LA-UR-16-25648

Sensitivity and Uncertainty Techniques for Use in Nuclear Criticality Safety

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

Don Mueller
Nuclear Data and Criticality Safety Group, ORNL

Los Alamos NATIONAL LABORATORY EST. 1943

NCSP NUCLEAR CRITICALITY SAFETY PROGRAM

OAK RIDGE National Laboratory

Abstract

Sensitivity and Uncertainty Techniques for use in Nuclear Criticality Safety

Organizer: **James Kuropatwinski**
Nuclear Criticality Safety Division, LANL

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

Don Mueller
Nuclear Data and Criticality Safety Group, ORNL

The sensitivity and uncertainty analysis course will introduce students to k_{eff} sensitivity data, cross-section uncertainty data, how k_{eff} sensitivity data and k_{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 Use in Nuclear Criticality Safety

Day 1

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

Day 2

4. 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
   d) Scale/Tsunami
5. Examples
   a) Pyrochemical Processing – Geometry, Materials, Reflection, Moderation
   b) General Studies
6. 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**

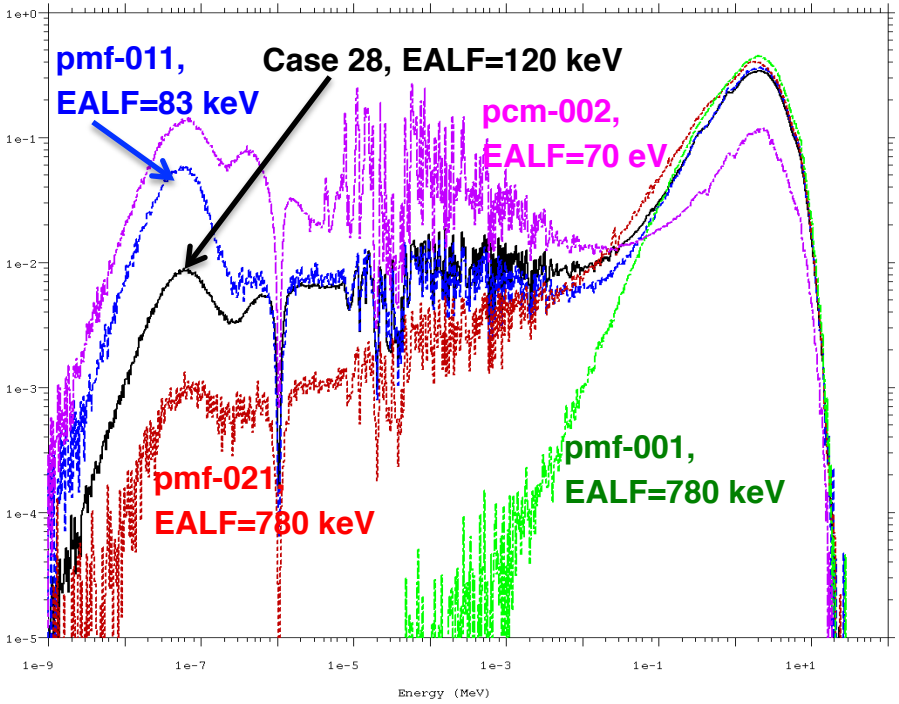
- **Conservatism is fundamental to NCS**
  - **Always consider uncertainties in calculations, data, measurements**
  - **Use additional margin for uncertainties that cannot be calculated**
  - **Subtract uncertainties from upper subcritical limits on K_{eff}**

- **Codes are great, but analyst judgment is required for everything**

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 cross-sections, etc.
- The figure shows neutron production spectra for an application (Case 28) & 4 benchmarks for Pu systems (pmf-001, pmf-011, pcm-002, pmf-021). Which of the benchmarks are similar to the application?
- Simple metrics cannot capture the complexity of fissile systems

$\nu\Sigma_F\Phi$ production spectrum



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

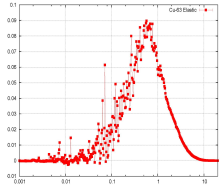
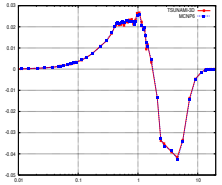
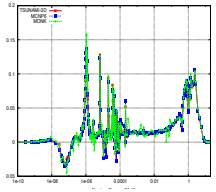
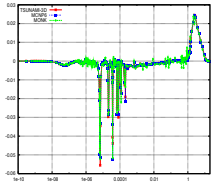
- **From ORNL, the Scale system includes Keno, Tsunami, Tsurfer, & other tools**
- **From LANL, the MCNP6 & Whisper tools are now available**
- **Other tools have been developed by groups in England, France, Germany, Japan, S. Korea, China**

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:

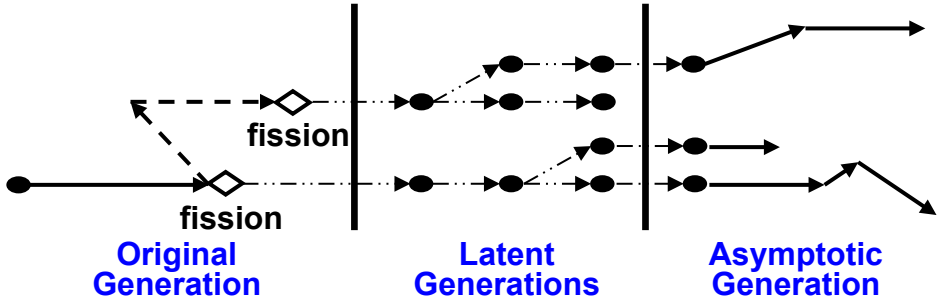
$$S_{k,x} = \frac{dk/k}{dx/x} = -\frac{\left\langle \psi^{\dagger}, \left(\Sigma_x - \mathrm{S}_x - k^{-1}\mathrm{F}_x\right)\psi \right\rangle}{\left\langle \psi^{\dagger}, k^{-1}\mathrm{F}\psi \right\rangle}$$

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

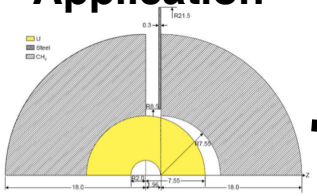
etc.

- **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

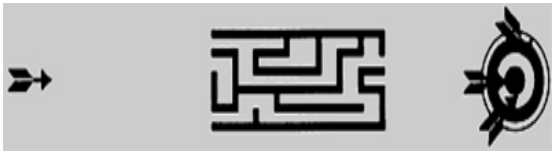


Introduction - Sensitivity-Uncertainty Methodology for NCS

Application



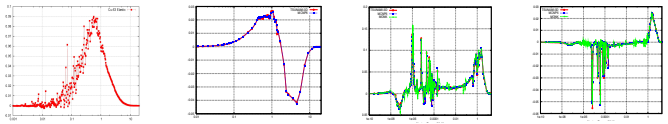
Nuclear Cross-section Data



MCNP6 or Scale/Tsunami

Monte Carlo

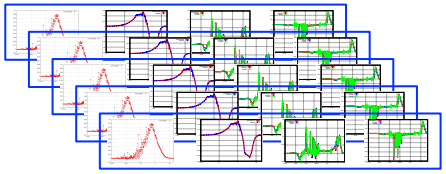
Criticality Calculation



Application

Sensitivity Profile

Nuclear Cross-section Covariance Data



Catalog of sensitivity profiles for 1000s of experiments

SU-based Analysis

Whisper or Tsurfer

Pattern matching – application sensitivity profile vs catalog

Select similar experiments

Statistical analysis to determine bias & uncertainty & extra margin

USL

Upper Subcritical Limit

for criticality safety analysis

Introduction – Workshop 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_{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

Los Alamos
NATIONAL LABORATORY
EST. 1943

NCSP
NUCLEAR CRITICALITY SAFETY PROGRAM

Background

- **Why do we care about Validation?**
  - ANSI/ANS-8.24 Foreword: ***"…the industry need to optimize operations and reduce unnecessary conservatism has increased. Thus, the scrutiny and importance placed on validation has increased in recent years."***
  - **Ensure what NCS determines to be subcritical is actually subcritical**
    - 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 Borosilcate-Glass Raschig Rings as a Neutron Absorber in Solutions of Fissile Material
- ANSI/ANS 8.7-1998(R2012), Nuclear Criticality Safety in the Storage of Fissile Materials
- ANSI/ANS-8.10-2005, Criteria for Nuclear Criticality Safety Controls in Operations with Shielding and Confinement
- ANSI/ANS 8.14-2004, Use of Soluble Neutron Absorbers in Nuclear Facilities Outside Reactors
- ANSI/ANS 8.17-2004, Criticality Safety Criteria for the Handling, Storage, and Transportation of LWR Fuel Outside Reactors
- ANSI/ANS-8.19-2014, Administrative Practices for Nuclear Criticality Safety
- ANSI/ANS 8.20-1991(R2005), Nuclear Criticality Safety Training
- ANSI/ANS-8.21-1995(R2001), Use of Fixed Neutron Absorbers in Nuclear Facilities Outside Reactors
- ANSI/ANS-8.23-2007, Nuclear Criticality Accident Emergency Planning and Response

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

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

- **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. [$k_{eff\ calculated} - k_{eff\ experment}$]
  - **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_{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_{eff} must be less than the Upper Subcritical Limit (USL):

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

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

[additional AoA margin may be appropriate]

Note: Bias = calculated – experiment,
For conservatism – can set positive biases to zero; only consider negative biases

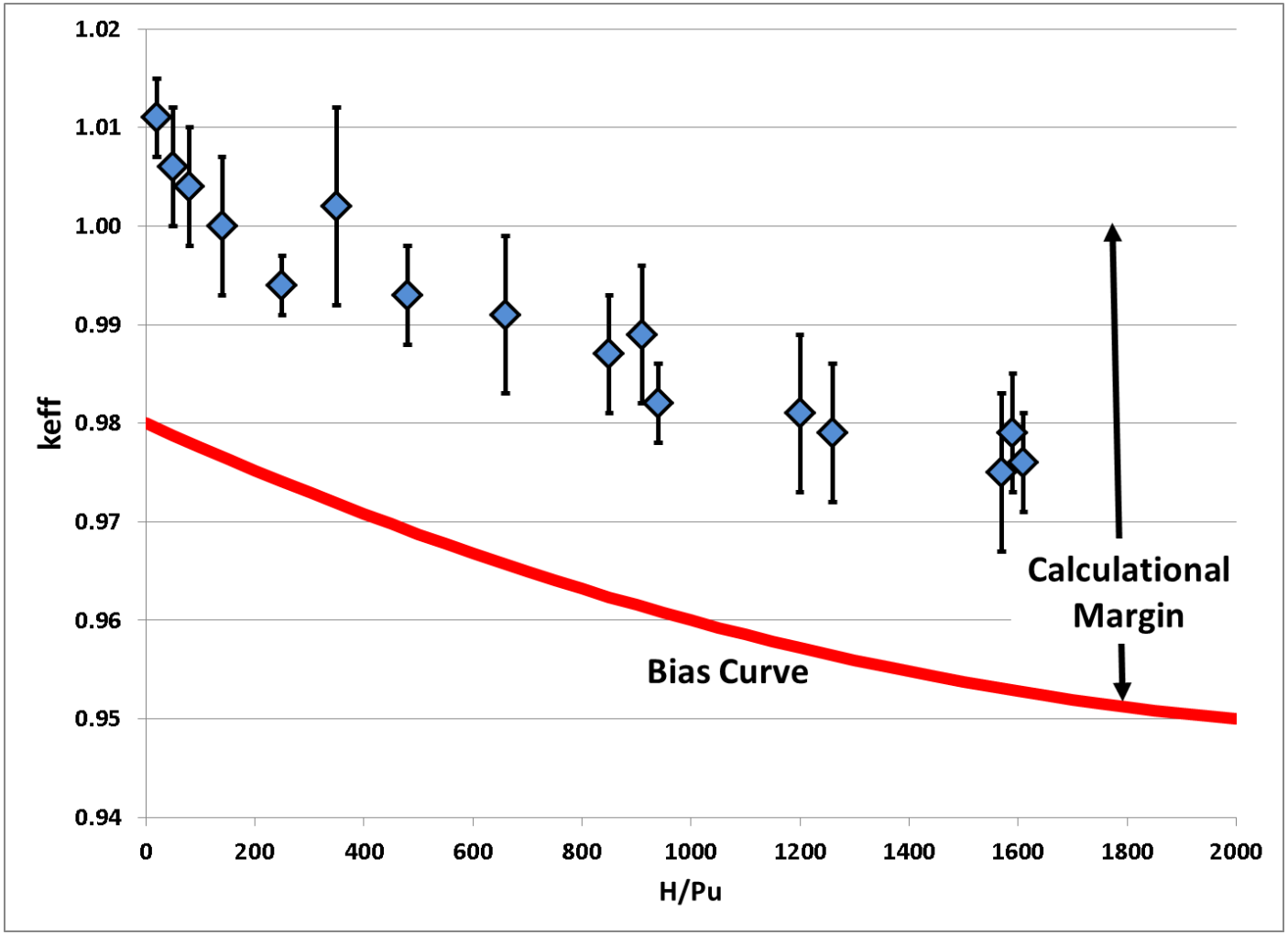
- The bias and bias uncertainty are at some confidence level, typically 95% or 99%.
  - These confidence intervals may be derived from a normal distribution, but the normality of the bias data must be justified.
  - Alternatively, the confidence intervals can be set using non-parametric methods.

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.**

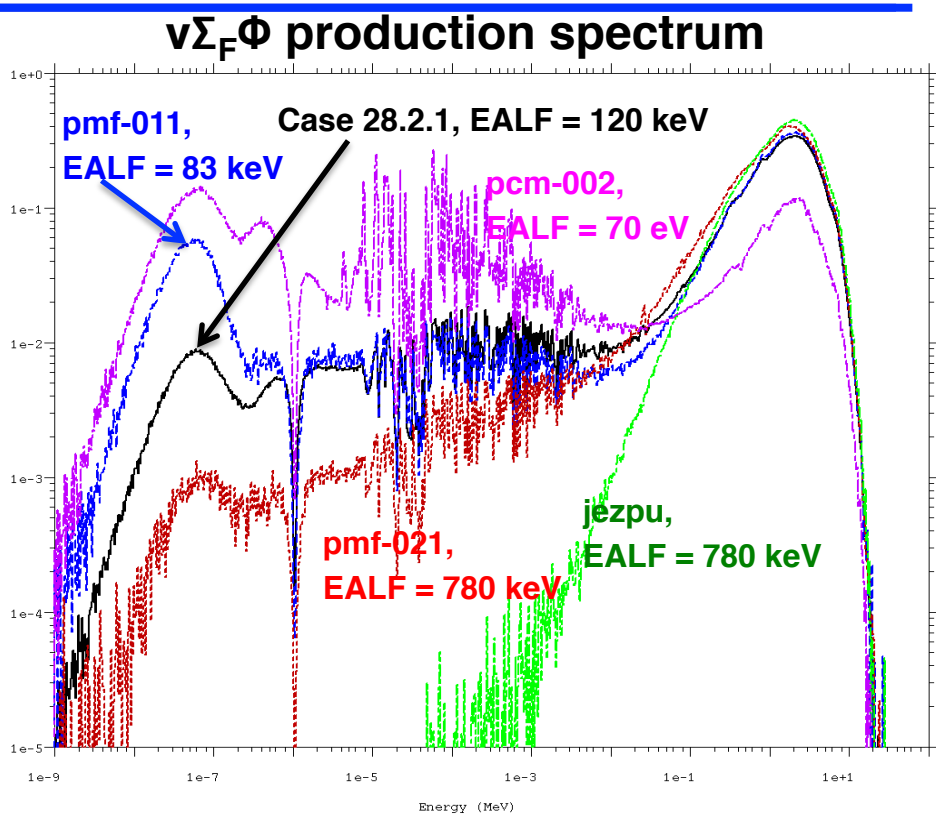
Selection of Benchmarks

- **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
- 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)



- 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 $k_{eff,normalized}$
- 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_{eff}.**

- **Techniques:**

  - **Manual model variation**
    - Change material densities or temperatures
    - Change dimensions
    - Used to justify simplifications and to quantify the impact of manufacturing tolerances and uncertainties
    - Used to support margin adopted for validation weaknesses

  - **Perturbation theory methods (Whisper and TSUNAMI)**
    - These systems use perturbation theory to provide nuclide, reaction, energy, and location dependent sensitivity data
    - Typically in units of $(\Delta k/k)/(\Delta\sigma/\sigma)$, or the fractional change in k_{eff} due to a fractional change in the nuclear data value.
    - Sensitivity analysis improves understanding of what is important for k_{eff} determination

S/U Analysis

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

S/U Analysis

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

Documentation and Independent Technical Review

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

- **Independent Technical Review:**
  - review benchmark applicability
  - Input files and output files
  - Methodology for determining bias, bias uncertainty, margins
  - Concurrence with validation applicability

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 >> kT)** $\Sigma_S(E' \rightarrow E) = \dfrac{\Sigma_S(E')}{E'}$

- **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**

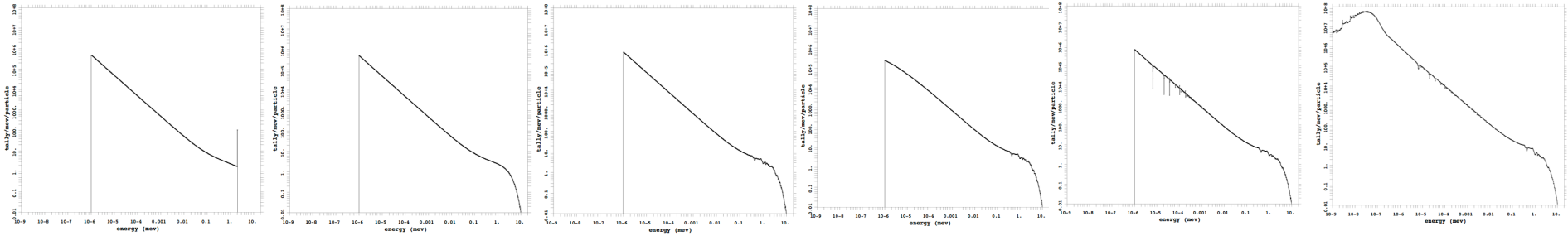
- **Consider slowing down flux in hydrogen, $E<E_S$**

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

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

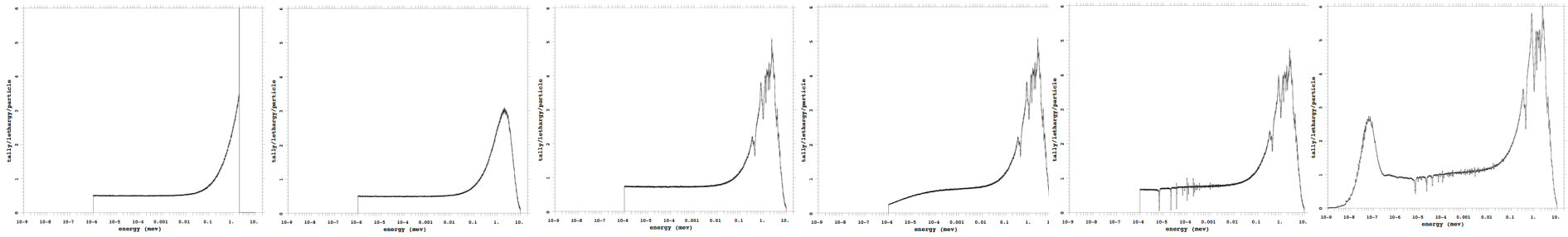
Flux Spectra for Neutron Slowing Down & Criticality

loglog plots of ϕ(E) vs E



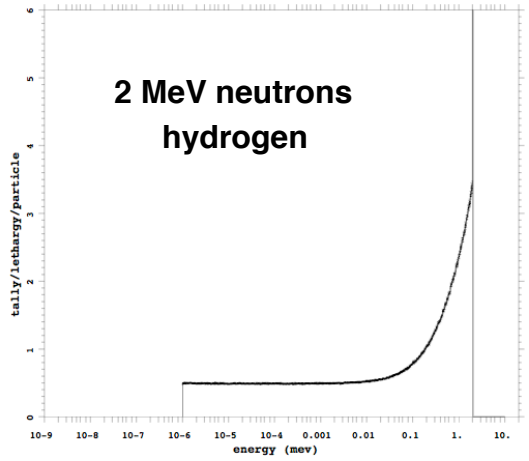
2 MeV neutrons hydrogen	fission neutrons hydrogen	fission neutrons water	fission neutrons water + B^{10}	fission neutrons water + U^{238}	Fuel Pin Unit Cell

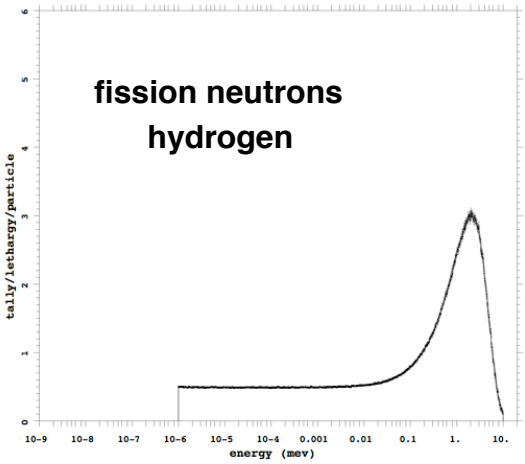
loglin plots of ϕ(u) vs u



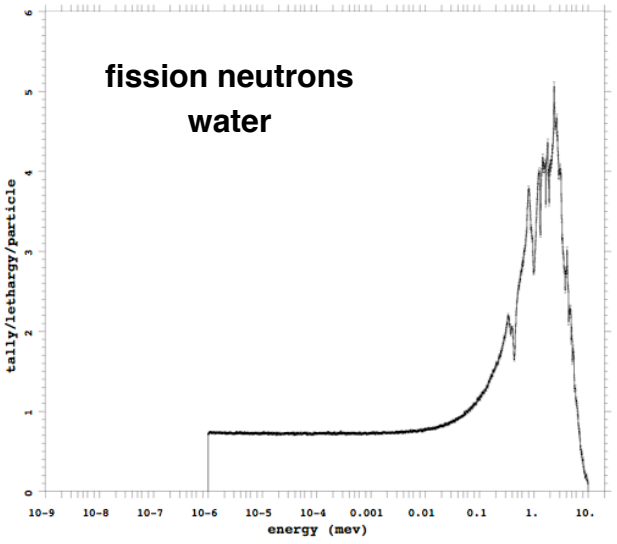
2 MeV neutrons hydrogen	fission neutrons hydrogen	fission neutrons water	fission neutrons water + B^{10}	fission neutrons water + U^{238}	Fuel Pin Unit Cell

