LA-UR-03- 3603

Approved for public release; distribution is unlimited.

99

Title: MCNP5 DEVELOPMENT, VERIFICATION, & PERFORMANCE

Author(s): FORREST B. BROWN

Submitted to: International Conference on Supercomputing in Nuclear Applications, Paris, France, September 22-24, 2003

Los Alamos NATIONAL LABORATORY

MCNP5 DEVELOPMENT, VERIFICATION, AND PERFORMANCE

Forrest B. Brown
Los Alamos National Laboratory

Abstract

MCNP is a well-known and widely used Monte Carlo code for neutron, photon, and electron transport simulations. During the past 18 months, MCNP was completely reworked to provide MCNP5, a modernized version with many new features, including plotting enhancements, photon Doppler broadening, radiography image tallies, enhancements to source definitions, improved variance reduction, improved random number generator, tallies on a superimposed mesh, and edits of criticality safety parameters. Significant improvements in software engineering and adherence to standards have been made. Over 100 verification problems have been used to ensure that MCNP5 produces the same results as before and that all capabilities have been preserved. Testing on large parallel systems shows excellent parallel scaling.

Introduction

MCNP [1] is a well-known and widely used Monte Carlo code for neutron, photon, and electron transport simulations. MCNP was first released in the mid-1970s for neutron and photon transport, and was enhanced over the years to include generalized sources and tallies, K-effective eigenvalue calculations, volume calculations, electron physics, coupled electron-photon calculations, interactive plotting of geometry and tallies, cross-section plotting, repeated structures, lattice geometry, parallel processing, perturbation theory, detectors and pulse height tallies, automated weight window generation, many variance reduction options, an unresolved resonance treatment, macrobody geometry, statistical convergence tests, and many other user-requested features. It is estimated that there are approximately 300 MCNP users at LANL and over 3,000 users worldwide.

The previous version of MCNP, Version 4C2, was released in 2001. As for many other large, mature, production-oriented code systems, the evolution of the coding itself has lagged the very rapid changes in computer hardware and software technologies. Coding style, Fortran language features, version control, issues tracking, compilation/installation procedures, etc., were based on 1970s practices. Rigid adherence to these practices served to maintain high code quality over the years, but has increasingly impeded further advanced development.

With the advent of the DOE Accelerated Strategic Computing Initiative (ASCI), there has been significant focus on upgrading the software development and quality assurance practices at DOE laboratories to ensure that codes can fully utilize the ASCI teraflop computers and can be enhanced to incorporate more advanced physics modeling. In support of the ASCI Program, an intensive effort to modernize MCNP has been carried out by the Monte Carlo Team at LANL for the past 18 months. The result of this effort is Version 5 of MCNP (MCNP5). The code is supported by a detailed 1,000-page manual, a series of application classes, and an extensive verification/validation effort.

Modernization of MCNP

The effort to modernize MCNP was driven by the need to provide for:

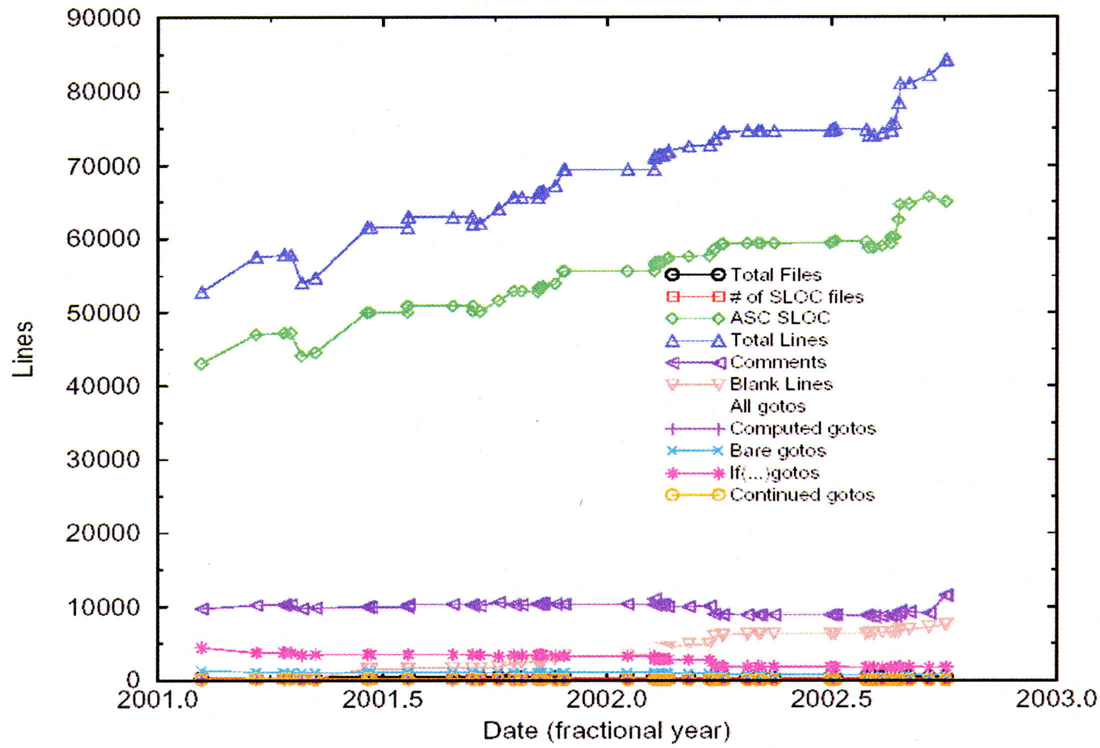
- Modern software engineering and software quality assurance practices,
- Strict adherence to current standards for Fortran-90 and parallel processing,
- Preservation of all existing code capabilities,
- Flexibility for rapid introduction of new features and adaption to advanced computers.

