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

Nuclear Criticality Safety Validation – I
- Introduction - background, standards, definitions, USL, calculational margin, margin of subcriticality
- Selection of benchmarks
- Bias & bias uncertainty
- Sensitivity-uncertainty analysis
- Validation approaches & technical review

Nuclear Criticality Safety Validation – II, Using MCNP & Whisper
- Whisper – overview, methodology, status
- Sensitivity profiles, covariance data, correlation coefficients
- USLs & validation
- Using Whisper – whisper_mcnp, whisper_usl, Whisper.out

Nuclear Criticality Safety Validation – III, Whisper Examples for NCS Analysts
Pyrochemical Processing
1. Typical model: ingot
2. Geometry: Annular
3. Material: Pu-NaCl
4. Reflection: Ta
5. Moderation: Oil

General Studies
6. Revisit a Practical Application of the Single-Parameter-Subcritical-Mass Limit for Pu Metal with Whisper
7. Critical-mass curves and USL-mass curves comparison

Appendix: Monte Carlo parameter studies & uncertainty analyses with MCNP6

References
(blank)
Nuclear Criticality Safety

Validation - I
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 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-1158-2010, Self-Assessment Standard for DOE Contractor Criticality Safety Programs
- 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-8.3-2003, Criticality Accident Alarm System
- ANSI/ANS-8.5-1996(R2007), Use of Borosilicate-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 8.14-2004, Use of Soluble Neutron Absorbers in Nuclear Facilities Outside Reactors
- ANSI/ANS 8.20-1991(R2005), Nuclear Criticality Safety Training


- ANSI/ANS 8.26-2007, Criticality Safety Engineer Training and Qualification Program
# MCNP Verification & Validation Suites

## Verification Suites

<table>
<thead>
<tr>
<th>Suite</th>
<th>Problems / Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>REGRESSION</strong></td>
<td>- 161 code test problems&lt;br&gt;- Run by developers for QA checking (100s of times per day)</td>
</tr>
</tbody>
</table>
| **VERIFICATION KEFF**      | - 75 analytic benchmarks (0-D and 1-D)<br>- Exact solutions for $k_{\text{eff}}$
                                | - Past – multigroup, New – continuous-energy                                          |
| **VERIFICATION GENTIME**   | - 10 benchmarks (analytic or comparisons to Partisan) for reactor kinetics parameters |
| **KOBAYASHI**              | - 6 void & duct streaming problems, with point detectors, exact solutions            |
| **Ganapol Benchmarks**     | [in progress] - Exact, semi-analytic benchmark problems<br>- Fixed source, not criticality |
| **Gonzales Benchmark**     | [in progress] - Exact analytic benchmark with elastic scatter, including free-gas scatter |

## Validation Suites

<table>
<thead>
<tr>
<th>Suite</th>
<th>Problems / Features</th>
</tr>
</thead>
</table>
| **VALIDATION_CRITICALITY**| - 31 ICSBEP Cases<br>- Too small a suite for serious V&V<br>- 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<br>- Broad-range validation, for developers                      |
| **VALIDATION_CRIT_WHISPER**| - 1101 ICSBEP Cases<br>- 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)
• 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)

- **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.
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.
To consider a simulated system subcritical, the computed $k_{eff}$ must be less than the Upper Subcritical Limit (USL):

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

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

[additional 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.
• **Hypothetical bias curve**
  – Selected experiments with Pu metal and water mixtures
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

- **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, positive biases* (calculated value is higher than experiment value) are not credited 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 $k_{\text{eff}}$.

- 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 $k_{\text{eff}}$ due to a fractional change in the nuclear data value.
    - Sensitivity analysis improves understanding of what is important for $k_{\text{eff}}$ determination
Uncertainty analysis combines sensitivity data with nuclear data uncertainty information to yield:

- Uncertainty in $k_{\text{eff}}$ 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)
  – $K_{\text{eff}}$ 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)

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


- 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
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 \( \text{MOS}_{\text{data+code}} \) of 2-5%. Quantitative, physics-based, repeatable \( \text{MOS}_{\text{data+code}} \) 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
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, $S(\text{energy, reaction, isotope})$, $S = (dk/k) / (d\sigma/\sigma)$

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

\[ S_{k,x} = \frac{d k / k}{d x / x} = \frac{x}{k} \frac{d k}{d x} \]

• This may be expressed using perturbation theory:

\[ S_{k,x} = \frac{x}{k} \frac{d k}{d x} = - \frac{\left\langle \psi^\dagger, \left( \Sigma_x - S_x - k^{-1} F_x \right) \psi \right\rangle}{\left\langle \psi^\dagger, k^{-1} F \psi \right\rangle} \]

  - 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
  - \((\text{Score}) \times (\text{importance})\) is tallied for adjoint-weighted results
**Sensitivity Profiles - Examples**

**U-238:** total cross-section sensitivity

OECD/NEA UACSA Benchmark Phase III.1

![Graph showing sensitivity of total cross-section for U-238 across neutron energies.](image1)

**H-1:** elastic scattering cross-section sensitivity

OECD/NEA UACSA Benchmark Phase III.1

![Graph showing sensitivity of elastic scattering for H-1 across neutron energies.](image2)

**Pu-239:** fission chi(E) sensitivity

OECD/NEA UACSA Benchmark Phase III.1

![Graph showing sensitivity of fission chi for Pu-239 across neutron energies.](image3)

**Cu-63:** Elastic Scattering Sensitivity
Copper-Reflected Zeus experiment:

![Graph showing sensitivity of Cu-63 elastic scattering across incident energies.](image4)
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 $K_{eff}$ to a particular reaction type & energy bin is:
    \[ S_{k,x} = \frac{\Delta k}{k} = \frac{x}{k} \frac{dk}{dx} \]
    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 $S_A$

\[ \text{Isotopes } \Rightarrow \]