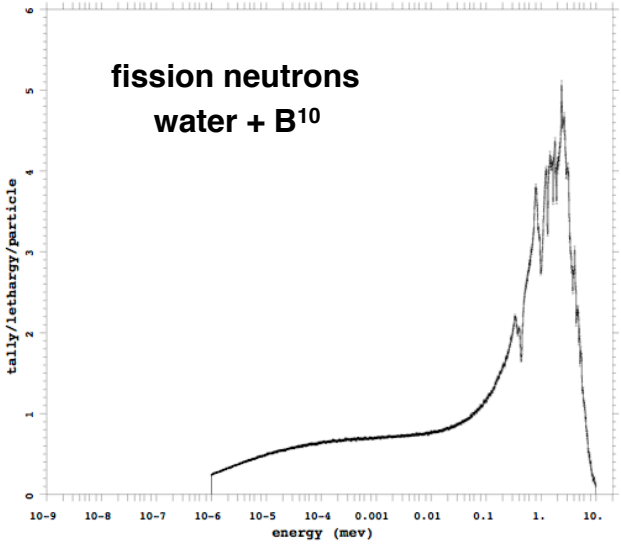
Flux Spectra for Neutron Slowing Down

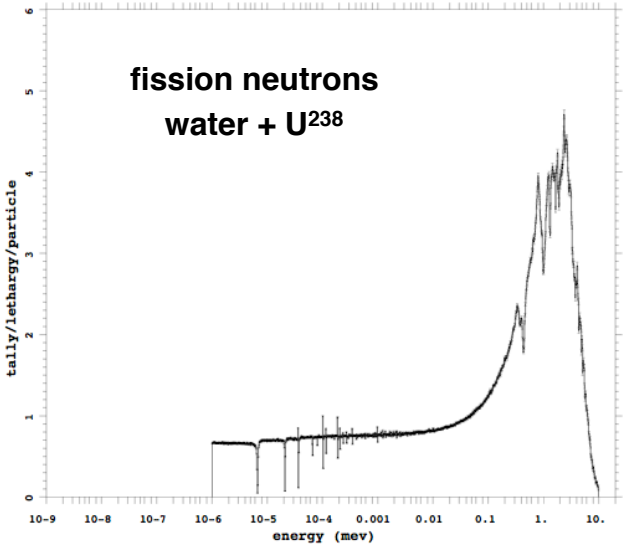




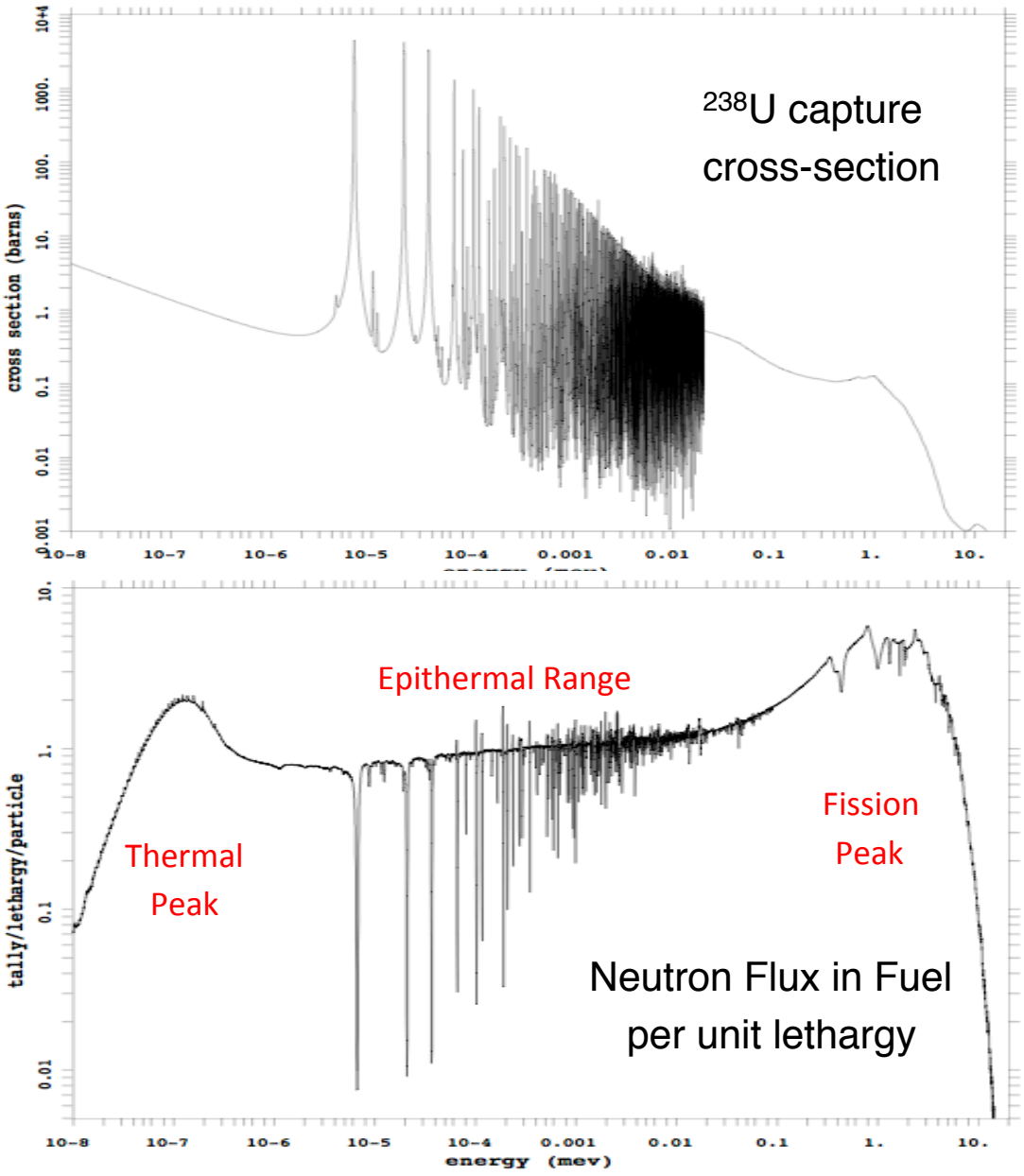
loglin plots of ϕ(u) vs u







UO_2 Fuel Pin



UO_2 Fuel Pin

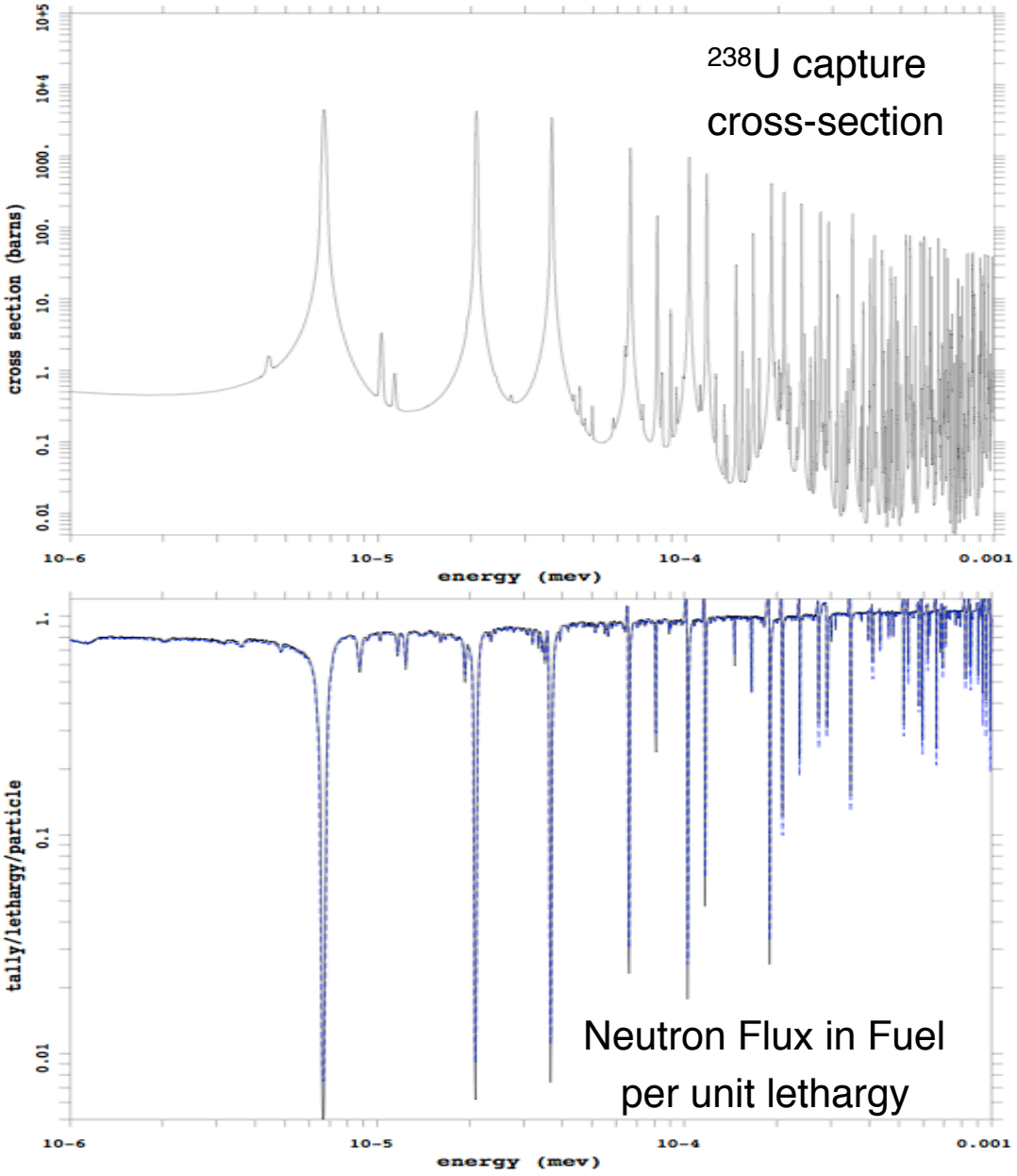
3.1% Enriched

293.6 ^{o}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 ^{238}U capture in epithermal energy range during slowing down**

UO_2 Fuel Pin



UO_2 Fuel Pin

3.1% Enriched

293.6 °K

Detail for
1 eV – 1 KeV

1/3 of neutron losses are due to ^{238}U capture at epithermal energies during slowing down

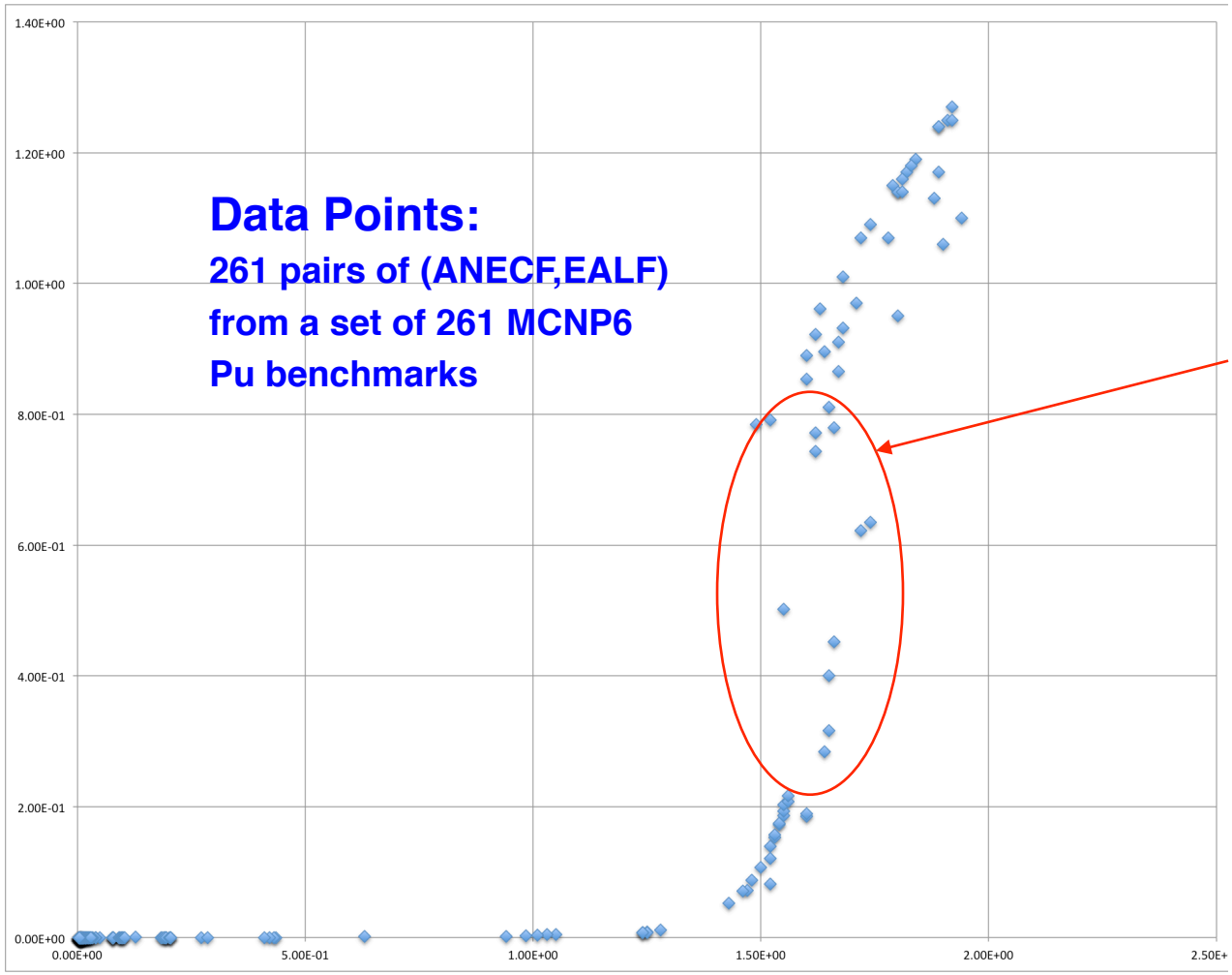
Characterizing the Neutron Spectrum

- 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 ^{239} or H/U^{235} ratios, for solutions
  - Fraction of fissions caused by fast (E > 100 keV), intermediate (1 eV < E < 100 keV), and thermal (E < 1 eV) neutrons
  - $^{238}U(n,f)/^{235}U(n,f)$, $^{237}Np(n,f)/^{235}U(n,f)$, 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

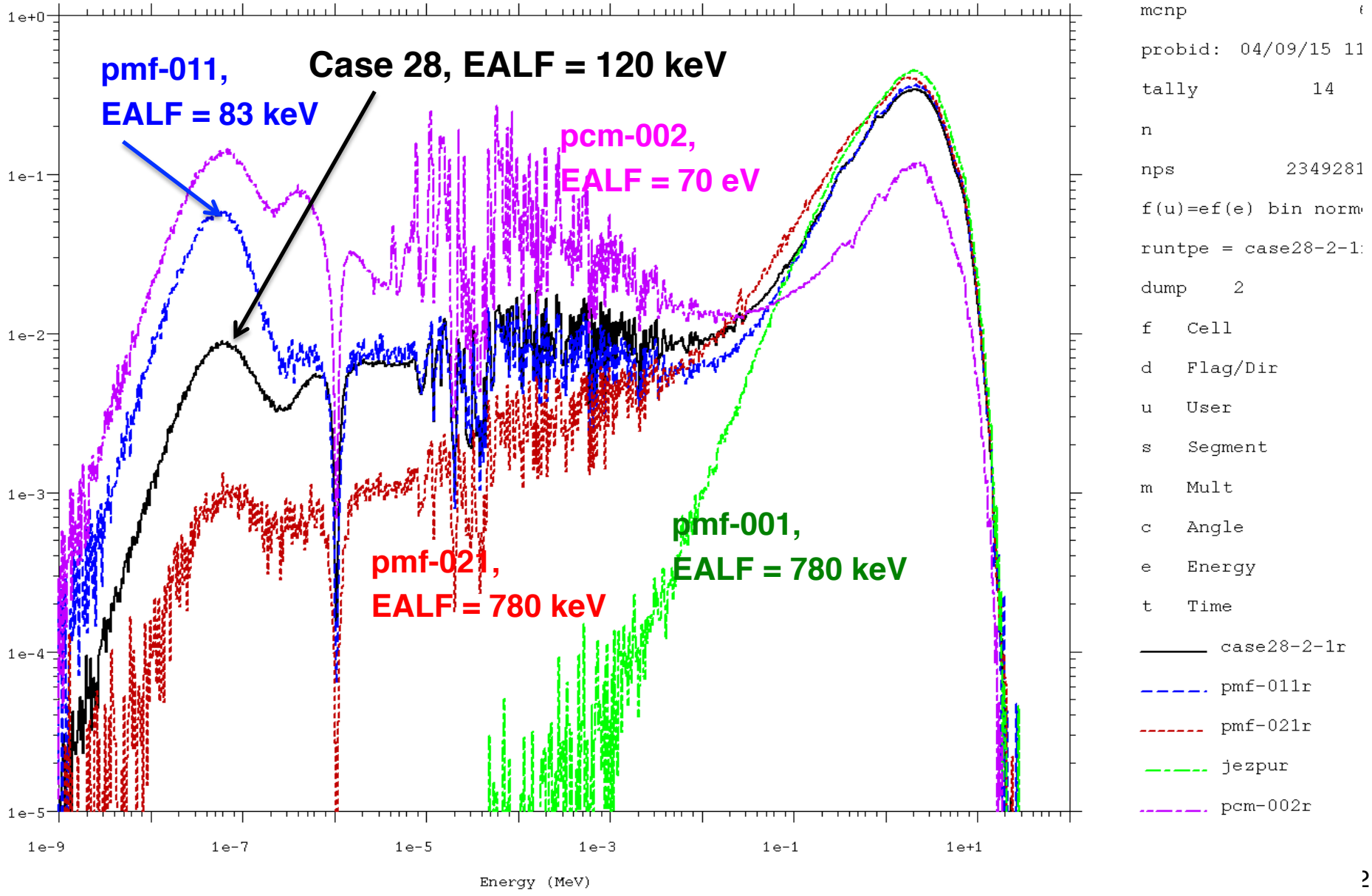
EALF vs ANECF

ANECF = average neutron energy causing fission

EALF = energy of the average neutron lethargy causing fission



Pu Systems – $\nu\Sigma_F\Phi$ 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_{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 Keff 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:

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

- 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:**

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

- **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

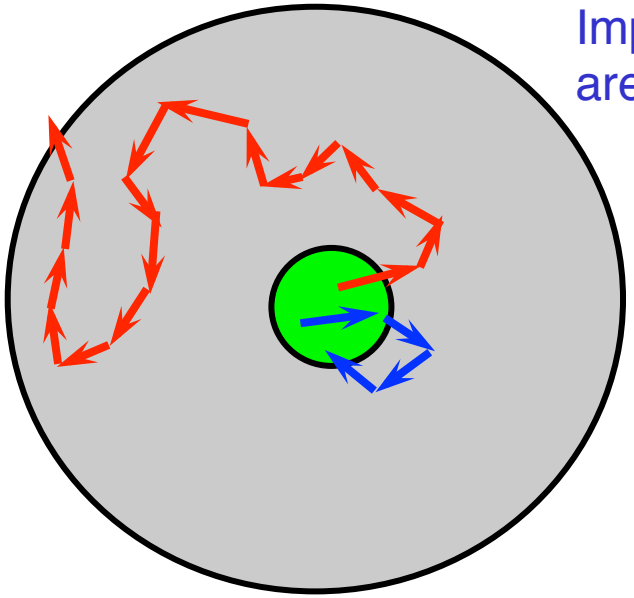
$$\Lambda_{rem} = \frac{\langle 1, {}^{1}\!/\!_{v}\,\psi \rangle}{\langle 1, F\psi \rangle}$$

