
  use mcnp_debug

  implicit none

  real(dknd) :: am
  integer    :: i

  ! The variably dimensioned block SPF holds the SI, SP, SB arrays.

  ! RDUM(2), RDUM(3),RDUM(4) -- Directional cosines for the source reference
  ! direction.

  am = pbl%r%u*rdum(2) + pbl%r%v*rdum(3) + pbl%r%w*rdum(4)

  ! KSD(4,1) is the length of the distribution.
  ! KSD(13,1) is the offset into the SPF block.

  do i=1,ksd(4,1)-1
    if ( spf(i,ksd(13,1)+1) <= am .and. spf(i+1,ksd(13,1)+1) >= am) then
      psc = (spf(i+1,ksd(13,1)+2)-spf(i,ksd(13,1)+2))/ &
        &   (spf(i+1,ksd(13,1)+1)-spf(i,ksd(13,1)+1))
      psc = psc * spf(i+1,ksd(13,1)+3)
      exit
    else
     psc = ZERO
    endif
  enddo

  return
end subroutine srcdx
```

Caution

It is important to note that the case in Listing 10.36 applies only when the source is anisotropic with azimuthal symmetry.

For the general case,

$$\text{PSC} = 2\pi p(\mu_0, \varphi_0). \tag{10.19}$$

The 2π factor must be applied by the user because MCNP6 assumes azimuthal symmetry and, in effect, divides the user-defined PSC by 2π.

For a continuous $p(\mu, \varphi)$ function, PSC is calculated as above. In the case of a discrete probability density function,

$$\text{PSC} = 2\pi \cdot \overline{p(\mu_0, \varphi_0)} = \frac{2\pi(P_i - P_{i-1})}{(\mu_i - \mu_{i-1})(\varphi_i - \varphi_{i-1})} = \frac{2\pi(P_i - P_{i-1})}{\Delta\mu_i \Delta\varphi_i}, \tag{10.20}$$

where $\mu_{i-1} \le \mu_0 < \mu_i$, $\varphi_{i-1} \le \varphi_0 < \varphi_i$, and $\overline{p(\mu_0, \varphi_0)}$ is an average probability density function in the specified values of μ_0 and φ_0 and $P_i - P_{i-1}$ is the probability of selecting μ_0 and φ_0 in these intervals. For N equally probable bins and n equally spaced $\Delta\varphi$s, each $2\pi/n$ wide,

$$\text{PSC} = \frac{n}{N}\frac{1}{\Delta\mu_i}. \tag{10.21}$$

Another way to view this general case is by considering solid angles on the unit sphere. For an isotropic source, the probability $(P_i - P_{i-1})$ of being emitted into a specified solid angle is the ratio of the total solid angle (4π) to the specified solid angle $(\Delta\mu\Delta\varphi)$. Then, PSC $\equiv 0.5$. Thus, for the general case (normalized to PSC $\equiv 0.5$ for an isotropic source)

$$\text{PSC} = \frac{(0.5)(P_i - P_{i-1})4\pi}{\Delta\mu_i \Delta\varphi_i} = \frac{2\pi(P_i - P_{i-1})}{\Delta\mu_i \Delta\varphi_i}. \tag{10.22}$$

Note that PSC is greater than 0.5 if the specified solid angle $\Delta\mu\Delta\varphi_i$ is less than $(P_i - P_{i-1})4\pi$. This is the same as the previous general expression.

Caution

Be careful when using your own subroutine **SOURCE** with either detectors or DXTRAN. This caution applies to the calculation of the direct contribution from the source to a point detector, point on a ring, or point on a DXTRAN sphere. Not only is there the calculation of the correct value of PSC for an anisotropic source, but there may also be problems with a biased source.

For example, if an isotropic source is biased to start only in a cone of a specified angle (for example, Ψ), the starting weight of each particle should be WGT $\times (1 - \cos\Psi)/2$, where WGT is the weight of the unbiased source (that is, WGT is the expected weight from a total source). The weight in **SRCDX** *must* be changed to the expected weight WGT to calculate the direct contribution to a point detector correctly if PSC is defined to be 0.5.

This example can be viewed in a different way. The probability density function for the above biased source is

$$p(\mu) = \begin{cases} \dfrac{1}{1 - \cos\Psi} & \cos\Psi \le \mu \le 1 \\ 0 & -1 \le \mu < \cos\Psi \end{cases}. \tag{10.23}$$

Thus, PSC is this constant everywhere in the cone and zero elsewhere. Multiplying this PSC and biased starting weight gives

$$\text{WGT} \times (1 - \cos\Psi) \times \frac{0.5}{1 - \cos\Psi}$$

or WGT $\times$ 0.5, which is the expected result for an isotropic source.

Another source type that requires caution is a user-supplied source that is energy-angle correlated. For example, assume a source has a Gaussian distribution in energy where the mean of the Gaussian is correlated in some manner with μ. In subroutine **SRCDX**, the μ_0 to a point detector must be calculated and the energy of the starting particle must be sampled from the Gaussian based on this μ_0. This must be done for each point detector in the problem, thus guaranteeing that the direct source contribution to each detector will be from the proper energy spectrum. The original energy of the starting particle as well as all the other starting parameters selected in subroutine **SOURCE** are automatically restored after the direct source contribution to detectors is made. Thus, the subroutine **SOURCE** is still sampled correctly.

10.4 Material Examples

10.4.1 Table Data/Model Physics Mix and Match

Consider a neutron problem with deuterium and tritium. The available deuterium library contains values valid up to 150 MeV, but the tritium library goes up to only 20 MeV. Previously, either neutron physics models above 20 MeV (neglecting the deuterium table data up to 150 MeV) or nuclear data tables below 150 MeV (using the 20-MeV tritium data throughout the 20–150-MeV range) had to be used. Using the mix-and-match capability available through the *tabl* parameter of the PHYS:$\mathscr{P}$ card, the user can specify that deuterium use tables up to 150 MeV and use physics models above 150 MeV and that tritium use data tables up to 20 MeV and use physics models above 20 MeV.

Figure 10.47 shows an example of the energy-matching capability. The 100-MeV neutrons are incident on an 8.433-cm-long, 3.932-cm-radius BGO crystal. The crystal contains 21% bismuth, 16% germanium, and 63% oxygen. Assume no germanium libraries are available. The solid line represents flux in the crystal with the full mix-and-match capability, which uses all libraries up to their energy limits and physics models above those limits and for germanium. The dashed-line calculation uses the old method of substituting arsenic for the missing germanium library, using the libraries up to 20 MeV and using physics models above. The dotted line uses bismuth and oxygen libraries up to their limits of 150 MeV; the arsenic library is used up to its limit of 20 MeV, and then the 20-MeV data are used from 20 to 150 MeV; above 150 MeV, physics models are used for all three nuclides. This last option is least desirable but often was used in past code versions to take advantage of the 150-MeV libraries, even though many data libraries go only to 20 MeV.

10.5 Physics Models

10.5.1 Neutron Production from a Spallation Target

One of the fundamental quantities of interest in most spallation target applications is the number of neutrons produced per beam particle incident on target. For targets fed by proton accelerators, this quantity is typically denoted as "n/p". Here, we demonstrate how one goes about calculating this quantity for a simple target geometry using MCNP6.

The geometry consists of a simple right circular cylinder of lead, 10 cm in diameter by 30 cm long. A beam of 1-GeV protons is launched onto the target. The beam has a 7-cm-diameter spot size, with a parabolic spatial profile. See Fig. 10.48.

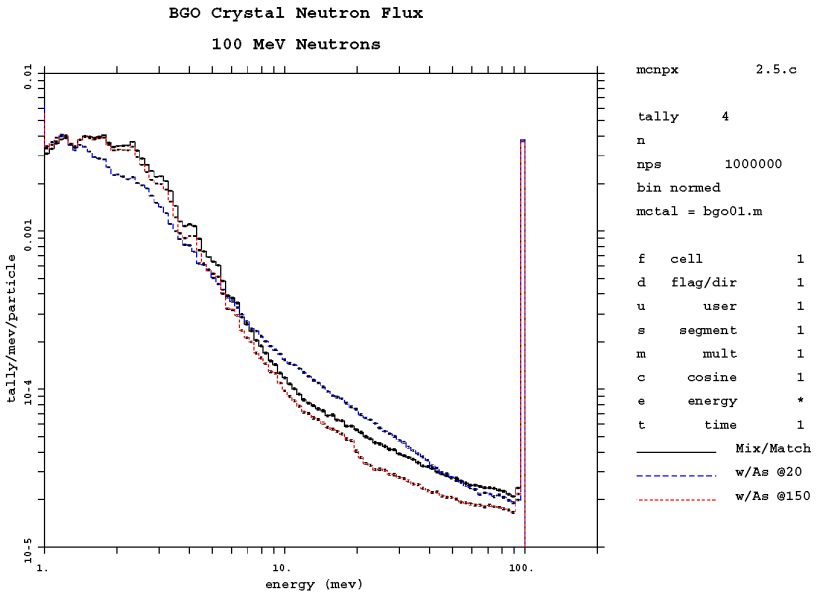


Figure 10.47: Comparison of different germanium library and model options.

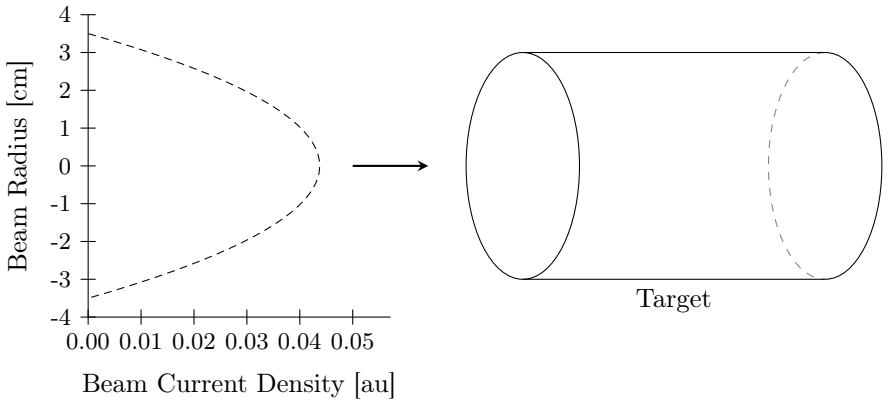


Figure 10.48: Neutron production from a spallation target.

Table 10.3: Neutron Problem Summaries

Case	INC Model	Particles transported
base	Bertini	`n h /`
1	Bertini	`n h / d t s a`
2	ISABEL	`n h /`
3	CEM	`n h /`
4	INCL	`n h /`

In MCNP6, net neutron production is tallied implicitly and is provided by default in the problem summary for neutrons. The problem summary shows net neutron production resulting from nuclear interactions (the component that accounts for neutron production by all particles transported using INC/Preequilibrium/Evaporation physics) and net production by (n,xn) reactions (neutrons created in inelastic nuclear interactions by neutrons below the transition energy, using evaluated nuclear data). Net production from nuclear interactions is given by the difference of the neutron weights in the "neutron creation" and "neutron loss" columns. A similar approach is taken to calculate net (n,xn) production. Net neutron production may also be calculated by realizing that the only loss mechanisms for neutrons are escape and capture. The sum of the weights in the "neutron loss" column under "escape" and "capture" is thus equal to the net neutron production. The values listed in the problem summary are "collision estimators," meaning they are tallied when a collision occurs during transport. Uncertainties are not calculated by MCNP6 for these collision-estimated quantities. A reasonable upper limit on the relative uncertainty would be given by the inverse square root of the number of source particles launched.

We provide here four different variations for the calculation of net neutron production for this simple target geometry. In the "base case" we transport protons, neutrons, and charged pions. The transition energy between LAHET physics and neutron transport using tabular nuclear data is set to the default ($tabl = -1$), which means that "mix and match" (see §5.6.3) will be turned on and the ENDF/B-VI.6 neutron libraries are used. All protons are transported using LAHET physics. Nucleon and pion interactions simulated by LAHET physics use the Bertini intranuclear cascade model. Variations from this base case are outlined in Table 10.3. For each case 20,000 source protons were transported. Note that in MCNPX, Bertini INC was the default physics option. In MCNP6, however, the default option is CEM03.03; therefore, we need to specify the following `LCA` card to activate Bertini INC: `lca 8j 0`. In this example we refer to Bertini INC as the "base case."

For the sake of brevity, we reproduce here just the neutron problem summaries from the MCNP6 output decks.

10.5.1.1 Base Case

The base-case MCNP input file is shown in Listing 10.37 with corresponding output excerpt shown in Listing 10.38.

Listing 10.37: example_bertini_inc_1.mcnp.inp.txt

```
Sample problem: spallation target
c     neutron production with Bertini physics
c     EJ Pitcher, 1 Nov 99
c     MR James, 31 Oct 2007
C     SG Mashnik, February 27, 2013
c
c --- cell cards ---
c
1 1 -11.4   1 -2 -3     $ Pb target
2 0         (-1:2:3) -4 $ bounding sphere
```

```
3 0         4           $ outside universe

c --- surface cards ---
c
1 pz   0.0
2 pz  30.0
3 cz   5.0
4 so  90.0

c --- material cards ---
c
c     Material #1: Pb without Pb-204
m1    82206 0.255  82207 0.221  82208 0.524 nlib=.66c hlib=.24h
c
c --- data cards ---
mode        n h /
imp:n,h,/   1 1 0
phys:n      1000. j j
phys:h      1000. j j
c lca         j j j
lca         8j  0
nps         20000
prdmp       j -30 j 1
c
c --- source definition ---
c     1-GeV proton beam, 7-cm-diam, parabolic spatial profile
sdef   sur 1 erg 1000. dir 1 vec 0. 0. 1. rad d1 pos 0. 0. 0. par 9
si1  a   0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3
         1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7
         2.8 2.9 3.0 3.1 3.2 3.3 3.4 3.5
sp1      0.00000 0.09992 0.19935 0.29780 0.39478 0.48980 0.58237
         0.67200 0.75820 0.84049 0.91837 0.99135 1.05894 1.12065
         1.17600 1.22449 1.26563 1.29894 1.32392 1.34008 1.34694
         1.34400 1.33078 1.30678 1.27151 1.22449 1.16522 1.09322
         1.00800 0.90906 0.79592 0.66808 0.52506 0.36637 0.19151
         0.00000
```

Listing 10.38: example_bertini_inc_1.mcnp.out.txt

```
     Sample problem: spallation target     probid =  12/28/20 15:18:39
*********************************************************
Calls to event-generator models, counted by particle type.

particle        BERTINI            CEM           INCL         ISABEL         LAQGSM      LAQGSM_H1            HYD        ELASTIC

neutron            5506              0              0              0              0              0              0           2519
proton            27269              0              0              0              0              0              0          13394
pi_plus             595              0              0              0              0              0              0            273

totals            33370              0              0              0              0              0              0          16186


neutron creation    tracks      weight        energy           neutron loss        tracks      weight        energy
                                (per source particle)                                        (per source particle)

source                   0    0.            0.                 escape             347410    1.7355E+01    2.1397E+02
nucl. interaction   292426    1.4621E+01    3.1152E+02         energy cutoff           0    0.            0.
particle decay           0    0.            0.                 time cutoff             0    0.            0.
weight window            0    0.            0.                 weight window           0    0.            0.
cell importance          0    0.            0.                 cell importance         0    0.            0.
weight cutoff            0    0.            0.                 weight cutoff           0    0.            0.
e or t importance        0    0.            0.                 e or t importance       0    0.            0.
dxtran                   0    0.            0.                 dxtran                  0    0.            0.
forced collisions        0    0.            0.                 forced collisions       0    0.            0.
exp. transform           0    0.            0.                 exp. transform          0    0.            0.
upscattering             0    0.            0.                 downscattering          0    0.            9.3802E+00
photonuclear             0    0.            0.                 capture                 0    1.3592E-02    7.2872E-02
(n,xn)               75783    3.7856E+00    1.8490E+01         loss to (n,xn)      24395    1.2182E+00    4.7746E+01
prompt fission           0    0.            0.                 loss to fission         0    0.            0.
delayed fission          0    0.            0.                 nucl. interaction    3587    1.7935E-01    6.0729E+01
prompt photofis          0    0.            0.                 particle decay          0    0.            0.
tabular boundary         0    0.            0.                 tabular boundary        0    0.            0.
tabular sampling      7183    3.5915E-01    1.8827E+00         elastic scatter         0    0.            0.
    total           375392    1.8766E+01    3.3189E+02             total           375392    1.8766E+01    3.3189E+02

  number of neutrons banked                 350997        average time of (shakes)              cutoffs
  neutron tracks per source particle     1.8770E+01         escape            5.7124E+00         tco   1.0000E+33
  neutron collisions per source particle 2.6493E+01         capture           4.6539E-01         eco   0.0000E+00
  total neutron collisions                  529856         capture or escape 5.7083E+00         wc1  -5.0000E-01
  net multiplication              0.0000E+00 0.0000         any termination   5.2838E+00         wc2  -2.5000E-01
```

The two methods for calculating total neutron production give the following results:

$$\begin{array}{ccccccc} \text{net nuclear interactions} & + & \text{net (n,}x\text{n)} & + & \text{tabular sampling} & & \\ (14.621 - 0.179) & + & (3.786 - 1.218) & + & 0.359 & = 17.369 \text{ n/p} \end{array}$$

and

$$\begin{array}{ccccc} \text{escapes} & + & \text{captures} & & \\ 17.355 & + & 0.014 & = 17.369 \text{ n/p} & . \end{array}$$

Both methods give the same answer. Since "escapes + captures" is easier to calculate, this is the method typically used. A reasonable upper limit on the relative uncertainty of n/p is $\sqrt{20000} \approx 0.7\%$.

10.5.1.2 Case 1

In the first variation we transport not only nucleons (denoted by the symbols `n` and `h` on the `MODE` card) and charged pions (`/`), but also light ions (deuterons, tritons, ^{3}He, and alphas, denoted by `d`, `t`, `s`, and `a`, respectively). Thus, the only differences are on the `MODE` and `IMP` cards, with the changed cards shown in Listing 10.39.

Listing 10.39: Excerpt from example_bertini_inc_2.mcnp.inp.txt

```
mode        n h / d t s a
imp:n,h,/,d,t,s,a   1 1 0
```

Note that nuclear interactions by light ions are simulated using the ISABEL INC model. The problem summary for this case is shown in Listing 10.40.

Listing 10.40: example_bertini_inc_2.mcnp.out.txt

```
     Sample problem: spallation target     probid =  12/28/20 15:32:46
************************************************************
Calls to event-generator models, counted by particle type.

particle        BERTINI            CEM           INCL         ISABEL         LAQGSM      LAQGSM_H1            HYD        ELASTIC

neutron            5486              0              0              0              0              0              0           2550
proton            27337              0              0              0              0              0              0          13406
pi_plus             618              0              0              0              0              0              0            265
deuteron              0              0              0             10              0              0              0              9
triton                0              0              0              2              0              0              0              2
alpha                 0              0              0              0              0              0              0              1

totals            33441              0              0             12              0              0              0          16233


neutron creation    tracks      weight        energy           neutron loss        tracks      weight        energy
                                (per source particle)                                        (per source particle)

source                   0    0.            0.                  escape              348003    1.7384E+01    2.1419E+02
nucl. interaction   293076    1.4654E+01    3.1260E+02          energy cutoff            0    0.            0.
particle decay           0    0.            0.                  time cutoff              0    0.            0.
weight window            0    0.            0.                  weight window            0    0.            0.
cell importance          0    0.            0.                  cell importance          0    0.            0.
weight cutoff            0    0.            0.                  weight cutoff            0    0.            0.
e or t importance        0    0.            0.                  e or t importance        0    0.            0.
dxtran                   0    0.            0.                  dxtran                   0    0.            0.
forced collisions        0    0.            0.                  forced collisions        0    0.            0.
exp. transform           0    0.            0.                  exp. transform           0    0.            0.
upscattering             0    0.            0.                  downscattering           0    0.            9.3861E+00
photonuclear             0    0.            0.                  capture                  0    1.3764E-02    7.4671E-02
(n,xn)               76089    3.8008E+00    1.8531E+01          loss to (n,xn)       24536    1.2252E+00    4.7737E+01
prompt fission           0    0.            0.                  loss to fission          0    0.            0.
delayed fission          0    0.            0.                  nucl. interaction     3628    1.8140E-01    6.1518E+01
prompt photofis          0    0.            0.                  particle decay           0    0.            0.
tabular boundary         0    0.            0.                  tabular boundary         0    0.            0.
tabular sampling      7002    3.5010E-01    1.7728E+00          elastic scatter          0    0.            0.
    total           376167    1.8805E+01    3.3291E+02              total          376167    1.8805E+01    3.3291E+02

  number of neutrons banked                 351631        average time of (shakes)              cutoffs
  neutron tracks per source particle     1.8808E+01          escape            5.7390E+00          tco   1.0000E+33
```

```
42  neutron collisions per source particle 2.6648E+01          capture           4.5961E-01          eco   0.0000E+00
43  total neutron collisions                    532968          capture or escape 5.7348E+00          wc1  -5.0000E-01
44  net multiplication              0.0000E+00 0.0000          any termination   5.3065E+00          wc2  -2.5000E-01
```

Net neutron production for this case is 17.398 n/p, or 0.17% above the base case value. Examination of the net nuclear interactions and net (n,xn) figures show very similar results to the base case. The implication of this result is that we need not concern ourselves with light ion transport if the quantity with which we are concerned is related solely to neutrons, as neutron production by light ions is small when we start with a proton beam.

10.5.1.3 Case 2

In the second variation we replace the Bertini INC model used in the base case for the simulation of nucleon and pion interactions with nuclei by the ISABEL INC model (in this example, both INC models utilize the same GCCI level-density model). We invoke the ISABEL INC model by changing the `LCA` card and adding the `LCB` card as shown in Listing 10.41.

Listing 10.41: Excerpt from example_isabel_inc_1.mcnp.inp.txt

```
lca         2j 2 5j 0
lcb         4j 1000 1000
```

This changes the value of the variable ***iexisa*** (3rd value on the `LCA` card) from its default value of 1 to 2, and increases the ***flenb5*** and ***flenb6*** parameters controlling the ISABEL INC transition energies to the proton source energy of 1 GeV. The neutron problem summary for this case is shown in Listing 10.42.

Listing 10.42: example_isabel_inc_1.mcnp.out.txt

```
     Sample problem: spallation target     probid =  08/17/22 21:05:49
*********************************************************
Calls to event-generator models, counted by particle type.

particle        BERTINI            CEM           INCL         ISABEL         LAQGSM      LAQGSM_H1            HYD        ELASTIC

neutron               0              0              0           6904              0              0              0           2800
proton                0              0              0          30878              0              0              0          13770
pi_plus               0              0              0            703              0              0              0            278

totals                0              0              0          38485              0              0              0          16848


neutron creation    tracks      weight        energy           neutron loss        tracks      weight        energy
                                (per source particle)                                        (per source particle)

source                   0    0.            0.                  escape             327457    1.6358E+01    2.2057E+02
nucl. interaction   275042    1.3752E+01    3.2733E+02          energy cutoff           0    0.            0.
particle decay           0    0.            0.                  time cutoff             0    0.            0.
weight window            0    0.            0.                  weight window           0    0.            0.
cell importance          0    0.            0.                  cell importance         0    0.            0.
weight cutoff            0    0.            0.                  weight cutoff           0    0.            0.
e or t importance        0    0.            0.                  e or t importance       0    0.            0.
dxtran                   0    0.            0.                  dxtran                  0    0.            0.
forced collisions        0    0.            0.                  forced collisions       0    0.            0.
exp. transform           0    0.            0.                  exp. transform          0    0.            0.
upscattering             0    0.            0.                  downscattering          0    0.            8.4098E+00
photonuclear             0    0.            0.                  capture                 0    1.2987E-02    6.8450E-02
(n,xn)               71266    3.5599E+00    1.7703E+01          loss to (n,xn)      22514    1.1243E+00    4.5670E+01
prompt fission           0    0.            0.                  loss to fission         0    0.            0.
delayed fission          0    0.            0.                  nucl. interaction    4026    2.0130E-01    7.2281E+01
prompt photofis          0    0.            0.                  particle decay          0    0.            0.
tabular boundary         2    1.0000E-04    1.4997E-02          tabular boundary        2    1.0000E-04    1.4997E-02
tabular sampling      7689    3.8445E-01    1.9679E+00          elastic scatter         0    0.            0.
    total           353999    1.7697E+01    3.4702E+02              total          353999    1.7697E+01    3.4702E+02

  number of neutrons banked                 331485        average time of (shakes)              cutoffs
  neutron tracks per source particle     1.7700E+01         escape            5.7844E+00         tco   1.0000E+33
  neutron collisions per source particle 2.5083E+01         capture           4.8834E-01         eco   0.0000E+00
  total neutron collisions                  501662         capture or escape 5.7802E+00         wc1  -5.0000E-01
  net multiplication              0.0000E+00 0.0000        any termination   5.3479E+00         wc2  -2.5000E-01
```

In the MCNP6 summary table, information is provided about the event-generator models used in the calculation. This information assists users to better understand the results and how they were calculated. For this particular case, we specified `lca 2J 2 5J 0` and `lcb 4J 1000 1000` to invoke ISABEL INC for all energies 1 GeV and lower. Note the net neutron production calculated using the ISABEL INC model is 16.371 n/p, which is 5.75% below the predicted base case value using the Bertini INC model. This result is consistent with other studies that reveal slightly lower neutron production resulting from ISABEL as compared to Bertini.

10.5.1.4 Case 3

In this variation, we use the CEM03.03 model for neutrons, protons, and pions. CEM is turned on by setting the 9th entry of the LCA card, $icem = 1$, as shown in Listing 10.43.

Listing 10.43: Excerpt from example_cem_physics_1.mcnp.inp.txt

```
lca         8j  1
```

Because CEM03.03 is the default physics option in MCNP6, no LCA card is required to invoke CEM03.03 physics. Unlike the other INC models in the code, CEM03.03 includes its own Modified Preequilibrium Model (MEM) [200, 204] and its own extension (see details in [216]) of the Generalized Evaporation Model (GEM) by Furihata [259]. Therefore, the Multistage Pre-equilibrium Model (MPM) by Prael and Bozoian [256] and evaporation model settings used for the Bertini INC and ISABEL models have no effect when CEM03.03 is specified.

The neutron summary table for this case is shown in Listing 10.44.

Listing 10.44: example_cem_physics_1.mcnp.out.txt

```
     Sample problem: spallation target     probid =  12/28/20 16:23:29
*********************************************************
Calls to event-generator models, counted by particle type.

particle        BERTINI            CEM           INCL         ISABEL         LAQGSM      LAQGSM_H1            HYD        ELASTIC

neutron               0           3640              0              0              0              0              0           2513
proton                0          18321              0              0              0              0              0          12801
pi_plus               0            340              0              0              0              0              0            209

totals                0          22301              0              0              0              0              0          15523


neutron creation    tracks      weight        energy           neutron loss        tracks      weight        energy
                                (per source particle)                                        (per source particle)

source                 0    0.            0.                  escape             366645    1.8316E+01    2.0833E+02
nucl. interaction   317483    1.5874E+01    3.1426E+02          energy cutoff           0    0.            0.
particle decay           0    0.            0.                  time cutoff             0    0.            0.
weight window            0    0.            0.                  weight window           0    0.            0.
cell importance          0    0.            0.                  cell importance         0    0.            0.
weight cutoff            0    0.            0.                  weight cutoff           0    0.            0.
e or t importance        0    0.            0.                  e or t importance       0    0.            0.
dxtran                   0    0.            0.                  dxtran                  0    0.            0.
forced collisions        0    0.            0.                  forced collisions       0    0.            0.
exp. transform           0    0.            0.                  exp. transform          0    0.            0.
upscattering             0    0.            0.                  downscattering          0    0.            1.1376E+01
photonuclear             0    0.            0.                  capture                 0    1.4435E-02    8.0051E-02
(n,xn)               72827    3.6374E+00    1.6405E+01          loss to (n,xn)      24677    1.2321E+00    4.3468E+01
prompt fission           0    0.            0.                  loss to fission         0    0.            0.
delayed fission          0    0.            0.                  nucl. interaction    3640    1.8200E-01    6.8654E+01
prompt photofis          0    0.            0.                  particle decay          0    0.            0.
tabular boundary         0    0.            0.                  tabular boundary        0    0.            0.
tabular sampling      4652    2.3260E-01    1.2436E+00          elastic scatter         0    0.            0.
    total           394962    1.9744E+01    3.3191E+02              total           394962    1.9744E+01    3.3191E+02

  number of neutrons banked                 370285        average time of (shakes)              cutoffs
  neutron tracks per source particle     1.9748E+01          escape            5.4050E+00          tco   1.0000E+33
  neutron collisions per source particle 2.8020E+01          capture           4.2740E-01          eco   0.0000E+00
  total neutron collisions                  560403          capture or escape 5.4011E+00          wc1  -5.0000E-01
  net multiplication              0.0000E+00 0.0000          any termination   5.0148E+00          wc2  -2.5000E-01
```

Note the net neutron production calculated with the CEM03.03 model is 18.33 n/p, which is 5.53% above the value predicted by the Bertini INC model.

10.5.1.5 Case 4

In the final variation from the base case we use the INCL model coupled with the ABLA evaporation mode by changing the `LCA` card as shown in Listing 10.45.

Listing 10.45: example_incl_abla_physics_1.mcnp.inp.txt

```
lca         8j  2
```

Note: The ABLA evaporation model is automatically chosen when INCL is specified.

The neutron problem summary for this case is shown in Listing 10.46.

Listing 10.46: Excerpt from example_incl_abla_physics_1.mcnp.out.txt

```
     Sample problem: spallation target     probid =  12/28/20 16:26:57
************************************************************
Calls to event-generator models, counted by particle type.

particle        BERTINI            CEM           INCL         ISABEL         LAQGSM      LAQGSM_H1            HYD        ELASTIC

neutron               0              0           8869              0              0              0              0           2464
proton                0              0          42469              0              0              0              0          13674
pi_plus               0              0           1163              0              0              0              0            299

totals                0              0          52501              0              0              0              0          16437


neutron creation    tracks      weight        energy           neutron loss        tracks      weight        energy
                                (per source particle)                                        (per source particle)

source                   0    0.            0.                  escape             333031    1.6637E+01    2.2398E+02
nucl. interaction   272213    1.3611E+01    3.1956E+02          energy cutoff           0    0.            0.
particle decay           0    0.            0.                  time cutoff             0    0.            0.
weight window            0    0.            0.                  weight window           0    0.            0.
cell importance          0    0.            0.                  cell importance         0    0.            0.
weight cutoff            0    0.            0.                  weight cutoff           0    0.            0.
e or t importance        0    0.            0.                  e or t importance       0    0.            0.
dxtran                   0    0.            0.                  dxtran                  0    0.            0.
forced collisions        0    0.            0.                  forced collisions       0    0.            0.
exp. transform           0    0.            0.                  exp. transform          0    0.            0.
upscattering             0    0.            0.                  downscattering          0    0.            9.2432E+00
photonuclear             0    0.            0.                  capture                 0    1.2483E-02    6.7357E-02
(n,xn)               81634    4.0784E+00    2.0691E+01          loss to (n,xn)      25419    1.2695E+00    5.2979E+01
prompt fission           0    0.            0.                  loss to fission         0    0.            0.
delayed fission          0    0.            0.                  nucl. interaction    3588    1.7940E-01    5.6038E+01
prompt photofis          0    0.            0.                  particle decay          0    0.            0.
tabular boundary         0    0.            0.                  tabular boundary        0    0.            0.
tabular sampling      8191    4.0955E-01    2.0607E+00          elastic scatter         0    0.            0.
    total           362038    1.8099E+01    3.4231E+02              total           362038    1.8099E+01    3.4231E+02

  number of neutrons banked                 336619        average time of (shakes)              cutoffs
  neutron tracks per source particle     1.8102E+01          escape            5.4388E+00          tco   1.0000E+33
  neutron collisions per source particle 2.5088E+01          capture           4.5154E-01          eco   0.0000E+00
  total neutron collisions                  501752          capture or escape 5.4350E+00          wc1  -5.0000E-01
  net multiplication              0.0000E+00 0.0000          any termination   5.0005E+00          wc2  -2.5000E-01
```

Table 10.4: Results Compiled for Summary Cases

Case	Variation from base case	n/p
Base	N/A	17.369
1	Bertini INC and light ion transport	17.398
2	ISABEL INC for nucleons and pions	16.371
3	CEM INC for nucleons and pions	18.33
4	INCL INC for nucleons and pions; ABLA evaporation model	16.649

Net neutron production for this case is 16.649 n/p, which is 4.14% less than the base case value.

10.5.1.6 Summary

Results compiled for each case of this example are shown in Table 10.4. Runtimes between the model physics options do vary. In general, the Bertini and ISABEL models have comparable runtimes for the cases within this exercise. The CEM model was the most computationally efficient while the INCL/ABLA model was roughly double the computational time of CEM.

This example demonstrates how to calculate neutron production from a spallation target. When the quantity of interest depends only on neutrons and one starts with a proton beam, there is no need to transport any particles other than protons, neutrons, and charged pions, as neutron production by other particles is negligible compared to production by these three particle types. All particles should be included for energy deposition calculations, as discussed in §5.9.1.1. Use of the various physics model options, such as the CEM03.03. Bertini, and INCL modules, within MCNP6 is encouraged—this ability allows the user to test the sensitivity of the quantity of interest to the different physics models. If significant differences are observed, the user should evaluate which physics model is most appropriate for their particular application. For example, total neutron production from actinide targets is known to be more accurate if the multi-step pre-equilibrium model (MPM) is turned off while using Bertini INC and/or ISABEL INC.

10.6 Variance Reduction Examples

10.6.1 Pathological Concrete Shell Example

This simple, but pathological, problem illustrates how to use and interpret results from point and ring detectors that is discussed in greater detail in §2.6.10. It also shows how the statistical checks can reveal deficiencies in the tallies of an otherwise seemingly well-behaved problem.

The problem consists of a spherical shell of concrete with a 390 cm outer radius and a 360 cm inner radius [147]. A 14 MeV point isotropic neutron source is at $(0, 0, 0)$, the center of the void region. It is a neutron-only problem (`MODE n`; this is the default mode and thus the `MODE` card does not appear in the input file), with a neutron lower-energy cutoff at 12 MeV. A surface-flux and track-length tally is used in addition to point and ring detectors.

Even though this is a simple problem, it is difficult, and even inappropriate, for the `F5`-type point detector. Detectors are usually inappropriate when particles can be transported readily to the region of interest and another type of tally, such as the `F2` surface flux tally, or even better (because there is no grazing-angle approximation), the `F4` track-length tally can be used. Also, detectors do not generally work well close to or in scattering regions. This problem is especially difficult for point detectors because the largest history scores occur for neutrons that not only have several collisions near the detector point but also stay above the

12-MeV energy cutoff. These histories are extremely rare, important, and generally undersampled (leading to the long tail in the high-score region of the EHSPDF).

To demonstrate the long-tail behavior, two calculations are used with 10^4 and 10^8 histories, shown in Listings 10.47 and 10.48, respectively. Interactive-plotter command-input files [§6.2.1] to generate the figures shown in §2.6.10 are given in Listings 10.49, 10.50, and 10.51.

Listing 10.47: example_vr_conc_shell_ehspdf_10k.mcnp.inp.txt

```
 concrete shell problem with point & ring dets (TD-6-27-78: Estes and Cashwell)
1    0 -1
2    1 -2.25 1 -2
3    0       2 -3
4    0       3

1    so 360
2    so 390
3    so 390.0001

imp:n  1 1 1 0
sdef   erg=14.
m1     8016 0.65  14000 0.25  1001 0.1
cut:n  j 12.
f2:n   2
fc2    f(x) for f2:n surface flux leakage tally
f4:n   3
fc4    f(x) for f4:n volume flux leakage tally
f5:n   0 -390 0 -0.5
fc5    f(x) for f5:n point detector leakage tally
f15y:n 0 390 -0.5
fc15   f(x) for f15y:n ring detector leakage tally
print
prdmp  4j 500
nps    1e4
```

Listing 10.48: example_vr_conc_shell_ehspdf_100m.mcnp.inp.txt

```
 concrete shell problem with point & ring dets (TD-6-27-78: Estes and Cashwell)
1    0 -1
2    1 -2.25 1 -2
3    0       2 -3
4    0       3

1    so 360
2    so 390
3    so 390.0001

imp:n  1 1 1 0
sdef   erg=14.
m1     8016 0.65  14000 0.25  1001 0.1
cut:n  j 12.
f2:n   2
fc2    f(x) for f2:n surface flux leakage tally
f4:n   3
fc4    f(x) for f4:n volume flux leakage tally
f5:n   0 -390 0 -0.5
fc5    f(x) for f5:n point detector leakage tally
f15y:n 0 390 -0.5
fc15   f(x) for f15y:n ring detector leakage tally
```

```
print
prdmp  4j 5e6
nps    1e8
```

Listing 10.49: example_vr_conc_shell_tallies_10k.mcnp.comin.txt

```
run=example_vr_conc_shell_ehspdf_10k.mcnp.inp.txtr.h5 file all &
tfc m xlims 0 10000 ylims 0 1.4-7 &
title 2 "mean leakage fluxes as a function of the number of histories" &
noerr tal 4 lab "track length" cop tal 15 lab "ring detector" cop
tal 5 lab "point detector"
tfc e linlog ylims 1-3 1 &
title 2 "relative errors as a function of the number of histories" &
tal 4 lab "track length" cop tal 15 lab "ring detector" cop
tal 5 lab "point detector"
tfc v title 2 "variance of the variance as a function of the number of histories" &
tal 4 lab "track length" cop tal 15 lab "ring detector" cop
tal 5 lab "point detector"
tfc s linlin ylims 0 4 &
title 2 "slope of f(x) for large scores as a function of the number of histories" &
tal 4 lab "track length" cop tal 15 lab "ring detector" cop
tal 5 lab "point detector"
end
```

Listing 10.50: example_vr_conc_shell_tallies_100m.mcnp.comin.txt

```
run=example_vr_conc_shell_ehspdf_100m.mcnp.inp.txtr.h5 file all &
tfc m xlims 0 1+8 ylims 5.5-8 7.5-8 &
title 2 "mean leakage fluxes as a function of the number of histories" &
noerr tal 4 lab "track length" cop tal 15 lab "ring detector" cop
tal 5 lab "point detector"
tfc e linlog ylims 1-4 0.2 &
title 2 "relative errors as a function of the number of histories" &
tal 4 lab "track length" cop tal 15 lab "ring detector" cop
tal 5 lab "point detector"
tfc v ylims 1-4 1 &
title 2 "variance of the variance as a function of the number of histories" &
tal 4 lab "track length" cop tal 15 lab "ring detector" cop
tal 5 lab "point detector"
tfc s linlin ylims 0 10.1 &
title 2 "slope of f(x) for large scores as a function of the number of histories" &
tal 4 lab "track length" cop tal 15 lab "ring detector" cop
tal 5 lab "point detector"
end
```

Listing 10.51: example_vr_conc_shell_ehspdf.mcnp.comin.txt

```
run=example_vr_conc_shell_ehspdf_10k.mcnp.inp.txtr.h5 file all tfc p xlims 1-8 .2 ylims 1-7 1+9 &
title 2 "empirical f(x)'s for concrete shell leakage flux tallies for 10k histories" &
tal 4 fac x 5.23-3 0 fac y 191.135 0 lab "track length" cop
tal 15 lab "ring detector" cop tal 5 lab "point detector"
run=example_vr_conc_shell_ehspdf_100m.mcnp.inp.txtr.h5 &
title 2 "empirical f(x)'s for concrete shell leakage flux tallies for 100m histories" &
tfc p tal 4 fac x 5.23-3 0 fac y 191.135 0 lab "track length" cop
tal 15 lab "ring detector" cop tal 5 lab "point detector"
end
```

Part IV

Appendices

Appendix A

Mesh-Based WWINP, WWOUT, and WWONE File Format

The mesh-based weight-window input file WWINP and the mesh-based weight-window output files WWOUT and WWONE are ASCII files with a common format. The files consist of three blocks:

1. Block 1 contains the header information, energy and time group numbers, and basic mesh information.
2. Block 2 contains the mesh geometry.
3. Block 3 contains the energy and time group boundaries and lower weight-window bounds.

The three-dimensional array of fine mesh cells is stored by assigning an index number to each cell. The three dimensions are (x, y, z) for rectangular meshes, (r, z, θ) for cylindrical meshes, and (r, φ, θ) for spherical meshes. These may be indexed as $[i, j, k]$ with a total of I, J, K meshes in each coordinate direction. The assignment of mesh cells is illustrated in Fig. A.1 for an (x, y, z) mesh. The cell index number is related to the fine mesh number in each coordinate direction through the formula

$$\text{cell index number} = i + (j-1) \cdot I + (k-1) \cdot I \cdot J. \tag{A.1}$$

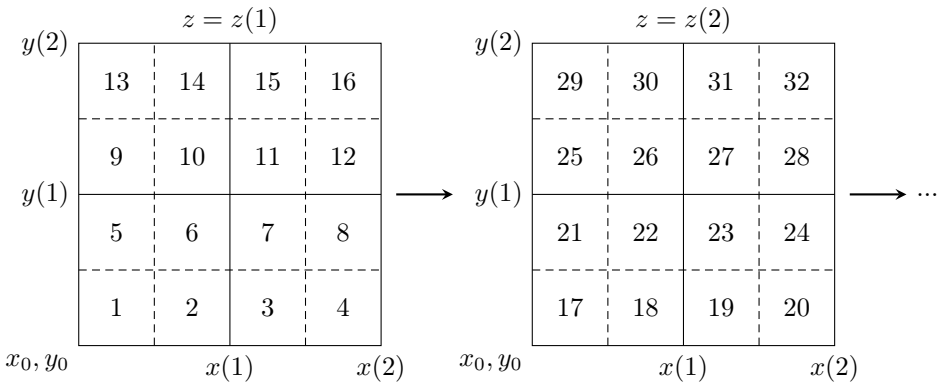


Figure A.1: Superimposed mesh cell indexing.

Table A.1 presents the file format using generic variables. Table A.2 describes the variables and gives the equivalent variables from the WWINP, WWOUT, and WWONE files. A description of the variables follows.

Table A.1: Format of the Mesh-Based WWINP, WWOUT and WWONE Files

Block	Format	Variable List
1	`4i10, 20x, a19`	`if iv ni nr probid`
1	`7i10`	`nt(1) ... nt(ni) [if iv=2]`
1	`7i10`	`ne(1) ... ne(ni)`
1	`6g13.5`	`nfx nfy nfz x0 y0 z0`
1	`6g13.5`	`ncx ncy ncz nwg [if nr=10]`
1	`6g13.5`	`ncx ncy ncz x1 y1 z1 [if nr=16]`
1	`6g13.5`	`x2 y2 z2 nwg [if nr=16]`
2	`6g13.5`	`x0 (qx(i), px(i), sx(i), i=1,ncx)`
2	`6g13.5`	`y0 (qy(i), py(i), sy(i), i=1,ncy)`
2	`6g13.5`	`z0 (qz(i), pz(i), sz(i), i=1,ncz)`
3	`6g13.5`	`t(i,1) ... t(i,nt(i)) [if nt(i)>1]`
3	`6g13.5`	`e(i,1) ... e(i,ne(i))`
3	`6g13.5`	`(((w(i,j,k,l,1) j=1,nft), k=1,ne(i)), l=1,nt(i))`

Table A.2: Correspondence of Variable Names

WWINP	WWOUT / WWONE	DESCRIPTION
ip	*ip*	Particle type
ic	*ic*	Mesh cell index
ie	*ie*	Energy index
it	*it*	Time index
ia	*ia*	Angle index (multigroup calculations)
im	*im*	Multitasking index
NWGM	NWGMA	Length of WGM/WGMA
NWWM	NWWMA	Total number of fine meshes
MWWTF(*ip*)	MWWTG(*ip*)	Time bins
NWW(*ip*)	NGWW(*ip*)	Energy bins
WWM(26)	WWMA(26)	Geometry description
WGM(*i*)	WGMA(*i*)	Geometry boundaries, fine meshes
WWE1(*ip*,*ie*)	WWGE(*ip*,*ie*)	Energy bounds
WWT1(*ip*,*it*)	WWGT(*ip*,*it*)	Time bounds
WWF(*ip*,*ic*,*ie*,*it*,*ia*)	WWFA(*ip*,*ic*,*ie*,*it*,*im*)	Weight-window lower bounds

`if`	File type. Only 1 is supported. Unused. (MCNP name: `if`)
`iv`	Time-dependent windows flag (1 / 2 = no / yes) (MCNP name: `iv`)
`ni`	Number of particle types (MCNP name: `NI`)
`nr`	= 10 / 16 / 16 for rectangular / cylindrical / spherical (MCNP name: `NR`) = number of words to describe mesh
`probid`	Problem identification description (MCNP name: `probid`)
`i`	Particle type (MCNP name: `KP`)
`nt(i)`	Number of time bins for particle type `i` (MCNP name: `NWW(KP)`)
`ne(i)`	Number of energy bins for particle type `i` (MCNP name: `NGWW(KP)`)
`nfx,nfy,nfz`	Total number of fine (x, y, z), (r, z, θ), or (r, φ, θ) mesh bins (MCNP name: `WWM(1:3)`)
`x0,y0,z0`	Corner of (x, y, z) Cartesian geometry, bottom center of (r, z, θ) cylindrical geometry, or center of (r, φ, θ) spherical geometry (MCNP name: `WWM(4:6)`)
`ncx,ncy,ncz`	Number of coarse (x, y, z), (r, z, θ), or (r, φ, θ) mesh bins (MCNP name: `WWM(7:9)`)
`x1,y1,z1`	Vector from (x_0, y_0, z_0) to (x_1, y_1, z_1) defines (r, z, θ) cylinder or (r, φ, θ) polar axis (MCNP name: `WWN(10:12)`)
`x2,y2,z2`	Vector from (x_0, y_0, z_0) to (x_2, y_2, z_2) defines (r, z, θ) cylinder or (r, φ, θ) azimuthal axis (MCNP name: `WWN(13:15)`)
`nwg`	Geometry type 1, 2, 3 = (x, y, z), (r, z, θ), (r, φ, θ) (MCNP name: `WWM(NR)`)
`px(i),py(i),pz(i)`	Coarse mesh coordinates for (x, y, z), (r, z, θ), or (r, φ, θ) (MCNP name: `WGM(k)`)
`qx(i),qy(i),qz(i)`	Fine mesh ratio (presently = 1 always) in each coarse mesh for (x, y, z), (r, z, θ), or (r, φ, θ) (MCNP name: `WGM(k)`)
`sx(i),sy(i),sz(i)`	Number of fine meshes in each coarse mesh for (x, y, z), (r, z, θ), or (r, φ, θ) (MCNP name: `WGM(k)`)
`t(i,j)`	Upper time bounds for particle `i`, bin `j` (given only if `nt(i)` > 1) (MCNP name: `WWT1(KP,j)`)
`e(i,j)`	Upper energy bounds for particle `i`, bin `j` (MCNP name: `WWE1(KP,j)`)
`nft`	Total number of fine meshes (`nfx` $\times$ `nfy` $\times$ `nfz`) (MCNP name: `NWWM`)
`w(i,j,k,l,1)`	Weight-window lower bounds. These are written in blocks of `j` $= 1 :$ `NWWMA` geometry meshes for each energy `k` $= 1,$ `NGWW(KP)` and for each time `l` $= 1,$ `MWWTG(KP)` (MCNP name: `WWF(KP,j,k,l,1)`)

A.1 Example Mesh Description and Files

Input file mesh description:

```
mesh geom=rzt ref= -4.2419 4.2419 -2
     origin 0 0 -9.0001
     imesh  3.02  6.0001
     iints  3  5
     jmesh 8.008 14.002
     jints  4  3
     kmesh .25 .50 .75  1
     kints  2   1   2   3
```

Resultant WWINP, WWOUT and WWONE file:

```
         1         1         1        16
         1
   6.0000       7.0000       8.0000       0.0000       0.0000      -9.0001
   2.0000       2.0000       4.0000       0.0000       0.0000       5.0001
   6.0001       0.0000      -9.0001       2.0000
   0.0000       3.0000       3.0200       1.0000       5.0000       6.0001
   1.0000
   0.0000       4.0000       8.0080       1.0000       3.0000       14.002
   1.0000
   0.0000       2.0000      0.25000       1.0000       1.0000      0.50000
   1.0000       2.0000      0.75000       1.0000       3.0000       1.0000
   1.0000
   100.00
   0.0000       0.0000       1.1924      0.48566      0.60746       1.0653
  0.10454       0.9993      0.11065      0.16738      0.37556      0.94980
...
```

Appendix B

XSDIR Data Directory File

A cross-section directory file, commonly referred to as the **xsdir** file, is used to locate and read the ACE-formatted data files. The **xsdir** file is a sequentially formatted ASCII file containing free-field entries delimited by blanks. The default **xsdir** file provided with the MCNP code, version 6.3, is named **xsdir_mcnp6.3**. MCNP6 uses two *types* and fifteen *classes* of data. These data are kept in individual *tables* that are often organized into libraries. These terms and tables are described in this appendix.

MCNP6 reads fifteen classes of data from two types of data tables. The two types of data tables are the following:

Type 1 standard formatted tables (sequential, up to 128 characters per record). These portable libraries are used to transmit data from one installation to another. They are bulky and slower to read. Often installations generate Type 2 tables from Type 1 tables using the **makxsf** code [344]. See §E.5 for more information on the the **makxsf** code.

Type 2 standard unformatted tables (direct-access, binary) locally generated from Type 1 tables. They are generally not portable between different systems. Type 2 tables are used mostly because they are more compact and faster to read than Type 1 tables.

Data tables exist for fifteen classes of data. These classes are identified by the last letter of the table identifier in Table B.1.

A user should think of a data table as an entity that contains evaluation-dependent information about one of the fifteen classes of data for a specific target isotope, isomer, element, or material. For how the data are used in the MCNP code, a user does not need to know whether a particular table is Type 1 or Type 2. For a given table identifier [§1.2.3], the data contained on Type 1 and Type 2 tables are identical. Problems run with one data type will track problems run with the same data in another format type.

When we refer to data libraries, we are talking about a series of data tables concatenated into one file. All tables on a single library must be of the same type but not necessarily of the same class. There is no reason, other than convenience, for having data libraries; the MCNP code could read exclusively from individual data tables not in libraries. The MCNP code determines where to find data tables for each target identifier in a problem based on information contained in a version-dependent **xsdir** file, following the rules in [§2.3.1].

The **xsdir** file has three sections. In the first section, the first line is an optional entry of the form `DATAPATH = datapath` where the word `DATAPATH` (case insensitive) must start in columns 1–5. The "=" sign is optional. The directory where the data libraries are stored is named *datapath*. The **xsdir** file can be renamed on the MCNP execution line [Table 3.4]. The search hierarchy to find **xsdir** and/or the data libraries is the following:

1. `xsdir` = *filename* on the MCNP6 execution line, where *filename* is the name of a cross-section directory file,

Table B.1: MCNP6 Data Classes

	Class description	Physics identifier
1	S(α,β) data tables	t
2	Continuous-energy neutron data libraries	c
3	Discrete-energy neutron data libraries	d
4	Coupled neutron-photon data multigroup library—neutron	m
5	Coupled neutron-photon data multigroup library—photon	g
6	Photoatomic data libraries	p
7	Photonuclear data libraries	u
8	Dosimetry data libraries	y
9	Electron data libraries	e
10	Proton data libraries	h
11	Photoatomic data libraries with atomic relaxation data	p
12	Deuteron data libraries	o
13	Triton data libraries	r
14	Helion data libraries	s
15	Alpha data libraries	a

2. `DATAPATH=`*`datapath`* in the MCNP input file message block,
3. the current working directory,
4. the `DATAPATH` entry on the first line of the **`xsdir`** file,
5. the system environment variable `DATAPATH`, or
6. the individual data table in the **`xsdir`** file (see below under Access Route).

The second section of the **`xsdir`** file is the atomic weight ratios. This section starts with the words "ATOMIC WEIGHT RATIOS" (case insensitive) beginning in columns 1–5. The following lines are free-format pairs of target identifiers and AWRs, where the target identifier is any form in [§1.2.2] and AWR is the atomic weight ratio. These atomic weight ratios are used for converting from weight fractions to atom fractions and for getting the average Z in computing electron stopping powers. If the atomic weight ratio is missing for any nuclide requested on an `Mm` card, it must be provided on the `AWTAB` card.

The third section of the **`xsdir`** file is the listing of available data tables. This section starts with the word "DIRECTORY" (case insensitive) beginning in columns 1–5. The lines following consist of the seven- to eleven-entry description of each table. The table identifier of each table must be the first entry. If a table requires more than one line, the continuation is indicated by a "+" at the end of the line. A zero indicates the entry is not applicable. Unneeded entries at the end of the line can be omitted.

The directory file has seven to eleven entries for each table. They are the following:

1. Name of the Table, character
2. Atomic Weight Ratio, real
3. File Name, character
4. Access Route, character
5. File Type, integer
6. Address, integer

7. Table Length, integer
8. Record Length, integer
9. Number of Entries per Record, integer
10. Temperature, real
11. Probability Table Flag, character

which are

1. Name of the Table, in the form of §1.2.3. The target identifier cannot be "0", "model", or "none" and may be up to 32 characters in length.
2. Atomic Weight Ratio. This is the atomic mass divided by the mass of a neutron. The atomic weight ratio here is used only for neutron kinematics and should be the same as it appears in the cross-section table so that threshold reactions are correct. It is the quantity A used in all the neutron interaction equations of Section 2 of the **`xsdir`** file. This entry is used only for neutron tables.
3. File Name. The file name is the name of the library that contains the table. The file name can include a directory path. It cannot include spaces.
4. Access Route. The access route is a string that tells how to access the file if it is not already accessible, such as a directory path. If there is no access route, this entry is zero. It cannot include spaces.
5. File Type. Either 1 for Type 1 files or 2 for Type 2.
6. Address. For Type 1 files, the address is the line number in the file where the table starts. For Type 2 files, it is the record number of the first record of the table.
7. Table Length. A data table consists of two blocks of information. The first block is a collection of pointers, counters, and character information. The second block is a solid sequence of numbers. For Type 1 and Type 2 tables, the table length is the length (total number of words) of the second block.
8. Record Length. This entry is unused for Type 1 files and therefore is zero. For Type 2 direct access files, it is a compiler-dependent attribute.
9. Number of Entries per Record. This is unused for Type 1 files and therefore is zero. For Type 2 files it is the number of entries per record. Usually this entry is set to 512.
10. Temperature. This is the temperature in MeV at which a neutron table is processed. This entry is used only for neutron data.
11. Probability Table Flag. The character word "ptable" indicates a continuous-energy neutron nuclide has unresolved resonance range probability tables.

Appendix C

Transportable Heavy Ions

This appendix lists those heavy ions, by target identifier [§1.2.2] in ZAID format, that can be transported.

$Z = 2$, Helium

2005, 2006, 2007, 2008

$Z = 3$, Lithium

3005, 3006, 3007, 3008, 3009, 3010, 3011

$Z = 4$, Beryllium

4006, 4007, 4008, 4009, 4010, 4011, 4012, 4013, 4014

$Z = 5$, Boron

5008, 5009, 5010, 5011, 5012, 5013, 5014, 5015, 5016, 5017

$Z = 6$, Carbon

6008, 6009, 6010, 6011, 6012, 6013, 6014, 6015, 6016, 6017, 6018, 6019, 6020

$Z = 7$, Nitrogen

7011, 7012, 7013, 7014, 7015, 7016, 7017, 7018, 7019, 7020, 7021, 7022, 7023

$Z = 8$, Oxygen

8013, 8014, 8015, 8016, 8017, 8018, 8019, 8020, 8021, 8022, 8023, 8024

$Z = 9$, Fluorine

9015, 9016, 9017, 9018, 9019, 9020, 9021, 9022, 9023, 9024, 9025, 9026, 9027

$Z = 10$, Neon

10017, 10018, 10019, 10020, 10021, 10022, 10023, 10024, 10025, 10026, 10027, 10028

$Z = 11$, Sodium

11019, 11020, 11021, 11022, 11023, 11024, 11025, 11026, 11027, 11028, 11029, 11030, 11031, 11032, 11033, 11034, 11035

$Z = 12$, Magnesium

12020, 12021, 12022, 12023, 12024, 12025, 12026, 12027, 12028, 12029, 12030, 12031, 12032, 12033, 12034

$Z = 13$, Aluminum

13022, 13023, 13024, 13025, 13026, 13027, 13028, 13029, 13030, 13031, 13032, 13033, 13034, 13035

$Z = 14$, Silicon

14024, 14025, 14026, 14027, 14028, 14029, 14030, 14031, 14032, 14033, 14034, 14035, 14036, 14037, 14038, 14039

$Z = 15$, Phosphorus

15026, 15027, 15028, 15029, 15030, 15031, 15032, 15033, 15034, 15035, 15036, 15037, 15038, 15039, 15040, 15041, 15042

$Z = 16$, Sulfur

16029, 16030, 16031, 16032, 16033, 16034, 16035, 16036, 16037, 16038, 16039, 16040, 16041, 16042, 16043, 16044

$Z = 17$, Chlorine

17031, 17032, 17033, 17034, 17035, 17036, 17037, 17038, 17039, 17040, 17041, 17042, 17043, 17044, 17045

$Z = 18$**, Argon**

18032, 18033, 18034, 18035, 18036, 18037, 18038, 18039, 18040, 18041, 18042, 18043, 18044, 18045, 18046

$Z = 19$**, Potassium**

19035, 19036, 19037, 19038, 19039, 19040, 19041, 19042, 19043, 19044, 19045, 19046, 19047, 19048, 19049, 19050, 19051

$Z = 20$**, Calcium**

20036, 20037, 20038, 20039, 20040, 20041, 20042, 20043, 20044, 20045, 20046, 20047, 20048, 20049, 20050, 20051

$Z = 21$**, Scandium**

21040, 21041, 21042, 21043, 21044, 21045, 21046, 21047, 21048, 21049, 21050, 21051

$Z = 22$**, Titanium**

22041, 22042, 22043, 22044, 22045, 22046, 22047, 22048, 22049, 22050, 22051, 22052, 22053, 22054

$Z = 23$**, Vanadium**

23044, 23045, 23046, 23047, 23048, 23049, 23050, 23051, 23052, 23053, 23054, 23055, 23056

$Z = 24$**, Chromium**

24045, 24046, 24047, 24048, 24049, 24050, 24051, 24052, 24053, 24054, 24055, 24056, 24057, 24058, 24059

$Z = 25$**, Manganese**

25049, 25050, 25051, 25052, 25053, 25054, 25055, 25056, 25057, 25058, 25059, 25060, 25061, 25062

$Z = 26$**, Iron**

26049, 26050, 26051, 26052, 26053, 26054, 26055, 26056, 26057, 26058, 26059, 26060, 26061, 26062, 26063, 26064

$Z = 27$**, Cobalt**

27053, 27054, 27055, 27056, 27057, 27058, 27059, 27060, 27061, 27062, 27063, 27064

$Z = 28$**, Nickel**

28053, 28054, 28055, 28056, 28057, 28058, 28059, 28060, 28061, 28062, 28063, 28064, 28065, 28066, 28067, 28068

$Z = 29$**, Copper**

29057, 29058, 29059, 29060, 29061, 29062, 29063, 29064, 29065, 29066, 29067, 29068, 29069, 29070, 29071, 29072, 29073

$Z = 30$**, Zinc**

30057, 30058, 30059, 30060, 30061, 30062, 30063, 30064, 30065, 30066, 30067, 30068, 30069, 30070, 30071, 30072, 30073, 30074, 30075, 30076, 30077, 30078

$Z = 31$**, Gallium**

31062, 31063, 31064, 31065, 311066, 31067, 31068, 31069, 31070, 31071, 31072, 31073, 31074, 31075, 31076, 31077, 31078, 31079, 31080, 31081, 31082, 31083

$Z = 32$**, Germanium**

32061, 32062, 32063, 32064, 32065, 32066, 32067, 32068, 32069, 32070, 32071, 32072, 32073, 32074, 32075, 32076, 32077, 32078, 32079, 32080, 32081, 32082, 32083, 32084

$Z = 33$**, Arsenic**

33066, 33067, 33068, 33069, 33070, 33071, 33072, 33073, 33074, 33075, 33076, 33077, 33078, 33079, 33080, 33081, 33082, 33083, 33084, 33085, 33086, 33087

$Z = 34$**, Selenium**

34068, 34069, 34070, 34071, 34072, 34073, 34074, 34075, 34076, 34077, 34078, 34079, 34080, 34081, 34082, 34083, 34084, 34085, 34086, 34087, 34088, 34089, 34090, 34091

$Z = 35$, **Bromine**

35070, 35071, 35072, 35073, 35074, 35075, 35076, 35077, 35078, 35079, 35080, 35081, 35082, 35083, 35084, 35085, 35086, 35087, 35088, 35089, 35090, 35091, 35092

$Z = 36$, **Krypton**

36071, 36072, 36073, 36074, 36075, 36076, 36077, 36078, 36079, 36080, 36081, 36082, 36083, 36084, 36085, 36086, 36087, 36088, 36089, 36090, 36091, 36092, 36093, 36094, 36095, 36096, 36097

$Z = 37$, **Rubidium**

37074, 37075, 37076, 37077, 37078, 37079, 37080, 37081, 37082, 37083, 37084, 37085, 37086, 37087, 37088, 37089, 37090, 37091, 37092, 37093, 37094, 37095, 37096, 37097, 37098, 37099, 37100

$Z = 38$, **Strontium**

38077, 38078, 38079, 38080, 38081, 38082, 38083, 38084, 38085, 38086, 38087, 38088, 38089, 38090, 38091, 38092, 38093, 38094, 38095, 38096, 38097, 38098, 38099, 38100

$Z = 39$, **Yttrium**

39080, 39081, 39082, 39083, 39084, 39085, 39086, 39087, 39088, 39089, 39090, 39091, 39092, 39093, 39094, 39095, 39096, 39097, 39098, 39099, 39100, 39101, 39102

$Z = 40$, **Zirconium**

40081, 40082, 40083, 40084, 40085, 40086, 40087, 40088, 40089, 40090, 40091, 40092, 40093, 40094, 40095, 40096, 40097, 40098, 40099, 40100, 40101, 40102

$Z = 41$, **Niobium**

41084, 41085, 41086, 41087, 41088, 41089, 41090, 41091, 41092, 41093, 41094, 41095, 41096, 41097, 41098, 41099, 41100, 41101, 41102, 41103, 41104, 41105, 41106

$Z = 42$, **Molybdenum**

42087, 42088, 42089, 42090, 42091, 42092, 42093, 42094, 42095, 42096, 42097, 42098, 42099, 42100, 42101, 42102, 42103, 42104, 42105, 42106, 42107, 42108

$Z = 43$, **Technetium**

43090, 43091, 43092, 43093, 43094, 43095, 43096, 43097, 43098, 43099, 43100, 43101, 43102, 43103, 43104, 43105, 43106, 43107, 43108, 43109, 43110

$Z = 44$, **Ruthenium**

44092, 44093, 44094, 44095, 44096, 44097, 44098, 44099, 44100, 44101, 44102, 44103, 44104, 44105, 44106, 44107, 44108, 44109, 44110, 44111, 44112, 44113

$Z = 45$, **Rhodium**

45094, 45095, 45096, 45097, 45098, 45099, 45100, 45101, 45102, 45103, 45104, 45105, 45106, 45107, 45108, 45109, 45110, 45111, 45112, 45113, 45114

$Z = 46$, **Palladium**

46096, 46097, 46098, 46099, 46100, 46101, 46102, 46103, 46104, 46105, 46106, 46107, 46108, 46109, 46110, 46111, 46112, 46113, 46114, 46115, 46116, 46117, 46118

$Z = 47$, **Silver**

47096, 47097, 47098, 47099, 47100, 47101, 47102, 47103, 47104, 47105, 47106, 47107, 47108, 47109, 47110, 47111, 47112, 47113, 47114, 47115, 47116, 47117, 47118, 47119, 47120, 47121, 47122, 47123

$Z = 48$, **Cadmium**

48097, 48098, 48099, 48100, 48101, 48102, 48103, 48104, 48105, 48106, 48107, 48108, 48109, 48110, 48111, 48112, 48113, 48114, 48115, 48116, 48117, 48118, 48119, 48120, 48121, 48122, 48123, 48124, 48125, 48126

$Z = 49$, **Indium**

49100, 49101, 49102, 49103, 49104, 49105, 49106, 49107, 49108, 49109, 49110, 49111, 49112, 49113, 49114, 49115, 49116, 49117, 49118, 49119, 49120, 49121, 49122, 49123, 49124, 49125, 49126, 49127, 49128, 49129, 49130, 49131, 49132

$Z = 50$, **Tin**

50103, 50104, 50105, 50106, 50107, 50108, 50109, 50110, 50111, 50112, 50113, 50114, 50115, 50116, 50117, 50118, 50119, 50120, 50121, 50122, 50123, 50124, 50125, 50126, 50127, 50128, 50129, 50130, 50131, 50132, 50133, 50134

$Z = 51$, **Antimony**

51108, 51109, 51110, 51111, 51112, 51113, 51114, 51115, 51116, 51117, 51118, 51119, 51120, 51121, 51122, 51123, 51124, 51125, 51126, 51127, 51128, 51129, 51130, 51131, 51132, 51133, 51134, 51135, 51136

$Z = 52$, **Tellurium**

52106, 52107, 52108, 52109, 52110, 52111, 52112, 52113, 52114, 52115, 52116, 52117, 52118, 52119, 52120, 52121, 52122, 52123, 52124, 52125, 52126, 52127, 52128, 52129, 52130, 52131, 52132, 52133, 52134, 52135, 52136, 52137, 52138

$Z = 53$, **Iodine**

53110, 53111, 53112, 53113, 53114, 53115, 53116, 53117, 53118, 53119, 53120, 53121, 53122, 53123, 53124, 53125, 53126, 53127, 53128, 53129, 53130, 53131, 53132, 53133, 53134, 53135, 53136, 53137, 53138, 53139, 53140, 53141, 53142

$Z = 54$, **Xenon**

54110, 54111, 54112, 54113, 54114, 54115, 54116, 54117, 54118, 54119, 54120, 54121, 54122, 54123, 54124, 54125, 54126, 54127, 54128, 54129, 54130, 54131, 54132, 54133, 54134, 54135, 54136, 54137, 54138, 54139, 54140, 54141, 54142, 54143, 54144, 54145

$Z = 55$, **Cesium**

55114, 55115, 55116, 55117, 55118, 55119, 55120, 55121, 55122, 55123, 55124, 55125, 55126, 55127, 55128, 55129, 55130, 55131, 55132, 55133, 55134, 55135, 55136, 55137, 55138, 55139, 55140, 55141, 55142, 55143, 55144, 55145, 55146, 55147, 55148

$Z = 56$, **Barium**

56117, 56118, 56119, 56120, 56121, 56122, 56123, 56124, 56125, 56126, 56127, 56128, 56129, 56130, 56131, 56132, 56133, 56134, 56135, 56136, 56137, 56138, 56139, 56140, 56141, 56142, 56143, 56144, 56145, 56146, 56147, 56148

$Z = 57$, **Lanthanum**

57123, 57124, 57125, 57126, 57127, 57128, 57129, 57130, 57131, 57132, 57133, 57134, 57135, 57136, 57137, 57138, 57139, 57140, 57141, 57142, 57143, 57144, 57145, 57146, 57147, 57148, 57149

$Z = 58$, Cerium

58124, 58125, 58126, 58127, 58128, 58129, 58130, 58131, 58132, 58133, 58134, 58135, 58136, 58137, 58138, 58139, 58140, 58141, 58142, 58143, 58144, 58145, 58146, 58147, 58148, 58149, 58150, 58151

$Z = 59$, Praseodymium

59129, 59130, 59131, 59132, 59133, 59134, 59135, 59136, 59137, 59138, 59139, 59140, 59141, 59142, 59143, 59144, 59145, 59146, 59147, 59148, 59149, 59150, 59151, 59152

$Z = 60$, Neodymium

60129, 60130, 60131, 60132, 60133, 60134, 60135, 60136, 60137, 60138, 60139, 60140, 60141, 60142, 60143, 60144, 60145, 60146, 60147, 60148, 60149, 60150, 60151, 60152, 60153, 60154

$Z = 61$, Promethium

61132, 61133, 61134, 61135, 61136, 61137, 61138, 61139, 61140, 61141, 61142, 61143, 61144, 61145, 61146, 61147, 61148, 61149, 61150, 61151, 61152, 61153, 61154, 61155

$Z = 62$, Samarium

62133, 62134, 62135, 62136, 62137, 62138, 62139, 62140, 62141, 62142, 62143, 62144, 62145, 62146, 62147, 62148, 62149, 62150, 62151, 62152, 62153, 62154, 62155, 62156, 62157, 62158

$Z = 63$, Europium

63138, 63139, 63140, 63141, 63142, 63143, 63144, 63145, 63146, 63147, 63148, 63149, 63150, 63151, 63152, 63153, 63154, 63155, 63156, 63157, 63158, 63159, 63160

$Z = 64$, Gadolinium

64142, 64143, 64144, 64145, 64146, 64147, 64148, 64149, 64150, 64151, 64152, 64153, 64154, 64155, 64156, 64157, 64158, 64159, 64160, 64161, 64162, 64163

$Z = 65$, Terbium

65144, 65145, 65146, 65147, 65148, 65149, 65150, 65151, 65152, 65153, 65154, 65155, 65156, 65157, 65158, 65159, 65160, 65161, 65162, 65163, 65164, 65165

$Z = 66$, Dysprosium

66145, 66146, 66147, 66148, 66149, 66150, 66151, 66152, 66153, 66154, 66155, 66156, 66157, 66158, 66159, 66160, 66161, 66162, 66163, 66164, 66165, 66166, 66167, 66168

$Z = 67$, Holmium

67147, 67148, 67149, 67150, 67151, 67152, 67153, 67154, 67155, 67156, 67157, 67158, 67159, 67160, 67161, 67162, 67163, 67164, 67165, 67166, 67167, 67168, 67169, 67170

$Z = 68$, Erbium

68147, 68148, 68149, 68150, 68151, 68152, 68153, 68154, 68155, 68156, 68157, 68158, 68159, 68160, 68161, 68162, 68163, 68164, 68165, 68166, 68167, 68168, 68169, 68170, 68171, 68172, 68173

$Z = 69$, Thulium

69151, 69152, 69153, 69154, 69155, 69156, 69157, 69158, 69159, 69160, 69161, 69162, 69163, 69164, 69165, 69166, 69167, 69168, 69169, 69170, 69171, 69172, 69173, 69174, 69175, 69176

$Z = 70$, Ytterbium

70153, 70154, 70155, 70156, 70157, 70158, 70159, 70160, 70161, 70162, 70163, 70164, 70165, 70166, 70167, 70168, 70169, 70170, 70171, 70172, 70173, 70174, 70175, 70176, 70177, 70178, 70179

$Z = 71$, Lutetium

71151, 71152, 71153, 71154, 71155, 71156, 71157, 71158, 71159, 71160, 71161, 71162, 71163, 71164, 71165, 71166, 71167, 71168, 71169, 71170, 71171, 71172, 71173, 71174, 71175, 71176, 71177, 71178, 71179, 71180, 71181, 71182, 71183

$Z = 72$, Hafnium

72154, 72155, 72156, 72157, 72158, 72159, 72160, 72161, 72162, 72163, 72164, 72165, 72166, 72167, 72168, 72169, 72170, 72171, 72172, 72173, 72174, 72175, 72176, 72177, 72178, 72179, 72180, 72181, 72182, 72183, 72184

$Z = 73$, Tantalum

73157, 73158, 73159, 73160, 73161, 73162, 73163, 73164, 73165, 73166, 73167, 73168, 73169, 73170, 73171, 73172, 73173, 73174, 73175, 73176, 73177, 73178, 73179, 73180, 73181, 73182, 73183, 73184, 73185, 73186

$Z = 74$, **Tungsten**

74158, 74159, 74160, 74161, 74162, 74163, 74164, 74165, 74166, 74167, 74168, 74169, 74170, 74171, 74172, 74173, 74174, 74175, 74176, 74177, 74178, 74179, 74180, 74181, 74182, 74183, 74184, 74185, 74186, 74187, 74188, 74189, 74190

$Z = 75$, **Rhenium**

75161, 75162, 75163, 75164, 75165, 75166, 75167, 75168, 75169, 75170, 75171, 75172, 75173, 75174, 75175, 75176, 75177, 75178, 75179, 75180, 75181, 75182, 75183, 75184, 75185, 75186, 75187, 75188, 75189, 75190, 75191, 75192

$Z = 76$, **Osmium**

76163, 76164, 76165, 76166, 76167, 76168, 76169, 76170, 76171, 76172, 76173, 76174, 76175, 76176, 76177, 76178, 76179, 76180, 76181, 76182, 76183, 76184, 76185, 76186, 76187, 76188, 76189, 76190, 76191, 76192, 76193, 76194, 76195, 76196

$Z = 77$, **Iridium**

77166, 77167, 77168, 77169, 77170, 77171, 77172, 77173, 77174, 77175, 77176, 77177, 77178, 77179, 77180, 77181, 77182, 77183, 77184, 77185, 77186, 77187, 77188, 77189, 77190, 77191, 77192, 77193, 77194, 77195, 77196, 77197, 77198

$Z = 78$, **Platinum**

78168, 78169, 78170, 78171, 78172, 78173, 78174, 78175, 78176, 78177, 78178, 78179, 78180, 78181, 78182, 78183, 78184, 78185, 78186, 78187, 78188, 78189, 78190, 78191, 78192, 78193, 78194, 78195, 78196, 78197, 78198, 78199, 78200, 78201

$Z = 79$, **Gold**

79175, 79176, 79177, 79178, 79179, 79180, 79181, 79182, 79183, 79184, 79185, 79186, 79187, 79188, 79189, 79190, 79191, 79192, 79193, 79194, 79195, 79196, 79197, 79198, 79199, 79200, 79201, 79202, 79203, 79204

$Z = 80$, **Mercury**

80177, 80178, 80179, 80180, 80181, 80182, 80183, 80184, 80185, 80186, 80187, 80188, 80189, 80190, 80191, 80192, 80193, 80194, 80195, 80196, 80197, 80198, 80199, 80200, 80201, 80202, 80203, 80204, 80205, 80206

$Z = 81$, Thallium

81184, 81185, 81186, 81187, 81188, 81189, 81190, 81191, 81192, 81193, 81194, 81195, 81196, 81197, 81198, 81199, 81200, 81201, 81202, 81203, 81204, 81205, 81206, 81207, 81208, 81209, 81210

$Z = 82$, Lead

82183, 82184, 82185, 82186, 82187, 82188, 82189, 82190, 82191, 82192, 82193, 82194, 82195, 82196, 82197, 82198, 82199, 82200, 82201, 82202, 82203, 82204, 82205, 82206, 82207, 82208, 82209, 82210, 82211, 82212, 82213, 82214

$Z = 83$, Bismuth

83188, 83189, 83190, 83191, 83192, 83193, 83194, 83195, 83196, 83197, 83198, 83199, 83200, 83201, 83202, 83203, 83204, 83205, 83206, 83207, 83208, 83209, 83210, 83211, 83212, 83213, 83214, 83215

$Z = 84$, Polonium

84192, 84193, 84194, 84195, 84196, 84197, 84198, 84199, 84200, 84201, 84202, 84203, 84204, 84205, 84206, 84207, 84208, 84209, 84210, 84211, 84212, 84213, 84214, 84215, 84216, 84217, 84218

$Z = 85$, Astatine

85196, 85197, 85198, 85199, 85200, 85201, 85202, 85203, 85204, 85205, 85206, 85207, 85208, 85209, 85210, 85211, 85212, 85213, 85214, 85215, 85216, 85217, 85218, 85219

$Z = 86$, Radon

86199, 86200, 86201, 86202, 86203, 86204, 86205, 86206, 86207, 86208, 86209, 86210, 86211, 86212, 86213, 86214, 86215, 86216, 86217, 86218, 86219, 86220, 86221, 86222, 86223, 86224, 86225, 86226

$Z = 87$, Francium

87201, 87202, 87203, 87204, 87205, 87206, 87207, 87208, 87209, 87210, 87211, 87212, 87213, 87214, 87215, 87216, 87217, 87218, 87219, 87220, 87221, 87222, 87223, 87224, 87225, 87226, 87227, 87228, 87229

$Z = 88$, Radium

88206, 88207, 88208, 88209, 88210, 88211, 88212, 88213, 88214, 88215, 88216, 88217, 88218, 88219, 88220, 88221, 88222, 88223, 88224, 88225, 88226, 88227, 88228, 88229, 88230

$Z = 89$, Actinium

89209, 89210, 89211, 89212, 89213, 89214, 89215, 89216, 89217, 89218, 89219, 89220, 89221, 89222, 89223, 89224, 89225, 89226, 89227, 89228, 89229, 89230, 89231, 89232

$Z = 90$, Thorium

90212, 90213, 90214, 90215, 90216, 90217, 90218, 90219, 90220, 90221, 90222, 90223, 90224, 90225, 90226, 90227, 90228, 90229, 90230, 90231, 90232, 90233, 90234, 90235, 90236

$Z = 91$, Protactinium

91215, 91216, 91217, 91218, 91219, 91220, 91221, 91222, 91223, 91224, 91225, 91226, 91227, 91228, 91229, 91230, 91231, 91232, 91233, 91234, 91235, 91236, 91237, 91238

$Z = 92$, Uranium

92222, 92223, 92224, 92225, 92226, 92227, 92228, 92229, 92230, 92231, 92232, 92233, 92234, 92235, 92236, 92237, 92238, 92239, 92240, 92241, 92242

$Z = 93$, Neptunium

93227, 93228, 93229, 93230, 93231, 93232, 93233, 93234, 93235, 93236, 93237, 93238, 93239, 93240, 93241, 93242

$Z = 94$, Plutonium

94232, 94233, 94234, 94235, 94236, 94237, 94238, 94239, 94240, 94241, 94242, 94243, 94244, 94245, 94246

$Z = 95$, Americium

95232, 95233, 95234, 95235, 95236, 95237, 95238, 95239, 95240, 95241, 95242, 95243, 95244, 95245, 95246, 95247

$Z = 96$, Curium

96238, 96239, 96240, 96241, 96242, 96243, 96244, 96245, 96246, 96247, 96248, 96249, 96250, 96251

$Z = 97$, Berkelium

97240, 97241, 97242, 97243, 97244, 97245, 97246, 97247, 97248, 97249, 97250, 97251

$Z = 98$, Californium

98239, 98240, 98241, 98242, 98243, 98244, 98245, 98246, 98247, 98248, 98249, 98250, 98251, 98252, 98253, 98254, 98255, 98256

$Z = 99$, Einsteinium

99243, 99244, 99245, 99246, 99247, 99248, 99249, 99250, 99251, 99252, 99253, 99254, 99255, 99256

$Z = 100$, Fermium

100242, 100243, 100244, 100245, 100246, 100247, 100248, 100249, 100250, 100251, 100252, 100253, 100254, 100255, 100256, 100257, 100258, 100259

Appendix D

File Formats

This appendix describes the formats of files that are produced or processed by the MCNP code.

D.1 Overview of HDF5 in the MCNP Code

Several files in the MCNP code have been replaced with HDF5 files beginning with MCNP code, version 6.3. The HDF5 format provides a standardized binary file format that allows writing and storing large numerical data collections efficiently. In HDF5, numerical data are stored in datasets, and nested groups are used to organize data via names (similar to the directory structures of a file system). This format offers the efficiency of binary data, while allowing for organization and direct access to data. The HDF5 format is described further at https://portal.hdfgroup.org/display/HDF5/Introduction+to+HDF5.

For the MCNP code, the HDF5 format is being used in place of Fortran unformatted IO binary files from previous releases. The primary motivation for this change is that individual data in an HDF5 file can be accessed directly by name, which eliminates the need to process the files in a sequential order. Another benefit of HDF5 is that it is far easier to manipulate than Fortran unformatted IO in languages other than Fortran, and there are preexisting tools to read and manipulate data. The **h5dump** and **h5view** utilities (https://www.hdfgroup.org/downloads/) provide human-readable access to the binary data contained in HDF5 files, which is useful for quick inspection of output results. The **h5py** library (https://www.h5py.org/) allows manipulation and processing of data via the Python language.

D.2 Restart File Format

This section details the file format of the current MCNP HDF5-based restart file, with default name **runtpe.h5**. At the moment, most dataset names in the restart file directly correspond to an MCNP variable. This format is subject to change, and breaking changes are expected as the MCNP source code undergoes reorganization. The current file kind string is stored inside of the `kind_file` attribute in the `config_control` group, defined as "runtape". The current file version is stored inside of the `version_file` attribute in the `config_control` group. The semantic versioning approach defines how the version number will evolve. Once the format stabilizes, backwards compatibility will be attempted, either through the presence or absence of groups (for new features) or through conversion scripts. When reading in a restart file for either restarted calculations or for plotting purposes, both the `kind_file` and `version_file` attributes are checked for consistency with what is supported by the current code version.

In the file description below, components without children are datasets, and components with children (and those that are hyperlinks) are groups of the restart file. As an example, the dataset `Loc_iso` can be found inside the `fdac` group, which is inside the `fixed` group, which is inside of the root group `/`. The complete path is `/fixed/fdac/Loc_iso`. A few datasets, such as `xss` and `ufil`, are written in multiple parts. Arrays of

structures are often written with the array name followed by some index. Components labeled "optional" are optional relative to their parent.

⚠ Caution

This section is automatically generated from the output files of the MCNP testing suite, and may not be 100% complete. If any abnormalities are observed, please notify mcnp_help@lanl.gov and provide the input file that caused the abnormality.

D.2.1 Main Layout

This is the highest level of the runtape. The majority of the content is contained in the `fixed` (data that is constant throughout a problem) and `variable` (data that changes as the simulation runs) groups.

