

## LA-UR-17-27058

Approved for public release; distribution is unlimited.

Title: Sensitivity-Uncertainty Techniques for Nuclear Criticality Safety

Author(s): Brown, Forrest B.  
Rising, Michael Evan  
Alwin, Jennifer Louise

Intended for: DOE-NCSP Training Class

Issued: 2017-08-07

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# **Sensitivity-Uncertainty Techniques for Nuclear Criticality Safety**

**Forrest Brown, Michael Rising, Jennifer Alwin**

**Monte Carlo Methods, Codes, & Applications (XCP-3)  
X Computational Physics Division**



# Abstract

---

## Sensitivity-Uncertainty Techniques for Nuclear Criticality Safety

Forrest Brown, Michael Rising, Jennifer Alwin  
Monte Carlo Codes Group, LANL

The sensitivity and uncertainty analysis course will introduce students to  $k_{\text{eff}}$  sensitivity data, cross-section uncertainty data, how  $k_{\text{eff}}$  sensitivity data and  $k_{\text{eff}}$  uncertainty data are generated and how they can be used. Discussion will include how sensitivity/uncertainty data can be used to select applicable critical experiments, to quantify a defensible margin to cover validation gaps and weaknesses, and in development of upper subcritical limits.

Acknowledgment: This work was supported by the  
US DOE-NNSA Nuclear Criticality Safety Program.



# Outline

---

## Sensitivity-Uncertainty Techniques for Nuclear Criticality Safety

1. Introduction
2. Validation For Nuclear Criticality Safety
3. Neutron Physics & Statistical Methods
  - a) Neutron Spectra
  - b) Nuclear Data Sensitivities
  - c) Covariance Data For Nuclear Cross-sections
  - d) Correlation Coefficients
4. Application To Nuclear Criticality Safety Validation
  - a) Introduction
  - b) Benchmark Selection –  $C_k$ 's
  - c) Extreme Value Theory – Bias, Bias Uncertainty
  - d) MOS For Nuclear Data Uncertainty – GLLS
5. Practical Use Of Sensitivity-Uncertainty Tools
  - a) Review: Best Practices For Monte Carlo Criticality Calculations
  - b) Introduction – Scale/Tsunami & Mcnp6/Whisper
  - c) MCNP/Whisper - Whisper\_mcnp, Whisper\_usl
6. Examples
  - a) Pu Pyrochemical Processing – Geometry, Materials, Reflection, Moderation
  - b) HEU examples
  - c) General Studies
7. Using Whisper to Support NCS Validation ANSI/ANS-8.24 Requirements
8. References

## Introduction – NCS Validation

---

### Big Picture:

- **Calculations used to support nuclear criticality safety evaluations must make use of validated computer codes**
- **Computer code validation:**
  - Compare calculated results to nature (ie, experimental measurements)
  - Must compare to experiments similar to application of interest
  - Determine how accurate the codes are
- **Conservatism is fundamental to NCS**
  - Always consider uncertainties in calculations, data, measurements
  - Use additional margin for uncertainties that cannot be calculated
  - Subtract uncertainties from upper subcritical limits on  $K_{\text{eff}}$
- **Codes are great, but analyst judgment is required for everything**

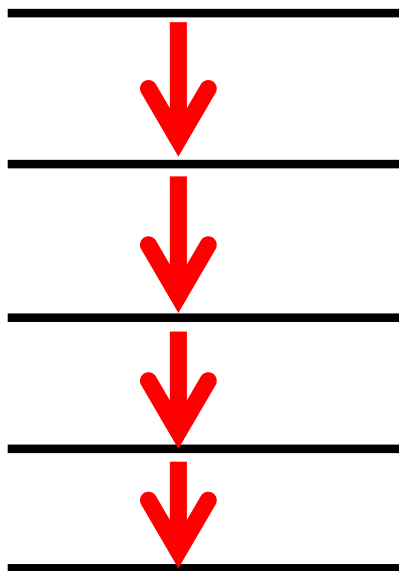
## Introduction – NCS validation

### Upper Subcritical Limit (USL)

- For an application:

- A calculated  $K_{\text{eff}} < 1.0$  is NOT sufficient to ensure subcriticality
- Must conservatively account for
  - Bias & uncertainties in the calculational method
  - Uncertainties in the physical model (eg, mass, isotopics, geometry, ...)

$K_{\text{eff}} = 1$



**Bias** = mean ( $K_{\text{calc}} - K_{\text{exp}}$ ) for a set of experiments that are similar to the application

**Bias Uncertainty**, at 95% or 99% confidence level

**Margin of Subcriticality (MOS)** = code & data uncertainties

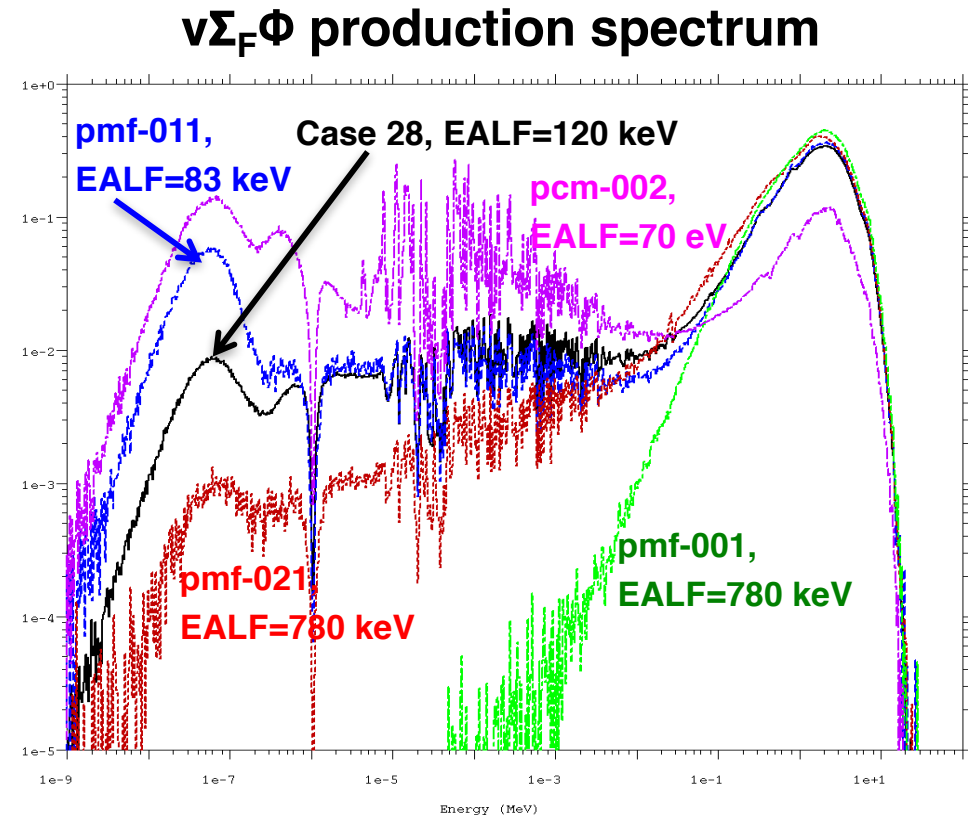
**MOS for Area of Applicability (AOA)** = if benchmarks are not similar enough to application

**USL**

**Must have:**  $K_{\text{calc}} + 2\sigma_{\text{calc}} < \text{USL}$

## Introduction – NCS validation

- Nuclear Criticality Safety requires validation of computational methods
- Validation involves comparing calculation vs experiment for many benchmarks similar to the application of interest
- Neutron spectra are complex functions of geometry, materials, nuclear data, etc.
- The figure shows neutron production spectra for 5 Pu systems:
  - an application (Case 28)
  - 4 benchmarks for Pu systems
- Which of the benchmarks are similar to the application?



**During the past 20 years, powerful tools have been developed based on sensitivity-uncertainty methods**

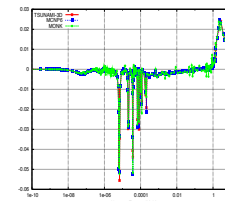
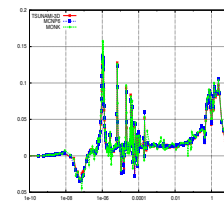
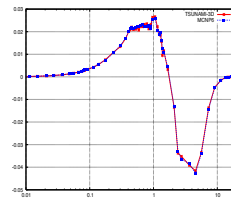
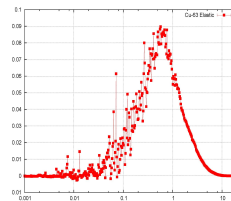
- From ORNL, the Scale system includes Keno, Tsunami, Tsurfer, & other tools
- From LANL, the MCNP6 & Whisper tools are now available
- Other tools have been developed by groups in England, France, Germany, Japan, S. Korea, China

## Introduction - Sensitivity Profiles for Nuclear Data

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

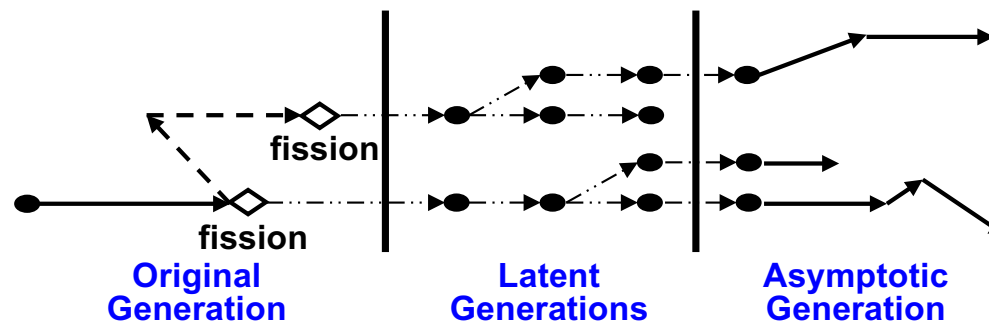
$$S_{k,x} = \frac{dk/k}{dx/x} = - \frac{\langle \psi^\dagger, (\Sigma_x - S_x - k^{-1}F_x) \psi \rangle}{\langle \psi^\dagger, k^{-1}F \psi \rangle}$$

- $S_{k,x}(E)$  is the **sensitivity profile**, that includes all isotopes, reactions, & energies for a system:

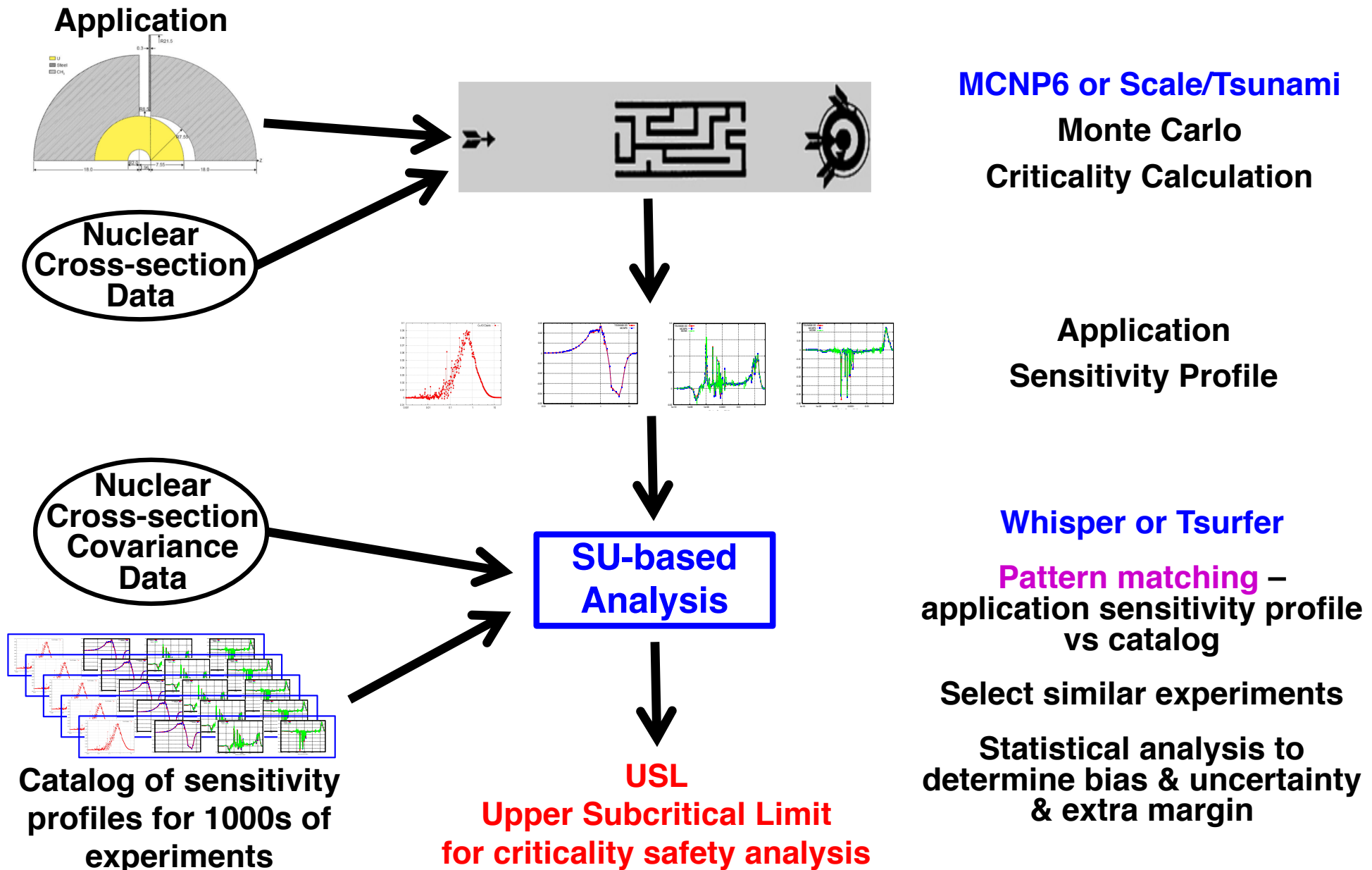


etc.

- MCNP6 & Scale/Tsunami Monte Carlo** can use the Iterated Fission Probability method to compute adjoint-weighted integrals for the sensitivity profiles
  - Tally scores are collected in original generation, adjoint-weighting is based on the progeny in the asymptotic generation



# Introduction - Sensitivity-Uncertainty Methodology for NCS



## Introduction – Goals

---

- **Review of Validation for NCS**
  - Overview, requirements, standards, definitions, USL
  - Selecting benchmarks, bias & bias uncertainty, validation approaches
- **Fundamental Concepts for Sensitivity-Uncertainty Methods**
  - Spectra
  - Sensitivity of  $K_{\text{eff}}$  to nuclear data
  - Covariances for the nuclear data
  - Correlation coefficients & the sandwich rule
- **Application to NCS Validation**
  - Computing correlation coefficients & selecting benchmarks
  - Determining bias & bias uncertainty
  - Determining (minimum) extra margin for data & other uncertainties
- **Practical Examples**





# Nuclear Criticality Safety Validation

Introduction - background, standards, definitions,  
USL, calculational margin, margin of subcriticality

Selection of benchmarks

Bias & bias uncertainty

Sensitivity-uncertainty analysis

Validation approaches & technical review

## 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**
    - Computer codes have approximations and errors
    - Nuclear data have approximations and errors
  - **Criticality safety:**
    - Focus on avoiding worst-case combination of mistakes, uncertainties, errors
    - Rigor & conservatism always; never wishful thinking or "close enough"
  - **How can we be confident in assessing subcriticality?**
    - Verify that codes work as intended
    - Validate codes + data + methods against nature (experiments)

# Orders, Standards, Guides for NCS

- 10 CFR 830 Subpart A, Quality Assurance
- 10 CFR 830 Subpart B, Nuclear Safety Management
  
- DOE O 414.1C, Quality Assurance
- DOE G 414.1-4, Safety Software Guide for use with 10CFR 830 Subpart A, Quality Assurance Requirements
- DOE G 421.1-2, Implementation Guide for Use in Developing Documented Safety Analyses to Meet Subpart B of 10 CFR 830
- DOE O 420.1C, Facility Safety
- DOE O 426.2 Personnel Selection, Training, Qualification, and Certification Requirements
  
- **DOE-STD-3007-2007, Guidelines for Preparing Criticality Safety Evaluations at DOE Nonreactor Nuclear Facilities**
- DOE STD 1134-1999 Review Guide for Criticality Safety Evaluations
- DOE-STD-1158-2010, Self-Assessment Standard for DOE Contractor Criticality Safety Programs
- DOE-STD-3009-1994, Preparation Guide for U.S. Department of Energy Nonreactor Nuclear Facility Safety Analysis
- DOE-STD-1186-2004, Specific Administrative Controls
- DOE-STD-1027-1992, Hazard Categorization and Accident Analysis Techniques for Compliance with DOE Order 5480.23, Nuclear Safety Analysis Reports
  
- **ANSI/ANS-8.1-2014, Nuclear Criticality Safety in Operations with Fissionable Materials Outside Reactors**
  
- ANSI/ANS-8.3-2003, Criticality Accident Alarm System
- ANSI/ANS-8.5-1996(R2007), Use of 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-8.10-2005, Criteria for Nuclear Criticality Safety Controls in Operations with Shielding and Confinement
- ANSI/ANS 8.14-2004, Use of Soluble Neutron Absorbers in Nuclear Facilities Outside Reactors
- ANSI/ANS 8.17-2004, Criticality Safety Criteria for the Handling, Storage, and Transportation of LWR Fuel Outside Reactors
- ANSI/ANS-8.19-2014, Administrative Practices for Nuclear Criticality Safety
- ANSI/ANS 8.20-1991(R2005), Nuclear Criticality Safety Training
- ANSI/ANS-8.21-1995(R2001), Use of Fixed Neutron Absorbers in Nuclear Facilities Outside Reactors
- ANSI/ANS-8.23-2007, Nuclear Criticality Accident Emergency Planning and Response
  
- **ANSI/ANS 8.24-2007, Validation of Neutron Transport Methods for Nuclear Criticality Safety Calculations**
  
- ANSI/ANS 8.26-2007, Criticality Safety Engineer Training and Qualification Program
  
- **Validation with Limited Benchmark Data, Response to CSSG Tasking 2014-02**

# Background

---

## Establishing Subcriticality

- ***Any method*** used to determine the subcritical state of a fissionable material system must be validated.
- **Direct use of experimental data is preferred (ANSI/ANS-8.1-2014 4.2.7)**
  - 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 4.3)**
  - Validation shall be performed by comparison to experiments and AoA should be established from this comparison.
    - 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.

## Validation: Definitions (1)

---

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

## Validation: Definitions (2)

---

- From ANSI/ANS-8.24-2007, Validation of Neutron Transport Methods for Nuclear Criticality Safety Calculations:
  - **Bias:** The systematic difference between calculated results and experimental data. [ $k_{\text{eff calculated}} - k_{\text{eff experiment}}$ ]
  - **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.
  - **Benchmark Applicability:** The benchmark parameters and their bounding values from which bias and bias uncertainty of a calculational method are established. [AoA]
  - **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. [extension of AoA]

# MCNP Verification & Validation Suites

## Verification Suites

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

## Validation Suites

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

## Overview of Validation Methods

---

- **Identify the range of applications to be considered**
  - Fissile material, geometry, reflection, moderation, etc.
  - Metrics to help characterize neutronics – EALF, % fast/thermal fissions, H/U or H/Pu for solutions, etc.
- **Select a set of experimental benchmarks from ICSBEP Handbook that are neutronically similar to the applications**
  - Must select sufficient number for valid statistical analysis
  - Analyze the set of benchmarks with Monte Carlo
- **Statistical analysis**
  - Determine bias & bias uncertainty for the set of benchmarks
  - For conservatism, usually set positive biases to zero & only consider negative biases for individual benchmarks
- **Estimate additional margin of subcriticality (MOS)**
  - Extra margin to account for nuclear data uncertainty
  - Extra margin to account for unknown code errors
  - Extra margin if applications not similar enough to benchmark set



## Upper Subcritical Limit

---

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

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

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

[additional AoA margin may be appropriate]

Note: Bias = calculated – experiment,

For conservatism – can set positive biases to zero; only consider negative biases

- 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.

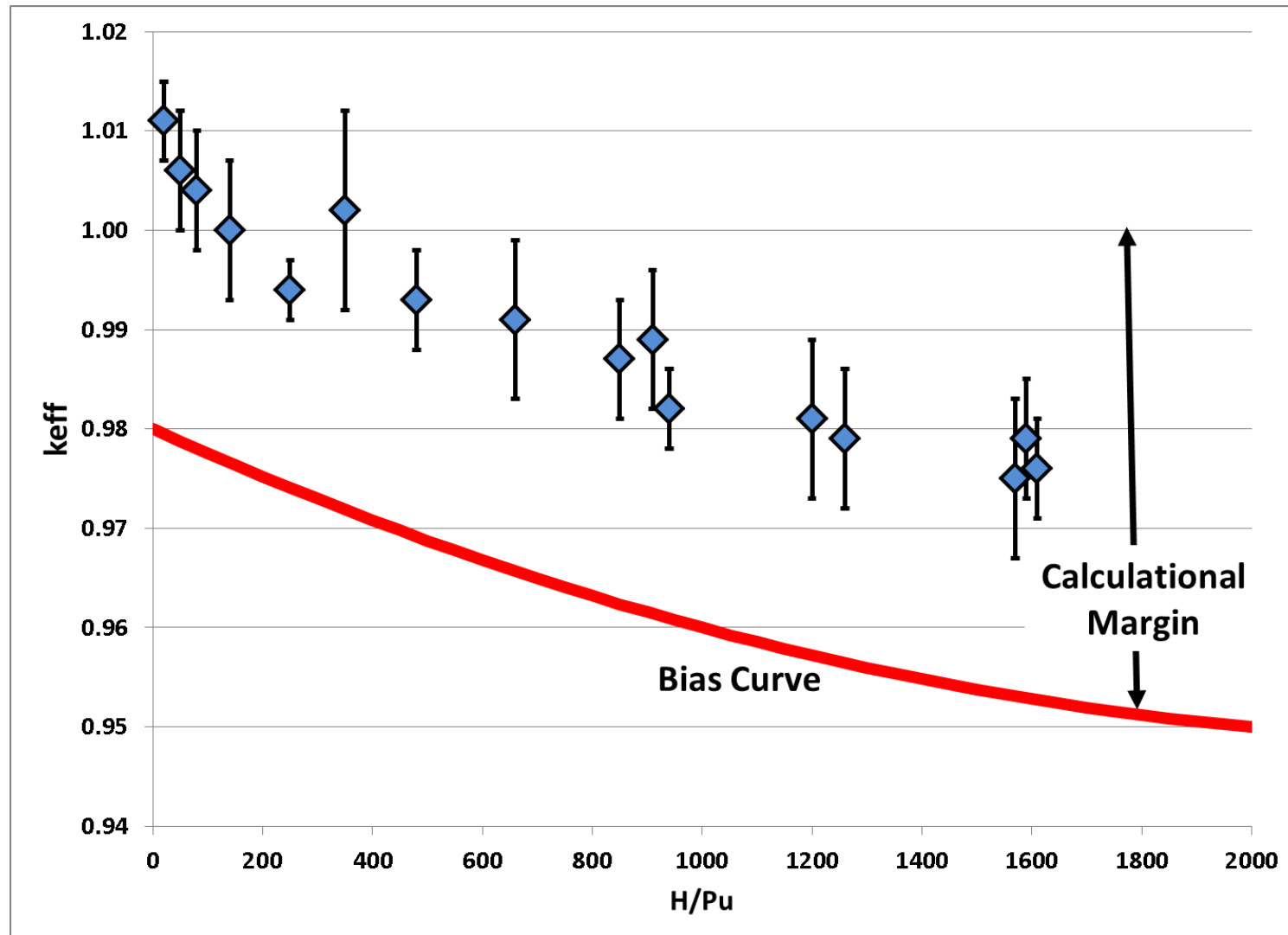
## Calculational Margin

---

- 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).

# Calculational Margin Example

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



## Margin of Subcriticality

---

- **To establish a Margin of Subcriticality (MOS) need to consider the process, validation, codes, data, etc. holistically.**
  - **Confidence in the codes and data.**
    - More mature codes that are widely used have greater confidence than newer ones.
    - Deterministic methods may 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.**

## Selection of Benchmarks

---

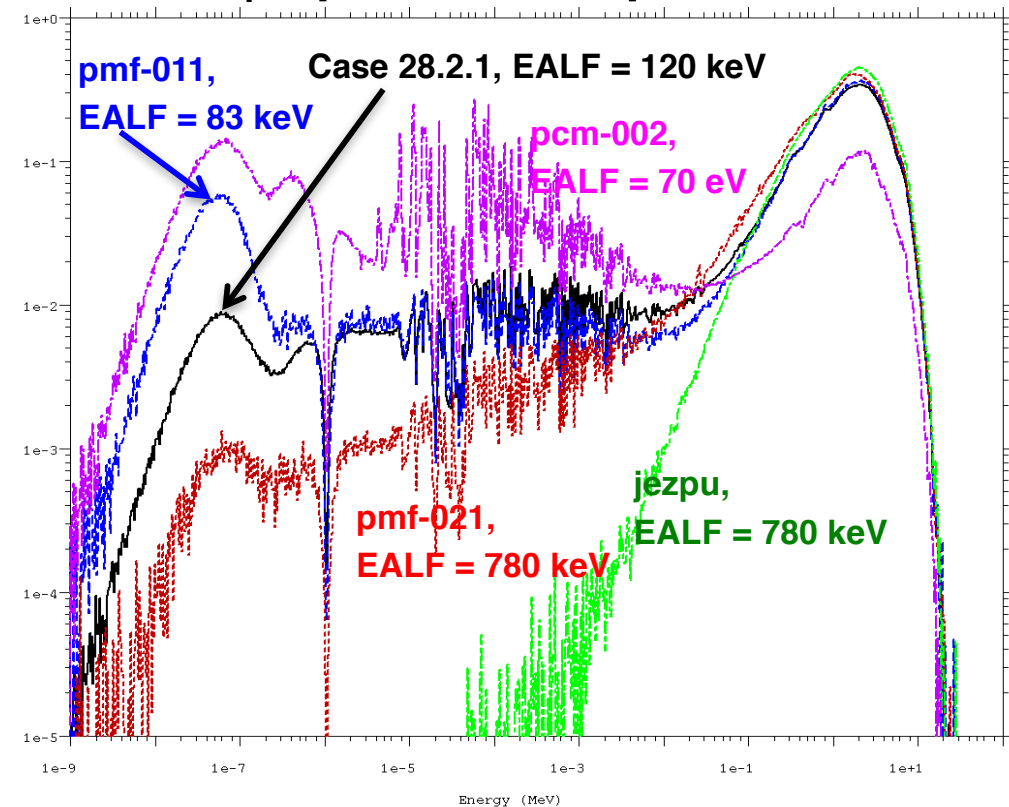
- **Select critical experiments that you expect to have the same bias as the criticality safety evaluation models**
  - Similar neutron energy spectrum (EALF, ANECF, etc.)
  - Similar fissionable materials and isotopics
  - Similar neutron absorbers (Cd, Gd, B, Fe, Ti, etc.)
  - Similar neutron reflectors (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**
  - Especially when considering upset conditions
- **How many experiments are needed?**
  - As many experiments that are similar or “applicable” to the criticality safety evaluation models for valid statistical analysis
  - If an experiment is exactly the same as the fissionable material operation, subcritical limits may be derived directly from experiments with no need to calculate the result
  - “Response to CSSG Tasking 2014-02, Validation with Limited Benchmark Data,” September 21, 2015, [http://ncsp.llnl.gov/cssg/taskandresponse/2014/2014-02\\_Response\\_on\\_Validation\\_with\\_Limited\\_Data\\_09-21-15.pdf](http://ncsp.llnl.gov/cssg/taskandresponse/2014/2014-02_Response_on_Validation_with_Limited_Data_09-21-15.pdf)
- If no benchmark experiments exist that match the system being evaluated, it may be possible to interpolate or extrapolate from existing benchmark data to that system. Sensitivity and uncertainty analysis tools may be used to assess the applicability of benchmark problems to the system being analyzed. (DOE-STD-3007-2007)

## Selection of Benchmarks

- **Historically, engineering judgement (“expert”) has been used**
  - Based on the analysts understanding of what is important to the problem
- **This can 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.

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)

### $\nu\Sigma_F\Phi$ production spectrum



### 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

---

### **ANSI/ANS-8.24 Section 5**

- **Identify the parameters that correlate experiments to the system or process being analyzed in the criticality safety evaluation**
- **Normal and credible abnormal conditions shall be considered when determining the parameters and range of parameters**
  - 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**
  - 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**
  - 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**

## Selection of Benchmarks

---

### ANSI/ANS-8.24 Section 5

- **Benchmarks should be drawn from multiple sources to minimize systemic error**
- **Methods used to analyze benchmarks shall be the same *computational method* being used in the criticality safety evaluation**
  - 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***
- **Benchmark models prepared by outside organizations should be evaluated for appropriateness, completeness & accuracy**
  - 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:**
  - 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 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  $k_{\text{eff,normalized}}$ )
- 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.
- 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

## S/U Analysis

---

- **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:**
  - Data can be used to
    - Select benchmarks that are similar to the application
    - 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)
    - 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
    - As a trending parameter in USL determination
- **CSSG Response on Validation with Limited Data:** *“For those situations where a nuclide is determined to be important and limited data exist, validation may still be possible. However, an additional margin should be used to compensate for the limited data. This margin is separate from, and in addition to, any margin needed for extending the benchmark applicability to the validation. Sensitivity and uncertainty tools may be used as part of the technical basis for determining the magnitude of the margin.”*

## Comparison of Validation Approaches (Simplified)

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

# Documentation and Independent Technical Review

---

- **Documentation:**

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

- **Independent Technical Review:**

- review benchmark applicability
- Input files and output files
- Methodology for determining bias, bias uncertainty, margins
- Concurrence with validation applicability

---



# Neutron Spectra

- Neutron slowing down theory
- Lethargy
- Neutron spectra
- Resonance absorption
- Spectral indicators
- Examples

# Neutron Slowing Down Theory

- Consider the transport equation for:

- Infinite medium of hydrogen
- Steady source at energy  $E_s$
- Isotropic elastic scatter
- Scattering nuclides are stationary, no upscattering occurs
- No absorption

$$\cancel{\Omega \cdot \nabla} \phi(E) + \Sigma_T(E)\phi(E) = \int_E^{E_s} dE' \Sigma_s(E' \rightarrow E)\phi(E') + S \cdot \delta(E - E_s)$$

- For hydrogen at rest ( $E \gg kT$ )  $\Sigma_s(E' \rightarrow E) = \frac{\Sigma_s(E')}{E'}$

- Slowing down in hydrogen at rest:

$$\Sigma_s(E)\phi(E) = \int_E^{E_s} dE' \frac{\Sigma_s(E')}{E'} \phi(E') + S \cdot \delta(E - E_s)$$

- Solution

$$\phi(E) = \frac{S}{\Sigma_s(E) \cdot E} + \frac{S}{\Sigma_s(E)} \delta(E - E_s)$$