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

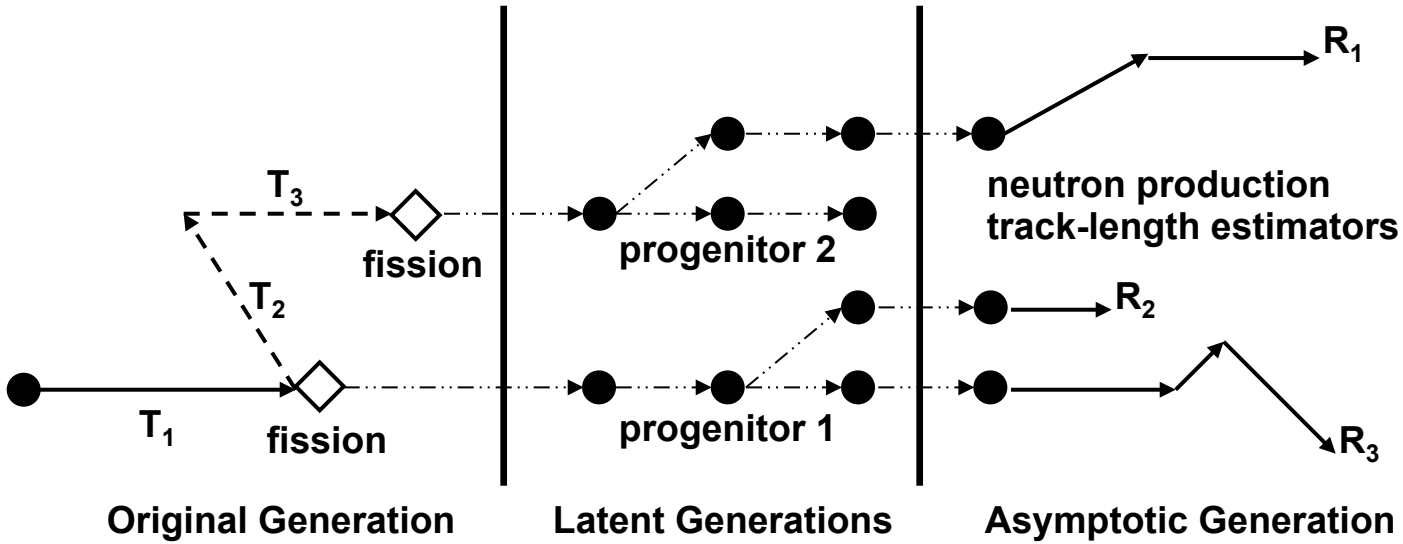


Important neutrons are often short-lived

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

MCNP Implementation

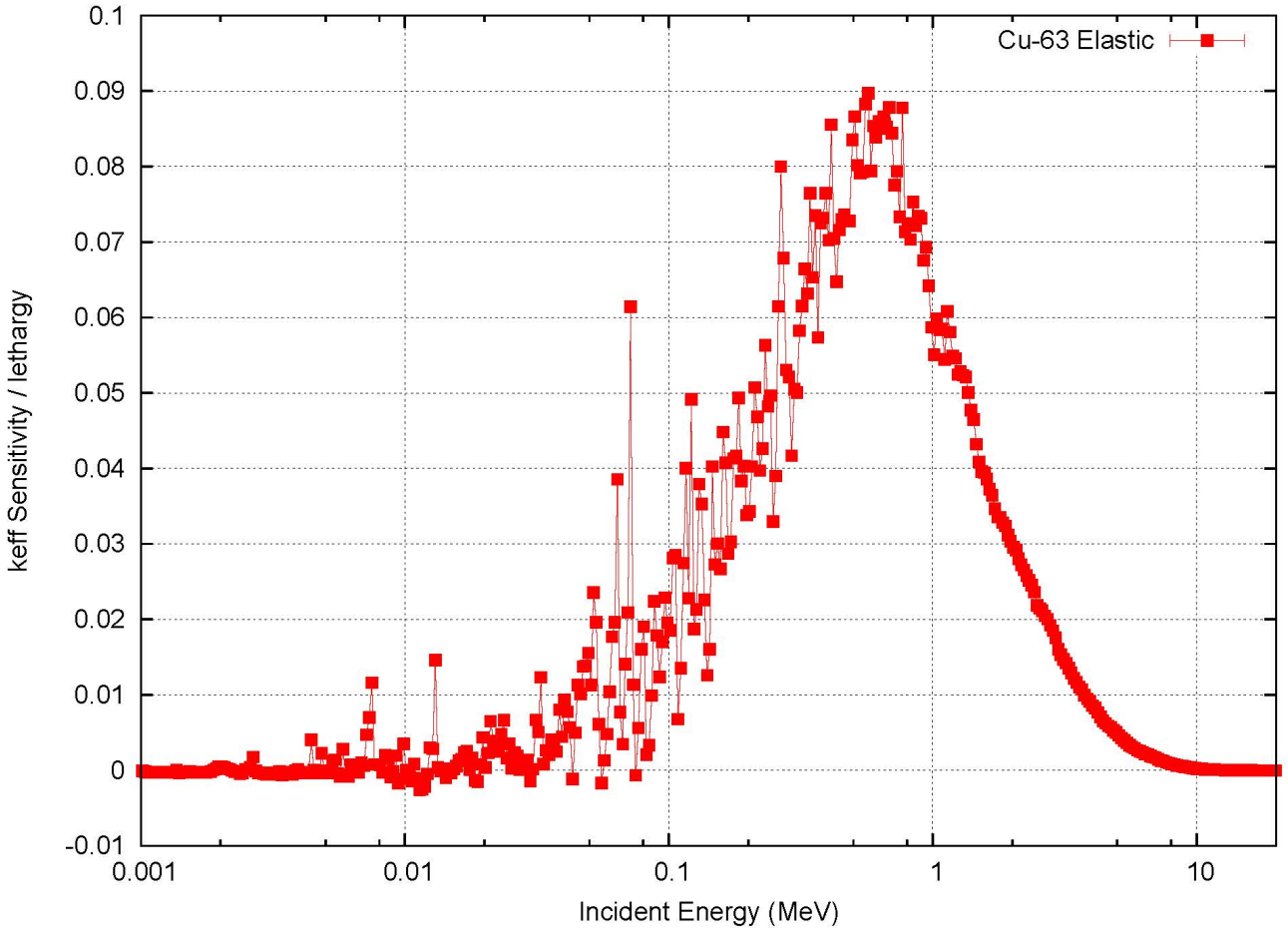
- **MCNP performs adjoint-weighting of tallies using a technique called the iterated fission probability**
- **MCNP breaks active cycles into consecutive *blocks*:**
  - **Tally contributions collected in first generation, progenitor neutrons tagged and linked with tally contributions.**
  - **All subsequent progeny within the block remember their progenitor.**
  - **After *N* cycles, the population of progeny from each progenitor is measured. This is multiplied by the previously recorded tally contributions to form a tally score.**



Example Sensitivity Coefficient Profile

Cu-63: Elastic Scattering Sensitivity

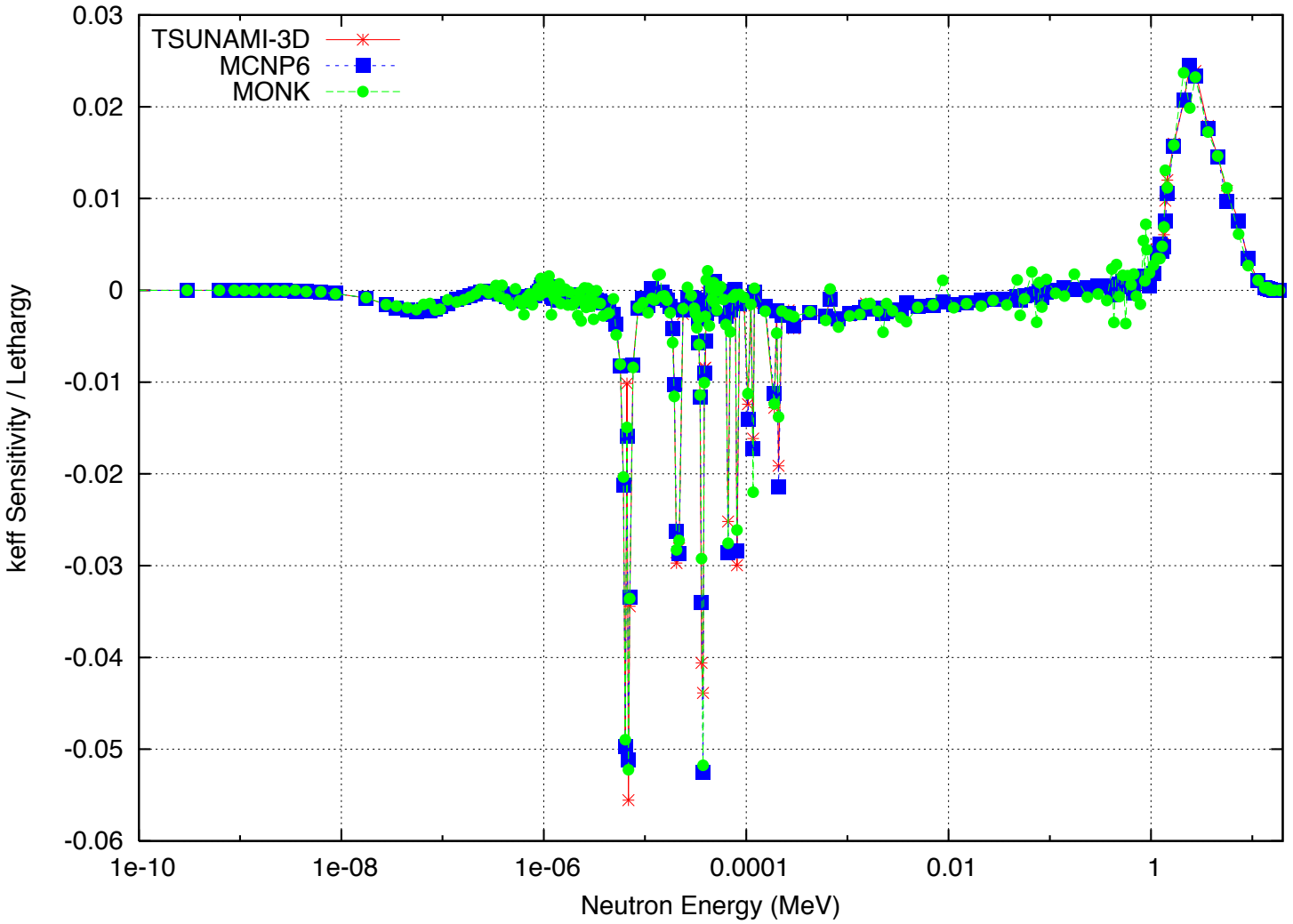
Copper-Reflected Zeus experiment:



Example Sensitivity Coefficient Profile

U-238: total cross-section sensitivity

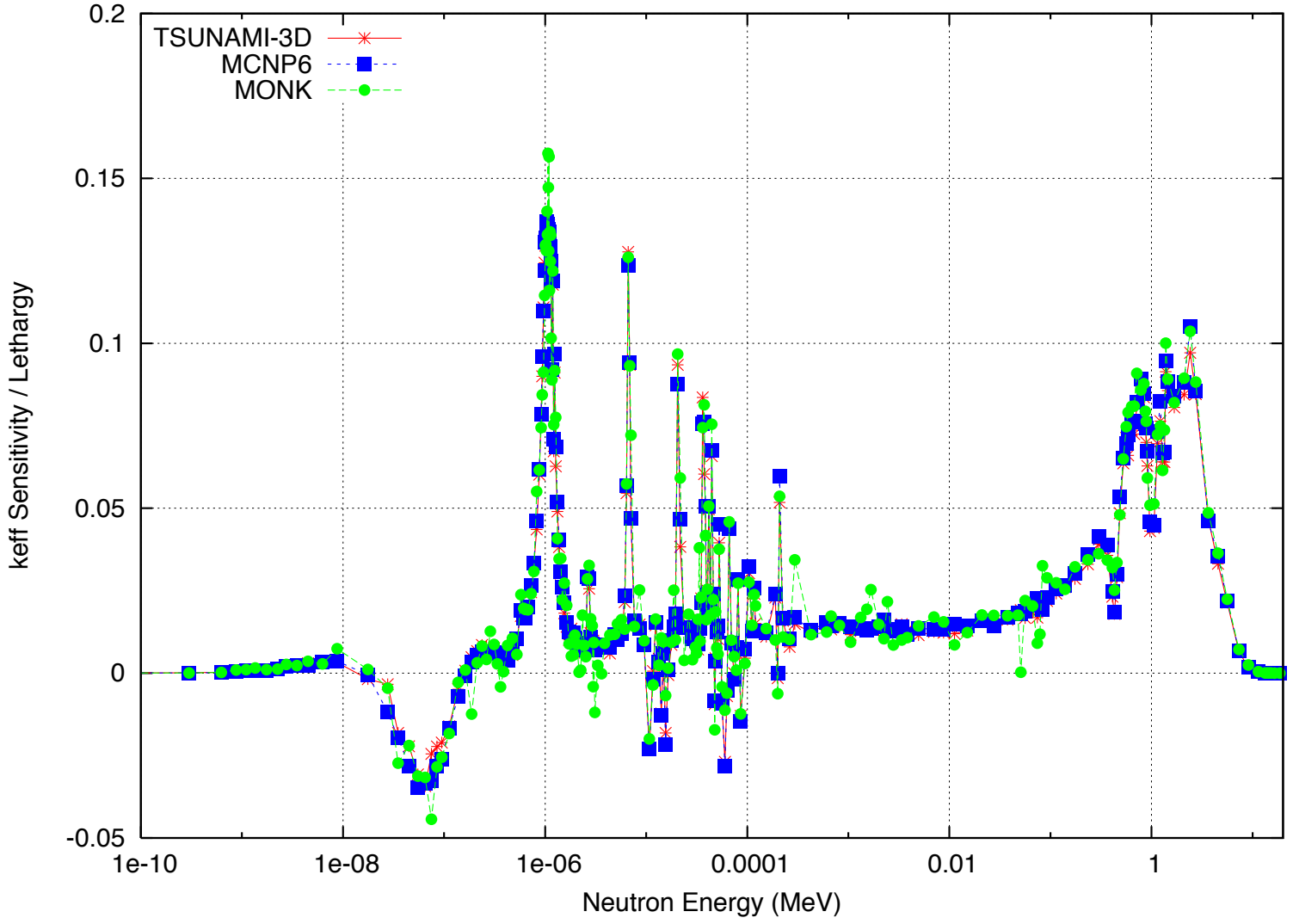
OECD/NEA UACSA Benchmark Phase III.1



Example Sensitivity Coefficient Profile

H-1: elastic scattering cross-section sensitivity

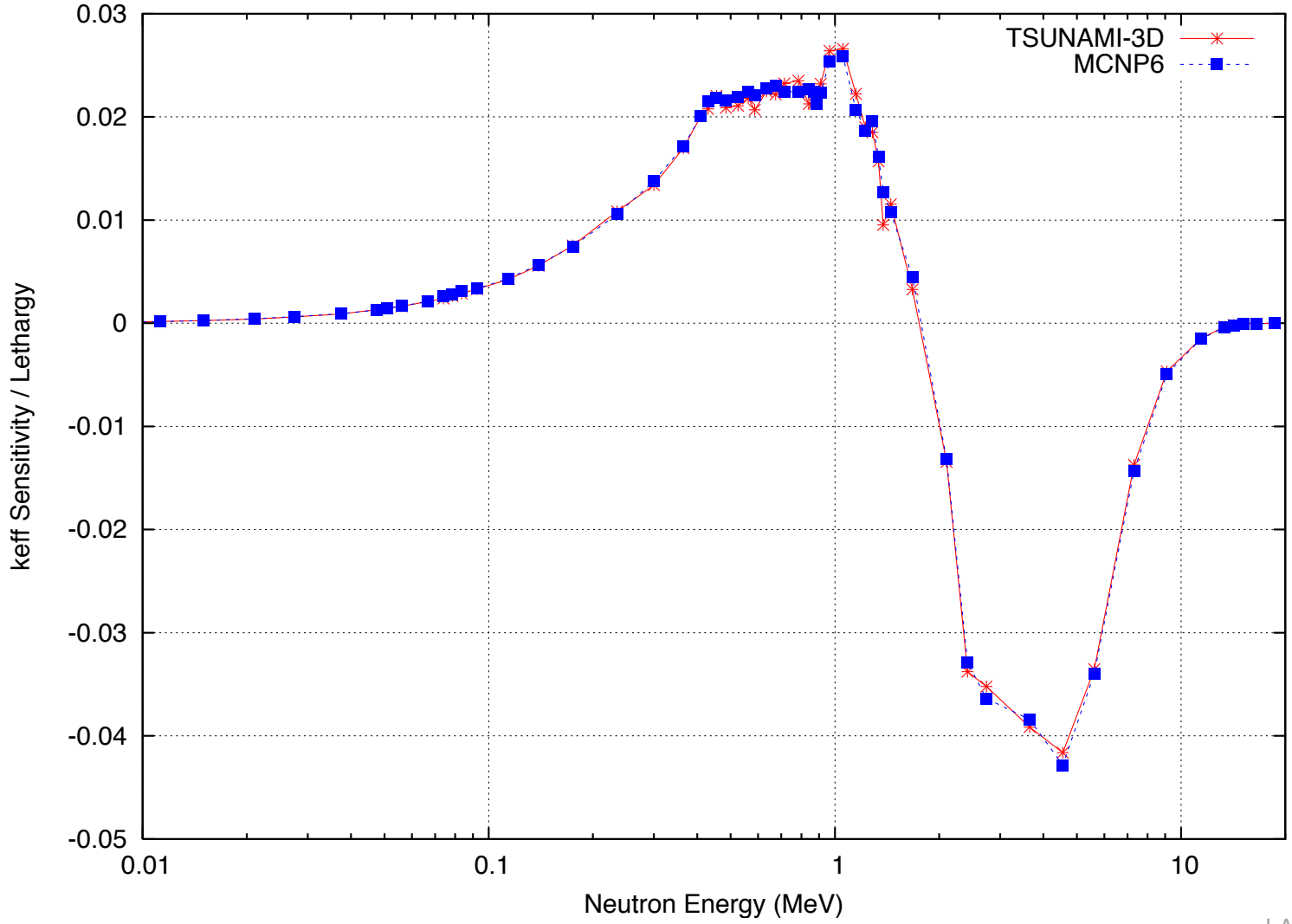
OECD/NEA UACSA Benchmark Phase III.1



Example Sensitivity Coefficient Profile

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

OECD/NEA UACSA Benchmark Phase III.1



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:**

  KSEN*j* 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**

```
ksen1    xs    rxn= -2
```

- **U-238 elastic and inelastic scattering sensitivities**

```
ksen2    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**