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

\[ \text{size of } S_A = (44 \ E \ bins) \times (12 \ reactions) \times (\text{number of isotopes}) \]
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)

$$C_{xx} = \begin{bmatrix}
\end{bmatrix}$$

size of $C_{xx} = \left[ (44 \text{ E bins}) \times (12 \text{ reactions}) \times (\text{number of isotopes}) \right]^2$
Correlation Coefficient

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

  $\rho = +1$ total positive correlation
  $\rho = -1$ total negative correlation
  $\rho = 0$ no correlation
Variance in $K_{eff}$ & Correlation Between Problems

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

- **Variance in $K_{eff}$ due to nuclear data uncertainties**:
  \[
  \text{Var}_k(A) = \bar{S}_A \bar{C}_{xx} \bar{S}_A^T \\
  \text{Var}_k(B) = \bar{S}_B \bar{C}_{xx} \bar{S}_B^T
  \]

- **Covariance between A & B due to nuclear data uncertainties**:
  \[
  \text{Cov}_k(A,B) = \bar{S}_A \bar{C}_{xx} \bar{S}_B^T
  \]

- **Correlation between Problems A & B due to nuclear data**:
  \[
  c_k(A,B) = \frac{\text{Cov}_k(A,B)}{\sqrt{\text{Var}_k(A) \cdot \text{Var}_k(B)}} = \frac{\bar{S}_A \bar{C}_{xx} \bar{S}_B^T}{\sqrt{\bar{S}_A \bar{C}_{xx} \bar{S}_A^T \cdot \bar{S}_B \bar{C}_{xx} \bar{S}_B^T}}
  \]

$= \text{scalar}$
Sandwich Rule – Variance & Covariance

- Matrix-vector operations

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

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

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

Problem-dependent sensitivity vector, S.
Based on flux spectrum, adjoint spectrum, nuclear data, problem isotopes, geometry, temperature

Size = G x MT x NI

Nuclear Data Covariances
Size = (G x MT x NI)^2

= scalar

\[S^T\]
To consider a simulated system subcritical, the computed keff must be less than the Upper Subcritical Limit (USL):

\[ K_{\text{calc}} + 2\sigma_{K\text{calc}} < \text{USL} \]

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

\[ \text{MOS} = \text{MOS}_{\text{data}} + \text{MOS}_{\text{code}} + \text{MOS}_{\text{application}} \]

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: $k_{\text{eff}}$ only
  - S/U-based: $k_{\text{eff}}$ & 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**
  - 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 $k_{\text{application}} + 2\sigma < \text{USL}$ ?
Whisper Methodology

- **MCNP6**
  - Determine Sensitivity Profiles for Benchmarks $B_1 \ldots B_N$ [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
    cp some-dir/myjob.i WTEST
    ```

• Set up batch job file, job.txt
  
  ```bash
  #!/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
  
  ```bash
  msub job.txt
  ```
To try it, on Moonlight HPC:

- **Set & export WHISPER_PATH environment variable**
  
  - **bash:**
    
    ```bash
    export WHISPER_PATH="/usr/projects/mcnp/ncs/WHISPER"
    export PATH="$WHISPER_PATH/bin:$PATH"
    ```
  
  - **csh, tcsh:**
    
    ```csh
    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:
bash:
cp some-dir/myjob.i WTEST
bash:
bash:
ls WTEST
mjob.i
bash:
```
Using whisper_mcnp (1)

• From the front-end on an HPC system:

\texttt{whisper\_mcnp.pl myjob.i}

  – myjob.i is an MCNP6 input file
    • Must NOT include any of these cards: \texttt{kopts, ksen, prdmp}
    • May list more than 1 input file on \texttt{whisper\_mcnp} command line
    • Lots of options, see next 2 slides

  – Creates files & dirs:
    \texttt{MCNPInputList.toc}
    \texttt{Calcs/}
    \texttt{Calcs/myjob.i} \quad \text{modified to include kopts, ksen, prdmp, & new kcode}
    \texttt{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

