Los Alamos
NATIONAL LABORATORY
EST.1943

LA-UR-16-29071
Approved for public release; distribution is unlimited.

Title: Criticality Calculations with MCNP6 - Practical Lectures

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

Intended for: Teaching MCNP usage to nuclear criticality safety analysts

Issued: 2016-11-29

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

Criticality Calculations with MCNP6 –

Practical Lectures

Forrest Brown, Michael Rising, Jennifer Alwin

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

NCSP
NUCLEAR CRITICALITY SAFETY PROGRAM

Los Alamos
NATIONAL LABORATORY
EST. 1943

Lecture Topics

Principal lectures
Supplemental

@ Copyright 2016, LANL. For educational use only

Course Information

Criticality Calculations with MCNP6 – Practical Lectures

Solving particle transport problems with the Monte Carlo method is simple - just simulate the particle behavior. The devil is in the details, however. A basic understanding of the theory behind Monte Carlo methods is needed to provide a solid foundation for practical application of today's sophisticated Monte Carlo production codes. These lectures provide a balanced approach to learning both the theory and practice of Monte Carlo simulation codes.

Theory: The lectures provide an overview of Monte Carlo simulation methods, including the transport equation, random sampling, computational geometry, collision physics, and statistics. Several lectures focus on the state-of-the-art in Monte Carlo criticality simulations, covering eigenvalue calculations, convergence analysis, dominance ratio calculations, bias in k-effective and tallies, bias in uncertainties, and case studies of realistic calculations. Other lectures cover advanced topics, including stochastic geometry, temperature dependence, fission energy deposition, depletion calculations, parallel calculations, and parameter studies.

Practice: This portion of the class focuses on using MCNP6 to perform criticality calculations for criticality safety and reactor physics applications. Class exercises provide hands-on experience at running the code, plotting both geometry and results, and understanding the code output. There is extensive discussion of "best practices" for performing Monte Carlo calculations.

The class includes lectures and hands-on computer use for a variety of criticality calculations. All class materials and an extensive MCNP® reference collection are provided on a browser-based DVD. Time will be available to discuss individual questions and problems with the MCNP developers, and to pursue additional advanced topics.

MCNP® and Monte Carlo N-Particle® are registered trademarks owned by Los Alamos National Security, LLC, manager and operator of Los Alamos National Laboratory. Any third party use of such registered marks should be properly attributed to Los Alamos National Security, LLC, including the use of the ® designation as appropriate. Any questions regarding licensing, proper use, and/or proper attribution of Los Alamos National Security, LLC marks should be directed to trademarks@lanl.gov.

Preliminaries

- **Welcome …**
  - Thank you for attending
  - Thanks to DOE Nuclear Criticality Safety Program for support
  - Previous classes at LANL, SNL, Y12, PNNL, INL, ANL, LLNL, SRL, …
  - Introductions …..
- **Instructors**
  - From the XCP-3 Monte Carlo Codes team at LANL
- **Questions - any time**
  - Plenty of material , but flexible -- just ask
- **DVD - all class notes, solutions, schedule, & info**
  - **- MCNP Reference Collection (1.5 GB of PDF files)**
  - **- DVD is for educational purposes only**

Learning Objectives from DOE-STD-1135A

After completion of this course, you should be able to:

A. Develop an input model for MCNP.

B. Describe how cross section data impact Monte Carlo and deterministic codes.

C. Describe the importance of validation of computer codes and how it is accomplished.

D. Describe the methodology supporting Monte Carlo codes and deterministic codes.

E. Describe pitfalls of Monte Carlo calculations.

F. Discuss the strengths and weaknesses of Monte Carlo and Discrete Ordinants codes.

G. The diffusion theory model is not strictly valid for treating fissile systems in which neutron absorption, voids, and/or material boundaries are present. In the context of these limitations, identify a fissile system for which a diffusion theory solution would be adequate.

Use MCNP5, MCNPX, or MCNP6 ???

For criticality-safety work:

- **Always use MCNP5-1.60, MCNP6.1, MCNP6.1.1, or MCNP6.2**
  - Thorough V&V for crit-safety & reactors
  - Extensive V&V documentation
  - Fully parallel – MPI + threading
  - Supported by DOE/NNSA Nuclear Criticality Safety Program
- **Never use MCNPX (any version)**
  - No specific V&V testing & documentation for crit-safety & reactors
  - Does not have convergence diagnostics (eg, Shannon entropy)
  - Not funded by DOE/NNSA Nuclear Criticality Safety Program
- **Never use MCNP4 (any version) or older versions**

MCNP
Monte Carlo
–
Introduction

MCNP Overview

Monte Carlo method for simulating radiation transport,

1940s - Von Neumann, Ulam, Fermi, Metropolis, Richtmyer

Common sense approach – simulate reality

- **Geometry:**

  Ray-tracing through "exact" model of problem geometry to determine location of interactions

- **Physics:**

  Cross-section data & physics models used as probabilities for interactions, **random sampling**

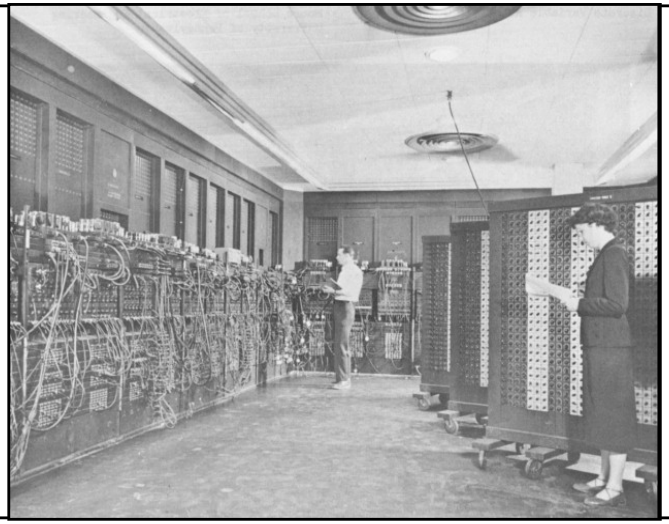
- **Tallies:**

  Bookkeeping, to record how often certain events occur during the simulation.

Monte Carlo & MCNP History

ENIAC – 1945

30 tons
20 ft x 40 ft room
18,000 vacuum tubes
0.1 MHz
20 word memory
patchcords



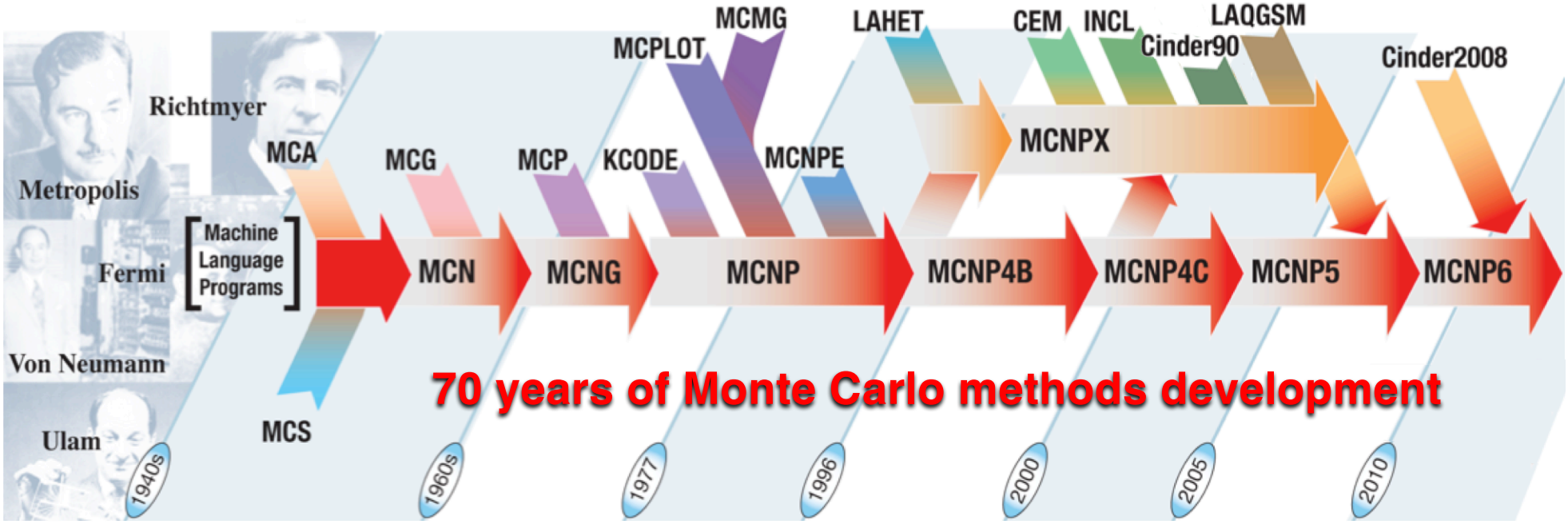
Manhattan Project – 1945...

Discussions on using ENIAC

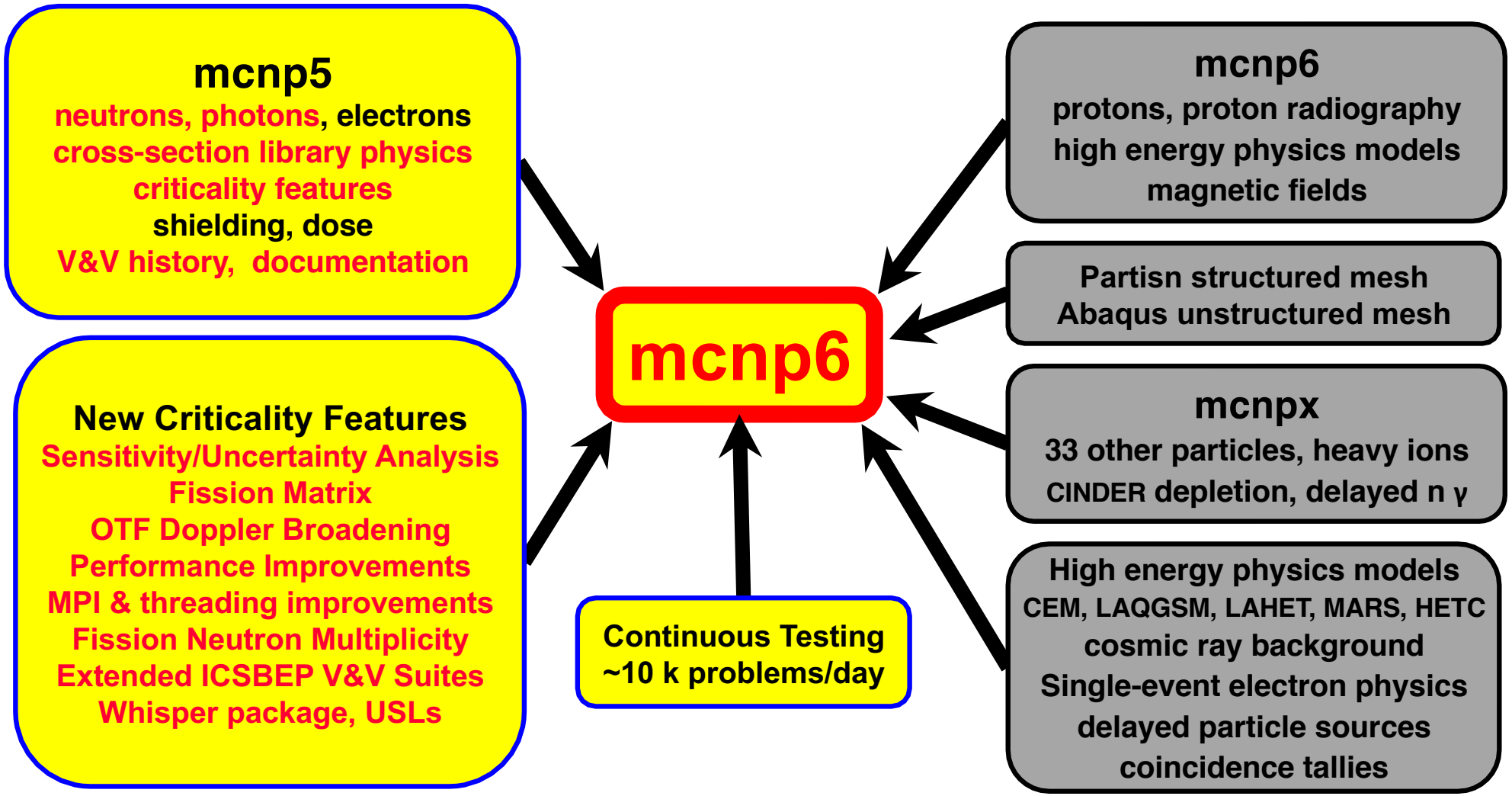
Ulam suggested using the "method of statistical trials"

Metropolis suggested the name "Monte Carlo"

Von Neumann developed the first computer code



MCNP6 Status



mcnp5 – 100 K lines of code
mcnp6 – 500 K lines of code

MCNP6 Overview

MCNP6 can simulate these particles:

neutron (n)	tau neutrino (ν_t)	cascade$_c^0$ (Ξ_c^0)	D_s^+
anti-neutron (n)	proton (p)	lambda$_b^0$ (Λ_b^0)	B^+
photon (γ)	anti-proton ($^-$p)	pion$^+$ (π^+)	B^0
electron (e^-)	lambda^0 (Λ^0)	pion$^-$ (π^-)	B_s^0
positron (e^+)	sigma$^+$ (Σ^+)	neutral pion (π^0)	deuteron
muon- (μ^-)	sigma$^-$ (Σ^-)	kaon$^+$ (K^+)	triton
anti-muon- (μ^+)	cascade^0 (Ξ^0)	kaon$^-$ (K^-)	helium-3
tau- (τ^-)	cascade$^-$ (Ξ^-)	K_0 short	helium-4 (α)
electron neutrino (ν_e)	omega$^-$ (Ω^-)	K_0 long	heavy ions
anti-electron neutrino ($^-\nu_e$)	lambda$_c^+$ (Λ_c^+)	D^+	
muon neutrino (ν_m)	cascade$_c^+$ (Ξ_c^+)	D^0	

What Can MCNP Do?

Detailed models of geometry & physics

- General 3D combinatorial geometry
- Repeated structures
- Lattice geometries
- Geometry, cross section, tally plotting
- ENDF/B-VII physics interaction data

Calculate nearly any physical quantity

- Flux & current
- Energy & charge deposition
- Heating & reaction rates
- Response functions
- Mesh tallies & radiography images
- K-effective, beta-eff, lambda-eff
- Fission distributions

Unique features for criticality calc's

- Shannon entropy of the fission source for assessing convergence
- Dominance ratio, k_1 / k_0
- Stochastic geometry
- Isotopic changes with burnup

> 20,000 users around the world

- Fission and fusion reactor design
- Nuclear criticality safety
- Radiation shielding
- Waste storage/disposal
- Detector design and analysis
- Nuclear well logging
- Health physics & dosimetry
- Medical physics and radiotherapy
- Transmutation, activation, & burnup
- Aerospace applications
- Decontamination & decommissioning
- Nuclear safeguards

Portable to any computer

- Windows, Linux, Mac, Unix
- Multicore, clusters, netbooks, ASC, …
- Parallel, scalable - MPI + threads
- Built-in plotting

Support

- Extensive V&V against experiments
- Web site, user groups, email forum

MCNP & Nuclear Criticality Safety (1)

HEU-MET-THERM-003

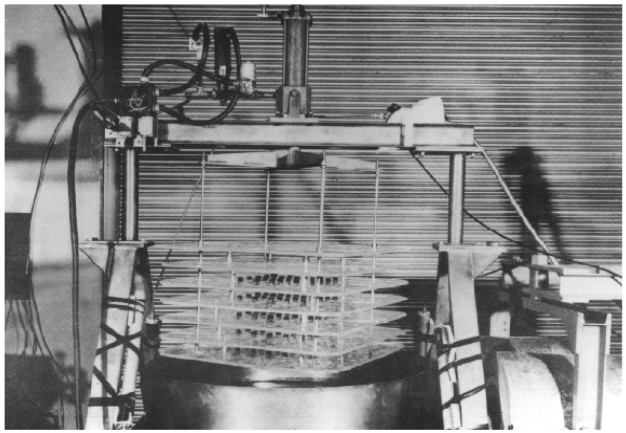
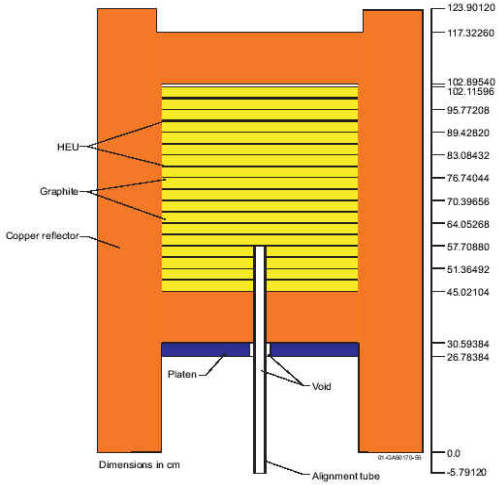
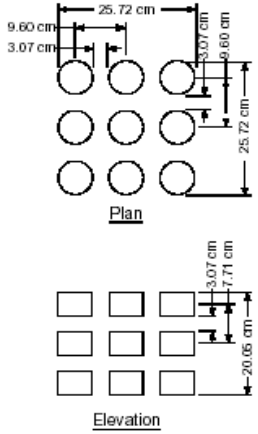


Figure 2. Array of 0.5-in. Cubes Prior to Immersion.

Zeus-2, HEU-MET-INTER-006, case 2



PU-MET-FAST-003, case 3



IEU-COMP-THERM-002, case 3

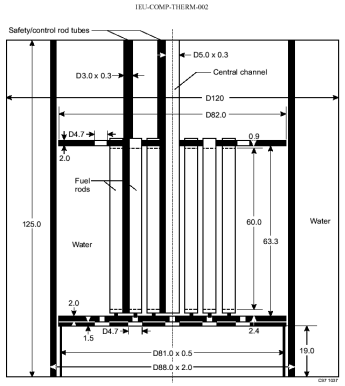


Figure 9. Model of Core.
(dimensions in cm; outer diameters are shown; the value "n" in the notation "× n" is the wall thickness)

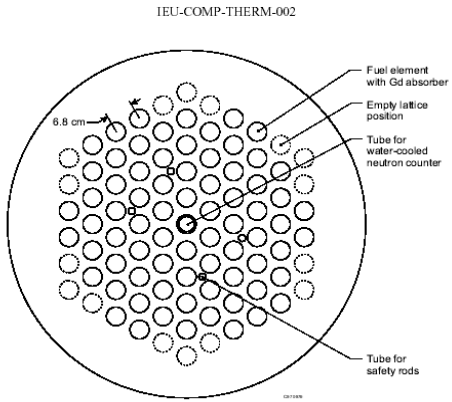


Figure 6. Lattice Plate Loading Chart for Assemblies with Gd Absorber (Case 3 and Case 4).

PNL-33 - MIX-COMP-THERM-002

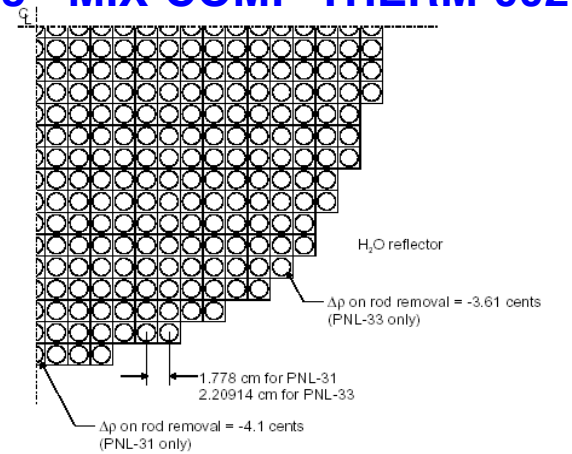


Figure 7. Fuel Loading for PNL-31 and PNL-33.

MCNP & Nuclear Criticality Safety (2)

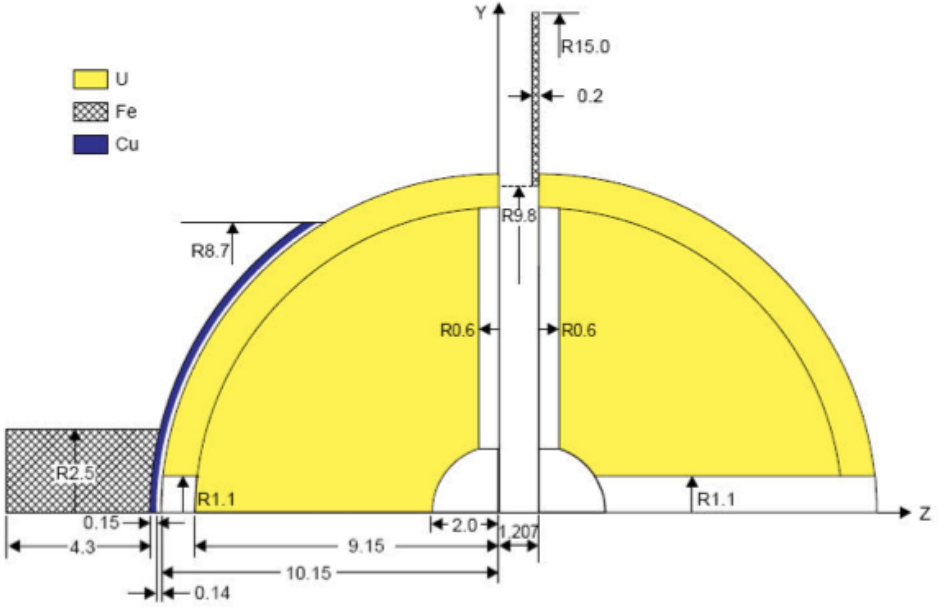


Figure 1. Unreflected HEU Sphere, heu-met-fast-008

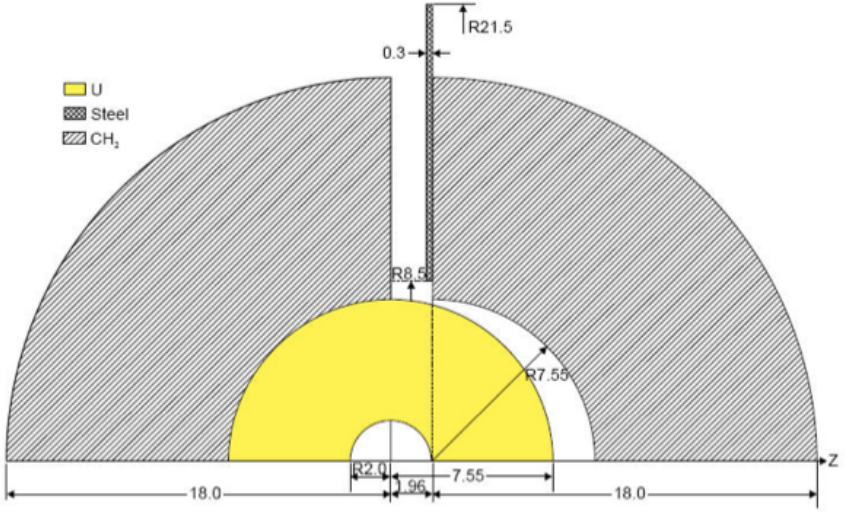


Figure 4. HEU Sphere Reflected by Polyethylene, heu-met-fast-011

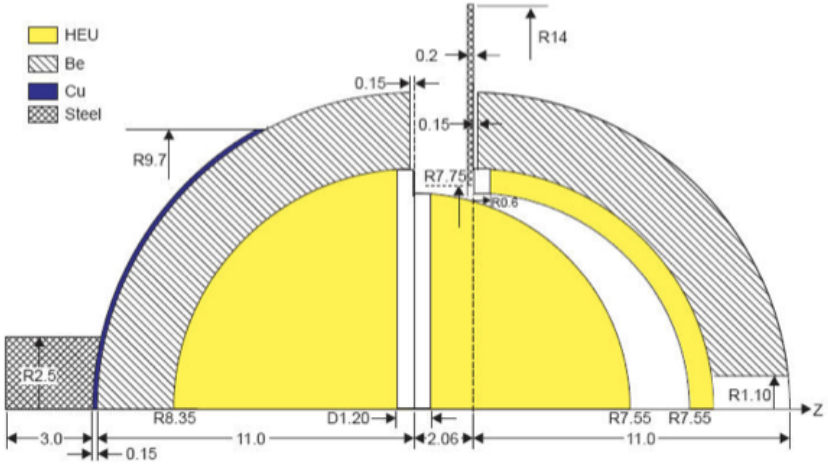
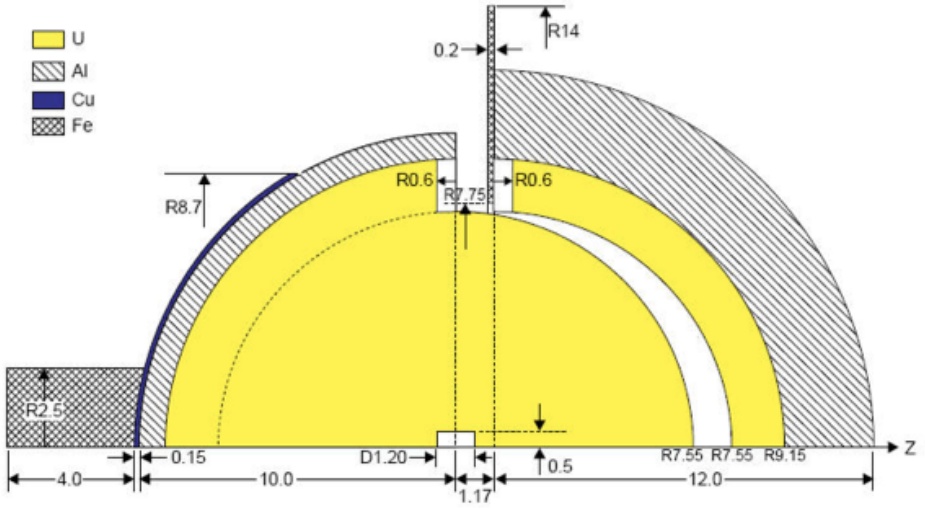


Figure 2. Incomplete HEU Sphere Reflected by Beryllium, heu-met-fast-009-case-1



e 5. Incomplete HEU Sphere Reflected by Aluminum, heu-met-fast-012

Nuclear Regulatory Commission Use of MCNP

Criticality Safety:

- To assess the criticality safety of licensed facilities that handle fissionable materials.

Radiation Shielding:

- To benchmark other shielding and dose calculation computer codes and methods used by NRC staff.
- To verify licensees' shielding and dosimetry calculations.

Radiation Dosimetry:

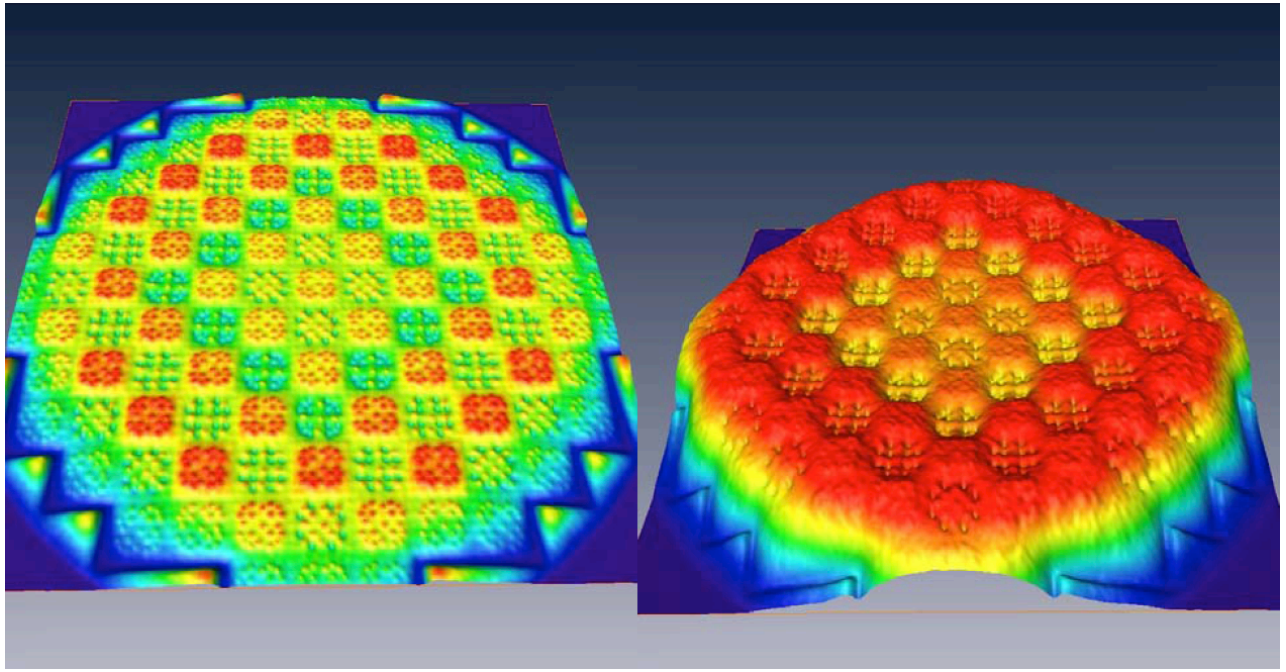
- Assess planned and unplanned worker radiation exposures.
- Assess public exposure from planned licensing actions.

Medical:

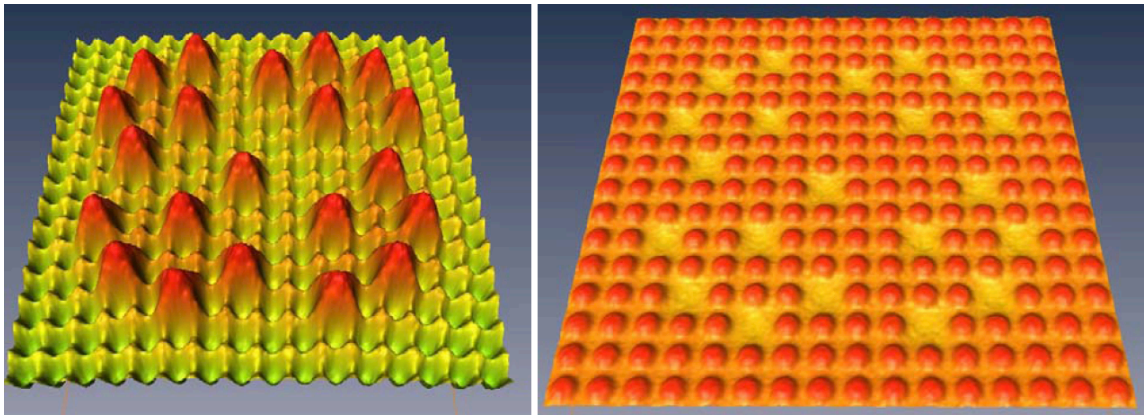
- To understand the radiation safety implications of using radiation in medical diagnosis and treatments.

MCNP & Reactors (1) - PWR Analysis

Whole-core Thermal & Fast Flux from MCNP5 Analysis



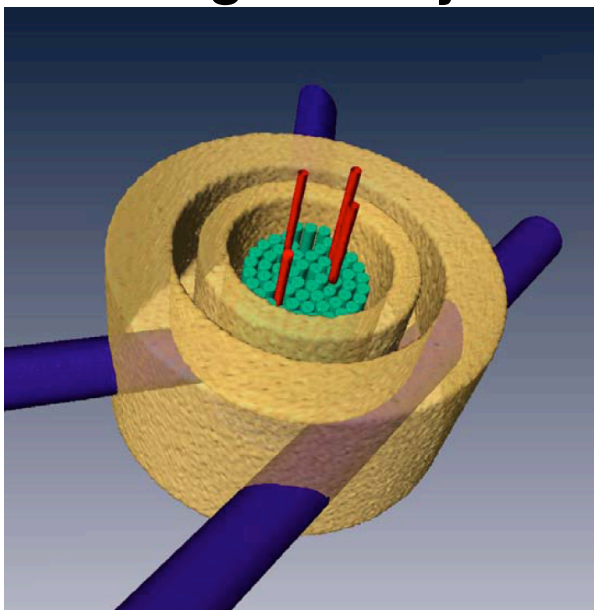
Assembly Thermal & Fast Flux from MCNP5 Analysis



(from Luka Snoj, Jozef Stefan Inst.)

MCNP & Reactors (2) - TRIGA Reactor LEU Conversion

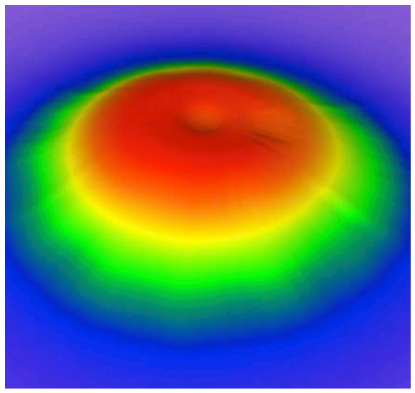
3D geometry

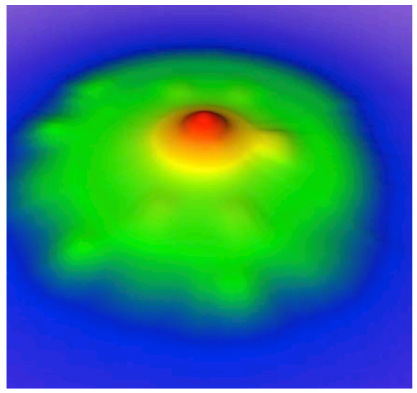


Fast Flux

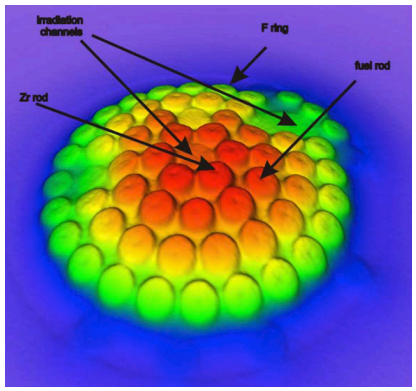
Thermal Flux

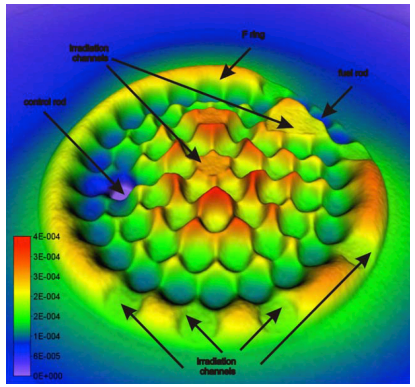
Diffusion Theory Codes

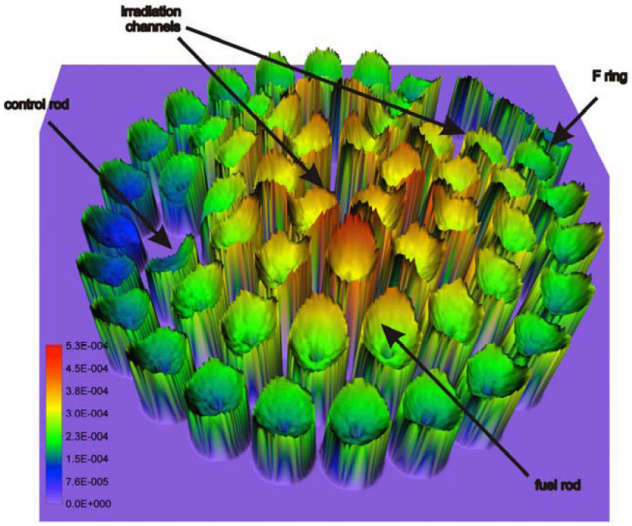




MCNP Analysis



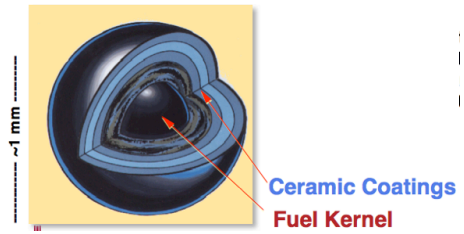




Radial Power Density From MCNP Analysis

(from Luka Snoj, Jozef Stefan Inst.)

MCNP & Reactors (3) - Advanced Reactor Design

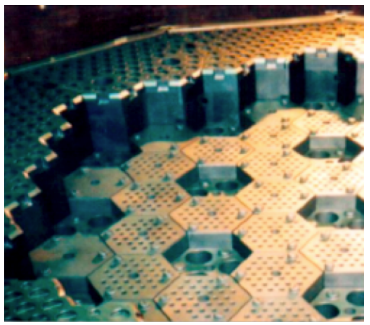


Very High Temperature Reactor – VHTR

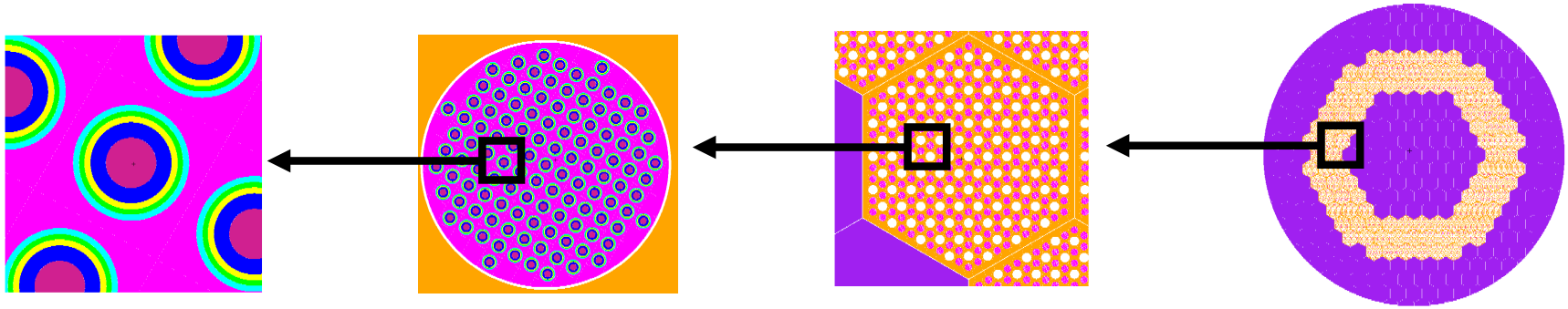
TRISO FUEL PARTICLES

FUEL COMPACTS

FUEL BLOCK

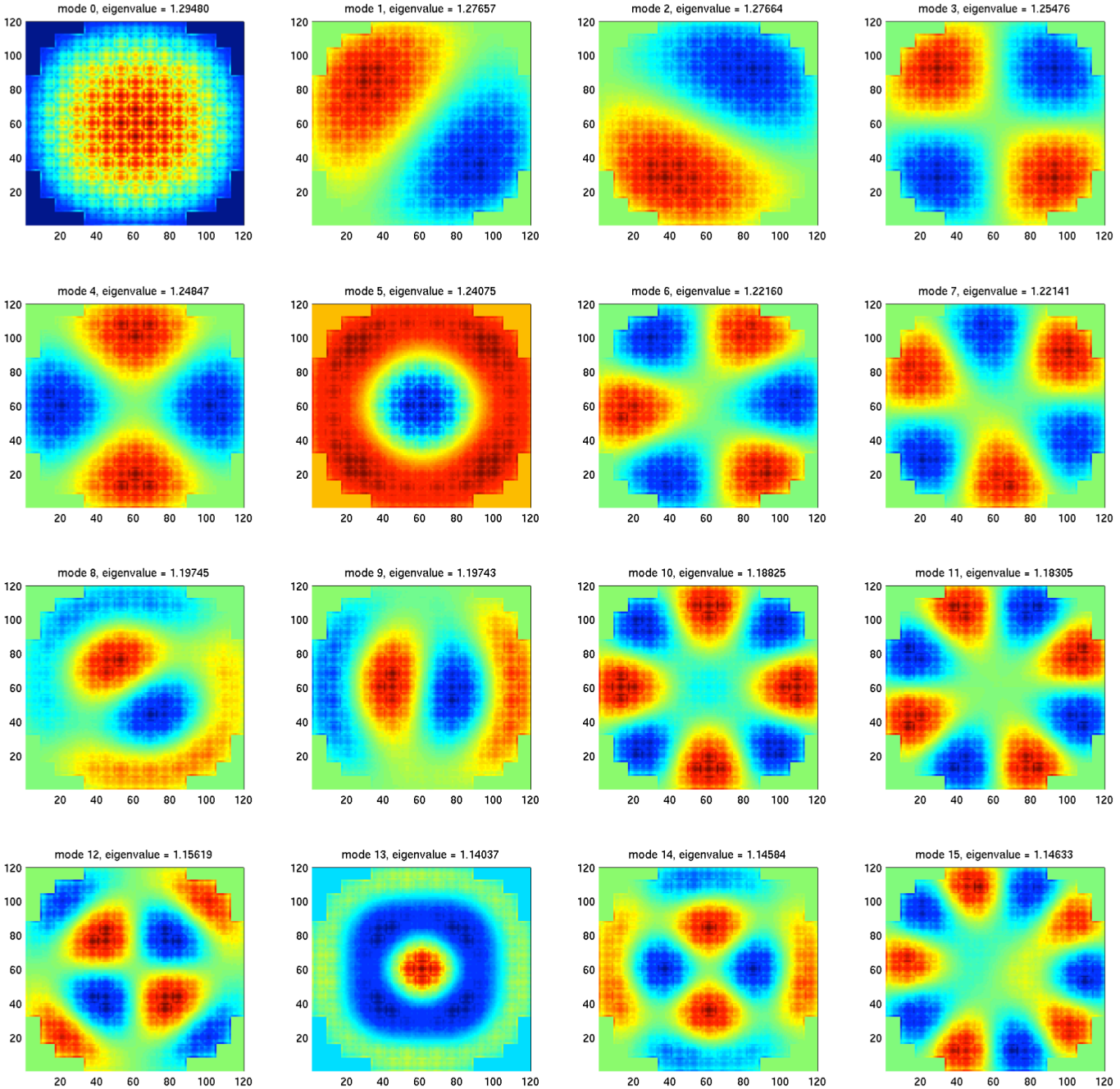


CORE



MCNP model - accurate & explicit at multiple levels

MCNP & Reactors (4) – Higher Eigenmode Analysis

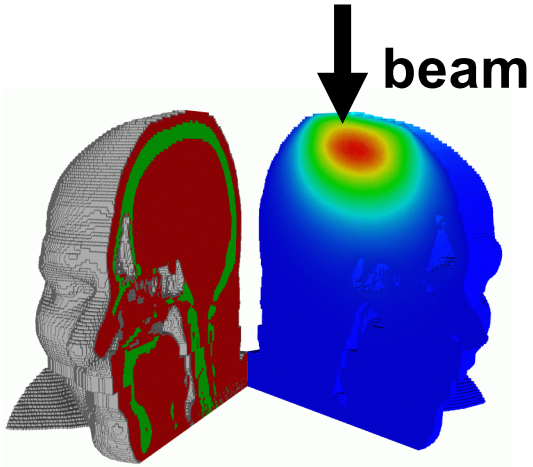


Higher eigenmodes for PWR

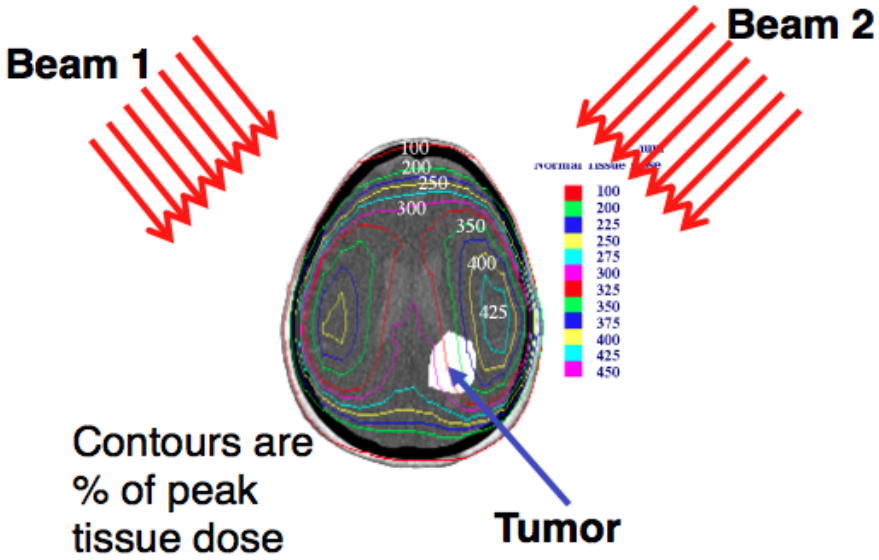
n	K_n
0	1.29480
1	1.27664
2	1.27657
3	1.25476
4	1.24847
5	1.24075
6	1.22160
7	1.22141
8	1.19745
9	1.19743
10	1.18825
11	1.18305
12	1.15619
13	1.14633
14	1.14617
15	1.14584

Medical Physics (1) - Treatment Planning

- **Calculate dose distributions** for different beam orientations.
- Post-process to combine different beams to **maximize dose to tumor & minimize dose to healthy tissue**.
- While the peak tumor and tissue dose are usually based on in-phantom dose rate measurements, the simulation is necessary to determine more advanced parameters, such as the dose volume histogram.