```
ksen3    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:**

KSEN*j* 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

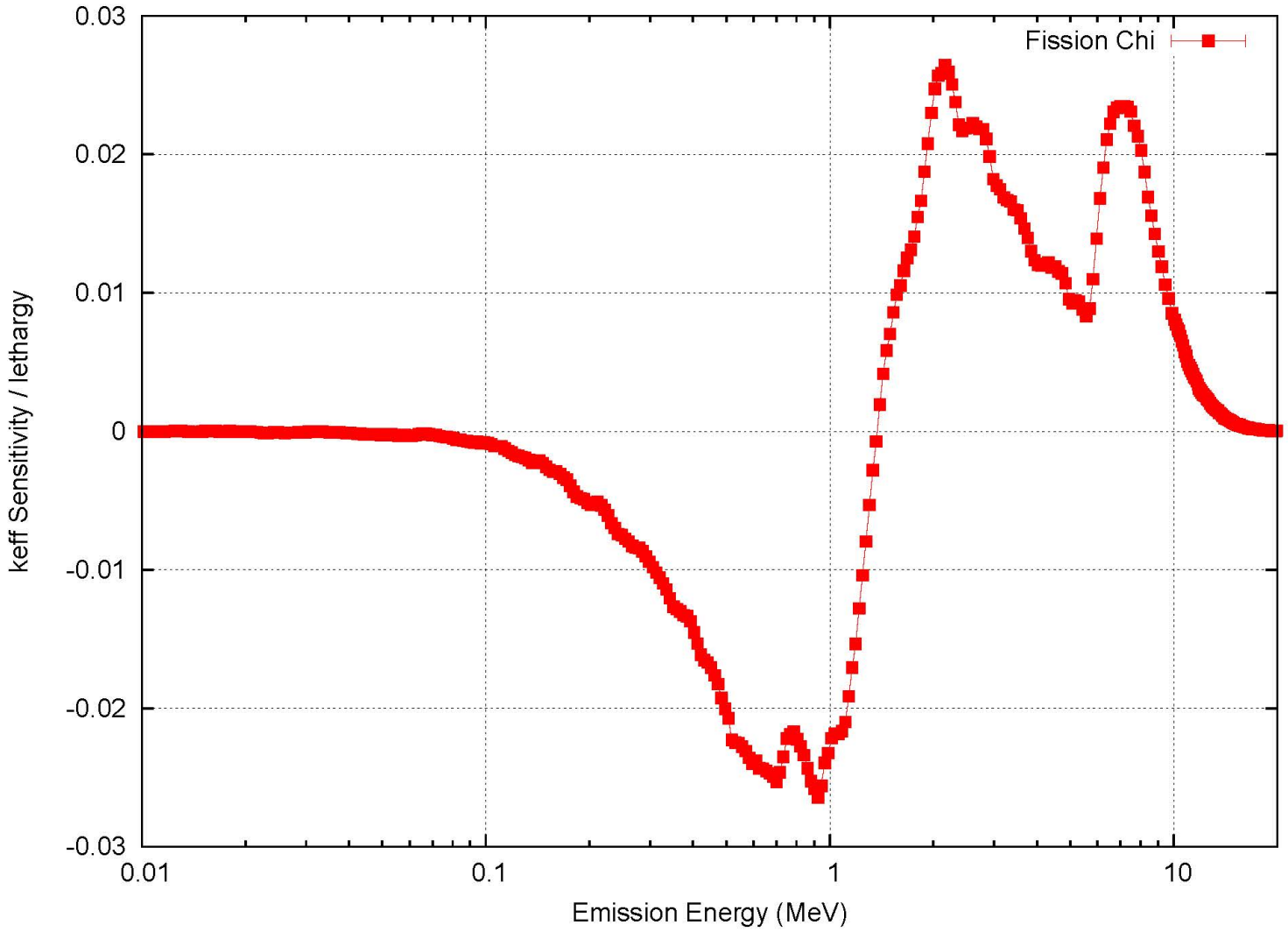
- **KSEN card of Pu-239 chi sensitivity:**

```
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

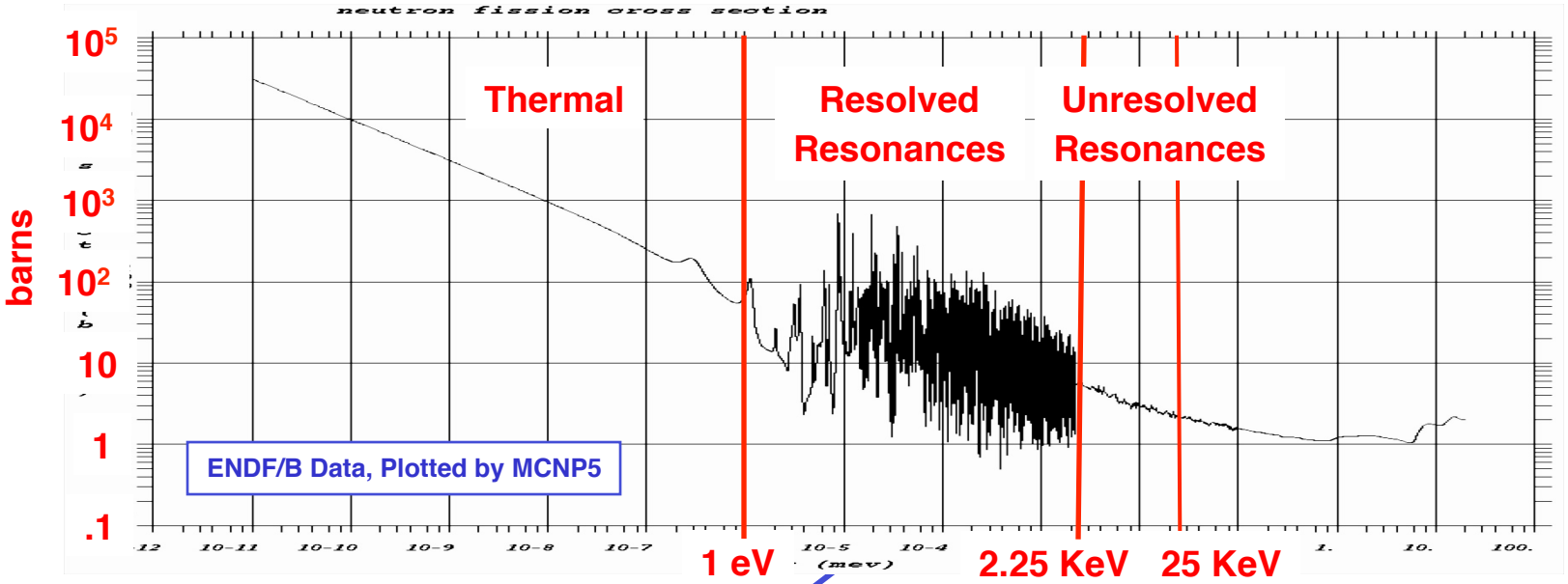
Nuclear Data

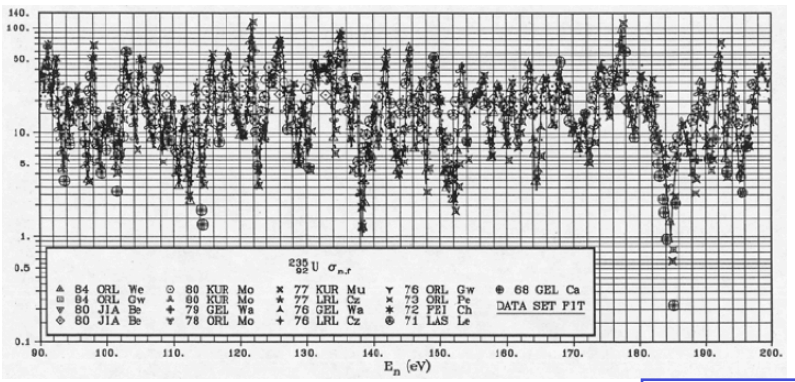
- **"Nuclear" data involves interactions of incident particles with the nucleus.**
  - **Data libraries include cross-section and scattering data with interpolation laws, various parameters, etc., derived from both experiments and theory**
  - **Typically there are "ladders" of (E_J,σ_J) pairs, but many other formats are also used.**

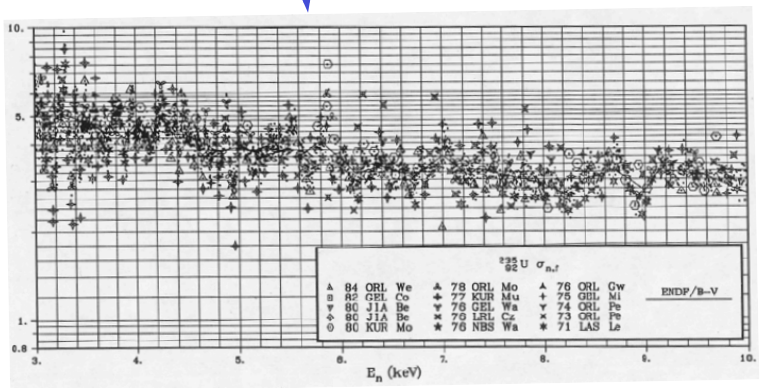
- **Results obtained from a calculation depend upon both the code and the nuclear data it employs**

- **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**

U^{235} Fission Cross-section



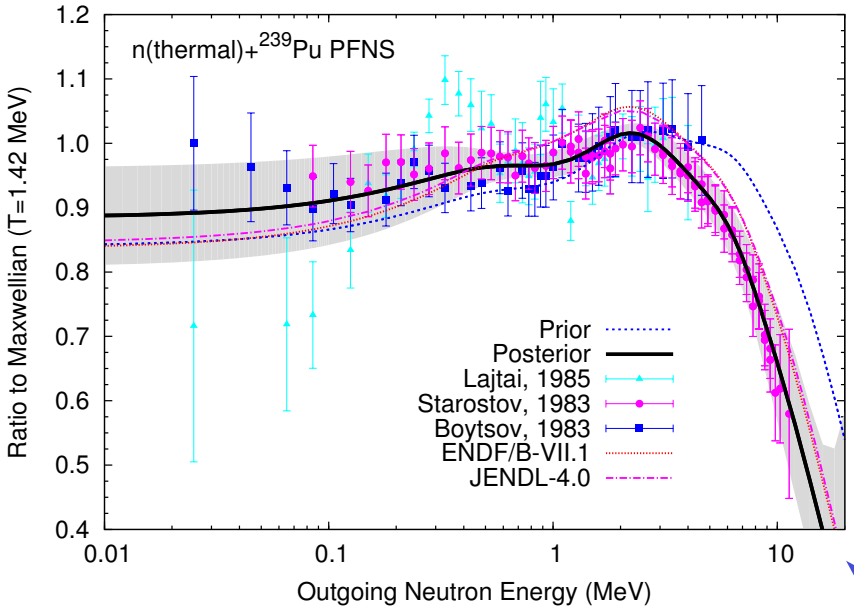


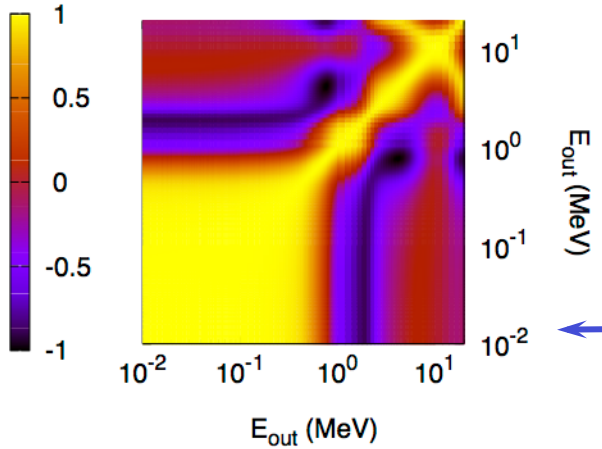


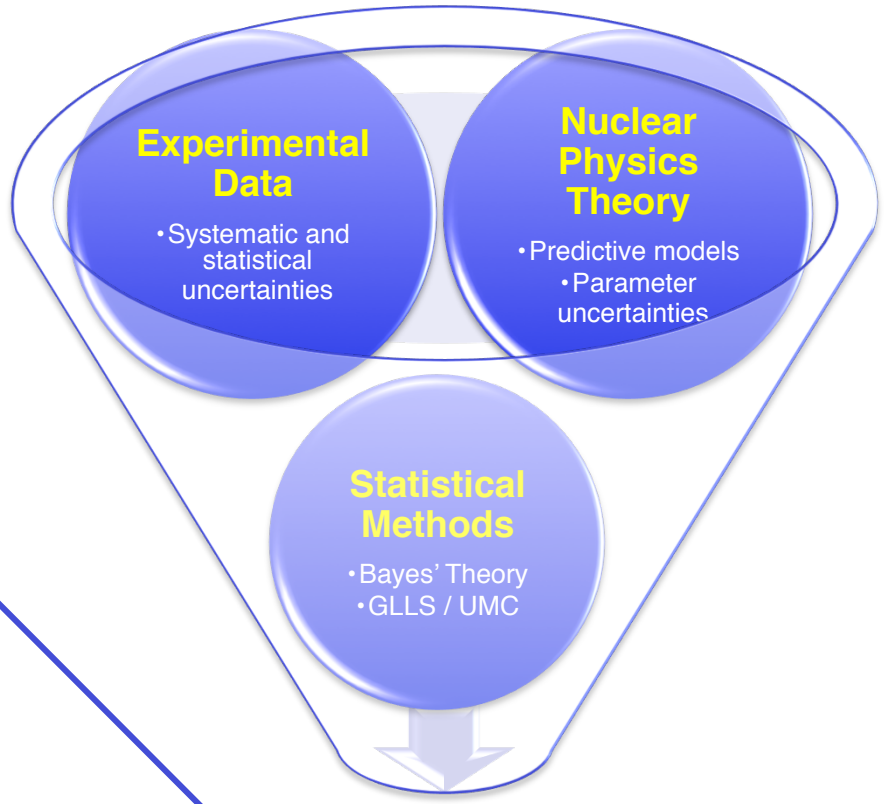
Experimental Data used by CSWEG

Pu^{239} 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,γ
103	n,p
104	n,d
105	n,t
106	n,^{3}He
107	n,α
452	ν
1018	χ

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 SCALE system

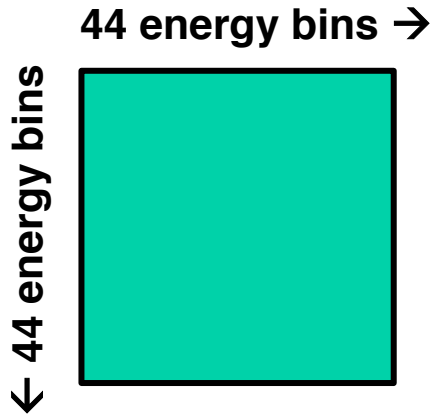
Energy bin bounds (MeV)

```
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 x G matrix, where G = number of energy groups = 44



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

Cross-section Covariance Data (4)

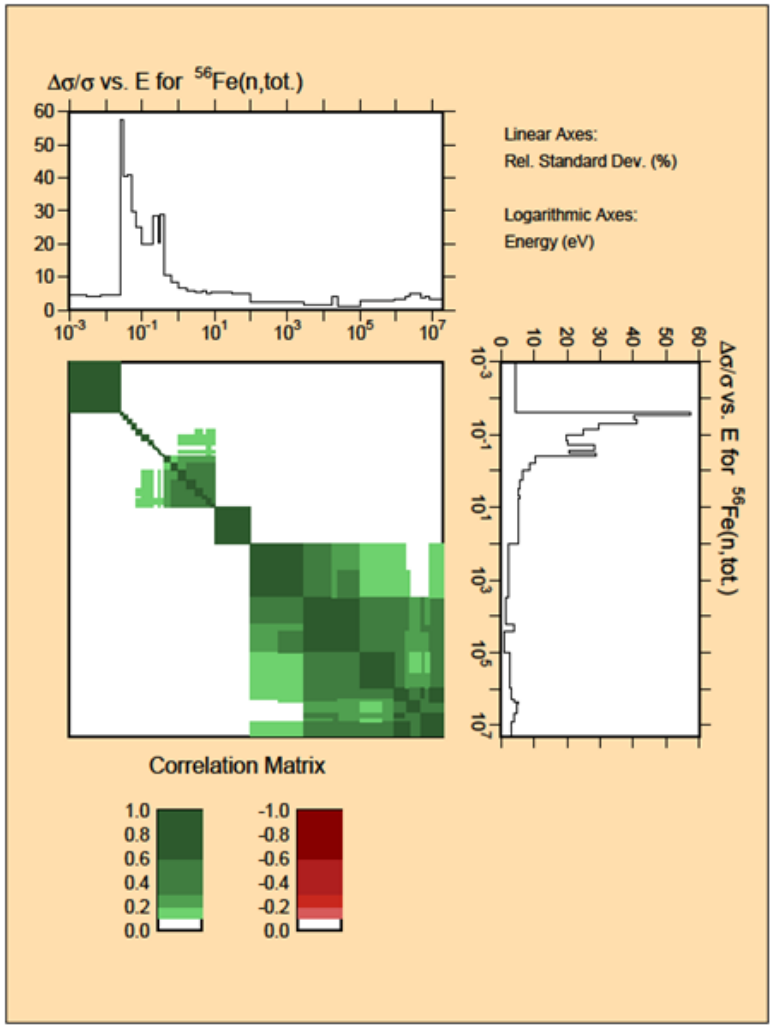


FIG. 9: A typical NJOY-generated plot of ENDF/B-VII.0 data downloaded from the National Nuclear Data Center, BNL, USA.

Cross-section Covariance Data (5)

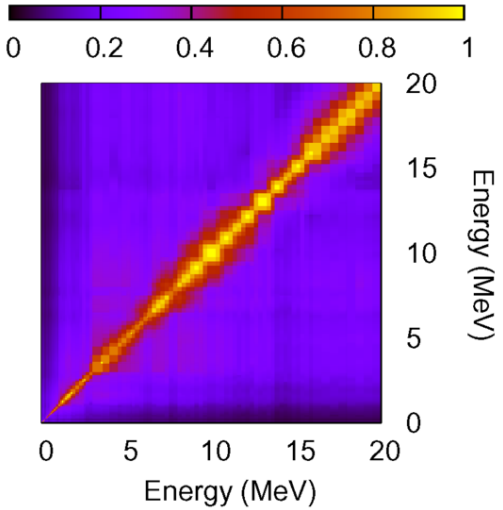


FIG. 3: Correlation matrix for the neutron-induced fission cross section on ^{235}U. It was evaluated by Pronyaev *et al.* as part of the cross section standards evaluation [19].

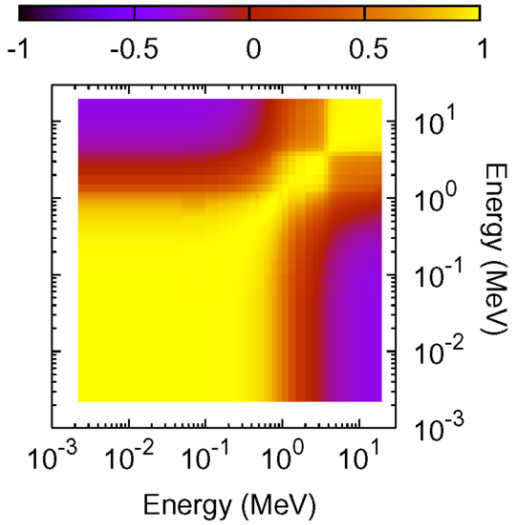


FIG. 6: Correlation matrix for the capture cross section of n+^{235}U.

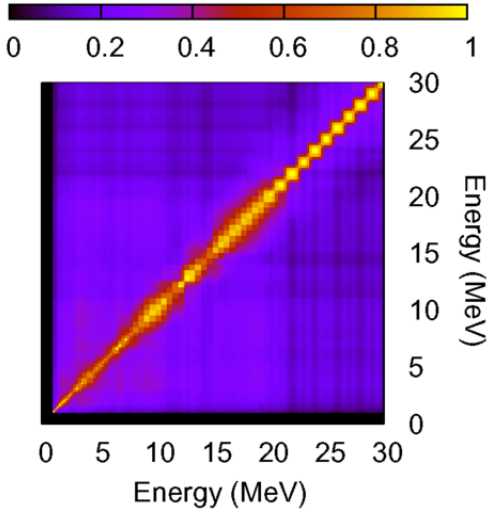


FIG. 13: ^{238}U fission cross-section correlation matrix.

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)

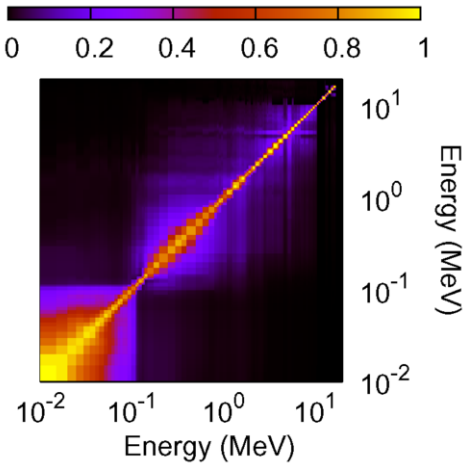


FIG. 25: Correlation matrix evaluated for the ^{238}Pu (n,fission) cross section.

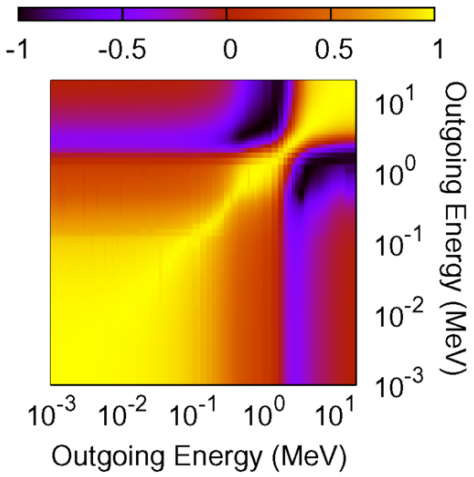


FIG. 40: Correlation matrix evaluated for the n(0.5 MeV)+^{239}Pu prompt fission neutron spectrum.

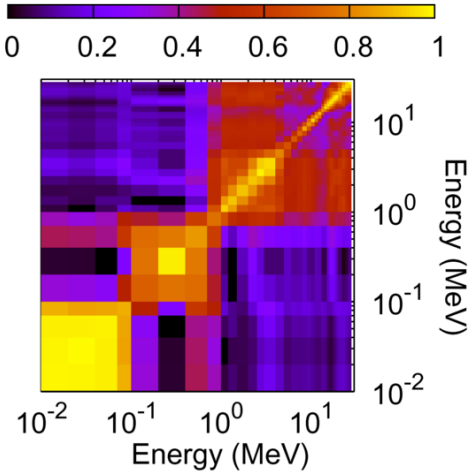


FIG. 43: Evaluated correlation matrix for the neutron-induced fission cross section of ^{240}Pu in the fast energy range.

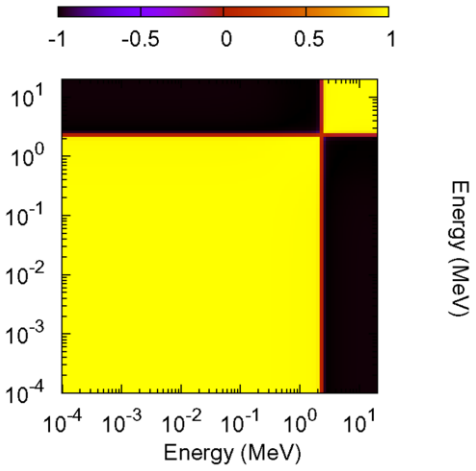


FIG. 30: Correlation matrix for the n(0.5 MeV)+^{238}Pu prompt fission neutron spectrum.

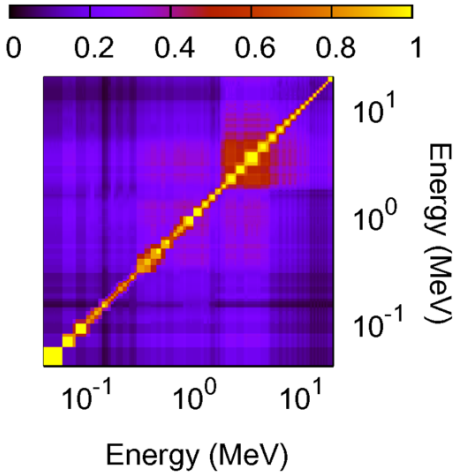


FIG. 45: ^{240}Pu (n,total) cross section correlation matrix.

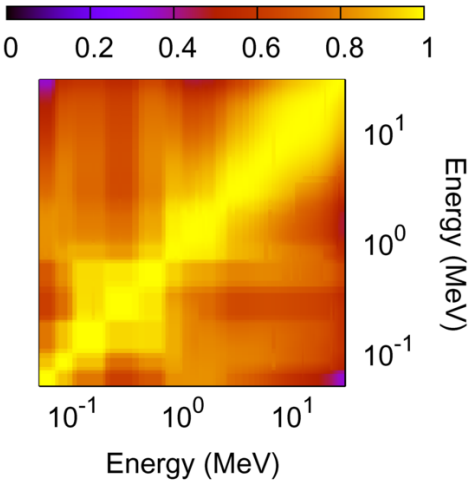


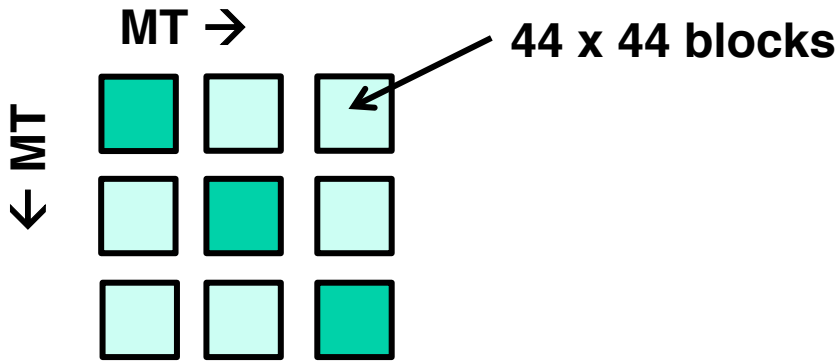
FIG. 47: Correlation matrix for the n+^{240}Pu capture cross section. Large off-diagonal elements are due mostly to model uncertainties, since no experimental data exist above 300 keV.

