LA-UR-16-23533

Lecture Notes on Sensitivity-Uncertainty Based Nuclear Criticality Safety Validation

Jennifer Alwin, Forrest Brown, Michael Rising

Monte Carlo Codes Group, XCP-3
Los Alamos National Laboratory

2016-05-02

Contents

Lecture Notes on Sensitivity-Uncertainty Based Nuclear Criticality Safety Validation

(blank)

(blank)

Nuclear Criticality Safety

-

Validation - I

Los Alamos
NATIONAL LABORATORY
EST.1943

NCSP
NUCLEAR CRITICALITY SAFETY PROGRAM

Background

- **Why do we care about Validation?**
  - ANSI/ANS-8.24 Foreword: "…the industry need to optimize operations and reduce unnecessary conservatism has increased. Thus, the scrutiny and importance placed on validation has increased in recent years."
  - **Ensure what NCS determines to be subcritical is actually subcritical**
    - People make mistakes
    - Computer codes and nuclear data have approximations and errors
  - **Criticality safety:**
    - Focus on avoiding worst-case combination of mistakes, uncertainties, errors, ...
    - Rigor & conservatism always; never wishful thinking or "close enough"
  - **How can we be confident in assessing subcriticality?**
    - Verify that codes work as intended
    - Validate codes + data + methods against nature (experiments)

Orders, Standards, Guides for NCS

- 10 CFR 830 Subpart A, Quality Assurance
- 10 CFR 830 Subpart B, Nuclear Safety Management

- DOE O 414.1C, Quality Assurance
- DOE G 414.1-4, Safety Software Guide for use with 10CFR 830 Subpart A, Quality Assurance Requirements
- DOE G 421.1-2, Implementation Guide for Use in Developing Documented Safety Analyses to Meet Subpart B of 10 CFR 830
- DOE O 420.1C, Facility Safety
- DOE O 426.2 Personnel Selection, Training, Qualification, and Certification Requirements

- **DOE-STD-3007-2007, Guidelines for Preparing Criticality Safety Evaluations at DOE Nonreactor Nuclear Facilities**
- DOE STD 1134-1999 Review Guide for Criticality Safety Evaluations
- DOE-STD-1158-2010, Self-Assessment Standard for DOE Contractor Criticality Safety Programs
- DOE-STD-3009-1994, Preparation Guide for U.S. Department of Energy Nonreactor Nuclear Facility Safety Analysis
- DOE-STD-1186-2004, Specific Administrative Controls
- DOE-STD-1027-1992, Hazard Categorization and Accident Analysis Techniques for Compliance with DOE Order 5480.23, Nuclear Safety Analysis Reports

- **SD130,R3 Nuclear Criticality Safety Program**
- **NCS-GUIDE-01,R2 Criticality Safety Evaluations**

- **ANSI/ANS-8.1-2014, Nuclear Criticality Safety in Operations with Fissionable Materials Outside Reactors**

- ANSI/ANS-8.3-2003, Criticality Accident Alarm System
- ANSI/ANS-8.5-1996(R2007), Use of Borosilcate-Glass Raschig Rings as a Neutron Absorber in Solutions of Fissile Material
- ANSI/ANS 8.7-1998(R2012), Nuclear Criticality Safety in the Storage of Fissile Materials
- ANSI/ANS-8.10-2005, Criteria for Nuclear Criticality Safety Controls in Operations with Shielding and Confinement
- ANSI/ANS 8.14-2004, Use of Soluble Neutron Absorbers in Nuclear Facilities Outside Reactors
- ANSI/ANS 8.17-2004, Criticality Safety Criteria for the Handling, Storage, and Transportation of LWR Fuel Outside Reactors
- ANSI/ANS-8.19-2014, Administrative Practices for Nuclear Criticality Safety
- ANSI/ANS 8.20-1991(R2005), Nuclear Criticality Safety Training
- ANSI/ANS-8.21-1995(R2001), Use of Fixed Neutron Absorbers in Nuclear Facilities Outside Reactors
- ANSI/ANS-8.23-2007, Nuclear Criticality Accident Emergency Planning and Response

- **ANSI/ANS 8.24-2007, Validation of Neutron Transport Methods for Nuclear Criticality Safety Calculations**

- ANSI/ANS 8.26-2007, Criticality Safety Engineer Training and Qualification Program

MCNP Verification & Validation Suites

Verification Suites

- **REGRESSION**
  - 161 code test problems
  - Run by developers for QA checking (100s of times per day)
- **VERIFICATION_KEFF**
  - 75 analytic benchmarks (0-D and 1-D)
  - Exact solutions for k_{eff}
  - Past – multigroup, New – continuous-energy
- **VERIFICATION_GENTIME**
  - 10 benchmarks (analytic or comparisons to Partisn) for reactor kinetics parameters
- **KOBAYASHI**
  - 6 void & duct streaming problems, with point detectors, exact solutions
- **Ganapol Benchmarks** [in progress]
  - Exact, semi-analytic benchmark problems
  - Fixed source, not criticality
- **Gonzales Benchmark** [in progress]
  - Exact analytic benchmark with elastic scatter, including free-gas scatter

Validation Suites

- **VALIDATION_CRITICALITY**
  - 31 ICSBEP Cases
  - Too small a suite for serious V&V
  - Today, used for
    - Code-to-code verification, with real problems & data
    - Compiler-to-compiler verification, with real problems & data
    - Timing tests for optimizing MCNP coding & threading
- **VALIDATION_CRIT_EXPANDED**
  - 119 ICSBEP Cases
  - Broad-range validation, for developers
- **VALIDATION_CRIT_WHISPER**
  - 1101 ICSBEP Cases
  - Used with Whisper methodology for serious validation
  - Will be expanded, as time permits

Establishing Subcriticality

- ***Any method*** used to determine the subcritical state of a fissionable material system must be validated.

- **Preferred method is direct use of experimental data**
  - Where applicable data are available, subcritical limits shall be established on bases derived from experiments, with adequate allowance for uncertainties in the data. In the absence of directly applicable experimental measurements, the limits may be derived from calculations made by a method shown by comparison with experimental data to be valid in accordance with Sec. 4.3 (ANSI/ANS-8.1-2014 4.2.7)
    - Code-to-code comparison doesn't meet requirement

  - Use of subcritical limit data provided in ANSI/ANS standards or accepted reference publications does not require further validation (ANSI/ANS-8.1-2014 4.3)

Validation: Definitions (1)

- **From ANSI/ANS-8.24-2007, Validation of Neutron Transport Methods for Nuclear Criticality Safety Calculations:**
  - **Verification:** **The process of confirming that the *computer code system* correctly performs numerical calculations.**
  - **Validation:** **The process of quantifying (e.g., establishing the appropriate *bias* and *bias uncertainty*) the suitability of the computer code system for use in nuclear criticality safety analyses.**
  - **Computer code system:** **A *calculational method*, computer hardware, and computer software (including the operating system).**
  - **Calculational Method:** **The mathematical procedures, equations, approximations, assumptions, and associated numerical parameters (e.g., cross sections) that yield the calculated results.**

Validation: Definitions (2)

- **From ANSI/ANS-8.24-2007, Validation of Neutron Transport Methods for Nuclear Criticality Safety Calculations:**
  - **Bias:** The systematic difference between calculated results and experimental data.
  - **Bias Uncertainty:** The uncertainty that accounts for the combined effects of uncertainties in the benchmarks, the calculational models of the benchmarks, and the calculational method.
  - **Calculational Margin:** An allowance for bias and bias uncertainty plus considerations of uncertainties related to interpolation, extrapolation, and trending.
  - **Margin of Subcriticality:** An allowance beyond the calculational margin to ensure subcriticality.
  - **Validation Applicability:** A domain, which could be beyond the bounds of the benchmark applicability, within which the margins derived from validation of the calculational method have been applied.

Excerpts from ANSI/ANS - 8.24-2007

5.1 Appropriate system or process parameters that correlate the experiments to the system or process under consideration shall be identified. .....

5.2 Normal and credible abnormal conditions for the system or process shall be identified when determining the appropriate parameters and their range of values.

5.4 Selected benchmarks should encompass the appropriate parameter values spanning the range of normal and credible abnormal conditions anticipated for the system or process to which the validation will be applied.

7.2 The validation applicability should not be so large that a subset of the data with a high degree of similarity to the system or process would produce an upper subcritical limit that is lower than that determined for the entire set. This criterion is recommended to ensure that a subset of data that is closely related to the system or process is not nonconservatively masked by benchmarks that do not match the system as well.

8.1 The validation activity shall be documented with sufficient detail to allow for independent technical review.

8.1.5 The margin of subcriticality and its basis shall be documented.

8.2 An independent technical review of the validation shall be performed. The independent technical review should include, but is not limited to, the following:

(1) a review of the benchmark applicability;

(2) a review of the input files and output files to ensure accurate modeling and adequate convergence;

(3) a review of the methodology, and its use, for determining bias, bias uncertainty, and margins;

(4) concurrence with the validation applicability.

Upper Subcritical Limit

- To consider a simulated system subcritical, the computed keff must be less than the Upper Subcritical Limit (USL):

$$K_{calc} + 2\sigma < \text{USL}$$

$$\text{USL} = 1 + (\text{Bias}) - (\text{Bias uncertainty}) - \text{MOS}$$

[additonal AoA margin may be appropriate, case-by-case basis]

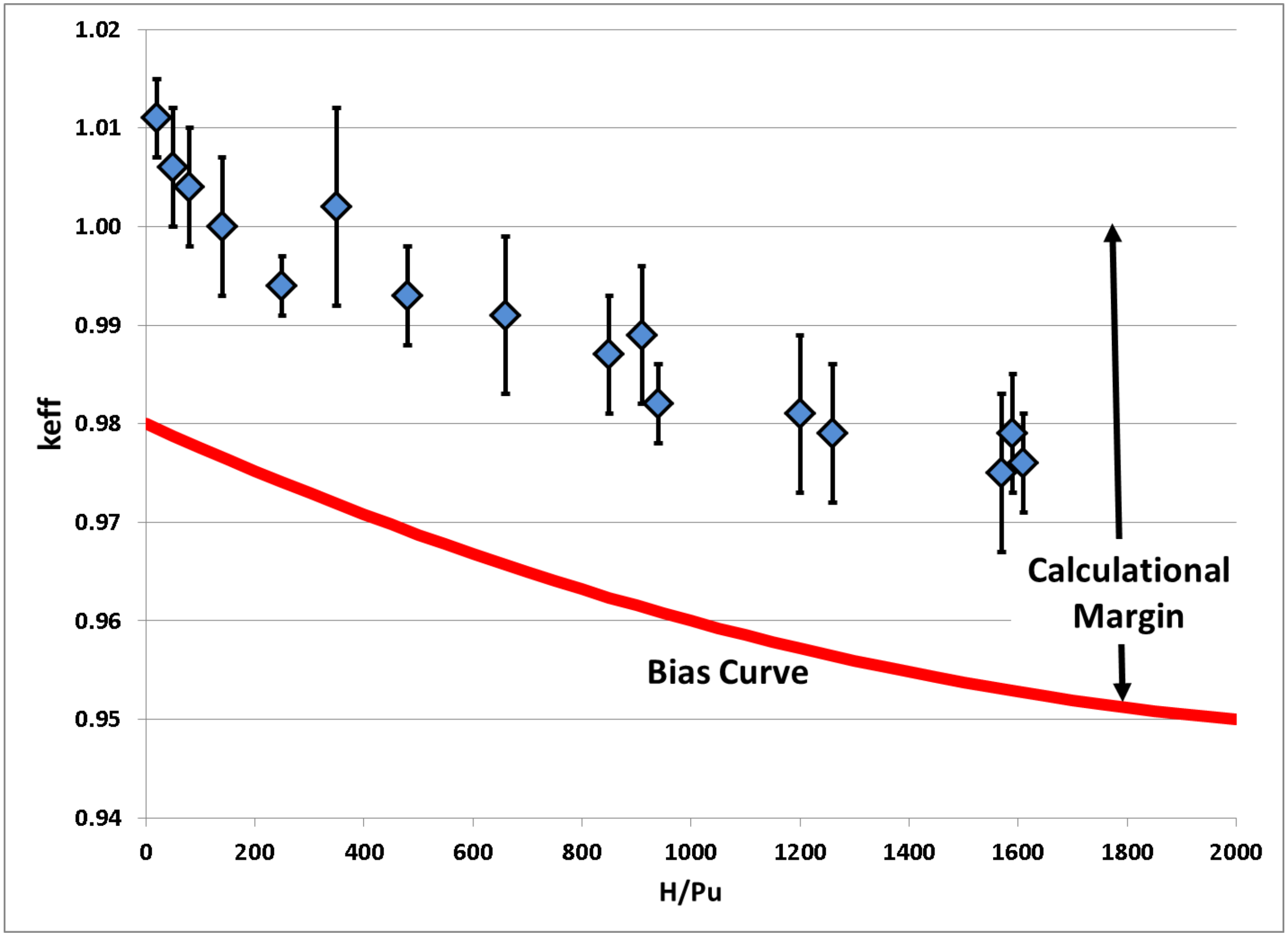
- The bias and bias uncertainty are at some confidence level, typically 95% or 99%.
  - These confidence intervals may be derived from a normal distribution, but the normality of the bias data must be justified.
  - Alternatively, the confidence intervals can be set using non-parametric methods.

Calculational Margin

- **The calculational margin is the sum of the bias and the bias uncertainty.**
  - **Bias: represents the systematic difference between calculation and benchmark experiments.**
  - **Bias uncertainty: relates to uncertainties in the experimental benchmarks and the calculations.**
  - **Bias & bias uncertainty are routine calculations, for a given application & set of benchmarks**
  - **Bias & bias uncertainty are only credible when the application & chosen benchmarks are neutronically similar**
  - **Often quoted as 95/95 confidence, meaning that the calculation margin bounds 95% of the benchmark deviations at the 95% confidence level (assuming normality).**
  - **May trend calculational margin based upon physical parameters.**

Calculational Margin Example

- **Hypothetical bias curve**
  - Selected experiments with Pu metal and water mixtures



Margin of Subcriticality

- **To establish a Margin of Subcriticality (MOS) need to consider the process, validation, codes, data, etc. holistically.**
  - **Confidence in the codes and data.**
    - More mature codes that are widely used have greater confidence than newer ones.
    - Deterministic methods require additional margin beyond Monte Carlo because of numerical issues (e.g., ray effects, discretization errors, self-shielding approximations, etc.).
  - **Adequacy of the validation**
    - Unlikely to find a benchmark experiment that is exactly like the model being simulated.
    - Based on trending analysis of physical parameters and/or sensitivity and uncertainty studies, can quantify "similarity".
    - Sparsity of benchmark data, extrapolations, and wide interpolations necessitate larger margins.

- **Major contributors**
  - **Margin for uncertainties in nuclear cross-section data**
  - **Margin for unknown errors in codes**
  - **Additional margin to consider the limitations of describing process conditions based upon sensitivity studies, operating experience, administrative limits, etc.**

Potential Bias Cause

- **Nuclear Data**
  - **Different use of nuclear data lead to different biases**
    - Requires different critical experiments to validate different energy ranges
    - Systems with higher sensitivity to highly uncertain cross sections may have larger biases
    - Material missing from either experiments or safety models can affect bias accuracy
  - **Ideally, critical experiments used for validation will use the same data in the same way the criticality safety evaluations models do, thus they will have the same bias**
    - Sensitivity and uncertainty analysis techniques can be used to do a quantitative comparison

Selection of Benchmarks

- **Select critical experiments that you expect have the same bias and the criticality safety evaluation models**
  - Similar neutron energy spectrum (EALF, AEG, etc.)
  - Similar fissionable materials and isotopics
  - Similar neutron absorbers (Cd, Gd, B, Fe, Ti, etc.)
  - Similar neutron reflectors (air, water, steel, lead, concrete, etc.)
  - Similar geometries
- **Due to variation in criticality safety evaluation models, you may need multiple sets or sets covering a parameter range**
- **How many experiments are needed?**
  - As many experiments that are similar or “applicable” to the criticality safety evaluation models
  - If an experiment is exactly the same as the fissionable material operation, subcritical limits may be derived directly from experiments with no need to calculate the result
  - “Response to CSSG Tasking 2014-02, Validation with Limited Benchmark Data,” September 21, 2015, http://ncsp.llnl.gov/cssg/taskandresponse/2014/2014-02_Response_on_Validation_with_Limited_Data_09-21-15.pdf
- If no benchmark experiments exist that match the system being evaluated, it may be possible to interpolate or extrapolate from existing benchmark data to that system. Sensitivity and uncertainty analysis tools may be used to assess the applicability of benchmark problems to the system being analyzed. (DOE-STD-3007-2007)

Selection of Benchmarks

- **Historically, engineering judgement ("expert") has been used**
- **Based on the analysts understanding of what is important to the problem**
- **This can, in some cases, lead to questions**
  - Validation of U solution with U metal experiments
  - Experiments with strong absorbers included that were not present in safety models
  - Validation of fuel rod lattices with solution or metal experiments
  - Overly broad critical experiment set (i.e., single broad validation set) used. There is a temptation to try to create a validation that covers all operations.
    - The validation applicability should not be so large that a subset of the data with a high degree of similarity to the system or process would produce an upper subcritical limit that is lower than that determined for the entire set. This criterion is recommended to ensure that a subset of data that is closely related to the system or process is not nonconservatively masked by benchmarks that do not match the system as well (ANSI/ANS-8.24 7.2)

How do NCS analysts develop engineering judgement?

- Could take years of experience and study of individual benchmarks
- Could rely on guidance from other qualified analysts to caution (missing materials, neutron absorbers present in typical materials not always obvious, etc.)

Selection of Benchmarks

- **Identify the parameters that correlate experiments to the system or process being analyzed in the criticality safety evaluation (ANSI/ANS-8.24 5.1)**
- **Normal and credible abnormal conditions shall be considered when determining the parameters and range of parameters (ANSI/ANS-8.24 5.2)**
  - The experiments selected need to be similar to the normal and abnormal conditions you need to evaluate
- **Experiments shall be reviewed for completeness and accuracy before being used in a validation (ANSI/ANS-8.24 5.3)**
  - An experiment may be useful for setting limits, but not be sufficiently complete or accurate to use as a benchmark (This can happen with subcritical experiments, process specific experiments, and in-situ experiments)
- **Benchmarks should cover the parameter range (ANSI/ANS-8.24 5.4)**
  - Avoid the need to extrapolate beyond the range of the available data
- **Benchmarks selected should be consistent with the modeling capabilities of the code system being validated (ANSI/ANS-8.24 5.5)**

Selection of Benchmarks

- **Benchmarks should be drawn from multiple sources to minimize systemic error** (ANSI/ANS-8.24 5.6)

- **Methods used to analyze benchmarks shall be the same *computational method* being used in the criticality safety evaluation** (ANSI/ANS-8.24 5.7)
  - **Albedos, variance reduction techniques, cross section processing, sometimes geometry options**
- **Benchmark modeling shall be the responsibility of individuals experienced in the use of the *computational method*** (ANSI/ANS-8.24 5.8)

- **Benchmark models prepared by outside organizations should be evaluated for appropriateness, completeness & accuracy** (ANSI/ANS-8.24 5.9)

  - **ICSBEP handbook cautions against using their input files without review**
  - **Modeling techniques used may not be adequately similar to that used in the criticality safety evaluation models**

Calculating Bias and Bias Uncertainty

- **There are many methods and codes used to calculate bias and bias uncertainty. Some examples are:**
  - Site specific statistical analysis procedures
  - NUREG/CR-6698 (Methods originally developed at SRNL)
  - USLSTATS
  - Whisper
- **The validation study should describe (i.e., either directly or by reference) the method used to calculate the bias and bias uncertainty.**
- **Make sure the data meets all prerequisites (e.g., normality, number of points, etc.) for the method used.**
- **If it does not, use a different method.**
- **In general, <u>positive biases*</u> (calculated value is higher than experiment value) are <u>not credited</u> for criticality safety purposes. If they are used, shall be justified based on an understanding of the cause of bias.** (Positive biases are sometimes used in reactor or nuclear experiment design.)

  *The sign of the bias is arbitrary. For the purposes of ANSI/ANS-8.24, it has been defined to be positive when the calculated values exceed the experimental values, but it could be defined otherwise.

Results Distribution

- **Some bias and bias uncertainty determination methods require that the distribution be "normal"**
- **Some examples of normality tests**
  - Visual inspection of frequency bar charts (qualitative chi-square)
  - Chi-squared tests
  - Kolomogrov-Smirnov
  - Shapiro-Wilk
  - Anderson-Darling
- **For trending analysis, look at normality of residuals (difference between best fit line and $k_{eff,normalized}$**
- **Most normality tests (e.g., those used in USLSTATS and NUREG/CR-6698) accept the distribution as normal unless 95% sure that it is not normal. This is a pretty low threshold.**
- **You should do numerical tests for normality, but a histogram plot is sometimes adequate.**
- **Look out for distributions with multiple peaks, skewed distributions, and tails that are obviously inconsistent with normal distribution**
- **Even if you do use numerical tests for normality, you should still do the histogram, and verify to yourself that the pictures and the numbers match.**

S/U Analysis

- **Sensitivity analysis quantifies how variation of material properties or nuclear data affects k_{eff}.**

- **Techniques:**
  - **Manual model variation**
    - Change material densities or temperatures
    - Change dimensions
    - Used to justify simplifications and to quantify the impact of manufacturing tolerances and uncertainties
    - Used to support margin adopted for validation weaknesses
  - **Perturbation theory methods (Whisper and TSUNAMI)**
    - These systems use perturbation theory to provide nuclide, reaction, energy, and location dependent sensitivity data
    - Typically in units of $(\Delta k/k)/(\Delta\sigma/\sigma)$, or the fractional change in k_{eff} due to a fractional change in the nuclear data value.
    - Sensitivity analysis improves understanding of what is important for k_{eff} determination

S/U Analysis

- **Uncertainty analysis combines sensitivity data with nuclear data uncertainty information to yield:**
  - Uncertainty in k_{eff} due to uncertainty in nuclear data for specific nuclides and reactions
  - These uncertainties can be used to provide a defensible basis for margin to cover validation weaknesses
  - The uncertainty information for two different systems may be compared to quantify how much uncertainty the systems have in common
  - If two systems are similarly sensitive to the same nuclear data, then they should have the same bias
  - The c_k correlation coefficient compares two systems, assessing the potential for common bias for each nuclide, reaction, and energy group
  - C_k = 1 means two systems use same data in same way

S/U Analysis

- **S/U analysis:**
  - Sensitivity data can be used:
    - Improve understanding of systems
    - Suggest or defend modeling simplifications
    - Suggest critical experiments that might be useful for validation
    - Critical experiment design
    - In GLLS for estimating margin for data uncertainties (Whisper and TSURFER)
  - K_{eff} uncertainty data can be used:
    - Improve understanding of potential bias causes
    - Estimate how large biases related to a mixture or nuclide might be and provide a defensible basis for margin selection to cover validation weaknesses
  - C_k can be used:
    - Select critical experiments
    - As a trending parameter in USL determination
- **CSSG Response on Validation with Limited Data:** *"For those situations where a nuclide is determined to be important and limited data exist, validation may still be possible. However, an additional margin should be used to compensate for the limited data. This margin is separate from, and in addition to, any margin needed for extending the benchmark applicability to the validation. Sensitivity and uncertainty tools may be used as part of the technical basis for determining the magnitude of the margin."*

Comparison of Validation Approaches (Simplified)

	Traditional, Simple	Traditional, Enhanced	Modern
Benchmark Collection	Expert judgment, 1 set, Geometry & materials cover applications	Expert judgment, Several subsets (metal, solutions, other)	Large collection with sensitivity profile data, Reject outliers, Estimate missing uncertainties
Selecting Benchmarks		Expert judgment, Select subset based on geometry & materials	Automatically select benchmarks with sensitivity profiles closest to application
Calculational Margin	Determine bias & bias uncertainty	Determine bias & bias uncertainty, Possible trending within subset	Determine bias & bias uncertainty, Automatically use weighting based on application-specific Ck similarities
Margin of Subcriticality	Expert judgment, Very large	Expert judgment, Large	Automatically determine specific margin for data uncertainty by GLLS, Code-expert judgment for code, Expert judgment for additional
Comment	Easy to use, Highly dependent on expert judgment, Requires large conservative MOS	More work if trending, Very dependent on expert judgment, Subsets & trending may permit smaller MOS	Computer-intensive, quantitative, Less reliance on expert judgment, Calculated estimate for most of MOS

Documentation and Independent Technical Review

- **Documentation:**
  - Sufficient detail to allow for independent technical review
  - Describe computer code system being validated
  - Justify selection of benchmarks
    - Identify data sources through references
    - Document benchmark applicability (AoA)
  - Methods and calculations supporting the determination of bias and bias uncertainty, calculational margin, validation applicability
    - If using trending analysis, document technical bases
  - Validation applicability (extension beyond AoA)
    - Justification for extrapolations or wide interpolations
    - Discuss and justify differences between validation applicability and system or process parameters
    - Describe limitations (e.g., gaps in data, missing data)
- **Independent Technical Review:**
  - review benchmark applicability
  - Input files and output files
  - Methodology for determining bias, bias uncertainty, margins
  - Concurrence with validation applicability

Nuclear Criticality Safety

-

Validation – II

-

Using MCNP & Whisper

Los Alamos NATIONAL LABORATORY EST.1943

NCSP NUCLEAR CRITICALITY SAFETY PROGRAM

Topics

- Whisper
  - Summary, methodology, status
  - Sensitivity profiles
  - Covariance data
  - Correlation coefficients
  - USLs & Validation
- whisper_mcnp
  - Usage, files, output
- whisper_usl
  - Usage, files, output
- Whisper.out
- Conclusions & Discussion

Whisper – Summary

Whisper - Software for Sensitivity-Uncertainty-Based Nuclear Criticality Safety Validation

Whisper is computational software designed to assist the nuclear criticality safety (NCS) analyst with validation studies with the Monte Carlo radiation transport package MCNP. Standard approaches to validation rely on the selection of benchmarks based upon expert judgment. Whisper uses sensitivity/uncertainty (S/U) methods to select relevant benchmarks to a particular application or area of applicability (AOA), or set of applications being analyzed. Using these benchmarks, Whisper computes a calculational margin from an extreme value distribution. In NCS, a margin of subcriticality (MOS) that accounts for unknowns about the analysis. Typically, this MOS is some prescribed number by institutional requirements and/or derived from expert judgment, encompassing many aspects of criticality safety. Whisper will attempt to quantify the margin from two sources of potential unknowns, errors in the software and uncertainties in nuclear data. The Whisper-derived calculational margin and MOS may be used to set a baseline upper subcritical limit (USL) for a particular AOA, and additional margin may be applied by the NCS analyst as appropriate to ensure subcriticality for a specific application in the AOA.

Whisper provides a benchmark library containing over 1,100 MCNP input files spanning a large set of fissionable isotopes, forms (metal, oxide, solution), geometries, spectral characteristics, etc. Along with the benchmark library are scripts that may be used to add new benchmarks to the set; this documentation provides instructions for doing so. If the user desires, Whisper may analyze benchmarks using a generalized linear least squares (GLLS) fitting based on nuclear data covariances and identify those of lower quality. These may, at the discretion of the NCS analyst and their institution, be excluded from the validation to prevent contamination of potentially low quality data. Whisper provides a set of recommended benchmarks to be optionally excluded.

Whisper also provides two sets of 44-group covariance data. The first set is the same data that is distributed with SCALE 6.1 in a format that Whisper can parse. The second set is an adjusted nuclear data library based upon a GLLS fitting of the benchmarks following rejection. Whisper uses the latter to quantify the effect of nuclear data uncertainties within the MOS. Whisper also has the option to perform a nuclear covariance data adjustment to produce a custom adjusted covariance library for a different set of benchmarks.

Acknowledgements: Thanks to the XCP & NCS Division Leaders at LANL for promoting and supporting the XCP3-NCS interchange sessions. Thanks to the DOE Nuclear Criticality Safety Program for its long-term support for developing advanced MCNP6 capabilities, including the iterated fission probability, adjoint-weighted tallies, sensitivity/uncertainty features, and Whisper statistical analysis. Thanks to the LANL PF4-Restart program for supporting some of the LANL-specific portions of this work, including direct support for assisting the NCS criticality safety analysts.