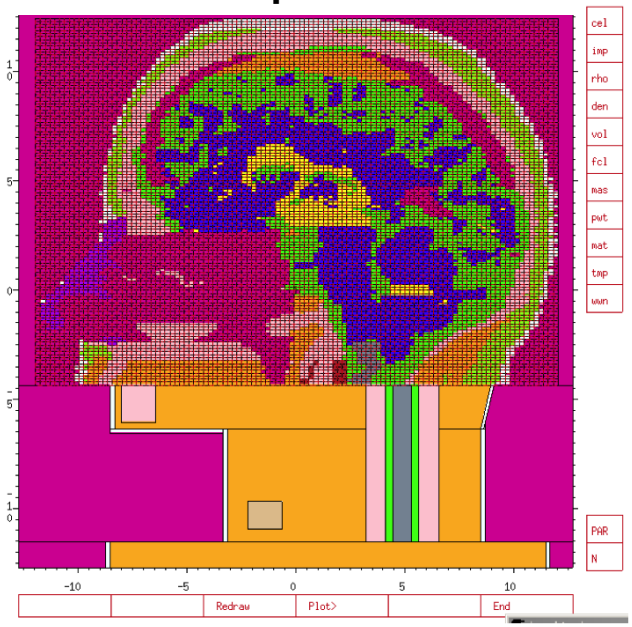


CT Based geometries are possible to represent in MCNP

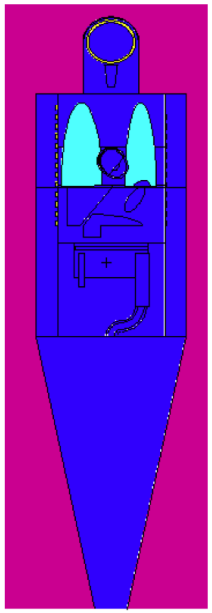


Medical Physics (2) – Phantoms & Voxel Models

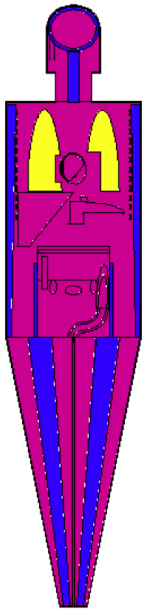
Zubal phantom



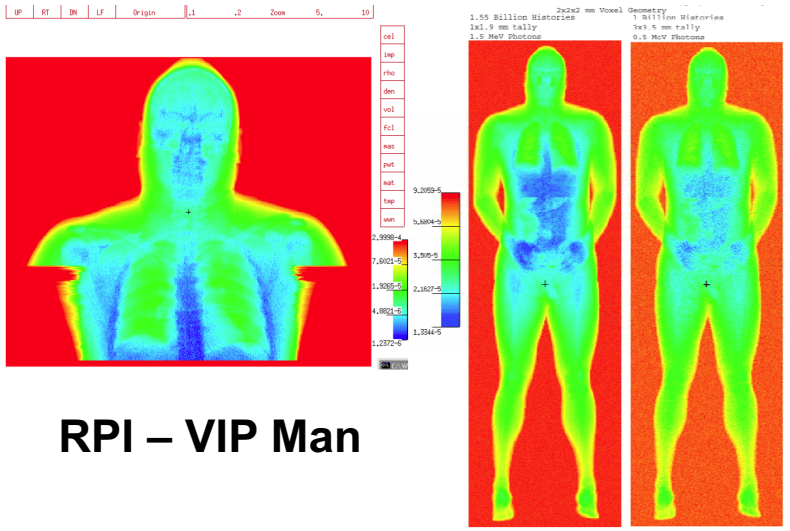
Yanch, MIT



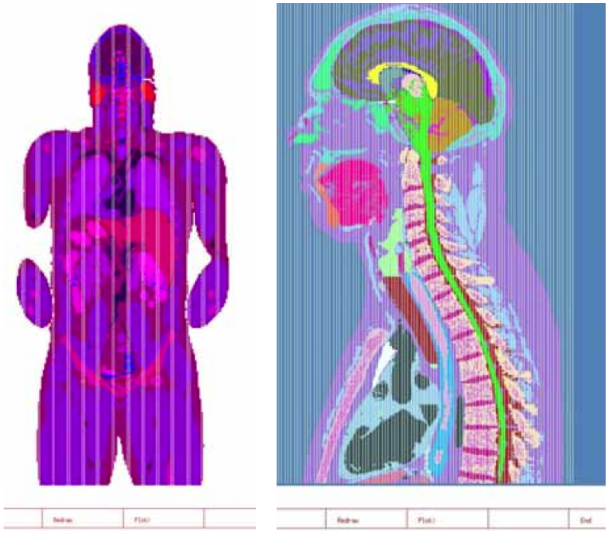
ORNL



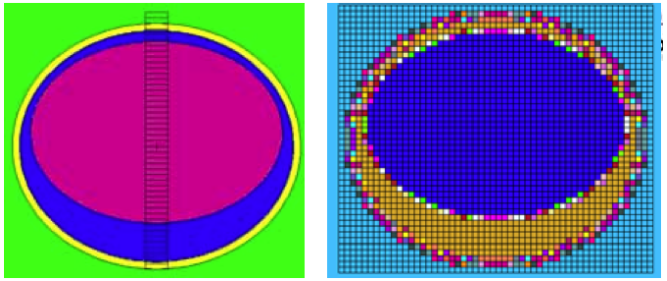
RPI – VIP Man



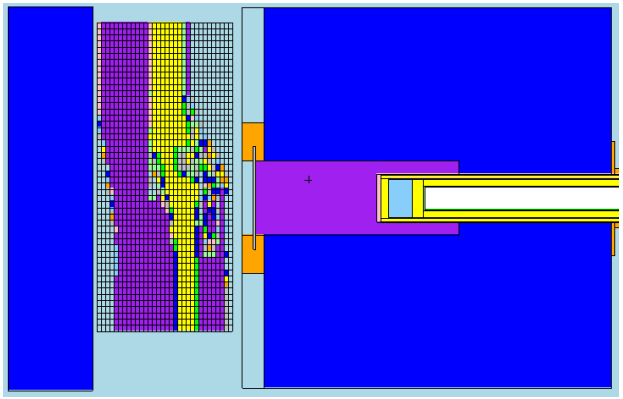
VIP Man



Snyder head phantom

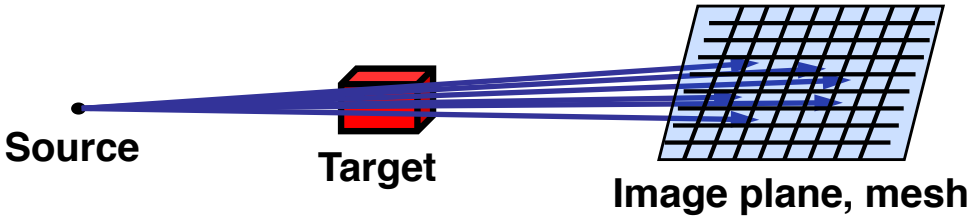


Boron-Neutron Therapy



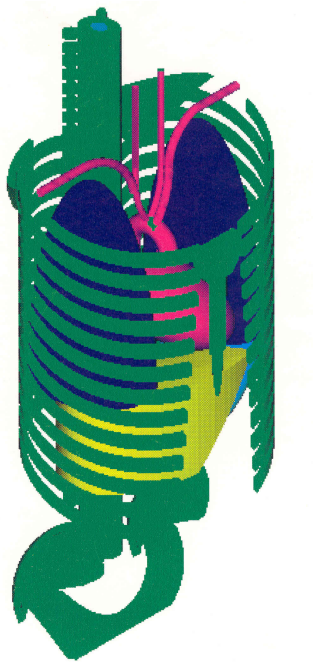
MCNP Radiography Calculations (1)

- **Radiography tallies**

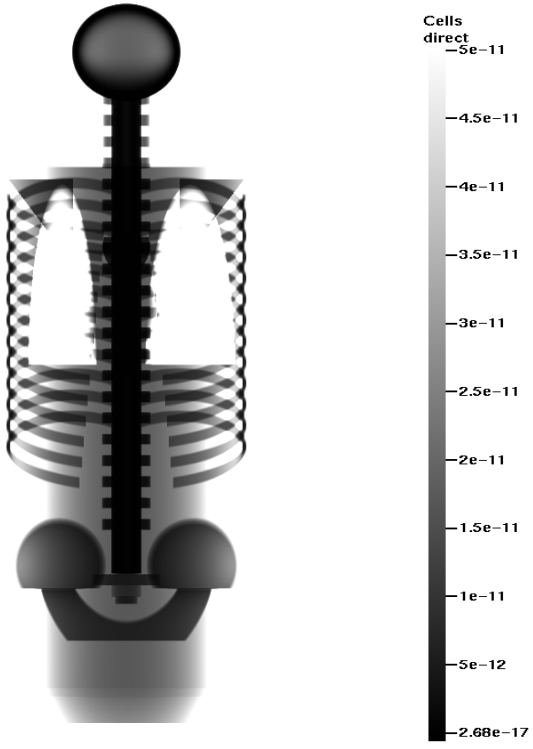


- **Neutron and photon radiography uses a grid of point detectors (pixels)**
- **Each source and collision event contributes to all pixels**

MCNP Model of Human Torso



Simulated Radiograph - 1 M pixels



MCNP Radiography Calculations (2) – Proton Radiography

- Experiments at LANL & BNL use high-energy proton beams directed at test objects

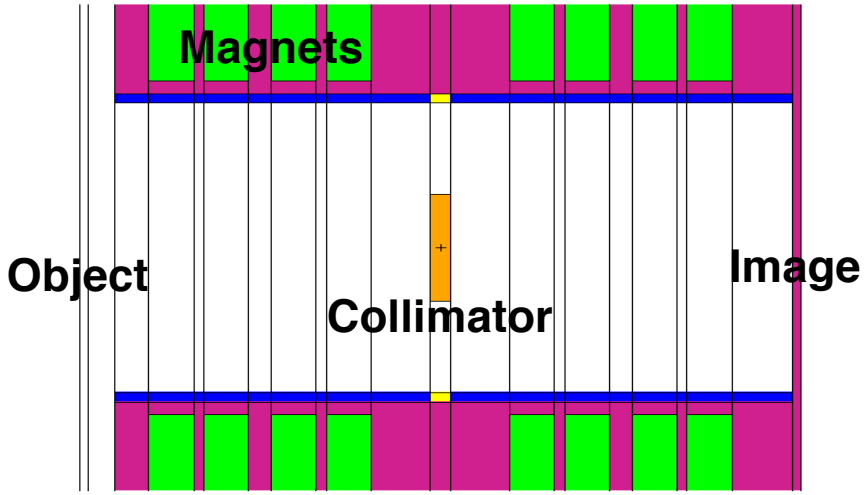
  LANL: 800 MeV protons
  BNL: 24 GeV protons
  Proposed: 50 GeV protons

- Proton beams are collimated & focused by magnetic lenses

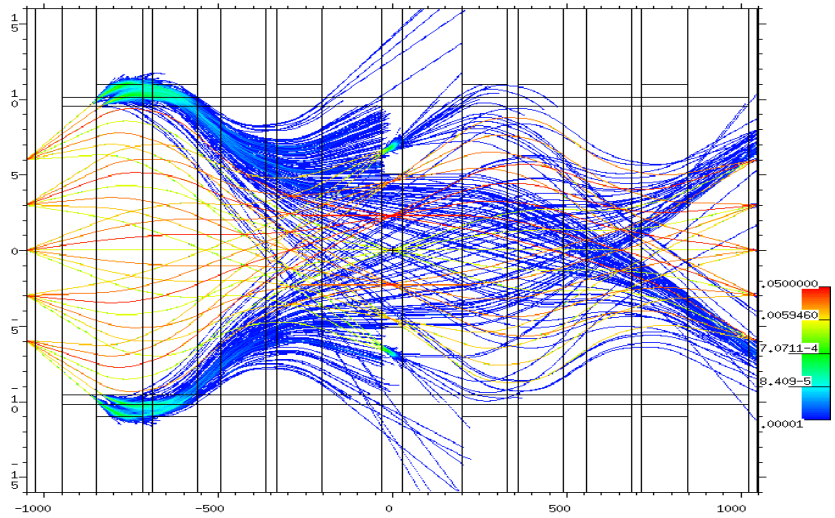
- Radiography tallies simulate pixels from detectors

- Experiment design & analysis are modeled with MCNP6

LANSCE pRad, MCNP6 calculations

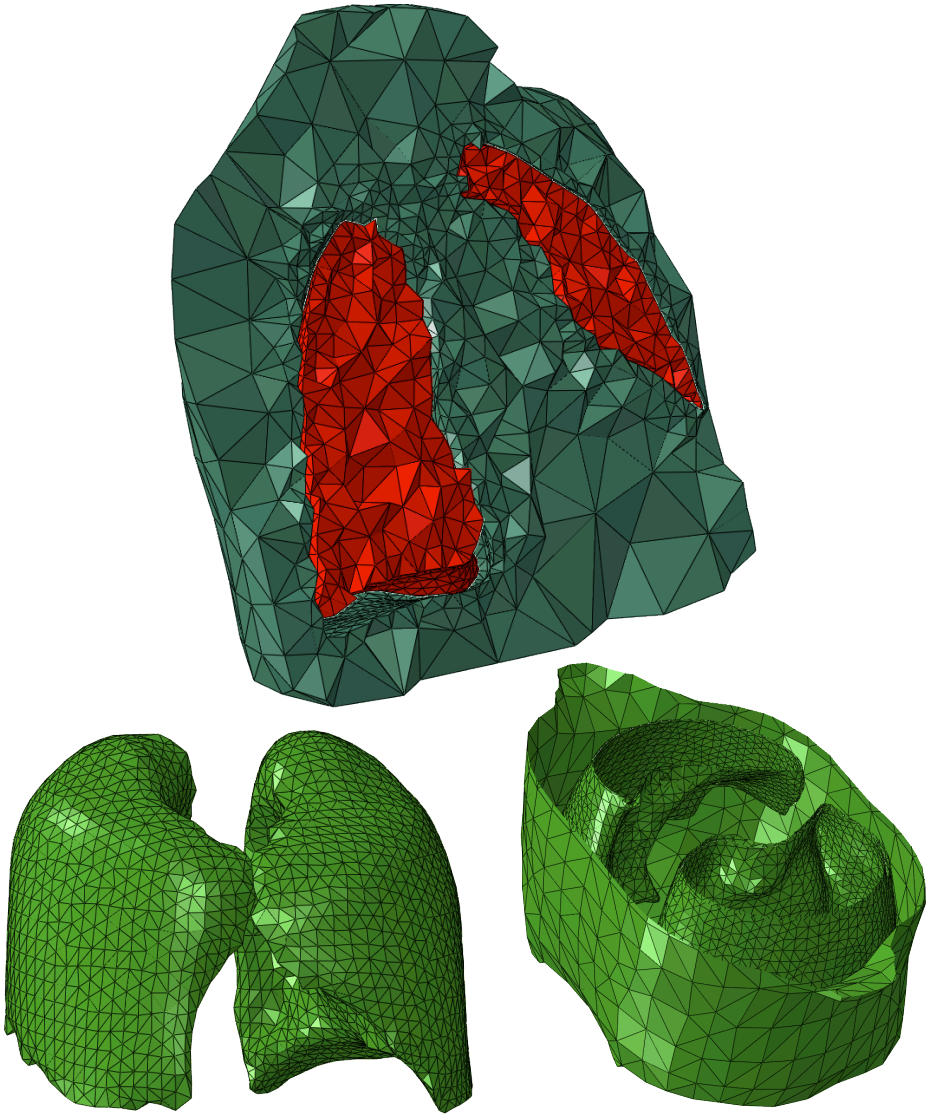


Horizontal axis - 0, 3, 6, and 9 mrads angles

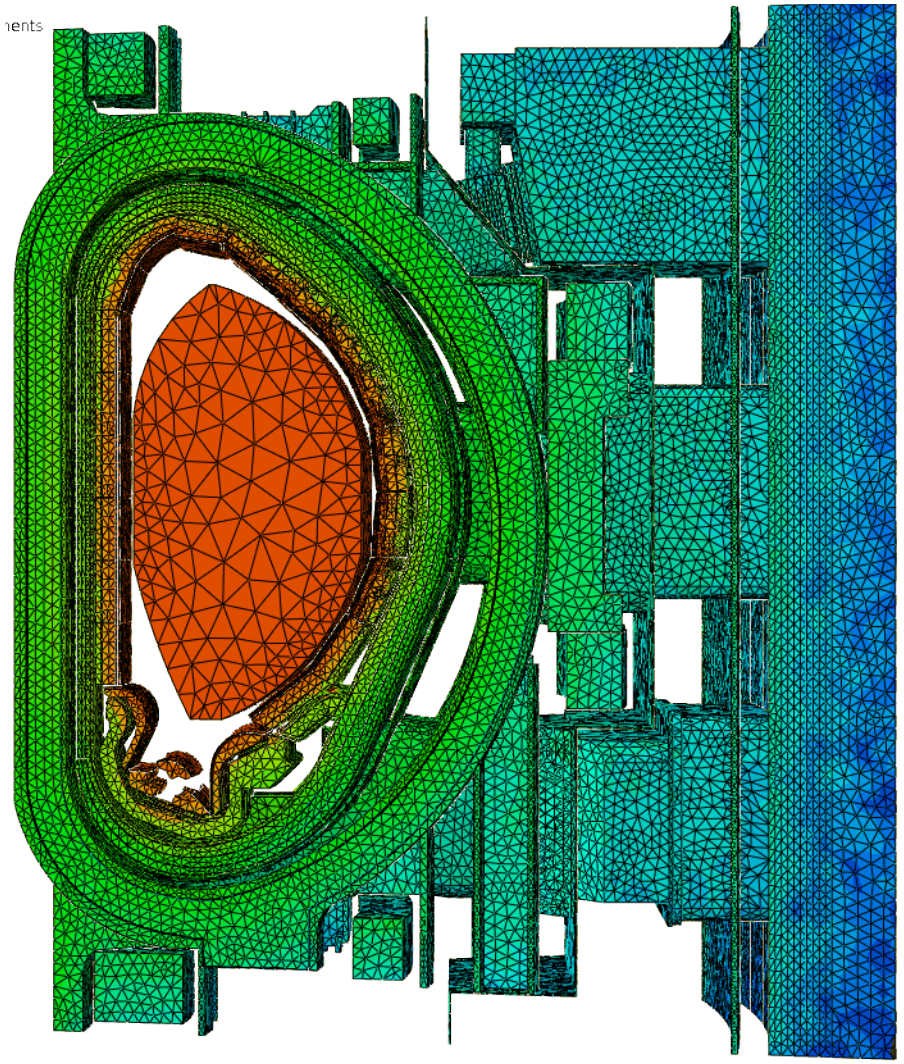


MCNP – Unstructured Mesh Geometry

Medical Physics
Radiation treatment planning

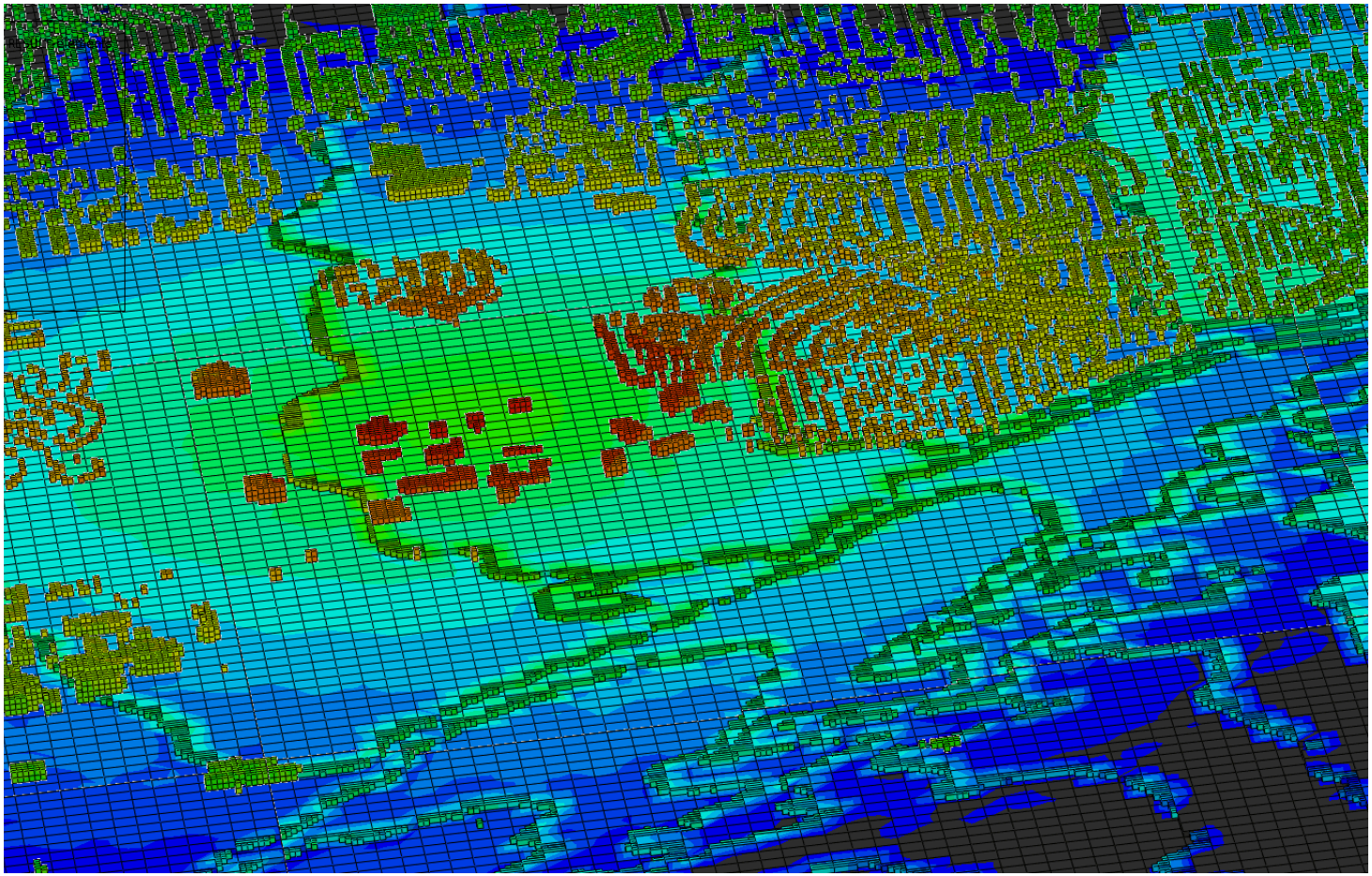


ITER Fusion Project
Neutron Flux Calculations



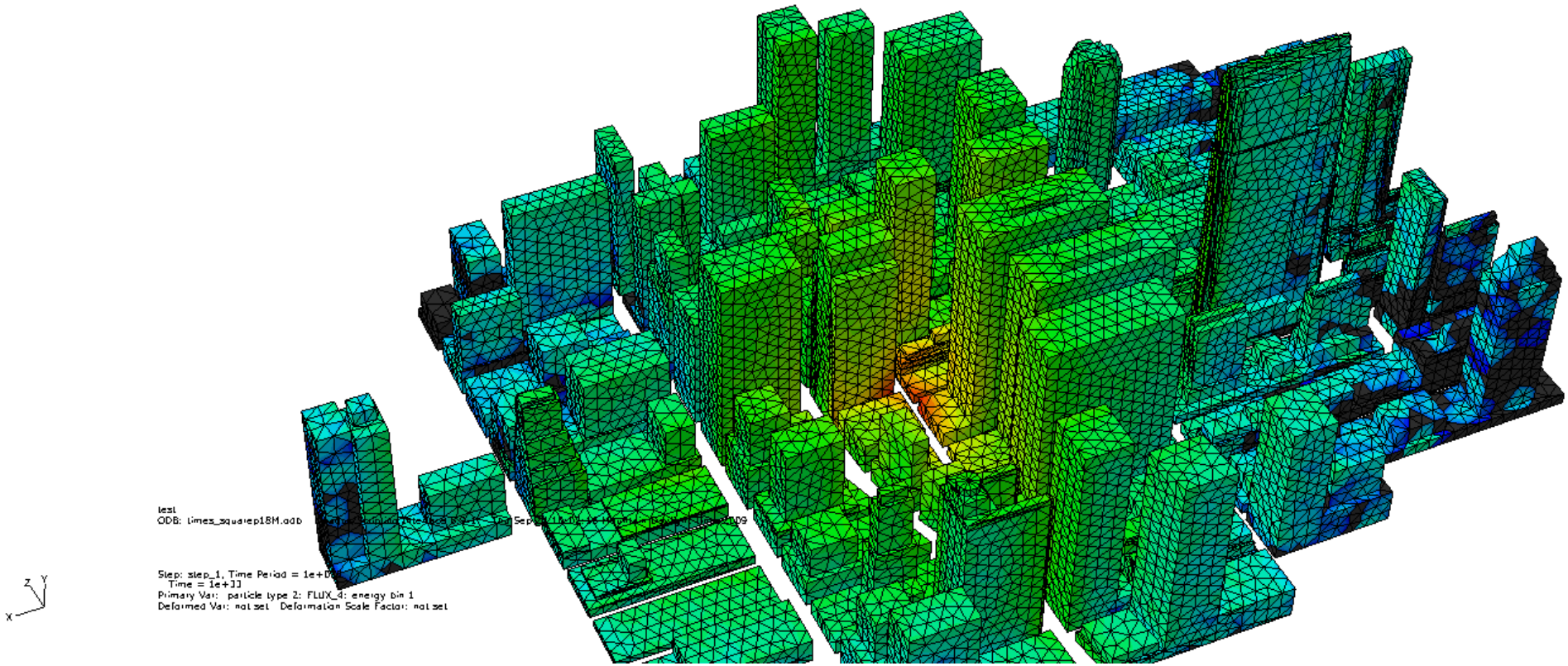
Kirtland, AFB, NM

Gamma flux from an above ground “Fat Man-like” explosion

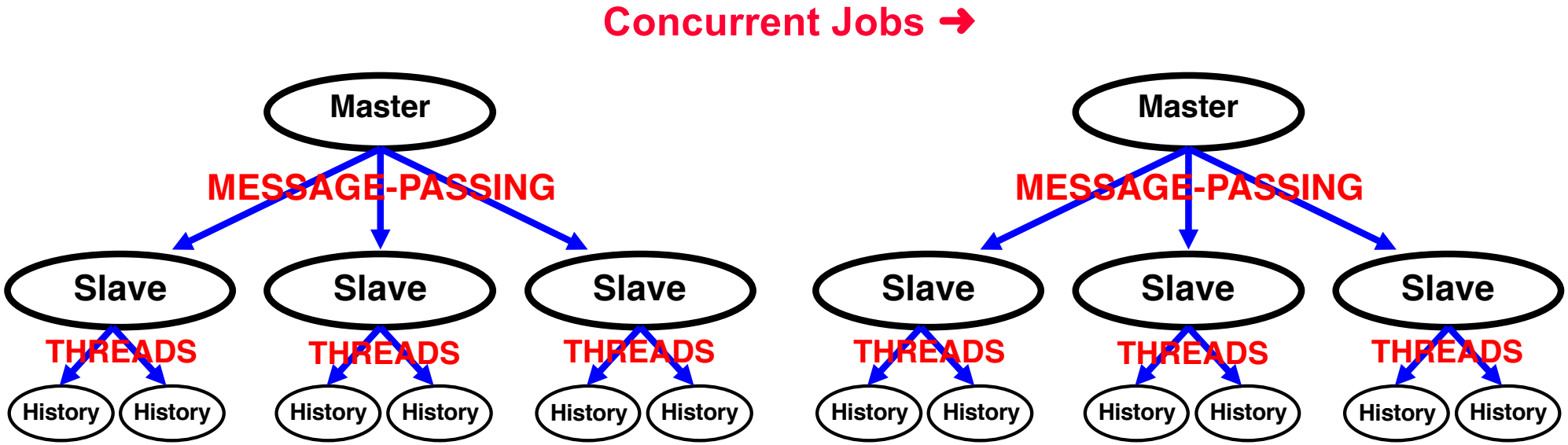


Times Square, New York City

Gamma flux from a “Fat Man-like” explosion



MCNP - Hierarchical Parallelism

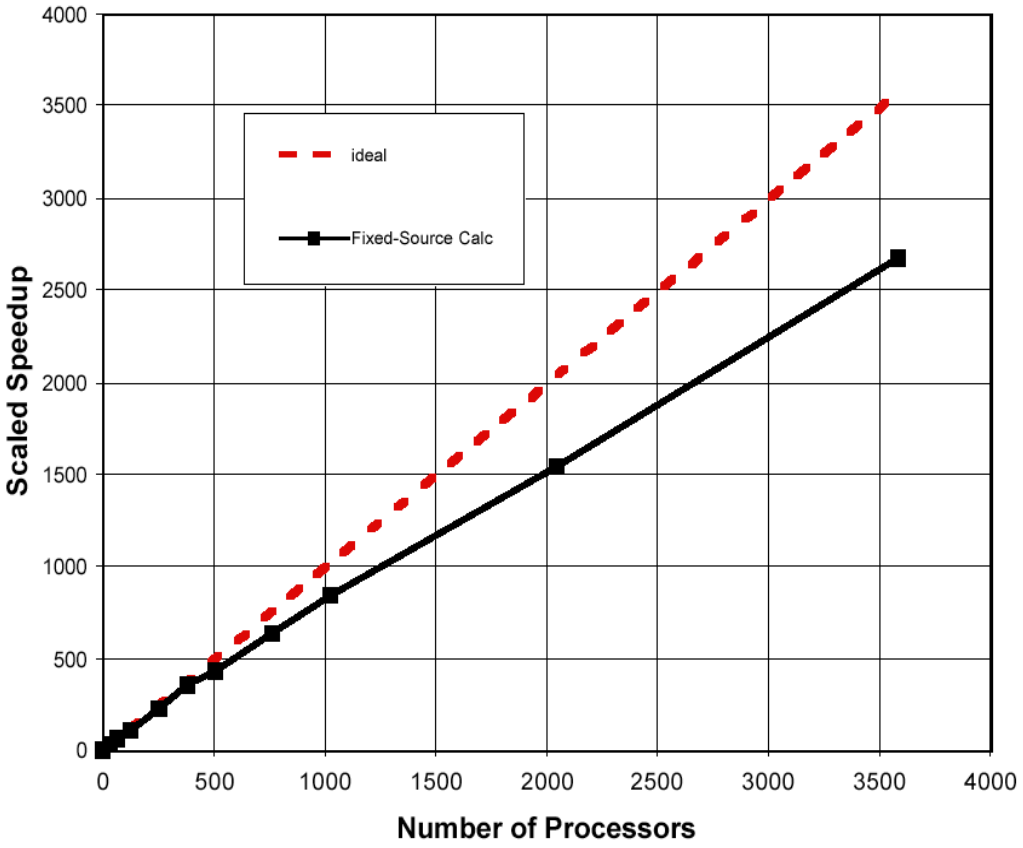


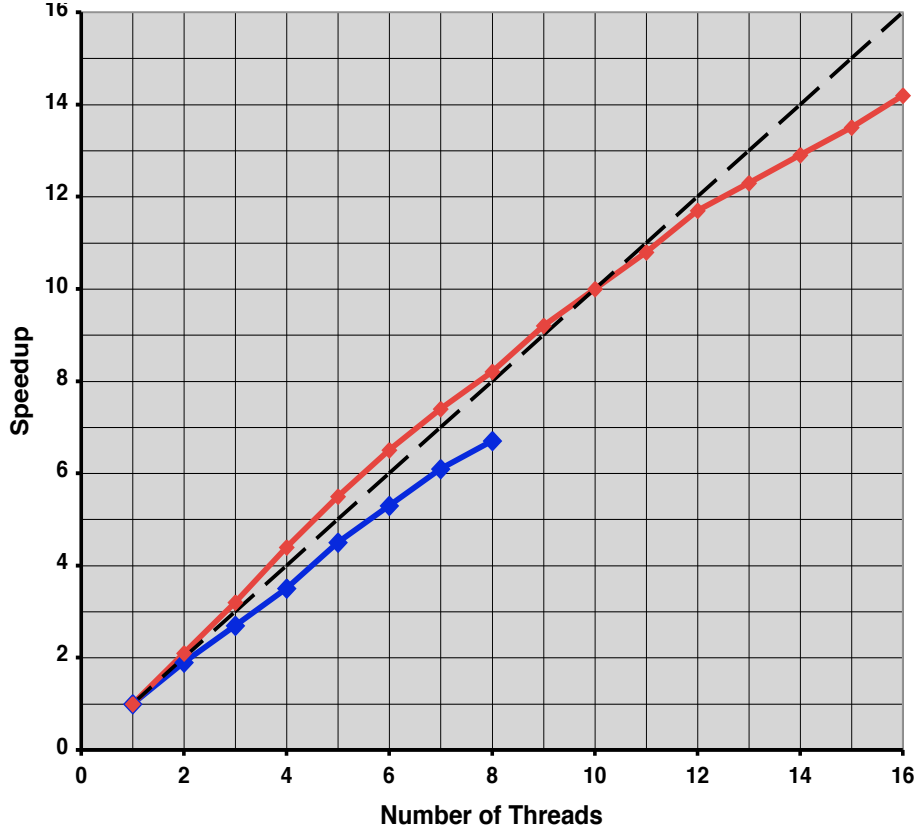
Parallel Processes

- Total processes = (# jobs) x (# MPI processes) x (# threads)
- Tradeoffs:
  - More MPI processes - lots more memory & messages
  - More threads - contention from lock/unlock shared memory
  - More jobs - system complexity, combining results

MCNP - Hierarchical Parallelism

- Use MPI to distribute work among slaves ("boxes")
- Use threading to distribute histories among individual cpus on box





ASCI Q, MPI+OpenMP,
4 threads/MPI-task

Lobo - 4 x Quad-core, 16 threads/node
Mac Pro - 2 x Quad-core, 8 threads/node

MCNP Distribution

- **Latest release from RSICC - codes, data, manuals (1 package)**
  - MCNP5 - 1.60: Sept 2010
  - MCNPX - 2.7.0: May 2011
  - ENDF/B-VII.1 + older data libraries
  - MCNP6.1: June 2013
  - MCNP6.1.1: July 2014
  - MCNP6.2: (2017)

- **Code distribution center:**
  - Radiation Safety Information Computational Center, Oak Ridge, TN
    www-rsicc.ornl.gov

- **Help:**
  - Read the manual
  - User forums: mcnp-forum@lanl.gov
  - MCNP home page: mcnp.lanl.gov
  - MCNP Developers (limited): mcnp6@lanl.gov

MCNP Development Team

Monte Carlo Development, XCP-3 & NEN-5

Avneet Sood (GL)
Jennifer Alwin
Art Forster
Gregg McKinney
Tony Zukaitis
Chris Werner (DGL)
Forrest Brown
Grady Hughes
Mike Rising
Jerawan Armstrong
Jeffrey Bull
Roger Martz
CJ Solomon

Computer Support: Laura Casswell

Data Team, XCP-5 Kent Parsons, Jeremy Conlin, Morgan White, Beth Lee

NJOY, T-2 Skip Kahler

University R&D William Martin, Leslie Kerby, Anil Prinja

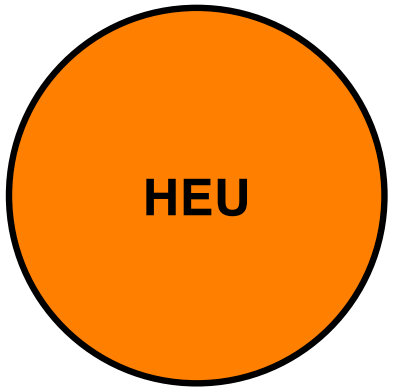
Example

Godiva Critical Experiment

Godiva Critical Experiment (1)

- Simplified model of bare HEU sphere
- Sphere radius = 8.741 cm
- HEU density = 18.74 g/cm3
- Nuclide ZAID Mass-fraction

Nuclide	ZAID	Mass-fraction
U235	92235	94.73 %
U238	92238	5.27 %



ZAID = Z * 1000 + A

MCNP units: g, cm

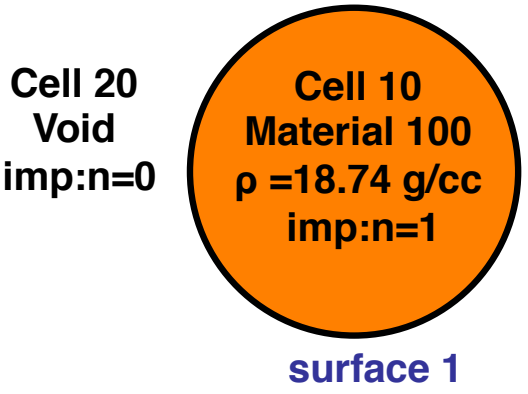
MCNP material density: + = atom density, atoms / barn-cm
- = mass density, g / cm^3

MCNP nuclide fractions in a material: + = atom fraction
- = mass fraction

1 barn = 10^{-24} cm^2

Godiva Critical Experiment (2)

Create & edit file g1.txt,
or copy it from C:\MCNP\SOLUTIONS folder




```
Godiva critical                          $ Title
c   CELL CARDS                           $ Comment
10    100 -18.74   -1    imp:n=1         $ Density in g/cc
20     0           +1    imp:n=0

                                         $ NOTE: 1 blank line
c   SURFACE CARDS
1  so    8.741

                                         $ NOTE: 1 blank line
c   DATA CARDS
kcode   1000    1.0    10    50
ksrc    0.0   0.0   0.0
m100    92235   -0.9473      $ Material- with mass fractions
        92238   -0.0527
```

Godiva Critical Experiment (3)

```
Godiva critical
c   CELL CARDS
10    100 -18.74    -1    imp:n=1
20     0            +1    imp:n=0

c   SURFACE CARDS
1   so    8.741

c   DATA CARDS
kcode   1000   1.0   10   50
ksrc    0.0   0.0   0.0
m100    92235   -0.9473
        92238   -0.0527
```

Indent 5 or more spaces for continuing a card

- **Cell 10**
  - Contains material 100 at a density of 18.74 g/cc
  - See m100 card for material specification
  - Boundary surface list contains only one surface: -1 (inside of 1)
  - Importance = 1, so track neutrons normally
- **Cell 20**
  - Contains void (material 0), no density needed
  - Boundary surface list contains only one surface: +1 (outside 1)
  - Importance = 0, so kill any neutrons reaching this cell
- **kcode & ksrc cards**
  - Starting guess: neutrons at center
  - 1000 neutrons/cycle, guess k=1, discard 10 cycles, run 50 total

Godiva Critical Experiment (4)

- Save file g1.txt in a working folder, C:\MCNP\WORK
- Open the MCNP command window

```
                          . . . Go to the folder with  g1.txt
cd    C:\MCNP\WORK
dir
                          . . . Should see g1.txt
mcnp6    i=g1.txt
                          . . . Lots of stuff to the screen
dir
                          . . . See what files are created
                          . . . Look at file outp with text editor
```

- Do all of the above again
- Cleanup

```
del   out?   srct?   runtp?
```

Godiva Critical Experiment (5)

Screen output

```
 mcnp     ver=6    , ld=06/02/14  07/03/14 14:16:42
          Code Name & Version = MCNP6_DEVEL, 6-1-02
          Copyright LANS/LANL/DOE - see output file

     _/        _/        _/_/_/       _/      _/       _/_/_/         _/_/_/
    _/_/    _/_/      _/             _/_/    _/       _/     _/      _/
   _/  _/  _/ _/     _/             _/  _/  _/       _/_/_/       _/_/_/
  _/    _/    _/     _/           _/    _/_/       _/            _/     _/
 _/      _/      _/_/_/         _/      _/       _/              _/_/

  warning.  Physics models disabled.
  comment.  total fission nubar data are being used.
comment. using random number generator  1, initial seed = 19073486328125
imcn    is done

ctm =         0.00    nrn =                    0
dump     1 on file runtpe    nps =             0    coll =                   0

 source distribution written to file srctp         cycle=     0
xact    is done

cp0 =    0.04
```

Godiva Critical Experiment (6)

```
comment.
comment. entropy of the fission source distribution will be computed
comment.
comment. the mesh for source entropy is based on the site coordinates
comment.   using  4 x  4 x  4 =    64  mesh cells
comment.
comment.     Xmin= -1.1321E+01     Xmax=  1.1351E+01
comment.     Ymin= -1.1567E+01     Ymax=  1.1732E+01
comment.     Zmin= -1.1973E+01     Zmax=  1.1442E+01
comment.
comment. the mesh will be automatically expanded if necessary to
comment.   encompass the entire fission source distribution
comment.

cycle    k(col)        ctm   entropy    active     k(col)    std dev         fom
    1   1.37485      0.00  3.77E+00
    2   1.14104      0.00  4.11E+00
 . . . . .
   10   1.03116      0.01  4.33E+00
********************  begin active keff cycles  **********************
   11   1.02102      0.02  4.35E+00
   12   0.96299      0.02  4.34E+00         2    0.99201    0.02902     405471
. . . . .
   49   0.97984      0.07  4.32E+00        39    1.00218    0.00372    1337790
   50   1.01116      0.07  4.42E+00        40    1.00241    0.00363    1371357
```

Godiva Critical Experiment (7)

```
source  distribution written to file srctp        cycle=    50
     run terminated when      50 kcode cycles were done.

=====>     576.69 M neutrons/hr    (based on wall-clock time in mcrun)

comment.
comment. Average fission-source entropy for the last half of cycles:
comment.      H=  4.37E+00  with population std.dev.=  4.88E-02
comment.
comment. Cycle    3 is the first cycle having fission-source
comment.   entropy within 1 std.dev. of the average
comment.   entropy for the last half of cycles.
comment.   At least this many cycles should be discarded.
comment.
comment. Source entropy convergence check passed.
comment.

final k(col/abs/trk len) = 1.00341     std dev = 0.00327

ctm =        0.07   nrn =           2589160
dump    2 on file runtpe   nps =       50095   coll =           163564
mcrun  is done
```

MCNP Basics

Input Files
Cell, Surface, & Data Cards
Basic Geometry

Random Number Seeds
Continue Runs

Introduction to Advanced Geometry

MCNP Input File (1)

Title Line ... (required)

Cell Cards ...

blank line separator

Surface Cards ...

blank line separator

Data Cards ...

blank line separator (optional)

... these lines are ignored - useful for notes or saving options

- **Card names begin in first 5 columns**
- **80 columns or fewer**
- **Free field format**
- **Not case sensitive: UC, lc, MiXeD**
- **Continuation: 5 leading blanks or &**
- **Comment cards begin with "c "**
- **In-line comments begin with $**
- **Use spaces or tabs** (tab stops – 8 columns)
- **For most numbers, these are the same:**

  1 1. 1.0 1e0 1e+00 1.0e+0

- **Units**

Length:	**cm**
Mass:	**g**
Energy & Temp.:	**MeV**
Number density:	**atoms/barn-cm**
Time:	**shakes**

(1 barn = 10^{-24} cm^2, 1 sh = 10^{-8} sec)

MCNP Input File (2)

```
Godiva critical - using ksrc & surfaces      $ Title
c  CELL CARDS                                 $ Comment
10   100     -18.74      -1                   $ Cells
20    0                   1

c  SURFACE CARDS
1  so    8.741

c  DATA CARDS
kcode    1000    1.0    10    50
ksrc   0.0   0.0   0.0
imp:n   1    0
m100    92235   -94.73
        92238   -5.27
```

Indent 5 or more spaces for continuation

MCNP Input File (3)

- **Highly recommended convention, to aid in setup & debugging**

Use different numbers of digits for cells, surfaces, materials

	Problem Size			
	small	medium	large	huge
Surfaces:	1	12	123	1234
Cells:	12	123	1234	12345
Materials:	123	1234	12345	123456

Max number of digits for cells, surfaces, materials: 8

MCNP Cells (1)

- **Cells are the basic geometry unit**
  - Volume of space bounded by surfaces
  - Cartesian coordinate system
  - Volumes calculated for some simple cells, not for complicated ones
- **Cells are used for:**
  - constructing the model
  - specifying the materials
  - variance reduction methods
  - performing tallies
- **All of space must be defined**
  - Every xyz point will lie either on a surface or within a uniquely defined cell.
  - No gaps or overlaps for cells
  - At least one cell will describe the problem exterior (outside world)
- **Repeated structure and lattice ability**
  - Cells may contain embedded geometry - lattice or repeated structure

MCNP Cells (2)

Cell #	Mat #	Density	Surface List	Cell Data (Optional)
30	300	9.65e-2	1 -2 3 -4 5 -6	
		Positive Density ➜ atoms/barn-cm		
10	300	-1.0	1 -2 3 -4 5 -6	imp:n=1.0
		Negative Density ➜ g/cm^3		
20	0		-7:8: -9	
		Voids ➜ Material # = 0, omit Density		

Material Cards (1)

Mn zaid1 fraction1 zaid2 fraction2 …..

n = material number

zaid = element or nuclide identifier: ZZZAAA

ZZZ = atomic number

AAA = atomic mass

Examples: ^{235}U ➜ 92235 ^{16}O ➜ 8016 C ➜ 6000

fractions: positive = atom fraction of zaid

negative = mass fraction of zaid

- MCNP normalizes the fractions for a material to sum up to 1.0
- Density (g/cc or atoms/barn-cm) comes from the cell cards

- In ENDF/B-VII.1, only 1 element: C 6000
- For list of isotopes, see LA-UR-13-21822, "Listing of Available ACE Files"

Material Cards (2)

- **Cell & material cards should be consistent**
  - **Overall material density comes from the cell card where a material is used**
    - g/cc or atoms/barn-cm
  - **Mass fractions or atom fractions are on the material card**
    - Fractions on a material card are normalized to sum to 1.0

- **Examples**

```
Water
m10   1001   2
      8016   1

HEU
m20   92235   -.93
      92238   -.07

HEU
m30   92235   0.930823
      92238   0.069177
```

```
STAINLESS-STEEL
c        Cr number density = 1.6540e-2
c        Fe number density = 6.3310e-2
c        Ni number density = 6.5100e-3
c     total number density = 8.6360e-2
c
c
m200   24050   7.1866e-4     $ Cr-50   4.345%
       24052   1.3859e-2     $ Cr-52  83.789%
       24053   1.5715e-3     $ Cr-53   9.501%
       24054   3.9117e-4     $ Cr-54   2.365%
       26054   3.7005e-3     $ Fe-54   5.845%
       26056   5.8090e-2     $ Fe-56  91.754%
       26057   1.3415e-3     $ Fe-57   2.119%
       26058   1.7853e-4     $ Fe-58   0.282%
       28058   4.4318e-3     $ Ni-58  68.0769%
       28060   1.7071e-3     $ Ni-60  26.2231%
       28061   7.4207e-5     $ Ni-61   1.1399%
       28062   2.3661e-4     $ Ni-62   3.6345%
       28064   6.0256e-5     $ Ni-64   0.9256%
```

Importance Cards

- **Each cell must have an "importance" for each type of particle**
  - **Imp:n for neutrons, imp:p for photons, ….., for each cell**

Imp:n = 1 Track particle in the cell in normal manner

Imp:n = 0 Kill particles that enter the cell

Outside world cell usually **0**

Imp:n = any other value

Invokes splitting &/or Russian roulette

Used for variance reduction

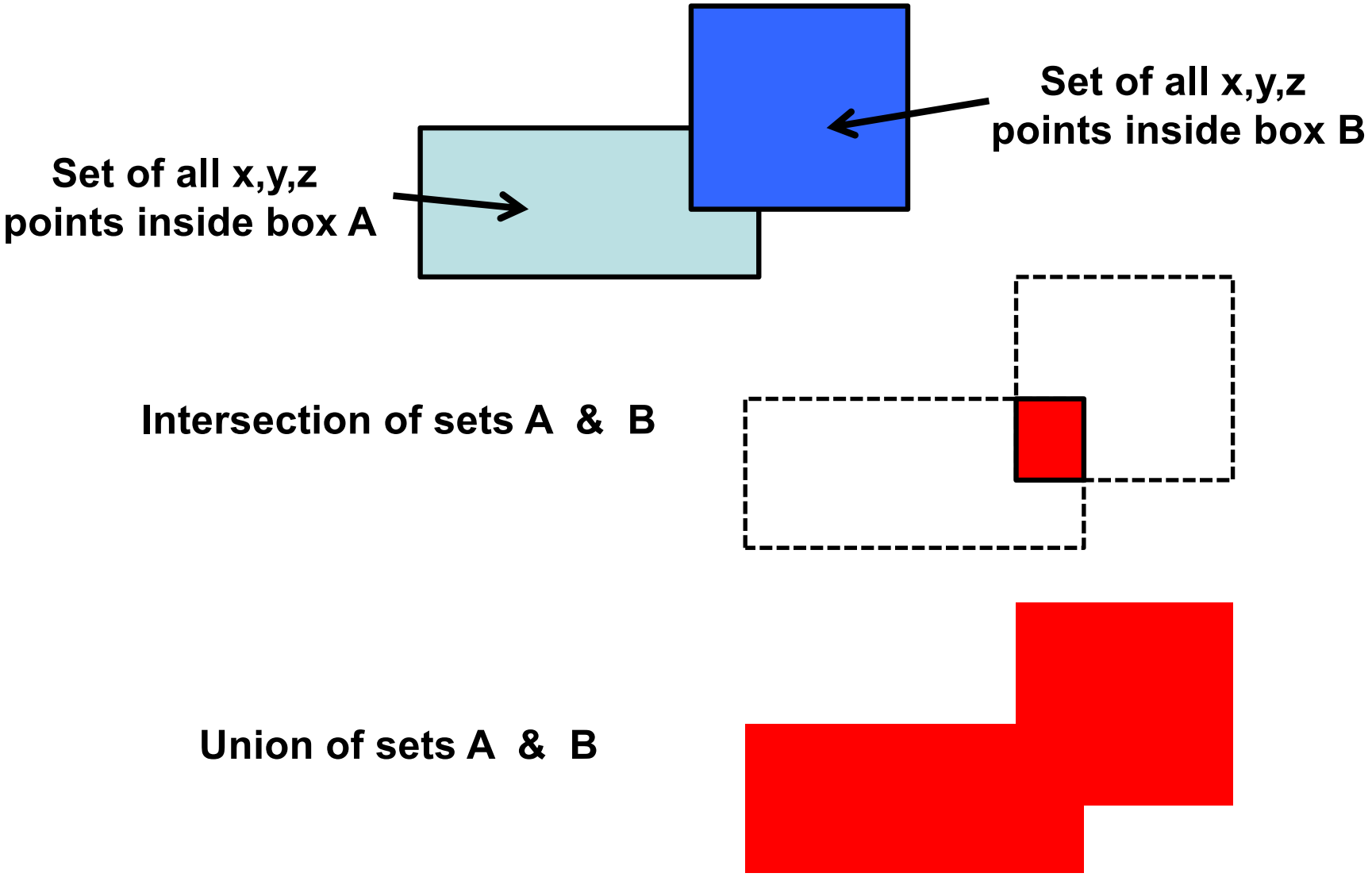
- **Can be in data card block or after surfaces on all cell cards**

```
Godiva critical
10   100 -18.74   -1
 9     0           1

1   so 8.741

kcode   1000 1.   10   50
ksrc    0.0  0.0  0.0
m100    92235  -94.73
        92238  -5.27
imp:n   1    0
```

```
Godiva critical
10   100 -18.74   -1   imp:n=1
 9     0           1   imp:n=0

1   so 8.741

kcode   1000 1.   10   50
ksrc    0.0  0.0  0.0
m100    92235  -94.73
        92238  -5.27
```

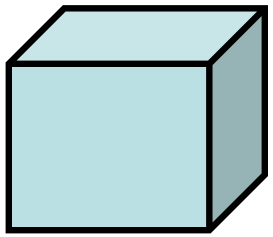
Comments on 3D Computational Geometry (1)

- **At the most fundamental level, 3D computational geometry is an exercise in Set Theory, the same concepts we all learned as kids**



Comments on 3D Computational Geometry (2)

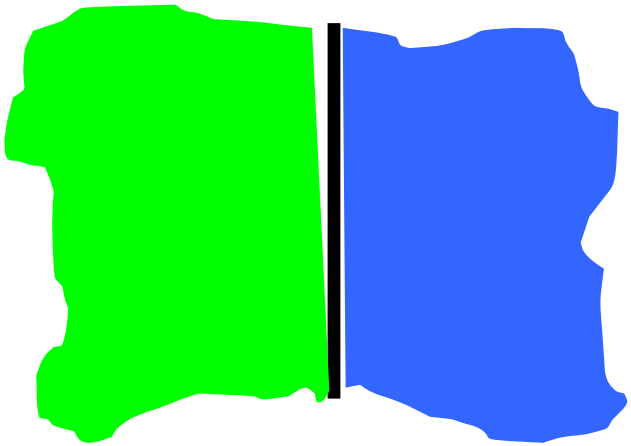
- Some codes use primitive bodies (box, sphere, etc.) in defining the sets of points to consider



Set of all x,y,z points inside box

- Other codes use half-spaces – all the points on one side of a surface

Set of all x,y,z points on minus side of surface



Set of all x,y,z points on plus side of surface

- A little thought should convince you that either scheme can be used in set theory for defining objects in space

MCNP Surfaces (1)

- Surfaces are used to define space
- Sign defines surface "sense"

  +3 ➜ half-space on + side of surface 3

  -5 ➜ half-space on - side of surface 5

- Boolean operators & parentheses
  - intersection space
  - union :
  - grouping ( )
- Cells are defined by intersections & unions of half-spaces
  - List of signed surfaces, spaces, colons, parentheses
  - Cell can also be defined as complement of another cell, using #cell

- 1^{st}, 2^{nd}, 4^{th} order equations (26):
  - planes
  - spheres, cylinders, cones
  - ellipsoid, hyperboloid, paraboloid
  - torus (elliptical or circular)
- Macrobodies
  - Primitive bodies - box, finite cylinder, hex, wedge, …
  - MCNP internally translates to collections of surfaces
- Can also specify surface by giving a few points (see manual)
- Special boundary surface types
  - reflecting (mirror) *10
  - white (isotropic) +10
  - Periodic see manual
- Some surface areas calculated

MCNP Surfaces (2)