Cross-section Covariance Data (7)

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

  C_{xx}^{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)

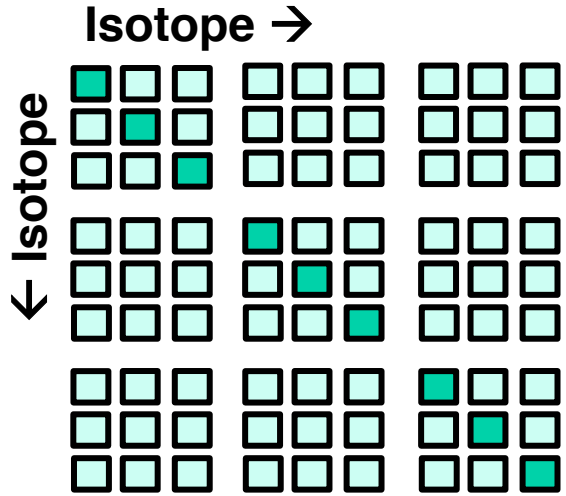


  - Each C_{xx}^{Iso} entry is produced by SCALE or NJOY based on covariance data from the ENDF/B libraries (with some adjustments if needed)
  - The C_{xx} data is universal, independent of benchmark or application problem

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**

C_{xx} =



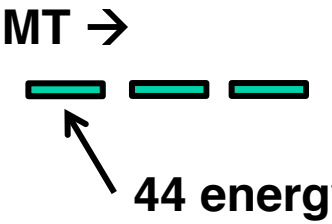
  - Each diagonal element of C_{xx} is the variance of the cross-section for a particular isotope, MT, & energy bin
  - Off-diagonal elements of C_{xx} are the shared variance between pairs of Iso-MT-E & Iso'-MT'-E'
  - Very sparse (lots of zeros), block-structured matrix
    (Off-diagonal I-I' blocks 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 Keff to a particular reaction type & energy bin is:**

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

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

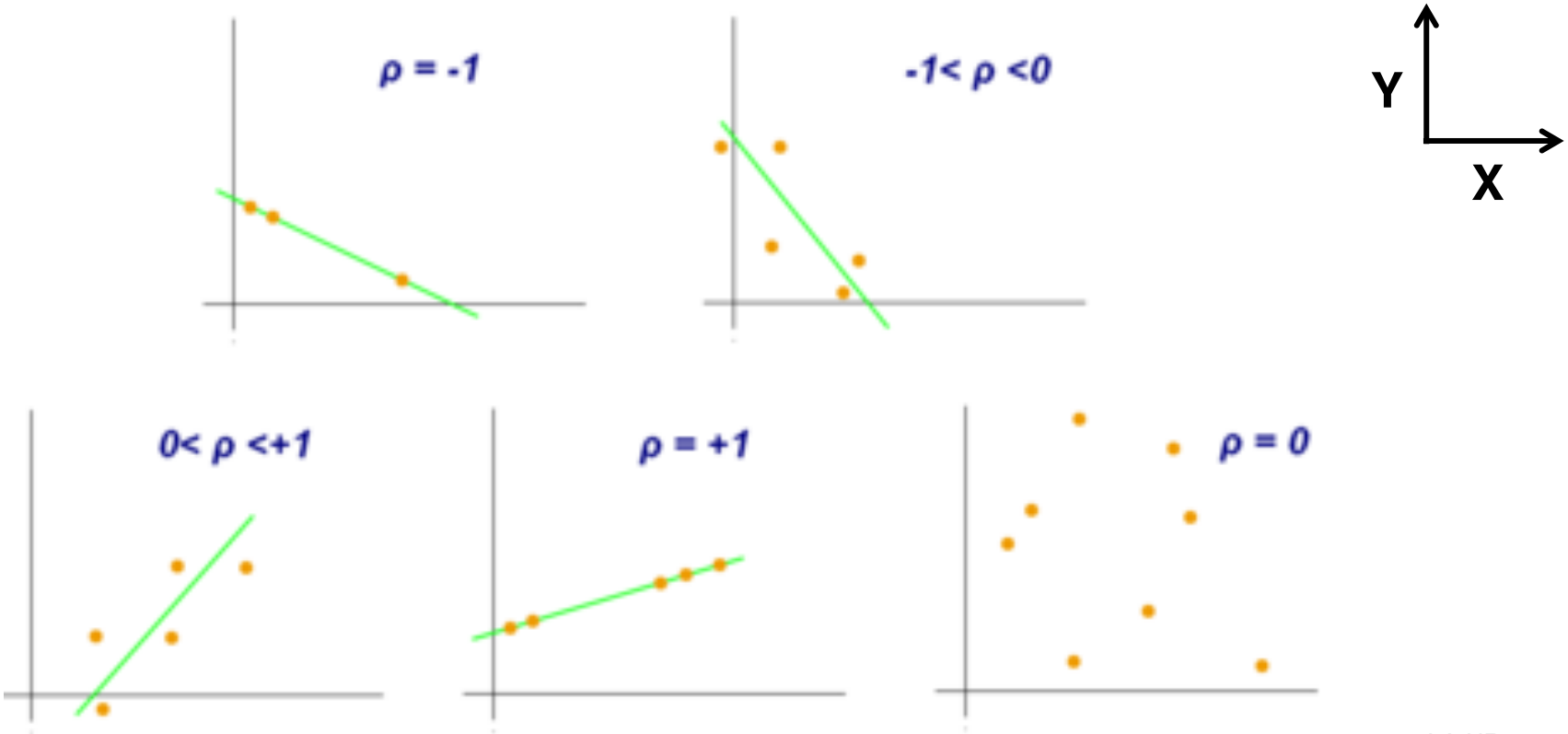


- **For a particular application problem, A, the sensitivity profiles for all isotopes are combined into one sensitivity vector S_A**

Correlation Coefficients

Correlation Coefficient (1)

- **Correlation coefficient**
  - **Pearson product-moment correlation coefficient, *r* or *ρ***
  - **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, ρ**
  - **Distribution of X, with mean μ_x, standard deviation σ_x**
  - **Distribution of Y, with mean μ_y, standard deviation σ_y**

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

$$\mu_X = E(X) \qquad \sigma_X^2 = E[(X-E(X))^2] = E(X^2)-E(X)^2$$

$$\mu_Y = E(Y) \qquad \sigma_Y^2 = E[(Y-E(Y))^2] = E(Y^2)-E(Y)^2$$

- **Sample correlation coefficient, r**
  - **Dataset for X: { x_1, x_2, ..., x_n }, mean x-bar, std dev s_x**
  - **Dataset for Y: { y_1, y_2, ..., y_n } mean y-bar, std dev s_y**

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

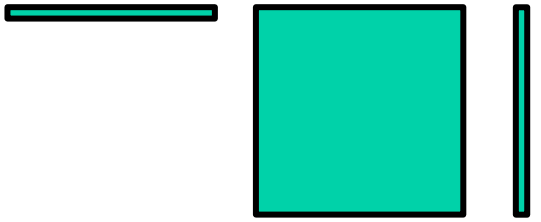
Variance in Keff & Correlation Between Problems

- **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**

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

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

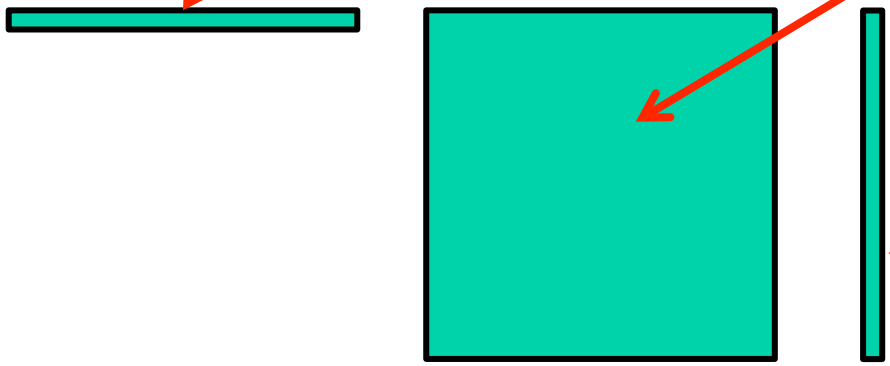
Problem-dependent sensitivity vector, S.

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

Size = G x MT x NI

Nuclear Data Covariances

Size= (G x MT x NI)^2



= scalar

S^T

Error Propagation (1)

- **Define a linear relationship**

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

- **Determine expected (mean) value of y**

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

- **Determine covariance matrix of y**

$$\begin{aligned}
\mathbf{C}_{\mathbf{y}} &= \mathrm{cov}(\mathbf{y},\mathbf{y}) = E[(\mathbf{y} - \mu_{\mathbf{y}})(\mathbf{y} - \mu_{\mathbf{y}})^T] \\
&= E[(\mathbf{A}\mathbf{x} + \mathbf{b} - \mathbf{A}\mu_{\mathbf{x}} - \mathbf{b})(\mathbf{A}\mathbf{x} + \mathbf{b} - \mathbf{A}\mu_{\mathbf{x}} - \mathbf{b})^T] \\
&= E[(\mathbf{A}(\mathbf{x} - \mu_{\mathbf{x}}))(\mathbf{A}(\mathbf{x} - \mu_{\mathbf{x}}))^T] \\
&= E[\mathbf{A}(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{x} - \mu_{\mathbf{x}})^T \mathbf{A}^T] \\
&= \mathbf{A}\, E[(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{x} - \mu_{\mathbf{x}})^T]\mathbf{A}^T \\
&= \mathbf{A}\,\mathrm{cov}(\mathbf{x},\mathbf{x})\,\mathbf{A}^T \\
\mathbf{C}_{\mathbf{y}} &= \mathbf{A}\,\mathbf{C}_{\mathbf{x}}\,\mathbf{A}^T
\end{aligned}$$

"Sandwich" Rule!

Error Propagation (2)

- First-order Taylor series expansion of k about cross section, Σ

$$k(\Sigma_1',\Sigma_2',\ldots,\Sigma_N') \cong k(\Sigma_1^0,\Sigma_2^0,\ldots,\Sigma_N^0) + \sum_{i=1}^{N} \left.\frac{\partial k}{\partial \Sigma_i}\right|_{\Sigma_i^0} (\Sigma_i' - \Sigma_i^0)$$

- Define vectors for cross sections and sensitivity profiles

$$\vec{\Sigma}' = \begin{bmatrix} \Sigma_1' & \Sigma_2' & \cdots & \Sigma_N' \end{bmatrix} \qquad \vec{S} = \begin{bmatrix} \left.\frac{\partial k}{\partial \Sigma_1}\right|_{\Sigma_1^0} & \left.\frac{\partial k}{\partial \Sigma_2}\right|_{\Sigma_2^0} & \cdots & \left.\frac{\partial k}{\partial \Sigma_N}\right|_{\Sigma_N^0} \end{bmatrix}$$

$$\vec{\Sigma}^0 = \begin{bmatrix} \Sigma_1^0 & \Sigma_2^0 & \cdots & \Sigma_N^0 \end{bmatrix}$$

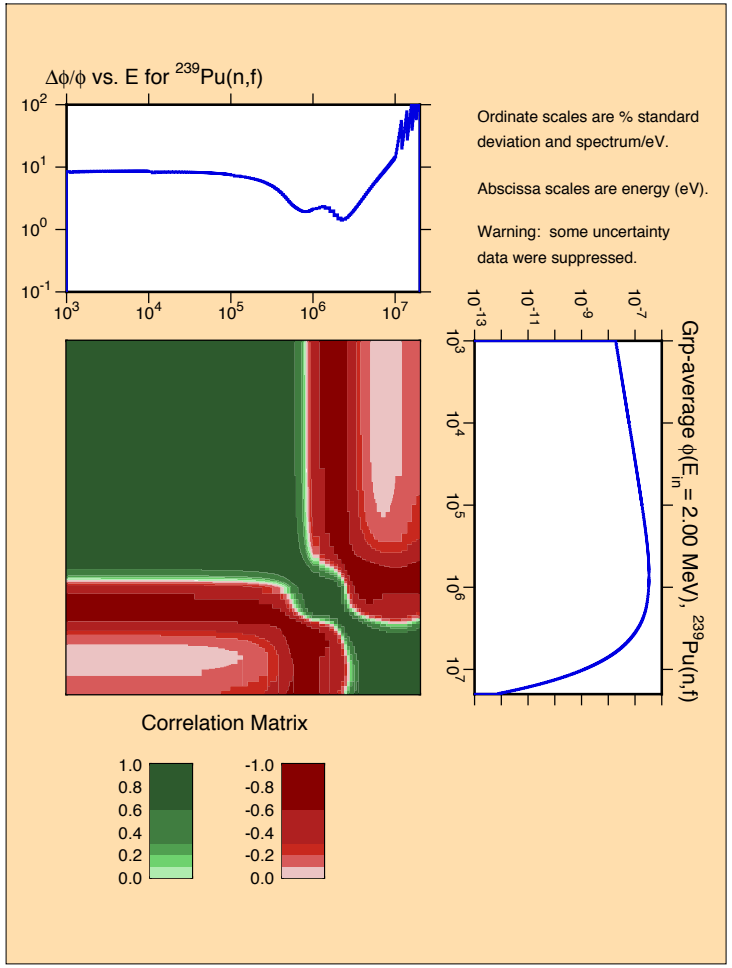
- Determine covariance matrix (variance) of k

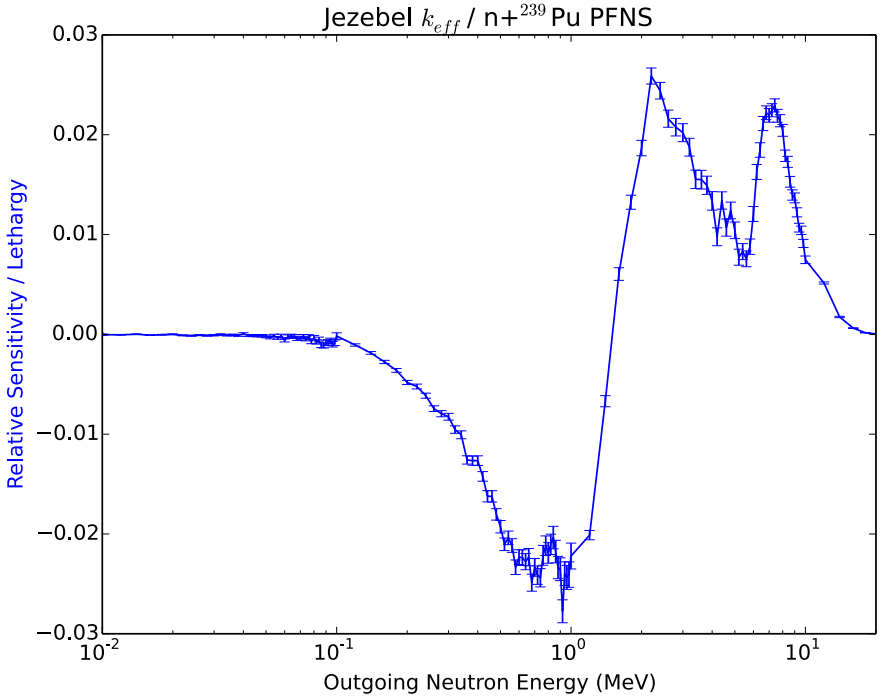
$$\begin{aligned} k(\vec{\Sigma}') &\cong k(\vec{\Sigma}^0) + \vec{S}(\vec{\Sigma}' - \vec{\Sigma}^0)^T \\ &= \vec{S}\,\vec{\Sigma}'^T + \left[k(\vec{\Sigma}^0) - \vec{S}\,\vec{\Sigma}^{0T}\right] \\ &= \mathbf{Ax} \quad + \quad \mathbf{b} \end{aligned}$$

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

Error Propagation (3)

- **Example using sandwich rule, ^{239}Pu PFNS impact on *k***





Uncertainty in *k* due to ^{239}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

Los Alamos NATIONAL LABORATORY EST. 1943

NCSP NUCLEAR CRITICALITY SAFETY PROGRAM

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

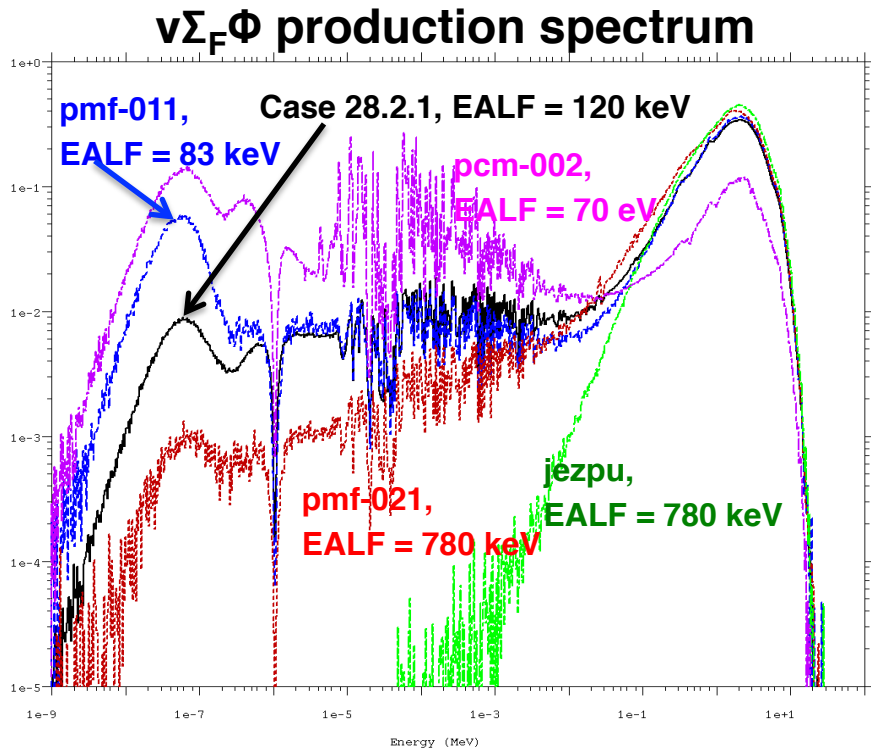
- **Whisper History, Background, SQA Status, Documentation**

- **Whisper Methodology**
  - **Capabilities**
  - **Correlation Coefficients**
  - **Cross-section Covariance Data**
  - **Sensitivity Profiles**
  - **Variance in Keff & 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

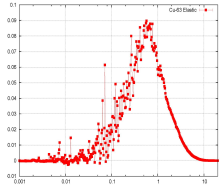
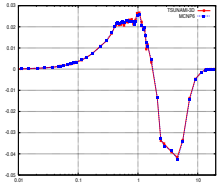
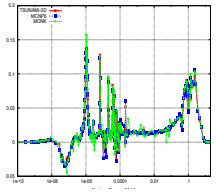
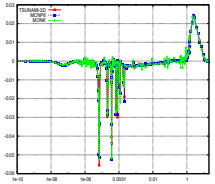
- **MCNP determines sensitivity profiles to characterize the neutronics of an application or benchmark, S(energy, reaction, isotope), S = (dk/k) / (dσ/σ)**
- **WHISPER uses sensitivity profiles & data covariances to select similar benchmarks, determine bias, bias-uncertainty, & margin-of-subcriticality 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:

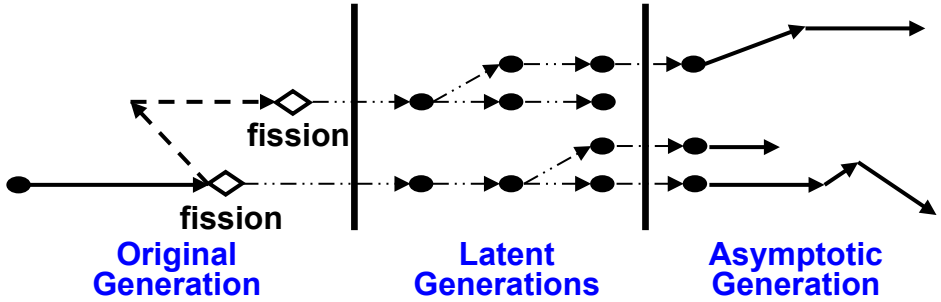
$$S_{k,x} = \frac{dk/k}{dx/x} = -\frac{\left\langle \psi^{\dagger}, \left(\Sigma_x - S_x - k^{-1}F_x\right)\psi \right\rangle}{\left\langle \psi^{\dagger}, k^{-1}F\psi \right\rangle}$$

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

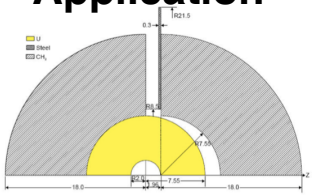





etc.