## Slowing Down Theory - Lethargy

- For theory, visualization, understanding, it is useful to change variables from energy (E) to lethargy (u)

$$u = \ln \frac{E_0}{E}, \quad \text{where } E_0 \text{ is large, eg 20 MeV}$$

$$du = -\frac{dE}{E}, \quad E = E_0 e^{-u}$$

$$\phi(u) = \left| \frac{dE}{du} \right| \phi(E) = E \cdot \phi(E)$$

– As energy decreases, lethargy increases

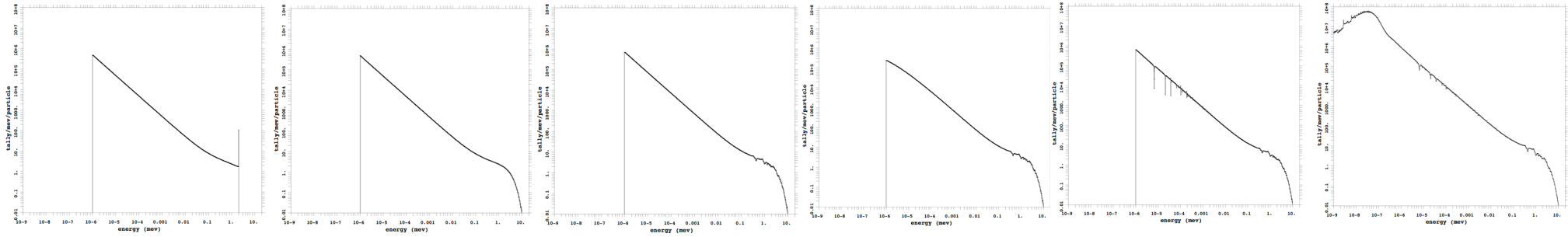
- Consider slowing down flux in hydrogen,  $E < E_s$

$$\phi(E) = \frac{S}{\Sigma_s(E) \cdot E} \sim \frac{1}{E}$$

$$\phi(u) = \frac{S}{\Sigma_s(u)} \sim \text{constant}$$

# Flux Spectra for Neutron Slowing Down & Criticality

## loglog plots of $\phi(E)$ vs $E$



2 MeV neutrons  
hydrogen

fission neutrons  
hydrogen

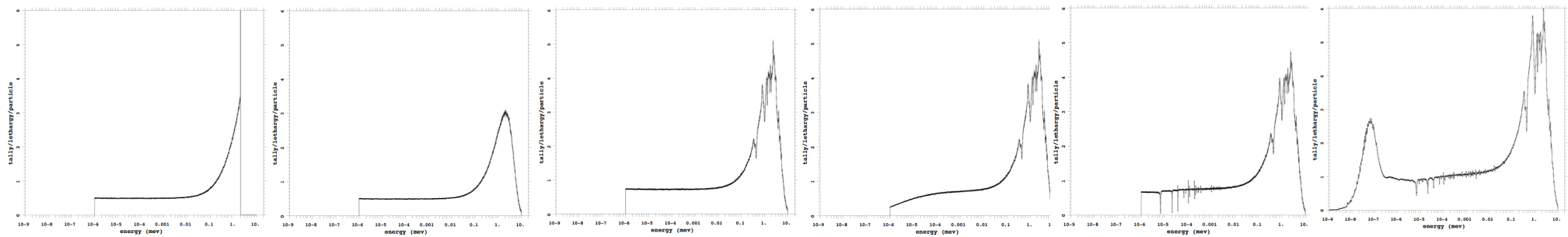
fission neutrons  
water

fission neutrons  
water + B<sup>10</sup>

fission neutrons  
water + U<sup>238</sup>

Fuel Pin  
Unit Cell

## loglin plots of $\phi(u)$ vs $u$



2 MeV neutrons  
hydrogen

fission neutrons  
hydrogen

fission neutrons  
water

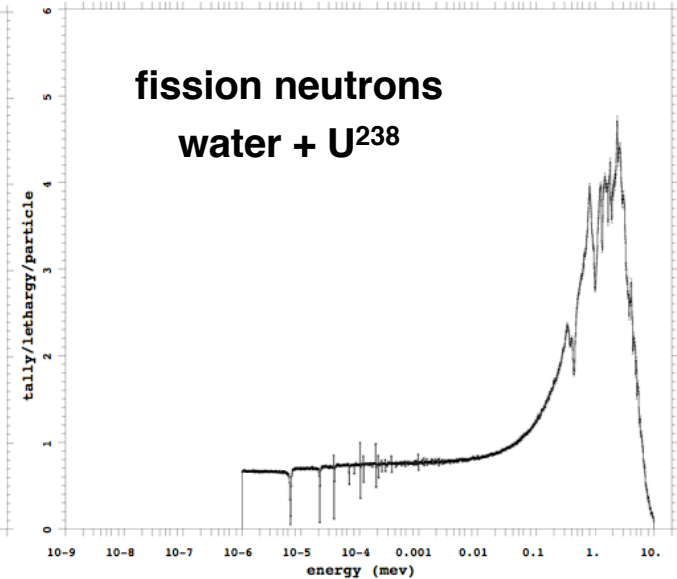
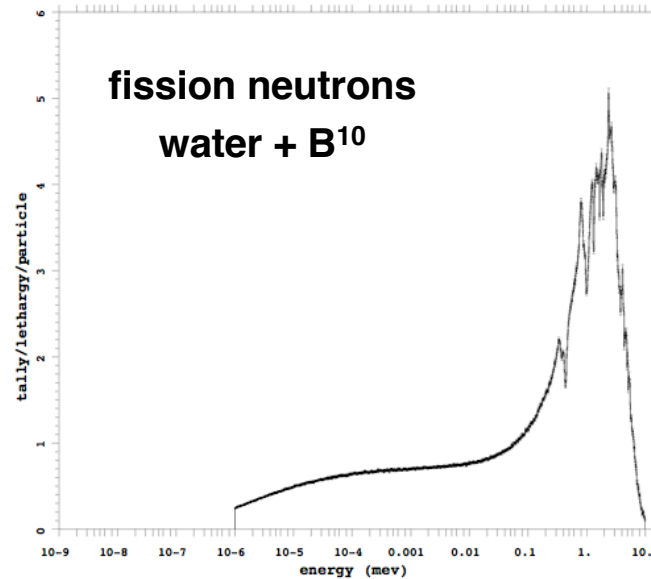
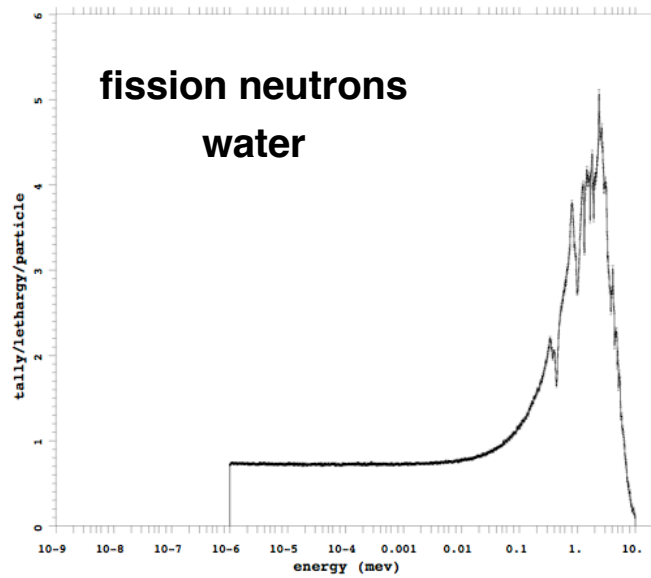
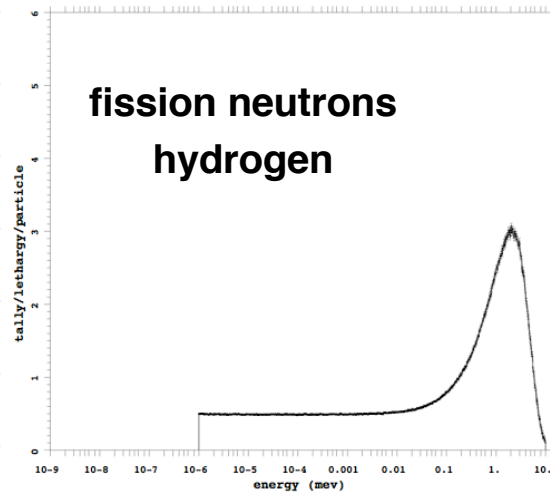
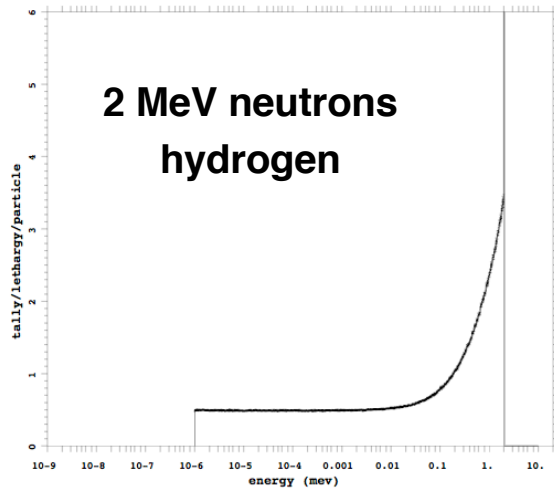
fission neutrons  
water + B<sup>10</sup>

fission neutrons  
water + U<sup>238</sup>

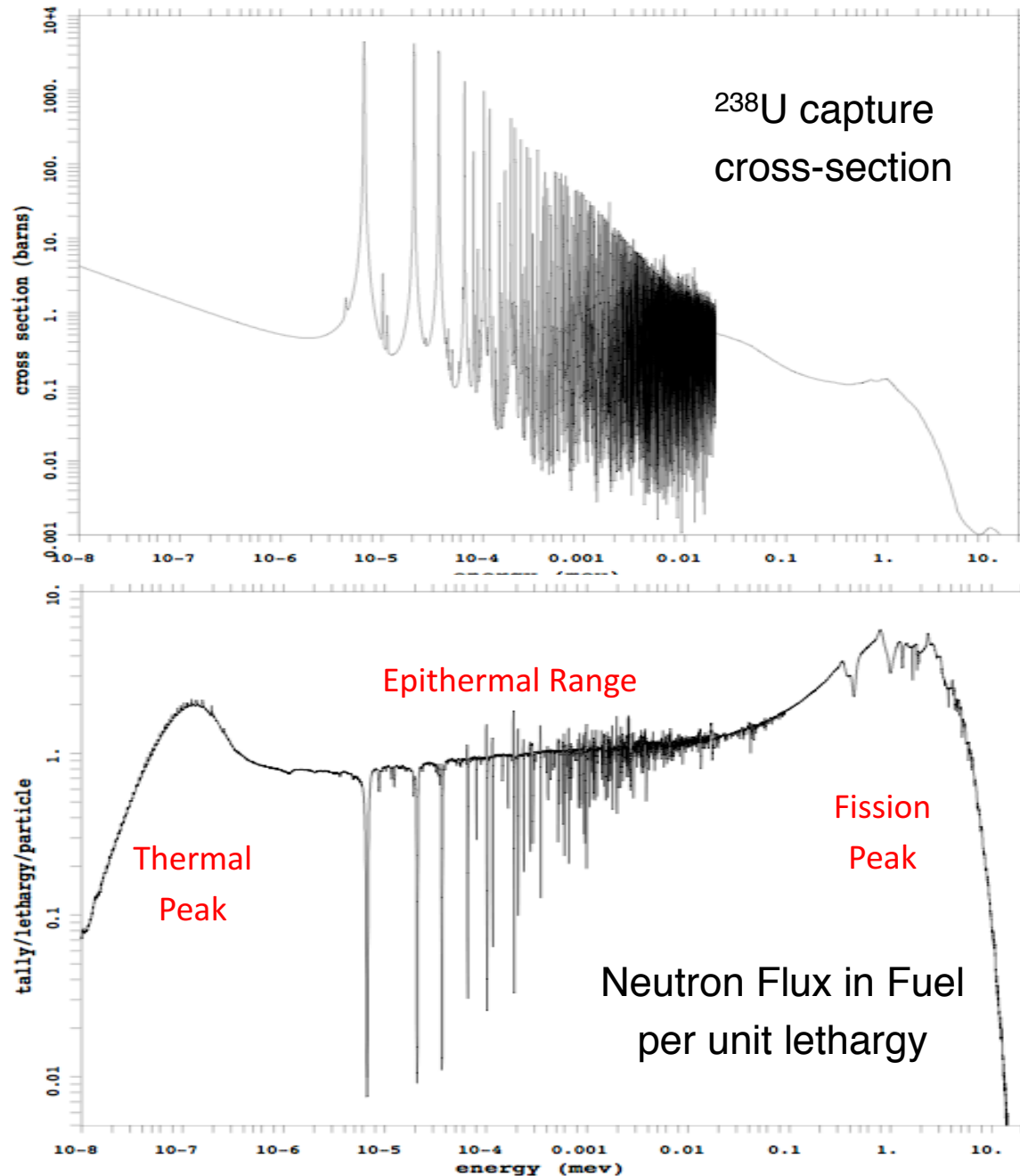
Fuel Pin  
Unit Cell

# Flux Spectra for Neutron Slowing Down

loglin plots of  
 $\phi(u)$  vs  $u$



# UO<sub>2</sub> Fuel Pin



## UO<sub>2</sub> Fuel Pin

3.1% Enriched

293.6 °K

.01 eV – 20 MeV

- Neutrons born in MeV range from fission
- Most fissions caused by thermal neutrons
- **1/3 of neutron losses are due to <sup>238</sup>U capture in epithermal energy range during slowing down**

## Characterizing the Neutron Spectrum

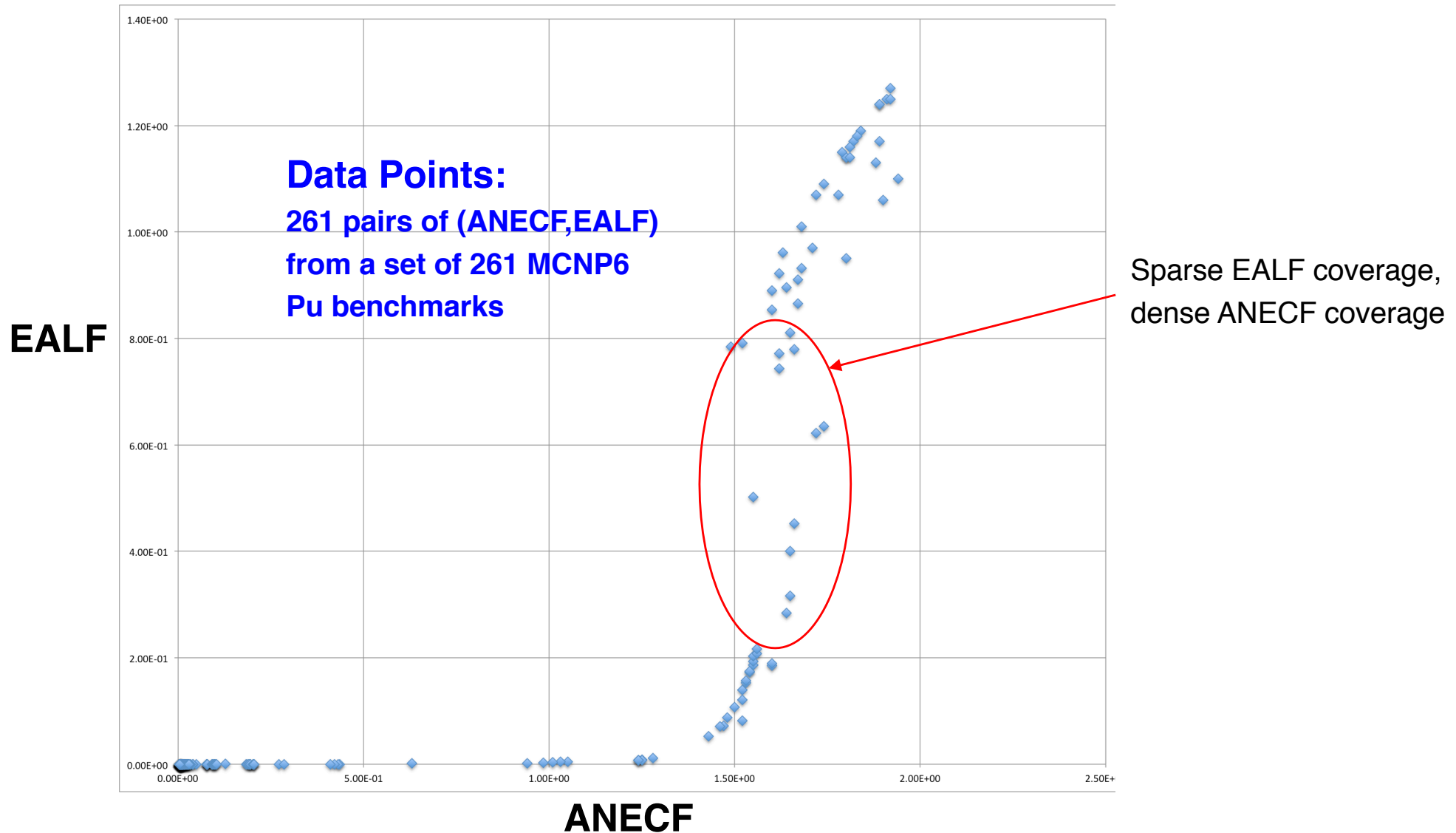
---

- The neutron spectrum –  $\Phi(E)$  or  $\Phi(u)$  – is a complex function of geometry, materials, isotopes, reflectors, temperature, cross-sections, ...
- Many different **spectral index** parameters can be used to characterize the spectrum
  - **EALF** – energy corresponding to the average lethargy of neutrons causing fission
  - **ANECF** – average energy of neutrons causing fission
  - Above thermal leakage fraction
  - H/Pu <sup>239</sup> or H/U<sup>235</sup> ratios, for solutions
  - % fissions caused by fast, intermediate, thermal neutrons
  - (U238 fission)/(U235 fission), and other ratios
  - etc.
- These parameters are useful for comparing different reactors or benchmark experiments, in looking for trends in code or cross-section accuracy
- **Spectrum hardness** is often characterized by one of these parameters
- No single parameter tells the whole story

# EALF vs ANECF

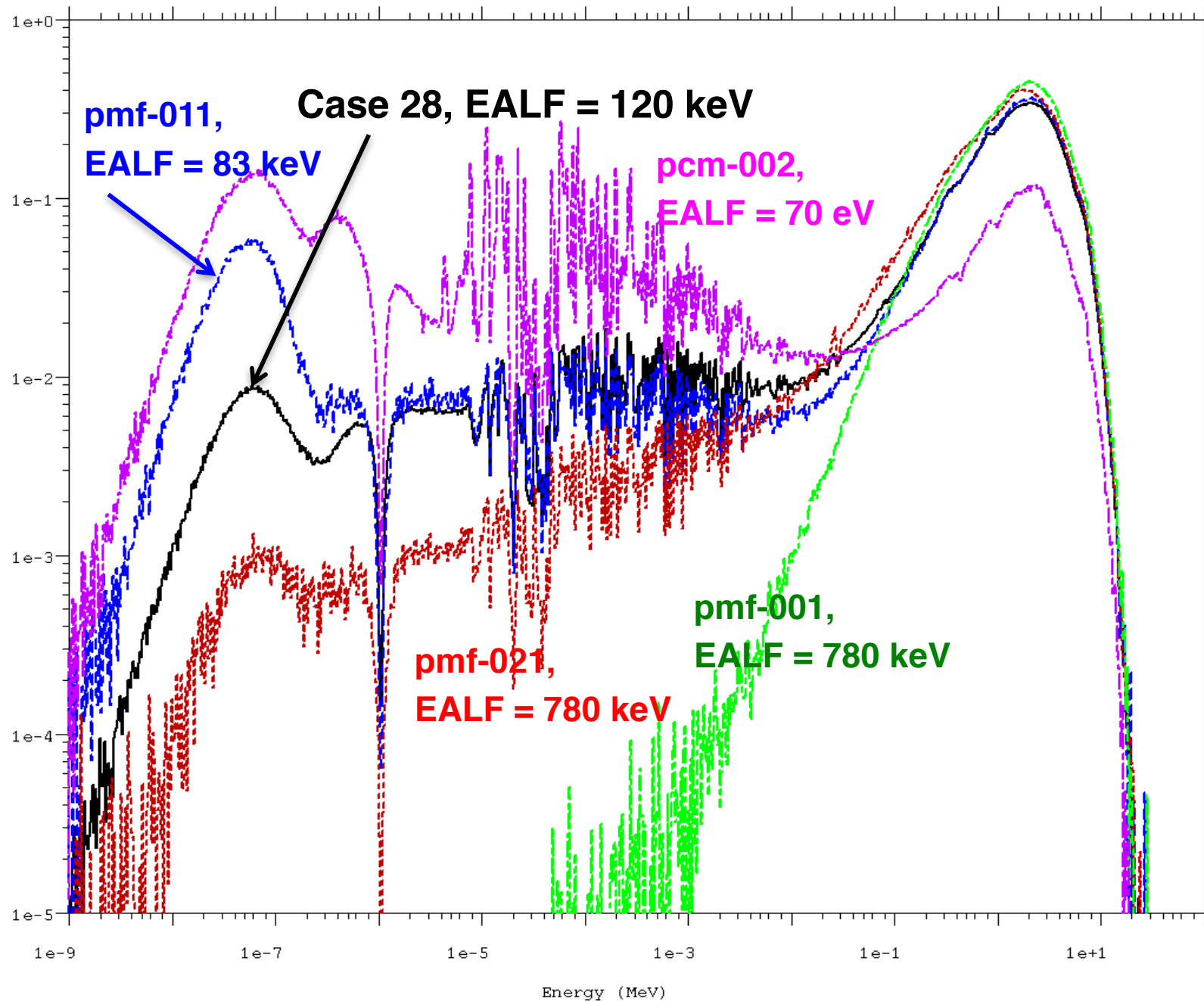
**ANECF = average neutron energy causing fission**

**EALF = energy of the average neutron lethargy causing fission**





# Pu Systems – $\nu\Sigma_\phi\Phi$ production & spectrum hardness





# Nuclear Data Sensitivities

## Introduction & Objectives

---

- **MCNP can produce sensitivity profiles to determine which data most impacts criticality.**
- **Learning Objectives:**
  - Understand the meaning of a sensitivity coefficient
  - Comprehend the techniques used by MCNP to estimate those tallies
  - Use the KSEN card to generate both energy-integrated and energy-resolved sensitivity profiles for specific reactions
  - Understand sensitivity output file information

## Motivation (1)

---

- **Nuclear cross sections are a major driver for criticality, and their uncertainties usually the largest source of bias in calculations.**
- **Knowing which data most impacts criticality is useful for:**
  - Critical experiment design
  - Uncertainty quantification and bias assessment
  - Code validation
  - Nuclear data adjustment and qualification
- **Validation requires selecting benchmarks that are appropriate for the process being analyzed.**
  - One method of picking appropriate benchmarks is to find the ones where the system multiplication is impacted by the same nuclear data.
  - For example, if the process  $k_{\text{eff}}$  is very sensitive to thermal plutonium capture, you should find benchmarks where the same is true.
- **Critical experiment design**
  - Often experiments are performed to address some defined nuclear data need.
  - Nuclear data sensitivities can determine if the as-designed experiment meets that need.

# Sensitivity Coefficient

---

- For criticality problems, often want to know:
  - How sensitive is  $K_{eff}$  to uncertainty in some parameter ?
- The **sensitivity coefficient** is defined as the ratio of relative change in a response to a relative change in a system parameter:

$$S_{R,x} = \frac{\Delta R / R}{\Delta x / x}$$

- Here, the response is the system multiplication  $k$  and the parameter  $x$  is some nuclear data (cross section).
- For a very small change in system parameter  $x$ :

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

## Sensitivity Coefficient

---

- This may be expressed using perturbation theory:

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

- This includes both the forward and adjoint neutron fluxes.
- The boldface S and F are shorthand for scattering and fission integrals of the transport equation.
- The x subscript implies that the quantity is just for data x.

## Adjoint Transport Equation

---

- The adjoint transport equation:

$$\begin{aligned}
 -\mathbf{\Omega} \cdot \nabla \psi^\dagger(\mathbf{r}, \mathbf{\Omega}, E) + \Sigma_t \psi^\dagger(\mathbf{r}, \mathbf{\Omega}, E) = \\
 \iint dE' d\mathbf{\Omega}' \Sigma_s(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}, E \rightarrow E') \psi^\dagger(\mathbf{r}, \mathbf{\Omega}', E') \\
 + \frac{1}{k_{\text{eff}}} \iint dE' d\mathbf{\Omega}' \chi(E \rightarrow E') \nu \Sigma_f(\mathbf{r}, E) \psi^\dagger(\mathbf{r}, \mathbf{\Omega}', E')
 \end{aligned}$$

- Adjoint fundamental mode has physical meaning:

The importance at a location in phase space is proportional to the expected value of a measurement, caused by a neutron introduced into a critical system at that location, after infinitely many fission generations.

- The iterated fission probability method is based on this concept, & can be used to determine adjoint or importance weighting for Monte Carlo tallies



## Example – Need for Adjoint-Weighting

- MCNP can compute lifetimes (prompt removal times) with non-importance weighted tallies:

unweighted

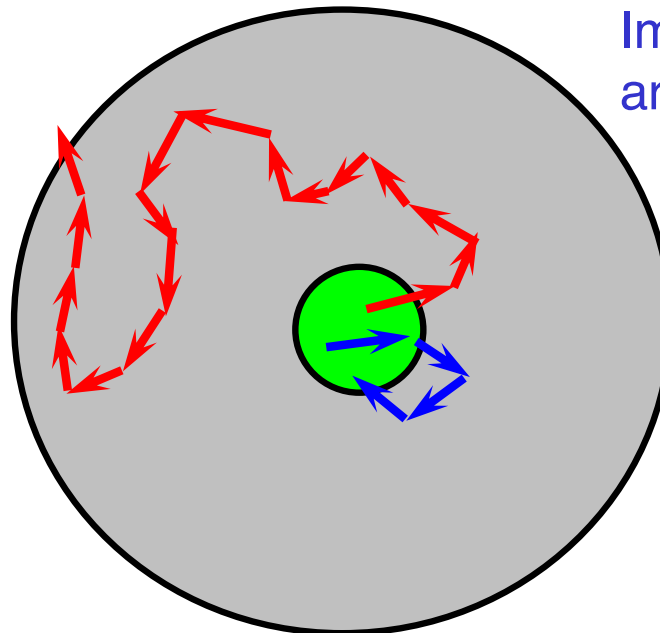
$$\Lambda_{rem} = \frac{\langle 1, 1/v \psi \rangle}{\langle 1, F\psi \rangle}$$

adjoint-weighted

$$\Lambda_{eff} = \frac{\langle \psi^\dagger, 1/v \psi \rangle}{\langle \psi^\dagger, F\psi \rangle}$$

- Example: Importance weighting is necessary in systems with thick reflectors. Unweighted lifetimes are often very much larger than effective lifetimes (adjoint-weighted)**

Neutrons spending significant time deep in the reflector are unlikely to cause fission and are therefore unimportant



Important neutrons are often short-lived

**Net Effect:** Not weighting by importance overvalues long-lived neutrons leading to lifetimes much too long.

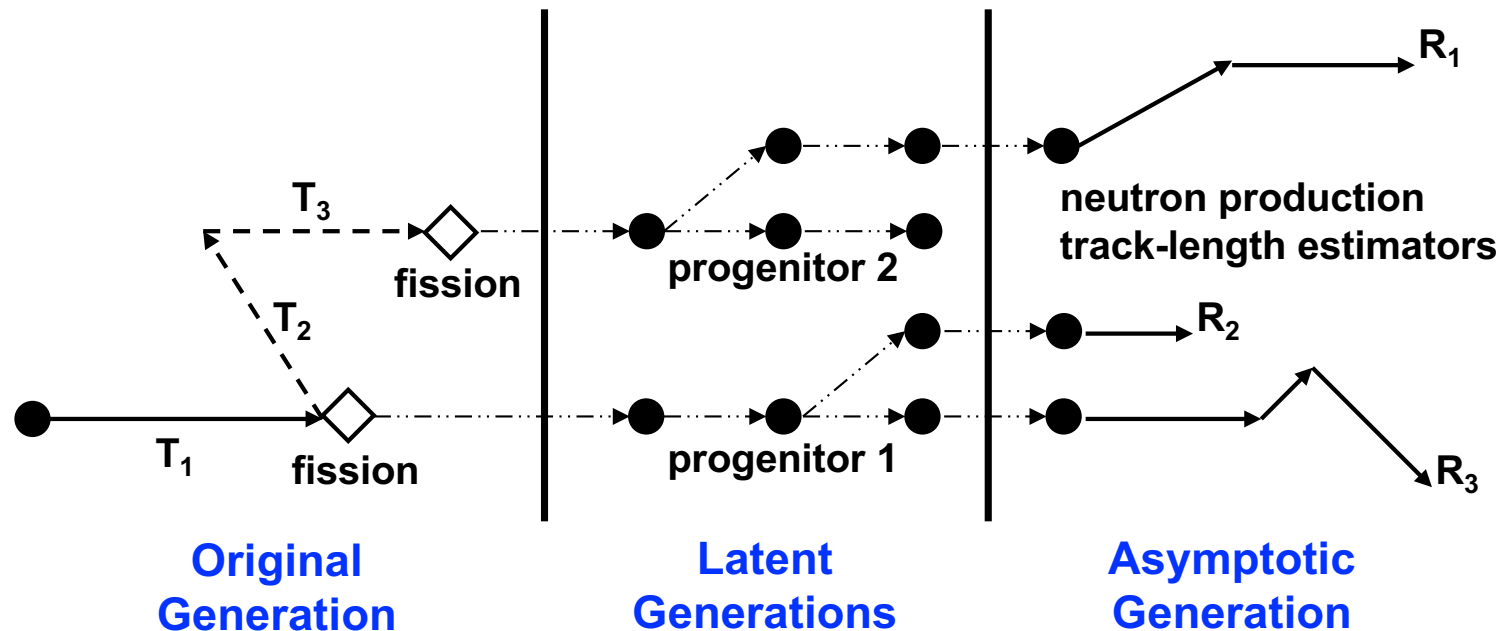
Example: Flattop

$$\lambda_{No-wgt} = 67.1 \text{ ns}$$

$$\lambda_{Adj-wgt} = 17.5 \text{ ns}$$

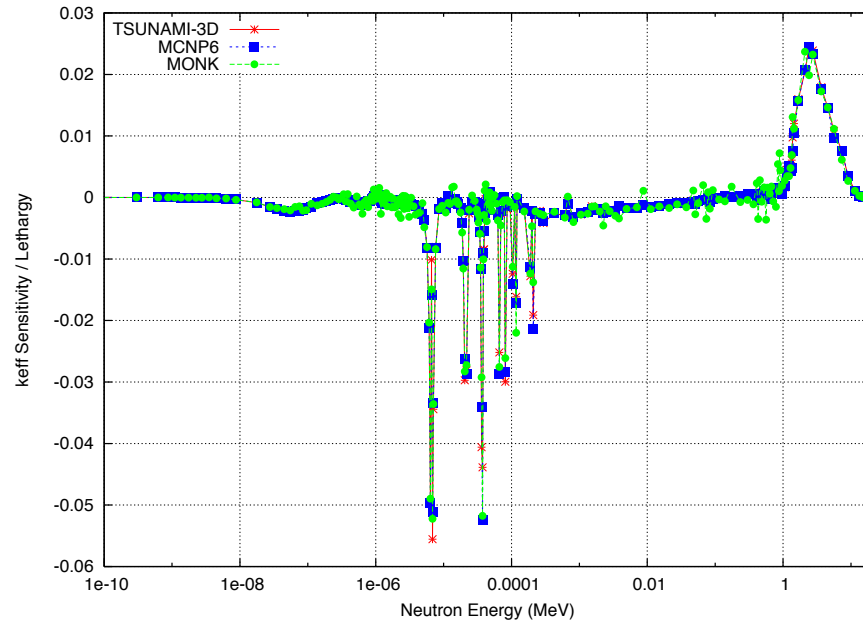
# MCNP Implementation

- MCNP breaks active cycles into consecutive blocks:
  - Tally **scores** are collected in **original generation**, & progenitor neutrons tagged
  - All subsequent progeny within the **latent generations** remember their progenitor
  - **Importance** is the population of progeny from each progenitor in the **asymptotic generation**
  - **(Score)\*(importance)** is tallied for adjoint-weighted results

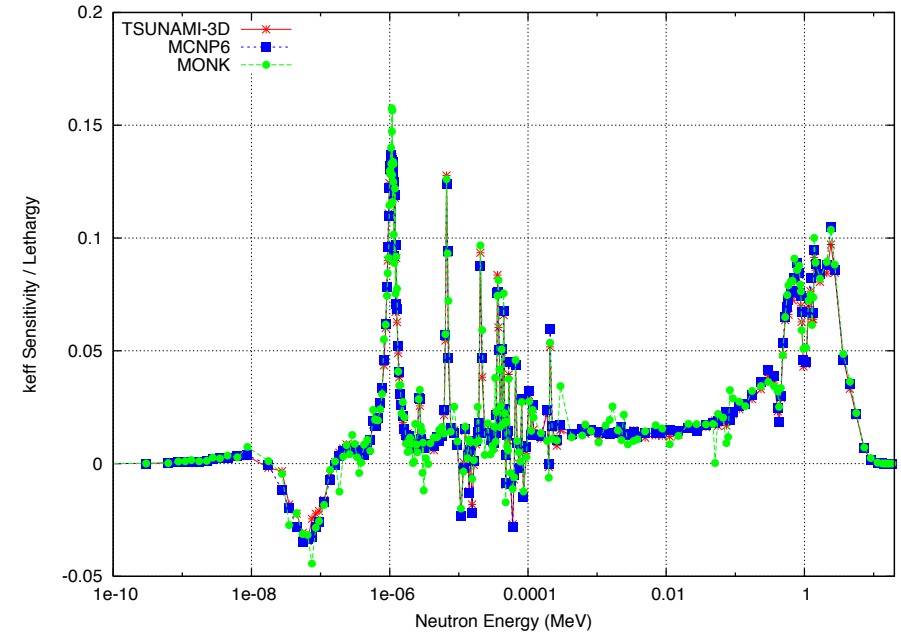


# Example Sensitivity Coefficient Profiles

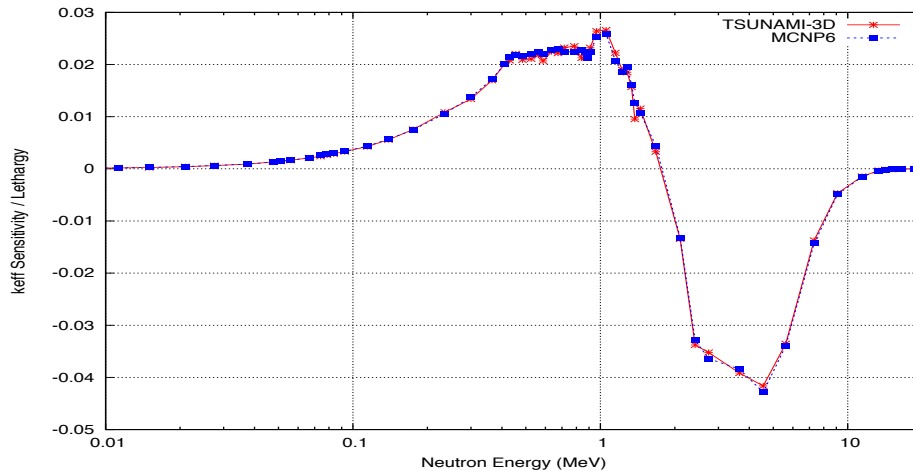
**U-238: total cross-section sensitivity**  
OECD/NEA UACSA Benchmark Phase III.1



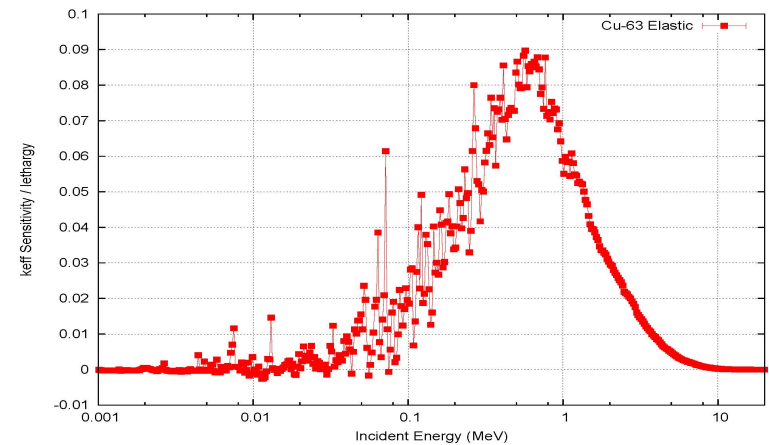
**H-1: elastic scattering cross-section sensitivity**  
OECD/NEA UACSA Benchmark Phase III.1



**Pu-239: fission  $\chi(E)$  sensitivity**  
OECD/NEA UACSA Benchmark Phase III.1



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



## MCNP6 - KOPTS Card

---

- **KOPTS controls many special features for KCODE calculations**
- **For keff sensitivity calculations, KOPTS is used to control the following:**
  - **Size of the blocks (default is 10 cycles)**
  - **Sensitivity output printing (default is just to the output file).**
- **Format:**  
**KOPTS   BLOCKSIZE= N   KSENTAL= FILEOPT**
- **For now, the only “FILEOPT” allowed is MCTAL, which has MCNP produce a special MCTAL results file**

## MCNP6 - KSEN Card

---

- **Format for nuclear data:**

**KSENj XS ISO= ZAID1 ZAID2 ... RXN= MT1 MT2 ...  
ERG= E1 E2 ...**

- **Notes:**

- *j* is an arbitrary user index (> 0).
- XS defines the type of sensitivity (XS only allowed for now).
- ISO is followed by a list of ZAIDS or S(a,b) identifiers (e.g., 92235.70c, default is all isotopes).
- RXN is a list of MT numbers (default is total, see next slide for a shortened list).
- ERG is a user-defined energy grid in MeV (default 0 to infinity).
- More options available for secondary distributions (e.g., chi).
- Multiple instances of KSEN are allowed, so long as they have a different user index *j*.

## MCNP6 - KSEN Reaction MT numbers

---

- **Partial list of valid reaction MTs for KSEN**

– Total	1
– Capture	-2
– N,Gamma	102
– Elastic Scattering	2
– Inelastic Scattering	4
– Fission	-6
– Fission Nu	-7
– N,2N	16
– Fission Chi	-1018
– Elastic Law	-1002

## MCNP6 - KSEN Examples

---

- Capture cross section sensitivity for all isotopes

```
kzen1    xs    rxn= -2
```

- U-238 elastic and inelastic scattering sensitivities

```
kzen2    xs    iso= 92238.70c    rxn=  2  4
```

- H-1 and light-water S(a,b) total sensitivity with uniform lethargy grid from 1e-5 eV to 100 MeV

```
kzen3    xs    iso=  1001.70c    lwtr.10t  
          rxn=   1  
          erg= 1.e-11  12ilog 1e+2
```

## MCNP6 Example 1: KSEN Card

---

- Copy **puc6.txt** from SOLUTIONS directory to **ksen1.txt**.
- Find sensitivities to 3 x 2 array of cans containing plutonium nitrate solution.
  - Set KCODE card to use 5000 neutrons per cycle, skip 50, and run 250 cycles total.
  - Set KOPTS card to have a BLOCKSIZE of 5.
  - Add a cross section sensitivity card with no arguments, i.e., use all defaults

```
kcode    5000    1.0    50    250
...
c
c ### keff sensitivity cards
c
kopts    blocksize = 5
c
c default ksen, get total xs sensitivity to all isotopes
ksen1    xs
```

- Run the problem and analyze output.