Surface #	Name	Data
10	px	5.0
	plane normal to x-axis,	$x - D = 0$ data = D
50	so	11.1
	sphere at origin,	$x^2 + y^2 + z^2 - R^2 = 0$ data = R
30	rcc	-6.0 0.0 0.0 12.0 0.0 0.0 4.0
	right circular cylinder:	
	- center of base at (-6.0, 0, 0)	
	- 12.0–cm high can about x-axis	
	- radius 4.0	

MCNP Surfaces (3)

Table 3.1: MCNP Surface Cards

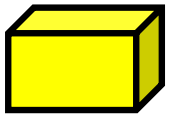
Mnemonic	Type	Description	Equation	Card Entries
P	Plane	General	$Ax + By + Cz - D = 0$	ABCD
PX		Normal to X-axis	$x - D = 0$	D
PY		Normal to Y-axis	$y - D = 0$	D
PZ		Normal to Z-axis	$z - D = 0$	D
SO	Sphere	Centered at Origin	$x^2 + y^2 + z^2 - R^2 = 0$	R
S		General	$(x-\bar{x})^2 + (y-\bar{y})^2 + (z-\bar{z})^2 - R^2 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ R$
SX		Centered on X-axis	$(x-\bar{x})^2 + y^2 + z^2 - R^2 = 0$	$\bar{x}\ R$
SY		Centered on Y-axis	$x^2 + (y-\bar{y})^2 + z^2 - R^2 = 0$	$\bar{y}\ R$
SZ		Centered on Z-axis	$y^2 + y^2 + (z-\bar{z})^2 - R^2 = 0$	$\bar{z}\ R$
C/X	Cylinder	Parallel to X-axis	$(y-\bar{y})^2 + (z-\bar{z})^2 - R^2 = 0$	$\bar{y}\ \bar{z}\ R$
C/Y		Parallel to Y-axis	$(x-\bar{x})^2 + (z-\bar{z})^2 - R^2 = 0$	$\bar{x}\ \bar{z}\ R$
C/Z		Parallel to Z-axis	$(x-\bar{x})^2 + (y-\bar{y})^2 - R^2 = 0$	$\bar{x}\ \bar{y}\ R$
CX		On X-axis	$y^2 + z^2 - R^2 = 0$	R
CY		On Y-axis	$x^2 + z^2 - R^2 = 0$	R
CZ		On Z-axis	$x^2 + y^2 - R^2 = 0$	R

MCNP Surfaces (4)

K/X	Cone	Parallel to X-axis	$\sqrt{(y-\bar{y})^2+(z-\bar{z})^2}-t(x-\bar{x})=0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
K/Y		Parallel to Y-axis	$\sqrt{(x-\bar{x})^2+(z-\bar{z})^2}-t(y-\bar{y})=0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
K/Z		Parallel to Z-axis	$\sqrt{(x-\bar{x})^2+(y-\bar{y})^2}-t(z-\bar{z})=0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
KX		On X-axis	$\sqrt{y^2+z^2}-t(x-\bar{x})=0$	$\bar{x}\ t^2\ \pm 1$
KY		On Y-axis	$\sqrt{x^2+z^2}-t(y-\bar{y})=0$	$\bar{y}\ t^2\ \pm 1$
KZ		On Z-axis	$\sqrt{x^2+y^2}-t(z-\bar{z})=0$	$\bar{z}\ t^2\ \pm 1$ ± 1 used only for 1 sheet cone
SQ	Ellipsoid Hyperboloid Paraboloid	Axis parallel to X-, Y-, or Z-axis	$A(x-\bar{x})^2+B(y-\bar{y})^2+C(z-\bar{z})^2+2D(x-\bar{x})+2E(y-\bar{y})+2F(z-\bar{z})+G=0$	A B C D E F G $\bar{x}$ $\bar{y}$ $\bar{z}$
GQ	Cylinder Cone Ellipsoid Hyperboloid Paraboloid	Axes not parallel to X-, Y-, or Z-axis	$Ax^2+By^2+Cz^2+Dxy+Eyz+Fzx+Gx+Hy+Jz+K=0$	A B C D E F G H J K
TX	Elliptical or circular torus. Axis is parallel to X-,Y-, or Z-axis		$(x-\bar{x})^2/B^2+(\sqrt{(y-\bar{y})^2+(z-\bar{z})^2}-A)^2/C^2-1=0$	$\bar{x}\ \bar{y}\ \bar{z}$ A B C
TY			$(y-\bar{y})^2/B^2+(\sqrt{(x-\bar{x})^2+(z-\bar{z})^2}-A)^2/C^2-1=0$	$\bar{x}\ \bar{y}\ \bar{z}$ A B C
TZ			$(z-\bar{z})^2/B^2+(\sqrt{(x-\bar{x})^2+(y-\bar{y})^2}-A)^2/C^2-1=0$	$\bar{x}\ \bar{y}\ \bar{z}$ A B C
XYZP		Surfaces defined by points		See pages 3–15 and 3–17

MCNP Surfaces (5)

- **Rectangular Parallelepiped**



RPP xmin xmax ymin ymax zmin zmax

for infinite in a direction, use min=0 & max=0

- **Right Circular Cylinder**

RCC Vx Vy Vz Hx Hy Hz R

Vx Vy Vz = center of base

Hx Hy Hz = axis of cylinder, magnitude = height

R = radius

- **Others**

ARB, BOX, ELL, HEX, REC, RHP, TRC, WED

MCNP Surfaces (6)

F(x,y,z) = S

where

F = 0 is a surface equation

x,y,z arbitrary 3-D coordinate

S result of xyz point in equation

S is the "sense" of a point with respect to the surface

S > 0 - point is outside the surface

S = 0 - point is on the surface

S < 0 - point is inside the surface

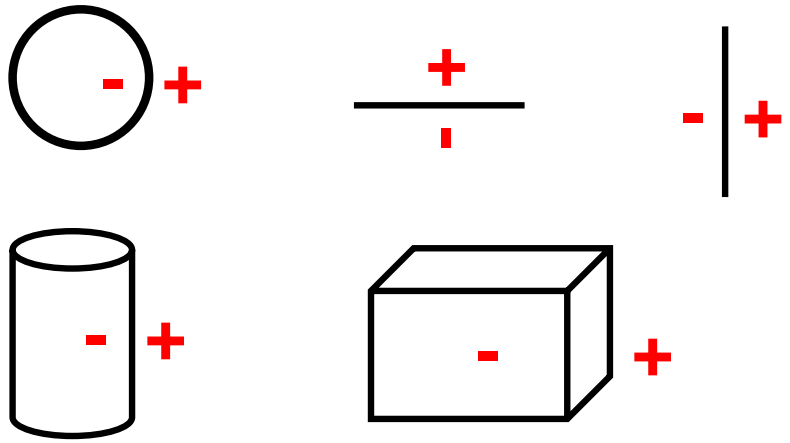
For macrobodies,

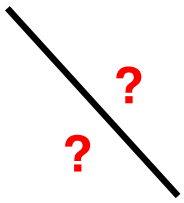
- inside the body ➜ negative
- outside the body ➜ positive

Alternate determination of sense:

- Surface normal points in + direction

Examples





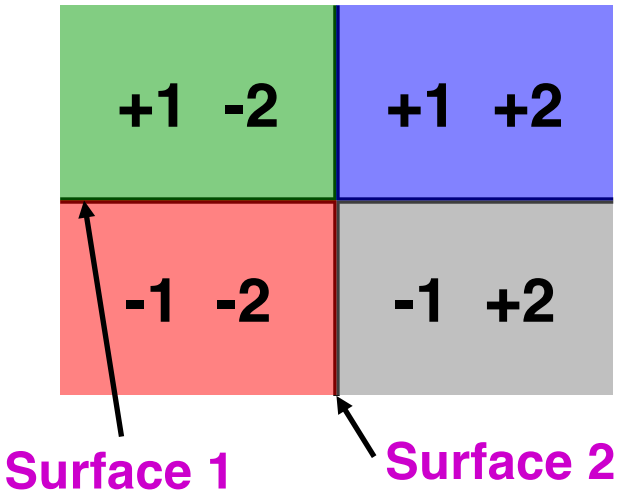
Note – General Plane:
The sense depends on the normalization of the surface equation. Multiplying both sides of the equation by -1 flips the sense. If in doubt, pick a convenient (x,y,z) point, substitute into surface expression to find the sense, + or -.

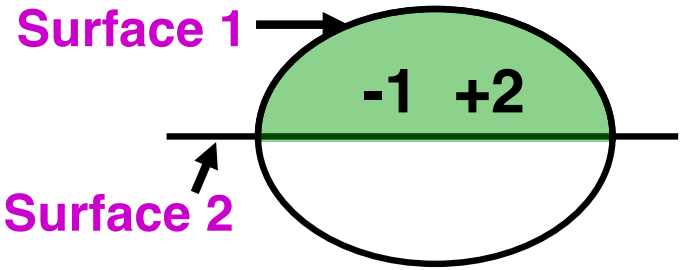
Intersection & Union of Sides

MCNP convention:

Blank signifies intersection

+1 -2 = intersection of
+side of surface 1 and
-side of surface 2

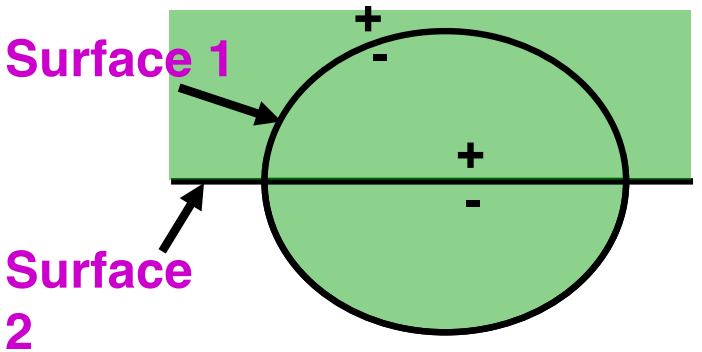




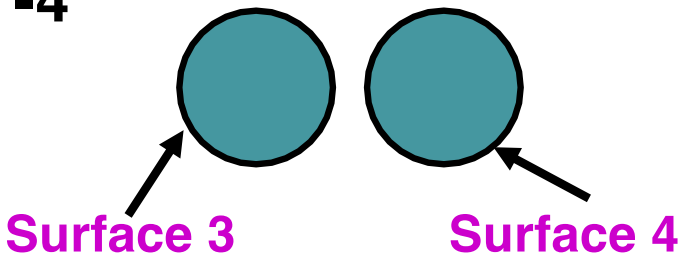
MCNP convention:

Colon signifies union

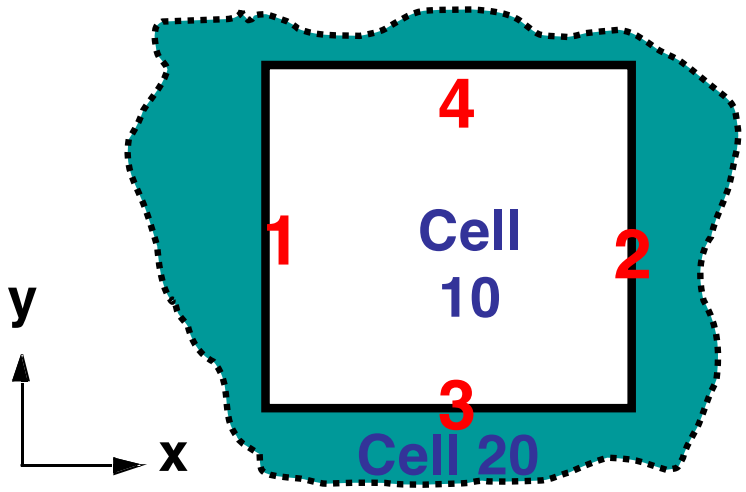
-1 : 2 == union of
-side of surface 1 with
+side of surface 2



-3 : -4



MCNP Geometry (7)



Surfaces:

1	px	-5.0
2	px	5.0
3	py	-5.0
4	py	5.0

Intersection logic for Cell 10 definition:

```
10  0     +1  -2  +3  -4    imp:n=1
```

Union logic for Cell 20 definition:

```
20  0     -1 : 2 : -3 : 4   imp:n=1
```

Complement operator for Cell 20 definition:

```
20  0     #10               imp:n=1
```

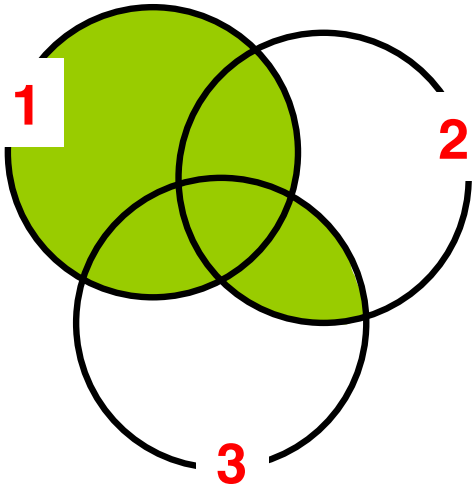
MCNP Geometry (9)

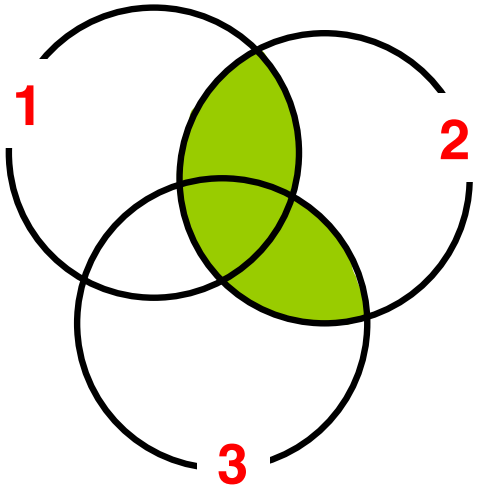
- **Intersections are done before unions**

-1 : -3 -2 is equivalent to -1 : (-3 -2)

- **Example**

-1 : -3 -2 is not the same as (-1 : -3) -2





Example Problem - puc1 (1)

Plutonium Nitrate Solution
In a Cylindrical tank
--

Sample input file: puc1.txt

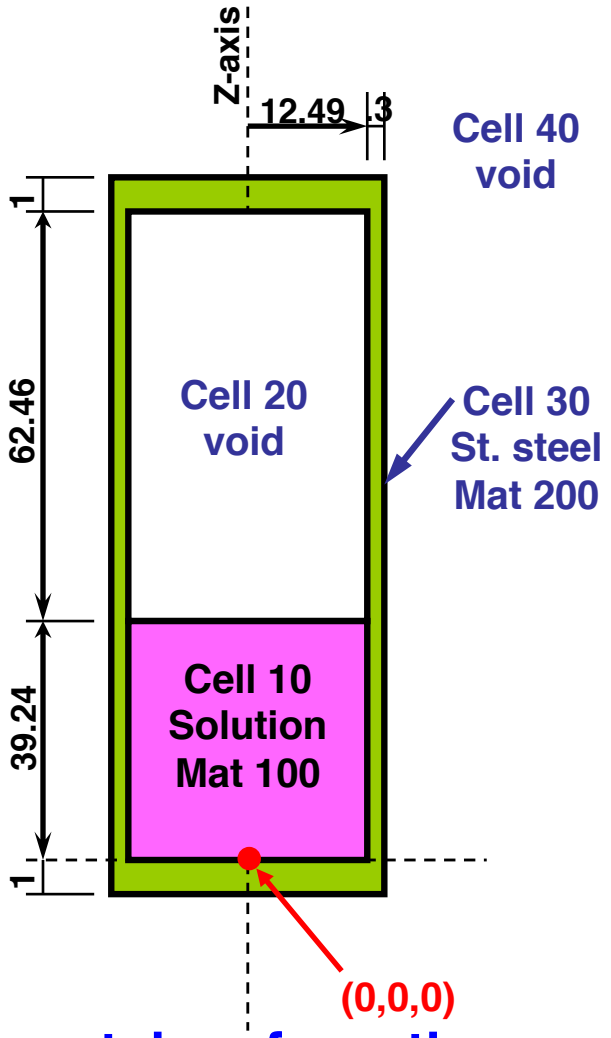
Example Problem - puc1 (2)

Cell 10:

Radius = 12. 49 cm
Height = 39.24 cm
Density = 9.9270e-2 atoms/b-cm

Cell 30:

Tank thickness = 0.3 cm
Bottom thickness = 1.0 cm
Top thickness = 1.0 cm
Inside height = 101.7 cm
Density = 8.6360e-2 atoms/b-cm



Note: Dimensions & material specifications taken from the example used in Section 5.3 of the MCNP Criticality Primer

Example Problem - puc1 (3)

Material 100

```
rho = 9.927e-2

 1001   6.0070e-2
 8016   3.6540e-2
 7014   2.3699e-3
94239   2.7682e-4
94240   1.2214e-5
94241   8.3390e-7
94242   4.5800e-8
```

Material 200

```
rho = 8.636e-2

24050   7.1866e-4
 24052   1.3859e-2
 24053   1.5715e-3
 24054   3.9117e-4
 26054   3.7005e-3
 26056   5.8090e-2
 26057   1.3415e-3
 26058   1.7853e-4
 28058   4.4318e-3
 28060   1.7071e-3
 28061   7.4207e-5
 28062   2.3661e-4
 28064   6.0256e-5
```

Material 100 - Pu-nitrate solution

Material 200 - stainless-steel, for container

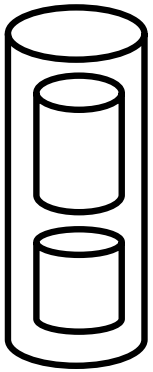
Void - outside container, and
inside container above solution

To save typing, just set up the problem geometry (MCNP cells & surfaces), and then:

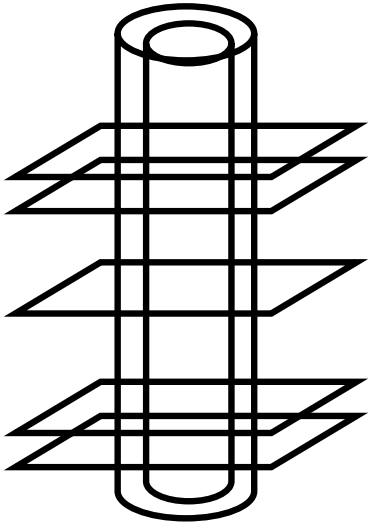
- Cut & paste DATA cards from file puc1_data_cards.txt in SOLUTIONS folder
- Includes KCODE, KSRC, M100, MT100, M200 cards
- mt100 lwtr card invokes S(α,β) thermal scattering treatment for hydrogen in material 100

Example Problem - puc1 (4)

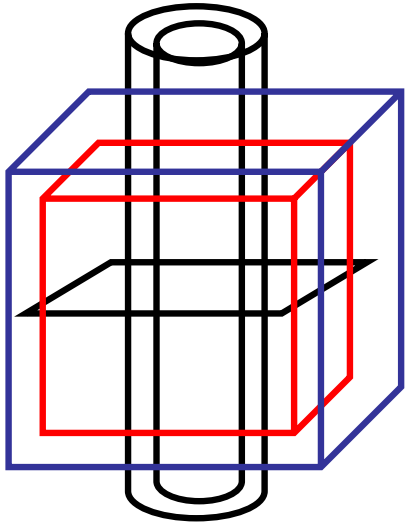
Possible geometry setups



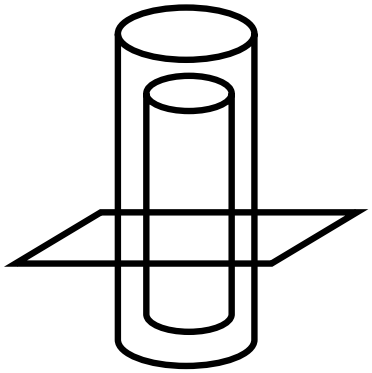
3 RCC bodies



2 infinite z-cylinders
+ 5 planes



2 infinite z-cylinders
+ 2 RPP bodies
+ 1 plane



2 RCC bodies
+ plane

Try it this way
(you'll see why later …)

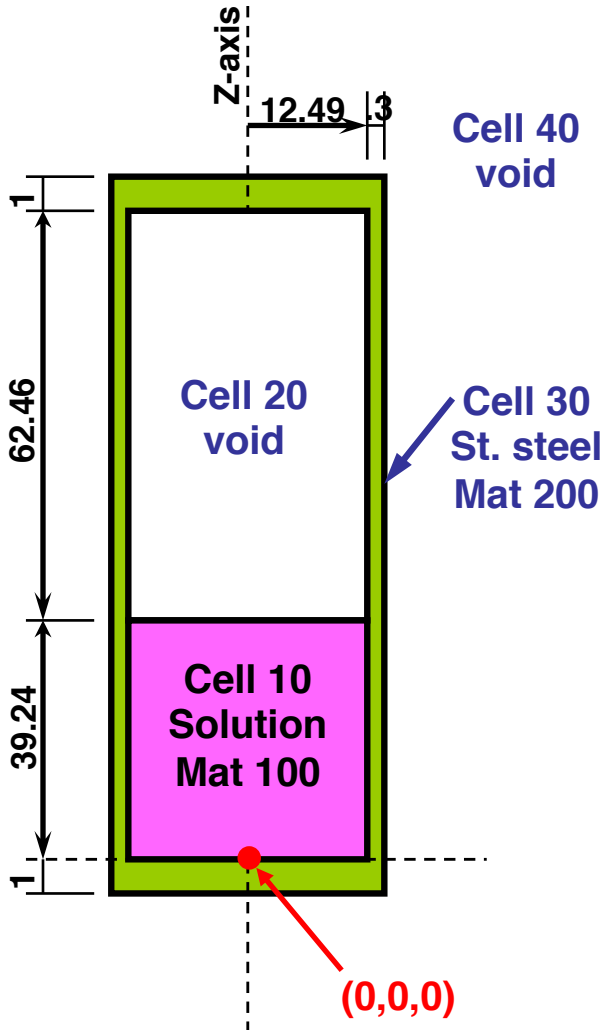
Example Problem - puc1 (5)

Construct a single tank

- Cell 10 – material 100
- Cell 20 – void
- Cell 30 – material 200
- Cell 40 – void

- Use ksrc, center of cell 10
- Use 1000 neutrons/cycle
- Discard 25 cycles, run 100 total
- Don't forget imp:n

- Edit file puc1.txt
- Plot
- Compute keff



Example Problem - puc1 (6)

```
puc1 - single cylinder
c    CELL CARDS
10   100    9.9270e-2     -1 -3     imp:n=1       $ solution
20   0                    -1  3     imp:n=1       $ void above solution
30   200    8.6360e-2      1 -2     imp:n=1       $ can
40   0                     2        imp:n=0       $ outer void

c    SURFACE CARDS
1    RCC   0. 0.  0.    0. 0. 101.7    12.49      $ inner can
2    RCC   0. 0. -1.    0. 0. 103.7    12.79      $ outer can
3    pz    39.24                                  $ solution height

c      DATA CARDS (from puc1_data_cards.txt)
kcode    1000   1.0   25   100 $ criticality calc
ksrc     0. 0. 19.62           $ source guess
c
m100          . . .            $ solution material
mt100 lwtr                     $ use h2o  S(a,b)
m200          . . .            $ can material
```

Result:
keff = 0.88778 ± 0.00363

MCNP - Running & Plotting (1)

mcnp6 i=in.txt o=out.txt ... [options]

Default File Name	Description
inp	Input file
outp	ascii output file
runtpe	Binary restart file
mctal	ascii tally results
meshtal	mesh tallies
srctp	source guess

Options	Operation
i	Process
p	Plot geometry
x	Process xsec's
r	Particle transport
z	Plot tally results
	Plot cross sections

default: ixr

Examples:

```
mcnp6   inp=test1.txt   outp=test1o.txt   run=test1r   ip
mcnp6   i=gdv.txt
mcnp6   name=test1.  i=in.txt
mcnp6   i=test1.txt   ixz
```

Must explicitly include any file suffixes, such as ".txt"

MCNP - Running & Plotting (2)

- **Geometry plotting:**

  mcnp6 i=inp ip **<-- process input & plot geometry**

- **Run a problem:** **mcnp6 i=inp.txt**
  - **Creates files:**

outp	output file	text
runtpe	restart file	binary
mctal	tally file (if prdmp card set)	text
meshtal	mesh tally file	text
srctp	source (for restart)	binary
comout	(if plotting)	text

  - **Last letter changed, if file already exists**

- **Run a problem:** **mcnp6 n=prob. i=prob.txt**
  - **Creates files:**

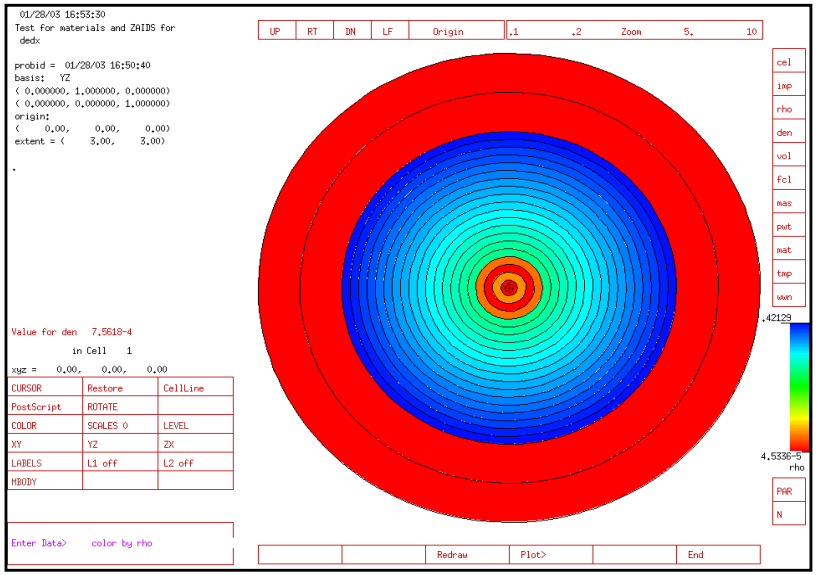
prob.o	output file
prob.r	restart file
prob.m	tally file
prob.s	source
prob.c	(if plotting)

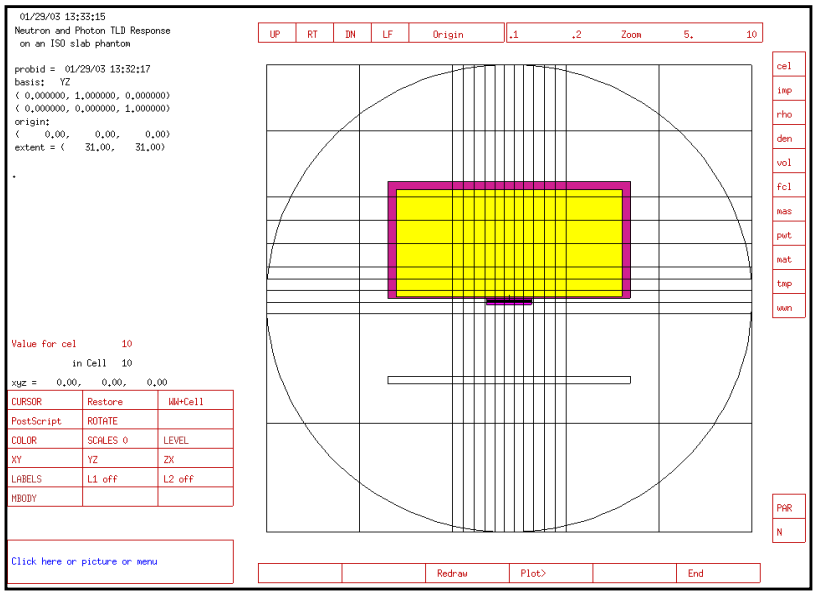
  - **Aborts if any of these already exist - MCNP will not overwrite these files**

MCNP - Running & Plotting (3)

Plotting geometry

- **Interactive 2-dimensional slices**
- **Errors displayed as dashed (red) lines**
- **Many problem variables can be shown:**
  - cell & surface numbers
  - macrobody facets
  - importances imp:n
  - lattice variables u, lat, fill, level
  - material properties rho, den, ...
  - variance reduction parameters
  - weight windows mesh





MCNP - Running & Plotting (4)

Plotting commands

ORIGIN X Y Z or 15.0 0.0 5.0	Position the center of the plot window at (X,Y,Z).
EXTENT EH ex 25.0	Scale the plot with extent EH Smaller EH, closer view
PX VX px 3.0 PY VY py 5.0 PZ VZ pz 0.01	Set the view plane to x=VX y=VY z=VZ

…. Lots of other options & commands - see MCNP manual

Number of Cycles or Histories

- **Number of cycles - for criticality problems:**

  KCODE npc kguess ndiscard ncycles

  - **npc** = neutrons/cycle
  - **kguess** = initial guess for k-effective, usually 1.0
  - **ndiscard** = number of initial cycles to discard (no tallies)
  - **ncycles** = total number of cycles to run (including ndiscard), in a "continue" run, **ncycle** includes those from previous runs

- **Number of histories - for fixed-source problems:**

  NPS N

  - Terminate the Monte Carlo calculation after **N** histories have been run
  - In a continue-run, NPS is the total number of particles - including runs before the continue run (cumulative)
  - Negative entry will print output file at time of last history run

Particle Types & Physics

- **Problem type:**

  MODE q_1 q_2 … q_n

  - List of all particles, q_1, q_2, ..., to be transported in space-delimited format
  - Neutron only: mode n (default)
  - Neutron+photon: mode n p
  - Neutron+photon+electron: mode n p e

- **Particle physics options:**

  PHYS:*q* (*list of input parameters*)

  - Controls physics options; format varies based on particle type (q)
  - Includes control of charged-particle straggling, upper energy limit, light ion recoil, implicit vs analog capture, bremsstrahlung production, secondary particle production, delayed gamma production, photonuclear particle production, fission multiplicity, etc.
  - For criticality problems (mode n), almost always leave out phys:n card & just use defaults

Source Cards (1)

- For criticality calculations, can use KSRC card to define initial neutron starting points

KSRC x1 y1 z1 x2 y2 z2 x3 y3 z3 …..

  - Can define any number of points, reused as needed
  - Points are used ONLY for initial source guess, ignored on subsequent cycles

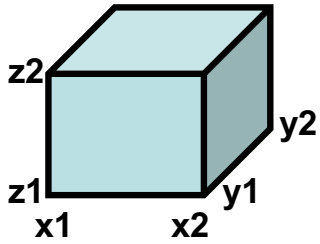
- For fixed-source or criticality calculations, can use SDEF card to define starting parameters for histories
  - Very general sources can be described
  - For criticality calculations, only used for initial source guess, ignored on subsequent cycles
  - See "Sources" lecture in MCNP Intro Class for more details

- Cannot use both SDEF and KSRC in same calculation

SDEF Recipes for Volume Sampling

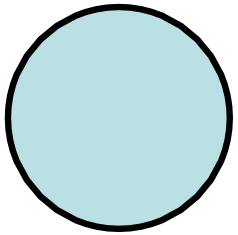
- **Uniform source within the volume of a box, isotropic in direction**

```
sdef   x=d1  y=d2  z=d3
  si1   x1  x2
  sp1   0   1
  si2   y1  y2
  sp2   0   1
  si3   z1  z2
  sp3   0   1
```



- **Uniform source within the volume of a sphere, isotropic in direction**

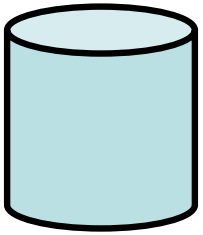
```
sdef   pos= x0 y0 z0    rad=d1
  si1 h  0  R
  sp1    -21  2    $ sample ~ r**2
```



center = x0, y0, z0
radius = R

- **Uniform source within the volume of a cylinder, isotropic in direction**

```
sdef   pos= x0 y0 z0    rad=d1
       axs= 0 0 1       ext=d2
  si1 h  0  R
  sp1    -21  1    $ sample ~ r
  si2    0  H
  sp2    0  1
```



center of base = x0, y0, z0
radius = R
height = H

Example Problem - dtpoly (1)

Simple Shielding Calculation

--

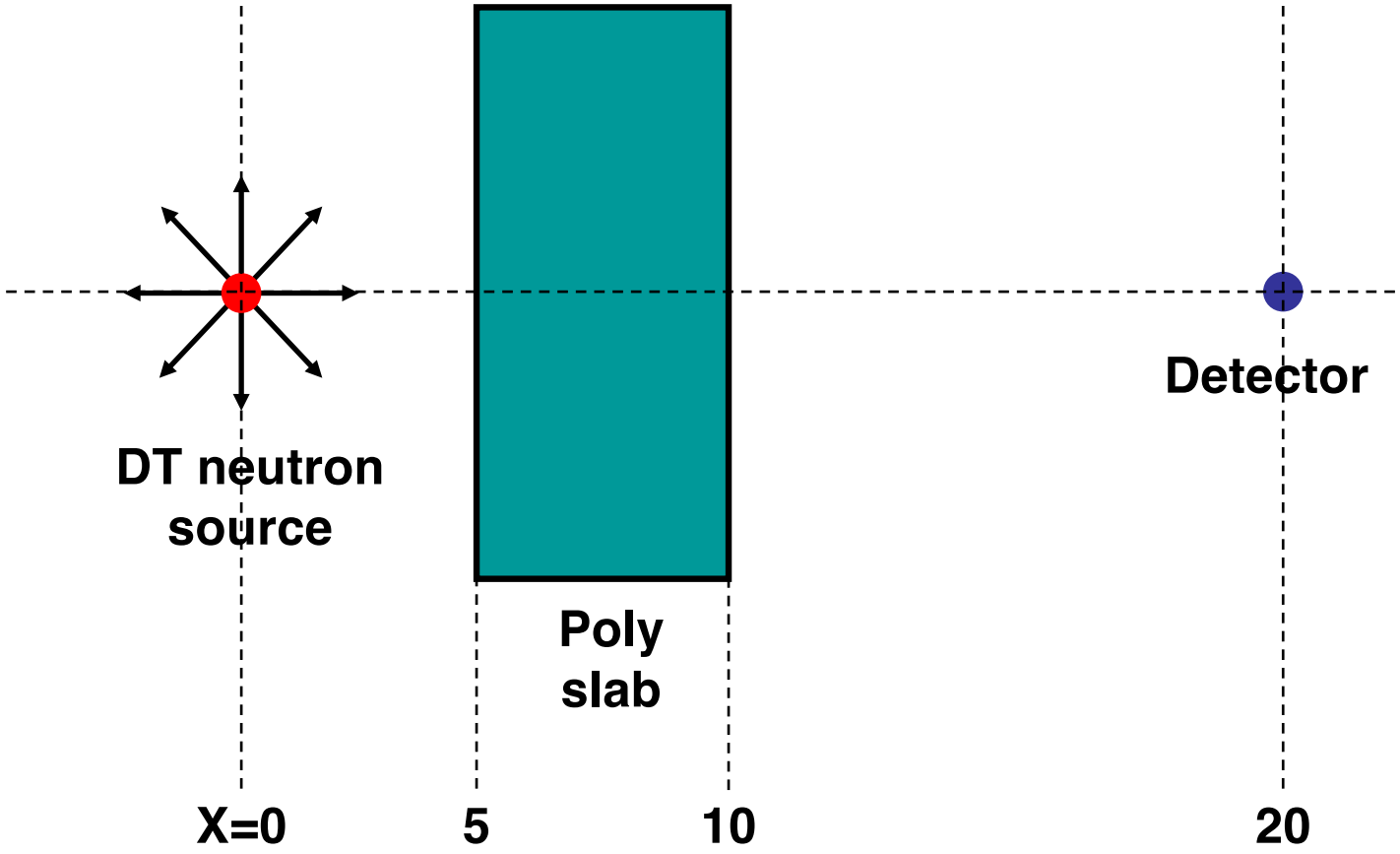
DT Source through Poly Slab, with point detector

Sample input file: dtpoly.txt

(optional)

Simple Shielding Calculation (2)

- **Neutrons from D-T generator hitting a polyethylene slab**

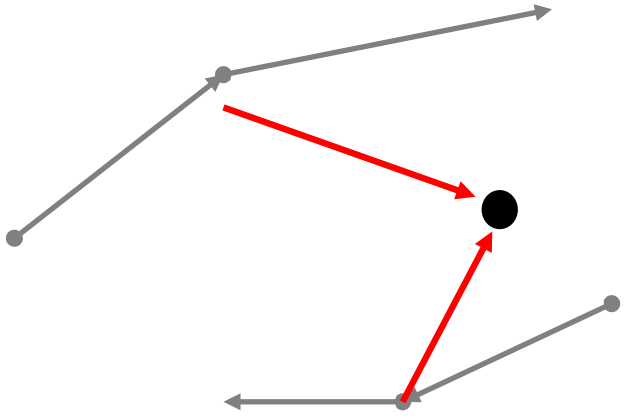


PNNL-15870, Polyethylene: ρ= 0.93 g/cc, 2 H atoms, 1 C atom

DT neutons: emitted isotropically, 14 MeV

Simple Shielding Calculation (3)

- **Tally for Flux at Point - deterministic estimate, for every collision**



- **Form:** **Fn:q X Y Z R**
  - **n = tally number ending in 5, ≤ 995**
  - **q = particle type = particle <u>symbol</u>, n or p (no charged particles)**
  - **X, Y, Z = position of tally point where flux is desired**
  - **R = radius of a "sphere of constant flux"**
    - **Required to keep tally variance finite**
    - **Recommended to be about one mean free path**
    - **Can use 0.0 for points in a void region**
- **Example** - neutron flux at point (20,0,0) in void region

F5:n 20.0 0.0 0.0 0.0

Simple Shielding Calculation (4)

- **Material** reference: PNNL-15870, on class DVD

Polyethylene: 2 H atoms, 1 C atom, ρ = 0.93 g/cc

H: 1001
C: 6000 - strange aspect of ENDF/B-VII, 6000 instead of 6012

```
m1000     1001  2      6000  1
```

- **Source**

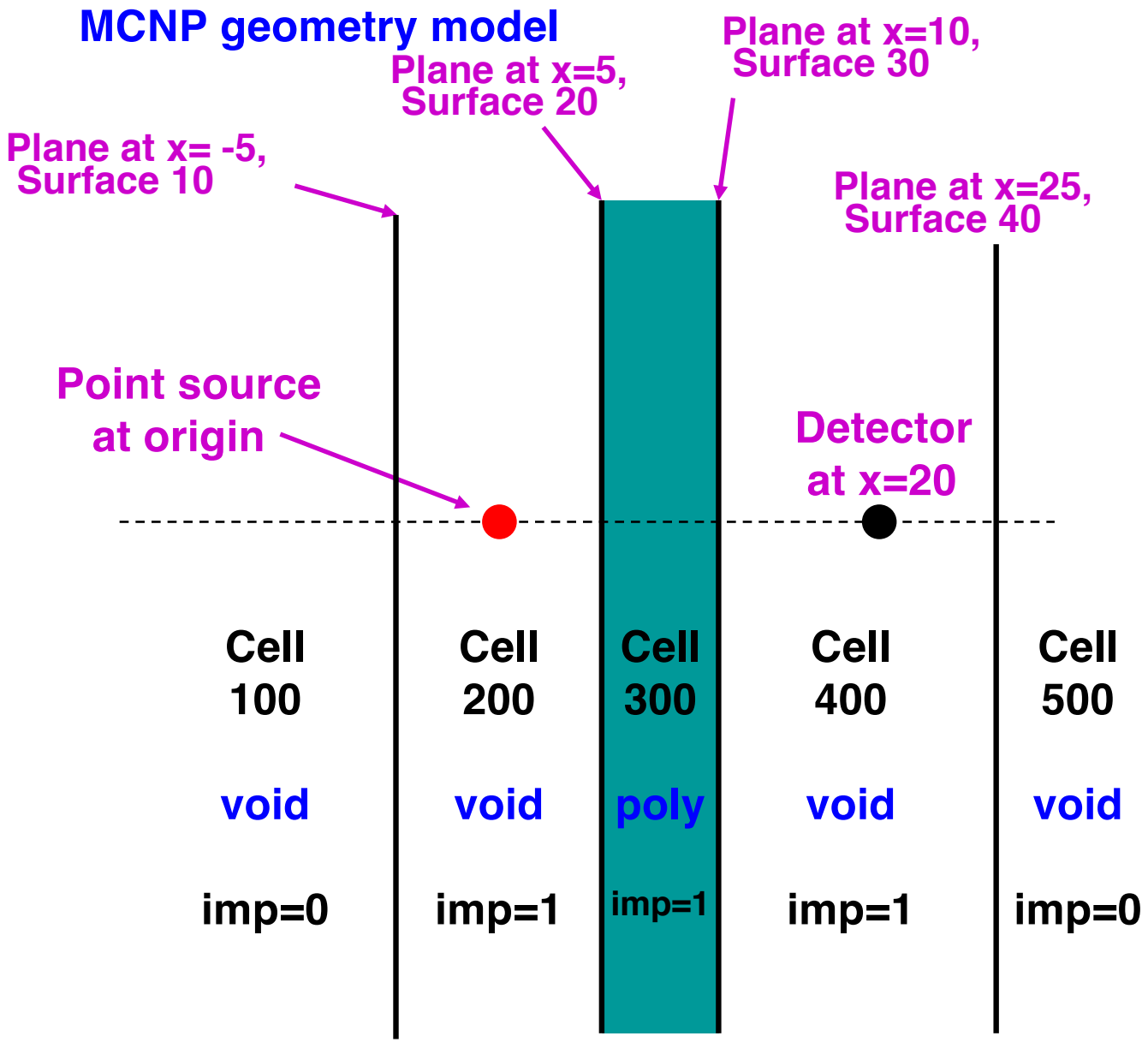
DT neutons: emitted isotropically, 14 MeV

SDEF card defaults (if not specified):
direction: sampled isotropically in 3D
position: origin
energy: 14.0 MeV

For this example, can specify

```
sdef      pos= 0.   0.   0.     erg= 14.
```

Simple Shielding Calculation (5)



Coordinates

- X-Y-Z coordinates
- Units: cm

Poly slab

- Region of space right of surface 20, left of surface 30

Cells 100 & 500 ???

- Outside of problem
- Kill any neutrons that enter Cell 100 or Cell 500 (importance=0)
- This scheme gives us a finite problem domain

Simple Shielding Calculation (6)

Create & edit file dtpoly.txt, or copy from SOLUTIONS folder, with these lines:

```
DT neutron beam, poly slab, detector
c GEOMETRY - CELLS
100     0                -10        imp:n=0    $ void to left, kill neutrons
200     0                +10 -20    imp:n=1    $ void to left  of slab
300     1000   -0.93      +20 -30    imp:n=1    $ poly slab, rho=.93 g/cc
400     0                +30 -40    imp:n=1    $ void to right of slab
500     0                +40        imp:n=0    $ void to right, kill neutrons
                                                        $ NOTE: 1 blank line here !
c GEOMETRY - SURFACES
10      px      -5.0                            $ x-plane, problem left  boundary
20      px       5.0                            $ x-plane, left  side of poly slab
30      px      10.0                            $ x-plane, right side of poly slab
40      px      25.0                            $ x-plane, problem right boundary
                                                        $ NOTE: 1 blank line here !
c OTHER STUFF
nps     1e5                                     $ number of source neutrons
c MATERIALS - poly
m1000   1001 2    6000 1                       $ could also have "mt1000 poly" card
c SOURCE
sdef    pos= 0. 0. 0.      erg=14.              $ DT isotropic source
c POINT DETECTOR
f5:n    20.0  0.0  0.0      0.0                 $ x,y,z,   radius
```

Simple Shielding Calculation (7)

- Save file dtpoly.txt in a working folder, C:\MCNP\WORK
- Open a command window & go to the working folder

```
dir
                        . . . Should see dtpoly.txt
mcnp6   i=dtpoly.txt
                        . . . Lots of stuff to the screen
dir
                        . . . See what files are created
                        . . . Look at file outp with text editor
```

- Do all of the above again
- Cleanup

```
del  out?  srct?  runtp?
```

Simple Shielding Calculation (8)

Output to screen:

```
  mcnp      ver=6     , ld=06/02/14  07/08/14 20:23:45
            Code Name & Version = MCNP6_DEVEL, 6-1-02
            Copyright LANS/LANL/DOE - see output file

    _/      _/      _/_/_/      _/      _/      _/_/_/      _/_/_/
   _/_/  _/_/     _/           _/_/  _/_/     _/    _/    _/
  _/ _/_/ _/      _/           _/ _/_/ _/     _/_/_/      _/_/_/
 _/   _/  _/     _/           _/   _/  _/    _/              _/
_/       _/      _/_/_/      _/       _/   _/        _/_/_/

  warning.  Physics models disabled.
  comment.  total nubar used if fissionable isotopes are present.
  warning.    1 materials had unnormalized fractions. print table 40.
 comment. using random number generator  1, initial seed = 19073486328125
imcn    is done

ctm =         0.00    nrn =                    0
dump     1 on file runtpe    nps =             0    coll =                  0
xact    is done

cp0 =    0.01
      run terminated when       100000  particle histories were done.
ctm =         0.14    nrn =              4740086
dump     2 on file runtpe    nps =        100000    coll =             270916

mcrun   is done
```

Simple Shielding Calculation (9)

- **Output file outp**

```
Code Name & Version = MCNP6_DEVEL, 6-1-02                    _

. . . Listing of input file, cell info, random number info, xsec info, ...
1problem summary
      run terminated when      100000  particle histories were done.
. . . Particle production & loss tables, cell activity, other info, ...

1tally        5        nps =      100000
           tally type 5    particle flux at a point detector.   units
  1/cm**2
           particle(s): neutrons

 detector located at x,y,z = 2.00000E+01 0.00000E+00 0.00000E+00
                2.10851E-04 0.0073

 detector located at x,y,z = 2.00000E+01 0.00000E+00 0.00000E+00
 uncollided neutron flux
                1.16592E-04 0.0000

. . . Statistical checks on tallies, other info, ...
 run terminated when      100000  particle histories were done.
 computer time =    0.13 minutes
```

Random Numbers (1)

- **Random numbers**
  - MCNP is a Monte Carlo code & uses random numbers to sample from probability densities
  - If a calculation is repeated using the same input & cross-section libraries, bit-for-bit identical results will be obtained.