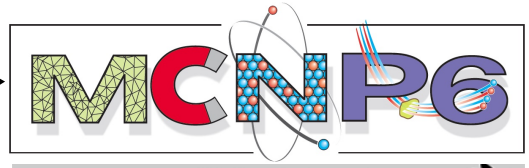
- **MCNP Monte Carlo** uses the Iterated Fission Probability method to compute adjoint-weighted integrals for the sensitivity profiles
  - Tally scores are collected in original generation, adjoint-weighting is based on the progeny in the asymptotic generation



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

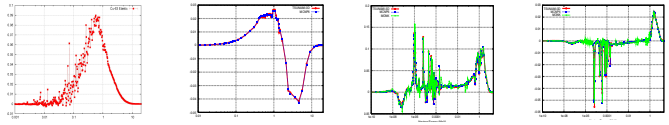


Application



MCNP Monte Carlo
Criticality Calculation

Nuclear Cross-section Data



Application
Sensitivity Profile

Nuclear Cross-section Covariance Data

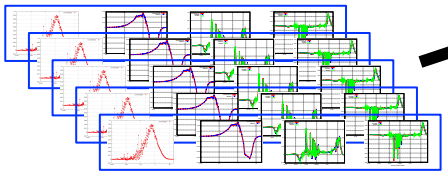
WHISPER analysis

WHISPER

Pattern matching – application sensitivity profile vs catalog

Select similar experiments

Statistical analysis to determine bias & uncertainty & extra margin



Catalog of sensitivity profiles for 1100+ experiments

USL
Upper Subcritical Limit
for criticality safety analysis

Whisper Methodology for Validation & USLs

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

- **Using Whisper**

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

  Run Whisper:

  ① Automated, physics-based selection of benchmarks that are neutronically similar to the application, ranked & weighted
  - Compare Application S_A to each of the Benchmark sensitivities $S_{B(i)}$
  - Select most-similar benchmarks (highest S_A-$S_{B(i)}$ correlation coefficients)

  ② Bias + bias uncertainty from Extreme Value Theory
  - Statistical analysis - based on most-similar Benchmarks selected

  ③ Margin for nuclear data uncertainty estimated by GLLS method
  - Use benchmark sensitivities & cross-section covariance data to estimate the MOS for nuclear data uncertainties

MCNP6 & Whisper Status

- **MCNP releases by RSICC**

  MCNP6.1 – 2013, production version

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

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

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

  Reference Collection – 700+ technical reports

  V&V Test Collection – 1434 test problems

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

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

Whisper-1.1.0 Update

Whisper code updates: 1.0.0 → 1.1.0

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

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

Whisper support updates: 1.0.0 → 1.1.0

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

  whimcnp → whisper_mcnp.pl

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

Whisper files updates: 1.0.0 → 1.1.0

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

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 Methodology

- MCNP6
  - Determine Sensitivity Profiles for Benchmarks B_1 ... 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_{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_{Bj} computed by MCNP, J = 1, ..., N (N = number of benchmarks)
- **Find correlation between Application A & Benchmark B_J, J = 1 ... 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 ... 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:**

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

where $w_{min} = 25$ **(default, user opt)**
$w_{penalty} = 100$ **(default, user opt)**

  - **That is, must select enough benchmarks so that $sum\{ w_J \} \geq w_{req}$**
  - **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 ($k_{calc} – k_{bench}$) 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 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 – Calculational Margin (2)

- Let $\beta_J = k_{calc\ J} - k_{bench\ J}$ and $\sigma^2_J = \sigma^2_{bench\ J} + \sigma^2_{calc\ J}$
  - For convenience, the X_J below are opposite in sign to β_J
- For a set of N benchmarks, let X_J be a random variable normally distributed about β_J with uncertainty σ_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:

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

- The cumulative distribution function (CDF) for X is

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

Whisper Details – Calculational Margin (3)

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

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

- For all benchmarks J = 1, ..., N, Whisper computes
  - Benchmark weight, w_J
  - Bias, β_J
  - Bias uncertainty, σ_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(CM) = .99 (.99 is default, user opt)

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

Whisper Details – Calculational Margin (4)

- **Bias & bias uncertainty**

  USL = 1 - CM - MOS

  = 1 + bias - bias-uncert - $\Delta_{\text{non-conserv}}$ - 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(CM) = .99**

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

$$\text{bias} = -\int_{-\infty}^{\infty} x \cdot f(x)\,dx = -\int_{-\infty}^{\infty} xF(x)\sum_{J=1}^{N} w_J \frac{f_J(x)}{F_J(x)}\,dx$$

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

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

  if bias>0, CM = CM + bias

Whisper Details – Calculational 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

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

- **MOS = additional margin "that is sufficiently large to ensure that the calculated conditions will actually be subcritical" (ANSI/ANS-8.24)**

- **$MOS_{software}$ (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: $MOS_{software} = 0.0050$
      → 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

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

- $MOS_{application}$
  - 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

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

- **MOS_{data}**
  - **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_{data} 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_{data}, which led to large conservative margins**
  - **Whisper (LANL) & Tsunami (ORNL) both use essentially the same methodology to address MOS_{data} – GLLS**
  - **Generalized Linear Least Squares (GLLS) takes into account the experiments, calculations, sensitivities, & data covariance data to predict MOS_{data}**

Margin of Subcriticality - GLLS

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

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

  - **Δk is a vector of the relative changes in the ratio of calculated k to benchmark k, due to the change in cross-section data Δx. The length of Δk is the number of benchmarks**
  - **Δx is a vector of the relative differences of cross-section data from their mean values. The length of Δx is (isotopes)*(reacions)*(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 Δx (and the resulting Δk) such that χ^2 is minimized**

Margin of Subcriticality - GLLS

- **The goal of GLLS:**
  - Determine adjustments to the nuclear data, Δx, which produce changes in computed k_{eff} for benchmarks, Δk, 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, Δx = 0, so χ^2 determined only by differences in calculated & benchmark k's
  - If data is adjusted to decrease 1st term, then 2nd term increases
  - GLLS determines optimum tradeoff (minimum χ^2) between Δx & Δk

GLLS

Measured k_{eff} values for benchmarks:

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

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

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

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

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

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, $\vec{x}$:

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

Recall that:

Sensitivity matrix for a set of benchmarks:

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

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

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

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:

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

or

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

GLLS

For the vector $\vec{d}$, $(d_i) = \dfrac{k_i(\vec{x}) - m_i}{k_i(\vec{x})} \qquad i = 1,...,I$

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:

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

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:

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

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

GLLS

GLLS gives the data adjustments (& resulting Δk's) that minimize the Q or R functions (also called χ^2)

The adjustments also give reduced uncertainties:

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

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

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 1σ uncertainties in k for the adjusted data.

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

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

where n_σ = 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):

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

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

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

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

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_usl.pl
```

- **Submit batch job file**

```
msub   job.txt
```

Whisper-1.1.0 – Interactive

To try it, on Moonlight HPC:

- **Set & export WHISPER_PATH environment variable**
  - **bash:**

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

  - **csh, tcsh:**

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

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

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

Using whisper_mcnp (1)

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

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

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

  - **Creates files & dirs:**

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

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

Using whisper_mcnp (2)

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

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

  - **Note that there are large numbers of neutrons/cycle & cycles for the KCODE input. While it may be tempting to reduce these to get shorter runs, that is discouraged since it is important to achieve reasonable statistical uncertainties on the sensitivity profiles for a large number of reactions, isotopes, & energies.**
- **After using whisper_mcnp, after the MCNP6 jobs complete:**
  - **The Calcs/ directory will contain these files**

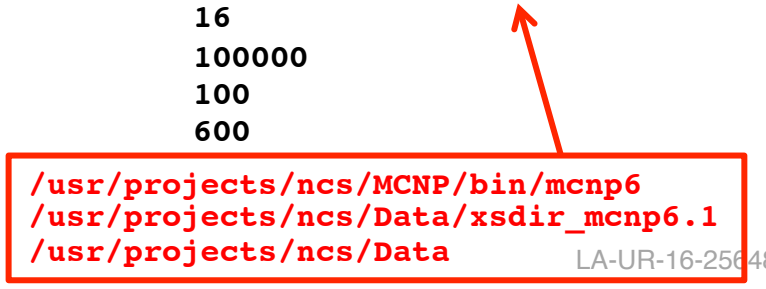
myjob.i	modified MCNP6 input file, with kcode, ksen, kopts, prdmp
myjob.io	output file from MCNP6 jobs
myjob.ir	runtpe file
myjob.is	srctp file

whisper_mcnp.pl - Usage

```
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/SCALE6.1
 Covariance Adjusted Data Path  =
 Application TOC File           = KeffSenList.toc
 Application Sensitivity Path   = KeffSenLib/
 User Options File              =
 Output File                    = Whisper.out
```

whisper_usl.pl (4)

```
........

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

         ........

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

    myjob.i                             0.01329      0.00120      0.97860     -0.00972
```

Whisper.out (1)


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

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

                    ..........

  246 benchmarks read,       0 benchmarks excluded.

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

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

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

Whisper.out (2)

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

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

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

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

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

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

Benchmark rankings shown below

Conclusions & Discussion

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

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

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

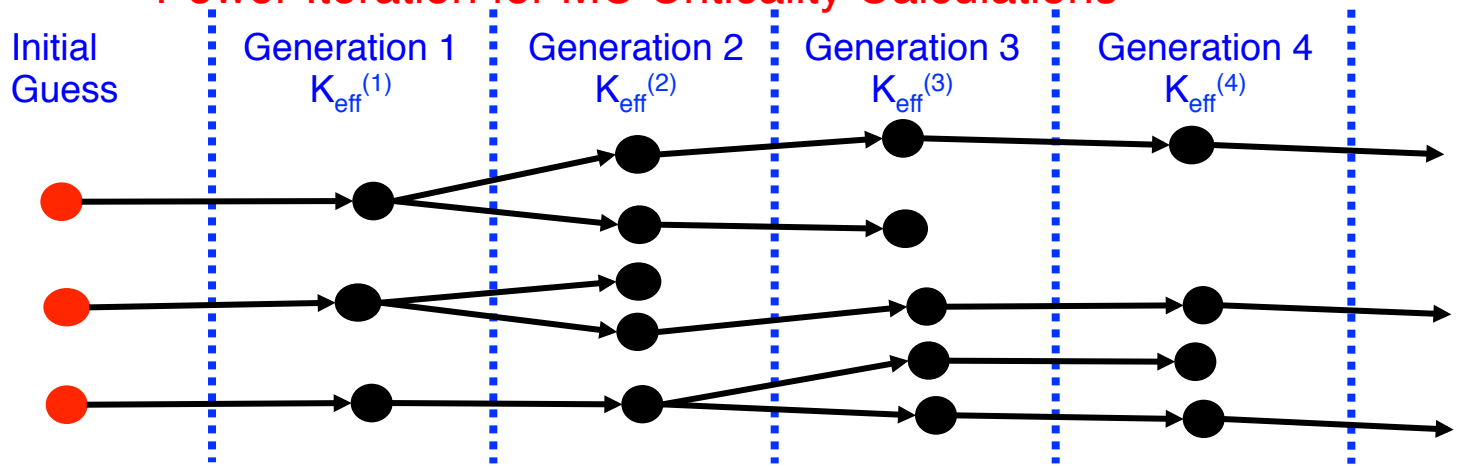
In the near-term, S/U-based methods provide powerful tools for supporting, complementing, and extending traditional validation methods. It is expected that the use of S/U-based tools will expand as more experience & knowledge is acquired.

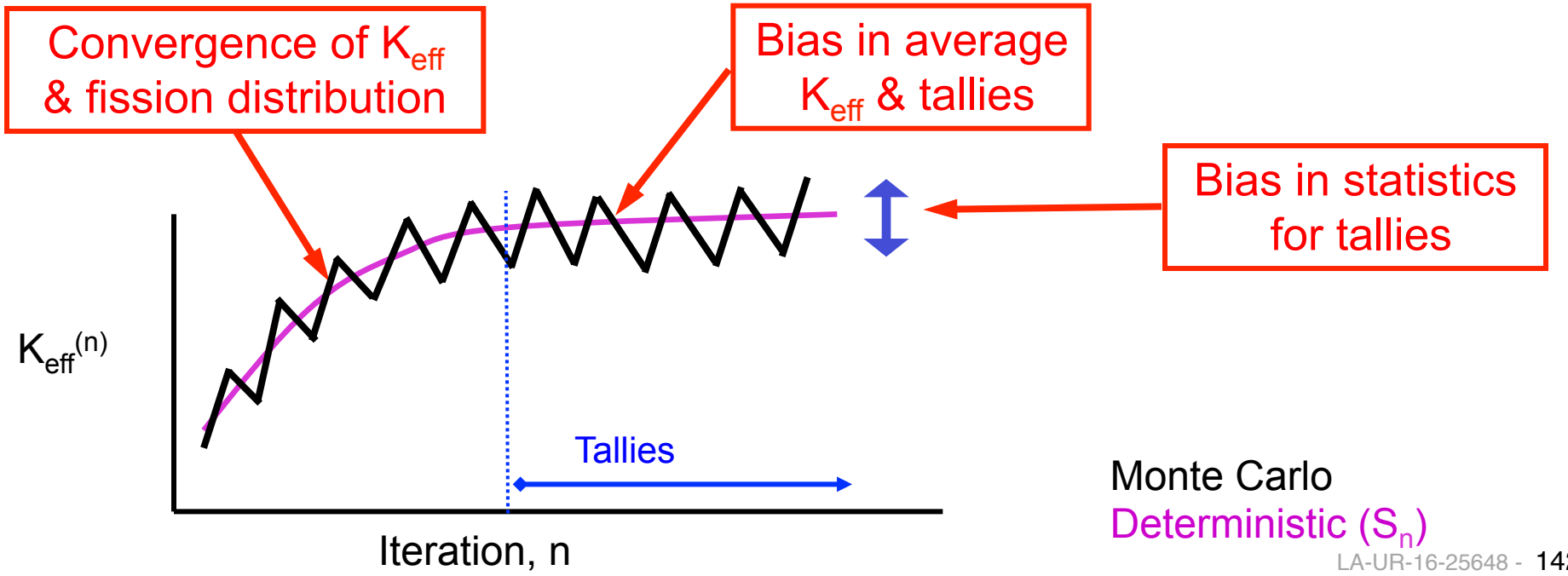
Best Practices for Monte Carlo Criticality Calculations

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

Methodology & Concerns

Power Iteration for MC Criticality Calculations





Monte Carlo
Deterministic (S_n)

Convergence

- **Monte Carlo codes use power iteration to solve for K_{eff} & Ψ for eigenvalue problems**
- **Power iteration convergence is well-understood:**

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

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

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

  - **First-harmonic source errors die out as ρ^n, $\rho = k_1 / k_0 < 1$**
  - **First-harmonic K_{eff} errors die out as $\rho^{n-1}(1-\rho)$**
  - **Source converges slower than K_{eff}**

- **Most codes only provide tools for assessing K_{eff} convergence.**

➜ **MCNP also looks at Shannon entropy of the source distribution, H_{src}.**

Bias in K_{eff} & Tallies

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

- **Bias in Keff, due to renormalization**

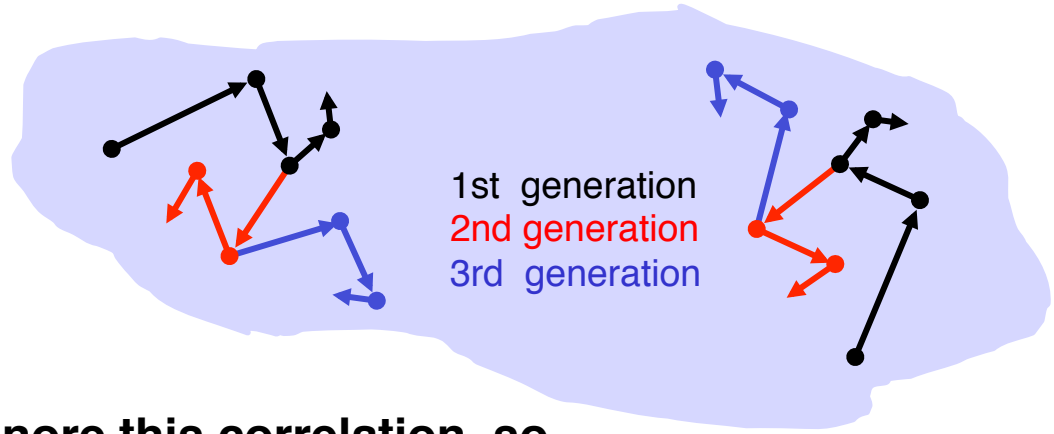
$$\text{Bias in } K_{eff} \propto \frac{1}{M}$$

M = 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_{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 σ^2) > (computed σ^2), since correlations are positive**

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

Best Practices – MC Crit Calcs - Summary

- **To avoid bias in K_{eff} & tally distributions:**
  - **Use 10K or more neutrons/cycle (maybe 100K+ for large system)**
  - **Always check convergence of both K_{eff} & H_{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_{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
  - keff ± σ (combined col/trk/abs only)
  - EALF, ANECF, % fast/intermed/thermal fissions
  - For solutions, H/Pu^{239} or H/U^{235}
  - 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

Monte Carlo Methods

F. B. Brown, "Fundamentals of Monte Carlo Particle Transport," LA-UR-05-4983, available at 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", 8th 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

Los Alamos NATIONAL LABORATORY EST. 1943

NCSP NUCLEAR CRITICALITY SAFETY PROGRAM

Examples using Whisper

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

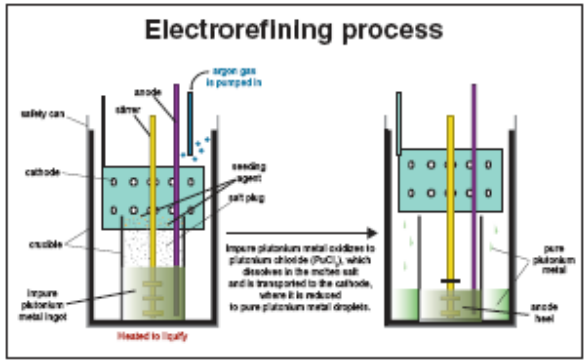
- **General Studies**
  - Example 6: "Revisiting a Practical Application of the Single-Parameter-Subcritical-Mass Limit for Plutonium Metal with Whisper"
  - Example 7: Critical-mass curves and USL-mass curves comparison

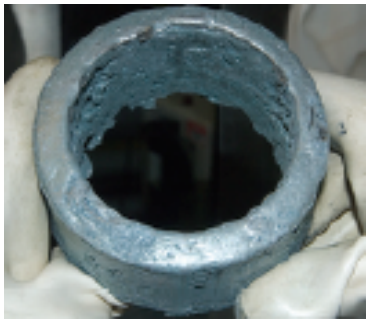
Note for examples & demo:

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.

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**

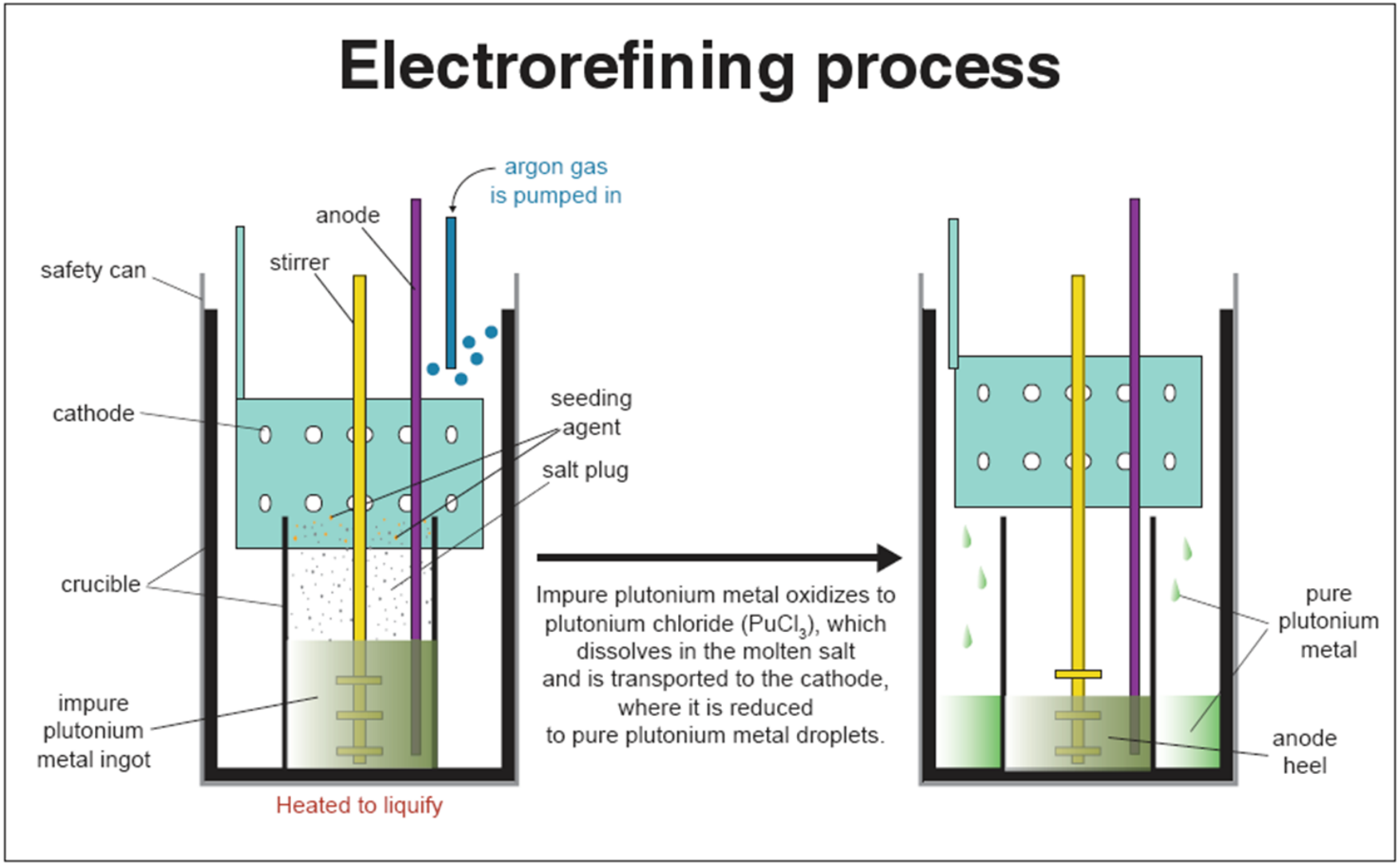




Ref. Actinide Research Quarterly 3rd Quarter 2008

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

Pyrochemical Processing

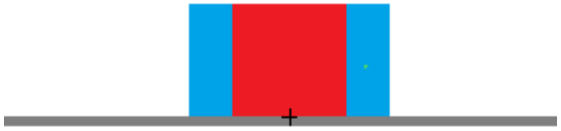


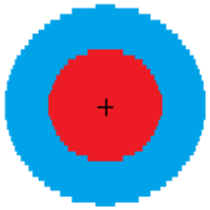
Ref. Actinide
Research Quarterly
3rd Quarter 2008

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^3
- Reflected radially with 1 inch of water
- Reflected on the bottom with ¼ inch steel





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

    mcnp6 i=wval1.txt

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

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

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

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

    - Use parameters for dimensions & location of KSRC point

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


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

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

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

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

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

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

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

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

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

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

```
whisper_mcnp.pl   inp_case*
```

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