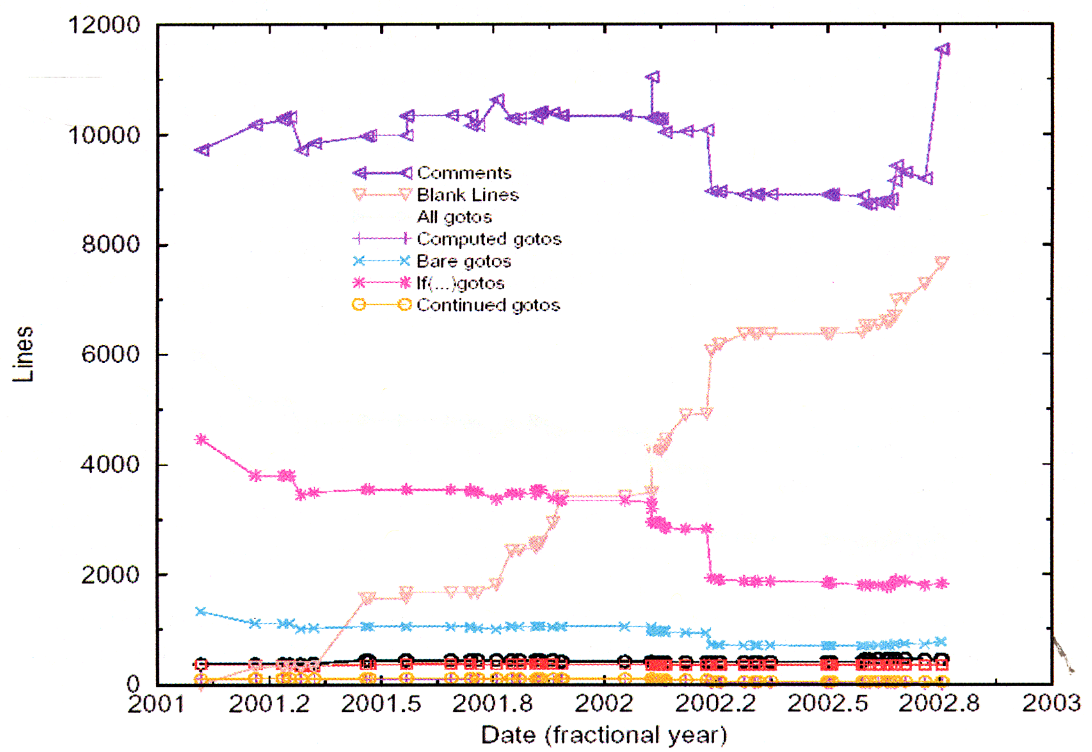
An evolutionary approach to MCNP modernization was followed to minimize the chances of the introduction of new errors.

Software Engineering and Quality Assurance

MCNP development adheres to the software engineering requirements established for the LANL ASCI program [2]. All source coding, test problems, documentation, Unix scripts for compilation, *makefiles*, etc., have been placed under strict, formal, version control. We currently use the Razor™ system to provide for centralized source code management and version control. Files are checked-out, modified, and then checked-in, with Razor managing versions of individual files and associating a set of

Figure 1. Evolution of Code Metrics During Development





files with each code release. Razor is also used for issues management, associating issues with each code release. The MCNP modernization has proceeded in a series of small steps. For each step, a new thread in the version control system was produced and thoroughly tested. To date, over 50 threads have been produced.

The entire installation and test process has been revised to use the GNU *make* utility [4]. Simple configuration files are used to specify system-dependent features (e.g., compiler, libraries, optimization controls, etc.). The *makefile* includes targets for building the executable, running a set of standard installation tests, and installing the code. The source coding has been split from one very large monolithic file into roughly 330 separate files containing individual subprograms or modules. The combination of these two changes allows for very fast incremental compilations when changes are made to the source coding. Code revisions can be distributed using the GNU utilities *diff* and *patch*, which are available for all systems, including PCs. This new build system has been used on a wide variety of computer systems, including Sun, SGI, HP-Compaq, IBM AIX, PC/Windows, PC/Linux, Itanium/Linux and Mac OS X. For PC/Windows, a Windows installer is also provided that installs a pre-compiled executable, the User Manual, the installation test set, and necessary scripts.

Standards for Fortran-90 and Parallel Processing

Every line of coding was reworked under formal version control to provide strict compliance with ANSI-standard Fortran-90. This conversion will aid in the long-term viability of MCNP, since Fortran-77 compilers are becoming obsolete. A consistent style was adopted to enhance readability and understanding of the coding. Modern Fortran-90 language features have been used to upgrade much of the coding, resulting in simplified code flow logic, a greatly reduced number of GOTO statements, encapsulation of selected features in modules, and much more readable, understandable coding. All source coding was changed to free-format Fortran-90; "comdecks" were converted to Fortran-90 modules; dynamic arrays are now constructed using Fortran-90 *allocate* statements; computed GOTOs were replaced by *case* statements; very many GOTOs were eliminated through the use of structured *if-else-endif* constructs; DO-loops were changed to eliminate shared terminations, and invoke cycle and exit statements; Hollerith usage was eliminated. To improve code readability, a consistent indentation style was applied to DO-loops, *if* statements, and *case* statements; very many blank lines were inserted; inline comments were used where appropriate. To encapsulate and consolidate related routines and data, Fortran-90 modules were created for random number generation, OpenMP parallelism, message-passing, criticality, geometry plotting, and tally plotting. A large portion of this conversion was accomplished using specially developed tools (e.g., *perl* scripts) to ensure consistency and error-free conversion. For manual changes, the use of the Razor version control and frequent testing served to (nearly) eliminate errors, and to quickly identify and resolve the inevitable errors that did occur.