Whisper Methodology for Validation & USLs

- **Whisper**
  - **Statistical analysis code to determine baseline USLs**
  - **Uses sensitivity profiles from continuous-energy MCNP6**
  - **Uses covariance data for nuclear cross-sections**

- **Using Whisper**

Run MCNP6 for an Application, & get Application sensitivity profile, S_A

Run Whisper:

① Automated, physics-based selection of benchmarks that are neutronically similar to the application, ranked & weighted

- Compare Application S_A to each of the Benchmark sensitivities $S_{B(i)}$
- Select most-similar benchmarks (highest S_A-$S_{B(i)}$ correlation coefficients)

② Bias + bias uncertainty from Extreme Value Theory

- Statistical analysis - based on most-similar Benchmarks selected

③ Margin for nuclear data uncertainty estimated by GLLS method

- Use benchmark sensitivities & cross-section covariance data to estimate the MOS for nuclear data uncertainties

MCNP6 & Whisper Status

- **MCNP releases by RSICC**

  MCNP6.1 – 2013, production version

  MCNP6.1.1 – 2014, same criticality, faster, beta features for DHS

  MCNP6.2 – 2016 (Fall), with Whisper code & benchmarks

  ENDF/B-VII.1 data, updates, & older data

  Reference Collection – 700+ technical reports

  V&V Test Collection – 1434 test problems

- **Whisper-1.1.0 (2016) [original Whisper-1.0.0 (2014)]**
  - **SQA**
    - Whisper is now part of MCNP6, rigorous SQA
    - Portable to Linux, Mac, & Windows, same results
  - **Benchmark Suite**
    - 1101 ICSBEP benchmarks, with sensitivity profiles from MCNP6 for all isotopes & reactions
  - **Software**
    - Available to any DOE crit-safety group
    - Will be included with MCNP6.2 release (Fall 2016)
  - **Documentation**

    mcnp.lanl.gov → "Reference Collection" → "Whisper – NCS Validation"

Whisper-1.1.0 Update

Whisper code updates: 1.0.0 → 1.1.0

- Robust numerics, to avoid memory problems on Mac & Windows
  - Explicit threaded loops, to replace many instances of F90 matrix operators
  - Replaced Linpack coding by modern Fortran
  - Additional threading for some slow sections
  - No change to any results
- Methods
  - Chi-square & benchmark rejection changed from based on dk to dk/k. Gives some very minor diffs in list of rejected benchmarks
  - For USL, the list of benchmarks selected is sorted by weight (or Ck)
- Files
  - up to 256-character filenames
  - printed list of all files in use, full pathnames
  - TOC files permit blank lines & comment lines

    BenchmarkTOC.dat, ExcludedBenchmarks.dat
- Control
  - deprecate use of environment variables for filenames
  - use explicit command-line options instead (for whisper)
  - revised scripts handle this automatically

Whisper support updates: 1.0.0 → 1.1.0

- Build & test procedures completely revised, to be similar to mcnp6
- Previous C-shell scripts replaced by portable perl scripts

  whimcnp → whisper_mcnp.pl

  ww → whisper_usl.pl

- Mods to mcnp_pstudy.pl, to run on Windows & support Whisper scripts

Whisper files updates: 1.0.0 → 1.1.0

- Benchmarks
  - Updated 27 files (per NCS)
    - 1 significant error
    - trivial Δk changes in others
  - Added 15 new files
- Reran 42 benchmarks
  - new sensitivity profiles
  - new BenchmarkTOC.dat & ExcludedBenchmarks.dat
  - new adjusted covariance data files

Introduction (1)

Whisper? Who cares?

- Sensitivity/Uncertainty methods for validation have been under development for > 18 yrs at ORNL (Broadhead, Rearden, Perfetti, ...)
- Kiedrowski & Brown developed MCNP iterated fission probability, adjoint weighted tallies, & S/U capabilities, 2008-2013. Whisper in 2014.
- There are now 2 US calculational paths for S/U based validation:
  - SCALE/Tsunami
  - MCNP/Whisper
- International effort for comparisons being planned
  - LANL, ORNL, IRSN
- S/U based validation methods can supplement, support, & extend traditional validation methods

Introduction (2)

Traditional validation methods are 40+ years old; S/U methods are new

Should not argue for exclusive use of either traditional or S/U methods

The foundation of criticality safety includes conservatism, continuous improvement, state-of-the-art tools & data, thorough checking, …..

Traditional & S/U methods complement each other, & provide greater assurance for setting USLs

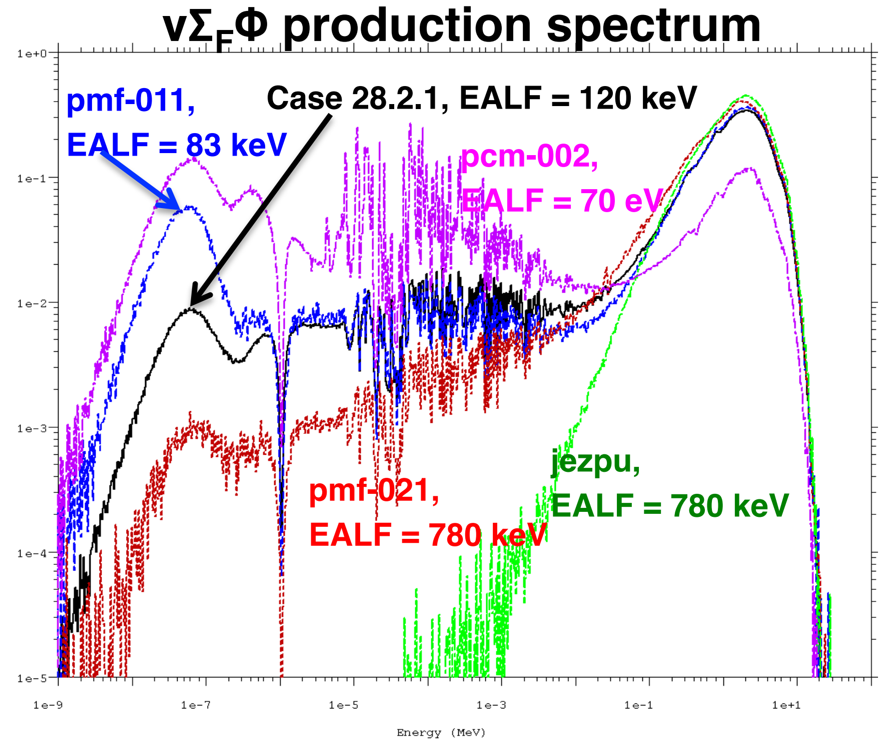
- Traditional methods provide a check on S/U methods
- S/U approach to automated benchmark selection is quantitative, physics-based, & repeatable. Provides a check on traditional selection
- Traditional methods use $MOS_{data+code}$ of 2-5%.
  Quantitative, physics-based, repeatable $MOS_{data+code}$ from S/U usually smaller

The next 5 years or so should be a transition period, where both traditional & S/U methods should be used

In today's environment of audits, reviews, & "justify everything", it is prudent to use both traditional & S/U methods for validation

Neutronics of Applications & Benchmarks

- Neutron spectra are complex functions of geometry, materials, nuclear cross-sections, etc.
- Simple metrics such as EALF or ANECF cannot capture the complexity of a fissile system
- During the past 20 years, a powerful set of tools has been developed based on sensitivity-uncertainty methods



- **Characterize the neutronics of an application or benchmark by means of sensitivity profiles, S(energy, reaction, isotope), S = (dk/k) / (dσ/σ)**
- **Fold in the uncertainties in nuclear data using covariance matrices**
- **MCNP6 determines sensitivity profiles for an application**
- **Whisper uses sensitivity profiles & data covariances to select similar benchmarks, determine bias, bias-uncertainty, & MOS**

Sensitivity Profiles for Criticality Safety

- **The sensitivity coefficient is defined as the ratio of relative change in k-effective to relative change in a system parameter:**

$$S_{k,x} = \frac{dk/k}{dx/x} = \frac{x}{k}\frac{dk}{dx}$$

- **This may be expressed using perturbation theory:**

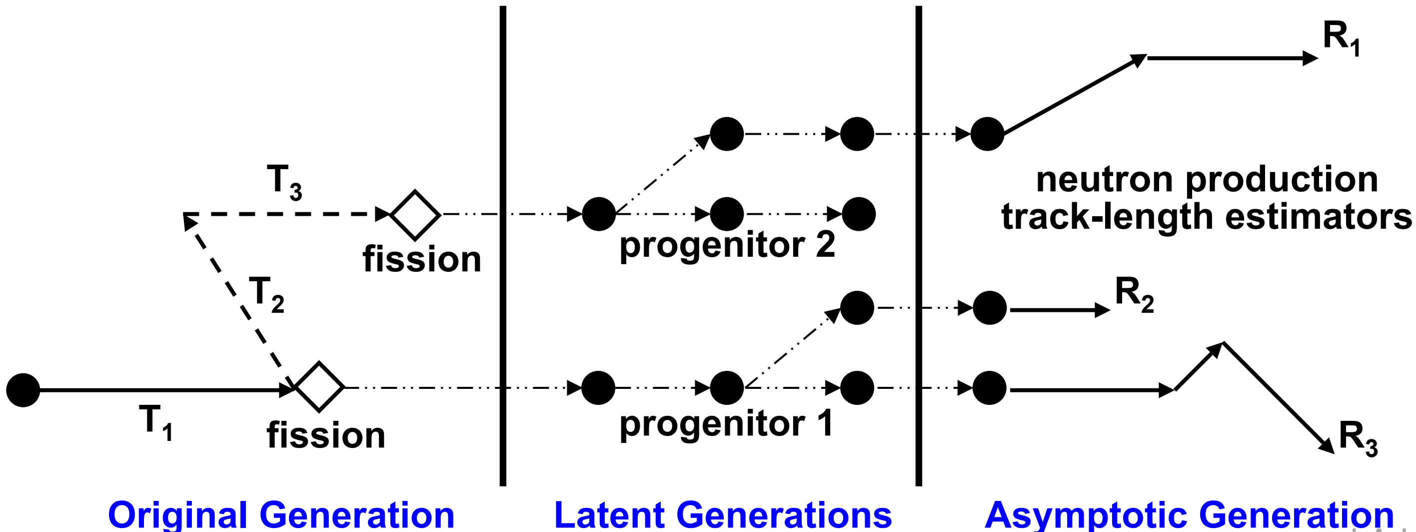
$$S_{k,x} = \frac{x}{k}\frac{dk}{dx} = -\frac{\left\langle \psi^{\dagger}, \left(\Sigma_x - \mathrm{S}_x - k^{-1}\mathrm{F}_x\right)\psi \right\rangle}{\left\langle \psi^{\dagger}, k^{-1}\mathrm{F}\psi \right\rangle}$$

  - **Includes both the forward & adjoint neutron fluxes.**
  - **S = scatter operator, F = fission operator in integral transport eq**
  - **x subscript implies that the perturbation is just for data x**

- **$S_{k,x}(E)$ is the sensitivity profile, a function of neutron energy**

Sensitivity Profiles – Adjoint Weighting

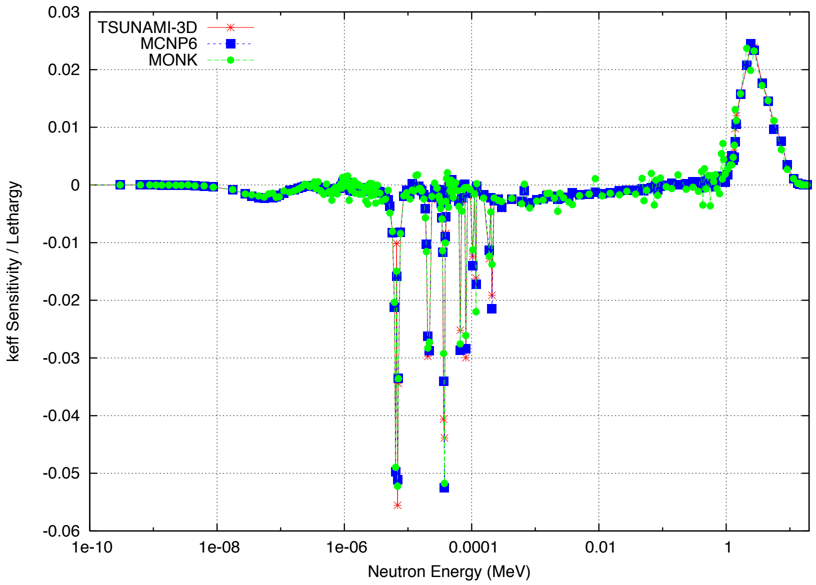
- **Using the Iterated Fission Probability method, MCNP6 can compute adjoint-weighted integrals of any quantity.**
- **MCNP breaks active cycles into consecutive *blocks*:**
  - **Tally scores are collected in original generation, & progenitor neutrons tagged**
  - **All subsequent progeny within the latent generations remember their progenitor**
  - **Importance is the population of progeny from each progenitor in the asymptotic generation**
  - **(Score)*(importance) is tallied for adjoint-weighted results**



Sensitivity Profiles - Examples

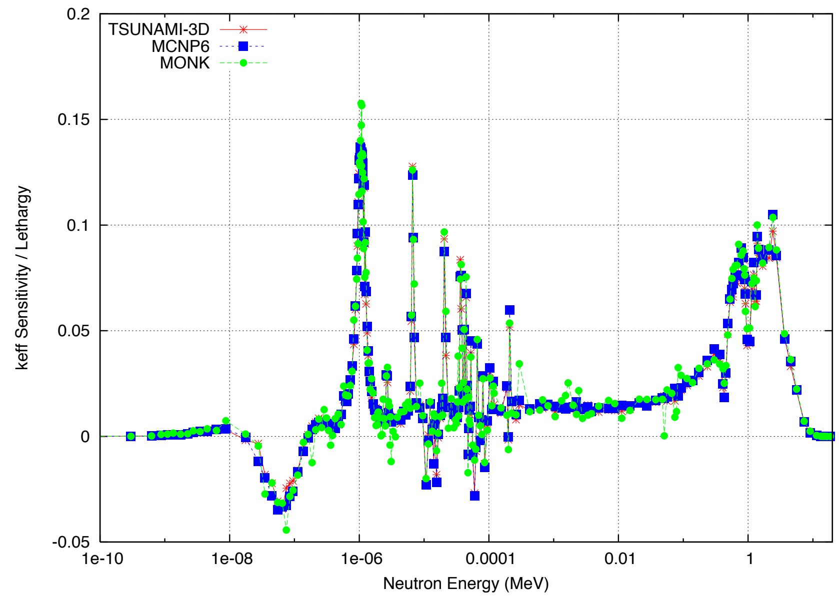
U-238: total cross-section sensitivity

OECD/NEA UACSA Benchmark Phase III.1



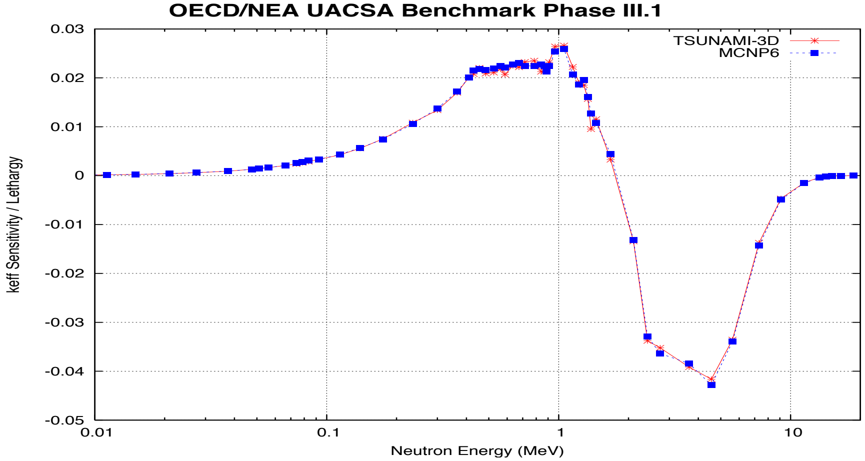
H-1: elastic scattering cross-section sensitivity

OECD/NEA UACSA Benchmark Phase III.1



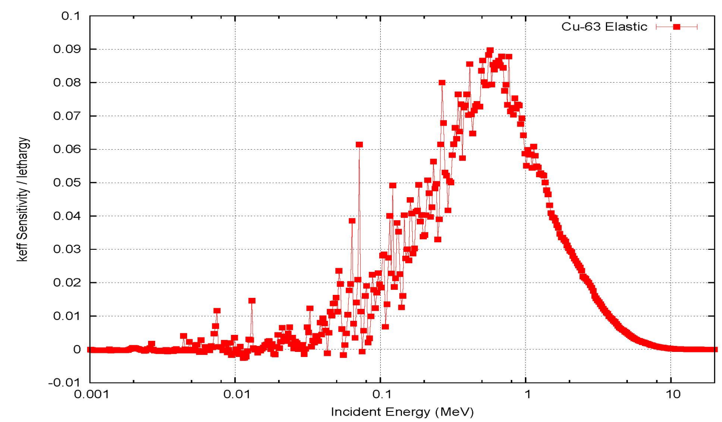
Pu-239: fission chi(E) sensitivity

OECD/NEA UACSA Benchmark Phase III.1



Cu-63: Elastic Scattering Sensitivity

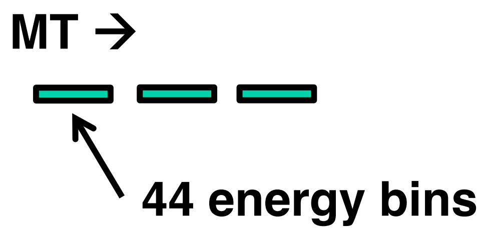
Copper-Reflected Zeus experiment:



Sensitivity Profiles (Vectors)

- **For each isotope, the sensitivity coefficients for a specific problem are stored consistent with the layout of the covariance data**
  - **Recall that the sensitivity of Keff to a particular reaction type & energy bin is:**

$$S_{k,x} = \frac{\Delta k / k}{\Delta x / x} = \frac{x}{k}\frac{dk}{dx}$$



where x is the cross-section for a particular isotope, reaction (MT), & energy bin

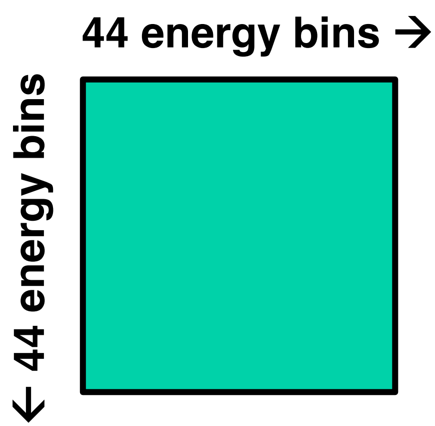
- **For a particular application problem, A, the sensitivity profiles for all isotopes are combined into one sensitivity vector S_A**

The sensitivity profile S_A(E, MT, iso) completely characterizes the neutronics of an application

size of S_A = (44 E bins) x (12 reactions) x (number of isotopes)

Cross-section Covariance Data

- For a particular isotope & particular reaction (MT), the nuclear data uncertainties are a G x G matrix, where G = number of energy groups = 44



  - Each diagonal is the variance of the cross-section for a particular energy bin
  - Off-diagonal elements are the shared variance between the data for pairs of energy bins

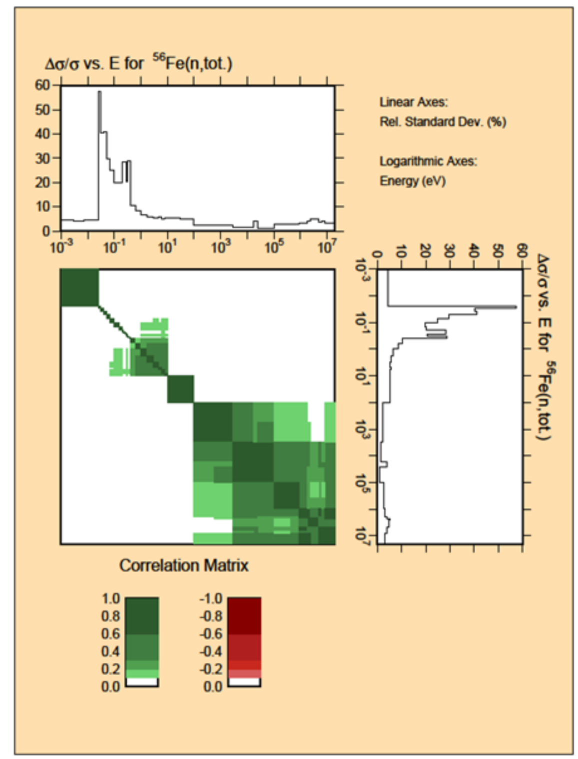


FIG. 9: A typical NJOY-generated plot of ENDF/B-VII.0 data downloaded from the National Nuclear Data Center, BNL, USA.

Cross-section Covariance Data

- **The covariance matrices for all isotopes can be combined, including off-diagonal blocks that relate uncertainties in one iso-MT-E with a different iso-MT-E**

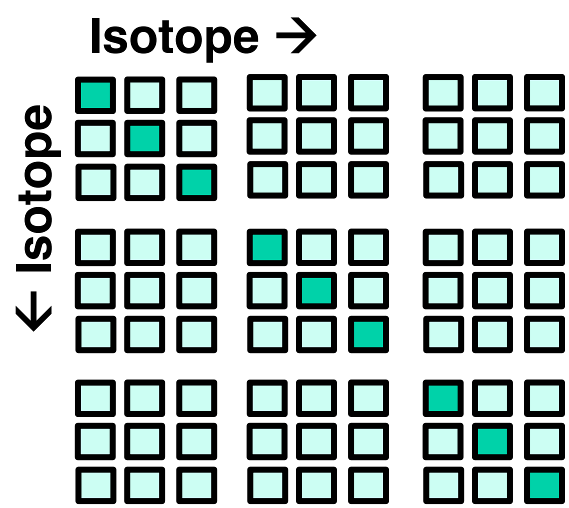
  – **Each diagonal element of C_{xx} is the variance of the cross-section for a particular isotope, MT, & energy bin**

  – **Off-diagonal elements of C_{xx} are the shared variance between pairs of Iso-MT-E & Iso'-MT'-E'**

  – **Very sparse (lots of zeros), block-structured matrix**

  (Off-diagonal I-I' blocks would generally be zero)

$C_{xx} =$



size of C_{xx} = [(44 E bins) x (12 reactions) x (number of isotopes)]^2

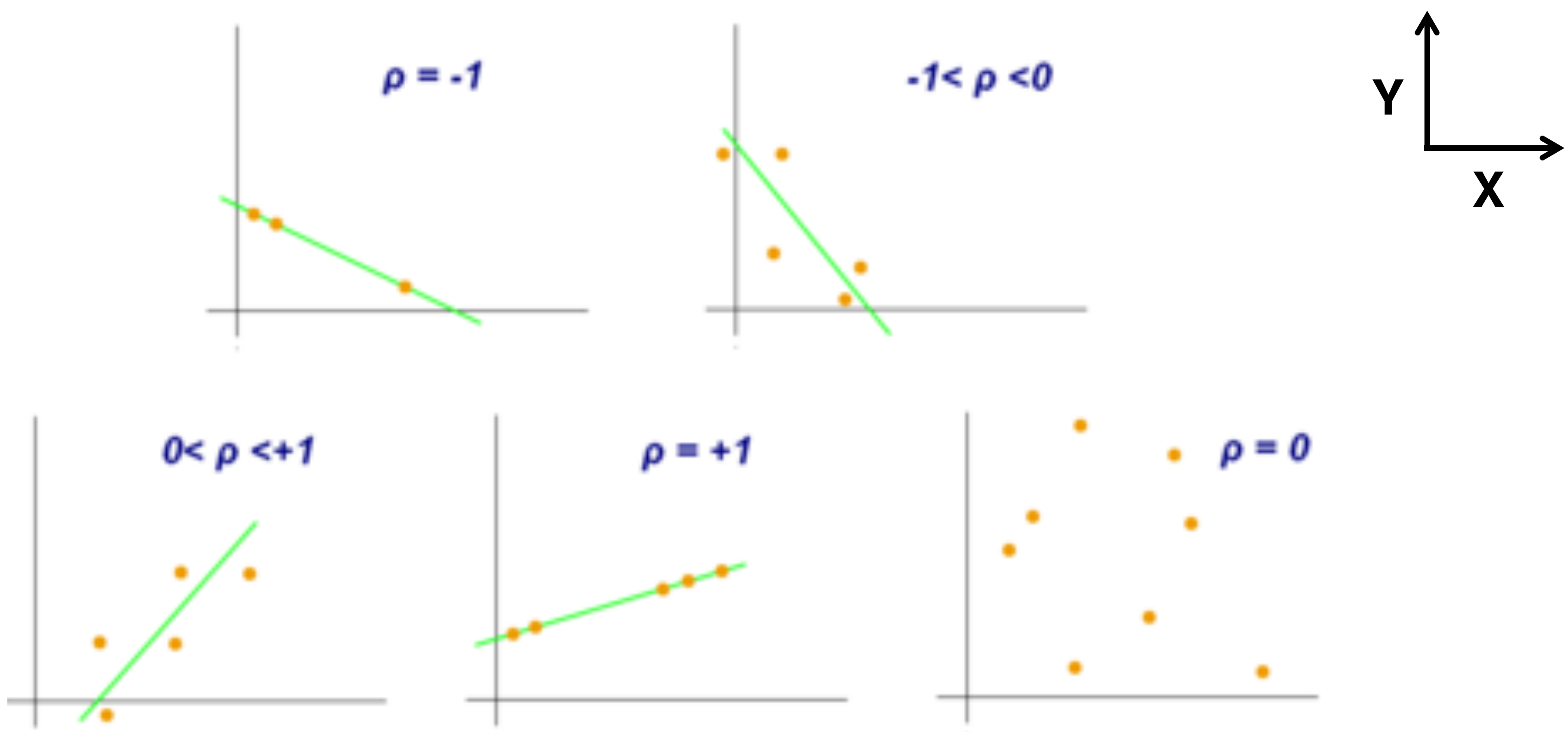
Correlation Coefficient

- **Correlation coefficient**
  - **Pearson product-moment correlation coefficient, *r* or *ρ***
  - **A measure of the linear correlation between variables X & Y**

    ρ = +1 total positive correlation

    ρ = -1 total negative correlation

    ρ = 0 no correlation



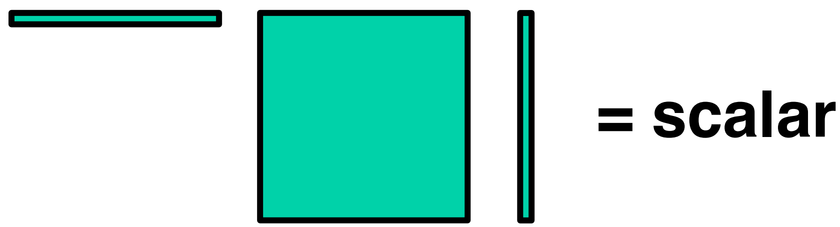
Variance in Keff & Correlation Between Problems

- **Given:** **Application A,** **Sensitivity S_A computed by MCNP**
  Benchmark B, **Sensitivity S_B computed by MCNP**

- **Variance in Keff due to nuclear data uncertainties:**

$$Var_k(A) = \vec{S}_A \bar{C}_{xx} \vec{S}_A^T$$

$$Var_k(B) = \vec{S}_B \bar{C}_{xx} \vec{S}_B^T$$



- **Covariance between A & B due to nuclear data uncertainties:**

$$Cov_k(A,B) = \vec{S}_A \bar{C}_{xx} \vec{S}_B^T$$

- **Correlation between Problems A & B due to nuclear data:**

$$c_k(A,B) = \frac{Cov_k(A,B)}{\sqrt{Var_k(A)} \cdot \sqrt{Var_k(B)}} = \frac{\vec{S}_A \bar{C}_{xx} \vec{S}_B^T}{\sqrt{\vec{S}_A \bar{C}_{xx} \vec{S}_A^T} \cdot \sqrt{\vec{S}_B \bar{C}_{xx} \vec{S}_B^T}}$$

- **Matrix-vector operations**

$$Var_k(A) = \vec{S}_A \bar{C}_{xx} \vec{S}_A^T$$

$$Cov_k(A,B) = \vec{S}_A \bar{C}_{xx} \vec{S}_B^T$$

$$c_k(A,B) = \frac{Cov_k(A,B)}{\sqrt{Var_k(A)} \cdot \sqrt{Var_k(B)}}$$

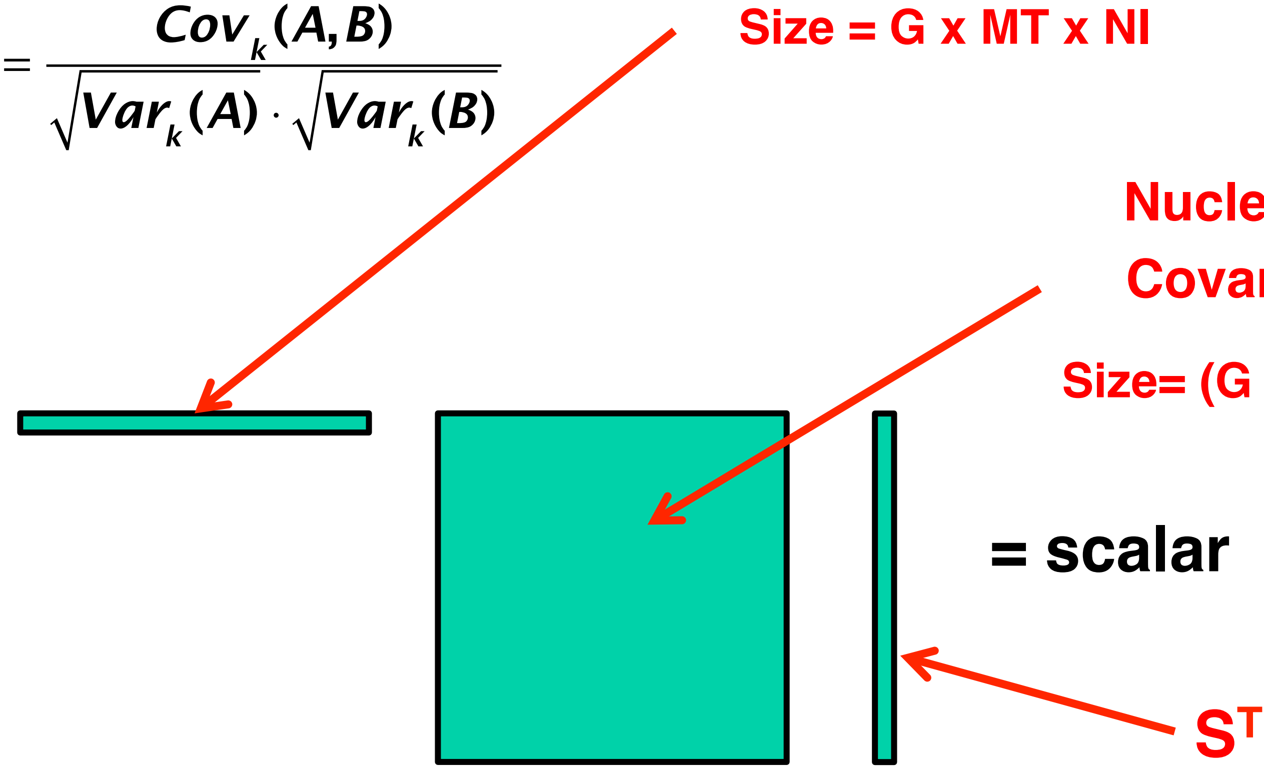
Problem-dependent sensitivity vector, S.