```
-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

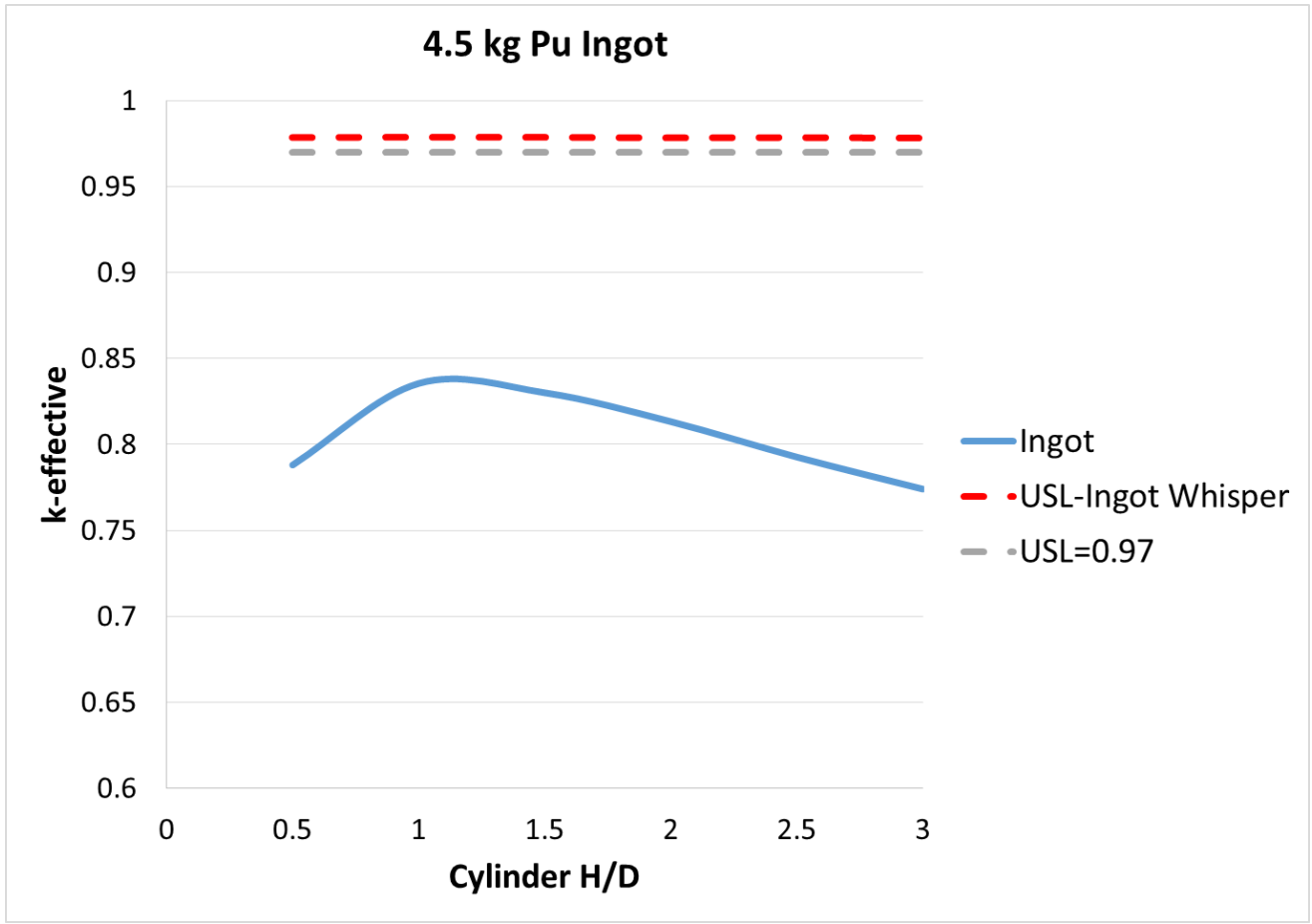
```
mcnp6 i=wval1.txt
```

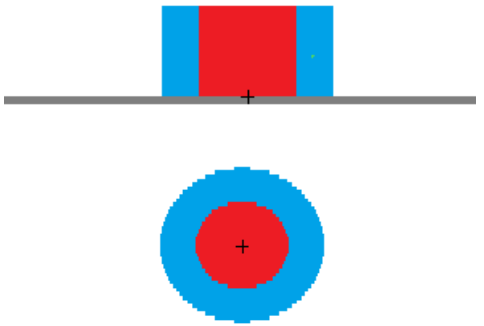
k = 0.83491 (41)

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:

USL = 0.99-MOS-AoA = 0.97 - AoA

Example 2

–

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

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

- **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?**

5.3 Metallic units

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

Table 3 – Single-parameter subcritical limits for metal units

Parameter	Subcritical limits for		
	^{233}U [15]	^{235}U [16]	^{239}Pu [17]
Mass of fissile nuclide (kg)	6.0	20.1	5.0

- **Is annular cylinder validated geometry?**

From a typical traditional validation report

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

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

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

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

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

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

    mcnp6 i=wval2.txt

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

      `c @@@ HD = 1.0 2.0 3.0`
    - For a given H/D, compute Pu height
    - Define list for inner radius RIN_PU

      `c @@@ RIN_PU = 0.001 0.5 1.0 2.0`
    - Then other dimensions & source

Solid cylinder

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

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

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

Hollow cylinder

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

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

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

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

mcnp_pstudy -i wval2p.txt -whisper

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

whisper_mcnp.pl inp_case*

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

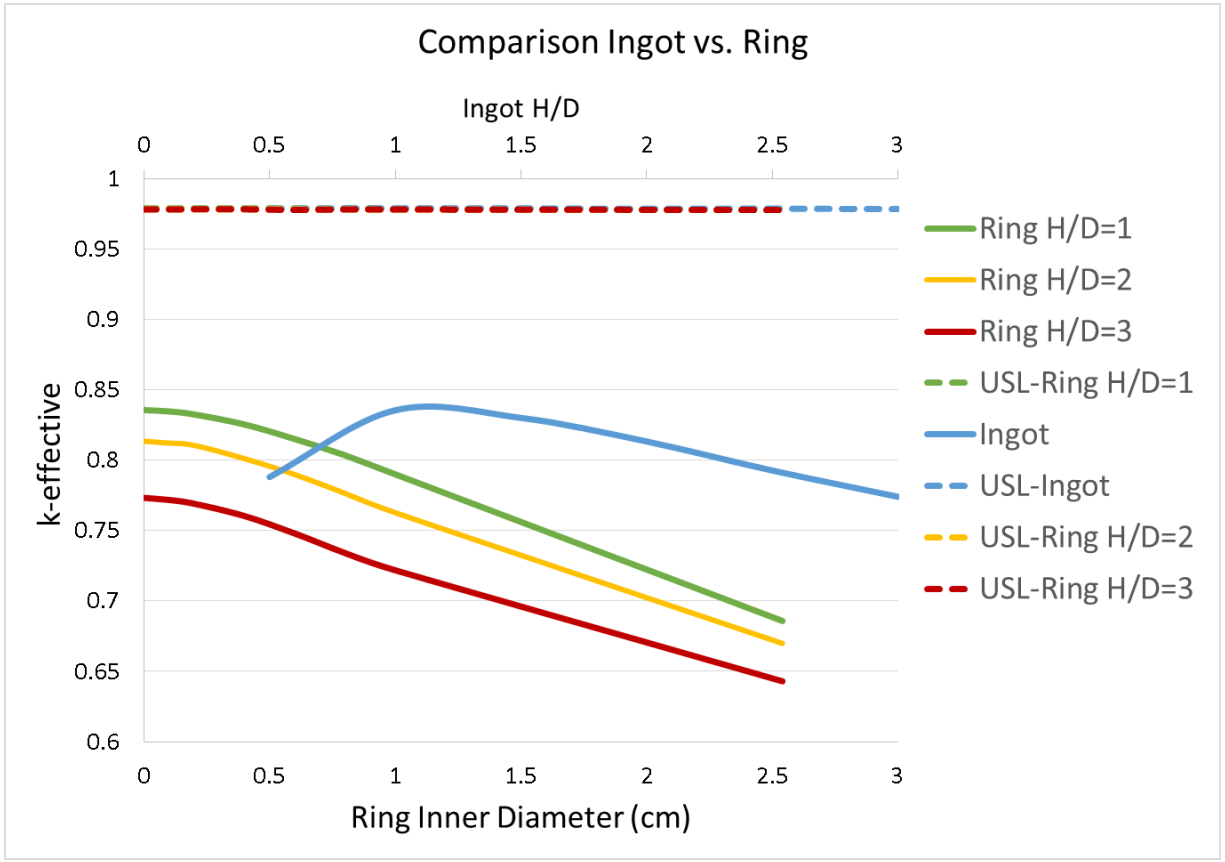
whisper_usl.pl

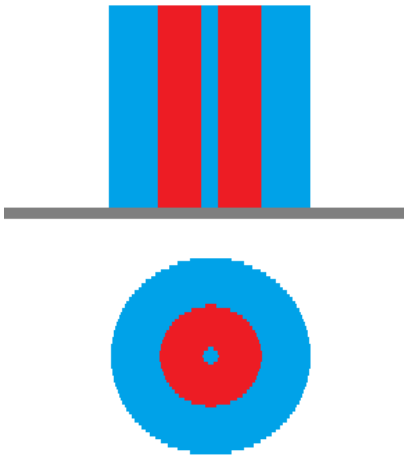
use whisper_usl to run Whisper & determine USL for each case

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

MCNP6-Whisper Results

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

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

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

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

Traditional Validation Results:

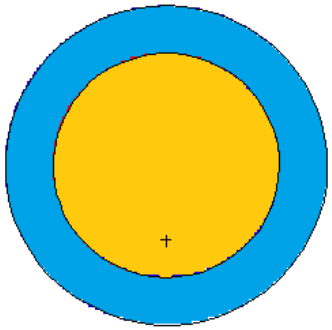
USL = 0.99-MOS-AoA = 0.97 - AoA

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

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^3
- Density of NaCl = 1.556 g/cm^3



- Run commands:

```
mcnp_pstudy  -i  wval3p.txt  -whisper
whisper_mcnp.pl    inp_case*
whisper_usl.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
 si1  0  5.989
 sp1  -21 2
c
m1     1001.80c 2      8016.80c 1
mt1    lwtr.20t
m4     94239.80c -0.81117881
       11023.80c -0.07427730
       17035.80c -0.08561650
       17037.80c -0.02893221
```

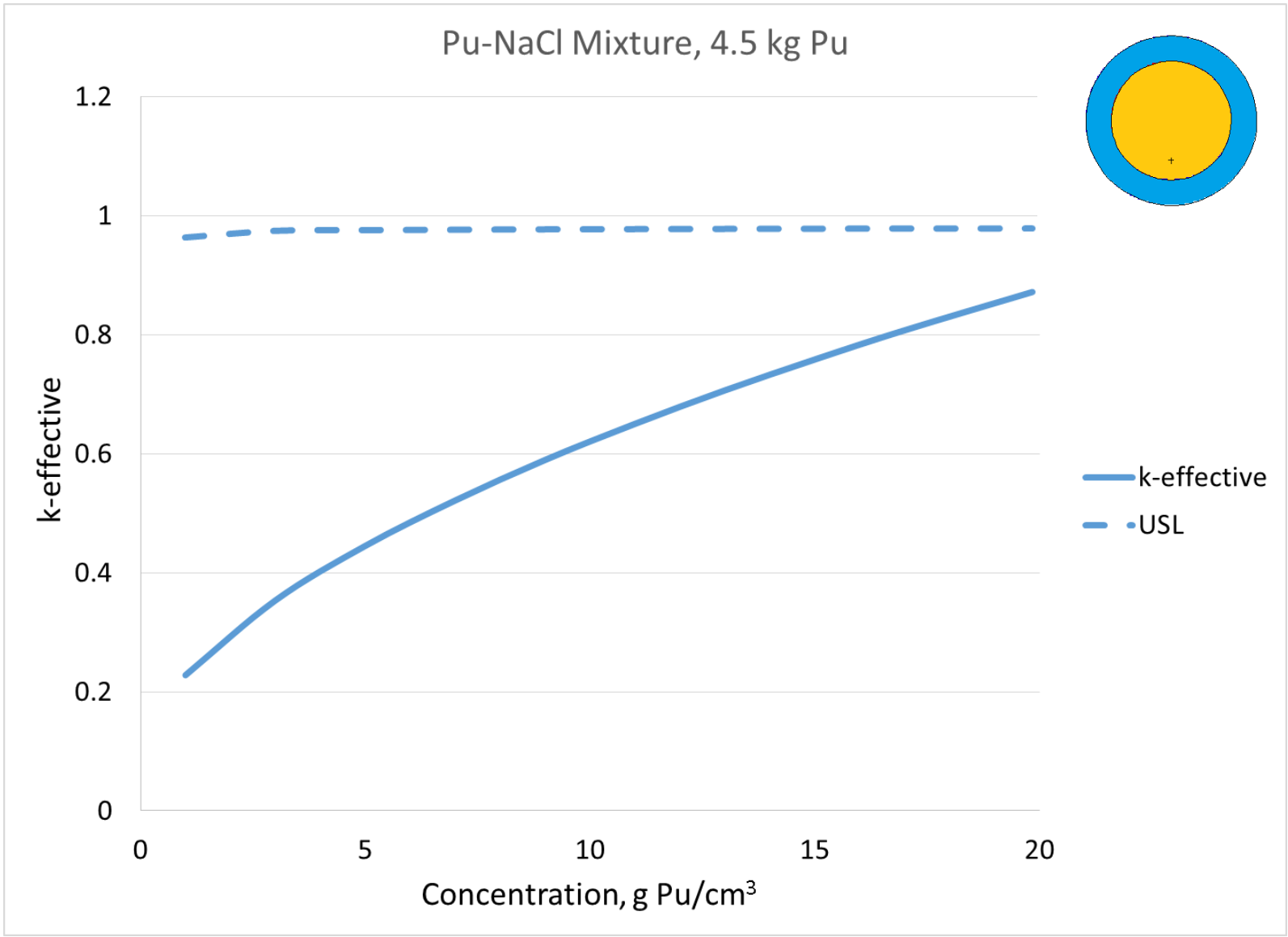
```
wval3p:  Pu mixed with NaCl
c @@@         PI = 3.141592654
c @@@ PU_MASS    = 4500
c @@@ PU_VOL     = ( PU_MASS / 19.86 )
c @@@ NACL_MASS = 1.e-6  500 1000 1500 2000
c @@@ NACL_VOL   = ( NACL_MASS / 1.556 )
c
c    Pu    mass = PU_MASS g
c    NaCl mass = NACL_MASS g
c    Pu    density (pure) = 19.86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

 1   so   RADIUS
 2   so   OUTER_H2O

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

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

MCNP6-Whisper Results



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

MCNP6-Whisper Results

USL baseline = .979

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

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

Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 – AoA

*bold indicates same benchmark selected for Pu ingot

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**

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?**

From a typical traditional validation report

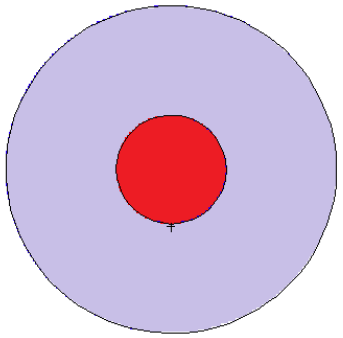
Parameter	Area of Applicability
Fissile Material	^{239}Pu
Fissile Material Form	Pu Metal, PuO_2, and $Pu(NO_3)_4$
H/^{239}Pu	$0 \leq H/^{239}Pu \leq 2807$
Average Neutron Energy Causing Fission (MeV)	$0.003 \leq ANECF \leq 1.935$
^{240}Pu	0 to 42.9 wt% ^{240}Pu
Moderating Materials	none, water, graphite,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

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

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

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

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



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

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

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

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

Example 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
  si1   0 3.78
  sp1   -21 2
c
 m1   73180.80c 0.00012    73181.80c 0.99988
 m4   94239.80c 1
 prdmp 9e9 9e9 1 9e9
```

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

  1 so    R_INNER
  2 so    R_OUTER

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

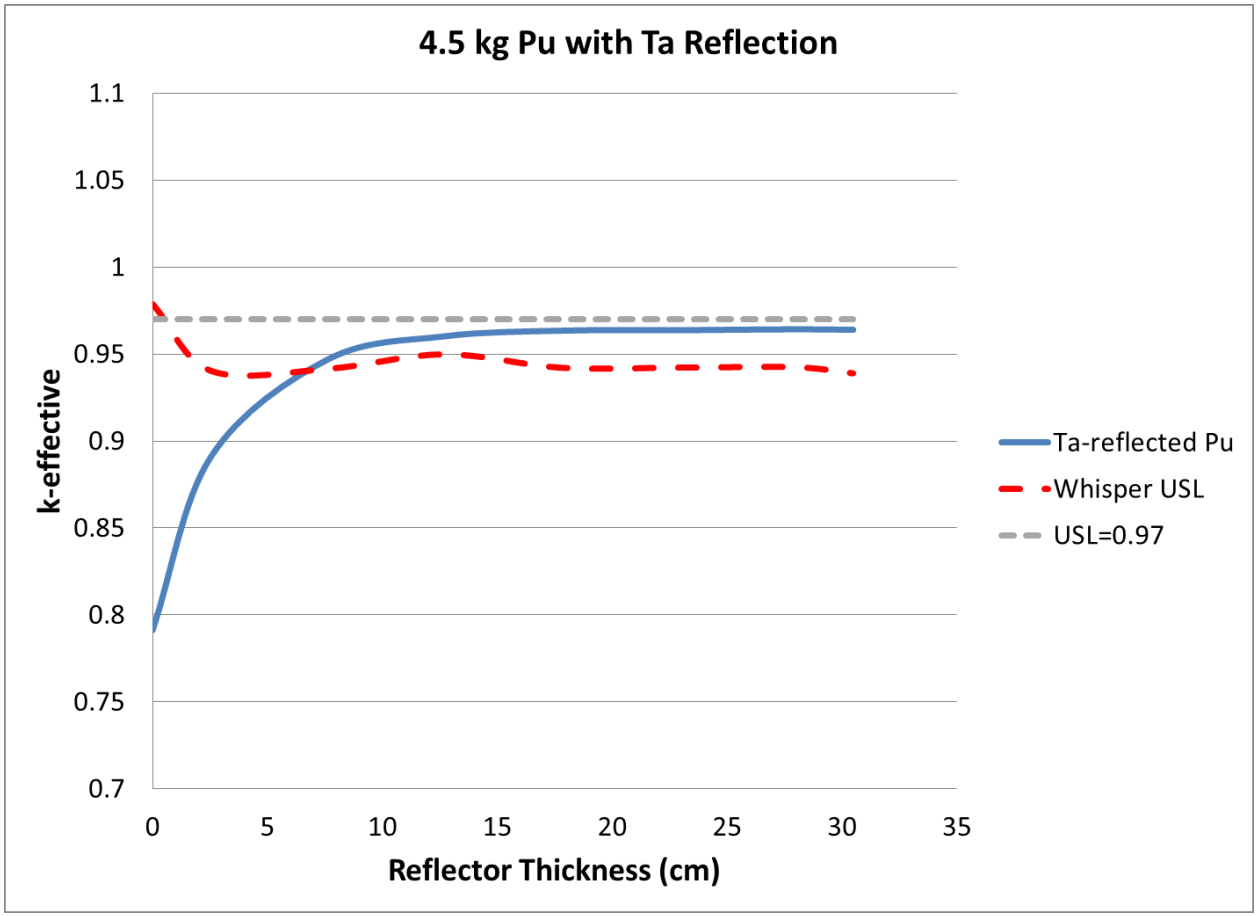
Example 4: 4.5 kg Pu Sphere, Ta-reflected (3)

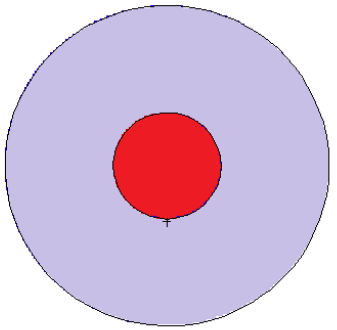
wval4, thick=7.62
mcnp6 i=wval4.txt

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





Example 4: Ta-reflected Pu

MCNP6 and Whisper Results

Run using all 1101 Whisper benchmarks, not just 261 benchmarks for class

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

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

Trouble ! Benchmarks are not very similar to application

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

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

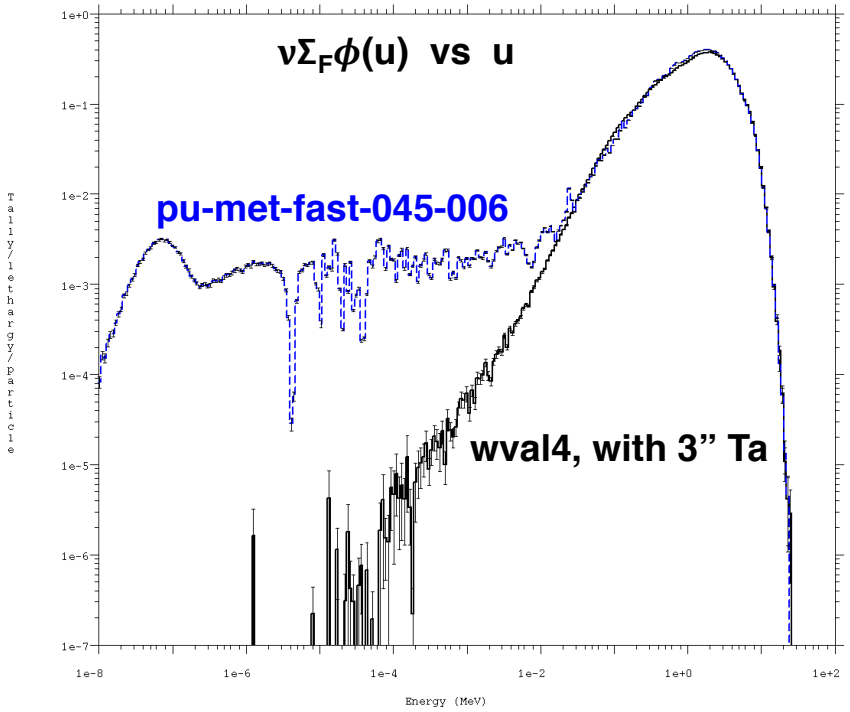
    $0.90 < C_k < 0.95$ → good

    $C_k < 0.90$ → not so good

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

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

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



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

Example 5
–
4.5 kg Pu Sphere,
Oil moderated

Example 5: Oil-Moderated Pu

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

From a typical traditional validation report

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

  - **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**

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

Example 5: Oil-Moderated Pu

- **MCNP6 Input**
- **4.5 kg Pu (0) sphere mixed with variable amounts of Hydraulic oil**
- **Pu concentration range:**
  -19.8 g Pu/cm^3
- **Hydraulic oil composition:**
  $C_{40}H_{33}O_4Cl_6P$
- **Hydraulic oil density:**
  0.871 g/cm^3
- **Reflected with 1 inch of water**