The evolution of the code is illustrated in Figure 1, where various code metrics are plotted over the development period. These metrics are useful for measuring progress, as well as for characterizing the overall code contenbt. For example, total source lines of coding (SLOC) increased from about 50,000 to 90,000; the number of IF(…) GOTO statements was reduced by about 50%; the final code has about 12,000 comments and 8,000 blank lines; computed GOTO statements were completely eliminated.

MCNP5 is required to execute on many different varieties of parallel computer systems. To achieve parallelism in a portable manner, we rely on strict adherence to the MPI standard [3] for message-passing (for distributed-memory parallelism) and the OpenMP standard [4] for threading (for shared-memory parallelism). MCNP5 can execute sequentially (no parallelism) or in parallel using only MPI message-passing, using only OpenMP threads, or using both MPI and OpenMP. In addition, we continue to support PVM message-passing. For small calculations, we generally run MCNP5 sequentially or with a moderate number of OpenMP threads (typically 4 or 8). For larger calculations, MCNP5 has been run using 1000s of processors and combined MPI/OpenMP, with excellent parallel speedups.

Preservation of All Existing Code Capabilities

All previously existing code capabilities have been preserved, including physics options, geometry, tallying, plotting, cross-section handling, etc. Tally results from MCNP5 are expected to match the tally results of problems that can be run with the previous MCNP4C2, except where bugs were discovered and fixed in the conversion process. Changes in the format and presentation of some of the printed output are allowed, but the tally results (*mctal* files) are required to match 4C2 results in all installation/regression tests. All user input files that were used with previous versions should still work; no changes to input are required for using MCNP5 except to utilize new features.

Flexibility for New Features and Advanced Computers

One of the goals in MCNP modernization was providing flexibility for adding new features and adapting to advanced ASCI computer systems. This goal has been demonstrably met: Many new features have already been introduced into MCNP5, as described in the next section. Recently, the same version of MCNP5 that is used on PCs, Unix workstations, and the SGI Origin-2000 was installed and successfully tested on: a PC cluster using either MPI or PVM, the 3-Teraflop ASCI Blue Mountain system, the 12-Teraflop ASCI Blue Pacific system, and the 20-Teraflop ASCI Q system.

New Features

While a large amount of effort has been focused on modernization of MCNP, a number of new features have been developed for Version 5, including:

- **MCNP manual [5]**: The manual has been completely reviewed and modified, including a new command summary in Appendix A. The 1,000 page manual has been split into 3 volumes: Overview/Theory, Users Guide, and Developers Guide. The new features are included in our MCNP classes.

- **Criticality Safety Parameters**: Coding was added to MCNP5 to produce edits of quantities important to criticality safety analysts - energy corresponding to the average lethargy of neutrons causing fission, average energy of neutrons causing fission, fractions of fissions at low/medium/high energies, etc. These edits have been requested for years and will aid in comparing MCNP5 to other codes.

- **Doppler broadening for photon collisions**: This capability is important for low-energy photon transport. Past versions of MCNP have neglected the precollision motion of the electron. MCNP5 now includes the capability to handle this low-energy correction.

- **Radiography tallies**: Neutral particle radiography tallies have been added to support neutron and photon imaging simulations. This feature uses multiple point detectors to determine the particle flux at pixel locations in a user-defined grid. As many detector points as desired can be used to create both the direct (unscattered) and scattered flux image contributions. Each source and collision event contributes to all detectors, resulting in a smooth image. Radiography simulations have been run using millions of detector points.

- **Generalized source options**: Enhancements to sources provide source description options that are especially appropriate to accelerator beam applications, including options to select from Gaussian spatial distributions, transformation and replication of a defined source, and user selection of the type of source particle for surface sources.

- **Variance reduction**: Time-dependent importances can now be specified to allow varying splitting and Russian roulette parameters with time. The capability can be used for all particle types and is fully integrated with both implicit absorption and weight windows. This capability has been successfully tested on several problems where it is important to have good flux estimates at late problem times.

- **Random numbers**: In addition to modernizing the coding of the random number generator, the basic algorithm for random number generation has been extended [6]. This work has lengthened the period of the random number sequence by a factor of $\sim 10^5$, from $\sim 7\text{x}10^{13}$ to $\sim 9.2\text{x}10^{18}$. The increase in period is important, given today's faster computers, since it is becoming common to run problems involving $\sim 10^9$ histories, with a stride of $\sim 10^5$ random numbers allocated per history. Such calculations exceed the period of the previous random number generator, resulting in wraparound in the random sequence and reuse of some random numbers. For MCNP5, the default random number generator is identical to the previous one (to provide backward compatibility), but the extended-period generators may be optionally selected on the new RAND card. These new MCNP generators were selected after performing many random number tests.

- **Superimposed mesh-based tallies**: These tallies provide for tallying on a user-specified mesh. As for the superimposed mesh for weight windows, the tally mesh is independent of the problem geometry. More than one such mesh tally can be defined independently; their spatial extents may or may not overlap.

- **Setup and visualization**: We are providing support for the MCNP Visual Editor (VISED) [7], developed by R. Schwartz and L.L. Carter, which provides a convenient GUI-based means of preparing input and visualizing geometry. The PC version is included with the MCNP5 distribution. See URL http://www.mcnpvised.com/vised5.html for more VISED information.