# MCNP6 Exercise 1: Results

nuclear data keff sensitivity coefficients

sensitivity profile 1

energy range: 0.0000E+00 1.0000E+36 MeV

isotope	reaction	sensitivity	rel. unc.
1001.70c	total	4.7564E-01	0.0589
7014.70c	total	-1.0670E-02	0.5088
8016.70c	total	1.2197E-01	0.1225
24050.70c	total	-9.1837E-05	4.4999
24052.70c	total	2.5948E-03	0.3650
24053.70c	total	7.2096E-04	0.8493
24054.70c	total	1.5180E-05	7.5290
26054.70c	total	-4.5558E-04	0.8763
26056.70c	total	1.3197E-02	0.1791
26057.70c	total	7.9241E-04	0.5101
...			
94239.70c	total	8.1218E-02	0.0919
94240.70c	total	-4.5498E-02	0.0288
94241.70c	total	7.6258E-04	0.1957
94242.70c	total	-6.0798E-05	0.0480
lwtr.10t	total	1.6518E-01	0.1716

- Total cross section sensitivities can also be thought of as the sensitivity to the atomic density
- Observations:
  - Water (hydrogen and oxygen) have the most impact on k in this system.
  - Pu-239 has a significant, but smaller impact.
  - Other significant, but less important, isotopes are Pu-240 and Fe-56.
- Pu-239 total sensitivity is small for a dominant fissile isotope
  - Investigate this by decomposing this into specific reactions

## MCNP6 Exercise 2: Sensitivities by Reaction

---

- Copy **ksen1.txt** to **ksen2.txt**.
- Find sensitivities of total, capture, elastic, inelastic, and fission for H-1, light-water S(a,b), O-16, and Pu-239
  - Delete the old KSEN card and insert a new one

```
c
c ### keff sensitivity cards
c
kopts    blocksize= 5
c
c reaction sensitivities for h-1, o-16, pu-239
c capture, elastic, inelastic, fission
ksen2    xs      iso= 1001.70c lwtr.10t 8016.70c 94239.70c
          rxn=    1 -2  2  4 -6
```

- Run the problem and analyze output.

## MCNP6 Exercise 2: Results

1001.70c	total	4.7564E-01	0.0589
1001.70c	capture	-4.1980E-02	0.0110
1001.70c	elastic	5.1762E-01	0.0541
1001.70c	inelastic	0.0000E+00	0.0000
1001.70c	fission	0.0000E+00	0.0000
lwtr.10t	total	1.6518E-01	0.1716
lwtr.10t	capture	0.0000E+00	0.0000
lwtr.10t	elastic	0.0000E+00	0.0000
lwtr.10t	inelastic	1.6518E-01	0.1716
lwtr.10t	fission	0.0000E+00	0.0000
8016.70c	total	1.2197E-01	0.1225
8016.70c	capture	-1.3346E-03	0.0491
8016.70c	elastic	1.2219E-01	0.1219
8016.70c	inelastic	1.1203E-03	0.2583
8016.70c	fission	0.0000E+00	0.0000
94239.70c	total	8.1218E-02	0.0919
94239.70c	capture	-3.0413E-01	0.0076
94239.70c	elastic	-1.3872E-03	1.2795
94239.70c	inelastic	6.1685E-04	0.8563
94239.70c	fission	3.8605E-01	0.0140

- Elastic scattering with H-1 and O-16 are important, as is inelastic thermal scattering with H-1 in H2O molecule.
- Pu-239 fission and capture are of similar opposing magnitude, which is the cause of a lower than normal sensitivity to keff.
- Analyze Pu-239 capture and fission as function of energy.

## MCNP6 Exercise 3: Sensitivities by Energy

---

- Copy **ksen2.txt** to **ksen3.txt**.
- Find sensitivities of Pu-239 capture and fission as function of energy.
  - Delete the old KSEN card and insert a new one.
  - For the energy bins, use 0 to 0.625 eV, 0.625 eV to 100 keV, and 100 keV to 100 MeV as thermal, intermediate, and fast.

```
c
c ### keff sensitivity cards
c
kopts    blocksize = 5
c
c pu-239 capture and fission sensitivity for thermal,
intermediate, and fast
ksen3    xs      iso = 94239.70c
          rxn    = -2  -6
          erg    = 0   0.625e-6    0.1    100
```

- Run the problem and analyze output.

## MCNP6 Exercise 3: Results

94239.70c capture

energy range (MeV)		sensitivity	rel. unc.
0.0000E+00	6.2500E-07	-2.7413E-01	0.0084
6.2500E-07	1.0000E-01	-2.9833E-02	0.0124
1.0000E-01	1.0000E+02	-1.7170E-04	0.0066

94239.70c fission

energy range (MeV)		sensitivity	rel. unc.
0.0000E+00	6.2500E-07	3.3226E-01	0.0184
6.2500E-07	1.0000E-01	4.2493E-02	0.0556
1.0000E-01	1.0000E+02	1.1298E-02	0.1122

- Most of the effect for fission and capture are in the thermal range (as expected).
- Both thermal and intermediate Pu-239 capture and fission are of similar magnitude.
- Fast Pu-239 capture is negligible relative to Pu-239 fission.

## MCNP6 - KSEN with Secondary Distributions

---

- **More complete KSEN:**

**KSENj XS**

**ISO = ZAID1 ZAID2 ...**

**RXN = MT1 MT2 ...**

**ERG = E1 E2 ...**

**COS = C1 C2 ...**

**EIN = I1 I2 ...**

**CONSTRAIN = YES/NO**

- **Comments:**

- For secondary distributions ERG is with respect to outgoing energies (default 0 to infinity).
- COS defines direction cosine changes from the collision (default -1 to 1)
- EIN defines the incident energy range (default 0 to infinity)
- CONSTRAIN tells MCNP whether the distribution must be renormalized to preserve probability (default is YES)
- If cross sections or fission nu listed in RXN, MCNP will calculate those as normal.

## MCNP6 - Constrained Chi Sensitivity Example

---

- **KSEN card of Pu-239 chi sensitivity:**

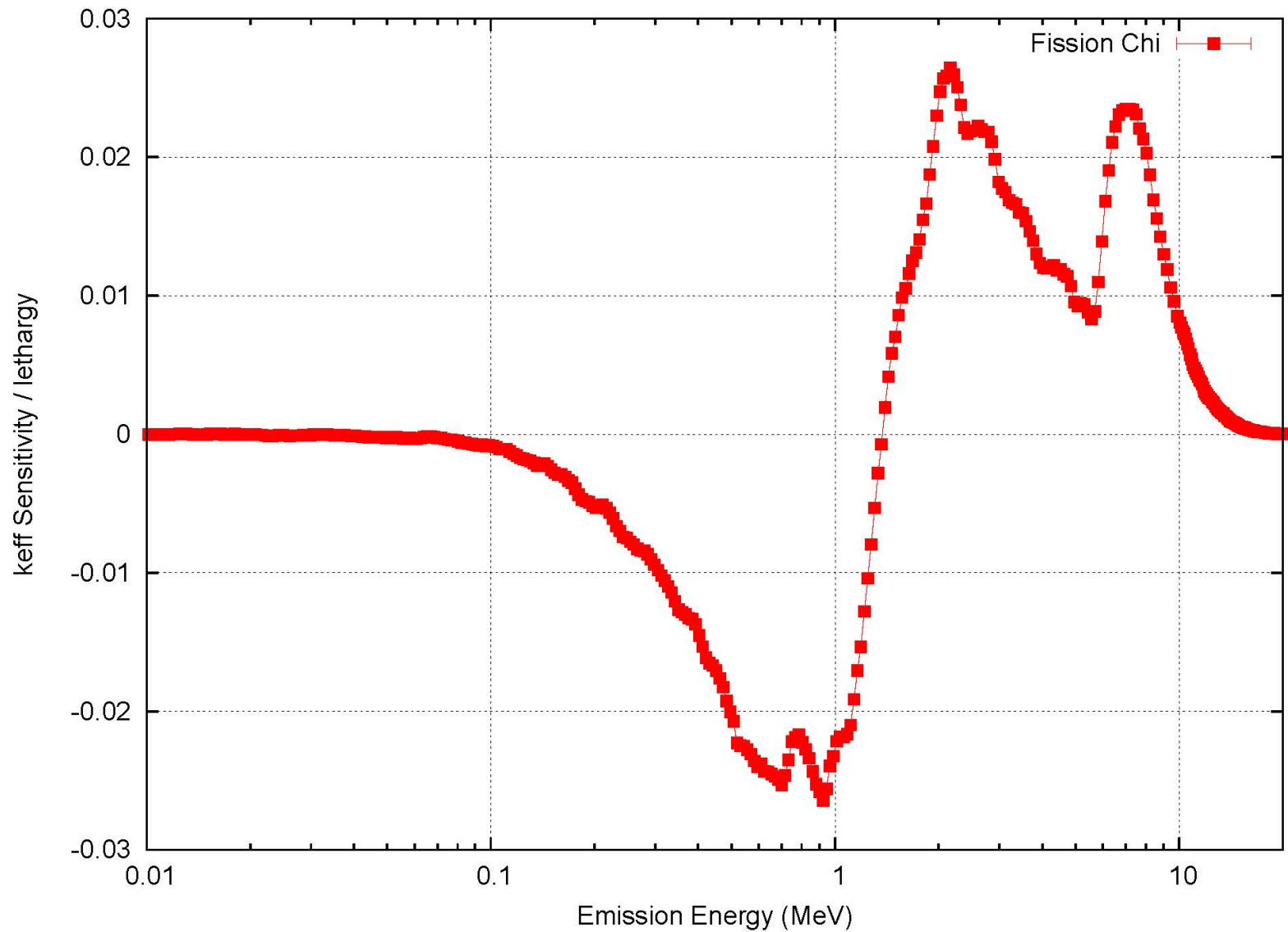
```
ksen94  xs    iso= 94239.70c  
          rxn= -1018  
          erg= 1e-11 999ilog 20  
          ein= 0 19i 20  
          constrain= yes
```

- **Comments:**

- Fine outgoing energy binning in lethargy
- Incident energy bins are in 1 MeV intervals from 0 to 20 MeV
- MCNP should give a sensitivity to a distribution that is renormalized

## Constrained Chi Sensitivity Example

- Pu-239 chi sensitivity in Jezebel (Pu Sphere):







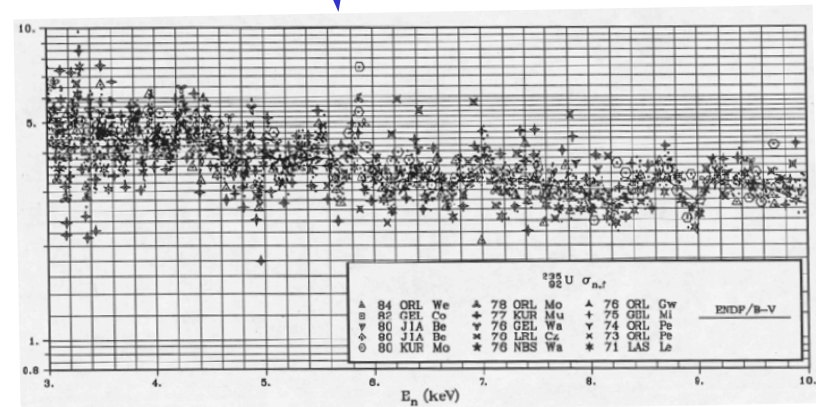
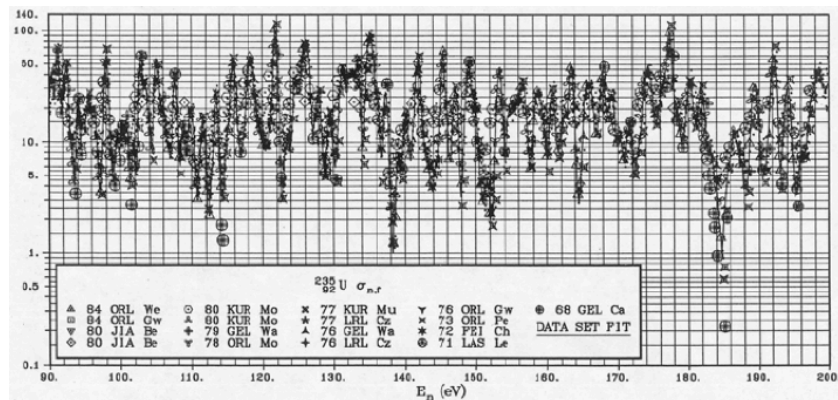
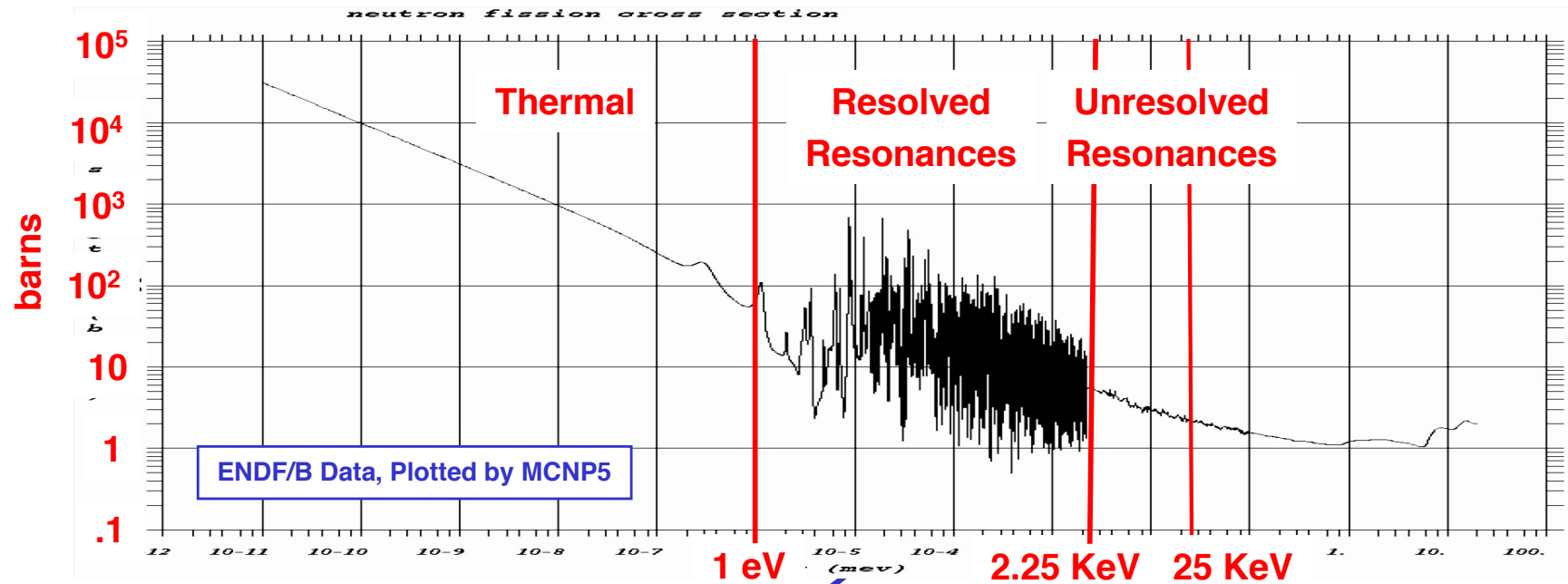
# Covariance Data

## Nuclear Data

---

- **“Nuclear” data involves interactions of incident particles with the nucleus.**
  - Data libraries include cross-section and scattering data with interpolation laws, various parameters, etc., derived from both experiments and theory
  - Typically there are "ladders" of  $(E_J, \sigma_J)$  pairs, but many other formats are also used.
- **Results obtained from a calculation depend upon both the code and the nuclear data it employs**
- **Along with the evaluated nuclear cross sections, angular distributions, energy spectra, etc., the uncertainties of the nuclear data can be a large source of the overall uncertainty in any application**
  - Both experimental and theoretical uncertainties contribute to the evaluated nuclear data uncertainties
  - The uncertainties are given in the form of a covariance matrix

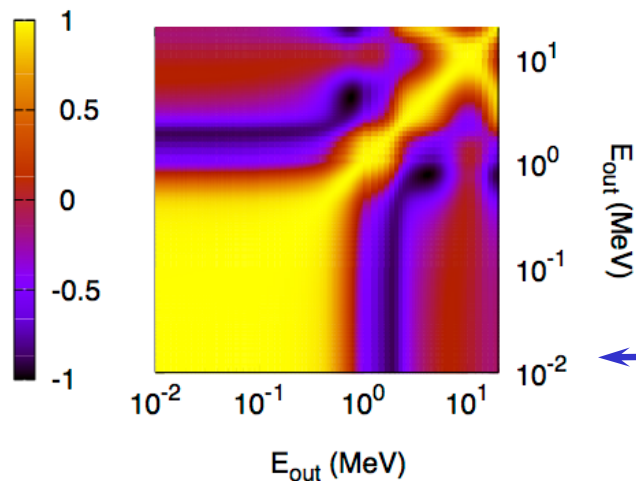
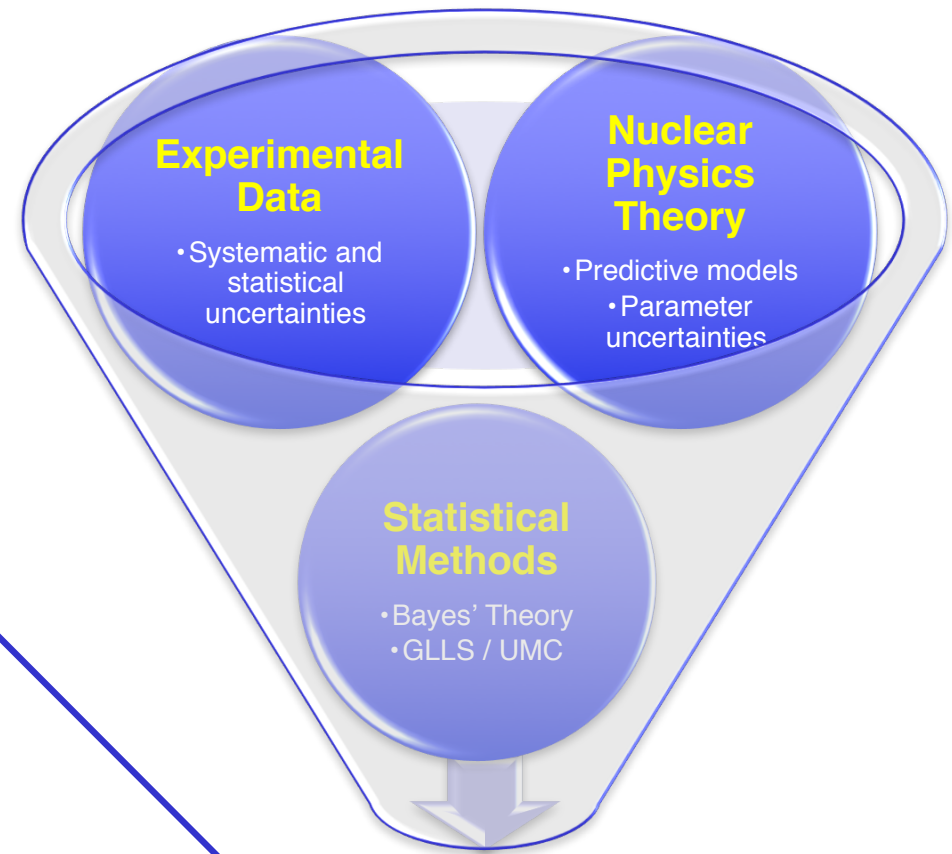
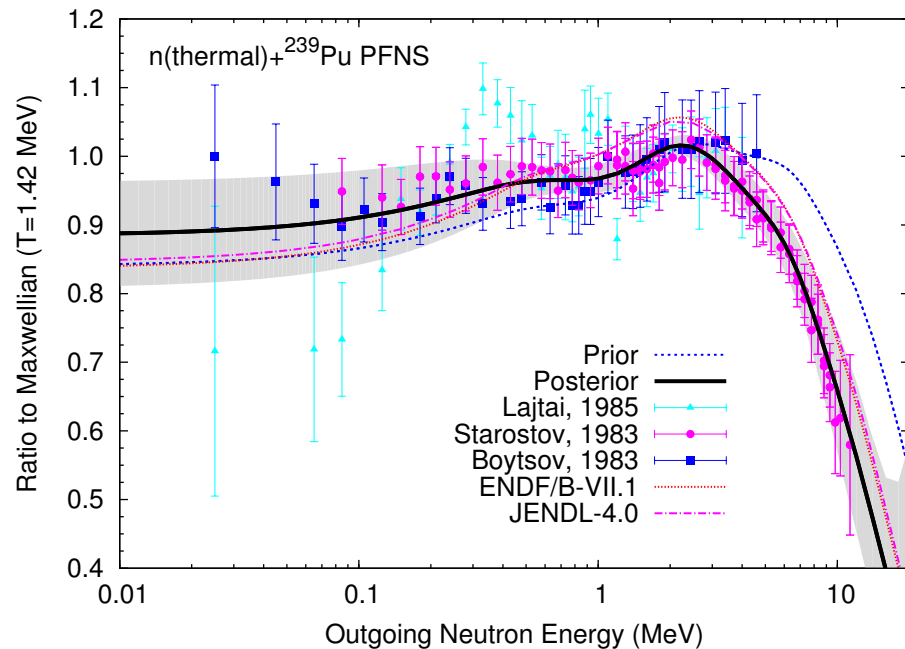
# $^{235}\text{U}$ Fission Cross-section



Experimental Data used by CSWEG

# Pu<sup>239</sup> Prompt Fission Neutron Spectrum

## How is the nuclear data determined?



**Evaluated Data  
(with  
uncertainties)**

## ENDF/B & Other Libraries

---

- **ENDF/B**

- In the early 1960s, the Cross Section Evaluation Working Group (CSEWG) was founded to generate reliable nuclear data
- CSEWG continues to produce and maintain the Evaluated Nuclear Data File (ENDF)
- ENDF/B-VI.0 was released in 1990, ENDF/B-VI.8 in 2000
- ENDF/B-VII.0 was released in December 2006  
**ENDF/B-VII.1 was released in December 2011**  
**(Included upgraded covariance matrix evaluations)**

- **Other Libraries**

- JEF                - Joint European File
- JENDL           - Japanese Evaluated Nuclear Data Library
- CENDL           - Chinese Evaluated Nuclear Data Library
- BROND           - Russian
- ENDL            - Livermore National Laboratory
- EFF              - European File - Fusion
- FENDL           - Fusion Evaluated Nuclear Data Library
- UK Nuclear Data Library

## Cross-section Covariance Data (1)

---

- For a given isotope, these 12 cross-sections & sensitivities are used within Whisper:

MT	reaction
2	elastic scatter
4	inelastic
16	n,2n
18	fission
102	n, $\gamma$
103	n,p
104	n,d
105	n,t
106	n, $^3\text{He}$
107	n, $\alpha$
452	$\nu$
1018	$\chi$

## Cross-section Covariance Data (2)

- MCNP uses continuous-energy cross-section data & collision physics, but **sensitivity profiles** are tallied in 44 energy bins
- The 44 energy bins reflect the cross-section **covariance data** files obtained for each isotope & reaction from the BLO data (low-fidelity covariance data from the Brookhaven-LANL-ORNL covariance project)

### Energy bin bounds (MeV)

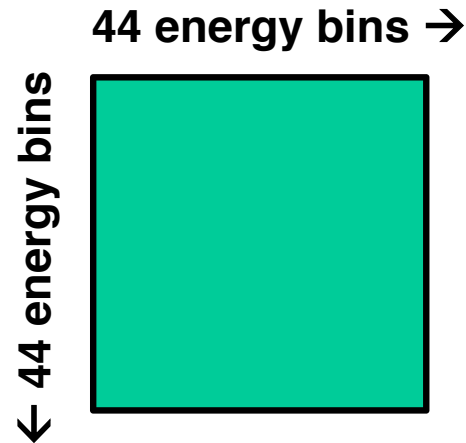
1.0000e-11	3.0000e-09	7.5000e-09	1.0000e-08	2.5300e-08	3.0000e-08
4.0000e-08	5.0000e-08	7.0000e-08	1.0000e-07	1.5000e-07	2.0000e-07
2.2500e-07	2.5000e-07	2.7500e-07	3.2500e-07	3.5000e-07	3.7500e-07
4.0000e-07	6.2500e-07	1.0000e-06	1.7700e-06	3.0000e-06	4.7500e-06
6.0000e-06	8.1000e-06	1.0000e-05	3.0000e-05	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			

- When better cross-section covariance data become available, more energy bins will be used



## Cross-section Covariance Data (3)

- For a particular isotope & particular reaction (MT), the nuclear data uncertainties are a  $G \times G$  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

## Cross-section Covariance Data (4)

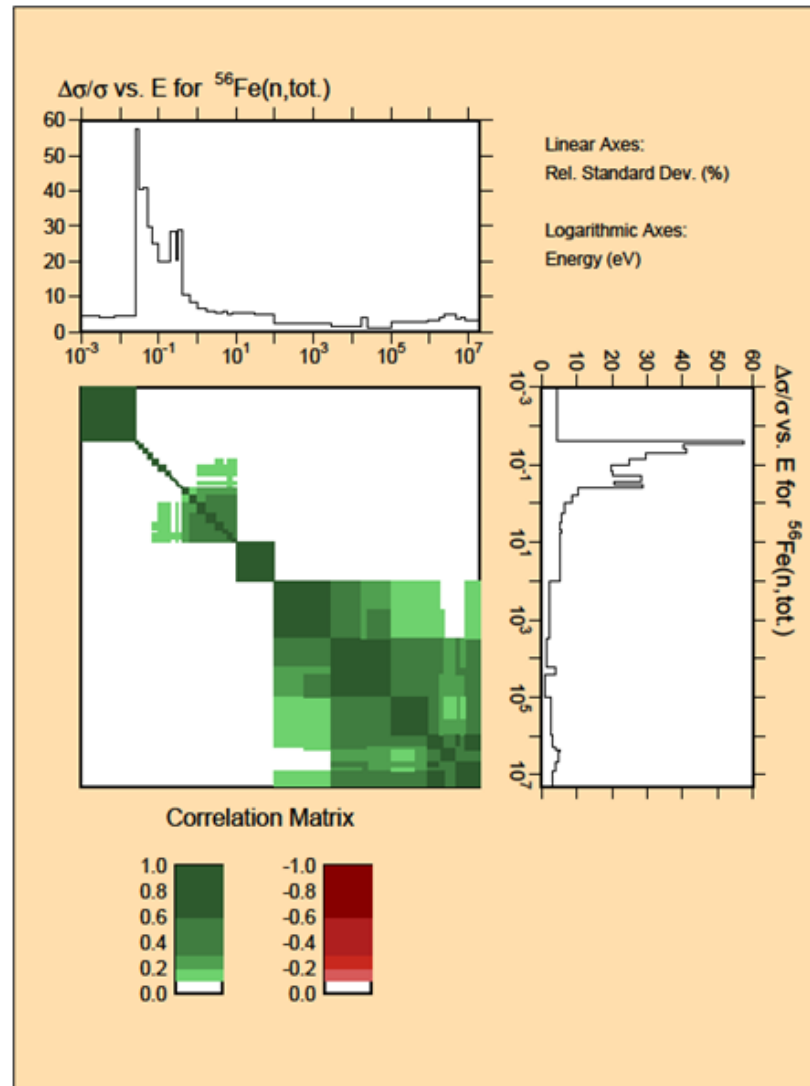


FIG. 9: A typical NJOY-generated plot of ENDF/B-VII.0 data downloaded from the National Nuclear Data Center, BNL, USA.

## Cross-section Covariance Data (5)

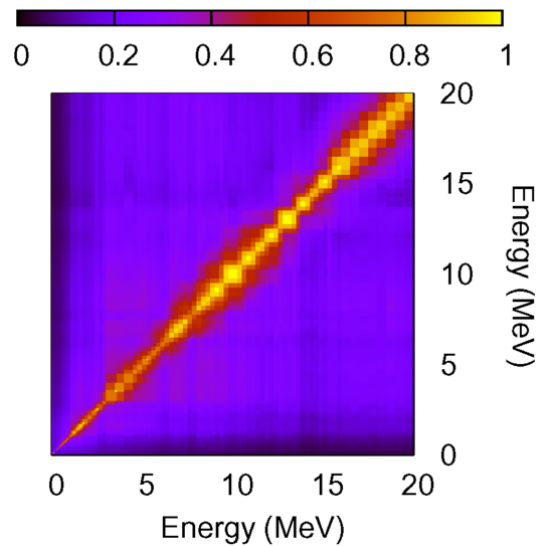


FIG. 3: Correlation matrix for the neutron-induced fission cross section on  $^{235}\text{U}$ . It was evaluated by Pronyaev *et al.* as part of the cross section standards evaluation [19].

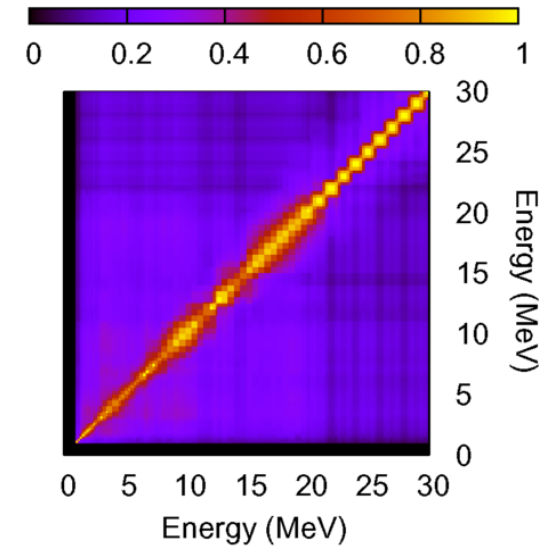


FIG. 13:  $^{238}\text{U}$  fission cross-section correlation matrix.

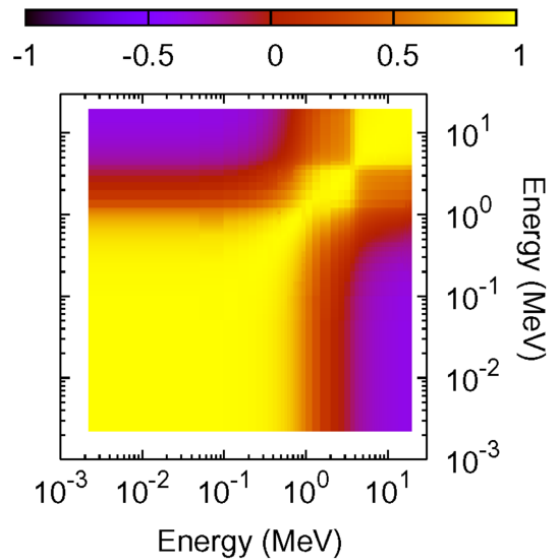


FIG. 6: Correlation matrix for the capture cross section of  $n+^{235}\text{U}$ .

Covariance plots on this & next page taken from:

P. Talou, P.G. Young, T. Kawano, M. Rising, M.B. Chadwick,  
 “Quantification of Uncertainties for Evaluated Neutron-Induced  
 Reactions on Actinides in the Fast Energy Range”,  
 Nuclear Data Sheets 112, 3054–3074 (2011)

# Cross-section Covariance Data (6)

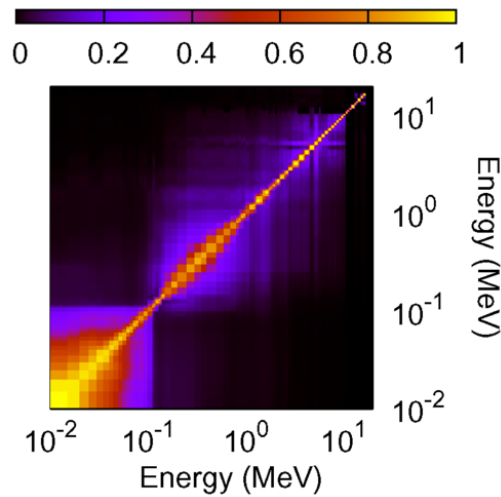


FIG. 25: Correlation matrix evaluated for the  $^{238}\text{Pu}$  (n,fission) cross section.

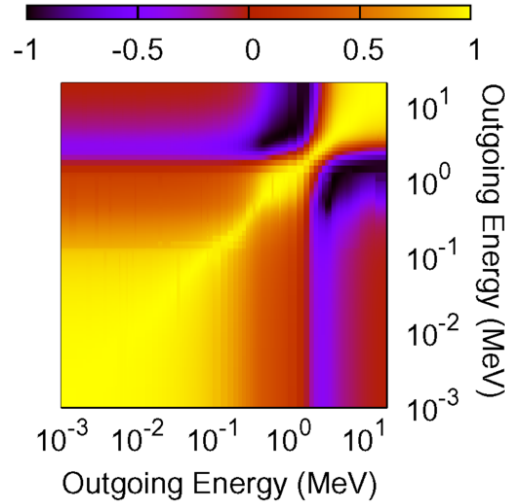


FIG. 40: Correlation matrix evaluated for the  $n(0.5 \text{ MeV}) + ^{239}\text{Pu}$  prompt fission neutron spectrum.

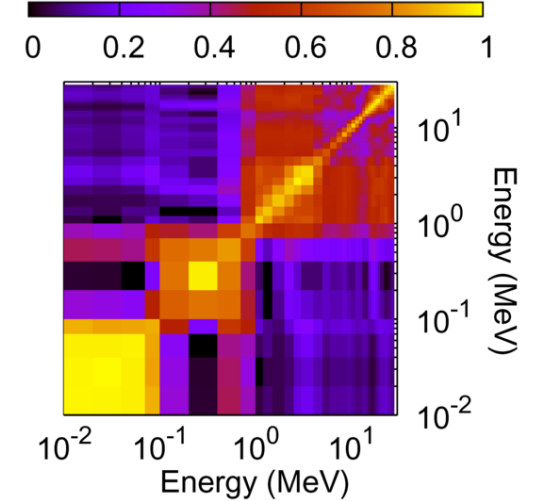


FIG. 43: Evaluated correlation matrix for the neutron-induced fission cross section of  $^{240}\text{Pu}$  in the fast energy range.

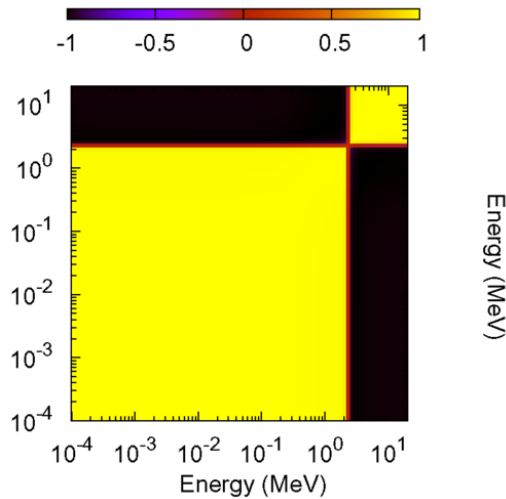


FIG. 30: Correlation matrix for the  $n(0.5 \text{ MeV}) + ^{238}\text{Pu}$  prompt fission neutron spectrum.