Based on flux spectrum, adjoint spectum, nuclear data, problem isotopes, geometry, temperature

Size = G x MT x NI

Nuclear Data Covariances

Size= (G x MT x NI)^2



Upper Subcritical Limit

- To consider a simulated system subcritical, the computed keff must be less than the Upper Subcritical Limit (USL):

$$K_{calc} + 2\sigma_{Kcalc} < USL$$

$$USL = 1 + (Bias) - (Bias\ uncertainty) - MOS$$

$$MOS = MOS_{data} + MOS_{code} + MOS_{application}$$

- The bias and bias uncertainty are at some confidence level, typically 95% or 99%.
  - These confidence intervals may be derived from a normal distribution, but the normality of the bias data must be justified.
  - Alternatively, the confidence intervals can be set using non-parametric methods.

Validation

To determine USL for applications

- **Run MCNP6 for applications**
  – Traditional: k_{eff} only
  – S/U-based: k_{eff} & sensitivity profiles

- **Select benchmarks similar to applications**
  – Traditional: Expert judgment
  – S/U-based: Select benchmarks with highest C_k's

- **Statistical analysis**
  – Standard statistical methods, determine bias & bias-uncertainty using the set of selected benchmarks

- **Determine appropriate $MOS_{data,code,applicability}$**
  – Traditional: Expert judgment, usually 2% or 5%, more if warranted
  – S/U-based: Use GLLS to estimate MOS_{data}, code-expert for MOS_{code}

- **Determine USL**
  – Is $k_{application} + 2\sigma < USL$?

Whisper Methodology

- **MCNP6**
  - **Determine Sensitivity Profiles for Benchmarks B_1 ... B_N [setup, not user]**
  - **Determine Sensitivity Profiles for Application A**

- **Whisper**
  - **Determine Benchmark c_k's**
    - For each benchmark B_J, determine $c_k^{(J)}$ correlation coefficient between A & B_J
  - **Determine Benchmark Weights & Select Benchmarks**
    - Iterative procedure using $c_k^{(J)}$ values, $c_{k,max}$, $c_{k,acc}$
  - **Determine Calculational Margin (CM)**
    - Extreme Value Theory, with weighted data, nonparametric
    - Compute bias & bias uncertainty
    - Adjustment for non-conservative bias
    - Handling small sample sizes
  - **Determine portions of MOS**
    - Margin for nuclear data uncertainties
    - Margin for unknown code errors

Using Whisper for Validation

- **As part of Whisper installation (not day-to-day use),**
  - **For each of the ~1100 benchmarks**
    - MCNP6 is run to generate the sensitivity vector S_B for that benchmark
    - The sensitivity vector S_B for each benchmark is saved in a folder
  - **The nuclear data covariance files are saved in a folder**
  - **Benchmarks are checked for consistency, some may be rejected**
  - **Missing uncertainties for some benchmarks are estimated**
  - **All of this is the responsibility of the Admin person & needs to be done only once at installation (or repeated if the code, data, or computer change)**

- **To use Whisper for validation:**

  ① Use the whisper_mcnp script to make 1 run with MCNP6 for a particular application, to generate the sensitivity vector for the application, S_A

  ② Run Whisper, using the whisper_usl script

Whisper-1.1.0 – Batch Job

To try it, on Moonlight HPC front end:

- **Make a directory, copy MCNP6 input files to it**
  - No blanks in pathname, directory name, input file names
  - Put mcnp6 input files in the directory

```
bash:       mkdir   WTEST
bash:       cp      some-dir/myjob.i    WTEST
```

- **Set up batch job file, job.txt**

```
#!/bin/bash
#PBS  -V
#PBS  -l nodes=1:ppn=16,walltime=01:00:00
export  WHISPER_PATH="/usr/projects/mcnp/ncs/WHISPER"
export          PATH="$WHISPER_PATH/bin:$PATH"

cd WTEST

whisper_mcnp.pl  -local    myjob.i
whisper_usl.pl
```

- **Submit batch job file**

```
msub   job.txt
```

Whisper-1.1.0 – Interactive

To try it, on Moonlight HPC:

- Set & export WHISPER_PATH environment variable
  - bash:
```
export   WHISPER_PATH="/usr/projects/mcnp/ncs/WHISPER"
export           PATH="$WHISPER_PATH/bin:$PATH"
```
  - csh, tcsh:
```
setenv   WHISPER_PATH "/usr/projects/mcnp/ncs/WHISPER"
setenv           PATH "$WHISPER_PATH/bin:$PATH"
```

- Make a directory, copy MCNP6 input files to it
  - No blanks in pathname, directory name, input file names
  - Put mcnp6 input files in the directory

```
bash:       mkdir   WTEST
bash:       cp      some-dir/myjob.i   WTEST
bash:       ls      WTEST
mjob.i
bash:
```

Using whisper_mcnp (1)

- **From the front-end on an HPC system:**

```
whisper_mcnp.pl   myjob.i
```

- **myjob.i is an MCNP6 input file**
  - Must NOT include any of these cards: `kopts, ksen, prdmp`
  - May list more than 1 input file on whisper_mcnp command line
  - Lots of options, see next 2 slides

- **Creates files & dirs:**

  MCNPInputList.toc
  Calcs/
  Calcs/myjob.i ← **modified to include kopts, ksen, prdmp, & new kcode**
  KeffSenLib/

- **Submits jobs to HPC compute nodes**
  - **Single-node jobs, 16 threads each**
  - **Default time limit of 1 hr**

Using whisper_mcnp (2)

- **For each MCNP6 input file listed on the whisper_mcnp command line:**
  - **KCODE line is deleted & these lines are inserted:**

```
kcode     100000   1.0    100    600
kopts blocksize = 5
ksen1 xs
     rxn = +2 +4 -6 +16 102 103 104 105 106 107 -7 -1018
     erg = 1.0000e-11 3.0000e-09 7.5000e-09 1.0000e-08 2.5300e-08 3.0000e-08
           4.0000e-08 5.0000e-08 7.0000e-08 1.0000e-07 1.5000e-07 2.0000e-07
           2.2500e-07 2.5000e-07 2.7500e-07 3.2500e-07 3.5000e-07 3.7500e-07
           4.0000e-07 6.2500e-07 1.0000e-06 1.7700e-06 3.0000e-06 4.7500e-06
           6.0000e-06 8.1000e-06 1.0000e-05 3.0000e-05 1.0000e-04 5.5000e-04
           3.0000e-03 1.7000e-02 2.5000e-02 1.0000e-01 4.0000e-01 9.0000e-01
           1.4000e+00 1.8500e+00 2.3540e+00 2.4790e+00 3.0000e+00 4.8000e+00
           6.4340e+00 8.1873e+00 2.0000e+01
prdmp j 9999999
```

  - **Note that there are large numbers of neutrons/cycle & cycles for the KCODE input. While it may be tempting to reduce these to get shorter runs, that is discouraged since it is important to achieve reasonable statistical uncertainties on the sensitivity profiles for a large number of reactions, isotopes, & energies.**

- **After using whisper_mcnp, after the MCNP6 jobs complete:**
  - **The Calcs/ directory will contain these files**

myjob.i	modified MCNP6 input file, with kcode, ksen, kopts, prdmp
myjob.io	output file from MCNP6 jobs
myjob.ir	runtpe file
myjob.is	srctp file

whisper_mcnp.pl - Usage

mcnp

whisper_mcnp.pl [Options] Filelist

```
Options:
        -help              print this information
        -local             run MCNP jobs locally, on this computer
        -submit            submit batch MCNP jobs, using msub    [default]
        -walltime x        walltime limit for submitted batch jobs (eg, 01:00:00)
        -mcnp x            pathname for MCNP6 executable
        -xsdir x           pathname for MCNP6 xsdir file
        -data x            pathname for MCNP6 data, DATAPATH
        -threads x         number of threads for MCNP6
        -neutrons x        number of neutrons/cycle for MCNP6
        -discard x         number of inactive cycles for MCNP6
        -cycles x          total number of cycles for MCNP6
Filelist:
        Names of MCNP6 input files. The names should not contain blanks.
        The files must include a KCODE card (that will be replaced), &
        must not contain KSENn, KOPTS, or PRDMP cards (they will be supplied)

Defaults:                  **for local**              **for submit**
        -submit
        -mcnp              hardwired in script        /usr/projects/mcnp/mcnpexe  -6
        -xsdir             hardwired in script     /usr/projects/mcnp/MCNP_DATA/xsdir_mcnp6.1
        -data              hardwired in script        /usr/projects/mcnp/MCNP_DATA
        -walltime                                     01:00:00
        -threads           12                         16
        -neutrons          10000                      100000
        -discard           100                        100
        -cycles            600                        600
```

```
/usr/projects/ncs/MCNP/bin/mcnp6
/usr/projects/ncs/Data/xsdir_mcnp6.1
/usr/projects/ncs/Data
```

Using whisper_mcnp (4)

- **Use whisper_mcnp.pl to run mcnp6 & get sensitivity profiles**

```
bash:   cd   WTEST
bash:   whisper_mcnp.pl    myjob.i
```

Screen output:

```
*****************
*               *
*  whisper_mcnp  *      a utility script to set up input & run MCNP for Whisper
*               *
*****************

        Input File TOC           = MCNPInputList.toc
        Calculation directory    = Calcs
        Sensitivity directory    = KeffSenLib

        Neutrons/cycle           = 100000
        Cycles to discard        = 100
        Total Cycles to run      = 600

        MCNP6 executable         = /usr/projects/mcnp/mcnpexe -6
        XSDIR file               = /usr/projects/mcnp/MCNP_DATA/xsdir_mcnp6.1
        DATAPATH                 = /usr/projects/mcnp/MCNP_DATA
        Threads                  = 16
        Wall-clock time for job = 01:00:00

        All jobs will be submitted using moab


        ...process  mcnp input file:  myjob.i
        ...modified mcnp input file:  Calcs/myjob.i

        ...submit mcnp job to cluster using moab:  myjob.i
```

Using whisper_mcnp (5)

- **After running whisper_mcnp in directory WTEST:**

```
whisper_mcnp.pl     myjob.i
```

Use moab commands to check job status: showq –u username

When the submitted job is complete:

Files created by whisper_mcnp & mcnp6:

```
WTEST/
    myjob.i                ← original
    MCNPInputlist.toc
    Calcs/
        myjob.i  myjob.io  myjob.ir  myjob.is
    KeffSenLib/
```

Using whisper_usl (1)

- **From the front-end or compute node on an HPC system, run Whisper using the whisper_usl script:**

```
cd   WTEST
whisper_usl.pl
```

  - Can optionally include ExcludeFile.dat, list of benchmark files to exclude from Whisper calculations
  - Runs Whisper for application(s) myjob.i (etc)

- **For each input file listed in `MCNPInputList.toc`:**
  - Extract sensitivity profiles from `Calcs/myjob.io`, place into directory `KeffSenLib/`
  - Create (or add to) file `KeffSenList.toc`
  - Run Whisper using the sensitivity profiles for the application (myjob.i) and the collection of Whisper benchmark sensitivity profiles
  - Output to screen & file `Whisper.out`

Using whisper_usl (2)

- **After running whisper_mcnp & whisper_usl:**

```
whisper_mcnp.pl   myjob.i
..... [wait for submitted mcnp6 job to complete]
whisper_usl.pl
```

Files created by whisper_mcnp, mcnp6, & whisper_usl:

```
myjob.i                ← original
MCNPInputlist.toc
Calcs/
      myjob.i  myjob.io  myjob.ir  myjob.is
KeffSenList.toc
KeffSenLib/
      myjob.ik
Whisper.out
```

whisper_usl.pl (3)

```
bash: whisper_usl.pl

*****************
*               *
*  whisper_usl  *      set up & run Whisper validation calculations
*               *
*****************

   =====> setup files for whisper

        ---> setup for problem myjob.i
                ...extract sensitivity profile data from:  Calcs/myjob.io
                ...copy    sensitivity profile data to:    KeffSenLib/myjob.ik
                ...extract calc Keff & Kstd    data from:  Calcs/myjob.io
                ...  KeffCalc= 0.96740 +- 0.00057,  ANECF= 1.4904E+00 MeV,  EALF= 1.2150E-01 MeV

   =====> run whisper

/Users/fbrown/CODES/WHISPER/WHISPER.git/bin/whisper -a KeffSenList.toc -ap KeffSenLib
 whisper-1.1.0                  2016-02-02   (Copyright 2016 LANL)
 WHISPER_PATH                 = /Users/fbrown/CODES/WHISPER
 Benchmark TOC File           = /Users/fbrown/CODES/WHISPER/Benchmarks/TOC/BenchmarkTOC.dat
 Benchmark Sensitivity Path   = /Users/fbrown/CODES/WHISPER/Benchmarks/Sensitivities
 Benchmark Correlation File   =
 Benchmark Exclusion File     =
 Benchmark Rejection File     =
 Covariance Data Path         = /Users/fbrown/CODES/WHISPER/CovarianceData/SCALE6.1
 Covariance Adjusted Data Path =
 Application TOC File         = KeffSenList.toc
 Application Sensitivity Path  = KeffSenLib/
 User Options File            =
 Output File                  = Whisper.out
```

whisper_usl.pl (4)

```
........

 Reading benchmark data ...
 Reading application data ...
 Reading covariance data ...
 Reading adjusted covariance data ...
 Calculating application nuclear data uncertainties ...
 Calculating upper subcritical limits ...
......case     1  Ck=  0.41263
......case     4  Ck=  0.36554     ← all Ck's printed in Whisper.out,
......case     3  Ck=  0.63497       only a few printed to the screen

        ........

......case   246  Ck=  0.18901
                                            calc        data unc     baseline    k(calc)
   application                              margin      (1-sigma)    USL         > USL

   myjob.i                                  0.01329     0.00120      0.97860    -0.00972
```

Whisper.out (1)

```
whisper-1.1.0                  2016-02-02   (Copyright 2016 LANL)
WHISPER_PATH                 = /Users/fbrown/CODES/WHISPER
Benchmark TOC File           = /Users/fbrown/CODES/WHISPER/Benchmarks/TOC/BenchmarkTOC.dat
Benchmark Sensitivity Path   = /Users/fbrown/CODES/WHISPER/Benchmarks/Sensitivities
Benchmark Correlation File   =
Benchmark Exclusion File     =
Benchmark Rejection File     =
Covariance Data Path         = /Users/fbrown/CODES/WHISPER/CovarianceData/SCALE6.1
Covariance Adjusted Data Path =
Application TOC File         = KeffSenList.toc
Application Sensitivity Path = KeffSenLib/
User Options File            =
Output File                  = Whisper.out

Reading benchmark data ...
                       benchmark    k(bench)    unc        k(calc)     unc        bias        unc
                         myjob.i    1.00000     0.01100    1.01174     0.00007    -0.01174    0.01100

                   ..........

  246 benchmarks read,      0 benchmarks excluded.

Reading application data ...
                               application     k(calc)     unc
                                   myjob.i     0.96802     0.00052

Reading covariance data ...
 Reading covariance data for 1001 ...

                   ..........
Reading adjusted covariance data ...
 Reading covariance data for 1001 ...
```

Whisper.out (2)

mcnp

```
Calculating application nuclear data uncertainties ...
    application                                  adjusted    prior
    myjob.i                                      0.00209     0.01221

 Calculating upper subcritical limits ...
                                                 calc        data unc    baseline    k(calc)
    application                                  margin      (1-sigma)   USL          > USL
    myjob.i                                      0.01334     0.00209     0.97623     -0.00686

    Benchmark population    =    48
    Population weight       =   28.56732
    Maximum similarity      =    0.96434

    Bias                    =    0.00850
    Bias uncertainty        =    0.00484
    Nuc Data uncert margin  =    0.00209
    Software/method margin  =    0.00500
    Non-coverage penalty    =    0.00000

    benchmark                                     ck          weight
    pu-met-fast-011-001.i                         0.9643      1.0000
    pu-met-fast-044-002.i                         0.9641      0.9958
    pu-met-fast-021-002.i                         0.9618      0.9545
    pu-met-fast-003-103.i                         0.9602      0.9252
    pu-met-fast-026-001.i                         0.9594      0.9099
    pu-met-fast-025-001.i                         0.9584      0.8912
    pu-met-fast-032-001.i                         0.9572      0.8699
    pu-met-fast-016-001.i                         0.9546      0.8221
    pu-met-fast-027-001.i                         0.9546      0.8217
                      ........
    pu-met-fast-012-001.i                         0.9167      0.1283
    pu-met-fast-040-001.i                         0.9166      0.1269
    pu-met-fast-045-003.i                         0.9163      0.1209
    pu-met-fast-045-004.i                         0.9147      0.0909
    pu-met-fast-002-001.i                         0.9145      0.0874
```

For this application, 48 of the 1101 benchmarks were selected as neutronically similar & sufficient for valid statistical analysis

Benchmark rankings shown below

Conclusions & Discussion

The sensitivity-uncertainty-based tools provided by MCNP/Whisper & SCALE/Tsunami are relatively new. They should be used with caution, and results should be critically reviewed.

One particular strength of the S/U-based tools is the selection of the most appropriate benchmarks to use for an application. The S/U-based tools provide quantitative, physics-based results for identifying which benchmarks are most similar to an application.

Another unique strength of the S/U-based tools is the use of GLLS methods to provide a quantitative, physics-based estimate of the MOS_{data} due to nuclear data uncertainties. For applications where the traditional 2-5% MOS is too limiting, the S/U-based tools may provide quantitative evidence for a reduced MOS. Caution and judgment are required.

In the near-term, S/U-based methods provide powerful tools for supporting, complementing, and extending traditional validation methods. It is expected that the use of S/U-based tools will expand as more experience & knowledge is acquired.

Nuclear Criticality Safety

-

Validation - III

-

Using Whisper

Examples for NCS Analysts

Examples using Whisper

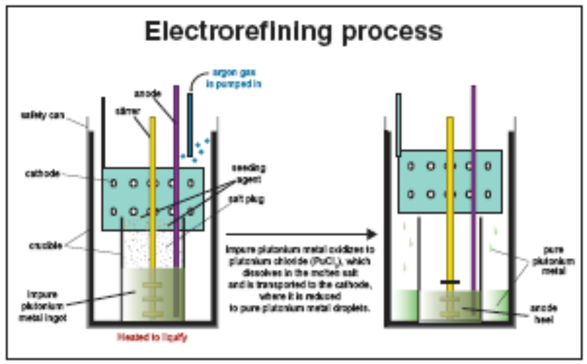
- **Pyrochemical Processing**
  - **Example 1: Typical computational model: ingot**
  - **Example 2: Geometry: Annular**
  - **Example 3: Material: Pu-NaCl**
  - **Example 4: Reflection: Ta**
  - **Example 5: Moderation: Oil**

- **General Studies**
  - **Example 6: “Revisiting a Practical Application of the Single-Parameter-Subcritical-Mass Limit for Plutonium Metal with Whisper”**
  - **Example 7: Critical-mass curves and USL-mass curves comparison**

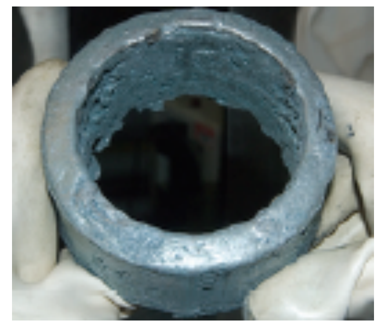
Pyrochemical Processing

- **Electrorefining is a batch plutonium metal purification process**
  - **Feed: impure plutonium metal ingot, up to 4,500 g Pu**
  - **Product: ER ring**
  - **Waste: salt, anode heel, crucible**

→

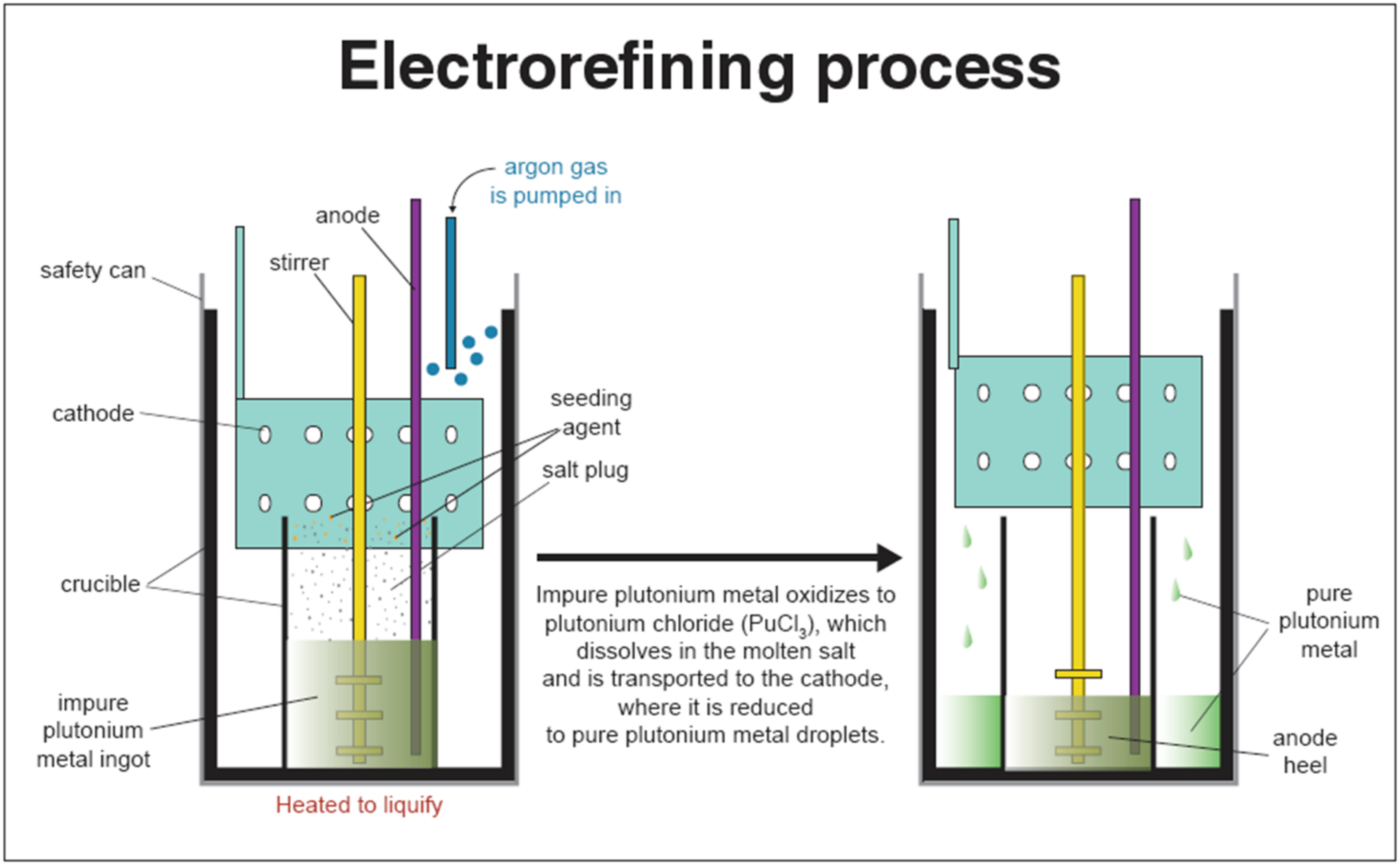


→



Ref. Actinide Research Quarterly 3rd Quarter 2008

- **Purification media is an equimolar NaCl/KCl molten salt at 740°C**
  - **A small amount of plutonium chloride seed to charge the electrolyte with Pu(III).**
- **Liquid plutonium oxidizes at the anode (ingot) into the electrolyte**
- **Pu(III) ion in transported through the electrolyte to the cathode**
- **Reduced to metal dripping into the outer cup**



Ref. Actinide
Research Quarterly
3rd Quarter 2008

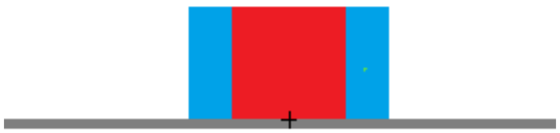
Example 1

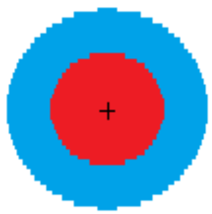
–

4.5 kg Pu Ingot, varying H/D

Example 1 - wval1: 4.5 kg Pu Ingot, varying H/D   (1)

- **4.5 kg Pu-239 right-circular cylinder**
- **Pu density = 19.86 g/cm^3**
- **Reflected radially with 1 inch of water**
- **Reflected on the bottom with ¼ inch steel**





- **Vary the height-to-diameter (H/D) over the range 0.5 – 3.0**
  - **Start with wval1.txt, input for H/D = 1**

    mcnp6   i=wval1.txt

  - **Copy wval1.txt to wval1p.txt, then insert directives for mcnp_pstudy**
    - Define list for HD:

      ```
      c @@@ HD =  0.5  1.0  1.5  2.0  2.5  3.0
      ```

    - For a given H/D, compute Pu radius, then other dimensions

      $$V = (\text{Pu mass})/(\text{Pu density})$$
      $$V = H\pi R^2 = (\text{H/D}) \cdot 2\pi R^3$$
      $$R = \left[V/2\pi(\text{H/D})\right]^{1/3}$$