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 

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
MCNPIInputlist.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
```
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

application calc  data unc  baseline  k(calc)
          margin   (1-sigma)  USL     > USL

myjob.i  0.01329  0.00120  0.97860  -0.00972
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 ...

<table>
<thead>
<tr>
<th>benchmark</th>
<th>k(bench)</th>
<th>unc</th>
<th>k(calc)</th>
<th>unc</th>
<th>bias</th>
<th>unc</th>
</tr>
</thead>
<tbody>
<tr>
<td>myjob.i</td>
<td>1.00000</td>
<td>0.01100</td>
<td>1.01174</td>
<td>0.00007</td>
<td>-0.01174</td>
<td>0.01100</td>
</tr>
</tbody>
</table>

............

246 benchmarks read, 0 benchmarks excluded.

Reading application data ...

<table>
<thead>
<tr>
<th>application</th>
<th>k(calc)</th>
<th>unc</th>
</tr>
</thead>
<tbody>
<tr>
<td>myjob.i</td>
<td>0.96802</td>
<td>0.00052</td>
</tr>
</tbody>
</table>

Reading covariance data ...

Reading covariance data for 1001 ...

............

Reading adjusted covariance data ...

Reading covariance data for 1001 ...
Calculating application nuclear data uncertainties ...

<table>
<thead>
<tr>
<th>application</th>
<th>adjusted</th>
<th>prior</th>
</tr>
</thead>
<tbody>
<tr>
<td>myjob.i</td>
<td>0.00209</td>
<td>0.01221</td>
</tr>
</tbody>
</table>

Calculating upper subcritical limits ...

<table>
<thead>
<tr>
<th>calc</th>
<th>data unc</th>
<th>baseline</th>
<th>k(calc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>myjob.i</td>
<td>0.01334</td>
<td>0.00209</td>
<td>0.97623</td>
</tr>
</tbody>
</table>

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 rankings shown below

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

For this application, 48 of the 1101 benchmarks were selected as neutronically similar & sufficient for valid statistical analysis.
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\textsubscript{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 7: Critical-mass curves and USL-mass curves comparison
• **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
Electrorefining process

Impure plutonium metal oxidizes to plutonium chloride (PuCl₃), which dissolves in the molten salt and is transported to the cathode, where it is reduced to pure plutonium metal droplets.

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/cm³
- 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
      \[ V = \frac{\text{Pu mass}}{\text{Pu density}} \]
      \[ V = H\pi R^2 = \frac{H\pi}{D} \cdot 2\pi R^3 \]
      \[ R = \left[ \frac{V}{2\pi(H/D)} \right]^{1/3} \]
    - 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
wval1p: 4500 g Pu metal, various H/D
  c reflected 1 inch water radially,
  c 0.25 in steel bottom

Pu cylinder:
  mass       = 4500 g
  density    = 19.86 g/cc
  volume     = VOL_PU
  radius Pu  = R_PU
  height Pu  = H_PU
  H/D        = HD
  H2O outer radius = R_H2O

Pu cylinder with various H/D:
  mass       = 4500 g
  density    = 19.86 g/cc
  volume     = VOL_PU
  radius Pu  = R_PU
  height Pu  = H_PU
  H/D        = HD
  H2O outer radius = R_H2O
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

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

k = 0.83491 (41)

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

<table>
<thead>
<tr>
<th>H/D</th>
<th>Case</th>
<th>KEFF</th>
<th>KSIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>case001</td>
<td>7.87229E-01</td>
<td>4.09191E-04</td>
</tr>
<tr>
<td>1.0</td>
<td>case002</td>
<td>8.34430E-01</td>
<td>4.20715E-04</td>
</tr>
<tr>
<td>1.5</td>
<td>case003</td>
<td>8.29652E-01</td>
<td>4.19130E-04</td>
</tr>
<tr>
<td>2.0</td>
<td>case004</td>
<td>8.11958E-01</td>
<td>4.18723E-04</td>
</tr>
<tr>
<td>2.5</td>
<td>case005</td>
<td>7.93676E-01</td>
<td>4.63720E-04</td>
</tr>
<tr>
<td>3.0</td>
<td>case006</td>
<td>7.73434E-01</td>
<td>4.19664E-04</td>
</tr>
</tbody>
</table>

4.5 kg Pu Ingot

- Ingot
- USL-Ingot Whisper
- USL=0.97
### MCNP6-Whisper Results

<table>
<thead>
<tr>
<th>application</th>
<th>calc data unc</th>
<th>baseline k(calc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>margin</td>
<td>(1-sigma) USL &gt; USL</td>
</tr>
<tr>
<td>ingot.txt_1_in</td>
<td>0.01441</td>
<td>0.00076</td>
</tr>
<tr>
<td>Benchmark population</td>
<td>= 44</td>
<td></td>
</tr>
<tr>
<td>Population weight</td>
<td>= 25.38028</td>
<td></td>
</tr>
<tr>
<td>Maximum similarity</td>
<td>= 0.99621</td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>= 0.00858</td>
<td></td>
</tr>
<tr>
<td>Bias uncertainty</td>
<td>= 0.00583</td>
<td></td>
</tr>
<tr>
<td>Nuc Data uncert margin</td>
<td>= 0.00076</td>
<td></td>
</tr>
<tr>
<td>Software/method margin</td>
<td>= 0.00500</td>
<td></td>
</tr>
<tr>
<td>Non-coverage penalty</td>
<td>= 0.00000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>benchmark</th>
<th>calc weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>pu-met-fast-036-001.i</td>
<td>0.9962</td>
</tr>
<tr>
<td>pu-met-fast-022-001.i</td>
<td>0.9957</td>
</tr>
<tr>
<td>pu-met-fast-024-001.i</td>
<td>0.9956</td>
</tr>
<tr>
<td>pu-met-fast-001-001.i</td>
<td>0.9940</td>
</tr>
<tr>
<td>pu-met-fast-023-001.i</td>
<td>0.9937</td>
</tr>
<tr>
<td>pu-met-fast-039-001.i</td>
<td>0.9932</td>
</tr>
<tr>
<td>mix-met-fast-009-001.i</td>
<td>0.9923</td>
</tr>
<tr>
<td>pu-met-fast-044-005.i</td>
<td>0.9917</td>
</tr>
<tr>
<td>pu-met-fast-035-001.i</td>
<td>0.9913</td>
</tr>
<tr>
<td>pu-met-fast-025-001.i</td>
<td>0.9902</td>
</tr>
<tr>
<td>pu-met-fast-009-001.i</td>
<td>0.9898</td>
</tr>
</tbody>
</table>

### Traditional Validation Results:

**USL = 0.99-MOS-AoA = 0.97 - AoA**
Example 2

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

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

Example 2: 4.5 kg Pu Annulus, varying H & R_{in} (1)
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/cm³
- 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
    \[
    \text{mcnp6 i=wval2.txt}
    \]

  - Copy wval2.txt to wval2p.txt, then insert directives for mcnp_pstudy

    - Define list for solid HD:
      \[
      \text{c @@@ HD = 1.0 2.0 3.0}
      \]
    - For a given H/D, compute Pu height
    - Define list for inner radius R_{IN_PU}
      \[
      \text{c @@@ R_{IN_PU} = 0.001 0.5 1.0 2.0}
      \]
    - Then other dimensions & source

Solid cylinder
\[
V = \frac{\text{Pu mass}}{\text{Pu density}}
\]
\[
V = H\pi R^2 = (H/D) \cdot 2\pi R^3
\]
\[
H = \left[\frac{4V(H/D)^2}{\pi}\right]^{1/3}
\]
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)

<table>
<thead>
<tr>
<th>wval2: 4500 g Pu metal ring, fixed Rin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 3 -1.0 -1 imp:n=1</td>
</tr>
<tr>
<td>2 1 -19.860000 +1 -2 imp:n=1</td>
</tr>
<tr>
<td>11 3 -1.0 +2 -11 imp:n=1</td>
</tr>
<tr>
<td>14 6 -7.92 -30 imp:n=1</td>
</tr>
<tr>
<td>15 0 +11 +30 -20 imp:n=1</td>
</tr>
<tr>
<td>20 0 +20 imp:n=0</td>
</tr>
</tbody>
</table>

| 1 rcc 0 0 0 0 6.608 0.100000          |
| 2 rcc 0 0 0 0 6.608 3.305259         |
| 11 rcc 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    |

### wval2p: 4500 g Pu metal ring, various H & Rin

| 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 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 for hollow cylinder: |
| c use same height as for solid ingot |
| c set various inner radii |

| 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) |

| 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 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 0.100 3.305259 |
| sp1 -21 1 |
| si2 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 0.01460423 |
| 24053.80c 0.001656024 |
| 24054.80c 0.00012220 |
| 26054.80c 0.003469592 |
| 26056.80c 0.0544651574 |
| 26057.80c 0.001257838 |
| 26058.80c 0.0000174 |
| 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 |

| prdmp 9e9 9e9 1 9e9 |

| prdmp 9e9 9e9 1 9e9 |

| prdmp 9e9 9e9 1 9e9 |
Example 2 - wval2p: 4.5 kg Pu Annulus, varying H & R_{in}  (4)

- **Parameter study using** `mcnp_pstudey`, `whisper_mcnp`, & `whisper_usl`:

  ```
  mcnp_pstudey  -i  wval2p.txt  -whisper
  
  use mcnp_pstudey 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\textsubscript{in} (5)**

\[ k = 0.83413 \ (42) \]

```
wval2
mcnp6  i=wval2.txt
```

```
wval2p, varying H & R\textsubscript{in}
mcnp_pstudy  -i wval2p.txt  -setup  -run
```

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

**Comparison of 4.5 kg Pu Ingot and Rings**

![Comparison Graph]

```
Example 2 - wval2p: 4.5 kg Pu Annulus, varying H & R\textsubscript{in} (5)
```
## MCNP6-Whisper Results

<table>
<thead>
<tr>
<th>application</th>
<th>calc</th>
<th>data unc</th>
<th>baseline</th>
<th>k(calc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ringhd2.txt_0.4_in</td>
<td>0.01464</td>
<td>0.00075</td>
<td>0.97840</td>
<td>-0.17760</td>
</tr>
</tbody>
</table>

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

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

### Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 - AoA

Example 2 - wval2p: 4.5 kg Pu Annulus, varying $H$ & $R_{in}$ (6)
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/cm³
- Density of NaCl = 1.556 g/cm³

- 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

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
sil 0 5.989
sp1 -21 2

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

MCNP6-Whisper Results

Pu-NaCl Mixture, 4.5 kg Pu

Concentration, g Pu/cm$^3$

k-effective

k-effective

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

**MCNP6-Whisper Results**

- 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

**USL baseline = .979**

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>ck</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>pu-met-fast-044-002.i</td>
<td>0.9734</td>
<td>0.6832</td>
</tr>
<tr>
<td>pu-met-fast-042-007.i</td>
<td>0.9734</td>
<td>0.6832</td>
</tr>
<tr>
<td>pu-met-fast-042-008.i</td>
<td>0.9722</td>
<td>0.6645</td>
</tr>
<tr>
<td>pu-met-fast-042-009.i</td>
<td>0.9709</td>
<td>0.6426</td>
</tr>
<tr>
<td>pu-met-fast-042-010.i</td>
<td>0.9705</td>
<td>0.6356</td>
</tr>
<tr>
<td>pu-met-fast-042-011.i</td>
<td>0.9699</td>
<td>0.6257</td>
</tr>
<tr>
<td>pu-met-fast-023-001.i</td>
<td>0.9691</td>
<td>0.6133</td>
</tr>
<tr>
<td>pu-met-fast-042-012.i</td>
<td>0.9687</td>
<td>0.6054</td>
</tr>
<tr>
<td>pu-met-fast-039-001.i</td>
<td>0.9683</td>
<td>0.5993</td>
</tr>
<tr>
<td>pu-met-fast-042-014.i</td>
<td>0.9681</td>
<td>0.5961</td>
</tr>
<tr>
<td>pu-met-fast-042-013.i</td>
<td>0.9681</td>
<td>0.5959</td>
</tr>
<tr>
<td>pu-met-fast-042-015.i</td>
<td>0.9676</td>
<td>0.587</td>
</tr>
<tr>
<td>pu-met-fast-022-001.i</td>
<td>0.9644</td>
<td>0.534</td>
</tr>
<tr>
<td>pu-met-fast-009-001.i</td>
<td>0.964</td>
<td>0.5284</td>
</tr>
<tr>
<td>pu-met-fast-035-001.i</td>
<td>0.9629</td>
<td>0.5093</td>
</tr>
<tr>
<td>mix-met-fast-009-001.i</td>
<td>0.9618</td>
<td>0.4919</td>
</tr>
<tr>
<td>pu-met-fast-044-001.i</td>
<td>0.9612</td>
<td>0.482</td>
</tr>
<tr>
<td>pu-met-fast-001-001.i</td>
<td>0.9602</td>
<td>0.4653</td>
</tr>
<tr>
<td>pu-met-fast-025-001.i</td>
<td>0.9593</td>
<td>0.4499</td>
</tr>
<tr>
<td>pu-met-fast-021-001.i</td>
<td>0.9588</td>
<td>0.4424</td>
</tr>
<tr>
<td>pu-met-fast-030-001.i</td>
<td>0.9559</td>
<td>0.3941</td>
</tr>
<tr>
<td>pu-met-fast-018-001.i</td>
<td>0.9555</td>
<td>0.3863</td>
</tr>
<tr>
<td>pu-met-fast-029-001.i</td>
<td>0.951</td>
<td>0.3115</td>
</tr>
<tr>
<td>pu-met-fast-045-005.i</td>
<td>0.9509</td>
<td>0.3097</td>
</tr>
<tr>
<td>mix-met-fast-007-022.i</td>
<td>0.9496</td>
<td>0.2897</td>
</tr>
<tr>
<td>mix-met-fast-007-023.i</td>
<td>0.9448</td>
<td>0.2093</td>
</tr>
<tr>
<td>pu-met-fast-019-001.i</td>
<td>0.9421</td>
<td>0.1637</td>
</tr>
<tr>
<td>pu-met-fast-038-001.i</td>
<td>0.9384</td>
<td>0.1032</td>
</tr>
<tr>
<td>mix-met-fast-001-001.i</td>
<td>0.9374</td>
<td>0.0871</td>
</tr>
<tr>
<td>pu-met-fast-040-001.i</td>
<td>0.9355</td>
<td>0.055</td>
</tr>
<tr>
<td>pu-met-fast-003-103.i</td>
<td>0.9352</td>
<td>0.0505</td>
</tr>
</tbody>
</table>