- **Additional Macrobodies**: MCNP5 now provides the full set of macrobodies available to other codes such as MORSE, KENO, ITS. Additional macrobodies in MCNP5 that were not in MCNP4C2 include REC, TRC, ELL, WED, and ARB.

- **Plotting enhancements**: Improvements have been made to the geometry and tally plotting, with more colors and smooth gradients for 2D contour plot shading.

- **New MCNP Nuclear and Atomic Data Libraries:** In a separate but closely related effort, the Data Team at LANL has released new data libraries containing updates from the ENDF/B-VI.6, ACTI, and EPDL97 libraries [8]. The ENDF66 library, a follow-on to ENDF60, includes data for 173 nuclides from ENDF/B-VI release 6. There are 58 new nuclides, and another 40 have significant updates. All of the nuclides were processed with tighter tolerances and include recent new features, where appropriate, such as unresolved resonance probability tables, delayed neutron time and energy spectra, charged-particle production data, and tabular angular distributions. ENDF66 subsumes the many special purpose libraries (ENDF6DN, URES, and LA150N) released since ENDF60. The ACTI research effort sought to provide more detailed neutron-induced photon spectra for prompt gamma-ray spectroscopy. These updated photon spectra are now included in ENDF/B-VI as part of release 8, and the 41 nuclides updated have been processed into the special purpose ACTI library. The latest LLNL photon and electron data (EPDL97, EEDL97, and EADL97) have been included in ENDF/B-VI as part of release 8. The improved photoatomic interaction cross-sections and fluorescence data will be available in a new library. Future code updates will enable better sampling of the fluorescence data to more accurately reproduce atomic relaxation.

Verification

During this massive recoding effort, a fundamental requirement was that all previously-existing code capabilities must be preserved and no new code errors could be introduced. This requirement was inviolable, and was enforced by extensive testing throughout the entire code development effort. The MCNP5 developers have verified that MCNP5 produces the same results as the previous version, MCNP4C2, for a set of over 100 verification test problems. Four sets of verification problems were used to ensure code correctness: a suite of 42 regression tests, a suite of 26 criticality benchmark problems [9], a suite of 10 analytic benchmarks for criticality [10], and a suite of 19 radiation shielding validation problems [9].

Regression Test Suite

For many years, the MCNP distribution has included a set of installation tests to verify that installation and compilation of the code are carried out correctly on a given computer system. For these tests, reference "templates" are provided for both the printed code output and resulting tally files (*mctal* files), and are compared with the actual output and *mctal* files. Due to the short running time, the test set is typically run many times each day by an individual code developer and is now used for regression testing, rather than just installation testing. Today's code development process typically consists of modifying a few subroutines, incremental recompilation using GNU *make*, and then running the regression test set.

During the development of MCNP5, the regression test set was expanded from 28 to 42 problems, with new tests added to cover new code features or to explicitly test that particular bugs were fixed. Previous analysis of MCNP has indicated that the tests cover approximately 80-90% of the total lines of coding. (Test coverage analysis for MCNP5 is in progrss.) The MCNP5 build system specifically includes capabilities for running any or all of the regression tests and for comparing results with the reference templates.

It is important to note that the regression tests do not verify code correctness; they are used only for the purpose of detecting unintended changes to the code. Nevertheless, their extensive use on a daily basis serves to prevent the inadvertent introduction of bugs.

Criticality Validation Suite

The criticality validation suite [9] contains 26 cases that encompass a wide variety of fissile materials and spectra. Specifically, they include the three major fissile isotopes — ^{233}U, ^{235}U, and ^{239}Pu — in configurations that produce fast, intermediate, and thermal spectra. Furthermore, the ^{235}U cases were chosen so that they include highly enriched uranium (HEU), intermediate-enriched uranium (IEU), and low-enriched uranium (LEU) fuels.

The cases in the suite also were chosen to include a variety of configurations. The fast-spectrum cases include bare spheres, cores reflected by a heavy material (normal U), and cores reflected by a light material (Be or water). The thermal-spectrum cases include lattices of fuel pins as well as homogeneous solutions. The number of experiments with intermediate spectra is much more limited, and those cases were chosen primarily for availability rather than specific attributes.

The specifications for all 26 cases in the criticality validation suite are taken from the *International Handbook of Evaluated Criticality Benchmark Experiments* [11].The 26 cases are summarized in Table I. All of the cases are at room temperature and pressure.

The calculations all were performed in sequential (single-processor) mode on a Silicon Graphics Origin 2000 supercomputer at Los Alamos National Laboratory. Each of the cases employed 250 generations of 5,000 neutron histories each, and the results from the first 50 generations were discarded. Consequently, the results reported herein are based on 1,000,000 active neutrons histories for each case. For each case, calculations were run with both code versions using ENDF60+URES data [12] and also using the newer ENDF66 data [8].

The values of k_{eff} for these 26 cases are given in Table II. MCNP5 and MCNP4C2 produce identical answers for 49 of the 52 cases and agree within statistics for the other 3 cases. For the Zeus(2) cases, both code versions agree exactly using ENDF66 data. Using the ENDF60+URES data, the Zeus(2) cases tracked identically for 125 generations (0.625M histories), and final results agree within statistics.For the HEU-MT-003 (4) cases with the ENDF60+URES data, both codes agreed exactly. Using the ENDF66 data, the codes track for the first 225 generations (1.125M histories), and the final results agree within statistics. Similarly, the IEU-CT-002 (3) cases matched using ENDF60+URES data, and differed slightly using ENDF66 data, with final results agreeing within statistics.