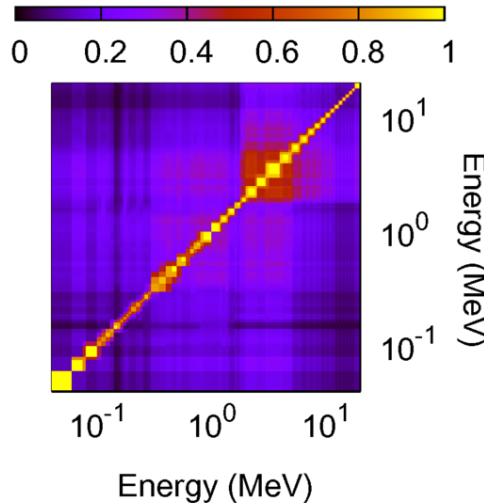


FIG. 45:  $^{240}\text{Pu}$  (n,total) cross section correlation matrix.

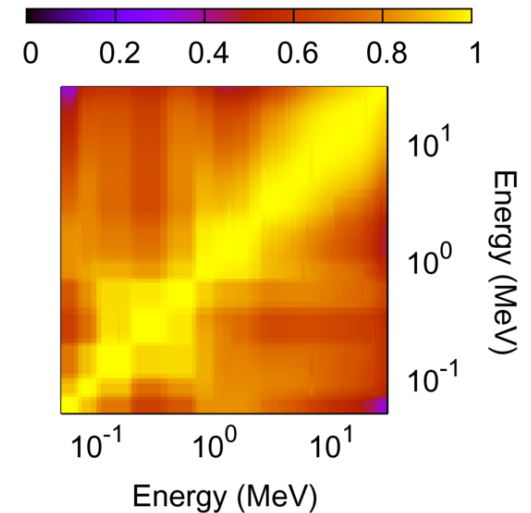


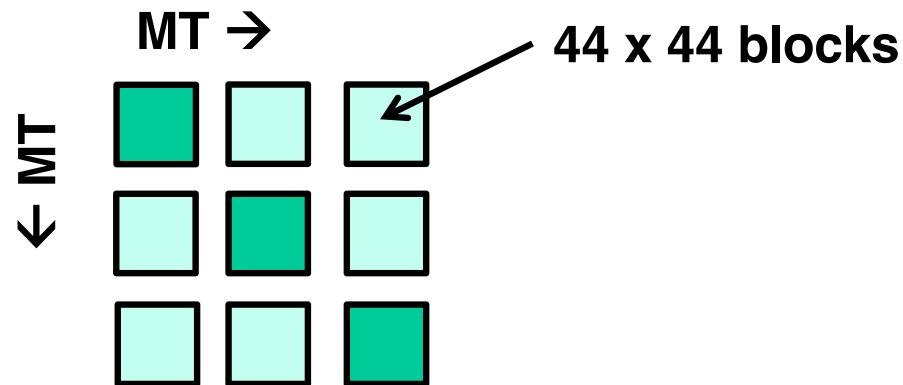
FIG. 47: Correlation matrix for the  $n + ^{240}\text{Pu}$  capture cross section. Large off-diagonal elements are due mostly to model uncertainties, since no experimental data exist above 300 keV.

## Cross-section Covariance Data (7)

- For each isotope, with 44 energies & 12 reactions:

$$C_{xx}^{\text{Iso}} : c(44, 44, 12, 12)$$

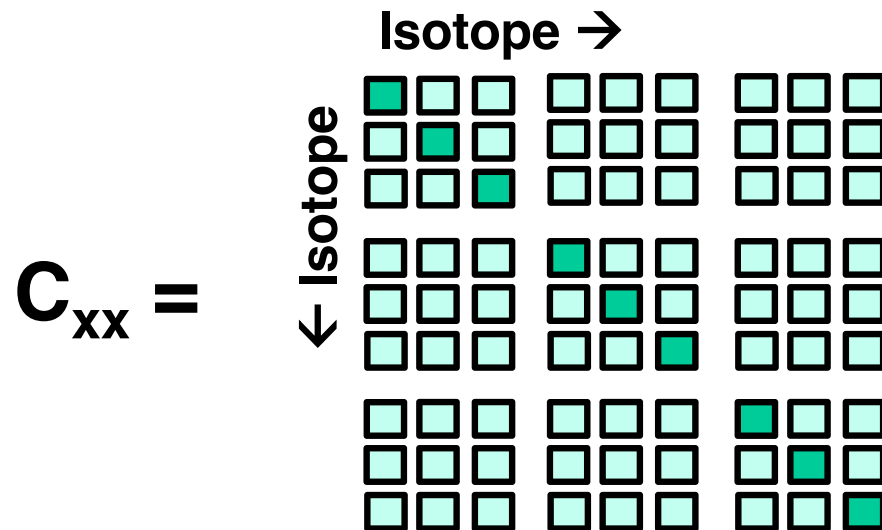
- Each diagonal element of  $C_{xx}$  is the **variance** of the cross-section for a particular MT & energy bin
- Off-diagonal elements of  $C_{xx}$  are the **shared variance** between pairs of MT-E & MT'-E' (Off-diagonal MT-MT' blocks would generally be 0)



- Each  $C_{xx}^{\text{Iso}}$  entry is produced by SCALE or NJOY based on covariance data from the ENDF/B libraries (with some adjustments if needed)
- The  $C_{xx}$  data is universal, independent of benchmark or application problem

## Cross-section Covariance Data (8)

- 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)

## 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:

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

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

MT →



44 energy bins

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

Isotopes →



---



# Correlation Coefficients

## Correlation Coefficient (1)

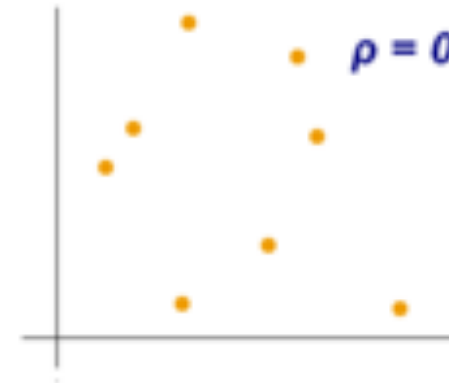
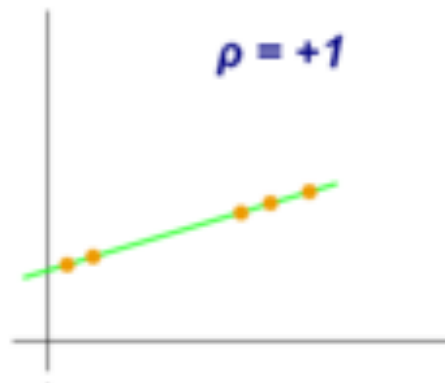
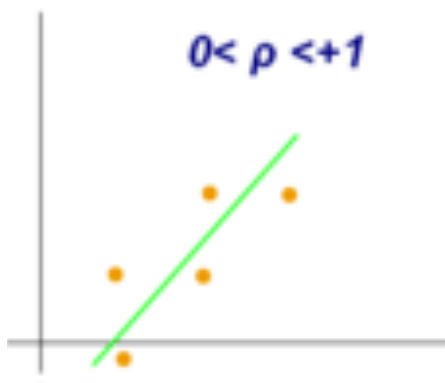
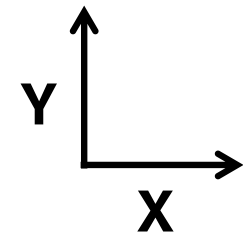
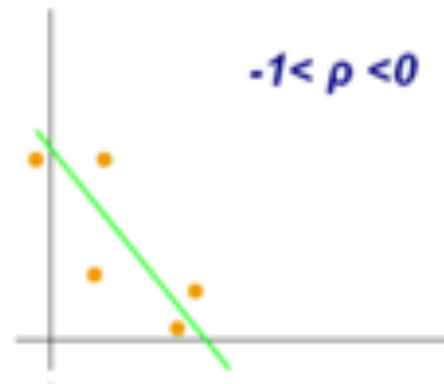
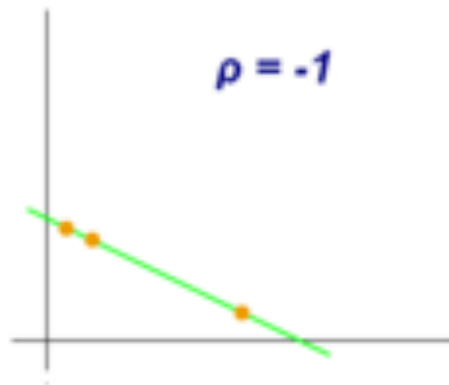
- Correlation coefficient

- 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



## Correlation Coefficient (2)

- Population correlation coefficient,  $\rho$**

- Distribution of X, with mean  $\mu_x$ , standard deviation  $\sigma_x$
- Distribution of Y, with mean  $\mu_y$ , standard deviation  $\sigma_y$

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \cdot \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \cdot \sigma_Y} = \frac{E(XY) - E(X) \cdot E(Y)}{\sigma_X \cdot \sigma_Y}$$

$$\begin{aligned} \mu_X &= E(X) & \sigma_X^2 &= E[(X - E(X))^2] = E(X^2) - E(X)^2 \\ \mu_Y &= E(Y) & \sigma_Y^2 &= E[(Y - E(Y))^2] = E(Y^2) - E(Y)^2 \end{aligned}$$

- Sample correlation coefficient,  $r$**

- Dataset for X:  $\{ x_1, x_2, \dots, x_n \}$ , mean  $\bar{x}$ , std dev  $s_x$
- Dataset for Y:  $\{ y_1, y_2, \dots, y_n \}$ , mean  $\bar{y}$ , std dev  $s_y$

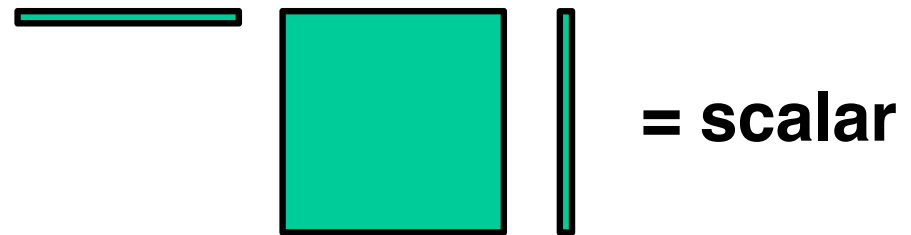
$$r = r_{xy} = \frac{\frac{1}{n} \sum x_i y_i - \bar{x} \cdot \bar{y}}{s_x \cdot s_y}$$

## Variance in Keff & Correlation Between Problems

- Given: Problem A, Sensitivity  $S_A$  computed by MCNP  
Problem B, Sensitivity  $S_B$  computed by MCNP
- Variance in Keff due to nuclear data uncertainties:

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

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



= scalar

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

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

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

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

## Sandwich Rule – Variance & Covariance

- Matrix-vector operations

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

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

**Problem-dependent sensitivity vector,  $S$ .**

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

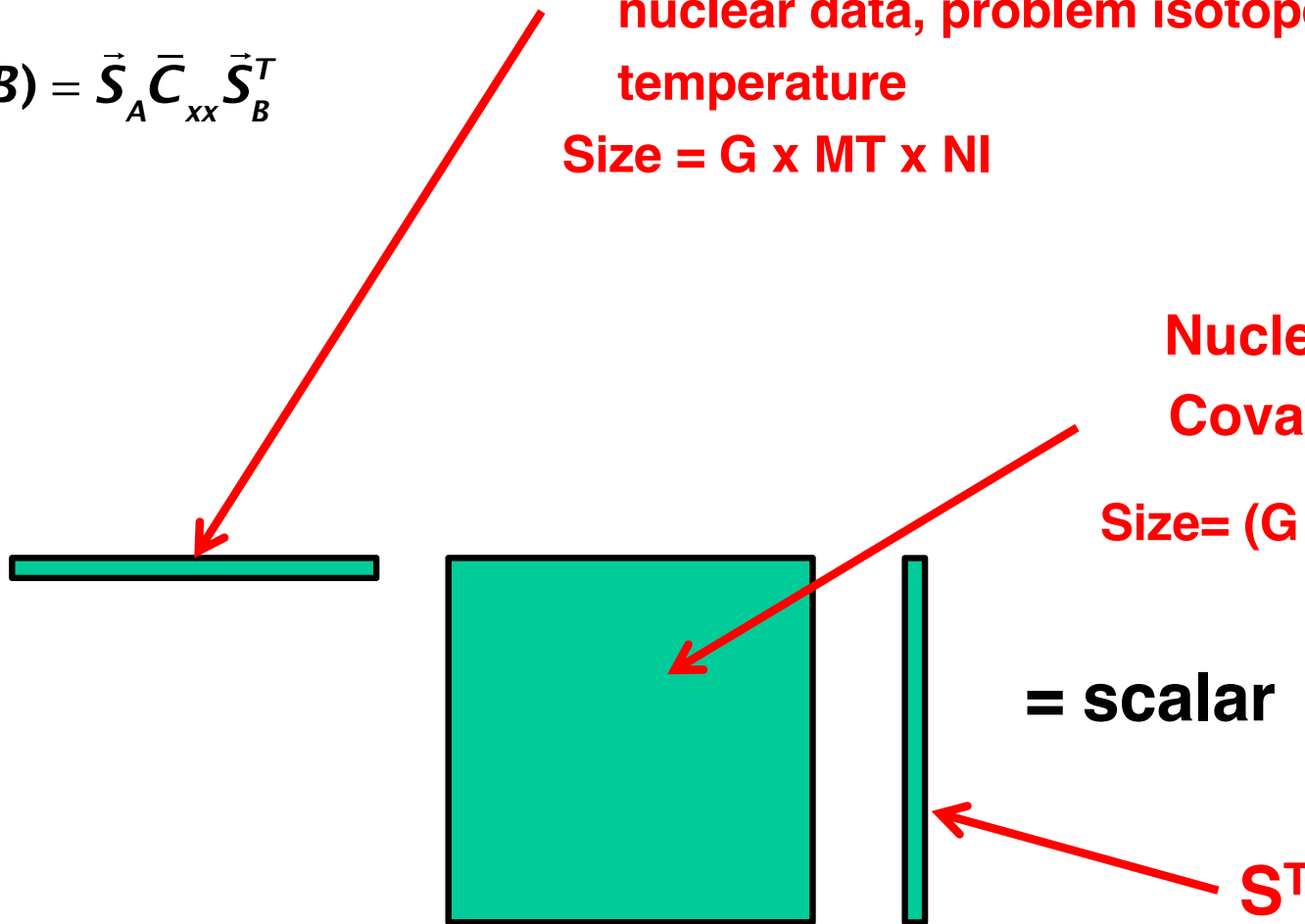
Size = G x MT x NI

**Nuclear Data Covariances**

Size = (G x MT x NI)<sup>2</sup>

= scalar

$S^T$



## Error Propagation (1)

---

- Define a linear relationship

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$$

- Determine expected (mean) value of  $\mathbf{y}$

$$\mu_y = E[\mathbf{y}] = E[\mathbf{A}\mathbf{x} + \mathbf{b}] = \mathbf{A}E[\mathbf{x}] + \mathbf{b} = \mathbf{A}\mu_x + \mathbf{b}$$

- Determine covariance matrix of  $\mathbf{y}$

$$\begin{aligned} \mathbf{C}_y &= \text{cov}(\mathbf{y}, \mathbf{y}) = E[(\mathbf{y} - \mu_y)(\mathbf{y} - \mu_y)^T] \\ &= E[(\mathbf{A}\mathbf{x} + \mathbf{b} - \mathbf{A}\mu_x - \mathbf{b})(\mathbf{A}\mathbf{x} + \mathbf{b} - \mathbf{A}\mu_x - \mathbf{b})^T] \\ &= E[(\mathbf{A}(\mathbf{x} - \mu_x))(\mathbf{A}(\mathbf{x} - \mu_x))^T] \\ &= E[\mathbf{A}(\mathbf{x} - \mu_x)(\mathbf{x} - \mu_x)^T \mathbf{A}^T] \\ &= \mathbf{A} E[(\mathbf{x} - \mu_x)(\mathbf{x} - \mu_x)^T] \mathbf{A}^T \\ &= \mathbf{A} \text{cov}(\mathbf{x}, \mathbf{x}) \mathbf{A}^T \end{aligned}$$

$$\mathbf{C}_y = \mathbf{A} \mathbf{C}_x \mathbf{A}^T \quad \leftarrow \text{“Sandwich” Rule!}$$

## Error Propagation (2)

- First-order Taylor series expansion of  $k$  about cross section,  $\Sigma$

$$k(\Sigma'_1, \Sigma'_2, \dots, \Sigma'_N) \cong k(\Sigma^0_1, \Sigma^0_2, \dots, \Sigma^0_N) + \sum_{i=1}^N \left. \frac{\partial k}{\partial \Sigma_i} \right|_{\Sigma^0_i} (\Sigma'_i - \Sigma^0_i)$$

- Define vectors for cross sections and sensitivity profiles

$$\begin{aligned} \bar{\Sigma}' &= \begin{bmatrix} \Sigma'_1 & \Sigma'_2 & \cdots & \Sigma'_N \end{bmatrix} \\ \bar{\Sigma}^0 &= \begin{bmatrix} \Sigma^0_1 & \Sigma^0_2 & \cdots & \Sigma^0_N \end{bmatrix} \end{aligned} \quad \bar{S} = \begin{bmatrix} \left. \frac{\partial k}{\partial \Sigma_1} \right|_{\Sigma^0_1} & \left. \frac{\partial k}{\partial \Sigma_2} \right|_{\Sigma^0_2} & \cdots & \left. \frac{\partial k}{\partial \Sigma_N} \right|_{\Sigma^0_N} \end{bmatrix}$$

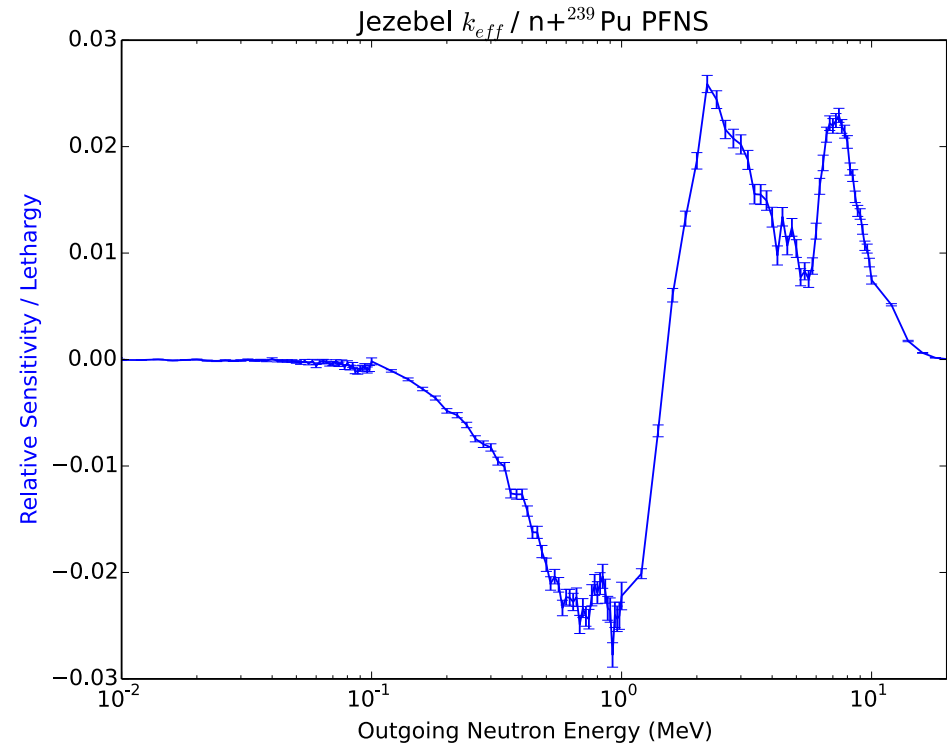
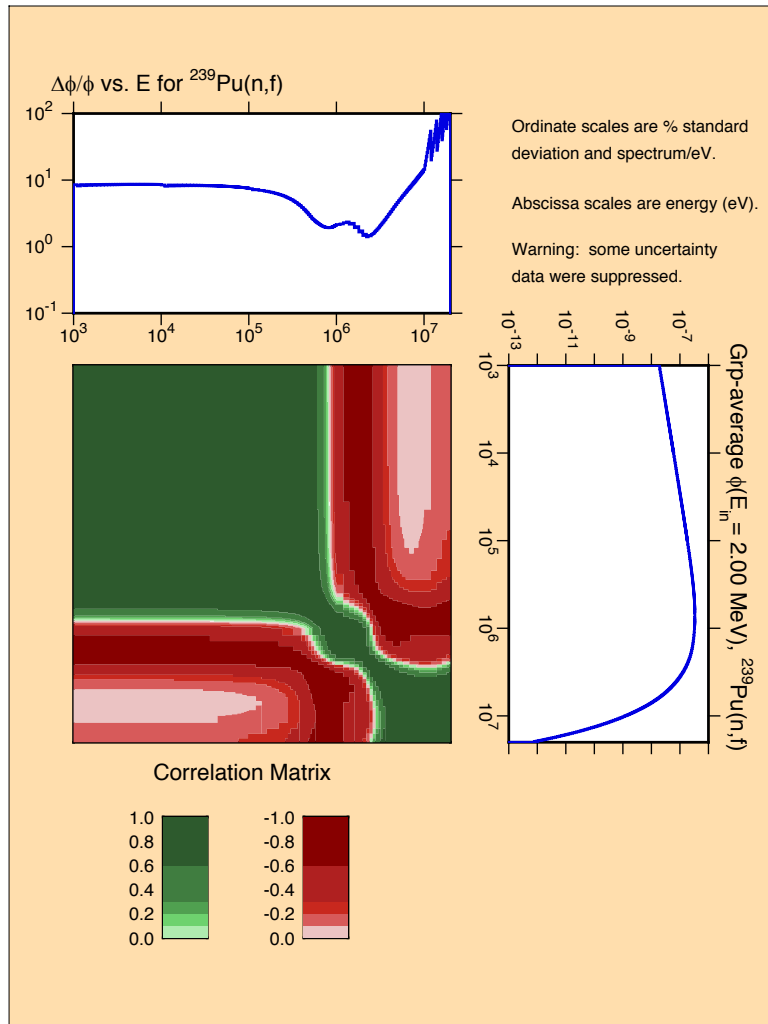
- Determine covariance matrix (variance) of  $k$

$$\begin{aligned} k(\bar{\Sigma}') &\cong k(\bar{\Sigma}^0) + \bar{S} (\bar{\Sigma}' - \bar{\Sigma}^0)^T \\ &= \bar{S} \bar{\Sigma}'^T + \left[ k(\bar{\Sigma}^0) - \bar{S} \bar{\Sigma}^{0T} \right] \\ &= \mathbf{Ax} + \mathbf{b} \end{aligned}$$

$$\mathbf{C}_k = \bar{S} \mathbf{C}_\Sigma \bar{S}^T$$

## Error Propagation (3)

- Example using sandwich rule,  $^{239}\text{Pu}$  PFNS impact on  $k$



Uncertainty in  $k$  due to  $^{239}\text{Pu}$  PFNS only!

$$\sigma_k^2 = \vec{S} \mathbf{C}_X \vec{S}^T$$

$$\frac{\sigma_k}{k} \cong 0.160\%$$





# MCNP-WHISPER

## for Nuclear Criticality Safety Validation

**This work supports:**

**US DOE Nuclear Criticality Safety Program**

**US DOE Stockpile Stewardship Program**

**LANL Nuclear Criticality Safety Division**

**LANL PF4 Restart**

**Contributors:**

**Forrest Brown, Michael Rising, Jennifer Alwin**

**Monte Carlo Codes Group, XCP-3**

# 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 US DOE-NNSA 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.

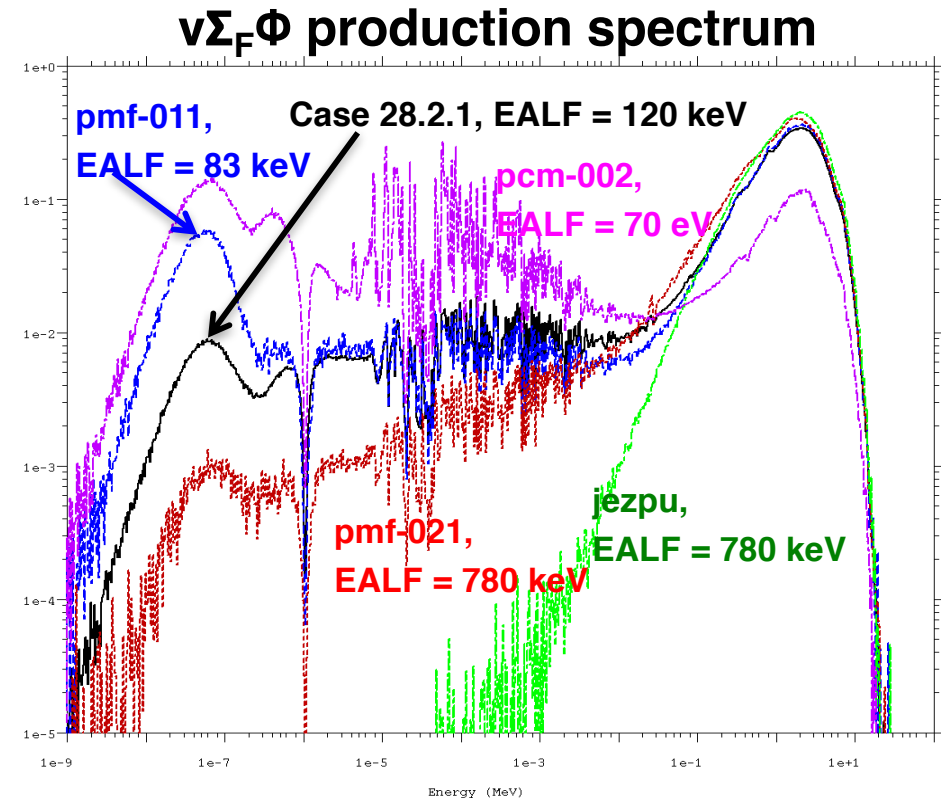
# Whisper

---

- **Whisper History, Background, SQA Status, Documentation**
- **Whisper Methodology**
  - Capabilities
  - Correlation Coefficients
  - Cross-section Covariance Data
  - Sensitivity Profiles
  - Variance in  $K_{eff}$  & Correlation Between Problems
  - Determining benchmark  $C_k$ 's
  - Determining bias & bias uncertainty
  - Determining portions of the MOS
- **Using Whisper for Validation**
  - Overview
  - Using whisper\_mcnp
  - Using whisper\_usl
  - Examples

## MCNP-WHISPER Methodology for Nuclear Criticality Safety Analysis (1)

- Nuclear Criticality Safety requires validation of computational methods
- Validation involves comparing calculation vs experiment for many benchmarks similar to the application of interest
- Neutron spectra are complex functions of geometry, materials, nuclear cross-sections, etc.
- Simple metrics 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



## MCNP-WHISPER Methodology for Nuclear Criticality Safety Validation

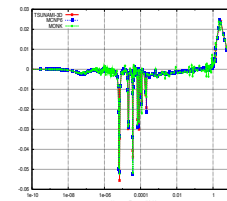
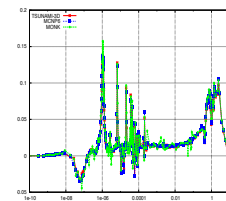
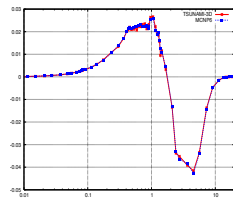
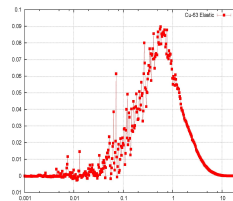
- **MCNP** determines **sensitivity profiles** to characterize the neutronics of an application or benchmark,  $S(\text{energy, reaction, isotope})$ ,  $S = (dk/k) / (d\sigma/\sigma)$
- **WHISPER** uses sensitivity profiles & data covariances to select similar benchmarks, determine bias, bias-uncertainty, & margin-of-subcriticality for setting the **Upper-Subcritical-Limit (USL)**

## MCNP-WHISPER Methodology for Nuclear Criticality Safety Analysis (2)

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

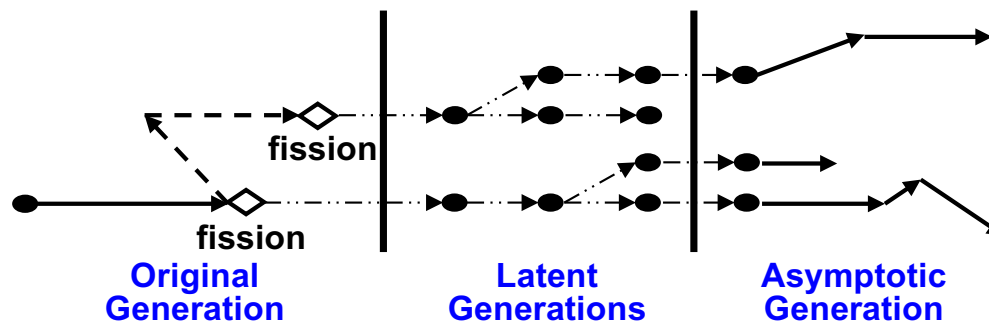
$$S_{k,x} = \frac{dk/k}{dx/x} = - \frac{\langle \psi^\dagger, (\Sigma_x - S_x - k^{-1}F_x) \psi \rangle}{\langle \psi^\dagger, k^{-1}F \psi \rangle}$$

- $S_{k,x}(E)$  is the **sensitivity profile**, that includes all isotopes, reactions, & energies for a system:

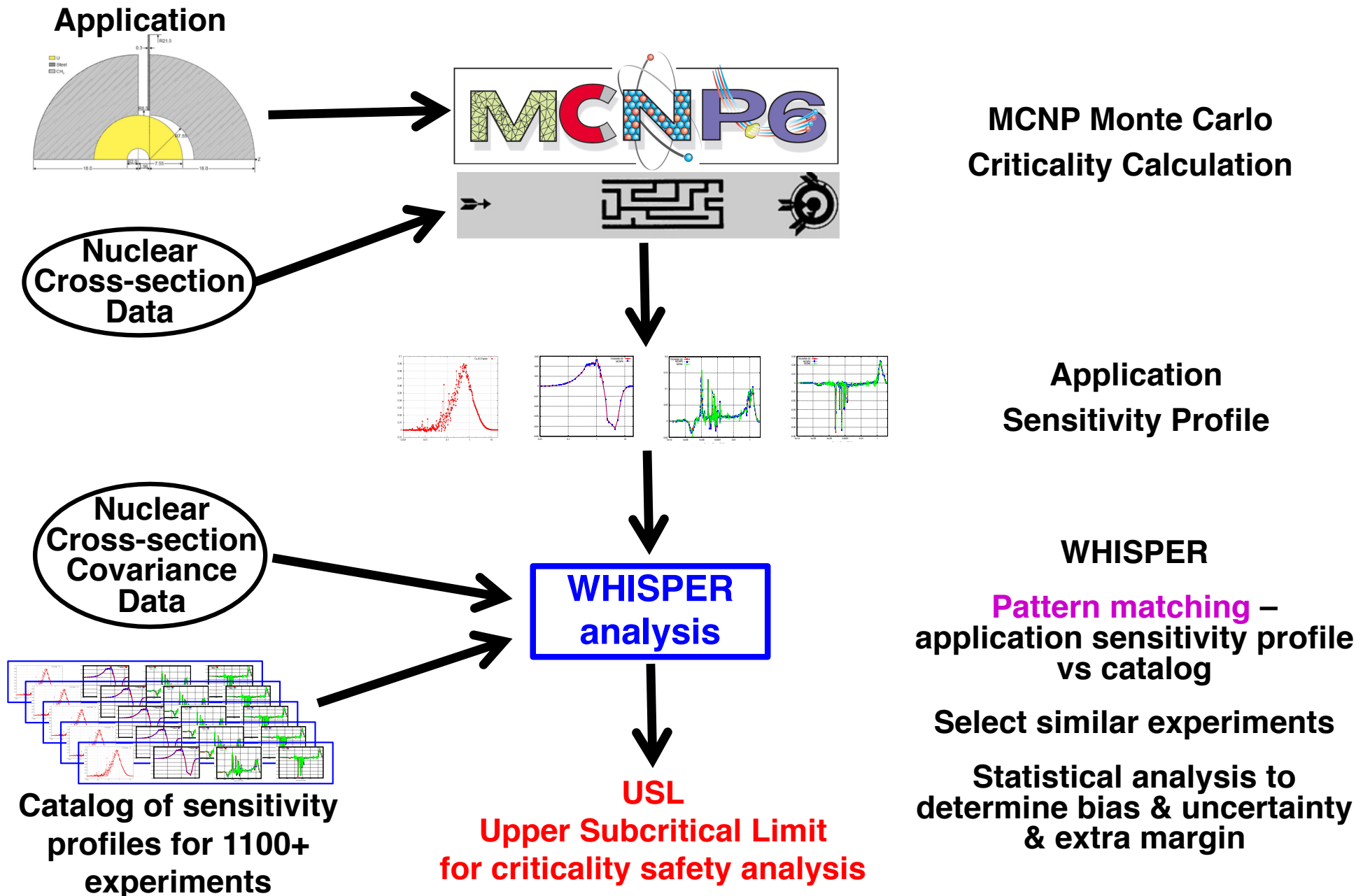


etc.