**Traditional Validation Results:**

- USL = 0.99-MOS-AoA = 0.97 – AoA
Example 4

- 4.5 kg Pu Sphere, Ta Reflector, various thicknesses
Example 4: Ta-reflected Pu

- **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?

- **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/cm³
- Reflected radially with Ta

- Vary the Ta-reflector thickness over the range 0.01 – 30.0 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 5.0 10.0 15.0 20.0 25.0 30.0
    - 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

Pu mass = 4500 g
Pu density = 19.8 g/cc
Pu volume = 227.272727

reflector definition:
reflector thickness = 7.62
reflector inner radius = 3.7857584
reflector outer radius = 11.405758

1 4 -19.80 -1 imp:n=1
2 1 -16.69 +1 -2 imp:n=1
20 0 +2 imp:n=0

so 3.7857584
so 11.405758

kcode 10000 1.0 50 250
sdef pos=0 0 0 rad=d1
si1 0 3.78
sp1 -21 2

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

Pu mass = 4500 g
Pu density = 19.8 g/cc
Pu volume = 227.272727

vary reflector thickness from 0+ to 30 cm

@@@ THICK = .01 5. 10. 15. 20. 25. 30.
@@@ R_INNER = 3.7857584
@@@ R_OUTER = ( R_INNER + THICK )

reflector definition:
reflector thickness = THICK cm
reflector inner radius = R_INNER cm
reflector outer radius = R_OUTER cm

1 4 -19.80 -1 imp:n=1
2 1 -16.69 +1 -2 imp:n=1
20 0 +2 imp:n=0

so R_INNER
so R_OUTER

kcode 10000 1.0 50 250
sdef pos=0 0 0 rad=d1
si1 0 R_INNER
sp1 -21 2

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**

<table>
<thead>
<tr>
<th>T</th>
<th>Case</th>
<th>KEFF</th>
<th>KSIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>.01</td>
<td>case001</td>
<td>7.91693E-01</td>
<td>3.14948E-04</td>
</tr>
<tr>
<td>5.0</td>
<td>case002</td>
<td>9.27157E-01</td>
<td>4.47334E-04</td>
</tr>
<tr>
<td>10.</td>
<td>case003</td>
<td>9.54775E-01</td>
<td>4.11031E-04</td>
</tr>
<tr>
<td>15.</td>
<td>case004</td>
<td>9.61644E-01</td>
<td>4.34033E-04</td>
</tr>
<tr>
<td>20.</td>
<td>case005</td>
<td>9.62867E-01</td>
<td>4.37235E-04</td>
</tr>
<tr>
<td>25.</td>
<td>case006</td>
<td>9.63899E-01</td>
<td>4.04508E-04</td>
</tr>
<tr>
<td>30.</td>
<td>case007</td>
<td>9.63160E-01</td>
<td>4.27633E-04</td>
</tr>
</tbody>
</table>

$k = 0.94638 (41)$

**mcnp6 i=wval4.txt**

**mcnp_pstudy -i wval4p.txt -setup -run**

**4.5 kg Pu with Ta Reflection**

- Ta-reflected Pu
- Whisper USL
- USL=0.97
Example 4: Ta-reflected Pu

### MCNP6 and Whisper Results

<table>
<thead>
<tr>
<th>application</th>
<th>calc margin</th>
<th>data unc (1-sigma)</th>
<th>baseline</th>
<th>k(calc)</th>
<th>benchmark</th>
<th>ck</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>tarefl.txt_7.62_in</td>
<td>0.01707</td>
<td>0.01502</td>
<td>0.93889</td>
<td>0.00750</td>
<td>mix-met-fast-009-001.i</td>
<td>0.4193</td>
<td>0.5919</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-009-001.i</td>
<td>0.4190</td>
<td>0.5914</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-035-001.i</td>
<td>0.4189</td>
<td>0.5913</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-022-001.i</td>
<td>0.4185</td>
<td>0.5904</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-025-001.i</td>
<td>0.4183</td>
<td>0.5900</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-036-001.i</td>
<td>0.4180</td>
<td>0.5896</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-001-001.i</td>
<td>0.4180</td>
<td>0.5895</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-021-002.i</td>
<td>0.4176</td>
<td>0.5887</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-030-001.i</td>
<td>0.4171</td>
<td>0.5879</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-024-001.i</td>
<td>0.4171</td>
<td>0.5878</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-021-001.i</td>
<td>0.4165</td>
<td>0.5867</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-044-003.i</td>
<td>0.4164</td>
<td>0.5866</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-044-005.i</td>
<td>0.4162</td>
<td>0.5863</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-044-002.i</td>
<td>0.4160</td>
<td>0.5858</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-029-001.i</td>
<td>0.4155</td>
<td>0.5850</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-044-004.i</td>
<td>0.4146</td>
<td>0.5832</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-003-103.i</td>
<td>0.4141</td>
<td>0.5823</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-015.i</td>
<td>0.4134</td>
<td>0.5811</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-012.i</td>
<td>0.4134</td>
<td>0.5811</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mix-met-fast-007-022.i</td>
<td>0.4134</td>
<td>0.5811</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-011.i</td>
<td>0.4134</td>
<td>0.5810</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-009.i</td>
<td>0.4134</td>
<td>0.5810</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-013.i</td>
<td>0.4133</td>
<td>0.5808</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-014.i</td>
<td>0.4133</td>
<td>0.5808</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-010.i</td>
<td>0.4133</td>
<td>0.5808</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-007.i</td>
<td>0.4132</td>
<td>0.5807</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-018-001.i</td>
<td>0.4132</td>
<td>0.5806</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-006.i</td>
<td>0.4131</td>
<td>0.5806</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pu-met-fast-042-008.i</td>
<td>0.4131</td>
<td>0.5805</td>
</tr>
</tbody>
</table>

### Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 – AoA

**Trouble! Benchmarks are not very similar to application**
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:
    
    \[
    \begin{align*}
    0.95 < C_k & \rightarrow \text{great} \\
    0.90 < C_k < 0.95 & \rightarrow \text{good} \\
    C_k < 0.90 & \rightarrow \text{not so good}
    \end{align*}
    \]
    
    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?

- 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 $k_{eff}$. 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

$$\text{USL} = 1.0 + (\text{bias}) - (\text{bias uncertainty}) - (\text{margin of subcriticality}) - (\text{AoA margin})$$
Example 5: Oil-Moderated Pu

- **MCNP6 Input**
  
  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

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

- **Pu concentration range:**
  -19.8 g Pu/cm³

- **Hydraulic oil composition:**
  \( \text{C}_{40}\text{H}_{33}\text{O}_{4}\text{Cl}_{6}\text{P} \)

- **Hydraulic oil density:**
  0.871 g/cm³

- **Reflected with 1 inch of water**
Example 5: Oil-Moderated Pu

- MCNP6 and Whisper Results
Example 5: Oil-Moderated Pu

MCNP6 and Whisper Results

<table>
<thead>
<tr>
<th>application</th>
<th>calc</th>
<th>data unc</th>
<th>baseline</th>
<th>k(calc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>puoilmix.txt_7_in</td>
<td>0.01477</td>
<td>0.00109</td>
<td>0.97739</td>
<td>-0.41445</td>
</tr>
</tbody>
</table>

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 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pu-met-fast-042-001.i</td>
<td>0.9643</td>
<td>1.0000</td>
</tr>
<tr>
<td>pu-met-fast-011-001.i</td>
<td>0.9641</td>
<td>0.9973</td>
</tr>
<tr>
<td>pu-met-fast-027-001.i</td>
<td>0.9580</td>
<td>0.9377</td>
</tr>
<tr>
<td>pu-met-fast-042-002.i</td>
<td>0.9561</td>
<td>0.9199</td>
</tr>
<tr>
<td>pu-met-fast-042-003.i</td>
<td>0.9483</td>
<td>0.8436</td>
</tr>
<tr>
<td>pu-met-fast-044-004.i</td>
<td>0.9474</td>
<td>0.8343</td>
</tr>
<tr>
<td>pu-met-fast-042-004.i</td>
<td>0.9444</td>
<td>0.8048</td>
</tr>
<tr>
<td>pu-met-fast-031-001.i</td>
<td>0.9425</td>
<td>0.7861</td>
</tr>
<tr>
<td>pu-met-fast-044-005.i</td>
<td>0.9404</td>
<td>0.7658</td>
</tr>
</tbody>
</table>

Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 - AoA
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].


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

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

N = 15,
Mass Per Cube = ~1.48 g,
Spacing = 1 cm
Example 6: Revisiting a Practical Application of the SPSL for Pu Metal

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Subcritical limits for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass of fissile nuclide (kg)</td>
<td>6.0</td>
</tr>
<tr>
<td>Cylinder diameter (cm)</td>
<td>4.5</td>
</tr>
<tr>
<td>Slab thickness (cm)</td>
<td>0.38</td>
</tr>
<tr>
<td>Uranium enrichment (wt% $^{235}$U)</td>
<td>–</td>
</tr>
<tr>
<td>Maximum density for which mass and dimension limits are valid (g/cm³)</td>
<td>18.65</td>
</tr>
</tbody>
</table>
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

Bare Pu Critical-Mass Curve
Comparison with USL-Mass Curve

Pu Concentration (g/cm³)

Mass (s)
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.

**Example 7: Critical-Mass and USL-Mass Curves**

[ANSI/ANS-8.24 7.2]

**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.
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**
mcnp_pstudy

- 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

<table>
<thead>
<tr>
<th>gdv</th>
<th>c</th>
<th>1 100 -18.74 -1 imp:n=1</th>
<th>2 0 1 imp:n=0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 so 8.741</td>
<td></td>
</tr>
<tr>
<td>kcode</td>
<td>10000</td>
<td>1.0 15 115</td>
<td></td>
</tr>
<tr>
<td>ksrc</td>
<td>0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m100</td>
<td>92235</td>
<td>-94.73 92238 -5.27</td>
<td></td>
</tr>
<tr>
<td>prdmp</td>
<td>0 0 1 1 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>gdv-A</th>
<th>C @@@ RADIUS = 8.500 8.741 8.750</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 100 -18.74 -1 imp:n=1</td>
</tr>
<tr>
<td></td>
<td>2 0 1 imp:n=0</td>
</tr>
<tr>
<td></td>
<td>1 so RADIUS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>kcode</th>
<th>10000</th>
<th>1.0 15 115</th>
</tr>
</thead>
<tbody>
<tr>
<td>ksrc</td>
<td>0 0 0</td>
<td></td>
</tr>
<tr>
<td>m100</td>
<td>92235</td>
<td>-94.73 92238 -5.27</td>
</tr>
<tr>
<td>prdmp</td>
<td>0 0 1 1 0</td>
<td></td>
</tr>
</tbody>
</table>
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: \( R_1, r_1, U\_density, U\_den \)

- **Single value**
  \( C \ associates \ P1 = value \)