The statistically insignificant differences observed in 3 of the 52 cases are attributed to roundoff associated with compiler differences. The MCNP4C2 code was compiled approximately 2 years previously using a Fortran-77 compiler and associated math libraries; the MCNP5 code was compiled

Table I. Summary of MCNP Criticality Validation Suite

	Name	Spectrum	Handbook ID	Description
1	Jezebel-233	Fast	U233-MET-FAST-001	Bare sphere of ^{233}U
2	Flattop-23	Fast	U233-MET-FAST-006	Sphere of ^{233}U reflected by normal U
3	U233-MF-005 (2)	Fast	U233-MET-FAST-005, case 2	Sphere of ^{233}U reflected by beryllium
4	Falstaff (1)	Intermediate	U233-SOL-INTER-001, case 1	Sphere of uranyl fluoride solution enriched in ^{233}U
5	ORNL-11	Thermal	U233-SOL-THERM-008	Large sphere of uranyl nitrate solution enriched in ^{233}U
6	Godiva	Fast	HEU-MET-FAST-001	Bare HEU sphere
7	Flattop-25	Fast	HEU-MET-FAST-028	HEU sphere reflected by normal U
8	Godiver	Fast	HEU-MET-FAST-004	HEU sphere reflected by water
9	HISS/HUG	Intermediate	HEU-COMP-INTER-004	Infinite, homogeneous mixture of HEU, H, and graphite
10	ZEUS (2)	Intermediate	HEU-MET-INTER-006, case2	HEU platters moderated by graphite and reflected by Cu
11	HEU-MT-003 (4)	Thermal	HEU-MET-THERM-003, case 4	Lattice of HEU cubes reflected by water
12	ORNL-10	Thermal	HEU-SOL-THERM-032	Large sphere of HEU nitrate solution
13	IEU-MF-003	Fast	IEU-MET-FAST-003	Bare sphere of IEU (36 wt.%)
14	BIG TEN	Fast	IEU-MET-FAST-007	Cylinder of IEU (10 wt.%) reflected by normal U
15	IEU-MF-004	Fast	IEU-MET-FAST-004	Sphere of IEU (36 wt.%) reflected by graphite
16	IEU-CT-002 (3)	Thermal	IEU-COMP-THERM-002, case 3	Lattice of IEU (17 wt.%) fuel rods in water
17	BAW XI (2)	Thermal	LEU-COMP-THERM-008, case 2	Large lattice of PWR fuel pins in borated water
18	SHEBA-2	Thermal	LEU-SOL-THERM-001	Cylinder of LEU fluoride solution enriched to 5 wt.%
19	Jezebel	Fast	PU-MET-FAST-001	Bare sphere of Pu
20	Jezebel-240	Fast	PU-MET-FAST-002	Bare sphere of Pu (20.1 at.% ^{240}Pu)
21	Flattop-Pu	Fast	PU-MET-FAST-006	Pu sphere reflected by normal U
22	PU-MF-011	Fast	PU-MET-FAST-011	Pu sphere reflected by water
23	Pu Buttons	Fast	PU-MET-FAST-003, case 3	3 x 3 x 3 array of small cylinders of Pu
24	HISS/HPG	Intermediate	PU-COMP-INTER-001	Infinite, homogeneous mixture of Pu, hydrogen, and graphite
25	PNL-33	Thermal	MIX-COMP-THERM-002, case 4	Lattice of mixed-oxide fuel pins in borated water
26	PNL-2	Thermal	PU-SOL-THERM-021, case 3	Sphere of plutonium nitrate solution

Table II. Criticality Suite Results, Using Old and New Data on SGI Computer

Name		K-effective Results Using ENDF60+URES Data		K-effective Results Using ENDF66 Data	
		MCNP5	MCNP4C2	MCNP5	MCNP4C2
1	Jezebel-233	0.99241 (57)	"	0.99106 (56)	"
2	Flattop-23	0.99931 (71)	"	0.99960 (72)	"
3	U233-MF-005 (2)	0.99785 (64)	"	0.99900 (59)	"
4	Falstaff (1)	0.99040 (104)	"	0.99017 (106)	"
5	ORNL-11	0.99596 (41)	"	0.99708 (37)	"
6	Godiva	0.99728 (63)	"	0.99647 (60)	"
7	Flattop-25	0.99790 (63)	"	0.99660 (59)	"
8	Godiver	0.99539 (80)	"	0.99675 (79)	"
9	HISS/HUG	1.01264 (47)	"	1.01016 (46)	"
10	ZEUS (2)	0.99722 (73)	0.99655 (71)	0.99538 (75)	"
11	HEU-MT-003 (4)	0.98257 (88)	"	0.98413 (79)	0.98374 (80)
12	ORNL-10	0.99874 (39)	"	0.99835 (40)	"
13	IEU-MF-003	1.00046 (57)	"	0.99973 (61)	"
14	BIG TEN	1.00987 (55)	"	1.00725 (54)	"
15	IEU-MF-004	1.00381 (62)	"	1.00315 (67)	"
16	IEU-CT-002 (3)	1.00024 (70)	"	1.00029 (74)	0.99987 (71)
17	BAW XI (2)	0.99837 (60)	"	0.99863 (70)	"
18	SHEBA-2	1.01064 (77)	"	1.01018 (82)	"
19	Jezebel	0.99694 (57)	"	0.99772 (60)	"
20	Jezebel-240	0.99883 (60)	"	0.99884 (57)	"
21	Flattop-Pu	1.00138 (66)	"	1.00266 (70)	"
22	PU-MF-011	0.99736 (76)	"	0.99700 (72)	"
23	Pu Buttons	0.99581 (67)	"	0.99735 (68)	"
24	HISS/HPG	1.01126 (59)	"	1.00936 (56)	"
25	PNL-33	1.00578 (79)	"	1.00545 (80)	"
26	PNL-2	1.00031 (104)	"	1.00219 (95)	"
	Notes:	" = result identical to that of column at left (NN) = std deviation is NN x 10^{-5}			

using the current version of the SGI Fortran-90 compiler and associated libraries.