- MCNP Monte Carlo** uses the Iterated Fission Probability method to compute adjoint-weighted integrals for the sensitivity profiles
  - Tally scores are collected in original generation, adjoint-weighting is based on the progeny in the asymptotic generation



# MCNP-WHISPER Methodology for Nuclear Criticality Safety Analysis (3)



# 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 – 2017 (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](http://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  $\Delta k$  to  $\Delta k/k$ . Gives some very minor diffs in list of rejected benchmarks
  - For USL, the list of benchmarks selected is sorted by weight (or  $C_k$ )
- **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

## Whisper SQA

---

- **Whisper is part of the MCNP software package**
  - Will be distributed to the criticality-safety community via future RSICC releases of MCNP
  - Feedback from criticality-safety analysts at DOE sites will be factored into future development
  - Potential for world-wide feedback/review/improvements
- **Maintained under MCNP version control system (GIT, TeamForge)**
  - LANL standard
  - WHISPER GIT Module for checkout into MCNP source tree
  - All revisions, additions, improvements tracked under Artifact 36407
- **MCNP SQA methodology**
  - Encompasses Whisper
  - Previous audits & reviews of MCNP SQA determined that methodology was compliant with DOE/ASC & LANL P1040 requirements
  - Review is in progress to assess current MCNP SQA P1040 compliance, and make any revisions required to continue compliance

---

# Whisper Methodology

# Whisper

---

## Whisper Methodology

- **MCNP6**
  - Determine Sensitivity Profiles for Benchmarks  $B_1 \dots B_N$
  - 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 &  $B_j$
- **Whisper – Determine Benchmark Weights & Select Benchmarks**
  - Iterative procedure using  $c_k^{(j)}$  values,  $c_{k,max}$ ,  $c_{k,acc}$
- **Whisper – Determine Calculational Margin (CM)**
  - Extreme Value Theory, with weighted data, nonparametric
  - Compute bias & bias uncertainty
  - Adjustment for non-conservative bias
  - Handling small sample sizes
- **Whisper – Determine portions of MOS**

# Whisper Capabilities

## Admin

- Install code, scripts, benchmarks, covariance files, correlations
- Test the installation
- Identify inconsistent benchmarks to be rejected
- Estimate missing benchmark uncertainties
- Can add additional benchmarks
- Can reject additional benchmarks

## User

- Use **whisper\_mcnp** script to run MCNP6 for process models, to obtain  $k_{\text{eff}}$  & sensitivity profiles for all isotopes & reactions
- Use **whisper\_usl** script to run Whisper for process models
  - Whisper matches process model sensitivity profiles with benchmark library profiles, selects most similar benchmarks
  - Compute calculational margin for each process model, based on selected benchmarks (bias + bias uncertainty)
  - Estimate cross-section portion of MOS based on GLLS
  - Use 0.005 for code unknowns portion of MOS
  - Estimate baseline USL for each process model (not including additional AOA or other margin)

## 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**
  - **Details will be covered later. 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:**
  - Use the **whisper\_mcnp** script to make 1 run with MCNP6 for a particular application, to generate the sensitivity vector for the application,  $S_A$
  - Run Whisper, using the **whisper\_usl** script



## Whisper – Overview of Application Use

---

- Given  $S_A$  for an application, the nuclear data covariance files, and the collection of 1100+  $S_B$  vectors for the benchmarks
  - For each of the benchmarks, compute the correlation between the benchmark & application problem,  $c_k(A,B)$
  - Use the  $c_k(A,B)$  values for the benchmarks to compute relative weights for each benchmark
  - Select the a set of benchmarks with the highest weights (i.e., the highest neutronics correlations between benchmarks & application)
  - Using the selected benchmarks, compute bias, bias uncertainty, & extra margin based on nuclear data uncertainty
  - There are of course details, such as acceptable  $c_k$  values, determining weights using  $c_k$  values, extra penalty if not enough similar benchmarks, benchmark correlation, .....

## Whisper Details – Compute $c_k$ Values

- Given:

- Problem A, Application Sensitivity  $S_A$  computed by MCNP
- Problem  $B_J$ , Benchmark Sensitivity  $S_{B_J}$  computed by MCNP,  
 $J = 1, \dots, N$  ( $N$  = number of benchmarks)

- Find correlation between Application A & Benchmark  $B_J$ ,  $J = 1 \dots N$ :

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

- Eliminate any negative correlation coefficients

- If  $c_k^{(J)} < 0$ , set  $c_k^{(J)} = 0$ ,  $J = 1 \dots N$

- Determine maximum  $c_k^{(J)}$ ,  $C_{k,max}$

## Whisper Details – Benchmark Weights (1)

- Benchmarks are assigned weights  $w_J$  based on their  $c_k^{(J)}$  values,  $c_{k,max}$ , and a (to-be-determined) acceptance threshold,  $c_{k,acc}$ 
  - Benchmarks similar to the application,  $c_k^{(J)} > c_{k,acc}$ :  $0 < w_J \leq 1$
  - Benchmarks not similar to the application,  $c_k^{(J)} < c_{k,acc}$ :  $w_J = 0$
  - Scheme for determining  $w_J$  is on next slide
- The minimum required total weight,  $w_{req}$ , for the set of selected benchmarks is:

$$w_{req} = w_{min} + (1 - c_{k,max}) * w_{penalty}$$

$$\begin{array}{ll} \text{where } w_{min} &= 25 & \text{(default, user opt)} \\ w_{penalty} &= 100 & \text{(default, user opt)} \end{array}$$

- That is, must select enough benchmarks so that  $\text{sum}\{ w_J \} \geq w_{req}$
- Rationale
  - 25 or more are needed for reliable statistical treatment
  - If benchmarks are not close to application ( $c_{k,max}$  not close to 1.0), want to require more of them. Simple linear penalty.

## Whisper Details – Benchmark Weights (2)

- The determination of benchmark weights is iterative, based on an acceptance criteria  $c_{k,acc}$

- $c_{k,acc}$  is the minimum threshold for  $c_k^{(j)}$  values
- Benchmarks with  $c_k^{(j)} < c_{k,acc}$  are assigned  $w_j = 0$
- Benchmarks with  $c_k^{(j)} \geq c_{k,acc}$  are assigned weight

$$w_j = \frac{c_k^{(j)} - c_{k,acc}}{c_{k,max} - c_{k,acc}}$$

- Iterative procedure determines largest  $c_{k,acc}$  that satisfies requirement that  $\sum\{w_j\} \geq w_{req}$

- Select a value for  $c_{k,acc}$  close to  $c_{k,max}$
- Determine benchmark weights (by above scheme)
- If  $\sum\{w_j\} < w_{req}$ , decrease  $c_{k,acc}$  by  $10^{-5}$  & repeat above step
- The iteration ends when enough benchmarks with highest  $w_j$ 's are selected so that  $\sum\{w_j\} \geq w_{req}$

If not enough benchmarks to satisfy total weight requirement, adjustment scheme is used. Discussed later, at end.....

## Whisper Details – Calculational Margin (1)

---

- **Whisper uses a nonparametric statistical approach to determining the calculational margin (bias + bias uncertainty)**
  - Does not rely on assumption that  $(k_{\text{calc}} - k_{\text{bench}})$  is normally distributed for the set of benchmarks
  - Can handle weighted benchmarks (Tsunami rank-order scheme can't)
  - Based on **Extreme Value Theory**
    - The addition of less-relevant benchmarks cannot reduce the calculational margin
    - Irrelevant benchmarks (i.e., low  $c_k$ ) will not non-conservatively affect results
    - Accounting for weighting avoids overly conservative calculational margin
- **Whisper uses EVT to find the value of a calculational margin that bounds the worst-case bias to some probability of a weighted population**

Note in following discussion:

- There is the fundamental assumption that for a single benchmark, the bias for that benchmark is normally distributed, according to the experimental uncertainty & Monte Carlo statistics
- There is no assumption of normality across the collection of benchmarks, however. The method is nonparametric.

## Whisper Details – Computational Margin (2)

- Let  $\beta_J = k_{\text{calc } J} - k_{\text{bench } J}$  and  $\sigma_J^2 = \sigma_{\text{bench } J}^2 + \sigma_{\text{calc } J}^2$ 
  - For convenience, the  $X_J$  below are opposite in sign to  $\beta_J$
- For a set of  $N$  benchmarks, let  $X_J$  be a random variable normally distributed about  $\beta_J$  with uncertainty  $\sigma_J$ . The cumulative distribution function (CDF) for  $X_J$  is

$$F_J(x) = \text{Prob}(X_J < x) = \frac{1}{\sqrt{2\pi} \cdot \sigma_J} \int_{-\infty}^x \exp\left[-\frac{1}{2} \left(\frac{y + \beta_J}{\sigma_J}\right)^2\right] dy = \frac{1}{2} \left[ 1 + \text{erf}\left(\frac{x + \beta_J}{\sqrt{2\sigma_J^2}}\right) \right]$$

Note:  $+\beta_J$ , due to opposite sign

- Let the random variable  $X$  be the maximum (opposite-signed) bias for the benchmark collection:

$$X = \max\{ X_1, \dots, X_N \}$$

- The cumulative distribution function (CDF) for  $X$  is

$$F(x) = \text{Prob}(X \leq x) = \prod_{J=1}^N F_J(x)$$

## Whisper Details – Computational Margin (3)

- When benchmarks are weighted, the following form is used for  $F_J(x)$

$$F_J(x) = (1 - w_J) + \frac{w_J}{2} \left[ 1 + \operatorname{erf} \left( \frac{x + \beta_J}{\sqrt{2\sigma_J^2}} \right) \right]$$

- For all benchmarks  $J = 1, \dots, N$ , Whisper computes

- Benchmark weight,  $w_J$
- Bias,  $\beta_J$
- Bias uncertainty,  $\sigma_J$

- Those quantities & the weighted  $F_J(x)$  determine  $F(x)$ : 
$$F(x) = \prod_{J=1}^N F_J(x)$$

- Whisper determines the calculational margin (bias + bias uncertainty) by numerically solving:

$$F(\text{CM}) = .99 \quad (.99 \text{ is default, user opt})$$

CM is the calculational margin that bounds the worst-case benchmark bias & bias uncertainty with probability .99 (default)

## Whisper Details – Computational Margin (4)

- Bias & bias uncertainty**

$$\text{USL} = 1 - \text{CM} - \text{MOS}$$

$$= 1 + \text{bias} - \text{bias-uncert} - \Delta_{\text{non-conserv}} - \text{MOS}$$

- **ANSI/ANS-8.24:**

"Individual elements (e.g., bias and bias uncertainty) of the calculational margin need not be computed separately. Methods may be used that combine the elements into the calculational margin."

- Whisper computes CM by numerically solving  $F(\text{CM}) = .99$**

- Whisper computes bias & bias uncertainty numerically as:**

$$\text{bias} = - \int_{-\infty}^{\infty} x \cdot f(x) dx = - \int_{-\infty}^{\infty} x F(x) \sum_{j=1}^N w_j \frac{f_j(x)}{F_j(x)} dx$$

$$\sigma_{\text{bias}} = \text{CM} + \text{bias}$$

- If the bias is non-conservative (positive), then the CM is adjusted so that no credit is taken for non-conservative bias**

$$\text{if } \text{bias} > 0, \quad \text{CM} = \text{CM} + \text{bias}$$



## Whisper Details – Computational Margin (5)

What if there are not enough benchmarks to meet the requirement that  $\sum \{ w_J \} = w_{req}$  ?

- Define these quantities:

$W_{sum} = \sum \{ w_J \}$  - sum of all benchmark weights,  $w_{sum} < w_{req}$

$CM_0$  = calculational margin computed with all benchmark weights set to 1.0

- $CM_0$  is an upper bound, wide application space but not specific enough for the application being analyzed
- Typically large & very over-conservative

$CM'$  = calculation margin with weighted benchmarks, but  $w_{sum} < w_{req}$

- Note that  $CM_0 \geq CM'$

- Compute CM from: 
$$CM = CM' \cdot \frac{W_{sum}}{W_{req}} + CM_0 \cdot \left( 1 - \frac{W_{sum}}{W_{req}} \right)$$
- Should probably question the benchmark suite, & include extra conservative margin of subcriticality

## Margin of Subcriticality

---

$$\text{MOS} = \text{MOS}_{\text{software}} + \text{MOS}_{\text{data}} + \text{MOS}_{\text{application}}$$

- **MOS = additional margin "that is sufficiently large to ensure that the calculated conditions will actually be subcritical" (ANSI/ANS-8.24)**
- **MOS<sub>software</sub> (for MCNP)**
  - **No approximations from mesh or multigroup**
  - **Exact answers to analytical benchmarks with given xsecs**
  - **Many years testing with collision physics & random sampling**
  - **Only realistic concern is unknown bugs**
    - MCNP is used a lot, for many different criticality applications
    - Bugs that produce  $\Delta k < 0.0010$  are difficult to distinguish from data uncertainties
    - Past bugs that produced  $\Delta k > 0.0020$  are very few, but reported & fixed
    - Historical detection limit for bugs is  $\Delta k \sim 0.0020$
    - Expert judgment, conservative:  $\text{MOS}_{\text{software}} = 0.0050$   
→ Any unknown bug larger than this would have certainly been found & fixed
    - Other MC codes should almost certainly use a larger margin
    - **Analysts may use a larger number, but have no basis for a smaller number**

## Margin of Subcriticality

---

$$\text{MOS} = \text{MOS}_{\text{software}} + \text{MOS}_{\text{data}} + \text{MOS}_{\text{application}}$$

- **MOS<sub>application</sub>**
  - **Analyst: analyses, scoping, judgment**
  - **Consider uncertainties in dimensions, densities, isotopics, etc.**
  - **Consider the number of similar benchmark cases**
  - **Consider area-of-applicability**
  - **Expert judgment, backed up by analysis**

## Margin of Subcriticality

---

$$\text{MOS} = \text{MOS}_{\text{software}} + \text{MOS}_{\text{data}} + \text{MOS}_{\text{application}}$$

- **MOS<sub>data</sub>**

- The largest portion of MOS comes from uncertainties in the nuclear cross-section data
- Data uncertainties could be as large as 0.5% - 1% in extra MOS, possibly more, possibly less
- **MOS<sub>data</sub> depends on the application**
  - For common applications, where there are lots of benchmark experiments, the relevant ENDF/B-VII data was adjusted based on those benchmarks
  - For less common applications, where there are few benchmark experiments, ENDF/B-VII adjustments for benchmarks plays little or no role in the data
- In the past, very difficult to assess MOS<sub>data</sub>, which led to large conservative margins
- Whisper (LANL) & Tsunami (ORNL) both use essentially the same methodology to address MOS<sub>data</sub> – GLLS
- Generalized Linear Least Squares (GLLS) takes into account the experiments, calculations, sensitivities, & data covariance data to predict MOS<sub>data</sub>

## Margin of Subcriticality - GLLS

- **The goal of GLLS:** (start at the end.....)
  - Determine adjustments to the nuclear data,  $\Delta x$ , which produce changes in computed  $k_{\text{eff}}$  for benchmarks,  $\Delta k$ , such that this quantity is minimized for the set of benchmarks:

$$\chi^2 = \Delta \vec{k} \cdot \bar{C}_{kk} \cdot \Delta \vec{k}^T + \Delta \vec{x} \cdot \bar{C}_{xx} \cdot \Delta \vec{x}^T$$

- $\Delta k$  is a vector of the relative changes in the ratio of calculated  $k$  to benchmark  $k$ , due to the change in cross-section data  $\Delta x$ . The length of  $\Delta k$  is the number of benchmarks
- $\Delta x$  is a vector of the relative differences of cross-section data from their mean values. The length of  $\Delta x$  is (isotopes)\*(reactions)\*(energies)
- $C_{kk}$  is the relative covariance matrix for the benchmark experiment  $k$ 's
  - Diagonal elements are variance of each benchmark experiment
  - Off-diagonals are correlation between benchmark measurements. (From DICE, often zero or not well-known)
- $C_{xx}$  is the relative covariance matrix for the nuclear data
- GLLS finds  $\Delta x$  (and the resulting  $\Delta k$ ) such that  $\chi^2$  is minimized

## Margin of Subcriticality - GLLS

---

- **The goal of GLLS:**

- Determine adjustments to the nuclear data,  $\Delta x$ , which produce changes in computed  $k_{\text{eff}}$  for benchmarks,  $\Delta k$ , such that this quantity is minimized for the set of benchmarks:

$$\chi^2 = \Delta \vec{k} \cdot \bar{C}_{kk} \cdot \Delta \vec{k}^T + \Delta \vec{x} \cdot \bar{C}_{xx} \cdot \Delta \vec{x}^T$$

- With no data adjustment,  $\Delta x = 0$ , so  $\chi^2$  determined only by differences in calculated & benchmark k's
- If data is adjusted to decrease 1<sup>st</sup> term, then 2<sup>nd</sup> term increases
- GLLS determines optimum tradeoff (minimum  $\chi^2$ ) between  $\Delta x$  &  $\Delta k$

## GLLS

---

Measured  $k_{eff}$  values for benchmarks:

$$\vec{m} = (m_i), \quad i = 1, \dots, I \quad (I = \# \text{ benchmarks})$$

Covariance matrix for  $\vec{m}$ , relative to calculated  $k_{eff}$ 's:

$$\bar{C}_{mm} = \left( \frac{m_i}{k_i} \cdot \frac{\text{cov}(m_i, m_j)}{m_i m_j} \cdot \frac{m_j}{k_j} \right), \quad i, j = 1, \dots, I$$

Covariance between measured benchmark k's (m's) & cross-section data:

$$\bar{C}_{xm} = \left( \frac{\text{cov}(x_n, m_i)}{x_n m_i} \cdot \frac{m_i}{k_i} \right), \quad n = 1, \dots, M \quad i = 1, \dots, I$$

This represents correlations between cross-section data & the measured benchmark k's. **At present, these data do not exist. Neither Tsunami nor Whisper use  $C_{xm}$ .**

# GLLS

Linear changes in calculated  $k_{eff}$  due to perturbation in data,  $\vec{x}$ :

$$k_i(\vec{x}') = k_i(\vec{x} + \delta\vec{x}) = k_i(\vec{x}) + \delta k_i = k_i(\vec{x}) \cdot \left[ 1 + \sum_{n=1}^M S_n^{(i)} \cdot \frac{\delta x_n}{x_n} \right]$$

Recall that:

Sensitivity matrix for a set of benchmarks:

$$\bar{S}_k = \left( \frac{x_n}{k_i} \cdot \frac{\partial k_i}{\partial x_n} \right) \quad i = 1, \dots, I \text{ (rows)} \quad n = 1, \dots, M \text{ (cols)}$$

Covariance matrix for nuclear data,  $\vec{x}$  :

$$\bar{C}_{xx} = \left( \frac{\text{cov}(x_n, x_p)}{x_n x_p} \right) \quad n = 1, \dots, M \quad p = 1, \dots, M$$

Uncertainty matrix for the set of benchmarks, due to data:

$$\bar{C}_{kk} = \bar{S}_k \cdot \bar{C}_{xx} \cdot \bar{S}_k^T$$

Express the relative changes in k for a set of benchmarks due to data perturbations:

$$\frac{k_i(\vec{x}') - m_i}{k_i(\vec{x})} = \frac{k_i(\vec{x}) - m_i}{k_i(\vec{x})} + \left[ \sum_{n=1}^M S_n^{(i)} \cdot \frac{\delta x_n}{x_n} \right]$$

or

$$\vec{y} = \vec{d} + \bar{S}_k \cdot \vec{z}$$



## GLLS

For the vector  $\vec{d}$ ,  $(d_i) = \frac{k_i(\vec{x}) - m_i}{k_i(\vec{x})} \quad i = 1, \dots, I$

the uncertainty matrix for the set of benchmarks is

$$\begin{aligned}\bar{C}_{dd} &= \bar{C}_{kk} + \bar{C}_{mm} - \bar{S}_k \bar{C}_{xm} - \bar{C}_{mx} \bar{S}_k^T \\ &= \bar{S}_k \bar{C}_{xx} \bar{S}_k^T + \bar{C}_{mm} - \bar{S}_k \bar{C}_{xm} - \bar{C}_{mx} \bar{S}_k^T\end{aligned}$$

GLLS involves minimizing this quantity:

$$Q(\vec{z}, \vec{y}) = (\vec{y}, \vec{z}) \cdot \begin{pmatrix} \bar{C}_{mm} & \bar{C}_{mx} \\ \bar{C}_{xm} & \bar{C}_{xx} \end{pmatrix}^{-1} \cdot (\vec{y}, \vec{z})^T,$$

subject to the constraint  $\vec{y} = \vec{d} + \bar{S}_k \vec{z}$

This is accomplished using Lagrange multipliers & minimizing this quantity:

$$R(\vec{z}, \vec{y}) = Q(\vec{z}, \vec{y}) + 2\lambda(\bar{S}_k \vec{z} - \vec{y})$$

$\vec{z}$  and  $\vec{y}$  satisfy these relations:

$$\frac{\partial R(\vec{z}, \vec{y})}{\partial \vec{z}} = \frac{\partial R(\vec{z}, \vec{y})}{\partial \vec{y}} = 0$$

The results, giving the adjusted data & k's that minimize R are:

$$\Delta \text{data:} \quad \vec{z} = (\bar{C}_{xm} - \bar{C}_{xx} \bar{S}_k^T) \cdot \bar{C}_{dd}^{-1} \cdot \vec{d}$$

$$\Delta k: \quad \vec{y} = (\bar{C}_{mm} - \bar{C}_{mx} \bar{S}_k^T) \cdot \bar{C}_{dd}^{-1} \cdot \vec{d}$$

# GLLS

GLLS gives the data adjustments (& resulting  $\Delta k$ 's) that minimize the Q or R functions (also called  $\chi^2$ )

**The adjustments also give reduced uncertainties:**

$$\bar{C}_{m'm'} = \bar{C}_{mm} - (\bar{C}_{mm} - \bar{C}_{mx} \bar{S}_k^T) \cdot \bar{C}_{dd}^{-1} \cdot (\bar{C}_{mm} - \bar{S}_k \bar{C}_{xm})$$

$$\bar{C}_{x'x'} = \bar{C}_{xx} - (\bar{C}_{xm} - \bar{C}_{xx} \bar{S}_k^T) \cdot \bar{C}_{dd}^{-1} \cdot (\bar{C}_{mx} - \bar{S}_k \bar{C}_{xx})$$

The adjusted uncertainty matrix in k for a set of applications is:

$$\bar{C}_{k'k'} = \bar{S}_{k,A} \cdot \bar{C}_{x'x'} \cdot \bar{S}_{k,A}^T$$

where each row of  $\bar{S}_{k,A}$  is the sensitivity vector for an application.

The square roots of diagonal elements in  $\bar{C}_{k'k'}$  are the relative  $1\sigma$  uncertainties in k for the adjusted data.

For a particular application i, the portion of MOS for nuclear data uncertainty is:

$$MOS_{data} = n_{\sigma} \cdot \sqrt{(\bar{C}_{k'k'})_{i,i}}$$

where  $n_{\sigma} = 2$  for 95% confidence, 2.6 for 99%

## Upper Subcritical Limit

---

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

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

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

$$\text{MOS} = \text{MOS}_{\text{data}} + \text{MOS}_{\text{code}} + \text{MOS}_{\text{application}}$$

- 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.

---

# Whisper Usage

## Using Whisper for Validation

---

- **As part of Whisper installation (not day-to-day use),**
  - **For each of the ~1100 benchmarks**
    - MCNP6 is run to generate the sensitivity vector  $S_B$  for that benchmark
    - The sensitivity vector  $S_B$  for each benchmark is saved in a folder
  - **The nuclear data covariance files are saved in a folder**
  - **Benchmarks are checked for consistency, some may be rejected**
  - **Missing uncertainties for some benchmarks are estimated**
  - **All of this is the responsibility of the Admin person & needs to be done only once at installation (or repeated if the code, data, or computer change)**
- **To use Whisper for validation:**
  - ① Use the **whisper\_mcnp** script to make 1 run with MCNP6 for a particular application, to generate the sensitivity vector for the application,  $S_A$
  - ② Run Whisper, using the **whisper\_usl** script

## Whisper-1.1.0 – Batch Job

---

To try it, on Moonlight HPC front end:

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

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

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

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

cd WTEST

whisper_mcnp.pl  -local    myjob.i
whisper_us1.pl
```

- **Submit batch job file**

```
msub    job.txt
```

## Whisper-1.1.0 – Interactive

---

To try it, on Moonlight HPC:

- Set & export WHISPER\_PATH environment variable

- **bash:**

```
export WHISPER_PATH="/usr/projects/mcnp/ncs/WHISPER"
export PATH="$WHISPER_PATH/bin:$PATH"
```

- **csh, tcsh:**

```
setenv WHISPER_PATH "/usr/projects/mcnp/ncs/WHISPER"
setenv PATH "$WHISPER_PATH/bin:$PATH"
```

- Make a directory, copy MCNP6 input files to it

- No blanks in pathname, directory name, input file names
  - Put mcnp6 input files in the directory

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



## Using whisper\_mcnp (1)

---

- From the front-end on an HPC system:

**whisper\_mcnp.pl      myjob.i**

- **myjob.i** is an MCNP6 input file

- Must NOT include any of these cards: **kopts, ksen, prdmp**
- May list more than 1 input file on whisper\_mcnp command line
- Lots of options, see next 2 slides

- **Creates files & dirs:**

MCNPInputList.toc

Calcs/

Calcs/myjob.i

KeffSenLib/

← modified to include kopts, ksen, prdmp, & new kcode

- **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.i○	output file from MCNP6 jobs
myjob.i <sup>r</sup>	runtp file
myjob.i <sup>s</sup>	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

## Using whisper\_mcnp (4)

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

```
bash: cd WTEST
```

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

### Screen output:

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

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

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

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

```
All jobs will be submitted using moab
```

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

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

## Using whisper\_mcnp (5)

---

- After running whisper\_mcnp in directory WTEST:

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

Use moab commands to check job status: `showq -u username`

When the submitted job is complete:

Files created by whisper\_mcnp & mcnp6:

WTEST/

myjob.i ← original

MCNPInputlist.toc

Calcs/

myjob.i myjob.io myjob.ir myjob.is

KeffSenLib/

## Using whisper\_usl (1)

---

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

```
cd    WTEST
whisper_usl.pl
```

- Can optionally include `ExcludeFile.dat`, list of benchmark files to exclude from Whisper calculations
  - Runs Whisper for application(s) `myjob.i` (etc)
- 
- For each input file listed in `MCNPInputList.toc`:
    - Extract sensitivity profiles from `Calcs/myjob.io`, place into directory `KeffSenLib/`
    - Create (or add to) file `KeffSenList.toc`
    - Run Whisper using the sensitivity profiles for the application (`myjob.i`) and the collection of Whisper benchmark sensitivity profiles
    - Output to screen & file `whisper.out`

## Using whisper\_usl (2)

---

- After running whisper\_mcnp & whisper\_usl:

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

Files created by whisper\_mcnp, mcnp6, & whisper\_usl:

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

## whisper\_usl.pl (3)

```
bash: whisper_usl.pl
```

```
*****
```

```
*                               *
```

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

```
*                               *
```

```
*****
```

```
=====> setup files for whisper
```

```
---> setup for problem myjob.i
```

```
...extract sensitivity profile data from: Calcs/myjob.io
```

```
...copy      sensitivity profile data to:   KeffSenLib/myjob.ik
```

```
...extract calc Keff & Kstd      data from: Calcs/myjob.io
```

```
... KeffCalc= 0.96740 +- 0.00057,  ANECF= 1.4904E+00 MeV,  EALF= 1.2150E-01 MeV
```

```
=====> run whisper
```

```
/Users/fbrown/CODES/WHISPER/WHISPER.git/bin/whisper -a KeffSenList.toc -ap KeffSenLib
```

```
whisper-1.1.0          2016-02-02   (Copyright 2016 LANL)
```

```
WHISPER_PATH           = /Users/fbrown/CODES/WHISPER
```

```
Benchmark TOC File     = /Users/fbrown/CODES/WHISPER/Benchmarks/TOC/BenchmarkTOC.dat
```

```
Benchmark Sensitivity Path = /Users/fbrown/CODES/WHISPER/Benchmarks/Sensitivities
```

```
Benchmark Correlation File =
```

```
Benchmark Exclusion File =
```

```
Benchmark Rejection File =
```

```
Covariance Data Path   = /Users/fbrown/CODES/WHISPER/CovarianceData/BLO-44g
```

```
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

.....case        3    Ck=   0.63497

← all Ck's printed in Whisper.out,  
only a few printed to the screen

.....

.....case    246    Ck=   0.18901

application	calc margin	data unc (1-sigma)	baseline USL	k(calc) > USL
myjob.i	0.01329	0.00120	0.97860	-0.00972

# Whisper.out (1)

```

whisper-1.1.0          2016-02-02   (Copyright 2016 LANL)
WHISPER_PATH           = /Users/fbrown/CODES/WHISPER
Benchmark TOC File      = /Users/fbrown/CODES/WHISPER/Benchmarks/TOC/BenchmarkTOC.dat
Benchmark Sensitivity Path = /Users/fbrown/CODES/WHISPER/Benchmarks/Sensitivities
Benchmark Correlation File =
Benchmark Exclusion File  =
Benchmark Rejection File  =
Covariance Data Path     = /Users/fbrown/CODES/WHISPER/CovarianceData/BLO-44g
Covariance Adjusted Data Path =
Application TOC File      = KeffSenList.toc
Application Sensitivity Path = KeffSenLib/
User Options File        =
Output File              = Whisper.out

```

Reading benchmark data ...

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

.....

246 benchmarks read, 0 benchmarks excluded.

Reading application data ...

application	k(calc)	unc
myjob.i	0.96802	0.00052

Reading covariance data ...

Reading covariance data for 1001 ...

.....

Reading adjusted covariance data ...

Reading covariance data for 1001 ...

## Whisper.out (2)

Calculating application nuclear data uncertainties ...

application	adjusted	prior
myjob.i	0.00209	0.01221

Calculating upper subcritical limits ...

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

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

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

### benchmark

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

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

Benchmark rankings shown below

## Conclusions & Discussion

---

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

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

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

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

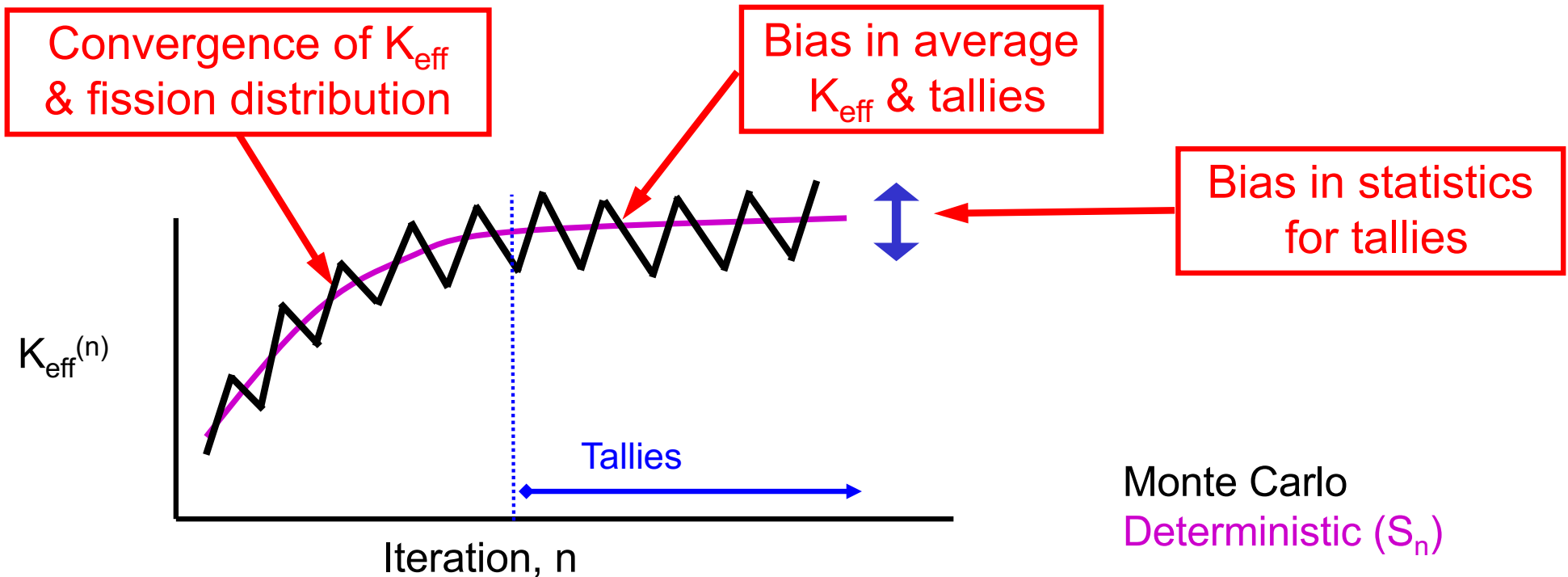
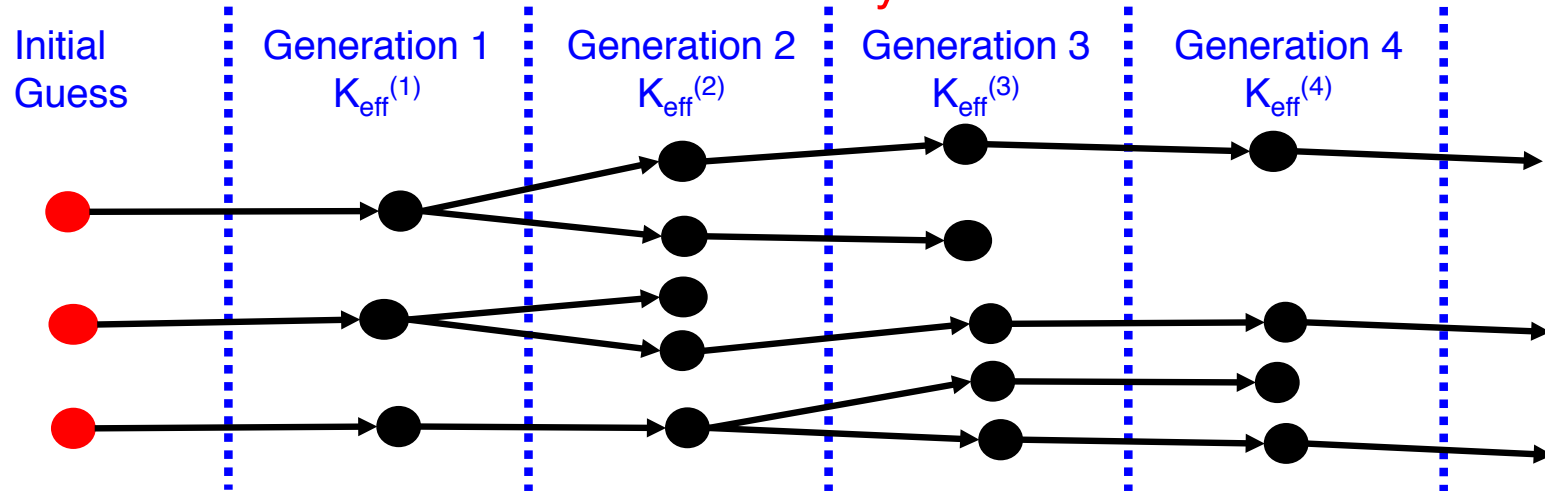
---

# Best Practices for Monte Carlo Criticality Calculations

- **Monte Carlo Criticality Calculations**
  - Methodology & Concerns
  - Convergence
  - Bias
  - Statistics
- **Best Practices**
  - Discussion
  - Conclusions

# Methodology & Concerns

## Power Iteration for MC Criticality Calculations



# Convergence

- Monte Carlo codes use power iteration to solve for  $K_{\text{eff}}$  &  $\Psi$  for eigenvalue problems

- Power iteration convergence is well-understood:

$n$  = cycle number,  $k_0, u_0$  - fundamental,  $k_1, u_1$  - 1st higher mode

$$\Psi^{(n)}(\vec{r}) = \vec{u}_0(\vec{r}) + a_1 \cdot \rho^n \cdot \vec{u}_1(\vec{r}) + \dots$$

$$k_{\text{eff}}^{(n)} = k_0 \cdot \left[ 1 - \rho^{n-1} (1 - \rho) \cdot g_1 + \dots \right]$$

- First-harmonic source errors die out as  $\rho^n$ ,  $\rho = k_1 / k_0 < 1$
  - First-harmonic  $K_{\text{eff}}$  errors die out as  $\rho^{n-1} (1 - \rho)$
  - Source converges slower than  $K_{\text{eff}}$
- Most codes only provide tools for assessing  $K_{\text{eff}}$  convergence.
- MCNP also looks at Shannon entropy of the source distribution,  $H_{\text{src}}$ .