- **How can you repeat a calculation using different random numbers?**

  Add this data card:

  rand gen= k seed= n

  gen= k - choose the random number generator
  - gen=1 for default 48-bit generator, period ~10^{14}
  - gen= 2, 3, 4 for 63-bit generators, period ~10^{18}

  seed= n - set the initial random seed for the problem to n
  - n should be an odd integer, 18 digits or fewer

Random Numbers (2)

- For problem puc1, change the random seed, rerun the problem, compare answers.
- Results should differ, but should agree within statistics.
- If calculations are run using different random seeds, the results are statistically independent & may be averaged together. (Using weights proportional to $1/\sigma^2$)

```
puc1 - single cylinder
10       100      9.9270e-2       -1 -3       imp:n=1
20       0                        -1  3       imp:n=1
30       200      8.6360e-2        1 -2       imp:n=1
40       0                         2          imp:n=0

1        RCC    0. 0.  0.     0. 0. 101.7      12.49
2        RCC    0. 0. -1.     0. 0. 103.7      12.79
3        pz     39.24

c DATA CARDS from file puc1_data_cards.txt
kcode    1000 1.0 25 100
ksrc     0. 0. 19.62
m100      . . .
mt100     . . .
m200      . . .
rand     seed= 123456789
```

Result:

Default RN:	keff = 0.88778 ± 0.00363
New RN:	keff = 0.87804 ± 0.00377

Continue Runs (1)

mcnp6 i= more.txt r=runtpe C

- Short input file (2 lines): more.txt
  - Special input file to change parameters
  - Criticality Example:

```
continue
kcode  1000 1.0 25  325
```

Can only change the 4th entry - total number of cycles

  - Fixed-source Example:

```
continue
nps  1e6
```

- Picks up exactly where previous calculation left off, given the runtpe file from the previous run
- Don't forget the "C" on the command line
- Some entries on these cards can be changed for continue runs (see manual): kcode, nps, dd, ctme, prdmp, print, mplot, dbcn, lost

Continue Runs (2)

- Continue problem puc1.txt, so that a total of 325 active cycles are run.
- Below, assume restart file (runtpe) is called runtpe

- Results should differ, but should agree within statistics.
- Statistics will be smaller, $\sigma \sim 1 / \sqrt{\text{number of histories}}$

note: original run: 100-25 = 75 active cycles
continue to: 325-25 = 300 active cycles

4x histories ➔ cuts statistics in half

File more.txt

```
continue
kcode  1000 1.0 25  325
```

Commands

```
mcnp6 i=more.txt r=runtpe c
```

Results:

75 active cycles: keff= 0.88778 0.00363
300 active cycles: keff= 0.88445 0.00185

Criticality Calculations

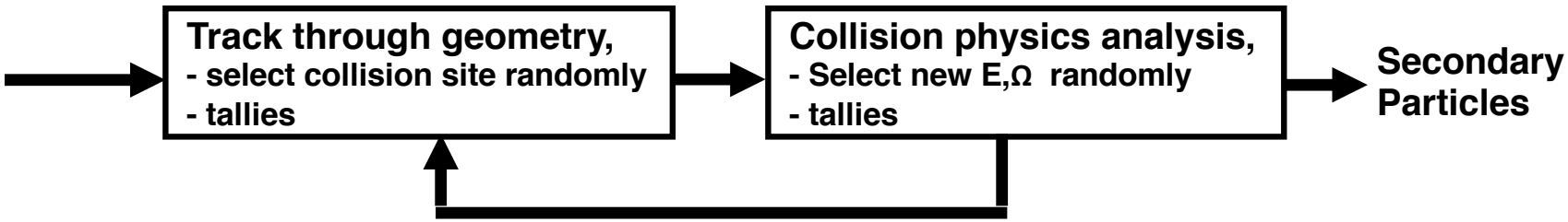
Criticality Overview
Criticality Estimators
KCODE & KSRC Cards
Examples & Output

Convergence
Plotting
Examples & Plots

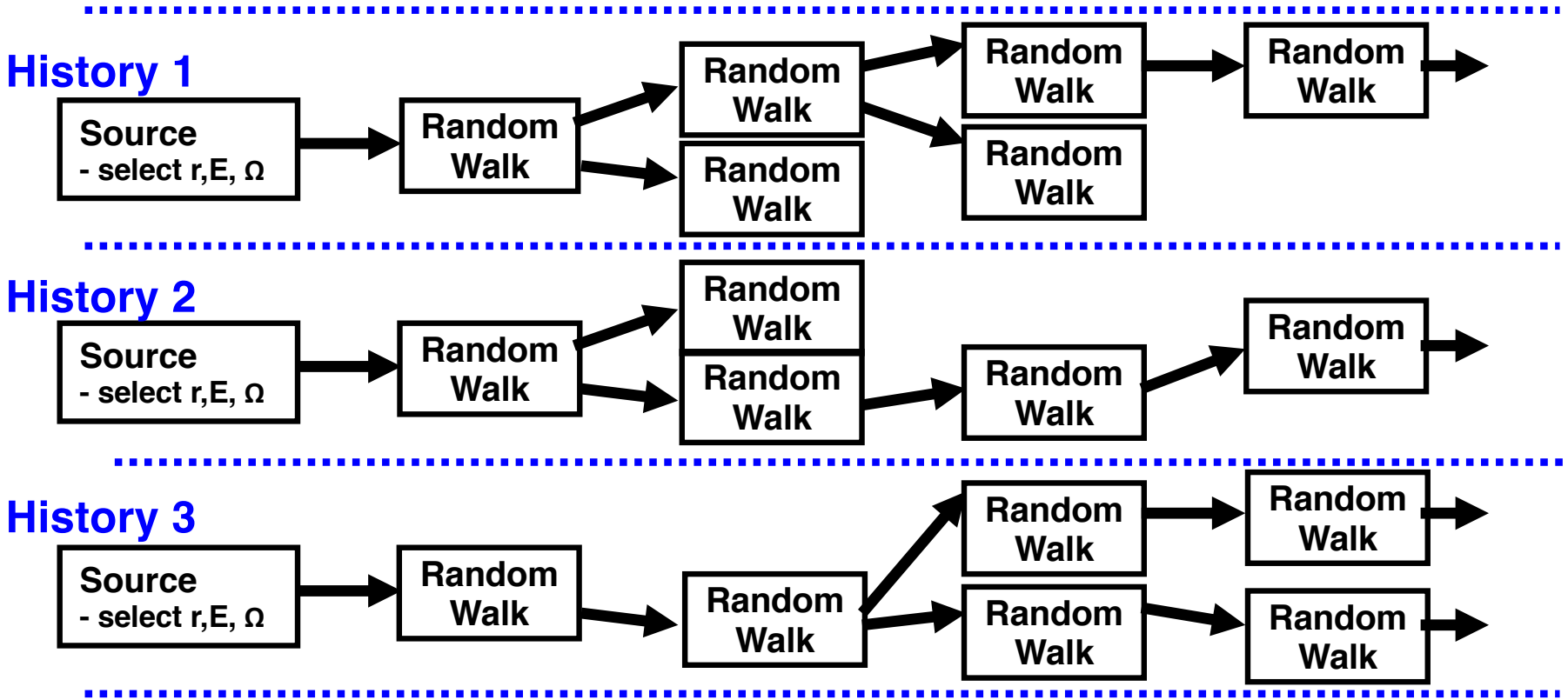
Criticality Overview

Fixed-source Monte Carlo Calculation

Random Walk for a particle



Particle Histories



Time-dependent Transport

- **Time-dependent linear Boltzmann transport equation for neutrons, with prompt fission source & external source**

$$\frac{1}{v}\frac{\partial\psi(\vec{r},E,\vec{\Omega},t)}{\partial t} = Q(\vec{r},E,\vec{\Omega},t) + \iint \psi(\vec{r},E',\vec{\Omega}',t)\,\Sigma_S(\vec{r},E'\rightarrow E,\vec{\Omega}\cdot\vec{\Omega}')\,d\vec{\Omega}'dE'$$

$$+ \frac{\chi(\vec{r},E,t)}{4\pi}\iint \nu\Sigma_F(\vec{r},E')\,\psi(\vec{r},E',\vec{\Omega}',t)\,d\vec{\Omega}'dE'$$

$$-\left[\,\vec{\Omega}\cdot\nabla + \Sigma_T(\vec{r},E)\,\right]\cdot\psi(\vec{r},E,\vec{\Omega},t)$$

$$\frac{1}{v}\frac{\partial\psi(\vec{r},E,\vec{\Omega},t)}{\partial t} = Q + [S+M]\cdot\psi - [L+T]\cdot\psi$$

- **This equation can be solved directly by Monte Carlo, assuming:**
  - **Each neutron history is an IID trial (independent, identically distributed)**
  - **All neutrons must see same probability densities in all of phase space**
  - **Usual method: geometry & materials fixed over solution interval Δt**

K_{eff} Eigenvalue Equations

- Another approach to creating a static eigenvalue problem from the time-dependent transport equation is to introduce K_{eff}, a scaling factor on the multiplication (ν)

- Assume:
  1. Fixed geometry & materials
  2. No external source: $Q(r,E,\Omega,t) = 0$
  3. $\partial\Psi/\partial t = 0$: $\nu \rightarrow \nu / k_{eff}$

- Setting $\partial\Psi/\partial t = 0$ and introducing the K_{eff} eigenvalue gives

$$\left[L+T\right]\Psi_k(\vec{r},E,\vec{\Omega}) = \left[S+\frac{1}{K_{eff}}M\right]\Psi_k$$

  - This is a static equation, an eigenvalue problem for K_{eff} and Ψ_k without time-dependence
  - K_{eff} is called the effective multiplication factor
  - K_{eff} and Ψ_k should never be used to model time-dependent problems.
  - K_{eff}-eigenvalue problems can be solved by Monte Carlo methods

K_{eff} Eigenvalue Equation

Leakage Collisions Scattering

$$\left[\vec{\Omega}\cdot\nabla + \Sigma_T(\vec{r},E)\right]\cdot\psi(\vec{r},E,\vec{\Omega}) = \iint\psi(\vec{r},E',\vec{\Omega}')\Sigma_S(\vec{r},E'\rightarrow E,\vec{\Omega}\cdot\vec{\Omega}')d\vec{\Omega}'dE'$$

Multiplication

$$+ \frac{1}{k_{eff}}\frac{\chi(\vec{r},E)}{4\pi}\iint\nu\Sigma_F(\vec{r},E',t)\psi(\vec{r},E',\vec{\Omega}')d\vec{\Omega}'dE'$$

$$[\ L + T\ ]\ \Psi_k = [S + 1/k\ M\ ]\ \Psi_k$$

➜ Jointly find K_{eff} and $\Psi(r,E,\Omega)$ such that equation balances

- The factor 1/k changes the relative level of the fission source, to balance the equation & permit a steady-state solution
- Never use K_{eff} and Ψ to model time-dependent problems.

- Criticality

Supercritical:	$K_{eff} > 1$	gains > losses, reduce gains
Critical:	$K_{eff} = 1$	gains = losses
Subcritical:	$K_{eff} < 1$	gains < losses, increase gains

K-eigenvalue equation

- **Use operator (or matrix) form to simplify notation**

$$(L+T)\Psi = S\Psi + \frac{1}{K_{eff}} M\Psi$$

where

L = leakage operator **S = scatter-in operator**

T = collision operator **M = fission multiplication operator**

- **Rearrange**

$$(L+T-S)\Psi = \frac{1}{K_{eff}} M\Psi$$

$$\Psi = \frac{1}{K_{eff}} \cdot (L+T-S)^{-1} M\Psi$$

$$\Psi = \frac{1}{K_{eff}} \cdot F\Psi$$

➜ **This eigenvalue equation will be solved by power iteration**

$$\Psi^{(n+1)} = \frac{1}{K_{eff}^{(n)}} \cdot F\Psi^{(n)}, \qquad n = 0,\ldots$$

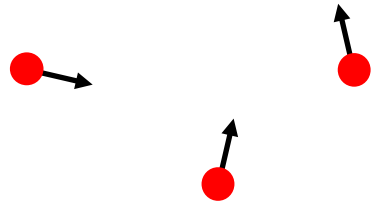
Power Iteration

Power iteration procedure:

1. **Initial guess for K_{eff} and Ψ**

   $K_{eff}^{(0)}$, $\Psi^{(0)}$

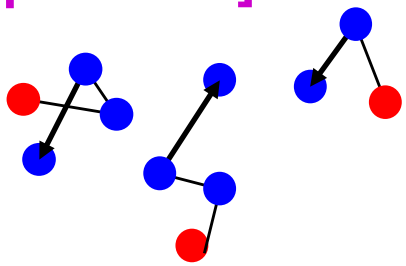
Source points for $\Psi^{(0)}$



2. **Solve for $\Psi^{(n+1)}$ [Monte Carlo random walk for N particles]**

$$\Psi^{(n+1)} = \frac{1}{K_{eff}^{(n)}} \cdot F\Psi^{(n)}$$

Source points For $\Psi^{(n+1)}$



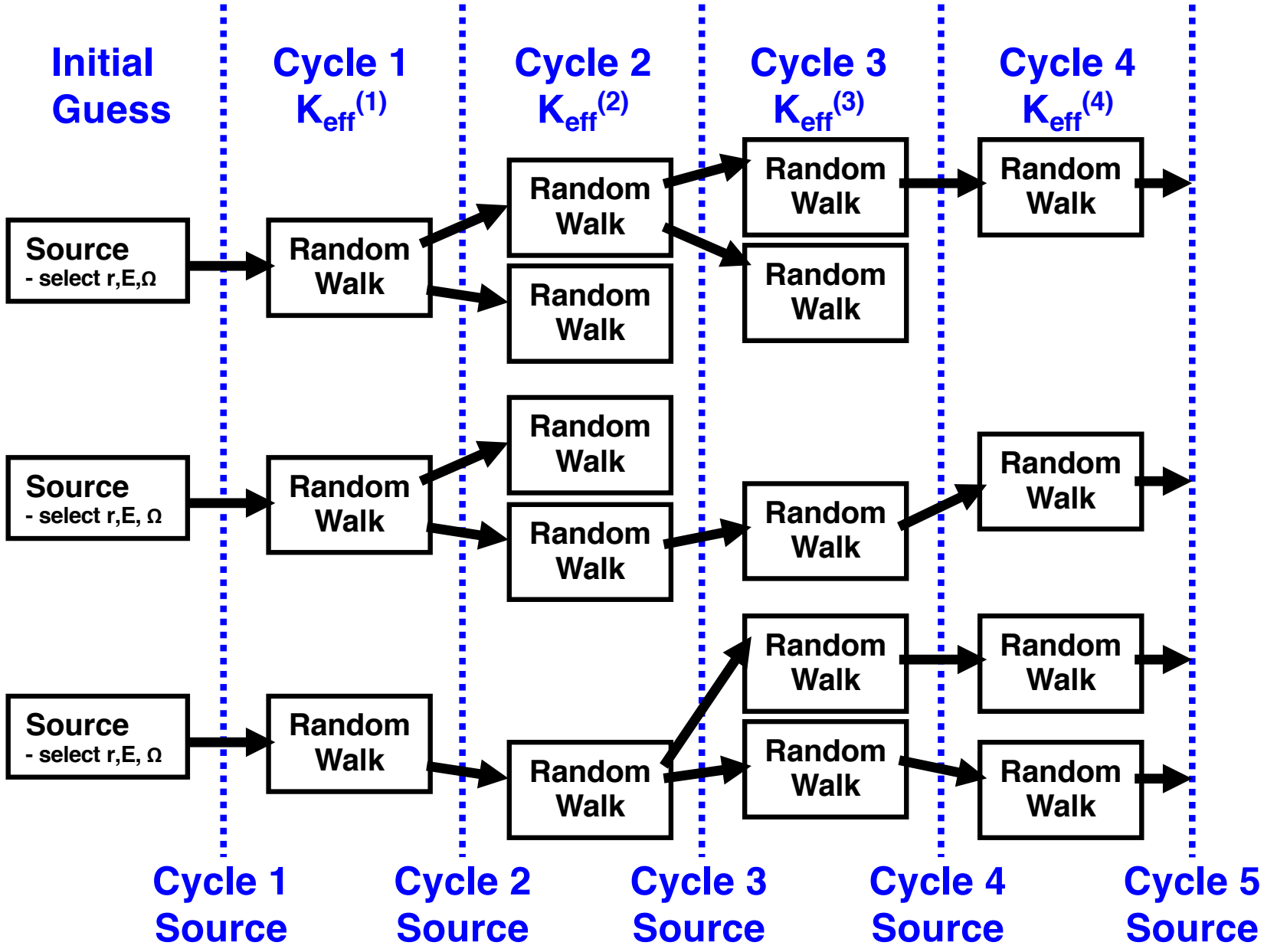
3. **Compute new K_{eff}**

   During iteration n+1, make MC tallies of neutron production from fission during the iteration. These tallies are pathlength, collision, & absorption estimators for

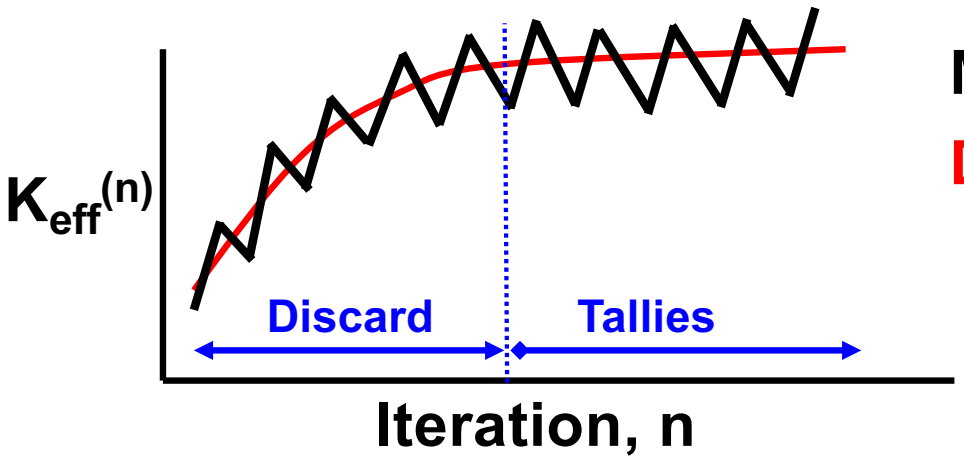
$$K_{eff}^{(n+1)} = K_{eff}^{(n)} \cdot \frac{\int M\Psi^{(n+1)} d\vec{r}}{\int M\Psi^{(n)} d\vec{r}}$$

2. **Repeat 1-3 until both $K_{eff}^{(n+1)}$ and $\Psi^{(n+1)}$ have converged**

Monte Carlo Eigenvalue calculation



Power Iteration



Monte Carlo

Deterministic (S_n)

- **Guess an initial source distribution**
- **Iterate until converged** **(How do you know ???)**
- **Then**
  - **For S_n code:** **done, print the results**
  - **For Monte Carlo:** **start tallies, keep running until uncertainties small enough**

- **Convergence? Stationarity? Bias? Statistics?**

MCNP Criticality Flow

batch = cycle = iteration = generation

- Histories are run in batches of N particles
- Spatial distribution:
  - For 1st cycle (only), fission source sites taken from KSRC, SDEF, or SRCTP
  - After 1st cycle, fission source sites taken from previous cycle
- Total weight in each cycle is N
- For each cycle, 3 estimates of Keff are made for the cycle
  - Keff track-length estimator
  - Keff collision estimator
  - Keff absorption estimator
- At the end of the problem, the cycle estimates are combined into 7 overall estimates (only the last one matters):
  - 3 cumulative Keff estimates using track-length, collision, & absorption
  - 3 cumulative Keff estimates using pairs
  - 1 overall combined cumulative estimate based on all data

K-effective Estimators

Background on Monte Carlo Estimators

- **Pathlength estimator for flux**
  - Flux = total pathlength traveled by all neutrons per unit volume per unit time
  - For flux in a cell, $$\phi \approx \frac{1}{W \cdot V} \sum_{\text{all flights in cell}} d_k \cdot wgt$$

V = cell volume

W = total starting weight

- **Collision estimator for flux**
  - Collision rate = $\Sigma_T \phi$, so ϕ = [collision rate]/ Σ_T
  - For flux in cell, $$\phi \approx \frac{1}{W \cdot V} \sum_{\text{all collisions in cell}} \frac{wgt}{\Sigma_T}$$

- **Absorption estimator for flux**
  - Absorption rate = $\Sigma_A \phi$, so ϕ = [absorption rate]/ Σ_A
  - For flux in cell, $$\phi \approx \frac{1}{W \cdot V} \sum_{\text{all absorptions in cell}} \frac{wgt}{\Sigma_A}$$

Single-cycle Keff Estimators

Neutron production rate = $\nu\Sigma_F\ \phi$

W = total weight starting cycle n

- **Pathlength estimator for Keff, for cycle n**

$$K^{(n)}_{path} = \left(\sum_{\substack{\text{all} \\ \text{flights}}} wgt_j \cdot d_j \cdot \nu\Sigma_F \right) \Big/ W$$

- **Collision estimator for Keff, for cycle n**

$$K^{(n)}_{collision} = \left(\sum_{\substack{\text{all} \\ \text{collisions}}} \frac{wgt_j}{\Sigma_T} \cdot \nu\Sigma_F \right) \Big/ W$$

- **Absorption estimator for Keff, for cycle n**

$$K^{(n)}_{absorption} = \left(\sum_{\substack{\text{all} \\ \text{absorptions}}} \frac{wgt_j}{\Sigma_A} \cdot \nu\Sigma_F \right) \Big/ W$$

Cumulative Eigenvalue Estimators

- 7 estimators of overall Keff -- $K_{p\text{-}c\text{-}a}$ is best
- Single estimators (ignore)

K_{path} = average over active cycles of $K^{(n)}_{path}$
$K_{collision}$ = average over active cycles of $K^{(n)}_{collision}$
$K_{absorption}$ = average over active cycles of $K^{(n)}_{absorption}$

- Double estimators (ignore)

$K_{p\text{-}c}$ = average over active cycles of $K^{(n)}_{path}$ and $K^{(n)}_{coll}$, with correlation
$K_{p\text{-}a}$ = average over active cycles of $K^{(n)}_{path}$ and $K^{(n)}_{absorp}$, with correlation
$K_{c\text{-}a}$ = average over active cycles of $K^{(n)}_{coll}$ and $K^{(n)}_{absorp}$, with correlation

- Overall Combined Estimator (best, only use this one)

$K_{p\text{-}c\text{-}a}$ = combined average over all active cycles of
$K^{(n)}_{path}$, $K^{(n)}_{collision}$, and $K^{(n)}_{absorption}$, including correlation

Confidence Intervals

- **Confidence interval:**
  Range that contains the true Keff with some specified probability

- **68% Confidence interval**

$$K_{eff} - \sigma \leq K_{true} \leq K_{eff} + \sigma$$

- **95% Confidence interval**

$$K_{eff} - 2\sigma \leq K_{true} \leq K_{eff} + 2\sigma$$

- **99% Confidence interval**

$$K_{eff} - 2.6\sigma \leq K_{true} \leq K_{eff} + 2.6\sigma$$

- **To get better confidence interval (smaller σ), run more histories (more cycles)**

$$\sigma \propto \frac{1}{\sqrt{N}}$$

KCODE & KSRC Cards

MCNP Criticality Input

- **Control for K-effective calculations**

  KCODE card

  – Number of particles per cycle
  – Initial guess for Keff
  – Number of initial cycles to skip
  – Total number of cycles to run

- **Guess for initial source** (only use 1 of these)

  KSRC card

  – Can specify any number of x,y,z points for initial location of fission neutrons

  SRCTP file

  – Can use a file of source points from a previous calculation

  SDEF card

  – Can specify source points should be sampled from a volume (eg, sphere, cylinder, box, etc.)

KCODE Card

KCODE N kest ndiscard ntotal

N = number of particles per cycle,

Typical: 1K - 5K for testing

10K - 100K+ for production

kest = initial guess for Keff

Usually use 1.0

ndiscard = number of inactive cycles to discard before beginning tallies

Determine this from convergence plots!

ntotal = total number of cycles to run,

Should be > ndiscard+100

(see manual for other optional entries)

Initial Source & Energy Spectrum

- **Can specify source points with KSRC or SDEF (not both)**

- **Use KSRC for small systems, & to put points in each irregular lump of fissile material in dispersed systems**

```
ksrc  0 0 0   .5 .5 .25   .1 .1 .1   [etc]
```

  - **For KSRC, source points are reused as needed to get starting locations for all the particles in the initial cycle**

- **Use SDEF for large systems, for uniform volume source in a box, sphere, or cylinder covering the fissile regions**
  - **Recipes for using SDEF volume source on next slide**

- **For the initial cycle only, neutron starting energy is sampled from a Watt fission spectrum. (Other cycles use actual (n,f) data & energy distributions.) Almost always – use default.**

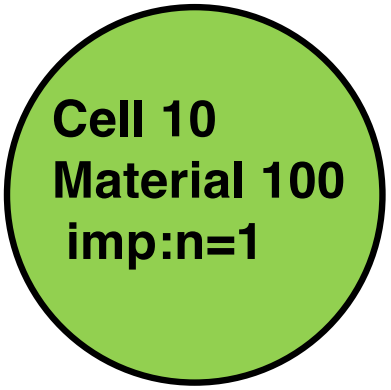
$$p(E) = Ce^{-E/a}\sinh\sqrt{bE}, \qquad a = 0.965\ \text{MeV},\ \ b = 2.29\ \text{MeV}^{-1}$$

Example Problems

Problem g1

Godiva critical -- using KSRC & surfaces

- Bare, high-enriched uranium sphere
- Sphere radius = 8.741 cm
- Material density =18.74 g/cm3
- Nuclide Wgt-fraction ZAID
  - U235 94.73 92235
  - U238 5.27 92238



(1) Create & edit file g1.txt

(2) Add title, cell cards, surface cards, data cards

- surface card (sphere at origin): # so radius

also use these data cards:

- kcode 1000 1.0 10 50
- ksrc 0 0 0

(3) Plot the geometry: mcnp6 i=g1.txt ip

(4) Run the problem: mcnp6 i=g1.txt

(5) Rerun the problem: mcnp6 i=g1.txt

Problem g1

File g1.txt

```
Godiva critical - using ksrc & surfaces
c
c  CELL CARDS
10    100    -18.74    -1
20    0                 1

c  SURFACE CARDS
1     so    8.741

c  DATA CARDS
imp:n  1    0
m100    92235  -94.73       92238  -5.27
c
kcode    1000    1.0    10    50
ksrc  0   0   0
```

Commands

mcnp6 i=g1.txt ip **<-- process input & plot geometry**

mcnp6 i=g1.txt **<-- run the problem...**

Source ?

Files created ?

Comments - g1

- **KSRC 0 0 0**
  - isotropic point source at (0,0,0)
  - used only for the source guess for initial cycle, ignored after that

- **KCODE 1000 1.0 10 50**
  - start 1000 particles
  - run 50 cycles, throw out the first 10
  - initial guess for Keff = 1.0

- **imp:n 1 0**
  - cell 1 has importance 1, cell 2 has importance 0
  - could put this information on cell cards instead:

```
c  CELL CARDS
10   100   -18.74   -1      imp:n 1
20   0                1     imp:n 0
```

- **cleanup:**

  rm out* src* run* com*

Problems g3, g4, g5

Optional

Same as problem g1, but:

g3: Use KSRC card to start points at center

g4: Use SDEF card to start points uniformly in sphere

g5: Use srctp file from g4, don't use KSRC or SDEF

For all: use 5000 neutrons/cycle, skip 50, run 150 total

Problem g3

```
Godiva - using KSRC
c
c CELL CARDS
10     100   -18.74   -1
20     0               1

c SURFACE CARDS
1      so 8.741

c DATA CARDS
kcode 5000 1.0 50 150
ksrc  0. 0. 0.
imp:n 1 0
m100     92235 -94.73    92238 -5.27
```

Problem g4

```
Godiva - using SDEF
c
c CELL CARDS
10     100   -18.74   -1
20     0               1

c SURFACE CARDS
1     so 8.741

c DATA CARDS
kcode 5000 1.0 50 150
Sdef   pos= 0 0 0    rad=d1
Si1    0     8.741
Sp1    -21    2      $ density ~ r**2
imp:n 1 0
m100     92235 -94.73    92238 -5.27
```

Problem g5

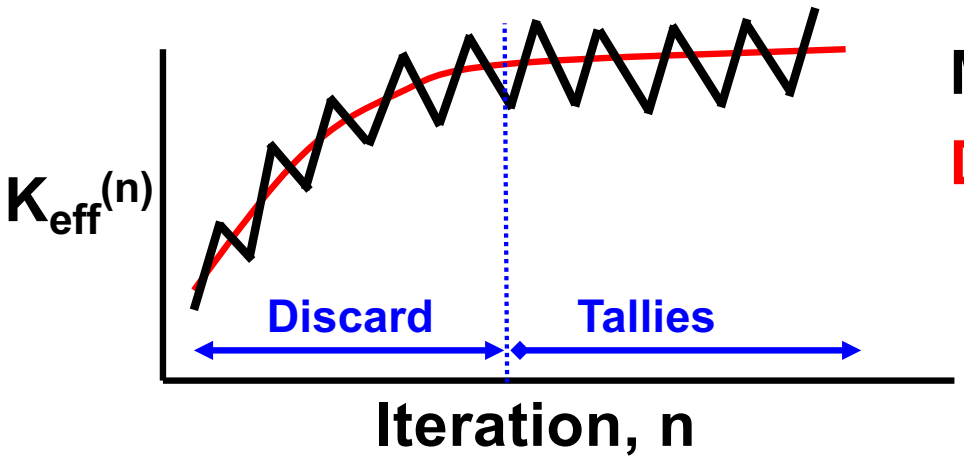
Assuming that problem g4 created source tape file srctw,

```
mcnp5  I=g5.txt   src=srctw

Godiva - using srctp, no ksrc or sdeff
c
c CELL CARDS
10    100  -18.74  -1
20    0             1

c SURFACE CARDS
1    so 8.741

c DATA CARDS
kcode 5000 1.0 50 150   $ 50 could be changed to 0 !
imp:n 1 0
m100    92235 -94.73   92238 -5.27
```

Examine output file
For KCODE calculation
.....

Convergence

Power Iteration



Monte Carlo

Deterministic (S_n)

- **Guess an initial source distribution**
- **Iterate until converged** **(How do you know ???)**
- **Then**
  - **For S_n code:** **done, print the results**
  - **For Monte Carlo:** **start tallies, keep running until uncertainties small enough**

- **Convergence? Stationarity? Bias? Statistics?**

K-eigenvalue equation

Transport equation, for K-eigenvalue problems

- **Use operator (or matrix) form to simplify notation**

$$(L+T)\Psi = S\Psi + \frac{1}{K_{eff}} M\Psi$$

where L = leakage operator S = scatter-in operator

T = collision operator M = fission multiplication operator

- **Rearrange**

$$(L+T-S)\Psi = \frac{1}{K_{eff}} M\Psi$$

➜ This eigenvalue equation will be solved by **power iteration**

$$(L+T-S)\Psi^{(n+1)} = \frac{1}{K_{eff}^{(n)}} M\Psi^{(n)}$$

Power Iteration - Convergence

- **Expand Ψ in terms of eigenfunctions $u_j(r,E,\Omega)$**

$$\Psi = \vec{u}_0 + \sum_{j=1}^{\infty} a_j \vec{u}_j = \vec{u}_0 + a_1\vec{u}_1 + a_2\vec{u}_2 + a_3\vec{u}_3 + ...$$

$$\int \vec{u}_j \vec{u}_k \, dV = \delta_{jk} \qquad a_j = \int \Psi \cdot \vec{u}_j \, dV$$

$$\vec{u}_j = \frac{1}{k_j} F \cdot \vec{u}_j \qquad k_0 > k_1 > k_2 > \ldots > 0 \qquad k_0 \equiv k_{effective}$$

Define: $\rho_j = \frac{k_j}{k_0}, \qquad \rho_1 = \frac{k_1}{k_0}, \qquad 1.0 > \rho_1 > \rho_2 > \rho_3 > \ldots$

- **Expand the initial guess in terms of the eigenmodes**

$$\Psi^{(0)} = \vec{u}_0 + \sum_{j=1} a_j^{(0)} \vec{u}_j$$

- **Substitute expansion for Ψ $^{(0)}$ into power iteration equation**

$$\Psi^{(n+1)} = \frac{1}{K^{(n)}} F \cdot \Psi^{(n)} = \frac{1}{k^{(n)}} \cdot \frac{1}{k^{(n-1)}} \ldots \frac{1}{k^{(0)}} \cdot F^n \cdot \Psi^{(0)}$$

$$\approx [\text{constant}] \cdot \left[\vec{u}_0 + a_1^{(0)} \rho_1^{n+1} \vec{u}_1 + a_2^{(0)} \rho_2^{n+1} \vec{u}_2 + \ldots \right]$$

Power Iteration - Convergence

$$\Psi^{(n+1)} \approx [\text{constant}] \cdot \left[\vec{u}_0 + a_1^{(0)} \rho_1^{n+1} \vec{u}_1 + \ldots \right]$$

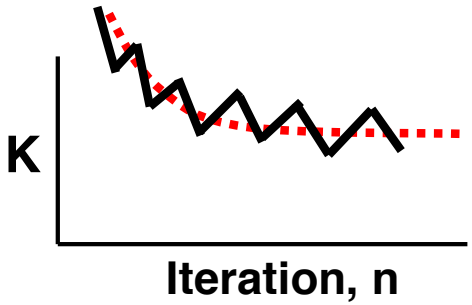
$$K^{(n+1)} \approx k_0 \cdot \left[1 - a_1^{(0)} \rho_1^{n} (1-\rho_1) G_1 + \ldots \right]$$

- u_0 & k_0 are the exact steady-state solution
- Convergence = higher modes have died away & are negligible
- Because $1 > \rho_1 > \rho_2 > \rho_3 > \ldots$, all of the red terms vanish as $n \rightarrow \infty$
  - $\psi^{(n+1)} \rightarrow u_0$
  - $K^{(n+1)} \rightarrow k_0$
- After the initial transient, error in $\Psi^{(n)}$ is dominated by first mode
  - (k_1 / k_0) is called the dominance ratio, DR or ρ
  - Errors in the source distribution $\Psi^{(n)}$ die off as ~ $(DR)^n$
- For problems with a high dominance ratio (e.g., DR ~ .99), the error in K_{eff} may be small, since the factor (1-ρ) is small.
  - K_{eff} may appear converged, even if the source distribution is not converged

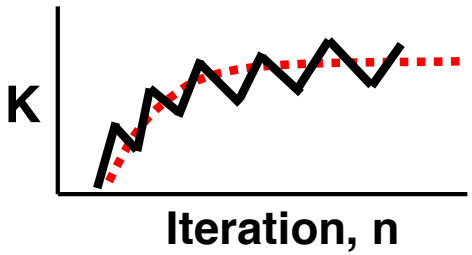
Typical K-effective convergence patterns

- **Higher mode error terms die out as $(k_1 / k_0)^n$, for n iterations**
  - **k_1 is the eigenvalue of the first higher mode, $k_0 = k_{eff}$**
  - **k_1 / k_0 is called the Dominance Ratio**

- **When initial guess is concentrated in center of reactor, initial K_{eff} is too high** (underestimates leakage)



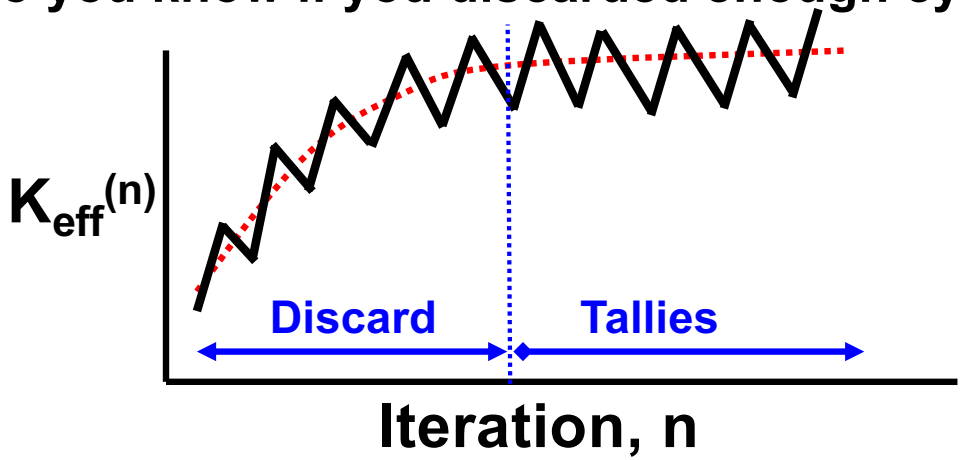
- **When initial guess is uniformly distributed, initial K_{eff} is too low** (overestimates leakage)



- **The Sandwich Method uses 2 K_{eff} calculations - one starting too high & one starting too low. Both calculations should converge to the same result. (Sometimes useful for difficult problems.)**

Keff Calculations

- **Initial cycles of a Monte Carlo K-effective calculation should be discarded, to avoid contaminating results with errors from initial guess**
  - **How many cycles should be discarded?**
  - **How do you know if you discarded enough cycles?**

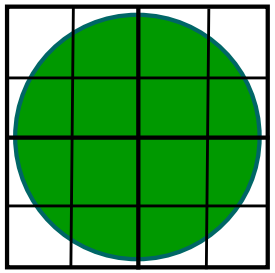


- **Analysis of the power iteration method shows that Keff is not a reliable indicator of convergence -- K_{eff} can converge faster than the source shape**

- **Based on concepts from information theory, Shannon entropy of the source distribution is useful for characterizing the convergence of the source distribution**

Shannon Entropy of the Fission Source

- **Divide the fissionable regions of the problem into N_S spatial bins**
  - Spatial bins should be consistent with problem symmetry
  - Typical choices: -- 1 bin for each assembly
    -- regular grid superimposed on core
  - Use dozens or hundreds of bins, not thousands



- **During the random walks for a cycle, tally the fission source points in each bin**
  - Provides a discretized approximation to the source distribution
  - $\{ p_J, \quad J=1,N_S \}$

- **Shannon entropy of the source distribution**

$$H(S) = -\sum_{J=1}^{N_S} p_J \cdot \ln_2(p_J), \quad \text{where} \quad p_J = \frac{(\text{\# source particles in bin J})}{(\text{total \# source particles in all bins})}$$

Shannon Entropy of the Fission Source

- **Shannon entropy of the source distribution**

$$H(S) = -\sum_{J=1}^{N_S} p_J \cdot \ln_2(p_J), \quad \text{where} \quad p_J = \frac{(\text{\# source particles in bin J})}{(\text{total \# source particles in all bins})}$$

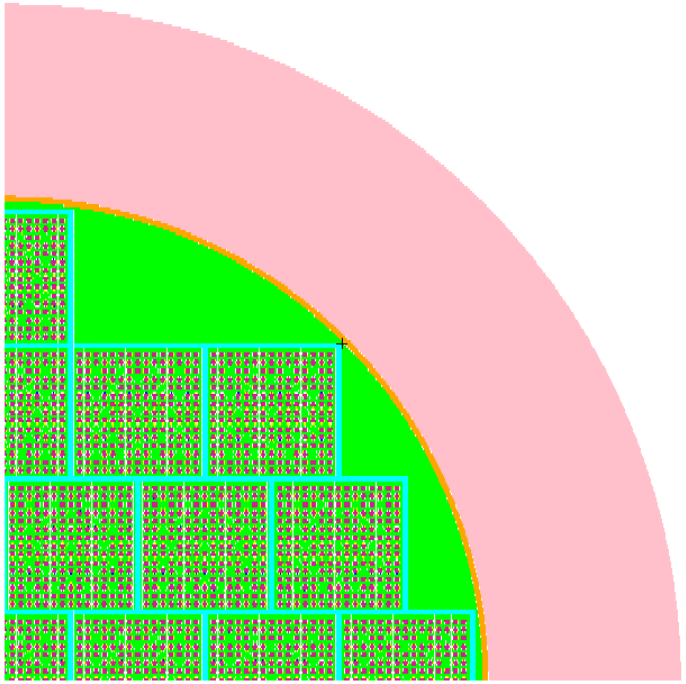
  - **$0 \le H(S) \le \ln_2(N_S)$**
  - **For a <u>uniform</u> source distribution, $H(S) = \ln_2(N_S)$**
  - **For a <u>point</u> source (in a single bin), $H(S) = 0$**

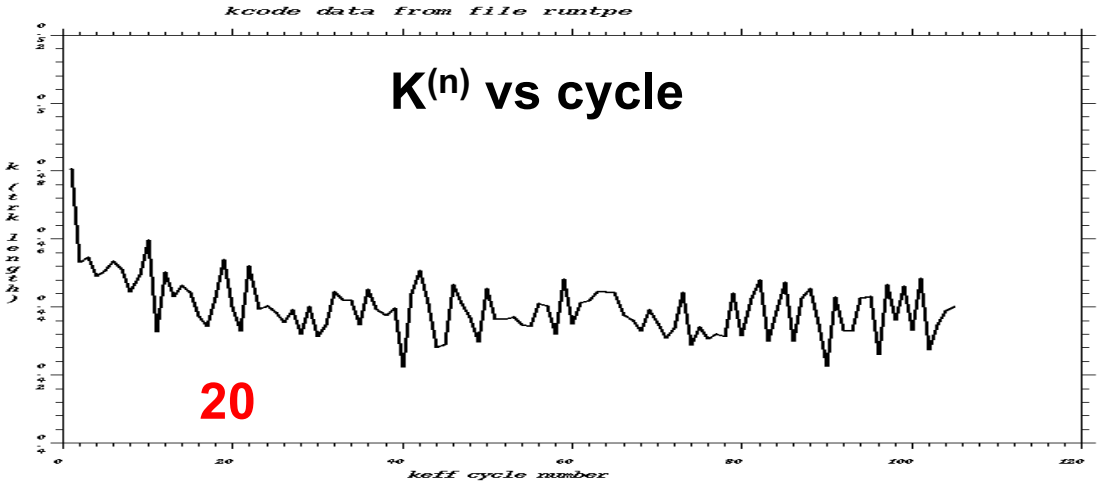
- **$H(S^{(n)})$ provides a single number to characterize the source distribution for iteration n (no physics!)**

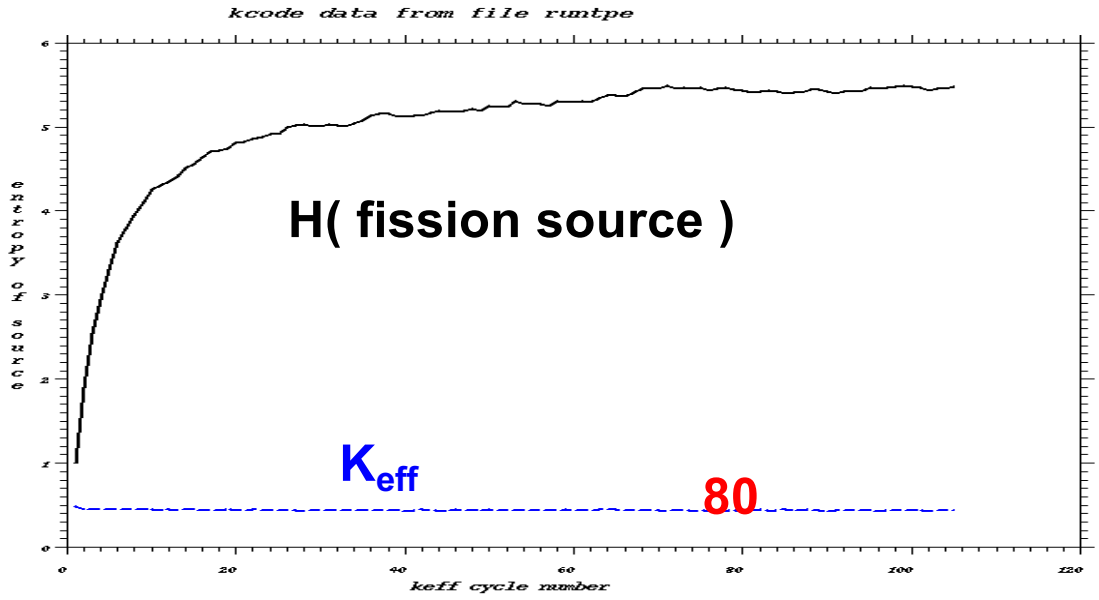
➜ As the source distribution converges in 3D space, a line plot of $H(S^{(n)})$ vs. n (the iteration number) converges