In addition, Monte Carlo eigenvalue calculations are very sensitive to computer roundoff due to their iterative nature – small differences in even a single particle history will propagate through all future generations. (Fixed source calculations are less sensitive to roundoff, since generations are not used; roundoff differences affect only a single history and do not propagate.)

Table III shows results for the criticality suite run with ENDF66 data on different computer systems. For the SGI and HP/Alpha systems, most results argree exactly, while a few cases show differences which are within statistics. The results for the PC/Linux system shows more differences attributable to roundoff due to a much different Fortran-90 compiler, different optimization level, and different precision in some

Table III. Criticality Suite Results, Using New Data and Different Computers

Name		K-effective Results Using ENDF66 Data			
		MCNP4C2 SGI Origin2K	MCNP5 SGI Origin2K	MCNP5 HP Alpha	MCNP5 PC/Linux
1	Jezebel-233	0.99106 (56)	"	"	"
2	Flattop-23	0.99960 (72)	"	"	0.99934 (74)
3	U233-MF-005 (2)	0.99900 (59)	"	"	"
4	Falstaff (1)	0.99017 (106)	"	"	0.98807 (111)
5	ORNL-11	0.99708 (37)	"	"	"
6	Godiva	0.99647 (60)	"	"	"
7	Flattop-25	0.99660 (59)	"	"	0.99680 (60)
8	Godiver	0.99675 (79)	"	"	0.99546 (73)
9	HISS/HUG	1.01016 (46)	"	"	1.01010 (49)
10	ZEUS (2)	0.99538 (75)	"	"	0.99449 (72)
11	HEU-MT-003 (4)	0.98374 (80)	0.98413 (79)	0.98413 (79)	0.98364 (75)
12	ORNL-10	0.99835 (40)	"	"	"
13	IEU-MF-003	0.99973 (61)	"	"	"
14	BIG TEN	1.00725 (54)	"	"	"
15	IEU-MF-004	1.00315 (67)	"	"	"
16	IEU-CT-002 (3)	0.99987 (71)	1.00029 (74)	1.00035 (74)	1.00169 (77)
17	BAW XI (2)	0.99863 (70)	"	"	0.99843 (62)
18	SHEBA-2	1.01018 (82)	"	"	"
19	Jezebel	0.99772 (60)	"	"	"
20	Jezebel-240	0.99884 (57)	"	"	"
21	Flattop-Pu	1.00266 (70)	"	"	1.00195 (70)
22	PU-MF-011	0.99700 (72)	"	"	"
23	Pu Buttons	0.99735 (68)	"	"	"
24	HISS/HPG	1.00936 (56)	"	1.00977 (55)	1.01046 (56)
25	PNL-33	1.00545 (80)	"	"	"
26	PNL-2	1.00219 (95)	"	"	"
	Notes:	" = result identical to that for MCNP4C2 on SGI (NN) = std deviation is NN x 10^{-5}			

of the arithmetic and data constants. Only one of the cases, IEU-CT-002(3), differs outside of the 1-sigma interval, and it agrees at the 2-sigma level.

Analytic Benchmarks for Criticality

Reference [10] provides a set of 75 criticality problems found in the literature for which exact analytical solutions are known. Number densities, geometry, and cross-section data are specified exactly for these problems. As part of the MCNP5 verification, 10 of these analytic benchmark problems were run to high precision using MCNP5 on 2 different computer systems - a Silicon Graphics Origin 2000 supercomputer and a Pentium-III PC running Windows-2000. The 10 cases selected from [10] are listed in Table IV along with both the analytic results and the MCNP5 results. For all cases, a total of 210

Table IV. Results for Analytic Criticality Benchmarks

	Name	Description	Exact K-eff	MCNP5 K-eff
1	Ua-1-0-IN	Infinite medium, 1 group	2.25	2.24996 (24)
2	Ua-1-0-SP	Sphere, 1 group	1.0	0.99990 (23)
3	Uc-H2O(2)-1-0-SP	Reflected sphere, 1 group	1.0	0.99985 (23)
4	UD2O-1-0-CY	Cylinder, 1 group	1.0	0.99996 (15)
5	PUa-1-1-SL	Slab, 1 grp, P1 scatter	1.0	0.99989 (26)
6	UD2OB-1-1-SP	Sphere, 1 grp, P1 scatter	1.0	0.99993 (17)
7	PU-2-0-IN	Infinite medium, 2 group	2.683767	2.68375 (07)
8	URRa-2-0-SL	Slab, 2 group	1.0	1.00001 (34)
9	URR-6-0-IN	Infinite medium, 6 group	1.60	1.59999 (02)
10	URRd-H2O(1)2-0-ISLC	Slab, 2 group	1.0	0.99986 (41)
		Note: (NN) = std deviation is NN x 10^{-5}		

generations were run, with the first 10 discarded for settling. For cases 1-9, 40,000 histories were used per generation, for a total of 8M histories in the 200 active cycles. For case 10, only 5,000 histories per generation were run, for

a total of 1M histories in the active generations. In all cases, MCNP5 results were identical on the SGI system and PC, and all results were in statistical agreement with the exact k-effective values.

Radiation Shielding Validation Suite

The radiation-shielding validation suite [9] contains three subcategories: time-of-flight spectra for neutrons from pulsed spheres, neutron and photon spectra at shield walls within a simulated fusion reactor, and photon dose rates. Two of the cases are coupled neutron-photon calculations, while the others are exclusively neutron or exclusively photon calculations.