## Bias in $K_{\text{eff}}$ & Tallies

---

- **Power iteration is used for Monte Carlo  $K_{\text{eff}}$  calculations**
  - **For one cycle (iteration):**
    - $M_0$  neutrons start
    - $M_1$  neutrons produced,  $E[ M_1 ] = K_{\text{eff}} \cdot M_0$
  - **At end of each cycle, must renormalize by factor  $M_0 / M_1$**
  - **Dividing by stochastic quantity ( $M_1$ ) introduces bias in  $K_{\text{eff}}$  & tallies**
- **Bias in  $K_{\text{eff}}$ , due to renormalization**

$$\text{Bias in } K_{\text{eff}} \propto \frac{1}{M}$$

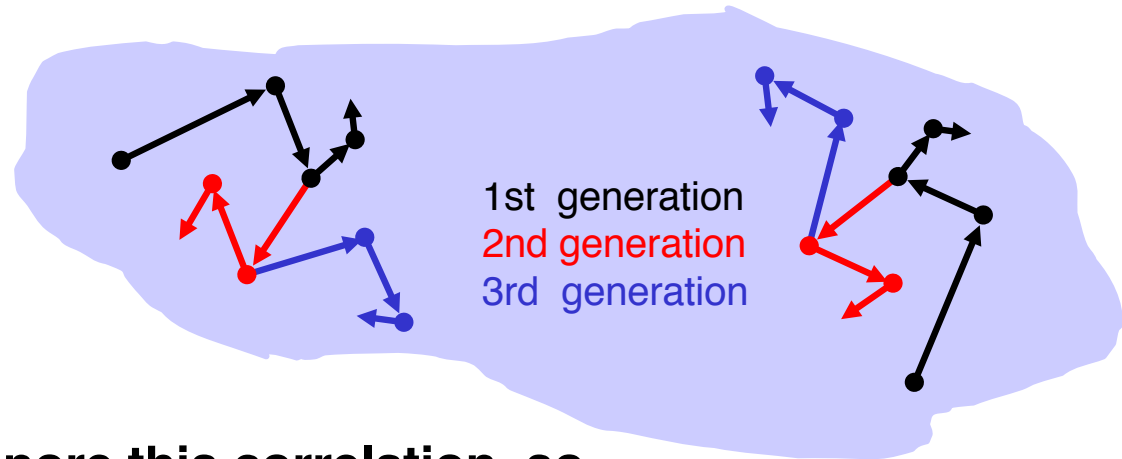
$M = \text{neutrons / cycle}$

- **Power & other tally distributions are also biased, produces “tilt”**

## Bias in Statistics

- MC eigenvalue calculations are solved by power iteration

- Tallies for one generation are spatially correlated with tallies in successive generations
- The correlation is positive
- MCNP & other MC codes ignore this correlation, so computed statistics are smaller than the real statistics
- Errors in statistics are small/negligible for  $K_{\text{eff}}$ , may be significant for local tallies (eg, fission distribution)
- Running more cycles or more neutrons/cycle does not reduce the underprediction bias in statistics
- (True  $\sigma^2$ ) > (computed  $\sigma^2$ )**, since correlations are positive



$$\frac{\text{True } \sigma_{\bar{X}}^2}{\text{Computed } \sigma_{\bar{X}}^2} = \frac{\sigma_{\bar{X}}^2}{\tilde{\sigma}_{\bar{X}}^2} \approx 1 + 2 \cdot \left( \frac{\text{sum of lag-i correlation}}{\text{coeff's between tallies}} \right)$$

## Best Practices – MC Crit Calcs - Summary

---

- **To avoid bias in  $K_{\text{eff}}$  & tally distributions:**
  - Use 10K or more neutrons/cycle (maybe 100K+ for large system)
  - Always check convergence of both  $K_{\text{eff}}$  &  $H_{\text{src}}$
  - Discard sufficient initial cycles
- **To help with convergence & coverage:**
  - Take advantage of problem symmetry, if possible
  - Use good initial source guess, cover fissionable regions --  
points in each fissile region, or volume source for large systems
- **Run at least a few 100 active cycles  
to allow codes to compute reliable statistics**
- **Statistics on tallies from codes are underestimated, often by 2-5x;  
possibly make multiple independent runs**  
[note: statistics on  $k_{\text{eff}}$  are OK, not underestimated]

## Other Suggestions

---

**For serious work, my work-flow includes the actions below:**

- In MCNP input files, include a summary of { date, names, changes }
- Confirm that calculations used correct versions of code, data, scripts
- Always look at geometry with MCNP plotter
- Always check convergence plots for Keff & Hsrc
- Always check output file (not screen) for lost particles
- Check details if any unusual warnings appear
- Record for each run:
  - Name, date, computer, input/output file names
  - $k_{eff} \pm \sigma$  (combined col/trk/abs only)
  - EALF, ANECF, % fast/intermed/thermal fissions
  - For solutions, H/Pu<sup>239</sup> or H/U<sup>235</sup>
  - Any issues?

**If I'm in a hurry & skip some of the above, I usually end up paying big-time later on – having to repeat work to resolve errors or confusion**

# References

## Previous discussion of details concerning bias, convergence, & statistics and "Best Practices" presented at

- 2008 - PHYSOR Monte Carlo workshop
- 2009 - M&C Monte Carlo workshop
- 2009 - Paper at NCSD topical meeting (best paper award)
- 2010 - PHYSOR Monte Carlo Workshop
- 2008 – present – MCNP Criticality Classes

Presentations available at [mcnp.lanl.gov](http://mcnp.lanl.gov)

### Monte Carlo Methods

F. B. Brown, "Fundamentals of Monte Carlo Particle Transport," LA-UR-05-4983, available at [mcnp.lanl.gov](http://mcnp.lanl.gov) (2005).

### Monte Carlo k-effective Calculations

F.B. Brown, "Review of Best Practices for Monte Carlo Criticality Calculations", ANS NCSD-2009, Richland, WA, Sept 13-17 (2009).

G.E. Whitesides, "Difficulty in Computing the k-effective of the World," *Trans. Am. Nucl. Soc.*, 14, No. 2, 680 (1971).

J. Lieberoth, "A Monte Carlo Technique to Solve the Static Eigenvalue Problem of the Boltzmann Transport Equation," *Nukleonik* 11,213 (1968).

M. R. Mendelson, "Monte Carlo Criticality Calculations for Thermal Reactors," *Nucl. Sci Eng.* 32, 319-331 (1968).

### Shannon entropy & convergence

T. Ueki & F.B. Brown, "Stationarity and Source Convergence in Monte Carlo Criticality Calculations", ANS Topical Meeting on Mathematics & Computation, Gatlinburg, TN April 6-11, 2003 [also, LA-UR-02-6228] (September, 2002).

T. Ueki & F.B. Brown, "Stationarity Modeling and Informatics-Based Diagnostics in Monte Carlo Criticality Calculations", *Nucl. Sci. Eng.* 149, 38-50 [also LA-UR-03-8343] (2005).

F.B. Brown, "On the Use of Shannon Entropy of the Fission Distribution for Assessing Convergence of Monte Carlo Criticality Calculations", proceedings PHYSOR-2006, Vancouver, British Columbia, Canada [also LA-UR-06-3737 and LA-UR-06-6294] (Sept 2006).

R.N. Blomquist, et al., "NEA Expert Group on Source Convergence Phase II: Guidance for Criticality Calculations", 8<sup>th</sup> International International Conference on Criticality Safety, St. Petersburg, Russia, May 28 – June 1, 2007 (May 2007).

### Bias in Keff & Tallies

E.M. Gelbard and R.E. Prael, "Monte carlo Work at Argonne National Laboratory", in Proc. NEACRP Meeting of a Monte Carlo Study Group, ANL-75-2, Argonne National Laboratory, Argonne, IL (1974).

R. C. Gast and N. R. Candelore, "Monte Carlo Eigenfunction Strategies and Uncertainties," in Proc. NEACRP Meeting of a Monte Carlo Study Group, ANL-75-2, Argonne National Laboratory, Argonne, IL (1974).

R. J. Brissenden & A. R. Garlick, "Biases in the Estimation of Keff and Its Error by Monte Carlo Methods," *Ann. Nucl. Energy*, 13, 2, 63-83 (1986)

### Correlation & Bias in Uncertainties

T. Ueki and F. B. Brown, "Autoregressive Fitting for Monte Carlo K-effective Confidence Intervals," *Trans. Am. Nucl. Soc.*, 86, 210 (2002).

T. Ueki, "Time Series Modeling and MacMillan's Formula for Monte Carlo Iterated-Source Methods," *Trans. Am. Nucl. Soc.*, 90, 449 (2004).

T. Ueki, "Intergenerational Correlation in Monte Carlo K-Eigenvalue Calculations", *Nucl. Sci. Eng.* 141, 101-110 (2002)

T. Ueki & B. R. Nease, "Times Series Analysis of Monte Carlo Fission Sources - II: Confidence Interval Estimation", *Nucl. Sci. Eng.*, 153, 184-191 (2006).

D. B. MacMillan, "Monte Carlo Confidence Limits for Iterated-Source Calculations," *Nucl. Sci. Eng.*, 50, 73 (1973).

E. M. Gelbard and R. E. Prael, "Computation of Standard Deviations in Eigenvalue Calculations," *Prog. Nucl. Energy*, 24, 237 (1990).

---

# Practical Examples for NCS Analysts

## Examples using Whisper

---

- **Pu 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
- **U Metal Examples**
  - Example 6: U billet with graphite/furnace insulation reflection
  - Example 7: U cylinder with Ta Reflection
- **General Studies**
  - Example 8: Revisiting a Practical Application of the Single-Parameter-Subcritical-Mass Limit for Plutonium Metal with Whisper
  - Example 9: Critical-mass curves and USL-mass curves comparison

Note for examples & demo:

To save time for class demos & running on a laptop, the full suite of 1101 Whisper Benchmarks is not used. Rather, a set of 246 benchmarks including sensitivity profiles from a LANL NCS traditional validation suite is used as the catalog. Parameters for running MCNP6 to get application sensitivity profiles use reduced values to save run time.



## Upper Subcritical Limit (USL)

- **For an application:**
  - A calculated  $K_{\text{eff}} < 1.0$  is NOT sufficient to ensure subcriticality
  - Must conservatively account for
    - Bias & uncertainties in the calculational method
    - Uncertainties in the physical model (eg, mass, isotopics, geometry, ...)

$K_{\text{eff}} = 1$



**Bias** = mean ( $K_{\text{calc}} - K_{\text{exp}}$ ) for a set of experiments that are similar to the application



**Bias Uncertainty**, at 95% or 99% confidence level



**Margin of Subcriticality (MOS)** = code & data uncertainties



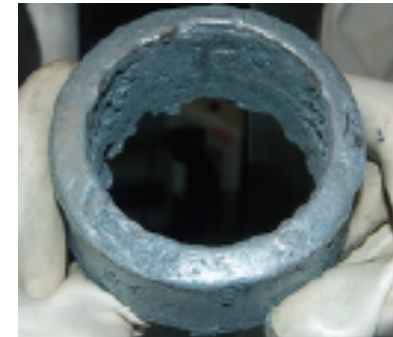
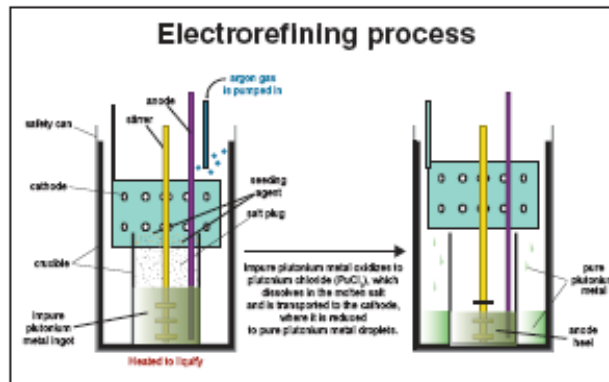
**USL**

**MOS for Area of Applicability (AOA)** = if benchmarks are not similar enough to application

**Must have:**  $K_{\text{calc}} + 2\sigma_{\text{calc}} < \text{USL}$

# Pyrochemical Processing

- **Electrorefining is a batch plutonium metal purification process**
  - **Feed:** impure plutonium metal ingot
  - **Product:** pure plutonium metal ring
  - **Waste:** salt, anode heel, crucible

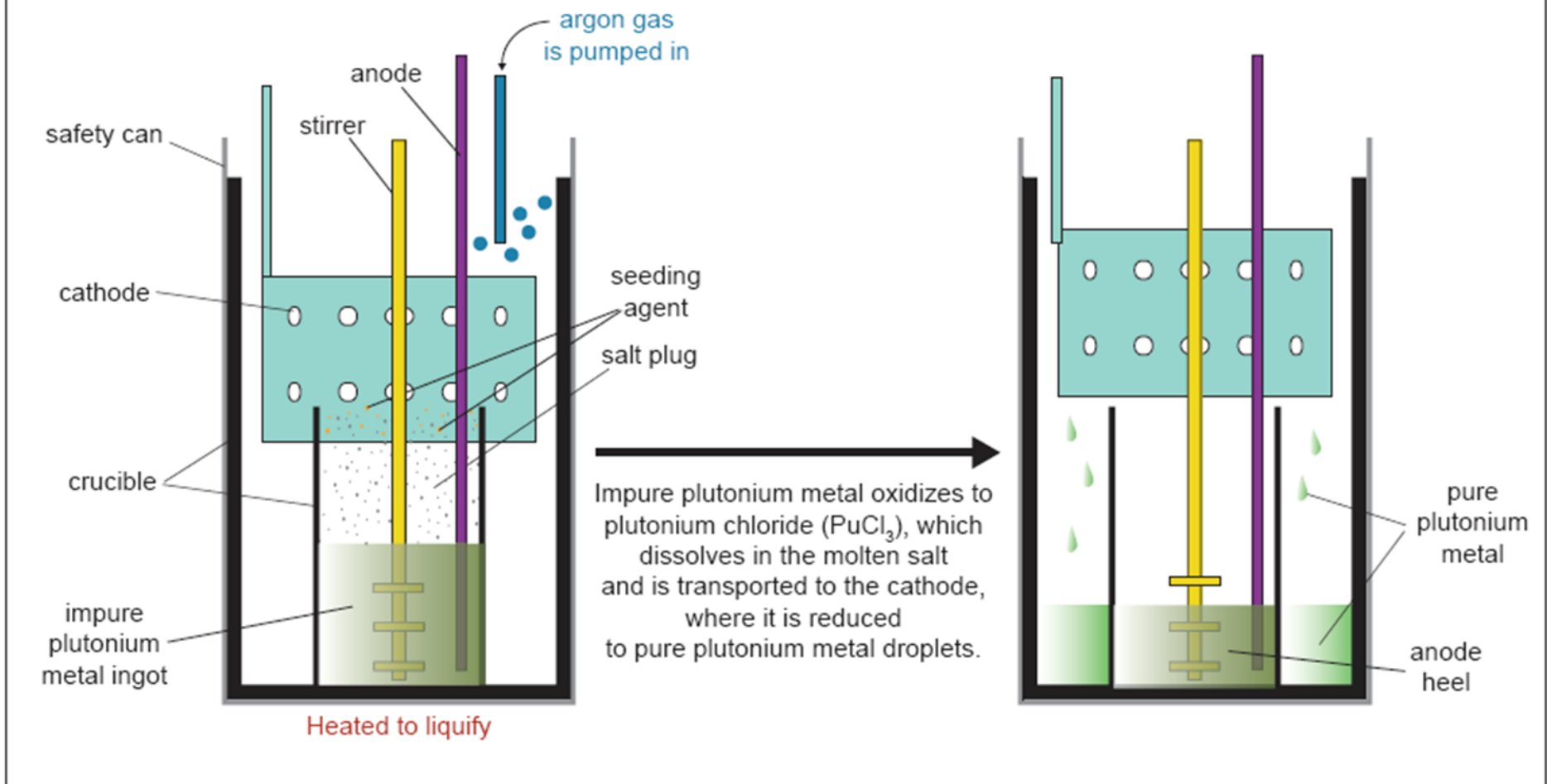


Ref. Actinide  
Research Quarterly  
3<sup>rd</sup> Quarter 2008

- **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 is transported through the electrolyte to the cathode
- Reduced to metal dripping into the outer cup

# Pyrochemical Processing

## Electrorefining process



Ref. Actinide  
Research Quarterly  
3<sup>rd</sup> Quarter 2008

# Example 1

—

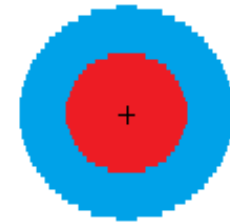
**4.5 kg Pu Ingot,  
varying H/D**

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

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



- Vary the height-to-diameter (H/D) over the range 0.5 – 3.0



- Start with **wval1.txt**, input for H/D = 1

mcnp6 i=wval1.txt

- Copy **wval1.txt** to **wval1p.txt**, then insert directives for mcnp\_pstudy

- Define list for HD:

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

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

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

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

$$R = [V / 2\pi(H/D)]^{1/3}$$

- Use parameters for dimensions & location of KSRC point

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

```
wvall1: 4500 g Pu metal, H/D = 1
c reflected 1 inch water radially,
c 0.25 in steel bottom
c
1 1 -19.860000 -1 imp:n=1
11 3 -1.0 +1 -11 imp:n=1
14 6 -7.92 -30 imp:n=1
15 0 +11 +30 -20 imp:n=1
20 0 +20 imp:n=0

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

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

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

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

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

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

---

- Parameter study using **mcnp\_pstudy**, **whisper\_mcnp**, & **whisper\_usl**:

```
mcnp_pstudy.pl -i wvallp.txt -whisper
```

```
use mcnp_pstudy to create inp files  
inp_case001, inp_case002, ... inp_case_006
```

```
whisper_mcnp.pl inp_case*
```

```
use whisper_mcnp to run mcnp6 for each case &  
produce  $k_{\text{eff}}$  & sensitivity profile tallies  
items in green are for class demo, so that cases run quickly,  
& should not be used for serious work  
-neutrons 10000 -discard 50 -cycles 250 -threads 4
```

```
whisper_usl.pl
```

```
use whisper_usl to run Whisper & determine USL for each case
```

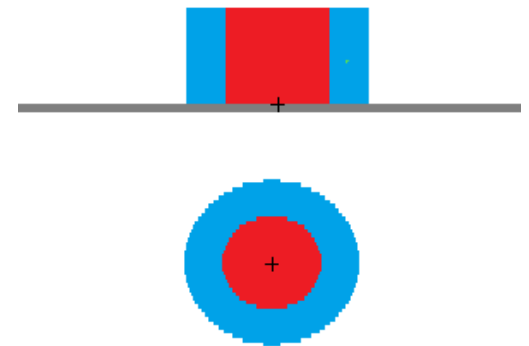
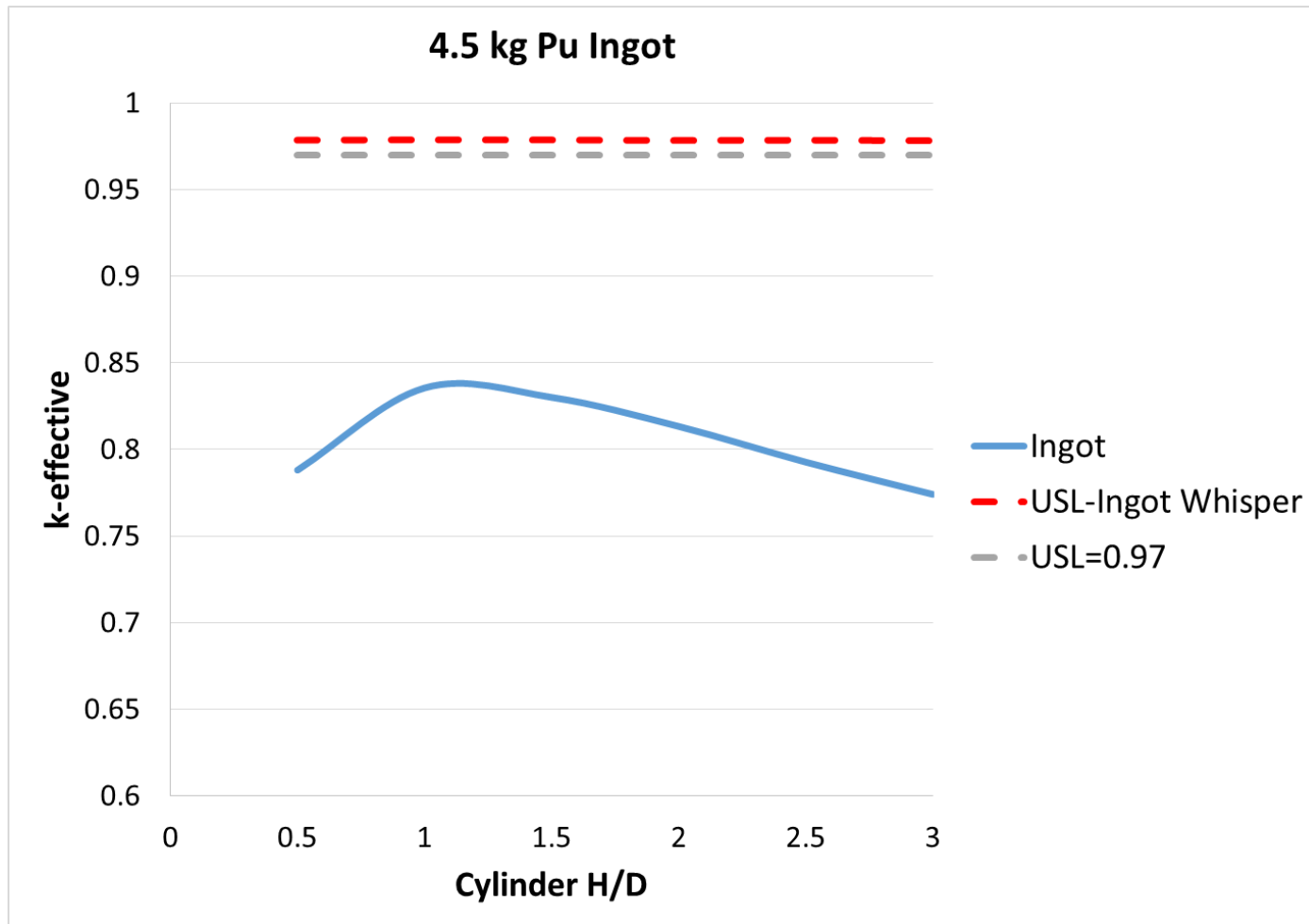
## Example 1: 4.5 kg Pu Ingot, varying H/D (4)

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

$k = 0.83491$  (41)

wval1p, varying H/D  
mcnp\_pstudy -i wval1p.txt -setup -run

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





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

### MCNP6-Whisper Results

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

Benchmark population	=	44
Population weight	=	25.38028
Maximum similarity	=	0.99621
Bias	=	0.00858
Bias uncertainty	=	0.00583
Nuc Data uncert margin	=	0.00076
Software/method margin	=	0.00500
Non-coverage penalty	=	0.00000

benchmark	ck	weight
pu-met-fast-036-001.i	0.9962	1.0000
pu-met-fast-022-001.i	0.9957	0.9850
pu-met-fast-024-001.i	0.9956	0.9813
pu-met-fast-001-001.i	0.9940	0.9319
pu-met-fast-023-001.i	0.9937	0.9207
pu-met-fast-039-001.i	0.9932	0.9069
mix-met-fast-009-001.i	0.9923	0.8774
pu-met-fast-044-005.i	0.9917	0.8598
pu-met-fast-035-001.i	0.9913	0.8449
pu-met-fast-025-001.i	0.9902	0.8117
pu-met-fast-009-001.i	0.9898	0.7976

pu-met-fast-044-003.i	0.9896	0.7926
pu-met-fast-044-004.i	0.9894	0.7867
pu-met-fast-044-002.i	0.9887	0.7646
pu-met-fast-029-001.i	0.9867	0.7006
pu-met-fast-021-002.i	0.9865	0.6966
pu-met-fast-011-001.i	0.9848	0.6430
pu-met-fast-030-001.i	0.9845	0.6328
pu-met-fast-031-001.i	0.9844	0.6284
pu-met-fast-042-004.i	0.9823	0.5620
pu-met-fast-042-006.i	0.9820	0.5543
pu-met-fast-021-001.i	0.9815	0.5387
pu-met-fast-042-003.i	0.9813	0.5304
pu-met-fast-042-007.i	0.9812	0.5301
pu-met-fast-042-005.i	0.9809	0.5189
pu-met-fast-042-009.i	0.9808	0.5153
pu-met-fast-042-008.i	0.9807	0.5119
pu-met-fast-042-010.i	0.9802	0.4971
pu-met-fast-042-012.i	0.9802	0.4959
pu-met-fast-042-011.i	0.9800	0.4908
pu-met-fast-042-002.i	0.9799	0.4873
pu-met-fast-042-015.i	0.9795	0.4759
pu-met-fast-042-013.i	0.9794	0.4707
pu-met-fast-042-014.i	0.9793	0.4690
pu-met-fast-027-001.i	0.9752	0.3389
pu-met-fast-042-001.i	0.9748	0.3267
pu-met-fast-044-001.i	0.9743	0.3134
pu-met-fast-018-001.i	0.9741	0.3057
mix-met-fast-007-022.i	0.9733	0.2819
pu-met-fast-003-103.i	0.9714	0.2215
mix-met-fast-007-023.i	0.9709	0.2041
mix-met-fast-001-001.i	0.9675	0.0979
pu-met-fast-045-005.i	0.9668	0.0777
pu-met-fast-032-001.i	0.9644	0.0015

### Traditional Validation Results:

$$\text{USL} = 0.99\text{-MOS-AoA} = 0.97 - \text{AoA}$$

## Example 2

—

**4.5 kg Pu Annulus,  
varying H & R<sub>in</sub>**

## Example 2: 4.5 kg Pu Annulus, varying H & R<sub>in</sub> (1)

- Establishing Subcriticality – ANSI/ANS-8.1 mass subcritical limits apply to a single piece having no concave surfaces.  
–Does SPSL apply to a ring with concave surfaces?
- Is annular cylinder validated geometry?

From a typical traditional validation report

Parameter	Area of Applicability
Fissile Material	<sup>239</sup> Pu
Fissile Material Form	Pu Metal, PuO <sub>2</sub> , and Pu(NO <sub>3</sub> ) <sub>4</sub>
H/ <sup>239</sup> Pu	$0 \leq H/239Pu \leq 2807$
Average Neutron Energy Causing Fission (MeV)	$0.003 \leq \text{ANECF} \leq 1.935$
<sup>240</sup> Pu	0 to 42.9 wt% <sup>240</sup> Pu
Moderating Materials	none, water, graphite, polystyrene
Reflecting Materials	none, water, steel, oil, Plexiglas, polyethylene, graphite, W, Cu, U, Th, Al, Ni, Fe, Pb, Cd, Mo, Be, BeO
Other Materials	concrete, PVC, Ga, B, Gd, Ta
Geometry	cylinder array, cylinder, slab, sphere, hemisphere, stacked discs, cuboid, annular

### 5.3 Metallic units

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

Table 3 – Single-parameter subcritical limits for metal units

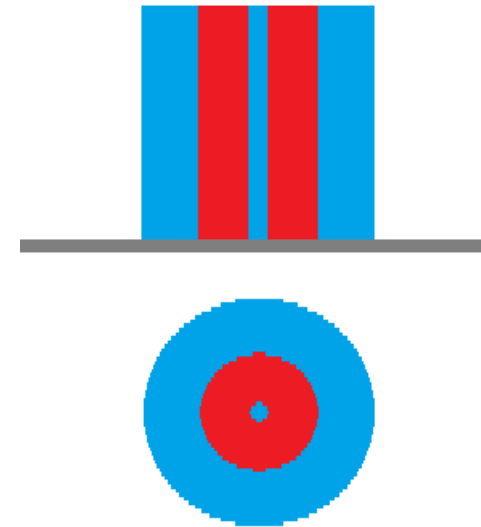
Parameter	Subcritical limits for		
	<sup>233</sup> U [15]	<sup>235</sup> U [16]	<sup>239</sup> Pu [17]
Mass of fissile nuclide (kg)	6.0	20.1	5.0

- How can this be established; what benchmarks include this geometry? Are these benchmarks similar to the ring?

Benchmark	<sup>240</sup> Pu wt%	Form	Geometry	Moderator / Reflector	H/ <sup>239</sup> Pu	Other Materials
pu-sol-therm-032-001	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	449.5	Steel
pu-sol-therm-032-002	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	488.2	Steel
pu-sol-therm-032-003	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	555.3	Steel
pu-sol-therm-032-004	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	622.5	Steel
pu-sol-therm-032-005	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	700.7	Steel
pu-sol-therm-032-006	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	800.5	Steel
pu-sol-therm-032-007	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	850.5	Steel
pu-sol-therm-032-008	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	949.6	Steel
pu-sol-therm-032-009	10.0	Pu(NO <sub>3</sub> ) <sub>4</sub>	Annular	Water/Water	1021.5	Steel

## Example 2: 4.5 kg Pu Annulus, varying H & R<sub>in</sub> (2)

- 4.5 kg Pu-239 right-circular cylinder, hollow
- Pu density = 19.86 g/cm<sup>3</sup>
- 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<sup>+</sup> - 2 cm



- Start with **wval2.txt** input

mcnp6 i=wval2.txt

- Copy **wval2.txt** to **wval2p.txt**, then insert directives for mcnp\_pstudy

- Define list for solid HD:

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

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

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

- Then other dimensions & source

Solid cylinder

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

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

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

Hollow cylinder

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

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

## Example 2: 4.5 kg Pu Annulus, varying H & R<sub>in</sub> (3)

```

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

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

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

```

```

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

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

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

```

## Example 2: 4.5 kg Pu Annulus, varying H & R<sub>in</sub> (4)

---

- Parameter study using **mcnp\_pstudy**, **whisper\_mcnp**, & **whisper\_usl**:

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

```
use mcnp_pstudy to create inp files  
inp_case001, inp_case002, ..., inp_case_012
```

```
whisper_mcnp.pl inp_case*
```

```
use whisper_mcnp to run mcnp6 for each case &  
produce keff & sensitivity profile tallies  
items in green are for class demo, so that cases run quickly,  
& should not be used for serious work  
-neutrons 10000 -discard 50 -cycles 250 -threads 4
```

```
whisper_usl.pl
```

```
use whisper_usl to run Whisper & determine USL for each case
```

## Example 2: 4.5 kg Pu Annulus, varying H & R<sub>in</sub> (5)

wval2

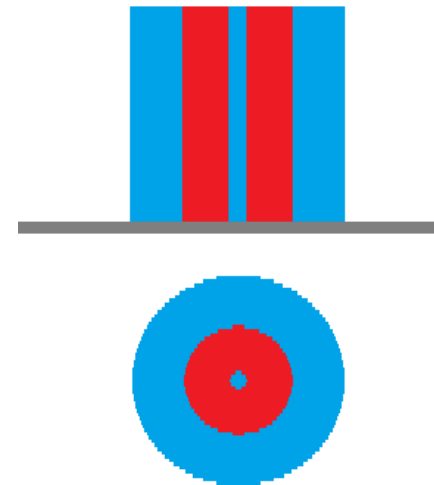
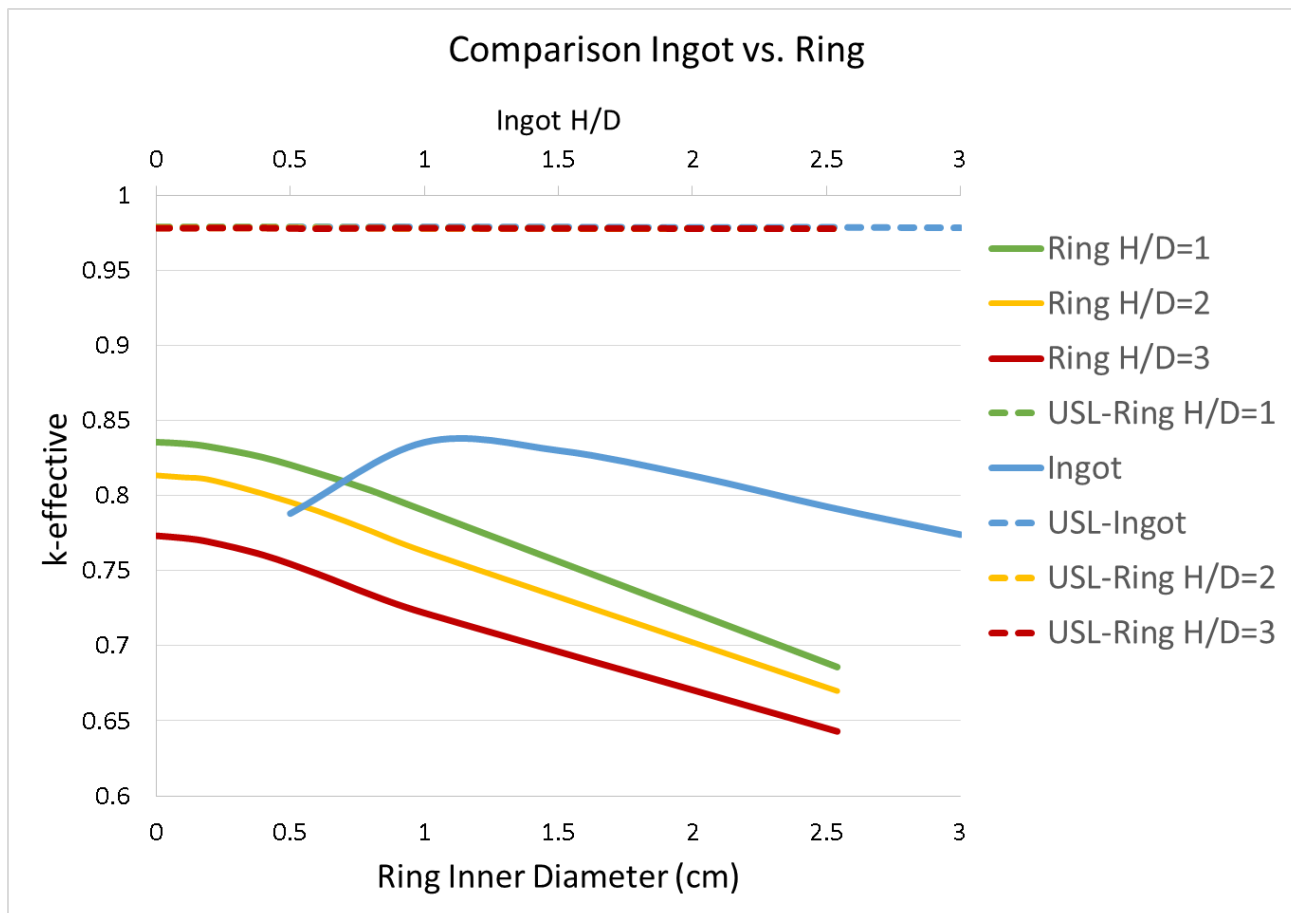
mcnp6 i=wval2.txt

k = 0.83413 (42)

wval2p

mcnp\_pstudy -i wval2p.txt  
-setup -run

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



## Example 2: 4.5 kg Pu Annulus, varying H & R<sub>in</sub> (6)

### MCNP6-Whisper Results

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

Benchmark population	=	41
Population weight	=	25.47164
Maximum similarity	=	0.99532
Bias	=	0.00836
Bias uncertainty	=	0.00628
Nuc Data uncert margin	=	0.00075
Software/method margin	=	0.00500
Non-coverage penalty	=	0.00000

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

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

Benchmarks are the same as those for the ingot in example 1

### Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 - AoA



## Example 3

—

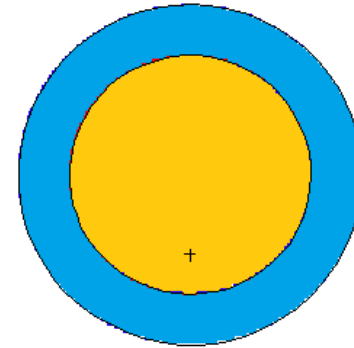
**4.5 kg Pu-NaCl Mixture**

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

---

- 4.5 kg Pu (0) sphere mixed with variable amounts (0-2 kg) of NaCl
- Reflected with 1 inch of water
- Density of Pu = 19.86 g/cm<sup>3</sup>
- Density of NaCl = 1.556 g/cm<sup>3</sup>
- Run commands:

```
mcnp_pstudy -i wval3p.txt -whisper  
whisper_mcnp.pl inp_case*  
whisper_us1.pl
```



For whisper\_mcnp.pl, these (nondefault) options are used for class:

**-neutrons 10000 -discard 50 -cycles 250 -threads 4**

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

wval3: Study of Pu mixed with NaCl

```
c
  1  4 -6.163863  -1    imp:n=1
  2  1 -1.0        +1 -2 imp:n=1
20  0              +2    imp:n=0
```

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

```
kcode 10000 1.0 150 500
sdef pos=0 0 0 rad=d1
  sil 0 5.989
  spl -21 2
```

```
c
m1 1001.80c 2 8016.80c 1
mt1 lwtr.20t
m4 94239.80c -0.81117881
  11023.80c -0.07427730
  17035.80c -0.08561650
  17037.80c -0.02893221
```

wval3p: Pu mixed with NaCl

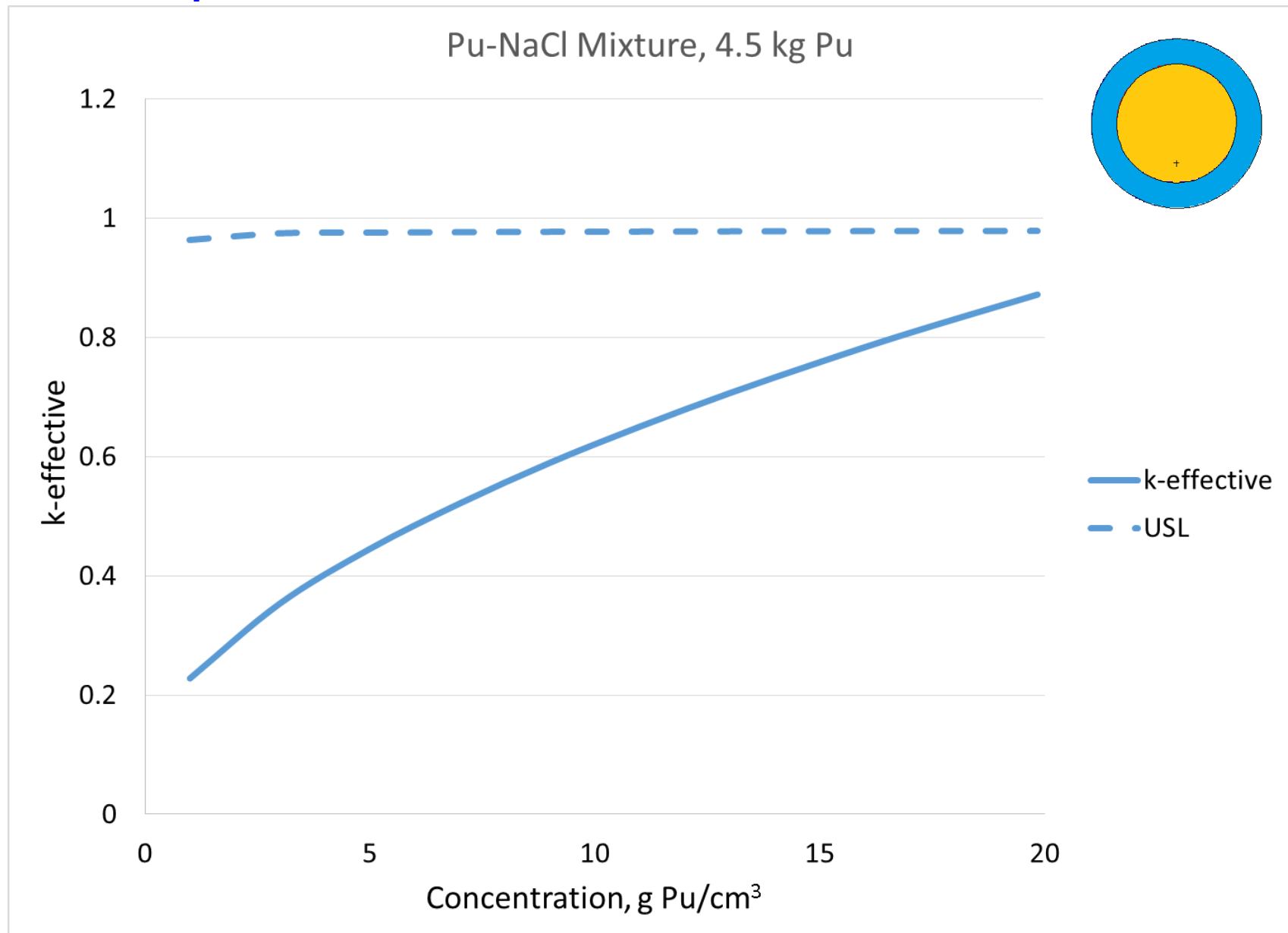
```
c @@@ PI = 3.141592654
c @@@ PU_MASS = 4500
c @@@ PU_VOL = ( PU_MASS / 19.86 )
c @@@ NACL_MASS = 1.e-6 500 1000 1500 2000
c @@@ NACL_VOL = ( NACL_MASS / 1.556 )
c
c Pu mass = PU_MASS g
c NaCl mass = NACL_MASS g
c Pu density (pure) = 19.86 g/cc
c NaCl density (pure) = 1.556 g/cc
c
c @@@ VOLUME = (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

```
  1 so RADIUS
  2 so OUTER_H2O
```

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

## Example 3: 4.5 kg Pu-NaCl Mixture (3)

### MCNP6-Whisper Results



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

### MCNP6-Whisper Results

\*bold indicates same benchmark selected for Pu ingot

USL baseline = .979

Benchmark population	=	46
Benchmark weight	=	25.75745
Benchmark similarity	=	0.99245
Bias	=	0.00796
Bias uncertainty	=	0.00682
Nuc Data	=	0.0012
Software/method margin	=	0.005
Non-coverage penalty	=	0

benchmark	ck	weight
pu-met-fast-011-001.i	0.9924	1
pu-met-fast-044-004.i	0.9842	0.8636
pu-met-fast-042-001.i	0.9831	0.8448
pu-met-fast-042-002.i	0.9828	0.8396
pu-met-fast-044-005.i	0.9827	0.8377
pu-met-fast-027-001.i	0.981	0.8107
pu-met-fast-036-001.i	0.9805	0.8018
pu-met-fast-042-003.i	0.9802	0.7965
pu-met-fast-031-001.i	0.9792	0.7798
pu-met-fast-042-004.i	0.9787	0.7727
pu-met-fast-024-001.i	0.978	0.7604
pu-met-fast-044-003.i	0.9768	0.7401
pu-met-fast-042-005.i	0.9757	0.7213
pu-met-fast-042-006.i	0.9746	0.7039
pu-met-fast-021-002.i	0.9737	0.6893

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

### Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 – AoA

## **Example 4**


—

**4.5 kg Pu Sphere,  
Ta Reflector, various thicknesses**

## 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?

Parameter	Area of Applicability
Fissile Material	$^{239}\text{Pu}$
Fissile Material Form	Pu Metal, $\text{PuO}_2$ , and $\text{Pu}(\text{NO}_3)_4$
$\text{H}/^{239}\text{Pu}$	$0 \leq \text{H}/^{239}\text{Pu} \leq 2807$
Average Neutron Energy Causing Fission (MeV)	$0.003 \leq \text{ANECF} \leq 1.935$
$^{240}\text{Pu}$	0 to 42.9 wt% $^{240}\text{Pu}$
Moderating Materials	none, water, graphite, polystyrene
Reflecting Materials	none, water, steel, oil, Plexiglas, polyethylene, graphite, W, Cu, U, Th, Al, Ni, Fe, Pb, Cd, Mo, Be, BeO
Other Materials	concrete, PVC, Ga, B, Gd, Ta 
Geometry	cylinder array, cylinder, slab, sphere, hemisphere, stacked discs, cuboid, annular

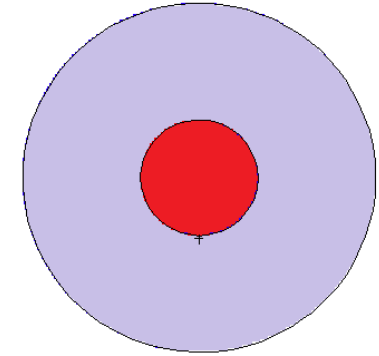
From a  
typical  
traditional  
validation  
report

### • 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 4: 4.5 kg Pu Sphere, Ta-reflected (1)

- 4.5 kg Pu-239 sphere
- Pu density = 19.8 g/cm<sup>3</sup>
- Reflected radially with Ta
- Vary the Ta-reflector thickness over the range 0.1 – 30. cm
  - Start with **wval4.txt**, input for thickness=7.62  
mcnp6 i=wval4.txt
  - Copy **wval4.txt** to **wval4p.txt**, then insert directives for mcnp\_pstudy
    - Define list for thickness:  
c @@@ THICK = 0.01 5. 10. 15. 20. 25. 30.
    - For a given THICK, compute reflector Rin & Rout
    - Use parameters for dimensions & location of KSRC point
    - Run:  
mcnp\_pstudy.pl -i wval4p.txt -whisper  
whisper\_mcnp.pl inp\_case\*  
whisper\_us1.pl





## Example 4: 4.5 kg Pu Sphere, Ta-reflected (2)

### wval4: Study of Pu reflected with Ta

```

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

  1 so  3.7857584
  2 so  11.405758

kcode 10000 1.0 50 250
sdef pos=0 0 0 rad=d1
  sil  0 3.78
  spl  -21 2
c
m1  73180.80c 0.00012      73181.80c 0.99988
m4  94239.80c 1
prtmp 9e9 9e9 1 9e9

```

### wval4p: Study of Pu reflected with Ta

```

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

  1 so      R_INNER
  2 so      R_OUTER

kcode 10000 1.0 50 250
sdef pos=0 0 0 rad=d1
  sil  0 R_INNER
  spl  -21 2
c
m1  73180.80c 0.00012      73181.80c 0.99988
m4  94239.80c 1
prtmp 9e9 9e9 1 9e9

```

## Example 4: 4.5 kg Pu Sphere, Ta-reflected (3)

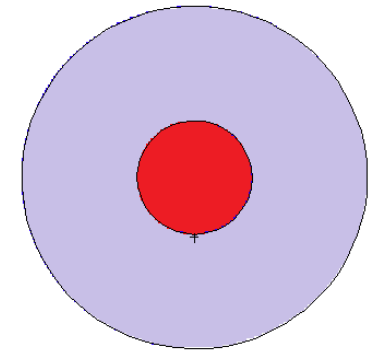
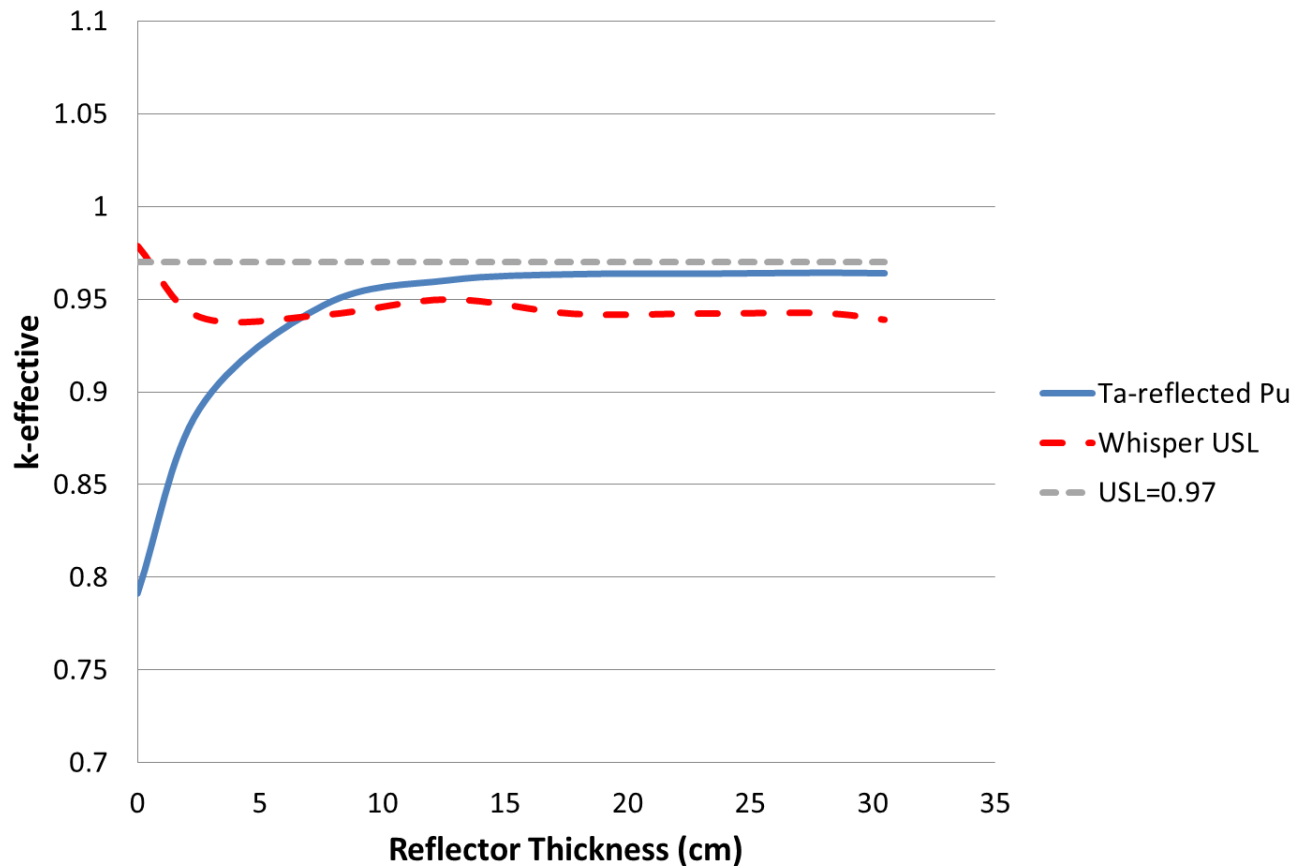
wval4, thick=7.62  
mcnp6 i=wval4.txt

$k = 0.94638$  (41)

wval4p, varying thick  
mcnp\_pstudy -i wval4p.txt -setup -run

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

4.5 kg Pu with Ta Reflection



## Example 4: Ta-reflected Pu

### MCNP6 and Whisper Results

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

Benchmark population = 119  
 Population weight = 60.92464  
 Maximum similarity = 0.64075  
 Bias = 0.00912  
 Bias uncertainty = 0.00795  
 Nuc Data uncert margin = 0.01502  
 Software/method margin = 0.00500  
 Non-coverage penalty = 0.00000

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

**Trouble !  
Benchmarks are  
not very similar  
to application**

Run using all 1101 Whisper benchmarks,  
not just 246 benchmarks for class

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

.....

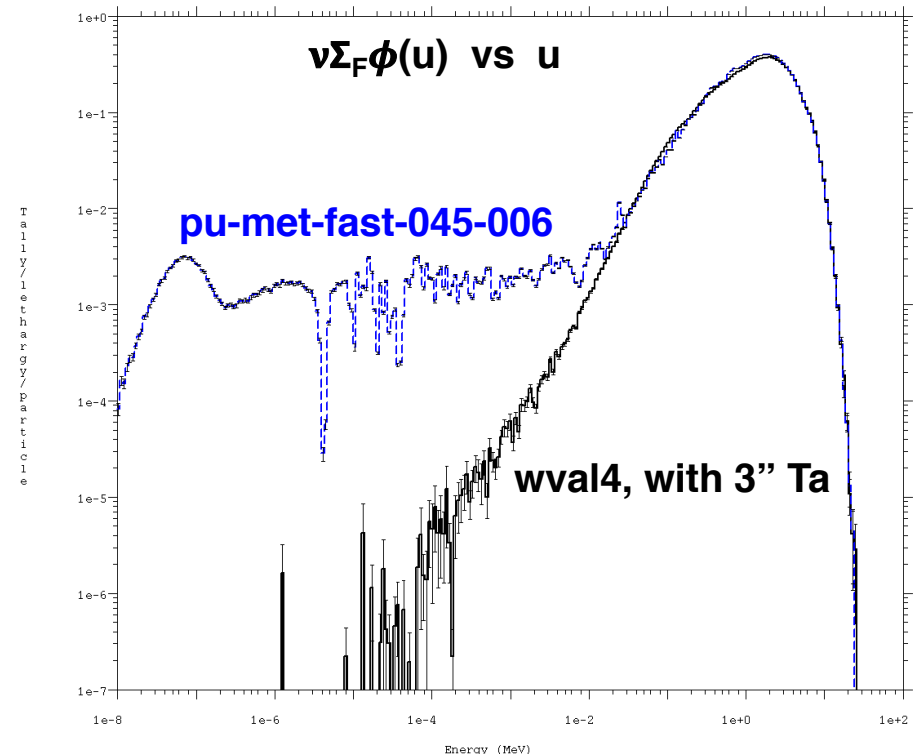
### Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 – AoA

## Example 4: Ta-reflected Pu

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

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

Parameter	Area of Applicability
Fissile Material	$^{239}\text{Pu}$
Fissile Material Form	Pu Metal, $\text{PuO}_2$ , and $\text{Pu}(\text{NO}_3)_4$
$\text{H}/^{239}\text{Pu}$	$0 \leq \text{H}/^{239}\text{Pu} \leq 2807$
Average Neutron Energy Causing Fission (MeV)	$0.003 \leq \text{ANECF} \leq 1.935$
$^{240}\text{Pu}$	0 to 42.9 wt% $^{240}\text{Pu}$
Moderating Materials	none, water, graphite, polystyrene 
Reflecting Materials	none, water, steel, oil, Plexiglas, polyethylene, graphite, W, Cu, U, Th, Al, Ni, Fe, Pb, Cd, Mo, Be, BeO
Other Materials	concrete, PVC, Ga, B, Gd, Ta
Geometry	cylinder array, cylinder, slab, sphere, hemisphere, stacked discs, cuboid, annular

From a  
typical  
traditional  
validation  
report

- Does the calculation model fit within the area of applicability of the benchmark critical experiments used for the code validation?
- For systems which are outside the validation applicability, an AoA margin may be warranted, depending on the specific problem being analyzed.
- 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**
- **4.5 kg Pu (0) sphere mixed with variable amounts of Hydraulic oil**
- **Pu concentration range:**  
**-19.8 g Pu/cm<sup>3</sup>**
- **Hydraulic oil composition:**  
**C<sub>40</sub>H<sub>33</sub>O<sub>4</sub>Cl<sub>6</sub>P**
- **Hydraulic oil density:**  
**0.871 g/cm<sup>3</sup>**
- **Reflected with 1 inch of water**

Pu mixed with hydraulic oil

c

```
1  4 -1.827099  -1      imp:n=1
2  1 -1.0        +1 -2   imp:n=1
20 0              +2      imp:n=0
```

```
1  so    10.2417609488294
```

```
2  so    12.7817609488294
```

```
kcode 10000 1.0 150 500
```

```
ksrc  0 0 0
```

c

```
m1    1001.80c 2
```

```
      8016.80c 1
```

```
mt1   lwtr.20t
```

```
m4    94239.80c -0.54731523
```

```
      1001.80c -0.01821054722413
```

```
      6000.80c -0.264852020155431
```

```
      8016.80c -0.0352799376428247
```

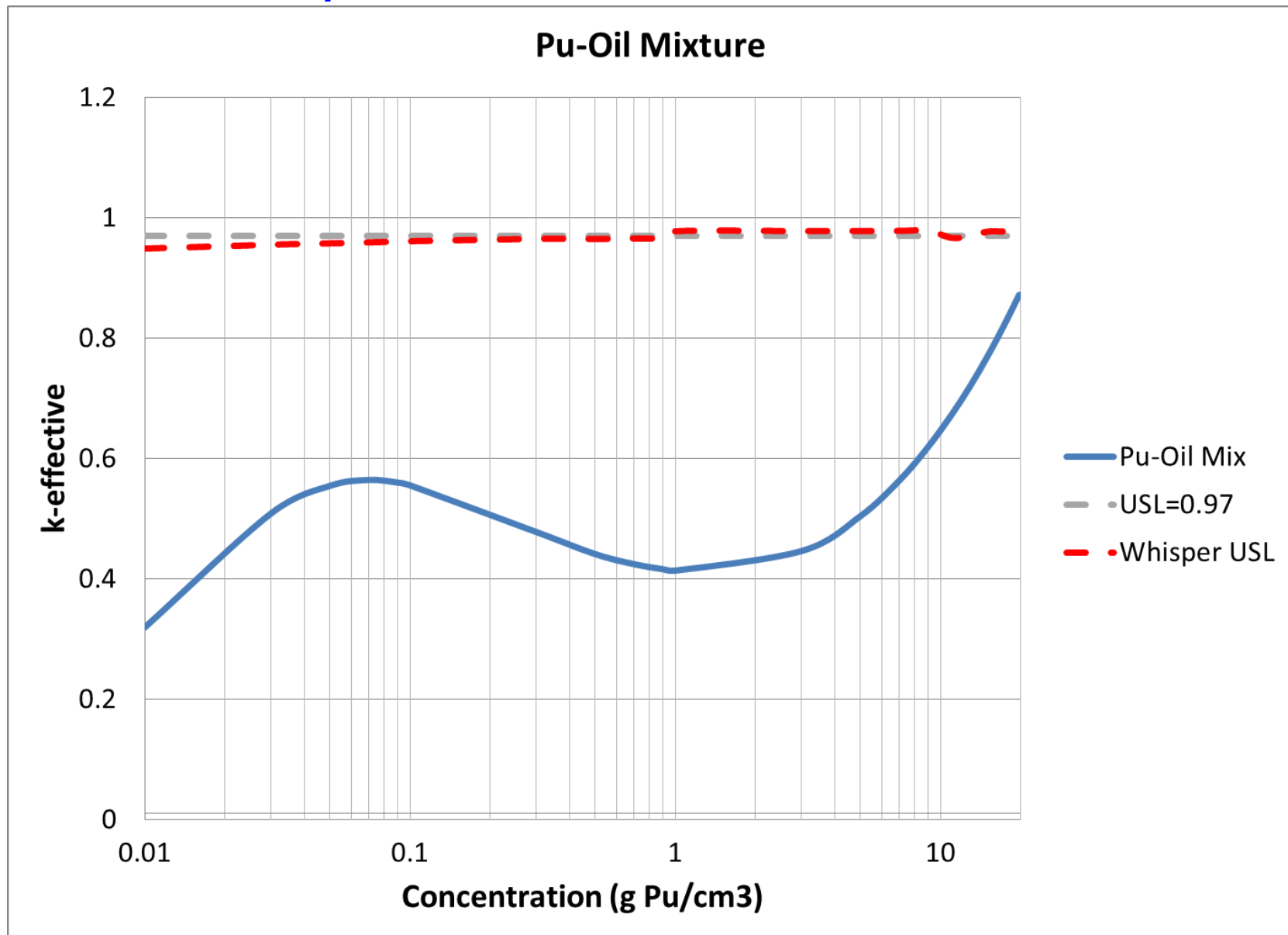
```
      15031.80c -0.0170753227802324
```

```
      17035.80c -0.0876520545992508
```

```
      17037.80c -0.0296143373586584
```

## Example 5: Oil-Moderated Pu

- MCNP6 and Whisper Results





## Example 5: Oil-Moderated Pu

### MCNP6 and Whisper Results

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

Benchmark population	=	65
Population weight	=	28.56693
Maximum similarity	=	0.96433
Bias	=	0.00720
Bias uncertainty	=	0.00757
Nuc Data uncert margin	=	0.00109
Software/method margin	=	0.00500
Non-coverage penalty	=	0.00000

benchmark	ck	weight
pu-met-fast-042-001.i	0.9643	1.0000
pu-met-fast-011-001.i	0.9641	0.9973
pu-met-fast-027-001.i	0.9580	0.9377
pu-met-fast-042-002.i	0.9561	0.9199
pu-met-fast-042-003.i	0.9483	0.8436
pu-met-fast-044-004.i	0.9474	0.8343
pu-met-fast-042-004.i	0.9444	0.8048
pu-met-fast-031-001.i	0.9425	0.7861
pu-met-fast-044-005.i	0.9404	0.7658

pu-comp-mixed-002-001.i	0.9388	0.7502
pu-met-fast-042-005.i	0.9373	0.7353
pu-comp-mixed-002-002.i	0.9344	0.7077
pu-met-fast-042-006.i	0.9344	0.7069
pu-met-fast-042-007.i	0.9320	0.6840
pu-met-fast-036-001.i	0.9310	0.6736
pu-met-fast-044-003.i	0.9307	0.6714
pu-met-fast-042-008.i	0.9303	0.6673
pu-met-fast-024-001.i	0.9277	0.6417
pu-met-fast-042-009.i	0.9271	0.6360
pu-met-fast-042-010.i	0.9268	0.6327
pu-comp-mixed-002-003.i	0.9267	0.6315
pu-met-fast-042-011.i	0.9255	0.6198
pu-met-fast-042-012.i	0.9228	0.5943
pu-met-fast-044-002.i	0.9224	0.5899
pu-met-fast-042-014.i	0.9224	0.5896
pu-met-fast-042-013.i	0.9222	0.5881
pu-met-fast-042-015.i	0.9209	0.5752
pu-comp-mixed-002-004.i	0.9191	0.5574
pu-met-fast-021-002.i	0.9184	0.5506
pu-met-fast-044-001.i	0.9145	0.5128
pu-met-fast-023-001.i	0.9046	0.4156
pu-met-fast-039-001.i	0.9031	0.4015
pu-comp-mixed-002-005.i	0.9030	0.3999
pu-met-fast-018-001.i	0.9008	0.3782
pu-met-fast-021-001.i	0.8989	0.3598
pu-met-fast-009-001.i	0.8985	0.3564
pu-met-fast-016-001.i	0.8965	0.3364
pu-met-fast-045-005.i	0.8954	0.3259

.....

### Traditional Validation Results:

$$\text{USL} = 0.99\text{-MOS-AoA} = 0.97 - \text{AoA}$$

## **Example 6**

—

**20 kg HEU Billet,  
Reflected by graphite and  
Furnace insulation**

## Ex 6: HEU Cylinder with Graphite & Furnace Reflection

### MCNP6 Input

- 20 kg HEU cylinder
- HEU density:  
18.95 g/cm<sup>3</sup>
- Graphite density:  
2.25 g/cm<sup>3</sup>
- Furnace insulation  
density:  
Al<sub>2</sub>O<sub>3</sub>, 0.5 g/cm<sup>3</sup>
- Reflected with 15 cm  
graphite, followed by 15  
cm insulation

```

billet: g U metal billet
c reflected various thicknesses of graphite,
c      2.25 g/cc CRC Handbook 68th Ed
c reflected various thicknesses of insulation Al2O3,
c      0.5 g/cc
c  V = h*pi*r^2 = h/d*2*pi*r^3
c  r = (V/(2*pi*H/D))^(1/3)
c   pi      = 3.141592654
c   mass_u  = 20000
c   dens_u  = 18.95
c   thick_graph = 15
c   thick_insul = 15
c   vol_u    = (mass_u/dens_u)
c   hd       = 1.0
c   r_u      = ((vol_u / (2*pi*hd))**(1/3))
c   h_u      = (2*r_u*hd)
c   r_graph  = (r_u + thick_graph)
c   r_insul  = (r_u + thick_graph + thick_insul)
c   ksrc_z   = (h_u/2)
1      2      -18.95  -1      imp:n=1
10     1      -2.25   +1 -10   imp:n=1
20     3      -0.50   +10 -20   imp:n=1
30     0              +20 -30   imp:n=1
40     0              +30      imp:n=0

1 rcc 0 0 0 0 0 h_u r_u
10 rcc 0 0 0 0 0 h_u r_graph
20 rcc 0 0 0 0 0 h_u r_insul
30 rcc 0 0 -3 0 0 50 50

kcode 10000 1.0 100 300
ksrc 0 0 ksrc_z
m1 6000.80c 1.0
mt1 grph.20t
m2 92235.80c -0.93 92238.80c -0.07
m3 8016.80c 0.6 13027.80c 0.4

```

## Ex 6: HEU Cylinder with Graphite & Furnace Reflection

calc	data unc	baseline	k(calc)
margin	(1-sigma)	USL	> USL
0.01023	0.00104	0.98208	-0.12937

Benchmark population	=	64
Population weight	=	26.06175
Maximum similarity	=	0.98953

Bias	=	0.00600
Bias uncertainty	=	0.00423
Nuc Data uncert margin	=	0.00104
Software/method margin	=	0.00500
Non-coverage penalty	=	0.00000

benchmark	ck	weight
heu-met-fast-084-004.i	0.9895	1.0000
heu-met-fast-041-003.i	0.9895	0.9993
heu-met-fast-084-023.i	0.9878	0.8775
heu-met-fast-019-001.i	0.9865	0.7800
heu-met-fast-020-002.i	0.9856	0.7194
heu-met-fast-084-001.i	0.9850	0.6748
heu-met-fast-084-017.i	0.9844	0.6315
heu-met-fast-078-025.i	0.9842	0.6183
heu-met-fast-100-001.i	0.9841	0.6131
heu-met-fast-041-004.i	0.9840	0.6063
heu-met-fast-100-002.i	0.9840	0.6041
heu-met-fast-084-015.i	0.9837	0.5794
heu-met-fast-089-001.i	0.9835	0.5638
heu-met-fast-084-011.i	0.9834	0.5602
heu-met-fast-001-001.i	0.9832	0.5439
heu-met-fast-022-002.i	0.9830	0.5324
heu-met-fast-078-027.i	0.9826	0.5022
heu-met-fast-051-002.i	0.9826	0.4993

benchmark	ck	weight
heu-met-fast-078-039.i	0.9825	0.4944
heu-met-fast-078-031.i	0.9825	0.4933
heu-met-fast-078-037.i	0.9824	0.4913
heu-met-fast-012-001.i	0.9824	0.4855
heu-met-fast-078-023.i	0.9823	0.4840
heu-met-fast-084-016.i	0.9823	0.4809
heu-met-fast-084-005.i	0.9823	0.4781
heu-met-fast-078-035.i	0.9821	0.4663
heu-met-fast-084-022.i	0.9821	0.4645
heu-met-fast-044-003.i	0.9818	0.4438
heu-met-fast-044-005.i	0.9818	0.4423
heu-met-fast-018-002.i	0.9815	0.4266
heu-met-fast-044-002.i	0.9814	0.4185
heu-met-fast-051-004.i	0.9814	0.4173
heu-met-fast-044-004.i	0.9814	0.4158
heu-met-fast-063-001.i	0.9811	0.3960
heu-met-fast-010-001.i	0.9810	0.3840
heu-met-fast-044-001.i	0.9808	0.3761
heu-met-fast-078-003.i	0.9808	0.3750
heu-met-fast-007-019.i	0.9807	0.3657
heu-met-fast-010-002.i	0.9806	0.3613
heu-met-fast-008-001.i	0.9806	0.3560
heu-met-fast-084-002.i	0.9804	0.3452
heu-met-fast-063-002.i	0.9803	0.3345
heu-met-fast-084-026.i	0.9802	0.3279
heu-met-fast-079-001.i	0.9801	0.3235
heu-met-fast-065-002.i	0.9796	0.2883
heu-met-fast-041-005.i	0.9796	0.2866
heu-met-fast-079-002.i	0.9794	0.2752
heu-met-fast-043-001.i	0.9794	0.2736
heu-met-fast-084-019.i	0.9794	0.2701
heu-met-fast-084-027.i	0.9793	0.2624

...