    - Use parameters for dimensions & location of KSRC point

Example 1 - wval1p: 4.5 kg Pu Ingot, varying H/D (2)

```
wval1:  4500 g Pu metal,  H/D = 1
c reflected 1 inch water radially,
c 0.25 in steel bottom
c
 1 1 -19.860000  -1            imp:n=1
11 3 -1.0        +1 -11        imp:n=1
14 6 -7.92       -30           imp:n=1
15 0             +11 +30 -20   imp:n=1
20 0 +20                       imp:n=0

 1  rcc  0 0 0       0 0 6.607662 3.303831
11  rcc  0 0 0       0 0 6.607662 5.843831
20  rcc  0 0 -2.54   0 0 91.44     91.44
30  rcc  0 0 -0.635  0 0 0.635     76.20

 kcode  10000 1.0 50 250
 ksrc   0 0 3.303831
 m1   94239.80c 1
 m3    1001.80c 0.66667    8016.80c 0.33333
 mt3   lwtr.20t
 m6   24050.80c 0.000757334
      24052.80c 0.014604423
      24053.80c 0.001656024
      24054.80c 0.000412220
      26054.80c 0.003469592
      26056.80c 0.054465174
      26057.80c 0.001257838
      26058.80c 0.000167395
      25055.80c 0.00174
      28058.80c 0.005255537
      28060.80c 0.002024423
      28061.80c 0.000088000
      28062.80c 0.000280583
      28064.80c 0.000071456
 prdmp 9e9 9e9 1 9e9
```

```
wval1p:  4500 g Pu metal, various H/D
c reflected 1 inch water radially,
c 0.25 in steel bottom
c
c   V = H pi R**2 = (H/D) 2pi R**3
c   R = (V/(2pi H/D)**1/3
c
c @@@  PI     = 3.141592654
c @@@  VOL_PU = ( 4500. / 19.86 )
c @@@  HD     = 0.5  1.0  1.5  2.0  2.5  3.0
c @@@  R_PU   = ( (VOL_PU/(2*PI*HD))**(1/3) )
c @@@  H_PU   = ( 2*R_PU*HD )
c @@@  R_H2O  = ( R_PU + 2.54 )
c @@@  KSRC_Z = ( H_PU * 0.5 )
c
c Pu cylinder:
c      mass        = 4500 g
c      density     = 19.86 g/cc
c      volume      = VOL_PU
c      radius Pu   = R_PU
c      height Pu   = H_PU
c      H/D         = HD
c
c H2O  outer radius = R_H2O
c
  1   1 -19.860000   -1            imp:n=1
 11   3 -1.0         +1 -11        imp:n=1
 14   6 -7.92        -30           imp:n=1
 15   0              +11 +30 -20   imp:n=1
 20   0              +20           imp:n=0

  1   rcc  0 0 0          0 0 H_PU   R_PU
 11   rcc  0 0 0          0 0 H_PU   R_H2O
 20   rcc  0 0 -2.540000  0 0 91.44  91.44
 30   rcc  0 0 -0.635000  0 0 0.635  76.20

 kcode  10000 1.0 50 250
 ksrc   0. 0. KSRC_Z
c
.................... etc.
```

Example 1 - wval1: 4.5 kg Pu Ingot, varying H/D (3)

- **Parameter study using mcnp_pstudy, whisper_mcnp, & whisper_usl:**

```
mcnp_pstudy  -i  wval1p.txt  -whisper
```

use mcnp_pstudy to create inp files
  inp_case001, inp_case002, … inp_case_006

```
whisper_mcnp.pl  -neutrons 10000  -discard 50  \
                  -cycles 250  -threads 4  \
                  inp_case*
```

use whisper_mcnp to run mcnp6 for each case &
  produce k_{eff} & sensitivity profile tallies
items in green are for class demo, so that cases run quickly,
  & should not be used for serious work

```
whisper_usl.pl
```

use whisper_usl to run Whisper & determine USL for each case

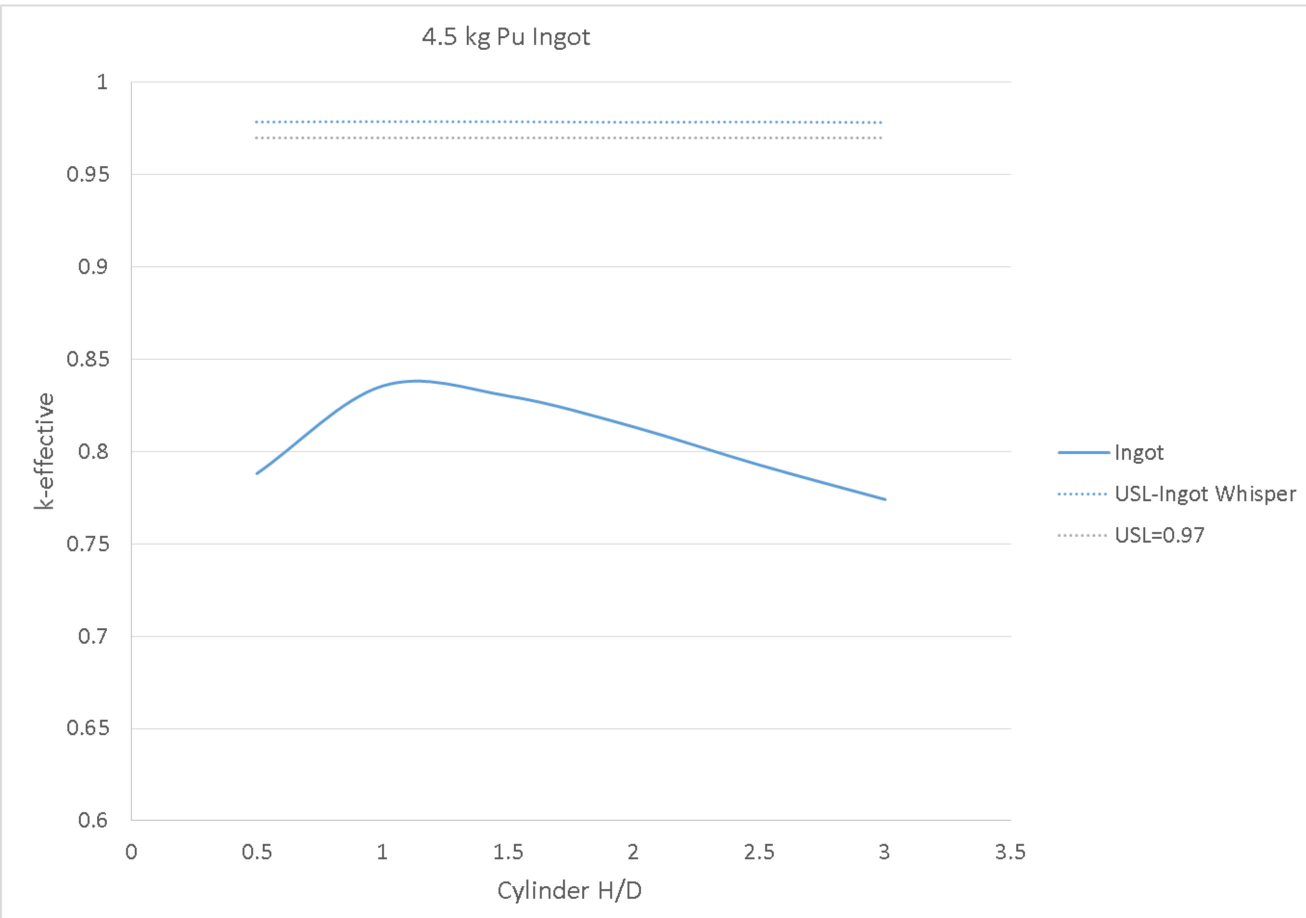
Example 1 - wval1: 4.5 kg Pu Ingot, varying H/D (4)

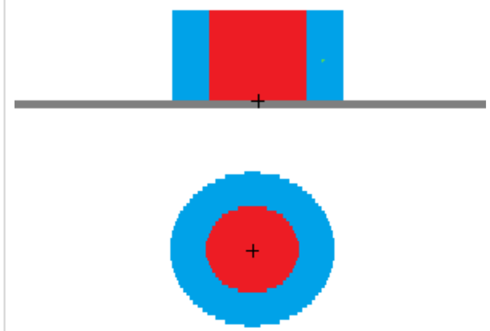
wval1, H/D = 1
mcnp6 i=wval1.txt

k = 0.83491 (41)

wval1p, varying H/D
mcnp_pstudy -i wval1p.txt -setup -run

```
HD=0.5   case001  KEFF     7.87229E-01      KSIG     4.09191E-04
HD=1.0   case002  KEFF     8.34430E-01      KSIG     4.20175E-04
HD=1.5   case003  KEFF     8.29652E-01      KSIG     4.19130E-04
HD=2.0   case004  KEFF     8.11958E-01      KSIG     4.18723E-04
HD=2.5   case005  KEFF     7.93676E-01      KSIG     4.63720E-04
HD=3.0   case006  KEFF     7.73434E-01      KSIG     4.19664E-04
```





Example 1 - wval1: 4.5 kg Pu Ingot, varying H/D (5)

MCNP6-Whisper Results

application	calc margin	data unc (1-sigma)	baseline USL	k(calc) > USL
ingot.txt_1_in	0.01441	0.00076	0.97862	-0.14366

```
Benchmark population   =   44
Population weight      =  25.38028
Maximum similarity     =   0.99621

Bias                   =   0.00858
Bias uncertainty       =   0.00583
Nuc Data uncert margin =   0.00076
Software/method margin =   0.00500
Non-coverage penalty   =   0.00000
```

benchmark	ck	weight
pu-met-fast-036-001.i	0.9962	1.0000
pu-met-fast-022-001.i	0.9957	0.9850
pu-met-fast-024-001.i	0.9956	0.9813
pu-met-fast-001-001.i	0.9940	0.9319
pu-met-fast-023-001.i	0.9937	0.9207
pu-met-fast-039-001.i	0.9932	0.9069
mix-met-fast-009-001.i	0.9923	0.8774
pu-met-fast-044-005.i	0.9917	0.8598
pu-met-fast-035-001.i	0.9913	0.8449
pu-met-fast-025-001.i	0.9902	0.8117
pu-met-fast-009-001.i	0.9898	0.7976
pu-met-fast-044-003.i	0.9896	0.7926
pu-met-fast-044-004.i	0.9894	0.7867
pu-met-fast-044-002.i	0.9887	0.7646
pu-met-fast-029-001.i	0.9867	0.7006
pu-met-fast-021-002.i	0.9865	0.6966
pu-met-fast-011-001.i	0.9848	0.6430
pu-met-fast-030-001.i	0.9845	0.6328
pu-met-fast-031-001.i	0.9844	0.6284
pu-met-fast-042-004.i	0.9823	0.5620
pu-met-fast-042-006.i	0.9820	0.5543
pu-met-fast-021-001.i	0.9815	0.5387
pu-met-fast-042-003.i	0.9813	0.5304
pu-met-fast-042-007.i	0.9812	0.5301
pu-met-fast-042-005.i	0.9809	0.5189
pu-met-fast-042-009.i	0.9808	0.5153
pu-met-fast-042-008.i	0.9807	0.5119
pu-met-fast-042-010.i	0.9802	0.4971
pu-met-fast-042-012.i	0.9802	0.4959
pu-met-fast-042-011.i	0.9800	0.4908
pu-met-fast-042-002.i	0.9799	0.4873
pu-met-fast-042-015.i	0.9795	0.4759
pu-met-fast-042-013.i	0.9794	0.4707
pu-met-fast-042-014.i	0.9793	0.4690
pu-met-fast-027-001.i	0.9752	0.3389
pu-met-fast-042-001.i	0.9748	0.3267
pu-met-fast-044-001.i	0.9743	0.3134
pu-met-fast-018-001.i	0.9741	0.3057
mix-met-fast-007-022.i	0.9733	0.2819
pu-met-fast-003-103.i	0.9714	0.2215
mix-met-fast-007-023.i	0.9709	0.2041
mix-met-fast-001-001.i	0.9675	0.0979
pu-met-fast-045-005.i	0.9668	0.0777
pu-met-fast-032-001.i	0.9644	0.0015

Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 - AoA

Example 2

–

4.5 kg Pu Annulus, varying H & R_{in}

Example 2: 4.5 kg Pu Annulus, varying H & R_{in} (1)

mcnp

- **Establishing Subcriticality - mass subcritical limits given in Table 3 apply to a single piece having no concave surfaces. Why? Can you use SPSL for a ring with concave surfaces?**
  - **If computational modeling, is a ring a validated geometry?**

From a typical traditional validation report

Parameter	Area of Applicability
Fissile Material	^{239}Pu
Fissile Material Form	Pu Metal, PuO_2, and $Pu(NO_3)_4$
H/^{239}Pu	$0 \le H/^{239}Pu \le 2807$
Average Neutron Energy Causing Fission (MeV)	$0.003 \le ANECF \le 1.935$
^{240}Pu	0 to 42.9 wt% ^{240}Pu
Moderating Materials	none, water, graphite, polystyrene
Reflecting Materials	none, water, steel, oil, Plexiglas, polyethylene, graphite, W, Cu, U, Th, Al, Ni, Fe, Pb, Cd, Mo, Be, BeO
Other Materials	concrete, PVC, Ga, B, Gd, Ta
Geometry	cylinder array, cylinder, slab, sphere, hemisphere, stacked discs, cuboid, annular

  - **How can this be established, what benchmarks include this geometry? Are these the benchmarks that are relevant (similar) to the ring?**

Benchmark	^{240}Pu wt%	Form	Geometry	Moderator / Reflector	H/^{239}Pu	Other Materials
pu-sol-therm-032-001	10.0	Pu(NO3)4	Annular	Water/Water	449.5	Steel
pu-sol-therm-032-002	10.0	Pu(NO3)4	Annular	Water/Water	488.2	Steel
pu-sol-therm-032-003	10.0	Pu(NO3)4	Annular	Water/Water	555.3	Steel
pu-sol-therm-032-004	10.0	Pu(NO3)4	Annular	Water/Water	622.5	Steel
pu-sol-therm-032-005	10.0	Pu(NO3)4	Annular	Water/Water	700.7	Steel
pu-sol-therm-032-006	10.0	Pu(NO3)4	Annular	Water/Water	800.5	Steel
pu-sol-therm-032-007	10.0	Pu(NO3)4	Annular	Water/Water	850.5	Steel
pu-sol-therm-032-008	10.0	Pu(NO3)4	Annular	Water/Water	949.6	Steel
pu-sol-therm-032-009	10.0	Pu(NO3)4	Annular	Water/Water	1021.5	Steel

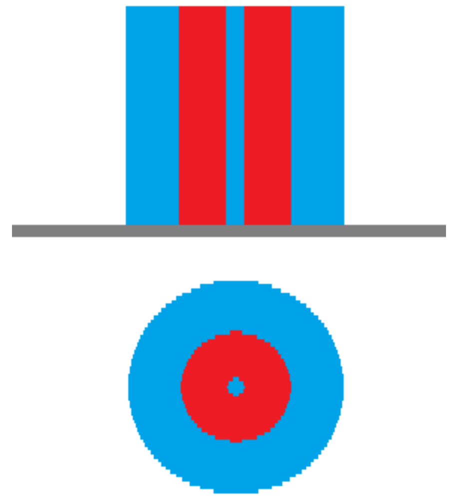
5.3 Metallic units

The enrichment subcritical limit for uranium and the mass subcritical limits given in Table 3 apply to a single piece having no concave surfaces.

Table 3 – Single-parameter subcritical limits for metal units

Parameter	Subcritical limits for		
	^{233}U [15]	^{235}U [16]	^{239}Pu [17]
Mass of fissile nuclide (kg)	6.0	20.1	5.0
Cylinder diameter (cm)	4.5	7.3	4.4
Slab thickness (cm)	0.38	1.3	0.65
Uranium enrichment (wt% ^{235}U)	–	5.0	–
Maximum density for which mass and dimension limits are valid (g/cm^3)	18.65	18.81	19.82

Example 2 - wval2p: 4.5 kg Pu Annulus, varying H & R_{in} (2)



- **4.5 kg Pu-239 right-circular cylinder, hollow**
- **Pu density = 19.86 g/cm^3**
- **Reflected radially with 1 inch of water**
- **Reflected on the bottom with ¼ inch steel**

- **Set the height to be same as solid cylinder with height-to-diameter (H/D) = 1.0, 2.0, 3.0**
- **For given height, vary inner radius over 0$^+$ - 2 cm**
  - **Start with wval2.txt input**

    mcnp6 i=wval2.txt

  - **Copy wval2.txt to wval2p.txt, then insert directives for mcnp_pstudy**
    - Define list for solid HD:

      ```
      c @@@ HD =  1.0  2.0  3.0
      ```

    - For a given H/D, compute Pu height
    - Define list for inner radius RIN_PU

      ```
      c @@@ RIN_PU =  0.001  0.5  1.0  2.0
      ```

    - Then other dimensions & source

Solid cylinder

$$V = (\text{Pu mass}) / (\text{Pu density})$$

$$V = H\pi R^2 = (\text{H/D}) \cdot 2\pi R^3$$

$$H = \left[4V(\text{H/D})^2 / \pi \right]^{1/3}$$

Hollow cylinder

$$V = H\pi(R_{out}^2 - R_{in}^2)$$

$$R_{out} = \left[R_{in}^2 + V/\pi H \right]^{1/2}$$

Example 2 - wval2p: 4.5 kg Pu Annulus, varying H & R_{in} (3)

mcnp

```
wval2: 4500 g Pu metal ring, fixed Rin
  1   3 -1.0          -1            imp:n=1
  2   1 -19.860000    +1 -2         imp:n=1
 11   3 -1.0          +2 -11        imp:n=1
 14   6 -7.92         -30           imp:n=1
 15   0               +11 +30 -20   imp:n=1
 20   0               +20           imp:n=0

  1 rcc  0 0 0        0 0  6.608   0.100000
  2 rcc  0 0 0        0 0  6.608   3.305259
 11 rcc  0 0 0        0 0  6.608   5.845259
 20 rcc  0 0 -2.540   0 0 91.44   91.44
 30 rcc  0 0 -0.635   0 0 0.635   76.20

kcode  10000 1.0 50 250
sdef pos=0 0 0  rad=d1 axs=0 0 1  ext=d2
 si1  0.100  3.305259
 sp1  -21 1
 si2  0.0   6.60800
 sp2  0    1
m1   94239.80c 1
m3   1001.80c 0.66667    8016.80c 0.33333
mt3  lwtr.20t
m6   24050.80c 0.000757334
     24052.80c 0.014604423
     24053.80c 0.001656024
     24054.80c 0.000412220
     26054.80c 0.003469592
     26056.80c 0.054465174
     26057.80c 0.001257838
     26058.80c 0.000167395
     25055.80c 0.00174
     28058.80c 0.005255537
     28060.80c 0.002024423
     28061.80c 0.000088000
     28062.80c 0.000280583
     28064.80c 0.000071456
prdmp 9e9 9e9 1 9e9
```

```
wval2p: 4500 g Pu metal ring, various H & Rin
c
c @@@  PI      = 3.141592654
c @@@  VOL_PU = ( 4500. / 19.86 )
c      Pu mass     = 4500 g
c      Pu density = 19.86 g/cc
c      Pu volume   = VOL_PU
c
c set height to match ingot with various H/D
c @@@  HD      =  1.0  2.0  3.0
c @@@  HEIGHT = ( (4*VOL_PU*(HD**2)/PI)**(1/3) )
c
c for hollow cylinder:
c   use same height as for solid ingot
c   set various inner radii
c   set Rout for given height, mass, Rin
c @@@  RIN_PU = .001  0.5  1.0  2.0
c @@@  ROUT_PU=(sqrt(RIN_PU**2+VOL_PU/(PI*HEIGHT)))
c @@@  ROUT_H2O = ( OUTER_PU + 2.54 )
c
  1   3 -1.0          -1             imp:n=1
  2   1 -19.860000    +1 -2          imp:n=1
 11   3 -1.0          +2 -11         imp:n=1
 14   6 -7.92         -30            imp:n=1
 15   0               +11 +30 -20    imp:n=1
 20   0               +20            imp:n=0

  1  rcc   0 0 0         0 0  HEIGHT  RIN_PU
  2  rcc   0 0 0         0 0  HEIGHT  ROUT_PU
 11  rcc   0 0 0         0 0  HEIGHT  ROUT_H2O
 20  rcc   0 0 -2.540    0 0 91.44    91.44
 30  rcc   0 0 -0.635    0 0 0.635    76.20

kcode  10000 1.0 50 250
sdef  pos= 0. 0. 0.   rad=d1  axs=0 0 1  ext=d2
 si1  RIN_PU  ROUT_PU
 sp1  -21 1
 si2  0 HEIGHT
 sp2  0  1
……………  etc.
```

Example 2 - wval2p: 4.5 kg Pu Annulus, varying H & R_{in} (4)

- **Parameter study using mcnp_pstudy, whisper_mcnp, & whisper_usl:**

```
mcnp_pstudy  -i  wval2p.txt  -whisper
```

use mcnp_pstudy to create inp files
  inp_case001, inp_case002, …, inp_case_012

```
whisper_mcnp.pl  -neutrons 10000  -discard 50  \
                  -cycles 250  -threads 4  \
                  inp_case*
```

use whisper_mcnp to run mcnp6 for each case &
  produce k_{eff} & sensitivity profile tallies

items in green are for class demo, so that cases run quickly,
  & should not be used for serious work

(For Windows, use ^ instead of \ for continuation)

```
whisper_usl.pl
```

use whisper_usl to run Whisper & determine USL for each case

Example 2 - wval2p: 4.5 kg Pu Annulus, varying H & R_{in} (5)

wval2

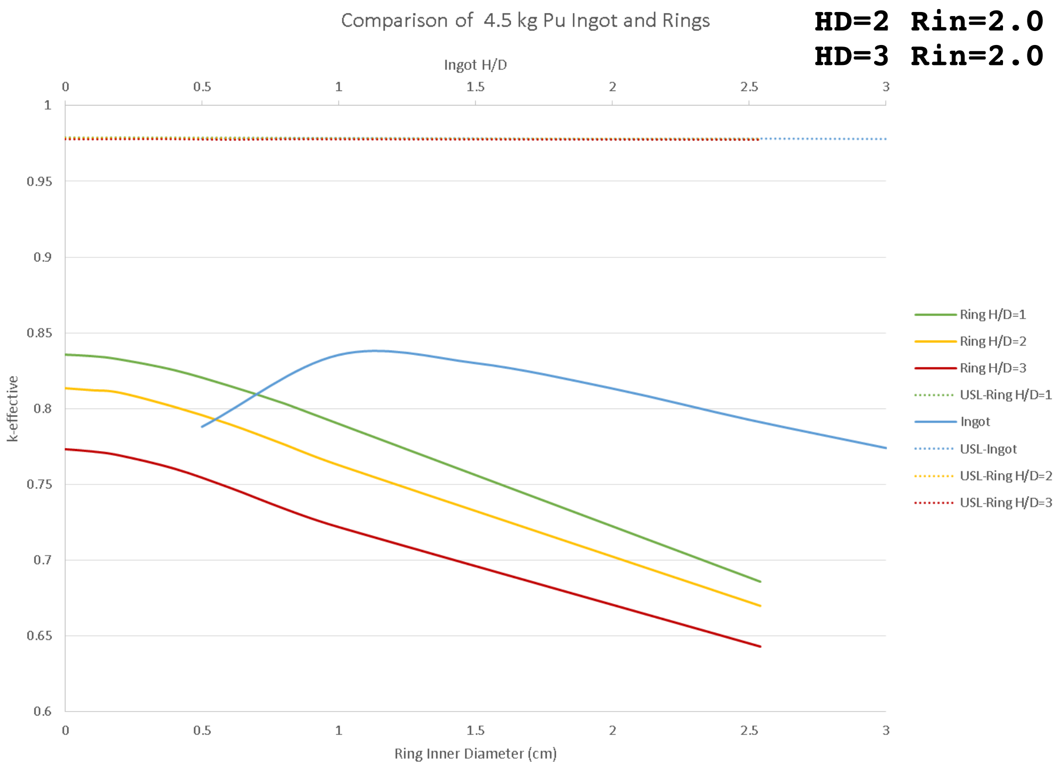
mcnp6 i=wval2.txt

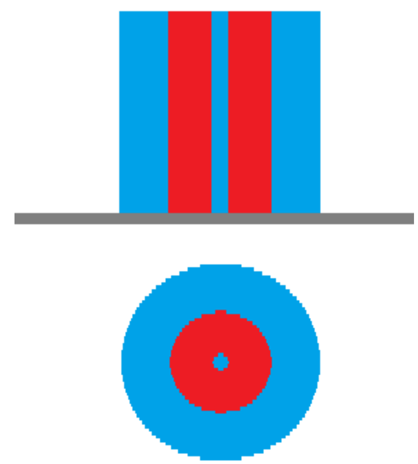
k = 0.83413 (42)

wval2p, varying H & R_{in}

mcnp_pstudy -i wval2p.txt -setup -run

HD=1	Rin=.001	case001	KEFF	8.34752E-01	4.35668E-04
HD=2	Rin=.001	case002	KEFF	8.12612E-01	4.09516E-04
HD=3	Rin=.001	case003	KEFF	7.72725E-01	3.82627E-04
HD=1	Rin=0.5	case004	KEFF	8.20432E-01	4.01135E-04
HD=2	Rin=0.5	case005	KEFF	7.95375E-01	4.60388E-04
HD=3	Rin=0.5	case006	KEFF	7.54174E-01	3.96580E-04
HD=1	Rin=1.0	case007	KEFF	7.88497E-01	3.95026E-04
HD=2	Rin=1.0	case008	KEFF	7.62394E-01	3.90299E-04
HD=3	Rin=1.0	case009	KEFF	7.20810E-01	4.27354E-04
HD=1	Rin=2.0	case010	KEFF	7.21523E-01	4.02775E-04
HD=2	Rin=2.0	case011	KEFF	6.97954E-01	4.88269E-04
HD=3	Rin=2.0	case012	KEFF	6.64037E-01	4.88326E-04





Example 2 - wval2p: 4.5 kg Pu Annulus, varying H & R_{in} (6)

MCNP6-Whisper Results

application	calc margin	data unc (1-sigma)	baseline USL	k(calc) > USL
ringhd2.txt_0.4_in	0.01464	0.00075	0.97840	-0.17760

```
Benchmark population   =   41
Population weight      =  25.47164
Maximum similarity     =   0.99532

Bias                   =   0.00836
Bias uncertainty       =   0.00628
Nuc Data uncert margin =   0.00075
Software/method margin =   0.00500
Non-coverage penalty   =   0.00000
```

benchmark	ck	weight
pu-met-fast-036-001.i	0.9953	1.0000
pu-met-fast-024-001.i	0.9941	0.9608
pu-met-fast-044-005.i	0.9933	0.9360
pu-met-fast-011-001.i	0.9928	0.9196
pu-met-fast-044-004.i	0.9925	0.9117
pu-met-fast-044-003.i	0.9898	0.8275
pu-met-fast-023-001.i	0.9890	0.8020
pu-met-fast-022-001.i	0.9886	0.7898
pu-met-fast-039-001.i	0.9884	0.7823

benchmark	ck	weight
pu-met-fast-044-002.i	0.9876	0.7587
pu-met-fast-031-001.i	0.9875	0.7561
pu-met-fast-021-002.i	0.9867	0.7284
pu-met-fast-042-002.i	0.9863	0.7158
pu-met-fast-042-004.i	0.9862	0.7124
pu-met-fast-042-003.i	0.9861	0.7104
pu-met-fast-001-001.i	0.9859	0.7051
mix-met-fast-009-001.i	0.9854	0.6873
pu-met-fast-035-001.i	0.9851	0.6798
pu-met-fast-009-001.i	0.9846	0.6633
pu-met-fast-042-006.i	0.9843	0.6536
pu-met-fast-042-005.i	0.9840	0.6446
pu-met-fast-042-007.i	0.9833	0.6237
pu-met-fast-042-001.i	0.9833	0.6230
pu-met-fast-025-001.i	0.9829	0.6103
pu-met-fast-042-008.i	0.9825	0.5980
pu-met-fast-027-001.i	0.9825	0.5975
pu-met-fast-042-009.i	0.9821	0.5843
pu-met-fast-042-010.i	0.9815	0.5667
pu-met-fast-042-011.i	0.9811	0.5543
pu-met-fast-042-012.i	0.9808	0.5435
pu-met-fast-042-013.i	0.9800	0.5202
pu-met-fast-042-014.i	0.9799	0.5175
pu-met-fast-042-015.i	0.9799	0.5159
pu-met-fast-030-001.i	0.9782	0.4626
pu-met-fast-021-001.i	0.9780	0.4560
pu-met-fast-029-001.i	0.9777	0.4468
pu-met-fast-044-001.i	0.9743	0.3409
pu-met-fast-018-001.i	0.9720	0.2678
mix-met-fast-007-022.i	0.9690	0.1754
mix-met-fast-007-023.i	0.9655	0.0635
pu-met-fast-045-005.i	0.9653	0.0586

Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 - AoA

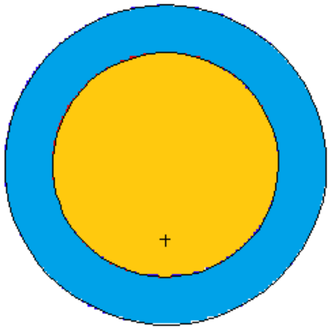
Example 3

–

4.5 kg Pu-NaCl Mixture

Example 3 – wval3: 4.5 kg Pu-NaCl Mixture (1)

- 4.5 kg Pu (0) sphere mixed with variable amounts (0-2 kg) of NaCl
- Reflected with 1 inch of water
- Density of Pu = 19.86 g/cm^3
- Density of NaCl = 1.556 g/cm^3



- Run commands:

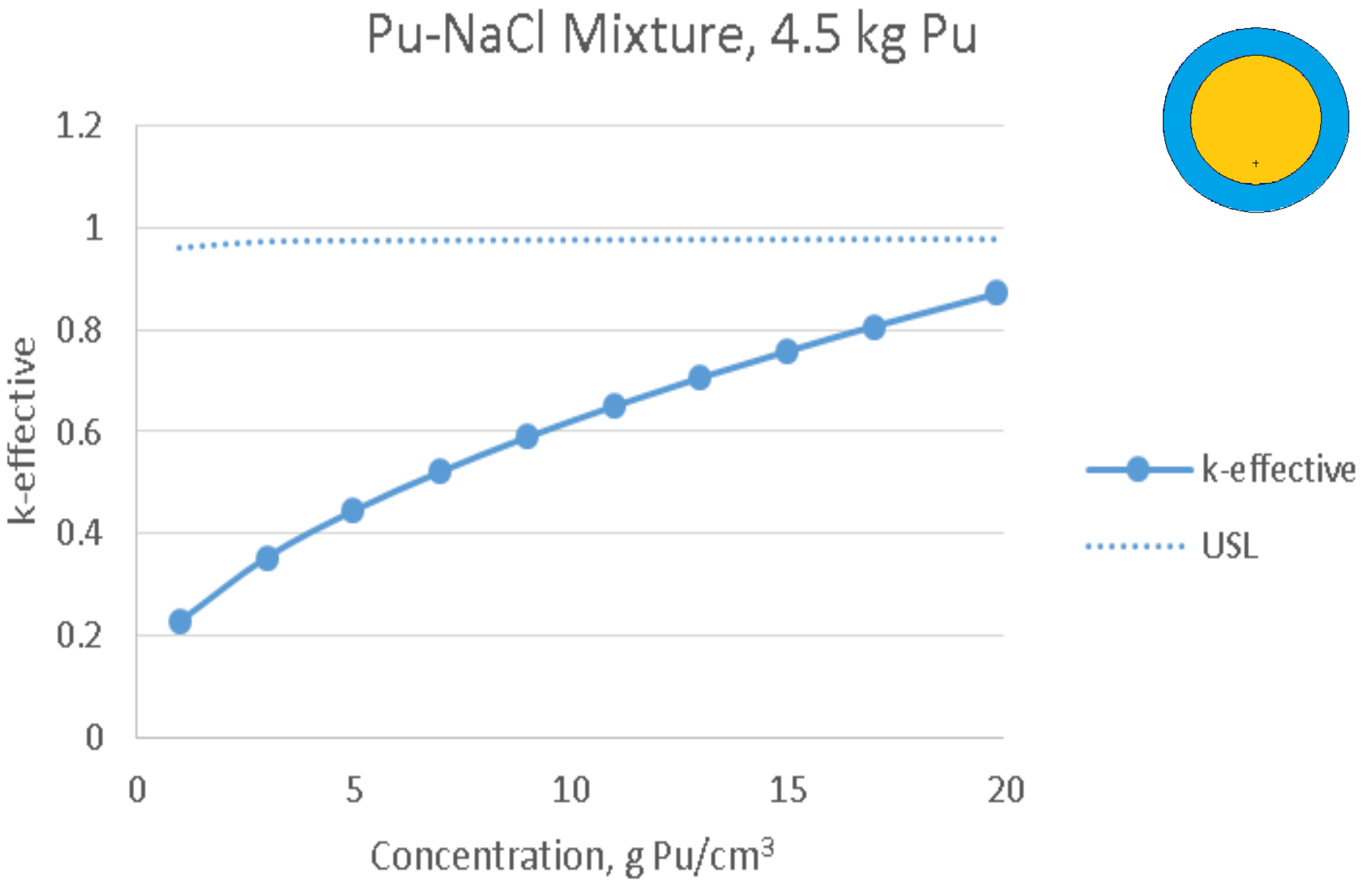
```
mcnp_pstudy  -i  wval3p.txt  -whisper
whisper_mcnp.pl  -neutrons 10000 -discard 50 \
                  -cycles 250  -threads 4  \
                 inp_case*
whisper_usl.pl
```

Example 3 – wval3: 4.5 kg Pu-NaCl Mixture (2)

```
wval3:  Study of Pu mixed with NaCl
c
 1   4 -6.163863   -1     imp:n=1
 2   1 -1.0        +1 -2  imp:n=1
20   0             +2     imp:n=0

 1 sph  0 0 0   5.98941813698262
 2 sph  0 0 0   8.52941813698262

kcode 10000 1.0 150 500
sdef  pos=0 0 0  rad=d1
 si1  0  5.989
 sp1  -21 2
c
m1    1001.80c 2     8016.80c 1
mt1   lwtr.20t
m4    94239.80c -0.81117881
      11023.80c -0.07427730
      17035.80c -0.08561650
      17037.80c -0.02893221
```

```
wval3p:  Pu mixed with NaCl
c @@@          PI = 3.141592654
c @@@ PU_MASS   = 4500
c @@@ PU_VOL    = ( PU_MASS / 19.86 )
c @@@ NACL_MASS = 1.e-6  500 1000 1500 2000
c @@@ NACL_VOL  = ( NACL_MASS / 1.556 )
c
c    Pu   mass = PU_MASS g
c    NaCl mass = NACL_MASS g
c    Pu   density (pure) = 19.86 g/cc
c    NaCl density (pure) = 1.556 g/cc
c
c @@@ VOLUME      = ( PU_VOL  + NACL_VOL  )
c @@@ MASS        = ( PU_MASS + NACL_MASS )
c @@@ DENSITY     = ( -MASS/VOLUME )
c @@@ DENSITY_PU  = ( PU_MASS/VOLUME )
c     Pu density  = DENSITY_PU   g/cc
c @@@ RADIUS      = ( (0.75*VOLUME/PI)**(1/3) )
c @@@ OUTER_H2O   = ( RADIUS + 2.54 )
c
c @@@ A11023 = 22.98976928
c @@@ A17035 = ( 34.96885268 * 0.7576 )
c @@@ A17037 = ( 36.96590259 * 0.2424 )
c @@@ A_NACL = ( A11023 + A17035 + A17037 )
c
c @@@ MF94239 = ( -PU_MASS/MASS  )
c @@@ MF11023 = ( -NACL_MASS*(A11023/A_NACL)/MASS )
c @@@ MF17035 = ( -NACL_MASS*(A17035/A_NACL)/MASS )
c @@@ MF17037 = ( -NACL_MASS*(A17037/A_NACL)/MASS )
c
 1   4 DENSITY     -1    imp:n=1
 2   1 -1.0        +1 -2 imp:n=1
20   0             +2    imp:n=0

 1  so   RADIUS
 2  so   OUTER_H2O

kcode 10000 1.0 50 250
sdef  pos=0 0 0  rad=d1
 si1  0  RADIUS
 sp1  -21 2
m1    1001.80c 2     8016.80c 1
mt1   lwtr.20t
m4    94239.80c  MF94239
      11023.80c  MF11023
      17035.80c  MF17035
      17037.80c  MF17037
prdmp 9e9 9e9 1 9e9
```

MCNP6-Whisper Results



Example 3 – wval3: 4.5 kg Pu-NaCl Mixture (4)

*bold indicates same benchmark selected for Pu ingot

MCNP6-Whisper Results

```
                                   USL baseline = .979
Benchmark population       =    46
Benchmark weight           =    25.75745
Benchmark similarity       =    0.99245

Bias                       =    0.00796
Bias uncertainty           =    0.00682
Nuc Data                   =    0.0012
Software/method margin     =    0.005
Non-coverage penalty       =    0
```

benchmark	ck	weight
pu-met-fast-011-001.i	**0.9924**	**1**
pu-met-fast-044-004.i	**0.9842**	**0.8636**
pu-met-fast-042-001.i	**0.9831**	**0.8448**
pu-met-fast-042-002.i	**0.9828**	**0.8396**
pu-met-fast-044-005.i	**0.9827**	**0.8377**
pu-met-fast-027-001.i	**0.981**	**0.8107**
pu-met-fast-036-001.i	**0.9805**	**0.8018**
pu-met-fast-042-003.i	**0.9802**	**0.7965**
pu-met-fast-031-001.i	**0.9792**	**0.7798**
pu-met-fast-042-004.i	**0.9787**	**0.7727**
pu-met-fast-024-001.i	**0.978**	**0.7604**
pu-met-fast-044-003.i	**0.9768**	**0.7401**
pu-met-fast-042-005.i	**0.9757**	**0.7213**
pu-met-fast-042-006.i	**0.9746**	**0.7039**
pu-met-fast-021-002.i	**0.9737**	**0.6893**
pu-met-fast-044-002.i	**0.9734**	**0.6832**
pu-met-fast-042-007.i	**0.9734**	**0.6832**
pu-met-fast-042-008.i	**0.9722**	**0.6645**
pu-met-fast-042-009.i	**0.9709**	**0.6426**
pu-met-fast-042-010.i	**0.9705**	**0.6356**
pu-met-fast-042-011.i	**0.9699**	**0.6257**
pu-met-fast-023-001.i	**0.9691**	**0.6133**
pu-met-fast-042-012.i	**0.9687**	**0.6054**
pu-met-fast-039-001.i	**0.9683**	**0.5993**
pu-met-fast-042-014.i	**0.9681**	**0.5961**
pu-met-fast-042-013.i	**0.9681**	**0.5959**
pu-met-fast-042-015.i	**0.9676**	**0.587**
pu-met-fast-022-001.i	**0.9644**	**0.534**
pu-met-fast-009-001.i	**0.964**	**0.5284**
pu-met-fast-035-001.i	**0.9629**	**0.5093**
mix-met-fast-009-001.i	**0.9618**	**0.4919**
pu-met-fast-044-001.i	**0.9612**	**0.482**
pu-met-fast-001-001.i	**0.9602**	**0.4653**
pu-met-fast-025-001.i	**0.9593**	**0.4499**
pu-met-fast-021-001.i	**0.9588**	**0.4424**
pu-met-fast-030-001.i	**0.9559**	**0.3941**
pu-met-fast-018-001.i	**0.9555**	**0.3863**
pu-met-fast-029-001.i	**0.951**	**0.3115**
pu-met-fast-045-005.i	**0.9509**	**0.3097**
mix-met-fast-007-022.i	**0.9496**	**0.2897**
mix-met-fast-007-023.i	**0.9448**	**0.2093**
pu-met-fast-019-001.i	0.9421	0.1637
pu-met-fast-038-001.i	0.9384	0.1032
mix-met-fast-001-001.i	**0.9374**	**0.0871**
pu-met-fast-040-001.i	0.9355	0.055
pu-met-fast-003-103.i	**0.9352**	**0.0505**

Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 – AoA

Example 4

–

4.5 kg Pu Sphere,

Ta Reflector, various thicknesses

- **Reflection: Ta**
  - **Is Ta validated as a reflector in the AoA?**
  - **What can be done to answer this question and, if needed, possibly extend AoA?**

From a typical traditional validation report

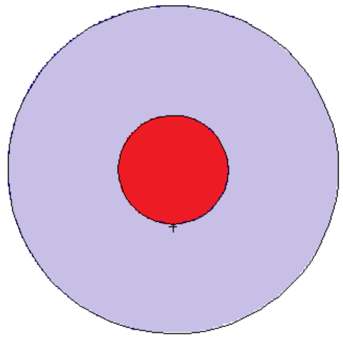
Parameter	Area of Applicability
Fissile Material	^{239}Pu
Fissile Material Form	Pu Metal, PuO_2, and $Pu(NO_3)_4$
$H/^{239}Pu$	$0 \leq H/^{239}Pu \leq 2807$
Average Neutron Energy Causing Fission (MeV)	$0.003 \leq ANECF \leq 1.935$
^{240}Pu	0 to 42.9 wt% ^{240}Pu
Moderating Materials	none, water, graphite, polystyrene
Reflecting Materials	none, water, steel, oil, Plexiglas, polyethylene, graphite, W, Cu, U, Th, Al, Ni, Fe, Pb, Cd, Mo, Be, BeO
Other Materials	concrete, PVC, Ga, B, Gd, Ta
Geometry	cylinder array, cylinder, slab, sphere, hemisphere, stacked discs, cuboid, annular

- **CSSG Response on Validation with Limited Benchmark Data:**

"For those situations where a nuclide is determined to be important and limited data exist, validation may still be possible. However, an additional margin should be used to compensate for the limited data. This margin is separate from, and in addition to, any margin needed for extending the benchmark applicability to the validation. Sensitivity and uncertainty tools may be used as part of the technical basis for determining the magnitude of the margin."

Example wval4: 4.5 kg Pu Sphere, Ta-reflected (1)

- 4.5 kg Pu-239 sphere
- Pu density = 19.8 g/cm^3
- Reflected radially with Ta



- Vary the Ta-reflector thickness over the range 0.$^+$ – 30. cm
  - Start with wval4.txt, input for thickness=7.62

    mcnp6 i=wval4.txt

  - Copy wval4.txt to wval4p.txt, then insert directives for mcnp_pstudy
    - Define list for thickness:

      ```
      c @@@ THICK =  0.01  5.  10.  15.  20.  25.  30.
      ```

    - For a given THICK, compute reflector Rin & Rout
    - Use parameters for dimensions & location of KSRC point
    - Run:

      ```
      mcnp_pstudy.pl -i wval4p.txt  -whisper
      whisper_mcnp.pl   inp_case*
      whisper_usl.pl
      ```

Example wval4: 4.5 kg Pu Sphere, Ta-reflected (2)

```
wval4:  Study of Pu reflected with Ta
c
c  Pu mass    = 4500 g
c  Pu density = 19.8 g/cc
c  Pu volume  = 227.272727
c
c  reflector definition:
c    reflector thickness    = 7.62
c    reflector inner radius = 3.7857584
c    reflector outer radius = 11.405758
c
  1   4 -19.80  -1         imp:n=1
  2   1 -16.69  +1 -2      imp:n=1
 20   0         +2         imp:n=0

  1 so  3.7857584
  2 so  11.405758

 kcode 10000 1.0 50 250
 sdef pos=0 0 0  rad=d1
  si1  0 3.78
  sp1  -21 2
c
 m1  73180.80c 0.00012   73181.80c 0.99988
 m4  94239.80c 1
 prdmp 9e9 9e9 1 9e9
```

```
wval4p:  Study of Pu reflected with Ta
c
c  Pu mass    = 4500 g
c  Pu density = 19.8 g/cc
c  Pu volume  = 227.272727
c
c  vary reflector thickness from 0+ to 30 cm
c
c  @@@  THICK   = .01  5. 10. 15. 20. 25. 30.
c  @@@  R_INNER = 3.7857584
c  @@@  R_OUTER = ( R_INNER + THICK )
c
c  reflector definition:
c    reflector thickness    = THICK cm
c    reflector inner radius = R_INNER cm
c    reflector outer radius = R_OUTER cm
c
  1   4 -19.80  -1         imp:n=1
  2   1 -16.69  +1 -2      imp:n=1
 20   0         +2         imp:n=0

  1 so    R_INNER
  2 so    R_OUTER

 kcode 10000 1.0 50 250
 sdef pos=0 0 0 rad=d1
  si1  0  R_INNER
  sp1  -21 2
c
 m1  73180.80c 0.00012   73181.80c 0.99988
 m4  94239.80c 1
 prdmp 9e9 9e9 1 9e9
```

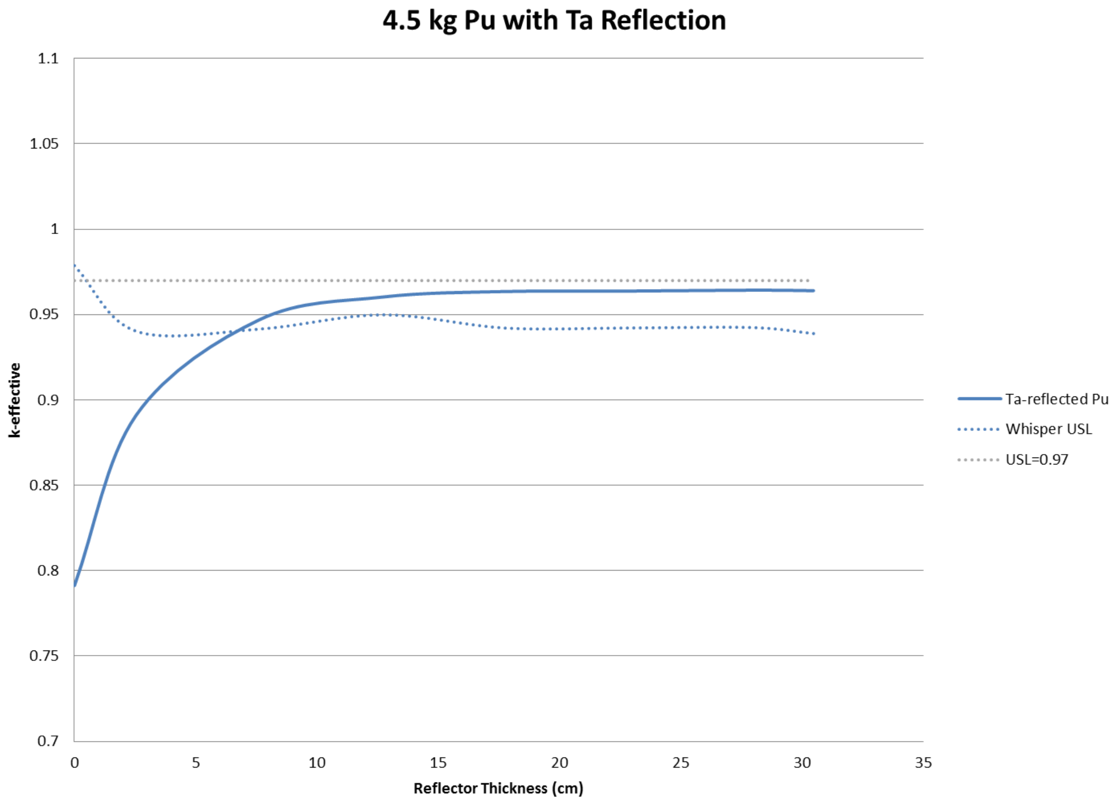
Example wval4: 4.5 kg Pu Sphere, Ta-reflected (3)

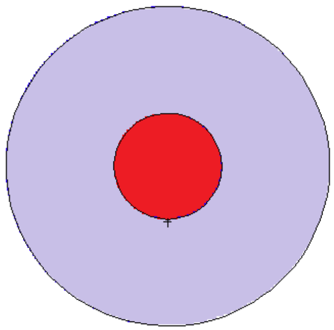
wval4, thick=7.62
mcnp6 i=wval4.txt

k = 0.94638 (41)

wval4p, varying thick
mcnp_pstudy -i wval4p.txt -setup -run

T=.01	case001	KEFF	7.91693E-01	KSIG	3.14948E-04
T=5.0	case002	KEFF	9.27157E-01	KSIG	4.47334E-04
T=10.	case003	KEFF	9.54775E-01	KSIG	4.11031E-04
T=15.	case004	KEFF	9.61644E-01	KSIG	4.34033E-04
T=20.	case005	KEFF	9.62867E-01	KSIG	4.37235E-04
T=25.	case006	KEFF	9.63899E-01	KSIG	4.04508E-04
T=30.	case007	KEFF	9.63160E-01	KSIG	4.27633E-04





Example 4: Ta-reflected Pu

MCNP6 and Whisper Results

application	calc margin	data unc (1-sigma)	baseline USL	k(calc) > USL
tarefl.txt_7.62_in	0.01707	0.01502	0.93889	0.00750

```
Benchmark population   =  119
Population weight      =  60.92464
Maximum similarity     =   0.64075

Bias                   =   0.00912
Bias uncertainty       =   0.00795
Nuc Data uncert margin =   0.01502
Software/method margin =   0.00500
Non-coverage penalty   =   0.00000
```

Trouble ! Benchmarks are not very similar to application

benchmark	ck	weight
pu-met-fast-045-006.i	0.6408	1.0000
pu-met-fast-045-004.i	0.6400	0.9986
pu-met-fast-045-003.i	0.6368	0.9926
pu-met-fast-045-002.i	0.6297	0.9796
pu-met-fast-045-007.i	0.6259	0.9725
pu-met-fast-045-001.i	0.6213	0.9641
pu-met-fast-045-005.i	0.5469	0.8270
pu-met-fast-023-001.i	0.4203	0.5937
pu-met-fast-039-001.i	0.4201	0.5935

Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 – AoA

benchmark	ck	weight
mix-met-fast-009-001.i	0.4193	0.5919
pu-met-fast-009-001.i	0.4190	0.5914
pu-met-fast-035-001.i	0.4189	0.5913
pu-met-fast-022-001.i	0.4185	0.5904
pu-met-fast-025-001.i	0.4183	0.5900
pu-met-fast-036-001.i	0.4180	0.5896
pu-met-fast-001-001.i	0.4180	0.5895
pu-met-fast-021-002.i	0.4176	0.5887
pu-met-fast-030-001.i	0.4171	0.5879
pu-met-fast-024-001.i	0.4171	0.5878
pu-met-fast-021-001.i	0.4165	0.5867
pu-met-fast-044-003.i	0.4164	0.5866
pu-met-fast-044-005.i	0.4162	0.5863
pu-met-fast-044-002.i	0.4160	0.5858
pu-met-fast-029-001.i	0.4155	0.5850
pu-met-fast-044-004.i	0.4146	0.5832
pu-met-fast-003-103.i	0.4141	0.5823
pu-met-fast-042-015.i	0.4134	0.5811
pu-met-fast-042-012.i	0.4134	0.5811
mix-met-fast-007-022.i	0.4134	0.5811
pu-met-fast-042-011.i	0.4134	0.5810
pu-met-fast-042-009.i	0.4134	0.5810
pu-met-fast-042-013.i	0.4133	0.5808
pu-met-fast-042-014.i	0.4133	0.5808
pu-met-fast-042-010.i	0.4133	0.5808
pu-met-fast-042-007.i	0.4132	0.5807
pu-met-fast-018-001.i	0.4132	0.5806
pu-met-fast-042-006.i	0.4131	0.5806
pu-met-fast-042-008.i	0.4131	0.5805

…….

Example 4: Ta-reflected Pu

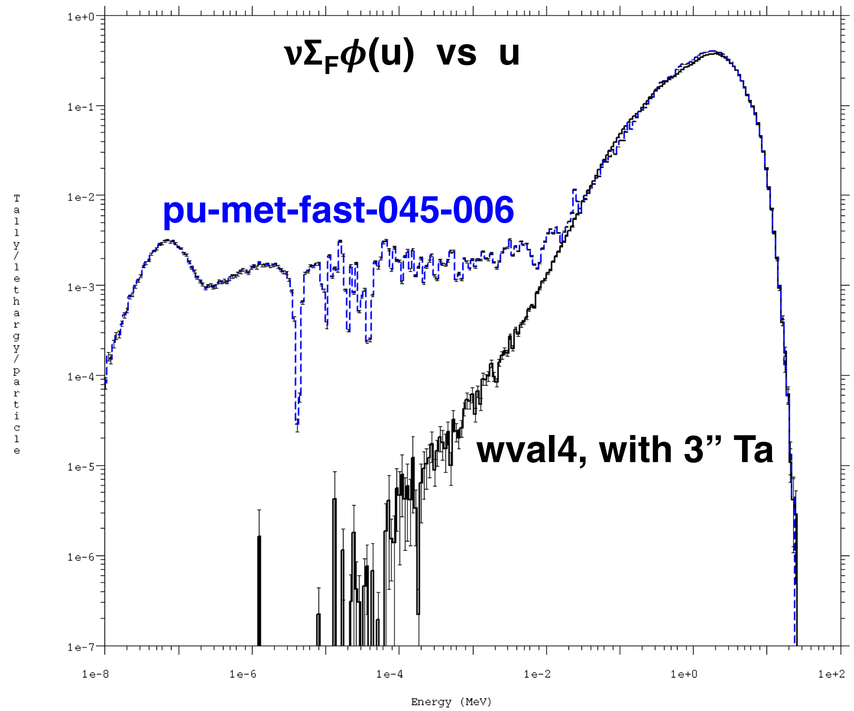
- **None of the benchmarks appear to have the same neutronics as the application**
  - **Largest C_k in the Whisper example output is 0.64 – very low**
  - **Guidance from ORNL Scale/Tsunami developers:**

$0.95 < C_k$	→	great
$0.90 < C_k < 0.95$	→	good
$C_k < 0.90$	→	not so good

For C_k's in range 0.9 – 1.0, at least 5-10 benchmarks needed

For C_k's in range 0.8 – 0.9, at least 10-20 benchmarks needed

  - **If all C_k's are low, there is a need to expand the benchmark suite, add similar benchmarks**
  - **If no similar benchmarks, need extra analysis, analyst judgment, & margin**



  - **The current benchmark suite for Whisper was focused on main needs for LANL validation, few benchmarks with Ta**
  - **Need to find more benchmarks with Ta reflector & add to Whisper suite, if Ta-reflected applications are expected**

Example 5

–

4.5 kg Pu Sphere,

Oil moderated

Example 5: Oil-Moderated Pu

- **Is Pu moderated with oil included in validation AoA?**
  - **If not, what can be done?**

From a typical traditional validation report

Parameter	Area of Applicability
Fissile Material	^{239}Pu
Fissile Material Form	Pu Metal, PuO_2, and $Pu(NO_3)_4$
$H/^{239}Pu$	$0 \leq H/^{239}Pu \leq 2807$
Average Neutron Energy Causing Fission (MeV)	$0.003 \leq ANECF \leq 1.935$
^{240}Pu	0 to 42.9 wt% ^{240}Pu
Moderating Materials	none, water, graphite, polystyrene
Reflecting Materials	none, water, steel, oil, Plexiglas, polyethylene, graphite, W, Cu, U, Th, Al, Ni, Fe, Pb, Cd, Mo, Be, BeO
Other Materials	concrete, PVC, Ga, B, Gd, Ta
Geometry	cylinder array, cylinder, slab, sphere, hemisphere, stacked discs, cuboid, annular

  - **Additionally the primary CSA shall determine that the calculation model(s) fits within the area of applicability of the benchmark critical experiments used for the code validation. The area of applicability determination quantifies parameters potentially important to the computational calculation of keff. This comparison of calculation models and the benchmark critical experiments insures that the selected USL is valid for the calculations being performed. For systems which are outside the validation area of applicability, an area of applicability margin (AoA) may also be warranted, depending on the specific problem being analyzed. The analyst must document and justify any extrapolation beyond the validation area of applicability, including any chosen margin. The resulting USL with an AoA margin is defined as**

USL = 1.0 + (bias) – (bias uncertainty) – (margin of subcriticality) – (AoA margin)