- **List of values**
  \( C \ associates \ P2 = value1 value2 ... valueN \)

- **List of N random samples from Probability Densities:**
  - **Uniform**
    \( C \ associates \ P3 = uniform \ N \ min \ max \)
  - **Normal**
    \( C \ associates \ P4 = normal \ N \ ave \ dev \)
  - **Lognormal**
    \( C \ associates \ P5 = lognormal \ N \ ave \ dev \)
  - **Beta**
    \( C \ associates \ P6 = beta \ N \ a \ b \ [a,b \ are \ integers] \)
Parameter Definition

- **Arithmetic expression**
  
  \( C @@@ P5 = (\text{arithmetic-statement}) \)
  
  - Can use numbers & previously defined parameters
  - Can use arithmetic operators \(+, -, *, /, \% (\text{mod}), ** (\text{exponentiation})\)
  - Can use parentheses \((\ )\)
  - Can use functions: \(\text{sin}(), \text{cos}(), \text{log}(), \text{exp}(), \text{int}(), \text{abs}(), \text{sqrt}()\)
  - Can generate random number in \((0,N): \text{rand}(N)\)
  - Can use \(\text{rn} \_\text{seed}()\) to get odd seed for mcnp RN generator in \([1,2^{48}-1]\)
  - Must evaluate to a single value
  - Examples:
    
    \[
    C @@@ \text{SEED} = (\text{rn} \_\text{seed}())
    \]
    \[
    C @@@ \text{FACT} = \text{normal} 1 1.0 .05
    \]
    \[
    C @@@ \text{UDEN} = (18.74 * \text{FACT})
    \]
    \[
    C @@@ \text{URAD} = (8.741 * (18.74/\text{UDEN})^{0.333333})
    \]

- **Repetition** (list of integers, 1..N)
  
  \( C @@@ P6 = \text{repeat} N \)
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() )
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
  B = 3 4
  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 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**

  ```plaintext
  JOBDIR
  ├── case001
  │    └── inp
  ├── case002
  │    └── inp
  ├── case003
  │    └── inp
  └── ..... 
  
  - Unix filesystem conventions followed
    ```plaintext
    JOBDIR/case001/inp,  JOBDIR/case002/inp, etc.
    ```

- **Values of parameters are substituted 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
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:  
    
    \texttt{-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
    
    \texttt{inp\_case001, inp\_case002, etc.}
    
  - Using \texttt{-case study01a -inponly} would result in files with names
    
    \texttt{inp\_study01a001, inp\_study01a002, etc.}
    
  - Other options \texttt{-run, -submit} cannot be used if \texttt{-inponly} is present
    
  - The option \texttt{-whisper} can be used, and is equivalent to \texttt{-inponly}
Combining Results

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

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

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

\[
\sigma_{\bar{X}}^2 \approx \sigma_{\bar{X}, \text{Monte Carlo}}^2 + \sigma_{\bar{X}, \text{Initial Conditions}}^2
\]

Varies as \( 1/M \) ~ Constant
## Examples

### 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
@@@ FACT= normal 50 1.0 .05
@@@ UDEN= ( 18.74*FACT )
@@@ 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
@@@ OPTIONS = -inner
c
@@@ DFACT = normal 50 1.0 .05
@@@ UDEN = ( DFACT * 18.74 )
@@@ UFACT = normal 50 1.0 .05
@@@ 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
```
## Examples

### Table 1. Results from varying parameters in the Godiva problem

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

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**
- Pu mass = 4500 g
- Pu density = 19.8 g/cc
- Pu volume = 227.272727

reflector definition:
- reflector thickness = 7.62 cm
- reflector inner radius = 3.7857584 cm
- reflector outer radius = 11.405758 cm

1 4 -19.80 -1 imp:n=1
2 1 -16.69 +1 -2 imp:n=1
20 0 +2 imp:n=0

kcode 10000 1.0 50 250
sdef pos=0 0 0 rad=d1
sil 0 3.78
spl -21 2

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**
- Pu mass = 4500 g
- Pu density = 19.8 g/cc
- Pu volume = 227.272727

vary reflector thickness from 0+ to 30 cm
@@@ THICK = .01 5. 10. 15. 20. 25. 30.
@@@ R_INNER = 3.7857584
@@@ R_OUTER = ( R_INNER + THICK )

reflector definition:
- reflector thickness = THICK cm
- reflector inner radius = R_INNER cm
- reflector outer radius = R_OUTER cm

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
sil 0 R_INNER
spl -21 2

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

wval4, thick=7.62
mcnp6 i=wval4.txt

mcnp_pstudy -i wval4p.txt -setup -run

k = 0.94638 (41)

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

4.5 kg Pu with Ta Reflection

La-UR-16-23533 - 135
Example

\texttt{wval1,}
\texttt{4.5 \text{kg Pu Ingot,}}
\texttt{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/cm³
- 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
      \[ V = \frac{\text{Pu mass}}{\text{Pu density}} \]
      \[ V = H\pi R^2 = (H/D) \cdot 2\pi R^3 \]
      \[ R = \left[ \frac{V}{2\pi(H/D)} \right]^{1/3} \]
    - 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 -2.540000  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 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 0.001673959
25055.80c 0.00174
28058.80c 0.005255537
28060.80c 0.002024423
28061.80c 0.000888000
28062.80c 0.00280583
28064.80c 0.00071456

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
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

kcode  10000 1.0 50 250
ksrc  0 0 .303831
Pu cylinder:
  mass       = 4500 g
  density    = 19.86 g/cc
  volume     = VOL_PU
  radius Pu  = R_PU
  height Pu  = H_PU
  H/D        = HD
  H2O  outer radius = R_H2O