```
/ ..... (root)
├─ config_control ..... (group)
├─ fixed ..... (group)
├─ header ..... (group)
├─ problem_info ..... (group)
├─ restart ..... (optional) (group)
├─ results ..... (optional) (group)
└─ variable ..... (group)
```

D.2.2 Configuration Control

[back to tree]
This group contains attributes used to identify the version of the file format, as well as other information on how the file was generated. Many of the HDF5 files the MCNP code generates include this group.

```
config_control ..... (group)
├─ build_date_code ..... (optional) (attribute)
├─ code_name ..... (optional) (attribute)
├─ gitinfo_code ..... (optional) (attribute)
├─ kind_file ..... (attribute)
├─ release_status_code ..... (optional) (attribute)
├─ version_code ..... (optional) (attribute)
└─ version_file ..... (attribute)
```

D.2.3 Fixed Data

[back to tree]
Data that are constant throughout the simulation and are only written to the file once.

```
fixed ..... (group)
├─ aplace ..... (dataset)
└─ dbrc ..... (optional) (group)
```

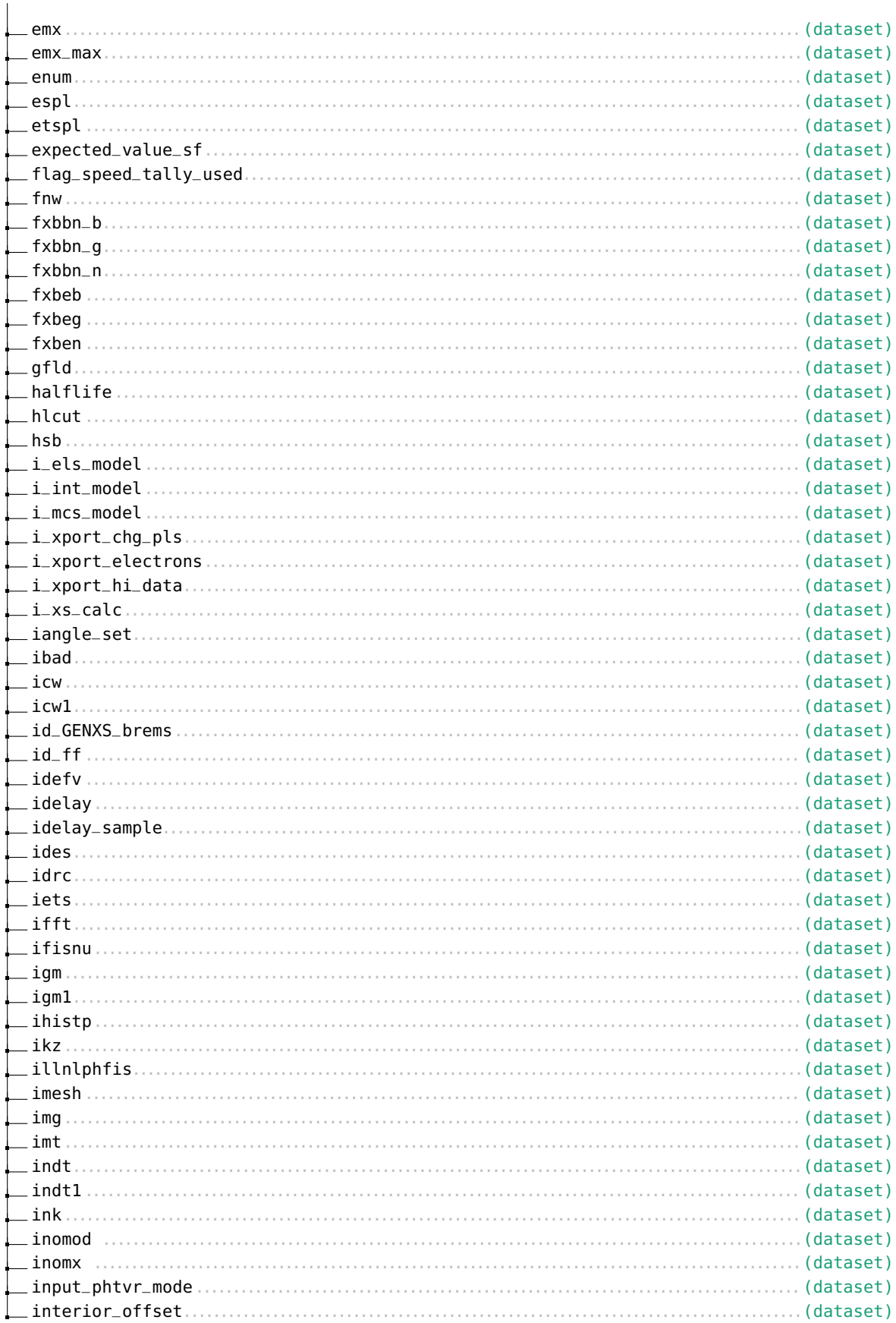
```
    ├── L ..... (dataset)
    ├── e ..... (dataset)
    ├── emax ..... (dataset)
    ├── idummy ..... (dataset)
    ├── info ..... (dataset)
    ├── ix_iex ..... (dataset)
    ├── n_iex ..... (dataset)
    ├── ne ..... (dataset)
    ├── ne_tot ..... (dataset)
    ├── nisos ..... (dataset)
    ├── sigs ..... (dataset)
    └── zaids ..... (dataset)
└── fdac ..... (group)
    ├── Loc_iso ..... (dataset)
    ├── P_com_lib ..... (dataset)
    ├── P_lib ..... (dataset)
    ├── Q1_lib ..... (dataset)
    ├── T_lib ..... (dataset)
    ├── aa9 ..... (dataset)
    ├── act ..... (dataset)
    ├── asm ..... (dataset)
    ├── asp ..... (dataset)
    ├── ave_a ..... (dataset)
    ├── ave_z ..... (dataset)
    ├── ave_zsq ..... (dataset)
    ├── awc ..... (dataset)
    ├── awc9 ..... (dataset)
    ├── awn ..... (dataset)
    ├── b_westphal ..... (dataset)
    ├── barpo ..... (optional) (group)
    ├── bvol ..... (dataset)
    ├── cmg ..... (dataset)
    ├── conb ..... (dataset)
    ├── da_cont ..... (dataset)
    ├── db_cont ..... (dataset)
    ├── delay_file ..... (dataset)
    ├── dg_cont ..... (dataset)
    ├── dn_cont ..... (dataset)
    ├── dp_cont ..... (dataset)
    ├── drs ..... (dataset)
    ├── drs_chg_pl ..... (dataset)
    ├── dxcp ..... (dataset)
    ├── eaa ..... (dataset)
    ├── ear ..... (dataset)
    ├── eba ..... (dataset)
    ├── ebd ..... (dataset)
    ├── ebl ..... (dataset)
    ├── ebt ..... (dataset)
    ├── ech ..... (dataset)
    ├── edg ..... (dataset)
    ├── eee ..... (dataset)
    ├── eek ..... (dataset)
    ├── egg ..... (dataset)
    └── elas_lib ..... (dataset)
```

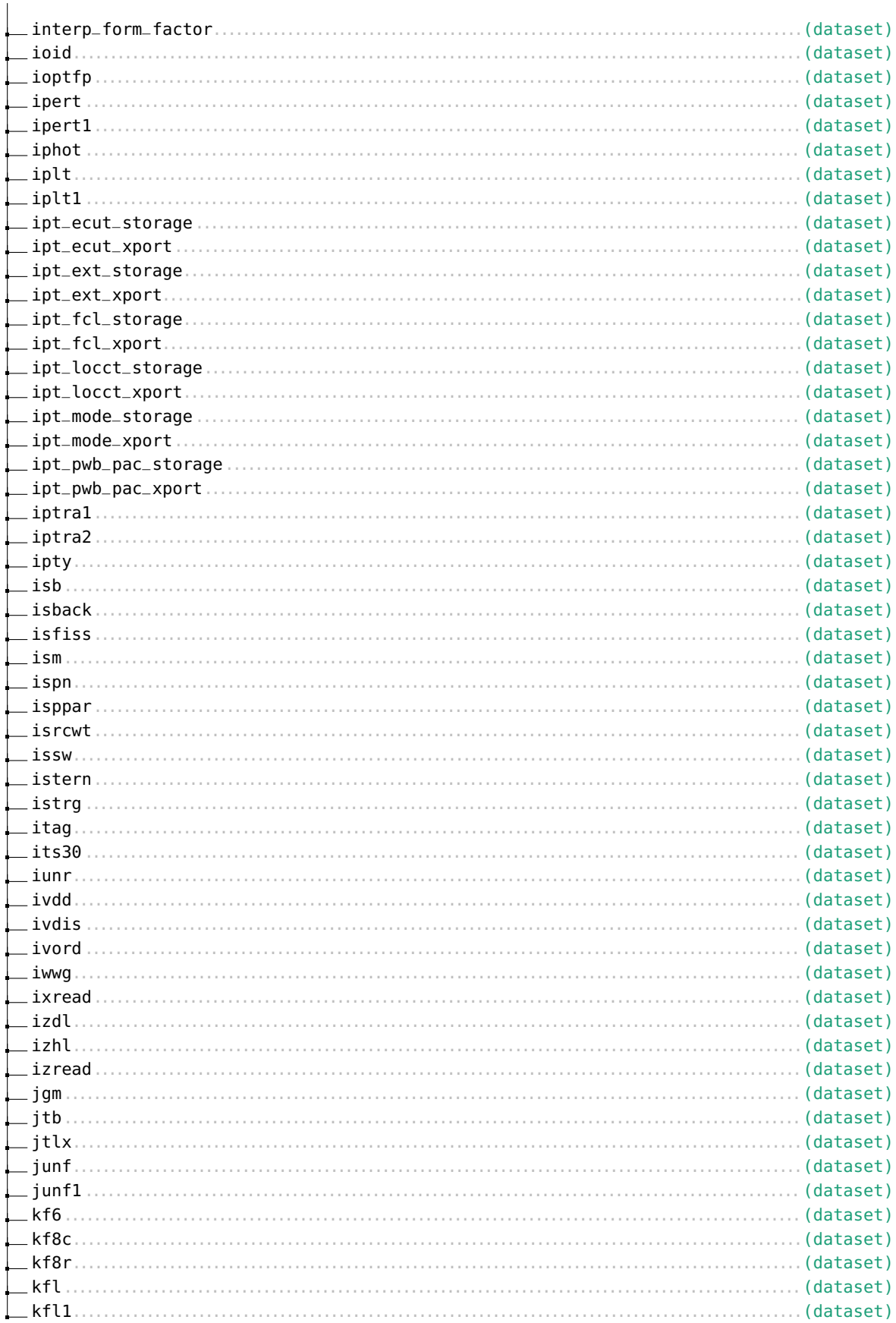
- elp (dataset)
- elxs_pS_all (dataset)
- elxs_pX_all (dataset)
- emaxt (dataset)
- emaxtm (dataset)
- embedded geometry (optional) (group)
- energies_and_params (dataset)
- esa (dataset)
- esxln (dataset)
- esxlp (dataset)
- esxmp (dataset)
- excitation_mean_MeV (dataset)
- fast_tbl_kbins (optional) (dataset)
- fast_tbl_nbins (dataset)
- fast_tbl_udel (optional) (dataset)
- fast_tbl_umax (optional) (dataset)
- fast_tbl_umin (optional) (dataset)
- fim (dataset)
- flag_laf_fill_u1 (dataset)
- flag_laf_fill_u2 (dataset)
- flag_laf_fill_u3 (dataset)
- flc (dataset)
- fmf (dataset)
- fmg (dataset)
- for (dataset)
- frc (dataset)
- fst (dataset)
- ftt (dataset)
- gauss_param_mcs (dataset)
- gmg (dataset)
- gvl (dataset)
- gwt (dataset)
- hsigg (dataset)
- ia_lib (dataset)
- iazdex (dataset)
- idna (dataset)
- idne (dataset)
- idns (dataset)
- idnt (dataset)
- iprmt (dataset)
- iss (dataset)
- ixl (dataset)
- ixs (dataset)
- iza (dataset)
- jasq (dataset)
- jasr (dataset)
- jasw (dataset)
- jmd (dataset)
- jscn (dataset)
- jsq (dataset)
- jss (dataset)
- jun (dataset)
- jvc (dataset)
- jxs (dataset)

```
├── jxs_mu_gam ..... (dataset)
├── kcp ..... (dataset)
├── kmm ..... (dataset)
├── ksc ..... (dataset)
├── ksd ..... (dataset)
├── ksm ..... (dataset)
├── kst ..... (dataset)
├── ksu ..... (dataset)
├── kxs ..... (dataset)
├── laf_bounds ..... (optional) (dataset)
├── laf_fill_size ..... (dataset)
├── laf_fill_u1 ..... (optional) (dataset)
├── laf_fill_u2 ..... (optional) (dataset)
├── laf_fill_u3 ..... (optional) (dataset)
├── lat ..... (dataset)
├── lca ..... (dataset)
├── lja ..... (dataset)
├── lme ..... (dataset)
├── lme2_of_max_z_in_mat ..... (dataset)
├── lme3_of_max_z_in_mat ..... (dataset)
├── lmn ..... (dataset)
├── lmt ..... (dataset)
├── locphc ..... (dataset)
├── lsc ..... (dataset)
├── mag_fields ..... (optional) (group)
├── mars ..... (optional) (group)
├── mat ..... (dataset)
├── matb ..... (dataset)
├── matburn ..... (dataset)
├── max_0_lib ..... (dataset)
├── mazp ..... (dataset)
├── mazu ..... (dataset)
├── mbd ..... (dataset)
├── mbi ..... (dataset)
├── mcs2 ..... (optional) (group)
├── mel ..... (dataset)
├── mfl ..... (dataset)
├── mu_shell_end ..... (dataset)
├── mu_shell_start ..... (dataset)
├── n_finite_lattices ..... (dataset)
├── nbzaid ..... (dataset)
├── ncl ..... (dataset)
├── nel ..... (dataset)
├── ngmfl ..... (dataset)
├── nhtfl ..... (dataset)
├── nhw ..... (dataset)
├── nmt ..... (dataset)
├── npq ..... (dataset)
├── nsb ..... (dataset)
├── nsfm ..... (dataset)
├── nslr ..... (dataset)
├── nucb ..... (dataset)
├── nxs ..... (dataset)
└── nxs_mu_gam ..... (dataset)
```

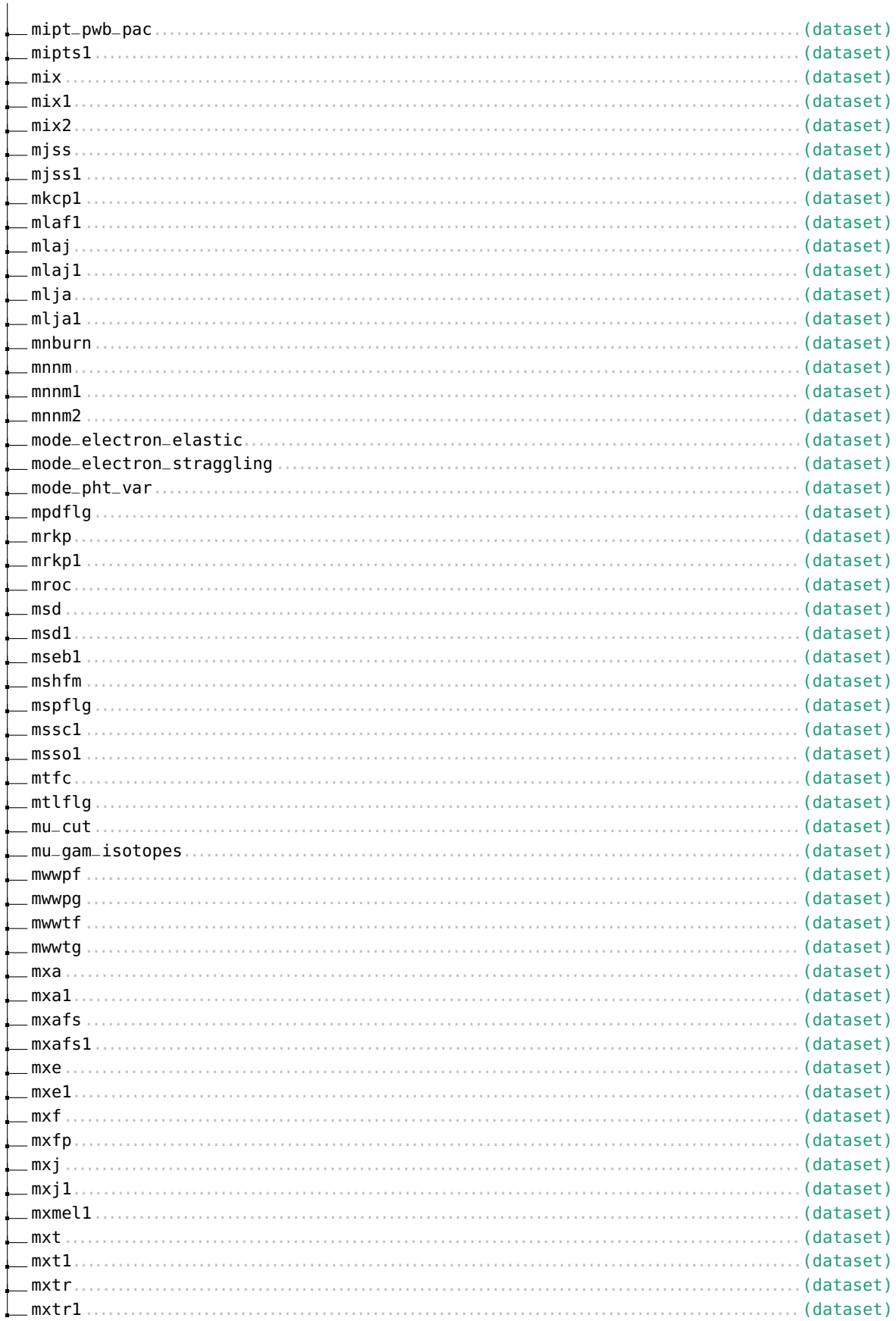
```
├── pbt ..... (dataset)
├── pcap ..... (dataset)
├── pkn ..... (dataset)
├── pmg ..... (dataset)
├── pnt ..... (dataset)
├── pru ..... (dataset)
├── ptb_data ..... (dataset)
├── ptb_iso ..... (dataset)
├── ptb_parflag ..... (dataset)
├── ptb_ptr ..... (dataset)
├── ptb_rho ..... (dataset)
├── pxr ..... (dataset)
├── qav ..... (dataset)
├── qav_chg_pl ..... (dataset)
├── qax ..... (dataset)
├── qcn ..... (dataset)
├── react_lib ..... (dataset)
├── ref_idx_coef ..... (dataset)
├── rej_lib ..... (dataset)
├── rng ..... (dataset)
├── rng_chg_pl ..... (dataset)
├── scf ..... (dataset)
├── sigg ..... (dataset)
├── sigmx ..... (dataset)
├── smg ..... (dataset)
├── spf ..... (dataset)
├── sqq ..... (dataset)
├── sso ..... (dataset)
├── swapb ..... (dataset)
├── table_max_energy ..... (dataset)
├── table_min_energy ..... (dataset)
├── table_type ..... (dataset)
├── tallies ..... (group)
├── targ_mass_lib ..... (dataset)
├── tbt ..... (dataset)
├── timeb ..... (dataset)
├── tmp ..... (dataset)
├── trf ..... (dataset)
├── tth ..... (dataset)
├── vcl ..... (dataset)
├── vec ..... (dataset)
├── wgm ..... (dataset)
├── wgma ..... (dataset)
├── ww_e_bin_upper ..... (dataset)
├── ww_low_bound ..... (dataset)
├── ww_t_bin_upper ..... (dataset)
├── wwge ..... (dataset)
├── wwgt ..... (dataset)
├── wwk ..... (dataset)
├── x_mu_gam_energy ..... (dataset)
├── x_mu_gam_prob ..... (dataset)
├── xnm ..... (dataset)
├── xs_bremsstrahlung ..... (dataset)
└── xs_max_by_isot ..... (dataset)
```

```
    ├── xs_max_by_mat ..... (dataset)
    ├── xse85 ..... (dataset)
    ├── xsiso ..... (dataset)
    ├── zaid_lib ..... (dataset)
    └── zz9 ..... (dataset)
└── fixcom ..... (group)
    ├── GENXS_brems_minimum ..... (dataset)
    ├── Photon_CEM_LAQGSM_Energy ..... (dataset)
    ├── RN_gen_input ..... (dataset)
    ├── RN_hist_input ..... (dataset)
    ├── RN_seed_input ..... (dataset)
    ├── RN_stride_input ..... (dataset)
    ├── back ..... (dataset)
    ├── bbrem ..... (dataset)
    ├── bnum ..... (dataset)
    ├── brem_pe_ratio_for_cos ..... (dataset)
    ├── brem_pe_ratio_for_erg ..... (dataset)
    ├── burn ..... (dataset)
    ├── burn_calculation ..... (dataset)
    ├── cadis_lng ..... (dataset)
    ├── cadis_nerg ..... (dataset)
    ├── cadis_npart ..... (dataset)
    ├── cadis_nsrc ..... (dataset)
    ├── calph ..... (dataset)
    ├── cdgb ..... (dataset)
    ├── cdnb ..... (dataset)
    ├── ckvnum ..... (dataset)
    ├── coincd ..... (dataset)
    ├── coincd_rel ..... (dataset)
    ├── cr_flag ..... (dataset)
    ├── csa ..... (dataset)
    ├── csda_table ..... (dataset)
    ├── cum_dist ..... (dataset)
    ├── cut_mix ..... (dataset)
    ├── dawwg_control ..... (dataset)
    ├── dbeb ..... (dataset)
    ├── ddg ..... (dataset)
    ├── ddx ..... (dataset)
    ├── dgb ..... (dataset)
    ├── dgeb ..... (dataset)
    ├── dgthresh ..... (dataset)
    ├── disable_burn_stats ..... (dataset)
    ├── disable_interupt ..... (dataset)
    ├── dnb ..... (dataset)
    ├── dneb ..... (dataset)
    ├── dray_cutoff ..... (dataset)
    ├── dxw ..... (dataset)
    ├── dxx ..... (dataset)
    ├── ecf ..... (dataset)
    ├── ecf_min_chg_pl ..... (dataset)
    ├── efac ..... (dataset)
    ├── electron_method_boundary ..... (dataset)
    ├── emcf ..... (dataset)
    └── emin_by_particle ..... (dataset)
```


emx (dataset)
emx_max (dataset)
enum (dataset)
espl (dataset)
etspl (dataset)
expected_value_sf (dataset)
flag_speed_tally_used (dataset)
fnw (dataset)
fxbbn_b (dataset)
fxbbn_g (dataset)
fxbbn_n (dataset)
fxbeb (dataset)
fxbeg (dataset)
fxben (dataset)
gfld (dataset)
halflife (dataset)
hlcut (dataset)
hsb (dataset)
i_els_model (dataset)
i_int_model (dataset)
i_mcs_model (dataset)
i_xport_chg_pls (dataset)
i_xport_electrons (dataset)
i_xport_hi_data (dataset)
i_xs_calc (dataset)
iangle_set (dataset)
ibad (dataset)
icw (dataset)
icw1 (dataset)
id_GENXS_brems (dataset)
id_ff (dataset)
idefv (dataset)
idelay (dataset)
idelay_sample (dataset)
ides (dataset)
idrc (dataset)
iets (dataset)
ifft (dataset)
ifisnu (dataset)
igm (dataset)
igm1 (dataset)
ihistp (dataset)
ikz (dataset)
illnlphfis (dataset)
imesh (dataset)
img (dataset)
imt (dataset)
indt (dataset)
indt1 (dataset)
ink (dataset)
inomod (dataset)
inomx (dataset)
input_phtvr_mode (dataset)
interior_offset (dataset)


interp_form_factor (dataset)
ioid (dataset)
ioptfp (dataset)
ipert (dataset)
ipert1 (dataset)
iphot (dataset)
iplt (dataset)
iplt1 (dataset)
ipt_ecut_storage (dataset)
ipt_ecut_xport (dataset)
ipt_ext_storage (dataset)
ipt_ext_xport (dataset)
ipt_fcl_storage (dataset)
ipt_fcl_xport (dataset)
ipt_locct_storage (dataset)
ipt_locct_xport (dataset)
ipt_mode_storage (dataset)
ipt_mode_xport (dataset)
ipt_pwb_pac_storage (dataset)
ipt_pwb_pac_xport (dataset)
iptra1 (dataset)
iptra2 (dataset)
ipty (dataset)
isb (dataset)
isback (dataset)
isfiss (dataset)
ism (dataset)
ispn (dataset)
isppar (dataset)
isrcwt (dataset)
issw (dataset)
istern (dataset)
istrg (dataset)
itag (dataset)
its30 (dataset)
iunr (dataset)
ivdd (dataset)
ivdis (dataset)
ivord (dataset)
iwwg (dataset)
ixread (dataset)
izdl (dataset)
izhl (dataset)
izread (dataset)
jgm (dataset)
jtb (dataset)
jtlx (dataset)
junf (dataset)
junf1 (dataset)
kf6 (dataset)
kf8c (dataset)
kf8r (dataset)
kfl (dataset)
kfl1 (dataset)

- kfq (dataset)
- kjaq (dataset)
- kjaq1 (dataset)
- knods (dataset)
- knrm (dataset)
- kpt (dataset)
- ktgd (dataset)
- ktl1 (dataset)
- ktls (dataset)
- kufil (dataset)
- lcaopt (dataset)
- leaopt (dataset)
- lemsh (dataset)
- len_dynamic_spb_data (dataset)
- length_elxs_neutron (dataset)
- length_elxs_proton (dataset)
- level_of_pht (dataset)
- level_of_var_red (dataset)
- lfcdg (dataset)
- lfcdj (dataset)
- lgd1 (dataset)
- lgd2 (dataset)
- lgd3 (dataset)
- list_dxt_to_go_to (dataset)
- list_secs_primaries (dataset)
- lit1 (dataset)
- loc_edep (dataset)
- locdt (dataset)
- lrt1 (dataset)
- ltal (dataset)
- ltd1 (dataset)
- ltgd (dataset)
- lvcdg (dataset)
- lvcdj (dataset)
- lxs (dataset)
- mai (dataset)
- max_mu_gam_lines (dataset)
- max_num_isotopes (dataset)
- max_num_reactions (dataset)
- mbbm1 (dataset)
- mbnk (dataset)
- mcal (dataset)
- mcal1 (dataset)
- mct (dataset)
- mdxflg (dataset)
- medflg (dataset)
- mem_reduct (dataset)
- mgegbt (dataset)
- mgm (dataset)
- mipt_ecut (dataset)
- mipt_ext (dataset)
- mipt_fcl (dataset)
- mipt_locct (dataset)
- mipt_mode (dataset)


mipt_pwb_pac (dataset)
mipts1 (dataset)
mix (dataset)
mix1 (dataset)
mix2 (dataset)
mjss (dataset)
mjss1 (dataset)
mkcp1 (dataset)
mlaf1 (dataset)
mlaj (dataset)
mlaj1 (dataset)
mlja (dataset)
mlja1 (dataset)
mnburn (dataset)
mnnm (dataset)
mnnm1 (dataset)
mnnm2 (dataset)
mode_electron_elastic (dataset)
mode_electron_straggling (dataset)
mode_pht_var (dataset)
mpdflg (dataset)
mrkp (dataset)
mrkp1 (dataset)
mroc (dataset)
msd (dataset)
msd1 (dataset)
mseb1 (dataset)
mshfm (dataset)
mspflg (dataset)
mssc1 (dataset)
msso1 (dataset)
mtfc (dataset)
mtlflg (dataset)
mu_cut (dataset)
mu_gam_isotopes (dataset)
mwwpf (dataset)
mwwpg (dataset)
mwwtf (dataset)
mwwtg (dataset)
mxa (dataset)
mxa1 (dataset)
mxafs (dataset)
mxafs1 (dataset)
mxe (dataset)
mxe1 (dataset)
mxf (dataset)
mxfp (dataset)
mxj (dataset)
mxj1 (dataset)
mxmel1 (dataset)
mxt (dataset)
mxt1 (dataset)
mxtr (dataset)
mxtr1 (dataset)

```
├── mxxs ..... (dataset)
├── mxxs1 ..... (dataset)
├── n_dxt_to_go_to ..... (dataset)
├── n_half_energy ..... (dataset)
├── napgg2 ..... (dataset)
├── naw1 ..... (dataset)
├── nbmx ..... (dataset)
├── ncobrn ..... (dataset)
├── ncom ..... (dataset)
├── ncom1 ..... (dataset)
├── nconb ..... (dataset)
├── ndae ..... (dataset)
├── ndbe ..... (dataset)
├── ndet ..... (dataset)
├── ndge ..... (dataset)
├── ndgnf ..... (dataset)
├── ndla ..... (dataset)
├── ndlb ..... (dataset)
├── ndlg ..... (dataset)
├── ndln ..... (dataset)
├── ndlp ..... (dataset)
├── ndnd ..... (dataset)
├── ndnd1 ..... (dataset)
├── ndne ..... (dataset)
├── ndpe ..... (dataset)
├── ndtt ..... (dataset)
├── ndx ..... (dataset)
├── nee ..... (dataset)
├── nee3 ..... (dataset)
├── nee_ech ..... (dataset)
├── nee_max ..... (dataset)
├── nemsh1 ..... (dataset)
├── nets ..... (dataset)
├── nf8c ..... (dataset)
├── nf8ed ..... (dataset)
├── nf8ed1 ..... (dataset)
├── nfmult ..... (dataset)
├── nfmult1 ..... (dataset)
├── ngam ..... (dataset)
├── ngd1 ..... (dataset)
├── ngd2 ..... (dataset)
├── ngd3 ..... (dataset)
├── ngdt ..... (dataset)
├── ngww ..... (dataset)
├── nhb ..... (dataset)
├── nilr ..... (dataset)
├── nilr1 ..... (dataset)
├── nilw ..... (dataset)
├── nilw1 ..... (dataset)
├── nips ..... (dataset)
├── nipt ..... (dataset)
├── niss ..... (dataset)
├── niwr1 ..... (dataset)
└── njsr ..... (dataset)
```

```
├── num_elxs_neutron_energies ..... (dataset)
├── num_elxs_neutron_isotopes ..... (dataset)
├── num_elxs_proton_energies ..... (dataset)
├── num_elxs_proton_isotopes ..... (dataset)
├── num_energy_bands ..... (dataset)
├── num_primaries_active ..... (dataset)
├── num_terms_in_gs ..... (dataset)
├── numb ..... (dataset)
├── nvec ..... (dataset)
├── nvec1 ..... (dataset)
├── nwang ..... (dataset)
├── nwgc1 ..... (dataset)
├── nwge1 ..... (dataset)
├── nwgeoa ..... (dataset)
├── nwgeom ..... (dataset)
├── nwgm ..... (dataset)
├── nwgm1 ..... (dataset)
├── nwgma ..... (dataset)
├── nwgma1 ..... (dataset)
├── nwgp1 ..... (dataset)
├── nwgt1 ..... (dataset)
├── nwma1 ..... (dataset)
├── nwng ..... (dataset)
├── nww ..... (dataset)
├── nwwa1 ..... (dataset)
├── nwwc1 ..... (dataset)
├── nwwe1 ..... (dataset)
├── nwwm ..... (dataset)
├── nwwma ..... (dataset)
├── nwwma1 ..... (dataset)
├── nwwp1 ..... (dataset)
├── nwwt1 ..... (dataset)
├── nxnx ..... (dataset)
├── nxnx1 ..... (dataset)
├── optlcb ..... (dataset)
├── optlcc ..... (dataset)
├── optleb ..... (dataset)
├── particleuse ..... (dataset)
├── pecut ..... (dataset)
├── phia ..... (dataset)
├── photon_solo ..... (dataset)
├── proj_mass_lib ..... (dataset)
├── rcoilf ..... (dataset)
├── regl_vol_src ..... (dataset)
├── rga ..... (dataset)
├── rim ..... (dataset)
├── rmem_r ..... (dataset)
├── rnfb ..... (dataset)
├── rnfs ..... (dataset)
├── rngb ..... (dataset)
├── rngs ..... (dataset)
├── rnmult ..... (dataset)
├── rnok ..... (dataset)
└── rrg ..... (dataset)
```

- rtc_size (dataset)
- scale_lib (dataset)
- smod (dataset)
- srv (dataset)
- tco (dataset)
- thgf (dataset)
- ttc (dataset)
- type_of_rssa (dataset)
- use_muon_II_data (dataset)
- wc1 (dataset)
- wc2 (dataset)
- wwg (dataset)
- wwm (dataset)
- wwma (dataset)
- wwp (dataset)
- xmugam (dataset)
- xunrl (dataset)
- xunru (dataset)

keff_bank (optional) (group)
- fso_decay_const (dataset)
- fso_delayed (dataset)
- fso_erg_inc (dataset)
- fso_fmat_bin0 (dataset)
- fso_kpert_max (dataset)
- fso_kpert_min (dataset)
- fso_max_count (dataset)
- fso_max_count1 (dataset)
- fso_max_items (dataset)
- fso_precursor (dataset)
- fso_progenitor (dataset)
- fso_srctp_count (dataset)
- fso_uran_max (dataset)
- fso_uran_min (dataset)

kplace (dataset)

lplace (dataset)

options (group)
- global_options (dataset)

xss (optional) (group)
- n : one for each data array (dataset)

XSS arrays (xss) - [back to tree]
The structure of these arrays corresponds to the ACE format specification [345].

D.2.3.1 Barpo

[back to tree]

barpo (optional) (group)
- aai1 (dataset)
- aai11 (dataset)

- aai2 (dataset)
- aai22 (dataset)
- iener (dataset)
- isig (dataset)
- ne (dataset)
- nel (dataset)

D.2.3.2 Embedded Mesh Geometry

[back to tree]
Variables related to embedded mesh geometries.

- embedded geometry (optional) (group)
  - edits_maxnum_embee (dataset)
  - edits_maxnum_energy (dataset)
  - edits_maxnum_response (dataset)
  - edits_maxnum_time (dataset)
  - embedded geometries (group)
    - n : one for each embedded geometry (group)
      - calc_vols (dataset)
      - card_identifier_number (dataset)
      - debug (dataset)
      - edits_file_type (dataset)
      - file_edit_in (dataset)
      - file_edit_out (dataset)
      - file_geom_in (dataset)
      - file_gmv_out (dataset)
      - file_type (dataset)
      - filetype_number (dataset)
      - length_conversion (dataset)
      - list_of_materials (optional) (dataset)
      - lnk3dnt (optional) (group)
      - mass_of_materials (dataset)
      - matcell (dataset)
      - mesh_instance_number (dataset)
      - mesh_volume (dataset)
      - multi_track (dataset)
      - num_materials (dataset)
      - track_opt (dataset)
      - volume_of_materials (dataset)
  - embedded_eBins2 (optional) (dataset)
  - embedded_eMults2 (optional) (dataset)
  - embedded_ee2file_map (optional) (dataset)
  - embedded_eeType (optional) (dataset)
  - embedded_embee_errors (optional) (dataset)
  - embedded_energy_conv2 (optional) (dataset)
  - embedded_meBins2 (optional) (dataset)
  - embedded_mrBins2 (optional) (dataset)
  - embedded_mtBins2 (optional) (dataset)
  - embedded_part_ee_map (optional) (dataset)
  - embedded_rBins2 (optional) (dataset)

- embedded_rMults2 (optional) (dataset)
- embedded_tBins2 (optional) (dataset)
- embedded_tMults2 (optional) (dataset)
- embedded_time_conv2 (optional) (dataset)
- embee2embed (optional) (dataset)
- embees (optional) (group)
- num_embedded_geoms (dataset)
- num_embees (dataset)
- num_regl_geoms (dataset)
- num_structured_geoms (dataset)
- rgli_file_type (optional) (dataset)

D.2.3.2.1 LNK3DNT Geometries

[back to tree]

- lnk3dnt (optional) (group)
  - constant_spacing (dataset)
  - den (dataset)
  - dx (dataset)
  - file_name (dataset)
  - file_number (dataset)
  - gram2num (dataset)
  - hname (dataset)
  - huse1 (dataset)
  - huse2 (dataset)
  - idc (dataset)
  - igom (dataset)
  - ilevel (dataset)
  - intrec (dataset)
  - ivers (dataset)
  - ninti (dataset)
  - nintj (dataset)
  - nintk (dataset)
  - nmxsp (dataset)
  - num_lnks (dataset)
  - nzone (dataset)
  - reserved (dataset)
  - xmesh (dataset)
  - ymesh (dataset)
  - zmesh (dataset)

D.2.3.2.2 Embedded Elemental Edits

[back to tree]

- embees (optional) (group)
  - n : one for each elemental edit (group)

```
├── comment ..... (dataset)
├── editNo ..... (dataset)
├── editType ..... (dataset)
├── embed ..... (dataset)
├── energy ..... (dataset)
├── errors ..... (dataset)
├── particle ..... (dataset)
├── time ..... (dataset)
├── wtmmesh_atom ..... (dataset)
├── wtmmesh_factor ..... (dataset)
├── wtmmesh_mat ..... (dataset)
├── wtmmesh_mtype ..... (dataset)
└── wtmmesh_nList ..... (dataset)
```

D.2.3.3 Magnetic Fields

[back to tree]
Variables related to magnetic fields.

```
mag_fields ..... (optional) (group)
├── bfield_cell ..... (dataset)
├── n :  one for each field ..... (optional) (group)
│   ├── axis ..... (dataset)
│   ├── cross ..... (dataset)
│   ├── deflect_max ..... (dataset)
│   ├── ff_kick_surf ..... (optional) (dataset)
│   ├── field ..... (dataset)
│   ├── itype ..... (dataset)
│   ├── nffedges ..... (dataset)
│   ├── ref_point ..... (dataset)
│   ├── step_max ..... (dataset)
│   └── vec ..... (dataset)
└── n_mag_fields ..... (dataset)
```

D.2.3.4 MARS Physics

[back to tree]
Variables used for MARS physics routines.

```
mars ..... (optional) (group)
├── AELEMT ..... (dataset)
├── EE5 ..... (dataset)
├── EMODEL ..... (dataset)
├── IQGSM ..... (dataset)
├── KEI ..... (dataset)
├── NELEMT ..... (dataset)
├── SIB ..... (dataset)
├── SINE5 ..... (dataset)
└── SIP ..... (dataset)
```

- SPRO5 (dataset)
- SSS (dataset)
- STOT5 (dataset)
- ZELEMT (dataset)

D.2.3.5 MCS2

[back to tree]
Variables for the MCS model.

- mcs2 (optional) (group)
  - s (dataset)
  - s2 (dataset)
  - s4 (dataset)

D.2.3.6 Tally Data

[back to tree]
Contains most tally data.

- tallies (group)
  - basic_tallies (group)
    - comdat (dataset)
    - f8fdep (dataset)
    - fmultd (dataset)
    - i_secondary_uncollided (dataset)
    - iptal (dataset)
    - itds (dataset)
    - jptal (dataset)
    - jtf (dataset)
    - lft (dataset)
    - locct (dataset)
    - locph (dataset)
    - locst (dataset)
    - tally_for_par (dataset)
    - tds (dataset)
  - particle_activity (group)
    - ipan (dataset)
    - npn (dataset)
    - pan_disabled (dataset)
  - tmesh (group)
    - emsh (dataset)

D.2.4 Header

[back to tree]
Contains MCNP-specific information about the problem and the code that generated this runtape.

```
header ..... (group)
├── aid ..... (dataset)
├── chcd ..... (dataset)
├── hdpath ..... (dataset)
├── hdpth ..... (dataset)
├── idtm ..... (dataset)
├── mxe ..... (dataset)
├── probid ..... (dataset)
├── probs ..... (dataset)
├── thrd_num_runtpe ..... (dataset)
├── thrd_runtpe ..... (dataset)
└── ufil ..... (group)
    └── nm :  n = 1:3 and m = 1:6 ..... (dataset)
```

D.2.5 Problem Information

[back to tree]
This group contains attributes that identify the problem that generated this file.

```
problem_info ..... (group)
├── environment_datapath ..... (optional) (attribute)
├── execution_start_time ..... (optional) (attribute)
├── problem_timestamp ..... (optional) (attribute)
└── problem_title ..... (attribute)
```

D.2.6 Restart

[back to tree]
This group contains information used during a restart run.

```
restart ..... (optional) (group)
└── unstructured_mesh ..... (group)
```

D.2.6.1 Unstructured Mesh Restart Information

[back to tree]
Used for restarting an unstructured mesh simulation. Contents are split into `fixed` and `variable` components, similar to the main MCNP restart capability.

```
unstructured_mesh ..... (group)
```

- latest (attribute)
- um_n : one for each embedded unstructured mesh (group)
  - fixed (group)
    - edits (group)
      - edit_n : one for each unstructured mesh edit (group)
        - contrib (dataset)
        - eBins (dataset)
        - eMults (dataset)
        - energy (attribute)
        - errors (attribute)
        - nEBins (attribute)
        - nRBins (attribute)
        - nTBins (attribute)
        - pNumber (attribute)
        - pNumberMaster (attribute)
        - pTag (attribute)
        - rBins (dataset)
        - rMults (dataset)
        - rxn (attribute)
        - rxn2 (attribute)
        - rxn3 (attribute)
        - tBins (dataset)
        - tMults (dataset)
        - time (attribute)
    - elementDensity (dataset)
    - meshFileType (attribute)
    - nElements (attribute)
    - nEmbee (attribute)
    - nInstances (attribute)
    - nMaterials (attribute)
    - nNodes (attribute)
    - nParticles (attribute)
    - numEELists (attribute)
    - particleEditList (dataset)
    - particles (group)
      - particle_n : one for each particle (group)
        - nTBins (attribute)
        - tBins (dataset)
  - variable (group)
    - n : one for each runtape dump (group)
      - edit_n : one for each unstructured mesh edit (group)
        - edit2 (dataset)
        - edits (dataset)
      - normFactor (attribute)
      - numHistories (attribute)

D.2.7 Simulation Results

[back to tree]
Processed simulation results.

```
results ..... (optional) (group)
├── fission_matrix ..... (optional) (group)
└── mesh_tally ..... (optional) (group)
```

D.2.7.1 Fission Matrix

[back to tree]
Processed fission matrix data. See §D.5 for more details.

```
fission_matrix ..... (optional) (group)
├── data ..... (dataset)
├── delta_xyz ..... (dataset)
├── indices ..... (dataset)
├── indptr ..... (dataset)
├── n ..... (dataset)
├── n_xyz ..... (dataset)
└── origin ..... (dataset)
```

D.2.7.2 Mesh Tallies

[back to tree]
Mesh tally processed results. See §D.4.1 for more details.

```
mesh_tally ..... (optional) (group)
└── mesh_tally_n :  n corresponds to tally ID ..... (optional) (group)
    ├── basis_axis ..... (optional) (dataset)
    ├── basis_cross ..... (optional) (dataset)
    ├── basis_vector ..... (optional) (dataset)
    ├── coordinate_system ..... (attribute)
    ├── coords ..... (optional) (dataset)
    ├── grid_energy ..... (dataset)
    ├── grid_r ..... (optional) (dataset)
    ├── grid_t ..... (optional) (dataset)
    ├── grid_time ..... (dataset)
    ├── grid_x ..... (optional) (dataset)
    ├── grid_y ..... (optional) (dataset)
    ├── grid_z ..... (dataset)
    ├── has_collision_binning ..... (attribute)
    ├── has_comment_lines ..... (attribute)
    ├── has_dose_response_function ..... (attribute)
    ├── has_flagged_cells ..... (attribute)
    ├── has_flagged_surfaces ..... (attribute)
    ├── has_score_multiplier ..... (attribute)
    ├── has_transformation ..... (attribute)
    ├── is_isotopic_reaction_rate_tally ..... (attribute)
    ├── mean ..... (dataset)
    ├── multiplicative_factor ..... (attribute)
    ├── particle_number ..... (attribute)
    └── relative_standard_error ..... (dataset)
```

- tally_quantity (attribute)
- tally_type (attribute)

D.2.8 Variable Data

[back to tree]

Data that changes throughout the simulation. Each `n` inside of `variable` is a separate runtape dump. The latest one is given by the variable `latest`.

- variable (group)
  - latest (dataset)
  - n : one for each runtape dump (group)
    - Russian_Roulette (dataset)
    - chcd (dataset)
    - entropy (optional) (group)
    - fmat (optional) (group)
    - fmesh (optional) (group)
    - idtm (dataset)
    - idum (dataset)
    - kadjoint (optional) (group)
    - kblock (optional) (group)
    - mcnpx_mesh (optional) (group)
    - mesh_xyz (optional) (group)
    - phtvr_flag (dataset)
    - probid (dataset)
    - rdum (dataset)
    - stop_flag (dataset)
    - uran (optional) (group)
    - varcom (group)
      - afis (dataset)
      - batch_count (dataset)
      - bcw (dataset)
      - coll (dataset)
      - cpk (dataset)
      - cts (dataset)
      - dbcn (dataset)
      - dmp (dataset)
      - erg_gt_emax (dataset)
      - ffis (dataset)
      - fission_stats (dataset)
      - init (dataset)
      - inpd (dataset)
      - ion_chg (dataset)
      - ion_s (dataset)
      - ion_src_a (dataset)
      - ion_src_chg (dataset)
      - ion_src_z (dataset)
      - iqtal (dataset)
      - iroc (dataset)
      - jrad (dataset)
      - kc8 (dataset)

- kcsf (dataset)
- kct (dataset)
- kcy (dataset)
- kcz (dataset)
- keff_cycle_col (dataset)
- keff_pop_control (dataset)
- knod (dataset)
- ksdef (dataset)
- ksource_rn_count (dataset)
- ksource_rn_seed (dataset)
- lost (dataset)
- lost_count (dataset)
- model_count (dataset)
- monod (dataset)
- n_per_batch (dataset)
- nbhwm (dataset)
- nbov (dataset)
- nbt (dataset)
- ncburn (dataset)
- ndmp (dataset)
- ndpw (dataset)
- nesm (dataset)
- netb (dataset)
- nfer (dataset)
- notal (dataset)
- notrn (dataset)
- npd (dataset)
- npnm (dataset)
- npp (dataset)
- nppm (dataset)
- npred (dataset)
- nps (dataset)
- nps_start (dataset)
- nps_stop (dataset)
- npsmg (dataset)
- npsout (dataset)
- npsr (dataset)
- npum (dataset)
- npxm (dataset)
- nqss (dataset)
- nqsw (dataset)
- nrnh (dataset)
- nrrs (dataset)
- nrsw (dataset)
- nskk (dataset)
- nsom (dataset)
- nssi (dataset)
- ntc (dataset)
- ntc1 (dataset)
- ntprt (dataset)
- ntss (dataset)
- nwer (dataset)
- nwsb (dataset)
- nwse (dataset)

```
├── nwsg ..... (dataset)
├── nwst ..... (dataset)
├── nwws ..... (dataset)
├── nziy ..... (dataset)
├── osum ..... (dataset)
├── osum2 ..... (dataset)
├── pax ..... (dataset)
├── prn ..... (dataset)
├── qital ..... (dataset)
├── racc ..... (dataset)
├── rani ..... (dataset)
├── ranj ..... (dataset)
├── rijk ..... (dataset)
├── rkk ..... (dataset)
├── rlt ..... (dataset)
├── rnr ..... (dataset)
├── rssp ..... (dataset)
├── rsum ..... (dataset)
├── rsum2 ..... (dataset)
├── smul ..... (dataset)
├── snit ..... (dataset)
├── sumk ..... (dataset)
├── swtm ..... (dataset)
├── tfis ..... (dataset)
├── tmav ..... (dataset)
├── twac ..... (dataset)
├── twss ..... (dataset)
├── walltime_start ..... (dataset)
├── walltime_stop ..... (dataset)
├── wcs1 ..... (dataset)
├── wcs2 ..... (dataset)
├── wgts ..... (dataset)
├── wssi ..... (dataset)
└── wt0 ..... (dataset)
vdac ..... (group)
    ├── ara ..... (dataset)
    ├── atcin ..... (dataset)
    ├── awamu ..... (dataset)
    ├── bbank ..... (dataset)
    ├── bumtu ..... (dataset)
    ├── burn_table ..... (dataset)
    ├── cosy ..... (optional) (group)
    ├── den ..... (dataset)
    ├── eflxb ..... (dataset)
    ├── findcel ..... (group)
    │   └── source_cell ..... (dataset)
    ├── flxb ..... (dataset)
    ├── fma ..... (dataset)
    ├── fme ..... (dataset)
    ├── isef ..... (dataset)
    └── keff_bank ..... (optional) (group)
        ├── fso_bnk ..... (dataset)
        ├── fso_bnk_count ..... (dataset)
        └── fso_bnk_next ..... (dataset)
```

- fso_src (dataset)
- fso_src_count (dataset)
- fso_src_next (dataset)
- laj (dataset)
- lcaj (dataset)
- lfcl (dataset)
- lse (dataset)
- maze (dataset)
- ndr (dataset)
- npsw (dataset)
- nsl (dataset)
- numvse (dataset)
- pfburn (dataset)
- ptb_keff (optional) (dataset)
- rho (dataset)
- rkpl (dataset)
- roc (dataset)
- sinbrn (dataset)
- tallies (group)
- vol (dataset)
- wns (dataset)
- wwg_entering_all_e_t (dataset)
- wwg_entering_by_e_t (dataset)
- wwg_score_all_e_t (dataset)
- wwg_score_by_e_t (dataset)
- wwlhs (dataset)
- xflxb (dataset)
- xlk (dataset)
- xs_calc (optional) (group)
- yla (dataset)
- zwc (dataset)

D.2.8.1 Shannon Entropy

[back to tree]
Variables related to the calculation of Shannon Entropy.

- entropy (optional) (group)
  - hsrc_card_used (dataset)
  - hsrc_init_mesh (dataset)
  - hsrc_ixyz (dataset)
  - hsrc_xyz1 (dataset)
  - hsrc_xyz2 (dataset)

D.2.8.2 Fission Matrix

[back to tree]
Variables related to the fission matrix.

- fmat (optional) (group)
  - fmat%J (optional) (dataset)
  - fmat%L (optional) (dataset)
  - fmat%R (optional) (dataset)
  - fmat%chunk (optional) (dataset)
  - fmat%dom_ratio (optional) (dataset)
  - fmat%eigval (optional) (dataset)
  - fmat%eigvec (optional) (dataset)
  - fmat%entropy (optional) (dataset)
  - fmat%kcycle1 (optional) (dataset)
  - fmat%kcycle2 (optional) (dataset)
  - fmat%n (optional) (dataset)
  - fmat%neutrons (optional) (dataset)
  - fmat%nnz (optional) (dataset)
  - fmat%nnz_max (optional) (dataset)
  - fmat%src (optional) (dataset)
  - fmat%src_active (optional) (dataset)
  - fmat%src_ncyc (optional) (dataset)
  - fmat%srcp (optional) (dataset)
  - fmat_accel (dataset)
  - fmat_autosrc (dataset)
  - fmat_convrg (dataset)
  - fmat_debug (dataset)
  - fmat_initialized (dataset)
  - fmat_iters (dataset)
  - fmat_ncyc (dataset)
  - fmat_nx (dataset)
  - fmat_ny (dataset)
  - fmat_nz (dataset)
  - fmat_opt (dataset)
  - fmat_rnseed (dataset)
  - fmat_skip (dataset)
  - fmat_space (dataset)
  - n (optional) (dataset)
  - nnzmax (optional) (dataset)

D.2.8.3 FMESH Tallies

[back to tree]
Variables related to FMESH tallies.

- fmesh (optional) (group)
  - n : one for each tally (group)
    - axs (dataset)
    - batches (optional) (attribute)
    - bins_per_region (attribute)
    - comments (optional) (group)
      - n : one for each comment line (dataset)
    - crs (dataset)
    - de (optional) (dataset)
    - df (optional) (dataset)

- enbin (dataset)
- fact (dataset)
- fmult (dataset)
- icrd (dataset)
- icx (dataset)
- id (dataset)
- ifm_card (dataset)
- inc_lower (dataset)
- inc_upper (dataset)
- intrpol (dataset)
- ipt (dataset)
- isotopic_reac_rate_fm_tal (dataset)
- itr (dataset)
- kclear (dataset)
- lemesh (dataset)
- lenorm (dataset)
- ltmesh (dataset)
- ltnorm (dataset)
- mat (dataset)
- mytype (dataset)
- n_comment_lines (dataset)
- ncf (dataset)
- ndfb (dataset)
- nenb (dataset)
- nireact (dataset)
- nreact (dataset)
- nsf (dataset)
- ntib (dataset)
- nxrb (dataset)
- nyzb (dataset)
- nztb (dataset)
- org (dataset)
- outf (dataset)
- react (optional) (dataset)
- results_1 (dataset)
- results_2 (dataset)
- tally_mode (dataset)
- tibin (dataset)
- tot_energy_bin (dataset)
- tot_time_bin (dataset)
- total_bins (attribute)
- total_histories (optional) (attribute)
- total_weight (optional) (attribute)
- vec (dataset)
- weight_norm (attribute)
- xrbin (dataset)
- yzbin (dataset)
- ztbin (dataset)

- nmesh (dataset)
- write_all_xdmf (attribute)

D.2.8.4 Adjoint Capabilities

[back to tree]
Variables related to perturbation theory.

```
kadjoint ..... (optional) (group)
├── current_generation ..... (dataset)
├── kadjoint_keff ..... (dataset)
├── kinetics ..... (optional) (group)
├── kpert ..... (optional) (group)
├── ksen ..... (optional) (group)
├── max_progenitor_id ..... (optional) (dataset)
├── max_progenitors ..... (dataset)
├── neutron_production ..... (optional) (dataset)
├── progenitor_block_size ..... (dataset)
└── progenitor_root ..... (dataset)
```

D.2.8.4.1 Point-Kinetics Tallies

[back to tree]
Variables from the point-kinetics functionality from `KOPTS`.

```
kinetics ..... (optional) (group)
├── do_delay ..... (dataset)
├── do_precursor ..... (dataset)
├── kin_delay ..... (optional) (dataset)
├── kin_delgp ..... (optional) (dataset)
├── kin_dsrc_var ..... (optional) (dataset)
├── kin_fisrc ..... (dataset)
├── kin_fission ..... (dataset)
├── kin_fsrc_dsrc_covar ..... (optional) (dataset)
├── kin_fsrc_var ..... (dataset)
├── kin_nden_dsrc_covar ..... (dataset)
├── kin_nden_fsrc_covar ..... (dataset)
├── kin_nden_var ..... (dataset)
├── kin_ndensity ..... (dataset)
├── kin_precursor_info ..... (optional) (dataset)
├── kin_src_wgt ..... (dataset)
├── kin_track ..... (dataset)
└── num_delgrps ..... (dataset)
```

D.2.8.4.2 Reactivity Perturbations

[back to tree]
Variables for the `KPERT` functionality.

```
kpert ..... (optional) (group)
```

- cell data (group)
  - n : one for each perturbation (group)
    - nperts (dataset)
    - nrxns (dataset)
    - rxn (optional) (dataset)
- kpert_src_wgt (dataset)
- mem_kpert (dataset)
- mt_flags (dataset)
- num_kperts (dataset)
- perturbations (group)
  - n : one for each perturbation (group)
    - cell (dataset)
    - denom (dataset)
    - erg (optional) (dataset)
    - id (dataset)
    - ital (dataset)
    - mat (dataset)
    - ncells (dataset)
    - nerg (dataset)
    - niso (dataset)
    - nrxns (dataset)
    - rho (dataset)
    - rxn (optional) (dataset)
    - user_id (dataset)
- tallies (group)
  - kpert_covar (dataset)
  - kpert_dcolrate (dataset)
  - kpert_denom (dataset)
  - kpert_denom_var (dataset)
  - kpert_dfission (dataset)
  - kpert_dscatter (dataset)
  - kpert_numer (dataset)
  - kpert_numer_var (dataset)
  - kpert_pfission (dataset)

D.2.8.4.3 k_{eff} Sensitivity Coefficients

[back to tree]
Variables for the KSEN functionality.

- ksen (optional) (group)
  - ksen_file_format (dataset)
  - ksen_kinds (dataset)
  - ksen_user_ids (dataset)
  - number_of_ksens (dataset)
  - xs (group)
    - dk_store (dataset)
    - dk_store_loc (dataset)
    - dk_store_size (dataset)
    - inelastic grid = n : one for each cross section table if needed (group)
      - inelastic (optional) (dataset)

```
    │
    ├── inelastic_num ..... (dataset)
    ├── inelastic_start ..... (dataset)
    ├── nufission_num ..... (dataset)
    ├── nufission_start ..... (dataset)
    ├── nxn_num ..... (dataset)
    ├── nxn_start ..... (dataset)
    ├── total_prod_num ..... (dataset)
    └── total_prod_start ..... (dataset)
├── ksen = n :  one for each sensitivity profile ..... (group)
    ├── chi ..... (dataset)
    ├── chi_exists ..... (dataset)
    ├── chi_mts ..... (optional) (dataset)
    ├── chi_needed ..... (dataset)
    ├── chi_nmts ..... (dataset)
    ├── chi_nuprod ..... (optional) (dataset)
    ├── chi_tal ..... (optional) (dataset)
    ├── chi_total ..... (optional) (dataset)
    ├── chi_var ..... (optional) (dataset)
    ├── cos ..... (dataset)
    ├── ein ..... (dataset)
    ├── erg ..... (dataset)
    ├── id ..... (dataset)
    ├── iso ..... (dataset)
    ├── mts ..... (dataset)
    ├── ncos ..... (dataset)
    ├── nein ..... (dataset)
    ├── nerg ..... (dataset)
    ├── niso ..... (dataset)
    ├── nleg ..... (dataset)
    ├── nmts ..... (dataset)
    ├── nzone ..... (dataset)
    ├── slaw ..... (dataset)
    ├── slaw_exists ..... (dataset)
    ├── slaw_mts ..... (optional) (dataset)
    ├── slaw_needed ..... (dataset)
    ├── slaw_nmts ..... (dataset)
    ├── slaw_nuprod ..... (optional) (dataset)
    ├── slaw_tal ..... (optional) (dataset)
    ├── slaw_total ..... (optional) (dataset)
    ├── slaw_var ..... (optional) (dataset)
    ├── tal ..... (dataset)
    ├── tmts ..... (dataset)
    ├── var ..... (dataset)
    ├── zaids ..... (group)
    │   └── n :  one for each ZAID ..... (dataset)
    ├── zone = n :  one for each spatial region ..... (optional) (group)
    │   ├── k ..... (dataset)
    │   └── n ..... (dataset)
    └── zone_type ..... (dataset)
├── ksen_xs_norm ..... (dataset)
├── ksen_xs_num ..... (dataset)
├── local_dk_loc ..... (dataset)
├── local_dk_num ..... (dataset)
└── need_inelastic_grid ..... (dataset)
```

- need_ksen_chi (dataset)
- need_ksen_slaw (dataset)

D.2.8.5 k-eigenvalue Convergence Information

[back to tree]
Contains generation information during k-eigenvalue simulations used for auto-convergence.

- kblock (optional) (group)
  - IKZ_USER_INPUT (dataset)
  - KCT_USER_INPUT (dataset)
  - KCY_CONVERGED (dataset)
  - TARGETS_INITIALIZED (dataset)
  - TARGET_CHISQUARE (dataset)
  - TARGET_DELTA_FMAT_NNZ (dataset)
  - TARGET_DELTA_FMAT_SRC (dataset)
  - TARGET_DISTRIB_TEST_CONF (dataset)
  - TARGET_H_DIFF (dataset)
  - TARGET_H_RELATIVE (dataset)
  - TARGET_H_SLOPE (dataset)
  - TARGET_H_SLOPE_CONF (dataset)
  - TARGET_KOLMOGOROV (dataset)
  - TARGET_K_SLOPE (dataset)
  - TARGET_K_SLOPE_CONF (dataset)
  - WTF_MAX_WGT (dataset)
  - WTF_MIN_WGT (dataset)
  - kblock%bcycle (dataset)
  - kblock%cycle (dataset)
  - kblock%d_block_cycle_ave (dataset)
  - kblock%d_block_neutfmat (dataset)
  - kblock%d_neutfmat (dataset)
  - kblock%h_block_cycle_ave (dataset)
  - kblock%h_block_fmat (dataset)
  - kblock%h_block_neut (dataset)
  - kblock%h_neut (dataset)
  - kblock%h_neut_x (dataset)
  - kblock%h_neut_y (dataset)
  - kblock%h_neut_z (dataset)
  - kblock%k_block_fmat (dataset)
  - kblock%keff (dataset)
  - kblock%keff_abs (dataset)
  - kblock%keff_col (dataset)
  - kblock%keff_trk (dataset)
  - kblock%kstd (dataset)
  - kblock%ncycles (dataset)
  - kblock%nnz_mat (dataset)
  - kblock%nnz_src (dataset)

D.2.8.6 TMESH Tallies

[back to tree]
Variables related to `TMESH` tallies.

- mcnpx_mesh (optional) (group)
  - ergh (dataset)
  - ergl (dataset)
  - gdata (dataset)
  - mdos (dataset)
  - mmult (dataset)
  - mshpt (dataset)
  - mshtr (dataset)
  - mstyp (dataset)
  - mxgc (dataset)
  - mxgt (dataset)
  - mxgv (dataset)
  - nemsh (dataset)
  - ng1 (dataset)
  - ng2 (dataset)
  - ng3 (dataset)
  - ngv (dataset)
  - nugd (dataset)
  - rdos (dataset)
  - rmult (dataset)

D.2.8.7 Fission Matrix/Shannon Entropy Mesh

[back to tree]
Contains geometric information on the shared fission matrix and Shannon entropy mesh.

- mesh_xyz (optional) (group)
  - delta (dataset)
  - dxyz (dataset)
  - extend (dataset)
  - ii (dataset)
  - n (dataset)
  - nold (dataset)
  - nxyz (dataset)
  - remap (optional) (dataset)
  - rr (dataset)
  - xyz1 (dataset)
  - xyz2 (dataset)

D.2.8.8 Random Universe Translation

[back to tree]
Variables related to random universe translation.

```
uran ..... (optional) (group)
├── uran_n ..... (dataset)
├── uran_univ ..... (dataset)
└── uran_xyz ..... (dataset)
```

D.2.8.9 COSY Magnetic Field Transfer Maps

[back to tree]
Variables related to COSY.

```
cosy ..... (optional) (group)
├── cmap ..... (dataset)
├── cosypar ..... (dataset)
├── icosyh ..... (dataset)
├── icosyl ..... (dataset)
├── icosyv ..... (dataset)
├── idl ..... (dataset)
├── ihaper ..... (dataset)
├── iidt ..... (dataset)
├── ixp ..... (dataset)
├── ixx ..... (dataset)
├── iyp ..... (dataset)
├── iyy ..... (dataset)
├── mapcell ..... (dataset)
├── mapname ..... (dataset)
├── mterms ..... (dataset)
├── nhaper ..... (dataset)
├── nmaps ..... (dataset)
└── nterms ..... (dataset)
```

D.2.8.10 Tally Data

[back to tree]
Contains most tally data.

```
tallies ..... (group)
└── basic_tallies ..... (group)
    ├── ddm ..... (dataset)
    ├── ddn ..... (dataset)
    ├── ddn_counter ..... (dataset)
    ├── dec ..... (dataset)
    ├── dxc ..... (dataset)
    ├── dxd ..... (dataset)
    ├── febl ..... (dataset)
    ├── flx ..... (dataset)
    ├── jfq ..... (dataset)
    ├── ndpf ..... (dataset)
    └── nhsd ..... (optional) (dataset)
```

- npc (dataset)
- ntbb (dataset)
- shsd (optional) (dataset)
- tal (optional) (dataset)
- tal_fircopy (optional) (dataset)
- tfc (dataset)

particle_activity (group)
- pac (dataset)
- pan (optional) (dataset)
- pcc (dataset)
- pwb (dataset)

tmesh (group)
- gd1 (dataset)
- gd2 (dataset)
- gd3 (dataset)

D.2.8.11 GENXS Tallies

[back to tree]
Variables related to the GENXS capability.

xs_calc (optional) (group)
- ergb (dataset)
- fnorm0 (dataset)
- i_type (dataset)
- i_typesup (dataset)
- ihiaz (dataset)
- ip_xc (dataset)
- jtitl1 (dataset)
- jtitl2n : one n for each edit (dataset)
- kplot0 (dataset)
- l_res (dataset)
- llang (dataset)
- llerg (dataset)
- lltyp (dataset)
- lowaz (dataset)
- lp_xc (dataset)
- lxstal (dataset)
- ncase (dataset)
- nheavy (dataset)
- xmu (dataset)

D.3 Particle Track Output File Format

This section details the current layout of the HDF5-formatted particle track output file. Examples of how to process and interpret the data in the particle track file using the Python programming language and MCNPTools can be found in [346].

Note that the deprecated ASCII and unformatted binary particle track output file formats [DEP-53382] are unchanged from the description of them given in [2, Appendix D], which is what should be used as the format reference for these output files.

All datasets in the HDF5-formatted particle track output file are composed of compound datatypes, and the datasets may be of length zero. Event types (and other uniquely identifiable fields, e.g., bank types) are stored as enumerations in HDF5. An enumeration is a numerical value with a corresponding unique string that describes the value. The enumeration mapping is written to the file as part of the meta data for the corresponding dataset. In the future, some compound data types may be extended and some fields with no value to users may be removed or replaced. However, the MCNPTools capability to parse the current HDF5 format will be maintained to provide users a stable interface [307].

D.3.1 Main Layout

The following sections detail the layout of the HDF5-formatted particle track output file. The **h5dump** utility provided with HDF5 distributions is a convenient way to inspect the numerical representation and available data fields of each compound data type in a human readable format.

The groups and datasets are organized as follows:

```
/ ..... (root)
├── config_control ..... (group)
├── problem_info ..... (group)
└── ptrack ..... (group)
    ├── RecordLog ..... (dataset)
    ├── Bank ..... (dataset)
    ├── Collision ..... (dataset)
    ├── Source ..... (dataset)
    ├── SurfaceCrossing ..... (dataset)
    └── Termination ..... (dataset)
```

D.3.2 Configuration Control

See §D.2.2 for more information on the contents of this group.

D.3.3 Problem Information

See §D.2.5 for more information on the contents of this group.

Table D.1: RecordLog compound data type fields

Field	Description
nps	History identifier for this event
node	(placeholder)
event_array_index	zero-based index of event in corresponding array
type	enumerated type of event

Table D.2: Event type HDF5 enumeration

Event type	Number
source	1000
bank	2000
surface_crossing	3000
collision	4000
termination	5000

D.3.4 RecordLog

Table D.1 describes the record log compound data type. The record log dataset in the **ptrack** group provides information on the order that events occurred during the simulation, for all histories. Each entry in this dataset corresponds to a particular event and contains the **NPS** number, event type (described in Table D.2), and the zero-based index into the corresponding event array, where index zero corresponds to the first entry in the corresponding event array. The **NPS** number uniquely identifies a history and may be unordered in the record log for simulations that used multiple threads. Record log entries also include the node number. The node number is just a placeholder for a future feature to identify relations between events and should not be depended upon in use of the record log.

The record log dataset only needs to be processed if the order of events during a history is necessary for the particular analysis being performed. Otherwise, the events in each of the individual event arrays can be processed independently.

D.3.4.1 Interpreting the Record Log with Secondary Particles

When secondary particles are involved, from the occurrence of physical particle production or variance reduction, the data in the record log will not appear in the chronological order of the particle simulation; the order in the table is governed by how MCNP processes secondary particles that have been added to the bank. Secondary particles are added and removed from the bank throughout the simulation as a stack, where the particle added last to the bank is removed and processed first. All of the events of the primary track are processed and added to the record log in order through termination, and then particles in the bank are fully processed with their events added to the log in order. The banked particles are processed in order of last added until the bank is emptied, noting that more secondary particles can be added to the bank during the processing of a banked particle. If reconstructing the branching of tracks within a history is required, the location and time of collision, surface crossing, and bank events can typically be used, but the process is not always straight forward. There are some cases, e.g., DXTRAN-related bank events, where it is not possible to explicitly reconstruct the history branching.

As an example of how to interpret the record log, consider the tabular representation of example entries given in Table D.3. This data represents the events for two histories, identified by **NPS** 10 and 11. Remember that the values in the **event_array_index** dataset use zero-based indices.

For the history with **NPS** 10, a particle is created with a source event and then terminates. The description of the source event is given by index 8 in the Source dataset. The termination event is given in index 13 of

Table D.3: Example entries of the RecordLog. Each row in the table represents the data for a single instance of the compound datatype in the RecordLog dataset. The value of **node** for the entries is omitted here.

nps	event_array_index	type
10	8	source
10	13	termination
11	9	source
11	23	collision
11	24	collision
11	14	termination
11	5	bank
11	25	collision
11	15	termination

the Termination dataset. For the history with **NPS** 11, a secondary particle track has been created, assumed here to be from a collision during the primary track. The primary track consisted of a source event with index 9, two collisions with indices 23 and 24, and a termination event with index 14. The secondary particle track is created with details given by the bank event with index 5. The secondary track then consisted of a collision with index 25 and termination with index 15. To determine which event created the secondary track, it would be necessary to look at the bank type enumeration given for the index 5 bank event and match it to the collision type of index 23 or 24. If the bank type does not uniquely identify the collision, then the bank event's particle location or time may be matched to the index 23 or 24 collision event that created the particle.

Table D.4: Particle type HDF5 enumeration

Particle type	Number
HEAVY_ION	37
K_MINUS	36
PI_MINUS	35
ALPHA	34
HELION	33
TRITON	32
DEUTERON	31
AOMEGA_MINUS	30
XI_PLUS	29
AXI0	28
ASIGMA_MINUS	27
ASIGMA_PLUS	26
ALAMBDA0	25
K0_LONG	24
K0_SHORT	23
K_PLUS	22
PI_ZERO	21
PI_PLUS	20
APROTON	19
ANU_M	18
ANU_E	17
MU_PLUS	16
SIGMA_ZERO	−1
OMEGA_MINUS	15
ASIGMA_ZERO	−2
XI_MINUS	14
PIDROGEN	−3
XI0	13
SIGMA_MINUS	12
SIGMA_PLUS	11
LAMBDA0	10
PROTON	9
POSITRON	8
NU_M	7
NU_E	6
ANEUTRON	5
MU_MINUS	4
ELECTRON	3
PHOTON	2
NEUTRON	1

Table D.5: Bank event compound data type fields

Data Field	Description
x	x coordinate of the particle position
y	y coordinate of the particle position
z	z coordinate of the particle position
u	Particle direction cosine relative to $+x$ axis
v	Particle direction cosine relative to $+y$ axis
w	Particle direction cosine relative to $+z$ axis
energy	Particle energy
weight	Particle weight
time	Time at particle position
nps	History identifier
node	Number of nodes in track from source to here
material_id	Program material ID of the cell containing event
cell_id	Problem number of the cell containing event
num_collisions_this_branch	Count of collisions per track
reaction_type	Number identifying the bank reaction type that created banked particle
bank_type	Bank type enumeration
zaid	**ZZZAAA** for reaction isotope, following the ZAID format [§1.2.2]
particle_type	Particle type enumeration

Table D.6: Reaction types that caused creation of banked particle for associated bank event. N.B.: for the specific reactions listed below, the reaction type number is different for bank events than the reaction type number for the corresponding collision event.

Number	Description
Incident neutron	
1	Inelastic $S(\alpha, \beta)$
2	Elastic $S(\alpha, \beta)$
-99	Elastic scatter / Inelastic Scatter
>5	ENDF Reaction ID (MT number)
Incident photon	
1	Incoherent scatter
2	Coherent scatter
3	Fluorescence / Single Fluorescence
4	Double Fluorescence
5	Pair production

D.3.5 Bank

Bank events represent the removal of a particle from the particle bank during transport, i.e., the birth of a secondary particle track. For bank events not created by a collision, e.g., a DXTRAN particle created from a source, the reaction_type and zaid entries will be zero.

Table D.7: Bank type HDF5 enumeration

bank_type	Number	Description
BANK_DXT_TRACK	1	DXTRAN track
BANK_ERG_TME_SPLIT	2	Energy or time split
BANK_WWS_SPLIT	3	Weight-window surface split
BANK_WWC_SPLIT	4	Weight-window collision split
BANK_UNC_TRACK	5	Forced collision-uncollided particle
BANK_IMP_SPLIT	6	Importance split
BANK_N_XN_F	7	Neutron from (n, xn) or (n, f) and secondary particle from library protons
BANK_N_XG	8	Photon from Neutron
BANK_FLUORESCENCE	9	Photon from double fluorescence
BANK_ANNIHILATION	10	Photon from annihilation
BANK_PHOTO_ELECTRON	11	Electron from photo-electric effect
BANK_COMPT_ELECTRON	12	Electron from Compton scatter
BANK_PAIR_ELECTRON	13	Electron from pair production
BANK_AUGER_ELECTRON	14	Auger electron from photon/x-ray
BANK_PAIR_POSITRON	15	Positron from pair production
BANK_BREMSSTRAHLUNG	16	Bremsstrahlung from electron
BANK_KNOCK_ON	17	Knock-on electron
BANK_K_X_RAY	18	X-rays from electron
BANK_N_XG_MG	19	Photon from multigroup neutron (n, p) reaction
BANK_N_XF_MG	20	Multigroup neutron (n, f) reaction
BANK_N_XN_MG	21	Multigroup neutron (n, xn) reaction
BANK_G_XG_MG	22	Multigroup (p, xp) (multiplying) reaction
BANK_ADJ_SPLIT	23	Adjoint weight split - multigroup
BANK_WWT_SPLIT	24	Weight-window pseudo-collision split
BANK_PHOTONUCLEAR	25	Secondary particles from photonuclear
BANK_DECAY	26	Secondary emission from a decay
BANK_NUCLEAR_INT	27	Nuclear interaction
BANK_RECOIL	28	Recoil particle
BANK_DXTRAN_ANNIHIL	29	DXTRAN annihilation photon from pulse-height tally variance reduction
BANK_N_CHARGED_PART	30	Light ions from neutrons
BANK_H_CHARGED_PART	31	Light ions from protons
BANK_N_TO_TABULAR	32	Library neutrons from model neutrons
BANK_MODEL_UPDAT1	33	Secondary particles from inelastic nuclear interactions
BANK_MODEL_UPDATE	34	Secondary particles from elastic nuclear interactions
BANK_DELAYED_NEUTRON	35	Delayed neutron
BANK_DELAYED_PHOTON	36	Delayed photon
BANK_DELAYED_BETA	37	Delayed electron
BANK_DELAYED_ALPHA	38	Delayed alpha
BANK_DELAYED_POSITRN	39	Delayed positron
BANK_SPON_FISS	40	Spontaneous fission source particle
BANK_SURF_SRC	41	Surface Source Read (SSR) source particle

Table D.8: Collision event compound data type fields

Data Field	Description
x	x coordinate of the particle position
y	y coordinate of the particle position
z	z coordinate of the particle position
u	Particle direction cosine relative to $+x$ axis
v	Particle direction cosine relative to $+y$ axis
w	Particle direction cosine relative to $+z$ axis
energy	Particle energy
weight	Particle weight
time	Time at particle position
nps	History identifier
node	Number of nodes in track from source to here
material_id	Program material ID of the cell containing event
cell_id	Problem number of the cell containing event
num_collisions_this_branch	Count of collisions per track
reaction_type	Number identifying the reaction type
zaid	**ZZZAAA** for reaction isotope, following the ZAID format [§1.2.2]
particle_type	particle type enumeration

Table D.9: Reaction types for collision events.

Number	Description
Incident neutron	
4	Inelastic $S(\alpha, \beta)$
-2	Elastic $S(\alpha, \beta)$
>0	ENDF Reaction ID (MT number)
Incident photon	
-1	Incoherent scatter
-2	Coherent scatter
-3	Fluorescence
-4	Pair production

D.3.6 Collision

Table D.8 describes the compound data type representing a particle collision event.

Table D.10: Source event compound data type fields

Data Field	Description
x	x coordinate of the particle position
y	y coordinate of the particle position
z	z coordinate of the particle position
u	Particle direction cosine relative to $+x$ axis
v	Particle direction cosine relative to $+y$ axis
w	Particle direction cosine relative to $+z$ axis
energy	Particle energy
weight	Particle weight
time	Time at particle position
nps	History identifier
node	Number of nodes in track from source to here
material_id	Program material ID of the cell containing event
cell_id	Problem number of the cell containing event
num_collisions_this_branch	Count of collisions per track
source_type	Number identifying the source type (See **nsr** in MCNP5 Vol. III manual [312])
particle_type	Particle type enumeration

D.3.7 Source

Table D.10 describes source events, which represent the creation of a particle at the beginning of a history.

Table D.11: Surface crossing event compound data type fields

Data Field	Description
x	x coordinate of the particle position
y	y coordinate of the particle position
z	z coordinate of the particle position
u	Particle direction cosine relative to $+x$ axis
v	Particle direction cosine relative to $+y$ axis
w	Particle direction cosine relative to $+z$ axis
energy	Particle energy
weight	Particle weight
time	Time at particle position
nps	History identifier
node	Number of nodes in track from source to here
material_id	Program material ID of the cell containing event
cell_id	Problem number of the cell containing event
num_collisions_this_branch	Count of collisions per track
surface_id	Problem number of surface crossed
surface_normal_cosine	Cosine between surface normal and particle direction
particle_type	Particle type enumeration

D.3.8 SurfaceCrossing

Table D.11 describes the surface crossing event compound data type. Note that the ID of the surface being crossed is a floating-point value rather than an integer value to represent both conventional surfaces (where an integer would be adequate) and macrobody facets (which are represented as floating-point values).

Table D.12: Termination event compound data type fields

Data Field	Description
x	x coordinate of the particle position
y	y coordinate of the particle position
z	z coordinate of the particle position
u	Particle direction cosine relative to $+x$ axis
v	Particle direction cosine relative to $+y$ axis
w	Particle direction cosine relative to $+z$ axis
energy	Particle energy
weight	Particle weight
time	Time at termination
nps	History identifier
node	Number of nodes in track from source to here
material_id	Program material ID of the cell containing event
cell_id	Problem number of the cell containing event
num_collisions_this_branch	Count of collisions per track
termination_type	Number identifying the termination type
particle_type	Particle type enumeration

D.3.9 Termination

Termination events represent the end of a track within a history.

Table D.13: Particle termination type values for different particle types

Number	termination_type	Description
All Particle Types		
1	ALL_PARS_LOSS_ESCAPE	Escape
2	ALL_PARS_LOSS_ENERGY_CUTOFF	Energy cutoff
3	ALL_PARS_LOSS_TIME_CUTOFF	Time cutoff
4	ALL_PARS_LOSS_WEIGHT_WINDOW	Weight-window roulette
5	ALL_PARS_LOSS_CELL_IMPORTANCE	Cell importance
6	ALL_PARS_LOSS_WEIGHT_CUTOFF	Weight cutoff
7	ALL_PARS_LOSS_E_OR_T_IMPORTANCE	Energy/time importance
8	ALL_PARS_LOSS_DXTRAN	Attempted DXTRAN region entry
9	ALL_PARS_LOSS_FORCED_COLLISIONS	Forced collisions
10	ALL_PARS_LOSS_EXP_TRANSFORM	Exponential transform
Neutrons		
11	NEUTRON_LOSS_DOWNSCATTERING	Loss to down scatter
12	NEUTRON_LOSS_CAPTURE	Capture
13	NEUTRON_LOSS_LOSS_TO_N_XN	Loss to (n,xn)
14	NEUTRON_LOSS_LOSS_TO_FISSION	Loss to fission
15	NEUTRON_LOSS_NUCL_INTERACTION	Nuclear interaction
16	NEUTRON_LOSS_PARTICLE_DECAY	Particle decay
17	NEUTRON_LOSS_TABULAR_BOUNDARY	Tabular boundary
18	NEUTRON_LOSS_ELASTIC_SCATTER	Elastic scatter
Photons		
11	PHOTON_LOSS_COMPTON_SCATTER	Compton scatter
12	PHOTON_LOSS_CAPTURE	Capture
13	PHOTON_LOSS_PAIR_PRODUCTION	Pair production
14	PHOTON_LOSS_PHOTONUCLEAR_ABS	Photonuclear absorption
15	PHOTON_LOSS_PHOTOFISSION	Loss to photofission
Electrons		
11	ELECTRON_LOSS_SCATTERING	Scattering loss
12	ELECTRON_LOSS_BREMSSTRAHLUNG	Bremsstrahlung loss
13	ELECTRON_LOSS_P_ANNIHILATION	Positron annihiliation
14	ELECTRON_LOSS_EXCITATION	Excitation event
16	ELECTRON_LOSS_IONIZATION	Ionization event
17	ELECTRON_LOSS_ERG_REJECTION	Energy rejection > emax
Other neutral particles		
11	NEUTRAL_LOSS_NUCL_INTERACTION	Nuclear interaction
12	NEUTRAL_LOSS_ELASTIC_SCATTER	Elastic scatter
13	NEUTRAL_LOSS_PARTICLE_DECAY	Particle decay
Other charged particles		
11	CHARGED_LOSS_COLL_ENERGY_LOSS	Collisional energy loss
13	CHARGED_LOSS_NUCL_INTERACTION	Nuclear interaction
14	CHARGED_LOSS_ELASTIC_SCATTER	Elastic scatter
15	CHARGED_LOSS_PARTICLE_DECAY	Particle decay
16	CHARGED_LOSS_CAPTURE	Capture
17	CHARGED_LOSS_TABULAR_SAMPLING	Tabular sampling
18	CHARGED_LOSS_COSY_APERTURE_HIT	Cosy aperture hit
19	CHARGED_LOSS_COSY_FAULTS	Cosy faults
20	CHARGED_LOSS_ERG_REJECTION	Energy rejection > emax

D.4 Mesh Tally XDMF Output Format

This section describes the file formats of the new MCNP mesh tally `xdmf` output option. Two files are used when a user selects `out=xdmf` on the `fmesh` card.

To produce the **`xdmf`** output, the HDF5-formatted restart file is modified to add a new results group at the root level as `/results`. Underneath that, a `mesh_tally` group is created and the results for each mesh tally are hierarchically organized therein. A variety of Boolean attributes are provided that indicate the presence of additional information on features used with the mesh tally (comments, transformations, reaction multipliers, etc.). In this way, the results can be easily interrogated using standard HDF5 libraries in a variety of programming languages.

In addition, the traditional mesh tally output file (`meshtal` or `...msht`) is written as a version-2 XDMF [324, 325] file (`meshtal.xdmf` or `...msht.xdmf`). The XDMF file can be used with custom applications and/or loaded into applications such as ParaView [326] or VisIt [327], which provide interactive 3-D visualization capabilities.

The remainder of this section is organized as follows: Section D.4.1 describes the file organization for the HDF5 [§D.4.2] file with its subsections describing attributes associated with mesh tally features. XDMF files are described in Section D.4.3. Appendix D.8 provides a Python script to process HDF5 elements into LaTeX `dirtree` listings.

D.4.1 File Organization

Both HDF5 files (by definition) and XDMF files (as XML-formatted files) are hierarchical in nature.

HDF5 files consist of groups (similar to directories) and data sets (multidimensional arrays of homogeneous but arbitrary data). In addition, HDF5 groups and data sets can have attributes assigned, which are relatively low-overhead scalar or array quantities. These three basic components can be arbitrarily arranged into a hierarchy that best suits an application's need(s).

Meanwhile, XDMF files use a standard hierarchy to define mesh geometries, and quantities on the mesh, for the purpose of post processing. Usually, this post-processing is enabled by a visualization application such as ParaView or Visit; however, C++ and Python XDMF libraries exist to quantitatively process XDMF files directly. Regardless, the XDMF format has a specific hierarchy and organization, but can point to data within the HDF5 files located arbitrarily. Thus, the XDMF can be used as a roadmap into the HDF5 file that defines where to retrieve the data of interest. It has been said that the HDF5 files contain the "heavy" data while the XDMF files contain the "light" data.

D.4.2 HDF5

An example HDF5 hierarchy for two mesh tallies (identified as 14 and 24) is given in Fig. D.1, which is generated from the MCNP input given in Listing D.1.

Additional mesh tallies would reside at the same level as the `mesh_tally_14` and `mesh_tally_24` groups. Multiple energy and/or time bins would be given at the respective levels. If the total over all energy and/or time bins are given, they will be labeled as `energy_total` and/or `time_total`, respectively. There is a duplication of datasets and attributes within each energy group to permit additional flexibility in the future such that energy- and/or time-dependent geometry variation is possible.