The time-of-flight cases are a subset of the pulsed-sphere experiments that were performed at Lawrence Livermore National Laboratory from the late 1960s into the 1980s. The objective of these experiments was to measure the neutron emission spectrum from a variety of materials bombarded by 14 MeV neutrons.

The second subset of cases in the radiation-shielding validation suite is based on a series of experiments that was performed at Oak Ridge National Laboratory in 1980. The objective of the experiments was to simulate the deuterium-tritium neutron spectrum that would exist at the first wall of a fusion reactor as well as the spectrum of secondary photons that would be produced from neutron interactions within that wall.

The cases in the last subset of the radiation-shielding validation suite are based on experimental measurements of photon dose rates. The first case is based on a 1980 measurement of air-scattered photon radiation far from the source ("skyshine"). The second case is an idealization of a number of measurements of the radiation environment in an open field covered by fallout. The remaining four cases model some of the Hupmobile thermoluminescent dosimeter (TLD) experiments performed at Lawrence Berkeley Laboratory between 1967 and 1969.

The MCNP calculations for the cases in this suite that include photons use the MCPLIB02 photon data library [13] for all nuclides. The calculations for the radiation-shielding validation suite all were performed in sequential mode on a Silicon Graphics Origin 2000 supercomputer. Each case employed 1,000,000 particle histories.

MCNP5 produces exactly the same tally values as MCNP4C2 for all the cases in the shielding validation suite listed in Tables IV-VI, given the same data library. This is true for both the older ENDF60 data and the new ENDF66 data.

Verification Summary

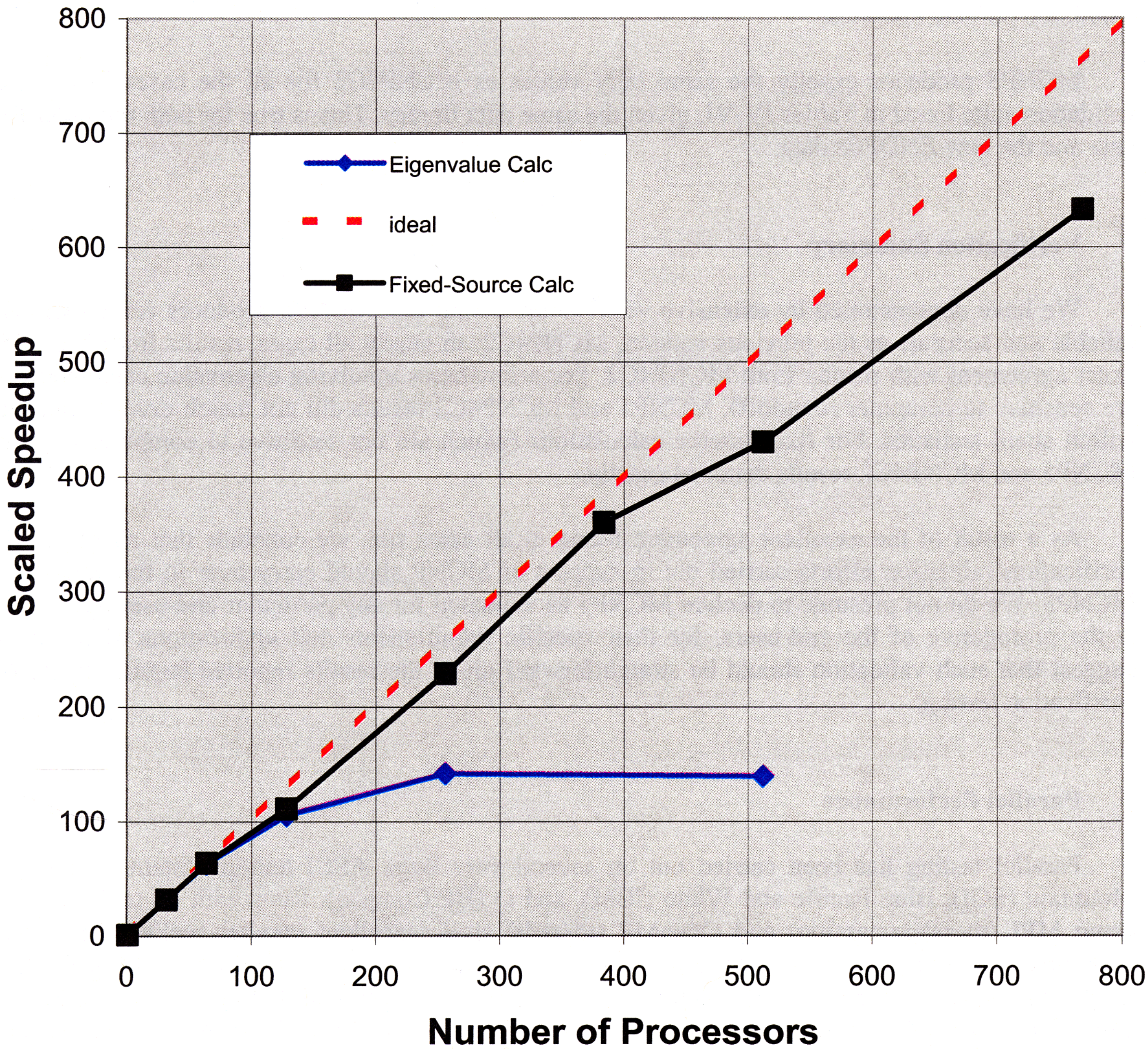
We have demonstrated by extensive verification testing that MCNP5 produces results which are as reliable and accurate as the previous version, MCNP4C2. In nearly all cases, results from MCNP5 are in exact agreement with results from MCNP4C2. For a few cases involving eigenvalue calculations (which are sensitive to computer roundoff), MCNP5 and MCNP4C2 results did not match exactly, but did agree within small statistics. For fixed-source calculations (which are not sensitive to computer roundoff), all MCNP5 and MCNP4C2 results matched exactly.