Example 5: Oil-Moderated Pu

- **MCNP6 Input**

- **4.5 kg Pu (0) sphere mixed with variable amounts of Hydraulic oil**

- **Pu concentration range:**

  -19.8 g Pu/cm^3

- **Hydraulic oil composition:**

  $C_{40}H_{33}O_4Cl_6P$

- **Hydraulic oil density:**

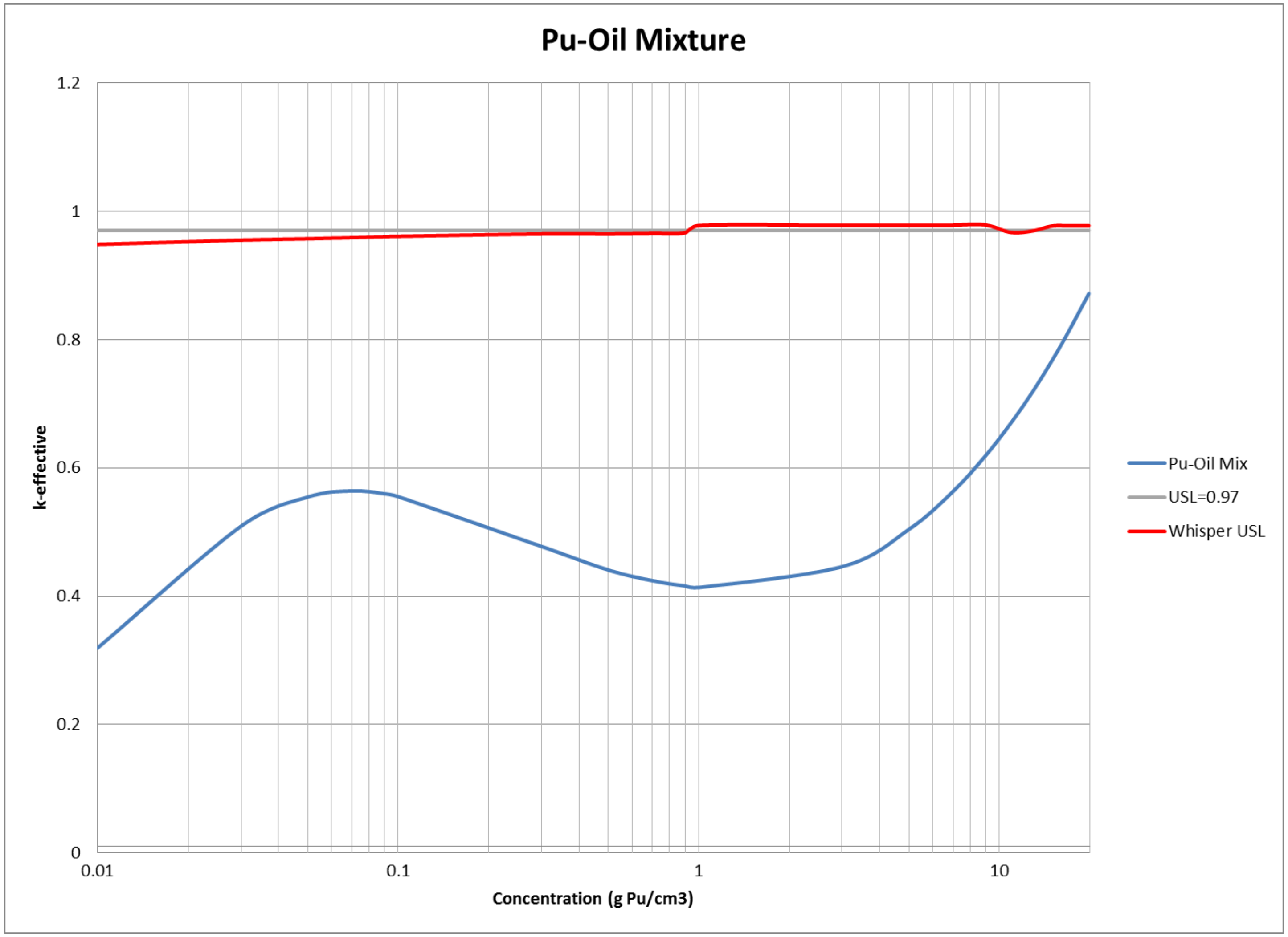
  0.871 g/cm^3

- **Reflected with 1 inch of water**

```
Pu mixed with hydraulic oil
c
1    4 -1.827099    -1      imp:n=1
2    1 -1.0         +1 -2   imp:n=1
20   0              +2      imp:n=0

1   so    10.2417609488294
2   so    12.7817609488294

kcode 10000 1.0 150 500
ksrc  0 0 0
c
m1    1001.80c 2
      8016.80c 1
mt1   lwtr.20t
m4    94239.80c  -0.54731523
       1001.80c  -0.01821054722413
       6000.80c  -0.264852020155431
       8016.80c  -0.0352799376428247
      15031.80c  -0.0170753227802324
      17035.80c  -0.0876520545992508
      17037.80c  -0.0296143373586584
```

Example 5: Oil-Moderated Pu

- **MCNP6 and Whisper Results**



Example 5: Oil-Moderated Pu

MCNP6 and Whisper Results

application	calc margin	data unc (1-sigma)	baseline USL	k(calc) > USL
puoilmix.txt_7_in	0.01477	0.00109	0.97739	-0.41445

```
Benchmark population    =   65
Population weight       =  28.56693
Maximum similarity      =   0.96433

Bias                    =   0.00720
Bias uncertainty        =   0.00757
Nuc Data uncert margin  =   0.00109
Software/method margin  =   0.00500
Non-coverage penalty    =   0.00000
```

benchmark	ck	weight
pu-met-fast-042-001.i	0.9643	1.0000
pu-met-fast-011-001.i	0.9641	0.9973
pu-met-fast-027-001.i	0.9580	0.9377
pu-met-fast-042-002.i	0.9561	0.9199
pu-met-fast-042-003.i	0.9483	0.8436
pu-met-fast-044-004.i	0.9474	0.8343
pu-met-fast-042-004.i	0.9444	0.8048
pu-met-fast-031-001.i	0.9425	0.7861
pu-met-fast-044-005.i	0.9404	0.7658
pu-comp-mixed-002-001.i	0.9388	0.7502
pu-met-fast-042-005.i	0.9373	0.7353
pu-comp-mixed-002-002.i	0.9344	0.7077
pu-met-fast-042-006.i	0.9344	0.7069
pu-met-fast-042-007.i	0.9320	0.6840
pu-met-fast-036-001.i	0.9310	0.6736
pu-met-fast-044-003.i	0.9307	0.6714
pu-met-fast-042-008.i	0.9303	0.6673
pu-met-fast-024-001.i	0.9277	0.6417
pu-met-fast-042-009.i	0.9271	0.6360
pu-met-fast-042-010.i	0.9268	0.6327
pu-comp-mixed-002-003.i	0.9267	0.6315
pu-met-fast-042-011.i	0.9255	0.6198
pu-met-fast-042-012.i	0.9228	0.5943
pu-met-fast-044-002.i	0.9224	0.5899
pu-met-fast-042-014.i	0.9224	0.5896
pu-met-fast-042-013.i	0.9222	0.5881
pu-met-fast-042-015.i	0.9209	0.5752
pu-comp-mixed-002-004.i	0.9191	0.5574
pu-met-fast-021-002.i	0.9184	0.5506
pu-met-fast-044-001.i	0.9145	0.5128
pu-met-fast-023-001.i	0.9046	0.4156
pu-met-fast-039-001.i	0.9031	0.4015
pu-comp-mixed-002-005.i	0.9030	0.3999
pu-met-fast-018-001.i	0.9008	0.3782
pu-met-fast-021-001.i	0.8989	0.3598
pu-met-fast-009-001.i	0.8985	0.3564
pu-met-fast-016-001.i	0.8965	0.3364
pu-met-fast-045-005.i	0.8954	0.3259

…….

Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 - AoA

Example 6

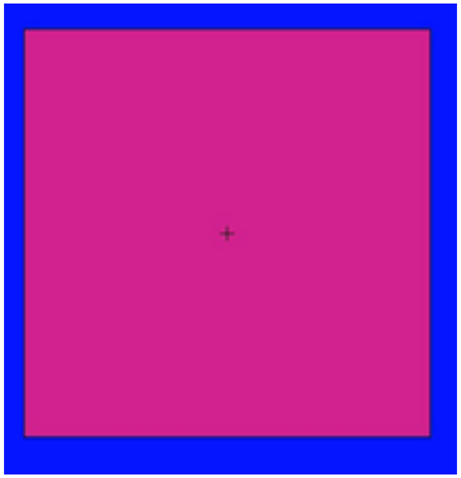
–

Revisiting a Practical Application of the SPSL for Pu Metal

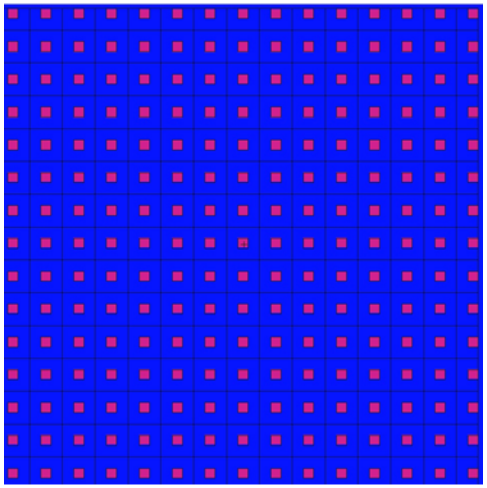
Example 6: Revisiting a Practical Application of the SPSL for Pu Metal

LANL's Nuclear Criticality Safety Group undertook an effort to define a threshold between un-moderated and moderated plutonium metal systems. This effort culminated in the issuing of LA-UR-07-0160, *Practical Application of the Single-Parameter Subcritical Mass Limit for Plutonium* [Ref. 1]. The stated goal of this document was to answer the question of when do plutonium metal and water mixtures cease to appear as "metal" systems and begin to appear more like "solution" systems. Even though the study involving plutonium (^{239}Pu) metal cubes in water was performed using MCNP [Ref. 2], the subject of code validation was intentionally ignored. This study is being revisited, and Upper Subcritical Limits (USLs) are being presented, using Whisper [Ref. 3].

1. LA-UR-07-0160, *Practical Application of the Single-Parameter Subcritical Mass Limit for Plutonium Metal,* 2007.
2. LA-12625-M, *MCNP - A General Monte Carlo N-Particle Transport Code*, 1997.
3. LA-UR-14-26558, *Whisper: Sensitivity/Uncertainty-Based Computational Methods and Software for Determining Baseline Upper Subcritical Limits,* 2014.



N = 1,
Mass Per Cube = 5,000 g,
Spacing = N/A



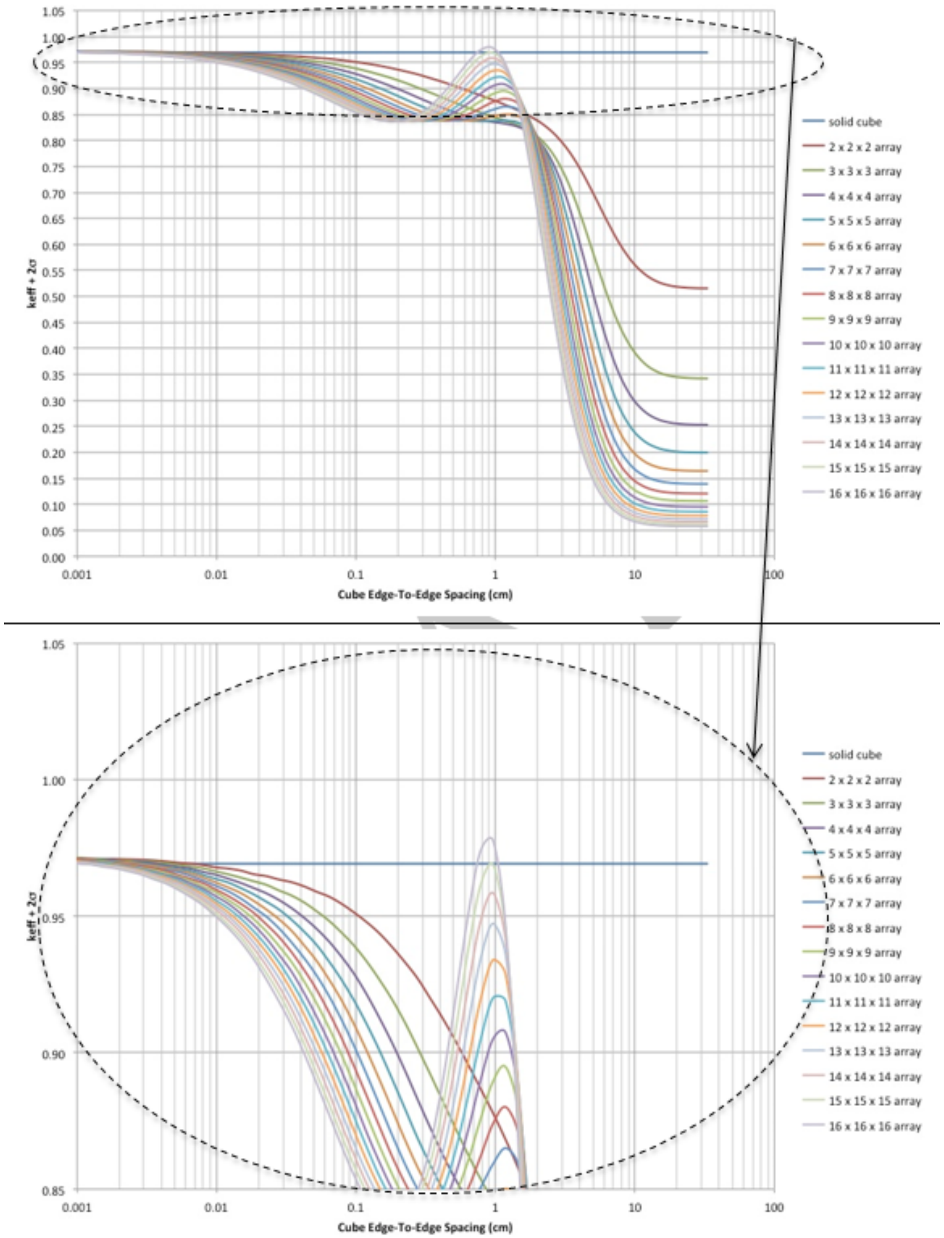
N = 15,
Mass Per Cube = ~1.48 g,
Spacing = 1 cm

5.3 Metallic units

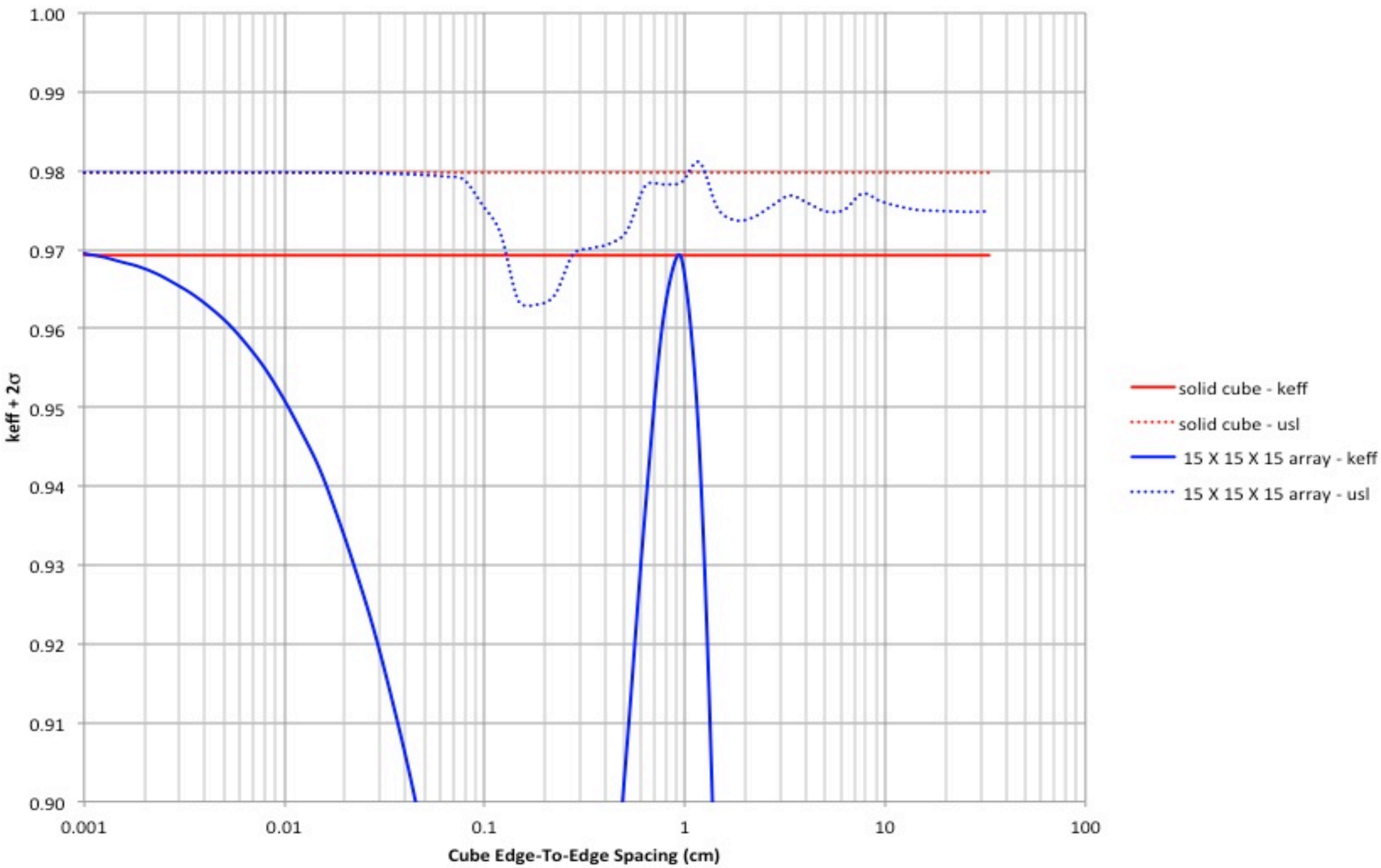
The enrichment subcritical limit for uranium and the mass subcritical limits given in Table 3 apply to a single piece having no concave surfaces.

Table 3 – Single-parameter subcritical limits for metal units

Parameter	Subcritical limits for		
	^{233}U [15]	^{235}U [16]	^{239}Pu [17]
Mass of fissile nuclide (kg)	6.0	20.1	5.0
Cylinder diameter (cm)	4.5	7.3	4.4
Slab thickness (cm)	0.38	1.3	0.65
Uranium enrichment (wt% ^{235}U)	–	5.0	–
Maximum density for which mass and dimension limits are valid (g/cm³)	18.65	18.81	19.82



Example 6: Revisiting a Practical Application of the SPSL for Pu Metal

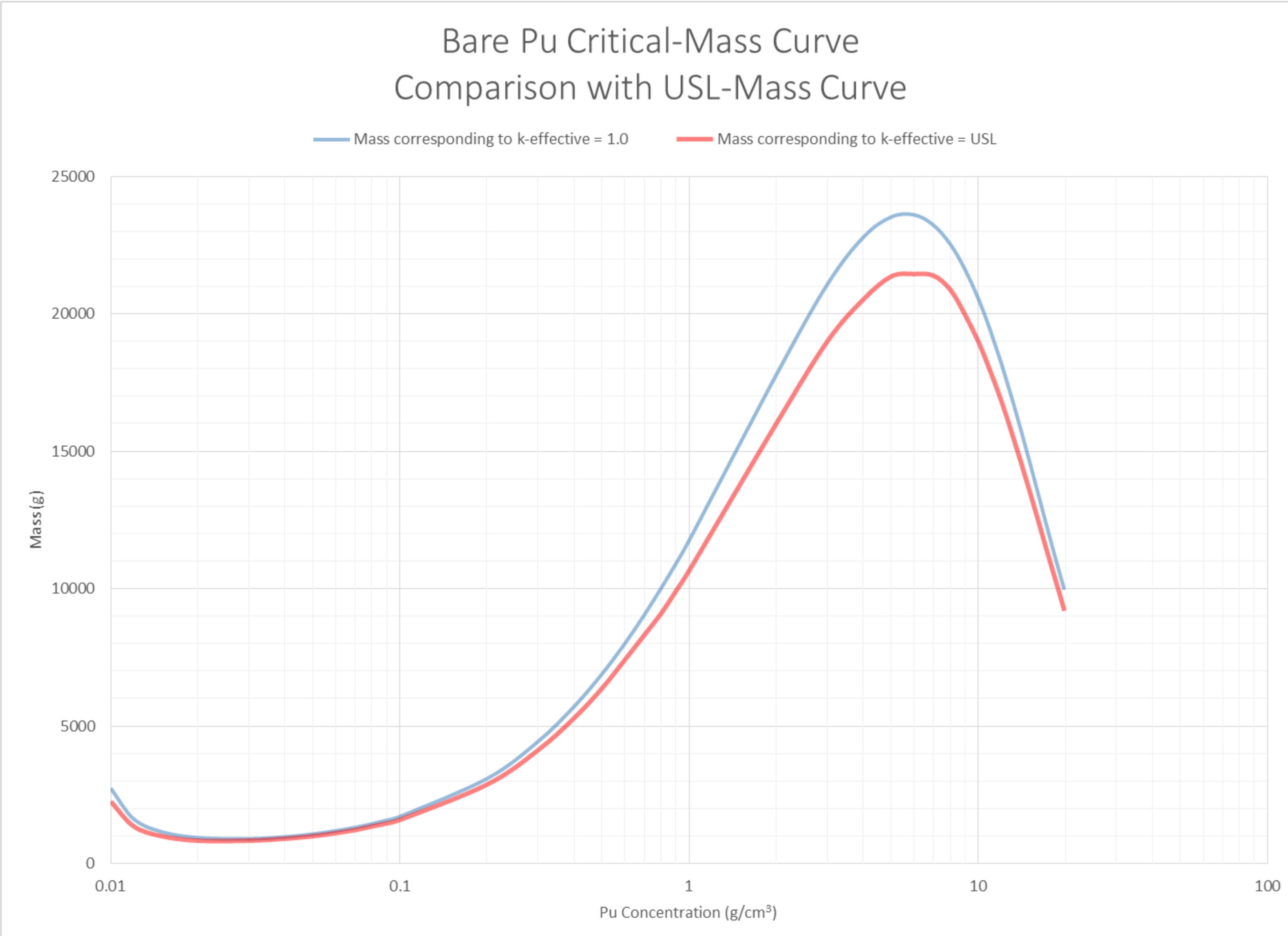


Example 7

–

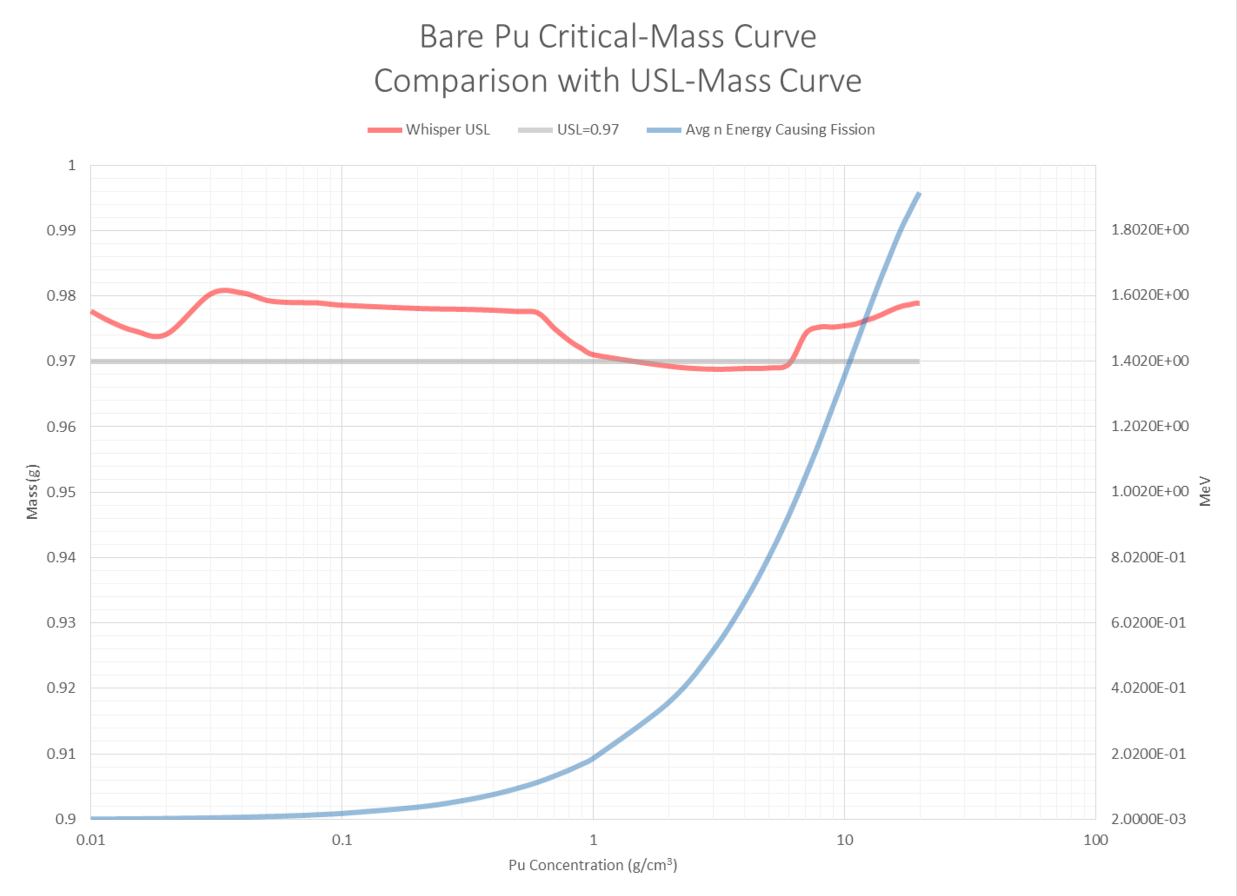
Critical Mass & USL Curves

Example 7: Critical-Mass and USL-Mass Curves



Example 7: Critical-Mass and USL-Mass Curves

[ANSI/ANS-8.24 7.2]
The validation applicability should not be so large that a subset of data with a high degree of similarity to the system or process would produce an upper subcritical limit that is lower than that determined for the entire set. This criterion is recommended to ensure that a subset of data that is closely related to the system or process is not nonconservatively masked by benchmarks that do not match the system as well.



THERMAL

- Average neutron energy causing fission: 0.00854 MeV
- % of fissions caused by neutrons: 96%; 3.5%; 0.5%
- Bias+bias uncertainty: 0.01306
- Nuclear data uncertainty: 0.00057
- USL = 0.98046

INTERMEDIATE

- Average neutron energy causing fission: 0.519 MeV
- % of fissions caused by neutrons: 18%; 55%; 27%
- Bias+bias uncertainty: 0.02197
- Nuclear data uncertainty: 0.00162
- USL = 0.96881

FAST

- Average neutron energy causing fission: 1.92 MeV
- % of fissions caused by neutrons: 0%; 2%; 98%
- Bias+bias uncertainty: 0.01419
- Nuclear data uncertainty: 0.00073
- USL = 0.97891

Monte Carlo Parameter Studies & Uncertainty Analyses With MCNP6

Outline

- Introduction
- mcnp_pstudy
- Examples
- Usage
  - Parameter definition
  - Parameter expansion
  - Constraints
  - Case setup & execution
  - Collecting & combining results
- Statistics
- Practical Examples from Criticality Safety
- Advanced Topics

Frequent Questions

How are calculated results affected by:

- **Nominal dimensions**
  - With minimum & maximum values ?
  - With as-built tolerances ?
  - With uncertainties ?
- **Material densities**
  - With uncertainties ?
- **Data issues**
  - Different cross-section sets ?
- **Stochastic materials**
  - Distribution of materials ?

Monte Carlo perturbation theory can handle the case of independent variations in material density, but does not apply to other cases.