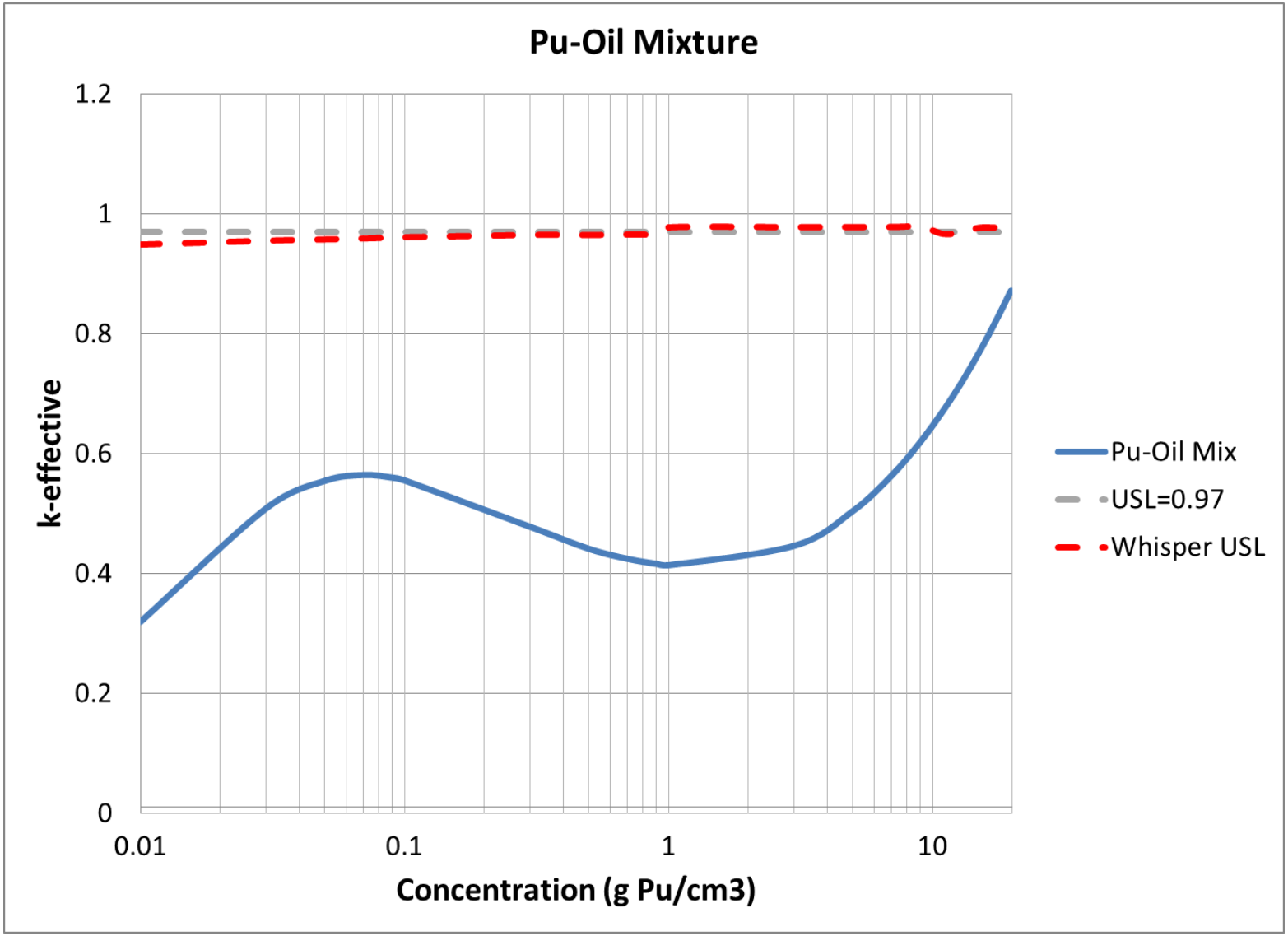
```
Pu mixed with hydraulic oil
c
1    4 -1.827099    -1      imp:n=1
2    1 -1.0         +1 -2   imp:n=1
20   0              +2      imp:n=0

1   so    10.2417609488294
2   so    12.7817609488294

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

Example 5: Oil-Moderated Pu

- **MCNP6 and Whisper Results**



Example 5: Oil-Moderated Pu

MCNP6 and Whisper Results

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

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

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

.....

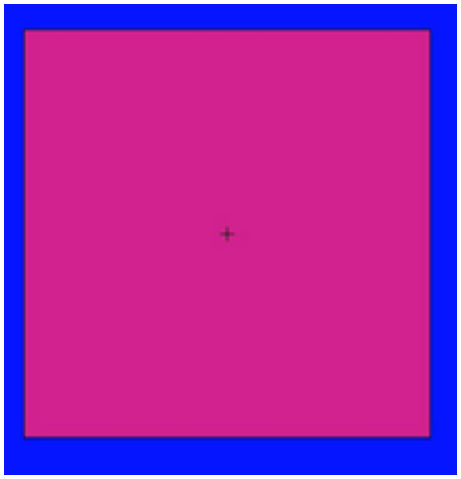
Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 - AoA

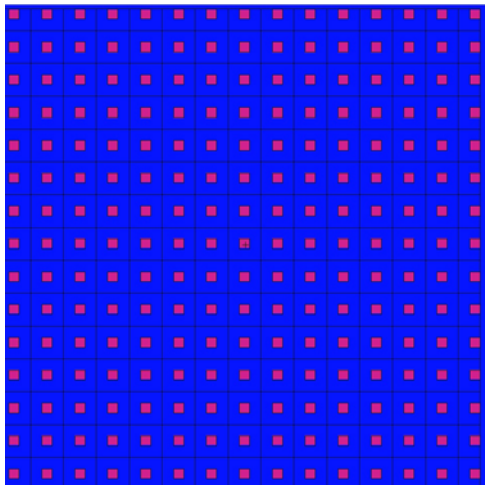
Example 6
–
Revisiting a Practical Application of the SPSL for Pu Metal

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

- LANL 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}Pu) metal cubes in water was performed using MCNP. This study is revisited, and Upper Subcritical Limits (USLs) are presented, using WHISPER.

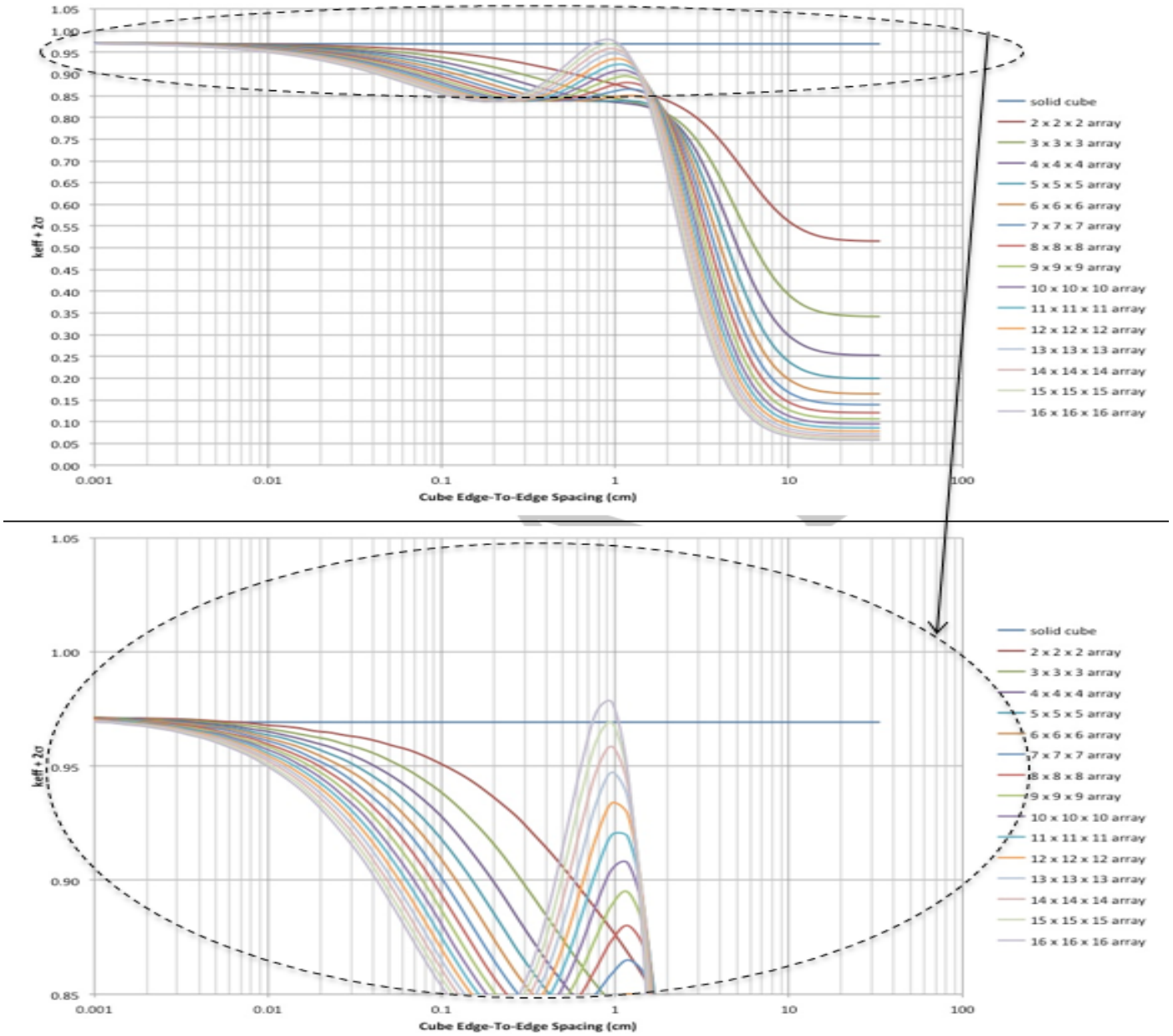


N = 1,
Mass Per Cube = 5,000 g,
Spacing = N/A

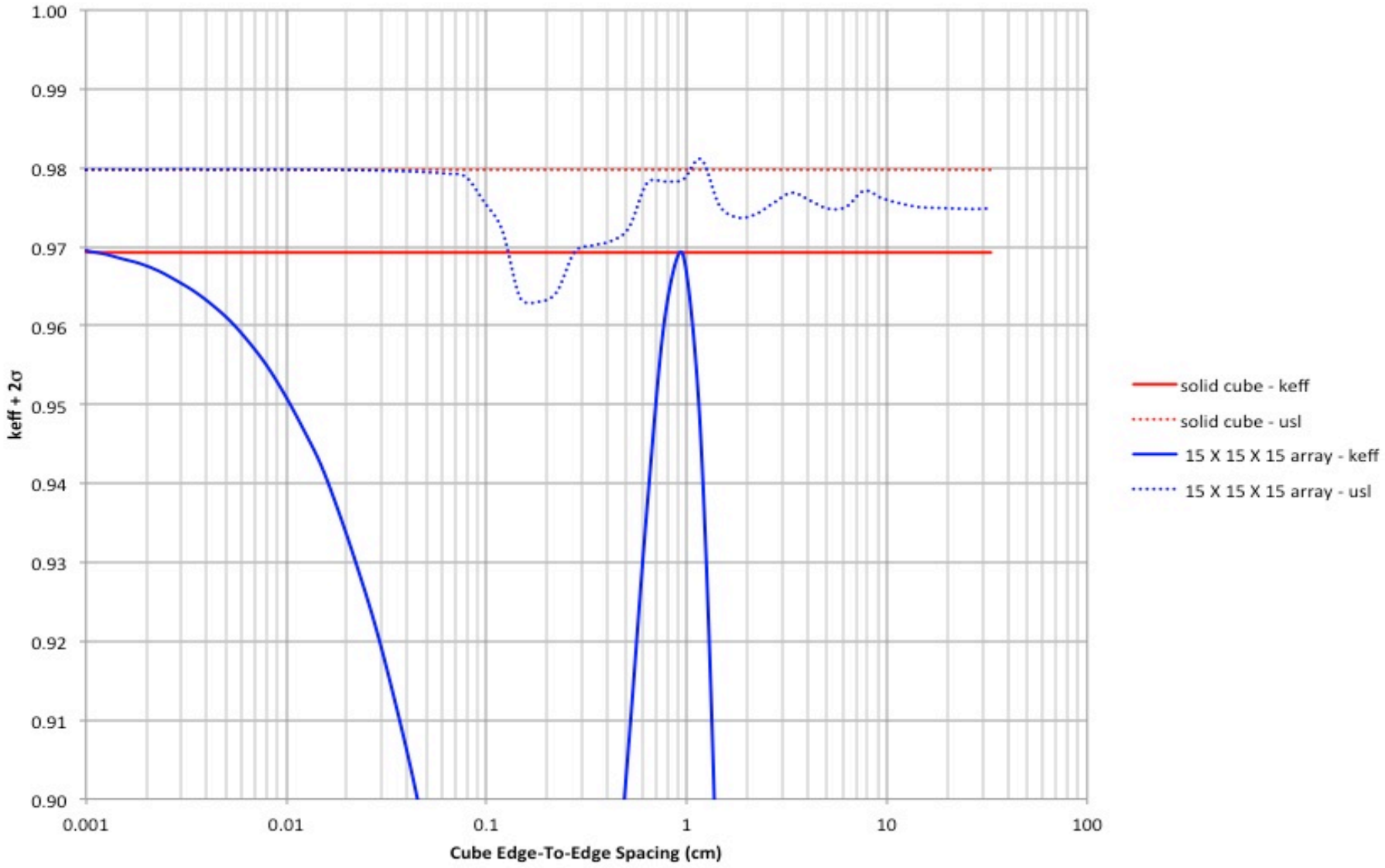


N = 15,
Mass Per Cube = ~1.48 g,
Spacing = 1 cm

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

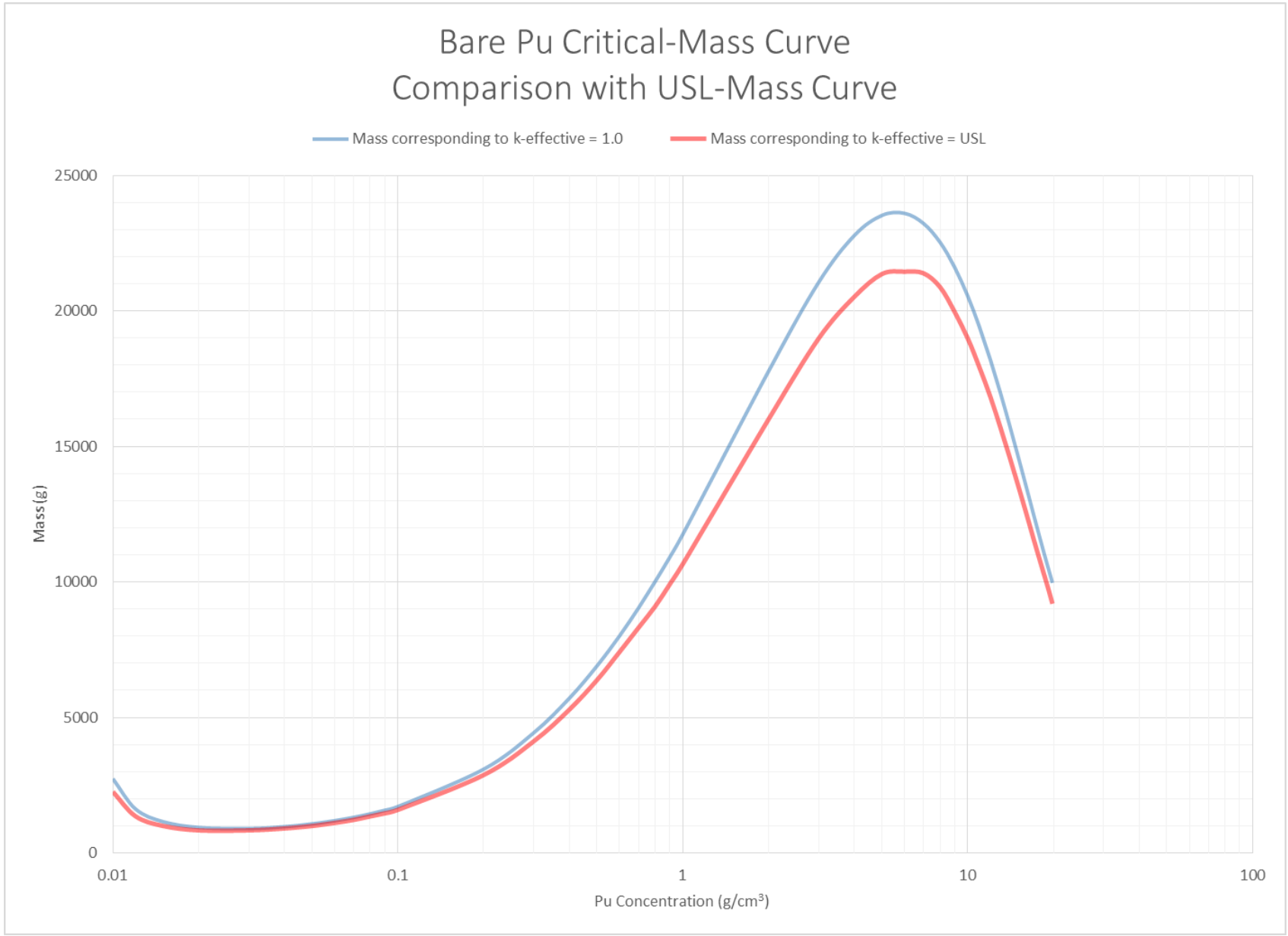


Example 6: Revisiting a Practical Application of the SPSL for Pu Metal



Example 7
–
Critical Mass & USL Curves

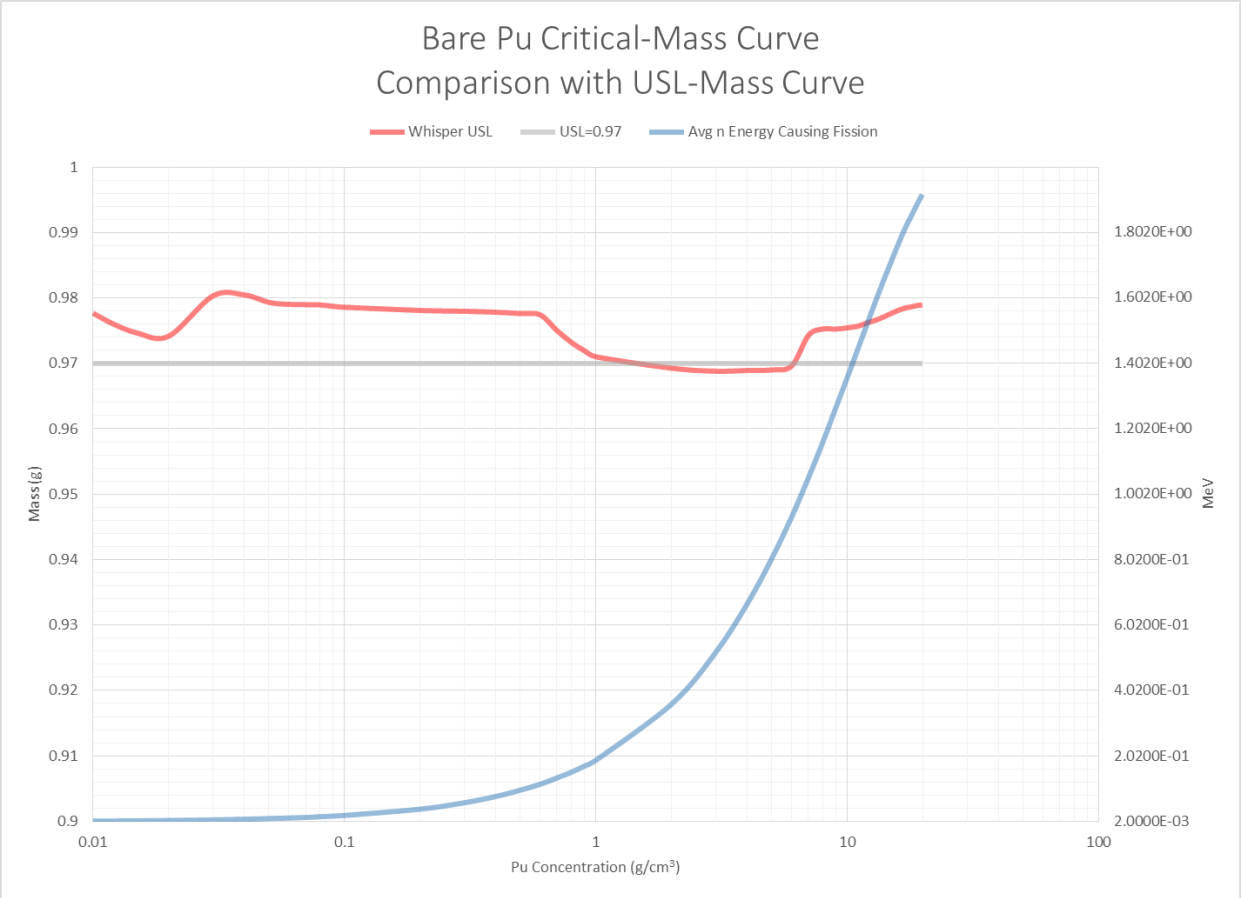
Example 7: Critical-Mass and USL-Mass Curves



Example 7: Critical-Mass and USL-Mass Curves

[ANSI/ANS-8.24 7.2]

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



THERMAL

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

INTERMEDIATE

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

FAST

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

References

References for Whisper & MCNP6 (1)

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

Abstract

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

Theory

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

User Manual

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

Application

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

References for Whisper & MCNP6 (2)

Software Quality Assurance

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

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

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

References for Whisper & MCNP6 (3)

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

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

General References on Adjoints, Perturbation, and Sensitivity Analysis

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

References for Whisper & MCNP6 (4)

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

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

References for Whisper & MCNP6 (5)

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

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