As a result of the excellent agreement found in all cases run, we conclude that all of the previous verification/validation efforts carried out in support of MCNP should carry over to the present version, MCNP5. We do not presume to declare MCNP5 as validated for any particular end-user application (that is the prerogative of the end-users, for their specific requirements and applications of the code), but suggest that such validation should be straightforward given the results reported herein for the MCNP5 verification testing.

Parallel Performance

Parallel testing has been carried out on several very large ASCI teraop systems, including Blue Mountain (SGI), Blue Pacific and White (IBM), and Q (HP/Compaq). Runs with up to 2048 processors using MPI (message-passing) and OpenMP (threads) show excellent parallel scaling for fixed-source problems, with efficiencies of 70-80% or higher. Eigenvalue calculations do not scale as well, due to the need for more frequent rendezvous operations at the end of each generation. Figure 2 shows the measured parallel scaling on the HP-Compaq Q system which is based on 64-bit Alpha chips. The results shown are for the parallel scaled speedup, where the number of histories per CPU is held constant, so that problems using more CPUs run more histories in proportion. Parallel scaling is excellent for the fixed-source calculations, with 82% parallel efficiency using 768 processors. Parallel performance asymptotes at a scaled speedup of about 140x for the eigenvalue problem, however. This is a result of the communications overhead incurred at the rendezvous points at the end of each generation, and is an unavoidable limit to the parallel performance of Monte Carlo eigenvalue calculations. See [14] for a discussion of the parallel scaling. Parallel calculations on Linux clusters and Windows PC clusters using MPI have been found to perform well, and results from these tests will be also presented.

Figure 2. MCNP5 Parallel Scaled Speedup for Fixed-Source and Eigenvalue Problem s
(ASCI Q system, using MPI and OpenMP, 4 threads/MPI-task)



Acknowledgments

MCNP5 was developed by the X-5 Monte Carlo Team at LANL: F.B. Brown, R.F. Barrett, T.E. Booth, J.S. Bull, L.J. Cox, R.A. Forster, J.T. Goorley, H.G. Hughes, R.D. Mosteller, S.E. Post, R.E. Prael, E.C. Selcow, A. Sood, J. Sweezy, T. Roberts. The MCNP5 data libraries were developed by the X-5 Nuclear Data Team at LANL: R.C. Little, J.M. Campbell, S.C. Frankle, M.C. White. This work was performed under the auspices of the U.S. Department of Energy by the University of California Los Alamos National Laboratory under contract No. W-7405-Eng-36.

References

1. J.F. Briesmeister, Ed., "MCNP – A General Monte Carlo N-Particle Transport Code – Version 4C," LA-13709-M, Los Alamos National Laboratory (March, 2000).
2. L.J. Cox, Editor, "LANL ASCI Software Engineering Requirements," LA-UR-02-888, Los Alamos National Laboratory (February, 2002).
3. "MPI: A Message Passing Interface", http://www-unix.mcs.anl.gov/mpi/index.html
4. "OpenMP Fortran Application Program Interface", http://www.openmp.org
5. X-5 Monte Carlo Team, "MCNP – A General Monte Carlo N-Particle Transport Code, Version 5," LA-UR-03-1987 (Vol I: Overview and Theory), LA-CP-03-0245 (Vol II: User's Guide), LA-CP-03-0284 (Vol III: Developer's Guide) (April, 2003).
6. F.B. Brown and Y. Nagaya, "The MCNP5 Random Number Generator", *Trans. Am. Nucl. Soc.* (Nov., 2002)
7. R. Schwarz., "The MCNP Visual Editor," http://www.mcnpvised.com
8. Joann M. Campbell, Stephanie C. Frankle, and Robert C. Little, "ENDF66: A Continuous-Energy Neutron Data Library for MCNP4C," *Proceedings of the 12th Biennial Topical Meeting of the Radiation Protection and Shielding Division of the American Nuclear Society*, pp. 19-38, Santa Fe, New Mexico, USA, April 14-18 (2002).
9. Russell D. Mosteller, "Validation Suites for MCNP," *Proceedings of the 12 Biennial Topical Meeting of the Radiation Protection and Shielding Division of the American Nuclear Society*, pp. 62-70, Santa Fe, New Mexico, USA, April 14-18 (2002).
10. A. Sood, R.A. Forster, D.K. Parsons, "Analytical Benchmark Test Set for Criticality Code Verification," *Prog. Nuc. Energy*, Vol 42, No. 1, pp. 55-106 (2003).
11. *International Handbook of Evaluated Criticality Safety Benchmark Experiments*, OECD Nuclear Energy Agency report NEA/NSC/DOC(95)03 (rev., September 2002).
12. Robert C. Little and Robert E. MacFarlane, "ENDF/B-VI Neutron Library for MCNP with Probability Tables," Los Alamos National Laboratory report LA-UR-98-5718 (1998).
13. H. G. Hughes, "Information on the Photon Library MCPLIB02," Los Alamos National Laboratory internal memorandum X-6:HGH-93-77 (revised 1996).
14. S. Matsuura, F.B. Brown, R.N. Blomquist, "Parallel Monte Carlo Eigenvalue Calculations," *Trans. Am. Nucl. Soc.* (Nov. 1994)