```
results
└── mesh_tally ..... (group)
    ├── mesh_tally_14 ..... (group)
    │   ├── coordinate_system ..... (attribute)
    │   ├── has_collision_binning ..... (attribute)
    │   ├── has_comment_lines ..... (attribute)
    │   ├── has_dose_response_function ..... (attribute)
    │   ├── has_flagged_cells ..... (attribute)
    │   ├── has_flagged_surfaces ..... (attribute)
    │   ├── has_score_multiplier ..... (attribute)
    │   ├── has_transformation ..... (attribute)
    │   ├── is_isotopic_reaction_rate_tally ..... (attribute)
    │   ├── multiplicative_factor ..... (attribute)
    │   ├── particle_number ..... (attribute)
    │   ├── tally_quantity ..... (attribute)
    │   ├── tally_type ..... (attribute)
    │   ├── grid_energy ..... (dataset)
    │   ├── grid_time ..... (dataset)
    │   ├── grid_x ..... (dataset)
    │   ├── grid_y ..... (dataset)
    │   ├── grid_z ..... (dataset)
    │   ├── mean ..... (dataset)
    │   └── relative_standard_error ..... (dataset)
    └── mesh_tally_24 ..... (group)
        ├── coordinate_system ..... (attribute)
        ├── has_collision_binning ..... (attribute)
        ├── has_comment_lines ..... (attribute)
        ├── has_dose_response_function ..... (attribute)
        ├── has_flagged_cells ..... (attribute)
        ├── has_flagged_surfaces ..... (attribute)
        ├── has_score_multiplier ..... (attribute)
        ├── has_transformation ..... (attribute)
        ├── is_isotopic_reaction_rate_tally ..... (attribute)
        ├── multiplicative_factor ..... (attribute)
        ├── particle_number ..... (attribute)
        ├── tally_quantity ..... (attribute)
        ├── tally_type ..... (attribute)
        ├── basis_axis ..... (dataset)
        ├── basis_cross ..... (dataset)
        ├── basis_vector ..... (dataset)
        ├── coords ..... (dataset)
        ├── grid_energy ..... (dataset)
        ├── grid_r ..... (dataset)
        ├── grid_t ..... (dataset)
        ├── grid_time ..... (dataset)
        ├── grid_z ..... (dataset)
        ├── mean ..... (dataset)
        └── relative_standard_error ..... (dataset)
```

Figure D.1: Mesh Tally HDF5 Hierarchy

Listing D.1: Excerpt from fmesh_xdmf.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
fmesh24:n geom=rzt origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=5 jints=5
                                   kmesh=1 kints=15
                                   out=xdmf
```

In addition to the groups and data sets in Fig. D.1, a variety of attributes are shown. These attributes are often Boolean indicators of additional features that may accompany mesh tallies and are meant as a convenience when parsing the HDF5 results directly. Other attributes are 256-character strings, integers, or floating-point values that provide supplemental information about the associated mesh tally. These attributes are described in the next subsections.

D.4.2.1 Attribute: number_of_normalizing_histories

This integer attribute indicates the number of histories (e.g., the `NPS` card entry) that is used to normalize the results by.

D.4.2.2 Attribute: coordinate_system

The this string attribute gives the coordinate system as `Cartesian` or `cylindrical`, as appropriate. This entry will also indicate which geometry data sets exist at the lowest level. If the mesh tally is Cartesian, at the lowest level the geometry data sets are:

`grid_x`	is a 1-D spatial grid along the x coordinate axis. This is defined based on the `fmesh` card `imesh` and `iints` entries. These coordinates are not modified by an additional `tr` card applied to the corresponding `fmesh` card.
`grid_y`	is a 1-D spatial grid along the y coordinate axis. This is defined based on the `fmesh` card `jmesh` and `jints` entries. These coordinates are not modified by an additional `tr` card applied to the corresponding `fmesh` card.
`grid_z`	is a 1-D spatial grid along the z coordinate axis. This is defined based on the `fmesh` card `kmesh` and `kints` entries. These coordinates are not modified by an additional `tr` card applied to the corresponding `fmesh` card.

and for cylindrical mesh tallies, at the lowest level the geometry data sets are:

`grid_r`	is a 1-D spatial grid along the r coordinate axis. This is defined based on the `fmesh` card `imesh` and `iints` entries. These coordinates are not modified by an additional `tr` card applied to the corresponding `fmesh` card.
`grid_t`	is a 1-D spatial grid along the θ coordinate axis. This is defined based on the `fmesh` card `jmesh` and `jints` entries. These coordinates are not modified by an additional `tr` card applied to the corresponding `fmesh` card.

`grid_z`	is a 1-D spatial grid along the z coordinate axis. This is defined based on the `fmesh` card `kmesh` and `kints` entries. These coordinates are not modified by an additional `tr` card applied to the corresponding `fmesh` card.
`coords`	is a 3-D spatial grid in a Cartesian coordinate system used to define an XDMF 3DSMesh structured curvilinear mesh. This array is modified according to the `fmesh` card's axis and vector (`axs` and `vec`) entries but is not modified by an additional `tr` card applied to the corresponding `fmesh` card.
`basis_axis`	is the axis of the cylinder, which is $(0, 0, 1)$ by default.
`basis_vector`	is the axis defining $\theta = 0$, which is $(1, 0, 0)$ by default.
`basis_cross`	is the cross product of the axis and vector, which is $(0, 1, 0)$ by default.

D.4.2.3 Attribute: has_collision_binning

This Boolean attribute indicates whether the `fmesh` has collision binning using the `inc` option. Examples of using this option are given in Listing D.2.

Listing D.2: Excerpt from fmesh_xdmf_inc.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   inc=1
fmesh24:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   inc=1 3
fmesh34:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   inc=1 infinite
```

These three mesh tallies show different ways of applying the `inc` option. In all cases, `has_collision_binning` is true. When true, two additional attributes are added to the corresponding energy group: `collision_bin_lower` and `collision_bin_upper`.

For `fmesh14`, `collision_bin_lower` is 1 (as entered) and `collision_bin_upper` is -2.

For `fmesh24`, `collision_bin_lower` is 1 (as entered) and `collision_bin_upper` is 3 (as entered).

For `fmesh34`, `collision_bin_lower` is 1 (as entered) and `collision_bin_upper` is -1.

D.4.2.4 Attribute: has_comment_lines

This Boolean attribute indicates whether the `FMESH` card has an associated `FC` card. Examples of using this option are given in Listing D.3.

Listing D.3: Excerpt from fmesh_xdmf_fc.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
fc14 single line comment
fmesh24:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
fc24 double line comment
     second line comment
```

These two mesh tallies show a single and multi-line comment card for mesh tallies 14 and 24, respectively. In all cases, `has_comment_lines` is true. When true, additional string data sets appear, one for each comment line. These data sets are `comment_lines_1`, `comment_lines_2`, etc. and contain a 256-character string. These are written in this way because of a current limitation regarding writing arrays of strings to an HDF5 file.

D.4.2.5 Attribute: has_dose_response_function

This Boolean attribute indicates whether the `FMESH` card has an associated set of `de`/`df` cards applied. An example using this option is given in Listing D.4.

Listing D.4: Excerpt from fmesh_xdmf_dedf.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
de14 log 1e-8 9ilog 1e2
df14 log 1e-8 9ilog 1e2
```

In this case, `has_dose_response_function` is true. Accordingly, another 256-character string attribute, `dose_response_interpolation`, indicates the interpolation method as `loglog` (other options are `linlog`, `loglin`, and `linlin`). Finally, two data sets are added at the lowest group level, `dose_response_function_domain` and `dose_response_function_range`, which correspond to the entries on the `de` and `df` cards, respectively.

D.4.2.6 Attribute: has_flagged_cells

This Boolean attribute indicates whether the `FMESH` card has an associated set of cell-flagging (`cf`) cards applied. An example using this option is given in Listing D.5.

In this case, `has_flagged_cells` is true. Accordingly, another integer attribute, `flagged_cell_count`, indicates the number of flagged cells (in this example: 1). Finally, another 1-D integer data set is added at the lowest group level, `flagged_cells`, which contains the ID numbers for each of the flagged cells.

Listing D.5: Excerpt from fmesh_xdmf_cf.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
cf14 100
```

D.4.2.7 Attribute: has_flagged_surfaces

This Boolean attribute indicates whether the FMESH card has an associated set of surface-flagging (sf) cards applied. An example using this option is given in Listing D.6.

Listing D.6: Excerpt from fmesh_xdmf_sf.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
sf14 10
```

In this case, has_flagged_surfaces is true. Accordingly, another integer attribute, flagged_surface_count, indicates the number of flagged surfaces (in this example: 1). Finally, another 1-D integer data set is added at the lowest group level, flagged_surfaces, which contains the ID numbers for each of the flagged surfaces.

D.4.2.8 Attribute: has_score_multiplier

This Boolean attribute indicates whether the FMESH card has an associated score multiplier applied using a tally multiplier (fm) card. An example using this option is given in Listing D.7.

Listing D.7: Excerpt from fmesh_xdmf_fm.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   factor=1.2
fm14 2.3
fmesh24:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   factor=1.2
fm24 2.3 1 1 2
fmesh34:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   factor=1.2
```

For mesh tally 14, has_score_multiplier is true and two additional attributes are added. First, a floating-point score_multiplier_constant is present and is set to $1.2 \times 2.3 = 2.76$. Next, a 256-character string attribute named score_multiplier_type is present and set to "arbitrary scaler".

For mesh tally 24, has_score_multiplier is true and four additional attributes are added. First, a floating-point score_multiplier_constant is present and is set to 2.3 (i.e., the factor entry on the FMESH card is ignored). Next, a 256-character string attribute named score_multiplier_type is present and set to "reaction". Finally, integer attribute score_multiplier_material is set to 1 and integer attribute score_multiplier_reaction_count is set to 2 (to indicate that two reactions are used from material one). In addition, a new floating-point 1-D data set is added at the lowest level indicating the reaction values from the fm card, which is $[1.0, 2.0]$ in this case.

For mesh tally 34, has_score_multiplier is true and four additional attributes are added. First, a floating-point score_multiplier_constant is present and is again set to 2.3 (i.e., the factor entry on the FMESH card is ignored). Next, a 256-character string attribute named score_multiplier_type is present and set to "reaction". Finally, integer attribute score_multiplier_material is set to 1 and integer attribute score_multiplier_reaction_count is set to 3 (to indicate that two reactions are used from material one as well as the ":" used to sum the reactions). In addition, a new floating-point 1-D data set is added at the lowest level indicating the reaction values from the fm card, which is $[1.0, 1050000003.0, 2.0]$ in this case. The value 1050000003.0 is effectively an enumeration that represents the colon.

Note that other available score_multiplier_types are "1/velocity" and "tracks".

D.4.2.9 Attribute: has_transformation

This Boolean attribute indicates whether the FMESH card has an associated geometry transformation applied using a TR card. An example using this option is given in Listing D.8.

Listing D.8: Excerpt from fmesh_xdmf_tr.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   tr=6
tr6 10 20 30 0 0 1 0 1 0 1 0 0
```

In this case, has_transformation is true and an additional 1-D floating-point data set is added:

transformation_matrix
: which contains the MCNP transformation matrix entries.

The entries in this "matrix" correspond to the entries on the TR card. In this case, transformation_matrix is equal to $[-6, -3, -2, -1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 2, 3]$. The rotation matrix is given in entries 5–13. The translation components are given in the final three entries. The other entries (2, 3, 4, 14) are intended for calculations internal to the MCNP code and are not described further.

D.4.2.10 Attribute: is_isotopic_reaction_rate_tally

This Boolean attribute indicates whether the FMESH card has an associated score multiplier applied using a tally multiplier (fm) card modified to act as an isotopic reaction rate tally. An example using this option is given in Listing D.9.

Listing D.9: Excerpt from fmesh_xdmf_fmrxnrate.mcnp.inp.txt

```
fmesh24:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   factor=1.2
```

In this case, is_isotopic_reaction_rate_tally is true. Otherwise, the additional attributes and data sets are consistent with mesh tally 24 in Section D.4.2.8.

D.4.2.11 Attribute: multiplicative_factor

This floating-point attribute indicates the multiplicative factor applied using the factor entry on the FMESH card. An example using this option is given in Listing D.10.

Listing D.10: Excerpt from fmesh_xdmf_factor.mcnp.inp.txt

```
fmesh14:n geom=xyz origin=-3 -3 -3 imesh=3 iints=3
                                   jmesh=4 jints=4
                                   kmesh=5 kints=5
                                   out=xdmf
                                   factor=2.3
```

In this case, multiplicative_factor is equal to 2.3. By default, this is equal to 1.0.

D.4.2.12 Attribute: particle_number

This integer attribute indicates the particle type that the mesh tally applies to. It follows MCNP conventions, so a neutron mesh tally corresponds to particle_number equal to 1, photon mesh tallies are identified as 2, etc.

D.4.2.13 Attribute: tally_quantity

This 256-character string attribute indicates the quantity being tallied. This is either:

track_length_particle_flux
: the default,

`track_length_energy_flux`	as specified by prefixing the `FMESH` card with an asterisk, or
`partial current...`	in a particular direction (implicitly specified by using a type-1 mesh tally).

D.4.2.14 Attribute: tally_type

This 256-character string attribute indicates the type of mesh tally. This is either `flux` (the default), `source` (as specified using the type parameter on the `FMESH` card), or `current` (implicitly specified by using a type-1 mesh tally).

D.4.3 XDMF

The XDMF standard is documented elsewhere [325] and it is rare that someone works with the file contents directly. Therefore, minimal details are given here regarding the detailed structure of the file. However, the general organization is described along with suggested methods for working with the file using ParaView.

The XDMF file used to interrogate mesh tallies is formatted according to the version 2 API (cf. version 3) and is an ASCII XML-formatted file. A temporal grid collection is used to represent time steps within time-dependent mesh tallies. General grid collections are used to group energy bins within time steps. When applicable, totals over all bins for energy and/or time are given by name (i.e., `_total`) otherwise time- and energy- bins are zero indexed (including unbinned mesh tallies). Mesh tally voxel volumes are referred to generically as a `volume` dataset. Individual mesh tally data sets for the tally and relative standard deviation values are prefixed by the mesh tally ID, e.g., `14_`, `44_`, `104_`, etc.

These naming approaches can lead to data sets that are not represented over all mesh voxels if multiple mesh tallies are displayed at once and/or if multiple energy bins are used. However, different mesh tallies can have different energy and/or time binning structures, so this method attempts to isolate data that is unique to each energy bin and time step at the cost of a suboptimal interactive manipulation and display experience.

Methods to load and work with an XDMF file follow.

D.4.3.1 Loading the XDMF File for Visualization

To load the mesh tally for visualization from Listing D.1 with ParaView, select File→Open... and select the appropriate XDMF file. Because the XDMF file uses format version 2, a dialog will likely appear to select a reader for the file, where "XDMF Reader" is the correct choice, as shown in Fig. D.2.

D.4.3.2 Navigating to a Dataset of Interest

Once the data is loaded into the ParaView pipeline and applied to the render view (by clicking the Apply button), one might see the two mesh tallies from Listing D.1 displayed as shown in Fig. D.3. Each mesh tally is a separate block that can be individually shown/hidden using either the Blocks or Hierarchy checkbox lists. In this case. the coloring is by volume so all voxels are shaded correctly. However, as noted previously, if shading by `14_tally`, `24_tally`, `14_relative_standard_deviation`, or `24_relative_standard_deviation`, then only the respective tally will be colored as shown in Fig. D.4, which is shaded by the `14_tally` field (i.e., the tally values for mesh tally 14).

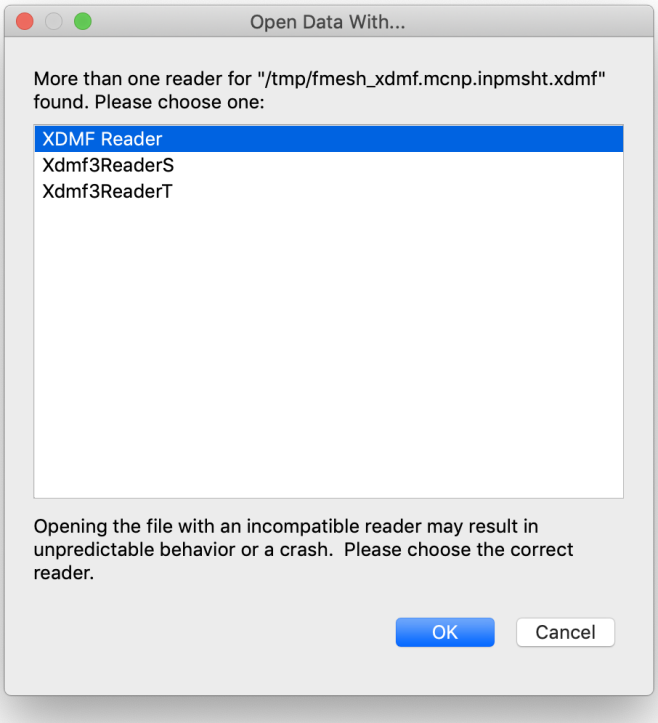


Figure D.2: Open Data With... Dialog Example

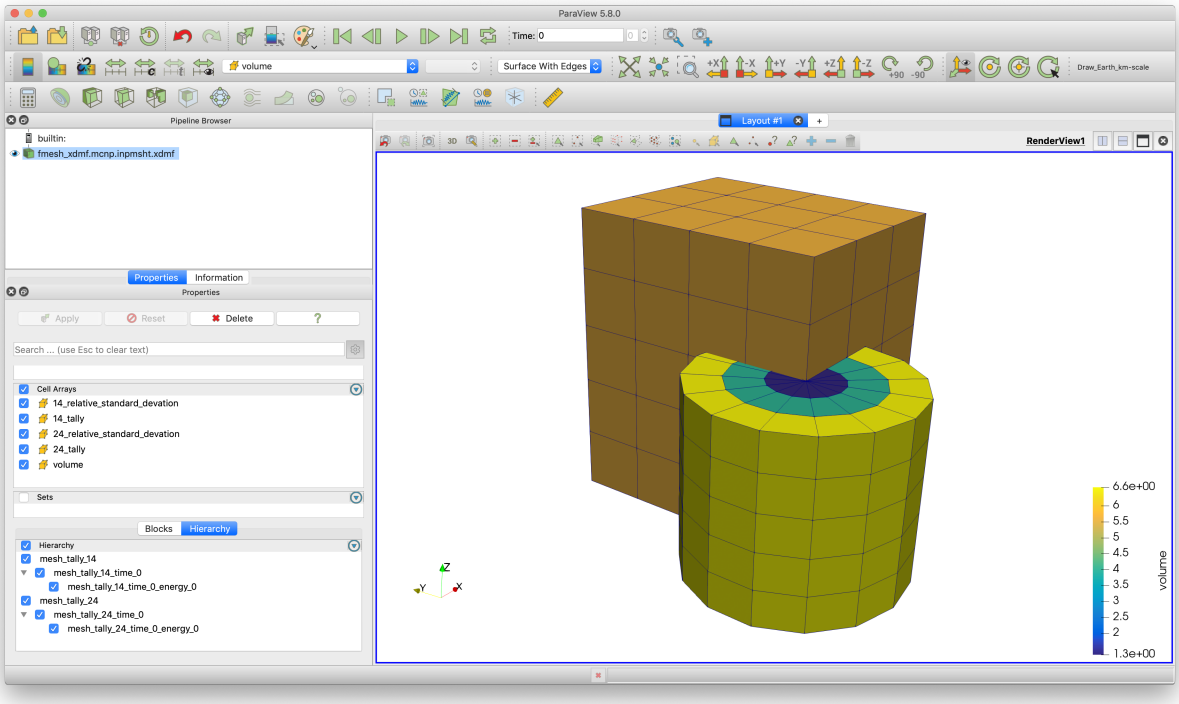


Figure D.3: Voxelwise Volume Example

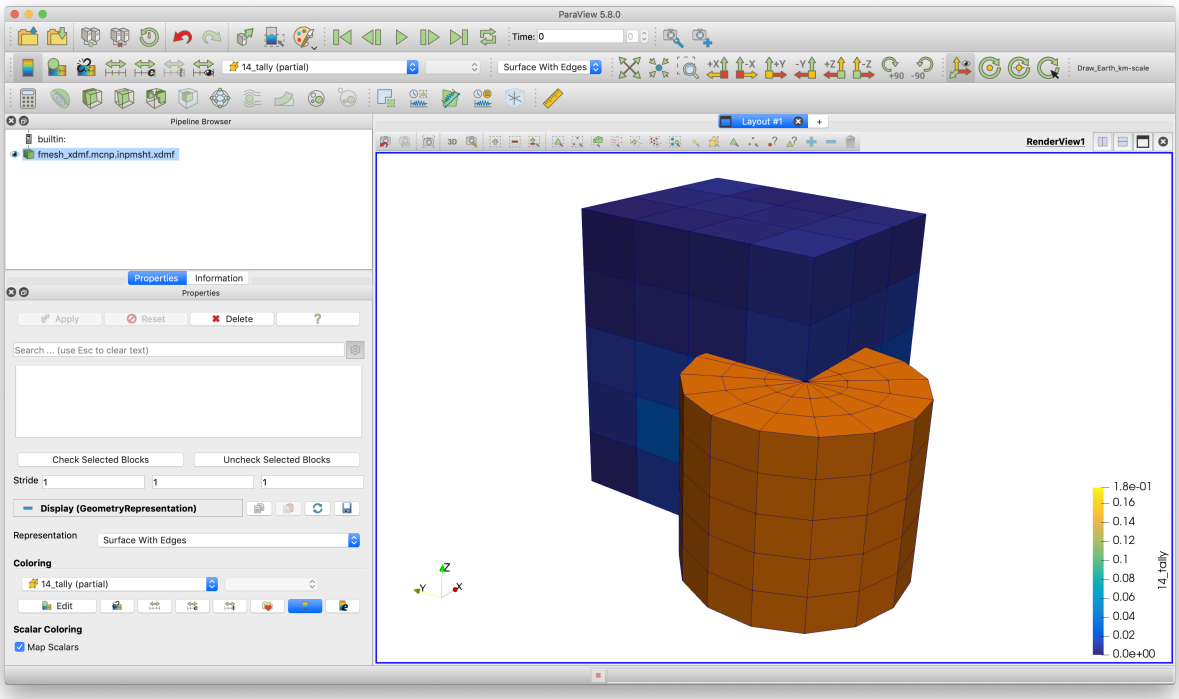


Figure D.4: Voxelwise `14_tally` Example

```
results
└── mesh_tally ..... (group)
    └── mesh_tally_14 ..... (group)
        ├── coordinate_system ..... (attribute)
        ├── has_collision_binning ..... (attribute)
        ├── has_comment_lines ..... (attribute)
        ├── has_dose_response_function ..... (attribute)
        ├── has_flagged_cells ..... (attribute)
        ├── has_flagged_surfaces ..... (attribute)
        ├── has_score_multiplier ..... (attribute)
        ├── has_transformation ..... (attribute)
        ├── is_isotopic_reaction_rate_tally ..... (attribute)
        ├── multiplicative_factor ..... (attribute)
        ├── particle_number ..... (attribute)
        ├── tally_quantity ..... (attribute)
        ├── tally_type ..... (attribute)
        ├── grid_energy ..... (dataset)
        ├── grid_time ..... (dataset)
        ├── grid_x ..... (dataset)
        ├── grid_y ..... (dataset)
        ├── grid_z ..... (dataset)
        ├── mean ..... (dataset)
        └── relative_standard_error ..... (dataset)
```

Figure D.5: Truncated Time & Energy-dependent Mesh Tally HDF5 File

Note that, as shown in Fig. D.3, the cylindrical mesh tally is shown with facets despite being a curvilinear structured grid. No visualization application is known at this time that can faithfully display the curvilinear nature of the mesh.

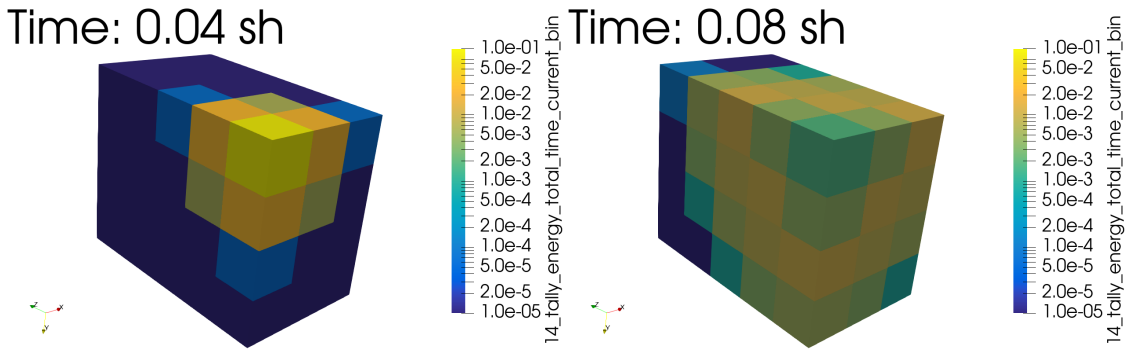
D.4.3.3 Saving Animation Frames from a Time-dependent Mesh Tally

An example time- and energy-dependent mesh tally is given in Listing D.11, which produces the (truncated)

Listing D.11: Excerpt from fmesh_xdmf_tdep_edep.mcnp.inp.txt

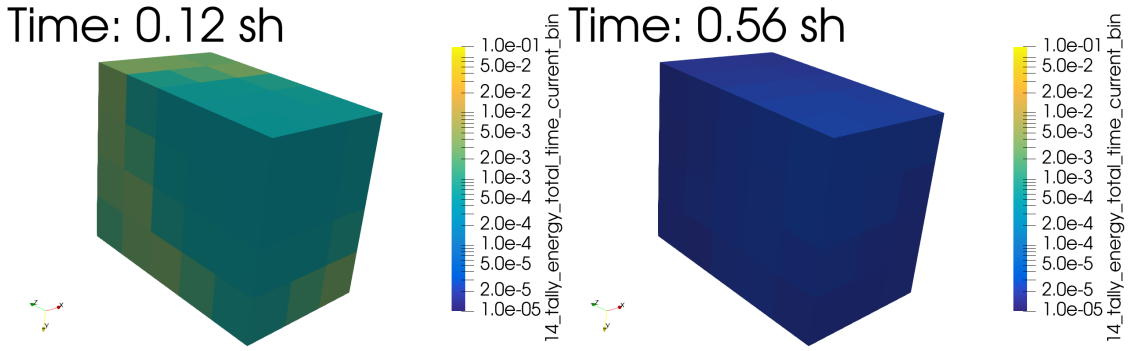
```
fmesh14:n geom=xyz origin= 0 0 0   imesh=3   iints=3
                                   jmesh=4   jints=4
                                   kmesh=5   kints=5
                                   tmesh=0 2 tints=1 50
                                   emesh=20  eints=2
                                   out=xdmf
```

HDF5 file shown in Fig. D.5. To export the time steps for the animation, select a data set to color by that corresponds to the current time step (e.g., `14_tally_energy_total_time_current_bin`). Having done so, one can then select File→Save Animation. . . , choose a prefix for the saved file(s), click OK, configure the frame attributes (size, file type, etc.), and click OK again. The export will the commence. Example frames from the resulting animation are shown in Fig. D.6.



(a) Timestep 1

(b) Timestep 2



(c) Timestep 3

(d) Timestep 14

Figure D.6: Animation Frames

D.5 Fission Matrix Format

The MCNP fission matrix is added to the runtape whenever the KOPTS option FMAT is set to yes. The contents of the /results/fission_matrix group in the restart file [§D.2] are:

- fission_matrix (group)
  - data (dataset)
  - delta_xyz (dataset)
  - indices (dataset)
  - indptr (dataset)
  - n (dataset)
  - n_xyz (dataset)
  - origin (dataset)

The variables indptr, indices, and data represent a 0-indexed compressed-sparse-row (CSR) matrix, which can be readily loaded by many sparse linear algebra packages. As an example, the Python [347] code in Listing D.12 can be used to load a fission matrix into the SciPy [348] sparse capability. The eigenvalues of the matrix can be computed using scipy.sparse.linalg.eigs.

Listing D.12: Fission Matrix HDF5 Reader

```python
#!/usr/bin/env python3

import h5py
import scipy.sparse as sparse
import scipy.sparse.linalg as sla

SUPPORTED_RUNTAPE = [1, 0, 0]


def extract_fmat(runtape):
    """Returns the last saved fission matrix as a scipy.sparse.csr_matrix"""
    with h5py.File(runtape, "r") as handle:
        # Check runtape version
        version_file = handle["config_control"].attrs["version_file"]
        if any(SUPPORTED_RUNTAPE != version_file):
            print("Possibly incompatible runtape detected.")

        fmat = handle["results/fission_matrix"]

        n = fmat["n"][()]
        indices = fmat["indices"][:]
        indptr = fmat["indptr"][:]
        data = fmat["data"][:]

        n_xyz = fmat["n_xyz"][:]
        delta_xyz = fmat["delta_xyz"][:]
        origin = fmat["origin"][:]

    return sparse.csr_matrix((data, indices, indptr), shape=(n, n)), n_xyz, delta_xyz, origin
```

The remaining variables in the group are used for converting the eigenvectors into a representation that has meaning in 3D. The eigenvectors, as computed from the fission matrix, are unrolled in a column-major way with x changing first. As NumPy [349] is row major by default, the easiest way to generate a 3D array indexed by $[x, y, z]$ is to use the approach in Listing D.13. The origin variable is the bottom-left-rear coordinate of the mesh, and delta_xyz is the spacing of the mesh in x, y, and z, in units of centimeters.

Listing D.13: Eigenvector to 3D Mesh

```
mat, n_xyz, delta_xyz, origin = extract_fmat("runtape.h5")

eigenvalues, eigenvectors = sla.eigs(mat)

# Reshape first eigenvector into 3D object
eigenvector = eigenvectors[:, 0].reshape(n_xyz[2], n_xyz[1], n_xyz[0]).transpose()
```

Finally, the eigenfunctions can be scaled by arbitrary coefficients. While it is common to interpret the fundamental eigenvector as everywhere-positive, some solvers may return an eigenvector that is everywhere-negative. It is safe to negate this eigenvector to make it positive. For more discussion on the interpretation of the results, see [350, 351].

D.6 Unstructured Mesh File Format: HDF5

D.6.1 Introduction

The Hierarchical Data Format version 5 (HDF5) is an open source file format [340]. An MCNP UM model geometry is derived from parts that are positioned relative to one another to form an assembly and an HDF5 UM model is designed by grouping the assembly into pseudo-cells (see §8.2 for a pseudo-cell definition). The HDF5 data model is chosen to store a UM geometry and elemental edit outputs since (i) an HDF5 file is designed to manage large complex data collections, (ii) an HDF5 file is portable among different computing platforms, and (iii) an HDF5 file is easy to view, edit, and analyze using public available software tools or Python scripts.

An HDF5 file is a container for an organized collection of objects where each object must have a unique identity within an HDF5 file and can be accessed only by its name within the hierarchy of the file. HDF5 group, dataset, and attribute objects are used to manage MCNP model data. In each HDF5 file, the objects are organized similar to a directory and file structure. A group and dataset are respectively comparable to a directory and file where a group may contain groups and datasets. An attribute is a named data value associated with a group or dataset.

Each HDF5 UM file contains up to three groups at the root level: `config_control`, `problem_info`, and `unstructured_mesh`.

The presence of the `config_control` group is required in order to process the HDF5 UM as an input mesh model. Both the `kind_file`, defined as "um_model", and `version_file`, defined as "[1,0,2]" for the MCNP code version 6.3.0 and 6.3.1, attributes of the `config_control` group are required to process the HDF5 UM file as input. When the `hdf5file` keyword on the `EMBED` data card is specified, the resulting HDF5 UM file is written with a `config_control` group with all appropriate attributes such that this file can be used as input in a subsequent calculation. See §D.2.2 for more information on the contents of this group that the MCNP code writes into the HDF5 UM file. If other codes are used to create HDF5 UM input files for MCNP UM simulations, then the `kind_file` and `version_file` attributes of the `config_control` group must be written into the mesh input files.

The `problem_info` group is generated by the MCNP code within the requested file specified using the `hdf5file` keyword on the `EMBED` data card. See §D.2.5 for more information on the contents of this group. The `problem_info` group is optional when other codes are used to generate the HDF5 UM input files.

The `unstructured_mesh` group, which is required, contains six attributes, one dataset, and one or more cell groups where the number of cell groups depends on the model. The HDF5 objects in the `unstructured_mesh` group are summarized in Table D.14. The `number_of_histories` attribute is not required in an HDF5 input model. The `model_description` attribute is a string describing the mesh model and its length is 128 UTF-8 characters. The `cell_name` dataset is a 1D string array whose size is equal to the `total_cells` attribute. Each entry value in the `cell_name` array must be distinct and its length is 128 ASCII characters. The number of *<unique_cell_name>* groups must be equal to the size of the `cell_name` array and each *<unique_cell_name>* group must be an entry of the `cell_name` array. The mesh model is built from part data where some parts are used multiple times or divided into multiple cells. The attributes related to total cell/part data identify how the parts are used in the mesh model.

- For a model where parts are divided into multiple cells and each part is used only once, the total cells is greater than the total parts but the total cell elements is equal to the total part elements.
- For a model where a whole part is used in multiple cells, the total cells is greater than the total parts and the total cell elements is greater than the total part elements.

Table D.14: HDF5 Attributes/Dataset/Groups in `unstructured_mesh` Group.

Name	HDF5 Object	Data type	Description
`model_description`	attribute	string	model description
`total_cells`	attribute	integer	total cells in a model
`total_elements`	attribute	integer	total elements of all cells
`total_parts`	attribute	integer	total parts in a model
`total_part_elements`	attribute	integer	total elements of all parts
`number_of_histories`	attribute	integer	number of particle histories run
`cell_name`	dataset	1D string array	unique name associated to cell group
<unique_cell_name>	group	attributes & groups	group containing cell geometries and edit outputs

Table D.15: HDF5 Attributes/Groups in *<unique_cell_name>* Group.

Path	HDF5 Object	Data Type
/`unstructured_mesh`/*<unique_cell_name>*/`part_name`	attribute	128 UTF-8 characters
/`unstructured_mesh`/*<unique_cell_name>*/`cell_id`	dataset	integer
/`unstructured_mesh`/*<unique_cell_name>*/`number_of_nodes`	attribute	integer
/`unstructured_mesh`/*<unique_cell_name>*/`number_of_elements`	attribute	1D integer array
/`unstructured_mesh`/*<unique_cell_name>*/`mesh`	group	datasets
/`unstructured_mesh`/*<unique_cell_name>*/`volume`	group	attribute & dataset
/`unstructured_mesh`/*<unique_cell_name>*/`material`	group	datasets
/`unstructured_mesh`/*<unique_cell_name>*/`source`	group	attribute & dataset
/`unstructured_mesh`/*<unique_cell_name>*/`edit`	group	groups

- For a model where each whole part is used only once, the total cells is equal to the total parts and the total cell elements is equal to the total part elements.

D.6.2 Cell Group

Each /`unstructured_mesh`/*<unique_cell_name>* group contains three attributes and four groups listed in Table D.15 The HDF5 input model requires only the `part_name`, `number_of_nodes`, `number_of_elements` attributes and the `mesh`, `material`, and `source` groups. The following describes the HDF5 attributes in each *<unique_cell_name>* group.

- The `part_name` attribute is the part name where the mesh data are extracted. The length of the `part_name` attribute is 128 UTF-8 characters.
- The `cell_id` dataset is a length-1 integer array that provides the MCNP (pseudo)cell identification number specified by the user in the MCNP input file. This field is present in MCNP UM HDF5 output files and is ignored in MCNP UM HDF5 input files.
- The `number_of_nodes` attribute is the number of nodes in a cell. The `number_of_nodes` must be greater than four.
- The `number_of_elements` attribute is the number of elements for each element type. Its size is six. Each entry in the `number_of_elements` array must be non-negative and the total number of elements must be positive. Since only linear pentahedron and hexahedron elements or quadratic pentahedron and hexahedron elements can be mixed in a part, the UM library will throw a fatal error if a cell contains other mixed element types. An entry of the `number_of_elements` array represents the number of elements based on the element type:

Table D.16: HDF5 Datasets in `mesh` Group.

Path	Data Type	
/unstructured_mesh/<*unique_cell_name*>/mesh/node_id	1D integer array	optional
/unstructured_mesh/<*unique_cell_name*>/mesh/node_point	2D real array	required
/unstructured_mesh/<*unique_cell_name*>/mesh/element_id	1D integer array	optional
/unstructured_mesh/<*unique_cell_name*>/mesh/XDMF_connectivity	1D integer array	required

  - The first and fourth entries are respectively the numbers of linear and quadratic tetrahedron elements.
  - The second and fifth entries are respectively the numbers of linear and quadratic pentahedron elements.
  - The third and sixth entries are respectively the numbers of linear and quadratic hexahedron elements.

D.6.2.1 Mesh Group

The /unstructured_mesh/<*unique_cell_name*>/mesh group contains four datasets describing the geometry and connectivity. The dataset names and dimensions are listed in Table D.16.

- The `node_id` dataset contains the node identifiers used to form the elements. It is a 1D integer array whose size is equal to the `number_of_nodes` attribute. All entries of the `node_id` array must be unique and positive. The `node_id` dataset is an optional data object. It is only required if the `node_id` array does not consecutively start from one.
- The `node_point` dataset contain the (x, y, z) points describing the node locations. This 2D real array dimension is $(n, 3)$ where n is the `number_of_nodes` attribute. The first and second indices of the `node_point` array represent the node identifiers and (x, y, z) locations, respectively. The dimensions of x, y, z are in centimeters.
- The `element_id` dataset contains the element identifiers. This 1D integer array is only required if the the `element_id` array does not consecutively start from one. If this array is required, its size must be equal to the summation of the `number_of_elements` array and each entry value in the `element_id` dataset must be distinct and positive. For a cell with mixed element types, the `element_id` array stores the element identifiers based on the element types ordering in the `number_of_elements` attribute.
- The `XDMF_connectivity` dataset contains the element connectivity. The mixed UM XDMF model is used to store this dataset. XDMF stands for eXtensible Data Model and Format and more information on XDMF can be found at [325]. The `XDMF_connectivity` dataset is a 1D integer array whose size depends on the `number_of_elements` attribute. The mixed UM XDMF model is made of an element type and a list of node indices describing an element; note that an element is called a cell in the XDMF literature. The `XDMF_connectivity` is then organized as follows: XDMF element type, list of node indices in an element, XDMF element type, list of node indices in an element, ..., and hence its size is equal to the summation of number of element type times number of nodes plus one. For example, if there is 10 linear pentahedra and 1000 linear hexahedra in a cell, then the length of `XDMF_connectivity` is equal to 9070. For a cell with mixed element type, the `XDMF_connectivity` array stores the element data based on the element types ordering in the `number_of_elements` attribute. This leads to the limitation that the XDMF element type must be prefixed for each element in a cell even if a cell has only single element type. The node indices used to form an element start from zero. The XDMF element type is an integer used to represent a physical element type:
  - 6 and 38 are respectively for a linear and quadratic tetrahedron,

Table D.17: HDF5 Attribute and Dataset in `volume` Group.

Path	HDF5 Object	Data Type
/unstructured_mesh/<*unique_cell_name*>/volume/cell_volume	attribute	real
/unstructured_mesh/<*unique_cell_name*>/volume/element_volume	dataset	1D real array

Table D.18: HDF5 Datasets in `material` Group.

Path	HDF5 Object	Data Type
/unstructured_mesh/<*unique_cell_name*>/material/material_id	dataset	1D integer array
/unstructured_mesh/<*unique_cell_name*>/material/mass_density	dataset	1D real array

  - 8 and 40 are respectively for a linear and quadratic pentahedron,
  - 9 and 48 are respectively for a linear and quadratic hexahedron.

D.6.2.2 Volume Group

The unstructured_mesh/<*unique_cell_name*>/volume group contains one attribute and one dataset describing the element volumes in units of cm^3. The volume object data are listed in Table D.17.

- The cell_volume attribute is the volume of the cell. It is the summation of the element_volume dataset.
- The element_volume attribute is the element volume. This 1D array size is equal to the summation of the number_of_elements attribute and it stores the value based on the element type ordering in the number_of_elements attribute. Each entry of this array must be positive.

D.6.2.3 Material Group

The /unstructured_mesh/<*unique_cell_name*>/material group contains two datasets describing the materials. The material object data are listed in Table D.18. The material properties present in an HDF5 mesh input file are not used for MCNP calculations. The MCNP code uses the material properties defined in the MCNP input file.

- The material_id dataset is the material identifier. This 1D array has only one entry and its value must be a positive integer number.
- The mass_density dataset is the material mass density in units of g/cm^3. This 1D array has only one entry and its value must be a non-negative real number.

D.6.2.4 Source Group

The /unstructured_mesh/<*unique_cell_name*>/source group contains one attribute and one dataset describing the source elements. The source group is optional and the source object data are listed in Table D.19. The source group is present only for the UM model that has source elements. If the source group is present in the cell, then the number_of_source_elements attribute and the source_element_value dataset are required and each cell in the model must have the source group.

Table D.19: HDF5 Attribute and Dataset in `source` Group.

/`unstructured_mesh`/<*`unique_cell_name`*>/`source`/ path	HDF5 Object	Data Type	
`number_of_source_elements`	attribute	integer	required
`source_element_value`	dataset	1D integer array	required
`source_element_id`	dataset	1D integer array	optional

Table D.20: HDF5 `value` and `error` Datasets in `edit` Group. *`X`* is an edit number in an MCNP input file, *`Y`* is a particle number (1 for neutron, 2 for photon, 3 for electron, and so on), *`Z`* is an energy bin index number starting from one, and *`W`* is a time bin index number starting from one. If several particles are requested in the same edit number, then all particle numbers will be included in *`X`* where the particle numbers are separated by "_". For example, `edit_`*`14`*`_particle_`*`1_2`* is a field name for neutron and photon of the edit number 14.

/`unstructured_mesh`/<*`unique_cell_name`*>/`edit` path to `value` and `error` Datasets	Data Type
/`edit_`*`X`*`_particle_`*`Y`*/`value` (`error`)	1D real array
/`edit_`*`X`*`_particle_`*`Y`*`_energy_bin_`*`Z`*/`value` (`error`)	1D real array
/`edit_`*`X`*`_particle_`*`Y`*`_time_bin_`*`W`*/`value` (`error`)	1D real array
/`edit_`*`X`*`_particle_`*`Y`*`_energy_bin_`*`Z`*`_time_bin_`*`W`*/`value` (`error`)	1D real array
/`edit_`*`X`*`_particle_`*`Y`*`_energy_total_time_bin_`*`W`*/`value` (`error`)	1D real array
/`edit_`*`X`*`_particle_`*`Y`*`_energy_bin_`*`Z`*`_time_total`/`value` (`error`)	1D real array
/`edit_`*`X`*`_particle_`*`Y`*`_energy_total_time_total`/`value` (`error`)	1D real array

- The `number_of_source_elements` attribute is the number of source elements. It must be a non-negative integer. It is equal to zero in the cell if the UM model has at least one source element but there is no source element in this cell.
- The `source_element_value` dataset is the source element value. This 1D array size is equal to the summation of the `number_of_elements` attribute and it stores the value based on the element type ordering in the `number_of_elements` attribute. Each entry is equal to one for a source element and zero for an non-source element. The number of non-zero entries is equal to the `number_of_source_elements` attribute. A `source_element_value` dataset is written for all cells in the HDF5 file for the UM model with source elements, whether a cell has elements that contain source or not, to simplify post-processing and/or visualization operations.
- The `source_element_id` dataset is the source element identifier. This 1D integer array is only required if the `source_element_id` array does not consecutively start from one. If this array is required, its size must be equal to the `number_of_source_elements` attribute and each entry in the `source_element_id` dataset must be unique and positive.

D.6.2.5 Edit Group

The /`unstructured_mesh`/<*`unique_cell_name`*>/`edit` group contains several groups whose names are shown Table D.20. The `value` and `error` datasets are 1D real arrays whose size is equal to the summation of the `number_of_elements` attribute. The `value` and `error` array store the data based on the element types ordering in the `number_of_elements` attribute. In addition, the subgroups in the `edit` group may contain the following attributes.

- `comment`
- `edit_energy_multiplier` and `edit_time_multiplier`

- `bin_energy_upper` and `bin_energy_multiplier`. These attributes are available if the energy domain is broken into bins.
- `bin_time_upper` and `bin_time_multiplier`. These attributes are available if the time domain is broken into bins.

D.6.3 Datatype and Array Dimension in HDF5 EEOUT Files

The binary HDF5 files are theoretically portable among platforms and easy to extract data from HDF5 files. The following must be considered when dealing with HDF5 EEOUT files.

- The `mesh_description` attribute and `cell_name` dataset must be fixed-length strings since the HDF5 library used to by the UM library can only handle fixed-length strings.
- The `node_point` dataset is a 2D array whose dimension is described as the row-major order in this manual. If Fortran code is used to write (or read) an HDF5 EEOUT file, then its dimension must be reversed to the column-major order.

D.7 Unstructured Mesh File Format: Legacy EEOUT

D.7.1 Introduction

Deprecation Notice **DEP-53294**

The EEOUT file format is deprecated. As an output file, it has been supplanted by an HDF5-formatted file [§D.6]. For the purpose of restarting a calculation, it has been supplanted by a location within the HDF5-formatted MCNP runtape (see the `hdf5file` argument on the `EMBED` card).

This section provides a brief description of version 6 of MCNP6's elemental edit output file (`EEOUT`) generated by the Revised Extended Grid Library (REGL).

D.7.2 EEOUT File

This section describes the elemental edit output file from MCNP6, otherwise known as the `EEOUT` file. This file contains a variety of information besides the edit results that have been calculated on a given mesh. Mainly, the information in this file consists of the results, known as edits, and a generic description of the mesh. What is meant by generic is that the mesh description in the file bears little resemblance to that created by the tool which generated the mesh. Therefore, many specific formats may eventually be read by the mesh library, but only one output format will be supported. In that regard, the format for the `EEOUT` file has been developed to accommodate what is thought to be all of the relevant data not only for post-processing but also for problem restart. Some of the data present in the file may be in a form that is only relevant to the REGL.

The following description is for version 6 of the `EEOUT` file and is similar to previous versions. An example `EEOUT` file follows this discussion and is a composite of two different problems. This composite file was done to make it easier to illustrate some data sets and to keep the example short. Some lines in this example are color coded for easier identification.

Note that this implementation of the unstructured mesh library is with Fortran and that both ASCII and binary versions of the `EEOUT` file are possible. Fortran inserts beginning and ending record markers around each binary file record; this should be taken into consideration when using a non-Fortran programming language to construct a routine that reads this file. However, if your distribution comes with the source code, consider using REGL to read the `EEOUT` file. See the UM utilities for examples of how to work with REGL.

D.7.3 Self-Describing File

The `EEOUT` file was designed to be a self-describing file with the goal of allowing easy access to and identification of the file's data for those developers who choose not to link with the mesh library and use its routines. The meta data and keyword-value pairs (KWV-pairs), both discussed below, permit the developer a number of options in terms of parsing through the file to extract relevant information.

The data in the file is grouped into data-sets with at least two data-set segments and at most three data-set segments per data-set. The three data-set segments are

- identification
- title line

- data

Except for the first line of the file, each data-set adheres to this convention. All data-sets must contain the data-set identification which is nothing more than the meta data that describes the segments following it. There is no justification for the meta data appearing in the file by itself, so either one or both of the other data-set segments follow it.

The `EEOUT` file also uses KWV-pairs. These appear anywhere there is character data. That is, these pairs may appear in either the title line segment or the data segment. The keyword-value pair is a convenient way to group a short description with either a numeric or alphanumeric value. Each pair consists of one or more keywords to the left of a colon (:) and a value to the right. When multiple KWV-pairs appear on a line, they are separated by a semicolon (;).

D.7.3.1 Identification Segment

This single meta data line always consists of six 8-byte integers, 1) through 6). Their significance is described next and accommodates some flexibility in use.

1. Number of characters in the title line (A value of 0 indicates no title line segment)
2. Number of records in the data-set after the title record (A value of 0 indicates no data segment)
3. Data type that appears in the data segment. No mixed types are permitted.

   0 - no data lines follow (redundant when 2) = 0)
   1 - character data
   2 - integer data
   3 - real data
4. Size in bytes of each 3) data item. If 2) = 0, then 4)'s value is meaningless.
5. Number of items in each data record. If 2) = 0, then 5)'s value is meaningless.
6. Parse length of each record. This is the number of entries formatted for each ASCII data line. If 2) = 0, then 6)'s value is meaningless.

D.7.3.2 Title Line Segment

The title line data segment is optional. However, it must be present if there is no data segment. In this sense, the data is contained within the title line as one or more KWV-pairs. This line is always interpreted as character data so that item 1) in the meta data line is a positive integer. Generally, this title line describes the data that follows it.

D.7.3.3 Data Segment

The data segment is optional. However, it must be present if there is no title line segment. This data may be character, integer, or real as indicated by item 3) in the meta data. Most of the data segments in the `EEOUT` file are either integer or real. In some instances where there is character data in this segment it may be something as simple as a list of material names or it may be KWV-pairs.

D.7.4 The EEOUT File Description

The following sections discuss the various data-sets that appear in the EEOUT file in the order that they appear. As mentioned above, an example file follows, Section D.7.5. In the example file all identification segments (meta data) appear in red and all title line segments appear in blue.

D.7.4.1 First Line

The first line of the EEOUT file is a description line that contains exactly 12 characters. If the file is the ASCII version, the 12 characters, ignoring the double quotes, are "MCNP EDITS A", where the "A" stands for ASCII. If the file is the binary version, the 12 characters, ignoring the double quotes, are "MCNP EDITS B", where the "B" stands for binary. Note that there is no meta data line preceding this line.

In the binary version of this file there will be Fortran inserted record markers before and after these 12 characters. If the developer is using a programming language other than Fortran to read the file, the length of the markers can be deduced from the total length of this line. With this information, subsequent records in the file can be read and the markers ignored to obtain the record information.

D.7.4.2 First Data Set

The first data set in the file does not contain a title line segment, but contains two KWV-pairs in two records. From the first pair, the value provides the mesh source. Since the Abaqus/CAE mesh input file is currently the only mesh input file that the library reads, the value is "ABAQUS". From the second KWV-pair, the value provides the version number of the EEOUT file.

D.7.4.3 Calling Code Labels

The second data-set consists of KWV-pairs containing descriptive information from the code that calls the mesh library. In the case of MCNP6, there are 8 labels that it passes to form the KWV-pairs in the output. Note that the calling code has inserted a special character, "|", to signify the end of meaningful characters on a line. The first one has keywords "Prob ID" and is the problem description supplied by the user in the MCNP6 run. The second and third KWV-pairs have the keywords "Calling Code" and "Code Version" which in this case confirms that the code using the library is MCNP6 and its associated build version. The fourth KWV-pair provides the Date & Time that the EEOUT file was generated. The fifth through eighth KWV-pairs supply four files associated with the MCNP6 calculation that generated the EEOUT file. These four files are

- the MCNP6 inp file
- the MCNP6 outp file
- the MCNP6 runtpe file
- the Abaqus inp file that contains the mesh description

Other associated files may by added to this data-set in the future.

NOTE: When the third entry on MCNP6's PRDMP card is set to -1, this data-set is not present.

D.7.4.4 Integer Parameters

The third data-set contains 12 KWV-pairs where the value part of the pair is an integer. The second through tenth pairs are parameters associated with the mesh geometry and their names are self-explanatory. They are the numbers of nodes, materials, instances, first-order tetrahedra, first-order pentahedra, first-order hexahedra, second-order tetrahedra, second-order pentahedra, and second-order hexahedra.

The first KWV-pair is the number of particles in the calculation.

The eleventh KWV-pair is the number of histories from the Monte Carlo calculation upon which the edit results are based. This is the number that is used in normalizing the edits.

The twelfth KWV-pair provides the number of edits or `EMBEE` cards that were specified in the input.

D.7.4.5 Real Parameters

The fourth data-set contains 2 KWV-value pairs where the value parts of the pairs are real numbers. In the first pair, the value is the length conversion for all of the spatial coordinates from the input mesh file and represents the multiplier needed to convert from the units of the original mesh model to centimeters (in this case the units required by MCNP6). This value has been applied to all coordinates appearing in the `EEOUT` file and consequently is reflected in all of the results. In the second pair, the value is the normalization factor that has been applied to all results in the file. This factor is used to un-normalize the results for continue runs.

D.7.4.6 Particle List

The fifth data-set contains a list of the particle numbers from the calling code. In the example given, there are two particles and the numbers in the data set are 1 and 2. Since this was MCNP6 writing the file, these number correspond to neutrons and photons. If the number would have been 2 and 3, the particles would be photons and electrons.

D.7.4.7 Particle Edit List

The sixth data-set is a mapping of the particles to the edits and is needed internally by the code. Inside the code this information is stored in a 2-D array where the first index is for the edit number (where the maximum number of entries corresponds to the total number of edits) and the second index is for the particle. The value stored in any array slot is the internal edit number to which the particle contributes. For the example here, it can be seen that neutrons contribute to both edits while gammas contribute only to the second edit. A value of 0 terminates the particle's list if there are fewer particles in the edit than the maximum number of particles in the problem.

D.7.4.8 Edit Description

The seventh data-set begins with the title EDIT DESCRIPTION and contains 6 integers: the number of different particles, the number of elemental edits -- this is for both the second and third entries (one of these will be removed at a later date), the maximum number of problem energy bins, the maximum number of problem time bins, and the maximum number of response bins.

Table D.21: Elemental Edit Data-set Set Entries

Integer	Description
1	internal edit number
2	user edit number; negative if errors requested
3	special combined energy deposition indicator; 9 if a combined edit, 0 otherwise
4	particle number in REGL
5	particle number from MCNP6
6	number of energy bins
7	number of time bins
8	number of response bins

D.7.4.9 Edit Data Groups

At this point in the file there begins a variable number of data-sets which describe details of the elemental edits. What follows for each particle in the problem are 5 data-sets beginning with EDIT DATA and ending with RESPONSE BINS. Except for the unit conversion factors, most of the information presented in these next data-sets also appear in the title lines of the edit data-sets that appear later in the file.

The EDIT DATA data-set set contains 8 integer values described in Table D.21.

The CONVERSION FACTORS data-set provides two real numbers in one record: the energy unit conversion factor followed by the time unit conversion factor.

The next three data-sets each contain two real records. The ENERGY BINS data-set supplies the upper energy cut points for the energy bins followed by the energy multipliers for these energy bins. The TIME BINS data-set provides similar information for the time bins. The RESPONSE BINS data-set provides similar information for the response bins. There should always be one energy, one time, and one response bin whether requested by the user or not. It is up to the calling code to enforce this.

D.7.4.10 Materials

This data-set contains the alphanumeric names of the materials to associate with the material numbers assigned to each element. The names are ordered alphanumerically.

D.7.4.11 Cumulative Instance Element Totals

Parts are instantiated into the global mesh model in the order directed by the mesh input file. As the parts are added, the number of elements in that part are totaled and stored sequentially in the cumulative element totals array. The first element of this array contains the number of elements in the first instance. For the number of elements in the remaining instances (2 through max number of instances), subtract the value in the preceding array location from the instance's array location value. The values appearing in this data-set are just the cumulative values stored in this array. This information is primarily of interest internally to the mesh library and may be eliminated at a later date from the `EEOUT` file.

D.7.4.12 Instance Element Names

This data-set contains the alphanumeric names of the pseudo-cells. There is one record in the data segment for each pseudo-cell and, generally, these names are allowed to be 256 characters long. The order of the names in this data-set is the same order with which they are added to the global mesh model as directed by the mesh input file.

The user should note that the pseudo-cell names are slightly altered from what appears in the Abaqus mesh input file. Abaqus can segment a part. Each of these segments in REGL becomes a pseudo-cell. Because of a recent infrastructure change in REGL, it was necessary to separate the pseudo-cells from the instances and promote the pseudo-cell as the entity that builds the assembly. When the pseudo-cells are separated from the instances, a name is assigned to the pseudo-cell based on the original instance name. The instance name is appended with the letter P and a number starting at 1. The number is incremented for each additional pseudo-cell removed from the instance.

In a future version of this file, the file name of this section may be changed.

D.7.4.13 Instance Element Type Totals

The elements in the global mesh model are ordered and numbered by element type. This standard order is first-order tetrahedra, first-order pentahedra, first-order hexahedra, second-order tetrahedra, second-order pentahedra, and second-order hexahedra. Element numbers proceed sequentially from 1 to the maximum number of elements in the model. Any first order tetrahedra has an element number that is less than the first first-order pentahedra that appears in the model. Similar statements can be made regarding the element numbers concerning the other element types. For example, the first instance added to the model may contain a mixture of first-order pentahedra and hexahedra. The second instance added may contain only first-order tetrahedra. Even though the instance containing the tetrahedra was added later, its element numbers will always be less than the instance containing the pentahedra and hexahedra.

This data-set contains one record of 12 integers for each pseudo-cell in the model and the records appear in the order in which the pseudo-cells were added to the global mesh model as directed by the mesh input file. These 12 integers are grouped into pairs with each pair providing the first global element number and the last global element number for each element type. The order of the pairs is in standard order. In the example provided in this document, the first pseudo-cell contains only second-order hexahedra and its first element has global element number 204 and its last global element number is 331.

In a future version of this file, the file name of this section may be changed.

D.7.4.14 Nodes Group

The next three data-sets contain node location data. The first set, "NODES X (cm)", lists all of the x-locations for nodes 1 through max number of nodes. The second set, "NODES Y (cm)", lists all of the y-locations for nodes 1 through max number of nodes. The third set, "NODES Z (cm)", lists all of the z-locations for nodes 1 through max number of nodes. As indicated in the title line, these values are in centimeters, the required unit for the calling code (in this case, MCNP6).

D.7.4.15 Element Type

This data-set contains integers that describe the element type for each of the global elements starting at 1 and proceeding to the maximum number of elements in the mesh model. First-order tetrahedra, pentahedra, and hexahedra are given the values 4, 5, and 6, respectively. These number are just the number of faces in each element type. Second-order tetrahedra, pentahedra, and hexahedra are given the values 14, 15, and 16, respectively. These number are just the number of faces in each element type plus 10.

D.7.4.16 Element Materials

This data-set contains integers that represent the material number assigned to each element. Each element in the global mesh model is associated with a material through its material number. The elements appear sequentially from 1 to the maximum number of elements in the global mesh model.

D.7.4.17 Connectivity Data Group

There are a variable number of connectivity data-sets appearing in the `EEOUT` file, depending upon the element types present in the model. If all six types appear, there will be six data-sets appearing in standard order. In the example provided in this document, there is only one data-set in this group and it is for the first-order hexahedra.

The title line in this data-set contains the text ELEMENT ORDERED. This means that nodes appear by element. All of the nodes for the first element appear before the nodes for the second element, etc. This is a change from earlier versions of this file where the information was NODE ORDERED where all of the first nodes of all elements appeared before all of the second nodes of all of the elements, etc.

D.7.4.18 Nearest Neighbor Data Group

There are a variable number of nearest neighbor data-sets appearing in the `EEOUT` file, depending upon the element types present in the model. If all six types appear, there will be six data-sets appearing in the standard order. In the example provided in this document, there is only one data-set in this group and it is for the first-order hexahedra.

This data is ordered in the same fashion as the connectivity data. All of the neighbors for the first element appear before all of the neighbors of the second element, etc. In addition, the ordering of the neighbors is by face number. Therefore, a 0 appearing in the third neighbor position means there is no element appearing as a neighbor on that face.

D.7.4.19 Edit Sets Group: Data Output and Data Sets

Depending upon the edit requests from the calling code, a variable number of edit set results appear after the nearest neighbor data. Starting with the first particle and continuing through the total number of particle types tracked on the mesh, all of the regular edits are output by particle type. The title line segment that appears in all of these data-sets contain KWV-pairs which provide details describing the edit set.

Each particle edit list combination comprises its own edit group. The start of this group of edits is signified with a data-set consisting of just the meta data segment and a title line segment with three KWV-pairs. The

keyword for the first KWV-pair is DATA OUTPUT PARTICLE and its value is the particle number. The keyword for the second KWV-pair is EDIT LIST and its value is just the edit list number for the particle. The edit list is a list used by the mesh library. The keyword for the third KWV-pair is TYPE and its value is a set of alphanumeric characters that are an amalgamation of the edit type (e.g., FLUX) and the edit number (e.g., 14) specified in the calling code.

The next data-set is the DATA OUTPUT COMMENT data-set and consists of the meta data segment and a title line with one KWV-pair. The keyword is always DATA OUTPUT COMMENT. If no comment was specified by the user for the edit, the value field is left blank; otherwise, it contains the comment that was provided in the input.

After the DATA OUTPUT COMMENT data-set, the remainder of the data-sets forming the edit list appear. These data-sets are full data-sets with title line and data segments. The title line segment has six KWV-value pairs containing the time and energy bin numbers, bounds, and multipliers. Note that in order to avoid a KWV-pair with a non-existent value, extra keywords were added to the first KWV-pair; these extra keywords are DATA SETS and flag the data-set as the one with the numerical results. The keywords TMULT and EMULT are shorthand for time bin multiplier and energy bin multiplier, respectively.

If either the time or energy domains are broken into bins, the mesh library will automatically sum the bins to produce a total result. When this appears in the file the bin number is replaced with the string TOTAL and the corresponding bin values and multipliers are replaced with the string N/A, indicating that this information is not applicable because it was not input by the user. If both the time and energy domains are broken into bins, the mesh library will automatically sum the bins to provide total time results for each energy bin and total energy results for each time bin in addition to total time and total energy results.

After all of the regular edit set information is written to the `EEOUT` file, any edit sets for composite edits appear. The only thing that differs with this edit group is the particle descriptor in the DATA OUTPUT title line. For the regular edits the value of the first KWV-pair is a particle number. For the composite edits the value is a string where the particle numbers have been blended to produce an unique identifier (e.g., 1_2).

NOTE: Users familiar with earlier versions of MCNP6 and the `EEOUT` file should recognize that the components of the composite edit are no longer handled separately. This was done to save memory for really large problems.

D.7.4.20 Centroids Group

After the edit set data-sets there appear three data-sets for the element centroids. These three data-sets are presented in a similar fashion as the node information. X-centroids for all elements appear first in their own data-set followed by data-sets for the y-centroids and z-centroids, respectively.

D.7.4.21 Densities and Volumes

The next to last data-set is the material density values for each element for elements number one to the maximum number of elements in the global mesh model. The units for these values are grams per cubic centimeter as indicated in the corresponding title line.

The last data-set contains the volumes for each element for elements number one to the maximum number of elements in the global mesh model. The units for these values are cubic centimeters.

D.7.5 Example EEOUT File

The following `EEOUT` file is a composite of two different problems.