Brute force approach:

Run many independent Monte Carlo calculations, varying the input parameters.

mcnp_pstudy

- **To simplify & streamline the setup, running, & analysis of Monte Carlo parameter studies & total uncertainty analyses, a new tool has been developed: mcnp_pstudy**

- **Control directives are inserted into a standard MCNP input file**
  - **Define lists of parameters to be substituted into the input file**
  - **Define parameters to be sampled from distributions & then substituted**
  - **Define arbitrary relations between parameters**
  - **Specify constraints on parameters, even in terms of other parameters**
  - **Specify repetitions of calculations**
  - **Combine parameters as outer-product for parameter studies**
  - **Combine parameters as inner-product for total uncertainty analysis**
- **Sets up separate calculations**
- **Submits or runs all jobs**
- **Collects results**

- **Completely automates the setup/running/collection for parameter studies & total uncertainty analyses**
  - Painless for users
  - 1 input file & run command can spawn 100s or 1000s of jobs
  - Fast & easy way to become the #1 user on a system

    (Added bonus: make lots of new friends in computer ops & program management.)
- **Ideal for Linux clusters & parallel ASC computers:**
  - Can run many independent concurrent jobs, serial or parallel
  - Faster turnaround: Easier to get many single-cpu jobs through the queues, rather than wait for scheduling a big parallel job
  - Clusters always have some idle nodes

mcnp_pstudy

- **mcnp_pstudy is written in *perl***
  - **640 lines of *perl* (plus 210 lines of comments)**
  - **Would have taken many thousands of lines of Fortran or C**
- **Portable to any computer system**
  - **Tested on Unix, Linux, Mac OS X, Windows**
  - **For Windows PCs, need to have *perl* installed**

    (ActivePerl is free at `activestate.com/activeperl`, easy to install)
- **Can be modified easily if needed**
  - **To add extra features**
  - **To accommodate local computer configuration**
    - **Node naming conventions for parallel cluster**
    - **Batch queueing system for cluster**
    - **Names & configuration of disk file systems (ie, local or shared)**
    - **Location of MCNP6 and MCNP6.mpi**

Examples

MCNP input for
simple Godiva calculation

MCNP input using *mcnp_pstudy*,
Run 3 different cases -
Each with a different radius

<pre>gdv
c
1 100 -18.74 -1   imp:n=1
2 0   1   imp:n=0
 	
1   so 8.741
 	
kcode 10000 1.0 15 115
ksrc 0 0 0
m100 92235 -94.73   92238 -5.27
prdmp 0 0 1 1 0</pre>
	<pre>gdv-A
	C @@@ RADIUS = 8.500 8.741 8.750
	1 100 -18.74 -1 imp:n=1
	2 0 1 imp:n=0
	
	1 so RADIUS
	
	kcode 10000 1.0 15 115
	ksrc 0 0 0
	m100 92235 -94.73 92238 -5.27
	prdmp 0 0 1 1 0</pre>

Basics

- Within an MCNP input file, all directives to mcnp_pstudy must begin with

```
c   @@@
```

- To continue a line, use "\" as the last character

```
c   @@@   XXX = 1    2    3    4    5   6   \
c   @@@         7    8    9   10
```

- Parameter definitions have the form

c   @@@   P =   *value or list*

c   @@@   P = (*arithmetic-expression*)

- Constraints have the form

c   @@@   CONSTRAINT = (*expression*   )

- Control directives have the form

c   @@@   OPTIONS =   *list-of-options*

Parameter Definition

- **Parameters**
  - Like C or Fortran variables
  - Start with a letter, contain only letters, integers, underscore
  - Case sensitive
  - Parameters are assigned values, either number(s) or string(s)
  - Examples: `R1, r1, U_density, U_den`
- **Single value**

```
c   @@@     P1   =   value
```

- **List of values**

```
c   @@@     P2   =   value1   value2   …   valueN
```

- **List of N random samples from Probability Densities:**
  - Uniform

```
c   @@@   P3   =   uniform   N    min    max
```

  - Normal

```
c   @@@   P4   =   normal   N    ave    dev
```

  - Lognormal

```
c   @@@   P5   =   lognormal   N    ave    dev
```

  - Beta

```
c   @@@   P6   =   beta N   a   b      [a,b are integers]
```

Parameter Definition

- **Arithmetic expression**

```
C   @@@     P5   = (   arithmetic-statement   )
```

  - Can use numbers & previously defined parameters
  - Can use arithmetic operators +, -, *, /, % (mod), ** (exponentiation)
  - Can use parentheses ( )
  - Can use functions: sin(), cos(), log(), exp(), int(), abs(), sqrt()
  - Can generate random number in (0,N): rand(N)
  - Can use rn_seed() to get odd seed for mcnp RN generator in $[1, 2^{48}-1]$
  - Must evaluate to a single value
  - Examples:

```
c   @@@    SEED = (   rn_seed()   )
c   @@@   FACT  = normal 1  1.0 .05
c   @@@   UDEN  = ( 18.74 * FACT )
c   @@@   URAD  = ( 8.741 * (18.74/UDEN)**.333333 )
```

- **Repetition (list of integers, 1..N)**

```
C   @@@     P6   =  repeat   N
```

Parameter Definition

- **Examples**

```
C   rod height in inches, for search
C   @@@  HROD = 5  10  15  20  25  30  35  40  45  50

C   nominal dimension, with uncertainty
C   @@@  X1 = normal  25   1.234  .002

C   dimension, with min & max
C   @@@  X2 = uniform 25   1.232  1.236

C   try different cross-sections
C   @@@  U235 = 92235.42c  92235.49c  92235.52c  \
C   @@@         92235.60c  92235.66c

C   different random number seeds (odd)
C   @@@  SEED = (  rn_seed()  )
```

Parameter Expansion

- After all parameters are defined, mcnp_pstudy expands them into sets to be used for each separate MCNP calculation
  - Outer product expansion: All possible combinations. Parameters specified first vary fastest.
  - Inner product expansion: Corresponding parameters in sequence. If not enough entries, last is repeated.

Example:

```
c @@@   A   =   1   2
c @@@   B   =   3   4
c @@@   C   =   5
```

Outer:

```
Case 1:      A=1,   B=3,   C=5
Case 2:      A=2,   B=3,   C=5
Case 3:      A=1,   B=4,   C=5
Case 4:      A=2,   B=4,   C=5
```

Inner:

```
Case 1:      A=1,   B=3,   C=5
Case 2:      A=2,   B=4,   C=5
```

Constraint Conditions

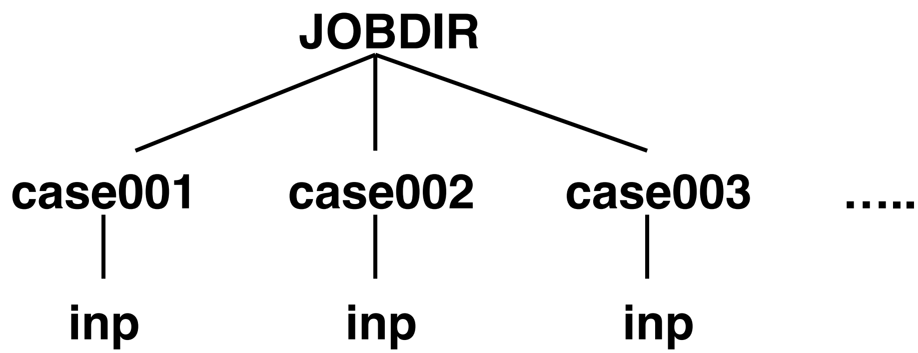
- After all parameters are defined & expanded, constraint conditions are evaluated
- Constraints involve comparison operators (>, <, >=, <=, ==, !=) or logical operators (&& (and), || (or), ! (not)), and may involve arithmetic or functions
- Constraints must evaluate to True or False
- If a any constraint is not met, the parameters for that case are discarded & re-evaluated until all of the constraints are satisfied

Example

```
C pick dimensions between min & max
C
C @@@  X1  =  uniform 1  3.9  4.1
C @@@  X2  =  uniform 1  5.9  6.1
C
C keep x1 & x2 if  x1+x2 <= 10.0,  otherwise reject & try again
C
C @@@  CONSTRAINT = ( X1 + X2  <=  10.0 )
```

Creating INP Files & Job Directories

- **Directory structure for MCNP5 jobs**



  - **Unix filesystem conventions followed**

    `JOBDIR/case001/inp,  JOBDIR/case002/inp, etc.`

- **Values of parameters are substitued into the original MCNP5 input file to create the input files for each case**
  - **Parameters substituted only when exact matches are found**
  - **Example: `UDEN` matches `UDEN`, and not `UDEN1`, `UDENS`, `uden`**

Job Options

- **Specifying options for running jobs**
  - Can be specified on the mcnp_pstudy command-line

    ```
    mcnp_pstudy  -inner    -setup -i inp01
    ```

  - Within the INP file

    ```
    c @@@ OPTIONS = -inner
    ```

- **Common options**

`-i str`	The INP filename is `str`, default = inp
`-jobdir str`	Use `str` as the name of the job directory
`-case str`	Use `str` as the name for case directories
`-mcnp_opts str`	Append `str` to the MCNP5 run command, may be a string such as `'o=outx tasks 4'`
`-bsub_opts str`	`str` is appended to the LSF bsub command
`-inner`	Inner product approach to case parameter substitution
`-outer`	Outer product approach to case parameter substitution
`-setup`	Create the cases & INP files for each
`-run`	Run the MCNP5 jobs on this computer
`-submit`	Submit the MCNP5 jobs using LSF bsub command
`-collect`	Collect results from the MCNP5 jobs

Running or Submitting Jobs

- Jobs can be run on the current system, or can be submitted to a batch queueing system (e.g., LSF)
- Tally results & K-effective can be collected when jobs finish

Examples:

```
bash:  mcnp_pstudy -inner -i inp01 -setup
bash:  mcnp_pstudy -inner -i inp01 -run
bash:  mcnp_pstudy -inner -i inp01 -collect
```

```
bash:  mcnp_pstudy -inner -i inp01 -setup -run -collect
```

```
bash:  mcnp_pstudy -inner -i inp01 -setup -submit
```

… wait till all jobs complete…

```
bash:  mcnp_pstudy -inner -i inp01   -collect
```

Creating Input Files ONLY

- To bypass the creation of job directories, and running/submitting problems:

  - A special command line option is available: -inponly

  - Invoking this option performs the parsing & setup of the input files for each case, but the resulting mcnp input files are placed in the current directory with default names of the form

    `inp_case001,   inp_case002, etc.`

  - Using `-case study01a  -inponly` would result in files with names

    `inp_study01a001,  inp_study01a002,  etc.`

  - Other options `-run, -submit` cannot be used if `-inponly` is present

  - The option `-whisper` can be used, and is equivalent to `-inponly`

- **Tally results & K-effective from separate cases can be combined using batch statistics:**

$$\overline{X} = \frac{1}{M} \cdot \sum_{k=1}^{M} X_k \qquad \sigma_{\overline{X}} = \sqrt{\frac{1}{M-1} \cdot \left[\frac{1}{M} \sum_{k=1}^{M} X_k^2 - \overline{X}^2 \right]}$$

where M is the number of cases & X_k is some tally or Keff for case k

- **Variance due to randomness in histories decreases as 1/M, but variance due to randomness in input parameters is constant**

$$\sigma_{\overline{X}}^2 \approx \sigma_{\overline{X},\ \text{Monte Carlo}}^2 + \sigma_{\overline{X},\ \text{Initial Conditions}}^2$$

Varies as 1/M **~ Constant**

Examples

mcnp

Vary the fuel density randomly & adjust radius for constant mass, for 50 cases

```
gdv-E
c vary fuel density - normal, 5%sd,
c adjust the radius to keep constant mass
c
c @@@ FACT= normal 50  1.0 .05
c @@@ UDEN= ( 18.74*FACT )
c @@@ URAD= ( 8.741*(18.74/UDEN)**.333333 )
c
1     100  -UDEN   -1   imp:n=1
2     0             1   imp:n=0

1     so  URAD

kcode 10000  1.0  15  115
ksrc  0. 0. 0.
m100  92235 -94.73   92238 -5.27
prdmp 0 0 1 1 0
```

Vary fuel density & mass independently, for 50 cases

```
gdv-F
c vary fuel radius - normal, 5%sd
c vary fuel density- normal, 5%sd
c
c @@@ OPTIONS = -inner
c
c @@@ DFACT = normal 50  1.0 .05
c @@@ UDEN = ( DFACT * 18.74 )
c
c @@@ UFACT = normal 50  1.0 .05
c @@@ URAD  = ( UFACT * 8.741 )
c
1     100  -UDEN   -1   imp:n=1
2     0             1   imp:n=0

1     so  URAD

kcode 10000  1.0  15  115
ksrc  0. 0. 0.
m100  92235 -94.73   92238 -5.27
prdmp 0 0 1 1 0
```

Table 1. Results from varying parameters in the Godiva problem

Problem	Description	K-effective	$\sigma_{K\text{-}eff}$
base	**Base case**, discard 15 initial cycles, retain 100 cycles with 10K histories/cycle, **1M total histories**	0.9970	**0.0005**
A	Repeat the base problem 50 times, **50M total histories**	0.9972	**0.0001**
B	**Vary the fuel density only**: sample from a normal distribution with 5% std.dev, **50M total histories**	0.9961	**0.0061**
C	**Vary the fuel radius only**: sample from a normal distribution with 5% std.dev, **50M total histories**	1.0057	**0.0051**
D	**Vary the enrichment only**, sample from a normal distribution with 5% std.dev, **50M total histories**	0.9890	**0.0027**
E	**Sample the fuel density from a normal distribution with 5% std.dev, and adjust the fuel radius to keep constant fuel mass, 50M total histories**	0.9966	**0.0042**
F	**Sample the fuel density from a normal distribution with 5% std.dev, and independently sample the radius from a normal distribution with 5% std.dev, 50M total histories**	1.0073	**0.0076**

Applications

- Parameter studies
  - Run a series of cases with different control rod positions
  - Run a series of cases with different soluble boron concentrations
  - Run a series of cases sampling certain dimensions from a Uniform or Normal probability density
  - Run a series of cases substituting different versions of a cross-section
- Total uncertainty analysis
  - Run a series of cases varying all input parameters according to their uncertainties
- Parallel processing using a "parallel jobs" approach
  - Running N separate jobs with 1 cpu each will be more efficient than running 1 job with N cpus
  - Eliminates queue waiting times while cpus are reserved
  - Take advantage of cheap Linux clusters
- Simulation of stochastic geometry
  - Run a series of cases with portions of geometry sampled randomly, with a different realization in each case

Conclusions

- **mcnp_pstudy works**
  - **In use regularly at LANL for a variety of real applications**
  - **Developed on Mac & PC, runs anywhere**
  - **Easy to customize, if you have special needs**

- **To get it:**
  - **Included with MCNP6 distribution**

FB Brown, JE Sweezy, RB Hayes, "Monte Carlo Parameter Studies and Uncertainty Analyses with MCNP5", PHYSOR-2004, Chicago, IL (April, 2004)

Examples

- **wval4:** **4.5 kg Pu Sphere, Ta-reflected with varying reflector thickness**
- **wval1:** **4.5 kg Pu Ingot, solid cylinder with varying H/D**
- **wval2:** **4.5 kg Pu Ring, hollow cylinder with varying H & R_{in}**

Example

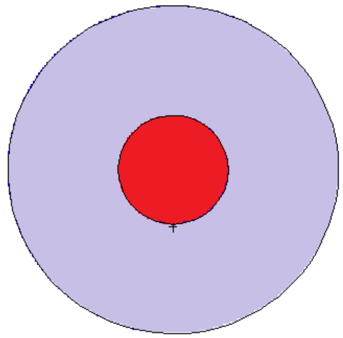
wval4,

4.5 kg Pu Sphere,

Ta-reflected

Example wval4: 4.5 kg Pu Sphere, Ta-reflected (1)

- 4.5 kg Pu-239 sphere
- Pu density = 19.8 g/cm^3
- Reflected radially with Ta



- Vary the Ta-reflector thickness over the range 0.$^+$ – 30. cm
  - Start with wval4.txt, input for thickness=7.62

    mcnp6 i=wval4.txt
  - Copy wval4.txt to wval4p.txt, then insert directives for mcnp_pstudy
    - Define list for thickness:

      ```
      c @@@ THICK =  0.01  5.  10.  15.  20.  25.  30.
      ```
    - For a given THICK, compute reflector Rin & Rout
    - Use parameters for dimensions & location of KSRC point
    - Run:

      ```
      mcnp_pstudy -i wval4.txt  -mcnp_opts 'tasks 4'  -setup
      ```
      …… examine files case*/inp
      ```
      mcnp_pstudy -i wval4.txt  -mcnp_opts 'tasks 4'  -run
      ```

Example wval4: 4.5 kg Pu Sphere, Ta-reflected (2)

```
wval4:  Study of Pu reflected with Ta
c
c  Pu mass     = 4500 g
c  Pu density = 19.8 g/cc
c  Pu volume  = 227.272727
c
c  reflector definition:
c    reflector thickness    = 7.62
c    reflector inner radius = 3.7857584
c    reflector outer radius = 11.405758
c
  1   4 -19.80  -1         imp:n=1
  2   1 -16.69  +1 -2      imp:n=1
 20   0         +2         imp:n=0

  1 so  3.7857584
  2 so  11.405758

 kcode 10000 1.0 50 250
 sdef pos=0 0 0  rad=d1
  si1  0 3.78
  sp1  -21 2
c
 m1  73180.80c 0.00012   73181.80c 0.99988
 m4  94239.80c 1
 prdmp 9e9 9e9 1 9e9
```

```
wval4p:  Study of Pu reflected with Ta
c
c  Pu mass     = 4500 g
c  Pu density = 19.8 g/cc
c  Pu volume  = 227.272727
c
c  vary reflector thickness from 0+ to 30 cm
c
c  @@@  THICK   = .01  5. 10. 15. 20. 25. 30.
c  @@@  R_INNER = 3.7857584
c  @@@  R_OUTER = ( R_INNER + THICK )
c
c  reflector definition:
c    reflector thickness    = THICK cm
c    reflector inner radius = R_INNER cm
c    reflector outer radius = R_OUTER cm
c
  1   4 -19.80  -1         imp:n=1
  2   1 -16.69  +1 -2      imp:n=1
 20   0         +2         imp:n=0

  1 so    R_INNER
  2 so    R_OUTER

 kcode 10000 1.0 50 250
 sdef pos=0 0 0 rad=d1
  si1  0  R_INNER
  sp1  -21 2
c
 m1  73180.80c 0.00012   73181.80c 0.99988
 m4  94239.80c 1
 prdmp 9e9 9e9 1 9e9
```

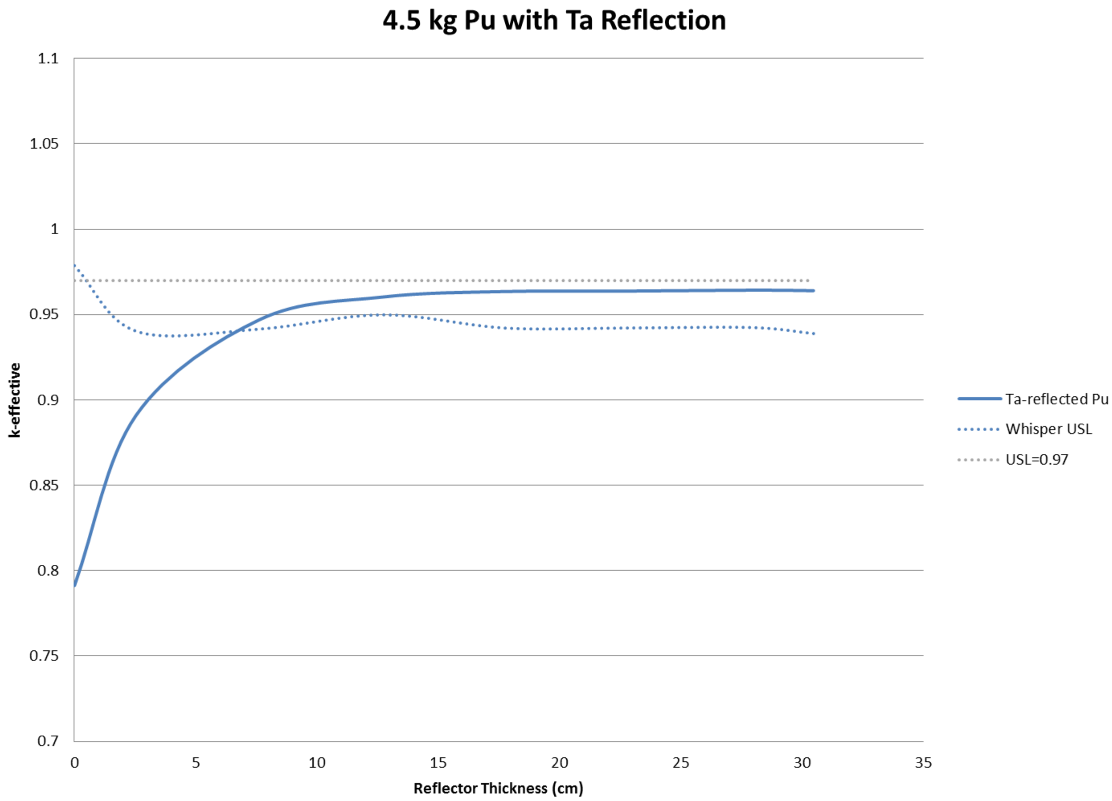
Example wval4: 4.5 kg Pu Sphere, Ta-reflected (3)

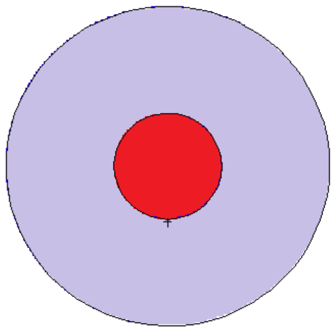
wval4, thick=7.62
mcnp6 i=wval4.txt

k = 0.94638 (41)

wval4p, varying thick
mcnp_pstudy -i wval4p.txt -setup -run

T=.01	case001	KEFF	7.91693E-01	KSIG	3.14948E-04
T=5.0	case002	KEFF	9.27157E-01	KSIG	4.47334E-04
T=10.	case003	KEFF	9.54775E-01	KSIG	4.11031E-04
T=15.	case004	KEFF	9.61644E-01	KSIG	4.34033E-04
T=20.	case005	KEFF	9.62867E-01	KSIG	4.37235E-04
T=25.	case006	KEFF	9.63899E-01	KSIG	4.04508E-04
T=30.	case007	KEFF	9.63160E-01	KSIG	4.27633E-04



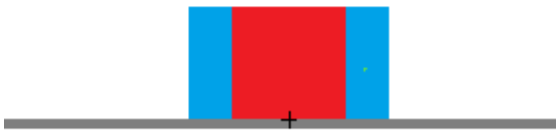


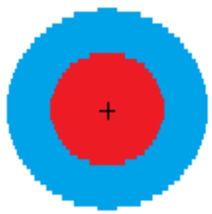
Example

wval1,
4.5 kg Pu Ingot,
varying H/D

Example wval1: 4.5 kg Pu Ingot, varying H/D (1)

- 4.5 kg Pu-239 right-circular cylinder
- Pu density = 19.86 g/cm^3
- Reflected radially with 1 inch of water
- Reflected on the bottom with ¼ inch steel





- Vary the height-to-diameter (H/D) over the range 0.5 – 3.0
  - Start with wval1.txt, input for H/D = 1

    mcnp6 i=wval1.txt

  - Copy wval1.txt to wval1p.txt, then insert directives for mcnp_pstudy
    - Define list for HD:

      ```
      c @@@ HD =  0.5  1.0  1.5  2.0  2.5  3.0
      ```

    - For a given H/D, compute Pu radius, then other dimensions

$$V = (\text{Pu mass})/(\text{Pu density})$$
$$V = H\pi R^2 = (\text{H/D}) \cdot 2\pi R^3$$
$$R = \left[V/2\pi(\text{H/D})\right]^{1/3}$$

    - Use parameters for dimensions & location of KSRC point

Example wval1: 4.5 kg Pu Ingot, varying H/D (2)

```
wval1:  4500 g Pu metal,  H/D = 1
c reflected 1 inch water radially,
c 0.25 in steel bottom
c
 1 1 -19.860000  -1            imp:n=1
11 3 -1.0        +1 -11        imp:n=1
14 6 -7.92       -30           imp:n=1
15 0             +11 +30 -20   imp:n=1
20 0 +20                       imp:n=0

 1  rcc  0 0 0       0 0 6.607662 3.303831
11  rcc  0 0 0       0 0 6.607662 5.843831
20  rcc  0 0 -2.54   0 0 91.44      91.44
30  rcc  0 0 -0.635  0 0 0.635      76.20

 kcode  10000 1.0 50 250
 ksrc   0 0 3.303831
 m1   94239.80c 1
 m3    1001.80c 0.66667    8016.80c 0.33333
 mt3   lwtr.20t
 m6   24050.80c 0.000757334
      24052.80c 0.014604423
      24053.80c 0.001656024
      24054.80c 0.000412220
      26054.80c 0.003469592
      26056.80c 0.054465174
      26057.80c 0.001257838
      26058.80c 0.000167395
      25055.80c 0.00174
      28058.80c 0.005255537
      28060.80c 0.002024423
      28061.80c 0.000088000
      28062.80c 0.000280583
      28064.80c 0.000071456
 prdmp 9e9 9e9 1 9e9
```

```
wval1p:  4500 g Pu metal, various H/D
c reflected 1 inch water radially,
c 0.25 in steel bottom
c
c   V = H pi R**2 = (H/D) 2pi R**3
c   R = (V/(2pi H/D)**1/3
c
c @@@  PI     = 3.141592654
c @@@  VOL_PU = ( 4500. / 19.86 )
c @@@  HD     = 0.5  1.0  1.5  2.0  2.5  3.0
c @@@  R_PU   = ( (VOL_PU/(2*PI*HD))**(1/3) )
c @@@  H_PU   = ( 2*R_PU*HD )
c @@@  R_H2O  = ( R_PU + 2.54 )
c @@@  KSRC_Z = ( H_PU * 0.5 )
c
c Pu cylinder:
c      mass        = 4500 g
c      density     = 19.86 g/cc
c      volume      = VOL_PU
c      radius Pu   = R_PU
c      height Pu   = H_PU
c      H/D         = HD
c
c H2O  outer radius = R_H2O
c
  1   1 -19.860000   -1            imp:n=1
 11   3 -1.0         +1 -11        imp:n=1
 14   6 -7.92        -30           imp:n=1
 15   0              +11 +30 -20   imp:n=1
 20   0              +20           imp:n=0

  1   rcc  0 0 0          0 0 H_PU   R_PU
 11   rcc  0 0 0          0 0 H_PU   R_H2O
 20   rcc  0 0 -2.540000  0 0 91.44  91.44
 30   rcc  0 0 -0.635000  0 0 0.635  76.20

 kcode  10000 1.0 50 250
 ksrc   0. 0. KSRC_Z
c
.................... etc.
```