Criticality Calculations - Convergence

- **Reactor core (Problem inp24)**



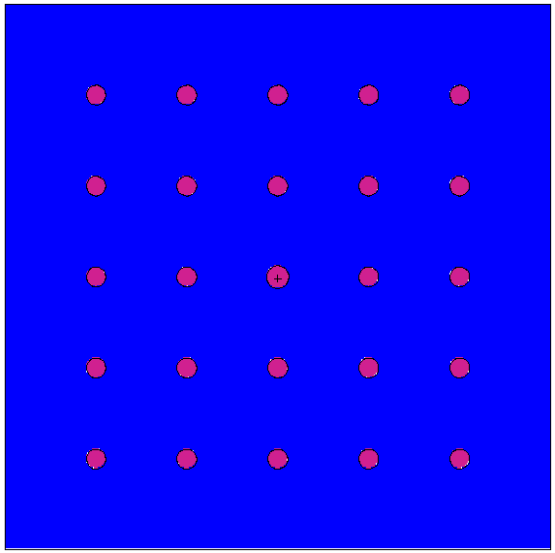




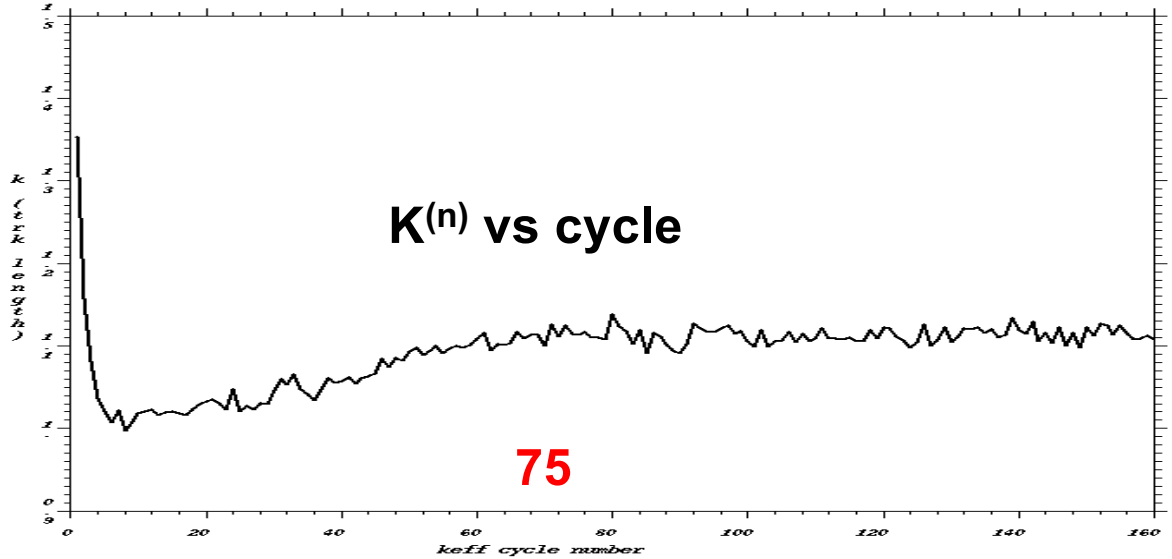
DR = .98

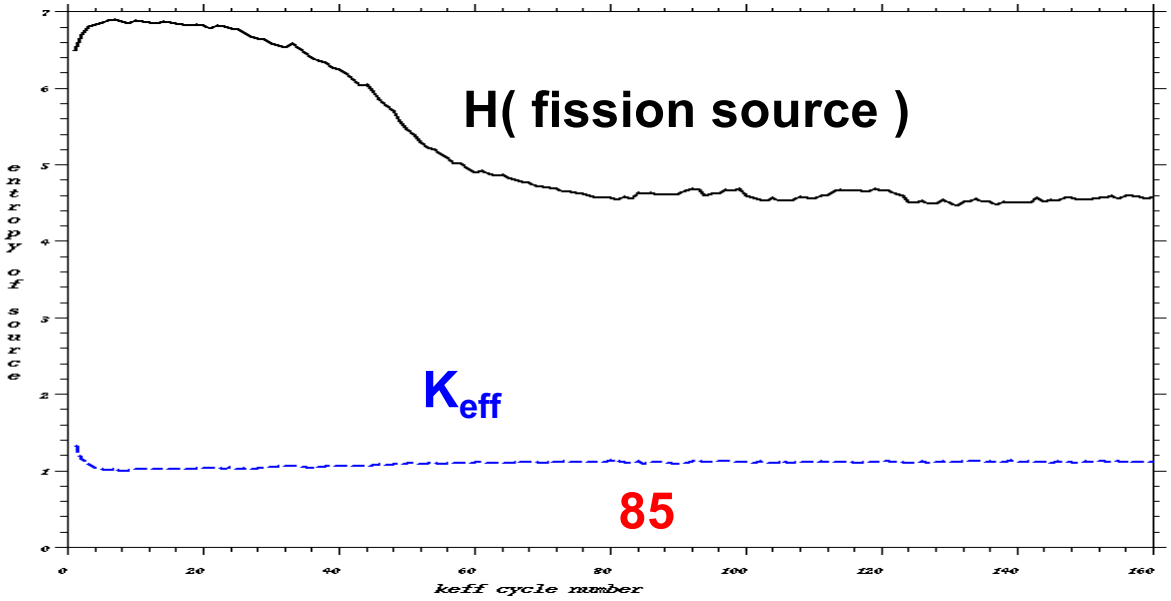
Criticality Calculations - Convergence

- **Loosely-coupled array of spheres (Problem test4s)**



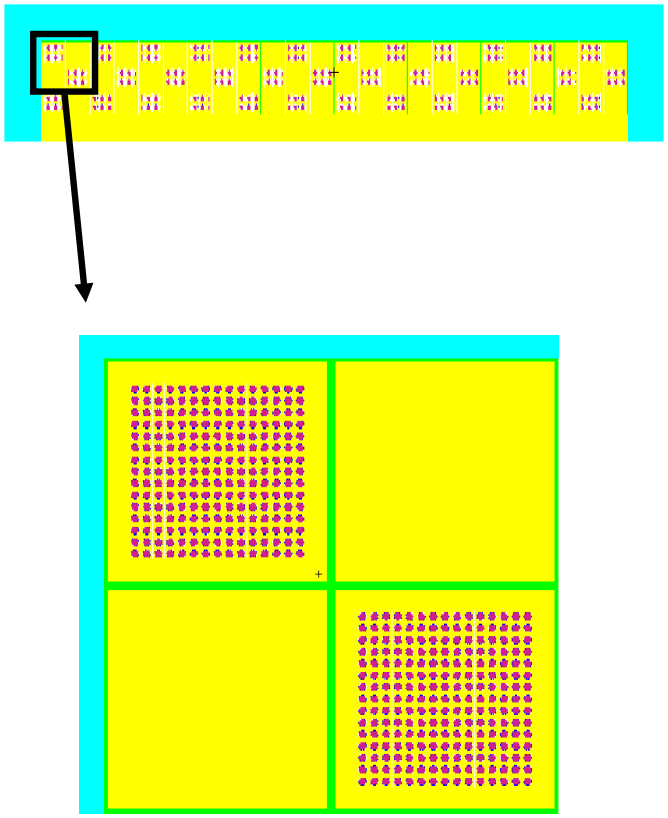
DR = .91

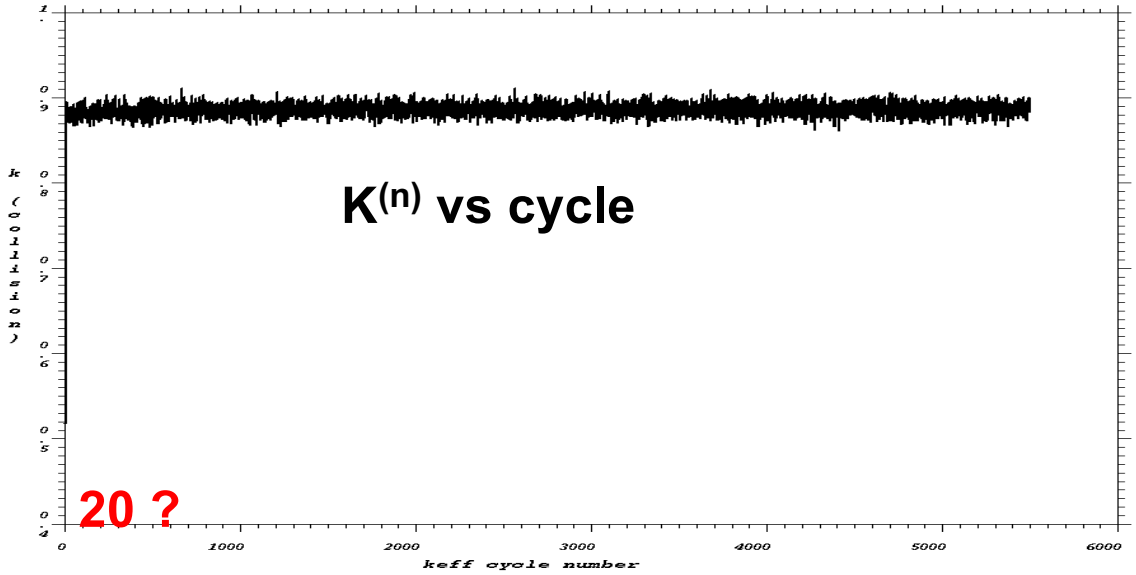


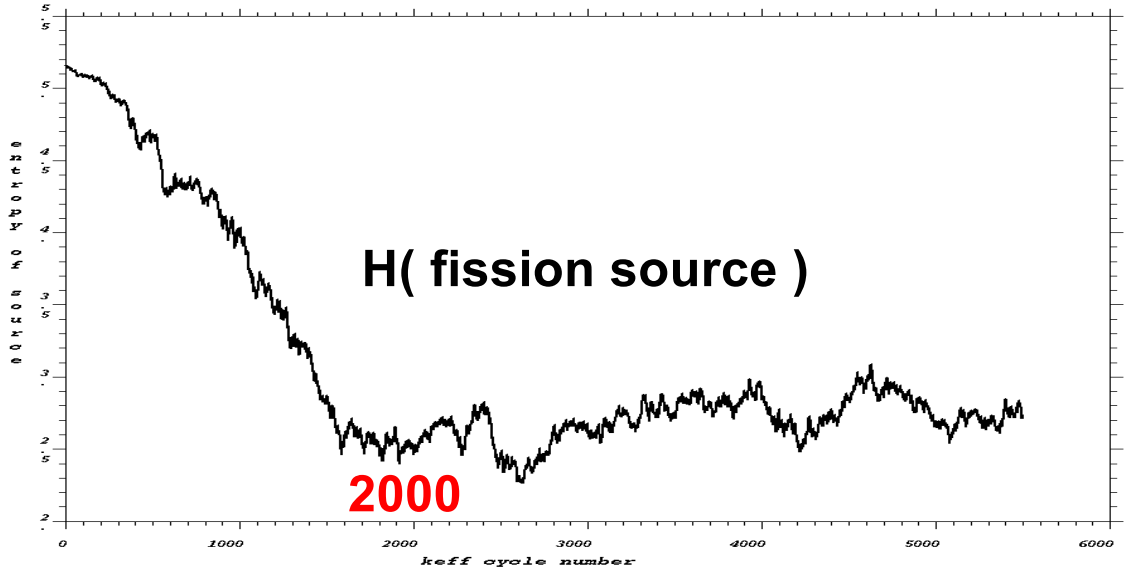


Criticality Calculations - Convergence

- **Fuel Storage Vault (Problem OECD_bench1)**



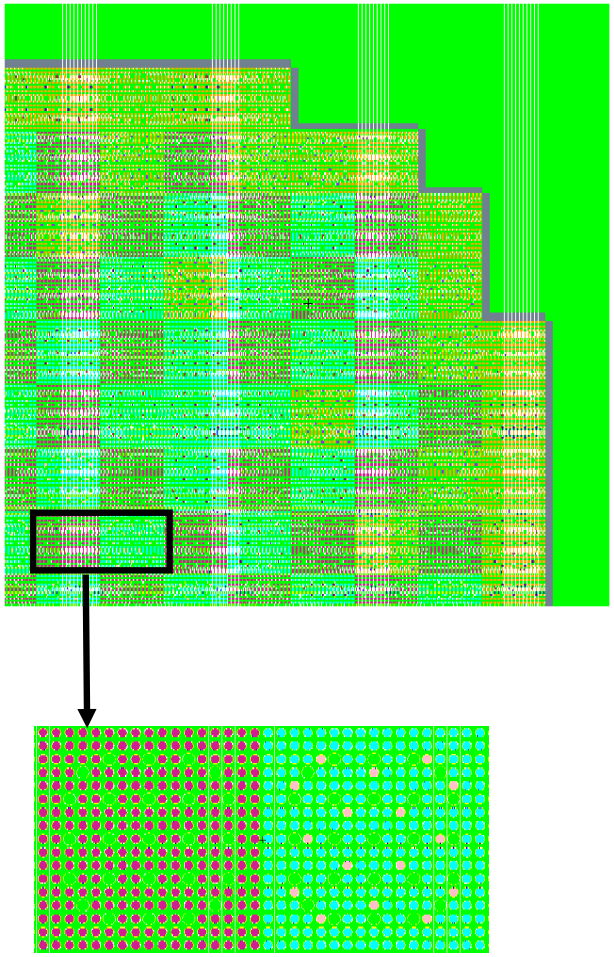


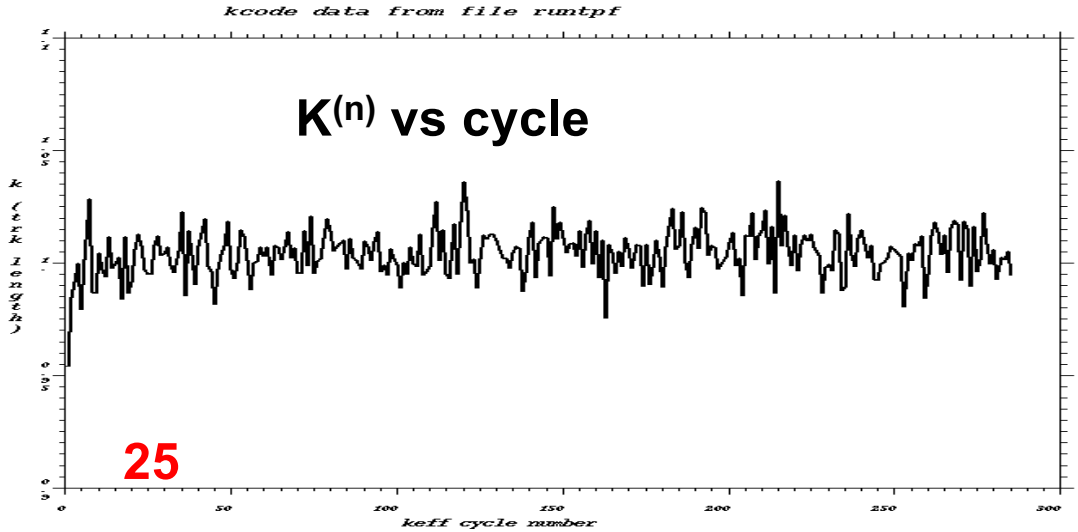


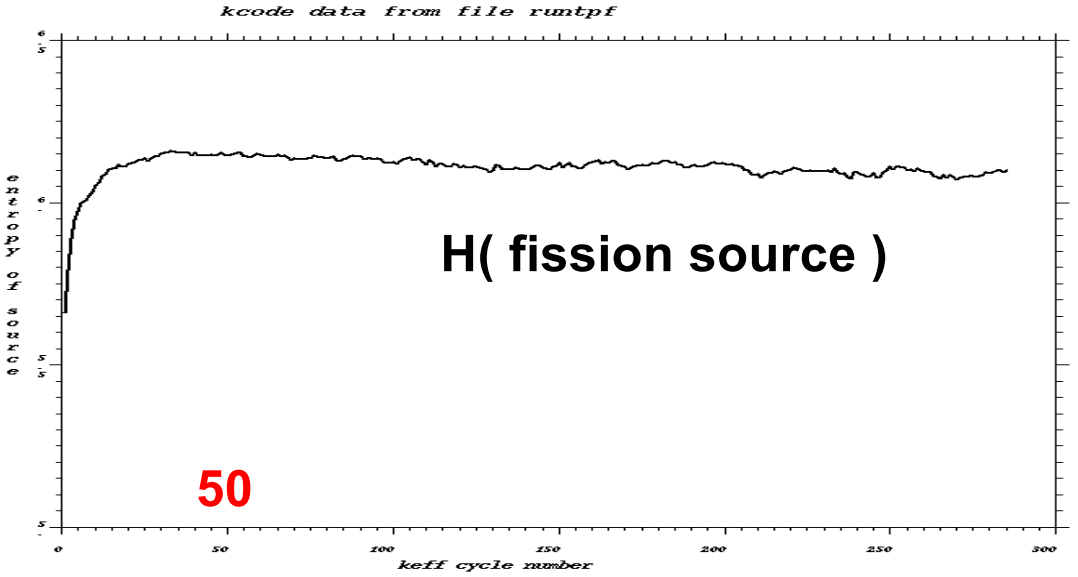
DR = .99+

Criticality Calculations - Convergence

- **PWR 1/4-Core (Napolitano)**



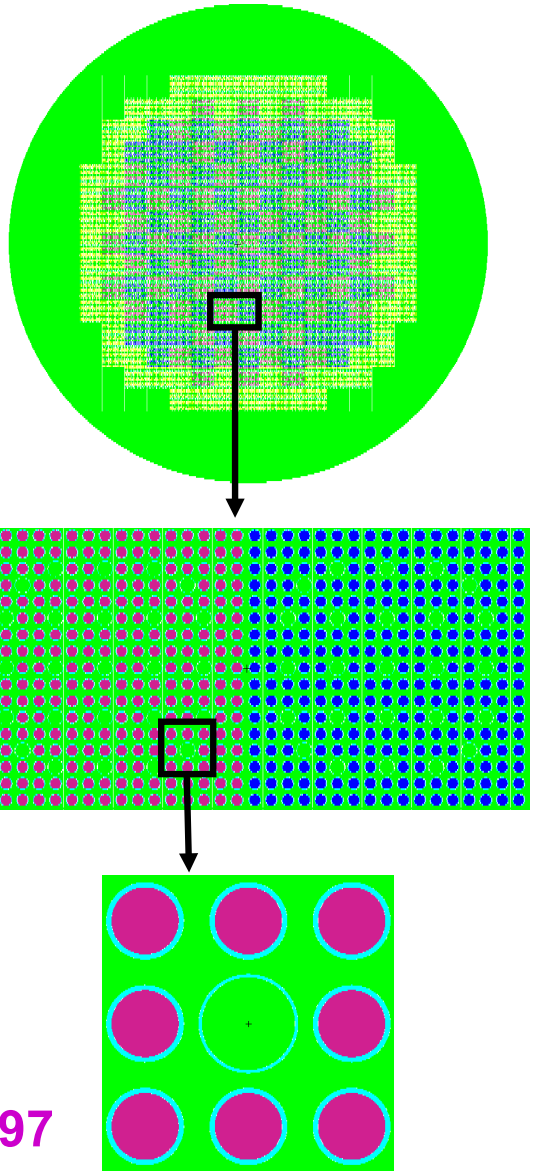




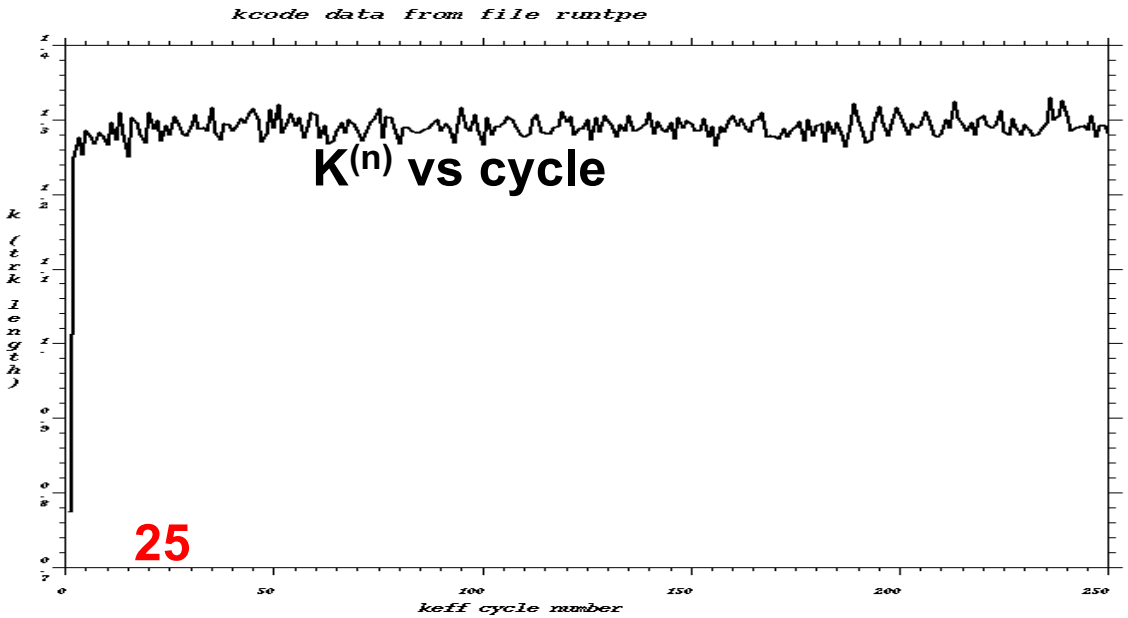
DR = .95

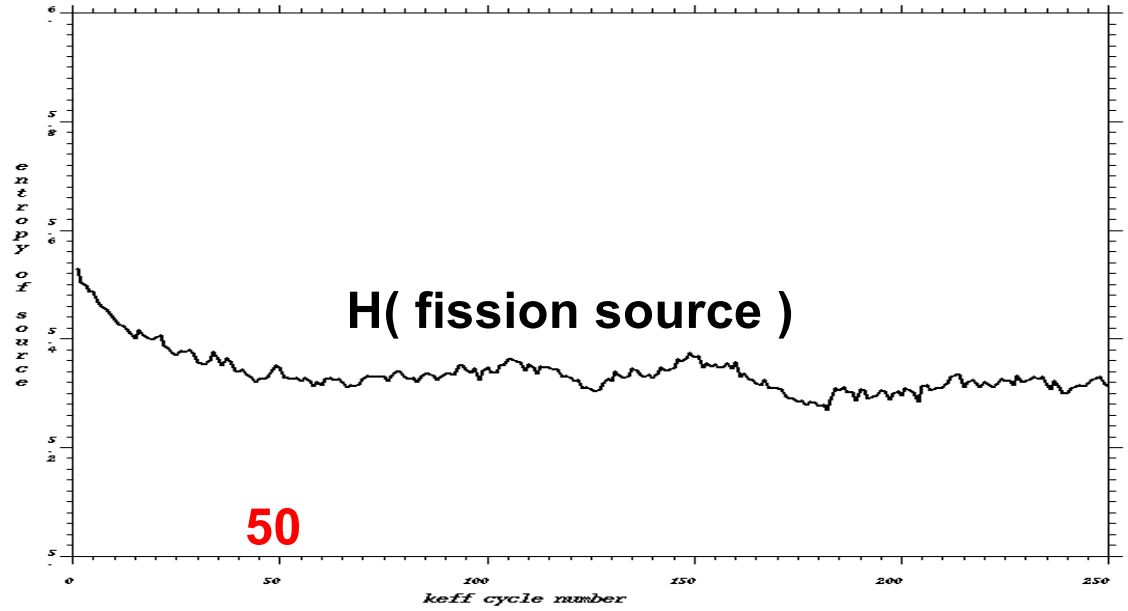
Criticality Calculations - Convergence

- 2D PWR (Ueki)



DR = .97





Guidance on Computing H_{src}

Source Entropy & MCNP5

- **Grid for computing H_{src}**
  - User can specify a rectangular grid in input

    hsrc n_x x_{min} x_{max} n_y y_{min} y_{max} n_z z_{min} z_{max}

    example: hsrc 5 0. 100. 5 0. 100. 1 -2. 50.

  - If hsrc card is absent, MCNP5 will choose a grid based on the fission source points, expanding it if needed during the calculation

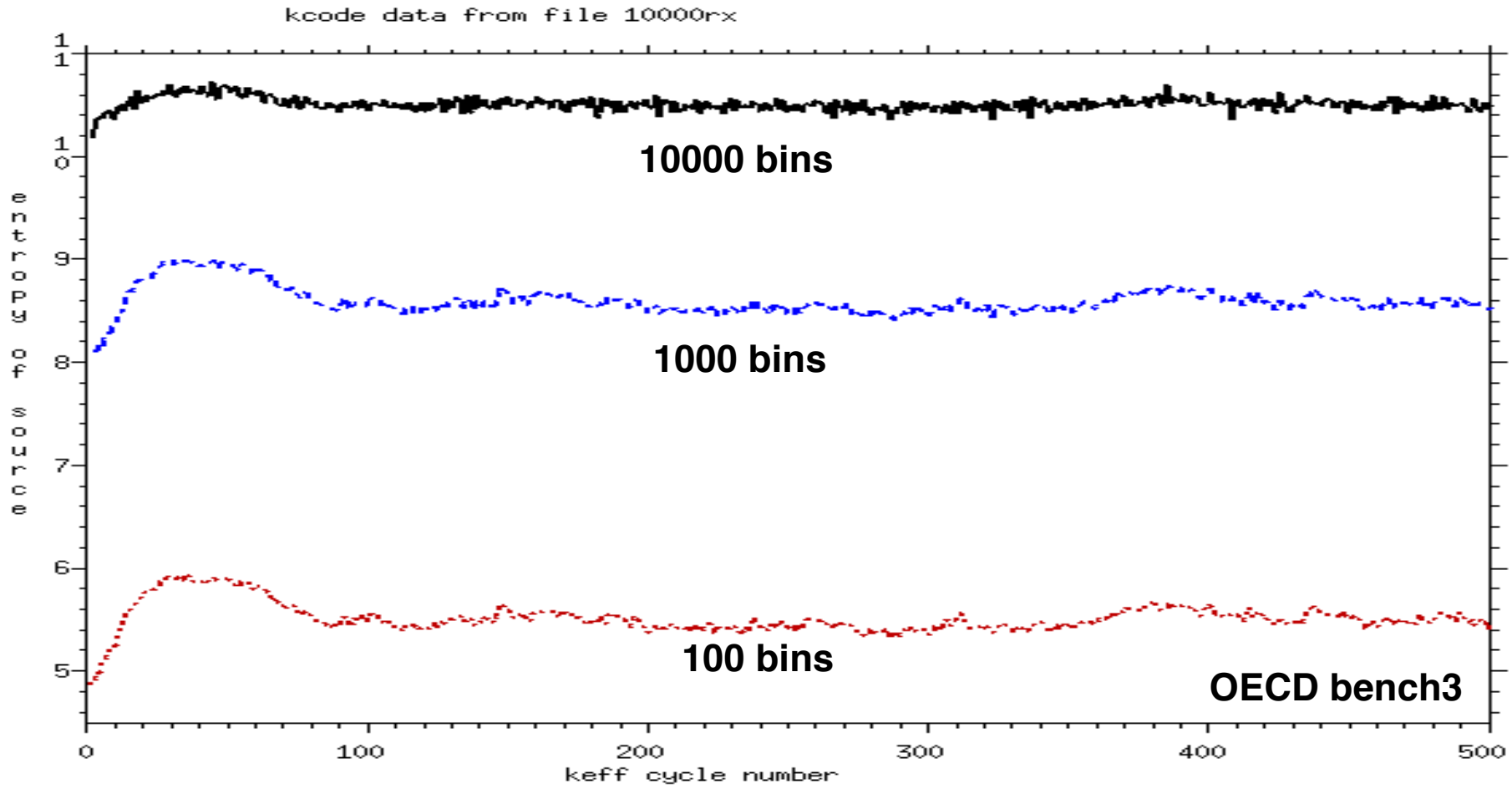
- **MCNP6 prints H_{src} for each cycle**

- **MCNP6 can plot H_{src} vs cycle**

- **Convergence check at end of problem**
  - MCNP6 computes the average H_{src} and its population variance σ_H^2 for the last half of the cycles
  - Then, finds the first cycle where H_{src} is within the band $\langle H_{src} \rangle \leq 2\sigma_H$
  - Then, checks to see if at least that many cycles were discarded

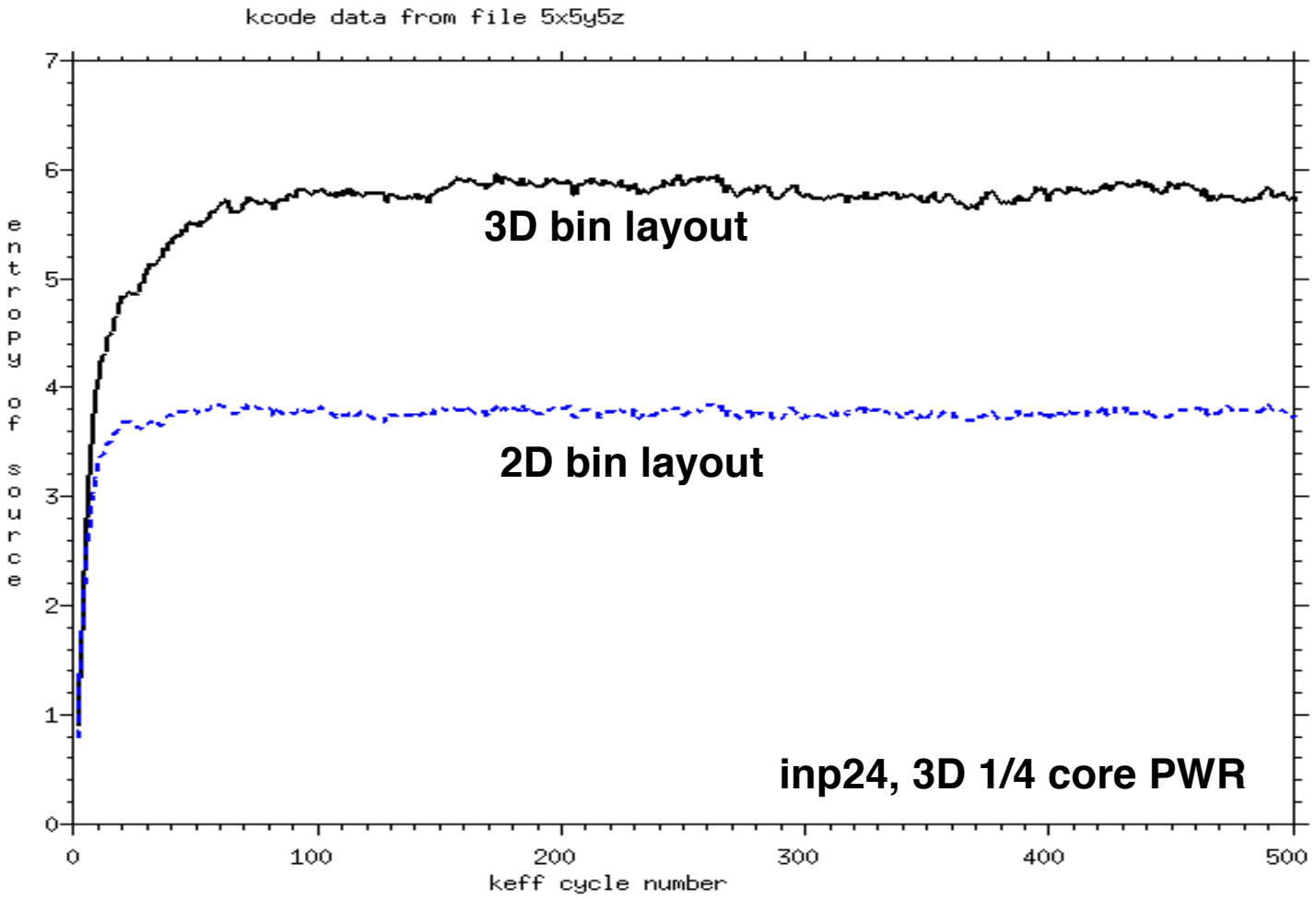
H_{src} Convergence vs Number of Spatial Bins

- For large number of bins, H_{src} approaches uniform upper limit
- Use 10s or 100s of bins, not 1000s or more



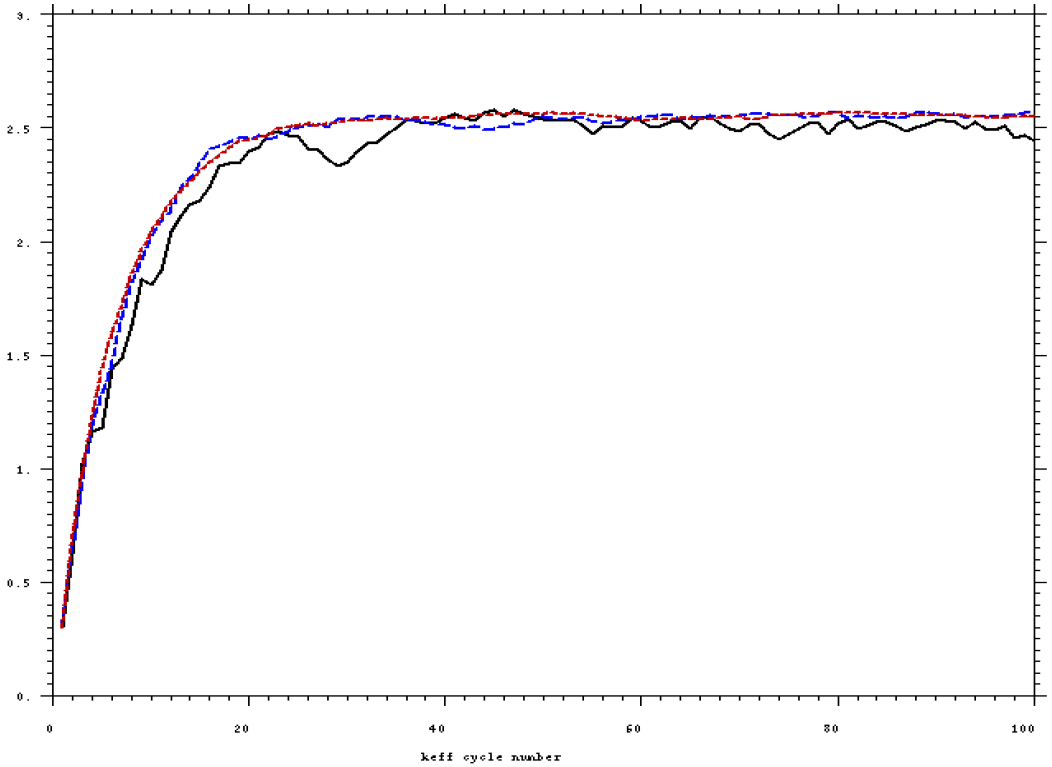
H_{src} and 2D vs 3D Spatial Bins

- For 3D problems, using a 2D bin layout for H_{src} may incorrectly assess convergence
- Important to use 3D bin layout for 3D problems



H_{src} Convergence vs Neutrons per Cycle

- **Problems converge at the same rate, for any number of neutrons/cycle. (The rate depends on dominance ratio, ie, physics & geom)**
- **More neutrons/cycle does not make problems converge faster**
- **More neutrons/cycle ➔ less noise in convergence plots**



Shannon Entropy
Of the Fission Source

for different neutrons / cycle

1000 - black
5000 - blue
20000 - red

Convergence Testing Using H_{src}

Example 1 - New Output

- Information on mesh used for calculating H_{src}

```
comment.
comment. entropy of the fission source distribution will be computed
comment.
comment. the mesh for source entropy is based on the site coordinates
comment.   using  4 x  4 x  4 =    64  mesh cells
comment.
comment.     Xmin=  2.2973E+01     Xmax=  3.6787E+02
comment.     Ymin=  5.3433E+01     Ymax=  1.8017E+02
comment.     Zmin= -1.1753E+01     Zmax=  1.1701E+01
comment.
comment. the mesh will be automatically expanded if necessary to
comment.   encompass the entire fission source distribution
comment.

cycle    k(col)        ctm   entropy    active     k(col)    std dev
    1   1.35561       0.00  1.58E+00
    2   1.12276       0.01  1.77E+00
  . . . . .
```

Example 1 - New Output (cont'd)

- At end of run, information on convergence, based on H_{src}

```
 . . . . .
 129   1.10703      0.17   2.43E+00        99    1.06324    0.00502
 130   1.14460      0.17   2.40E+00       100    1.06405    0.00503

source distribution to file srctt           cycle =   130
run terminated when     130 kcode cycles were done.
comment.
comment. Average fission-source entropy for the last half of cycles:
comment.      H=  3.02E+00  with population std.dev.=  4.97E-01
comment.
comment.
comment. Cycle   47 is the first cycle having fission-source
comment.   entropy within 1 std.dev. of the average
comment.   entropy for the last half of cycles.
comment.   At least this many cycles should be discarded.
warning.
warning. The fission-source entropy for the first active cycle, cycle=  31,
warning.   is NOT within 1 standard deviation of the average
warning.   source entropy for the last half of cycles.
warning.
warning. You should consider rerunning the problem,
warning.   discarding more initial cycles.
warning.
```

Example 2 - New Output

- At end of run, information on convergence, based on H_{src}

```
. . . . .
  174   1.15010      0.20  2.76E+00        99    1.11162    0.00292
  175   1.13930      0.21  2.73E+00       100    1.11190    0.00291
source distribution to file srctu           cycle =   175
run terminated when     175 kcode cycles were done.

comment.
comment. Average fission-source entropy for the last half of cycles:
comment.      H=  2.66E+00  with population std.dev.=  2.21E-01
comment.
comment.
comment. Cycle   36 is the first cycle having fission-source
comment.   entropy within 1 std.dev. of the average
comment.   entropy for the last half of cycles.
comment.   At least this many cycles should be discarded.
comment.
comment. Source entropy convergence check passed.
comment.
```

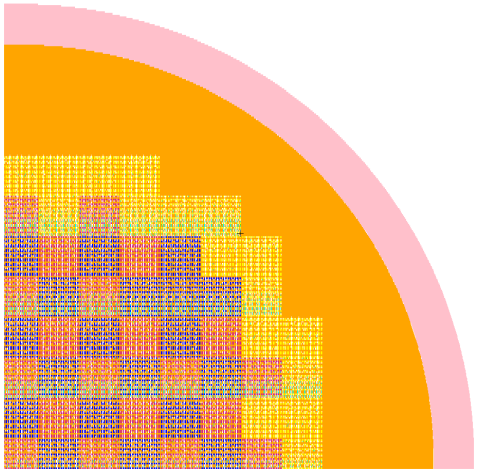
Example

- For Watts-Bar Unit #1, 3D whole-core model

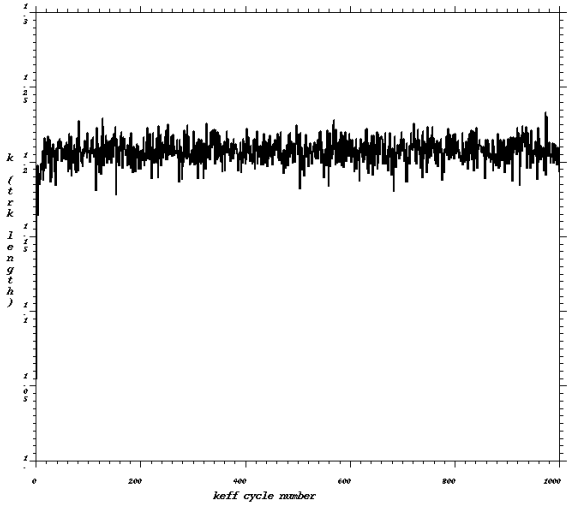
```
c
c   Mesh tally for assembly powers   (flux*fission*Q)
fmesh104:n  geom=xyz  origin= -161.25 -161.25 -194.492
            imesh=161.25 iints=15
            jmesh=161.25 jints=15
            kmesh=170.508 kints=1
fm104  -1.0  0  -6 -8       $ NOTE: 0 = use actual materials
c
c  Mesh tally for detailed fast & thermal flux
fmesh204:n  geom=xyz  origin= -161.25 -161.25 -194.492
            imesh=161.25 iints=150
            jmesh=161.25 jints=150
            kmesh=170.508 kints=1
            emesh  .625e-6  20.
```

Example (cont'd)

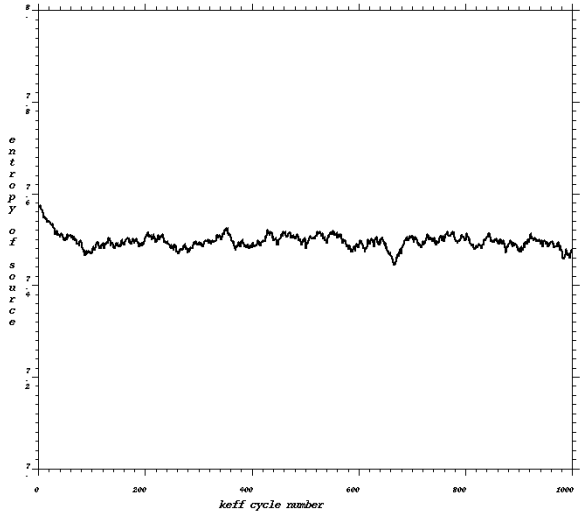
Geometry Model (1/4)



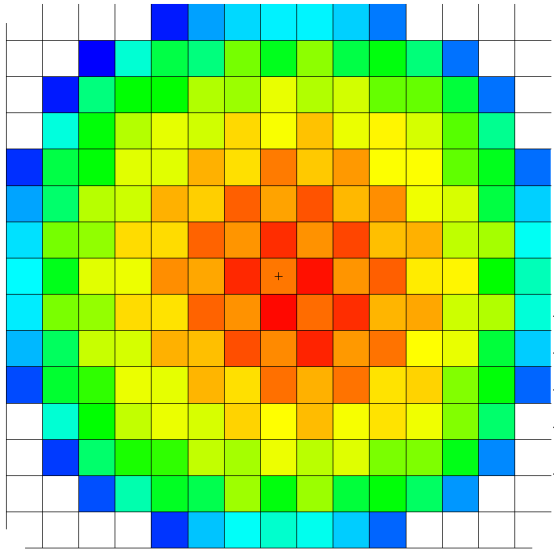
K vs cycle



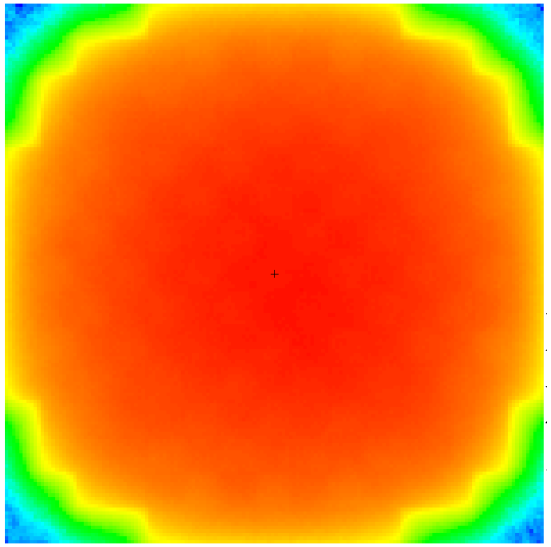
H_{src} vs cycle



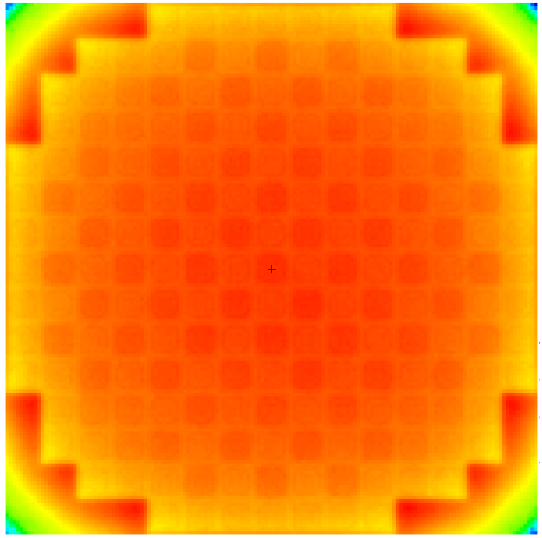
Assembly Powers



Fast Flux



Thermal Flux



Plotting

Criticality Calculation Workflow

- **Plot the geometry**
- **Make a trial run**
  - 50-100 cycles
  - 1000-5000 neutrons/cycle
- **Examine plots of:**
  - Keff vs cycle number
  - Shannon entropy of fission distribution vs cycle
- **Fix the KCODE card:**
  - Make sure that enough initial cycles are discarded
  - Change neutrons/cycle to 10K or more
- **Run calculation until statistics are small enough**
- **Look at convergence plots again, to verify…**

Criticality Output and Plotting

- **To invoke the MCNP tally plotter:**
  - While problem is running, type CTL-C, then m, or
  - After run complete, type: mcnp5 r=runtpe z
- **Commands for the tally plotter:**

  mcplot> kcode n

kcode 1	collision estimate of single-cycle K vs cycle
kcode 2	absorption estimate of single-cycle K vs cycle
kcode 3	track-length estimate of single-cycle K vs cycle
kcode 11-13	same as 1, 2, 3, but cumulative averages vs cycle
kcode 6	Shannon entropy of fission distribution vs cycle
kcode 16	Combined trk/abs/col ave. K vs cycle, with std dev
kcode 17 (what if)	Combined trk/abs/col ave K vs cycles skipped

- **Can plot several quantities together**

  mcplot> kcode 1 coplot kcode 2 coplot kcode 3
- **See MCNP Manual**

Convergence & Plotting Exercise

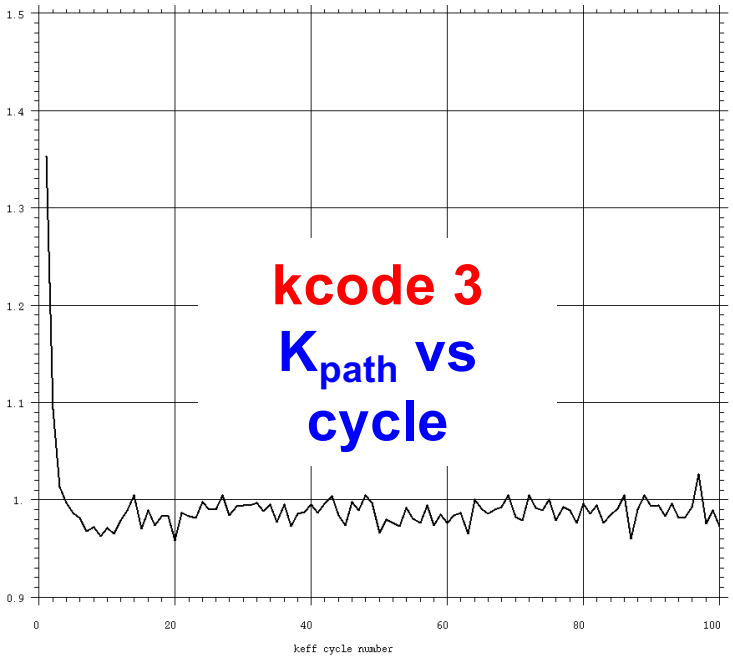
- **Make keff plots for Problem puc6.txt**

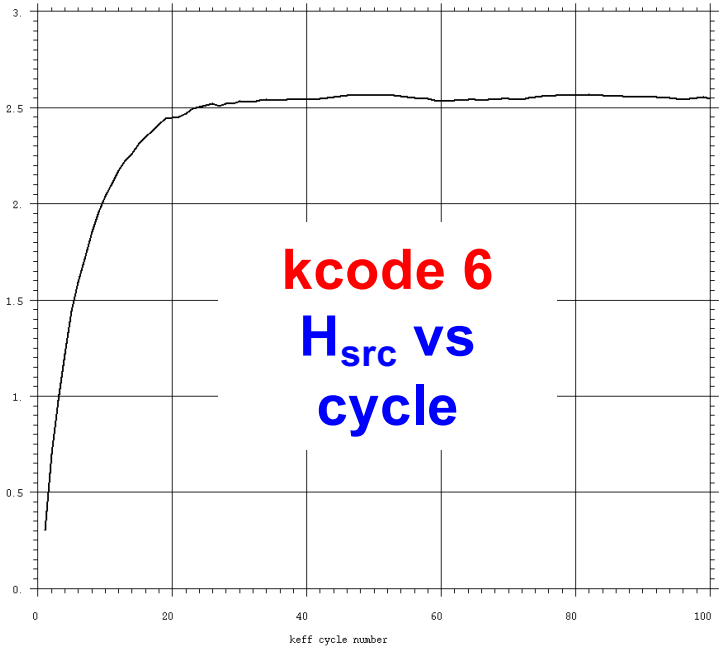
```
mcnp6   i= puc6.txt

mcnp6 r=runtpe  Z
mcplot>  kcode 3
mcplot>  scales 2
mcplot>  kcode 6
mcplot>  kcode 16
mcplot>  kcode 1 coplot kcode 2 coplot kcode 3
```

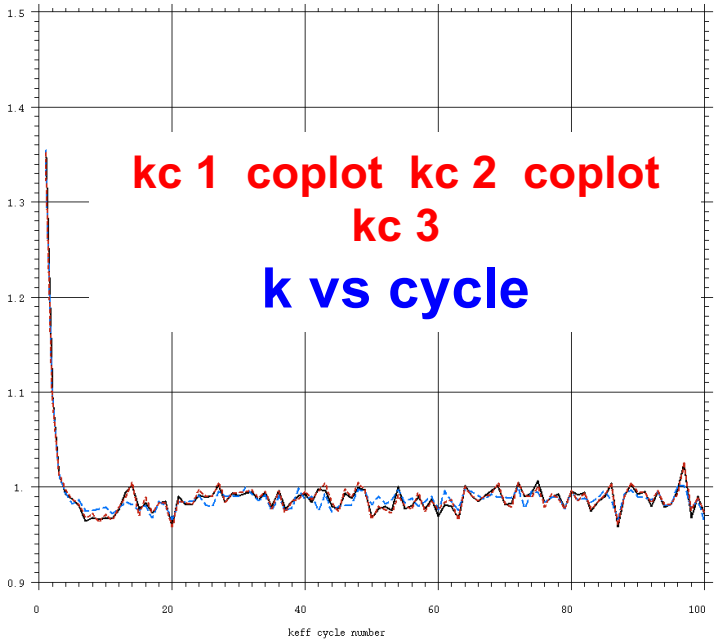
  - Note that it takes 40-50 cycles to converge H_{src}
  - Third entry on KCODE card (ndiscard) should be changed to 50

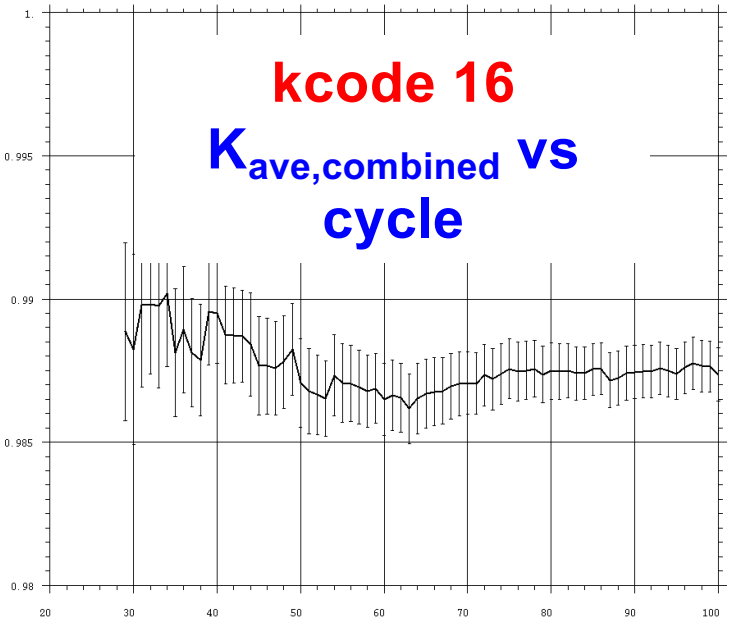
- **Examine & run & plot problem puc6k.txt**
  - Problem puc6 had 6 cans of fissile solution, with an initial source point in only 1 of the cans.
  - If the KSRC card is changed to start with 6 source points, 1 in each can, then the problem converges much quicker.

Convergence Plots for puc6

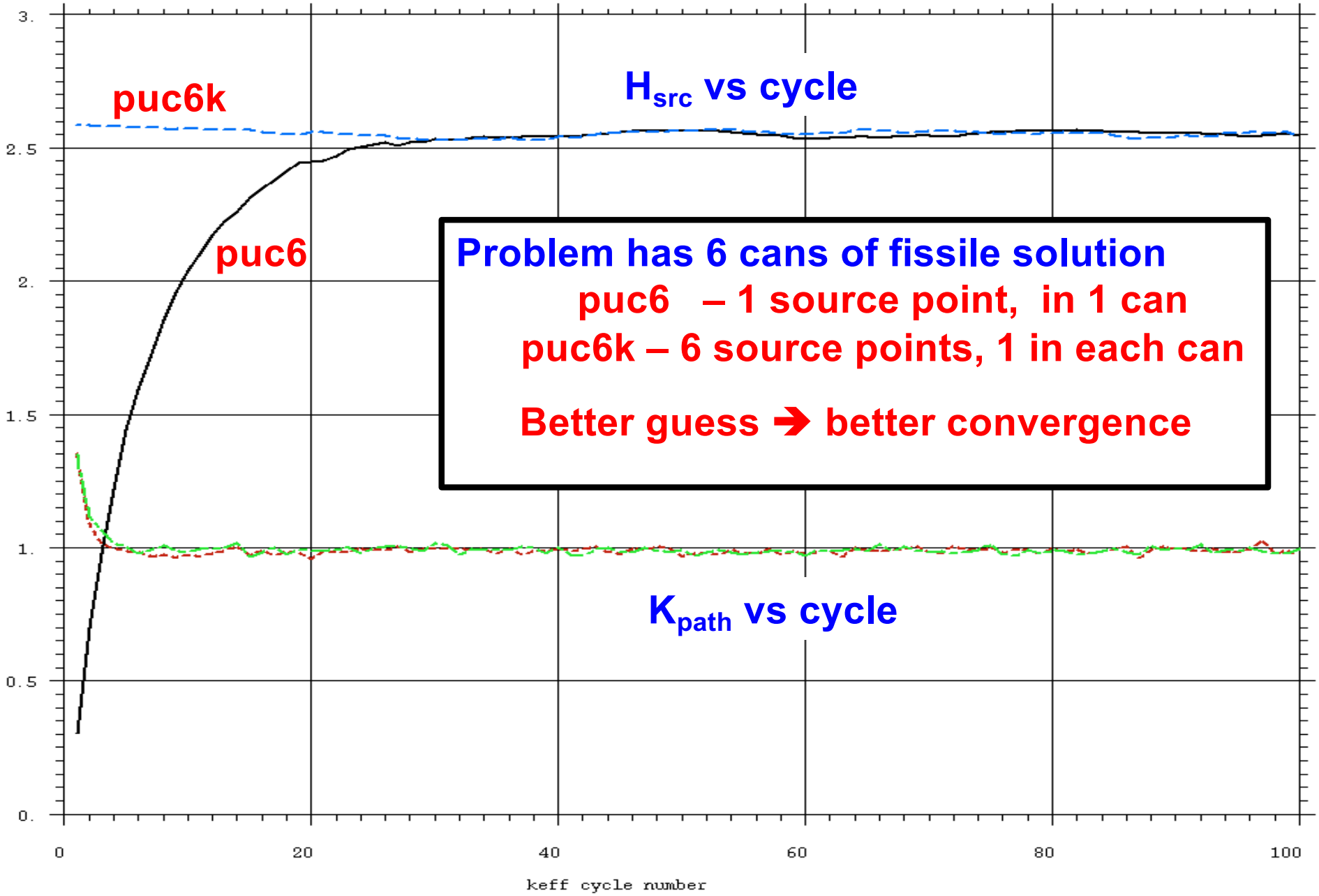








Convergence & Source Guess



K-Eigenvalue Equations

Monte Carlo Solution of K_{eff} Problems

Note: batch = cycle = iteration = generation

- **Initialize**
  - Assume a value for the initial K_{eff} (usually, $K_0 = 1$)
  - Sample M fission sites from the initial source distribution
- **For each cycle n, n = 1 … N+D**
  - Follow histories for all source particles in cycle
    - If fissions occur, bank the sites for use as source in next cycle
    - Make tallies for $K_{cycle}^{(n)}$ using path, collision, & absorption estimators
    - If $n \leq D$, discard any tallies
    - If $n > D$, accumulate tallies
  - Estimate $K_{cycle}^{(n)}$
- **Compute final results & statistics using last N cycles**

K-Calculations -- Banking Fission Sites

- During a particle random walk,

$$\text{wgt} \cdot \frac{\nu\Sigma_F}{\Sigma_T}$$ = expected number of fission neutrons created at collision point

- Averaged over all collisions for all histories, the expected value for $\text{wgt}\cdot\nu\Sigma_F/\Sigma_T$ is K_{eff}.