```
MCNP EDITS A
             0             2             1            16             1             1
 EEOUT  : ABAQUS
 VERSION:      6
            20             8             1           256             1             1
CALLING CODE LABELS
Prob ID         : simple cube, each element is a statistical set, 8 total|
Calling Code    : MCNP6|
Code Version    : 6.1.88|
Date & Time     :  02/08/11  09:48:4|
Inp File        : inp1007a|
Outp File       : inp1007ao|
Runtpe File     : inp1007ar|
Geom Inp File   : um1007.inp|
             0            12             1             1            42             1
NUMBER OF PARTICLES:                    2
NUMBER OF NODES    :                   27
NUMBER OF MATERIALS:                    6
NUMBER OF INSTANCES:                    6
NUMBER OF 1st TETS :                   30
NUMBER OF 1st PENTS:                    8
NUMBER OF 1st HEXS :                  128
NUMBER OF 2nd TETS :                   29
NUMBER OF 2nd PENTS:                    8
NUMBER OF 2nd HEXS :                  128
NUMBER OF HISTORIES:                 1000
NUMBER OF EDITS    :                    2
             0             2             1             1            43             1
LENGTH CONVERSION   :  1.0000000000000E+00
NORMALIZATION FACTOR:  1.0000000000000E-03
            14             1             2             4             2             2
 PARTICLE LIST
          1          2
            19             1             2             4             4             4
 PARTICLE EDIT LIST
          1          2          2          0
            18             1             2             4             6             6
 EDIT DESCRIPTION
        2       2       2       2       2       1
            10             1             2             4             8             8
 EDIT DATA
```

```
      1     -14       0       1       1       1       1       1
            19              1               3               8               2               2
 CONVERSION FACTORS
  1.00000E+00  1.00000E+00
            12              2               3               8               1               5
 ENERGY BINS
  1.00000E+36
  1.00000E+00
            10              2               3               8               1               5
 TIME BINS
  1.00000E+33
  1.00000E+00
            14              2               3               8               1               5
 RESPONSE BINS
  1.00000E+36
  1.00000E+00
            10              1               2               4               8               8
 EDIT DATA
      2     -36       9       2       2       2       2       1
            19              1               3               8               2               2
 CONVERSION FACTORS
  1.00000E+00  1.00000E+00
            12              2               3               8               2               5
 ENERGY BINS
  2.00000E+00  1.00000E+10
  1.00000E+00  1.00000E+00
            10              2               3               8               2               5
 TIME BINS
  1.00000E+00  1.00000E+39
  1.00000E+00  1.00000E+00
            14              2               3               8               1               5
 RESPONSE BINS
  1.00000E+36
  1.00000E+00
            10              6               1             256               1               1
 MATERIALS
Material_end_lin_hex_01
Material_end_quad_hex_06
Material_mid_lin_pent_04
Material_mid_lin_tet_02
Material_mid_quad_pent_05
Material_mid_quad_tet_03
```

```
           36              1              2              4              6              5
 INSTANCE CUMMULATIVE ELEMENT TOTALS
        128        136        166        174        203
        331
           23              6              1            256              1              1
 INSTANCE ELEMENT NAMES
Part-end_quad_hex-1P1
Part-mid_lin_pent-1P1
Part-mid_lin_tet-1P1
Part-mid_quad_pent-1P1
Part-mid_quad_tet-1P1
Part-end_lin_hex-1P1
           29              6              2              4             12             12
 INSTANCE ELEMENT TYPE TOTALS
        0         0         0         0         0         0         0         0         0         0       204       331
        0         0        31        38         0         0         0         0         0         0         0         0
        1        30         0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0       196       203         0         0
        0         0         0         0         0         0       167       195         0         0         0         0
        0         0         0         0        39       166         0         0         0         0         0         0
           13              1              3              8             27              5
 NODES X (cm)
 -5.00000E+00 -5.00000E+00 -5.00000E+00 -5.00000E+00 -5.00000E+00
 -5.00000E+00 -5.00000E+00 -5.00000E+00 -5.00000E+00  0.00000E+00
  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
  0.00000E+00  0.00000E+00  0.00000E+00  5.00000E+00  5.00000E+00
  5.00000E+00  5.00000E+00  5.00000E+00  5.00000E+00  5.00000E+00
  5.00000E+00  5.00000E+00
           13              1              3              8             27              5
 NODES Y (cm)
 -5.00000E+00  0.00000E+00  5.00000E+00 -5.00000E+00  0.00000E+00
  5.00000E+00 -5.00000E+00  0.00000E+00  5.00000E+00 -5.00000E+00
  0.00000E+00  5.00000E+00 -5.00000E+00  0.00000E+00  5.00000E+00
 -5.00000E+00  0.00000E+00  5.00000E+00 -5.00000E+00  0.00000E+00
  5.00000E+00 -5.00000E+00  0.00000E+00  5.00000E+00 -5.00000E+00
  0.00000E+00  5.00000E+00
           13              1              3              8             27              5
 NODES Z (cm)
  1.00000E+01  1.00000E+01  1.00000E+01  5.00000E+00  5.00000E+00
  5.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  1.00000E+01
  1.00000E+01  1.00000E+01  5.00000E+00  5.00000E+00  5.00000E+00
  0.00000E+00  0.00000E+00  0.00000E+00  1.00000E+01  1.00000E+01
```

```
 1.00000E+01  5.00000E+00  5.00000E+00  5.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00
            13           1           2           4           8          20
ELEMENT TYPE
 6  6  6  6  6  6  6  6
            17           1           2           4           8          10
ELEMENT MATERIAL
      1       1       1       1       1       1       1       1
            49           1           2           4          64           8
CONNECTIVITY DATA 1ST ORDER HEXS ELEMENT ORDERED
         10          11          13          14          19          20          22          23
         11          12          14          15          20          21          23          24
         14          15          17          18          23          24          26          27
         13          14          16          17          22          23          25          26
          1           2           4           5          10          11          13          14
          2           3           5           6          11          12          14          15
          5           6           8           9          14          15          17          18
          4           5           7           8          13          14          16          17
            37           1           2           4          48           6
NEAREST NEIGHBOR DATA 1ST ORDER HEXS
         55           0           0          40          47           0
         56           0           0          41          48          39
         57           0           0          42          49          40
         58           0           0          43          50          41
         59           0           0          44          51          42
         60           0           0          45          52          43
         61           0           0          46          53          44
         62           0           0           0          54          45
            58           0           0           0           0           0
DATA OUTPUT PARTICLE : 1 ; EDIT LIST : 1 ; TYPE : FLUX_14
            22           0           0           0           0           0
DATA OUTPUT COMMENT :
           145           1           3           8           9           5
DATA SETS RESULT TIME BIN : 1 ; TIME VALUE : 1.000E+33 ; TMULT : 1.00000E+00 ; ENERGY BIN : 1 ; ENERGY VALUE : 1.000E+36 ; EMULT : 1.00000E+00
0.00000E+00   4.43650E-02   4.52883E-02   4.73776E-02   4.99024E-02
4.24386E-02   4.63551E-02   4.10545E-02   4.90753E-02
            60           0           0           0           0           0
DATA OUTPUT PARTICLE : 1 ; EDIT LIST : 2 ; TYPE : ENERGY_36
            22           0           0           0           0           0
DATA OUTPUT COMMENT :
          145           1           3           8           9           5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : 1 ; ENERGY VALUE : 2.000E+00 ; EMULT : 1.00000E+00
```

```
 0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
 0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
           145             1             3             8             9             5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : 2 ; ENERGY VALUE : 1.000E+10 ; EMULT : 1.00000E+00
 0.00000E+00   2.39444E-02   2.43435E-02   2.58262E-02   2.70384E-02
 2.33694E-02   2.53041E-02   2.24486E-02   2.64320E-02
           137             1             3             8             9             5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : TOTAL ; ENERGY VALUE : N/A ; EMULT : N/A
  2.39444E-02   2.43435E-02   2.58262E-02   2.70384E-02  2.33694E-02
  2.53041E-02   2.24486E-02   2.64320E-02
            60             0             0             0             0             0
DATA OUTPUT PARTICLE : 2 ; EDIT LIST : 1 ; TYPE : ENERGY_36
            22             0             0             0             0             0
DATA OUTPUT COMMENT :
           145             1             3             8             9             5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : 1 ; ENERGY VALUE : 2.000E+00 ; EMULT : 1.00000E+00
 0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
 0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
           145             1             3             8             9             5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : 2 ; ENERGY VALUE : 1.000E+10 ; EMULT : 1.00000E+00
 0.00000E+00   1.23164E-03   1.33096E-03   1.34315E-03   1.38873E-03
 1.19235E-03   1.34903E-03   1.20800E-03   1.41040E-03
           137             1             3             8             9             5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : TOTAL ; ENERGY VALUE : N/A ; EMULT : N/A
  1.23164E-03   1.33096E-03   1.34315E-03   1.38873E-03  1.19235E-03
  1.34903E-03   1.20800E-03   1.41040E-03
            69             0             0             0             0             0
DATA OUTPUT PARTICLE : 1_2 ; EDIT LIST : 1 ; TYPE : ENERGY_36
            22             0             0             0             0             0
DATA OUTPUT COMMENT :
           131             1             3             8             9             5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : 1 ; ENERGY VALUE : 2.000E+00 ; EMULT : 1.00000E+00
 1.00000E-01   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
 0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
           131             1             3             8             9             5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : 2 ; ENERGY VALUE : 1.000E+10 ; EMULT : 1.00000E+00
 0.00000E+00   2.51760E-02   2.56744E-02   2.71694E-02   2.84271E-02
 2.45618E-02   2.66531E-02   2.36566E-02   2.78424E-02
           137             1             3             8             9             5
DATA SETS RESULTS TIME BIN : 1 ; TIME VALUE : 1.000E+39 ; TMULT : 1.00000E+00 ; ENERGY BIN : TOTAL ; ENERGY VALUE : N/A ; EMULT : N/A
  2.51760E-02   2.56744E-02   2.71694E-02   2.84271E-02  2.45618E-02
  2.66531E-02   2.36566E-02   2.78424E-02
```

```
            17              1              3              8              8              5
CENTROIDS X (cm)
-2.50000E+00 -2.50000E+00 -2.50000E+00 -2.50000E+00  2.50000E+00
 2.50000E+00  2.50000E+00  2.50000E+00
            17              1              3              8              8              5
CENTROIDS Y (cm)
-2.50000E+00  2.50000E+00 -2.50000E+00  2.50000E+00 -2.50000E+00
 2.50000E+00 -2.50000E+00  2.50000E+00
            17              1              3              8              8              5
CENTROIDS Z (cm)
 7.50000E+00  7.50000E+00  2.50000E+00  2.50000E+00  7.50000E+00
 7.50000E+00  2.50000E+00  2.50000E+00
            18              1              3              8              8              5
DENSITY (gm/cm^3)
 1.87401E+01  1.87401E+01  1.87401E+01  1.87401E+01  1.87401E+01
 1.87401E+01  1.87401E+01  1.87401E+01
            15              1              3              8              8              5
VOLUMES (cm^3)
 1.25000E+02  1.25000E+02  1.25000E+02  1.25000E+02  1.25000E+02
 1.25000E+02  1.25000E+02  1.25000E+02
```

D.8 Script to Generate HDF5 File Layouts

The script used to generate some of the LaTeX `dirtree` listings in this document by traversing an HDF5 file is given in Listing D.14.

Listing D.14: HDF5 Hierarchy Printing Utility Script (print_dirtree.py.txt)

```
#!/usr/bin/env python


class H5dirtree:
    def __init__(self, basename="/", offset=0):
        self.basename = basename
        self.offset = offset
        self.items = []

    def __call__(self, h5name, h5obj):
        import os

        # Nesting depth.
        d = h5name.count("/") + self.offset
        n = os.path.basename(h5name)
        separator = "{\color{lightgray}\dotfill}"
        label = (
            "{\color[HTML]{1b9e77}(dataset)}"
            if isinstance(h5obj, h5py.Dataset)
            else "{\color[HTML]{d95f02}(group)}"
        )
        self.items.append(d * "  " + ".{:} {:}{:}{:}".format(d, n, separator, label))
        e = d + 1
        separator = "{\color{lightgray}\dotfill}"
        label = "{\color[HTML]{7570b3}(attribute)}"
        for k, v in h5obj.attrs.items():
            self.items.append(
                e * "  " + ".{:} {:}{:}{:}".format(e, k, separator, label)
            )

    def make_dirtree(self):
        s = "\dirtree{%\n"
        s += ".1 {:}.\n".format(self.basename)
        for i in h5dt.items:
            s += "{:}.\n".format(i.replace("_", r"\_"))
        s += "}"
        self.dirtree = s


import __main__ as main

if __name__ == "__main__" and hasattr(main, "__file__"):

    import argparse
    import h5py
    import os
    import textwrap

    description = textwrap.dedent(
        """
    This script is used to traverse HDF5 files and collect the hierarchy to be
    printed in a tree-like way.
```

```
    """
    )

    epilog = textwrap.dedent(
        """
    Typical command line calls might look like:

    > python """
        + os.path.basename(__file__)
        + """ <h5filename> -g results
    """
        + u"\u2063"
    )

    parser = argparse.ArgumentParser(
        formatter_class=argparse.RawDescriptionHelpFormatter,
        description=description,
        epilog=epilog,
    )

    # Required positional argument(s).
    parser.add_argument("h5filename", type=str, help="HDF5 file to parse")

    # Optional named argument(s).
    parser.add_argument(
        "--group",
        "-g",
        type=str,
        default="/",
        help="parser start point (i.e., assumed root level)",
    )

    args = parser.parse_args()

    try:
        f = h5py.File(args.h5filename, "r")
    except:
        print("Couldn't process {:}".format(args.h5filename))
        raise

    try:
        r = f.get(args.group)
    except:
        print("Couldn't get group {:}".format(args.group))
        raise

    h5dt = H5dirtree(basename=args.group, offset=2)
    r.visititems(h5dt)
    h5dt.make_dirtree()
    print(h5dt.dirtree)
```

D.9 **inxc** File Structure

The **inxc** input is based on a 128-column "card" format and each requested case may require as many as seven cards. With the exception of the formatted title cards, all data provided in the **inxc** file are entered as

list-directed input. Repeat counts are allowed. A forward slash (/) may be used to terminate an input line; unread variables following a slash are assigned the default value(s). A description of the seven input cards follows:

Card 1	80-character problem title
Card 2	*ncase kplot l_res*

ncase	Defines the number of desired double-differential cross-section edits (DEFAULT: 0).
kplot	If nonzero, write cross-section edits to the MCNP6 **mctal** file (DEFAULT: 0). Plotting is available only with the **mctal** file.
l_res	If *l_res* $= 0$, no residual nuclei are calculated; if *l_res* $\neq 0$, perform a residual nuclei edit (DEFAULT: 0).

For each of the *ncase* cases, repeat the following cards 3 through 7, as required.

Card 3:	128-character case title
Card 4:	*nerg nang ntype fnorm imom iyield*

nerg	The number of energy (momentum) bin boundaries (DEFAULT: 0, i.e., produce only energy-integrated values).
nang	The absolute value, $\lvert$*nang*$\rvert$, provides the number of angle bin boundaries. For *nang* > 0, cosine bins are specified; for *nang* < 0, degree bins are specified (DEFAULT: 0, i.e., produce only angle-integrated energy spectra values).
ntype	The number of particle types to be tallied, including elastic scattering as a special case. If *ntype* $= 0$, all allowed particle types are included in the tally, including elastic scattering and elastic recoil (DEFAULT: 0).
fnorm	A normalization factor for the double-differential cross-section edit (DEFAULT: 1). For example, use *fnorm* $=$ 1000.0 to convert output to millibarns.
imom	If nonzero, treat the input energy bins as momentum bins (MeV/c) rather than energy bins (MeV). The output double-differential cross-section edits will be per unit momentum (DEFAULT: 0).
iyield	If nonzero, the output will be differential multiplicities or yields rather than differential cross sections. Multiplicities for nonelastic reactions are defined with respect to the nonelastic cross section; for elastic scattering, the differential multiplicity is with respect to the elastic cross section (DEFAULT: 0).

Card 5:	Energy (momentum) bin boundaries (present if *nerg* > 0).

Four modes of input are allowed. The values are energy in MeV or, if *imom* $\neq 0$, momentum in MeV/c:

1. All bins E_i for $i = 1, \ldots,$ *nerg* may be specified in increasing order.

2. If only one energy (momentum) value E_1 is entered, then $E_i = iE_1$ for $i = 2, \ldots, \mathit{nerg}$.
3. If $N < \mathit{nerg}$ bins E_i for $i = 1, \ldots, N$ are entered in increasing order, then $E_i = E_{i-1} + (E_N - E_{N-1})$ for $i = N+1, \ldots, \mathit{nerg}$.
4. If only two values, V_1 and V_2, are entered, with $V_1 < 0$ and $V_2 > 0$, then $E_{\mathit{nerg}} = V_2$ and $\log_{10}(E_{i-1}/E) = V_1$ for $i = 1, \ldots, \mathit{nerg} - 1$ (equal-lethargy spacing).

Card 6: Angle bin boundaries (present if $\mathit{nang} \neq 0$).

For $\mathit{nang} > 0$, cosine bins are entered by one of the following options:

1. Cosine bins μ_i for $i = 1, \ldots, \mathit{nang}$ are entered in increasing order; μ_{nang} is always set to 1.
2. If a null record "/" is present, nang equally spaced cosine bins $-1 < \mu_i \leq 1$ are defined with $\mu_{\mathit{nang}} = 1$.
3. If only one value is entered, then the entered value is μ_1 and $\mu_{\mathit{nang}} = 1$; the remaining cosine boundaries are interpolated uniformly.
4. If two (or more) values are entered, then the first entered value is μ_1, the second is $\mu_{\mathit{nang}-1}$, and $\mu_{\mathit{nang}} = 1$; the remaining cosine boundaries are interpolated uniformly.

For $\mathit{nang} < 0$, the degree bins are entered by one of the following options:

1. Degree bins φ_i for $i = 1, \ldots, \mathit{nang}$ are entered in decreasing order; φ_{nang} is always set to 0.
2. If a null record "/" is present, nang equally spaced degree bins $180 < \varphi_i \leq 0$ are defined with $\varphi_{\mathit{nang}} = 0$.
3. If only one value is entered, then the entered value is φ_1 and $\varphi_{\mathit{nang}} = 0$; the remaining cosine boundaries are interpolated uniformly.
4. If two (or more) values are entered, then the first entered value is φ_1, the second is $\varphi_{\mathit{nang}-1}$, and $\varphi_{\mathit{nang}} = 0$; the remaining cosine boundaries are interpolated uniformly.

Card 7: Particle types to be tallied for this case (present if $\mathit{ntype} > 0$).

Entries are a set of flags, k_i, for $i = 1, \ldots, \mathit{ntype}$ (see Table D.22). These flags identify the particle types to be included in a single cross-section edit case. Negative entries ($k_i < 0$) indicate tallies related to elastic scattering. Values of $k_i > 0$ designate the tallying of production of the indicated particles type by nonelastic processes.

In the absence of any nonelastic reaction models, only the elastic cases will produce a meaningful tally.

When the default ($\mathit{ntype} = 0$) is taken, all 26 edit types are allowed. Only brief output is produced when no secondaries of a given type occur. The ordering by particle type in the output is the following: proton, neutron, π^+, π^0, π^-, K^+, K^0, anti-K^0, K^-, anti-proton, anti-neutron, deuteron, triton, helion, alpha, photon, electron, positron, μ^-, μ^+, ν_e, anti-ν_e, ν_m, anti-ν_m, elastic scattered projectile, and elastic recoil nucleus.

⚠ Caution

The particle-type identifiers given in Table D.22 are not exactly the same as defined by the general MCNP6 numbering scheme for particle type [Table 4.3]. Users therefore should choose values for ntype carefully.

Table D.22: Particle-type Designators for the *ntype* k_i Flag

Flag k_i	Particle	Flag k_i	Particle
1	neutron (n)	13	muon neutrino ($\nu_{\rm m}$)
2	photon (γ)	14	anti muon neutrino ($\overline{\nu}_{\rm m}$)
3	electron (e^-)	15	positive kaon (K^+)
4	positron (e^+)	16	negative kaon (K^-)
5	proton (p^+)	17	kaon, short (K^0_S) (previously K^0)
6	positive pion (π^+)	18	kaon, long (K^0_L) (previously anti-K^0)
7	negative pion (π^-)	19	anti proton ($\overline{p}$)
8	neutral pion (π^0)	20	anti proton ($\overline{p}$)
9	negative muon (μ^-)	21	deuteron (d) (previously ^{2}H)
10	positive muon (μ^+)	22	triton (t) (previously ^{3}H)
11	electron neutrino (ν_e)	23	helion (^{3}He) (previously ^{3}He)
12	anti electron neutrino ($\overline{\nu}_e$)	24	alpha particle (α) (previously ^{4}He)
−1	elastic scattered projectile		
−2	elastic recoil nucleus		

The allowed k_i flag values (needed on card 7) are given in Table D.22 (now with particle descriptions consistent with Table 4.3 though with some symbols given previously also indicated).

Appendix E

Utilities

This appendix describes the use of the many utilities that come with the MCNP code.

All of these utilities are provided as-is. However, users of these utilities can contact mcnp_help@lanl.gov if any issues are encountered or to provide other feedback.

E.1 Doppler Broadening Resonance Correction Library Generation (**dbrc_make_lib**)

In order to use the DBRC capabilities, the code needs access to 0-K elastic-scattering data. This tool will scan **xsdir_mcnp6.3** within **DATAPATH** for ENDF/B-VII.1 data (with suffix **.85c**) and ENDF/B-VIII.0 data (with suffix **.05c**) and store them in **DBRC_endf71.txt** and **DBRC_endf80.txt** respectively. These should be moved into the root of **DATAPATH** for the code to find them.

E.1.1 User Interface

This utility has no command line options. If any files the utility is expecting are missing, it will print out the file it attempted to open and exit. If any files the utility expected are missing, it will print out the files it attempted to open and exit.

During operation, the utility will print out some diagnostic data, including which table is being processed, what file it was found in, the number of data points, and whether or not unresolved resonances adjusted the data points extracted. The utility will then note that the data were written to disk. This process will repeat, once for ENDF/B-VII.1 and once for ENDF/B-VIII.0.

E.2 Event Log Analyzer (**ela.pl**)

The Event Log Analyzer (ELA) is a Perl utility with a Tk interface used as a research tool to interrogate the event log produced by the MCNP code to understand event-by-event evolution of the random walk undertaken by a computational particle (i.e., a history). This utility has also been used as part of the MCNP "Advanced Variance Reduction" class. MCNP practitioners are welcome to use this utility as is; however, only limited support is available for it.

This utility was originally designed to work with the MCNP code, version 5.1.50. This is because a refined event log was added with version 5.1.50. Generating an event log with the MCNP code is usually performed by using the DBCN card with entries such as those shown in Listing E.1.

Listing E.1: example_event_log.mcnp.inp.txt

```
dbcn 2j 1 5 10000 $ Print event log for histories 1--5, limit to 10k lines
```

Further documentation can be found in [352, 353].

Note that [352] is the latest released documentation on ELA, but it is not current with all features. Around 2011, "Distance Analysis" was added and the "Required Data" tab was eliminated. Also note that checking track weight against weight window values only works for cell-based weight windows.

E.2.1 User Interface and Example

The ELA is primarily a GUI application, with its original documentation provided in [352]. However, a complete example follows based on Listing E.2.

Listing E.2: example_event_log.mcnp.inp.txt

```
Reduced-radius Godiva sphere showing an event log
c
c CELL CARDS
10   100  -18.74  -1  imp:n=1
20   0             1  imp:n=0

c SURFACE CARDS
1    so   8.5

c DATA CARDS
sdef erg=d1
sp1 -3 0.965 2.29
m100  92235.00c -.9473
      92238.00c -.0527
dbcn 2j 1 5 10000 $ Print event log for histories 1--5, limit to 10k lines
nps 100
print
rand gen=2 seed=12345
```

The example events counted are shown in Fig. E.1. The event-tree, surface-analysis, and distance-analysis settings shown in Figs. E.2, E.3, and E.4, respectively, produce the example event tree, surface analysis, and distance analysis shown in Figs. E.5, E.6, and E.7, respectively.

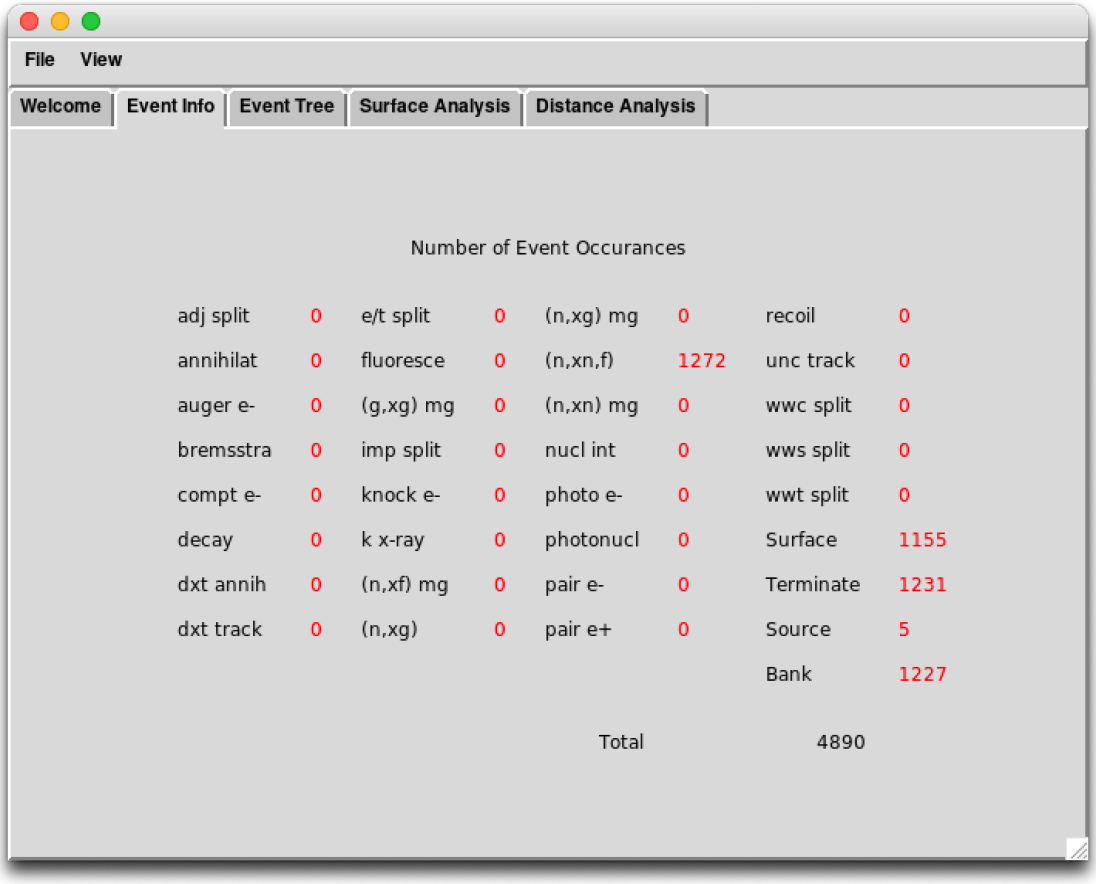
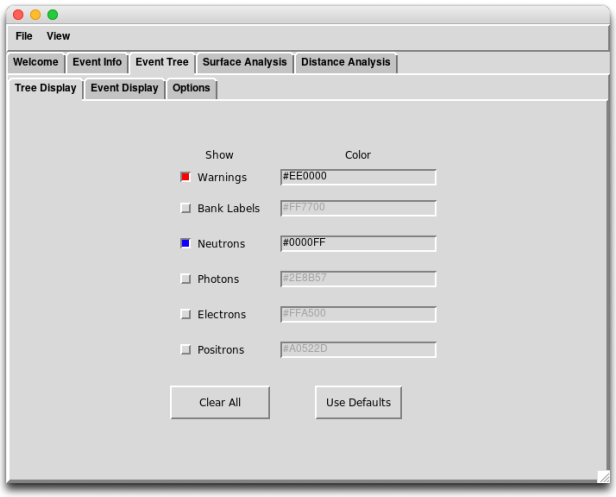
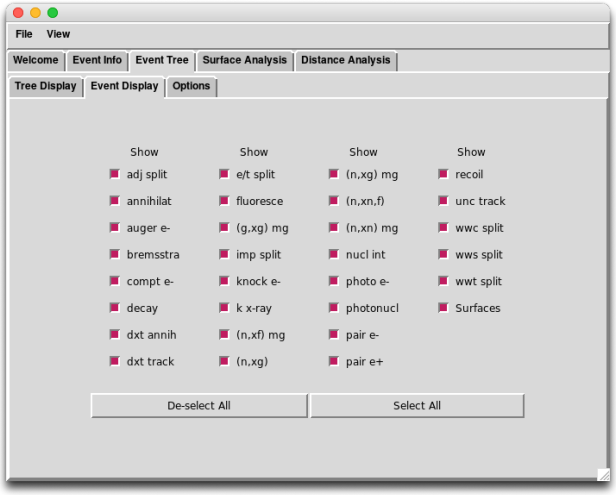


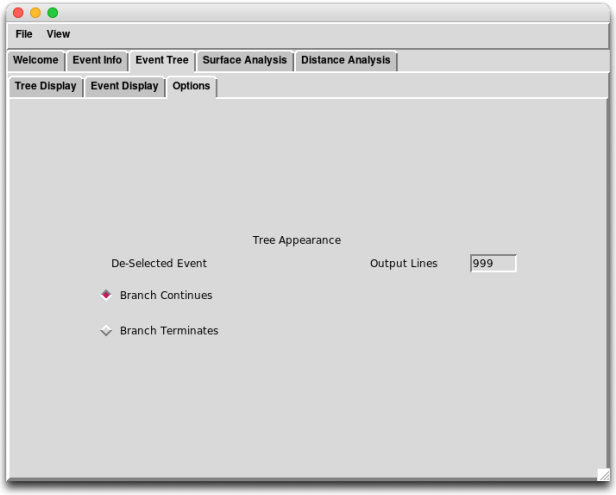
Figure E.1: ELA Event Counter



(a) View-enabled Events and Color Settings



(b) Enabled Subevent Settings



(c) Other Event-tree Options

Figure E.2: ELA Event-tree Settings

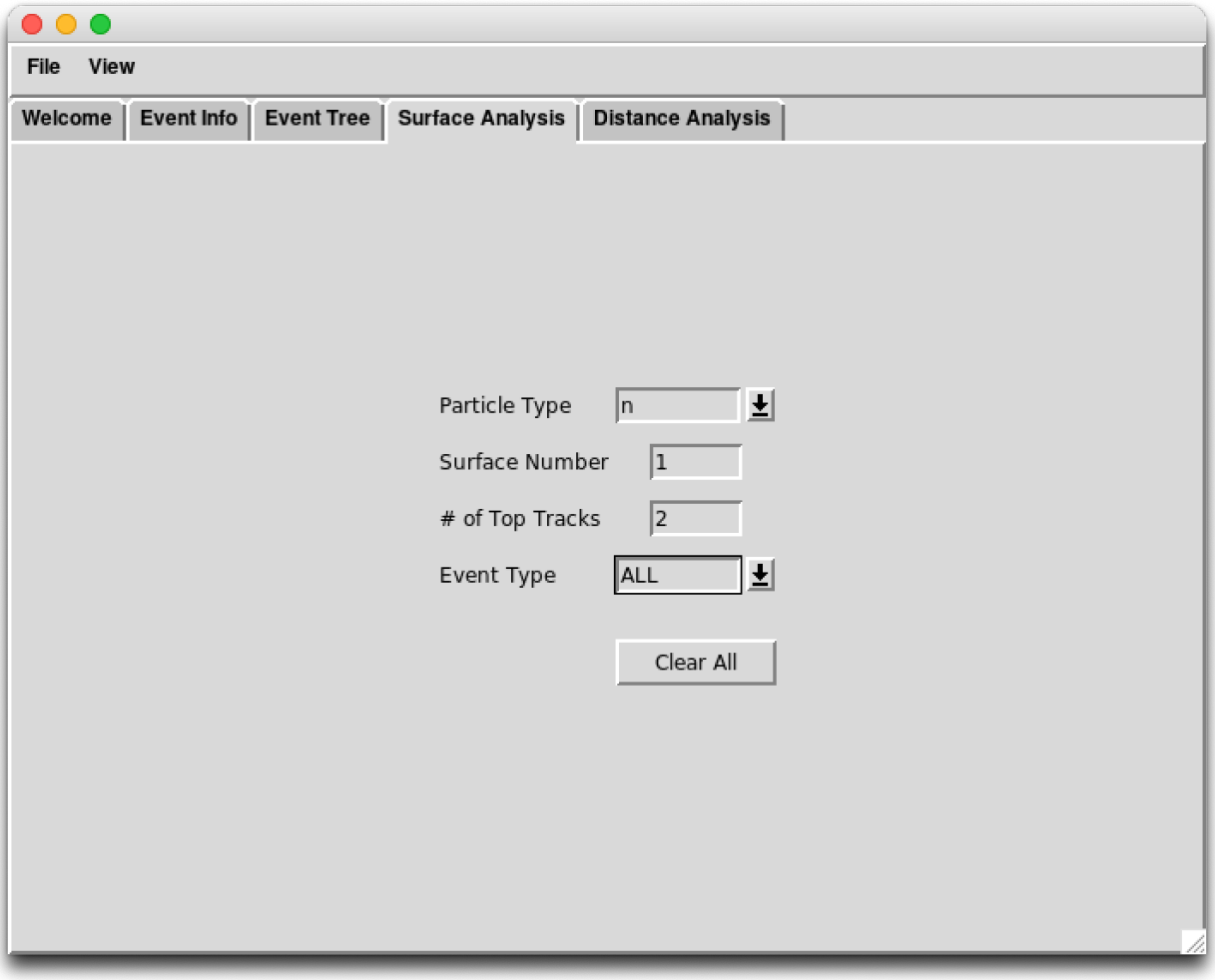


Figure E.3: ELA Surface-analysis Settings

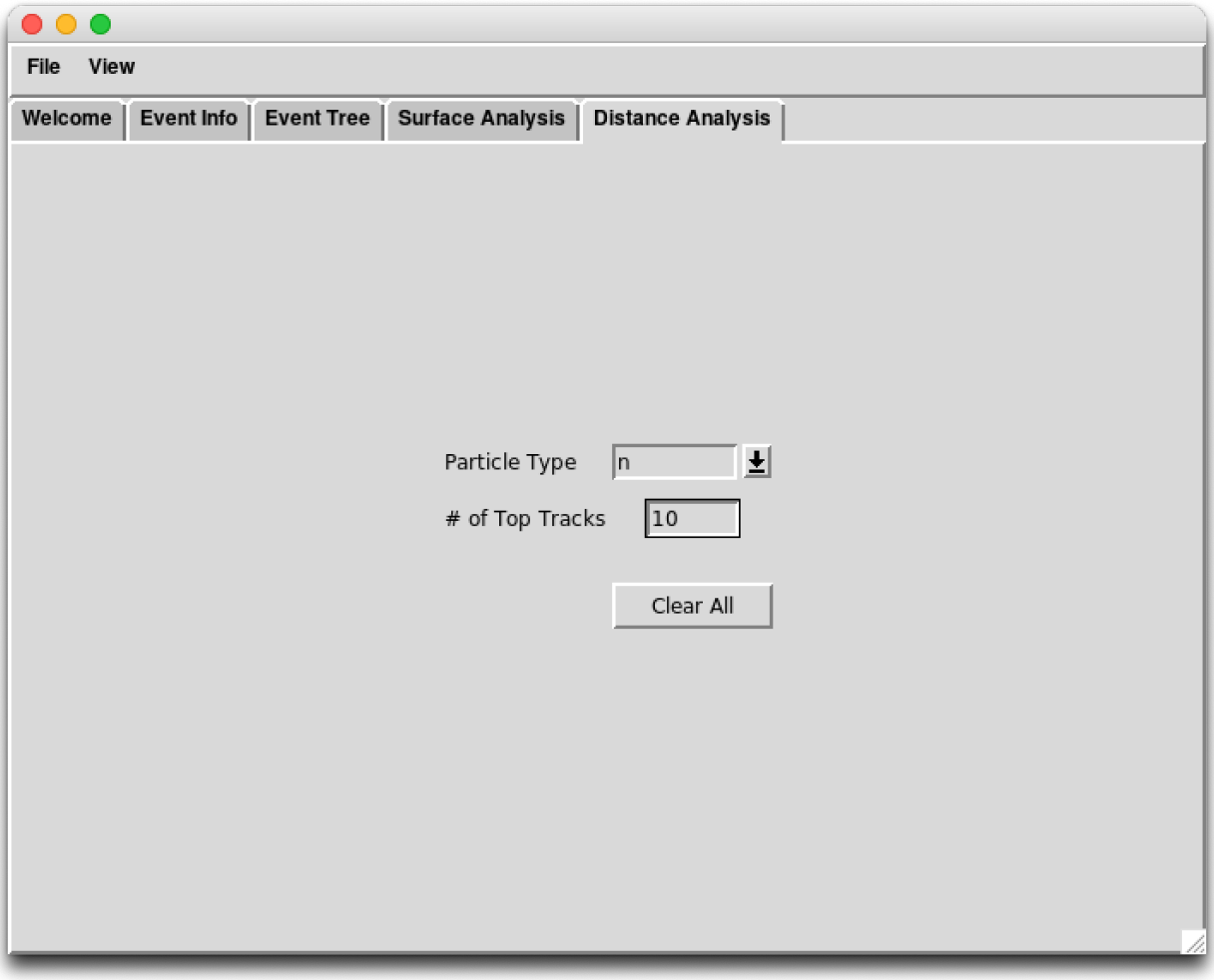


Figure E.4: ELA Distance-analysis Settings

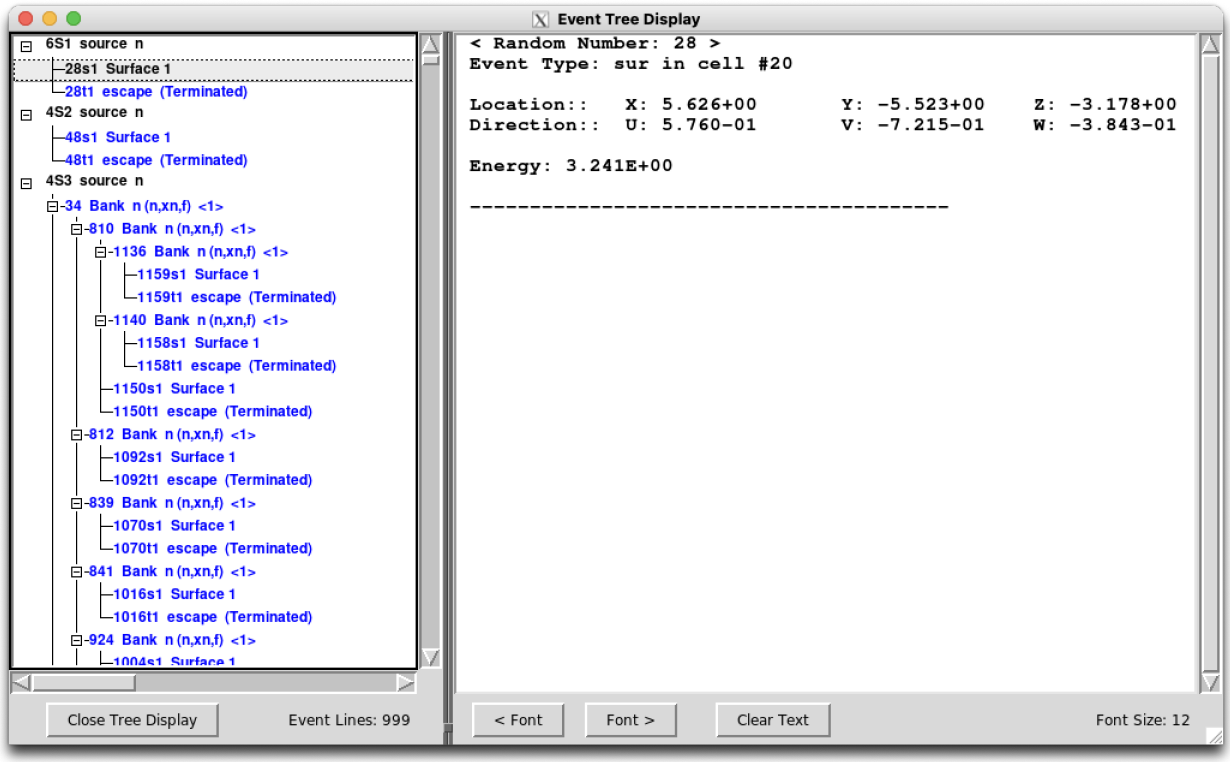


Figure E.5: ELA Event Tree

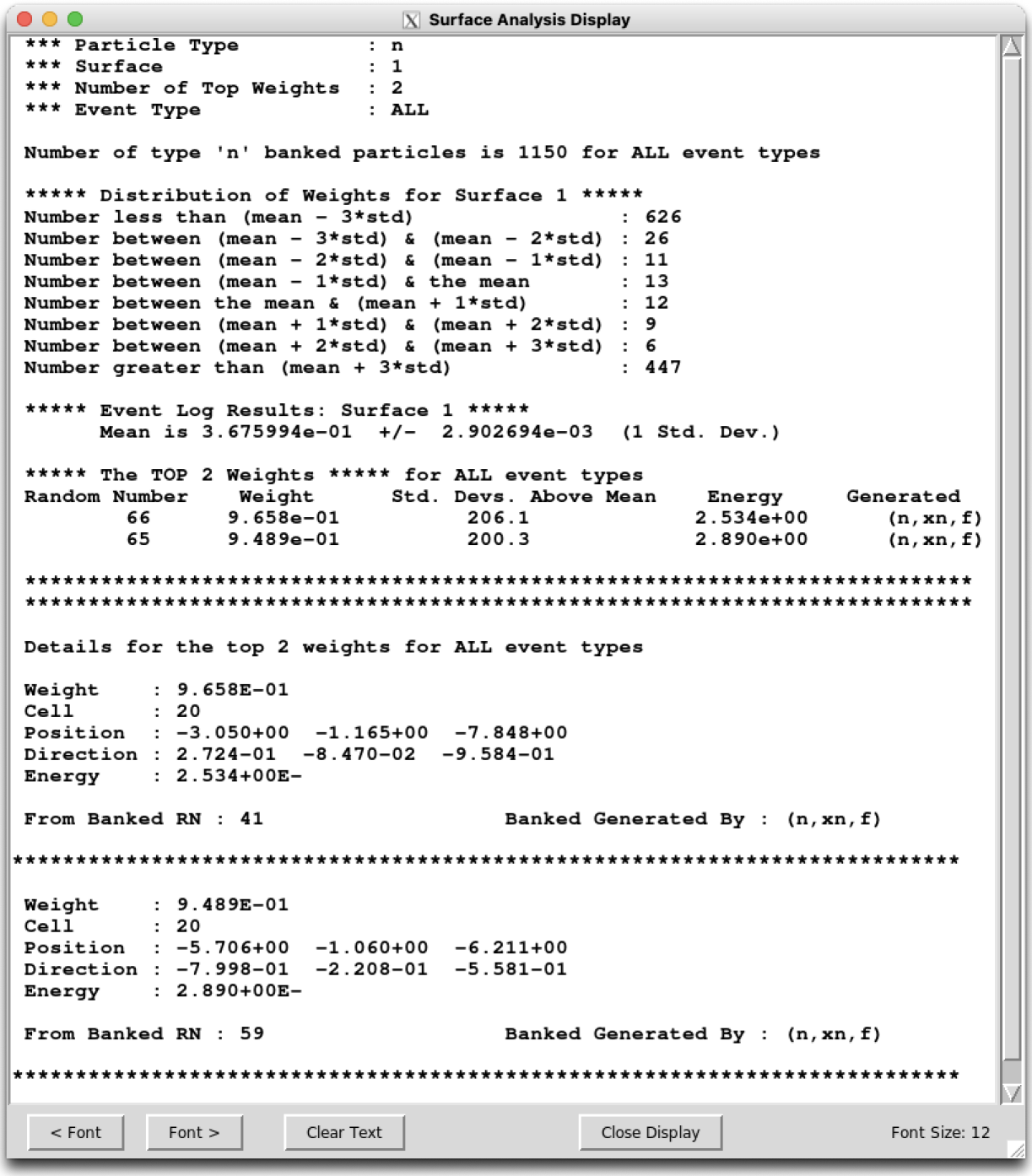


Figure E.6: ELA Surface-analysis Results

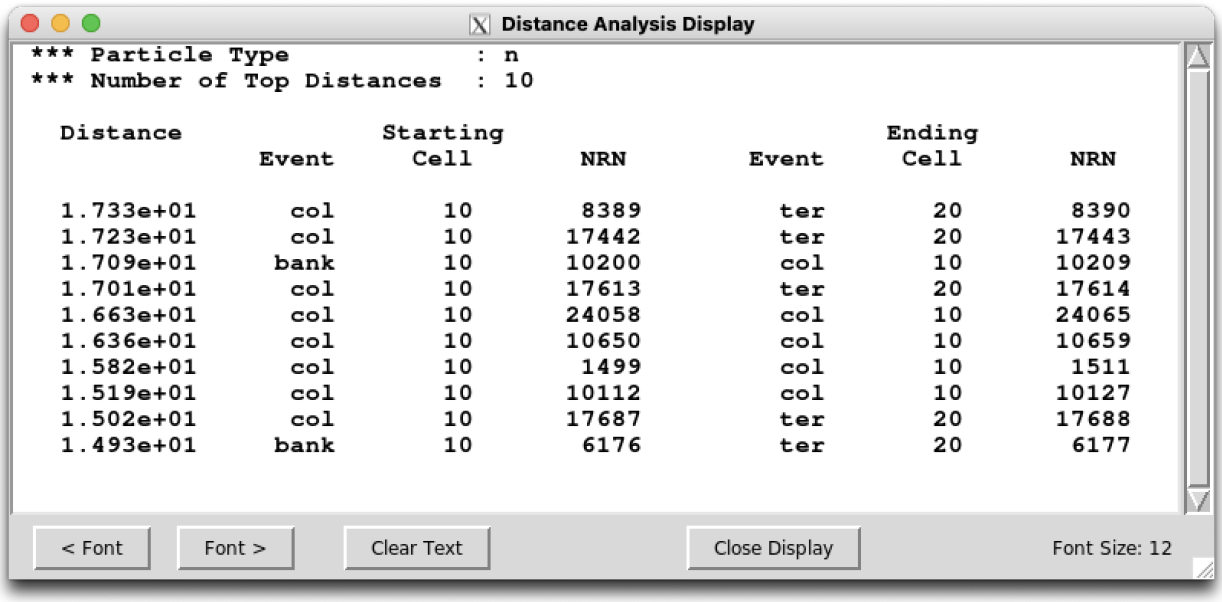


Figure E.7: ELA Distance-analysis Results

E.2.2 Change Log

This section describes the evolution of the Event Log Analyzer.

Version 1.0, 10 September 2007

- Original release.

Version 1.1, 3 March 2011

- Added code so that ELA knows the directory in which it is installed. This directory information is now used in opening its resource files.
- Added command-line interface operation. Try one of the following to learn more: `perl ela.pl --help` or `./ela.pl -h`.

Note that testing with some versions of Perl and Perl/Tk experience warning messages at start up and core dumps when terminating GUI operations.

Note also that use with Cygwin exhibits periodic misbehavior; with some installs and systems the menu bar is hidden. An alternative free Perl download for Windows is the Active Perl or Active State Perl.

Version 1.2, 19 August 2013

- Updated `#!` path to be generic.

Version 1.3, 18 January 2022

- Reformatted **`README.md`** file to Markdown and extracted content to MCNP manual. Revised wording to improve clarity and embed an example.

E.3 On-the-fly Doppler Broadened Data Fitting (fit_otf)

This tool generates data for the on-the-fly temperature dependent nuclear data capability (see card OTFDB). Generating a library is a two-stage process. First, the nuclear data library is scanned to generate a temperature-unionized energy grid in **-ugrid** mode. Then, in **-fit** mode, at each point in the unionized grid, a polynomial curve fit is generated as a function of temperature.

E.3.1 User Interface

Modes (one must be selected)

-fit	Enables fitting the nuclear data. This is the second stage to generating a library. Enables **fit**-specific arguments below.
-test	Tests the fitting procedure over a defined energy range. Mainly used for debugging. Enables **fit** and **test** specific arguments below.
-ugrid	Generates the unionized grid. This is the first stage to generating a library. Enables **ugrid**-specific arguments below.

Arguments valid in all modes

-ace_file	The ACE file to extract the nuclear data from. (OPTIONAL, DEFAULT: finds the table identifier set on **-zaid** in **xsdir_mcnp6.3** in the **DATAPATH**)
-tol_err	The target relative error to process the library to. During the ugrid stage, this is used to determine how fine the energy grid is. During the fit stage, this is used to determine what order of polynomial is necessary in temperature. (OPTIONAL, DEFAULT: 0.001)
-ugrid_file	The name of the file for the unionized grid. It is generated with **-ugrid**, and used with **-fit**. (REQUIRED)
-zaid	The table identifier to perform the operation on. This must exactly match the value in the **xsdir** file. (REQUIRED)

ugrid (unionized energy grid generation)-specific arguments

-ugrid_ace_zaid	Which table identifier to process into a unionized energy grid. Should match **-zaid**. (REQUIRED)
-ugrid_tmin	The lower temperature bound to use in Kelvin. If this value is lower than the temperature of the nuclear data, the nuclear data temperature will be used instead. (OPTIONAL, DEFAULT: 250)
-ugrid_tmax	The upper temperature bound to use in Kelvin. (OPTIONAL, DEFAULT: 3200)
-ugrid_tinc	The spacing between temperatures used during processing in Kelvin. Finer values can generate higher quality data at the cost of greater processing time. (OPTIONAL, DEFAULT: 50)

fit-specific arguments

-otf_file	The output filename for the coefficients. (OPTIONAL, DEFAULT: **otf_file.txt**)
-order	If present, both **-order_min** and **-order_max** are set to this value. (OPTIONAL, DEFAULT: unset)
-order_min	The minimum curve-fit order used to fit the temperature data. (OPTIONAL, DEFAULT: 1)
-order_max	The maximum curve-fit order used to fit the temperature data. In general, increasing this value will provide no benefit, as the numerical stability of the fitting algorithm gets worse beyond an order of 8. (OPTIONAL, DEFAULT: 8)
-tmin	The lower temperature bound to use in Kelvin. If this value is lower than the temperature of the nuclear data, the nuclear data temperature will be used instead. (OPTIONAL, DEFAULT: 250)
-tmax	The upper temperature bound to use in Kelvin. (OPTIONAL, DEFAULT: 3200)
-tinc	The spacing between temperatures used during processing in Kelvin. Finer values can generate higher quality data at the cost of greater processing time. (OPTIONAL, DEFAULT: 50)

test-specific arguments

-test_emin	Lower energy bound to test fitting approach in MeV. (REQUIRED)
-test_emax	Upper energy bound to test fitting approach in MeV. (REQUIRED)

E.3.2 Examples

In this example, the ENDF/B-VII.1 library for ^{238}U will be processed from 250 K to 3000 K with a target tolerance of 0.1%. A temperature step of 25 K will be used. In order to get to 250 K, a nuclear data library that is at or below this energy must be provided. Here, **92238.86c** has a temperature of 250 K. The first step is to generate the unionized energy grid:

```
fit_otf -zaid 92238.86c -ugrid -ugrid_ace_zaid 92238.86c \
        -ugrid_tmin 250 -ugrid_tmax 3000 -ugrid_tinc 25 \
        -ugrid_file ugrid_92238.86c
```

Once this process is completed, the file **ugrid_92238.86c** will contain the necessary unionized energy grid for phase two:

```
fit_otf -zaid 92238.86c -fit -tmin 250 -tmax 3000 -tinc 25 \
        -ugrid_file ugrid_92238.86c -otf_file otf_92238.86c.txt
```

This process will generate the necessary **otf_92238.86c.txt** file. This file can be added to the **DATAPATH**, or stored alongside the input file in the working directory. At the end of processing, the output will describe the quality of the library, including the number of energy points that had errors exceeding **-tol_err**, and by how much this error was exceeded:

```
Overall error checks:
     mt=  1  max-err= 0.100%     for   e=  10589.2     eV,  t=  275.0 K
     mt=101  max-err= 0.103%     for   e=  4266.55     eV,  t=  350.0 K
     mt=  2  max-err= 0.433%     for   e=  20.2818     eV,  t=  350.0 K
     mt=301  max-err= 0.232%     for   e=  718.922     eV,  t=  350.0 K
     mt=202  max-err= 0.108%     for   e=  4266.81     eV,  t=  350.0 K
     mt= 18  max-err= 0.041%     for   e=  723.763     eV,  t=  625.0 K
     mt=102  max-err= 0.108%     for   e=  4266.81     eV,  t=  350.0 K
     mt=444  max-err= 0.002%     for   e=  20.6298     eV,  t=  350.0 K

     Overall maximum error               =  0.433%

     Number of energies with err > 0.10% =    101
```

If these error values are acceptable, the file is ready for use with the `OTFDB` card. If it is not, one can tune the parameters using the testing mode prior to re-evaluating the whole library. In the example below, the 0.433% error at 20.2818 eV is re-examined with a maximum fit order of 10.

```
fit_otf -zaid 92238.86c -test -fit -tmin 250 -tmax 3000 -tinc 25 \
        -ugrid_file ugrid_92238.86c -order_max 10 \
        -test_emin 2.0e-5 -test_emax 2.1e-5
```

With this data, the new maximum error is 1.528%, indicating that increasing the order will not improve the result past 0.433% due to numerical instability.

E.4 Gridconv (**gridconv**)

The **gridconv** program is a post-processing code used with **mdata** and **mctal** output files. **Gridconv** converts the data in these files to formats compatible with various external graphics packages. Those permitted are:

IDL®	IDL (Interactive Data Language) is a product of Harris Geospatial Solutions, Inc. (https://www.l3harrisgeospatial.com/Software-Technology/IDL)
Tecplot	Tecplot is a product of Tecplot, Inc. (https://www.tecplot.com)
Gnuplot	Freeware. (http://www.gnuplot.info). Only 1D and 2D plots supported.

⚠ Caution

GRIDCONV has historically supported the IDL and Tecplot output formats; however, these output formats have not been tested in modern viewers at the time of writing.

E.4.1 User Interface

Gridconv has no command line options. Once started, the code will prompt the user for the information needed to create the desired formatted graphics input files.

After the header information from the **mdata** or **mctal** file has been read, **gridconv** can either produce an ASCII file from the data file or generate the required graphics input files as requested by the user. Note that the ASCII file contains raw data not normalized to the number of source particles. The reason for the option to write an ASCII file is that sometimes users will want to look at the values in the **mdata** file before doing any plotting, or check the numerical results for a test case. The ASCII option is also useful for porting the data in the **mdata** file to another computer platform and for reading the data into graphics packages not currently supported by **gridconv**.

Gridconv is currently set up to generate one-, two-, or three-dimensional graphics input files with any combination of binning choices. Once the graphics input file has been generated, **gridconv** gives the user the option of producing another file from the currently selected `TMESH` tally, selecting a different `TMESH` tally available in this **mdata** file, or reading information from a different file. There is always the option to exit the program.

Gridconv can also process any and all tallies written to the **mctal** file. The code is still interactive but now shows all tallies in the problem, from which any tally may be selected. The user has the option of generating one- or two-dimensional output. The user is then told about the bin structure so the one or two free variables may be selected. Energy is the default independent variable in the one-dimensional case. There are no default variables for the two-dimensional case. The order in which the two-dimensional bin variables are selected does not make any difference to the output; the order of the processing will be as it appears in the **mctal** file.

E.5 Cross Section Library Manipulation Tool (**makxsf**)

The **makxsf** code is a utility program for manipulating cross-section library files for use with the MCNP code and potentially other codes that make use of A Compact ENDF (ACE) formatted nuclear data files [344]. It can be used to convert ACE data files between ASCII and binary formats, to make customized libraries containing selected datasets, and to create temperature-dependent libraries.

⚠ Caution

The **makxsf** utility is no longer actively supported as the capabilities of this code have been usurped by other openly available software packages. If people rely on the capabilities of **makxsf**, please read about the known issues and alternative solutions in §E.5.2.

E.5.1 User Interface

As the interface to **makxsf** has remained unchanged for an extended period of time, please refer to the user guidance within the "The **makxsf** Code with Doppler Broadening" report [344].

E.5.2 Known Issues and Alternative Solutions

Temperature Interpolation of Continuous $S(\alpha, \beta)$ Thermal Scattering Data

The temperature interpolation capabilities are unavailable for use with the continuous form of the $S(\alpha, \beta)$ thermal scattering data tables. When **makxsf** was originally developed, the continuous form of the $S(\alpha, \beta)$ data did not exist.

Because **makxsf** was never updated to handle the continuous $S(\alpha, \beta)$ data, the alternative solutions to handling temperature-specific continuous thermal scattering data include:

- using NJOY [354] to generate $S(\alpha, \beta)$ data at the precise temperature needed.
- using stochastic mixing of $S(\alpha, \beta)$ data to approximate temperature effects (see the `MT0` card for more details).
- running bounding calculations with nearest lower and upper temperatures.
- using the nearest-temperature continuous $S(\alpha, \beta)$ data.

General Temperature Treatments

Both the Doppler broadening implementation used for resolved resonance data and the temperature interpolation scheme used for unresolved resonance data and discrete $S(\alpha, \beta)$ thermal scattering data are approximate methods that have not been updated nor re-validated in recent years.

It is recommended that people migrate toward using the production NJOY code [354], available as open-source software (https://github.com/njoy/NJOY2016), for all of their needs with respect to processing nuclear data at precise temperatures.

E.6 Merge ASCII Tally Files (**merge_mctal.pl**)

The **merge_mctal.pl** utility is a command-line interface-based Perl script that can be used to statistically merge multiple MCNP ASCII tally (**mctal**) files into a single resulting **mctal** file. MCNP practitioners are welcome to use this utility as is; however, only limited support is available for it. Note that a similar C++-based utility with the same name is provided with the MCNPTools software as of version 5.3.0.

Further documentation can be found in [355].

E.6.1 User Interface

To run this program, the user should have Perl version 5.8.5 or newer. All interaction with it is performed in a command-line interface.

The utility can be executed by typing `perl merge_mctal.pl` *`mctal1 mctal2`* etc. at the command line where *`mctal1`*, *`mctal2`*, and any other such entries form a space–delimited list of MCNP ASCII tally file names. Alternatively, one can make the script executable and provide the appropriate Perl path as the first line in the file. In either case, and optionally, the name of the resulting file can be given with the `-o` option such as `perl merge_mctal` *`mctal1 mctal2`* `-o` *`mctal.out`*.

E.6.2 Example

If one performs two statistically independent but otherwise identical MCNP calculations using the input shown in Listing E.3 and another modifying the random number generator as shown in Listing E.4 to produce ASCII tally files **mctal** and **mctam**, respectively, they can be merged with the command `perl merge_mctal.pl mctal mctam -o merged_mctal.txt`.

The expected output is shown in Listing E.5, where it is important to note that the merged file has individual tally results merged but the tally fluctuation chart values are not merged.

Listing E.3: merge_mctal1.mcnp.inp.txt

```
Generate mctal file to merge with another
1000 10 -9.98207e-1 -100      imp:n=1
9999  0              100      imp:n=0

100 so 90

mode n
sdef
m10       1001.70c     0.666657 $ Pseudo-Water, Liquid @ 23.15 deg-C
          8016.70c     0.333343 $ Density: 0.998207 g/cc from PNNL-15870, Rev. 1
mt10      lwtr.10t
f4:n 1000
c
print
prdmp 2j 1 $ Write MCTAL file at conclusion of calculation
rand gen=2 seed=12345
nps 10000
```

Listing E.4: merge_mctal2.mcnp.inp.txt

```
rand gen=2 seed=34567
```

Listing E.5: Expected Output from Merging Two ASCII Tally Files

```
...Reading MCTAL file:   mctal
...Reading MCTAL file:   mctam
...Merging  (except TFC)

...Creating merged MCTAL file = merged_mctal.txt
```

E.6.3 Change Log

This section describes the evolution of **merge_mctal.pl**.

Version 1.0, December 2003

- Original release.

Version 1.1, July 2010

- Added ability to handle MCNP6 **mctal** formats and larger numbers (up to 99,999,999).

Version 1.2, January 2022

- Reformatted README file to Markdown. Migrated content to MCNP manual. Revised wording and added example.
- Added **.pl** file extension.
- Assigned version numbers within utility.

E.7 Merge Mesh Tally Files (**merge_meshtal.pl**)

The **merge_meshtal.pl** utility is a command-line interface-based Perl script that drives a compiled binary based on C++ code can be used to statistically merge multiple MCNP Type B (MCNP5-style) ASCII mesh tally (**meshtal**) files, such as those created with the FMESH card using the `out = col` option, into a single resulting file. Note that only the `col` output option provides the format needed by this utility, so newer output options such as `cf`, `colsci`, `cfsci`, and `xdmf` are incompatible with this utility.

MCNP practitioners are welcome to use this utility as is; however, only limited support is available for it. Note that a similar C++-based utility with the same name is provided with the MCNPTools software as of version 5.3.0.

Further documentation can be found in [355].

E.7.1 User Interface

To run this program, the user should have Perl version 5.8.5 or newer. All interaction with it is performed in a command-line interface.

The utility can be executed by typing `perl merge_meshtal.pl` *meshtal1 meshtal2* etc. at the command line where *meshtal1*, *meshtal2*, and any other such entries form a space–delimited list of MCNP ASCII mesh tally file names. Alternatively, one can make the script executable and provide the appropriate Perl path as the first line in the file. In either case, and optionally, the name of the resulting file can be given with the `-o` option such as `perl merge_meshtal.pl` *meshtal1 meshtal2* `-o` *meshtal.out*.

Note that the **merge_meshtal_one** utility that **merge_meshtal.pl** requires is typically built as a part of the overall MCNP build process for utilities. However, this file can also be trivially built with most C++ compilers using a command such as `g++ -o merge_meshtal_one merge_meshtal_one.cpp` or `icpc -o merge_meshtal_one merge_meshtal_one.cpp` for the GNU and Intel C++ compilers, respectively. Regardless, the binary **merge_meshtal_one** utility must be in the user's path so it can be called by **merge_meshtal.pl**.

By default, these utilities keep the mesh-tally file together with all tally numbers present. This behavior can be disabled such that results for individual tally numbers are split into separate files with the `-split` command-line option.

Increased debug-type output can be enabled with the `-debug` command-line option.

E.7.2 Example

If one performs two statistically independent but otherwise identical MCNP calculations using the input shown in Listing E.6 and another modifying the random number generator as shown in Listing E.7 to produce ASCII tally files **meshtal** and **meshtam**, respectively, they can be merged with the command `perl merge_meshtal.pl meshtal meshtam -o merged_meshtal.txt`.

The expected output is shown in Listing E.8.

Listing E.6: merge_meshtal1.mcnp.inp.txt

```
Generate meshtal file to merge with another
1000 10 -9.98207e-1 -100      imp:n=1
9999  0              100      imp:n=0
```

```
100 so 90

mode n
sdef
m10       1001.70c     0.666657 $ Pseudo-Water, Liquid @ 23.15 deg-C
          8016.70c     0.333343 $ Density: 0.998207 g/cc from PNNL-15870, Rev. 1
mt10      lwtr.10t
fmesh14:n geom=xyz origin=-50 -50 -50 imesh=50 iints=50
                                      jmesh=50 jints=50
                                      kmesh=50 kints=50
                                      out=col
c
print
rand gen=2 seed=12345
nps 10000
```

Listing E.7: merge_meshtal2.mcnp.inp.txt

```
rand gen=2 seed=34567
```

Listing E.8: Expected Output from Merging Two ASCII Mesh Tally Files

```
  MESHTAL_FILES:
    meshtal
    meshtam

  MESHTAL_NUMBERS:
    14

 Processing:
    meshtal
    meshtam


  Combination of mesh tally files completed succesfully.

  Output stored in  merged_meshtal.txt


 *** done ***
```

E.7.3 Change Log

This section describes the evolution of **merge_meshtal**.

Version 1.0, July 2007

Original development.

Version 1.1, January 2010

- Extended C++ and Perl compatibility with each other.

Version 1.2, January 2022

- Reformatted README file to Markdown. Migrated content to MCNP manual. Revised wording and added example.
- Added **.pl** file extension.
- Assigned version numbers within utility.

E.8 Parameter Study and Uncertainty Analysis Tool (**mcnp_pstudy.pl**)

The **mcnp_pstudy.pl** utility is a Perl script developed to automate the setup, execution, and collection of results from a series of MCNP code calculations [356]. This tool provides a convenient means of performing parameter studies, total uncertainty analyses, parallel job execution on clusters, stochastic geometry modeling, and other types of calculations where a series of MCNP calculations must be performed with varying problem input specifications.

E.8.1 User Interface

To run this script, the user must have Perl available on their system. All direct interactions with it are performed in a command-line interface.

Syntax for Symbolic Parameters and Options

The **mcnp_pstudy.pl** utility performs various tasks based on a particular syntax it expects to encounter within the MCNP input file. All symbolic parameters, instructions, and options interpreted by **mcnp_pstudy.pl** appear as a general comment card (`C`, see §4.4.3) followed by a space, three at signs (`@`), another space, and the specific command defining an action taken by the code. Listing E.9 shows a variety of ways this syntax can be used.

Listing E.9: Syntax expected by **mcnp_pstudy.pl**

```
1  c @@@ symbol = value
2  c @@@ symbol = value_1 value_2 ... value_n
3  c @@@ symbol = normal n mean standard_deviation
4  c @@@ symbol = lognormal n mean standard_deviation
5  c @@@ symbol = uniform n lower_bound upper_bound
6  c @@@ symbol = beta n alpha beta
7  c @@@ symbol = repeat n
8  c @@@ symbol = ( expression )
9  c @@@ constraint = ( logical-expression )
10 c @@@ tied = list-of-symbols
11 c @@@ options = list-of-command-line-options
12 c @@@ options = list-of-command-line-options  \
13 c @@@           list-of-command-line-options
```

The keywords on the left-hand side of the equalities in Listing E.9 lead to various actions and are defined in the following fashion:

`symbol`	A user-defined symbolic parameter or variable name that can be used elsewhere in the MCNP input file to be substituted by a value, or can be used within an expression such as those shown on lines 8 and 9.
`constraint`	A literal used to define a logical-expression constraining the resulting values of a symbol or a combination of symbols.
`tied`	A literal used to associate multiple symbolic parameters to each other when expanding the full combination of resulting values of the listed symbols.
`options`	A literal used to define various command-line options as an alternative to directly specifying them on the command line.

Some notes on valid syntax on the right-hand side of the equalities in Listing E.9:

- For values defining a `symbol` (lines 1–2), strings with embedded whitespace are not allowed.
- The `normal`, `lognormal`, `uniform`, and `beta` (lines 3–6) represent built-in probability density functions (PDFs) used to sample random values from a respective PDF.
- The `repeat n` option (line 7) creates a list of integers `1..n`. This is a convenient way to create a dummy variable for the purpose of repeating a calculation.
- The `n` integer value (lines 3–7) is the number of values generated by the specified functions. This count can either be a single integer or a previously defined symbol that is also an integer.
- All parameters, including `value`, `mean`, `standard_deviation`, `lower_bound`, `upper_bound`, `alpha`, and `beta`, (lines 1–7) can be defined in terms of any previously defined symbol.
- An `expression` (line 8) may include arithmetic, previously defined symbols, and nested parentheses. The outer set of parentheses is required. The result must be a single scalar value.
- A `logical-expression` (line 9) may include previously defined symbols, nested parentheses, logical operators, and constants. The outer set of parentheses is required. The result must be true or false.
- A `list-of-symbols` (line 10) to tie together symbolic parameters must have the same number of values.
- A `list-of-command-line-options` (line 11–13) may be specified within the input file rather than from the command line. Options are set once, at the beginning of the setup, except for `-inner` and `-outer` which take effect immediately for subsequent parameter specifications. Command line options are discussed in §E.8.1.
- Any of the `c @@@` lines can be continued by ending the line with a backslash character (line 12). The subsequent continuation line(s) start with `c @@@` (line 13).

Command Line Arguments and Options

After defining parameters, expressions, and/or execution options within an MCNP input file, as described in §E.8.1, the typical workflow for using **`mcnp_pstudy.pl`** is:

1. Setup the full set of parameterized MCNP input files as individual cases ready to be executed. The original MCNP input file is retained and all individual cases are written with a unique filename or within a unique subdirectory location.
2. Run or submit each of the cases through the MCNP executable.
3. Collect results of the ensemble of cases after execution has completed. Note that averaging results across various input parameters may not be very meaningful depending on the application.

Execution Mode Flags

Setup jobs

`-setup`	Individual case input files will be created in a directory structure where each case will be placed in a unique directory. The directory and input file naming is specified

	by $JOBDIR/$CASE$N/$JOB, where each component of the full path and name can be set through the -jobdir, -case, and -job command line options described in §E.8.1. The $N value is a sequence of integers, a unique integer for each case, that contains $CASENUM digits padded on the left with 0.
-whisper, -inponly	Individual case input files will be created in the current working directory with a unique input file naming specified by inp_$CASE$N, where the components of this filename can be specified through the -case command line option described in §E.8.1. The $N value is a sequence of integers, a unique integer for each case, that contains $CASENUM digits padded on the left with 0. The run/submit jobs and collect results options described below are incompatible with this execution mode. This option is suitable for other tools, such as Whisper, to make use of the individual cases created by **mcnp_pstudy.pl** in this flattened directory structure.

Only a single setup option should be used on the command line. If the -whisper/-inponly flags are used along with the -setup flag, the code will default to the -whisper/-inponly behavior.

Run/submit jobs

-run	Execute the setup jobs on the current machine.
-rerun	Rerun any jobs that did not run to completion and produce a *case_finished* file.
-submit	Submit the jobs using the SLURM (**sbatch**), MOAB (**msub**), or LSF (**bsub**) job schedulers. SLURM (**sbatch**) is the default.
-resubmit	Resubmit any jobs which did not run to completion and produce a *case_finished* file.
-status	Report status from the file system whether cases are running or completed.

Of the four run/submit options, -run, -rerun, -submit, and-resubmit, only a single option should be used on the command line at once. If multiple run/submit flags are specified at once, only a single one will be executed based on the ordering of the run/submit flags shown above regardless of the ordering of the flags on the command line itself.

The -status flag simply checks on the existence of a sentinel file named *case_finished* within each case directory. At the completion of the MCNP calculation, **mcnp_pstudy.pl** creates this sentinel file solely for the purpose of this status check and/or use of the -rerun and -resubmit flags.

Collect results

-collect	Collect results from each job and average them.
-avgonly	Only print the average results.

The collection of results requires that a **MCTAL** is produced. See PRDMP.

Keyword-value Arguments and Option Flags

Required keyword-value arguments

`-i INFILE`	The template MCNP input file with variable, expression, and/or option directives specified for **mcnp_pstudy.pl**. The only alternative to specifying this keyword-argument pair is to stream in the input file via the standard input (**stdin**).

Optional keyword-value arguments

`-jobdir JOBDIR`	Use directory $`JOBDIR` to store case inputs and results. Can be absolute or relative, and will be created if it does not exist. Default is the local working directory.
`-case CASE`	Use name $`CASE` within case names used to uniquely identify individual jobs. Default is the string `case`.
`-casenum CASENUM`	Use $`CASENUM` digits appended to the case name, e.g., `case0001`, `case0002`, etc., when $`CASENUM` is `4`. Case numbers are padded on the left with `0`. Default is 3 digits.
`-job JOB`	MCNP input filename $`JOB`, created in each directory. Default is the string `inp`.
`-log LOGFILE`	Writes the summary of some of the actions completed and some results collected by **mcnp_pstudy.pl**. Default is `log.txt`.
`-mcnp MCNP`	Used as the run command. $`MCNP` may be the location of the executable or may be a string of commands in quotes, such as `"mpirun -n 4 /usr/local/MCNP6/bin/mcnp6.mpi"`
`-mcnp_opts MCNP_OPTS`	Appended to the MCNP execution command line. May be a series of options in quotes, e.g., `"o=outx tasks 4"`.
`-sbatch_opts SBATCH_OPTS`	Set of options for a **sbatch** command. Only used if the `-submit` option is invoked. By default, the SLURM **sbatch** job scheduler is used for submitting jobs.
`-msub_opts MSUB_OPTS`	Set of options for a **msub** command. Only used if the `-submit` option is invoked. Sets submission command to the MOAB **msub** job scheduler.
`-bsub_opts BSUB_OPTS`	Set of options for a **bsub** command. Only used if the `-submit` option is invoked. Sets submission command to the LSF **bsub** job scheduler.
`-ppn PPN`	Number of processes to run per node (i.e., number of cases to run concurrently per node). Default value is `1`.
`-symlink SYMLINK`	Create symbolic links in each case directory to the listed files. Can use Unix wildcards in filenames. If $`SYMLINK` is more than one file, separate list of files by blanks and use double or single quotes around entire list. Only used during the `-setup` step.
`-cmd_before CMD_BEFORE`	Command(s) to run before running the MCNP code. For Windows, $`CMD_BEFORE` should be a single command, inside double quotes if there are blanks. For Mac/Linux/Cygwin, $`CMD_BEFORE` can be a series of commands separated by semicolons, inside double quotes if there are blanks.

`-cmd_after CMD_AFTER`
: Command(s) to run after the MCNP calculation finishes. Run only if the MCNP calculation completed successfully.

Optional Flags

`-debug`, `-verbose`, `-v`
: Print additional output.

`-outer`, `-inner`
: If `-outer` is selected (the default), then jobs are created for all possible combinations of the parameters defined. The total number of jobs is the product of the number of times each parameter is specified.

  If `-inner` is selected, the defined parameters are substituted serially. The total number of jobs is the maximum number of values of a single parameter across all parameters defined.

Notes and Suggestions on Common Command Line Uses

While the workflow described above mentions three distinct steps, the typical usage of **mcnp_pstudy.pl** can often be accomplished in as few as one or two commands, invoking multiple steps at once. For example, running the command

```
mcnp_pstudy.pl -i inputfile -setup -run -collect
```

will result in all three steps being completed in order. This single command line execution approach assumes that the default MCNP6 executable (**mcnp6**) is found in the users' **PATH** environment variable on the local machine.

Note that the `-collect` step can only be executed once all cases have finished executing. Therefore, when submitting jobs to a cluster using one of the job submission scheduler options, it is likely necessary to separate the setup and execution steps from the collection step. Running the two commands

```
mcnp_pstudy.pl -i inputfile -inner -mcnp 'mpirun -np 72 /usr/local/MCNP6/bin/mcnp6.mpi' \
               -sbatch_opts '-N 2 -t 60' -setup -submit
mcnp_pstudy.pl -i inputfile -inner -collect
```

does effectively separate these steps. Note that the second command to collect results will only be successful if all jobs have completed running. The `-status` flag can be used to quickly check on if the full suite of jobs have completed.

Note that if any job directory, case name, case numbering, job name, inner/outer options are specified in the setup step, they must be consistent with the command line options for the subsequent run/submit and collect results steps if they are done during separate calls to **mcnp_pstudy.pl**. In the second example above, the `-inner` flag is specified in both steps so that the result collection step understands how many cases to collect results from.

E.8.2 Examples

Example 1

The example in Listing E.10 is setup with the intent to study the effect of independently varying both the mass density and the radius of a bare highly enriched uranium sphere that is effectively critical in it's nominal configuration.

Listing E.10: example_pstudy_1.mcnp_pstudy.inp.txt

```
1  Godiva independent density/radius mcnp_pstudy
2  c
3  c @@@ options =  -outer
4  c @@@ options =  -log godiva_pstudy_1.log.txt
5  c
6  c @@@  DEN = 18.  18.74   19.
7  c @@@  RAD = 8.   8.741  9.
8  c
9  c Below is normal MCNP input, with parameters
10 c
11 1    1  -DEN    -1
12 2    0           1
13
14 1    so  RAD
15
16 kcode 1000 1.0 10 40
17 ksrc  0. 0. 0.
18 imp:n 1 0
19 m1    92235 -94.73
20       92238 -5.27
21 prdmp j j 1
```

Some notes on various **mcnp_pstudy.pl**-specific features of the example input file in Listing E.10:

- Lines 3 and 4 contain options that are specified within the input file that could alternatively be specified on the command line.
- The `DEN` and `RAD` variables are defined on lines 6 and 7 as a list of values, each containing three items, which are ultimately substituted into lines 11 and 14, respectively, for each case created.
- Because the `-outer` flag is specified, a total of nine cases will be setup, executed and compared; one for each unique combination of `DEN` and `RAD` values. If the `-inner` flag were specified, a total of three cases would be setup.
- Line 21 contains a proper `PRDMP` card that will produce a **MCTAL** file at the conclusion of each case calculation. With this, the result collection option will work as intended.

To setup, run, and collect the results of all nine cases created from the templated input file in Listing E.10, the

```
1  mcnp_pstudy.pl -i example_pstudy_1.mcnp_pstudy.inp.txt -setup -run -collect
```

command can be used.

Example 2

The example in Listing E.11 is setup with the intent to study the effect of varying the mass density and the radius while preserving the total mass of a bare highly enriched uranium sphere that is effectively critical in it's nominal configuration. Additionally, an independent set of nuclear data libraries are studied for the same combination of selected densities and radii.

Listing E.11: example_pstudy_2.mcnp_pstudy.inp.txt

```
1  Godiva mass-preserving density/radius and library mcnp_pstudy
2  c
3  c @@@ options =  -outer
4  c @@@ options =  -log godiva_pstudy_2.log.txt
5  c
6  c @@@  DEN = 18.  18.74   19.
7  c @@@  RAD = ( 8.741*(18.74/DEN)**.333333 )
8  c
9  c @@@ tied = U235 U238
10 c @@@ U235 = 92235.80c 92235.00c
11 c @@@ U238 = 92238.80c 92238.00c
12 c
13 c Below is normal MCNP input, with parameters
14 c
15 1    1  -DEN    -1
16 2    0           1
17
18 1    so  RAD
19
20 kcode 1000 1.0 10 40
21 ksrc  0. 0. 0.
22 imp:n 1 0
23 m1     U235 -94.73
24        U238 -5.27
25 prdmp j j 1
```

Some notes on various **mcnp_pstudy.pl**-specific features of the example input file in Listing E.11:

- Lines 3 and 4 contain options that are specified within the input file that could alternatively be specified on the command line.
- The `DEN` variable is defined on line 6 as a list of three values, ultimately substituted into line 15.
- The `RAD` variable is defined on line 7 as an expression that requires evaluation based on the `DEN` value for that case. Each evaluated `RAD` value is ultimately substituted into line 18.
- Lines 10 and 11 define two more variables, `U235` and `U238`, containing two material identifiers for two separate nuclear data libraries. The values of these variables for each case will be substituted into lines 23 and 24, respectively.
- Line 9 specifies that the `U235` and `U238` variables are tied to one another such that the first and second entries will always be together, effectively representing two total values.
- Because the `-outer` flag is specified, a total of six cases will be setup, executed and compared; one for each unique combination of the three `DEN` and two `U235`/`U238` values.
- Line 25 contains a proper `PRDMP` card that will produce a **MCTAL** file at the conclusion of each case calculation. With this, the result collection option will work as intended.

To setup, run, and collect the results of the nine cases created from the templated input file in Listing E.11, the

```
mcnp_pstudy.pl -i example_pstudy_2.mcnp_pstudy.inp.txt -setup -run -collect
```

command can be used.

E.8.3 Changelog

Version 1.0, June 2003

- Original version. References LA-UR-04-0499 [356] and LA-UR-04-2506 [356].

Version 1.1, December 2008

- Version released with MCNP5-1.51.

Version 1.2, September 2010

- Version released with MCNP5-1.60.
- Improvements:
  - Allow for perturbed tallies.
- Bug fixes:
  - Minor fixes for parsing tallies.

Version 1.3, April 2013

- Version released with MCNP6.1.
- New features:
  - Added sampling from a Lognormal(μ, σ) probability density function.
  - Added sampling from a Beta(α, β) probability density function.
  - Added ability to run/rerun/submit/resubmit multiple cases per job, running concurrently (`-ppn`)
- Improvements:
  - Allow input lines that start with `#`.
  - Allow tied outer parameters.
  - Added option for status check of **`msub`** and **`bsub`** submitted jobs.
  - Modifications to permit inner/outer specifications for each parameter.
- Bug fixes:
  - Fix to prevent infinite loop if last line has no end-of-line (`\n`) character.

Version 1.4, April 2014

- Version released with MCNP6.1.1beta.
- Improvements:
  - Removed symbolic links for dump files named in input. Can be done explicitly in message block.

Version 1.5, April 2018

- Version released with MCNP6.2.
- New features:
  - New option `-whisper` or `-inponly`, to perform setup of input files in the of form inp_case* in the current directory. This option does not create case* directories. It is incompatible with the various `-run` and `-submit` options.
  - Added **SLURM** capability through the use of `-sbatch_opts` to specify job scheduler parameters. This is the default job scheduler option.
  - Added `-cmd_before` and `-cmd_after` options to specify commands to be executed before and/or after a submitted job.
- Improvements:
  - Check for, and forbid, duplicate symbol definitions. Check for, and forbid, expressions from using any undefined expressions.

Version 1.6, May 2022

- Version released with MCNP6.3.
- New features:
  - Added **README.md** Markdown and extracted user information and changelog content to MCNP manual.
- Improvements:
  - Added **.pl** suffix to utility to identify it as a Perl script.
- Bug fixes:
  - Fixed infinite loop when long inline (`$`) comment is encountered during long-line splitting logic.
  - Fixed title card splitting issue during long-line splitting logic. The new logic will not split any part of the optional message block nor the title card to avoid producing invalid input files.

E.9 Simple ACE File Generation Tools (**simple_ace.pl** and **simple_ace_mg.pl**)

The **simple_ace.pl** and **simple_ace_mg.pl** scripts have been developed to support the generation of simplified continuous-energy and multigroup nuclear data files in A Compact ENDF (ACE) format [357, 358]. In general, these scripts are used to generate nuclear data for use in analytic or semi-analytic verification testing that is useful to ensure that the code implementation is consistent with the underlying particle transport theory.

To run either the **simple_ace.pl** or the **simple_ace_mg.pl** script, the user must have Perl available on their system. All interactions with the scripts are performed in a command-line interface.

E.9.1 Continuous-energy Cross Sections with simple_ace.pl

The **simple_ace.pl** script generates continuous-energy nuclear data ACE-formatted datasets. It only considers capture, fission, and elastic scattering, prompt fission neutrons (no delayed fission), and a discrete delta function for the prompt fission neutron spectrum, $\chi(E)$.

User Interface Command Line Options

`-zaid ZAID`	String name for dataset, following §1.2.3, typically of the form `ZZAAA.nnc`. Default `99999.99c`.
`-file FILE`	String filename for output ACE dataset. Default `ZAID` (see above).
`-awr AWR`	Atomic weight ratio, mass/neutron-mass. Default `1000000.0`.
`-tmp TMP`	Temperature. In units of MeV if `TMP<1`. In units of Kelvin if `TMP>1`. Default is room temperature `2.5301E-8` MeV.
`-comment COMMENT`	Comment to include in ACE file header. Default is `ACE file created by simple_ace.pl`.
`-e ENERGIES`	List of energy points (MeV). Must include ≥ 2 values, provided in increasing order. Default is `1.E-11 100` MeV.
`-t T_XS`	Ignored. Total cross section, σ_t, constructed from `-s`, `-c`, and `-f` values below (1).
`-s S_XS`	List of scattering cross section values, σ_s, elastic only (1).
`-mu MU`	List of average cosine scattering angles, $\bar{\mu}$ (1, 2).
`-s1 S1`	List of P_1 scattering cross sections (1). May be used instead of $\bar{\mu}$ with same restrictions of the `MU` values above (2).
`-c C_XS`	List of capture cross section values, σ_c, not including fission (1).
`-f F_XS`	List of fission cross section values, σ_f (1).
`-echi ECHI`	Single energy (MeV) value for prompt fission spectrum delta function, $\chi(E) = \delta(E-$`ECHI`$)$.
`-nu NU`	List of average fission multiplicity values, $\bar{\nu}$ (1).
`-nloge NLOGE`	List of energy intervals to expand `-e` energy list into equally spaced bins in $log(e)$. Number of values in list must be 1 less than the number of values in the `-e` list. Use linear interpolation for cross section values.

`-broaden`	Flag to Doppler broaden cross sections. Assumes each input scatter cross section is at 0 K, and then Doppler broadens each cross section to `TMP` value, assuming constant cross section approximation. Default is False.

Details:

1. If 0 values supplied, set to 0. If 1 value supplied, use it for all energies. Otherwise, number of values must match the number of values given in the `-e` list of energies.
2. For P_1 scattering, $|\mu| \leq 1/3$. For scattering angles of $|\mu| > 1/3$, results can be seriously incorrect because the scattering probability density function is negative over portions of the scattering angle domain [358].

Example

To create a continuous-energy ACE dataset covering the incident neutron energy range from 10^{-11} to 100 MeV, with uniform capture (σ_c), scattering (σ_s), and fission (σ_f) cross sections, an uniform prompt fission neutron multiplicity ($\bar{\nu}$), and a 1 MeV discrete delta function for prompt neutrons born from fission, the following command line options can be used:

```
simple_ace.pl  -c .019584  -s .225216  -f .0816  -nu 3.24  -e 1e-11 100  -echi 1
```

Information from **simple_ace.pl** is provided to the standard output:

```
======> simple_ace.pl - create special purpose ACE file

             zaid = 99999.99c
             za   = 99999
             file = 99999.99c
              awr = 1000000
              tmp = 2.5301e-08
             echi = 1
       energy pts = 2
         xss size = 60

        e              sigt           sigc           sigs           nu       sigf
       1.0000e-11     3.2640e-01     1.9584e-02     2.2522e-01     3.24    8.1600e-02
       1.0000e+02     3.2640e-01     1.9584e-02     2.2522e-01     3.24    8.1600e-02

       XSDIR Info, to use on XSn card::

       XSn   99999.99c 1e+06        99999.99c  0 1 1 60  0 0 2.5301e-08


       Creating ACE file:  99999.99c
```

To utilize this ACE file within an MCNP calculation, the screen output provides the necessary `XS`n cross section input card. If the user provides a unique `n` value for this particular `XS` input card, the `99999.99c` identifier can be used within a `M` material specification input card. The generated ACE file for this example can be found in Listing E.12.

Listing E.12: Simple ACE File 99999.99c

```
 99999.99c       1e+06 2.53010e-08 2022-06-20
ACE file created by simple_ace.pl                                  2022-06-20
      0   0.000000      0   0.000000      0   0.000000      0   0.000000
      0   0.000000      0   0.000000      0   0.000000      0   0.000000
      0   0.000000      0   0.000000      0   0.000000      0   0.000000
      0   0.000000      0   0.000000      0   0.000000      0   0.000000
       60    99999        2        1        1        0        0        0
        0        0        0        0        0        0        0        0
        1       11       18       19       20       21       22       26
       28       38       39        0        0        0        0        0
        0        0        0        0       57       60        0        0
        0        0        0        0        0        0        0        0
              1e-11              100           0.3264           0.3264
           0.019584         0.019584         0.225216         0.225216
                  0                0                2                0
                  2            1e-11              100             3.24
               3.24               18                0               19
                  1                1                2           0.0816
             0.0816                1                6                2
              1e-11              100                0                0
                  2            1e-11              100                0
                  0                1                0                1
                 10                0                2            1e-11
                100                1                1                0
                  2            1e-11              100                2
           0.999999         1.000001         0.999999         1.000001
                  1                2           0.0816           0.0816
```

E.9.2 Multigroup Cross Sections with `simple_ace_mg.pl`

The **simple_ace_mg.pl** script generates multigroup nuclear data ACE-formatted datasets. It only considers capture, fission, and elastic scattering, prompt fission neutrons (no delayed fission), and isotropic or P_1 scattering distributions.

User Interface Command Line Options

`-zaid ZAID`	String name for dataset, following §1.2.3, typically of the form `ZZAAA.nnm`. Default `99999.99m`.
`-file FILE`	String filename for output ACE dataset. Default `ZAID` (see above).
`-awr AWR`	Atomic weight ratio, mass/neutron-mass. Default `1000000.0`.
`-tmp TMP`	Temperature. In units of MeV if `TMP<1`. In units of Kelvin if `TMP>1`. Default is room temperature `2.5301E-8` MeV.
`-comment COMMENT`	Comment to include in ACE file header. Default is `multigroup ACE file`.
`-groups NG`	Number of energy groups. Default is `1`.
`-e ENERGIES`	List of energy group boundaries (MeV). Must include `NG+1` values, provided in decreasing order. Default is `100 0` MeV (1).

`-t T_XS`	List of total cross section values, σ_t (2).
`-s S_XS`	List of scattering cross section values, σ_s, elastic only (3), (4).
`-s1 S1`	List of P_1 scattering cross section values (3), (4), (5).
`-c C_XS`	List of capture cross section values, σ_c, not including fission (2).
`-f F_XS`	List of fission cross section values, σ_f (2).
`-chi CHI`	List of prompt fission spectrum, χ, values for each energy group (2).
`-nu NU`	List of average fission multiplicity values, $\bar{\nu}$ (2).
`-bins NBINS`	Number of bins to expand P_1 scattering into equiprobable scattering angular distribution. Default `1000`.

Details:

1. If `NG=1`, default energy group structure is `100 0` MeV. If `NG=2`, default energy group structure is `100 0.625E-6 0` MeV. For any other `NG` value, no default group structure provided.
2. If 0 values supplied, set to 0. If 1 value supplied, use it for all energy groups. Otherwise, the number of values entered must match the number of `NG` groups.
3. If 0 values supplied, set to 0. If 1 value supplied, use it for all scattering transition groups. Otherwise, the number of values entered must match the number of scattering transition groups, `NS` = `NG` × `NG`.
4. The group-to-group scatter cross sections must be provided in this order: $1 \rightarrow 1$, $1 \rightarrow 2$, ..., $1 \rightarrow$ `NG`, $2 \rightarrow 1$, $2 \rightarrow 2$, ..., $2 \rightarrow$ `NG`, ... `NG` $\rightarrow 1$, `NG` $\rightarrow 2$, ..., `NG` $\rightarrow$ `NG`, with group 1 being the highest-energy group and group `NG` being the lowest-energy group.
5. For P_1 scattering, $|$`S1/S_XS`$| \leq 1/3$. For scattering angles of $|$`S1/S_XS`$| > 1/3$, results can be seriously incorrect because the scattering probability density function is negative over portions of the scattering angle domain [358].

Example

To create a 2-group multigroup ACE dataset with energy group 1 from `10-100` MeV and energy group 2 from `0-10` MeV, with total (σ_t), capture (σ_c), and fission (σ_f) cross sections, a prompt fission neutron multiplicity ($\bar{\nu}$) and emission spectrum (χ), and an isotropic group-to-group scattering matrix ($\sigma_{s,g\rightarrow g'}$), the following command line options can be used:

```
simple_ace.pl  -zaid 22089.01m    -comment 'la-ur-12-22089 analytic problem 1' \
               -groups 2   -e  100. 10. 0.   -t   2. 3.   -c  .5 1.   -f .5 1. \
               -nu .75 4.5   -chi 1. 0.   -s  .5 .5  0. 1.
```

Information from **simple_ace_mg.pl** is provided to the standard screen output:

```
======> simple_ace_mg.pl - create simple multigroup ACE file

              zaid = 22089.01m
```

```
             za   = 22089
             file = 22089.01m
              awr = 1000000
              tmp = 2.5301e-08
           groups = 2
         xss size = 21
          comment = la-ur-12-22089 analytic problem 1
             date = 2022-06-20

 group      Ehi     Elow       total     capture     scatter     fission          nu          chi
     1      100       10           2         0.5           1         0.5        0.75            1
     2       10        0           3           1           1           1         4.5            0

 scattering matrix,   group-I (down) --> group-J (across)
    J -->           1           2
    I --v
        1         0.5         0.5
        2           0           1

 XSDIR Info, to use on XSn card:

     XSn   22089.01m 1e+06        22089.01m  0 1 1 21  0 0 2.5301e-08


 Creating ACE file:  22089.01m
```

To utilize this ACE file within an MCNP calculation, the screen output provides the necessary `XS`n cross section input card. If the user provides a unique `n` value for this particular `XS` input card, the `22089.01m` identifier can be used within a `M` material specification input card. The generated ACE file for this example can be found in Listing E.13.

Listing E.13: Simple ACE Multigroup File 22089.01m

```
 22089.01m       1e+06 2.53010e-08 2022-06-20
la-ur-12-22089 analytic problem 1                                    2022-06-20
      0   0.000000      0   0.000000      0   0.000000      0   0.000000
      0   0.000000      0   0.000000      0   0.000000      0   0.000000
      0   0.000000      0   0.000000      0   0.000000      0   0.000000
      0   0.000000      0   0.000000      0   0.000000      0   0.000000
       21    22089        0        0        2        1        1        0
        0        1        0        1        0        0        0        0
        1        5        7        9       11       13        0        0
        0        0        0        0       15        0        0       20
       21        0        0        0        0        0        0        0
        0        0        0        0        0        0        0        0
                55                5               90               10
                 2                3              0.5                1
              0.75              4.5                1                0
               0.5                1               16              0.5
               0.5                0                1                0
                 0
```

E.10 Unstructured Mesh Format Converter (**um_convert**)

Deprecation Notice **DEP-53421**

The **um_convert** application is deprecated because of the deprecation of the MCNPUM file format [DEP-53424].

The *um_convert* (unstructured mesh convertor) program is a command-line utility program that takes the information in the Abaqus mesh input file and processes it with the UM input processing routines from REGL to produce the internal data structures that MCNP6 needs. The data from these internal data structures are written to a new file type, `MCNPUM` [DEP-53424], that MCNP6 can quickly read before launching into calculations. With the `MCNPUM` file type the UM input processing start up penalty need not happen every time the UM geometry is required. This can save substantial time for large mesh geometries that are used repeatedly. Details on the structure of this file and its contents are best learned from looking at the source code.

E.10.1 Command Line Options

To be reminded of *um_convert*'s functionality and to see the command line options, enter the following at the command line prompt:

```
um_convert_op --help
```

Note, your path must include the path to the program. A message similar to the following should appear in the command window:

```
** UNSTRUCTURED MESH CONVERSION PROGRAM **
Functions:
1) Convert ABAQUS inp file to mcnpum file
Command Line Arguments:
-h,   --help         summary of features & arguments
-b,   --binary       create mcnpum in binary format
-a,   --abaqus       ABAQUS input file                -- (1)
-l,   --length       length conversion factor
-o,   --output       um_convert output file name
-t,   --threads      number of threads
-um,  --mcnpum       mcnpum output file name
```

The -b Option

This argument (`-b, --binary`) requests that the `MCNPUM` file be created as a binary file instead of ASCII. ASCII is default and results if this option is not specified.

The -a Option

This argument (`-a, --abaqus`) followed by the file name of the Abaqus mesh input file communicates this information to the utility program. This information is required.

The -l Option

This argument (`-l, --length`) followed by a value provides a conversion factor for all dimensions in a similar fashion to the length parameter on the `EMBED` card.

The -o Option

This argument (`-o, --output`) followed by a file name tells the utility program where to write messages and information from the file conversion process. The information that MCNP6 would normally print to its `outp` file when building the unstructured mesh model is written to this file. This argument is optional. If no name is specified, the information is written to the `um_convert.out` file.

The -t Option

This argument (`-t, --threads`) followed by a number sets the number of OpenMP threads for use in the conversion process. The user should be careful and not oversubscribe threads by requesting too large of a number. (See Section E.10.2). This is an optional argument. The default value is 1.

The -um Option

This argument (`-um, --mcnpum`) followed by a file name tells the utility program what to call the `MCNPUM` file [DEP-53424] that it generates. If no name is specified, the information is written to the `um_convert.mcnpum` file.

E.10.2 Program Execution and Example

The *um_ convert* utility is a highly parallelized program that can be compiled to use MPI processes, OpenMP threads, and vectorized loops. As a note to those wishing to build the code on their systems from the source, the following is the appropriate command line (using the traditional MCNP6 make system) that will build the code with MPI processes, OpenMP threads, and vectorized loops once the mainline MCNP6 code has been built:

```
make depends build CONFI="intel openmpi omp" FC_OPT="-O3" GNUJ=4
```

Normal execution of *um_ convert* from the command line will result in messages similar to the following appearing in the command window:

```
 UM_CONVERT input processing begins.       11- 9-2015 @  9:46:31
```

```
  Max threads available:     16
  Global Tracking Model Complete
  Element Neighbors Found
  Part Cell Surfaces Complete
  SKD-Trees Build Complete
  Element Connectivity Complete
  um_convert execution time       19.6 sec
UM_CONVERT input processing ends.         11- 9-2015 @  9:46:50
```

Note that the program provides the user with the maximum number of available threads. The product of the number of MPI processes and the number of threads, specified with the -t Option, should not exceed the number of cpu cores present or performance will be degraded.

A combination of MPI processes and OpenMP threads should produce the shortest execution times on most systems. If the user doesn't have MPI available (e.g., a desktop Windows machine), executing with the maximum number of available threads should still produce acceptable execution times. The utility will process one part / instance at a time, using all of the requested threads as it needs them.

If the user is running on a Linux cluster where MPI has been installed, a combination of MPI processes and OpenMP threads is recommended. As always, performance is contingent on the number of parts / instances in the Abaqus mesh input file. If there are more cpu cores available than parts / instances, then specifying one MPI process for each part / instance with several threads per process is recommended. If fewer MPI processes are specified than parts / instances, then *um_ convert* will give each process a number of parts / instances to work on in a sequential fashion much like MCNP6 does with its parallel processing of parts / instances. In this later scenario where there are more parts / instances than cpu cores, it may be beneficial to reduce the number of MPI processes so that each process has two threads. This should help when one or a few parts have substantially more elements than the other parts.

Unlike MCNP6 where the manager MPI process basically functions as a controller during the calculational phase, all MPI processes in the *um_ convert* utility have a chunk of the parts / instances with which to work.

As a reminder when using MPI processes and OpenMP threads together on certain Linux clusters, the `mpi_paffinity_alone` and `bynode` switches (or their equivalent) may be necessary when using **mpirun** to ensure that threads are assigned to the correct hardware.

E.11 Unstructured Mesh Post-processing (**um_post_op**)

Deprecation Notice **DEP-53423**

The **um_post_op** application is deprecated because the legacy EEOUT ASCII and binary files upon which it operates is deprecated [DEP-53294].

The *um_post_op* (unstructured mesh post operations) program is a utility program that performs various manipulations on MCNP6's elemental edit output file, `EEOUT` [DEP-53294]. This program is written in Fortran and uses various routines and data structures from the Revised Extended Grid Library (REGL) in order to maintain consistency with MCNP6. Like MCNP6, *um_post_op* is designed to run from the command line. Current supported features include adding and merging multiple `EEOUT` files into one, converting binary files to ASCII, generating Visualization TooKit *(VTK)* visualization files, creating instance-based pseudo-tallies, writing a single edit to a file, and generating error histograms for those edits with errors. Some of these features support the processing of multiple files with one command.

E.11.1 Command Line Options

To be reminded of *um_post_op's* functionality and to see the command line options, enter the following at the command line prompt:

```
um_post_op --help
```

Note, your path must include the path to the program. A message similar to the following should appear:

```
** UTILITY PROGRAM FOR UNSTRUCTURED MESH EEOUT FILE **
Functions:




1) add many eeout files into one
2) merge many eeout files into one
3) convert binary files into ascii files
4) generate vtk files for VisIt visualization
5) generate pseudo-tallies by pseudo-cell
6) write a single edit to an ascii file
7) generate a histogram of edit errors




Command Line Arguments:
-h,   --help        summary of features & arguments
-a,   --add         add multiple files (no weighting)
-m,   --merge       merge multiple files
-o,   --output      single output file name
```

```
-p,   --pos         value range for wse and wsep
-bc,  --binconvert  convert binary file to ascii
-eh,  --errorhist   generate a histogram of edit errors
-ex,  --extension   multiple output file extension
-ta,  --tally       pseudo-tallies from file
-vtk, --vtkfile     generate ascii visualization file
-wse, --writesedit  write a single edit to file
```

Mutually Exclusive Options

This utility program has seven mutually exclusive options: merging (-m) many files into one ASCII file, adding (-a) many files together into one ASCII file, converting (-bc) any number of binary files into ASCII files, generating VTK files (-vtk) for visualization, generating pseudo-tallies (-ta) for instances, writing a single edit (-wse) to an ASCII file, and generating a histogram of edit errors (-eh) for those edits that have errors. Only one of these options may be requested at a time.

The -o and -ex Options

The output file name (-o, --output) and extension name (-ex, --extension) options are intended to be mutually exclusive. The user should receive error messages if both of these arguments appear on the same command line. However, one or the other must be used. The output file name is intended for use when there is one `EEOUT` file to manipulate or many files that are to be merged into one. The extension name is pre-appended with a period, '.', and then appears as the suffix to the input file name(s) when new files must be created after processing many input files (e.g., converting many files from ASCII to binary). The first argument following these arguments is interpreted as either the output file name or the extension name.

E.11.2 Examples

Merging Files

The original intent for this utility program was to establish a means of merging many `EEOUT` files into one file. These many files are expected to be from independent runs of a problem so that results are weighted by the number of histories in the file. This differs from adding files where there is no history weighting.

When the *um_post_op* utility is given a list of files to merge into one, it reads the header information (that includes number of nodes, materials, instances, tetrahedra, pentahedra, hexahedra) and checks the consistency of this header information for each subsequent file against the first file. For all files other than the first one, a message about that consistency is output to the terminal window. Without consistency among the files, the utility program can not make a meaningful and successful merge.

If there is only one file specified for merging, the program will print out an error message and stop. Since one file is created from many, the output file name argument is required.

Example command line:

```
um_post_op -m -o my_merge_file eeout1 eeout2 ... eeoutN
```

Note that the first argument after the -o argument is interpreted as the output file name.

At this time, the output file that is generated is ASCII, even if all of the input files are binary. The input files may be any mixture of ASCII or binary.

Adding Files

This capability provides a means of adding (or collecting) many `EEOUT` files into one file. These many files are expected to be from different calculational runs on the same mesh geometry; results are NOT weighted by the number of histories in the file. Rather, already normalized results are simply added together. This differs from merging files where there is history weighting. For example, this capability is useful if there are different runs because independent sources were used in different calculations and there is a need for the results to be combined.

Cautions and restrictions discussed under the merging files section apply here and are not repeated.

Example command line:

```
um_post_op -a -o my_add_file eeout1 eeout2 ... eeoutN
```

Note that the first argument after the -o argument is interpreted as the output file name.

Converting Files

This capability allows the conversion of `EEOUT` files from binary format to ASCII. In performing this operation there is a loss of precision since all double precision reals are written with only six significant digits. Currently, there is no capability to convert from ASCII to binary.

On the command line, one or many files may be specified for conversion. When many files are requested for conversion, there is no consistency check performed as there is when merging files since that is a meaningless action for this option.

When the conversion request asks for only one file, the -o argument may be used. Example command line:

```
um_post_op -bc -o eeout.ascii eeout.binary
```

It is also legitimate to use the -ex argument. Example command line:

```
um_post_op -bc -ex ascii eeout.binary
```

The resulting output file is named: eeout.binary.ascii

When more than one file is to be converted, the -ex argument must be used. Example command line:

```
um_post_op -bc -ex asc eeout1 eeout2 ... eeoutN
```

The resulting files appear with the names

```
eeout1.asc eeout2.asc ... eeoutN.asc
```

Creating Visualization Files

This capability should generate files in the VTK format for visualization from `EEOUT` files. The geometry data and the edit information is taken from the `EEOUT` file and reformatted to be consistent with version 4.2 of the *VTK* standard and written to an ASCII file. Details on the *VTK* file format and requirements can be found in the *VTK* documentation, available on the worldwide web and in text books.

On the command line, one or many files may be specified for conversion to the VTK format. When many files are requested for conversion, there is no consistency check performed as there is when merging is requested since that is a meaningless action for this option.

When the generation request asks for only one file, the `-o` argument may be used.

Example command line:

```
um_post_op -vtk -o eeout.vtk eeout1
```

It is also legitimate to use the -ex argument. Example command line:

```
um_post_op -vtk -ex vtk eeout1
```

The resulting output file is named: eeout1.vtk

When more than one file is to be generated, the -ex argument must be used. Example command line:

```
um_post_op -vtk -ex vtk eeout1 eeout2 ... eeoutN
```

The resulting files appear with the names

```
eeout1.vtk eeout2.vtk ... eeoutN.vtk
```

Note that while it is possible to specify any file extension or output file name for the *VTK* file, some visualization programs will not recognize it as such unless there is a *VTK* extension.

Note that this capability has not received extensive testing and may not be supported in the future.

Generating Pseudo-Tallies

This capability will generate a pseudo-tally for each pseudo-cell from the corresponding edit and write the results to an output file (see example at the end of this section). If no output file is specified, the output is written to a file named "fort.1001". These tallies are volume weighted according to the following equation:

$$tally_i = \frac{\sum_{n=1}^{N} vol_n \cdot edit_n}{\sum_{n=1}^{N} vol_n}$$

where

$tally_i$	tally for pseudo-cell i form corresponding edit
vol_n	volume of element n
$edit_n$	edit result of element n
N	total number of elements in i

These results are termed pseudo-tallies since they are equivalent to an MCNP tally averaged over a cell (*i.e.,* F4, F6, F7), but do not have an associated statistical uncertainty, tally fluctuation chart, etc. Note that these pseudo-tallies are over pseudo-cells.

On the command line one or many files may be specified for pseudo-tally creation. When many files are requested for pseudo-tally creation, there is no consistency check performed as there is when merging files since that is a meaningless action for this option.

When the conversion request asks for only one file, the -o argument may be used.

Example command line:

```
um_post_op -ta -o eeout.tally eeout.binary
```

It is also legitimate to use the -ex argument.

Writing A Single Edit To A File

This capability allows the user to write the edit results from a single edit in the `EEOUT` file (see example at the end of this section) to an ASCII file that is reformatted with detailed information. For each element in the problem (`EEOUT` file) the information that is available with each edit result is element number, element type number, material number, density, volume, and centroid location. The utility of this file is left to the imagination of the user. Results are ordered by increasing element number.

This request requires that an edit number be specified with the *um_post_op* command line argument, -wse or --writesedit; this number should be the argument immediately following this keyword argument. The correct edit number can be found in the output from the pseudo-tally option (see example at the end of this section for edit numbers in blue font), described previously. Since an edit my contain multiple energy, time, and particle bins, using the internal edit number requires less input on the *um_post_op* command line.

Example command line:

```
um_post_op -wse 1 -o eeout.wse eeout1
```

It is also legitimate to use the -ex argument.

It is possible to filter the output for this capability using the -p or --pos arguments. If the value following this argument is 1 or +1, only values greater than zero are included in the edit. Conversely, if the value following the argument is -1, only values less than or equal to zero are included. If a real value is specified instead of the integers just described, its value is the decision point with the sign of the value indicating whether the filter provides values greater than (+) or less than or equal to (−).

Example command line requesting to see all results less than or equal to 0.005.

```
um_post_op -wse 1 -p -5.e-3 -o eeout.wse eeout1
```

Writing A Single Edit To A File By Position

This capability is similar to that discussed in the previous section, except that the output is ordered by increasing position (i.e., x, y, z location). The appropriate arguments to use on the command line are: -wsep or --writeseditpos. Value filtering, as described in the previous section, works the same way with this capability.

Generating A Histogram Of Edit Errors

This capability allows the user to write error histograms to an output file for all of the edits in the `EEOUT` file for which errors were requested (see example at the end of this section). If no output file is specified, the output is written to a file named "fort.1001". The number of histogram bins can be specified directly after the -eh command line option. The default value is 10 if none is specified. The error bins are defined such that the smallest error is assigned to the first bin and the largest error is assigned to last bin. Bins are evenly spaced between the first and the last bins. Relative error values are in the range of 0 to 1, inclusive. See §8.4 for more details on errors and the `EEOUT` file.

The essential header information from the `EEOUT` file is written at the beginning of the error histogram file. Following this information, there is a section for each edit for which errors were requested. There is a description of each edit. In each section following the edit description, there are results by pseudo-cell and results over all mesh in the model. For each group of results there is the minimum and maximum errors on the edit in addition to a table with the error histogram. For each row in the histogram table there is the upper limit for the error bin, the absolute number of elements that fall into this bin, the relative percentage these elements represent of the total, and the cumulative percentage of the current row and all preceding rows.

Example command line specifying a table with 20 bins:

```
um_post_op -eh 20 -o my_error_histogram eeout1
```

It is also legitimate to use the -ex argument.

Miscellaneous

The REGL routine that reads valid `EEOUT` files [DEP-53294] has the ability to detect whether the file it is reading is ASCII or binary. If it can't make a determination that the file is a valid `EEOUT` file, an error message appears in the terminal window. Therefore, when a list of files is specified on the command line, for either merging, adding, or generating VTK files, they may be a mixture of ASCII or binary.

Example Pseudo-Tally File

What follows is not a complete example. Only enough details are provided to illustrate the main points.

```
Pseudo-tallies for eeout file via um_post_op
Eeout file: eeout1007

Created on    :  4- 3-2012 @  9: 0:37

Prob ID       : simple cube, each element is a statistical set, 8 total
Calling Code  : MCNP6
Inp File      : inp1007
Outp File     : inp1007o
Runtpe File   : inp1007r
Geom Inp File : um1007.inp

NUMBER OF NODES    :                   27
NUMBER OF MATERIALS:                    1
NUMBER OF INSTANCES:                    1
NUMBER OF 1st TETS :                    0
NUMBER OF 1st PENTS:                    0
NUMBER OF 1st HEXS :                    8
NUMBER OF 2nd TETS :                    0
NUMBER OF 2nd PENTS:                    0
NUMBER OF 2nd HEXS :                    0
NUMBER OF COMPOSITS:                    1
NUMBER OF HISTORIES:                 1000
NUMBER OF REG EDITS:                   19
NUMBER OF COM EDITS:                    9

------------------------------------------------------------------------
EDIT:    1 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_1_FLUX_14

 Energy Bin Boundary:  1.00000E+36  Energy Bin Multiplier:  1.00000E+00
 Time Bin Boundary  :  1.00000E+33  Time Bin Multiplier  :  1.00000E+00

Instance  Name                                 Volume        Result
--------  ----                                 ------        ------
       1  simple_cube-1                    1.00000E+03   4.77743E-02

------------------------------------------------------------------------
```

```
EDIT:    2 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_1_ENERGY_36

 Energy Bin Boundary:  2.00000E+00  Energy Bin Multiplier:  1.00000E+00
 Time Bin Boundary  :  1.00000E+00  Time Bin Multiplier  :  1.00000E+00

Instance  Name                                  Volume        Result
--------  ----                                  ------        ------
       1  simple_cube-1                     1.00000E+03   8.12612E-03

 ---------------------------------------------------------------------------
EDIT:    3 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_2_ENERGY_36

 Energy Bin Boundary:  1.00000E+10  Energy Bin Multiplier:  1.00000E+00
 Time Bin Boundary  :  1.00000E+00  Time Bin Multiplier  :  1.00000E+00

Instance  Name                                  Volume        Result
--------  ----                                  ------        ------
       1  simple_cube-1                     1.00000E+03   7.54778E-03

---------------------------------------------------------------------------
 EDIT:    4 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_2_ENERGY_BIN_1_ENERGY_36

 Energy Bin Boundary:  2.00000E+00  Energy Bin Multiplier:  1.00000E+00
 Time Bin Boundary  :  1.00000E+39  Time Bin Multiplier  :  1.00000E+00

Instance  Name                                  Volume        Result
--------  ----                                  ------        ------
       1  simple_cube-1                     1.00000E+03   7.84947E-03

 ---------------------------------------------------------------------------
EDIT:    5 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_2_ENERGY_36

 Energy Bin Boundary:  1.00000E+10  Energy Bin Multiplier:  1.00000E+00
 Time Bin Boundary  :  1.00000E+39  Time Bin Multiplier  :  1.00000E+00

Instance  Name                                  Volume        Result
--------  ----                                  ------        ------
       1  simple_cube-1                     1.00000E+03   2.26368E-03
```

Example Single Edit File

```
Write single edit for eeout file via um_post_op
Eeout file: eeout1007

Created on    :  4- 3-2012 @ 12:11:25

Prob ID       : simple cube, each element is a statistical set, 8 total
Calling Code  : MCNP6
Inp File      : inp1007
Outp File     : inp1007o
Runtpe File   : inp1007r
Geom Inp File : um1007.inp

NUMBER OF NODES    :                   27
NUMBER OF MATERIALS:                    1
NUMBER OF INSTANCES:                    1
NUMBER OF 1st TETS :                    0
NUMBER OF 1st PENTS:                    0
NUMBER OF 1st HEXS :                    8
NUMBER OF 2nd TETS :                    0
NUMBER OF 2nd PENTS:                    0
NUMBER OF 2nd HEXS :                    0
NUMBER OF COMPOSITS:                    1
NUMBER OF HISTORIES:                 1000
NUMBER OF REG EDITS:                   19
NUMBER OF COM EDITS:                    9

 ---------------------------------------------------------------
EDIT:    1 ::  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_1_FLUX_14

 Energy Bin Boundary:  1.00000E+36  Energy Bin Multiplier:  1.00000E+00
 Time Bin Boundary  :  1.00000E+33  Time Bin Multiplier  :  1.00000E+00

 ------------------------------------------------------------------------------------------------------------
Element   Type  Material     Density       Volume                      Centroid                      Result
                                                                 X    Y             Z
--------  ----  --------   -----------   -----------  ------------  ------------  ------------   -----------
       1    6          1   1.87401E+01   1.25000E+02  -2.50000E+00  -2.50000E+00   7.50000E+00   4.50075E-02
       2    6          1   1.87401E+01   1.25000E+02  -2.50000E+00   2.50000E+00   7.50000E+00   4.71156E-02
       3    6          1   1.87401E+01   1.25000E+02  -2.50000E+00  -2.50000E+00   2.50000E+00   4.99385E-02
```

```
        4    6          1   1.87401E+01   1.25000E+02  -2.50000E+00   2.50000E+00   2.50000E+00   4.99248E-02
        5    6          1   1.87401E+01   1.25000E+02   2.50000E+00  -2.50000E+00   7.50000E+00   4.59879E-02
        6    6          1   1.87401E+01   1.25000E+02   2.50000E+00   2.50000E+00   7.50000E+00   5.14196E-02
        7    6          1   1.87401E+01   1.25000E+02   2.50000E+00  -2.50000E+00   2.50000E+00   4.33516E-02
        8    6          1   1.87401E+01   1.25000E+02   2.50000E+00   2.50000E+00   2.50000E+00   4.94486E-02
```

Example Error Histogram File

```
Write error histograms for eeout file via um_post_op
Eeout file: block01_6part_6type.eeout

Created on    :  3-11-2014 @ 13: 8:21

Prob ID       : block01  8x8x6  6 parts, 6 element types
Calling Code  : MCNP6_DEVEL
Code Version  : 6-1-02
Date & Time   :  03/11/14 12.43.38
Inp File      : block01mgv1
Outp File     : outy
Runtpe File   : runtpn
Geom Inp File : job_block_6part_6type_01.inp

NUMBER OF NODES    :                 1258
NUMBER OF MATERIALS:                    6
NUMBER OF INSTANCES:                    6
NUMBER OF 1st TETS :                   30
NUMBER OF 1st PENTS:                    8
NUMBER OF 1st HEXS :                  128
NUMBER OF 2nd TETS :                   29
NUMBER OF 2nd PENTS:                    8
NUMBER OF 2nd HEXS :                  128
NUMBER OF COMPOSITS:                    0
NUMBER OF HISTORIES:              1000000
NUMBER OF REG EDITS:                    2
NUMBER OF COM EDITS:                    0

---------------------------------------------------------------
EDIT:  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_1_FLUX_4
```

```
    Energy Bin Boundary:  1.00000E+10  Energy Bin Multiplier:  1.00000E+00
    Time Bin Boundary  :  1.00000E+39  Time Bin Multiplier  :  1.00000E+00

    ---------------------------------------------------
    Results for Instance #     1  :: part-end_quad_hex-1

    Minmum Error       :  1.64393E-02
    Maximum Error      :  1.70379E-02
    Bin Width          :  2.99308E-05

    ------   --------   --------   --------    ----------
    Bin        Upper     Absolute   Relative    Cumulative
    Number     Bound      Number      (%)         (%)
    ------   --------   --------   --------    ----------
       1   1.6469E-02          1     0.7812        0.7812
       2   1.6499E-02          1     0.7812        1.5625
       3   1.6529E-02          3     2.3438        3.9062
       4   1.6559E-02          5     3.9062        7.8125
       5   1.6589E-02          0     0.0000        7.8125
       6   1.6619E-02          7     5.4688       13.2812
       7   1.6649E-02          6     4.6875       17.9688
       8   1.6679E-02         14    10.9375       28.9062
       9   1.6709E-02          5     3.9062       32.8125
      10   1.6739E-02          6     4.6875       37.5000
      11   1.6769E-02         13    10.1562       47.6562
      12   1.6798E-02         14    10.9375       58.5938
      13   1.6828E-02         12     9.3750       67.9688
      14   1.6858E-02         11     8.5938       76.5625
      15   1.6888E-02          5     3.9062       80.4688
      16   1.6918E-02         10     7.8125       88.2812
      17   1.6948E-02          4     3.1250       91.4062
      18   1.6978E-02          7     5.4688       96.8750
      19   1.7008E-02          3     2.3438       99.2188
      20   1.7038E-02          1     0.7812      100.0000

(Results for instances 2 through 6 were removed to make this example shorter.)

    ---------------------
    Results Over All Mesh

    Minmum Error       :  9.33224E-03
```

```
    Maximum Error      :  1.95299E-02
    Bin Width          :  5.09881E-04

    ------   --------   --------   --------    ----------
    Bin       Upper     Absolute   Relative    Cumulative
    Number    Bound      Number       (%)         (%)
    ------   --------   --------   --------    ----------
       1   9.8421E-03          4     1.2085        1.2085
       2   1.0352E-02          8     2.4169        3.6254
       3   1.0862E-02          0     0.0000        3.6254
       4   1.1372E-02          0     0.0000        3.6254
       5   1.1882E-02          4     1.2085        4.8338
       6   1.2392E-02          1     0.3021        5.1360
       7   1.2901E-02          0     0.0000        5.1360
       8   1.3411E-02          3     0.9063        6.0423
       9   1.3921E-02          3     0.9063        6.9486
      10   1.4431E-02          9     2.7190        9.6677
      11   1.4941E-02          9     2.7190       12.3867
      12   1.5451E-02          4     1.2085       13.5952
      13   1.5961E-02          0     0.0000       13.5952
      14   1.6471E-02         15     4.5317       18.1269
      15   1.6980E-02        241    72.8097       90.9366
      16   1.7490E-02         18     5.4381       96.3746
      17   1.8000E-02          5     1.5106       97.8852
      18   1.8510E-02          6     1.8127       99.6979
      19   1.9020E-02          0     0.0000       99.6979
      20   1.9530E-02          1     0.3021      100.0000
```

E.12 Unstructured Mesh Pre-processing (**um_pre_op**)

Deprecation Notice **DEP-53422**

The **um_pre_op** application is deprecated because its main capabilities have been duplicated elsewhere.

The skeleton-input-file generation functionality has been duplicated in an easier-to-maintain and distribute form such as that described in [359]. Elemental quality assessment is now a standard part of MCNP UM input processing, which is controlled using the `elementchk` argument on the `EMBED` card.

The lattice-conversion capability is not duplicated elsewhere. People interested in continuing to use these features should send an email to mcnp_help@lanl.gov.

The *um_pre_op* (unstructured mesh pre operations) program is a utility program that performs various manipulations on input designed to aid in problem setup with the unstructured mesh (UM). This program is written in Fortran and uses various routines and data structures from the Revised Extended Grid Library (REGL) in order to maintain consistency with MCNP6. Like MCNP6, *um_pre_op* is designed to run from the command line. Current supported features include creating a skeleton MCNP6 input deck (`-m`) from the Abaqus/CAE mesh input file, converting a simple lattice-voxel geometry (`-lc`) to an Abaqus mesh input file, volume checking (`-vc`) the finite element volumes, and element checking (`-ec`) the mesh input file for twisted and/or deformed elements. As with *um_post_op*, there is limited error handling.

E.12.1 Command Line Options

To be reminded of *um_pre_op's* functionality and to see the command line options, enter the following at the command line prompt:

```
um_pre_op --help
```

Note, your path must include the path to the program. A message similar to the following should appear:

```
** PRE-PROCESSOR PROGRAM FOR UM CAPABILITY **



Functions:
1) Create MCNP input file from Abaqus .inp file
2) Convert MCNP simple lattice to Abaqus .inp file
3) Volume check the Abaqus .inp file and pseudo-cells
4) Element check the Abaqus .inp file



Command Line Arguments:
-b,   --back        background material for input file
```

```
-h,   --help        summary of features & arguments
-m,   --mcnp        generate MCNP skeleton input file --(1)
-o,   --output      output file name
```

```
 -cf,  --controlfile  file with lattice conversion controls
-dc,  --datacards    data cards file to include
-ex,  --extension    output file extension
-ff,  --fillfile     file with lattice fill description
-lc,  --latconvert   convert simple lattice to Abaqus -- (2)
-vc,  --volcheck     volume check the .inp file       -- (3)
-ec,  --elementcheck element check the .inp file      -- (4)
-len, --length       scale factor for mesh dimensions
```

Mutually Exclusive Options

Currently, this utility program has four mutually exclusive options: generating (-m) a skeleton MCNP6 input file, converting (`-lc`) a simple lattice-voxel geometry to an Abaqus mesh input file, volume checking (`-vc`) the finite element volumes, and element checking (`-vc`) for twisted and/or deformed elements.

The -o and -ex Options

The output file name (`-o, --output`) and extension name (`-ex, --extension`) options are intended to be mutually exclusive. The user should receive error messages if both of these arguments appear on the same command line. However, one or the other must be used except where indicated in the following feature discussions. If the `-o` argument is present then the output is placed in a file with the name (or argument) that immediately follows on the command line. If the `-ex` argument is present, then the output is placed in a file with a name built from the input file name followed by a period, '.', and the argument immediately following on the command line.

The -b Option

The -b option is currently only used with the -m option to specify a background cell material number. See the discussion below for more information.

The -len Option

The -len option is currently only used with the -lc option. This is a scale factor to apply to dimensions from the lattice mesh file.

E.12.2 Examples

Generating an MCNP6 Input File

A skeleton MCNP6 input file can be created from the Abaqus mesh input file using the `-m` option. The name of the input file to be created is set with either the `-o` or `-ex` options. The intent of this option is to make it easier for users to get up and running with the unstructured mesh capability and not necessarily to generate a fully functional input file. The degree to which a fully functional input deck can be generated depends upon the completeness and correctness of the data card file provided with the `-dc` option.

The *um_pre_op* program can read the Abaqus mesh input file and generate a global mesh model just as if MCNP6 was performing this function. The information in the global mesh model is then used to create the appropriate pseudo-cell cards, background cell, and minimal CSG world to hold the mesh universe plus the `embed` control card for the data section. If more than a minimal CSG structure is required outside the mesh universe, the user must create this by hand.

If the `-b` option is not specified on the command line to supply a valid material number from the Abaqus mesh input file, *um_pre_op* will make the background cell void. If an invalid material number (i.e., a number for a material that is not defined in the Abaqus mesh input file) is specified with the `-b` option, *um_pre_op* will default to making the background cell void. At this time the `-b` option only works with the `-m` option.

When using the `-m` option it is possible to read a data cards file, `-dc` argument, for inclusion in the new MCNP6 input file. The *um_pre_op* program scans the data cards file for existing cards. For each particle on an existing and active mode card, a default flux edit (`EMBEE` card) is specified and written to the new input file. If active `IMP` cards are present in the data cards file, they are written to the new input file, otherwise *um_pre_op* creates default `IMP` cards for each particle present on the mode card. If an active `SDEF` card is present in the data cards file, it is written to the new input file, otherwise a skeleton `SDEF` card is written provided volume source elsets are present in the Abaqus mesh input file. All other cards in the data cards file, regardless if they are active cards or comments, are written to the new input file.

Note: At this point the material numbers for the material definitions in the data cards file should be consistent with those used in the Abaqus mesh input file. This may be the biggest source of error for some users.

Example command line with data cards argument and the `-b` argument to use material 7 from the Abaqus mesh input file as the background material for the mesh universe:

```
um_pre_op --mcnp -o newinput abaqus.inp -dc dc_cards -b 7
```

Converting a Simple Lattice Geometry

Simple lattice geometries in MCNP6 that use the `fill` parameter along with the `lat` parameter on a cell card can be converted to an Abaqus mesh input file for use with the -m option described previously or for viewing as an orphan mesh geometry in Abaqus. This lattice geometry is described as simple in that each voxel should have a homogenous structure since each voxel is converted to a first order hexahedra with a homogeneous material assignment.

For this feature, two input files are required and the mesh input file must be specified using the `-o` option; the `-ex` option is invalid here. In addition, a file named `lat2abq.summary` is created that contains details about the conversion process. The first of the two input files must contain only the fill information as it appears with the `fill` parameter on the MCNP6 lattice cell card. A short example is given in Listing E.14.

This is known as the fill file and is specified to *um_pre_op* with the `-ff` option. Any attempt to put other information is this file will undoubtedly cause *um_pre_op* to terminate in an unfriendly manner.

Listing E.14: Example fill file.

```
1 19R
2 7r  3 11R
2 2 4 2r
2 2 4 2r
3 4 3R 3 4 3R
```

The second of the two required input files is the control file and is specified to the *um_pre_op* program with the `-cf` option. An example is provided in Listing E.15.

Listing E.15: Example control file

```
Jacksonville 1000 x 1000 x 31 model; 1 meter resolution
Deltas 100 100 100
fill 0:999 0:999 0:30
Origin center
#
universe 1 -1.25000E-03 air
universe 2 -0.05 ext_building
universe 3 -0.01 int_building
universe 4 -1.2 ground
universe 5 -0.01 int_garage
universe 6 -0.087058 ext_garage
universe 7 -0.00125 air
#
exclude 1
extents 0 999 0 999 0 0
hints 200 200 50
threshhold 1
```

As can be seen from the description of this file that follows, there are a number of parameters that can be adjusted for this feature, making it tedious to implement and use as command line options.

The first line in the control file is the title line. The line is required, must be the first line in the file, and can contain 256 characters of information. This line is inserted in the Abaqus mesh input file on the line after the `*Heading` parameter at the beginning of the file. This is the line that is used for the MCNP6 input file title line if the *um_pre_op* -m option is invoked.

Any line after the first line with either a #, %, or $ in the first column is treated as a comment line by *um_pre_op* and ignored.

All of the other parameters for this feature are implemented with a set of keywords where the keyword appears at the beginning of the line before any values. The keywords do not need to start in the first column; they can be either upper case, lower case, or a mixture of both. Most keywords have default values. Those that do not have defaults are required keywords and should contain meaningful data.

The `deltas` keyword is required. Three values are needed that specify the length of the voxels in centimeters along the x, y, and z directions. These values will be used to size the hexahedra. All hexahedra will have these dimensions.

The `fill` keyword is required. Three sets of values for the x, y, and z directions are needed in the same format that MCNP6 requires for this keyword on the lattice cell card. Each set consists of two lattice

locations separated by a colon. The value to the left of the colon is the smallest index for that direction (for *um_pre_op* this value should be 0) while the value to the right of the colon is the largest index for that direction. The values specified for the `fill` keyword should be the full extents of the problem described in the fill file. A subset of this geometry can be specified with the `extents` parameter described below.

The `universe` keyword is required. There may be as many universes specified on separate lines in the control file as needed to fully describe the problem. For the sake of *um_pre_op* and converting a lattice description to an equivalent unstructured mesh equivalent, the concept of a universe is more restrictive than what MCNP6 allows in general. As stated above, each voxel in the lattice must be homogeneous so that one material can be assigned to it. Therefore, the universe numbers double as material numbers. If the universe and material numbers don't coincide in the existing description, it is up to the user to ensure that they do coincide (are identical). If the user wishes to convert a more complex voxel lattice to unstructured mesh, the complex voxels must be homogenized.

Three values are required for each `universe` keyword. The first is the universe number. There should be one for every universe number that is used in the fill file. The universe numbers will be used as the material numbers when describing the material elsets in the Abaqus mesh input file. There is no default value for the universe number; so valid input is required. The second value for the `universe` keyword is the material density (either number or physical). This value will be written to the pseudo-cell cards if *um_pre_op* is used with the `-m` option on this file. The third value for this keyword, is the universe / material name that can contain as many as 128 alphanumeric characters. This name is used in creating material and part names. More information on the parts created in this process can be found in the discussion for the `hints` keyword.

The following keywords are optional.

The `exclude` keyword is optional. It contains a single integer instructing *um_pre_op* to exclude the specified universe number from any of the parts. In the Fig. E.15 example, `universe 1` is part of the simple lattice, but because it is air that we don't want MCNP6 to track through as a mesh, we exclude it. This can save computation time, but will not let the program accumulate results on a mesh in these locations. When excluding any universe, it is probably a good idea to set the background material for the mesh universe to this material; see the `-b` option in conjunction with the `-m` option..

The `extents` keyword is optional and is used to select a contiguous extent of the lattice specified from the `fill` keyword. Default values are 0, but any values specified are taken to apply in the order lower x-index, upper x-index, lower y-index, upper y-index, lower z-index, and upper z-index.

The `hints` keyword is optional, but highly recommended since values associated with this keyword set the overall size of segments and parts. Three values, one for each direction, are permitted with the default for each being 9999999. The values are not physical units, but rather the number of columns (X), rows (Y), or planes (Z). Since MCNP6 input processing for parts in the unstructured mesh can be time intensive if the parts have more than ~50,000 elements, it is best to segment any geometry, whether it comes from a lattice or not, into smaller pieces. The values associated with this keyword provide guidance to *um_pre_op* in order to create these segments. *um_pre_op* will construct segments that are close to the size specified. Each segment has a set of i-j-k indices that describes its location from the lower left hand corner in the overall geometry. Once the segments are defined, the program can create parts from the segments. All elements with the same material are lumped together into a part whose name is derived from the i-j-k indices, the material number, and the material name. For example, a part composed of the material "ground" with an associated material number of 4 and possessing i-j-k indices of 2, 3, 1 would be given the name: `part_2_3_1_4_ground`.

The `origin` keyword is optional and is used to adjust the location of the mesh origin. If this keyword is not included, the origin defaults to 0 0 0, otherwise it is shifted to the value specified. An X Y Z location can be specified, or as a convenience, the characters `CENTER` may be input. With `CENTER` specified, the program calculates the problem's center based upon the overall extents specified with the `fill` and `deltas` keywords. Any triples of values causes the origin to shift to that location.

The `threshold` keyword is optional. It contains a single value instructing *um_pre_op* to make parts when the number of elements in the part exceed the value specified. The default value is 1. It is always a good idea to create parts with more than 1 element.

The information in the `lat2abq.summary` file is fairly self-explanatory. The information in this file can help the user set or adjust values for the hints keyword, among other things. It was decided that it was more appropriate to write this information to a file rather than to the terminal screen.

Example command line to convert a simple lattice geometry:

```
um_pre_op -lc -o geomlattice.inp -ff fillfile -cf control
```

Volume Checking

This option enables the user to check the finite element volumes (against a value) and obtain volumes and masses for the pseudo-cells. Results are printed to a file specified with either the `-o` or `-ex` options. See the results from the example file at the end of this section.

Any value appearing on the command line after the `-vc` argument is treated as the test value. If this value is positive, *um_pre_op* will print out all elements and their corresponding volumes that are greater than or equal to the specified value. If this value is negative, all elements and their corresponding volumes that are less than or equal to the specified value are printed. If no value follows the `-vc` argument, the test is for volumes less than or equal to zero.

Once the volume checks are performed on all of the finite elements, *um_pre_op* calculates the volumes and masses for all of the pseudo-cells. Masses are based on the densities that are present in the Abaqus mesh input file. This information appears in the output file after the element listing from the finite element volume check. After this, a list of the instance names appears followed by a list of the material names and associated densities.

Example command line to find all finite elements with a volume less than or equal to 15:

```
um_pre_op -vc -15 -o vc.out simple_cube_warped.inp
```

Element Checking

This option enables the user to check the Abaqus mesh input file for deformed and/or twisted elements (an example is shown in Fig. E.8) by calculating the determinant of the Jacobian at the appropriate Gauss points and at all node locations that define the finite element. Normal elements (i.e., not deformed or twisted) will have a positive Jacobian indicating that each point (finite volume) in the master space is mapped to an appropriate point (finite volume) in the global space (where some of the tracking algorithms operate). However, very small positive values indicate distortion in the mapping. With appropriate positive Jacobians, the volumes and masses will be correct (as modeled) and there should be no problem with the particle transport.

If a failed element is found (negative Jacobian) during the execution of this option, the global element number and appropriate location information are written to the terminal screen. This same information as well as the results for the Jacobian evaluation at each Gauss and node point are written to the file specified with either the `-o` or `-ex` options. Note that the information is organized by instance. See the results from the example

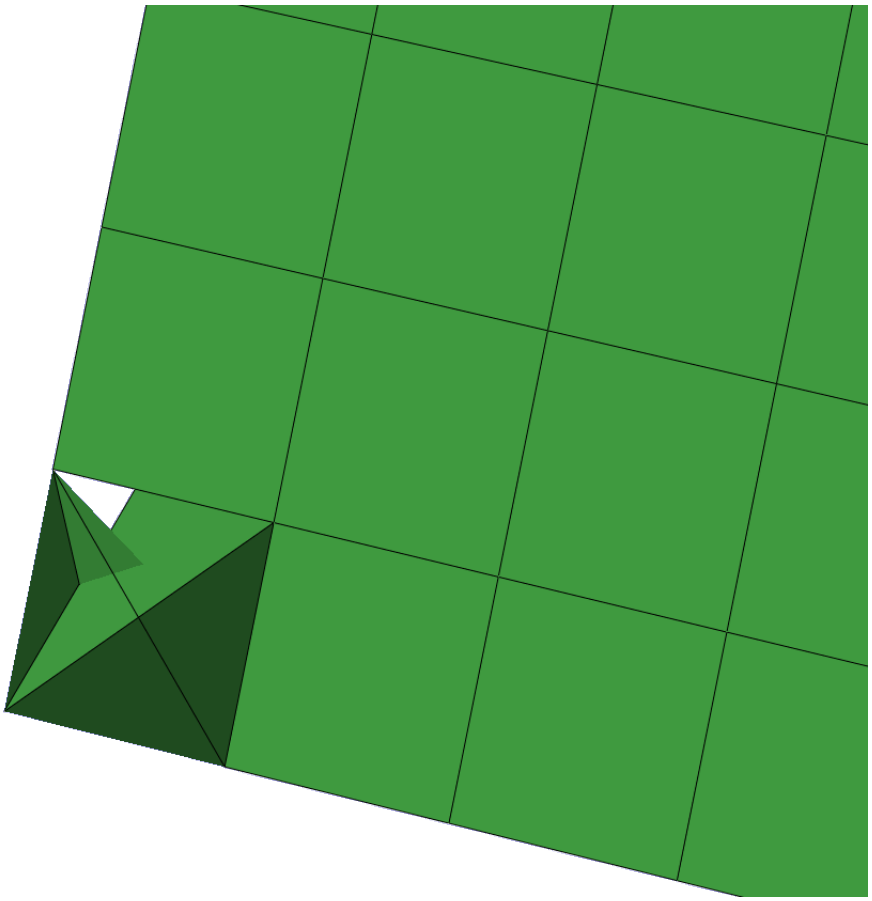


Figure E.8: Example twisted first-order tetrahedra.

file at the end of this section. It is the user's responsibility to fix the problem mesh with the appropriate meshing tool.

Example command line:

```
um_pre_op -ec -o warped.out simple_cube_warped.inp
```

Example Volume Check File

```
 simple warped cube
-
-  Data from file    : simple_cube_warped.inp
-  Created on        :  1-17-2014 @ 14: 0:58
-
    ---------------------------------------
    - Volume Check For Value  1.50000E+01 -
    ---------------------------------------

    Element       Volume
    -------    ------------
          1     7.81250E+00

    Elements with volumes <=  1.50000E+01 :        1

    ----------------------------------
    - Pseudo-Cell Volumes and Masses -
    ----------------------------------

    Cell    Instance    Part    Material    Denisty      Volume Mass
    ----    --------    ----    --------    -------   -----------   -----------
       1           1       1           1   -8.95000   9.99219E+03   8.94301E+04

     ------------------------------------------------------------------------
    Instance    Name
    --------    ------------------------------------------------------------
          1    simple_cube-1

    ------------------------------------------------------------------------
    Material    Denisty    Name
    --------    -------    ------------------------------------------------
           1   -8.95000    material-copper_01
           2   -2.25000    material-graphite_02
```

Example Element Check File

```

  --------------------------------
  - Checking Elements By Instance -
  --------------------------------

```

```
  Number  Name
------  ----------------------------------------------------------------------------------------
     1  part-cube-1

         Element:          2   failed.    Centroid:    1.50000E+00   5.00000E-01   1.50000E+00
                                            Nodes:        X   Y             Z
                                                1    2.00000E+00   1.00000E+00   1.00000E+00
                                                2    2.00000E+00   0.00000E+00   1.00000E+00
                                                3    2.00000E+00   0.00000E+00   2.00000E+00
                                                4    2.00000E+00   1.00000E+00   2.00000E+00
                                                5    1.00000E+00   1.00000E+00   1.00000E+00
                                                6    1.00000E+00   0.00000E+00   1.00000E+00
                                                7    1.00000E+00   1.00000E+00   2.00000E+00
                                                8    1.00000E+00   0.00000E+00   2.00000E+00

                                                    Determinate Values At Gauss Points
                                                            Gauss Points           Jacobian
                                                1    -0.57735  -0.57735  -0.57735   1.14E-01
                                                2     0.57735  -0.57735  -0.57735   1.14E-01
                                                3     0.57735   0.57735  -0.57735   8.33E-02
                                                4    -0.57735   0.57735  -0.57735   8.33E-02
                                                5    -0.57735  -0.57735   0.57735   8.33E-02
                                                6     0.57735  -0.57735   0.57735   8.33E-02
                                                7     0.57735   0.57735   0.57735  -3.05E-02
                                                8    -0.57735   0.57735   0.57735  -3.05E-02

                                                Determinate Values At Master Space Nodes
                                                            Gauss Points           Jacobian
                                                1    -1.00000  -1.00000  -1.00000   1.25E-01
                                                2     1.00000  -1.00000  -1.00000   1.25E-01
                                                3     1.00000   1.00000  -1.00000   1.25E-01
                                                4    -1.00000   1.00000  -1.00000   1.25E-01
                                                5    -1.00000  -1.00000   1.00000   1.25E-01
                                                6     1.00000  -1.00000   1.00000   1.25E-01
                                                7     1.00000   1.00000   1.00000  -1.25E-01
                                                8    -1.00000   1.00000   1.00000  -1.25E-01

 Total number of failed elements:      1
```

Appendix F

Response Functions

This appendix presents response functions that are appropriate for use on the DE and DF tally cards to convert from calculated particle flux to quantities of interest. Section F.1 provides several biological dose equivalent rates and Section F.2 provides data on material damage.

These sets of conversion factors are not the only ones in existence, nor are they recommended by this publication. Rather, they are presented only for convenience. The original publication cited and other sources of this information should be consulted to determine if they are appropriate for your application.

Be aware that conversion factor sets are subject to change based on the actions of various national and international organizations such as the National Council on Radiation Protection and Measurements (NCRP), the International Commission on Radiological Protection (ICRP), the International Commission on Radiation Units and Measurements (ICRU), the American National Standards Institute (ANSI), and the American Nuclear Society (ANS). Changes may be based on the reevaluation of existing data and calculations or on the availability of new information.

In addition to biological dose factors, a reference is given for silicon displacement kerma factors for potential use in radiation-effects assessment of electronic semiconductor devices. The use of these factors is subject to the same caveats stated above for biological dose rates.

For these response functions, ASCII files containing DE/DF cards that can be used with the READ card are electronically attached to this document for convenience to ease data retrieval, subsequent processing, and eventual use. Tabulated values and representative plots of the response functions given in the attachments are also provided here. Instructions on how to extract the response functions from this document can be found in the Preface (page 23).

⚠ Caution

The attached DE/DF cards cannot be directly used in an MCNP input. A tally number must be added. The omission of a tally number and therefore invalid attached input is intentional to ensure the person using the values has interacted with (and thought about) them and is not using them blindly.

F.1 Biological Conversion Factors

In the following discussions, dose rate will be used interchangeably with biological dose equivalent rate. The neutron quality factors implicit in the conversion factors are also tabulated for reference. For consistency with the original publication and to enable direct comparison with original sources, all conversion factors are given in the units they are published as. The interpolation mode chosen should correspond to that recommended

by the reference. For example, the ANSI/ANS publication recommends log-log interpolation; significant differences at interpolated energies can result if a different interpolation scheme is used (e.g., Figs. F.1 and F.26).

F.1.1 Incident Neutron

The ANSI/ANS-6.1.1-1977 neutron flux-to-dose conversion and quality factors are given in Listing F.1, which can be directly used as MCNP input for DE/DF cards. These flux-to-dose conversion factors are also plotted in Figure F.1 showing both linear and logarithmic interpolation. These values are extracted from [360] with permission of the publisher, the American Nuclear Society.

The ANSI/ANS-6.1.1-1991 standard provides a variety of neutron fluence-to-dose conversion factors assuming four irradiation-phantom orientations: anterior-posterior (AP), posterior-anterior (PA), lateral (LAT), and rotational (ROT). More details on these factors, and how to use them, are available in [361]. The AP, PA, LAT, and ROT responses are given in Listings F.2, F.3, F.4, and F.5, respectively, which can be directly used as MCNP input for DE/DF cards. In addition, the conversion factors are plotted in Figures F.2, F.3, F.4, and F.5.

The ICRP/21-1973 neutron fluence-to-dose conversion and quality factors are given in Listing F.6, which can be directly used as MCNP input for DE/DF cards. These values are modified from the original values in [362]. The values in Listing F.6 are the inverse of the original values. In addition, Listing F.6 includes extra significant figures in order to reconstruct the original values in [362].

These fluence-to-dose conversion factors are plotted in Figure F.6 showing both linear and logarithmic interpolation.

Similar to ANSI/ANS-6.1.1-1991, the ICRP/74-1996 standard provides a variety of neutron fluence-to-dose conversion factors assuming six irradiation-phantom orientations: anterior-posterior (AP), posterior-anterior (PA), left-lateral (LLAT), right-lateral (RLAT), rotational (ROT), and isotropic (ISO). For more information, please see [363]. The AP, PA, LLAT, RLAT, ROT, and ISO responses are given in Listings F.7, F.8, F.9, F.10, F.11, F.12, respectively. Similarly to before, these can be directly used as MCNP input for DE/DF cards. These conversion factors are plotted in Figures F.7, F.8, F.9, F.10, F.11, and F.12 respectively.

In addition, ICRP/74-1996 provides ambient and personal dose equivalent fluence-to-dose conversion factors assuming different orientations relative to an ICRU sphere and slab [363]. These are given in Listings F.13, F.14, F.15, F.16, F.17, F.18, and F.19, which are shown in Figures F.13, F.14, F.15, F.16, F.17, F.18, and F.19, respectively.

Similar to ICRP/74-1996, the ICRP/116-2010 standard provides a variety of neutron fluence-to-dose conversion factors assuming six irradiation-phantom orientations: anterior-posterior (AP), posterior-anterior (PA), left-lateral (LLAT), right-lateral (RLAT), rotational (ROT), and isotropic (ISO). For more information, please see [364]. The AP, PA, LLAT, RLAT, ROT, and ISO responses are given in Listings F.20, F.21, F.22, F.23, F.24, F.25, respectively. Similarly to before, these can be directly used as MCNP input for DE/DF cards. These conversion factors are plotted in Figures F.20, F.21, F.22, F.23, F.24, and F.25 respectively.

Listing F.1: Neutron_ANSIANS-611-1977_dedf.txt

```
c
c ANSI/ANS-6.1.1-1977, from Table 1:
c
c  Energy      Flux-to-dose Conversion Factor     Quality Factor
c   [MeV] [(rem/hr)/(cm$^{-2}\cdot$s$^{-1}$)]             [None]
#    de:n                                df:n
      log                                 log
   2.5e-8                             3.67e-6   $            2.0
   1.0e-7                             3.67e-6   $            2.0
   1.0e-6                             4.46e-6   $            2.0
   1.0e-5                             4.54e-6   $            2.0
   1.0e-4                             4.18e-6   $            2.0
   1.0e-3                             3.76e-6   $            2.0
     0.01                             3.56e-6   $            2.5
      0.1                             2.17e-5   $            7.5
      0.5                             9.26e-5   $           11.0
      1.0                             1.32e-4   $           11.0
      2.5                             1.25e-4   $            9.0
      5.0                             1.56e-4   $            8.0
      7.0                             1.47e-4   $            7.0
     10.0                             1.47e-4   $            6.5
     14.0                             2.08e-4   $            7.5
     20.0                             2.27e-4   $            8.0
c
```

Listing F.2: Neutron_ANSIANS-611-1991_Anterior-Posterior_AP_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Anterior-Posterior (AP), from Table 4:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   2.5e-8                               4.0
   1.0e-7                               4.4
   1.0e-6                              4.82
   1.0e-5                              4.46
   1.0e-4                              4.14
   1.0e-3                              3.83
     0.01                              4.53
     0.02                              5.87
     0.05                              10.9
      0.1                              19.8
      0.2                              38.6
      0.5                              87.0
      1.0                             143.0
      1.5                             183.0
      2.0                             214.0
      3.0                             264.0
      4.0                             300.0
      5.0                             327.0
      6.0                             347.0
      7.0                             365.0
      8.0                             380.0
     10.0                             410.0
     14.0                             480.0
c
```

Listing F.3: Neutron_ANSIANS-611-1991_Posterior-Anterior_PA_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Posterior-Anterior (PA), from Table 4:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   2.5e-8                             2.6
   1.0e-7                             2.7
   1.0e-6                            2.81
   1.0e-5                            2.78
   1.0e-4                            2.63
   1.0e-3                            2.49
     0.01                            2.58
     0.02                            2.79
     0.05                            3.64
      0.1                            5.69
      0.2                             8.6
      0.5                            30.8
      1.0                            53.5
      1.5                            85.8
      2.0                           120.0
      3.0                           174.0
      4.0                           215.0
      5.0                           244.0
      6.0                           265.0
      7.0                           283.0
      8.0                           296.0
     10.0                           321.0
     14.0                           415.0
c
```

Listing F.4: Neutron_ANSIANS-611-1991_Lateral_LAT_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Lateral (LAT), from Table 4:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   2.5e-8                               1.3
   1.0e-7                               1.4
   1.0e-6                              1.43
   1.0e-5                              1.33
   1.0e-4                              1.27
   1.0e-3                              1.19
     0.01                              1.27
     0.02                              1.46
     0.05                              2.14
      0.1                              3.57
      0.2                              6.94
      0.5                              18.7
      1.0                              33.3
      1.5                              52.1
      2.0                              71.8
      3.0                             105.0
      4.0                             131.0
      5.0                             151.0
      6.0                             167.0
      7.0                             181.0
      8.0                             194.0
     10.0                             218.0
     14.0                             280.0
c
```

Listing F.5: Neutron_ANSIANS-611-1991_Rotational_ROT_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Rotational (ROT), from Table 4:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   2.5e-8                               2.3
   1.0e-7                               2.4
   1.0e-6                              2.63
   1.0e-5                              2.48
   1.0e-4                              2.33
   1.0e-3                              2.18
     0.01                              2.41
     0.02                              2.89
     0.05                               4.7
      0.1                              8.15
      0.2                              15.3
      0.5                              38.8
      1.0                              65.7
      1.5                              93.7
      2.0                             120.0
      3.0                             162.0
      4.0                             195.0
      5.0                             219.0
      6.0                             237.0
      7.0                             253.0
      8.0                             266.0
     10.0                             292.0
     14.0                             365.0
c
```

Listing F.6: Neutron_ICRP21-1973_dedf.txt

```
c
c ICRP/21-1973, from Table 4, with Modifications:
c
c  Energy       Flux-to-dose Conversion Factor     Quality Factor
c   [MeV] [(mrem/hr)/(cm$^{-2}\cdot$s$^{-1}$)]             [None]
#    de:n                                 df:n
      log                                  log
   2.5e-8                             3.846e-3   $            2.3
   1.0e-7                             4.167e-3   $            2.0
   1.0e-6                             4.546e-3   $            2.0
   1.0e-5                             4.348e-3   $            2.0
   1.0e-4                             4.167e-3   $            2.0
   1.0e-3                             3.704e-3   $            2.0
     0.01                             3.571e-3   $            2.0
      0.1                              0.02083   $            7.4
      0.5                              0.07143   $           11.0
      1.0                               0.1176   $           10.6
      2.0                               0.1429   $            9.3
      5.0                               0.1471   $            7.8
     10.0                               0.1471   $            6.8
     20.0                               0.1538   $            6.0
     50.0                               0.1639   $            5.0
    100.0                               0.1786   $            4.4
    200.0                               0.1961   $            3.8
    500.0                               0.2778   $            3.2
    1.0e3                               0.4545   $            2.8
    2.0e3                                0.625   $            2.6
    3.0e3                               0.7143   $            2.5
c
```

Listing F.7: Neutron_ICRP74-1996_Anterior-Posterior_AP_dedf.txt

```
c
c ICRP/74-1996, Anterior-Posterior (AP), from Table A.41:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                             df:n
      log                              log
   1.0e-9                            5.24
   1.0e-8                            6.55
   2.5e-8                             7.6
   1.0e-7                            9.95
   2.0e-7                            11.2
   5.0e-7                            12.8
   1.0e-6                            13.8
   2.0e-6                            14.5
   5.0e-6                            15.0
   1.0e-5                            15.1
   2.0e-5                            15.1
   5.0e-5                            14.8
   1.0e-4                            14.6
   2.0e-4                            14.4
   5.0e-4                            14.2
   1.0e-3                            14.2
   2.0e-3                            14.4
   5.0e-3                            15.7
     0.01                            18.3
     0.02                            23.8
     0.03                            29.0
     0.05                            38.5
     0.07                            47.2
      0.1                            59.8
     0.15                            80.2
      0.2                            99.0
      0.3                           133.0
      0.5                           188.0
      0.7                           231.0
      0.9                           267.0
      1.0                           282.0
      1.2                           310.0
      2.0                           383.0
      3.0                           432.0
      4.0                           458.0
      5.0                           474.0
      6.0                           483.0
      7.0                           490.0
      8.0                           494.0
      9.0                           497.0
     10.0                           499.0
     12.0                           499.0
     14.0                           496.0
     15.0                           494.0
     16.0                           491.0
     18.0                           486.0
     20.0                           480.0
     30.0                           458.0
     50.0                           437.0
     75.0                           429.0
    100.0                           429.0
    130.0                           432.0
    150.0                           438.0
    180.0                           445.0
c
```

Listing F.8: Neutron_ICRP74-1996_Posterior-Anterior_PA_dedf.txt

```
c
c ICRP/74-1996, Posterior-Anterior (PA), from Table A.41:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                             3.52
   1.0e-8                             4.39
   2.5e-8                             5.16
   1.0e-7                             6.77
   2.0e-7                             7.63
   5.0e-7                             8.76
   1.0e-6                             9.55
   2.0e-6                             10.2
   5.0e-6                             10.7
   1.0e-5                             11.0
   2.0e-5                             11.1
   5.0e-5                             11.1
   1.0e-4                             11.0
   2.0e-4                             10.9
   5.0e-4                             10.7
   1.0e-3                             10.7
   2.0e-3                             10.8
   5.0e-3                             11.6
     0.01                             13.5
     0.02                             17.3
     0.03                             21.0
     0.05                             27.6
     0.07                             33.5
      0.1                             41.3
     0.15                             52.2
      0.2                             61.5
      0.3                             77.1
      0.5                            103.0
      0.7                            124.0
      0.9                            144.0
      1.0                            154.0
      1.2                            175.0
      2.0                            247.0
      3.0                            308.0
      4.0                            345.0
      5.0                            366.0
      6.0                            380.0
      7.0                            391.0
      8.0                            399.0
      9.0                            406.0
     10.0                            412.0
     12.0                            422.0
     14.0                            429.0
     15.0                            431.0
     16.0                            433.0
     18.0                            435.0
     20.0                            436.0
     30.0                            437.0
     50.0                            444.0
     75.0                            459.0
    100.0                            477.0
    130.0                            495.0
    150.0                            514.0
    180.0                            535.0
c
```

Listing F.9: Neutron_ICRP74-1996_L-Lateral_LLAT_dedf.txt

```
c
c ICRP/74-1996, L-Lateral (LLAT), from Table A.41:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                             1.68
   1.0e-8                             2.04
   2.5e-8                             2.31
   1.0e-7                             2.86
   2.0e-7                             3.21
   5.0e-7                             3.72
   1.0e-6                             4.12
   2.0e-6                             4.39
   5.0e-6                             4.66
   1.0e-5                              4.8
   2.0e-5                             4.89
   5.0e-5                             4.95
   1.0e-4                             4.95
   2.0e-4                             4.92
   5.0e-4                             4.86
   1.0e-3                             4.84
   2.0e-3                             4.87
   5.0e-3                             5.25
     0.01                             6.14
     0.02                             7.95
     0.03                             9.74
     0.05                             13.1
     0.07                             16.1
      0.1                             20.1
     0.15                             25.5
      0.2                             30.3
      0.3                             38.6
      0.5                             53.2
      0.7                             66.6
      0.9                             79.6
      1.0                             86.0
      1.2                             99.8
      2.0                            153.0
      3.0                            195.0
      4.0                            224.0
      5.0                            244.0
      6.0                            261.0
      7.0                            274.0
      8.0                            285.0
      9.0                            294.0
     10.0                            302.0
     12.0                            315.0
     14.0                            324.0
     15.0                            328.0
     16.0                            331.0
     18.0                            335.0
     20.0                            338.0
c
```

Listing F.10: Neutron_ICRP74-1996_R-Lateral_RLAT_dedf.txt

```
c
c ICRP/74-1996, R-Lateral (RLAT), from Table A.41:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                             df:n
      log                              log
   1.0e-9                             1.36
   1.0e-8                              1.7
   2.5e-8                             1.99
   1.0e-7                             2.58
   2.0e-7                             2.92
   5.0e-7                             3.35
   1.0e-6                             3.67
   2.0e-6                             3.89
   5.0e-6                             4.08
   1.0e-5                             4.16
   2.0e-5                              4.2
   5.0e-5                             4.19
   1.0e-4                             4.15
   2.0e-4                              4.1
   5.0e-4                             4.03
   1.0e-3                              4.0
   2.0e-3                              4.0
   5.0e-3                             4.29
     0.01                             5.02
     0.02                             6.48
     0.03                             7.93
     0.05                             10.6
     0.07                             13.1
      0.1                             16.4
     0.15                             21.2
      0.2                             25.6
      0.3                             33.4
      0.5                             46.8
      0.7                             58.3
      0.9                             69.1
      1.0                             74.5
      1.2                             85.8
      2.0                            129.0
      3.0                            171.0
      4.0                            198.0
      5.0                            217.0
      6.0                            232.0
      7.0                            244.0
      8.0                            253.0
      9.0                            261.0
     10.0                            268.0
     12.0                            278.0
     14.0                            286.0
     15.0                            290.0
     16.0                            293.0
     18.0                            299.0
     20.0                            305.0
     30.0                            324.0
     50.0                            358.0
     75.0                            397.0
    100.0                            433.0
    130.0                            467.0
    150.0                            501.0
    180.0                            542.0
c
```

Listing F.11: Neutron_ICRP74-1996_Rotational_ROT_dedf.txt

```
c
c ICRP/74-1996, Rotational (ROT), from Table A.41:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                             df:n
      log                              log
   1.0e-9                             2.99
   1.0e-8                             3.72
   2.5e-8                              4.4
   1.0e-7                             5.75
   2.0e-7                             6.43
   5.0e-7                             7.27
   1.0e-6                             7.84
   2.0e-6                             8.31
   5.0e-6                             8.72
   1.0e-5                              8.9
   2.0e-5                             8.92
   5.0e-5                             8.82
   1.0e-4                             8.69
   2.0e-4                             8.56
   5.0e-4                              8.4
   1.0e-3                             8.34
   2.0e-3                             8.39
   5.0e-3                             9.06
     0.01                             10.6
     0.02                             13.8
     0.03                             16.9
     0.05                             22.7
     0.07                             27.8
      0.1                             34.8
     0.15                             45.4
      0.2                             54.8
      0.3                             71.6
      0.5                             99.4
      0.7                            123.0
      0.9                            144.0
      1.0                            154.0
      1.2                            173.0
      2.0                            234.0
      3.0                            283.0
      4.0                            315.0
      5.0                            335.0
      6.0                            348.0
      7.0                            358.0
      8.0                            366.0
      9.0                            373.0
     10.0                            378.0
     12.0                            385.0
     14.0                            390.0
     15.0                            391.0
     16.0                            393.0
     18.0                            394.0
     20.0                            395.0
     30.0                            395.0
     50.0                            404.0
     75.0                            422.0
    100.0                            443.0
    130.0                            465.0
    150.0                            489.0
    180.0                            517.0
c
```

Listing F.12: Neutron_ICRP74-1996_Isotropic_ISO_dedf.txt

```
c
c ICRP/74-1996, Isotropic (ISO), from Table A.41:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                             df:n
      log                              log
   1.0e-9                              2.4
   1.0e-8                             2.89
   2.5e-8                              3.3
   1.0e-7                             4.13
   2.0e-7                             4.59
   5.0e-7                              5.2
   1.0e-6                             5.63
   2.0e-6                             5.96
   5.0e-6                             6.28
   1.0e-5                             6.44
   2.0e-5                             6.51
   5.0e-5                             6.51
   1.0e-4                             6.45
   2.0e-4                             6.32
   5.0e-4                             6.14
   1.0e-3                             6.04
   2.0e-3                             6.05
   5.0e-3                             6.52
     0.01                              7.7
     0.02                             10.2
     0.03                             12.7
     0.05                             17.3
     0.07                             21.5
      0.1                             27.2
     0.15                             35.2
      0.2                             42.4
      0.3                             54.7
      0.5                             75.0
      0.7                             92.8
      0.9                            108.0
      1.0                            116.0
      1.2                            130.0
      2.0                            178.0
      3.0                            220.0
      4.0                            250.0
      5.0                            272.0
      6.0                            282.0
      7.0                            290.0
      8.0                            297.0
      9.0                            303.0
     10.0                            309.0
     12.0                            322.0
     14.0                            333.0
     15.0                            338.0
     16.0                            342.0
     18.0                            345.0
     20.0                            343.0
c
```

Listing F.13: Neutron_ICRP74-1996_H10Phi_dedf.txt

```
c
c ICRP/74-1996, $H^{*}(10)/Phi$, from Table A.42:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                               6.6
   1.0e-8                               9.0
  2.53e-8                              10.6
   1.0e-7                              12.9
   2.0e-7                              13.5
   5.0e-7                              13.6
   1.0e-6                              13.3
   2.0e-6                              12.9
   5.0e-6                              12.0
   1.0e-5                              11.3
   2.0e-5                              10.6
   5.0e-5                               9.9
   1.0e-4                               9.4
   2.0e-4                               8.9
   5.0e-4                               8.3
   1.0e-3                               7.9
   2.0e-3                               7.7
   5.0e-3                               8.0
     0.01                              10.5
     0.02                              16.6
     0.03                              23.7
     0.05                              41.1
     0.07                              60.0
      0.1                              88.0
     0.15                             132.0
      0.2                             170.0
      0.3                             233.0
      0.5                             322.0
      0.7                             375.0
      0.9                             400.0
      1.0                             416.0
      1.2                             425.0
      2.0                             420.0
      3.0                             412.0
      4.0                             408.0
      5.0                             405.0
      6.0                             400.0
      7.0                             405.0
      8.0                             409.0
      9.0                             420.0
     10.0                             440.0
     12.0                             480.0
     14.0                             520.0
     15.0                             540.0
     16.0                             555.0
     18.0                             570.0
     20.0                             600.0
     30.0                             515.0
     50.0                             400.0
     75.0                             330.0
    100.0                             285.0
    130.0                             260.0
    150.0                             245.0
    175.0                             250.0
    201.0                             260.0
```

```
63 C
```

Listing F.14: Neutron_ICRP74-1996_H_textrmpslab100circPhi_dedf.txt

```
c
c ICRP/74-1996, $H_{\textrm{p,slab}}(10,0^{\circ})/Phi$, from Table A.42:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                              8.19
   1.0e-8                              9.97
  2.53e-8                              11.4
   1.0e-7                              12.6
   2.0e-7                              13.5
   5.0e-7                              14.2
   1.0e-6                              14.4
   2.0e-6                              14.3
   5.0e-6                              13.8
   1.0e-5                              13.2
   2.0e-5                              12.4
   5.0e-5                              11.2
   1.0e-4                              10.3
   2.0e-4                              9.84
   5.0e-4                              9.34
   1.0e-3                              8.78
   2.0e-3                              8.72
   5.0e-3                              9.36
     0.01                              11.2
     0.02                              17.1
     0.03                              24.9
     0.05                              39.0
     0.07                              59.0
      0.1                              90.6
     0.15                             139.0
      0.2                             180.0
      0.3                             246.0
      0.5                             335.0
      0.7                             386.0
      0.9                             414.0
      1.0                             422.0
      1.2                             433.0
      2.0                             442.0
      3.0                             431.0
      4.0                             422.0
      5.0                             420.0
      6.0                             423.0
      7.0                             432.0
      8.0                             445.0
      9.0                             461.0
     10.0                             480.0
     12.0                             517.0
     14.0                             550.0
     15.0                             564.0
     16.0                             576.0
     18.0                             595.0
     20.0                             600.0
c
```

Listing F.15: Neutron_ICRP74-1996_H_textrmpslab1015circPhi_dedf.txt

```
c
c ICRP/74-1996, $H_{\textrm{p,slab}}(10,15^{\circ})/Phi$, from Table A.42:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                              7.64
   1.0e-8                              9.35
  2.53e-8                              10.6
   1.0e-7                              11.7
   2.0e-7                              12.6
   5.0e-7                              13.5
   1.0e-6                              13.9
   2.0e-6                              14.0
   5.0e-6                              13.9
   1.0e-5                              13.4
   2.0e-5                              12.6
   5.0e-5                              11.2
   1.0e-4                              9.85
   2.0e-4                              9.41
   5.0e-4                              8.66
   1.0e-3                               8.2
   2.0e-3                              8.22
   5.0e-3                              8.79
     0.01                              10.8
     0.02                              17.0
     0.03                              24.1
     0.05                              36.0
     0.07                              55.8
      0.1                              87.8
     0.15                             137.0
      0.2                             179.0
      0.3                             244.0
      0.5                             330.0
      0.7                             379.0
      0.9                             407.0
      1.0                             416.0
      1.2                             427.0
      2.0                             438.0
      3.0                             429.0
      4.0                             421.0
      5.0                             418.0
      6.0                             422.0
      7.0                             432.0
      8.0                             445.0
      9.0                             462.0
     10.0                             481.0
     12.0                             519.0
     14.0                             552.0
     15.0                             565.0
     16.0                             577.0
     18.0                             593.0
     20.0                             595.0
c
```

Listing F.16: Neutron_ICRP74-1996_H_textrmpslab1030circPhi_dedf.txt

```
c
c ICRP/74-1996, $H_{\textrm{p,slab}}(10,30^{\circ})/Phi$, from Table A.42:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                             df:n
      log                              log
   1.0e-9                             6.57
   1.0e-8                              7.9
  2.53e-8                             9.11
   1.0e-7                             10.3
   2.0e-7                             11.1
   5.0e-7                             11.8
   1.0e-6                             12.0
   2.0e-6                             11.9
   5.0e-6                             11.5
   1.0e-5                             11.0
   2.0e-5                             10.4
   5.0e-5                             9.42
   1.0e-4                             8.64
   2.0e-4                             8.22
   5.0e-4                             7.66
   1.0e-3                             7.29
   2.0e-3                             7.27
   5.0e-3                             7.46
     0.01                             9.18
     0.02                             14.6
     0.03                             21.3
     0.05                             34.4
     0.07                             52.6
      0.1                             81.3
     0.15                            126.0
      0.2                            166.0
      0.3                            232.0
      0.5                            326.0
      0.7                            382.0
      0.9                            415.0
      1.0                            426.0
      1.2                            440.0
      2.0                            457.0
      3.0                            449.0
      4.0                            440.0
      5.0                            437.0
      6.0                            440.0
      7.0                            449.0
      8.0                            462.0
      9.0                            478.0
     10.0                            497.0
     12.0                            536.0
     14.0                            570.0
     15.0                            584.0
     16.0                            597.0
     18.0                            617.0
     20.0                            619.0
c
```

Listing F.17: Neutron_ICRP74-1996_H_textrmpslab1045circPhi_dedf.txt

```
c
c ICRP/74-1996, $H_{\textrm{p,slab}}(10,45^{\circ})/Phi$, from Table A.42:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                             df:n
      log                              log
   1.0e-9                             4.23
   1.0e-8                             5.38
  2.53e-8                             6.61
   1.0e-7                             7.84
   2.0e-7                             8.73
   5.0e-7                              9.4
   1.0e-6                             9.56
   2.0e-6                             9.49
   5.0e-6                             9.11
   1.0e-5                             8.65
   2.0e-5                              8.1
   5.0e-5                             7.32
   1.0e-4                             6.74
   2.0e-4                             6.21
   5.0e-4                             5.67
   1.0e-3                             5.43
   2.0e-3                             5.43
   5.0e-3                             5.71
     0.01                             7.09
     0.02                             11.6
     0.03                             16.7
     0.05                             27.5
     0.07                             42.9
      0.1                             67.1
     0.15                            106.0
      0.2                            141.0
      0.3                            201.0
      0.5                            291.0
      0.7                            348.0
      0.9                            383.0
      1.0                            395.0
      1.2                            412.0
      2.0                            439.0
      3.0                            440.0
      4.0                            435.0
      5.0                            435.0
      6.0                            439.0
      7.0                            448.0
      8.0                            460.0
      9.0                            476.0
     10.0                            493.0
     12.0                            529.0
     14.0                            561.0
     15.0                            575.0
     16.0                            588.0
     18.0                            609.0
     20.0                            615.0
c
```

Listing F.18: Neutron_ICRP74-1996_H_textrmpslab1060circPhi_dedf.txt

```
c
c ICRP/74-1996, $H_{\textrm{p,slab}}(10,60^{\circ})/Phi$, from Table A.42:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                             2.61
   1.0e-8                             3.37
  2.53e-8                             4.04
   1.0e-7                              4.7
   2.0e-7                             5.21
   5.0e-7                             5.65
   1.0e-6                             5.82
   2.0e-6                             5.85
   5.0e-6                             5.71
   1.0e-5                             5.47
   2.0e-5                             5.14
   5.0e-5                             4.57
   1.0e-4                              4.1
   2.0e-4                             3.91
   5.0e-4                             3.58
   1.0e-3                             3.46
   2.0e-3                             3.46
   5.0e-3                             3.59
     0.01                             4.32
     0.02                             6.64
     0.03                             9.81
     0.05                             16.7
     0.07                             27.3
      0.1                             44.6
     0.15                             73.3
      0.2                            100.0
      0.3                            149.0
      0.5                            226.0
      0.7                            279.0
      0.9                            317.0
      1.0                            332.0
      1.2                            355.0
      2.0                            402.0
      3.0                            412.0
      4.0                            409.0
      5.0                            409.0
      6.0                            414.0
      7.0                            425.0
      8.0                            440.0
      9.0                            458.0
     10.0                            480.0
     12.0                            523.0
     14.0                            562.0
     15.0                            579.0
     16.0                            593.0
     18.0                            615.0
     20.0                            619.0
c
```

Listing F.19: Neutron_ICRP74-1996_H_textrmpslab1075circPhi_dedf.txt