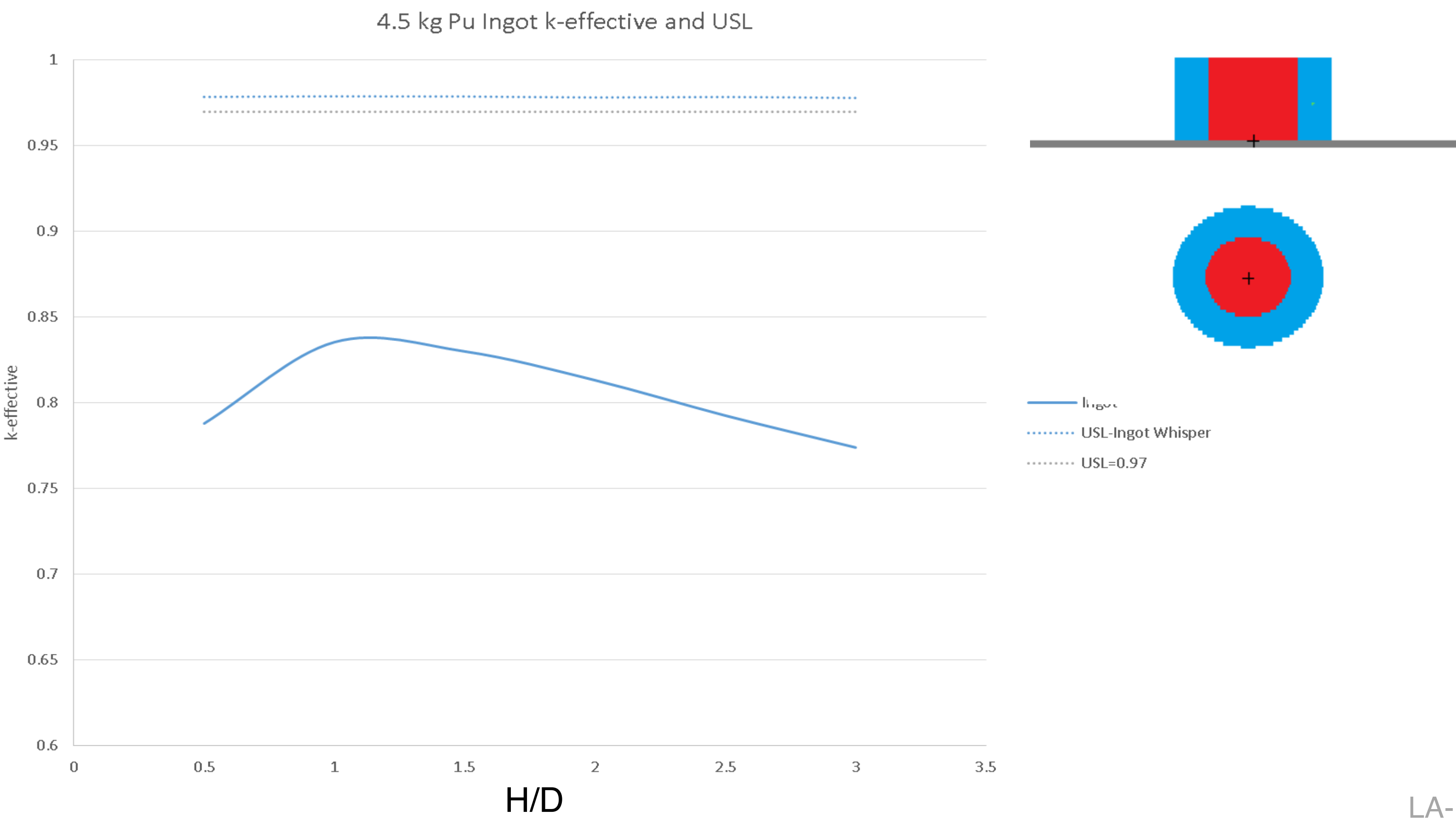
Example wval1: 4.5 kg Pu Ingot, varying H/D (3)

wval1, H/D = 1
 mcnp6 i=wval1.txt

k = 0.83491 (41)

wval1p, varying H/D
 mcnp_pstudy -i wval1p.txt -setup -run

HD=0.5	case001	KEFF	7.87229E-01	KSIG	4.09191E-04
HD=1.0	case002	KEFF	8.34430E-01	KSIG	4.20175E-04
HD=1.5	case003	KEFF	8.29652E-01	KSIG	4.19130E-04
HD=2.0	case004	KEFF	8.11958E-01	KSIG	4.18723E-04
HD=2.5	case005	KEFF	7.93676E-01	KSIG	4.63720E-04
HD=3.0	case006	KEFF	7.73434E-01	KSIG	4.19664E-04



Example

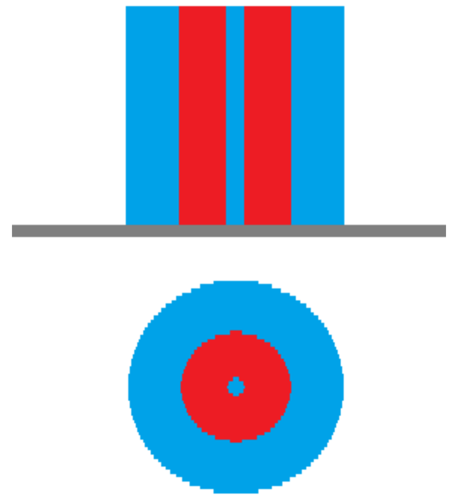
wval2,

4.5 kg Pu Annulus,

varying H & R_{in}

Example wval2: 4.5 kg Pu Annulus, varying H & R_{in} (1)

- **4.5 kg Pu-239 right-circular cylinder, hollow**
- **Pu density = 19.86 g/cm^3**
- **Reflected radially with 1 inch of water**
- **Reflected on the bottom with ¼ inch steel**



- **Set the height to be same as solid cylinder with height-to-diameter (H/D) = 1.0, 2.0, 3.0**
- **For given height, vary inner radius over 0$^+$ - 2 cm**
  - **Start with wval2.txt input**

    mcnp6 i=wval2.txt

  - **Copy wval2.txt to wval2p.txt, then insert directives for mcnp_pstudy**
    - Define list for solid HD:

      ```
      c @@@ HD =  1.0  2.0  3.0
      ```

    - For a given H/D, compute Pu height
    - Define list for inner radius RIN_PU

      ```
      c @@@ RIN_PU =  0.001  0.5  1.0  2.0
      ```

    - Then other dimensions & source

Solid cylinder

$$V = \text{(Pu mass)}/\text{(Pu density)}$$

$$V = H\pi R^2 = \text{(H/D)} \cdot 2\pi R^3$$

$$H = \left[4V\text{(H/D)}^2/\pi\right]^{1/3}$$

Hollow cylinder

$$V = H\pi(R_{out}^2 - R_{in}^2)$$

$$R_{out} = \left[R_{in}^2 + V/\pi H\right]^{1/2}$$

Example wval2: 4.5 kg Pu Annulus, varying H & R_{in} (2)

mcnp

```
wval2: 4500 g Pu metal ring, fixed Rin
  1    3 -1.0            -1            imp:n=1
  2    1 -19.860000      +1 -2         imp:n=1
 11    3 -1.0            +2 -11        imp:n=1
 14    6 -7.92           -30           imp:n=1
 15    0                 +11 +30 -20   imp:n=1
 20    0                 +20           imp:n=0

  1 rcc   0 0 0          0 0  6.608    0.100000
  2 rcc   0 0 0          0 0  6.608    3.305259
 11 rcc   0 0 0          0 0  6.608    5.845259
 20 rcc   0 0 -2.540     0 0 91.44    91.44
 30 rcc   0 0 -0.635     0 0 0.635    76.20

kcode   10000 1.0 50 250
sdef pos=0 0 0   rad=d1 axs=0 0 1   ext=d2
 si1   0.100   3.305259
 sp1   -21 1
 si2   0.0    6.60800
 sp2   0    1
m1    94239.80c 1
m3    1001.80c 0.66667    8016.80c 0.33333
mt3   lwtr.20t
m6    24050.80c 0.000757334
      24052.80c 0.014604423
      24053.80c 0.001656024
      24054.80c 0.000412220
      26054.80c 0.003469592
      26056.80c 0.054465174
      26057.80c 0.001257838
      26058.80c 0.000167395
      25055.80c 0.00174
      28058.80c 0.005255537
      28060.80c 0.002024423
      28061.80c 0.000088000
      28062.80c 0.000280583
      28064.80c 0.000071456
prdmp 9e9 9e9 1 9e9
```

```
wval2p: 4500 g Pu metal ring, various H & Rin
c
c @@@  PI      = 3.141592654
c @@@  VOL_PU = ( 4500. / 19.86 )
c      Pu mass     = 4500 g
c      Pu density = 19.86 g/cc
c      Pu volume  = VOL_PU
c
c set height to match ingot with various H/D
c @@@  HD      =  1.0  2.0  3.0
c @@@  HEIGHT = ( (4*VOL_PU*(HD**2)/PI)**(1/3) )
c
c for hollow cylinder:
c   use same height as for solid ingot
c   set various inner radii
c   set Rout for given height, mass, Rin
c @@@  RIN_PU = .001  0.5  1.0  2.0
c @@@  ROUT_PU=(sqrt(RIN_PU**2+VOL_PU/(PI*HEIGHT)))
c @@@  ROUT_H2O = ( OUTER_PU + 2.54 )
c
  1   3 -1.0           -1            imp:n=1
  2   1 -19.860000     +1 -2         imp:n=1
 11   3 -1.0           +2 -11        imp:n=1
 14   6 -7.92          -30           imp:n=1
 15   0                +11 +30 -20   imp:n=1
 20   0                +20           imp:n=0

  1   rcc    0 0 0          0 0  HEIGHT  RIN_PU
  2   rcc    0 0 0          0 0  HEIGHT  ROUT_PU
 11   rcc    0 0 0          0 0  HEIGHT  ROUT_H2O
 20   rcc    0 0 -2.540     0 0 91.44    91.44
 30   rcc    0 0 -0.635     0 0 0.635    76.20

kcode   10000 1.0 50 250
sdef   pos= 0. 0. 0.   rad=d1  axs=0 0 1  ext=d2
 si1   RIN_PU  ROUT_PU
 sp1   -21 1
 si2   0 HEIGHT
 sp2   0  1
…………… etc.
```

Example wval2: 4.5 kg Pu Annulus, varying H & R_{in} (3)

wval2

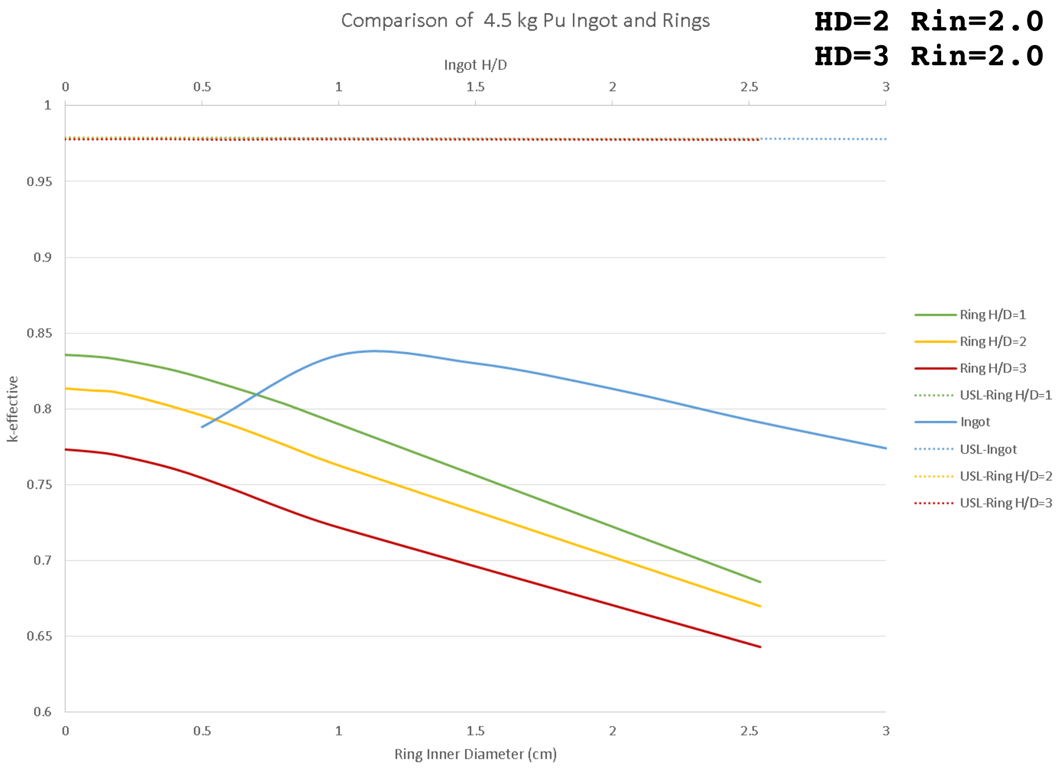
mcnp6 i=wval2.txt

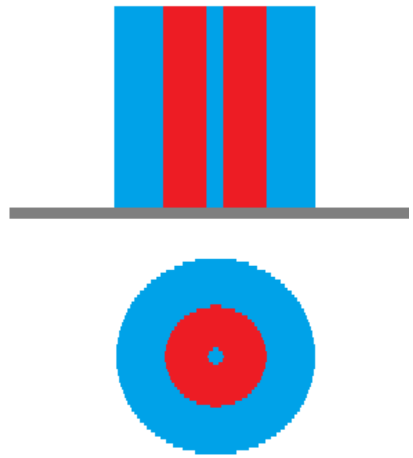
k = 0.83413 (42)

wval2p, varying H & R_{in}

mcnp_pstudy -i wval2p.txt -setup -run

```
HD=1 Rin=.001 case001 KEFF    8.34752E-01  4.35668E-04
HD=2 Rin=.001 case002 KEFF    8.12612E-01  4.09516E-04
HD=3 Rin=.001 case003 KEFF    7.72725E-01  3.82627E-04
HD=1 Rin=0.5  case004 KEFF    8.20432E-01  4.01135E-04
HD=2 Rin=0.5  case005 KEFF    7.95375E-01  4.60388E-04
HD=3 Rin=0.5  case006 KEFF    7.54174E-01  3.96580E-04
HD=1 Rin=1.0  case007 KEFF    7.88497E-01  3.95026E-04
HD=2 Rin=1.0  case008 KEFF    7.62394E-01  3.90299E-04
HD=3 Rin=1.0  case009 KEFF    7.20810E-01  4.27354E-04
HD=1 Rin=2.0  case010 KEFF    7.21523E-01  4.02775E-04
HD=2 Rin=2.0  case011 KEFF    6.97954E-01  4.88269E-04
HD=3 Rin=2.0  case012 KEFF    6.64037E-01  4.88326E-04
```





Advanced Topics

Tied parameters

Concurrent jobs

Parameter Expansion (1)

- **Standard inner & outer schemes for determining job parameters**

Example:

```
c @@@  A  =  1  2
c @@@  B  =  3  4
c @@@  C  =  5  6
c @@@  D  =  7  8
c @@@  E  =  9
```

```
Outer:    all combinations, 16 cases
          {1,3,5,7,9}, {2,3,5,7,9}, {1,4,5,7,9}, {2,4,5,7,9},
          {1,3,6,7,9}, {2,3,6,7,9}, {1,4,6,7,9}, {2,4,6,7,9},
          {1,3,5,8,9}, {2,3,5,8,9}, {1,4,5,8,9}, {2,4,5,8,9},
          {1,3,6,8,9}, {2,3,6,8,9}, {1,4,6,8,9}, {2,4,6,8,9},

Inner:    2 cases
          {1,3,5,7, 9},  {2,4,6,8, 9}
```

- **The inner & outer schemes for determining job parameters can be modified**
  - Often desirable to deal with groups of parameters that are varied
  - 2 or more parameters can be "tied" together, to vary in an inner manner
  - Tied parameter lists must have the same lengths

Parameter Expansion (2)

These examples assume that the -outer option is in effect for all parameter combinations

Example:

```
c @@@  tied = A B
c @@@  A   =   1   2
c @@@  B   =   3   4
c @@@  C   =   5   6
c @@@  D   =   7   8
c @@@  E   =   9
```

```
Cases, {A,B,C,D,E}:
 {1,3, 5, 7, 9},  {1,3, 6, 7, 9},
 {1,3, 5, 8, 9},  {1,3, 6, 8, 9},
 {2,4, 5, 7, 9},  {2,4, 6, 7, 9},
 {2,4, 5, 8, 9},  {2,4, 6, 8, 9}
```

Example:

```
c @@@  tied = A B C
c @@@  A   =   1   2
c @@@  B   =   3   4
c @@@  C   =   5   6
c @@@  D   =   7   8
c @@@  E   =   9
```

```
Cases, {A,B,C,D,E}:
 {1,3,5, 7,9},  {1,3,5, 8,9},
 {2,4,6, 7,9},  {2,4,6, 8,9}
```

Example:

```
c @@@  tied = A B
c @@@  A   =   1   2
c @@@  B   =   3   4
c @@@  tied = C D
c @@@  C   =   5   6
c @@@  D   =   7   8
c @@@  E   =   9
```

```
Cases, {A,B,C,D,E}:
 {1,3, 5,7, 9},  {1,3, 6,8, 9},
 {2,4, 5,7, 9},  {2,4, 6,8, 9}
```

Example:

```
c @@@  tied = A B C D
c @@@  A   =   1   2
c @@@  B   =   3   4
c @@@  C   =   5   6
c @@@  D   =   7   8
c @@@  E   =   9
```

```
Cases, {A,B,C,D,E}:
 {1,3,5,7, 9},  {2,4,6,8, 9}
```

Parameter Expansion (3)

The **-inner** & **-outer** options can be varied for different parameters, and mixed with **tied** parameters

Example:

```
c @@@  options = -inner
c @@@  A  =   1   2
c @@@  B  =   3   4
c @@@  C  =   5   6
c @@@  D  =   7   8
c @@@  E  =   9
```

```
Cases:
 {1,3,5,7, 9},  {2,4,6,8, 9},
```

Example:

```
c @@@ options = -inner
c @@@  A  =   1   2
c @@@  B  =   3   4
c @@@ options = -outer
c @@@  C  =   5   6
c @@@  D  =   7   8
c @@@  E  =   9
```

```
Cases:
 {1,3, 5, 7, 9},  {1,3, 6, 7, 9},
 {1,3, 5, 8, 9},  {1,3, 6, 8, 9},
 {2,4, 5, 7, 9},  {2,4, 6, 7, 9},
 {2,4, 5, 8, 9},  {2,4, 6, 8, 9}
```

Example:

```
c @@@ options = -outer
c @@@  tied = A B
c @@@  A  =   1   2
c @@@  B  =   3   4
c @@@  tied = C D
c @@@  C  =   5   6
c @@@  D  =   7   8
c @@@  E  =   9
```

```
Cases:
 {1,3, 5,7, 9},  {1,3, 6,8, 9},
 {2,4, 5,7, 9},  {2,4, 6,8, 9}
```

Example:

```
c @@@  tied = A B C D
c @@@  A  =   1   2
c @@@  B  =   3   4
c @@@  C  =   5   6
c @@@  D  =   7   8
c @@@  E  =   9
```

```
Cases:
 {1,3,5,7, 9},  {2,4,6,8, 9}
```

Concurrent Jobs (1)

- **By default, jobs for the different cases are run sequentially**
  - For **–run**: jobs for each case are run on the current computer, sequentially (one-at-a-time)
  - For **–submit**: separate batch jobs are submitted for each case,
  - For either **–run** or **-submit**, multiple threads can be used for the mcnp6 runs in each case, by using the option `–mcnp_opts 'tasks 8'`
- **For Linux & Mac systems, not Windows:**
  - Multiple concurrent cases can be run, even when threads are used
  - The `–ppn n` option specifies the number of processes per node (ie, cases to be run concurrently)
- **Examples:**
  - On a system with 24 hyperthreads, could run 6 cases at a time with 4 threads each:

```
mcnp_pstudy –i inp.txt -mcnp_opts 'tasks 4' -ppn 6 -setup –run
```

  - For a cluster with 16 cores/node, can submit jobs with 16 cases each:

```
mcnp_pstudy –i inp.txt -ppn 16  -setup –submit
```

References

References for Whisper & MCNP6 (1)

All references are available at URL: mcnp.lanl.gov → Recent Publications → Whisper – NCS Validation

Abstract

- Whisper - abstract from LANL TeamForge Tracker system, Artifact artf36407 (2015)

Theory

- B.C. Kiedrowski, F.B. Brown, et al., "Whisper: Sensitivity/Uncertainty-Based Computational Methods and Software for Determining Baseline Upper Subcritical Limits", Nuc. Sci. Eng. Sept. 2015, LA-UR-14-26558 (2014)
- B.C. Kiedrowski, "Methodology for Sensitivity and Uncertainty-Based Criticality Safety Validation", LA-UR-14-23202 (2014)
- F.B. Brown, M.E. Rising, J.L. Alwin, "Lecture Notes on Criticality Safety Validation Using MCNP & Whisper", LA-UR-16-21659 (2016)

User Manual

- B.C. Kiedrowski, "User Manual for Whisper (v1.0.0), Software for Sensitivity- and Uncertainty-Based Nuclear Criticality Safety Validation", LA-UR-14-26436 (2014)
- B.C. Kiedrowski, "MCNP6.1 k-Eigenvalue Sensitivity Capability: A Users Guide", LA- UR-13-22251 (2013)

Application

- B.C. Kiedrowski, et al., "Validation of MCNP6.1 for Criticality Safety of Pu-Metal, - Solution, and -Oxide Systems", LA-UR-14-23352 (2014)

References for Whisper & MCNP6 (2)

Software Quality Assurance

- R.F. Sartor, F.B. Brown, "Whisper Program Suite Validation and Verification Report", LA-UR-15-23972 (2015-05-28)
- R.F. Sartor, F.B. Brown, "Whisper Source Code Inspection Report", LA-UR-15-23986 (2015-05-28)
- R.F. Sartor, B.A. Greenfield, F.B. Brown, "MCNP6 Criticality Calculations Verification and Validation Report", LA-UR-15-23266 (2015-04-30)
- Monte Carlo Codes Group (XCP-3), "Whisper - Software for Sensitivity-Uncertainty-based Nuclear Criticality Safety Validation", LANL TeamForge Tracker system, Artifact artf36407 (2015)
- Monte Carlo Codes Group (XCP-3), WHISPER module in LANL TeamForge GIT repository (2015)
- Monte Carlo Codes Group (XCP-3), MCNP6 module in LANL TeamForge GIT repository
- Monte Carlo Codes Group (XCP-3), "MCNP Process Documents", LANL Teamforge wiki for MCNP
- Monte Carlo Codes Group (XCP-3), "Software Quality Assurance", LANL Teamforge wiki for MCNP, P1040-rev9 requirements

Recent MCNP6 & ENDF/B-VII.1 Verification/Validation

- F.B. Brown, "MCNP6 Optimization and Testing for Criticality Safety Calculations", Trans. ANS 111, LA-UR-15-20422 (2015)
- Monte Carlo Codes XCP-3, LANL F.B. Brown, B.C. Kiedrowski, J.S. Bull, "Verification of MCNP6.1 and MCNP6.1.1 for Criticality Safety Applications", LA-UR-14-22480 (2014).
- F.B. Brown, B.C. Kiedrowski, J.S. Bull, "Verification of MCNP5-1.60 and MCNP6.1 for Criticality Safety Applications", LA-UR-13-22196 (2013).
- L.J. Cox, S.D. Matthews, "MCNP6 Release 1.0: Creating and Testing the Code Distribution", LA-UR-13-24008 (2013)

References for Whisper & MCNP6 (3)

Recent MCNP6 & ENDF/B-VII.1 Verification/Validation (cont'd)

- R.D. Mosteller, F.B. Brown, B.C. Kiedrowski, "An Expanded Criticality Validation Suite for MCNP", LA-UR-11-00240 (2011).
- R.D. Mosteller, "An Expanded Criticality Validation Suite for MCNP", LA-UR-10-06230 (2010).
- R.C. Little, "V&V of MCNP and Data Libraries at Los Alamos", LA-UR-12-26307 (2012) A. Sood, R.A. Forster, D.K. Parsons, "Analytic Benchmark Test Set for Criticality Code
- Verification", LA-13511 and LA-UR-01-3082 (2001)
  XCP Data Team, "LANL Data Testing Support for ENDF/B-VII.1", LA-UR-12-20002 LA-UR-12-20002 (2012)

General References on Adjoints, Perturbation, and Sensitivity Analysis

- B.C. Kiedrowski, F.B. Brown, et al., "MCNP Sensitivity/Uncertainty Accomplishments for the Nuclear Criticality Safety Program", Trans. Am. Nuc. Soc 111, Nov 2014, LA-UR-14- 24458 (2014)
- B.C. Kiedrowski, "Adjoint Weighting Methods Applied to Monte Carlo Simulations of Applications and Experiments in Nuclear Criticality" seminar at University of Michigan, March 2014, LA-UR-14-21608 (2014)
- B.C. Kiedrowski, "MCNP Continuous-Energy Sensitivity and Uncertainty Progress and Application", Presentation at DOE-NNSA Nuclear Criticality Safety Program Technical Review, 26-27 March 2014, LA-UR-14-21919 (2014)
- B.C. Kiedrowski, "Application of Covariance Data in Nuclear Criticality", Nuclear Data Covariance Workshop, April 28 - May 1, Santa Fe, NM, LA-UR-14-22972 (2014)

References for Whisper & MCNP6 (4)

General References on Adjoints, Perturbation, and Sensitivity Analysis (cont'd)

- B.C. Kiedrowski & F.B. Brown, "Applications of Adjoint-Based Techniques in Continuous-Energy Monte Carlo Criticality Calculations", Supercomputing in Nuclear Applications and Monte Carlo 2013, Paris, Oct 27-31, LA-UR-13-27002 (2013)
- B.C. Kiedrowski, "Importance of Scattering Distributions on Criticality", ANS NCSD- 2013, Wilmington, NC, Sept 29 - Oct 1, LA-UR-13-24254 (2013).
- B.C. Kiedrowski, A.C. Kahler, M.E. Rising, "Status of MCNP Sensitivity/Uncertainty Capabilities for Criticality", ANS NCSD-2013, Wilmington, NC, Sept 29 - Oct 1, LA-UR- 13-24090 (2013)
- B.C. Kiedrowski, "K-Eigenvalue Sensitivity Coefficients to Legendre Scattering Moments", ANS 2013 Winter Meeting, LANL report LA-UR-13-22431 (2013)
- B.C. Kiedrowski, F.B. Brown, "Applications of Adjoint-Based Techniques in Continuous- Energy Monte Carlo Criticality Calculations", submitted to SNA+MC-2013, Paris, France [also LA-UR-12-26436] (2012)
- B.C. Kiedrowski, F.B. Brown, "K-Eigenvalue Sensitivities of Secondary Distributions of Continuous-Energy Data," M&C 2013, Sun Valley, ID, May 2013, report LA-UR-12- 25966, talk LA-UR-13-23208 (2013)
- B.C. Kiedrowski, F.B. Brown, "Methodology, Verification, and Performance of the Continuous-Energy Nuclear Data Sensitivity Capability in MCNP6," M&C 2013, Sun Valley, ID, May 2013, report LA-UR-12-25947, talk LA-UR-13-23199 (2012)
- B.C. Kiedrowski, F.B. Brown, "MCNP6 Nuclear Data Sensitivity Capability: Current Status and Future Prospects", presentation at MCNP/ENDF/NJOY Workshop, 2012-10- 30, LANL, LA-UR-12-25560 (2012)
- B.C. Kiedrowski, F.B. Brown, "Nuclear Data Sensitivities in Fast Critical Assemblies", presentation at NECDC-2012, LA-UR-12-25144 (2012)

References for Whisper & MCNP6 (5)

General References on Adjoints, Perturbation, and Sensitivity Analysis (cont'd)

- B.C. Kiedrowski, F.B. Brown, "Adjoint-Based k-Eigenvalue Sensitivity Coefficients to Nuclear Data Using Continuous-Energy Monte Carlo", submitted to Nuclear Science & Engineering [also LA-UR-12-22089] (2012)
- B.C. Kiedrowski, "MCNP6 Results for the Phase III Sensitivity Benchmark of the OCED/NEA Expert Group on Uncertainty Analysis for Criticality Safety Assessment", LA-UR-12-21048 (2012)
- B.C. Kiedrowski & F.B. Brown, "Continuous-Energy Sensitivity Coefficient Capability in MCNP6", Trans. Am. Nuc. Soc. 107, LA-UR-12-21010,
  presentation at 2012 ANS Winter Meeting, San Diego, CA, LA-UR-12-25949 (2012)
- B.C. Kiedrowski & F.B. Brown, "Comparison Of The Monte Carlo Adjoint-Weighted And Differential Operator Perturbation Methods", SNA+MC-2010, Tokyo, Oct 17-20, LA-UR- 10-05215 (2010)
- B.C. Kiedrowski, J.A. Favorite, & F.B. Brown, "Verification of K-eigenvalue Sensitivity Coefficient Calculations Using Adjoint-Weighted Perturbation Theory in MCNP", Trans. Am. Nuc. Soc, 103, Nov 2010, LA-UR-10-04285 (2010)
- B.C. Kiedrowski, F.B. Brown, & P. Wilson, "Adjoint-Weighted Tallies for k-Eigenvalue Calculations with Continuous-Energy Monte Carlo", Nucl. Sci. Eng. 168, 38-50, 2011, LA-UR-10-01824, (2010).
- B.C. Kiedrowski & F.B. Brown, "Adjoint-Weighting for Critical Systems with Continuous Energy Monte Carlo", ANS NCSD-2009, Richland, WA, Sept 13-17, paper LA-UR-09- 2594, presentation LA-UR-09-5624 (2009)

LA-UR-16-23533
Approved for public release; distribution is unlimited.

Title: **Lecture Notes on Sensitivity-Uncertainty Based Nuclear Criticality Safety Validation**

Author(s): Alwin, Jennifer
Brown, Forrest
Rising, Michael

Issued: 2016-05-02