- In order to bank approximately the same number of fission sites in each cycle, the current value of Keff is used to bias the selection of fission sites at a collision:

$$R = \text{wgt} \cdot \frac{\nu\Sigma_F}{\Sigma_T} \cdot \frac{1}{K}, \qquad n = \lfloor R \rfloor$$

If $\xi < R - n$, store $n+1$ sites in bank with $\text{wgt}' = K$

Otherwise, store n sites in bank with $\text{wgt}' = K$

K-Calculations -- Renormalization

- N_J = number of particles starting cycle J,

  N'_J = number of particles created by fission in cycle J (number of particles stored in fission bank)

  – The expected value for N'_J is: $E[N'_J] = K_{eff} \bullet N_J$
  – (N'_J/N_J) is a single-cycle estimator for K_{eff}

- To prevent the number of particles per cycle from growing exponentially (for K>1) or decreasing to 0 (for K<1), the particle population is renormalized at the end of each cycle:
  – In some Monte Carlo codes, the number of particles starting each cycle is a constant N. Russian roulette or splitting are used to sample N particles from the N' particles in the fission bank. (All particles in fission bank have a weight of 1.0)

  – In other codes, the total weight W starting each cycle is constant. The particle weights in the fission bank are renormalized so that the total weight is changed from W' to W. (Particles in fission bank have equal weights, but not necessarily 1.0)

K-Calculations -- Bias

- **The renormalization procedure used at the end of each cycle introduces a small bias into the computed Keff**
  - **Renormalization involves multiplying particle weights by (W/W'), where W = total weight starting a cycle, W'= total weight at the end of a cycle.**
  - **W' is a random variable, due to fluctuations in particle random walks.**

- **Theoretical analysis of the MC iteration process & propagation of history fluctuations gives**

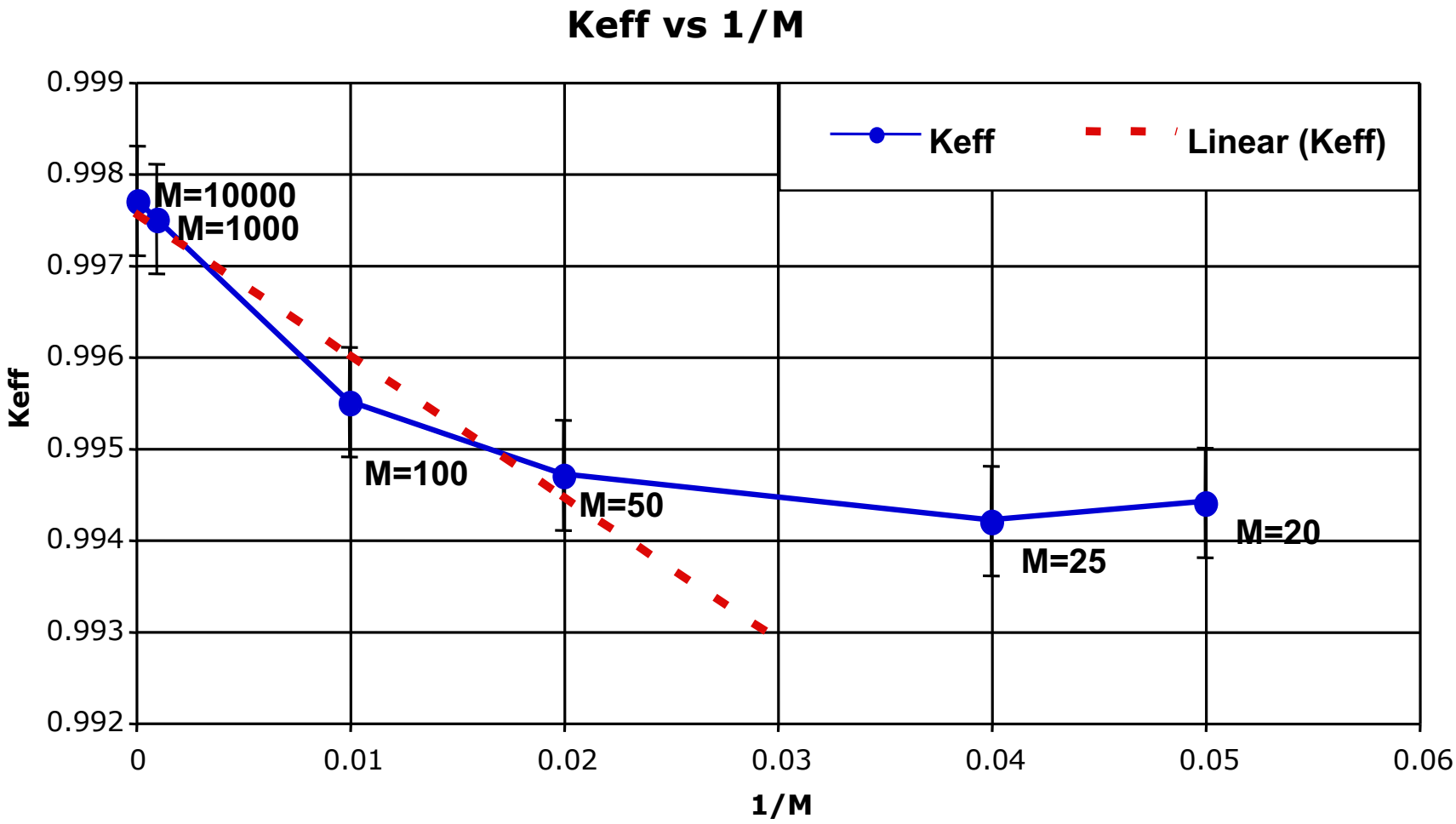
$$\text{bias in } K_{eff} = -\frac{\sigma_k^2}{K_{eff}} \cdot \begin{pmatrix} \text{sum of correlation coeff's} \\ \text{between batch K's} \end{pmatrix}$$

  - **M = histories/cycle**
  - **Bias in Keff ~ 1/M**
    - Smaller M ➜ larger cycle correlation ➜ larger bias in Keff & source
    - Larger M ➜ smaller cycle correlation ➜ smaller bias

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

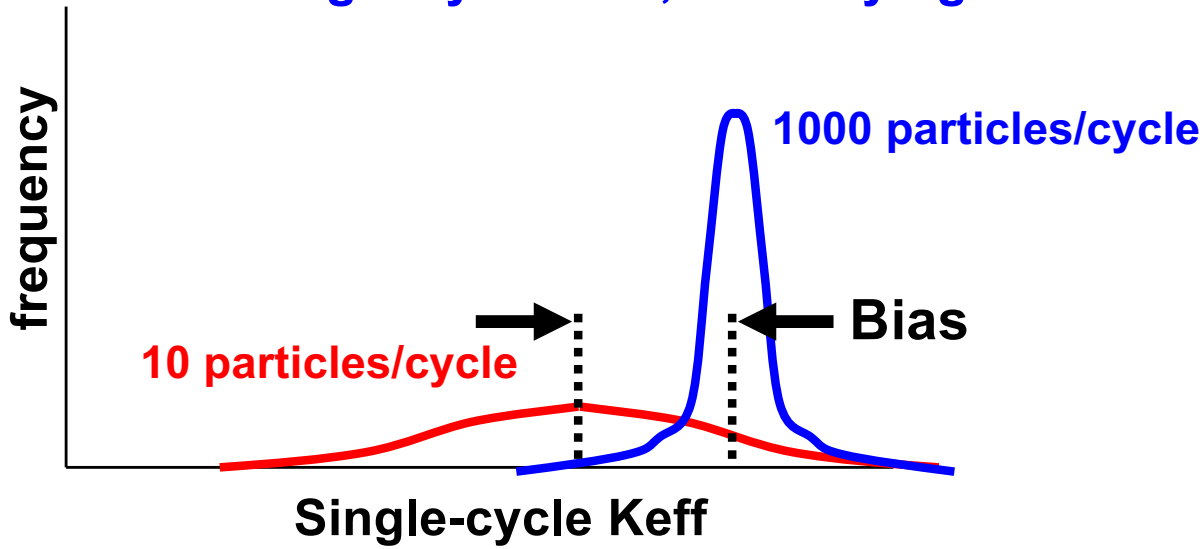
K-Calculations -- Bias

- **For a simple Godiva reactor calculation:**



K-Calculations -- Bias

- Observed PDF for single-cycle Keff, for varying M



- Bias in Keff is negative: $K_{calc} < K_{true}$

- Bias is

significant	for	M < 10	particles/cycle
small	for	M ~ 100	
negligible	for	M > 10000	
0	for	M $\rightarrow \infty$	

- Recommendation: Always use 10,000 or more particles/cycle

Cautions and Suggestions

- **You MUST make sure that enough initial cycles are discarded to ensure convergence of Keff & the fission source distribution**
- **Bias**
  - If you run with 10s or 100s of neutrons/cycle, there is a bias in Keff.
  - Bias ~ 1/N, so using more neutrons/cycle reduces the bias.
  - Use 10K or more neutrons/cycle for production runs
- **Correlation**
  - Tallies for one cycle are correlated to the previous (and next)
  - Cycle-to-cycle correlation may cause relative errors to be low.
  - Can check on this by examining "batching" results in MCNP output.
- **If anything looks suspicious,**
  - Run more cycles
  - Start over with more neutrons/cycle
  - Rerun the calculation with a new random number seed.

Cautions and Suggestions

- Use more neutrons/cycle, not fewer…, 10K or more per cycle
- For systems with high dominance ratio (ie, close to 1), may need to discard 100s or 1000s of initial cycles. For typical LWR's, probably need to discard ~50 initial cycles.
- Should run at least 100 cycles after convergence, to ensure that statistical analysis is reliable
- For reactors, initial source guess should cover most of core. Not necessary to worry about fuel vs non-fuel.
- For criticality safety, make sure initial source guess has some source points in each fissionable region.

Advanced Geometry

Introduction & Overview

Universe & Fill

'Like m But' & TRCL

Lattices & Fill

Hexagonal Geometry

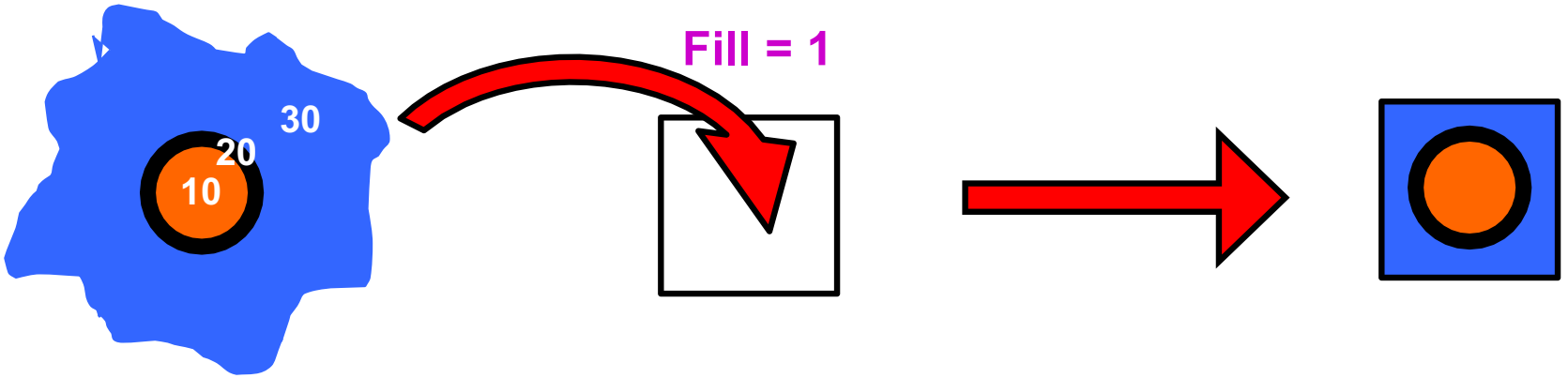
Introduction & Overview

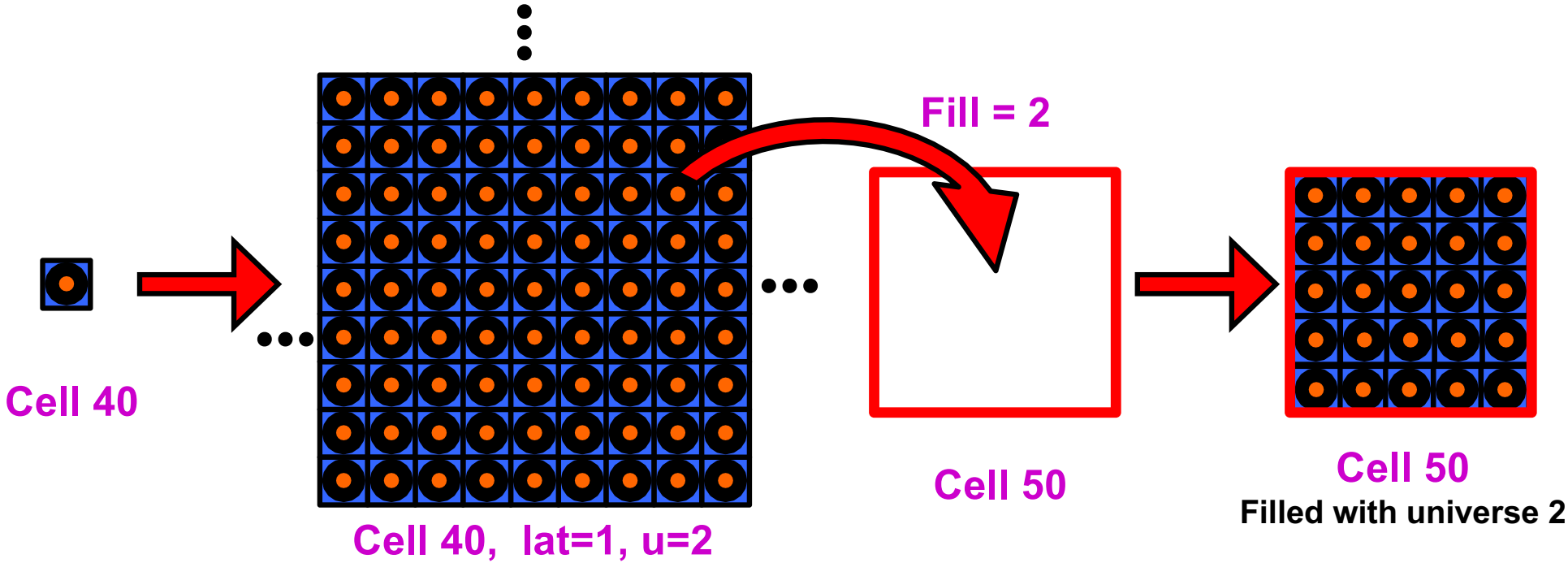
Universe, Fill, Lattices

Universe, Fill, & Lattice Concepts

- **Universe**
  - A collection of cells
  - Include a cell in universe n: put u=n on cell card, after surface list
    - n can be any number, need not be sequential
    - n must also appear on a fill= entry on another cell card (container)
    - All cells with the same u=n form a universe that fills another cell.
- **Fill**
  - Fill a container cell with a universe; insert universe into cell
  - Add fill=n on cell card, after surface list
    - n is the number of a universe
    - Usually, the cell being filled will contain a void material
    - Filled cell is a "window" - clips away any part of the filling universe which extends beyond the cell boundary
- **Lattice**
  - Define a cell, then flag it as the central element of an infinite lattice
  - MCNP replicates the central cell infinitely in x,y,z
  - Add lat=1 to cell definition, after surface list, for hexahedral lattice (6-sided)
    (use lat=2 for hexagonal prism lattices (8-sided))

Universe, Fill, Lattice, Fill

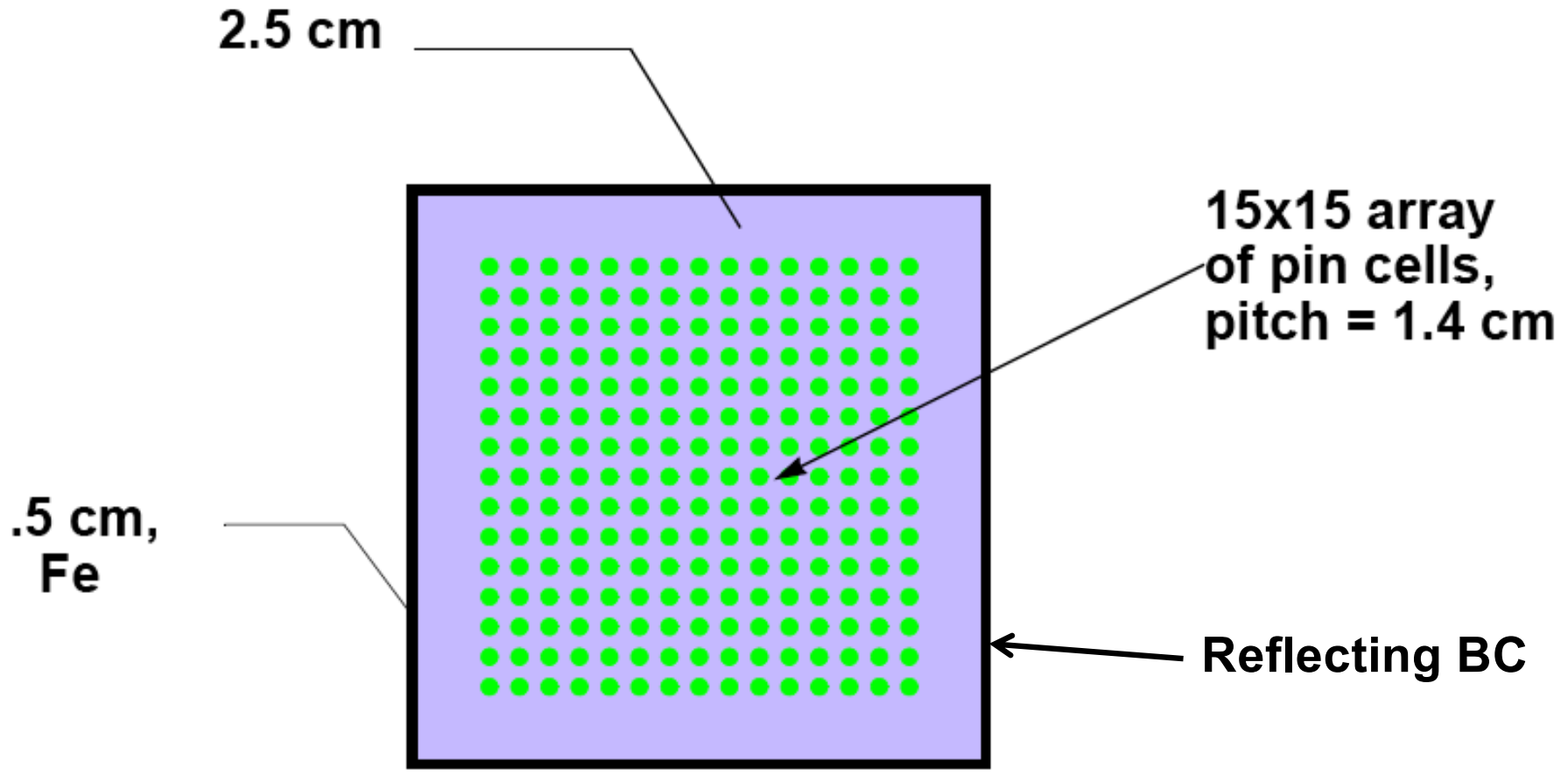




Problem – Fuel Assembly

- See Advanced Geometry lecture & criticality Case Study #2 for step-by-step explanations of this problem

- Fuel assembly - 15x15 Array of Pin Cells, infinite in z-direction



Problem assembly.txt

```
Problem assembly.txt  -  Fuel assembly - reflecting BC
 10    100  0.06925613  -1       u=1     imp:n=1                 $ fuel
 20    200  0.042910     1 -2    u=1     imp:n=1                 $ clad
 30    300  0.100059     2       u=1     imp:n=1                 $ water
 40    0                -3      fill=1   lat=1 u=2    imp:n=1
 50    0                -4      fill=2   imp:n=1                 $ lattice region
 60    300  0.100059     4 -5             imp:n=1                 $ outer water
 70    400   0.083770    5 -6             imp:n=1                 $ outer Fe

 1     cz   0.44
 2     cz   0.49
 3     RPP    -.7   .7    -.7   .7    0. 0.          $ small box for unit
 4     RPP  -10.5 10.5  -10.5 10.5    0. 0.          $ lattice region
 5     RPP  -13.0 13.0  -13.0 13.0    0. 0.          $ outer water
*6     RPP  -13.5 13.5  -13.5 13.5    0. 0.          $ outer Fe

kcode   1000    1.0  10  50
hsrc  15 -10.5 10.5   15 -10.5 10.5  1 -9e9 9e9
sdef    x=d1 y=d2 z=0.0
 si1     -10.5 10.5
 sp1        0    1
 si2     -10.5 10.5
 sp2        0    1
m100   92238 2.2380e-2   92235 8.2213e-4  8016 4.6054e-2  $ fuel
m200   40000 4.2910e-2                                   $ clad
m300    1001 6.6706e-2    8016 3.3353e-2                 $ water
mt300   lwtr
m400   26000 8.3770e-2                                   $ Fe
```

Result:

$K_{eff} = 1.13055 \pm 0.00268$

Universe & Fill

Universe Card (1)

- One or more cells can be grouped together into a collection, called a universe.

- A universe is either
  - a lattice cell OR
  - a collection of standard cells

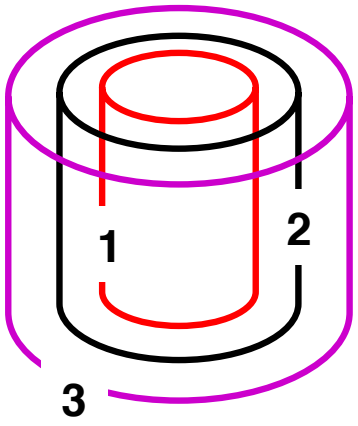
- Form: u=n

  - placed on the cell cards, after the surface info
  - n can be any number, u= numbers need not be sequential
  - n must also appear on a fill= entry on another cell card (container)
  - All cells with the same u=n form a universe that fills another cell.

- Cells of a universe can be finite or infinite, but must fill all of the space inside the container cell they fill.

Universe Card (2)

- Reactor fuel rod with gap & cladding, surrounded by infinite moderator




```
c  CELLS
10   110    0.069256    -1        u=9    $ fuel
20   0                   1 -2     u=9    $ gap
30   120    0.042910     2 -3     u=9    $ clad
40   130    0.100059     3        u=9    $ water, infinite

c SURFACES
1   RCC    0. 0. 0.      0. 0. 360.      0.43
2   RCC    0. 0. 0.      0. 0. 360.      0.44
3   RCC    0. 0. 0.      0. 0. 360.      0.49
```

- Universe 9 consists of cells 10, 20, 30, 40 - the fuel, gap, clad, & water
- Note that the Cell 40 (water) is infinite
- Universe 9 can be used to "fill" another cell (container cell), or to create a lattice of fuel rods

Fill Card

- **Fill a cell or lattice element with a universe**
- **Form: fill=n**
  - **placed on the cell cards, after the surface info**
  - **n is the number of a universe**
  - **Variations:**

fill=# (k)	where k is an optional transformation
fill=# (...)	where ... are optional TR entries
***fill=# (...)**	optional TR entries in **degrees** between this cell and filling universe

- **Usually, the cell being filled will contain a void material, since the material numbers and densities were assigned to the cells in the filling universe**
- **Filled cell is a "window" - clips away any part of the filling universe which extends beyond the cell boundary**

Example puc2 (1)

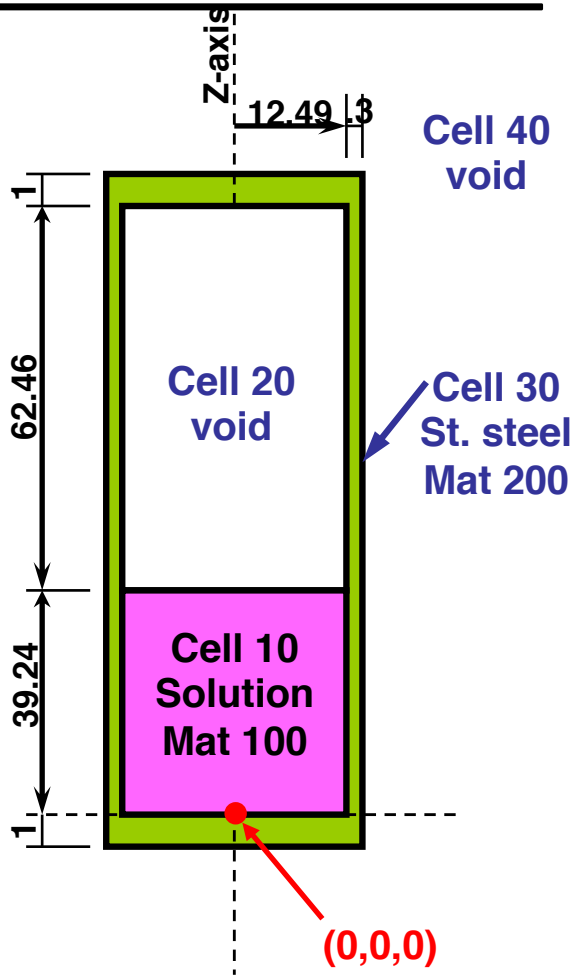
Problem puc2

Use UNIVERSE for solution & inner void
&
FILL the steel can with that universe

Example puc2 (2)

- **Copy file puc1.txt to puc2.txt**

- **Edit file puc2.txt**
  - **Modify definitions of cells 10 & 20:**
    - Identify cells 10 & 20 as **being in universe 1**
    - Remove surface 1 from their definitions
    - Both cells are infinite (in universe 1)
  - **Define a cell (25) for the interior of the container**
    - Bounded by the inner surface of the container (1)
    - **Fill it with universe 1**
    - Don't forget to add imp:n=1



- **Run the problem, with**

```
kcode  1000  1.0   25   100
```

- **Note that the answer is identical to the previous run**

Example puc2 (3)

```
puc2 - single cylinder, using universe & fill
c
C    ----- universe 1 -----
10   100   9.9270e-2     -3       u=1   imp:n=1    $ infinite solution
20   0                    3       u=1   imp:n=1    $ infinite void
C    ----- real world -----
25   0                   -1 fill=1   imp:n=1    $ inside container, filled
30   200   8.6360e-2      1 -2       imp:n=1    $ container
40   0                    2          imp:n=0    $ exterior

1      RCC    0. 0.  0.     0. 0. 101.7      12.49   $ inner
2      RCC    0. 0. -1.     0. 0. 103.7      12.79   $ outer
3      pz     39.24                                  $ solution height

C DATA CARDS from puc1_data_cards.txt
kcode    1000 1.0 25 100
ksrc     0. 0. 19.62
m100     . . .
mt100    . . .
m200     . . .
```

Result:

keff = 0.88778 ± 0.00363

Example puc2 (4)

- **Universe 1 is infinite**
  - Infinite void above surface 3, infinite solution below surface 3
  - Because cells 10 & 20 are infinite, MCNP can't compute their volume, & uses Volume=0 in the output
- **Cell 25 is filled with universe 1**
  - Universe 1 is clipped by the container cell (cell 25)
  - The container (cell 25) must be completely filled by the embedded universe (of course it is, since universe 1 is infinite…)
- **Plotting**
  - "XY" plot
  - See what happens when "Level" is changed -- Level 0, Level 1 (must click on "Redraw" to refresh the plot after changing Level)
- **Results**
  - Same as previous runs
  - Not always true when you use universe/fill - might have different roundoff …

'Like m But'
&
TRCL

'Like m But' Card

- "LIKE m BUT" cell description provides shorthand method for repeating similar cells

- Form: j LIKE m BUT list
  - Cell j takes all attributes of cell m except parameters in 'list'
  - Cell m must be defined before "j like m but" in INP file

- Parameters that can make up 'list' include:
  - imp, vol, pwt, ext, fcl, wwn, dxc, nonu, pd, tmp
  - u, trcl, lat, fill
  - mat, rho
  - U and/or TRCL, at minimum, must be in 'list'
  - Examples:

```
17   like 70 but     trcl=( 1 1 2)   u=66
23   like 70 but     mat=13   u=2
```

- Surface numbers cannot be altered with "like m but" format

Translation & Rotation

- Surfaces can be translated/rotated using the TR card
- Cells can be translated/rotated using the TRCL card
- Forms:
  - translate CELL by (dx,dy,dz):

    TRCL=(dx dy dz)

  - Translate & rotate CELL:

    TRCL=(dx dy dz xx' yx' zx' xy' yy' zy' xz' yz' zz')

    where

    xx' = cosine of angle between original x-axis and new x'-axis

    xy' = … similar …

  - Translate & rotate CELL:

    *TRCL=(dx dy dz xx' yx' zx' xy' yy' zy' xz' yz' zz')

    where

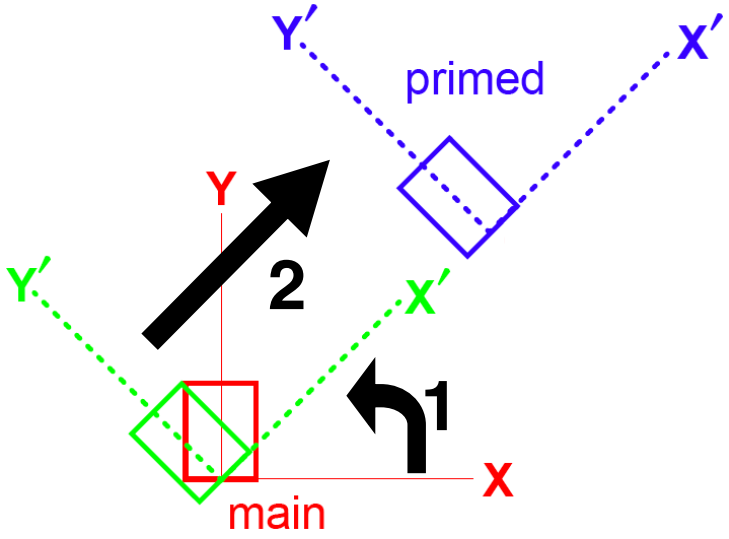
    xx' = angle in degrees between original x-axis and new x'-axis

    xy' = … similar …

- Rotation is done first, then translation

Translation & Rotation

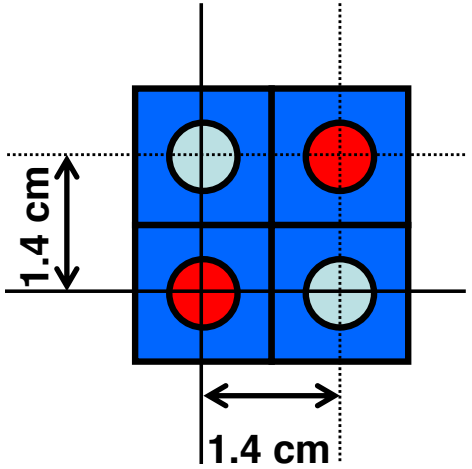
- **Rotation (red to green axes) is done first** (in original coord system)
- **Translation (green to blue axes) is done second** (in original coord system)



- **When TRCL is used, MCNP must create new surfaces**
  - **The new surfaces are assigned numbers of the form:**
    1000 * (new-cell-number) + (original-surface-number)
  - **Be careful to avoid those surface numbers in the rest of your input**
  - **If you use TRCL, make sure your surface numbers are <1000 !**
- **All universes that fill this cell inherit the TRCL**

'Like m But' & TRCL - Example

- **Cluster of several fuel rods, with different enrichments**




```
c Cell Cards
c ----- red universe, 7 -----
1 110   .069   -11   u=7      $ fuel-red
2 120   .100    11   u=7      $ water
c
c ----- green universe, 8 -----
3 130   .069   -11   u=8      $ fuel-green
4 120   .100    11   u=8      $ water
c
c ----- real world -----
5 0       -12   fill=7       $ unit cell,  lower left, at origin
6 like 5 but   fill=8 trcl=(  0 1.4 0) $ upper left
7 like 5 but   fill=8 trcl=(1.4   0 0) $ lower right
8 like 5 but   fill=7 trcl=(1.4 1.4 0) $ upper right

c Surfaces
11   RCC    0. 0. -180.      0. 0. 360.      0.49
12   RPP     -.7 .7    -.7 .7    -180. 180.
```

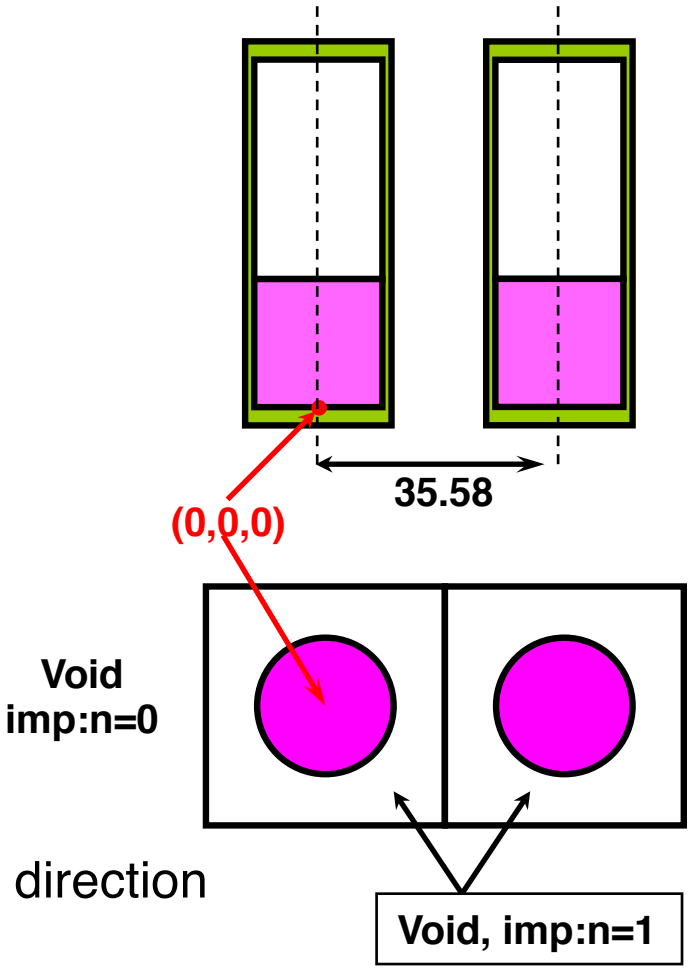
Example puc3 (1)

Problem puc3

Two Cans of Solution Using 'Like m But' and TRCL

Example puc3 (2)

- **Two cans, 35.58 cm separation between centers**
- **Copy file puc2.txt to puc3.txt**
- **Edit file puc3.txt**
  - **Identify cells 25, 30, 40 as being in universe 2 (change the importance of cell 40 to imp:n=1)**
  - **Define cell 50 & surface 4**
    - A box around the first can (cell 30)
    - Use RPP, x,y in range (-17.79,17.79), z in range (-1,102.7)
    - FILL cell 50 with universe 2
    - Importance = 1
  - **Define cell 60**
    - Same as cell 50, but translated 35.58 cm in +x direction
  - **Define cell 99**
    - Void, importance = 0, outside of 50 & 60
  - **Add another source point to KSRC, at (35.58, 0., 19.62)**



Example puc3 (3)

```
puc3 - TWO cylinders
C ----- universe 1, infinite solution & void -----
10  100  9.9270e-2     -3          u=1  imp:n=1   $ infinite solution
20  0                   3          u=1  imp:n=1   $ infinite void
C ----- universe 2, filled can & infinite exterior -----
25  0                  -1 fill=1   u=2  imp:n=1   $ inside of can, filled
30  200     8.6360e-2   1 -2       u=2  imp:n=1   $ can
40  0                   2          u=2  imp:n=1   $ infinite exterior
C ----- real world, 2 boxes (contining cans) & infinite exterior -----
50  0                  -4 fill=2        imp:n=1   $ 1st box at origin, with can
60  like 50 but  trcl=(35.58  0.  0.)             $ 2nd box shifted, with can
99  0                   #50 #60         imp:n=0   $ exterior to both boxes

1       RCC   0. 0.  0.     0. 0. 101.7     12.49
2       RCC   0. 0. -1.     0. 0. 103.7     12.79
3       pz    39.24
4       RPP  -17.79 17.79   -17.79 17.79   -1. 102.7

C DATA CARDS from puc1_data_cards.txt
kcode   1000 1.0 25 100
ksrc    0. 0. 19.62        35.58 0. 19.62
m100    . . .
mt100   . . .
m200    . . .
```

Result:
keff = 0.91260 ± 0.00381

Example puc3 (4)

- **Universe 1 is infinite**
  - Same as before, but now appears in 2 different places
  - Clipped by surface 1 when if FILLs cell 25
- **Universe 2 is infinite**
  - Can (containing universe 1) & exterior void
  - Embedded in Cell 50, and also in cell 60
- **Plotting**
  - "XY" plot, "ZX" plot
  - See what happens when "Level" is changed -- Level 0, Level 1, Level 2 (must click on "Redraw" to refresh the plot after changing Level)
  - Note surface 60004 -- what happened to other translated surfaces?
    (See output file for info on identical surfaces…)
  - Click on "MBODY On" - note the surface "facets" (internal label for body surfaces)
- **Results**
  - Higher Keff, as expected

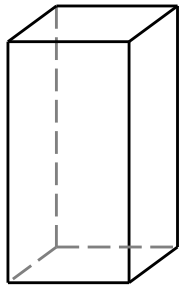
Lattices & Fill

Lattice Card (1)

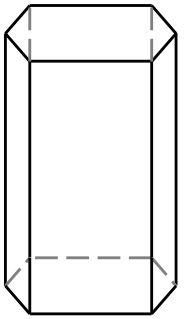
- **Defines cell as infinite array or lattice**
  - User input describes central element [0,0,0] in lattice
  - MCNP replicates the central element in all 3 directions
- **Form:** LAT=1 hexahedra (six face solid) square
  LAT=2 hexagonal prism (eight) triangle

  LAT=n should go on a cell card, after surface info
- **Space between elements must be filled exactly:**
  - hexahedra or hexagonal prisms need not be regular
  - Opposite sides of central element must be parallel
- **Lattice elements can be infinite along 1 or 2 axes**
- **Order of surfaces on the cell card is important**
  - Best to use macrobody -- will always increment along +axis

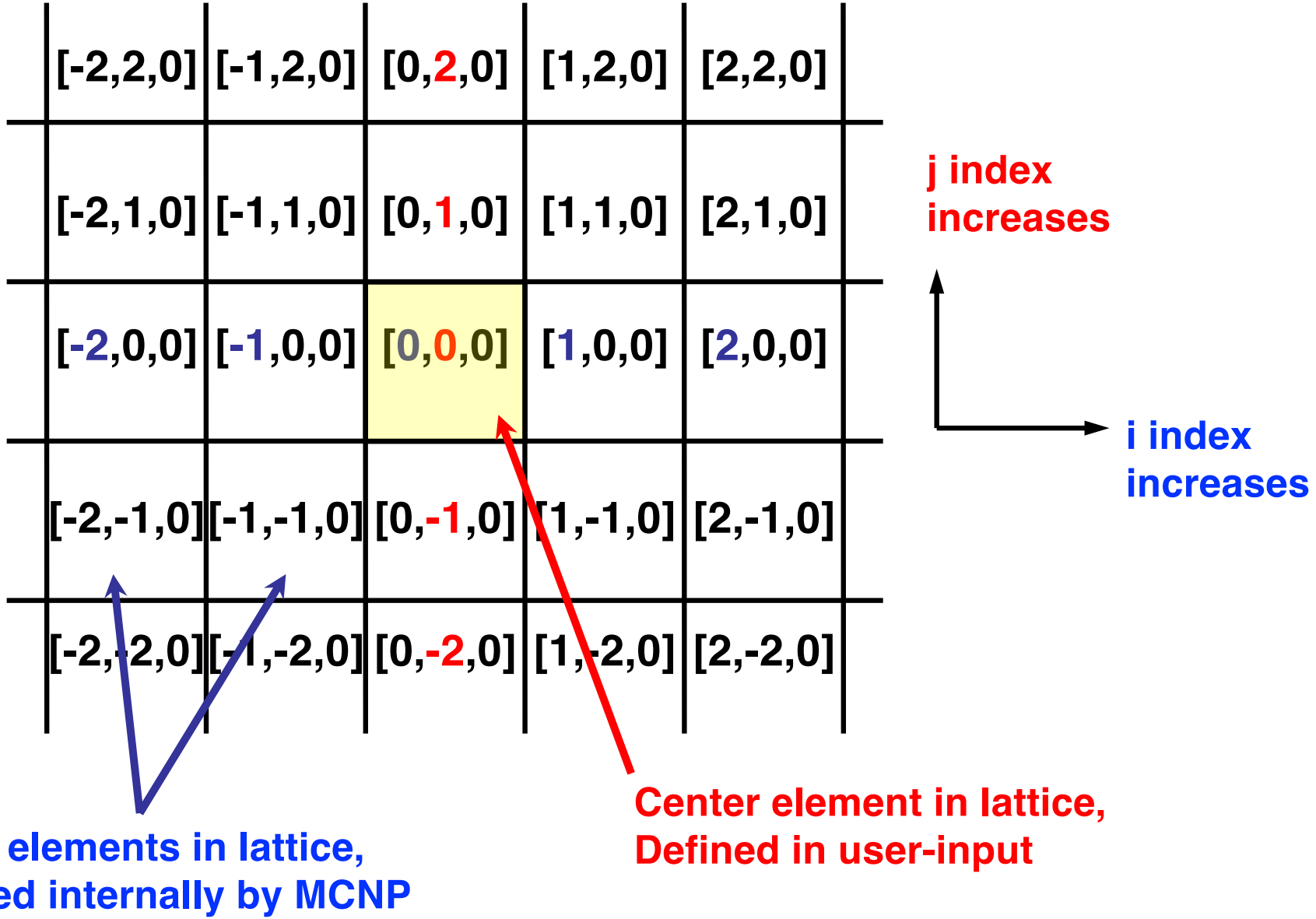
Hexahedra
(Quadrilateral Prism)
LAT=1



Hexagonal
Prism
LAT=2



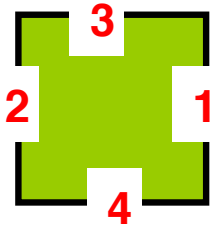
Lattice Card (2)



Lattice Card (3)

- **Elements identified by [i,j,k] labels determined by the order of surface entries on cell card**

- **Cell card specifies the [0,0,0] element**



5 - top
6 - bottom

```
11   0   -1 2    -3 4    -5 6   lat=1
```

  - For LAT=1, at least 4 surfaces or 2 vectors required

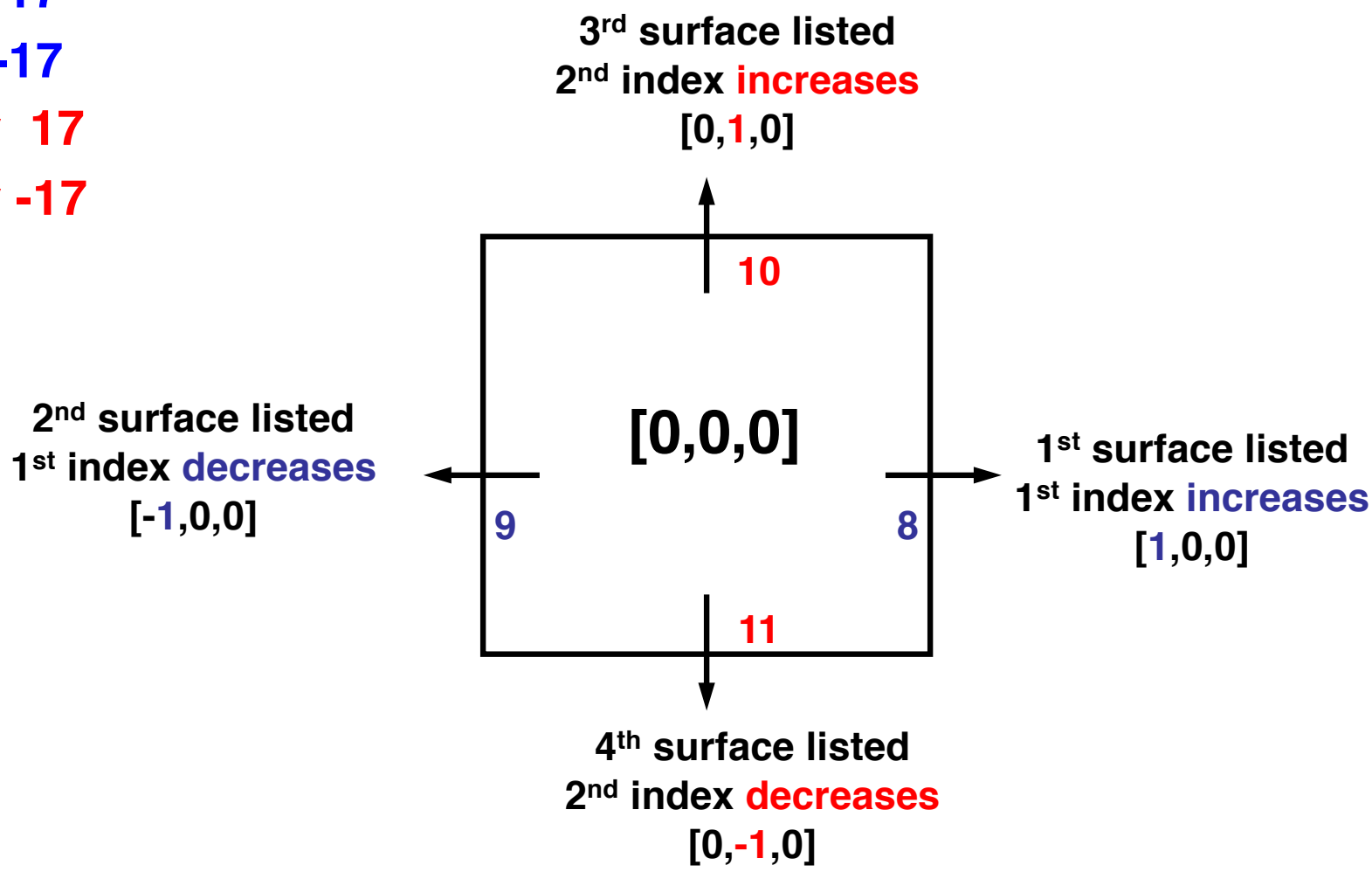
  - On + side of 1st surface = [1, 0, 0] lattice element
  - - side of 2nd [-1, 0, 0]
  - + side of 3rd [0, 1, 0]
  - - side of 4th [0,-1, 0]
  - + side of 5th [0, 0, 1]
  - - side of 6th [0, 0,-1]

- **If you don't list the surfaces in the order shown above, everything will get very confusing & you will have trouble.**

Lattice Card (4)

5 0 -8 9 -10 11 lat=1

8 px 17
9 px -17
10 py 17
11 py -17



Lattice Card (5)

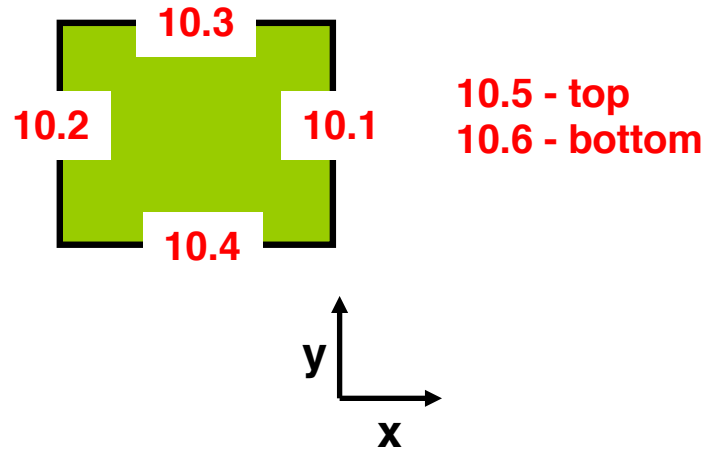
- **For macrobodies, MCNP internally replaces the body with a set of surfaces**
  - The surfaces created have the form S.F
  - "S" is the original surface number for the macrobody
  - "F" is a facet number, 1, 2, …

  - These cell & surface cards in MCNP input

```
25   111    -1.0     -10   lat=1    $ cell card

10   RPP   -1 1   -2 2    -3 3      $ surface card (body)
```

generate these surfaces internally

10.1	- "px" plane at x= 1
10.2	- "px" plane at x=-1
10.3	- "py" plane at y= 2
10.4	- "py" plane at y=-2
10.5	- "pz" plane at z=3
10.6	- "pz" plane at z=-3

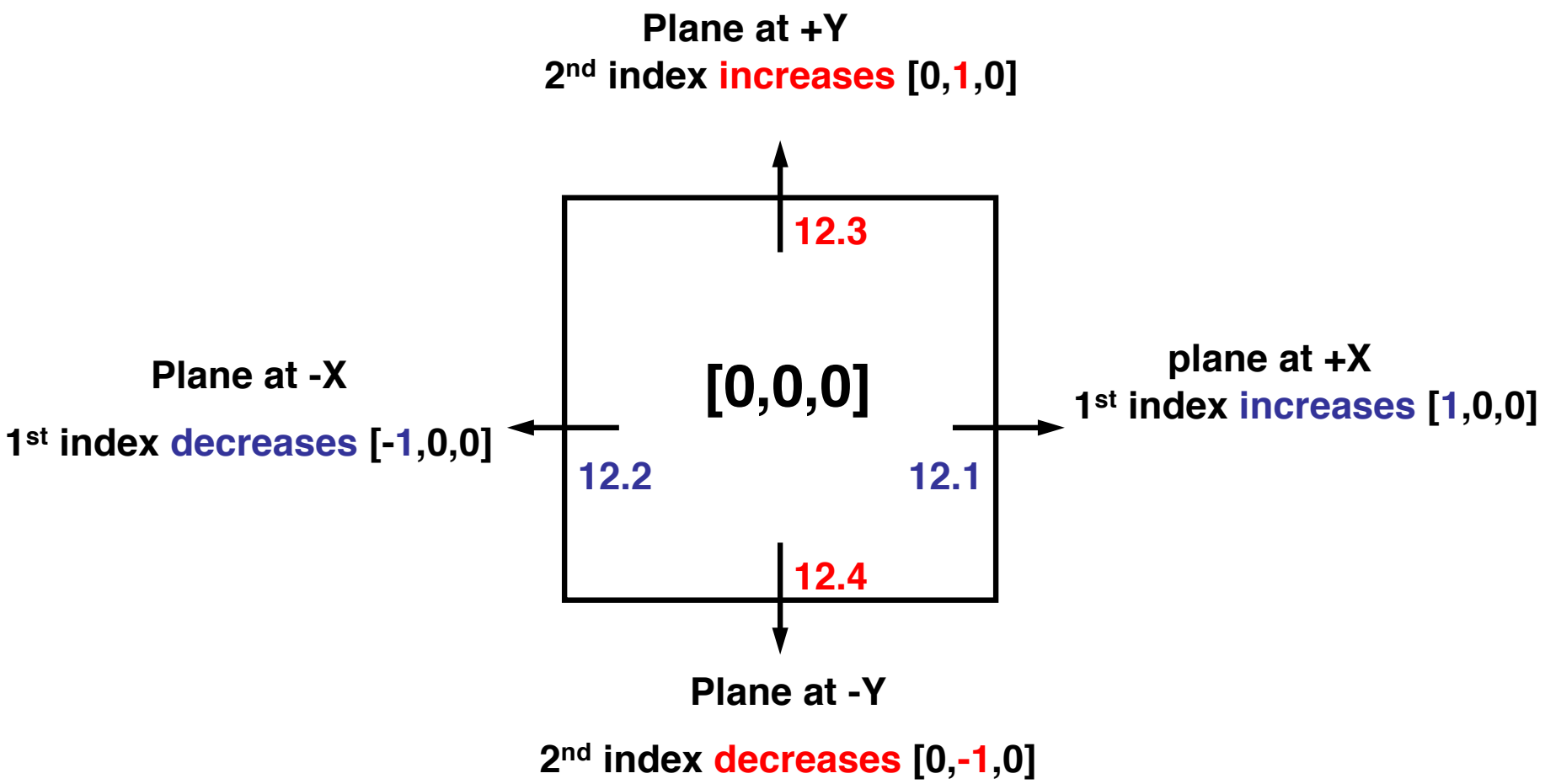


Lattice Card (6)