```
c
c ICRP/74-1996, $H_{\textrm{p,slab}}(10,75^{\circ})/Phi$, from Table A.42:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                             1.13
   1.0e-8                              1.5
  2.53e-8                             1.73
   1.0e-7                             1.94
   2.0e-7                             2.12
   5.0e-7                             2.31
   1.0e-6                              2.4
   2.0e-6                             2.46
   5.0e-6                             2.48
   1.0e-5                             2.44
   2.0e-5                             2.35
   5.0e-5                             2.16
   1.0e-4                             1.99
   2.0e-4                             1.83
   5.0e-4                             1.68
   1.0e-3                             1.66
   2.0e-3                             1.67
   5.0e-3                             1.69
     0.01                             1.77
     0.02                             2.11
     0.03                             2.85
     0.05                             4.78
     0.07                              8.1
      0.1                             13.7
     0.15                             24.2
      0.2                             35.5
      0.3                             58.5
      0.5                            102.0
      0.7                            139.0
      0.9                            171.0
      1.0                            180.0
      1.2                            210.0
      2.0                            274.0
      3.0                            306.0
      4.0                            320.0
      5.0                            331.0
      6.0                            345.0
      7.0                            361.0
      8.0                            379.0
      9.0                            399.0
     10.0                            421.0
     12.0                            464.0
     14.0                            503.0
     15.0                            520.0
     16.0                            535.0
     18.0                            561.0
     20.0                            570.0
c
```

Listing F.20: Neutron_ICRP116-2010_Anterior-Posterior_AP_dedf.txt

```
c
c ICRP/116-2010, Anterior-Posterior (AP), from Table A.5:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                             3.09
   1.0e-8                             3.55
   2.5e-8                              4.0
   1.0e-7                              5.2
   2.0e-7                             5.87
   5.0e-7                             6.59
   1.0e-6                             7.03
   2.0e-6                             7.39
   5.0e-6                             7.71
   1.0e-5                             7.82
   2.0e-5                             7.84
   5.0e-5                             7.82
   1.0e-4                             7.79
   2.0e-4                             7.73
   5.0e-4                             7.54
   1.0e-3                             7.54
   2.0e-3                             7.61
   5.0e-3                             7.97
     0.01                             9.11
     0.02                             12.2
     0.03                             15.7
     0.05                             23.0
     0.07                             30.6
      0.1                             41.9
     0.15                             60.6
      0.2                             78.8
      0.3                            114.0
      0.5                            177.0
      0.7                            232.0
      0.9                            279.0
      1.0                            301.0
      1.2                            330.0
      1.5                            365.0
      2.0                            407.0
      3.0                            458.0
      4.0                            483.0
      5.0                            494.0
      6.0                            498.0
      7.0                            499.0
      8.0                            499.0
      9.0                            500.0
     10.0                            500.0
     12.0                            499.0
     14.0                            495.0
     15.0                            493.0
     16.0                            490.0
     18.0                            484.0
     20.0                            477.0
     21.0                            474.0
     30.0                            453.0
```

```
     50.0                          433.0
     75.0                          420.0
    100.0                          402.0
    130.0                          382.0
    150.0                          373.0
    180.0                          363.0
    200.0                          359.0
    300.0                          363.0
    400.0                          389.0
    500.0                          422.0
    600.0                          457.0
    700.0                          486.0
    800.0                          508.0
    900.0                          524.0
    1.0e3                          537.0
    2.0e3                          612.0
    5.0e3                          716.0
    1.0e4                          933.0
c
```

Listing F.21: Neutron_ICRP116-2010_Posterior-Anterior_PA_dedf.txt

```
c
c ICRP/116-2010, Posterior-Anterior (PA), from Table A.5:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                             1.85
   1.0e-8                             2.11
   2.5e-8                             2.44
   1.0e-7                             3.25
   2.0e-7                             3.72
   5.0e-7                             4.33
   1.0e-6                             4.73
   2.0e-6                             5.02
   5.0e-6                              5.3
   1.0e-5                             5.44
   2.0e-5                             5.51
   5.0e-5                             5.55
   1.0e-4                             5.57
   2.0e-4                             5.59
   5.0e-4                              5.6
   1.0e-3                              5.6
   2.0e-3                             5.62
   5.0e-3                             5.95
     0.01                             6.81
     0.02                             8.93
     0.03                             11.2
     0.05                             15.7
     0.07                             20.0
      0.1                             25.9
     0.15                             34.9
      0.2                             43.1
      0.3                             58.1
      0.5                             85.9
      0.7                            112.0
      0.9                            136.0
      1.0                            148.0
      1.2                            167.0
      1.5                            195.0
      2.0                            235.0
      3.0                            292.0
      4.0                            330.0
      5.0                            354.0
      6.0                            371.0
      7.0                            383.0
      8.0                            392.0
      9.0                            398.0
     10.0                            404.0
     12.0                            412.0
     14.0                            417.0
     15.0                            419.0
     16.0                            420.0
     18.0                            422.0
     20.0                            423.0
     21.0                            423.0
     30.0                            422.0
```

```
     50.0                    428.0
     75.0                    439.0
    100.0                    444.0
    130.0                    446.0
    150.0                    446.0
    180.0                    447.0
    200.0                    448.0
    300.0                    464.0
    400.0                    496.0
    500.0                    533.0
    600.0                    569.0
    700.0                    599.0
    800.0                    623.0
    900.0                    640.0
    1.0e3                    654.0
    2.0e3                    740.0
    5.0e3                    924.0
    1.0e4                   1.17e3
c
```

Listing F.22: Neutron_ICRP116-2010_Left_Lateral_LLAT_dedf.txt

```
c
c ICRP/116-2010, Left Lateral (LLAT), from Table A.5:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                             1.04
   1.0e-8                             1.15
   2.5e-8                             1.32
   1.0e-7                              1.7
   2.0e-7                             1.94
   5.0e-7                             2.21
   1.0e-6                              2.4
   2.0e-6                             2.52
   5.0e-6                             2.64
   1.0e-5                             2.65
   2.0e-5                             2.68
   5.0e-5                             2.66
   1.0e-4                             2.65
   2.0e-4                             2.66
   5.0e-4                             2.62
   1.0e-3                             2.61
   2.0e-3                              2.6
   5.0e-3                             2.74
     0.01                             3.13
     0.02                             4.21
     0.03                              5.4
     0.05                             7.91
     0.07                             10.5
      0.1                             14.4
     0.15                             20.8
      0.2                             27.2
      0.3                             39.7
      0.5                             63.7
      0.7                             85.5
      0.9                            105.0
      1.0                            115.0
      1.2                            130.0
      1.5                            150.0
      2.0                            179.0
      3.0                            221.0
      4.0                            249.0
      5.0                            269.0
      6.0                            284.0
      7.0                            295.0
      8.0                            303.0
      9.0                            310.0
     10.0                            316.0
     12.0                            325.0
     14.0                            333.0
     15.0                            336.0
     16.0                            338.0
     18.0                            343.0
     20.0                            347.0
     21.0                            348.0
     30.0                            360.0
```

```
     50.0                          380.0
     75.0                          399.0
    100.0                          409.0
    130.0                          416.0
    150.0                          420.0
    180.0                          425.0
    200.0                          427.0
    300.0                          441.0
    400.0                          472.0
    500.0                          510.0
    600.0                          547.0
    700.0                          579.0
    800.0                          603.0
    900.0                          621.0
    1.0e3                          635.0
    2.0e3                          730.0
    5.0e3                          963.0
    1.0e4                         1.23e3
c
```

Listing F.23: Neutron_ICRP116-2010_Right_Lateral_RLAT_dedf.txt

```
c
c ICRP/116-2010, Right Lateral (RLAT), from Table A.5:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                            0.893
   1.0e-8                            0.978
   2.5e-8                             1.12
   1.0e-7                             1.42
   2.0e-7                             1.63
   5.0e-7                             1.86
   1.0e-6                             2.02
   2.0e-6                             2.11
   5.0e-6                             2.21
   1.0e-5                             2.24
   2.0e-5                             2.26
   5.0e-5                             2.24
   1.0e-4                             2.23
   2.0e-4                             2.24
   5.0e-4                             2.21
   1.0e-3                             2.21
   2.0e-3                              2.2
   5.0e-3                             2.33
     0.01                             2.67
     0.02                              3.6
     0.03                             4.62
     0.05                             6.78
     0.07                             8.95
      0.1                             12.3
     0.15                             17.9
      0.2                             23.4
      0.3                             34.2
      0.5                             54.4
      0.7                             72.6
      0.9                             89.3
      1.0                             97.4
      1.2                            110.0
      1.5                            128.0
      2.0                            153.0
      3.0                            192.0
      4.0                            220.0
      5.0                            240.0
      6.0                            255.0
      7.0                            267.0
      8.0                            276.0
      9.0                            284.0
     10.0                            290.0
     12.0                            301.0
     14.0                            310.0
     15.0                            313.0
     16.0                            317.0
     18.0                            323.0
     20.0                            328.0
     21.0                            330.0
     30.0                            345.0
```

```
     50.0                          370.0
     75.0                          392.0
    100.0                          404.0
    130.0                          413.0
    150.0                          418.0
    180.0                          425.0
    200.0                          429.0
    300.0                          451.0
    400.0                          483.0
    500.0                          523.0
    600.0                          563.0
    700.0                          597.0
    800.0                          620.0
    900.0                          638.0
    1.0e3                          651.0
    2.0e3                          747.0
    5.0e3                          979.0
    1.0e4                         1.26e3
c
```

Listing F.24: Neutron_ICRP116-2010_Rotational_ROT_dedf.txt

```
c
c ICRP/116-2010, Rotational (ROT), from Table A.5:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                               1.7
   1.0e-8                              2.03
   2.5e-8                              2.31
   1.0e-7                              2.98
   2.0e-7                              3.36
   5.0e-7                              3.86
   1.0e-6                              4.17
   2.0e-6                               4.4
   5.0e-6                              4.59
   1.0e-5                              4.68
   2.0e-5                              4.72
   5.0e-5                              4.73
   1.0e-4                              4.72
   2.0e-4                              4.67
   5.0e-4                               4.6
   1.0e-3                              4.58
   2.0e-3                              4.61
   5.0e-3                              4.86
     0.01                              5.57
     0.02                              7.41
     0.03                              9.46
     0.05                              13.7
     0.07                              18.0
      0.1                              24.3
     0.15                              34.7
      0.2                              44.7
      0.3                              63.8
      0.5                              99.1
      0.7                             131.0
      0.9                             160.0
      1.0                             174.0
      1.2                             193.0
      1.5                             219.0
      2.0                             254.0
      3.0                             301.0
      4.0                             331.0
      5.0                             351.0
      6.0                             365.0
      7.0                             374.0
      8.0                             381.0
      9.0                             386.0
     10.0                             390.0
     12.0                             395.0
     14.0                             398.0
     15.0                             398.0
     16.0                             399.0
     18.0                             399.0
     20.0                             398.0
     21.0                             398.0
     30.0                             395.0
```

```
     50.0                          395.0
     75.0                          402.0
    100.0                          406.0
    130.0                          411.0
    150.0                          414.0
    180.0                          418.0
    200.0                          422.0
    300.0                          443.0
    400.0                          472.0
    500.0                          503.0
    600.0                          532.0
    700.0                          558.0
    800.0                          580.0
    900.0                          598.0
    1.0e3                          614.0
    2.0e3                          718.0
    5.0e3                          906.0
    1.0e4                         1.14e3
c
```

Listing F.25: Neutron_ICRP116-2010_Isotropic_ISO_dedf.txt

```
c
c ICRP/116-2010, Isotropic (ISO), from Table A.5:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:n                              df:n
      log                               log
   1.0e-9                            1.29
   1.0e-8                            1.56
   2.5e-8                            1.76
   1.0e-7                            2.26
   2.0e-7                            2.54
   5.0e-7                            2.92
   1.0e-6                            3.15
   2.0e-6                            3.32
   5.0e-6                            3.47
   1.0e-5                            3.52
   2.0e-5                            3.54
   5.0e-5                            3.55
   1.0e-4                            3.54
   2.0e-4                            3.52
   5.0e-4                            3.47
   1.0e-3                            3.46
   2.0e-3                            3.48
   5.0e-3                            3.66
     0.01                            4.19
     0.02                            5.61
     0.03                            7.18
     0.05                            10.4
     0.07                            13.7
      0.1                            18.6
     0.15                            26.6
      0.2                            34.4
      0.3                            49.4
      0.5                            77.1
      0.7                           102.0
      0.9                           126.0
      1.0                           137.0
      1.2                           153.0
      1.5                           174.0
      2.0                           203.0
      3.0                           244.0
      4.0                           271.0
      5.0                           290.0
      6.0                           303.0
      7.0                           313.0
      8.0                           321.0
      9.0                           327.0
     10.0                           332.0
     12.0                           339.0
     14.0                           344.0
     15.0                           346.0
     16.0                           347.0
     18.0                           350.0
     20.0                           352.0
     21.0                           353.0
     30.0                           358.0
```

```
     50.0                          371.0
     75.0                          387.0
    100.0                          397.0
    130.0                          407.0
    150.0                          412.0
    180.0                          421.0
    200.0                          426.0
    300.0                          455.0
    400.0                          488.0
    500.0                          521.0
    600.0                          553.0
    700.0                          580.0
    800.0                          604.0
    900.0                          624.0
    1.0e3                          642.0
    2.0e3                          767.0
    5.0e3                         1.01e3
    1.0e4                         1.32e3
c
```

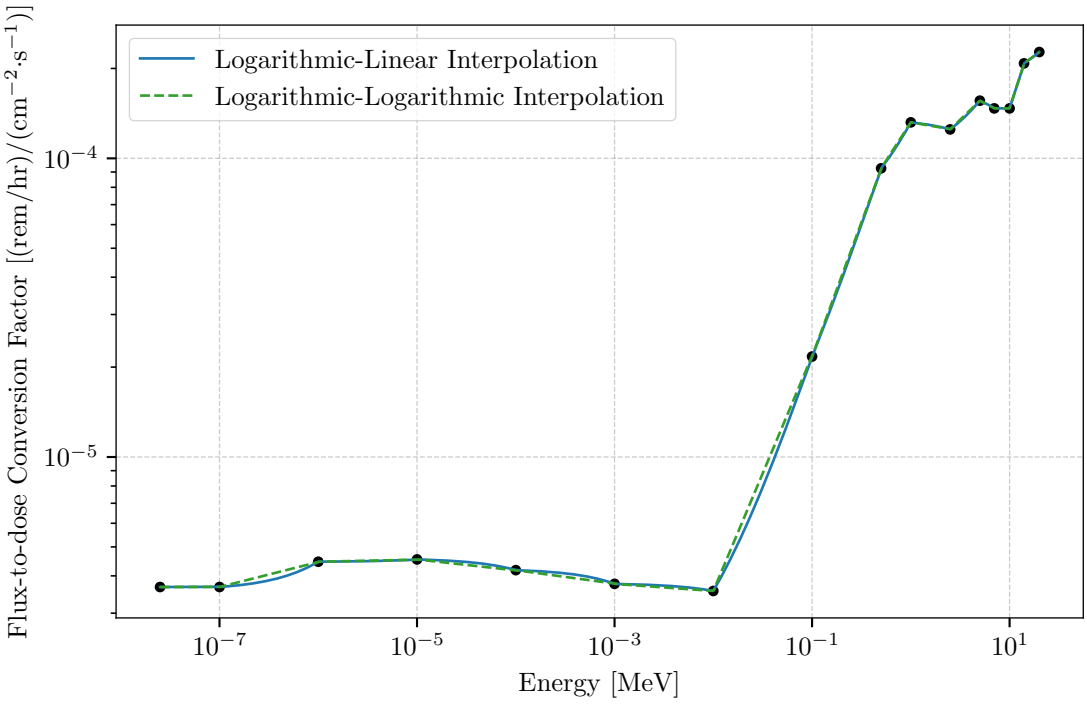


Figure F.1: ANSI/ANS-6.1.1-1977 Neutron Flux-to-dose Conversion Factors

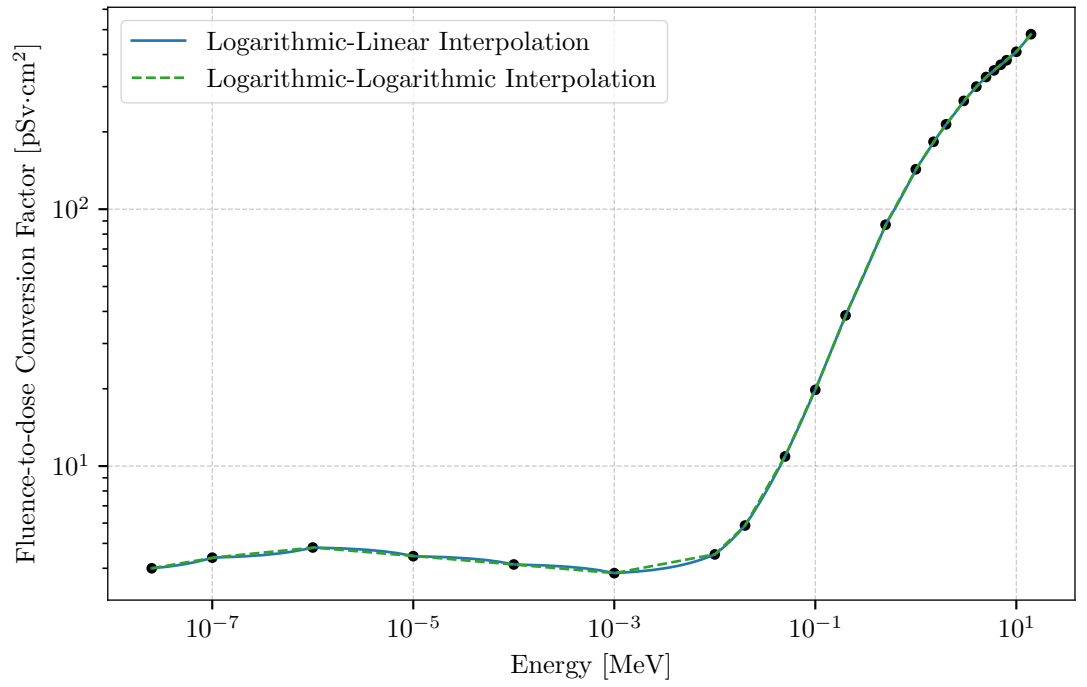


Figure F.2: ANSI/ANS-6.1.1-1991 Anterior-Posterior (AP) Neutron Fluence-to-dose Conversion Factors

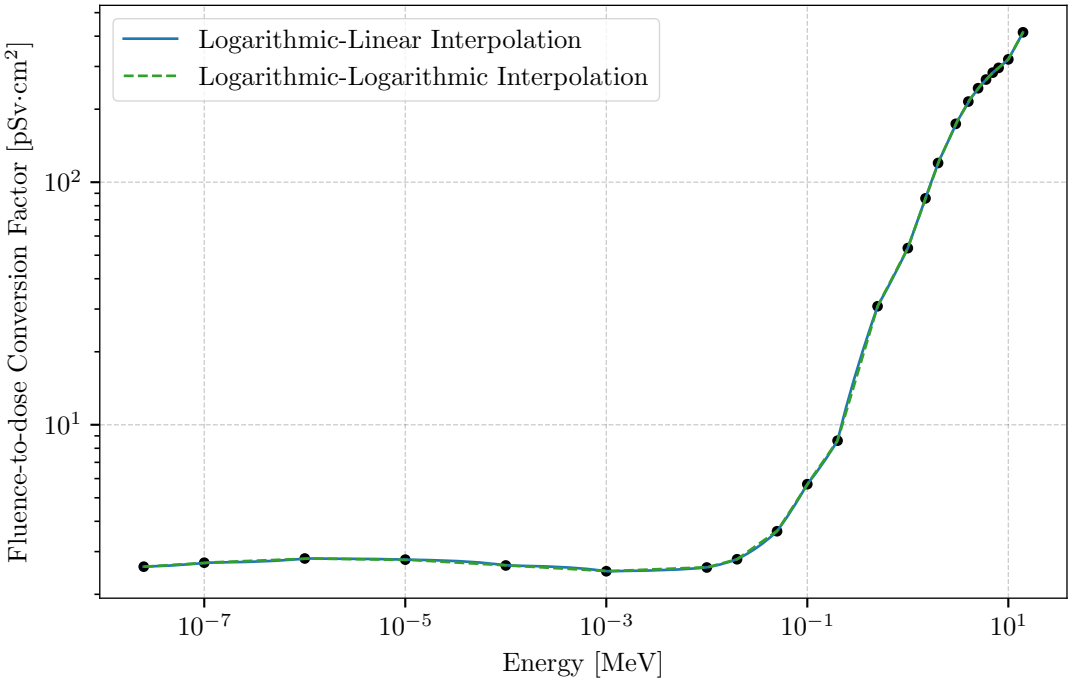


Figure F.3: ANSI/ANS-6.1.1-1991 Posterior-Anterior (PA) Neutron Fluence-to-dose Conversion Factors

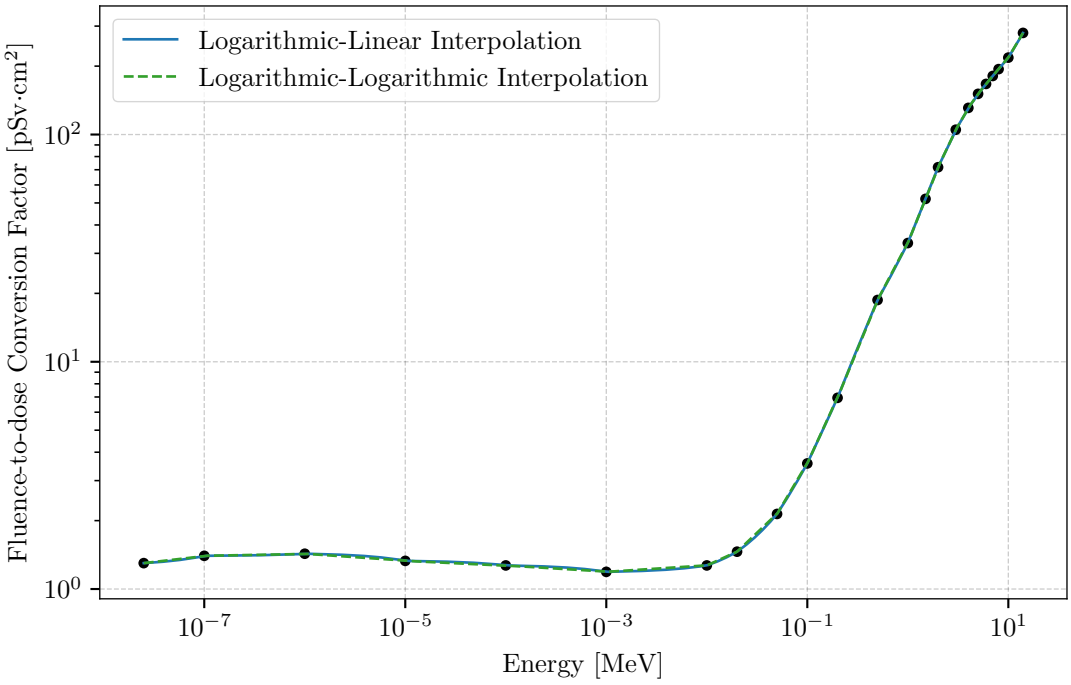


Figure F.4: ANSI/ANS-6.1.1-1991 Lateral (LAT) Neutron Fluence-to-dose Conversion Factors

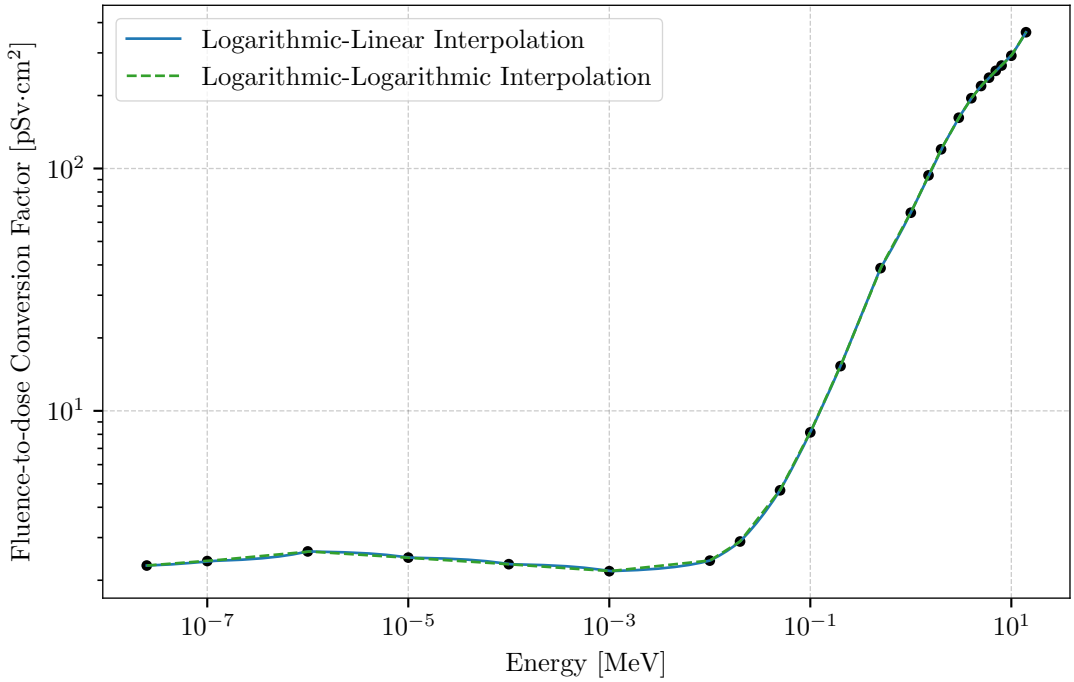


Figure F.5: ANSI/ANS-6.1.1-1991 Rotational (ROT) Neutron Fluence-to-dose Conversion Factors

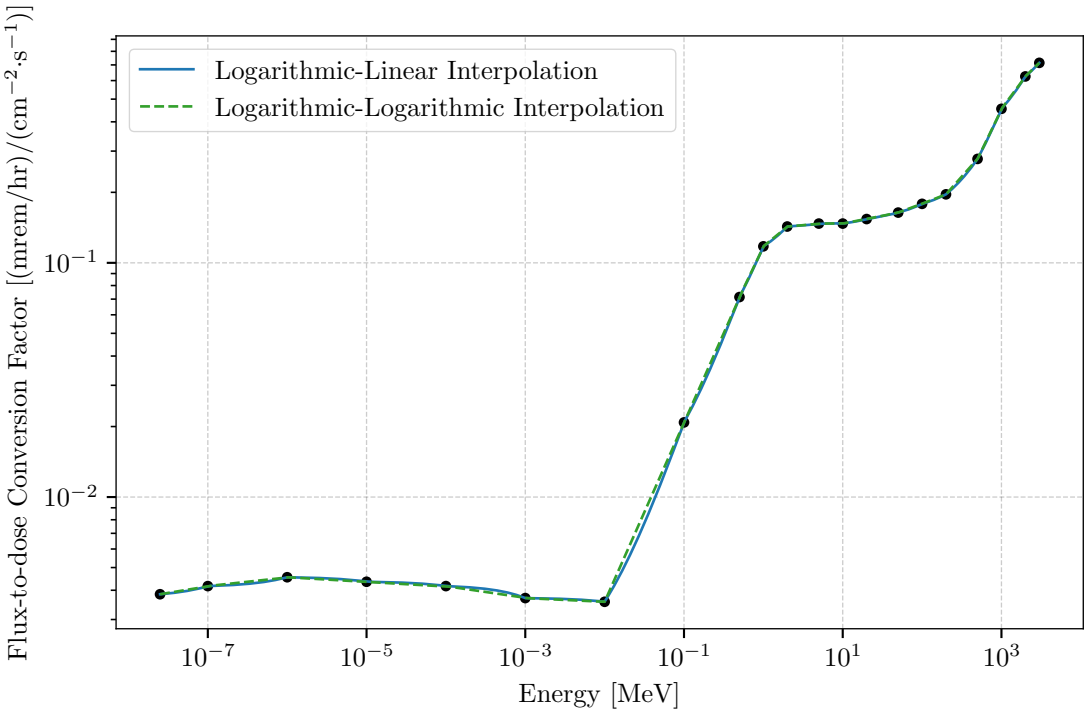


Figure F.6: ICRP/21-1973 Neutron Flux-to-dose Conversion Factors.

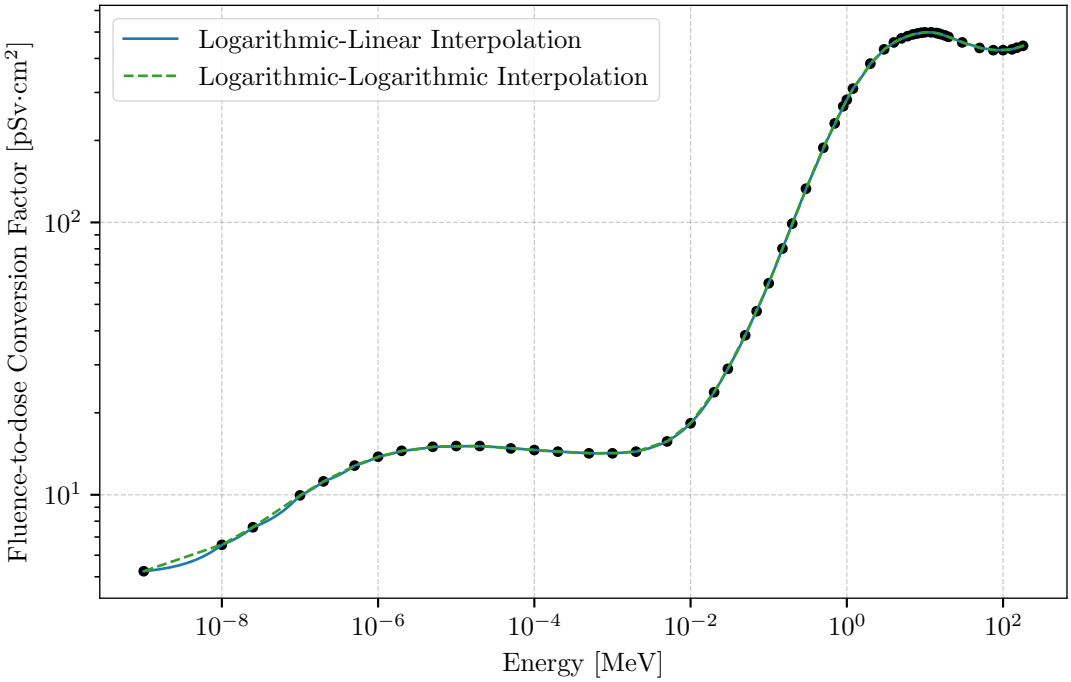


Figure F.7: ICRP/74-1996 Anterior-Posterior (AP) Neutron Fluence-to-dose Conversion Factors.

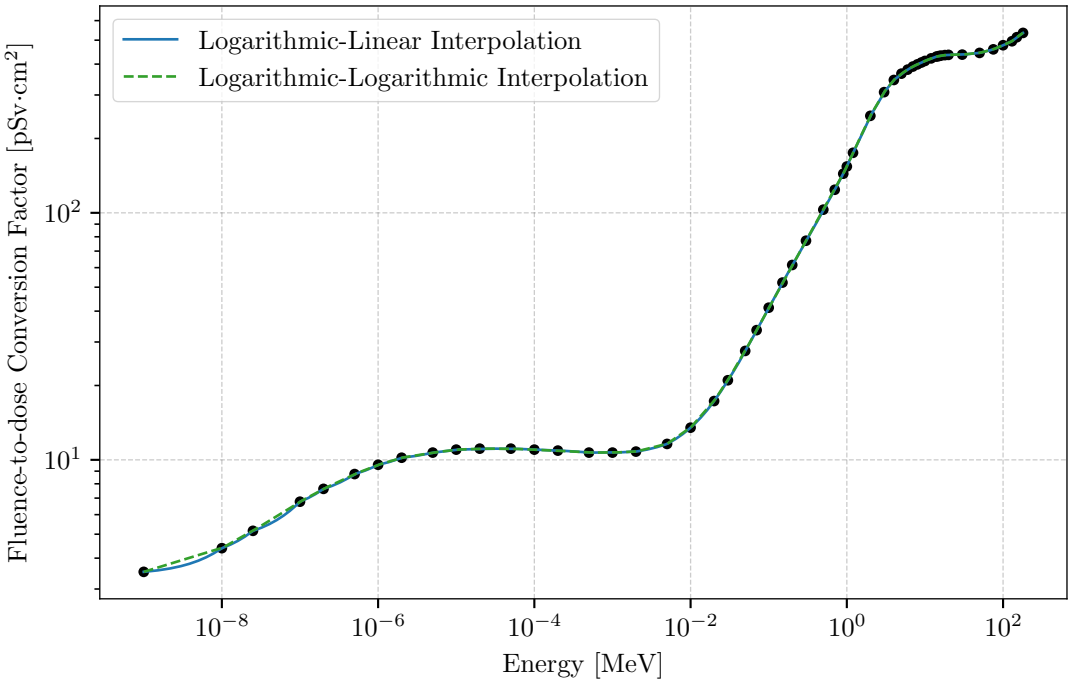


Figure F.8: ICRP/74-1996 Posterior-Anterior (PA) Neutron Fluence-to-dose Conversion Factors

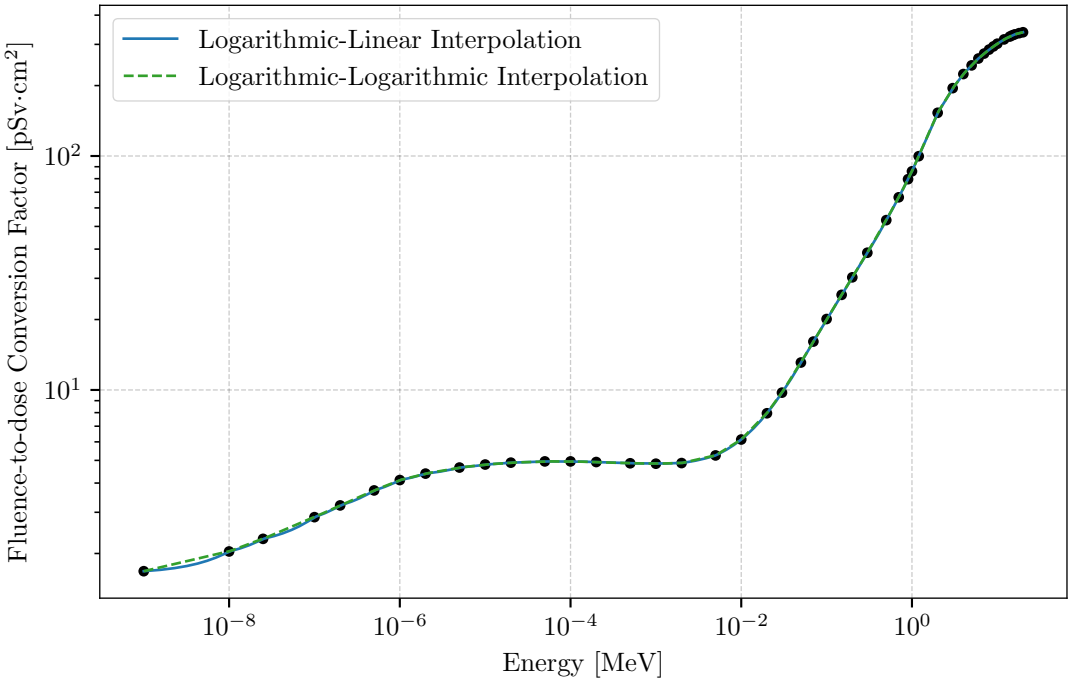


Figure F.9: ICRP/74-1996 Left Lateral (LLAT) Neutron Fluence-to-dose Conversion Factors

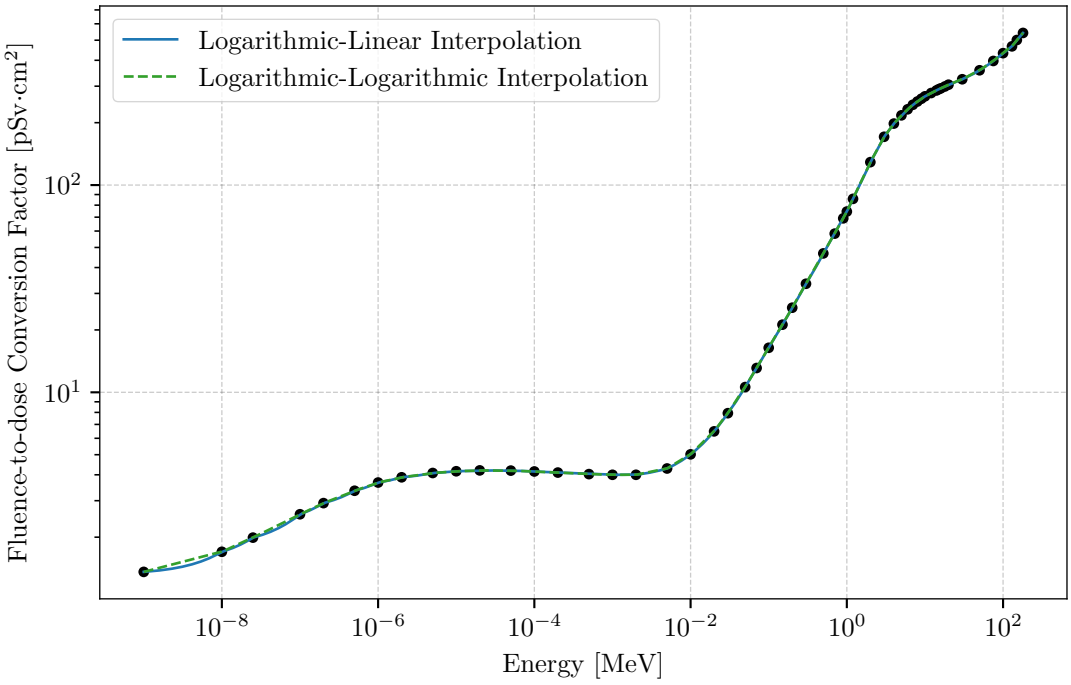


Figure F.10: ICRP/74-1996 Right Lateral (RLAT) Neutron Fluence-to-dose Conversion Factors

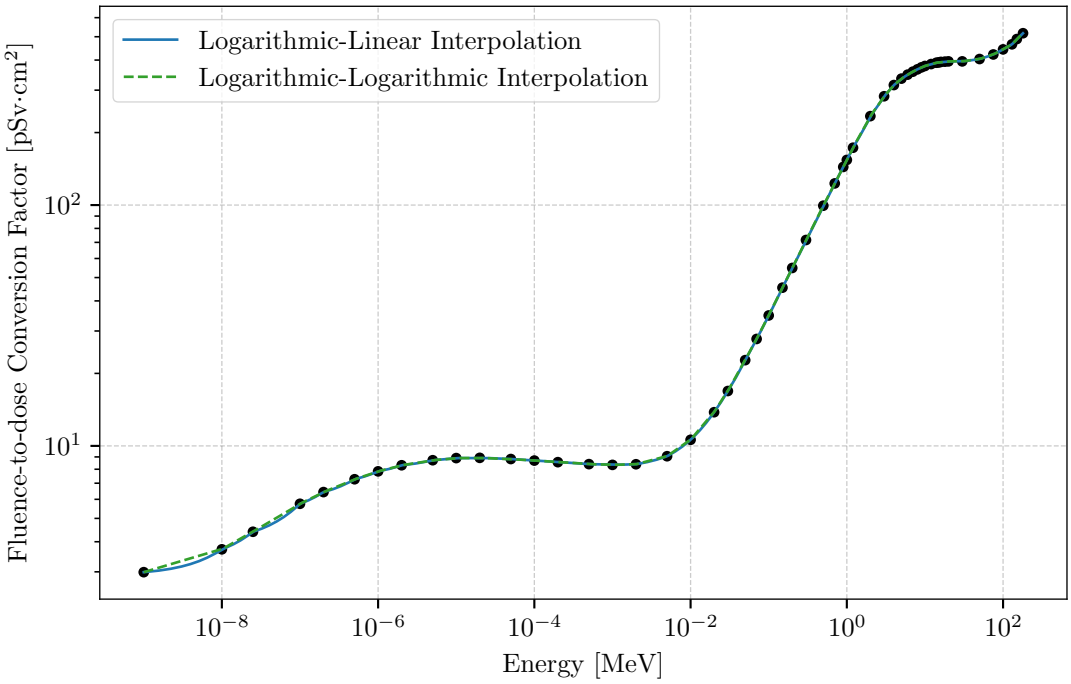


Figure F.11: ICRP/74-1996 Rotational (ROT) Neutron Fluence-to-dose Conversion Factors

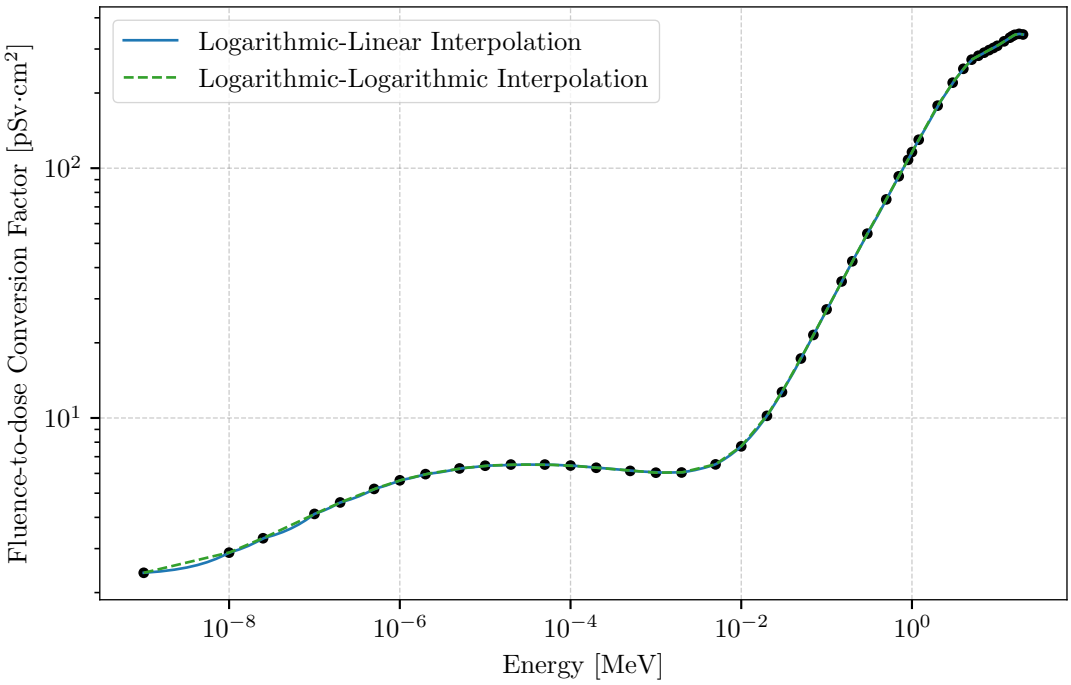


Figure F.12: ICRP/74-1996 Isotropic (ISO) Neutron Fluence-to-dose Conversion Factors

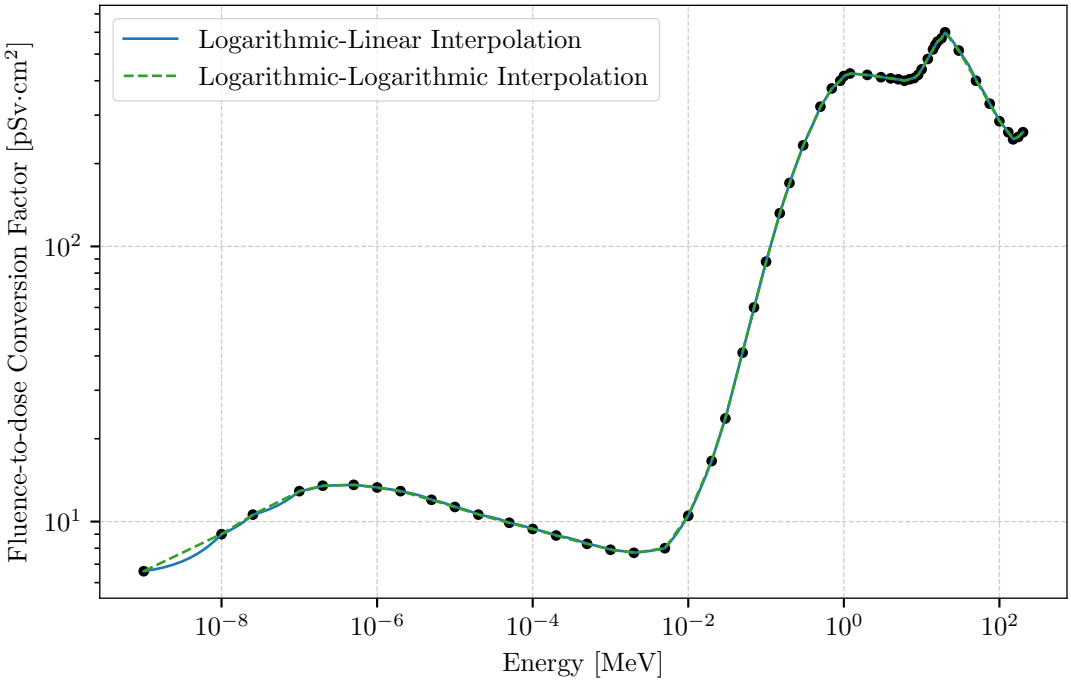


Figure F.13: ICRP/74-1996 Ambient Dose Equivalent ($H^*(10)/\phi$) Neutron Fluence-to-dose Conversion Factors

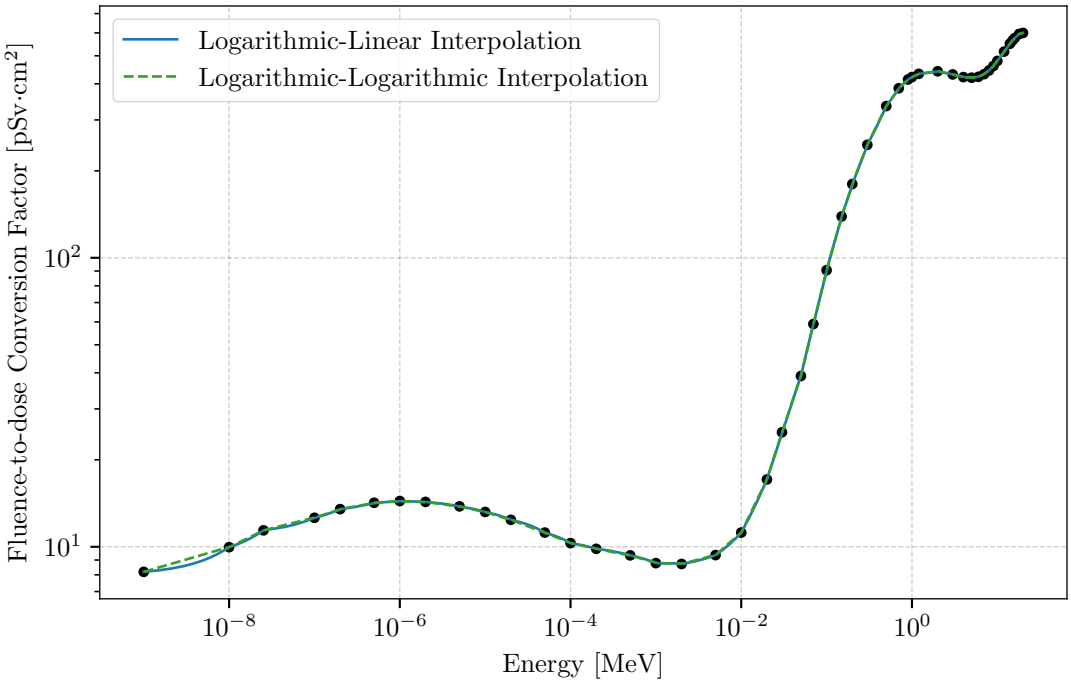


Figure F.14: ICRP/74-1996 Personal Dose Equivalent ($H_{\mathrm{p,slab}}(10, 0°)/\phi$) Neutron Fluence-to-dose Conversion Factors

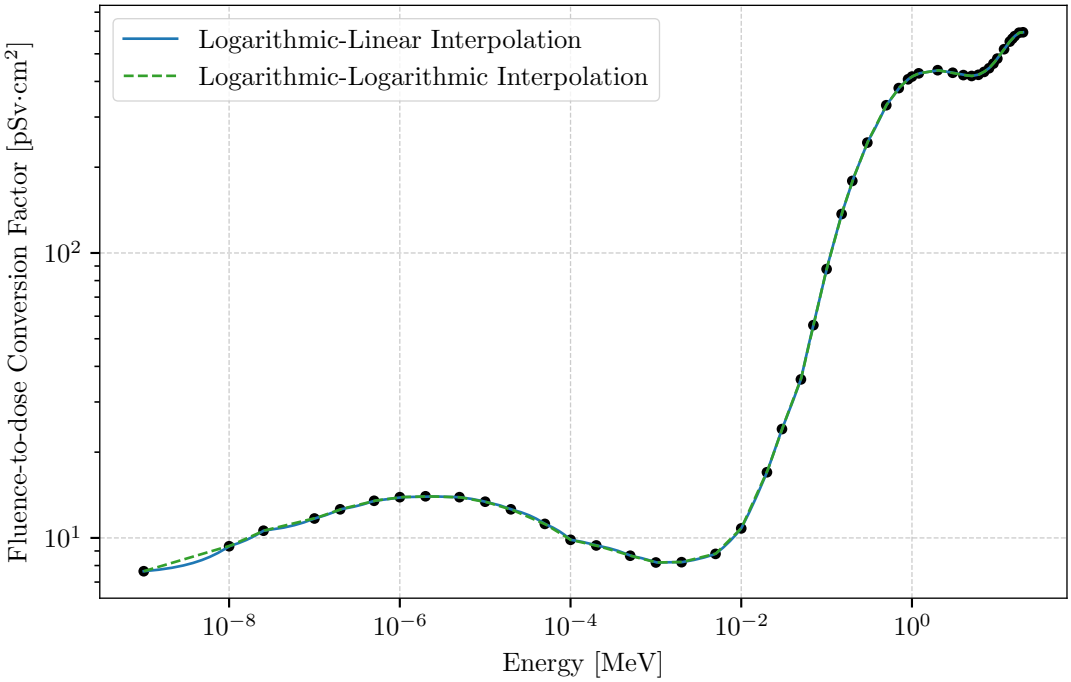


Figure F.15: ICRP/74-1996 Personal Dose Equivalent ($H_{\mathrm{p,slab}}(10, 15°)/\phi$) Neutron Fluence-to-dose Conversion Factors

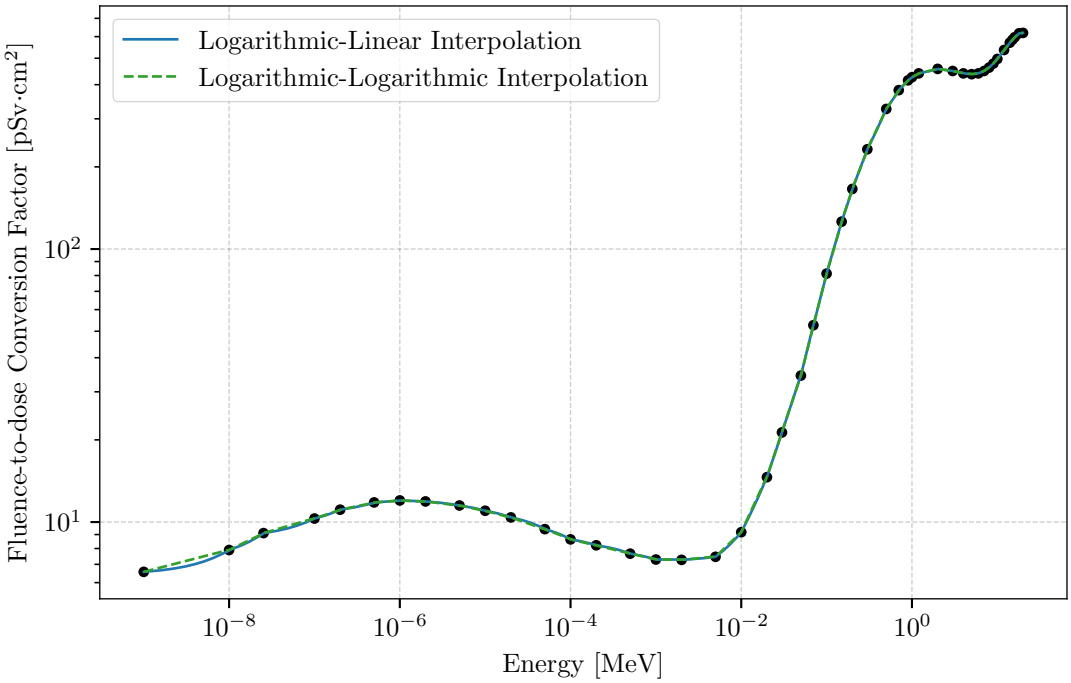


Figure F.16: ICRP/74-1996 Personal Dose Equivalent ($H_{\mathrm{p,slab}}(10, 30°)/\phi$) Neutron Fluence-to-dose Conversion Factors

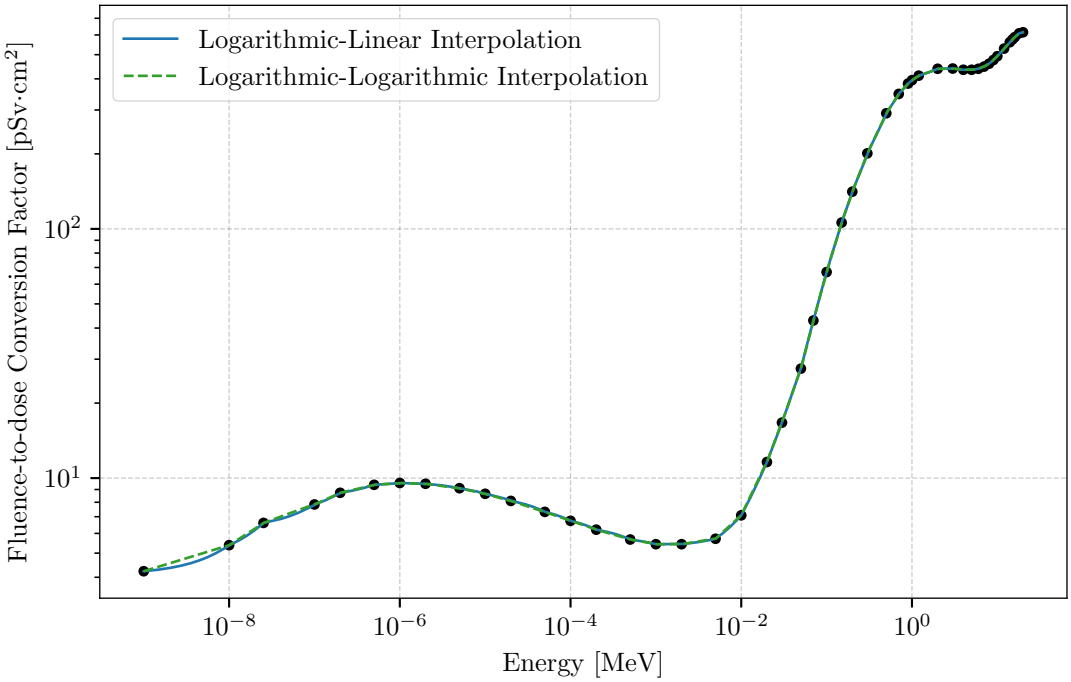


Figure F.17: ICRP/74-1996 Personal Dose Equivalent ($H_{\mathrm{p,slab}}(10, 45°)/\phi$) Neutron Fluence-to-dose Conversion Factors

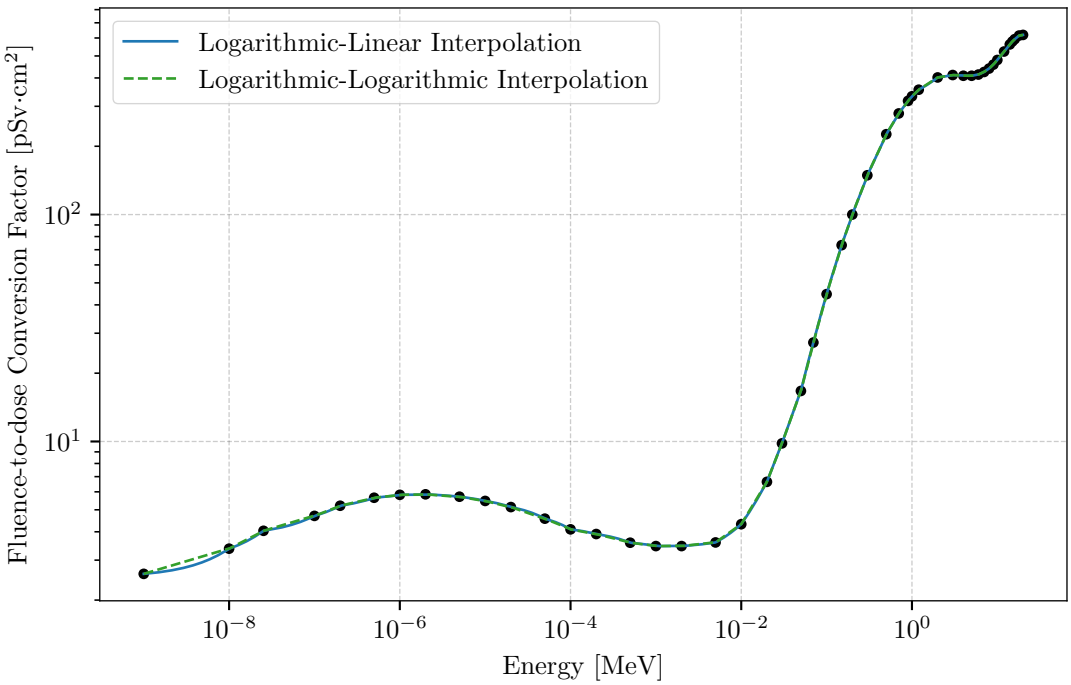


Figure F.18: ICRP/74-1996 Personal Dose Equivalent ($H_{\mathrm{p,slab}}(10, 60°)/\phi$) Neutron Fluence-to-dose Conversion Factors

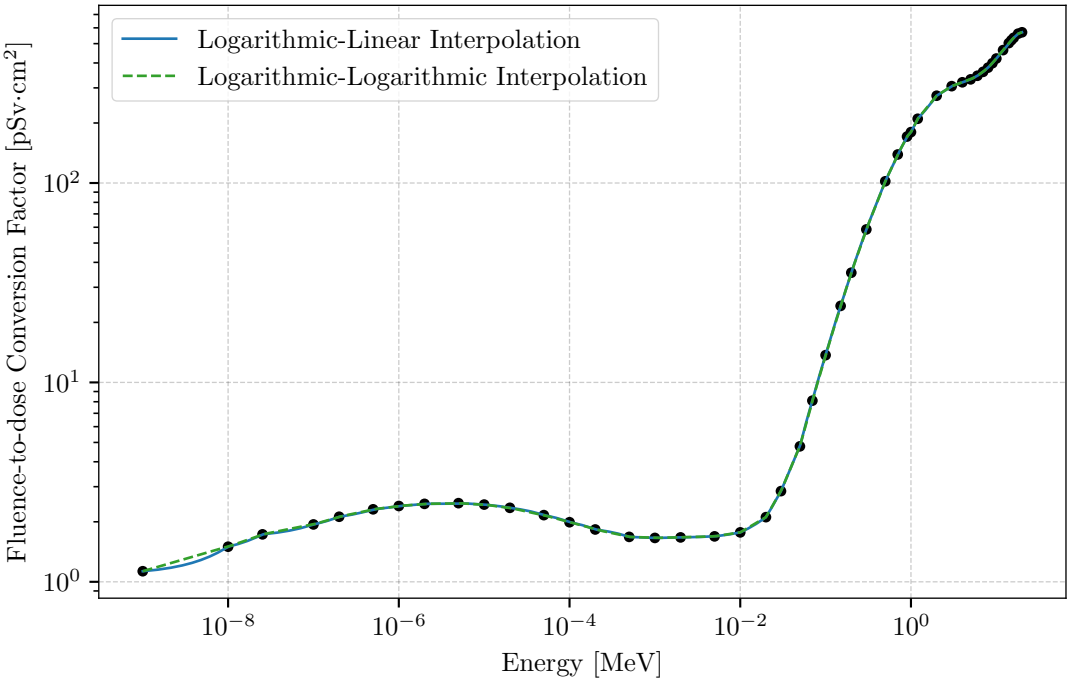


Figure F.19: ICRP/74-1996 Personal Dose Equivalent ($H_{\mathrm{p,slab}}(10, 75°)/\phi$) Neutron Fluence-to-dose Conversion Factors

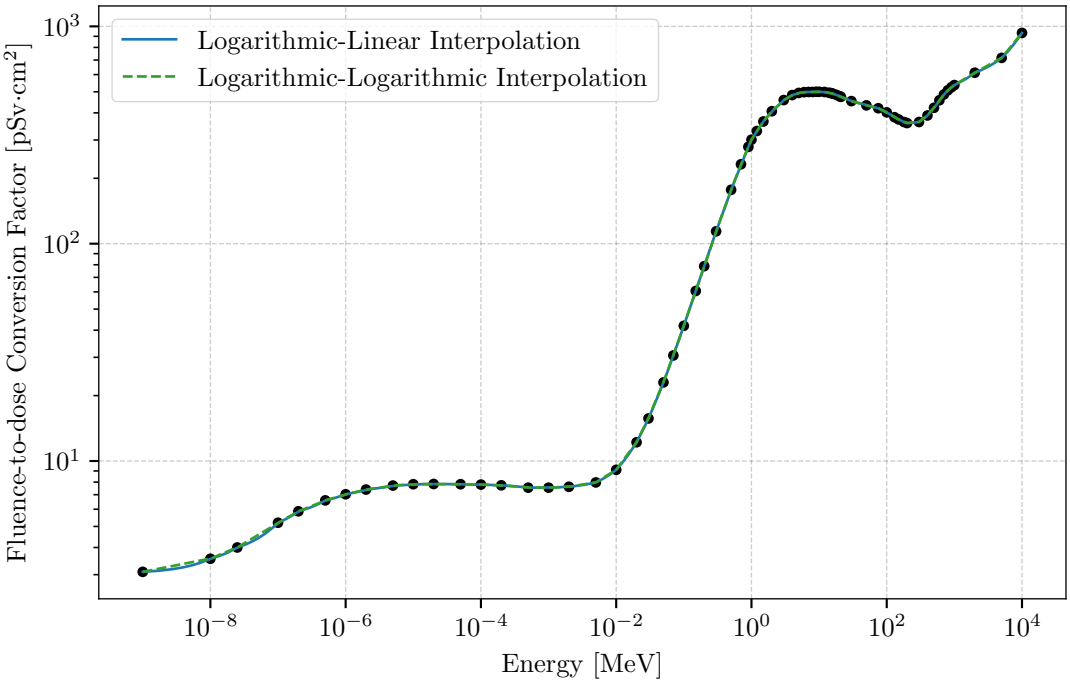


Figure F.20: ICRP/116-2010 Anterior-Posterior (AP) Neutron Fluence-to-dose Conversion Factors

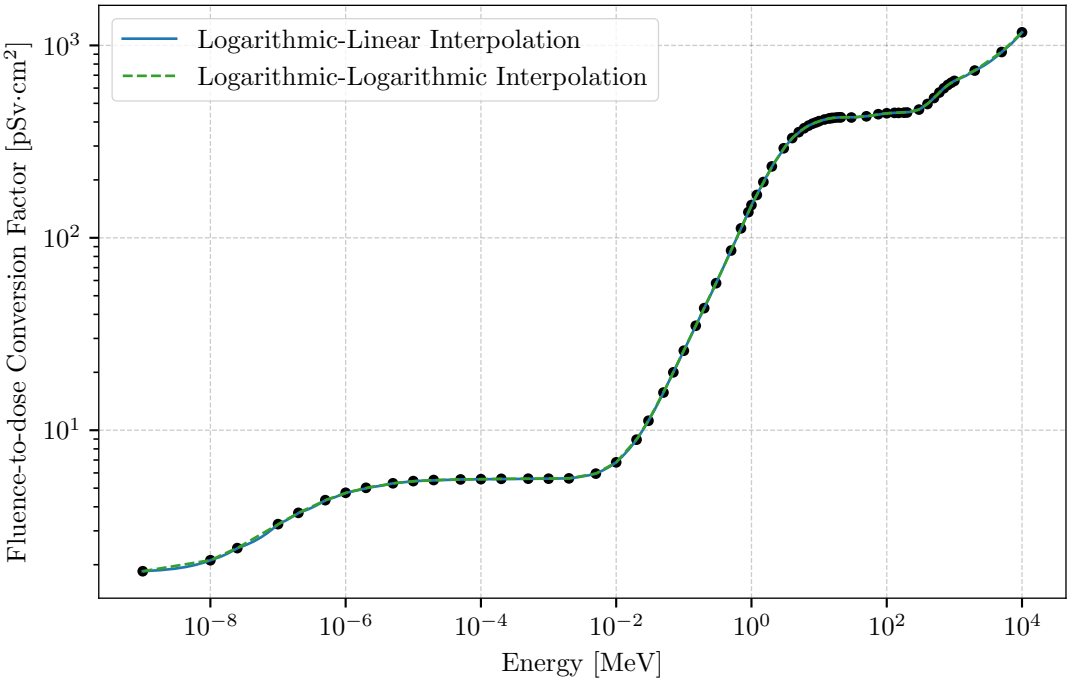


Figure F.21: ICRP/116-2010 Posterior-Anterior (PA) Neutron Fluence-to-dose Conversion Factors

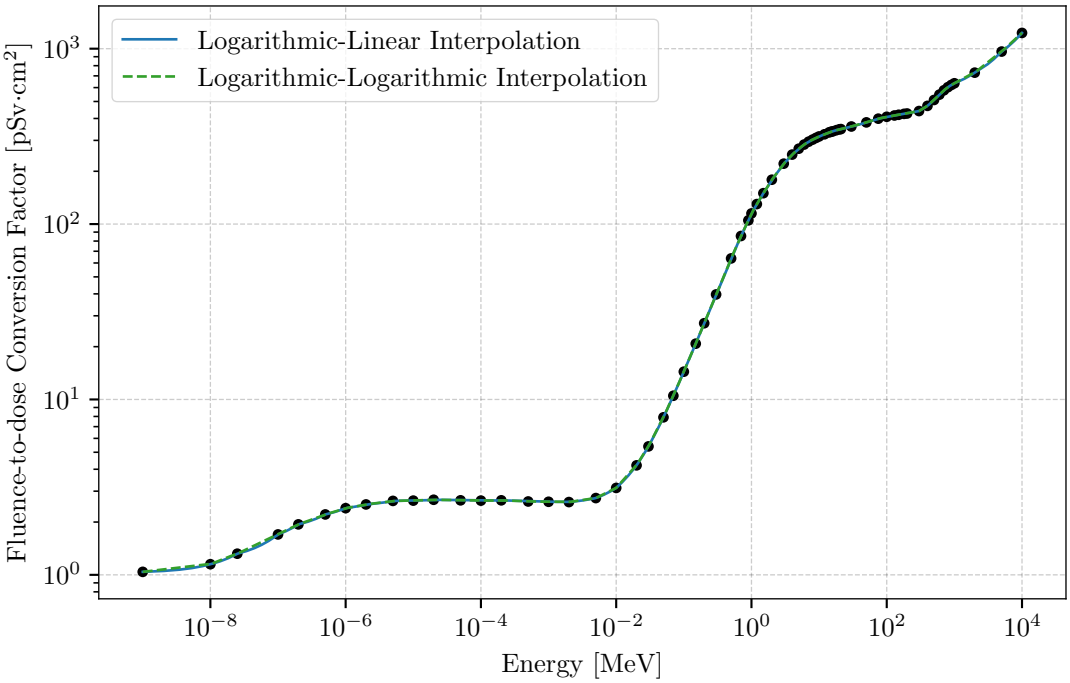


Figure F.22: ICRP/116-2010 Left Lateral (LLAT) Neutron Fluence-to-dose Conversion Factors

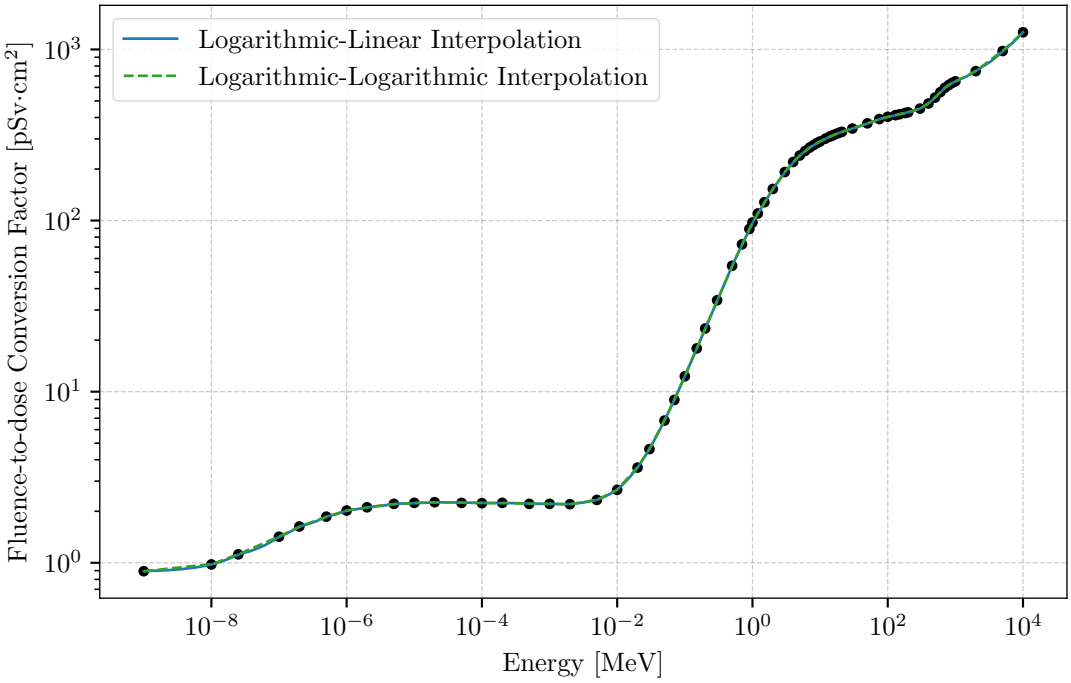


Figure F.23: ICRP/116-2010 Right Lateral (RLAT) Neutron Fluence-to-dose Conversion Factors

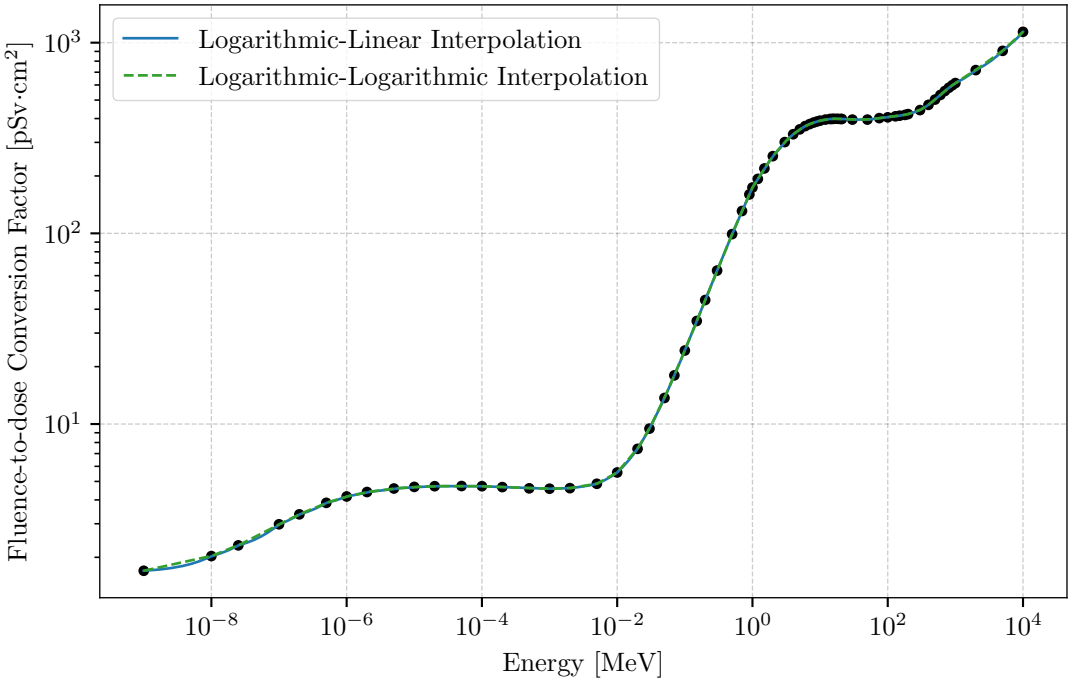


Figure F.24: ICRP/116-2010 Rotational (ROT) Neutron Fluence-to-dose Conversion Factors

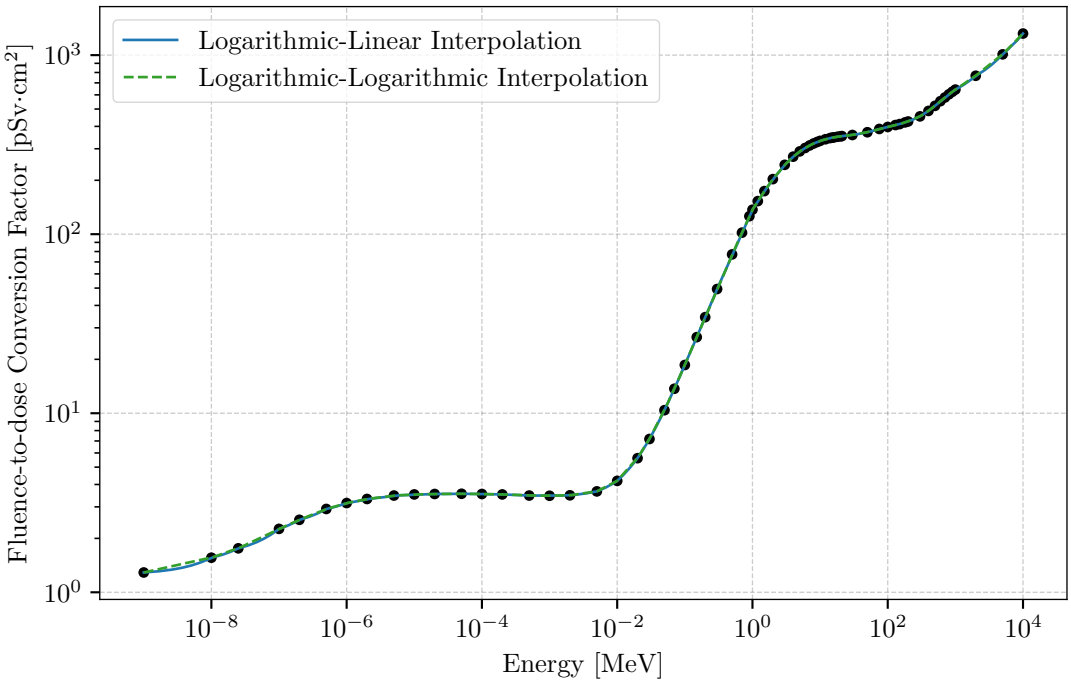


Figure F.25: ICRP/116-2010 Isotropic (ISO) Neutron Fluence-to-dose Conversion Factors

F.1.2 Incident Photon

The ANSI/ANS-6.1.1-1977 photon flux-to-dose conversion factors are given in Listing F.26 which can be directly used as MCNP input for DE/DF cards. The flux-to-dose conversion factors are also plotted in Figure F.26 showing both linear and logarithmic interpolation. These values are extracted from [360] with permission of the publisher, the American Nuclear Society.

The ANSI/ANS-6.1.1-1991 standard provides a variety of photon fluence-to-dose conversion factors assuming five irradiation-phantom orientations: anterior-posterior (AP), posterior-anterior (PA), lateral (LAT), rotational (ROT), and isotropic (ISO). More details on these factors, and how to use them, are available in [361]. The AP, PA, LAT, ROT, and ISO responses are given in Listings F.27, F.28, F.29, F.30, and F.31, respectively, which can be directly used as MCNP input for DE/DF cards. In addition, the conversion factors are plotted in Figures F.27, F.28, F.29, and F.30.

The ICRP/21-1973 photon fluence-to-dose conversion factors are given in Listing F.32, which can be directly used as MCNP input for DE/DF cards. These values are modified from the original values in [362]. The values in Listing F.32 are the inverse of the original values. In addition, Listing F.32 includes extra significant figures in order to reconstruct the original values in [362]. In addition, a duplicate entry for 10 MeV in [362] has been removed to provide a monotonic progression in energy, which is required by the DE card.

The fluence-to-dose conversion factors are also plotted in Figure F.32 showing both linear and logarithmic interpolation.

The ICRP/116-2010 standard provides a variety of photon fluence-to-dose conversion factors assuming six irradiation-phantom orientations: anterior-posterior (AP), posterior-anterior (PA), left lateral (LLAT), right lateral (RLAT), rotational (ROT), and isotropic (ISO). More details on these factors, and how to use them, are available in [364]. The AP, PA, LLAT, RLAT, ROT, and ISO responses are given in Listings F.33, F.34, F.35, F.36, F.37, and F.38, respectively, which can be directly used as MCNP input for DE/DF cards. In addition, the conversion factors are plotted in Figures F.33, F.34, F.35, F.36, F.37, and F.38.

Listing F.26: Photon_ANSIANS-611-1977_dedf.txt

```
c
c ANSI/ANS-6.1.1-1977, from Table 3:
c
c  Energy      Flux-to-dose Conversion Factor
c   [MeV] [(rem/hr)/(cm$^{-2}\cdot$s$^{-1}$)]
#    de:p                                df:p
      log                                 log
     0.01                              3.96e-6
     0.03                              5.82e-7
     0.05                               2.9e-7
     0.07                              2.58e-7
      0.1                              2.83e-7
     0.15                              3.79e-7
      0.2                              5.01e-7
     0.25                              6.31e-7
      0.3                              7.59e-7
     0.35                              8.78e-7
      0.4                              9.85e-7
     0.45                              1.08e-6
      0.5                              1.17e-6
     0.55                              1.27e-6
      0.6                              1.36e-6
     0.65                              1.44e-6
      0.7                              1.52e-6
      0.8                              1.68e-6
      1.0                              1.98e-6
      1.4                              2.51e-6
      1.8                              2.99e-6
      2.2                              3.42e-6
      2.6                              3.82e-6
      2.8                              4.01e-6
     3.25                              4.41e-6
     3.75                              4.83e-6
     4.25                              5.23e-6
     4.75                               5.6e-6
      5.0                               5.8e-6
     5.25                              6.01e-6
     5.75                              6.37e-6
     6.25                              6.74e-6
     6.75                              7.11e-6
      7.5                              7.66e-6
      9.0                              8.77e-6
     11.0                              1.03e-5
     13.0                              1.18e-5
     15.0                              1.33e-5
c
```

Listing F.27: Photon_ANSIANS-611-1991_Anterior-Posterior_AP_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Anterior-Posterior (AP), from Table 3:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                             df:p
      log                              log
     0.01                            0.062
    0.015                            0.157
     0.02                            0.238
     0.03                            0.329
     0.04                            0.365
     0.05                            0.384
     0.06                              0.4
     0.08                            0.451
      0.1                            0.533
     0.15                            0.777
      0.2                             1.03
      0.3                             1.56
      0.4                             2.06
      0.5                             2.54
      0.6                             2.99
      0.8                             3.83
      1.0                              4.6
      1.5                             6.24
      2.0                             7.66
      3.0                             10.2
      4.0                             12.5
      5.0                             14.7
      6.0                             16.7
      8.0                             20.8
     10.0                             24.7
     12.0                             28.9
c
```