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 0.0 3.303831

............... 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

<table>
<thead>
<tr>
<th>H/D</th>
<th>Case</th>
<th>KEFF</th>
<th>KSIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>case001</td>
<td>7.87229E-01</td>
<td>4.09191E-04</td>
</tr>
<tr>
<td>1.0</td>
<td>case002</td>
<td>8.34430E-01</td>
<td>4.20175E-04</td>
</tr>
<tr>
<td>1.5</td>
<td>case003</td>
<td>8.29652E-01</td>
<td>4.19130E-04</td>
</tr>
<tr>
<td>2.0</td>
<td>case004</td>
<td>8.11958E-01</td>
<td>4.18723E-04</td>
</tr>
<tr>
<td>2.5</td>
<td>case005</td>
<td>7.93676E-01</td>
<td>4.63720E-04</td>
</tr>
<tr>
<td>3.0</td>
<td>case006</td>
<td>7.73434E-01</td>
<td>4.19664E-04</td>
</tr>
</tbody>
</table>

4.5 kg Pu Ingot k-effective and USL
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:
      
      \[
      \text{c @@@ HD} = 1.0 \ 2.0 \ 3.0
      \]
    
    - For a given H/D, compute Pu height
    
    - Define list for inner radius RIN_PU
      
      \[
      \text{c @@@ RIN_PU} = 0.001 \ 0.5 \ 1.0 \ 2.0
      \]
    
    - Then other dimensions & source

Solid cylinder

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

\[
V = H\pi R^2 = (H/D) \cdot 2\pi R^3
\]

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

Hollow cylinder

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

\[
R_{out} = \left[R_{in}^2 + V/\pi H\right]^{1/2}
\]
Example \textit{wval2}: 4.5 kg Pu Annulus, varying H & R\textsubscript{in} (2)

\begin{verbatim}
\textbf{wval2: 4500 g Pu metal ring, fixed Rin}
1 3 -1.0 -1 \text{ imp:n=1}
2 1 -19.860000 +1 -2 \text{ imp:n=1}
11 3 -1.0 +2 -11 \text{ imp:n=1}
14 6 -7.92 -30 \text{ imp:n=1}
15 0 +11 +30 -20 \text{ imp:n=1}
20 0 +20 \text{ imp:n=0}

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

\textbf{kcode} 10000 1.0 50 250
\textbf{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 0.004122220
26054.80c 0.003469592
26056.80c 0.054465174
26057.80c 0.01257838
26058.80c 0.00167395
25055.80c 0.00174
28058.80c 0.005255537
28060.80c 0.002024423
28061.80c 0.00088000
28062.80c 0.000280583
28064.80c 0.00071456

\textbf{prdmp} 9e9 9e9 1 9e9
\end{verbatim}
Example \textit{wval2}: 4.5 kg Pu Annulus, varying H \& R_{in} (3)

\textbf{wval2}\newline \textbf{mcnp6} i=wval2.txt

\textbf{k} = 0.83413 (42)

\textbf{wval2p, varying H \& R_{in}}\newline \textbf{mcnp\_pstudy} -i wval2p.txt -setup -run

<table>
<thead>
<tr>
<th>HD</th>
<th>Rin</th>
<th>Case</th>
<th>KEFF (1)</th>
<th>KEFF (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.001</td>
<td>case001</td>
<td>8.3475E-01</td>
<td>4.3566E-04</td>
</tr>
<tr>
<td>2</td>
<td>.001</td>
<td>case002</td>
<td>8.1261E-01</td>
<td>4.0951E-04</td>
</tr>
<tr>
<td>3</td>
<td>.001</td>
<td>case003</td>
<td>7.7273E-01</td>
<td>3.8263E-04</td>
</tr>
<tr>
<td>1</td>
<td>.5</td>
<td>case004</td>
<td>8.2043E-01</td>
<td>4.0113E-04</td>
</tr>
<tr>
<td>2</td>
<td>.5</td>
<td>case005</td>
<td>7.9538E-01</td>
<td>4.6039E-04</td>
</tr>
<tr>
<td>3</td>
<td>.5</td>
<td>case006</td>
<td>7.5417E-01</td>
<td>3.9658E-04</td>
</tr>
<tr>
<td>1</td>
<td>1.0</td>
<td>case007</td>
<td>7.8849E-01</td>
<td>3.9502E-04</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>case008</td>
<td>7.6239E-01</td>
<td>3.9029E-04</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>case009</td>
<td>7.2081E-01</td>
<td>4.2735E-04</td>
</tr>
<tr>
<td>1</td>
<td>2.0</td>
<td>case010</td>
<td>7.2152E-01</td>
<td>4.0277E-04</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
<td>case011</td>
<td>6.9795E-01</td>
<td>4.8827E-04</td>
</tr>
<tr>
<td>3</td>
<td>2.0</td>
<td>case012</td>
<td>6.6404E-01</td>
<td>4.8832E-04</td>
</tr>
</tbody>
</table>

\textbf{Comparison of 4.5 kg Pu Ingot and Rings}
Advanced Topics

Tied parameters

Concurrent jobs
Parameter Expansion (1)

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

  Example:

  \[
  \begin{align*}
  c @@ A &= 1 \ 2 \\
  c @@ B &= 3 \ 4 \\
  c @@ C &= 5 \ 6 \\
  c @@ D &= 7 \ 8 \\
  c @@ E &= 9 \\
  \end{align*}
  \]

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

**Example:**

```plaintext
  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\}, \{1,3,6,8,9\}, \{2,4,5,7,9\}, \{2,4,6,8,9\}

**Example:**

```plaintext
  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\}, \{2,4,5,7,9\}, \{2,4,5,8,9\}

**Example:**

```plaintext
  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:**

```plaintext
  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\}
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},  {1, 3, 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
Abstract

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

Theory


User Manual


Application

Software Quality Assurance

- 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

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

- XCP Data Team, "LANL Data Testing Support for ENDF/B-VII.1", LA-UR-12-20002 LA-UR-12-20002 (2012)

General References on Adjoint, Perturbation, and Sensitivity Analysis

- 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)
General References on Adjoint, Perturbation, and Sensitivity Analysis  (cont’d)


References for Whisper & MCNP6  (5)

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


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

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

Issued: 2016-05-02