5 0 -12 lat=1

For infinite in z-dir, use 0 0

12 RPP -17. 17. -17. 17. -180. 180.

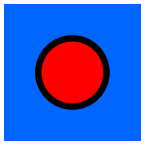


Lattice Card (7)

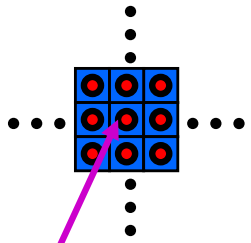
9 x 9 array of fuel rods

```
c Cells
1 110   .069   -10   u=7      $ fuel
2 120   .100    10   u=7      $ infinite water
c
5 0      -20   fill=7  lat=1  u=9  $ infinite lattice of pins
6 0      -30   fill=9           $ box with 9x9 pins

c Surfaces
10 RCC  0. 0. 0.   0. 0. 360.  0.49      $ cylinder for fuel
20 RPP  -.7 .7   -.7 .7   0 360          $ box for single pin
30 RPP  -6.3 6.3  -6.3 6.3  0 360        $ box holds 9x9 pins
```

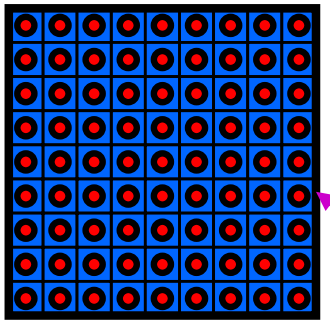
Universe 7



Universe 9



Real world, Cell 6



User defines center cell,
MCNP replicates into infinite lattice

Outer box (surface 30)
truncates infinite lattice

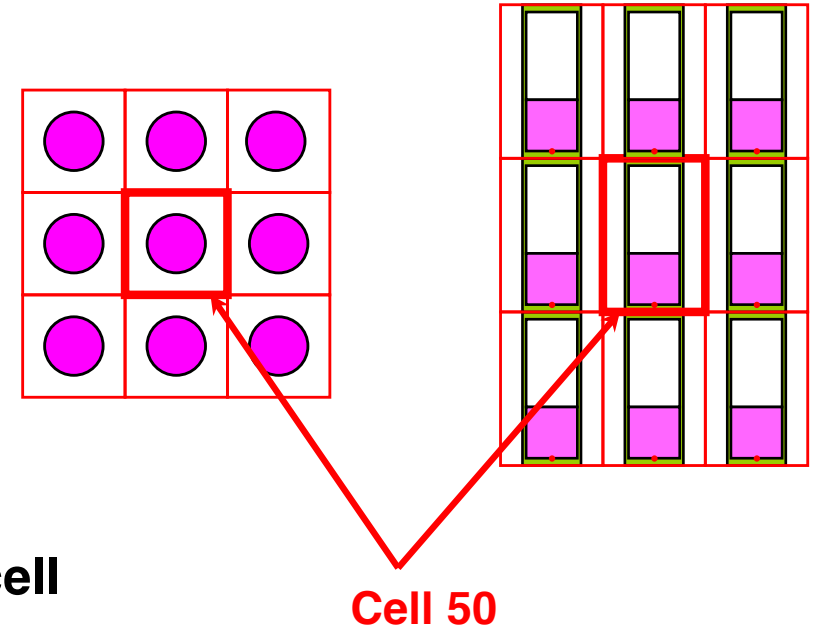
Example puc4 (1)

Problem puc4

Infinite Lattice of Cans
(3D Lattice)

Example puc4 (2)

Define the center lattice cell, filled with universe 2



- **Copy file puc3.txt to puc4.txt**
- **Edit file puc4.txt**
  - **Delete cells 60 & 99**
  - **Declare Cell 50 to be the center lattice cell in a hexahedral (box) lattice, LAT=1**
  - **Add this data card (turns off entropy calc for infinite lattice):**

```
hsrc  1 -1.e10 1.e10  1 -1.e10 1.e10  1 -1.e10 1.e10
```

  - **Remove the second point in KSRC**
  - **Plot:** `mcnp5 i=puc4 ip`
  - **Compute keff:** `mcnp5 i=puc4`

Example puc4 (3)

```
puc4 - infinite 3D lattice of cans
10   100   9.9270e-2   -3          u=1     imp:n=1   $ infinite solution
20   0                  3          u=1     imp:n=1   $ infinite void
25   0                 -1 fill=1   u=2     imp:n=1   $ inside can, filled
30   200   8.6360e-2    1 -2       u=2     imp:n=1   $ can
40   0                  2          u=2     imp:n=1   $ infinite exterior
50   0                 -4 fill=2   lat=1   imp:n=1   $ center lattice cell

1    RCC    0. 0.  0.     0. 0. 101.7      12.49
2    RCC    0. 0. -1.     0. 0. 103.7      12.79
3    pz     39.24
4    RPP   -17.79 17.79   -17.79 17.79    -1. 102.7   $ box to hold can

C DATA CARDS from puc1_data_cards.txt
kcode    1000 1.0 25 100
ksrc     0. 0. 19.62
m100     . . .
mt100    . . .
m200     . . .
hsrc     1 -1.e10 1.e10   1 -1.e10 1.e10   1 -1.e10 1.e10
```

Result:
keff = 1.61058 ± 0.00194

Example puc4 (4)

- **Keff is pretty large**
  - Infinite lattice, no leakage, no absorbers, …
- **Note that lattices are:**
  - Defined by creating the center cell & flagging it with LAT=1
  - Filled with 1 or more universes
  - Infinite in extent
- **How do you get a finite lattice?**

1. Make an infinite lattice, then give it a universe number
2. Create a container cell to hold some portion of the lattice
3. Then fill that container cell with the lattice universe
   - Infinite lattice is clipped (truncated) by the container cell boundaries
   - Lattice elements outside the container can never be reached

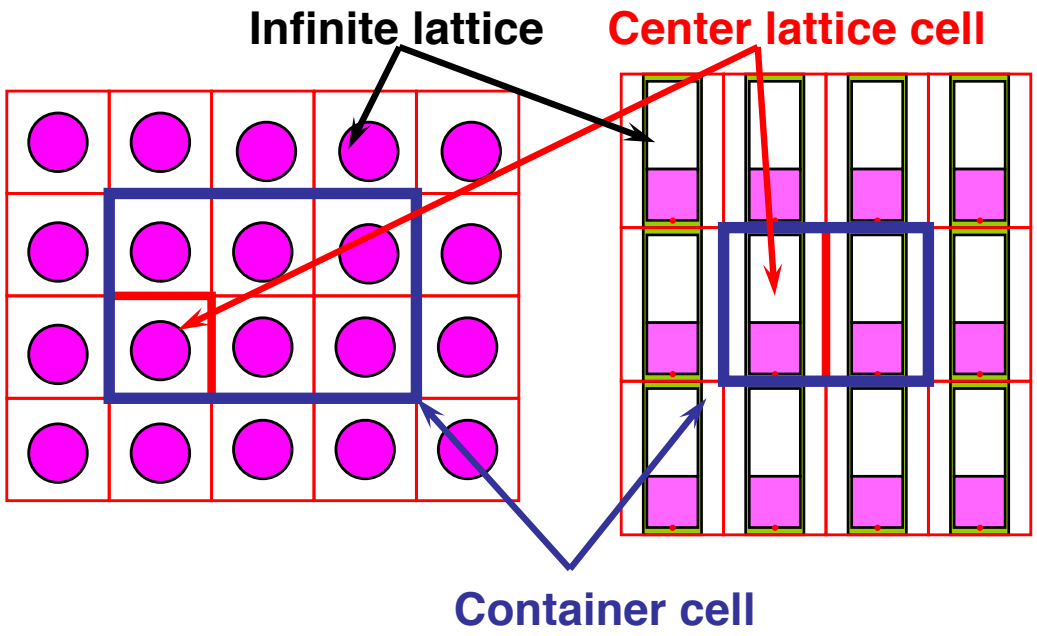
Example puc6 (1)

Problem puc6

Finite 2x3 Lattice of Cans

Example puc6 (2)

Start with infinite 3D lattice (Problem puc4) & create a 3x2 array of cans



- **Copy file puc4.txt to puc6.txt**
- **Edit file puc6.txt**
  - **Declare cell 50 as universe 3**
  - **Define the container cell 60, and an RPP body sized to hold 2x3 array of cell 50**
  - **Fill cell 60 with universe 3**
  - **Define cell 99, outside container, imp:n=0**
  - **Change KCODE card, use** `kcode 10000 1.0 25 100`
  - **Modify the HSRC card (optional):**

```
hsrc  3 -17.79 88.95   2 -17.79 53.37   1 -1. 102.7
```

  - **Plot:** mcnp6 i=puc6.txt ip
  - **Compute keff:** mcnp6 i=puc6.txt

Example puc6 (3)

```
puc6 3 x 2 array of cans
10   100   9.9270e-2   -3          u=1          imp:n=1    $ infinite solution
20   0                  3          u=1          imp:n=1    $ infinite void
25   0                 -1 fill=1   u=2          imp:n=1    $ contents of can, filled
30   200   8.6360e-2    1 -2       u=2          imp:n=1    $ can
40   0                  2          u=2          imp:n=1    $ outside of can, infinite
50   0                 -4 fill=2   lat=1 u=3    imp:n=1    $ infinite 3D lattice
60   0                 -5 fill=3                imp:n=1    $ container, fill by lattice
99   0                  5                       imp:n=0    $ outside container

1         RCC    0. 0.  0.     0. 0. 101.7      12.49
2         RCC    0. 0. -1.     0. 0. 103.7      12.79
3         pz     39.24
4         RPP   -17.79 17.79   -17.79 17.79    -1. 102.7
5         RPP   -17.79 88.95   -17.79 53.37    -1. 102.7  $ container, holds 3x2

C DATA CARDS from puc1_data_cards.txt
kcode    20000 1.0 25 100
ksrc     0. 0. 19.62
m100     . . .
mt100    . . .
m200     . . .
hsrc     1 -1.e10 1.e10   1 -1.e10 1.e10   1 -1.e10 1.e10
```

Result:
keff = 0.98736 ± 0.00094

Example puc6 (4)

- **Keff is close to 1.0 ….**

- **Note that lattices are:**
  - **Defined by creating the center cell & flagging it with LAT=1**
  - **Filled with 1 or more universes**
  - **Infinite in extent**

- **To get a finite lattice:**

1. **Make an infinite lattice, then give it a universe number**

2. **Create a container cell to hold some portion of the lattice**

3. **Fill the container cell with the lattice universe**
   - **Infinite lattice is clipped (truncated) by the container cell boundaries**
   - **Lattice elements outside the container can never be reached**

Fully Specified Fill for Lattices (1)

- When 'filling' a cell which is a lattice, you can specify which universe goes into each individual lattice element

- Infinite Lattice Form: fill = n
  - Fill all lattice elements with universe n

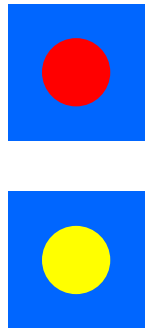
- Finite Lattice Form: fill = i1:i2 j1:j2 k1:k2 N_1 (...) N_2 (...) etc
  - i1:i2 j1:j2 k1:k2
    defines which elements of the lattice exist (i1≤ i2, j1≤j2, k1≤k2)
  - N_1, N_2, etc
    list of filling universe numbers that specify what universe fills each lattice element
  - Order of array entries follows FORTRAN convention:
    (i1,j1,k1), (i1+1,j1,k1), (all i,j1,k1), ... (all i,j2,k1)..... (all i,j3,k1) .....
  - For this fill card: `fill= -2:2 0:1 0:0`

    5 x 2 x 1 entries are required, as in

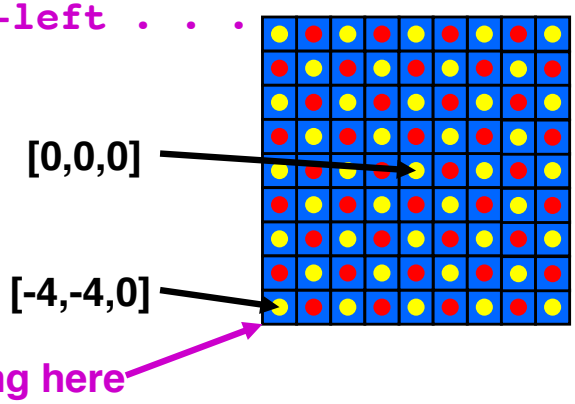
```
fill= -2:2 0:1 0:0   1   2   3   2   1
                     2   1   1   1   2
```

Fully Specified Fill for Lattices (2)

Finite lattice, 9x9 checkerboard arrangement



```
c Cell Cards
1 110   .069   -10   u=7       $ fuel-red
2 120   .100    10   u=7       $ infinite water
c
3 130   .069   -10   u=8       $ fuel-yellow
4 120   .100    10   u=8       $ infinite water
c
5 0           -20   u=9 lat=1          $ lattice of pin-cells
          fill= -4:4   -4:4   0:0      $ only fill central 9x9 elements,
         8 7 8 7 8 7 8 7 8             $   start at bottom-left . . .
         7 8 7 8 7 8 7 8 7
         8 7 8 7 8 7 8 7 8
         7 8 7 8 7 8 7 8 7
         8 7 8 7 8 7 8 7 8
         7 8 7 8 7 8 7 8 7
         8 7 8 7 8 7 8 7 8
         7 8 7 8 7 8 7 8 7
         8 7 8 7 8 7 8 7 8
6 0      -30   fill=9                 $ box to hold 9x9 pins, clip the rest

c Surfaces
10   RCC   0. 0. 0.      0. 0. 360.      0.49
20   RPP   -.7 .7    -.7 .7    0 360
30   RPP   -6.3 6.3   -6.3 6.3   0 360   $ box holds 9x9 pins
```



Fully Specified Fill for Lattices (3)

- **Special value for the fill array:**
  Own universe number means element is filled with material on cell card

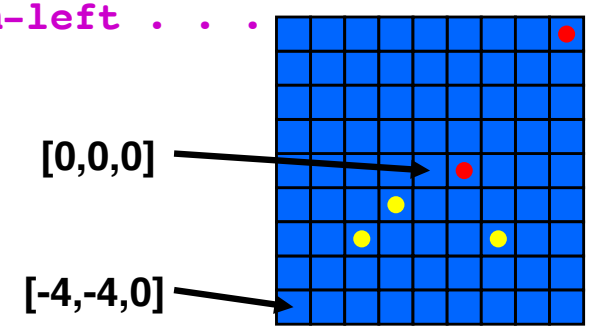
```
11   300   -1.0 -22   lat=1   u=9   fill= -1:1 -1:1 0:0
                                          9   9   9
                                          9   1   9
                                          9   9   9
```

  - The lattice elements in Cell 11 that are filled with Universe 9 (the universe assigned to this cell) are actually filled with Material 300
  - Note that when this special case is used, there can be no geometric detail inside of the lattice element

Fully Specified Fill for Lattices (4)

Finite lattice, 9x9 arrangement with only a few fuel pins

```
c Cell Cards
1 110   0.069   -10   u=7       $ fuel-red
2 120   0.100    10   u=7       $ infinite water
c
3 130   0.069   -10   u=8       $ fuel-yellow
4 120   0.100    10   u=8       $ infinite water
c
5 120   -1.0   -20   u=9 lat=1      $ lattice of pin-cells
         fill= -4:4   -4:4   0:0    $ only fill central 9x9 elements,
        9 9 9 9 9 9 9 9 9           $   start at bottom-left . . .
        9 9 9 9 9 9 9 9 9
        9 9 8 9 9 9 8 9 9
        9 9 9 8 9 9 9 9 9
        9 9 9 9 9 7 9 9 9
        9 9 9 9 9 9 9 9 9
        9 9 9 9 9 9 9 9 9
        9 9 9 9 9 9 9 9 9
        9 9 9 9 9 9 9 9 7
6 0      -30   fill=9      $ box to hold 9x9 pins, clip the rest

c Surfaces
10   RCC   0.0 0.0 0.0      0.0 0.0 360.0      0.49
20   RPP   -0.7 0.7   -0.7 0.7   0 360
30   RPP   -6.3 6.3   -6.3 6.3   0 360   $ box holds 9x9 pins
```



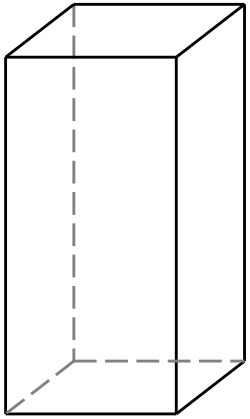
Geometry - General Suggestions

- **Don't set up geometry all at once**
  - start with small pieces, plot each as you go along
- **Always plot geometry !!!!!**
  - To see if it's correctly defined
  - To see if it's what you intended to define
- **Keep cells reasonably simple**
- **Use parentheses freely for clarity**
- **Only as much geometry detail as required for accuracy**
- **Check MCNP-calculated mass and volume against hand-calculated values**
- **2-D slices thru more than one plane**
  - Move plot plane origin around
  - Don't put plot plane directly on a surface
- **If all else fails…**
  - Use VOID card with inward-directed source
  - Lost particle: set plot origin to xyz location of lost particle,
    use uvw for plot basis vector, zoom-in with plotter

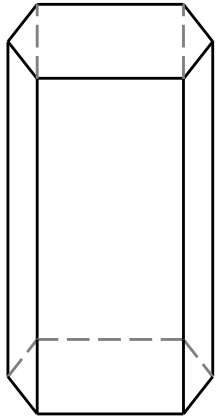
Hexagonal Geometry

Hexagonal Geometry (1)

Hexahedra (Quadrilateral Prism)



Hexagonal Prism



- Opposite sides must be identical and parallel.
- Hexagonal prism cross section must be convex
- Height of hexagonal prism can be infinite (if defined with surfaces)

Hexagonal Geometry (2)

- **Right Hexagonal Prism - RHP**
- **RHP and HEX are the same card**
- **RHP or HEX Card:**

RHP *v1 v2 v3* *h1 h2 h3* *r1 r2 r3* *s1 s2 s3* *t1 t2 t3*

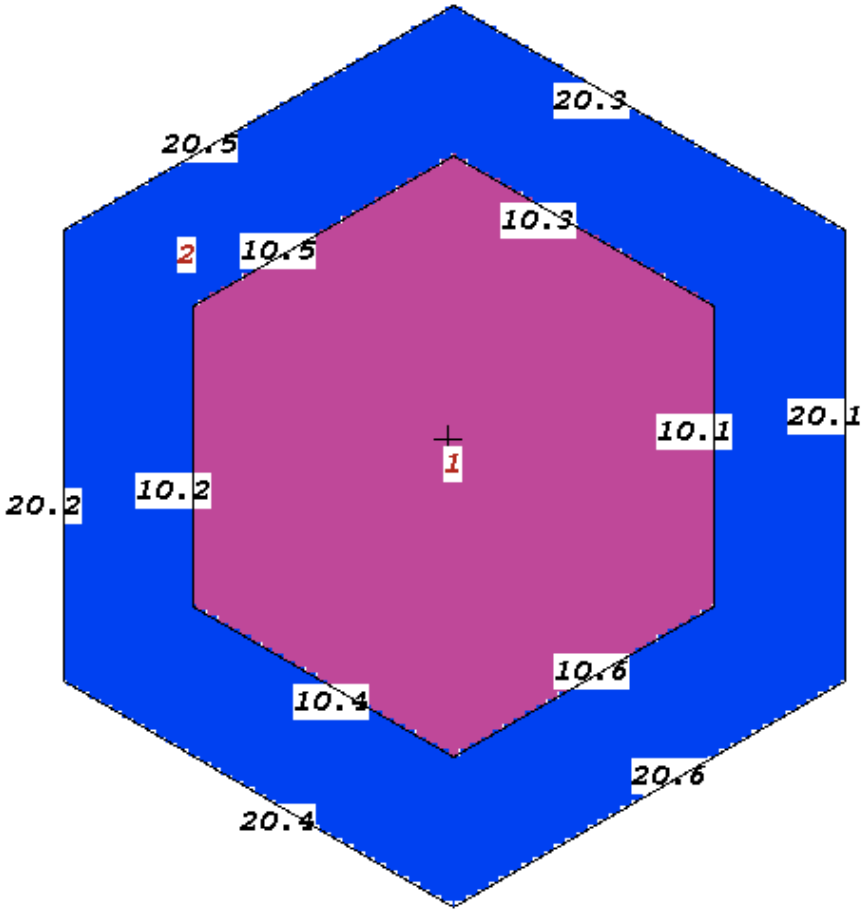
v1 v2 v3 = x, y , z coordinates of the bottom center of hex

h1 h2 h3 = vector from bottom to top, magnitude = height
for a z-hex with height h, h1 h2 h3 = 0 0 h

r1 r2 r3 = vector from the axis to the middle of the 1st facet,
for a pitch 2p facet normal to y-axis, r1 r2 r3 = 0 p 0

s1 s2 s3 = vector to center of the 2nd facet [optional]
t1 t2 t3 = vector to center of the 3rd facet [optional]

Hexagonal Geometry (3)



- **Example, hex using macrobodies:**

```
1  101  -1.0    -10
2  102  -7.8      10   -20

10  rhp  0  0  -4    0  0  8    2.  0  0
20  rhp  0  0  -4    0  0  8    3.  0  0
```

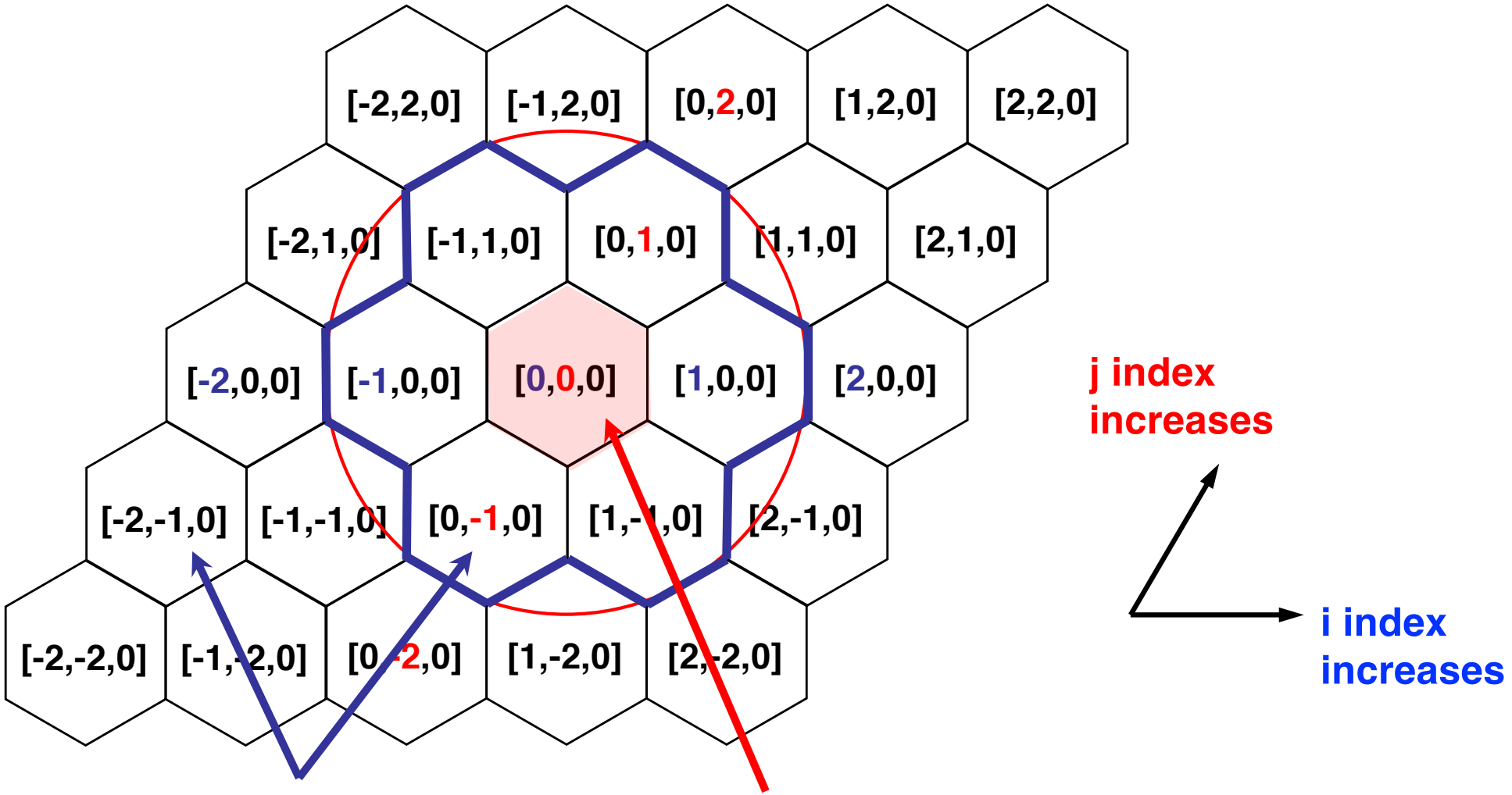
Center of base = (0,0,-4)

Height = 8, "out of the paper"

Surface 10.1 is x=2, 10.2 is x=-2

Surface 20.1 is x=3, 20.2 is x=-3

Hexagonal Geometry (4)



Tallies

Tally Fundamentals
Statistics
Reaction Rates
Flux & K-eff Estimators
Spectra & Plotting
Mesh Tallies

Optional
Dose, Tallies in Repeated Structures

Tally Fundamentals

Getting Results from MCNP

- **MCNP produces k-eff and various information in tables.**
  - **Only limited information about fluxes, spectra, reaction rates, etc.**
- **In fixed-source problems, MCNP gives no physical results by default.**
  - **Analogous to running an experiment without any detectors or measuring equipment!**
- **Tallies are analogous to measurement devices in experiments.**
- **MCNP has several "devices" available:**

Tally Types

- **Tallies in MCNP are often called edits in many other codes**
  - **Fluxes**
  - **Currents**
  - **Reaction rates**

- **MCNP tally types:**
  - **F1: Current on a surface**
  - **F2: Flux on a surface**
  - **F4: Flux in a cell (track-length estimate)**
  - **F5: Flux at a point or ring detector**
  - **F6: Energy deposition (track-length estimate)**
  - **F7: Fission energy deposition (track-length estimate)**
  - **F8: Pulse height tally**
  - **FMESH: Mesh tallies**

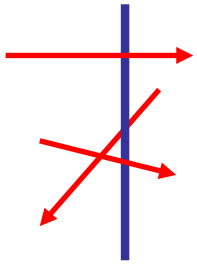
- **An energy weight can be applied to any tally by preceding it with an asterisk (e.g., `*F4`)**

- **A preceding plus-sign can be applied to F6 for collision heating (`+F6`) and F8 for charge deposition (`+F8`)**

Basic Tallies

F1 - Current across surface

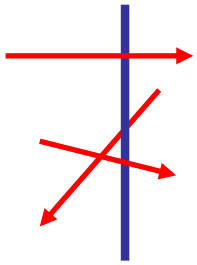
$$J = \frac{1}{W} \sum_{\text{all flights crossing surface}} wgt$$



W = total source weight

F2 - Flux on surface

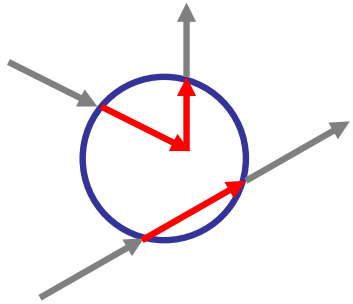
$$\phi = \frac{1}{A \cdot W} \sum_{\text{all flights crossing surface}} \frac{\text{wgt}}{|\mu|}$$



A = surface area
W = total source weight
μ = Ω • [surface normal]

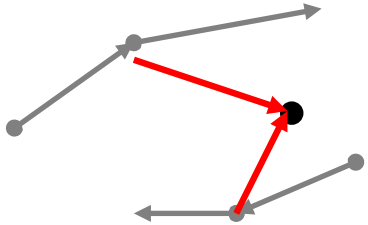
F4 - Flux in a cell

$$\phi = \frac{1}{V \cdot W} \sum_{\text{all flights in cell}} \text{wgt} \bullet \text{dist}$$



V = cell volume
W = total source weight

F5 - Flux at a point



$$\phi = \frac{1}{W} \sum_{\text{all collisions}} \text{wgt} \cdot \frac{p(\mu) e^{-\Sigma_T R}}{2\pi R^2}$$

Tally Normalization & Units

- **All MCNP tallies are normalized to be the response for 1 source particle**

- **If your actual source strength is 4000 particles/sec, then**
  - **MCNP tally results should be multiplied by 4000**
  - **Units for tallies should be "per second"**
  - **Applies when the rate of source particle production is known**

- **If your actual source strength is 4000 particles, then**
  - **MCNP tally results should be multiplied by 4000**
  - **Units for tallies should not be "per second"**
  - **Applies the total source particle production is known (eg, a pulse)**

- **You can have MCNP do the multiplication:**
  - **Supply the source strength on a tally multiplier card (FMn card)**

  OR
  - **Supply the source strength on the sdef card (include wgt=....)**

Tally Quantities Scored

Type	Where	Units
F1: Surface Current All particles	surface	# / sec
F2: Surface Flux All particles	surface	# / cm^2·sec
F4: Track length estimate of cell flux All particles	cell	# / cm^2·sec
F5: Flux at a point or ring detector N or P	point or ring	# / cm^2·sec
F6: Trk length est. of energy deposition All particles	cell	MeV / g·sec
F7: Trk-len est. of fission energy dep. N	cell	MeV / g·sec
F8: Pulse height tally All particles	cell	pulses / sec

*** = sec or nothing, depending on source units

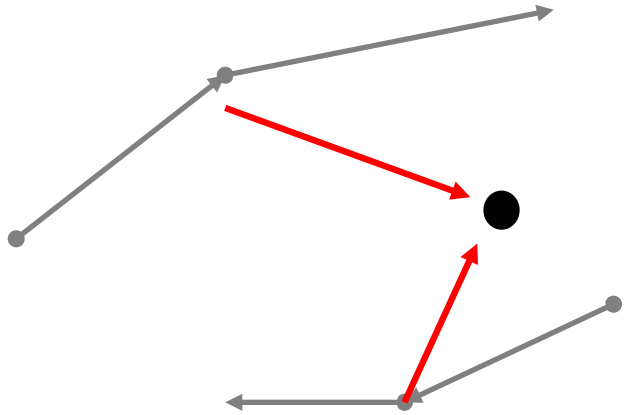
Tally Card Contents

- **F1, F2, F4, F6, F7, F8: Fn:s {list of surfaces or cells}**
  (not F5)

  - **n = tally number = i + 10j** **i = 1, 2, 4, 6, 7, 8**
    0 ≤ j ≤ 9999
    - **Last digit is tally type**
    - **F4:n, F14:n, F124:n are all "F4" tallies**
  - **s = particle type = particle symbol, n, p, e, etc.**
  - **May group entries by parentheses**
  - **Optional entry "T" at the end to give total**
  - *** before F1, F2, F4, F8 results in tally being multiplied by particle energy, and units by MeV**
  - *** before F6, F7 changes units to jerks/g (1 jerk = 1 GJ = 10^9 Joules)**

Tally Card Contents (Flux at Point)

- **Form:** **Fn:s X Y Z R**



  - **n = tally number = 5 + 10j** $0 \le j \le 99$
  - **s = particle type = particle symbol, n or p (no charged particles)**
  - **At every collision, makes a deterministic estimate of flux**
  - **X, Y, Z are position of tally where flux is desired**
  - **R is the radius of a "sphere of constant flux"**
    - **Required to keep tally variance finite**
    - **Recommended to be about one mean free path**
    - **Use 0.0 in a void (vacuum) region**
  - *** before F5 results in tally being multiplied by particle energy, and units by MeV**

Tally Examples

F14:n 10 30 50	F4 neutron cell flux, 3 bins
F994:n (10 30) 50	F4 neutron cell flux, 2 bins
F44:p 10 11 12 T	F4 photon cell flux, 4 bins
F105:p 3.2 4.1 5.7 0.0	F5 photon flux at point, in a void
F35:n 100. 17. 0.0 5.0	F5 neutron flux at point, in a material
*F8:p 10 25	F8 tally of photon energy deposited in cells, MeV / sec

Tally Fundamentals (continued)

Tallies Require Volumes or Areas

- **For tallies (except F5) to be valid, MCNP must know a volume or area to perform the division.**

- **Sometimes, MCNP will be unable to calculate the volume of cells or areas of surfaces. You must then provide them!**

- **Three methods of doing this:**

  1) Specify `vol = ####` on the respective cell card.

  2) Specify a list of volumes or areas for every cell or surface in the problem using the vol or area cards:

  VOL V_1 V_2 . . . V_m

  AREA A_1 A_2 . . . A_n

  3) Use a segment divisor (SD) card.

Segment Divisor Card (SD)

- **MCNP normalizes flux tallies by dividing by area, volume, or mass**
  - For cell flux tallies (F4), must divide by volume
  - For surface flux tallies (F2), must divide by area
  - For energy deposition (F6), must divide by mass
  - For fission heating (F7), must divide by mass

  - Can use SD card to supply areas, volumes, or masses
  - MUST do this if they are not calculated by MCNP

- **Form: SDn d1 d2 …**

  - n = tally number
  - d1, d2, … = divisors for each tally bin

  - Must have as many entries as there are tally bins for tally "n"

  - Can use 1.0 to avoid dividing by volume or area or mass
  - Can be used to get total absorption, rather than absorption/vol
  - Note: dividing by 1.0 instead of volume changes units, etc.

Commenting Tallies in the Output File

- **Commenting on what tallies are calculating is important, especially if others may look at your output file!**
- **Use of the FC card is recommended:**

  FCn A String that is a Comment

- **Example**

```
F114:n     10
FC114      Cell flux tally in cell 10.
```

- **Comment your tallies**
  - **Your coworkers will thank you!**

Tallies With Macrobodies (1)

- Surfaces of most macrobodies are formed by several distinct components (referred to as “facets”)
- Specific facet(s) must be specified for surface tallies
- Facet is identified as S.F, where S is the surface number for the macrobody and F is the facet number
- Facet numbers are fixed with respect to the orientation of the Macrobody
- Examples

Rectangular Parallelepiped (RPP)

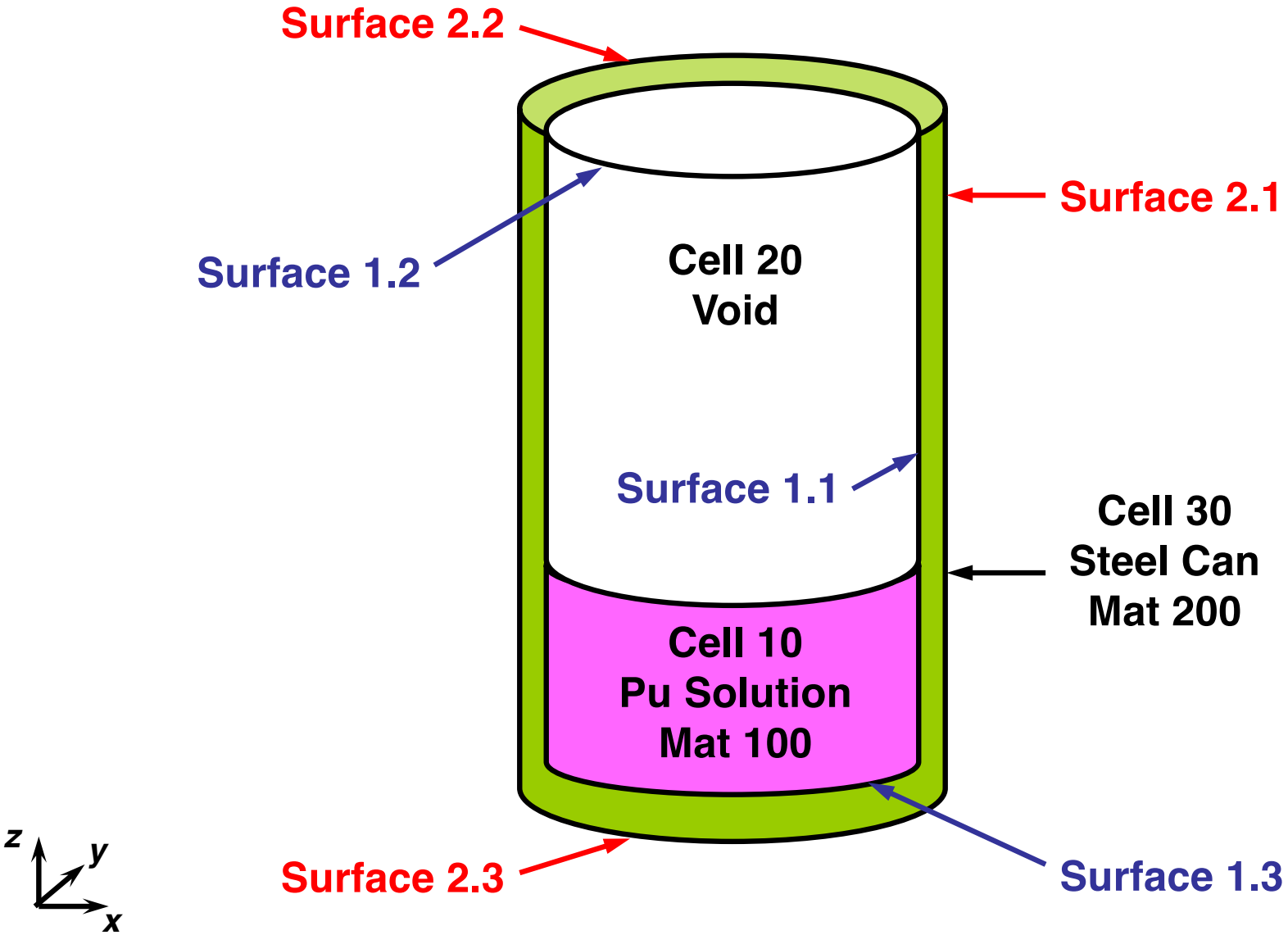
1 right side
2 left side
3 front
4 back
5 top
6 bottom

Right Circular Cylinder (RCC)

1 side of cylinder
2 top of cylinder
3 bottom of cylinder

Tallies With Macrobodies (2)

- **Surface and facet indices for puc1.txt:**



Statistics

Assessing Results

- The influence of statistical noise must be considered when assessing the reliability of Monte Carlo results.

- MCNP provides uncertainties and performs statistical checks to attempt to assess whether or not the results are reliable

  – Results of tests do not prove reliability!!!

  – The tests look for things that seem wrong.
  – The tests don't prove that results are correct.

- Confidence intervals assume that the Central Limit Theorem is satisfied.

Review: Basic Statistics

- **MCNP tally results have the form**

  RESULT RELERR

  Where

  RESULT = average score for the tally, after N histories

  RELERR = relative error in the average score, after N histories

  All tally results are normalized to be per starting particle

- **Exception: For KCODE calculations, K-effective results are reported as**

  RESULT STD

  Where

  RESULT = average score for the tally, after N histories

  STD = standard deviation in the average score, after N histories

Review: Basic Statistics

- **Average, standard deviation, relative error**
  - Let x_k = the value of a tally for the k^{th} history
    N = number of histories run (so far)
  - **Average tally**, after N histories

$$\bar{x} = \frac{1}{N}\sum_{k=1}^{N} x_k$$

  - **Standard deviation** of average tally, after N histories

$$S_{\bar{X}} = \frac{1}{\sqrt{N-1}}\sqrt{\frac{1}{N}\sum_{k=1}^{N} x_k^2 - \bar{X}^2} \approx \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N} x_k^2 - \bar{X}^2}$$

  - **Relative error** in average tally, after N histories

$$RELERR = \frac{S_{\bar{X}}}{\bar{X}} \qquad RELERR \propto S_{\bar{x}} \propto \frac{1}{\sqrt{N}}$$

Review: Basic Statistics

- **Relative error vs number of histories (N)**

$$RELERR \propto S_{\bar{x}} \propto \frac{1}{\sqrt{N}}$$

  - To cut the relative error in half, must run four times as many histories
  - To reduce relative error by 10x, must run 100x times as many histories

- **Precision**

  The RELERR or STD DEV reflect the precision of results, ie, the uncertainty in the result caused by statistical fluctuations in the Monte Carlo simulation

- **Accuracy**

  The accuracy of a result is how close the average tally is to the true physical quantity being estimated.

  Accuracy depends on the geometry approximations, cross-section data realism, material definitions, physics approximations, code approximations, etc.

- **Running more histories will improve the precision of a result, not the accuracy of a result.**

Confidence Intervals

- **Confidence interval**
  - **Using the computed STD DEV as an estimate of σ, we can estimate, by the Central Limit Theorem, the probability that the true mean lies with an interval:**

$$\text{Prob}\left\{ \bar{x} - 1 \cdot s_{\bar{x}} \le \mu \le \bar{x} + 1 \cdot s_{\bar{x}} \right\} = 68\%$$

$$\text{Prob}\left\{ \bar{x} - 2 \cdot s_{\bar{x}} \le \mu \le \bar{x} + 2 \cdot s_{\bar{x}} \right\} = 95\%$$

$$\text{Prob}\left\{ \bar{x} - 2.6 \cdot s_{\bar{x}} \le \mu \le \bar{x} + 2.6 \cdot s_{\bar{x}} \right\} = 99\%$$

  - **Think about what this means …..**
    - If you repeat a calculation many times, it is likely that 1/3 of the time the true result will lie outside of the computed 1σ confidence interval

MCNP - Ten Statistical Checks

MCNP performs 10 statistical checks on tallies to try & assess whether they are valid results

1. Estimated mean should have random behavior for last half of problem
2. Estimated relative error should be < 0.05 for point detector, < 0.10 for others
3. Estimated RELERR should monotonically decrease in last half of problem
4. Estimated RELERR should decrease as $1/N^{1/2}$ in last half of problem
5. Estimated variance of the variance (VOV) should be < 0.10
6. Estimated VOV should monotonically decrease in last half of problem
7. Estimated VOV should decrease as $1/N$ in last half of problem
8. Estimated FOM should not have obvious trends in last half of problem
9. Estimated FOM should show random behavior in last half of problem
10. Tail of tally probability density should fall off as $1/x^m$, with $m>3$

Reaction Rates

Reaction Rates

- **Often some reaction rate may be desired rather than just flux.**

$$R_x = N\sigma_x\phi$$

$$\left(\frac{\text{nuclides}}{\text{barn}\cdot\text{cm}}\right)\bullet\left(\frac{\text{barns}}{\text{nuclide}}\right)\bullet\left(\frac{1}{\text{cm}^2\cdot\text{sec}}\right) = \frac{1}{\text{cm}^3\cdot\text{sec}}$$

Units = Reactions per unit volume (per unit time)

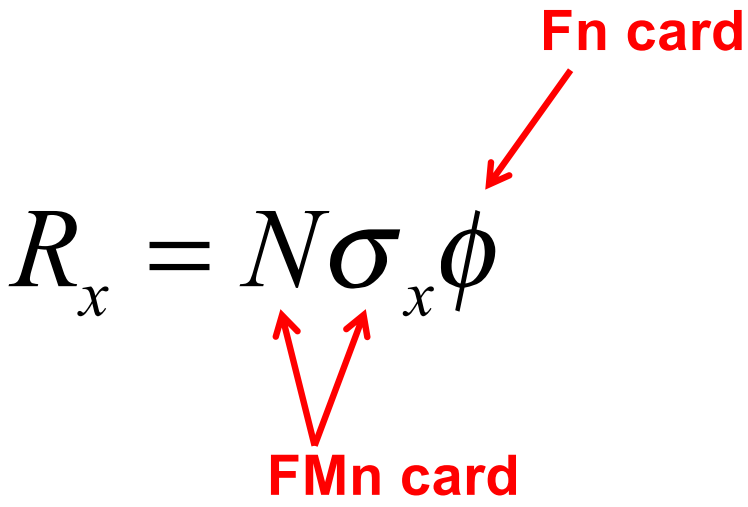
- **Need some way to multiply the flux tally scores by the number density and the microscopic cross section for reaction *x*.**

- **MCNP can do this with the tally multiplier, or FM, card.**

- **Tally multiplier card can also scale by constants and has additional uses.**

Reaction Rates

- **Reaction rate tallies**



- **Fn card specifies:**
  - Type of tally - last digit of n
  - Type of particle (eg, F4:n)
  - Where to make the tally - cells or surfaces

- **FMn card specifies:**
  - Multiplier – constant, N, &/or N's from a material
  - Type of cross-section – many options

Tally Multiplier Card

Note: Discussion here will be limited to the multiplier form of the tally multiplication card

- **Form:** FMn C m B_1 (B_2 . . . B_i) . . . (B_j . . . B_m) B_{last}
  - n is tally number (e.g., FM24)
  - C is a multiplicative constant
    - C > 0 means multiply tally by C (e.g., source intensity)
    - C < 0 means multiply tally by |C|, and by atom density in the tally cell
  - m is a material number (from an Mm card)
  - B_k is a reaction type identifier
  - B_{last} is either blank or T (T sums over previous tallies)

Combinations of Reaction Type Identifiers

- Reaction identifiers can be combined, either additively or multiplicatively, to form a single tally
- All component identifiers must be enclosed in a single set of parentheses
- Colon between two reaction type identifiers means they are to be added $(B_i : B_j : B_k)$
- Blank space between two reaction type identifiers means they are to be multiplied $(B_i \ B_j)$
- If no ( ), precedence of operations is multiply first, then add

Reaction Type Identifiers

- **Reaction type identifiers can be either positive or negative**
  - **Positive values correspond to ENDF reaction types (MT numbers)**
  - **Negative values are MCNP-specific to the type of library (multipgroup or continuous-energy) and particle (neutron or photon) employed**

MT	Reaction Type
1	Total
2	Elastic scatter
18	Fission
101	Capture
102	(n,γ)
etc.	see Manual

Type	Neutrons		Photons
	Continuous	Multigroup	
-1	total*	total	incoherent scatt
-2	capture	fission	coherent scatt
-3	elastic scattering*	ν (neutrons/fission)	photoelectric
-4	heating (MeV/coll)	χ (fission spectrum)	pair production
-5	γ production	capture	total
-6	fission	stopping power	photon heating
-7	ν (neutrons/fission)	momentum transfer	
-8	Q (MeV/fission)		

Note: The MCNP manual sometimes reverses the usual definitions of capture and absorption (physicists versus nuclear engineers). Throughout this presentation, we will use absorption = fission + capture

FM Card Examples

- **Fission rate in cell 10, which contains material 100, with continuous-energy neutron data**

```
F14:n    10
FM14     -1.0   100    -6
```

- **Fission energy deposition rate in cell 10, which contains material 100, with continuous-energy neutron data**

```
F24:n    10
FM24     -1.0   100     -6   -8     $ number densities from mat 100
                                     $ ( σF · Q )
```

- **Absorption rate in cell 10, which contains material 100, with continuous-energy neutron data**

```
F34:n    10
FM34     -1.0   100     -2 : -6     $ number densities from mat 100
                                     $ ( σC + σF )
```

Flux & K-effective Estimators

Background on Monte Carlo Estimators

- **Pathlength estimator for flux**
  - Flux = total pathlength traveled by all neutrons per unit volume per unit time
  - For flux in a cell, $\phi \approx \frac{1}{W \cdot V} \sum_{\text{all flights in cell}} d_k \cdot \text{wgt}$

V = cell volume

W = total starting weight

- **Collision estimator for flux**
  - Collision rate = $\Sigma_T \phi$, so ϕ = [collision rate]/ Σ_T
  - For flux in cell,

$$\phi \approx \frac{1}{W \cdot V} \sum_{\text{all collisions in cell}} \frac{\text{wgt}}{\Sigma_T}$$

- **Absorption estimator for flux**
  - Absorption rate = $\Sigma_A \phi$, so ϕ = [absorption rate]/ Σ_A
  - For flux in cell,

$$\phi \approx \frac{1}{W \cdot V} \sum_{\text{all absorptions in cell}} \frac{\text{wgt}}{\Sigma_A}$$

Single-cycle Keff Estimators

Neutron production rate = $\nu\Sigma_F\ \phi$

W = total weight starting cycle n

- **Pathlength estimator for Keff, for cycle n**

$$K_{path}^{(n)} = \left(\sum_{\substack{\text{all} \\ \text{flights}}} wgt_j \cdot d_j \cdot \nu\Sigma_F \right) \Big/ W$$

- **Collision estimator for Keff, for cycle n**

$$K_{collision}^{(n)} = \left(\sum_{\substack{\text{all} \\ \text{collisions}}} \frac{wgt_j}{\Sigma_T} \cdot \nu\Sigma_F \right) \Big/ W$$

- **Absorption estimator for Keff, for cycle n**

$$K_{absorption}^{(n)} = \left(\sum_{\substack{\text{all} \\ \text{absorptions}}} \frac{wgt_j}{\Sigma_A} \cdot \nu\Sigma_F \right) \Big/ W$$