Listing F.28: Photon_ANSIANS-611-1991_Posterior-Anterior_PA_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Posterior-Anterior (PA), from Table 3:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                            1.0e-4
    0.015                             0.031
     0.02                            0.0868
     0.03                             0.161
     0.04                             0.222
     0.05                              0.26
     0.06                             0.286
     0.08                             0.344
      0.1                             0.418
     0.15                             0.624
      0.2                             0.844
      0.3                               1.3
      0.4                              1.76
      0.5                               2.2
      0.6                              2.62
      0.8                              3.43
      1.0                              4.18
      1.5                               5.8
      2.0                              7.21
      3.0                              9.71
      4.0                              12.0
      5.0                              14.1
      6.0                              16.2
      8.0                              20.2
     10.0                              24.2
     12.0                              28.8
c
```

Listing F.29: Photon_ANSIANS-611-1991_Lateral_LAT_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Lateral (LAT), from Table 3:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                              0.02
    0.015                             0.033
     0.02                            0.0491
     0.03                            0.0863
     0.04                             0.123
     0.05                             0.152
     0.06                              0.17
     0.08                             0.212
      0.1                             0.258
     0.15                             0.396
      0.2                             0.557
      0.3                             0.891
      0.4                              1.24
      0.5                              1.58
      0.6                              1.92
      0.8                               2.6
      1.0                              3.24
      1.5                               4.7
      2.0                              6.02
      3.0                               8.4
      4.0                              10.6
      5.0                              12.6
      6.0                              14.6
      8.0                              18.5
     10.0                              22.3
     12.0                              26.4
c
```

Listing F.30: Photon_ANSIANS-611-1991_Rotational_ROT_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Rotational (ROT), from Table 3:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                             0.029
    0.015                             0.071
     0.02                              0.11
     0.03                             0.166
     0.04                             0.199
     0.05                             0.222
     0.06                              0.24
     0.08                             0.293
      0.1                             0.357
     0.15                             0.534
      0.2                             0.731
      0.3                              1.14
      0.4                              1.55
      0.5                              1.96
      0.6                              2.34
      0.8                              3.07
      1.0                              3.75
      1.5                              5.24
      2.0                              6.56
      3.0                               8.9
      4.0                              11.0
      5.0                              13.0
      6.0                              14.9
      8.0                              18.9
     10.0                              22.9
     12.0                              27.6
c
```

Listing F.31: Photon_ANSIANS-611-1991_Isotropic_ISO_dedf.txt

```
c
c ANSI/ANS-6.1.1-1991, Isotropic (ISO), from Table 3:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                            0.022
    0.015                            0.057
     0.02                           0.0912
     0.03                            0.138
     0.04                            0.163
     0.05                             0.18
     0.06                            0.196
     0.08                            0.237
      0.1                            0.284
     0.15                            0.436
      0.2                            0.602
      0.3                            0.949
      0.4                              1.3
      0.5                             1.64
      0.6                             1.98
      0.8                             2.64
      1.0                             3.27
      1.5                             4.68
      2.0                             5.93
      3.0                             8.19
      4.0                             10.2
      5.0                             12.1
      6.0                             14.0
      8.0                             17.8
     10.0                             21.6
     12.0                             25.8
c
```

Listing F.32: Photon_ICRP21-1973_dedf.txt

```
c
c ICRP/21-1973, from Table 6, with Modification:
c
c  Energy       Flux-to-dose Conversion Factor
c   [MeV] [(mrem/hr)/(cm$^{-2}\cdot$s$^{-1}$)]
#    de:p                                df:p
      log                                 log
     0.01                              2.778e-3
    0.015                              1.111e-3
     0.02                              5.882e-4
     0.03                              2.564e-4
     0.04                              1.563e-4
     0.05                              1.205e-4
     0.06                              1.111e-4
     0.08                              1.205e-4
      0.1                              1.471e-4
     0.15                              2.381e-4
      0.2                              3.448e-4
      0.3                              5.556e-4
      0.4                              7.692e-4
      0.5                              9.091e-4
      0.6                              1.136e-3
      0.8                               1.47e-3
      1.0                              1.786e-3
      1.5                              2.439e-3
      2.0                               3.03e-3
      3.0                                4.0e-3
      4.0                              4.762e-3
      5.0                              5.556e-3
      6.0                               6.25e-3
      8.0                              7.692e-3
     10.0                              9.091e-3
     20.0                               0.01563
     30.0                               0.02273
     40.0                               0.02941
     50.0                               0.03571
     60.0                               0.04348
     80.0                               0.05882
    100.0                               0.07143
    200.0                                0.1087
    500.0                                0.1724
    1.0e3                                0.2041
    2.0e3                                0.2326
    5.2e3                                0.2703
    1.0e4                                0.2941
    2.0e4                                0.3125
c
```

Listing F.33: Photon_ICRP116-2010_Anterior-Posterior_AP_dedf.txt

```
c
c ICRP/116-2010, Anterior-Posterior (AP), from Table A.1:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                            0.0685
    0.015                             0.156
     0.02                             0.225
     0.03                             0.313
     0.04                             0.351
     0.05                              0.37
     0.06                              0.39
     0.07                             0.413
     0.08                             0.444
      0.1                             0.519
     0.15                             0.748
      0.2                               1.0
      0.3                              1.51
      0.4                               2.0
      0.5                              2.47
    0.511                              2.52
      0.6                              2.91
    0.662                              3.17
      0.8                              3.73
      1.0                              4.49
    1.117                               4.9
     1.33                              5.59
      1.5                              6.12
      2.0                              7.48
      3.0                              9.75
      4.0                              11.7
      5.0                              13.4
      6.0                              15.0
    6.129                              15.1
      8.0                              17.8
     10.0                              20.5
     15.0                              26.1
     20.0                              30.8
     30.0                              37.9
     40.0                              43.1
     50.0                              47.1
     60.0                              50.1
     80.0                              54.5
    100.0                              57.8
    150.0                              63.3
    200.0                              67.3
    300.0                              72.3
    400.0                              75.5
    500.0                              77.5
    600.0                              78.9
    800.0                              80.5
    1.0e3                              81.7
    1.5e3                              83.8
    2.0e3                              85.2
    3.0e3                              86.9
```

```
    4.0e3                       88.1
    5.0e3                       88.9
    6.0e3                       89.5
    8.0e3                       90.2
    1.0e4                       90.7
c
```

Listing F.34: Photon_ICRP116-2010_Posterior-Anterior_PA_dedf.txt

```
c
c ICRP/116-2010, Posterior-Anterior (PA), from Table A.1:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                            0.0184
    0.015                            0.0155
     0.02                             0.026
     0.03                             0.094
     0.04                             0.161
     0.05                             0.208
     0.06                             0.242
     0.07                             0.271
     0.08                             0.301
      0.1                             0.361
     0.15                             0.541
      0.2                             0.741
      0.3                              1.16
      0.4                              1.57
      0.5                              1.98
    0.511                              2.03
      0.6                              2.38
    0.662                              2.62
      0.8                              3.13
      1.0                              3.83
    1.117                              4.22
     1.33                              4.89
      1.5                              5.39
      2.0                              6.75
      3.0                              9.12
      4.0                              11.2
      5.0                              13.1
      6.0                              15.0
    6.129                              15.2
      8.0                              18.6
     10.0                              22.0
     15.0                              30.3
     20.0                              38.2
     30.0                              51.4
     40.0                              62.0
     50.0                              70.4
     60.0                              76.9
     80.0                              86.6
    100.0                              93.2
    150.0                             104.0
    200.0                             111.0
    300.0                             119.0
    400.0                             124.0
    500.0                             128.0
    600.0                             131.0
    800.0                             135.0
    1.0e3                             138.0
    1.5e3                             142.0
    2.0e3                             145.0
    3.0e3                             148.0
```

```
    4.0e3                       150.0
    5.0e3                       152.0
    6.0e3                       153.0
    8.0e3                       155.0
    1.0e4                       155.0
c
```

Listing F.35: Photon_ICRP116-2010_Left_Lateral_LLAT_dedf.txt

```
c
c ICRP/116-2010, Left Lateral (LLAT), from Table A.1:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                            0.0189
    0.015                            0.0416
     0.02                            0.0655
     0.03                              0.11
     0.04                              0.14
     0.05                              0.16
     0.06                             0.177
     0.07                             0.194
     0.08                             0.214
      0.1                             0.259
     0.15                             0.395
      0.2                             0.552
      0.3                             0.888
      0.4                              1.24
      0.5                              1.58
    0.511                              1.62
      0.6                              1.93
    0.662                              2.14
      0.8                              2.59
      1.0                              3.23
    1.117                              3.58
     1.33                               4.2
      1.5                              4.68
      2.0                              5.96
      3.0                              8.21
      4.0                              10.2
      5.0                              12.0
      6.0                              13.7
    6.129                              13.9
      8.0                              17.0
     10.0                              20.1
     15.0                              27.4
     20.0                              34.4
     30.0                              47.4
     40.0                              59.2
     50.0                              69.5
     60.0                              78.3
     80.0                              92.4
    100.0                             103.0
    150.0                             121.0
    200.0                             133.0
    300.0                             148.0
    400.0                             158.0
    500.0                             165.0
    600.0                             170.0
    800.0                             178.0
    1.0e3                             183.0
    1.5e3                             193.0
    2.0e3                             198.0
    3.0e3                             206.0
```

```
    4.0e3                       212.0
    5.0e3                       216.0
    6.0e3                       219.0
    8.0e3                       224.0
    1.0e4                       228.0
c
```

Listing F.36: Photon_ICRP116-2010_Right_Lateral_RLAT_dedf.txt

```
c
c ICRP/116-2010, Right Lateral (RLAT), from Table A.1:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                            0.0182
    0.015                             0.039
     0.02                            0.0573
     0.03                            0.0891
     0.04                             0.114
     0.05                             0.133
     0.06                              0.15
     0.07                             0.167
     0.08                             0.185
      0.1                             0.225
     0.15                             0.348
      0.2                             0.492
      0.3                             0.802
      0.4                              1.13
      0.5                              1.45
    0.511                              1.49
      0.6                              1.78
    0.662                              1.98
      0.8                              2.41
      1.0                              3.03
    1.117                              3.37
     1.33                              3.98
      1.5                              4.45
      2.0                               5.7
      3.0                               7.9
      4.0                              9.86
      5.0                              11.7
      6.0                              13.4
    6.129                              13.6
      8.0                              16.6
     10.0                              19.7
     15.0                              27.1
     20.0                              34.4
     30.0                              48.1
     40.0                              60.9
     50.0                              72.2
     60.0                              82.0
     80.0                              97.9
    100.0                             110.0
    150.0                             130.0
    200.0                             143.0
    300.0                             161.0
    400.0                             172.0
    500.0                             180.0
    600.0                             186.0
    800.0                             195.0
    1.0e3                             201.0
    1.5e3                             212.0
    2.0e3                             220.0
    3.0e3                             229.0
```

```
    4.0e3                       235.0
    5.0e3                       240.0
    6.0e3                       244.0
    8.0e3                       251.0
    1.0e4                       255.0
c
```

Listing F.37: Photon_ICRP116-2010_Rotational_ROT_dedf.txt

```
c
c ICRP/116-2010, Rotational (ROT), from Table A.1:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                            0.0337
    0.015                            0.0664
     0.02                            0.0986
     0.03                             0.158
     0.04                             0.199
     0.05                             0.226
     0.06                             0.248
     0.07                             0.273
     0.08                             0.297
      0.1                             0.355
     0.15                             0.528
      0.2                             0.721
      0.3                              1.12
      0.4                              1.52
      0.5                              1.92
    0.511                              1.96
      0.6                               2.3
    0.662                              2.54
      0.8                              3.04
      1.0                              3.72
    1.117                               4.1
     1.33                              4.75
      1.5                              5.24
      2.0                              6.55
      3.0                              8.84
      4.0                              10.8
      5.0                              12.7
      6.0                              14.4
    6.129                              14.6
      8.0                              17.6
     10.0                              20.6
     15.0                              27.7
     20.0                              34.4
     30.0                              46.1
     40.0                              56.0
     50.0                              64.4
     60.0                              71.2
     80.0                              82.0
    100.0                              89.7
    150.0                             102.0
    200.0                             111.0
    300.0                             121.0
    400.0                             128.0
    500.0                             133.0
    600.0                             136.0
    800.0                             142.0
    1.0e3                             145.0
    1.5e3                             152.0
    2.0e3                             156.0
    3.0e3                             161.0
```

```
    4.0e3                       165.0
    5.0e3                       168.0
    6.0e3                       170.0
    8.0e3                       172.0
    1.0e4                       175.0
c
```

Listing F.38: Photon_ICRP116-2010_Isotropic_ISO_dedf.txt

```
c
c ICRP/116-2010, Isotropic (ISO), from Table A.1:
c
c  Energy Fluence-to-dose Conversion Factor
c   [MeV]              [pSv$\cdot$cm$^{2}$]
#    de:p                              df:p
      log                               log
     0.01                            0.0288
    0.015                             0.056
     0.02                            0.0812
     0.03                             0.127
     0.04                             0.158
     0.05                              0.18
     0.06                             0.199
     0.07                             0.218
     0.08                             0.239
      0.1                             0.287
     0.15                             0.429
      0.2                             0.589
      0.3                             0.932
      0.4                              1.28
      0.5                              1.63
    0.511                              1.67
      0.6                              1.97
    0.662                              2.17
      0.8                              2.62
      1.0                              3.25
    1.117                               3.6
     1.33                               4.2
      1.5                              4.66
      2.0                               5.9
      3.0                              8.08
      4.0                              10.0
      5.0                              11.8
      6.0                              13.5
    6.129                              13.7
      8.0                              16.6
     10.0                              19.6
     15.0                              26.8
     20.0                              33.8
     30.0                              46.1
     40.0                              56.9
     50.0                              66.2
     60.0                              74.1
     80.0                              87.2
    100.0                              97.5
    150.0                             116.0
    200.0                             130.0
    300.0                             147.0
    400.0                             159.0
    500.0                             168.0
    600.0                             174.0
    800.0                             185.0
    1.0e3                             193.0
    1.5e3                             208.0
    2.0e3                             218.0
    3.0e3                             232.0
```