## **Example 7**

—

**20 kg HEU Cylinder,  
Reflected by tantalum**

## Example 7: HEU Cylinder with Tantalum Reflection

### MCNP6 Input

- 20 kg HEU cylinder
- HEU density:  
18.95 g/cm<sup>3</sup>
- Tantalum density:  
16.65 g/cm<sup>3</sup>
- Tantalum reflector  
thickness:  
0 – 15 cm
- Reflected radially

```

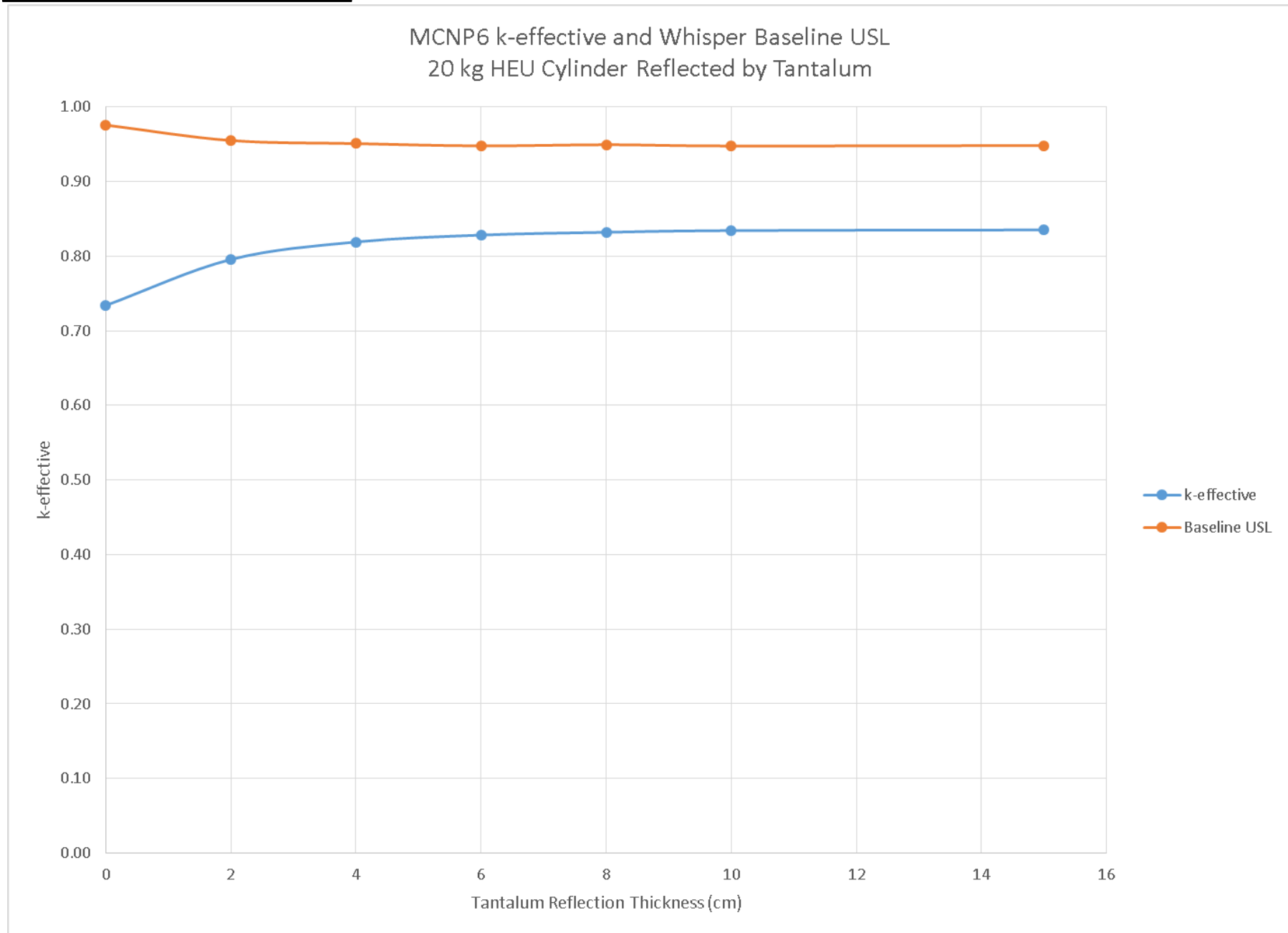
billet: g U metal billet, various thick of Ta
reflection
c
c  V = h*pi*r^2 = h/d*2*pi*r^3
c  r = (V/(2piH/D))^(1/3)
c
c  @@@ pi = 3.141592654
c  @@@ mass_u = 20000
c  @@@ dens_u = 18.95
c  @@@ thick_ta = 0.001,2,4,6,8,10,15
c  @@@ vol_u = (mass_u/dens_u)
c  @@@ hd = 1.0
c  @@@ r_u = ((vol_u / (2*pi*hd))**(1/3))
c  @@@ h_u = (2*r_u*hd)
c  @@@ r_ta = (r_u + thick_ta)
c  @@@ ksrc_z = (h_u/2)
c
c
1      2      -18.95 -1      imp:n=1
10     1      -16.65 +1 -10  imp:n=1
30     0              +10 -30 imp:n=1
40     0              +30      imp:n=0

1 rcc 0 0 0 0 0 h_u r_u
10 rcc 0 0 0 0 0 h_u r_ta
30 rcc 0 0 -3 0 0 50 50

kcode 10000 1.0 100 300
ksrc 0 0 ksrc_z
c
m1 73181.80c 1.0
c
m2 92235.80c -0.93 92238.80c -0.07

```

## Example 7: HEU Cylinder with Tantalum Reflection



## Example 7: HEU Cylinder with Tantalum Reflection

### No Reflection

calc	data unc	baseline	k(calc)
margin	(1-sigma)	USL	> USL
0.01603	0.00119	0.97588	-0.24173

Benchmark population	=	57
Population weight	=	27.93047
Maximum similarity	=	0.97078

Bias	=	0.00608
------	---	---------

Bias uncertainty	=	0.00995
------------------	---	---------

Nuc Data uncert margin	=	0.00119
------------------------	---	---------

Software/method margin	=	0.00500
------------------------	---	---------

Non-coverage penalty	=	0.00000
----------------------	---	---------

benchmark	ck	weight
heu-met-fast-100-002.i	0.9708	1.0000
heu-met-fast-100-001.i	0.9707	0.9989
heu-met-fast-001-001.i	0.9707	0.9978
heu-met-fast-018-002.i	0.9672	0.8851
heu-met-fast-065-002.i	0.9665	0.8642
heu-met-fast-015-001.i	0.9663	0.8577
heu-met-fast-007-019.i	0.9662	0.8559
heu-met-fast-051-002.i	0.9657	0.8403
heu-met-fast-008-001.i	0.9643	0.7960
heu-met-fast-051-004.i	0.9630	0.7530
heu-met-fast-044-001.i	0.9628	0.7478
heu-met-fast-079-001.i	0.9624	0.7355
heu-met-fast-044-002.i	0.9621	0.7259
heu-met-fast-007-001.i	0.9620	0.7204
heu-met-fast-043-001.i	0.9613	0.7013
heu-met-fast-012-001.i	0.9611	0.6922
heu-met-fast-084-001.i	0.9602	0.6645
heu-met-fast-079-002.i	0.9600	0.6580

benchmark	ck	weight
heu-met-fast-044-003.i	0.9595	0.6430
heu-met-fast-084-015.i	0.9591	0.6302
heu-met-fast-078-041.i	0.9587	0.6174
heu-met-fast-084-017.i	0.9582	0.6002
heu-met-fast-044-005.i	0.9576	0.5841
heu-met-fast-084-019.i	0.9576	0.5820
heu-met-fast-043-002.i	0.9575	0.5801
heu-met-fast-044-004.i	0.9572	0.5698
heu-met-fast-022-002.i	0.9565	0.5480
heu-met-fast-025-001.i	0.9563	0.5423
heu-met-fast-089-001.i	0.9557	0.5217
heu-met-fast-079-003.i	0.9551	0.5029
heu-met-fast-084-022.i	0.9550	0.5012
heu-met-fast-084-004.i	0.9530	0.4360
heu-met-fast-043-003.i	0.9520	0.4061
heu-met-fast-027-001.i	0.9518	0.3975
heu-met-fast-092-001.i	0.9511	0.3777
heu-met-fast-079-005.i	0.9505	0.3591
heu-met-fast-084-023.i	0.9505	0.3589
heu-met-fast-084-012.i	0.9501	0.3459
heu-met-fast-079-004.i	0.9497	0.3324
heu-met-fast-084-016.i	0.9493	0.3204
heu-met-fast-084-002.i	0.9490	0.3096
heu-met-fast-084-005.i	0.9488	0.3050
heu-met-fast-020-002.i	0.9483	0.2874
heu-met-fast-043-004.i	0.9473	0.2572
heu-met-fast-043-005.i	0.9464	0.2294
heu-met-fast-087-001.i	0.9461	0.2198
heu-met-fast-078-023.i	0.9453	0.1929
heu-met-fast-084-007.i	0.9451	0.1862
heu-met-fast-063-001.i	0.9441	0.1560
heu-met-fast-019-001.i	0.9427	0.1092
heu-met-fast-041-003.i	0.9426	0.1069...



## Example 7: HEU Cylinder with Tantalum Reflection

### 2cm Tantalum Reflection

calc	data unc	baseline	k(calc)
margin	(1-sigma)	USL	> USL
0.02083	0.00730	0.95519	-0.15947

Benchmark population	=	118
Population weight	=	71.52347
Maximum similarity	=	0.53482

Bias	=	0.01064
Bias uncertainty	=	0.01019
Nuc Data uncert margin	=	0.00730
Software/method margin	=	0.00500
Non-coverage penalty	=	0.00000

benchmark	ck	weight
heu-met-fast-100-002.i	0.5348	1.0000
heu-met-fast-100-001.i	0.5348	0.9984
heu-met-fast-001-001.i	0.5345	0.9892
heu-met-fast-051-002.i	0.5338	0.9673
heu-met-fast-018-002.i	0.5330	0.9437
heu-met-fast-007-019.i	0.5330	0.9426
heu-met-fast-051-004.i	0.5329	0.9406
heu-met-fast-065-002.i	0.5327	0.9324
heu-met-fast-015-001.i	0.5326	0.9284
heu-met-fast-084-001.i	0.5324	0.9239
heu-met-fast-008-001.i	0.5323	0.9208
heu-met-fast-044-001.i	0.5321	0.9147
heu-met-fast-044-002.i	0.5320	0.9115
heu-met-fast-084-017.i	0.5319	0.9090
heu-met-fast-079-001.i	0.5319	0.9073
heu-met-fast-084-015.i	0.5317	0.9024
heu-met-fast-007-001.i	0.5317	0.9016
heu-met-fast-012-001.i	0.5317	0.9013

**Trouble !  
Benchmarks  
are not very  
similar to  
application**

benchmark	ck	weight
heu-met-fast-044-003.i	0.5314	0.8928
heu-met-fast-043-001.i	0.5314	0.8924
heu-met-fast-078-041.i	0.5312	0.8845
-met-fast-079-002.i	0.5310	0.8796
-met-fast-044-005.i	0.5310	0.8779
-met-fast-044-004.i	0.5308	0.8716
-met-fast-089-001.i	0.5307	0.8702
-met-fast-022-002.i	0.5307	0.8691
heu-met-fast-084-004.i	0.5306	0.8674
heu-met-fast-084-019.i	0.5304	0.8613
heu-met-fast-084-022.i	0.5302	0.8533
heu-met-fast-084-023.i	0.5300	0.8486
heu-met-fast-043-002.i	0.5298	0.8424
heu-met-fast-079-003.i	0.5291	0.8204
heu-met-fast-084-005.i	0.5290	0.8149
heu-met-fast-025-001.i	0.5288	0.8107
heu-met-fast-084-016.i	0.5287	0.8069
heu-met-fast-020-002.i	0.5286	0.8043
heu-met-fast-027-001.i	0.5284	0.7957
heu-met-fast-084-002.i	0.5280	0.7858
heu-met-fast-078-023.i	0.5279	0.7828
heu-met-fast-043-003.i	0.5278	0.7787
heu-met-fast-063-001.i	0.5274	0.7650
heu-met-fast-041-003.i	0.5273	0.7630
heu-met-fast-079-005.i	0.5273	0.7614
heu-met-fast-079-004.i	0.5271	0.7571
heu-met-fast-084-012.i	0.5271	0.7557
heu-met-fast-019-001.i	0.5269	0.7503
heu-met-fast-087-001.i	0.5267	0.7419
heu-met-fast-092-001.i	0.5266	0.7391
heu-met-fast-078-025.i	0.5263	0.7314
heu-met-fast-043-004.i	0.5259	0.7187
.....		

## Example 7: HEU Cylinder with Tantalum Reflection

- **None of the benchmarks appear to have the same neutronics as the application**

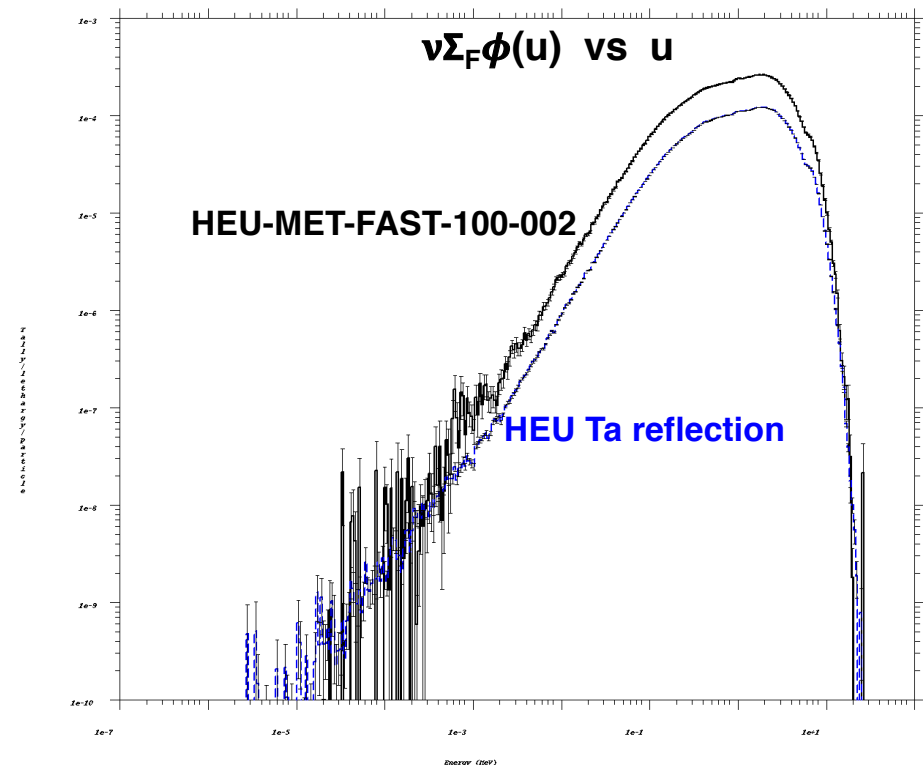
- Largest  $C_k$  in the Whisper example output is 0.53 – very low
- Guidance from ORNL Scale/Tsunami developers:

$0.95 < C_k \rightarrow$  great

$0.90 < C_k < 0.95 \rightarrow$  good

$C_k < 0.90 \rightarrow$  not so good

- 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 validation, few benchmarks with Ta
- Need to find more benchmarks with Ta reflector & add to Whisper suite, if Ta-reflected applications are expected

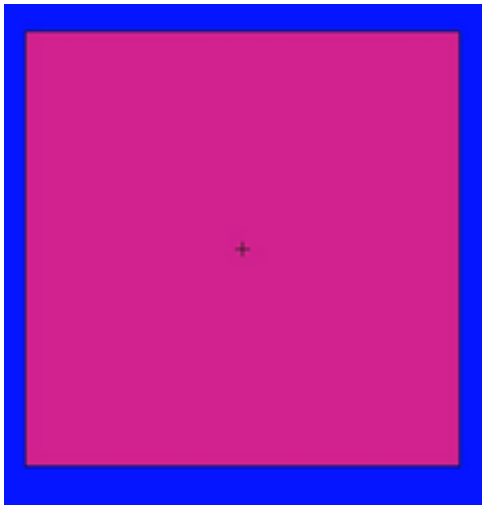
## **Example 8**

—

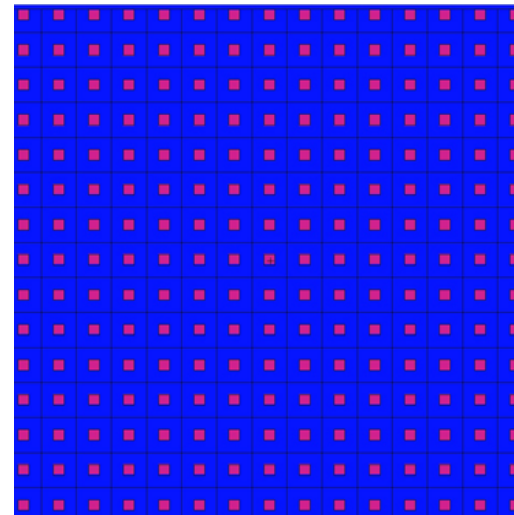
# **Revisiting a Practical Application of the SPSL for Pu Metal**

## Example 8: Revisiting a Practical Application of the SPSL for Pu Metal

- LANL undertook an effort to define a threshold between un-moderated and moderated plutonium metal systems in LA-UR-07-0160, *Practical Application of the Single-Parameter Subcritical Mass Limit for Plutonium*.
- The goal 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.
- The study involving plutonium ( $^{239}\text{Pu}$ ) metal cubes in water was performed using MCNP. This study is revisited, and Upper Subcritical Limits (USLs) are presented, using WHISPER.

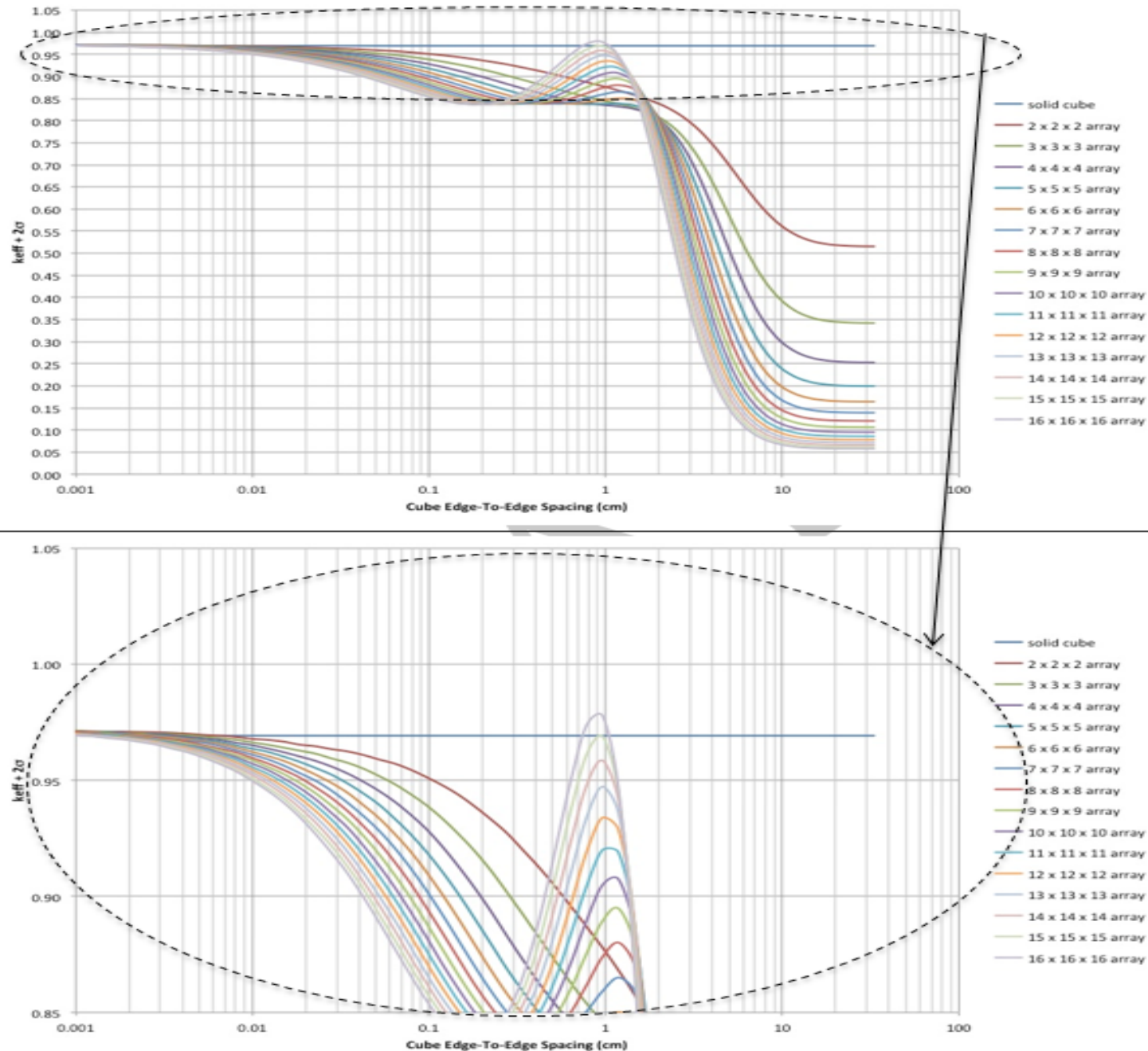


$N = 1$ ,  
Mass Per Cube = 5,000 g,  
Spacing = N/A

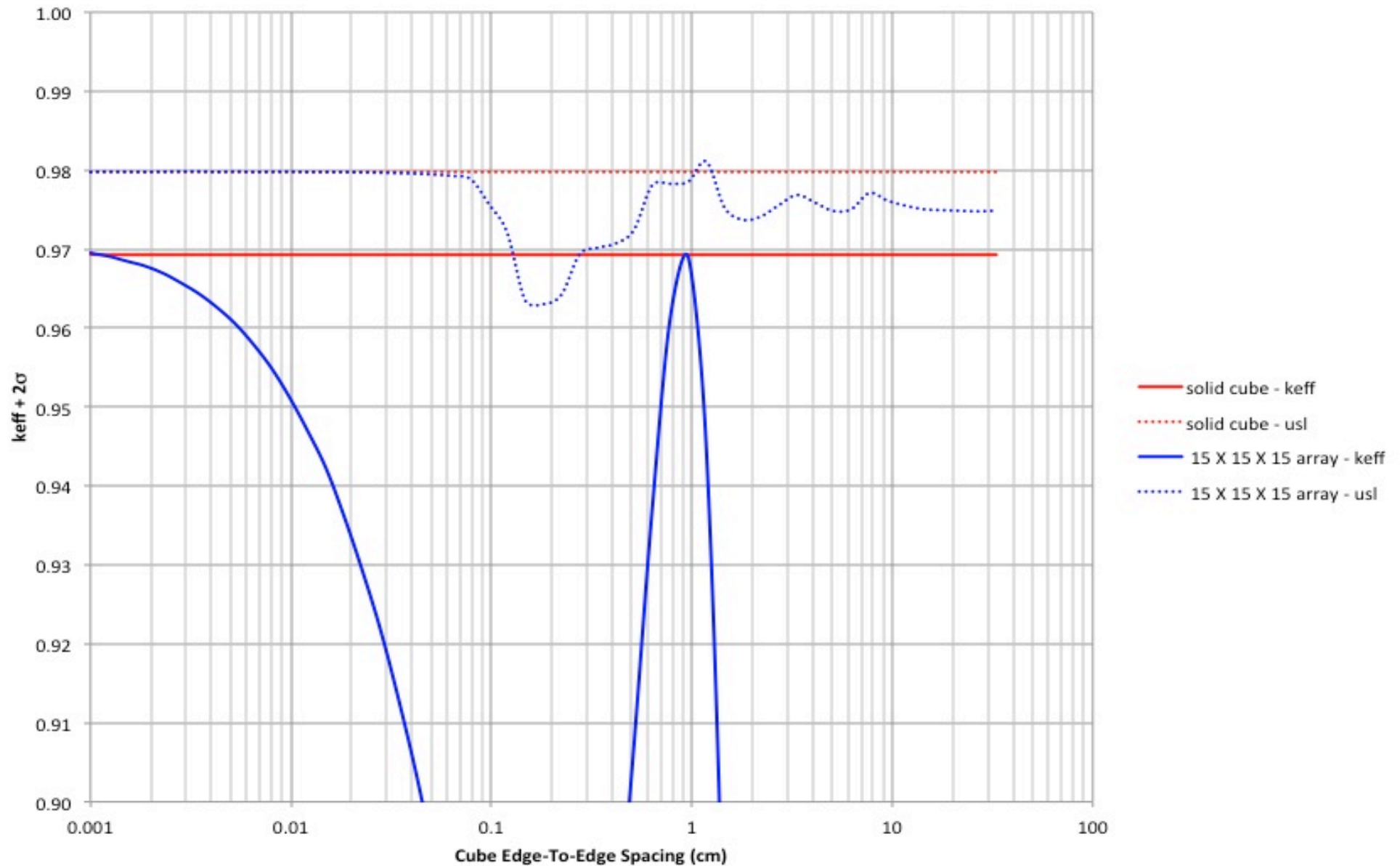


$N = 15$ ,  
Mass Per Cube =  $\sim 1.48$  g,  
Spacing = 1 cm

## Example 8: Revisiting a Practical Application of the SPSL for Pu Metal



## Example 8: Revisiting a Practical Application of the SPSL for Pu Metal

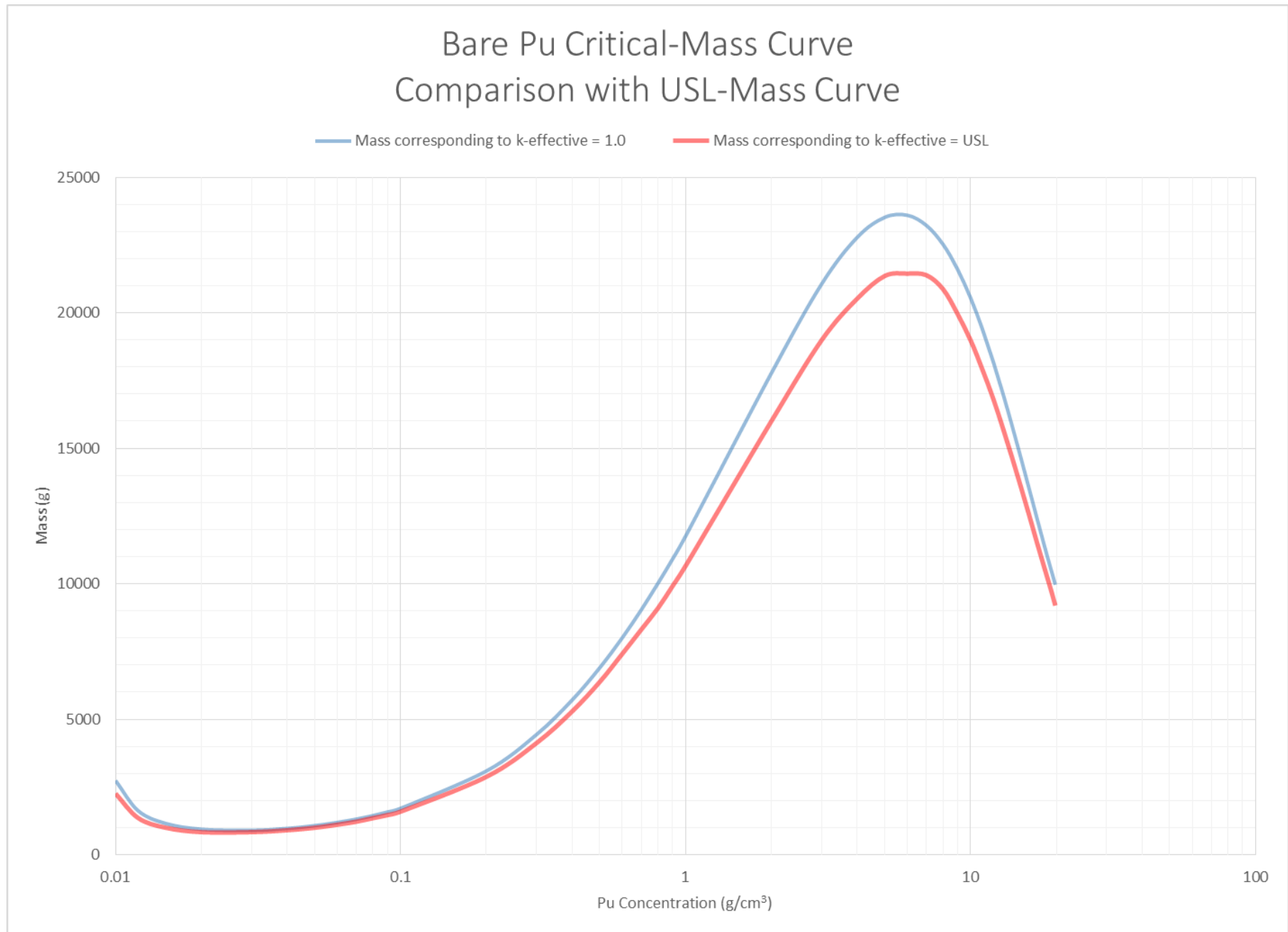


# Example 9

—

## Pu Critical Mass & USL Curves

## Example 9: Critical-Mass and USL-Mass Curves

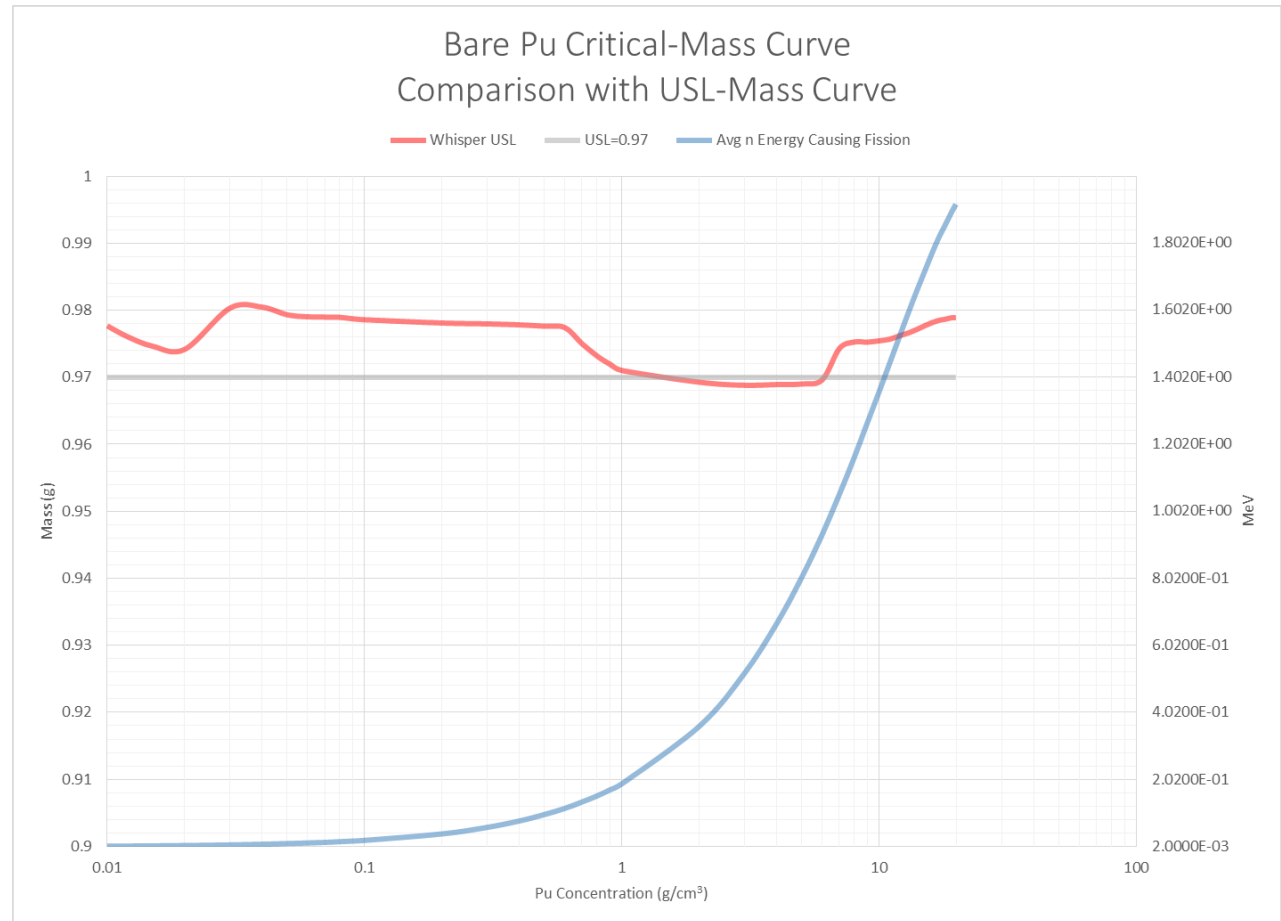




## Example 9: Critical-Mass and USL-Mass Curves

### [ANSI/ANS-8.24 7.2]

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



### THERMAL

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

### INTERMEDIATE

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

### FAST

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

---

# **Using Whisper to Support NCS Validation**

-

## **ANSI/ANS-8.24 Requirements & Recommendations**

## ANS-8.24

Computer Code System	Whisper-1.1
Verification prior to validation (document)	<p>Developers run verification suites and document results.</p> <p>Users must verify installation and operation prior to validation.</p>
Configuration Control	Users must manage configuration.
Changes evaluated to determine effect on validation	<p>Recommend running MCNP6 <b>validation_criticality</b> V&amp;V suite frequently (daily) to look for changes.</p> <p>If changes, determine the cause &amp; fix if possible.</p> <p>If necessary, complete new sensitivity profiles for Whisper benchmark library.</p>

## ANS-8.24

Selection and Modeling of Benchmarks		Whisper-1.1
Appropriate process parameters correlate experiment to application	Identify normal and credible abnormal conditions when determining parameters and values ( <b>benchmarks should encompass range</b> )	Whisper selects benchmark experiments that are most similar to the application using sensitivity profiles to characterize the neutronics of each application and benchmark for each isotope, reaction and energy.
Use the same methods and analysis to analyze benchmark and application		
Review benchmarks prior to use ( <b>should be consistent with modeling capabilities of method; drawn from multiple series; evaluated by organization performing validation</b> )	Experienced users responsible for modeling benchmarks	Benchmark models consistent with MCNP6 capabilities; drawn from multiple series; modeled by experienced MCNP6 users; must be reviewed and evaluated by organization performing validation.

## ANS-8.24

Establishment of Bias, Bias Uncertainty, Margins	Whisper-1.1
Justify positive bias	Does not use positive bias.
Base trending parameters on application	Establishes USL for each application.
Rejection of outliers based on physical behavior or established statistical rejection methods	Rejection based on GLLS with iterative-diagonal $\chi^2$ rejection technique.
Calculational margin consistent with quality and quantity of benchmarks	Selects similar (quality) benchmarks to conduct valid statistical analysis (quantity).
Method consistent with intended use	Consistent (no assumption of normality)
Bias uncertainty allowance for measurement uncertainties; limitations in representations, statistical and convergence uncertainties	Uses experimental and cross-section uncertainties; statistical and convergence uncertainties; parameter studies used for variations in geometry & materials.
Trends used for extrapolation/wide interpolation based on cause	Application-specific USL, possible to trend with output information or parameter study.

## ANS-8.24

<b>Establishment of Bias, Bias Uncertainty, and Margins Margin of Subcriticality (MOS)</b>	
<b>Sufficiently large to ensure calculated conditions will actually be subcritical</b>	<b>MOS<sub>data</sub> based on sensitivity profiles and nuclear data covariances,</b> <b>MOS<sub>code</sub> 0.005 based on MCNP developer expert judgment,</b> <b>MOS<sub>application</sub> must be applied by NCS analyst.</b>
<b>Take into account sensitivity of application to variations in fissile form, geometry, characteristics. Single trend might not be appropriate over entire validation applicability.</b>	<b>Application-specific, see case study for cubic array of metal pieces.</b>

## ANS-8.24

Adequacy of the Validation	
<p>Validation applicability based on benchmark applicability (may be extended)</p>	<p>Sensitivity profiles to select most similar benchmarks, ranked by <math>c_k</math>.</p> <p>Non-similar benchmarks (extrapolate or wide interpolate) have lower <math>c_k</math>;</p> <p><math>c_k &lt; 0.8</math> requires additional margin based on expert judgment.</p>
<p>USL based on CM and MOS</p>	<p>USL = 1 – CM – MOS</p>
<p>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 a higher USL than is lower than that determined for the entire set.</p> <p>Subset of data closely related to application is not nonconservatively masked by benchmarks that do not match the system as well.</p>	<p>Application-specific USL</p> <p>See Whisper Case Study Critical Mass Curve</p>



## ANS-8.24

---

- **Documentation and Independent Technical Review**
  - Trending analysis and technical basis
  - Validation applicability
  - Differences validation applicability – application
  - Limitations
  - MOS and its basis
  - USL and methods to determine
  - Independent technical review
    - Benchmark applicability
    - Input/output files
    - Methodology: CM, MOS
    - Concurrence with validation applicability

---

---

# References

## References for Whisper & MCNP6 (1)

---

All references are available at URL: [mcnp.lanl.gov](http://mcnp.lanl.gov) → Recent Publications → Whisper – NCS Validation

### Abstract

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

### Theory

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

### User Manual

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

### Application

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

## References for Whisper & MCNP6 (2)

---

### Software Quality Assurance

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

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

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

## References for Whisper & MCNP6 (3)

---

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

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

### General References on Adjoints, Perturbation, and Sensitivity Analysis

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

## References for Whisper & MCNP6 (4)

---

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

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

## References for Whisper & MCNP6 (5)

---

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

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