```
    4.0e3                       243.0
    5.0e3                       251.0
    6.0e3                       258.0
    8.0e3                       268.0
    1.0e4                       276.0
c
```

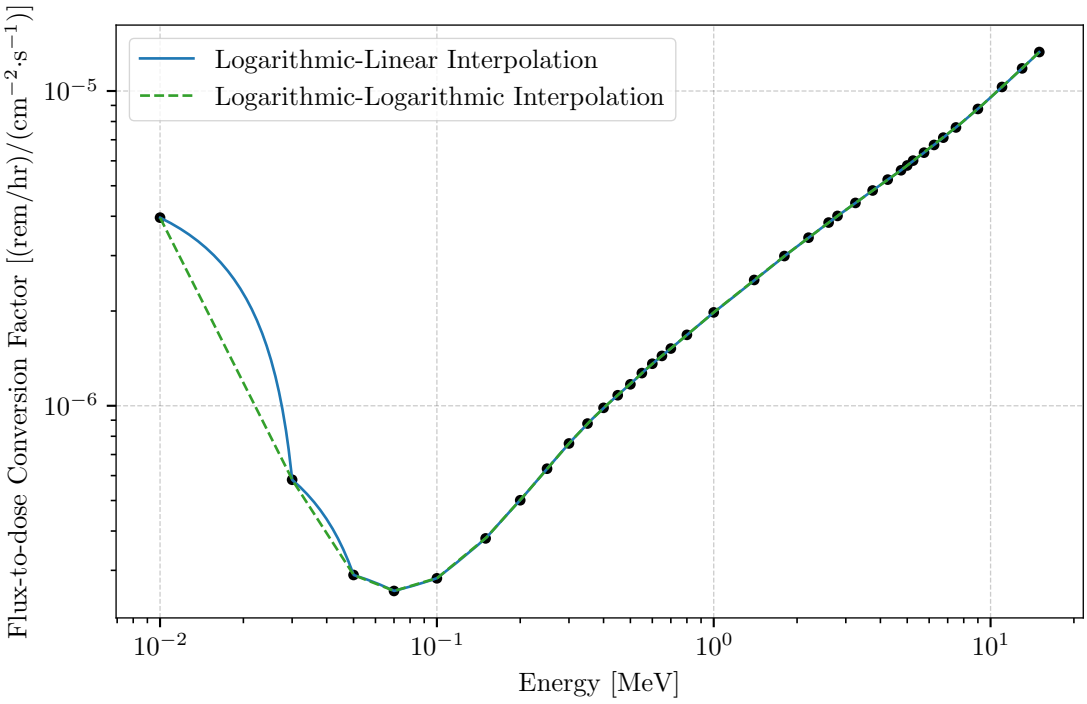


Figure F.26: ANSI/ANS-6.1.1-1977 Photon Flux-to-dose Conversion Factors

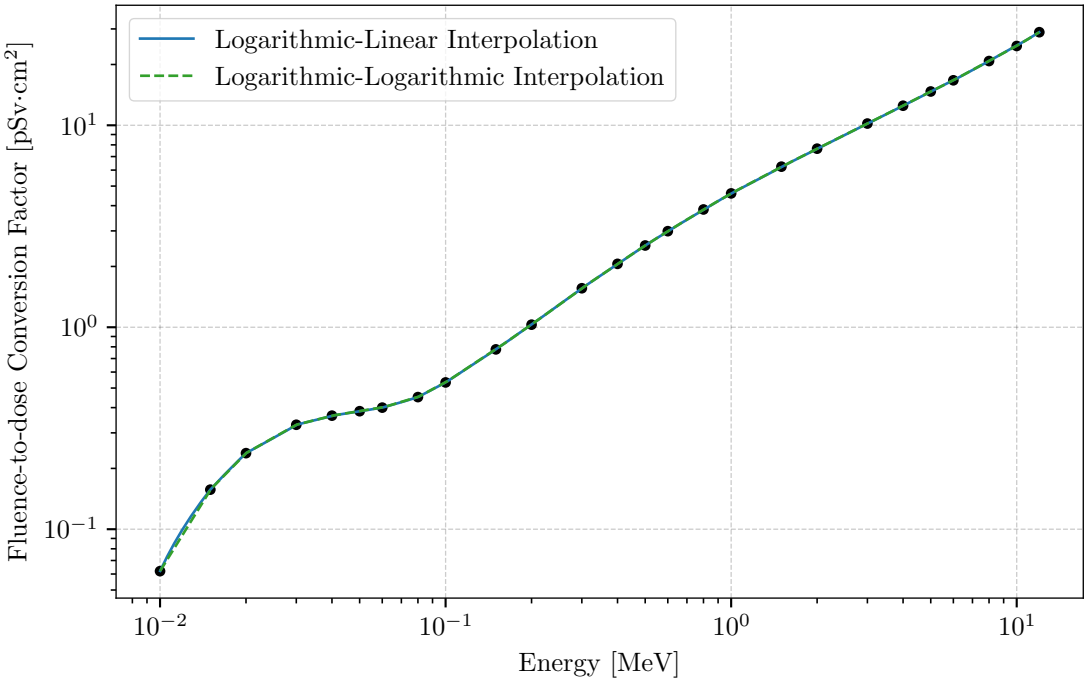


Figure F.27: ANSI/ANS-6.1.1-1991 Anterior-Posterior (AP) Photon Fluence-to-dose Conversion Factors

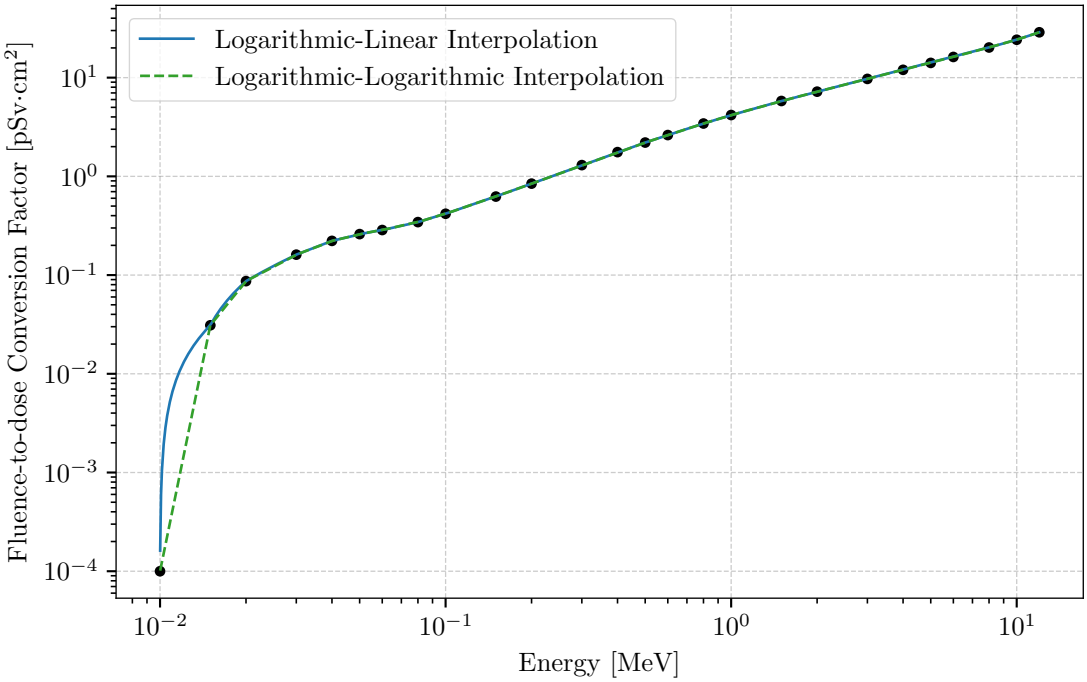


Figure F.28: ANSI/ANS-6.1.1-1991 Posterior-Anterior (PA) Photon Fluence-to-dose Conversion Factors

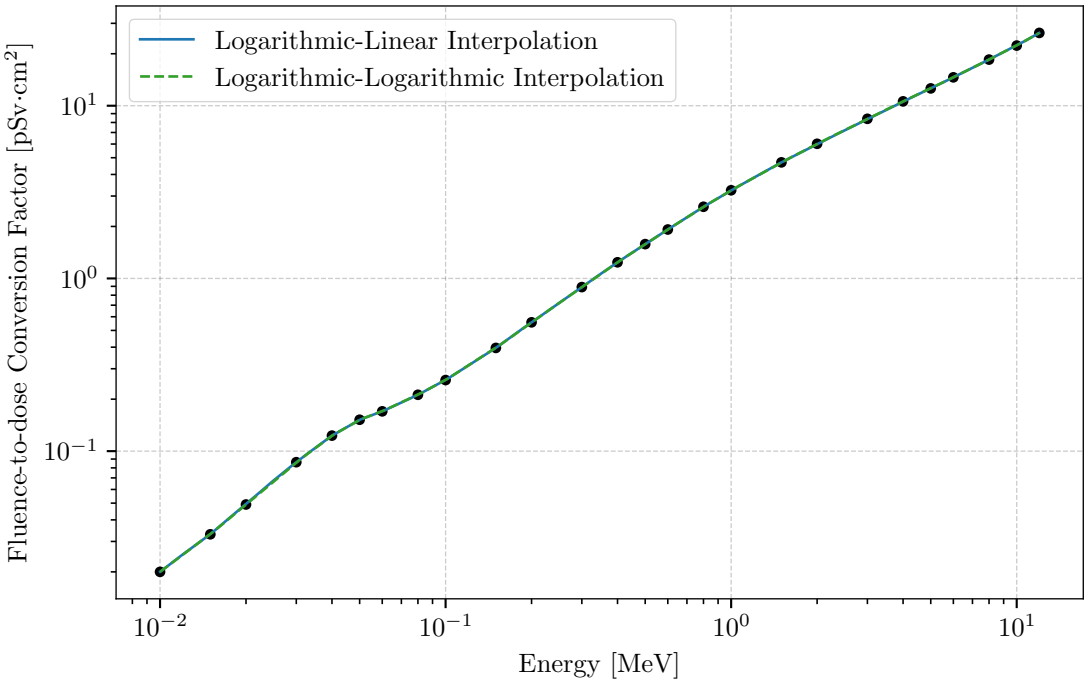


Figure F.29: ANSI/ANS-6.1.1-1991 Lateral (LAT) Photon Fluence-to-dose Conversion Factors

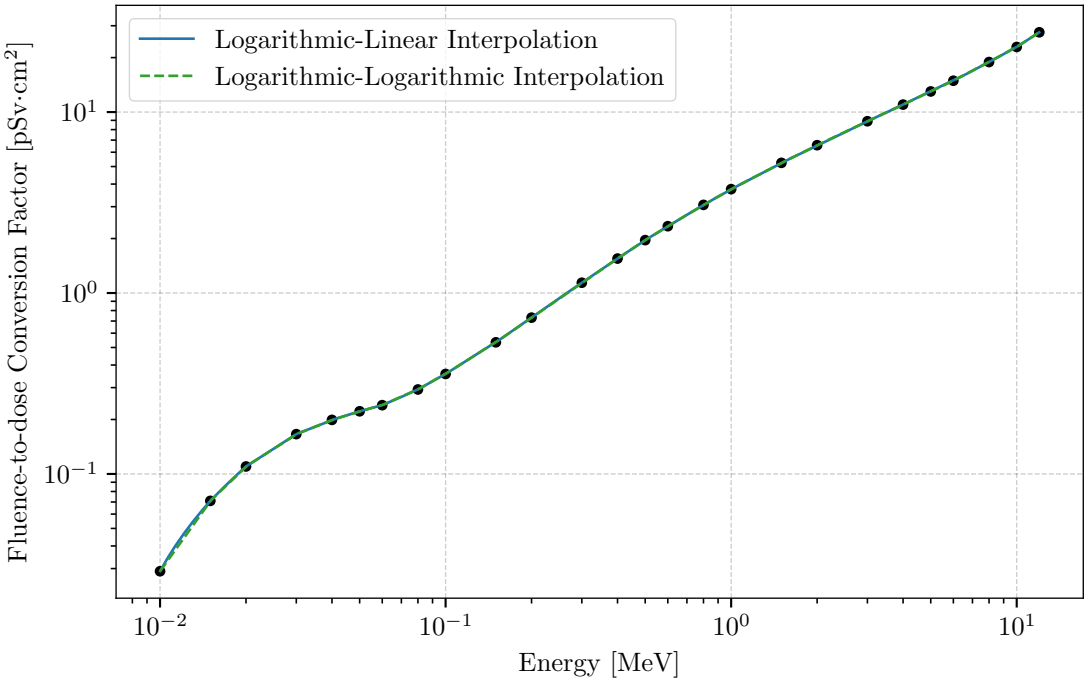


Figure F.30: ANSI/ANS-6.1.1-1991 Rotational (ROT) Photon Fluence-to-dose Conversion Factors

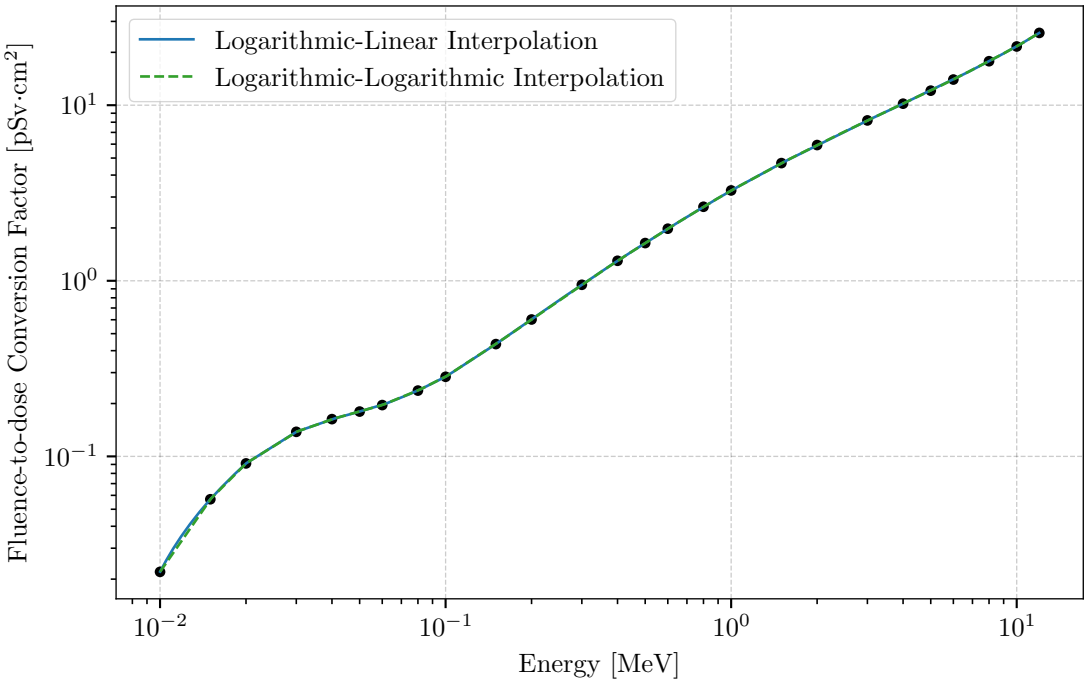


Figure F.31: ANSI/ANS-6.1.1-1991 Isotropic (ISO) Photon Fluence-to-dose Conversion Factors

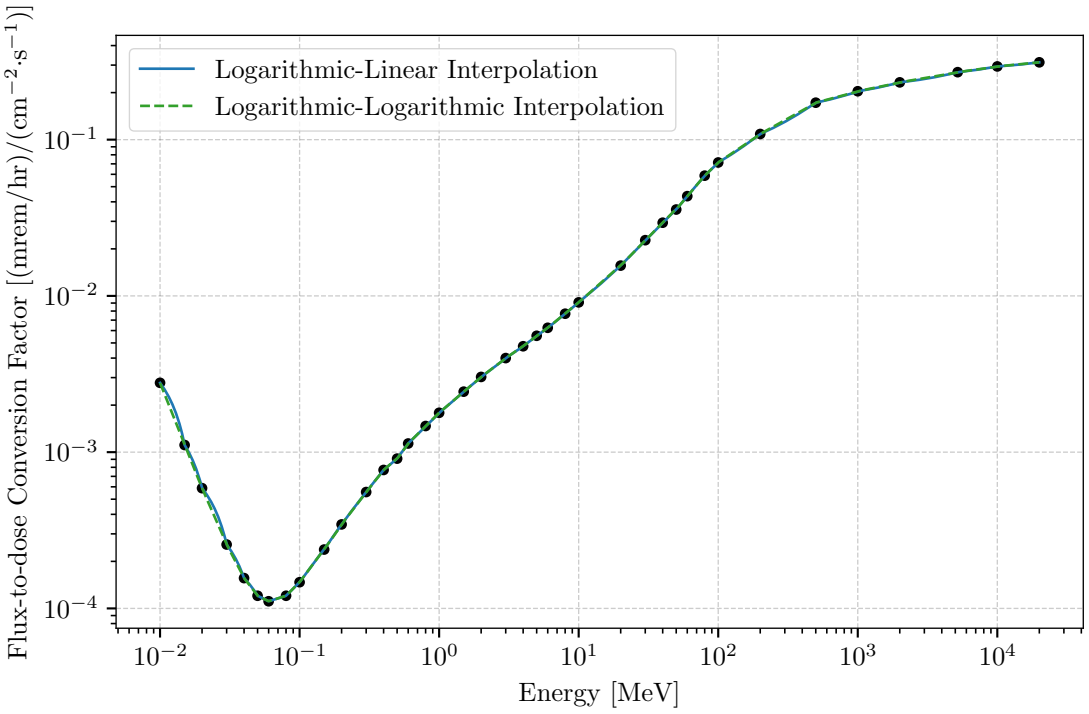


Figure F.32: ICRP/21-1973 Photon Flux-to-dose Conversion Factors

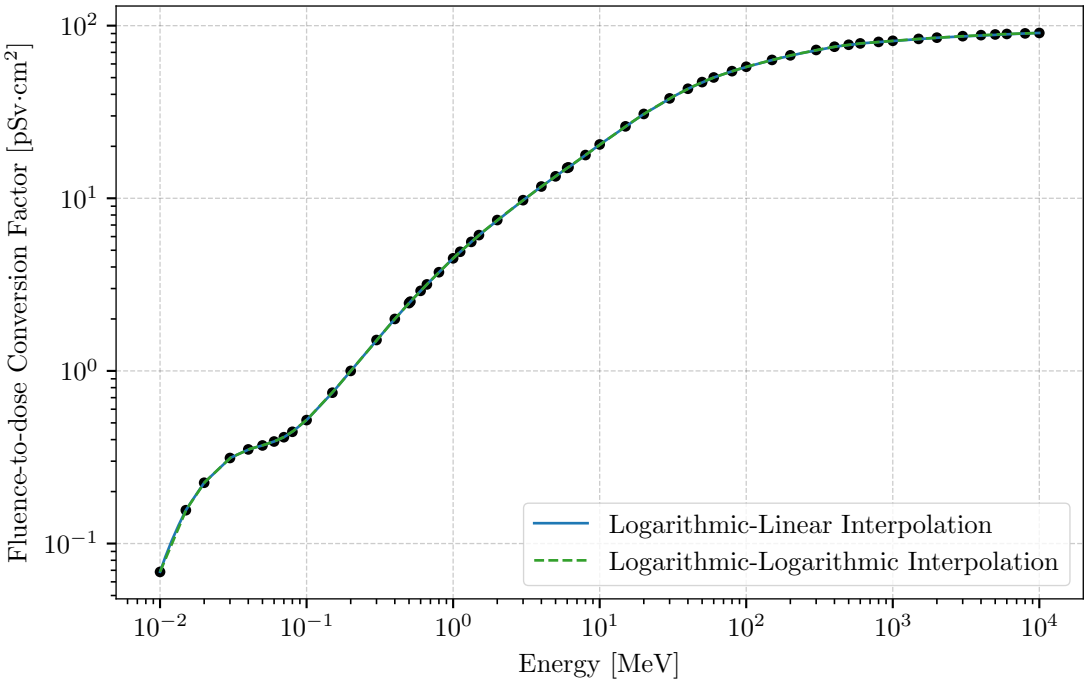


Figure F.33: ICRP/116-2010 Anterior-Posterior (AP) Photon Fluence-to-dose Conversion Factors

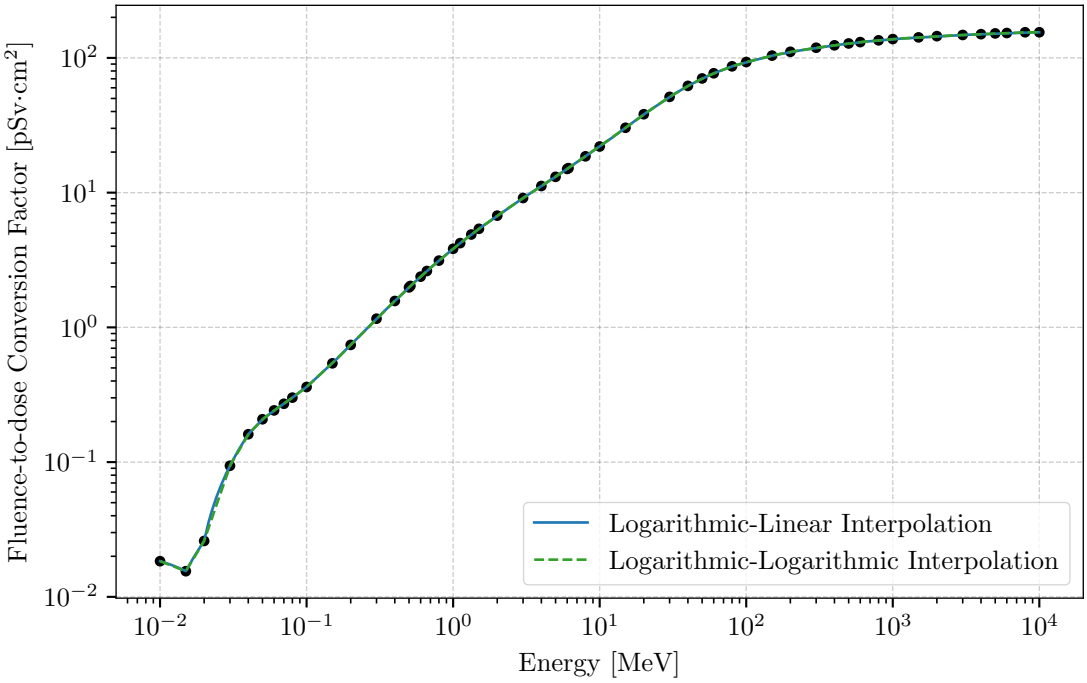


Figure F.34: ICRP/116-2010 Posterior-Anterior (PA) Photon Fluence-to-dose Conversion Factors

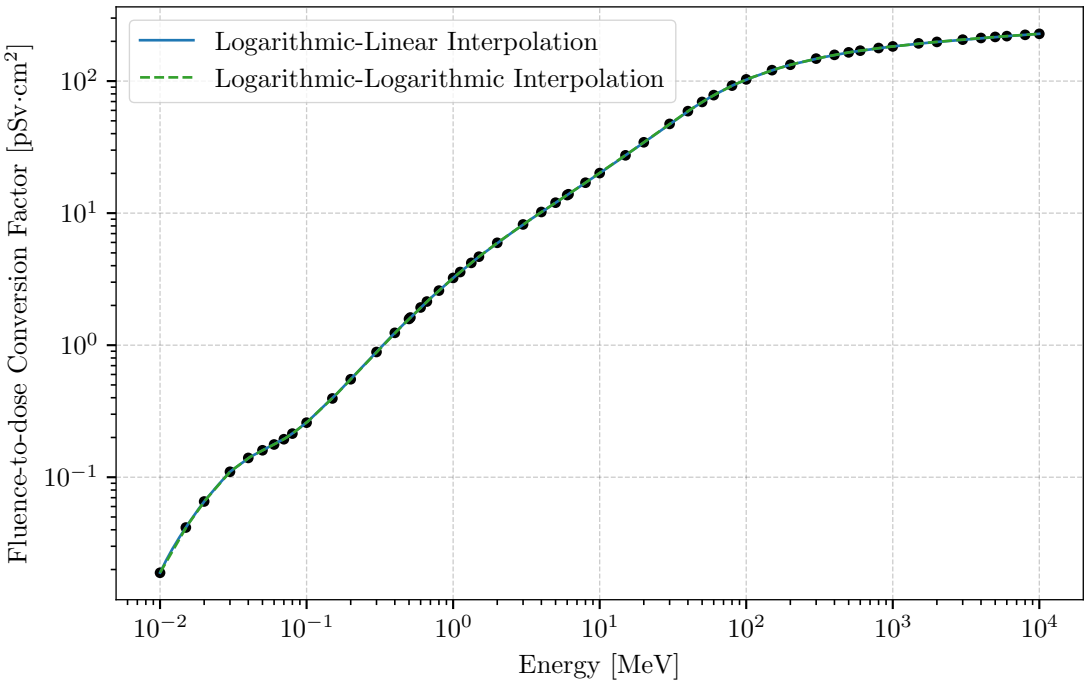


Figure F.35: ICRP/116-2010 Left Lateral (LLAT) Photon Fluence-to-dose Conversion Factors

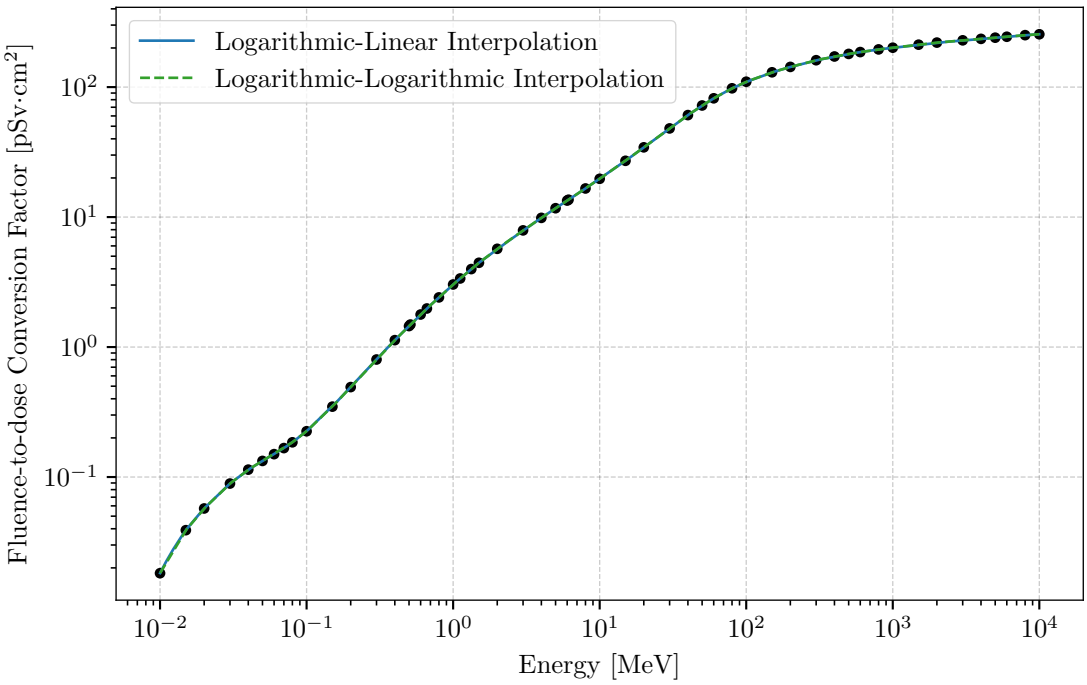


Figure F.36: ICRP/116-2010 Right Lateral (RLAT) Photon Fluence-to-dose Conversion Factors

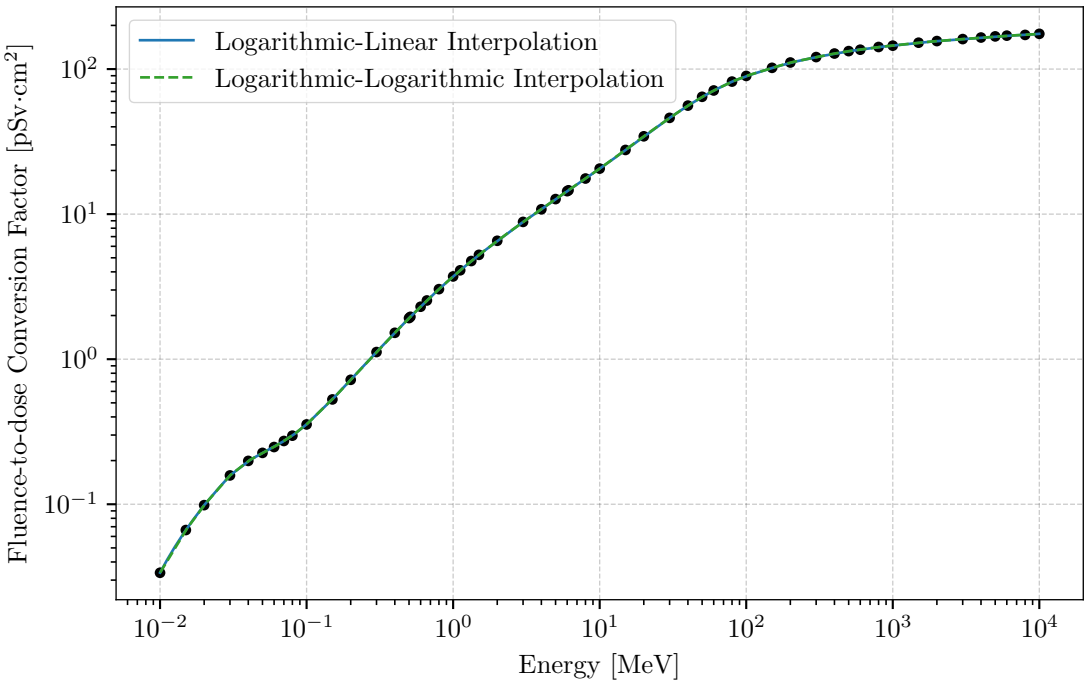


Figure F.37: ICRP/116-2010 Rotational (ROT) Photon Fluence-to-dose Conversion Factors

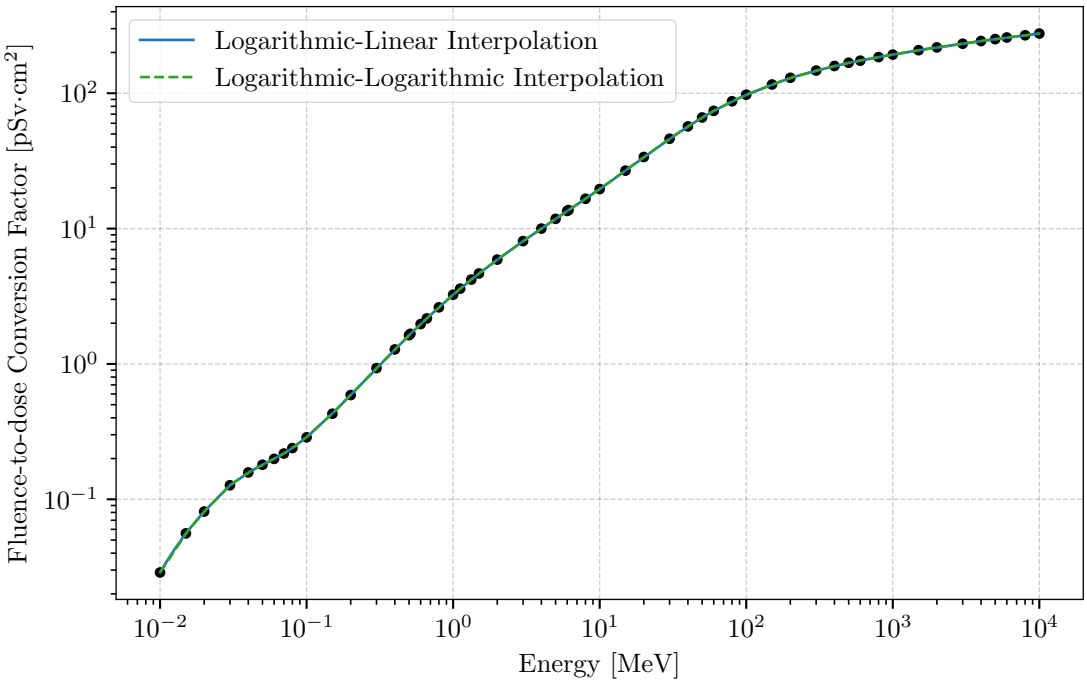


Figure F.38: ICRP/116-2010 Isotropic (ISO) Photon Fluence-to-dose Conversion Factors

F.2 Silicon Displacement Factors

Radiation damage to or effects on electronic components are often of interest. Of particular interest are the absorbed dose in rads and silicon displacement kerma factors. The absorbed dose may be calculated for a specific material by using the FM tally card with an appropriate multiplicative constant c to convert from the default MCNP units to rads. The silicon displacement kermas, however, are given as a function of energy, similar to the biological conversion factors. Therefore, they may be implemented on the DE and DF cards. One source of these kerma factors and a discussion of their significance is available in [365] with additional details in [366].

References

[Citing pages are listed after each reference.]

1. J. E. Sweezy, T. E. Booth, F. B. Brown, J. S. Bull, R. A. Forster, III, J. T. Goorley, H. G. Hughes, III, R. L. Martz, R. E. Prael, A. Sood, A. J. Zukaitis, R. C. Little, H. R. Trellue, M. C. White, M. B. Lee, and S. M. Girard, "MCNP - A General Monte Carlo N-Particle Transport Code, Version 5 - Volume 1: Overview and Theory," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-03-1987 (Revised 2/1/2008), Apr. 2003. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-03-1987 [Page 20]

2. C. J. Werner, J. C. Armstrong, F. B. Brown, J. S. Bull, L. Casswell, L. J. Cox, D. A. Dixon, R. A. Forster, III, J. T. Goorley, H. G. Hughes, III, J. A. Favorite, R. L. Martz, S. G. Mashnik, M. E. Rising, C. J. Solomon, Jr., A. Sood, J. E. Sweezy, A. J. Zukaitis, C. A. Anderson, J. S. Elson, J. W. Durkee, Jr., R. C. Johns, G. W. McKinney, G. E. McMath, J. S. Hendricks, D. B. Pelowitz, R. E. Prael, T. E. Booth, M. R. James, M. L. Fensin, T. A. Wilcox, and B. C. Kiedrowski, "MCNP User's Manual Code Version 6.2," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-17-29981, Oct. 2017. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-17-29981 [Pages 20, 743, and 910]

3. J. A. Kulesza, T. R. Adams, J. C. Armstrong, S. R. Bolding, F. B. Brown, J. S. Bull, T. P. Burke, A. R. Clark, R. A. Forster, III, J. F. Giron, A. S. Grieve, C. J. Josey, R. L. Martz, G. W. McKinney, E. J. Pearson, M. E. Rising, C. J. Solomon, Jr., S. Swaminarayan, T. J. Trahan, C. A. Weaver, S. C. Wilson, and A. J. Zukaitis, "MCNP® Code Version 6.3.1 Theory & User Manual," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-24-24602, Rev. 1, May 2024. [Page 23]

4. P. F. Rose, "ENDF-201, ENDF/B-VI Summary Documentation," Brookhaven National Laboratory, Brookhaven, NY, USA, Tech. Rep. BNL-NCS-17541, Oct. 1991. [Page 34]

5. S. C. Frankle, R. C. Reedy, and P. G. Young, Jr., "ACTI An MCNP Data Library for Prompt Gamma-ray Spectroscopy," in *Proceedings of the 12th Biennial Radiation Protection and Shielding Topical Meeting*, Santa Fe, NM, USA, Apr. 2002. [Page 34]

6. R. J. Howerton, D. E. Cullen, R. C. Haight, M. H. MacGregor, S. T. Perkins, and E. F. Plechaty, "The LLL Evaluated Nuclear Data Library (ENDL): Evaluation Techniques, Reaction Index, and Descriptions of Individual Reactions," Lawrence Livermore National Laboratory, Livermore, CA, USA, Tech. Rep. UCRL-50400, Vol. 15, Part A, Sep. 1975. [Pages 34 and 64]

7. D. E. Cullen, M. H. Chen, J. H. Hubbell, S. T. Perkins, E. F. Plechaty, J. A. Rathkopf, and J. H. Scofield, "Tables and Graphs of Photon-interaction Cross Sections from 10 eV to 100 GeV Derived from the LLNL Evaluated Photon Data Library (EPDL)," Lawrence Livermore National Laboratory, Livermore, CA, USA, Tech. Rep. UCRL-50400; Vol. 6, Oct. 1989. [Pages 34 and 66]

8. M. A. Gardner and R. J. Howerton, "ACTL: Evaluated Neutron Activation Cross-Section Library-evaluation Techniques and Reaction Index," Lawrence Livermore National Laboratory, Livermore, CA, USA, Tech. Rep. UCRL-50400, Vol. 18, 1978. [Pages 34 and 69]

9. E. D. Arthur and P. G. Young, Jr., "Evaluated Neutron-Induced Cross Sections for ^{54,56}Fe to 40 MeV," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-8626-MS (ENDF-304), Dec. 1980. [Pages 34 and 64]

10. D. G. Foster, Jr. and E. D. Arthur, "Average Neutronic Properties of "Prompt" Fission Products," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-9168-MS, Feb. 1982. [Pages 34 and 64]

11. E. D. Arthur, P. G. Young, Jr., A. B. Smith, and C. A. Philis, "New Tungsten Isotope Evaluations for Neutron Energies Between 0.1 and 20 MeV," in *Transactions of the American Nuclear Society*, vol. 39, 1981, p. 793. [Pages 34 and 64]

12. R. E. MacFarlane and D. W. Muir, "The NJOY Nuclear Data Processing System Version 91," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-12740-M, Oct. 1994. [Pages 34 and 123]

13. R. E. MacFarlane, D. W. Muir, and R. M. Boicourt, "The NJOY Nuclear Data Processing System, Volume I: User's Manual," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-9303-M, Vol. I (ENDF-324), May 1982. [Pages 34, 64, and 69]

14. R. E. MacFarlane, D. W. Muir, and R. M. Boicourt, "The NJOY Nuclear Data Processing System, Volume II: The NJOY, RECONR, BROADR, HEATR, and THERMR Modules," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-9303-M, Vol. II (ENDF-324), May 1982. [Pages 34, 64, and 69]

15. C. A. Toccoli, D. K. Parsons, and J. L. Conlin, "Verification of the Re-Released ENDF/B VIII.0 Based Thermal Scattering Libraries," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-21-26731, Jul. 2021. URL: https://www.osti.gov/biblio/1808797-verification-re-released-endf-viii-based-thermal-scattering-libraries [Page 34]

16. R. A. Forster, R. C. Little, J. F. Briesmeister, and J. S. Hendricks, "MCNP Capabilities For Nuclear Well Logging Calculations," *IEEE Transactions on Nuclear Science*, vol. 37, no. 3, pp. 1378–1385, Jun. 1992. DOI: 10.1109/23.57390 [Pages 40 and 174]

17. L. L. Carter and E. D. Cashwell, "Particle Transport Simulation with the Monte Carlo Method," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. TID-26607, 1975. [Pages 49, 83, 99, 144, and 147]

18. I. Lux and L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. CRC Press, 1991. DOI: 10.1201/9781351074834 [Pages 49 and 81]

19. E. D. Cashwell and C. J. Everett, *A Practical Manual on the Monte Carlo Method for Random Walk Problems*. New York, NY, USA: Pergamon Press, Inc., 1959. [Pages 49 and 50]

20. "Information technology — Programming languages — Fortran — Part 1: Base language," International Organization for Standardization, Geneva, Switzerland, ISO/IEC Standard 1539-1:2018, Nov. 2018. URL: https://www.iso.org/standard/72320.html [Pages 49 and 52]

21. "Programming languages — C," International Organization for Standardization, Geneva, Switzerland, ISO/IEC Standard 9899:1999, Dec. 1999, withdrawn. URL: https://www.iso.org/standard/29237.html [Page 49]

22. "Information technology — Programming languages — C++," International Organization for Standardization, Geneva, Switzerland, ISO/IEC Standard 14882:2014, Dec. 2014, withdrawn. URL: https://www.iso.org/standard/64029.html [Pages 49 and 52]

23. G. L. Buffon, *Essai d'Arithmetique Morale*, ser. Supplement a la Naturelle, 1777, vol. 4. [Page 49]

24. A. Hall, "On an Experimental Determination of Pi," *The Messenger of Mathematics*, vol. 2, pp. 113–114, 1873. [Page 49]

25. J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*. Dordrecht, Netherlands: Springer Dordrecht, 1964. DOI: 10.1007/978-94-009-5819-7 [Page 49]

26. M. P.-S. de Laplace, "Theorie Analytique des Probabilities," *Oeuvres Completes de Laplace de L'Academie des Sciences*, vol. 7, no. 2, pp. 356–366, 1786. [Page 49]

27. W. Thomson (Lord Kelvin), "Nineteenth Century Clouds Over the Dynamical Theory of Heat and Light," *Philosophical Magazine*, vol. 6, no. 2, pp. 1–40, Jul. 1901. DOI: 10.1080/14786440109462664 [Page 49]

28. W. W. Wood, "Early History of Computer Simulations in Statistical Mechanics and Molecular Dynamics," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-85-2292, Jun. 1985. URL: https://digital.library.unt.edu/ark:/67531/metadc1063911 [Pages 49 and 50]

29. N. G. Cooper, R. Eckhardt, and N. Shera, Eds., *From Cardinals to Chaos: Reflections on the Life and Legacy of Stanislaw Ulam*. New York, NY, USA: Cambridge University Press, 1989. [Page 50]

30. Unknown, "Fermi Invention Rediscovered at LASL," *The Atom*, Oct. 1966. [Page 50]

31. N. Metropolis and S. Ulam, "The Monte Carlo Method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, Sep. 1949. DOI: 10.1080/01621459.1949.10483310 [Page 50]

32. H. Kahn, "Modifications of the Monte Carlo Method," in *Seminar on Scientific Computation*, ser. Seminar on Scientific Computation, New York, NY, USA, Nov. 1950, pp. 20–27. [Page 50]

33. A. S. Householder, G. E. Forsythe, and H. H. Germond, Eds., *Monte Carlo Methods*, ser. NBS Applied Mathematics Series, 1951, vol. 12, no. 6. [Page 50]

34. D. H. Lehmer, "Mathematical Methods in Large-Scale Computing Units," in *Proceedings of a Second Symposium on Large Scale Digital Calculating Machinery, 1949*. Harvard University Press, 1951, pp. 141–146. [Pages 50 and 222]

35. H. Kahn, "Applications of Monte Carlo," RAND Corporation, Santa Monica, CA, USA, Research Memorandum AECU-3259/RM-1237-AEC, Apr. 1956. URL: https://www.rand.org/pubs/research_memoranda/RM1237.html [Pages 50, 99, and 101]

36. R. R. Johnston, "A General Monte Carlo Neutronics Code," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. LAMS-2856/TID-4500, May 1963. [Page 51]

37. E. D. Cashwell, J. R. Neergaard, W. M. Taylor, and G. D. Turner, "MCN: A Neutron Monte Carlo Code," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-4751, Jan. 1972. [Page 51]

38. E. D. Cashwell, J. R. Neergaard, C. J. Everett, R. G. Schrandt, W. M. Taylor, and G. D. Turner, "Monte Carlo Photon Codes: MCG and MCP," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-5157-MS, Mar. 1973. [Page 51]

39. J. A. Halbleib and T. A. Mehlhorn, "ITS: The Integrated TIGER Series of Coupled Electron/Photon Monte Carlo Transport Codes," Sandia National Laboratory, Albuquerque, NM, USA, Tech. Rep. SAND84-0573, 1984. [Page 51]

40. C. J. Werner, J. S. Bull, C. J. Solomon, Jr., F. B. Brown, G. W. McKinney, M. E. Rising, D. A. Dixon, R. L. Martz, H. G. Hughes, III, L. J. Cox, A. J. Zukaitis, J. C. Armstrong, R. A. Forster, III, and L. Casswell, "MCNP Version 6.2 Release Notes," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-18-20808, 2018. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-18-20808 [Page 52]

41. M. E. Rising, J. C. Armstrong, S. R. Bolding, F. B. Brown, J. S. Bull, T. P. Burke, A. R. Clark, D. A. Dixon, R. A. Forster, III, J. F. Giron, T. S. Grieve, H. G. Hughes, III, C. J. Josey, J. A. Kulesza, R. L. Martz, A. P. McCartney, G. W. McKinney, S. W. Mosher, E. J. Pearson, C. J. Solomon, Jr., S. Swaminarayan, J. E. Sweezy, S. C. Wilson, and A. J. Zukaitis, "MCNP® Code Version 6.3.0 Release

Notes," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-22-33103, Rev. 1, Jan. 2023. DOI: 10.2172/1909545 [Page 52]

42. K. Zieb, H. G. Hughes, III, M. R. James, and X. G. Xu, "Review of Heavy Charged Particle Transport in MCNP6.2," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 886, pp. 77–87, Apr. 2018. DOI: 10.1016/j.nima.2018.01.002 [Pages 54 and 70]

43. E. D. Cashwell and C. J. Everett, "Intersection of a Ray with a Surface of Third or Fourth Degree," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-4299, Sep. 1969. [Pages 58 and 215]

44. R. Kinsey, "Data Formats and Procedures for the Evaluated Nuclear Data File," Brookhaven National Laboratory, Brookhaven, NY, USA, Tech. Rep. BNL-NCS-50496 (ENDF 102), 2nd Ed. (ENDF/B-V), Oct. 1979. [Page 63]

45. M. Herman, A. Trkov, D. A. Brown, N. Holden, and G. Hedstrom, "ENDF-6 Formats Manual," Brookhaven National Laboratory, Upton, NY, USA, Tech. Rep. CSEWG Document ENDF-102 Report BNL-203218-2018-INRE SVN Commit: Revision 215, Feb. 2018. URL: https://www-nds.iaea.org/public/endf/endf-manual.pdf [Pages 63, 472, 573, 574, and 672]

46. J. L. Conlin, "Listing of Available ACE Data Tables," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-17-20709, Oct. 2017. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-17-20709 [Pages 64, 77, 95, and 97]

47. M. W. Asprey, R. B. Lazarus, and R. E. Seamon, "EVXS: A Code to Generate Multigroup Cross Sections from the Los Alamos Master Data File," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-4855, Jun. 1974. [Page 64]

48. R. J. Howerton, R. E. Dye, P. C. Giles, J. R. Kimlinger, S. T. Perkins, and E. F. Plechaty, "Omega Documentation," Lawrence Livermore National Laboratory, Livermore, CA, USA, Tech. Rep. UCRL-50400; Vol. 25, Aug. 1983. [Page 64]

49. E. Storm and H. I. Israel, "Photon Cross Sections from 0.001 to 100 Mev for Elements 1 Through 100," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-3753, Nov. 1967. [Page 66]

50. C. J. Everett and E. D. Cashwell, "MCP Code Fluorescence-routine Revision," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-5240-MS, May 1973. [Pages 66, 67, and 116]

51. H. G. Hughes, III, "Information on the MCPLIB02 Photon Library," Los Alamos National Laboratory, Los Alamos, NM, USA, Memorandum X-6:HGH-93-77/LA-UR-98-0539, Jan. 1998. [Page 66]

52. F. Biggs, L. B. Mendelsohn, and J. B. Mann, "Hartree-Fock Compton Profiles for the Elements," *Atomic Data and Nuclear Data Tables*, vol. 16, no. 3, pp. 201–309, Sep. 1975. DOI: 10.1016/0092-640X(75)90030-3 [Pages 66, 67, and 101]

53. D. E. Cullen, J. H. Hubbell, and L. D. Kissel, "EPDL97: The Evaluated Photon Data Library; '97 Version," Lawrence Livermore National Laboratory, Livermore, CA, USA, Tech. Rep. UCRL-50400; Vol. 6; Rev. 5, 1997. [Page 66]

54. M. C. White, "Photoatomic Data Library MCPLIB04: A New Photoatomic Library based on Data from ENDF/B-VI Release 8," Los Alamos National Laboratory, Los Alamos, NM, USA, Memorandum X-5:MCW-02-111(U)/LA-UR-03-1019, Feb. 2003. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-03-1019 [Page 67]

55. P. Obložinský, "Handbook on Photonuclear Data for Applications: Cross-Sections and Spectra," International Atomic Energy Agency, Vienna, Austria, Tech. Rep. IAEA-TECDOC-1178, 2000. [Page 67]

56. T. Kawano, Y. S. Cho, P. Dimitriou, D. Filipescu, N. Iwamoto, V. Plujko, X. Tao, H. Utsunomiya, V. Varlamov, R. Xu, R. Capote, I. Gheorghe, O. Gorbachenko, Y. L. Jin, T. Renstrøm, M. Sin, K. Stopani, Y. Tian, G. M. Tveten, J. M. Wang, T. Belgya, R. Firestone, S. Goriely, J. Kopecky, M. Krtička, R. Schwengner, S. Siem, and M. Wiedeking, "IAEA Photonuclear Data Library 2019," *Nuclear Data Sheets*, vol. 163, pp. 109–162, Jan. 2020. DOI: 10.1016/j.nds.2019.12.002 [Page 67]

57. J. A. Halbleib, R. P. Kensek, T. A. Mehlhorn, G. D. Valdez, S. M. Seltzer, and M. J. Berger, "ITS Version 3.0: The Integrated TIGER Series of Coupled Electron/Photon Monte Carlo Transport Codes," Sandia National Laboratories, Albuquerque, NM, USA, Tech. Rep. SAND91-1634, Mar. 1992. [Pages 68, 111, 278, and 280]

58. J. U. Koppel and D. H. Houston, "Reference Manual for ENDF Thermal Neutron Scattering Data," General Atomics, San Diego, CA, USA, Tech. Rep. GA-8744; Revised (ENDF-269), Jul. 1978. [Page 69]

59. J. C. Wagner, E. L. Redmond, II, S. P. Palmtag, and J. S. Hendricks, "MCNP: Multigroup/ Adjoint Capabilities," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-12704, Dec. 1993. [Page 70]

60. J. E. Morel, L. J. Lorence, Jr., R. P. Kensek, J. A. Halbleib, and D. P. Sloan, "A Hybrid Multigroup/Continuous-energy Monte Carlo Method for Solving the Boltzmann-Fokker-Planck Equation," *Nuclear Science & Engineering*, vol. 124, no. 3, pp. 369–389, Nov. 1996. DOI: 10.13182/NSE124-369 [Pages 70 and 117]

61. L. J. Lorence, Jr., J. E. Morel, and G. D. Valdez, "Physics Guide to CEPXS: A Multigroup Coupled Electron-photon Cross-section Generating Code; Version 1.0," Sandia National Laboratory, Albuquerque, NM, USA, Tech. Rep. SAND89-1685, Oct. 1989. [Pages 70 and 117]

62. L. J. Lorence, Jr., J. E. Morel, and G. D. Valdez, "User's Guide to CEPXS/ONED-ANT: A One-dimensional Coupled Electron-photon Discrete Ordinates Code Package; Version 1.0," Sandia National Laboratory, Albuquerque, NM, USA, Tech. Rep. SAND89-1661, 1989. [Pages 70 and 117]

63. L. J. Lorence, Jr., W. E. Nelson, and J. E. Morel, "Coupled Electron-photon Transport Calculations Using the Method of Discrete Ordinates," *IEEE Transactions on Nuclear Science*, vol. 32, no. 6, pp. 4416–4420, Dec. 1985. DOI: 10.1109/TNS.1985.4334134 [Pages 70 and 117]

64. S. G. Mashnik, "MCNP6 High-energy Event Generators," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-20-25982, Aug. 2020. DOI: 10.2172/1647182 [Page 70]

65. T. E. Booth, "Monte Carlo Variance Reduction Approaches for Non-Boltzmann Tallies," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-12433, Dec. 1992. URL: https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-12433 [Pages 71 and 458]

66. T. E. Booth, "Pulse Height Tally Variance Reduction in MCNP," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-13955, Aug. 2004. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-13955 [Pages 71, 456, and 458]

67. R. C. Little and R. E. Seamon, "Neutron-induced Photon Production in MCNP," in *Proceedings of Sixth International Conference on Radiation Shielding*, vol. 1, May 1983, p. 151. [Page 77]

68. J. E. Sweezy, "Neutron Next-Event Estimators Kinematics," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-23-30378, Rev. 1, Sep. 2023. DOI: 10.2172/2000872 [Page 82]

69. J. E. Sweezy, "Improvements to Contributions from Neutron Inelastic Scattering for Next-Event Estimators in MCNP® Software," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-23-31561, Rev. 1, Nov. 2023. DOI: 10.2172/2205018 [Page 84]

70. C. J. Everett and E. D. Cashwell, "A Third Monte Carlo Sampler (A Revision and Extension of Samplers I and II)," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-9721-MS, Mar. 1983. [Page 91]

71. H. G. Hughes and R. G. Schrandt, "Gaussian Sampling of Fission Neutron Multiplicity," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. X-6:HGH-86-264, 1986. [Page 93]

72. J. Terrell, "Distribution of Fission Neutron Numbers," *Physical Review*, vol. 108, no. 3, pp. 783–789, Nov. 1957. DOI: 10.1103/PhysRev.108.783 [Pages 93 and 369]

73. J. P. Lestone, "Energy and Isotope Dependence of Neutron Multiplicity Distributions," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-05-0288, 2005. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-05-0288 [Pages 17, 93, 94, and 368]

74. K. Böhnel, "The Effect of Multiplication on the Quantitative Determination of Spontaneously Fissioning Isotopes by Neutron Correlation Analysis," *Nuclear Science & Engineering*, vol. 90, no. 1, pp. 75–82, May 1985. DOI: 10.13182/NSE85-2 [Page 93]

75. R. D. Mosteller and C. J. Werner, "Reactivity Impact of Delayed Neutron Spectra on MCNP Calculations," in *Transactions of the American Nuclear Society*, vol. 82, 2000, pp. 235–236. [Page 95]

76. J. S. Hendricks and R. E. Prael, "Monte Carlo Next-Event Estimates from Thermal Collisions," *Nuclear Science & Engineering*, vol. 109, no. 2, pp. 150–157, Oct. 1991. DOI: 10.13182/NSE91-A28514 [Pages 96 and 139]

77. J. S. Hendricks and R. E. Prael, "MCNP $S(\alpha, \beta)$ Detector Scheme," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-11952, Aug. 1990. [Pages 96, 139, 450, and 569]

78. G. I. Bell and S. Glasstone, *Nuclear Reactor Theory*. New York, NY, USA: Van Nostrand Reinhold Company, 1970. [Pages 97, 118, 119, 120, and 196]

79. J. M. Otter, R. C. Lewis, and L. B. Levitt, "U3R; A Code to Calculate Unresolved Resonance Cross Section Probability Tables," Atomics International, Tech. Rep. AI-AEC-13024, Jul. 1972. [Page 97]

80. R. E. Prael, "Application of the Probability Table Method to Monte Carlo Temperature Difference Calculations," in *Transactions of the American Nuclear Society*, vol. 17, Nov. 1973, p. 261. [Page 97]

81. R. N. Blomquist, R. M. Lell, and E. M. Gelbard, "VIM — A Continuous Energy Monte Carlo Code at ANL; A Review of the Theory and Application of Monte Carlo Methods," Oak Ridge National Laboratory, Oak Ridge, TN, USA, Tech. Rep. ORNL/RSIC-44.,59, Apr. 1980. [Page 97]

82. L. L. Carter, R. C. Little, J. S. Hendricks, and R. E. MacFarlane, "New Probability Table Treatment in MCNP for Unresolved Resonances," in *Proceedings of the ANS Radiation Protection and Shielding Division Topical Conference*, vol. 2, Nashville, TN, USA, Apr. 1998, pp. 341–347. [Page 97]

83. R. D. Mosteller and R. C. Little, "Impact of MCNP Unresolved Resonance Probability Table Treatment on Uranium and Plutonium Benchmarks," in *Proceedings of the Sixth International Conference on Nuclear Criticality Safety (ICNC '99)*, Versailles, France, Sep. 1999, pp. 522–531. [Page 97]

84. L. Koblinger, "Direct Sampling from the Klein-Nishina Distribution for Photon Energies Above 1.4 MeV," *Nuclear Science & Engineering*, vol. 56, no. 2, pp. 218–219, Feb. 1975. DOI: 10.13182/NSE75-A26663 [Pages 99 and 101]

85. R. N. Blomquist and E. M. Gelbard, "An Assessment of Existing Klein-Nishina Monte Carlo Sampling Methods," *Nuclear Science & Engineering*, vol. 83, no. 3, pp. 380–384, Mar. 1983. DOI: 10.13182/NSE83-A17571 [Pages 99 and 101]

86. J. H. Hubbell, W. J. Veigele, E. A. Briggs, R. T. Brown, D. T. Cromer, and R. J. Howerton, "Atomic Form Factors; Incoherent Scattering Functions; and Photon Scattering Cross Sections," *Journal of Physical and Chemical Reference Data*, vol. 4, no. 3, p. 471, 1975. DOI: 10.1063/1.555523 [Pages 10, 100, 101, 102, and 115]

87. G. W. Grodstein, "X-Ray Attenuation Coefficients from 10 keV to 100 MeV," National Bureau of Standards, Gaithersburg, MD, USA, Tech. Rep. Circular No. 583, 1957. [Page 101]

88. A. Bohr and B. R. Mottelson, *Nuclear Structure*, 2nd ed. Singapore: World Scientific, 1998. [Page 105]

89. J. S. Levinger, "Neutron Production by Complete Absorption of High-energy Photons," *Nucleonics*, vol. 6, no. 5, pp. 64–67, 1950. [Page 105]

90. J. S. Levinger, "The High Energy Nuclear Photoeffect," *Physical Review*, vol. 84, no. 1, pp. 43–51, Oct. 1951. DOI: 10.1103/PhysRev.84.43 [Page 105]

91. M. B. Chadwick, P. Obložinský, P. E. Hodgson, and G. Reffo, "Pauli-blocking in the Quasideuteron Model of Photoabsorption," *Physical Review C*, vol. 44, no. 2, pp. 814–823, Aug. 1991. DOI: 10.1103/PhysRevC.44.814 [Page 105]

92. J. R. Wu and C. C. Chang, "Pre-equilibrium Particle Decay in the Photonuclear Reactions," *Physical Review C*, vol. 16, no. 5, pp. 1812–1824, Nov. 1977. DOI: 10.1103/PhysRevC.16.1812 [Page 105]

93. M. Blann, B. L. Berman, and T. T. Komoto, "Precompound-model Analysis of Photoneutron Reaction," *Physical Review C*, vol. 28, no. 6, pp. 2286–2298, Dec. 1983. DOI: 10.1103/PhysRevC.28.2286 [Page 105]

94. M. B. Chadwick, P. G. Young, Jr., and S. Chiba, "Photonuclear Angular-distribution Systematics in the Quasideuteron Regime," *Journal of Nuclear Science and Technology*, vol. 32, no. 11, pp. 1154–1158, Nov. 1995. DOI: 10.1080/18811248.1995.9731830 [Page 105]

95. A. Fassò, A. Ferrari, and P. R. Sala, "Total Giant Resonance Photonuclear Cross Sections for Light Nuclei: A Database for the FLUKA Monte Carlo Transport Code," in *Third Specialists Meeting on Shielding Aspects of Accelerators; Targets; and Irradiation Facilities; SATIF-3*. Sendai; Japan: Organization for Economic Cooperation and Development Nuclear Energy Agency, 1997. [Page 106]

96. M. C. White, "Development and Implementation of Photonuclear Cross-Section Data for Mutually Coupled Neutron-photon Transport Calculations in the Monte Carlo N-Particle (MCNP) Radiation Transport Code," Ph.D. dissertation, University of Florida, Gainesville, FL, USA, 2000. DOI: 10.2172/781606 [Page 106]

97. M. C. White, R. C. Little, and M. B. Chadwick, "Photonuclear Physics in MCNP(X)," in *Proceedings of the ANS Conference on Nuclear Applications of Accelerator Technology*, Long Beach, CA, USA, Nov. 1999. [Page 106]

98. S. Goudsmit and J. L. Saunderson, "Multiple Scattering of Electrons," *Physical Review*, vol. 57, no. 1, pp. 24–29, Jan. 1940. DOI: 10.1103/PhysRev.57.24 [Pages 107, 109, and 114]

99. L. D. Landau, "On the Energy Loss of Fast Particles by Ionization," *Journal of Physics (USSR)*, vol. 8, no. 4, pp. 201–205, 1944. [Pages 107 and 111]

100. O. Blunck and S. Leisegang, "Zum Energieverlust schneller Elektronen in dünnen Schichten," *Zeitschrift für Physik*, vol. 128, no. 4, pp. 500–505, Aug. 1950. DOI: 10.1007/BF01330032 [Pages 107 and 111]

101. M. J. Berger, *Monte Carlo Calculation of the Penetration and Diffusion of Fast Charged Particles*, ser. Methods in Computational Physics. New York, NY, USA: Academic Press, 1963, vol. 1, pp. 135–215. [Pages 107 and 109]

102. S. M. Seltzer, *An Overview of ETRAN Monte Carlo Methods*. Boston, MA, USA: Springer US, 1988, pp. 153–181. DOI: 10.1007/978-1-4613-1059-4_7 [Pages 107 and 114]

103. J. A. Halbleib, *Structure and Operation of the ITS Code System*. Boston, MA, USA: Springer US, 1988, pp. 249–262. DOI: 10.1007/978-1-4613-1059-4_10 [Page 107]

104. R. M. Sternheimer, M. J. Berger, and S. M. Seltzer, "Density Effect for the Ionization Loss of Charged Particles in Various Substances," *Physical Review B*, vol. 26, no. 11, pp. 6067–6076, Dec. 1982. DOI: 10.1103/PhysRevB.26.6067 [Pages 109 and 110]

105. R. M. Sternheimer and R. F. Peierls, "General Expression for the Density Effect for the Ionization Loss of Charged Particles," *Physical Review B*, vol. 3, no. 11, pp. 3681–3692, Jun. 1971. DOI: 10.1103/PhysRevB.3.3681 [Pages 109 and 110]

106. T. A. Carlson, *Photoelectron and Auger Spectroscopy*. New York, NY, USA: Plenum Press, 1975. [Page 110]

107. S. M. Seltzer, *Cross Sections for Bremsstrahlung Production and Electron-impact Ionization*. Boston, MA, USA: Springer US, 1988, pp. 81–114. DOI: 10.1007/978-1-4613-1059-4_4 [Pages 110 and 115]

108. S. M. Seltzer and M. J. Berger, "Bremsstrahlung Spectra from Electron Interactions with Screened Atomic Nuclei and Orbital Electrons," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 12, no. 1, pp. 95–134, Aug. 1985. DOI: 10.1016/0168-583X(85)90707-4 [Pages 110 and 115]

109. S. M. Seltzer and M. J. Berger, "Bremsstrahlung Energy Spectra from Electrons with Kinetic Energy 1 keV – 10 GeV Incident on Screened Nuclei and Orbital Electrons of Neutral Atoms with $Z = 1$ to 100," *Atomic Data and Nuclear Data Tables*, vol. 35, no. 3, pp. 345–418, Nov. 1986. DOI: 10.1016/0092-640X(86)90014-8 [Pages 110 and 115]

110. E. Rutherford, "The Scattering of α and β Particles by Matter and the Structure of the Atom," *Philosophical Magazine*, vol. 21, no. 125, pp. 669–688, 1911. DOI: 10.1080/14786440508637080 [Pages 111 and 114]

111. W. Börsch-Supan, "On the Evaluation of the Function $\phi(\lambda) = \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty} e^{\mu\ln\mu+\lambda\mu}\mathrm{d}\mu$ for Real Values of λ," *Journal of Reseach of the National Bureau of Standards—B. Mathematics and Mathematical Physics*, vol. 65B, no. 4, pp. 245–250, Dec. 1961. URL: https://nvlpubs.nist.gov/nistpubs/jres/65B/jresv65Bn4p245_A1b.pdf [Page 111]

112. O. Blunck and K. Westphal, "Zum Energieverlust energiereicher Elektronen in dünnen Schichten," *Zeitschrift für Physik*, vol. 130, no. 5, pp. 641–649, Oct. 1951. DOI: 10.1007/BF01329538 [Page 112]

113. V. A. Chechin and V. C. Ermilova, "The Ionization-loss Distribution at Very Small Absorber Thickness," *Nuclear Instruments and Methods*, vol. 136, no. 3, pp. 551–558, Aug. 1976. DOI: 10.1016/0029-554X(76)90380-3 [Page 112]

114. S. M. Seltzer, "Electron-photon Monte Carlo Calculations: The ETRAN Code," *International Journal of Radiation Applications and Instrumentation. Part A. Applied Radiation and Isotopes*, vol. 42, no. 10, pp. 917–941, 1991. DOI: 10.1016/0883-2889(91)90050-B [Page 112]

115. D. R. Schaart, J. T. Jansen, J. Zoetelief, and P. F. de Leege, "A Comparison of MCNP4C Electron Transport with ITS 3.0 and Experiment at Incident Energies Between 100 keV and 20 MeV: Influence of Voxel Size; Substeps; and Energy Indexing Algorithm," *Physics in Medicine & Biology*, vol. 47, no. 9, pp. 1459–1484, 2002. DOI: 10.1088/0031-9155/47/9/303 [Page 113]

116. H. G. Hughes, "Improved Logic for Sampling Landau Straggling in MCNP5," in *Proceedings of 2005 Mathematics and Computation Topical Meeting*. American Nuclear Society, 2005. [Page 114]

117. M. E. Riley, C. J. MacCallum, and F. Biggs, "Theoretical Electron-atom Elastic Scattering Cross Sections. Selected Elements; 1 keV to 256 keV," *Atomic Data and Nuclear Data Tables*, vol. 15, no. 5, pp. 443–476, May 1975. DOI: 10.1016/0092-640X(75)90012-1 [Page 114]

118. N. F. Mott, "The Scattering of Fast Electrons by Atomic Nuclei," *Proceedings of the Royal Society A; Mathematical, Physical and Engineering Sciences*, vol. 124, no. 794, pp. 425–442, 1929. DOI: 10.1098/rspa.1929.0127 [Page 114]

119. G. Molière, "Theorie der Streuung schneller geladener Teilchen II: Mehrfach- und Vielfachstreuung," *Zeitschrift für Naturforschung A*, vol. 3, no. 2, pp. 78–97, 1948. DOI: 10.1515/zna-1948-0203 [Page 114]

120. H. A. Bethe and W. Heitler, "On Stopping of Fast Particles and on the Creation of Positive Electrons," *Proceedings of the Royal Society A; Mathematical, Physical and Engineering Sciences*, vol. 146, no. 856, pp. 83–112, 1934. DOI: 10.1098/rspa.1934.0140 [Page 115]

121. H. W. Koch and J. W. Motz, "Bremsstrahlung Cross-section Formulas and Related Data," *Reviews of Modern Physics*, vol. 31, no. 4, pp. 920–955, Oct. 1959. DOI: 10.1103/RevModPhys.31.920 [Page 115]

122. M. J. Berger and S. M. Seltzer, "Bremsstrahlung and Photoneutrons from Thick Tungsten and Tantalum Targets," *Physical Review C*, vol. 2, no. 2, pp. 621–631, Aug. 1970. DOI: 10.1103/PhysRevC.2.621 [Page 115]

123. R. H. Pratt, H. K. Tseng, C. M. Lee, L. D. Kissel, C. MacCallum, and M. E. Riley, "Bremsstrahlung Energy Spectra from Electrons of Kinetic Energy 1 keV $< T <$ 2000 keV Incident on Neutral Atoms $2 < Z < 92$," *Atomic Data and Nuclear Data Tables*, vol. 20, no. 2, pp. 175–209, Aug. 1977. DOI: 10.1016/0092-640X(77)90045-6 [Page 115]

124. R. H. Pratt, H. K. Tseng, C. M. Lee, L. D. Kissel, C. MacCallum, and M. E. Riley, "Bremsstrahlung Energy Spectra from Electrons of Kinetic Energy 1 keV $< T <$ 2000 keV Incident on Neutral Atoms $2 < Z < 92$, Errata," *Atomic Data and Nuclear Data Tables*, vol. 26, no. 5, pp. 477–481, Sep. 1981. DOI: 10.1016/0092-640X(81)90015-2 [Page 115]

125. H. K. Tseng and R. H. Pratt, "Exact Screened Calculations of Atomic-field Bremsstrahlung," *Physical Review A*, vol. 3, no. 1, pp. 100–115, Jan. 1971. DOI: 10.1103/PhysRevA.3.100 [Page 115]

126. H. K. Tseng and R. H. Pratt, "Electron Bremsstrahlung from Neutral Atoms," *Physical Review Letters*, vol. 33, no. 9, pp. 516–518, Aug. 1974. DOI: 10.1103/PhysRevLett.33.516 [Page 115]

127. H. Davies, H. A. Bethe, and L. C. Maximom, "Theory of Bremsstrahlung and Pair Production. II. Integral Cross Section for Pair Production," *Physical Review*, vol. 93, no. 4, pp. 788–795, Feb. 1954. DOI: 10.1103/PhysRev.93.788 [Page 115]

128. H. Olsen, "Outgoing and Ingoing Waves in Final States and Bremsstrahlung," *Physical Review*, vol. 99, no. 4, pp. 1335–1336, Aug. 1955. DOI: 10.1103/PhysRev.99.1335 [Page 115]

129. G. Elwert, "Verschärfte Berechnung von Intensität und Polarisation im kontinuierlichen Röntgenspektrum," *Annalen der Physik*, vol. 426, no. 2, pp. 178–208, 1939. DOI: 10.1002/andp.19394260206 [Page 115]

130. R. J. Jabbur and R. H. Pratt, "High-frequency Region of the Spectrum of Electron and Positron Bremsstrahlung," *Physical Review*, vol. 129, no. 1, pp. 184–190, Jan. 1963. DOI: 10.1103/PhysRev.129.184 [Page 115]

131. R. J. Jabbur and R. H. Pratt, "High-frequency Region of the Spectrum of Electron and Positron Bremsstrahlung II," *Physical Review*, vol. 133, no. 4B, pp. B1090–B1091, Feb. 1964. DOI: 10.1103/PhysRev.133.B1090 [Page 115]

132. J. H. Hubbell and I. Øverbø, "Relativistic Atomic Form Factors and Photon Coherent Scattering Cross Sections," *Journal of Physical and Chemical Reference Data*, vol. 8, no. 1, pp. 69–105, 1979. DOI: 10.1063/1.555593 [Page 115]

133. H. K. Tseng and R. H. Pratt, "Electron Bremsstrahlung Energy Spectra Above 2 MeV," *Physical Review A*, vol. 19, no. 4, pp. 1525–1528, 1979. DOI: 10.1103/PhysRevA.19.1525 [Page 115]

134. E. Haug, "Bremsstrahlung and Pair Production in the Field of Free Electrons," *Zeitschrift für Naturforschung A*, vol. 30, no. 9, pp. 1099–1113, 1975. DOI: 10.1515/zna-1975-0901 [Page 115]

135. C. Møller, "Zur Theorie des Durchgangs schneller Elektronen durch Materie," *Annalen der Physik*, vol. 406, no. 5, pp. 531–585, 1932. DOI: 10.1002/andp.19324060506 [Page 116]

136. D. P. Sloan, "A New Multigroup Monte Carlo Scattering Algorithm Suitable for Neutral and Charged-particle Boltzmann and Fokker-Planck Calculations," Sandia National Laboratories, Tech. Rep. SAND83-7094, May 1983. [Page 117]

137. K. J. Adams and M. Hart, "Multigroup Boltzmann-Fokker-Planck Electron Transport Capability in MCNP," in *Transactions of the American Nuclear Society*, vol. 73, 1995, p. 334. [Page 117]

138. E. E. Lewis and W. F. Miller, *Computational Methods of Neutron Transport*. LaGrange Park, IL, USA: American Nuclear Society, 1993. URL: https://www.ans.org/store/item-350016 [Pages 118, 119, 120, and 157]

139. D. Mihalas and B. Weibel-Mihalas, *Foundations of Radiation Hydrodynamics*. Dover Publications, 1999. [Page 119]

140. D. Mihalas and B. Weibel-Mihalas, *Foundations of Radiation Hydrodynamics (eBook)*. Dover Publications, 1999. URL: https://store.doverpublications.com/0486135888.html [Page 119]

141. A. Dubi, *CRC Handbook of Nuclear Reactors Calculations*. Boca Raton, FL, USA: CRC Press; Inc., 1986, vol. 2, ch. Monte Carlo Calculations for Nuclear Reactors (Chapter 2). [Pages 120 and 122]

142. J. E. Stewart, "A General Point-on-a-ring Detector," in *Transactions of the American Nuclear Society*, vol. 28, 1978, p. 643. [Page 130]

143. R. A. Forster, "Ring Detector and Angle Biasing," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. TD-6-8-79, Jul. 1979. [Page 130]

144. E. C. Snow and J. D. Court, "Radiography Image Detector Capability in MCNP4B," in *Transactions of the American Nuclear Society*, vol. 79, 1998, p. 99. [Page 132]

145. L. S. Waters, J. S. Hendricks, and G. W. McKinney, "MCNPX User's Manual, Version 2.4.0," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-CP-02-408, Sep. 2002. [Page 132]

146. S. P. Pederson, R. A. Forster, III, and T. E. Booth, "Confidence Interval Procedures for Monte Carlo Transport Simulations," *Nuclear Science and Engineering*, vol. 127, no. 1, pp. 54–77, Sep. 1997, Los Alamos National Laboratory Tech. Rep. LA-UR-95-1265. DOI: 10.13182/NSE97-A1921 [Pages 143, 154, 155, 156, and 157]

147. G. P. Estes and E. D. Cashwell, "MCNP1B Variance Error Estimator," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. TD-6-27-78, Aug. 1978. [Pages 149, 154, 166, and 851]

148. A. Dubi, "On the Analysis of the Variance in Monte Carlo Calculations," *Nuclear Science and Engineering*, vol. 72, no. 1, pp. 108–110, Oct. 1979. DOI: 10.13182/NSE79-A19313 [Page 149]

149. I. Lux, "On Efficient Estimation of Variances," *Nuclear Science & Engineering*, vol. 92, no. 4, pp. 607–608, Apr. 1986. DOI: 10.13182/NSE86-A18617 [Page 149]

150. S. P. Pederson, "Mean Estimation in Highly Skewed Samples," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-12114-MS, Mar. 1991. [Pages 154 and 163]

151. T. E. Booth, "Analytic Comparison of Monte Carlo Geometry Splitting and Exponential Transform," in *Transactions of the American Nuclear Society*, 64, Ed., 1991, p. 303. [Page 156]

152. T. E. Booth, "A Caution on Reliability Using OptimalVariance Reduction Parameters," in *Transactions of the American Nuclear Society*, 66, Ed., 1991, p. 278. [Page 156]

153. T. E. Booth, "Analytic Monte Carlo Score Distributions for Future Statistical Confidence Interval Studies," *Nuclear Science & Engineering*, vol. 112, no. 2, pp. 159–169, Oct. 1992. DOI: 10.13182/NSE92-A28411 [Page 156]

154. R. A. Forster, "A New Method of Assessing the Statistical Convergence of Monte Carlo Solutions," in *Transactions of the American Nuclear Society*, vol. 64, 1991, p. 305. [Page 157]

155. R. A. Forster, S. P. Pederson, and T. E. Booth, "Two Proposed Convergence Criteria for Monte Carlo Solutions," in *Transactions of the American Nuclear Society*, vol. 64, 1991, p. 305. [Pages 157 and 167]

156. J. R. Hosking and J. R. Wallis, "Parameter and Quantile Estimation for the Generalized Pareto Distribution," *Technometrics*, vol. 29, no. 3, pp. 339–349, Aug. 1987. DOI: 10.2307/1269343 [Page 160]

157. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing (Fortran Version)*. Cambridge University Press, 1990. [Page 160]

158. M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods; Volume I: Basics*. New York, NY, USA: John Wiley & Sons, 1987. [Page 163]

159. E. E. Cureton, "The Teacher's Corner: Unbiased Estimation of the Standard Deviation," *The American Statistician*, vol. 22, no. 1, pp. 22–22, Feb. 1968. DOI: 10.1080/00031305.1968.10480435 [Page 167]

160. T. E. Booth, "A Sample Problem for Variance Reduction in MCNP," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-10363-MS, Oct. 1985, See also Los Alamos National Laboratory Tech. Rep. LA-UR-12-25907. URL: https://mcnp.lanl.gov/pdf_files/la-10363.pdf [Pages 173, 181, and 196]

161. T. E. Booth and J. S. Hendricks, "Importance Estimation in Forward Monte Carlo Calculations," *Fusion Science and Technology*, vol. 5, no. 1, pp. 90–100, Jan. 1984. DOI: 10.13182/FST84-A23082 [Page 181]

162. F. H. Clark, "The Exponential Transform as an Importance-sampling Device; A Review," Oak Ridge National Laboratory, Oak Ridge, TN, USA, Tech. Rep. ORNL-RSIC-14, Jan. 1966. [Page 183]

163. P. K. Sarkar and M. A. Prasad, "Prediction of Statistical Error and Optimization of Biased Monte Carlo Transport Calculations," *Nuclear Science & Engineering*, vol. 70, no. 3, pp. 243–261, Jun. 1979. DOI: 10.13182/NSE79-A20146 [Page 183]

164. J. S. Hendricks, "Construction of Equiprobable Bins for Monte Carlo Calculation," in *Transactions of the American Nuclear Society*, vol. 35, 1980, p. 247. [Pages 188 and 189]

165. T. J. Urbatsch, R. A. Forster, III, R. E. Prael, and R. J. Beckman, "Estimation and Interpretation of k_{eff} Confidence Intervals in MCNP," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-12658, Nov. 1995. [Pages 196, 199, 205, 206, 207, and 208]

166. C. D. Harmon, II, R. D. Busch, J. F. Briesmeister, and R. A. Forster, III, "Criticality Calculations with MCNP™: A Primer," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-12827-M, Aug. 1994. [Page 196]

167. F. B. Brown, "Fundamentals of Monte Carlo Particle Transport," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-05-4983, 2005. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-05-4983 [Pages 198 and 199]

168. S. Nakamura, *Computational Methods in Engineering and Science*. Malabar, FL, USA: R. E. Krieger Publishing Company, 1986. [Page 198]

169. T. Ueki and F. B. Brown, "Stationarity Diagnostics Using Shannon Entropy in Monte Carlo Criticality Calculation I: F Test," in *Transactions of the American Nuclear Society*, 87, Ed., 2002, p. 156. [Pages 199 and 209]

170. T. Ueki and F. B. Brown, "Stationarity and Source Convergence Diagnostics in Monte Carlo Criticality Calculation," in *Proceedings of 2003 American Nuclear Society Mathematics & Computation Division Topical Meeting (M&C2003)*, Gatlinburg, TN, USA, Apr. 2003. [Pages 199 and 209]

171. E. M. Gelbard and R. E. Prael, "Computations of Standard Deviations in Eigenvalue Calculations," *Progress in Nuclear Energy*, vol. 24, no. 1–3, pp. 237–241, 1990. DOI: 10.1016/0149-1970(90)90041-3 [Pages 199, 207, 210, and 212]

172. G. D. Spriggs and R. D. Busch, "On the Definition of Neutron Lifetimes in Multiplying and Nonmultiplying Systems," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-13260-MS, Jan. 1997. [Pages 200 and 205]

173. M. Halperin, "Almost Linearly-optimum Combination of Unbiased Estimates," *Journal of the American Statistical Association*, vol. 56, no. 293, pp. 36–43, 1961. DOI: 10.1080/01621459.1961.10482088 [Pages 205 and 207]

174. R. C. Gast and N. R. Candelore, "The Recap-12 Monte Carlo Eigenfunction Strategy and Uncertainties," Westinghouse Electric Corporation, Pittsburgh, PA, USA, Tech. Rep. WAPD-TM-1127, 1974. [Pages 206 and 210]

175. S. S. Shapiro and M. B. Wilk, "An Analysis of Variance Test for Normality," *Biometrika*, vol. 52, no. 3–4, pp. 591–611, Dec. 1965. DOI: 10.1093/biomet/52.3-4.591 [Page 208]

176. R. B. D'Agostino, "An Omnibus Test of Normality for Moderate and Large Size Samples," *Biometrika*, vol. 58, no. 2, pp. 341–348, Aug. 1971. DOI: 10.1093/biomet/58.2.341 [Page 208]

177. L. L. Carter, T. L. Miles, and S. E. Binney, "Quantifying the Reliability of Uncertainty Predictions in Monte Carlo Fast Reactor Physics Calculations," *Nuclear Science & Engineering*, vol. 113, no. 4, pp. 324–338, Apr. 1993. DOI: 10.13182/NSE93-A15332 [Page 210]

178. J. S. Hendricks, "Calculation of Cell Volumes and Surface Areas in MCNP," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-8113-MS, Jan. 1980. [Page 214]

179. K. M. Case, G. Placzek, F. de Hoffman, B. G. Carlson, and M. Goldstein, "Introduction to the Theory of Neutron Diffusion," Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. LAMD-1273, Jun. 1953. [Pages 220 and 286]

180. B. Spain, *Analytical Conics*. Pergamon Press, 1957. [Pages 221 and 222]

181. F. B. Brown, "Random Number Generation with Arbitrary Strides," in *Transactions of the American Nuclear Society*, vol. 71, 1994, p. 202. [Page 223]

182. F. B. Brown and Y. Nagaya, "The MCNP5 Random Number Generator," in *Transactions of the American Nuclear Society*, vol. 87, 2002, pp. 230–232. [Page 223]

183. J. S. Hendricks, "Effects of Changing the Random Number Stride in Monte Carlo Calculations," *Nuclear Science and Engineering*, vol. 109, no. 1, pp. 86–91, Sep. 1991. DOI: 10.13182/NSE91-A23846 [Page 223]

184. T. E. Booth, "Bad Estimates as a Function of Exceeding the MCNP Random Number Stride," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-14-23159, May 2014. DOI: 10.2172/1130484 [Page 223]

185. C. Doty-Humphrey, "PractRand," Website, 2019. URL: https://pracrand.sourceforge.net [Page 223]

186. C. J. Josey, "Reassessing the MCNP Random Number Generator," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-23-25111, Rev. 1, Sep. 2023. DOI: 10.2172/1998091 [Page 224]

187. J. E. Olhoeft, "The Doppler Effect for a Non-uniform Temperature Distribution in Reactor Fuel Elements," Westinghouse Electric Corporation, Pittsburgh, PA, USA, Tech. Rep. WCAP-2048, 1962. [Page 224]

188. H. Takahashi, "Monte Carlo Method for Geometrical Perturbation and its Application to the Pulsed Fast Reactor," *Nuclear Science & Engineering*, vol. 41, no. 2, pp. 259–270, Aug. 1970. DOI: 10.13182/NSE70-A20712 [Page 224]

189. M. C. Hall, "Monte Carlo Perturbation Theory in Neutron Transport Calculations," Ph.D. dissertation, University of London, London, UK, 1980. [Page 224]

190. M. C. Hall, "Cross-section Adjustment with Monte Carlo Sensitivities: Application to the Winfrith Iron Benchmark," *Nuclear Science & Engineering*, vol. 81, no. 3, pp. 423–431, Jul. 1982. DOI: 10.13182/NSE82-A20283 [Page 224]

191. H. Rief, "Generalized Monte Carlo Perturbation Algorithms for Correlated Sampling and a Second-order Taylor Series Approach," *Annals of Nuclear Energy*, vol. 11, no. 9, pp. 455–476, 1984. DOI: 10.1016/0306-4549(84)90064-1 [Page 224]

192. G. W. McKinney, "A Monte Carlo (MCNP) Sensitivity Code Development and Application," Master's thesis, University of Washington, Seattle, WA, USA, 1984. [Page 224]

193. G. W. McKinney, "Theory Related to the Differential Operator Perturbation Technique," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. X-6:GWM-94-124, 1994. [Page 225]

194. A. K. Hess, L. L. Carter, J. S. Hendricks, and G. W. McKinney, "Verification of the MCNP Perturbation Correction Feature for Cross-Section Dependent Tallies," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-13520, Oct. 1998. [Page 225]

195. G. W. McKinney and J. L. Iverson, "Verification of the Monte Carlo Differential Operator Technique for MCNP™," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-13098, Feb. 1996. [Page 229]

196. J. A. Favorite and D. K. Parsons, "Second-order Cross Terms in Monte Carlo Differential Operator Perturbation Estimates," in *Proceedings of International Conference; Mathematical Methods for Nuclear Applications*, Salt Lake City, UT, USA, Sep. 2001. [Page 229]

197. J. A. Favorite, "An Alternative Implementation of the Differential Operator (Taylor Series) Perturbation Method for Monte Carlo Criticality Problems," *Nuclear Science & Engineering*, vol. 142, no. 3, pp. 327–341, Nov. 2002. DOI: 10.13182/NSE02-A2311 [Pages 229 and 515]

198. J. F. Briesmeister *et al.*, "MCNP™—A General Monte Carlo N-Particle Transport Code," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-13709-M, Dec. 2000, Code Version 4C. [Page 251]

199. A. Fassò, A. Ferrari, J. Ranft, P. R. Sala, G. R. Stevenson, and J. M. Zazula, "FLUKA92," in *Proceedings of the Workshop on Simulating Accelerator Radiation Environments (SARE1)*, A. Palounek, Ed., Santa Fe, NM, USA, Jan. 1994, pp. 134–144, Los Alamos National Laboratory Tech. Rep. LA-12835-C. [Page 252]

200. K. K. Gudima, G. A. Osokov, and V. D. Toneev, "Model for Pre-Equilibrium Decay of Excited Nuclei," *Soviet Journal of Nuclear Physics*, vol. 21, no. 2, pp. 138–143, Feb. 1975. [Pages 252, 357, 361, and 847]

201. K. K. Gudima, S. G. Mashnik, and V. D. Toneev, "Cascade-exciton Model of Nuclear Reactions," *Nuclear Physics A*, vol. 401, pp. 329–361, 1983. DOI: 10.1016/0375-9474(83)90532-8 [Pages 252, 322, and 357]

202. K. K. Gudima, S. G. Mashnik, and A. J. Sierk, "User Manual for the Code LAQGSM," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-01-6804, Dec. 2001. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-01-6804 [Pages 252 and 357]

203. K. K. Gudima and S. G. Mashnik, "Extension of the LAQGSM03 Code to Describe Photonuclear Reactions up to Tens of GeV," in *Proceedings of the 11th International Conference on Nuclear Reaction Mechanisms*, E. Gadioli, Ed., vol. 126, Varenna, Italy, Jun. 2006, pp. 525–534, Los Alamos National Laboratory Tech. Rep. LA-UR-06-4693. [Pages 252 and 357]

204. S. G. Mashnik and V. D. Toneev, "MODEX—The Program for Calculation of the Energy Spectra of Particles Emitted in the Reactions of Pre-Equilibrium and Equilibrium Statistical Decays," *Communication of the Joint Institute for Nuclear Research (JINR)*, vol. P4-8417, pp. 3–24, 1974, Los

Alamos National Laboratory Tech. Rep. LA-UR-12-20390. DOI: 10.2172/1053861 [Pages 252, 357, 361, and 847]

205. S. G. Mashnik and A. J. Sierk, "Recent Developments of the Cascade-Exciton Model of Nuclear Reactions," in *Proceedings of the International Conference on Nuclear Data for Science and Technology (ND2001)*, Tsukuba, Ibaraki, Japan, Oct. 2001, Los Alamos National Laboratory Tech. Rep. LA-UR-01-5390. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-01-5390 [Pages 252 and 357]

206. S. G. Mashnik, K. K. Gudima, A. J. Sierk, and R. E. Prael, "Improved Intranuclear Cascade Models for the Codes CEM2k and LAQGSM," in *Proceedings of the International Conference on Nuclear Data for Science and Technology (ND2004)*, R. C. Haight, M. B. Chadwick, T. Kawano, and P. Talou, Eds., vol. 769, Santa Fe, NM, USA, Sep. 2004, pp. 1188–1192, Los Alamos National Laboratory Tech. Rep. LA-UR-05-0711. DOI: 10.1063/1.1945220 [Pages 252 and 357]

207. S. G. Mashnik, K. K. Gudima, M. I. Baznat, A. J. Sierk, R. E. Prael, and N. V. Mokhov, "CEM03.01 and LAQGSM03.01 Versions of the Improved Cascade-exciton Model (CEM) and Los Alamos Quark-Gluon String Model (LAQGSM) Codes," Los Alamos National Laboratory, Los Alamos, NM, USA, Research Note X-5-RN (U) 05-11, May 2005. [Pages 252, 322, and 357]

208. S. G. Mashnik, M. I. Baznat, K. K. Gudima, A. J. Sierk, and R. E. Prael, "CEM03 and LAQGSM03: Extension of the CEM2k+GEM2 and LAQGSM Codes to Describe Photonuclear Reactions at Intermediate Energies (30 MeV to 1.5 GeV)," *Journal of Nuclear Radiochemistry Science*, vol. 6, no. 2, pp. A1–A19, 2005, Los Alamos National Laboratory Tech. Rep. LA-UR-05-2013. DOI: 10.14494/jnrs2000.6.2_A1 [Pages 252, 322, and 357]

209. S. G. Mashnik, A. J. Sierk, K. K. Gudima, and M. I. Baznat, "CEM03 and LAQGSM03—New Modeling Tools for Nuclear Applications," *Journal of Physics: Conference Series*, vol. 41, no. 1, pp. 340–351, 2006, Los Alamos National Laboratory Tech. Rep. LA-UR-05-8130. DOI: 10.1088/1742-6596/41/1/037 [Pages 252 and 357]

210. S. G. Mashnik, R. E. Prael, and K. K. Gudima, "Implementation of CEM03.01 into MCNP6 and its Verification and Validation Running Through MCNP6. CEM03.02 Upgrade," Los Alamos National Laboratory, Los Alamos, NM, USA, Memorandum X-3-RN(U)-07-03/LA-UR-06-8652, 2006. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-06-8652 [Pages 252 and 357]

211. S. G. Mashnik, K. K. Gudima, N. V. Mokhov, and R. E. Prael, "LAQGSM03.03 Upgrade and Its Validation," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-07-6198, 2007. URL: https://arxiv.org/abs/0709.1736 [Pages 252 and 357]

212. S. G. Mashnik, K. K. Gudima, R. E. Prael, A. J. Sierk, M. I. Baznat, and N. V. Mokhov, "CEM03.03 and LAQGSM03.03 Event Generators for the MCNP6, MCNPX, and MARS15 Transport Codes," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-08-02931, May 2008. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-08-02931 [Pages 252, 322, and 357]

213. S. G. Mashnik, "Validation and Verification of MCNP6 Against Intermediate and High-Energy Experimental Data and Results by Other Codes," *European Physical Journal Plus*, vol. 126, no. 49, May 2011, Los Alamos National Laboratory Tech. Rep. LA-UR-10-07847. DOI: 10.1140/epjp/i2011-11049-1 [Pages 252 and 357]

214. S. G. Mashnik, "Validation and Verification of MCNP6 Against High-energy Experimental Data and Calculations by Other Codes. I. The CEM Testing Primer," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-11-05129, Sep. 2011. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-11-05129 [Pages 252 and 357]

215. S. G. Mashnik, "Validation and Verification of MCNP6 Against High-energy Experimental Data and Calculations by Other Codes. II. The LAQGSM Testing Primer," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-11-05627, Oct. 2011. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-11-05627 [Pages 252 and 357]

216. S. G. Mashnik and A. J. Sierk, "CEM03.03 User Manual," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-12-01364, Mar. 2012. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-12-01364 [Pages 252, 322, 357, 361, and 847]

217. N. I. of Standards and Technology, "CODATA Recommended Values of the Fundamental Physical Constants: 2018," U.S. Department of Commerce, Washington, D.C., USA, Tech. Rep. NIST SP 961, 2019. URL: https://physics.nist.gov/cuu/pdf/wall_2018.pdf [Page 254]

218. Y. Azuma, P. Barat, G. Bartl, H. Bettin, M. Borys, I. Busch, L. Cibik, G. D'Agostino, K. Fujii, H. Fujimoto, A. Hioki, M. Krumrey, U. Kuetgens, N. Kuramoto, G. Mana, E. Massa, R. Meeß, S. Mizushima, T. Narukawa, A. Nicolaus, A. Pramann, S. A. Rabb, O. Rienitz, C. Sasso, M. Stock, R. D. Vocke, A. Waseda, S. Wundrack, and S. Zakel, "Improved Measurement Results for the Avogadro Constant Using a ^{28}Si-enriched Crystal," *Metrologia*, vol. 52, no. 2, pp. 360–375, Mar. 2015. DOI: 10.1088/0026-1394/52/2/360 [Page 254]

219. K. Hagiwara, K. Hikasa, K. Nakamura, M. Tanabashi, M. Aguilar-Benitez, C. Amsler, R. M. Barnett, P. R. Burchat, C. D. Carone, C. Caso, G. Conforto, O. Dahl, M. Doser, S. Eidelman, J. L. Feng, L. Gibbons, M. Goodman, C. Grab, D. E. Groom, A. Gurtu, K. G. Hayes, J. J. Hernández-Rey, K. Honscheid, C. Kolda, M. L. Mangano, D. M. Manley, A. V. Manohar, J. March-Russell, A. Masoni, R. Miquel, K. Mönig, H. Murayama, S. Navas, K. A. Olive, L. Pape, C. Patrignani, A. Piepke, M. Roos, J. Terning, N. A. Törnqvist, T. G. Trippe, P. Vogel, C. G. Wohl, R. L. Workman, W. M. Yao, B. Armstrong, P. S. Gee, K. S. Lugovsky, S. B. Lugovsky, V. S. Lugovsky, M. Artuso, D. Asner, K. S. Babu, E. Barberio, M. Battaglia, H. Bichsel, O. Biebel, P. Bloch, R. N. Cahn, A. Cattai, R. S. Chivukula, R. D. Cousins, G. Cowan, T. Damour, K. Desler, R. J. Donahue, D. A. Edwards, V. D. Elvira, J. Erler, V. V. Ezhela, A. Fassò, W. Fetscher, B. D. Fields, B. Foster, D. Froidevaux, M. Fukugita, T. K. Gaisser, L. Garren, H. J. Gerber, F. J. Gilman, H. E. Haber, C. Hagmann, J. Hewett, I. Hinchliffe, C. J. Hogan, G. Höhler, P. Igo-Kemenes, J. D. Jackson, K. F. Johnson, D. Karlen, B. Kayser, S. R. Klein, K. Kleinknecht, I. G. Knowles, P. Kreitz, Y. V. Kuyanov, R. Landua, P. Langacker, L. Littenberg, A. D. Martin, T. Nakada, M. Narain, P. Nason, J. A. Peacock, H. R. Quinn, S. Raby, G. Raffelt, E. A. Razuvaev, B. Renk, L. Rolandi, M. T. Ronan, L. J. Rosenberg, C. T. Sachrajda, A. I. Sanda, S. Sarkar, M. H. Schmitt, O. Schneider, D. Scott, W. G. Seligman, M. H. Shaevitz, T. Sjöstrand, G. F. Smoot, S. Spanier, H. Spieler, N. J. C. Spooner, M. Srednicki, A. Stahl, T. Stanev, M. Suzuki, N. P. Tkachenko, G. Valencia, K. van Bibber, M. G. Vincter, D. R. Ward, B. R. Webber, M. Whalley, L. Wolfenstein, J. Womersley, C. L. Woody, and O. V. Zenin, "Review of Particle Physics: Particle Data Group," *Physical Review D*, vol. 66, no. 1, pp. 100 011–10 001 958, Jul. 2002. DOI: 10.1103/PhysRevD.66.010001 [Page 263]

220. "IEEE Standard for Floating-Point Arithmetic," Institute of Electrical and Electronics Engineers (IEEE), Tech. Rep. IEEE Std 754-2019 (Revision of IEEE 754-2008), 2019. DOI: 10.1109/IEEESTD.2019.8766229 [Page 264]

221. F. B. Brown, W. R. Martin, W. Ji, J. L. Conlin, and J. C. Lee, "Stochastic Geometry and HTGR Modeling with MCNP5," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-04-8668, Apr. 2004. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-04-8668 [Pages 299 and 300]

222. F. B. Brown and W. R. Martin, "Stochastic Geometry Capability in MCNP5 for the Analysis of Particle Fuel," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-04-5362, Nov. 2004. [Pages 299 and 300]

223. R. E. Alcouffe, R. S. Baker, J. A. Dahl, E. J. Davis, T. G. Saller, S. A. Turner, R. C. Ward, and R. J. Zerr, "PARTISN: A Time-Dependent, Parallel Neutral Particle Transport Code System Manual, Version 8.27," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-17-29704, Oct. 2017. [Page 300]

224. R. L. Martz and D. L. Crane, "The MCNP6 Book on Unstructured Mesh Geometry: Foundations (U)," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-12-25478, Rev. 1, Mar. 2014. [Page 301]

225. R. L. Martz, "Unstructured Mesh User's Startup Guide," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-12-00795, Feb. 2012. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-12-00795 [Page 301]

226. L. J. Cox, "LNK3DNT Geometry Support: User Guidance for Creating and Embedding," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-11-01654, 2011. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-11-01654 [Page 301]

227. R. L. Martz, "The MCNP6 Book on Unstructured Mesh Geometry: User's Guide for MCNP 6.2," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-17-22442, Mar. 2017. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-17-22442 [Page 307]

228. F. A. Ortega, J. M. Hinrichs, and R. Fresquez, "General Mesh Viewer User's Manual GMV Version 0.8," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-95-2986, Aug. 1995. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-95-2986 [Pages 309, 725, and 740]

229. R. S. Detwiler, R. J. McConn, Jr., T. F. Grimes, S. A. Upton, and E. J. Engel, "Compendium of Material Composition Data for Radiation Transport Modeling," Pacific Northwest National Laboratory, Richland, WA, USA, Tech. Rep. PNNL-15870, Rev. 2, 2021. DOI: 10.2172/1782721 [Page 319]

230. J. S. Hendricks, "MCNPX Model/Table Comparison," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-14030, Mar. 2003. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-14030 [Page 322]

231. W. R. Martin, S. J. Wilderman, F. B. Brown, and G. Yesilyurt, "Implementation of On-The-Fly Doppler Broadening in MCNP," in *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)*, Sun Valley, ID, USA, May 2013, Los Alamos National Laboratory Tech. Rep. LA-UR-13-20662. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-13-20662 [Page 323]

232. W. R. Martin, G. Yesilyurt, F. B. Brown, and S. J. Wilderman, "Implementation of On-the-fly Doppler Broadening in MCNP for Multiphysics Simulation of Nuclear Reactors," University of Michigan, Ann Arbor, MI, USA, Tech. Rep. DE-AC07-05ID14517, Nov. 2012, Final Report for US DOE Nuclear Energy University Programs Project No. 10-897. [Page 323]

233. F. B. Brown, W. R. Martin, G. Yesilyurt, and S. J. Wilderman, "Progress with On-The-Fly Neutron Doppler Broadening in MCNP," in *Transactions of the American Nuclear Society*, vol. 106. Chicago, IL, USA: American Nuclear Society, Jun. 2012, Los Alamos National Laboratory Tech. Rep. LA-UR-12-00423 (Paper) and Los Alamos National Laboratory Tech. Rep. LA-UR-12-22277 (Presentation). URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-12-00423 [Page 323]

234. F. B. Brown, W. R. Martin, G. Yesilyurt, and S. J. Wilderman, "On-the-fly Neutron Doppler Broadening for MCNP," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-12-20338, Apr. 2012. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-12-20338 [Page 323]

235. T. Wilcox, T. Kawano, G. W. McKinney, and J. S. Hendricks, "Correlated Gammas Using CGM and MCNPX," *Progress in Nuclear Energy*, vol. 63, pp. 1–6, Mar. 2013. DOI: 10.1016/j.pnucene.2012.10.002 [Pages 332 and 333]

236. R. E. Prael, L.-C. Liu, and S. Striganov, "Monte Carlo Model for Proton Elastic Scattering from Optical Model Calculations," in *Proceedings of Accelerator Applications in a Nuclear Renaissance (AccApp 2003)*. San Diego, CA, USA; June 1–5: American Nuclear Society, 2003, Los Alamos Tech. Rep. LA-UR-03-2275. URL: https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-03-2275 [Pages 332, 342, and 345]

237. J. M. Verbeke, C. Hagmann, and D. Wright, "Simulation of Neutron and Gamma Ray Emission from Fission and Photofission," Lawrence Livermore National Laboratory, Livermore, CA, USA, Tech. Rep. UCRL-AR-228518, Jan. 2014. [Pages 336 and 369]

238. J. S. Hendricks and B. J. Quiter, "MCNP/X Form Factor Upgrade for Improved Photon Transport," *Nuclear Technology*, vol. 175, no. 1, pp. 150–161, Jul. 2011, Los Alamos National Laboratory Tech. Rep. LA-UR-10-04266. DOI: 10.13182/NT10-17 [Page 336]

239. "Stopping Powers for Electrons and Positrons," International Commission on Radiation Units and Measurements (ICRU), Tech. Rep. ICRU-37, Oct. 1984. [Pages 343 and 346]

240. G. A. Rinker, "Static and Dynamic Muonic-atom Codes—MUON and RURP," *Computer Physics Communications*, vol. 16, no. 2, pp. 221–242, Feb. 1979. DOI: 10.1016/0010-4655(79)90090-0 [Page 344]

241. H. Armstrong, M. R. James, and G. W. McKinney, "Delayed Neutron and Photon Energy Biasing in MCNP6," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-13-23368, May 2013. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-13-23368 [Page 349]

242. R. R. Coveyou, R. R. Bate, and R. K. Osborn, "Effect of Moderator Temperature Upon Neutron Flux in Infinite, Capturing Medium," *Journal of Nuclear Energy (1954)*, vol. 2, no. 3-4, pp. 153–167, 1956. URL: https://www.sciencedirect.com/science/article/pii/0891391955900309 [Page 355]

243. M. Ouisloumen and R. Sanchez, "A Model for Neutron Scattering Off Heavy Isotopes That Accounts for Thermal Agitation Effects," *Nuclear Science and Engineering*, vol. 107, no. 3, pp. 189–200, 1991. DOI: 10.13182/NSE89-186 [Page 355]

244. B. Becker, R. Dagan, and G. Lohnert, "Proof and Implementation of the Stochastic Formula for Ideal Gas, Energy Dependent Scattering Kernel," *Annals of Nuclear Energy*, vol. 36, no. 4, pp. 470–474, 2009. DOI: 10.1016/j.anucene.2008.12.001 [Page 356]

245. E. E. Sunny, F. B. Brown, B. C. Kiedrowski, and W. R. Martin, "Temperature Effects of Resonance Scattering in Free Gas for Epithermal Neutrons," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-11-04886, Aug. 2011. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-11-04886 [Page 356]

246. E. E. Sunny, F. B. Brown, B. C. Kiedrowski, and W. R. Martin, "Temperature Effects of Resonance Scattering for Epithermal Neutrons in MCNP," in *Proceedings of PHYSOR—Advances in Reactor Physics—Linking Research, Industry, and Education*, Knoxville, TN, USA. April 15–20, 2012, pp. 15–20, Los Alamos National Laboratory Tech. Rep. LA-UR-11-06503. URL: https://www.osti.gov/biblio/1121992 [Page 356]

247. F. B. Brown, "Doppler Broadening Resonance Correction for Free-gas Scattering in MCNP6.2," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-19-24824, May 2019. DOI: 10.2172/1523218 [Page 356]

248. H. W. Bertini, "Low-energy Intranuclear Cascade Calculation," *Physical Review*, vol. 131, no. 4, pp. 1801–1821, Aug. 1963. DOI: 10.1103/PhysRev.131.1801 [Page 357]

249. H. W. Bertini, "Intranuclear Cascade Calculation of the Secondary Nucleon Spectra from Nucleon-nucleus Interations in the Energy Range 340 to 2900 MeV and Comparison with Experiment," *Physical Review*, vol. 188, no. 4, pp. 1711–1730, Dec. 1969. DOI: 10.1103/PhysRev.188.1711 [Page 357]

250. Y. Yariv and Z. Fraenkel, "Intranuclear Cascade Calculation of High-Energy Heavy-Ion Interactions," *Physical Review C*, vol. 20, no. 6, pp. 2227–2243, Dec. 1979. DOI: 10.1103/PhysRevC.20.2227 [Page 357]

251. Y. Yariv and Z. Fraenkel, "Inclusive Cascade Calculation of High Energy Heavy Ion Collisions: Effect of Interactions Between Cascade Particles," *Physical Review C*, vol. 24, no. 2, pp. 488–494, Aug. 1981. DOI: 10.1103/PhysRevC.24.488 [Page 357]

252. A. Boudard, J. Cugnon, S. Leray, and C. Volant, "Intranuclear Cascade Model for a Comprehensive Description of Spallation Reaction Data," *Physical Review C*, vol. 66, no. 4, pp. 044 615–044 643, Oct. 2002. DOI: 10.1103/PhysRevC.66.044615 [Page 357]

253. J. J. Gaimard and K. H. Schmidt, "A Reexamination of the Abrasion-ablation Model for the Description of the Nuclear Fragmentation Reaction," *Nuclear Physics A*, vol. 531, no. 3–4, pp. 709–745, Sep. 1991. DOI: 10.1016/0375-9474(91)90748-U [Page 357]

254. A. R. Junghans, M. de Jong, H. G. Clerc, A. V. Ignatyuk, G. A. Kudyaev, and K. H. Schmidt, "Projectile-Fragment Yields as a Probe for the Collective Enhancement in the Nuclear Level Density," *Nuclear Physics A*, vol. 629, no. 3–4, pp. 635–655, Feb. 1998. DOI: 10.1016/S0375-9474(98)00658-7 [Page 357]

255. R. E. Prael and H. Lichtenstein, "User Guide to LCS: The LAHET Code System," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-89-3014, Sep. 1989. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-89-3014 [Pages 357, 361, 364, and 365]

256. R. E. Prael and M. Bozoian, "Adaptation of the Multistage Preequilibrium Model for the Monte Carlo Method (I)," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-88-3238, Sep. 1988. [Pages 358, 365, and 847]

257. V. S. Barashenkov and V. D. Toneev, "Interaction of Particles and Nuclei with Atomic Nuclei [in Russian]," *Atomizdat*, 1972. [Page 361]

258. V. S. Barashenkov, A. S. Il'inov, N. M. Sobolevskiĭ, and V. D. Toneev, "Interaction of Particles and Nuclei of High and Ultrahigh Energy with Nuclei," *Soviet Physics Uspekhi*, vol. 16, no. 1, pp. 31–52, Jan. 1973. DOI: 10.1070/pu1973v016n01abeh005147 [Page 361]

259. S. Furihata, "Statistical Analysis of Light Fragment Production from Medium Energy Proton-induced Reactions," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 171, no. 3, pp. 251–258, Nov. 2000. DOI: 10.1016/S0168-583X(00)00332-3 [Pages 361 and 847]

260. I. Dostrovsky, Z. Fraenkel, and G. Friedlander, "Monte Carlo Calculations of Nuclear Evaporation Processes. III. Applications to Low-Energy Reactions," *Physical Review*, vol. 116, no. 3, pp. 683–702, Nov. 1959. DOI: 10.1103/PhysRev.116.683 [Page 361]

261. F. Atchison, "A Treatment of Medium-energy Particle Induced Fission for Spallation-systems' Calculations," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 259, no. 2, pp. 909–932, Jun. 2007. DOI: 10.1016/j.nimb.2007.03.004 [Page 361]

262. M. I. Baznat, K. K. Gudima, and S. G. Mashnik, "Proton-induced Fission Cross Section Calculation with the LANL Codes CEM2k+GEM2 and LAQGSM+GEM2," in *Proceedings of the American Nuclear Society Annual Meeting, Accelerator Applications 2003 Embedded Topical Meeting, "Accelerator Applications in a Nuclear Renaissance"*. San Diego, CA, USA; June 1—5: American Nuclear Society, Jun. 2003, pp. 976–985, Also Los Alamos National Laboratory Tech. Rep. LA-UR-03-0385. URL: https://www.ans.org/store/item-700302 [Page 361]

263. R. E. Prael, "The LAHET Code System: Introduction, Development, and Benchmarking," in *Proceedings of the Simulating Accelerator Radiation Environments Workshop*, Santa Fe, NM, USA, Jan. 1993, also: LA-UR-93-1216. URL: https://www.osti.gov/biblio/6235552 [Page 364]

264. P. Cloth, D. Filges, G. Sterzenbach, T. W. Armstrong, and B. L. Colborn, "The KFA-version of the High-energy Transport Code HETC and the Generalized Evaluation Code SIMPEL," Kernforschungsanlage Juelich G.m.b.H. Inst. fuer Reaktorentwicklung, Tech. Rep. JUEL-SPEZ-196, Mar. 1983. [Page 365]

265. F. Atchison, "Spallation and Fission in Heavy Metal Nuclei Under Medium Energy Proton Bombardment," in *Targets for Neutron Beam Spallation Sources, Jul-Conf-34*. Kernforschungsanlage Julich GmbH, Jan. 1980. [Page 365]

266. J. Barish, T. A. Gabriel, F. S. Alsmiller, and R. G. Alsmiller, Jr., "HETFIS High-Energy Nucleon-Meson Transport Code with Fission," Oak Ridge National Laboratory, Oak Ridge, TN, USA, Tech. Rep. ORNL-TM-7882, Jul. 1981. [Page 365]

267. N. Ensslin, W. C. Harker, M. S. Krick, D. G. Langner, M. M. Pickrell, and J. E. Stewart, "Application Guide to Neutron Multiplicity Counting," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-13422-M, Nov. 1998. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-13422-M [Pages 368 and 370]

268. P. A. Santi, D. H. Beddingford, and D. R. Mayo, "Revised Prompt Neutron Emission Multiplicity Distributions for ^{236,238}Pu," *Nuclear Physics A*, vol. 756, no. 3–4, pp. 325–332, Jul. 2005. DOI: 10.1016/j.nuclphysa.2005.04.002 [Page 368]

269. J. M. Verbeke, J. Randrup, and R. Vogt, "Fission Reaction Yield Algorithm FREYA User Manual 2.0," Lawrence Livermore National Laboratory, Livermore, CA, USA, Tech. Rep. LLNL-SM-705798, 2016. [Page 369]

270. P. Talou, I. Stetcu, P. Jaffke, M. E. Rising, A. E. Lovell, and T. Kawano, "Fission Fragment Decay Simulations with the CGMF Code," *Computer Physics Communications*, vol. 269, p. 108087, 2021. DOI: 10.1016/j.cpc.2021.108087 [Page 369]

271. J. S. Hendricks, "New Spontaneous Fission Data," Los Alamos National Laboratory, Los Alamos, NM, USA, Memorandum D-5:JSH-2005-064, Dec. 2004. [Page 370]

272. P. A. Santi, D. H. Beddingfield, and D. R. Mayo, "Revised Prompt Neutron Emission Multiplicity Distributions for ^{236,238}Pu," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-04-8040, Nov. 2004. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-04-8040 [Pages 370, 371, and 403]

273. N. E. Holden and M. S. Zucker, "IAEA Advisory Group Meeting on Neutron Standard Reference Data, Geel, Belgium, November 12–16," Brookhaven National Laboratory, Brookhaven, NY, USA, Tech. Rep. BNL-NCS-35513-R, Nov. 1984. [Page 370]

274. M. S. Zucker and N. E. Holden, "Californium-252 and ^{238}U Nuclear Parameters of Safeguards Interest," Brookhaven National Laboratory, Brookhaven, NY, USA, Tech. Rep. BNL-34804, 1983. [Page 370]

275. N. E. Holden and M. S. Zucker, "Prompt Neutron Multiplicity for the Transplutonium Nuclides," in *Proceedings of the International Conference on Nuclear Data for Basic and Applied Science*, Santa Fe, NM, USA, May 1985, Brookhaven National Laboratory Tech. Rep. BNL-NCS-36379. URL: https://www.osti.gov/biblio/5559210 [Page 370]

276. D. A. Hicks, J. Ise, and R. V. Pyle, "Probabilities of Prompt-Neutron Emission from Spontaneous Fission," *Physical Review*, vol. 101, no. 3, pp. 1016–1020, Feb. 1956. DOI: 10.1103/PhysRev.101.1016 [Page 370]

277. W. W. T. Crane, G. H. Higgins, and H. R. Bowman, "Average Number of Neutrons per Fission for Several Heavy-Element Nuclides," *Physical Review*, vol. 101, no. 6, pp. 1804–1805, Mar. 1956. DOI: 10.1103/PhysRev.101.1804 [Page 370]

278. J. W. Boldeman and M. G. Hines, "Prompt Neutron Emission Probabilities Following Spontaneous and Thermal Neutron Fission," *Nuclear Science and Engineering*, vol. 91, no. 1, pp. 114–116, Sep. 1985. DOI: 10.13182/NSE85-A17133 [Page 370]

279. B. C. Diven, H. C. Martin, R. F. Taschek, and J. Terrell, "Multiplicities of Fission Neutrons," *Physical Review*, vol. 101, no. 3, pp. 1012–1015, Feb. 1956. DOI: 10.1103/PhysRev.101.1012 [Page 370]

280. J. S. Bull, H. G. Hughes, III, P. L. Walstrom, J. D. Zumbro, and N. V. Mokhov, "Magnetic Field Tracking with MCNP5," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-04-2125, Mar. 2004. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-04-2125 [Page 379]

281. J. S. Bull, "Magnetic Field Tracking Features in MCNP6," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-11-00872, Feb. 2011. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-11-00872 [Page 379]

282. M. Berz and K. Makino, "COSY INFINITY Version 8.1—User's Guide and Reference Manual," Department of Physics and Astronomy, Michigan State University, East Lansing, MI, USA, Tech. Rep. MSUHEP-20704, 2002, See also http://cosy.pa.msu.edu. [Page 379]

283. N. V. Mokhov, "The MARS Code System User's Guide," Fermi National Accelerator Laboratory, Batavia, IL, USA, Tech. Rep. FERMILAB-FN-628, Apr. 1995. [Page 379]

284. N. V. Mokhov and O. E. Krivosheev, "MARS Code Status," in *Proceedings of International Conference on Advanced Monte Carlo for Radiation Physics, Particle Transport Simulation and Applications (M&C 2000)*, Lisbon, Portugal, Oct. 2000. [Page 379]

285. N. V. Mokhov, "Status of MARS Code," Fermi National Accelerator Laboratory, Batavia, IL, USA, Tech. Rep. FERMILAB-Conf-03/053, Mar. 2003. [Page 379]

286. J. A. Favorite and K. J. Adams, "Tracking Charged Particles Through a Magnetic Field Using MCNPX (U)," Los Alamos National Laboratory, Los Alamos, NM, USA, Research Note XCI-RN(U)99-002, Feb. 1999. [Page 380]

287. E. Forest, "Lie Algebraic Methods for Charged Particle Beams and Light Options," Ph.D. dissertation, University of Maryland, College Park, MD, USA, 1984. [Page 382]

288. W. C. Feldman, D. M. Drake, R. D. O'Dell, F. W. Brinkley, Jr., and R. C. Anderson, "Gravitational Effects on Planetary Neutron Flux Spectra," *Journal of Geophysical Research: Solid Earth*, vol. 94, no. B1, pp. 513–525, 1989. DOI: 10.1029/JB094iB01p00513 [Page 385]

289. G. Castagnoli and D. Lal, "Solar Modulation Effects in Terrestrial Production of Carbon-14," *Radiocarbon*, vol. 22, no. 2, pp. 133–158, 1980. DOI: 10.1017/S0033822200009413 [Pages 388 and 392]

290. J. Masarik and R. C. Reedy, "Gamma Ray Production and Transport in Mars," *Journal of Geophysical Research: Planets*, vol. 101, no. E8, pp. 18 891–18 912, Aug. 1996. DOI: 10.1029/96JE01563 [Page 388]

291. T. E. Booth, "The Combination of Rejection Sampling with Biased Probability Density Sampling (with Special Comments on MCNP "Cookie Cutter" Rejection)," Los Alamos National Laboratory, Tech. Rep. LA-UR-20-26688, 2020. DOI: 10.2172/1657100 [Page 391]

292. G. E. McMath and G. W. McKinney, "MCNP6 Elevation Scaling of Cosmic Ray Backgrounds," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-14-21331, Jul. 2014. URL: https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-14-21331 [Page 392]

293. M. Clem, G. de Angelis, P. Goldhagen, and J. W. Wilson, "New Calculations of the Atmospheric Cosmic Radiation Field—Results for Neutron Spectra," *Radiation Protection Dosimetry*, vol. 110, no. 1–4, pp. 423–428, 2004. DOI: 10.1093/rpd/nch175 [Page 392]

294. F. B. Brown, "Investigation of Clustering in MCNP6 Monte Carlo Criticality Calculations," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-17-25009, Jun. 2017. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-17-25009 [Page 428]

295. B. C. Kiedrowski and F. B. Brown, "Difficulties Computing k in Non-Uniform, Multi-Region Systems with Loose, Asymmetric Coupling," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-11-04541, Sep. 2011. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-11-04541 [Page 428]

296. F. B. Brown, "Monte Carlo Eigenvalue Calculations," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-06-7094, Oct. 2006. [Page 428]

297. F. B. Brown, "Revisiting the "K-Effective of the World" Problem," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-10-00189, Jun. 2010. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-10-00189 [Page 428]

298. F. B. Brown, ""K-effective of the World" and Other Concerns for Monte Carlo Eigenvalue Calculations," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-10-00289, Oct. 2010. URL: https://mcnp.lanl.gov/pdf_files/la-ur-10-05548.pdf [Page 428]

299. F. B. Brown, "A Review of Monte Carlo Criticality Calculations - Convergence, Bias, Statistics," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-08-06558, May 2008. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-08-06558 [Page 428]

300. F. B. Brown, "A Review of Best Practices for Monte Carlo Criticality Calculations," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-09-03136, 2009. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-09-03136 [Page 428]

301. F. B. Brown, C. J. Josey, S. Henderson, and W. R. Martin, "Automated Acceleration and Convergence Testing for Monte Carlo Criticality Calculations," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-19-23887, Aug. 2019. URL: https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-19-23887 [Page 430]

302. M. L. Fensin, "Development of the MCNPX Depletion Capability: a Monte Carlo Linked Depletion Method That Automates the Coupling between MCNPX and CINDER90 for High Fidelity Burnup Calculations," Ph.D. dissertation, University of Florida, Gainsville, FL, USA, 2008. URL: https://ufdc.ufl.edu/UFE0021946/00001 [Page 441]

303. J. A. Favorite, A. D. Thomas, and T. E. Booth, "On the Accuracy of a Common Monte Carlo Surface Flux Grazing Approximation," *Nuclear Science and Engineering*, vol. 168, no. 2, pp. 115–127, Jun. 2011, Los Alamos National Laboratory Tech. Rep. LA-UR-09-06258. DOI: 10.13182/NSE09-72 [Page 448]

304. J. A. Favorite, "Monte Carlo Surface Flux Tallies," in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)*. Rio de Janeiro, Brazil: American Nuclear Society, May 2011. [Page 448]

305. E. C. Snow, "Radiography Image Detector Patch for MCNP," Private Communication, 1996. [Page 451]

306. E. C. Snow, "Mesh Tallies and Radiography Images for MCNPX," in *Proceedings of 4th Workshop on Simulating Accelerator Radiation Environments (SARE4)*, T. A. Gabriel, Ed., Knoxville, TN, USA, Sep. 1998, p. 113. [Page 451]

307. C. J. Solomon, Jr. and C. R. Bates, "The MCNPTools Package: Installation and Use," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-17-21779, Mar. 2017, MCNPTools Version 3.8. URL: http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-17-21779 [Pages 455 and 910]

308. T. E. Booth, "A Supertrack Importance Generator for Pulse Height Tallies," in *Transactions of the American Nuclear Society*, vol. 71, Aug. 1994, pp. 400–407. [Page 457]

309. M. Herman and A. Trkov, "ENDF-6 Formats Manual," Brookhaven National Laboratory, Upton, NY, USA, Tech. Rep. CSEWG Document ENDF-102 BNL-90365-2009, Jun. 2009. URL: https://www.bnl.gov/isd/documents/70393.pdf [Page 472]

310. M. Herman and A. Trkov, "ENDF-6 Formats Manual," Brookhaven National Laboratory, Upton, NY, USA, Tech. Rep. CSEWG Document ENDF-102 Report BNL-90365-2009 Rev.1, Jul. 2010. URL: https://www.oecd-nea.org/dbdata/data/manual-endf/endf102.pdf [Page 472]

311. ICRP, "1990 Recommendations of the International Commission on Radiological Protection," *Annals of the ICRP*, vol. 21, no. 1–3, Jan. 1991, ICRP Publication 60. DOI: 10.1016/0146-6453(91)90065-O [Page 478]

312. J. E. Sweezy, T. E. Booth, F. B. Brown, J. S. Bull, R. A. Forster, III, J. T. Goorley, H. G. Hughes, III, R. L. Martz, R. E. Prael, A. Sood, A. J. Zukaitis, R. C. Little, H. R. Trellue, M. C. White, M. B. Lee, and S. M. Girard, "MCNP - A General Monte Carlo N-Particle Transport Code, Version 5 - Volume 3: Developer's Guide," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-CP-03-0284 (Revised 2/1/2008), Apr. 2003, This document is provided in the MCNP6 release package available from RSICC and is not accessible from the MCNP website. [Pages 486 and 917]

313. V. I. Tretyak, "Semi-empirical Calculation of Quenching Factors for Ions in Scintillators," *Astroparticle Physics*, vol. 33, no. 1, pp. 40–53, Feb. 2010. DOI: 10.1016/j.astropartphys.2009.11.002 [Page 492]

314. V. Avdeichikov, B. Jakobsson, V. A. Nikitin, P. V. Nomokonov, and A. Wegner, "Systematics in the Light Response of BGO, CsI(Tl) and GSO(Ce) Scintillators to Charged Particles," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 484, no. 1, pp. 251–258, May 2002. DOI: 10.1016/S0168-9002(01)01963-5 [Page 492]

315. A. Menchaca-Rocha, M. Buénerd, L. Gallin-Martel, F. Ohlsson-Malek, and T. Thuillier, "Response Measurements of NE102 to $Z = 2$–6 Ions at $E/A = 0.15$–1.5 GeV, and Predictions for Higher Z's," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 438, no. 2, pp. 322–332, Dec. 1999. DOI: 10.1016/S0168-9002(99)00837-2 [Page 492]

316. J. B. Birks, *The Theory and Practice of Scintillation Counting*. New York, NY, USA: MacMillan, 1964. DOI: 10.1016/C2013-0-01791-4 [Page 492]

317. M. T. Swinhoe, J. S. Hendricks, and D. R. Mayo, "MCNPX for Neutron Multiplicity Detector Simulation," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-04-8025, Nov. 2004. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-04-8025 [Pages 495 and 797]

318. B. T. Rearden, "TSUNAMI-3D: Control Module for Three-Dimensional Cross-Section Sensitivity and Uncertainty Analysis for Criticality," Oak Ridge National Laboratory, Oak Ridge, TN, USA, Tech. Rep. ORNL/TM-2005/39, Ver. 6, 2009. [Pages 512 and 521]

319. J. A. Favorite, "Using the MCNP Taylor Series Perturbation Feature (Efficiently) for Shielding Problems," in *Proceedings of the 13th International Conference on Radiation Shielding (ICRS-13) & 19th Topical Meeting of the Radiation Protection and Shielding Division of the American Nuclear Society (RPSD-2016)*. Paris, France: American Nuclear Society, Oct. 2016. [Pages 513, 514, and 516]

320. B. C. Kiedrowski and F. B. Brown, "Continuous-Energy Sensitivity Coefficient Capability in MCNP6," in *Proceedings of 2012 ANS Winter Meeting*. San Diego, CA, USA: American Nuclear Society, Dec. 2012, Los Alamos National Laboratory Tech. Rep. LA-UR-12-21010. [Page 521]

321. B. C. Kiedrowski, "MCNP6.1 k-eigenvalue Sensitivity Capability: A User's Guide," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-13-22251, Mar. 2013. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-13-22251 [Page 521]

322. J. A. Favorite, Z. Perkó, B. C. Kiedrowski, and C. M. Perfetti, "Adjoint-based Sensitivity and Uncertainty Analysis for Density and Composition: A User's Guide," *Nuclear Science and Engineering*, vol. 185, no. 3, pp. 384–405, Mar. 2017. DOI: 10.1080/00295639.2016.1272990 [Page 522]

323. "SCALE Code System," Oak Ridge National Laboratory, Oak Ridge, TN, USA, Tech. Rep. ORNL/TM-2005/39, Version 6.2.3, Mar. 2018, available from Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-834. DOI: 10.2172/1426571 [Pages 525 and 526]

324. J. A. Clarke and E. R. Mark, "Enhancements to the eXtensible Data Model and Format (XDMF)," in *HPCMP User's Group Conference 2007. High Performance Computing Modernization Program: A Bridge to Future Defense*, Pittsburgh, PA, USA; June 18–21, 2007, pp. 322–327. [Pages 535, 740, and 921]

325. "XDMF Model and Format," Website, Apr. 2020. URL: http://xdmf.org/index.php/XDMF_Model_and_Format [Pages 535, 740, 921, 929, and 938]

326. U. Ayachit, *The ParaView Guide*, community ed., L. Avila, K. Osterdahl, S. McKenzie, and S. Jordan, Eds. Kitware Inc., Jun. 2018. [Pages 535, 538, 701, 740, and 921]

327. "VisIt User's Manual," Lawrence Livermore National Laboratory, Livermore, CA, USA, Tech. Rep. UCRL-SM-220449, Oct. 2005. URL: https://wci.llnl.gov/simulation/computer-codes/visit/manuals [Pages 535, 740, and 921]

328. P. Pébay, T. B. Terriberry, H. Kolla, and J. Bennett, "Numerically Stable, Scalable Formulas for Parallel and Online Computation of Higher-order Multivariate Central Moments with Arbitrary Weights," *Computational Statistics*, vol. 31, no. 4, pp. 1305–1325, Mar. 2016. DOI: 10.1007/s00180-015-0637-z [Page 542]

329. J. S. Bull, J. A. Kulesza, C. J. Josey, and M. E. Rising, "MCNP® Code Version 6.3.0 Build Guide," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-22-32851, Rev. 1, Dec. 2022. DOI: 10.2172/1906011 [Page 543]

330. J. T. Goorley, "MCNP5 Tally Enhancement for Lattices," Los Alamos National Laboratory, Los Alamos, NM, USA, Memorandum X-5-RN(U)04-20/LA-UR-04-3400, Jun. 2004. [Pages 543 and 544]

331. T. E. Booth, K. C. Kelley, and S. S. McCready, "Monte Carlo Variance Reduction using Nested DXTRAN Spheres," *Nuclear Technology*, vol. 168, no. 3, pp. 765–767, Dec. 2009, Los Alamos National Laboratory Tech. Rep. LA-UR-08-02074. DOI: 10.13182/NT09-A9303 [Page 568]

332. D. A. Dixon, "An Update to the Computation of the Goudsmit-Saunderson Distribution in MCNP® Version 6.2," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-16-27959, Oct. 2016. DOI: 10.2172/1330070 [Page 613]

333. J. W. Durkee, Jr., M. R. James, and L. S. Waters, "MCNPX Graphics and Arithmetic Tally Upgrades," in *International Conference on Advances in Mathematics, Computational Methods, and Reactor Physics (M&C 2009)*, Saratoga Springs, NY, USA, May 2009, Los Alamos National Laboratory Tech. Rep. LA-UR-08-6562. [Page 639]

334. V. F. Dean, C. A. Atkinson, and N. R. Smith, "Water Moderated U(2.35)O_2 Fuel Rods in 2.032-cm Square-Pitched Arrays," in *International Handbook of Evaluated Criticality Safety Experiments*. Nuclear Energy Agency, Organization for Economic Co-Operation and Development, Sep. 2004, no. LEU-COMP-THERM-001. [Page 691]

335. R. J. LaBauve, J. Sapir, and C. A. Atkinson, "Bare, Highly Enriched Uranium Sphere (GODIVA)," in *International Handbook of Evaluated Criticality Safety Experiments*. Nuclear Energy Agency, Organization for Economic Co-Operation and Development, Sep. 2004, no. HEU-MET-FAST-001. [Page 693]

336. T. J. Tautges, P. P. Wilson, J. A. Kraftcheck, B. M. Smith, and D. L. Henderson, "Acceleration Techniques For Direct Use Of CAD-Based Geometries In Monte Carlo Radiation Transport," in *Proceedings of the International Conference On Mathematics, Computational Methods & Reactor Physics*. Saratoga Springs, NY, USA: American Nuclear Society, May 2009. [Page 722]

337. Y. Wu, Q. Zeng, and L. Lu, "CAD-Based Modeling For 3D Particle Transport," in *Proceedings of the International Conference On Mathematics, Computational Methods & Reactor Physics*. Saratoga Springs, NY, USA: American Nuclear Society, May 2009. [Page 722]

338. D. Große and H. Tsige-Tamirat, "Current Status of the CAD Interface Program McCad for MC Particle Transport Calculations," in *Proceedings of the International Conference On Mathematics, Computational Methods & Reactor Physics*. Saratoga Springs, NY, USA: American Nuclear Society, May 2009. [Page 722]

339. *Abaqus User Manuals, Version 6.9*, Dassault Systèmes Simulia, Inc., Providence, RI, USA, 2009. [Page 722]

340. The HDF Group, "Hierarchical Data Format, Version 5," Website, Apr. 2020. URL: http://www.hdfgroup.org/HDF5/ [Pages 739 and 936]

341. R. L. Martz, "The MCNPUM Capability for MCNP6's Unstructured Mesh Feature," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-16-23286, May 2016. URL: http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-16-23286 [Page 741]

342. K. L. Currie and M. E. Rising, "MCNP6 Source Primer: Release 1.0," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-18-21377, Rev. 1, Feb. 2020. DOI: 10.2172/1599011 [Page 743]

343. J. E. Sweezy, T. E. Booth, F. B. Brown, J. S. Bull, R. A. Forster, III, J. T. Goorley, H. G. Hughes, III, R. L. Martz, R. E. Prael, A. Sood, A. J. Zukaitis, R. C. Little, H. R. Trellue, M. C. White, M. B. Lee, and S. M. Girard, "MCNP - A General Monte Carlo N-Particle Transport Code, Version 5 - Volume 2: User's Guide," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-CP-03-0245 (Revised 2/1/2008), Apr. 2003, This document is provided in the MCNP6 release package available from RSICC and is not accessible from the MCNP website. [Page 788]

344. F. B. Brown, "The `makxsf` Code with Doppler Broadening," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-06-7002, Sep. 2006. URL: https://mcnp.lanl.gov/pdf_files/la-ur-06-7002.pdf [Pages 859 and 977]

345. J. L. Conlin and P. Romano, "A Compact ENDF (ACE) Format Specification," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-19-29016, Sep. 2019. DOI: 10.2172/1561065 [Page 889]

346. S. R. Bolding, J. A. Kulesza, M. J. Marcath, and M. E. Rising, "Particle Track Output (PTRAC) Improvements, Parallelism, and Post-Processing," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-21-26562, Jul. 2021. URL: https://www.osti.gov/biblio/1813812-particle-track-output-ptrac-improvements-parallelism-post-processing [Page 910]

347. G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA, USA: CreateSpace, 2009. [Page 934]

348. P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, Mar. 2020. DOI: 10.1038/s41592-019-0686-2 [Page 934]

349. C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2 [Page 934]

350. F. B. Brown, S. E. Carney, B. C. Kiedrowski, and W. R. Martin, "Fission Matrix Capability for MCNP, Part I - Theory," in *Proceedings of the International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2013)*. Sun Valley, ID, USA, May 5–9: American Nuclear Society, May 2013, Los Alamos National Laboratory Tech. Rep. LA-UR-13-20429. URL: https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-13-20429 [Page 935]

351. S. E. Carney, F. B. Brown, B. C. Kiedrowski, and W. R. Martin, "Fission Matrix Capability for MCNP, Part II - Applications," in *Proceedings of the International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2013)*. Sun Valley, ID, USA, May 5–9: American Nuclear Society, May 2013, Los Alamos National Laboratory Tech. Rep. LA-UR-13-20454. URL: https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-13-20454 [Page 935]

352. R. L. Martz, "ELA: Event Log Analyzer for MCNP5 (U)," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-06-7796, Nov. 2006. URL: https://mcnp.lanl.gov/pdf_files/la-ur-06-7796.pdf [Page 964]

353. R. L. Martz, "ELA: Event Log Analyzer," in *Proceedings of the 2006 Winter American Nuclear Society Meeting*, vol. 95, no. 1. Albuquerque, NM, USA, November 12–16: American Nuclear Society, Nov. 2006, p. 677, Los Alamos National Laboratory Tech. Rep. LA-UR-06-3910. URL: https://www.ans.org/pubs/transactions/article-6386 [Page 964]

354. R. E. MacFarlane, D. W. Muir, R. M. Boicourt, A. C. Kahler, III, and J. L. Conlin, "The NJOY Nuclear Data Processing System, Version 2016," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-17-20093, Jan. 2017. DOI: 10.2172/1338791 [Page 977]

355. F. B. Brown, "A Tutorial on Merging Tallies from Separate MCNP5 Runs," Los Alamos National Laboratory, Los Alamos, NM, USA, Presentation LA-UR-08-00249, Jan. 2008. URL: http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-08-00249 [Pages 978 and 980]

356. F. B. Brown, J. E. Sweezy, and R. B. Hayes, "Monte Carlo Parameter Studies and Uncertainty Analyses with MCNP5," in *Proceedings of PHYSOR 2004—The Physics of Fuel Cycles and Advanced Nuclear Systems: Global Developments*, Chicago, IL, USA. April 25–29, 2004, Los Alamos National Laboratory Tech. Rep. LA-UR-04-0499 (Paper) and Los Alamos National Laboratory Tech. Rep. LA-UR-04-2506 (Presentation). URL: https://www.osti.gov/biblio/977428-monte-carlo-parameter-studies-uncertainty-analyses-mcnp5 [Pages 983 and 990]

357. F. B. Brown, "New Tools to Prepare ACE Cross-section Files for MCNP Analytic Test Problems," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-16-24290, 2016. DOI: 10.2172/1258350 [Page 992]

358. F. B. Brown and N. Barnett, "One-group MCNP5 Criticality Calculations with Anisotropic Scattering," in *Transactions of the American Nuclear Society*, vol. 98. Anaheim, CA, USA. June 8–12: American Nuclear Society, Jun. 2008, pp. 518–519, Los Alamos National Laboratory Tech. Rep. LA-UR-08-00567. URL: https://www.ans.org/pubs/transactions/article-8112 [Pages 992, 993, and 995]

359. J. C. Armstrong and K. C. Kelley, "Generating MCNP Input Files for Unstructured Mesh Geometries," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-20-27139, Sep. 2020. DOI: 10.2172/1660578 [Page 1012]

360. ANS-6.1.1 Working Group, "American National Standard Neutron and Gamma-ray Flux-to-dose Rate Factors," American Nuclear Society, LaGrange Park, IL, USA, Tech. Rep. ANSI/ANS-6.1.1-1977, 1977. [Pages 1022 and 1068]

361. ANS-6.1.1 Working Group, "American National Standard Neutron and Gamma-ray Fluence-to-dose Rate Factors," American Nuclear Society, LaGrange Park, IL, USA, Tech. Rep. ANSI/ANS-6.1.1-1991, 1991. [Pages 1022 and 1068]

362. ICRP, "Data for Protection against Ionizing Radiation from External Sources: Supplement to ICRP Publication 15," *Annals of the ICRP*, vol. 21, no. 1, Jan. 1973, ICRP Publication 21. DOI: 10.1016/S0074-27407380002-9 [Pages 1022 and 1068]

363. ICRP, "Conversion Coefficients for use in Radiological Protection against External Radiation," *Annals of the ICRP*, vol. 26, no. 3–4, Jul. 1996, ICRP Publication 74. URL: https://journals.sagepub.com/toc/anib/26/3-4 [Page 1022]

364. ICRP, "Conversion Coefficients for Radiological Protection Quantities for External Radiation Exposures," *Annals of the ICRP*, vol. 40, no. 2–5, Apr. 2010, ICRP Publication 116. DOI: 10.1016/j.icrp.2011.10.001 [Pages 1022 and 1068]

365. ASTM E722-19, "Standard Practice for Characterizing Neutron Fluence Spectra in Terms of an Equivalent Monoenergetic Neutron Fluence for Radiation-Hardness Testing of Electronics," ASTM International, West Conshohocken, PA, USA, 2019. DOI: 10.1520/E0722-19 [Page 1095]

366. K. R. DePriest, "Historical Examination of the ASTM Standard E722 1-MeV Silicon Equivalent Fluence Metric," Sandia National Laboratory, Albuquerque, NM, USA, Tech. Rep. SAND2019-15194, Dec. 2019. DOI: 10.2172/1592863 [Page 1095]

Contributors

Casey A. Anderson prefers to keep an air of mystery about themselves.

Jerawan C. Armstrong, Ph.D., began working at Los Alamos National Laboratory in 2009. She has worked on various projects related to particle transport methods and applications.

Simon R. Bolding prefers to keep an air of mystery about themselves.

Thomas E. Booth, retired (LANL 1974–2011), developed novel Monte Carlo variance reduction techniques for Boltzmann transport and non-Boltzmann transport. Other Monte Carlo work: reliability assessment, adaptive Monte Carlo, higher eigenfunctions, probability of extinction, statistical physics, and elliptic PDEs.

Forrest B. Brown prefers to keep an air of mystery about themselves.

Jeffrey S. Bull prefers to keep an air of mystery about themselves.

Laura Casswell prefers to keep an air of mystery about themselves.

Alexander R. Clark, Ph.D., began working at LANL in 2016 as a Graduate Student Assistant, later as a Postdoctoral Research Associate, and is now an R&D Scientist. His technical interests include sensitivity analysis and uncertainty quantification applied to measures of subcriticality. He is a developer for the MCNP code, the Whisper package, and supports V&V.

Lawrence (Larry) J. Cox, Ph.D., is a retired computational physicist from Los Alamos and Livermore National Laboratories. In addition to getting a Ph.D. in Applied Science from UC/Davis, Larry is qualified as a Professional Software Engineering Master through the IEEE. In retirement, Larry continues to work on novel methods for parallel Monte Carlo radiation transport.

David Dixon, Ph.D, has worked on a variety of transport codes projects at Los Alamos National Laboratory but specializes in electron transport methods.

Joe W. Durkee prefers to keep an air of mystery about themselves.

Jay S. Elson retired from LANL in 2023 after 40 years of service to the laboratory. He served in various roles ranging from student, staff, and management. His professional interests include multiphase flow, computational fluid dynamics, atmospheric dispersion, and radiation transport.

Jeffrey A. Favorite, Ph.D., began working at Los Alamos National Laboratory in 1998. He is interested in deterministic and Monte Carlo radiation transport, particularly emphasizing sensitivity and perturbation methods.

Michael L. Fensin, Ph.D., is currently a research scientist at LANL. His research focuses on variance reduction strategies for small solid angle transport, radiation shielding design, radiation effects in materials, reactor burnup and design, and methods of radiographic interpretation of complex experiments.

R. Arthur Forster, Ph.D., is a retired LANL Fellow who has worked continuously on MCNP code development and applications since 1974. Two research areas of focus are the statistical aspects of Monte Carlo solutions to provide assurance that tally confidence intervals are valid and adaptive Monte Carlo solutions.

Jesse F. Giron, Ph.D., began working at Los Alamos National Laboratory in 2013 as an undergraduate and is currently a research scientist at LANL. His research focuses on mathematical methods, charged particle transport, and high energy particle physics.

John T. (Tim) Goorley, Ph.D., now leads the XCP-7 Radiation Transport Applications team, which heavily relies on MCNP6. He has used the MCNP code since his nuclear engineering undergraduate and counts himself fortunate that he can still use the code to do useful and interesting analyses for a wide variety of applications. He was on the MCNP Development team during 2002–2013, led the MCNP5/X merger, and was the MCNP6 initial release manager.

Avery S. Grieve joined LANL in 2021 and held the role of MCNP User Support Specialist until they departed LANL in 2023 to work with Commonwealth Fusion Systems.

John S. Hendricks, Ph.D. has worked in many areas including MCNP tallies, variance reduction, sources, plotters, and more. From 1989–1999 he was the principal integrator for MCNP features, from 2000–2006 he focused on MCNPX, and from 2006–2009 he focused on the merger to create MCNP6. From 2009 to the present, he has provided consulting services within a variety of industries.

Henry G. (Grady) Hughes prefers to keep an air of mystery about themselves.

Michael R. James prefers to keep an air of mystery about themselves.

Russell C. Johns prefers to keep an air of mystery about themselves.

Brian Kiedrowski, Ph.D., is a former LANL research scientist and currently a faculty member in the Nuclear Engineering & Radiological Sciences Department at the University of Michigan. His research focuses on theory and computational methods of neutral particle transport related to nuclear criticality, nonproliferation, reactor physics, and hybrid variance reduction.

Joel A. Kulesza, Ph.D., P.E., has worked at Compuware Corporation, Knolls Atomic Power Laboratory, Westinghouse Electric Company, and Los Alamos National Laboratory. His primary technical interests are hybrid variance-reduction techniques, unstructured-mesh analyses, and scientific visualization.

Roger L. Martz is a former LANL R&D scientist and has worked at the Bettis Atomic Power Laboratory serving on the SEAWOLF submarine and FORD carrier shield design teams. He is currently President & CEO of Stag Scientific Software, LLC, providing nuclear engineering, programming, and visualization services.

Stepan G. Mashnik prefers to keep an air of mystery about themselves.

Gregg W. McKinney, Ph.D., developed his first MCNP feature (`PERT` card) in 1983, while attending the University of Washington. He has developed more than 100 MCNP features and enhancements over the last four decades, including parallel processing, repeated-structure tallies, X-Window graphics, cosmic and terrestrial sources, delayed-particle emission, delta-ray production, and a long list of special tally options, to name a few.

Garrett E. McMath is a LANL R&D Engineer whose research interests include cosmic and terrestrial background radiation, design and testing of novel NDA instruments with international safeguards applications, and nuclear material accounting and control.

Eric J. Pearson prefers to keep an air of mystery about themselves.

Denise B. Pelowitz prefers to keep an air of mystery about themselves.

Richard E. Prael prefers to keep an air of mystery about themselves.

Michael E. Rising prefers to keep an air of mystery about themselves.

Clell J. (CJ) Solomon, Jr. prefers to keep an air of mystery about themselves.

Avneet Sood Ph.D., is currently a Senior Scientist at Los Alamos National Laboratory. His primary technical interests involve Monte Carlo transport methods and a variety of radiation transport applications supporting nuclear counter terrorism, non-proliferation, and counter-proliferation. His recent interest has been in nuclear policy analysis related to nuclear deterrence, arms control, national defense, and emerging threats.

Jeremy E. Sweezy, Ph.D., joined LANL in the Monte Carlo Codes group in 2002. He has been the team leader for the MCNP code development team (2005-2010) and the group leader of the XCP-3 Monte Carlo codes group (2019-2023). His primary technical interests are Monte Carlo software development and variance reduction methods.

Travis J. Trahan, Ph.D., is a LANL research scientist in the Monte Carlo Codes group and has been a project leader in the Advanced Simulation and Computing program. His primary technical interests are Monte Carlo software development, analysis of stochastic systems, and methods and algorithms for improving simulation efficiency.

Colin A. Weaver, Ph.D., is a Scientist at LANL.

Christopher J. Werner prefers to keep an air of mystery about themselves.

Trevor A. Wilcox prefers to keep an air of mystery about themselves.

Anthony Zukaitis prefers to keep an air of mystery about themselves.

Index

G

H

I

X

Z

Colophon

The look and style of this document is the result of the authors' experimentation, reader comments, and an attempt to capture "best practices" such as those described in *The Elements of Style* and *The Chicago Manual of Style* as the MCNP code's documentation has evolved over its lifetime. Additional feedback on this work can be provided to the editor via mcnp_help@lanl.gov.

This document, containing 208 figures and 82 tables, is assembled using LyX and typeset using LaTeX via PDFLaTeX with the Computer Modern type face. The figures are generally prepared using Python with matplotlib and/or TikZ. References are generally managed with JabRef, processed using BibTeX with natbib, and are shown using the `IEEEtranN` style (slightly modified).

This colophon and the accompanying software details are included to credit the indirect contributions to this work made by the community of developers and maintainers for these software products.