Spectra & Plotting

Obtaining Energy Spectra

- **Often, spectral information is often very important in criticality safety applications**

- **An energy binning can be added to tallies with an E card:**

```
En        e1   e2   . . .  ei   . . .  eK
```

  - **The index `n` corresponds to a tally index defined on the `F` card**
    - If `n = 0`, then it is the default for all tallies
  - **Each `ei` are energy bin boundaries in MeV**
  - **Implied lower bound always 0 MeV**
  - **Tallies are integrated over the entire energy bin (not in per MeV)**

- **For F1 tallies, can bin in direction cosine with the C card**
- **For fixed source problems, can bin in time with T card (not applicable for criticality)**

Exercise: puc1tal

- **Copy puc1.txt to puc1tal.txt**

- **Insert tallies for:**
  - **Surface leakage spectrum on outside of drum**
    - Top surface
    - Bottom surface
    - Side surface
    - Total leakage on outer surface
    - Energy binning
      - 100 logarithmic bins
      - Range = 1.0E-8 – 10 MeV
  - **Total fission neutron production in solution**

- **Run the problem, analyze the output file**
  - **Search for the string `1tally`**

Exercise: puc1tal input file

```
puc1tal - single cylinder
c >>>>> cell cards
. . .
c >>>>> surface cards
. . .
c >>>>> data cards
. . .
c tally specifications
c
f1c  Surface current on external surface of tank
f1:n  2.1  2.2  2.3  t
e1   1e-8 98ilog 10
c
f4c  Total neutron production nu*sigmaf*flux
f4:n 10
fm4  -1.0 100 -6 -7
sd4  1.0
```

Exercise: puc1tal results

```
 source distribution written to file srctp         cycle=   100
      run terminated when     100 kcode cycles were done.

 =====>   76.26 M histories/hr    (based on wall-clock time in mcrun)

. . .
. . .
. . .

 final k(col/abs/trk len) = 0.88778     std dev = 0.00363

 warning.  ratio of standard deviations for problem halves too large.
 warning.      1 of   2 tallies had bins with large relative errors.

 ctm =         0.08   nrn =           23634781
 dump    2 on file runtpe   nps =       100421   coll =       1179582
 mcrun  is done
```

Exercise: puc1tal results

```
1tally        1        nps =      100421
+  Surface current on external surface of tank
           tally type 1    number of particles crossing a surface.
           particle(s): neutrons
 number of histories used for normalizing tallies =        75000.00

 surface  2.1
      energy
    1.0000E-08   1.72462E-04 0.2439
    1.2328E-08   1.48487E-04 0.2733
. . .
 surface  2.2
. . .
 surface  2.3
. . .
 surface union total
. . .
 total       4.70666E-01 0.0038
```

Exercise: puc1tal results

```
1tally        4        nps =      100421
+  Total neutron production nu*sigmaf*flux
           tally type 4    track length estimate of particle flux.
           particle(s): neutrons
 number of histories used for normalizing tallies =        75000.00

           volumes
                   cell:      10
                         1.00000E+00

 cell  10
 multiplier bin:  -1.00000E+00       100          -6       -7
                 8.83523E-01 0.0060
```

Tally Plotting

- After looking at the output of puc1tal.txt plot the tallies
- Read in the runtpe file to plot energy spectrum (tally 1) of current leaving drum

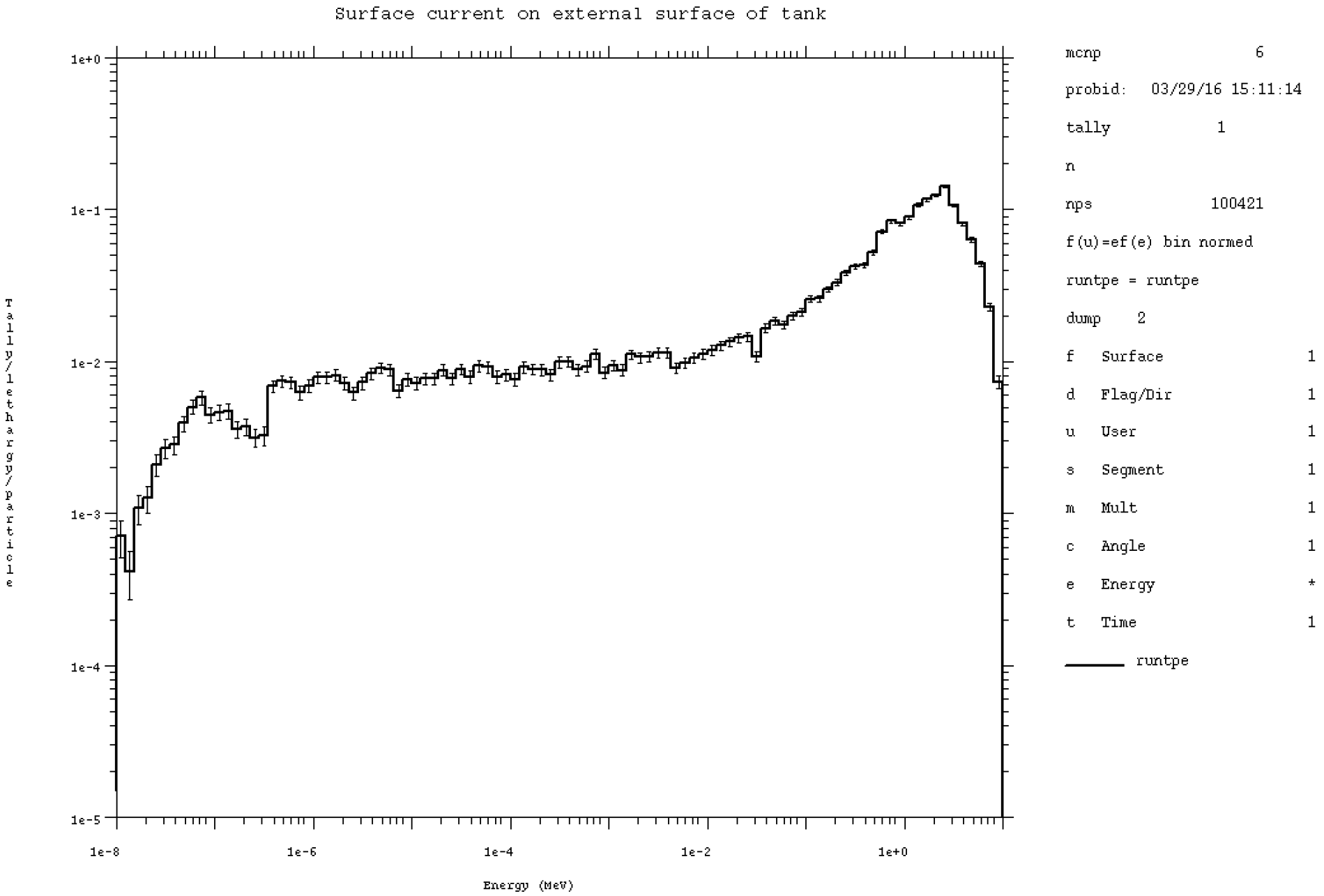
```
mcnp6  z   r = runtpe
```

  - Replace "runtpe" with the name of your runtpe file

- In the plotting command window, type text in `red`:

```
tal  1                          specifies the tally to plot
loglog                          has plot on log-log scale
lethargy                        changes to per lethargy normalization
xlim 1.e-8  1.e+1   sets the x-range of the plot window
```

Tally Plotting



Multiple Tally Plotting

- **In the plotting command window, next type text in `red`:**

```
fixed f 4 label "Surface Total" &
coplot fixed f 1 label "Surface 2.1" &
coplot fixed f 2 label "Surface 2.2" &
  1
coplot fixed f 3 label "Surface 2.3"
```

plots 4th surface of tally 1

plots 1st surface of tally 1

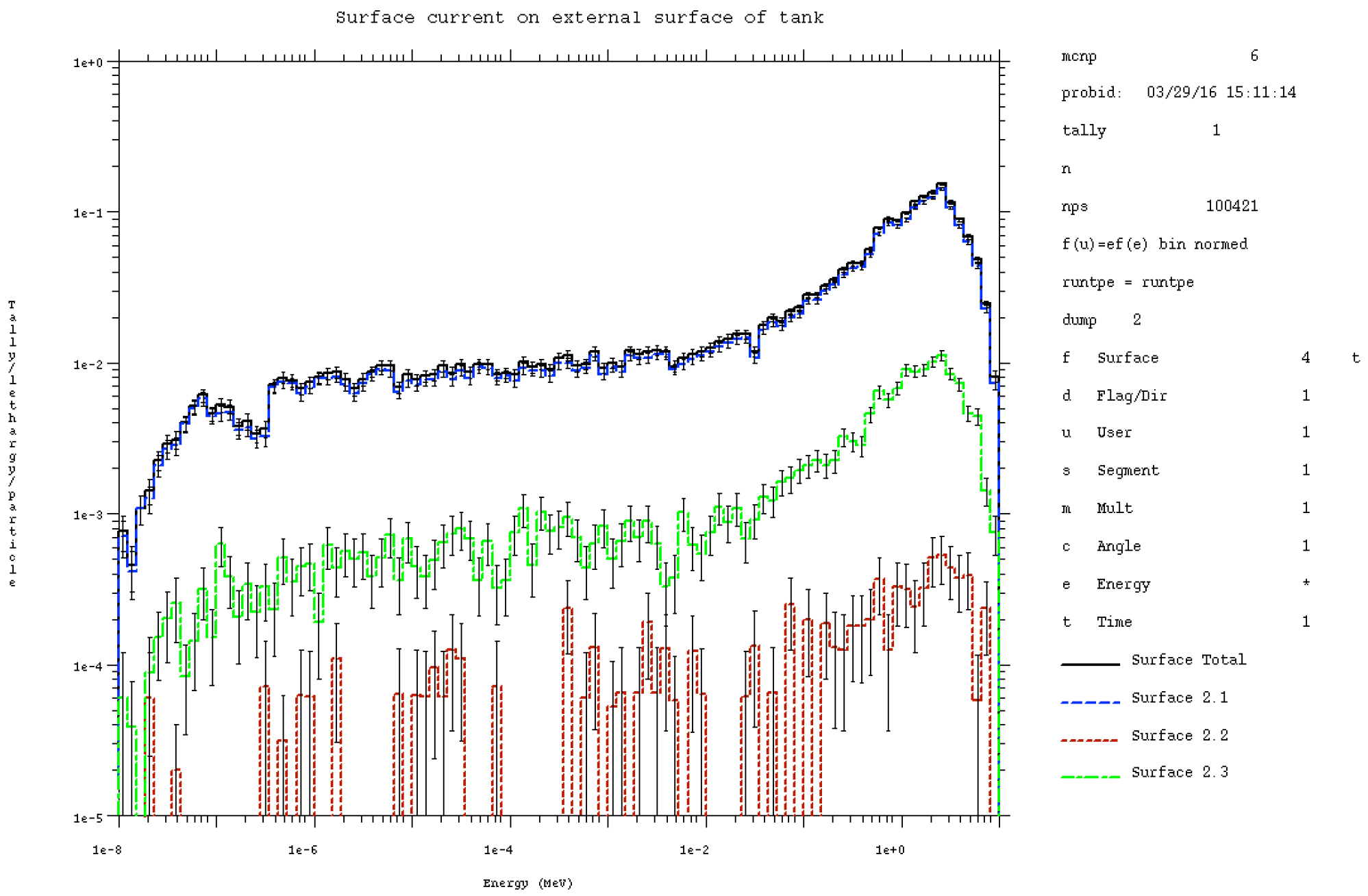
plots 2nd surface of tally 1

plots 3rd surface of tally 1

- **See manual for many more plotting options or type `help` at command prompt**

```
 mcplot>
help
Available commands:
            bar                    mt                    term
. . .
 Type "help all" for a verbose list of all help commands
  "help <command>" to list a specific help command,
  "help overview" for an overview of MCPLOT,
  or "help execute" for MCPLOT input & execution-line options.
```

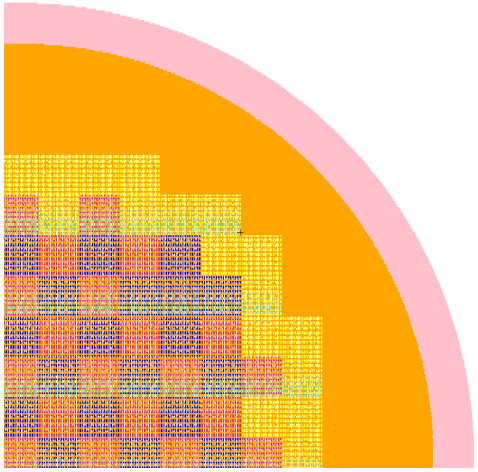
Multiple Tally Plotting



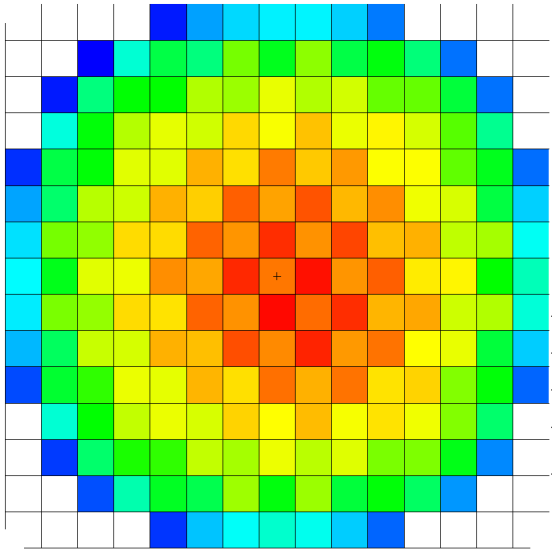
Mesh Tallies

Motivation

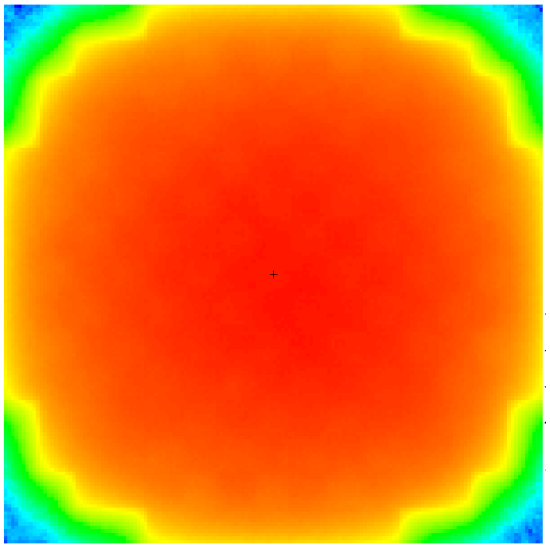
Geometry Model (1/4)



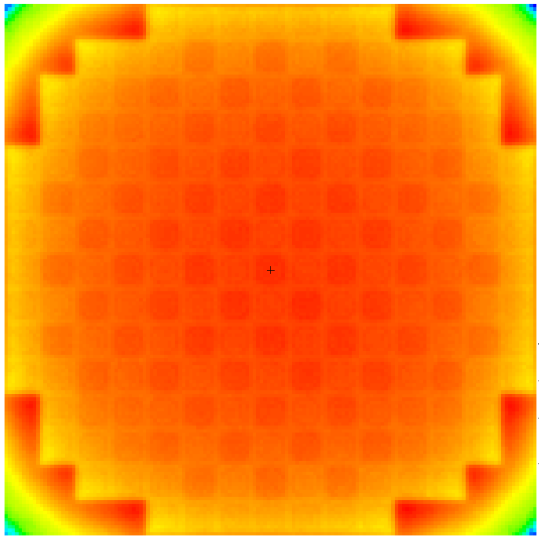
Assembly Powers



Fast Flux

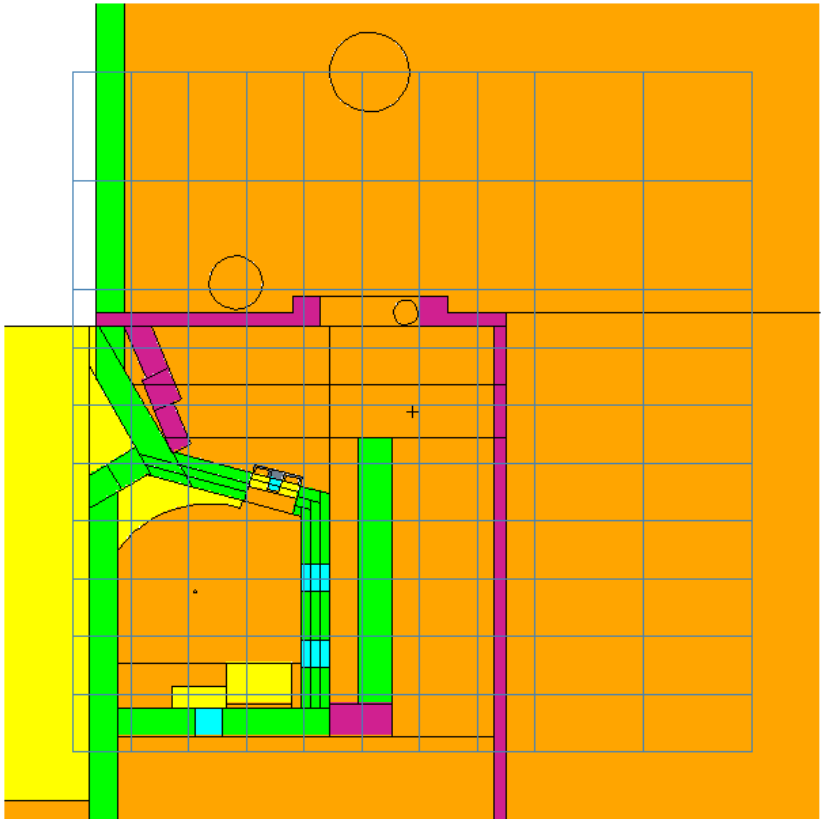


Thermal Flux



Mesh Tallies

- **Mesh tallies cover 3D regions of space independent of the problem geometry**
  - **Can be used to tally flux, reaction rates, heating, particle birth, fission source points, …**
  - **Rectangular, cylindrical, & spherical meshes**
  - **Bin on energy & time values**
  - **Unlimited number of meshes**
  - **Size of mesh limited only by computer parameters**
  - **Rotated by using a TR card**
  - **Modified by DE/DF or FM cards**
  - **Surface flagging & cell flagging**
  - **Plot results in MCNP**



Mesh Tally Card

```
FMESHn:p  GEOM=      ORIGIN=     IMESH=        IINTS=       JMESH=
          JINTS=     KMESH=    KINTS=
```

- Can be used with DEn, DFn, and FMn cards.
- Caution: It is easy to create huge mesh tallies that can overflow computer memory.

GEOM	= mesh geometry: Cartesian (xyz or rec) or cylindrical (rzt or cyl)	xyz
ORIGIN	= x,y,z coordinates in MCNP cell geometry superimposed mesh origin	0. 0. 0.
IMESH	= coarse mesh locations in x (rectangular) or r (cylindrical) direction	---
IINTS	= number of fine meshes within corresponding coarse meshes	1
JMESH	= coarse mesh locations in y (rectangular) or z (cylindrical) direction	---
JINTS	= number of fine meshes within corresponding coarse meshes	1
KMESH	= coarse mesh locations in z (rectangular) or theta (cylindrical) direction	---
KINTS	= number of fine meshes within corresponding coarse meshes	1
EMESH	= values of coarse meshes in energy	all energies
EINTS	= number of fine meshes within corresponding coarse energy meshes	1
FACTOR	= multiplicative factor for each mesh	1.

Example

- **Example: 5 x 10 x 20 fission rate mesh tally in 5x5x5 cm box centered about the origin.**

```
fmesh4:n   geom=xyz     origin=-2.5 -2.5 -2.5
           imesh=2.5    iints=5
           jmesh=2.5    jints=10
           kmesh=2.5    kints=20
fm4    -1.0      0       -6
```

- **Material index of zero is a wildcard, uses material in the current cell.**

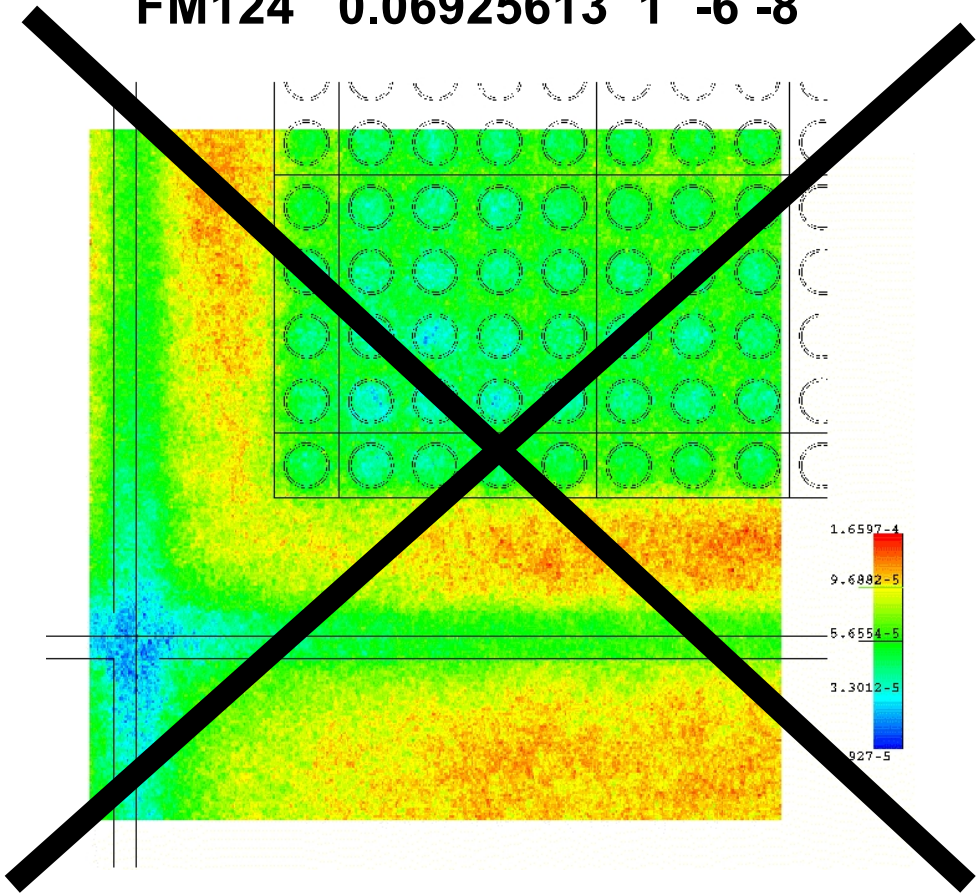
Example of the FM card

Calculate the average fission energy deposition

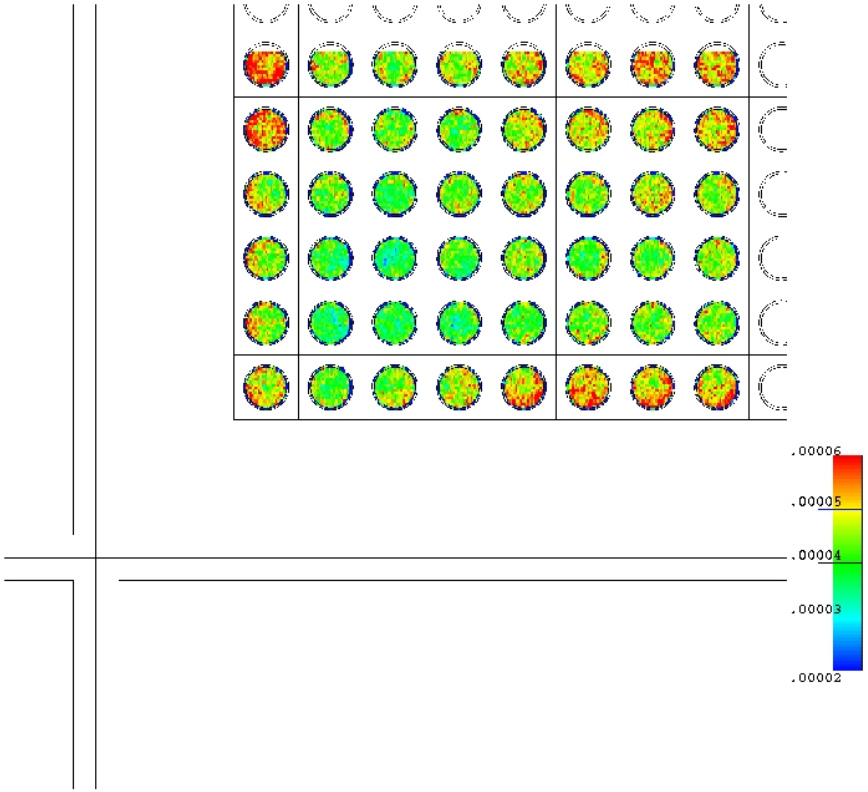
FM card format: FMn C m R1 R2

Put '0' as the material number

FM124 0.06925613 1 -6 -8



FM134 -1.0 0 -6 -8



Exercise: Mesh Tallies

- Copy puc6k.txt to puc6ktal.txt

- Insert a mesh tally to compute the neutron production from fission (#/cc) normalized per source particle as a function of space.
  - Mesh should cover the entire solution
  - Use 60 elements in x, 40 in y, and 20 in z
  - Revisit the table of special reaction numbers for the FM card
  - Remember the "0" wildcard

- Insert a second mesh tally to compute thermal and fast flux
  - Use same mesh extents as previous tally
  - Double the number of elements in each dimension
  - Set energy mesh at 0.625E-6 and 10 MeV

- Run the problem and wait for instructions on plotting

Exercise: Mesh Tallies

```
puc6ktal 3 x 2 array of cans
c >>>>> cell cards
. . .
c >>>>> surface cards
. . .
c >>>>> data cards
. . .
c tally specifications
c
fmesh14:n  geom=xyz  origin=-17.79 -17.79 0.0
           imesh=88.95 iints=60
           jmesh=53.37 jints=40
           kmesh=39.24 kints=20
fm14  -1.0  0  -6 -7
c
fmesh24:n  geom=xyz  origin=-17.79 -17.79 0.0
           imesh=88.95 iints=120
           jmesh=53.37 jints=80
           kmesh=39.24 kints=40
           emesh=0.625e-6 10.0
```

Mesh Tally Plotting

- **In the command line type:**

```
mcnp6   z   r = runpte
```

- **Where runtpe is the name of your runtpe file for puc6ktal**

- **The following commands in `red` are useful:**

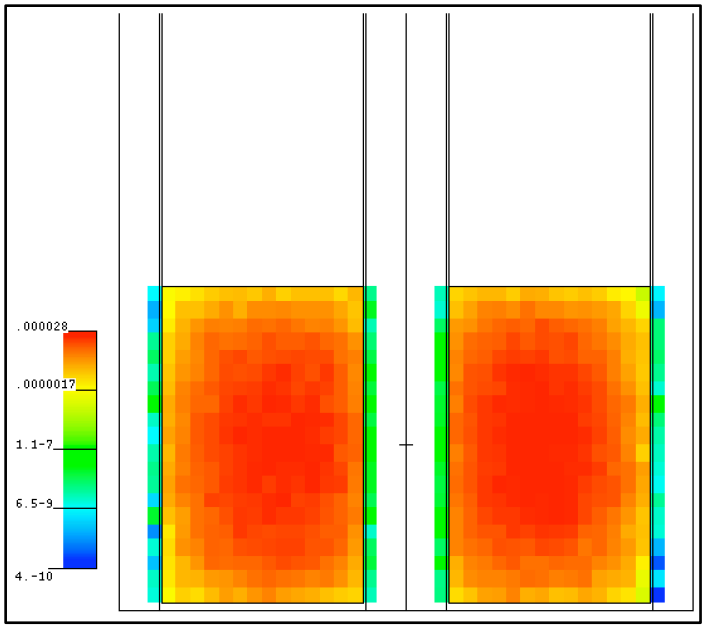
`mcplot> fmesh 24`	**Brings up the results**
`mcplot> fmrelerr`	**Plots the relative uncertainties**
`mcplot> ebin 1`	**Selects energy bin for results**

Mesh Tally Plotting

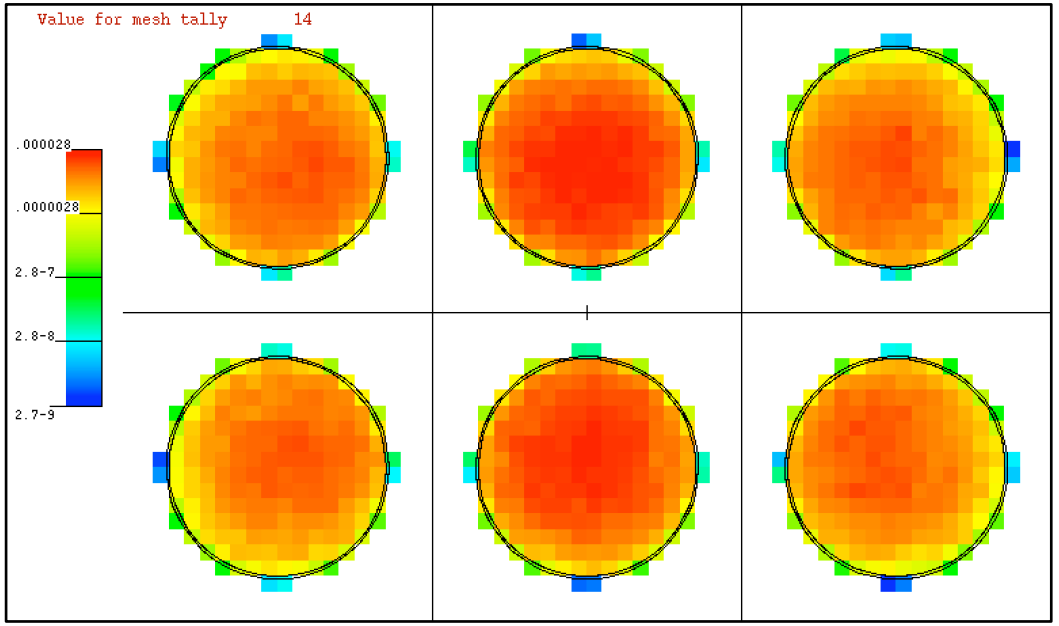
- **Neutron production rate from fission overlaid on geometry**

```
fmesh 14
fmrelerr
```

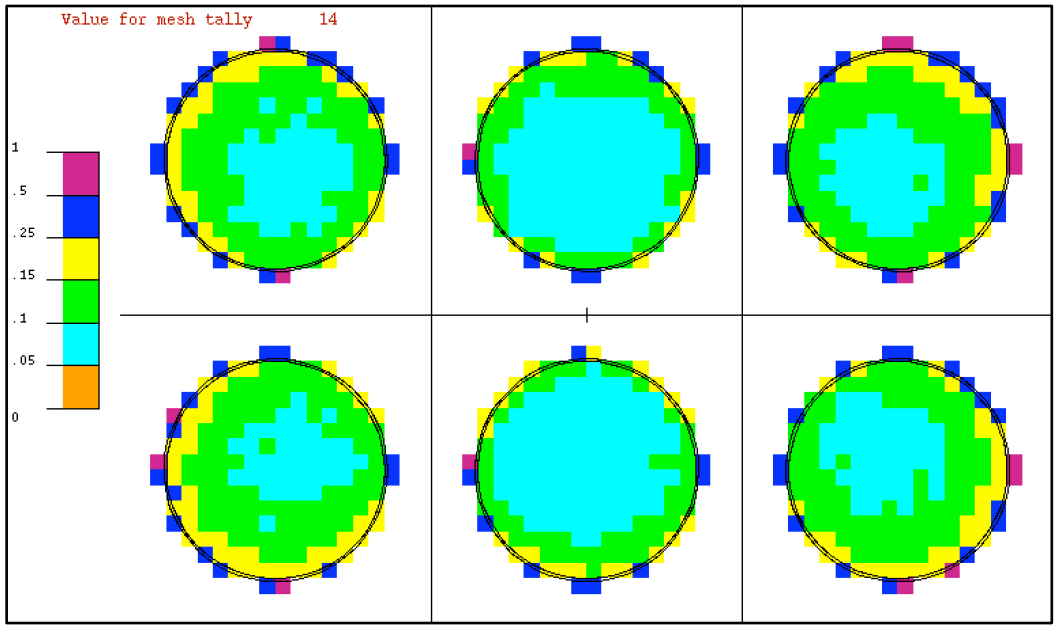
YZ Plane



XY Plane



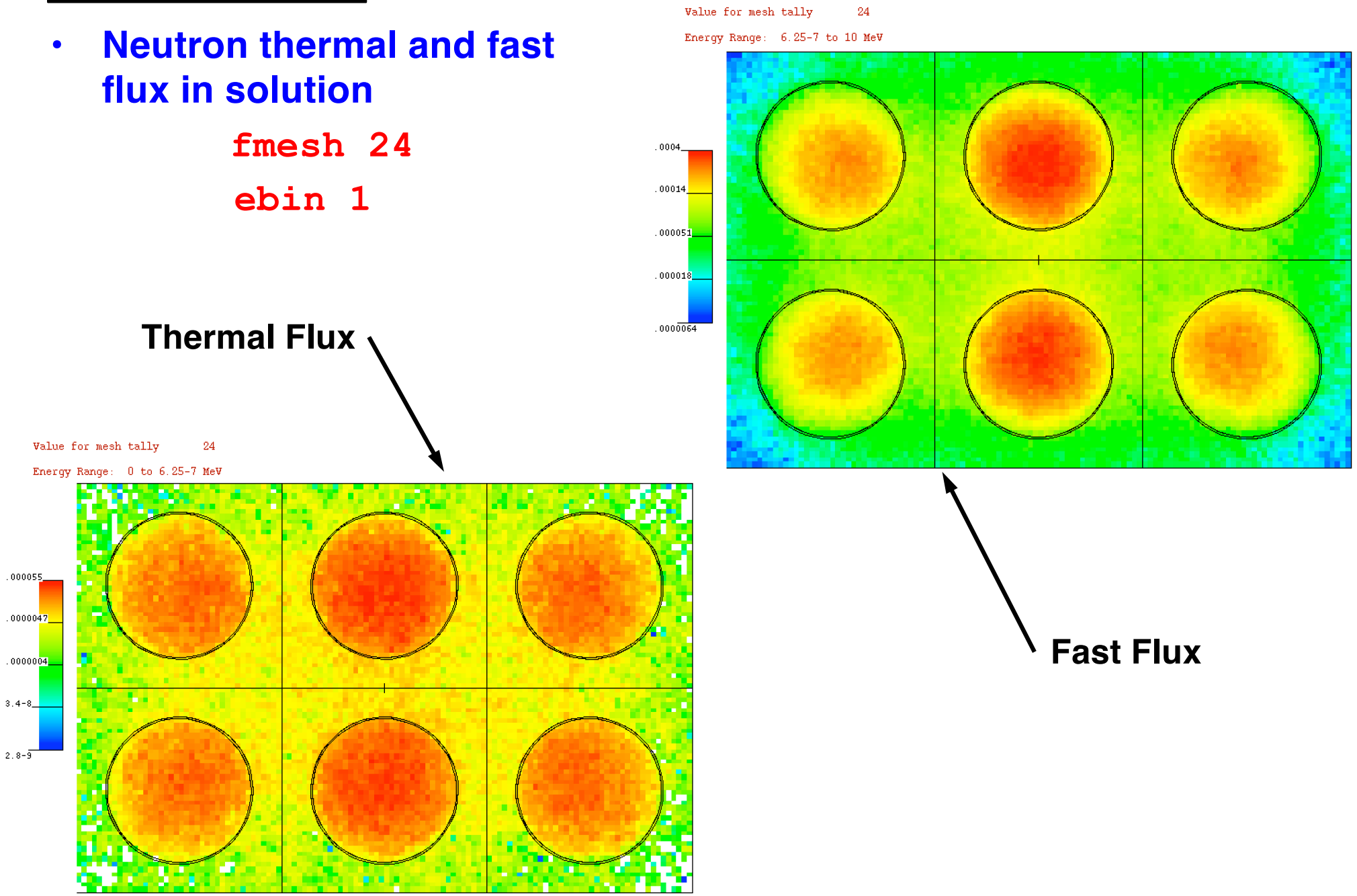
Relative Uncertainties



Mesh Tally Plotting

- **Neutron thermal and fast flux in solution**

```
fmesh 24
ebin 1
```



Discussion

- **Tallies (or edits) are necessary to obtain results in Monte Carlo**
- **In MCNP,**
  - **No flux, current or pulse height tallies are assumed**
  - **Criticality calculations (KCODE) give k-eff, ANECF, EALF, etc.**
  - **Tallies must be specified by the user in fixed source problems to obtain any results**
- **Usually flux is not very useful (unless comparing to other codes or analytic solutions)**
  - **Flux multipliers allow users to obtain reaction rates, energy deposition, neutron production, power distributions, etc.**
  - **Dose/response functions (see optional section) allow users to convolve special functions with the flux**
- **Mesh tallies are very useful and easy to apply to any problem**
  - **Geometry independent (superimposed mesh)**
  - **Watch out for:**
    - Poor statistics when volumes get smaller
    - Memory consumption with multiple mesh tallies

Dose Rates

OPTIONAL

Calculating Dose in MCNP

- **There are two methods to compute dose (energy deposited by unit mass) in MCNP:**
  - **Explicit modeling of exposed targets (e.g., detectors, phantoms, etc.) and use of energy deposition (F6) tallies**
  - **Flux tallies (F2, F4, or F5) with appropriate dose functions**

- **F6 tallies produce absorbed dose, not biological dose**
  - **Need "dose functions" to provide quality or tissue weighting factors**

Using the F6 Tally

- **If possible, model the target explicitly in the geometry and use an F6 tally to compute dose**
  - **Advantage: most exact as effects of target on radiation field capture**
  - **Disadvantage: not always practical to model everything (e.g., locations of individuals standing in a room)**

- **F6 tally is an F4 tally modified by the total cross section and heating number**
  - **Equivalent to an F4 with an FM4 card with (-1 -4)**
  - **Units are MeV/g, use FM card to convert units to rad or Gy**

- **Require to use DE and DF cards (next slide) with quality factors if biological dose required**

Dose/Response Function Cards (DE, DF)

- **Function to modify a tally response with some interpolated function (e.g. particle flux to human biological dose equivalent rate)**

$$\text{Dose} = \int_E D(E)\phi(E)dE$$

DEn A E1 E2 . . . Ek

- Ei = energy points (MeV)
- A = LOG or LIN energy interpolation method

DFn B F1 F2 . . . Fk

- Fi = corresponding value of the dose function at each energy on DEn
- B = LOG or LIN dose interpolation method

- **Appropriate for dosimetry when effect of "target" on the radiation field is small (e.g., a small detector)**

Exercise: shield04 – Dose Calculation

- Copy shield01.txt to shield05.txt.

- Add tally to compute the biological dose rate (rem/hr) from neutrons to a worker standing 1 meter from the back of the shield
  - Source strength of 3.e8 neutrons per second
  - Copy the DE and DF cards from the file: shield_dedf.txt

- Run the problem and examine the output file.

Exercise: shield05 – Dose Calculation

```
shield05 - shielding calculation with 252-Cf neutron source
c >>>>> cell cards
. . .

c >>>>> surface cards
. . .

c >>>>> data cards
. . .
c ### tally specification
fc2     average dose rate in rem/hr, 1 m from shield from 3e8 neutrons
f2:n    10.1
fm2     3.e8          $ multiply by source strength 3.e8 n/s
c ### neutron flux to dose (rem/hr) factors
c !!! NOT RECOMMENDED FOR "OFFICIAL" CALCULATIONS !!!
de2     log   2.50e-8   1.00e-7   1.00e-6   1.00e-5   1.00e-4   1.00e-3
              1.00e-2   1.00e-1   5.00e-1   1.00e+0   2.50e+0   5.00e+0
              7.00e+0   1.00e+1   1.40e+1   2.00e+1
df2     log   3.67e-6   3.67e-6   4.46e-6   4.54e-6   4.18e-6   3.76e-6
              3.56e-6   2.17e-5   9.26e-5   1.32e-4   1.25e-4   1.56e-4
              1.47e-4   1.47e-4   2.08e-4   2.27e-4
```

Exercise: Dose Calculation

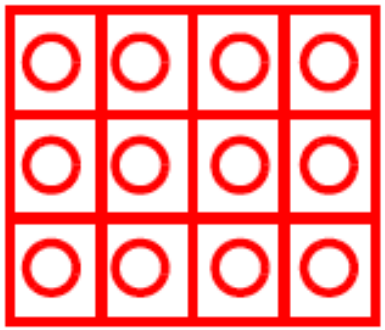
```
1tally        2        nps =      100000
   average dose rate rem/hr, 1 m from shield from 3e8
  neutrons
   tally type 2    particle flux averaged over a surface.
   tally for  neutrons

           this tally is modified by a dose function.

           this tally is all multiplied by  3.00000E+08

           areas
                surface:     10.1
                          4.00000E+04
 surface  10.1
                 1.63755E-03 0.0740
```

Tallies & Sources in Repeated Structures

OPTIONAL

Tallies in Repeated Structures

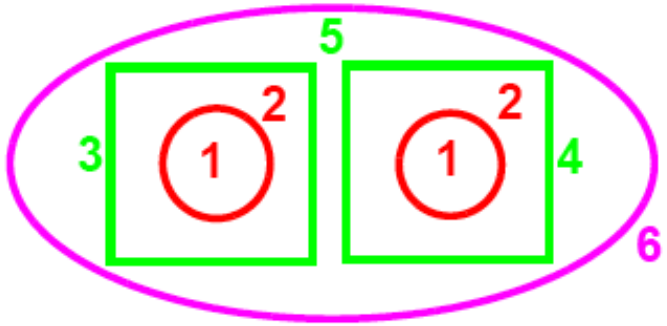
- **What's special about a tally or source in a repeated structure?**
  - **Must provide the path to tally or source cell**
  - **Which pin cell ??**



- **Enables tallies or sources in specific cells of repeated structures.**

Purpose and Definitions

- **Example - levels**




```
c cell cards
 1   0      -11   u=1
 2   0       11   u=1
 3   0      -22    fill=1                    u=2
 4   0      -33    fill=1   trcl(1 0 0)      u=2
 5   0       22 33                           u=2
 6   0      -44    fill=2
```

– Cell 6 is in the "real world", not a universe - called "Level 0"
Cells 3,4,5 are in Universe 2, which fills Cell 6 - called "Level 1"
Cells 1,2 are in Universe 1, which fills Cell 3 and Cell 4 - called "Level 2"

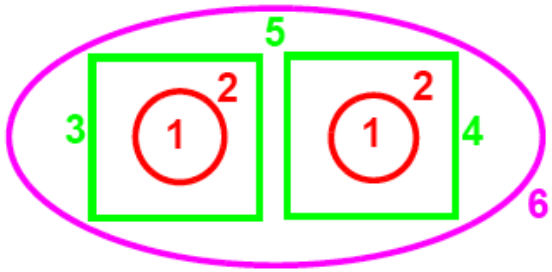
– The left pin (Cell 1) is contained in Cell 3, that is contained in Cell 6.

– Use the symbol "<" to mean "is contained in":

1 < 3 < 6 <-- This uniquely identifies the left pin
1 < 4 < 6 <-- This uniquely identifies the right pin

Geometric Paths & Tally Paths

- **Geometric chain**
  - A list of cells, one in each level of geometry, that uniquely specifies a particular cell in a repeated structure
  - Start at the deepest level of geometry, finish in the real world
- **Tally path**
  - A subset of a geometric chain



- **Example**

  C1 < C2 == "Cell C1 is contained in Cell C2"

  - All geometric paths: 1<3<6 2<3<6 1<4<6 2<4<6
    3<6 4<6 5<6 6
  - Possible tally paths:

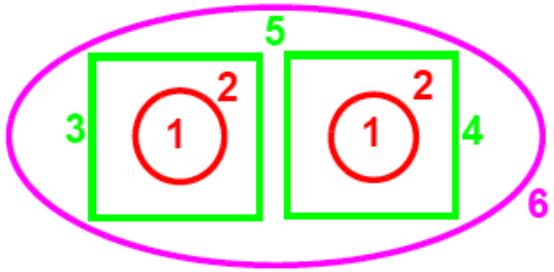
    1<3<6 <-- the left pin
    1<4<6 <-- the right pin
    1<6 <-- BOTH pins

Tally Paths

FORM: **Fn:p (E1 < C1 < C2) ...**

- Left arrow (<) identifies levels within a tally chain, translate it to: "is contained in"
- Requires an outer set of parentheses.
- First level entries (Ei) are either:
  - Tally surfaces if tally type 1 or 2.
  - Tally cells if tally type 4, 6, 7, or 8
- Upper level entries (Ci) must be:
  - Cells with a FILL entry that is nonzero.
  - Produces a tally only when a particle is in a geometric chain that corresponds to an input tally chain.

Tally Paths




```
c cell cards
 1   0      -11   u=1
 2   0       11   u=1
c
 3   0      -22    fill=1                  u=2
 4   0      -33    fill=1  trcl(1 0 0)    u=2
 5   0       22 33                        u=2
 6   0      -44    fill=2

C surface cards
. . .

C data cards -- flux tallies
F4:n (1<3<6)        <-- the left pin
     (1<4<6)        <-- the right pin
     (1<6)          <-- BOTH pins
```

Tally Paths - Lattices

Form: Fn:p ((E2 E3) < (C3 C4 [I1 ... I2]) < (C5 C6)) ...

- **Brackets, [ ], identify one or more elements of a lattice:**
  - **Must follow a filled lattice cell (C4).**
  - **A lattice cell listed without brackets gives the union**

- **Three possible formats: over all lattice elements.**

[I]

Indicating the I-th lattice element in fully specified FILL array.

[I1:I2 J1:J2 K1:K2]

Indicating one or more lattice elements (see FILL card).

[I1 J1 K1, I2 J2 K2, ...]

Indicating lattice element (I1, J1, K1), (I2, J2, K2), etc.

Tallies - Multiple Bin Format

- Automatically invoked for levels with multiple entries.
- Can be disabled at any level by ( ) around all entries.
- Number of total bins generated is given by the product of the number of bins at each level.
- The order of generated tally bins can be important,
  - (E4 E5 < (C1 C2) < C3 C4) becomes:

    (E4 < (C1 C2) < C3) (E5 < (C1 C2) < C3)
    (E4 < (C1 C2) < C4) (E5 < (C1 C2) < C4)

  - A segment divisor entry (SD card) may be input for each generated tally bin.

Sources in Repeated Structures

- KSRC Format
- SDEF CEL Format

KSRC Format

- No special format for repeated structures
- Enter x,y,z locations in the coordinate system of the highest level (the real world)
- All source points are absolute coordinates, ie, "real world" coordinates

Ksrc 1. 1. 1. 2. 3. 4. 5. 6. 7. . . .

SDEF CEL Format

From the manual, pages 3-59 - 3-61:

- The coordinate system for position and direction sampling (pds) is the coordinate system of the first negative or zero Ci in the geometric path starting from the right and proceeding left.
- Each entry in the source path represents a geometry level, where level zero is the last source path entry, level one the second to the left, etc., and level zero is above level one, level two is below level one.
- The pds level is the level associated with the pds cell or pds coordinate system. All levels above the pds level must be included in the source path. Levels below the pds level need not be specified, and when given, may include one or more zero entries.
- The default pds level is the first entry in the source path when the path has no negative or zero entry.
- Position rejection is done in cells at all levels where Ci≠0, but if any Ci has a negative universe number on its cell card and is at or above the pds level, higher level cells are not checked.

SDEF CEL Format

- **To use the SDEF card to select a particular cell inside a lattice,**

```
SDEF          x=d1   y=d2   z=d3    cel=d4
… (xyz position disributions) …
SI4   L   ( geometric-path-to-cell )
SP4     1
```

- **Example - center cell in Problem tal08**

```
SDEF     x=d1  y=d2  z=d3   cel=d4
SI1        -.7   .7
SP1         0    1
SI2        -.7   .7
SP2         0    1
SI3        -180. 180.
SP3         0    1
SI4   L    ( -10 < 40[ 0 0 0 ] < 50 )        <-- xyz coordinates in "Level 2"
SP4          1
```

SDEF CEL Format

- **Example - all fuel pins in Problem fv5**

```
SDEF     x=d1  y=d2  z=d3   cel=d4
SI1      -.7   .7
SP1       0    1
SI2      -.7   .7
SP2       0    1
SI3      -180. 180.
SP3       0    1
SI4   L  ( -10 < 40[-7:7  -7:7  0:0]< 50 )      $  xyz coordinates in "Level 2"
SP4        1   224r
```

Notes:

(1) 15 x 15 x 1 = 225 lattice elements listed on SI4

(2) Must give (relative) probability for each on SP4

(3) The minus sign in SI4 indicates which coordinate system the x,y,z sampling is done in (Level 2)

(3) Despite the order on the SDEF card,

the CEL parameter is chosen first, then x, then y, then z

Adjoint-Weighted Tallies

Perturbation Theory
Point-Kinetics Parameters
Nuclear Data Sensitivities

Perturbation Theory

Perturbation Theory

- **First-order estimate of the change in some response to a change in some parameter**

- **In the literature for some time,**
  - J. D. LEWINS, *Importance: The Adjoint Function*, Permagon Press, Oxford, United Kingdom (1965).
  - G. I. BELL and S. GLASSTONE, *Nuclear Reactor Theory*, Van Norstrand Reinhold, New York, New York (1970).

- **Example: change in reactivity can be estimated,**

$$\Delta\rho = -\frac{\left\langle \psi^{\dagger}, \left(\Delta\Sigma_t - \Delta S - \frac{1}{k} F \right) \psi \right\rangle}{\left\langle \psi^{\dagger}, F' \psi \right\rangle}$$

- **Requires adjoint functions to compute correctly.**

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.

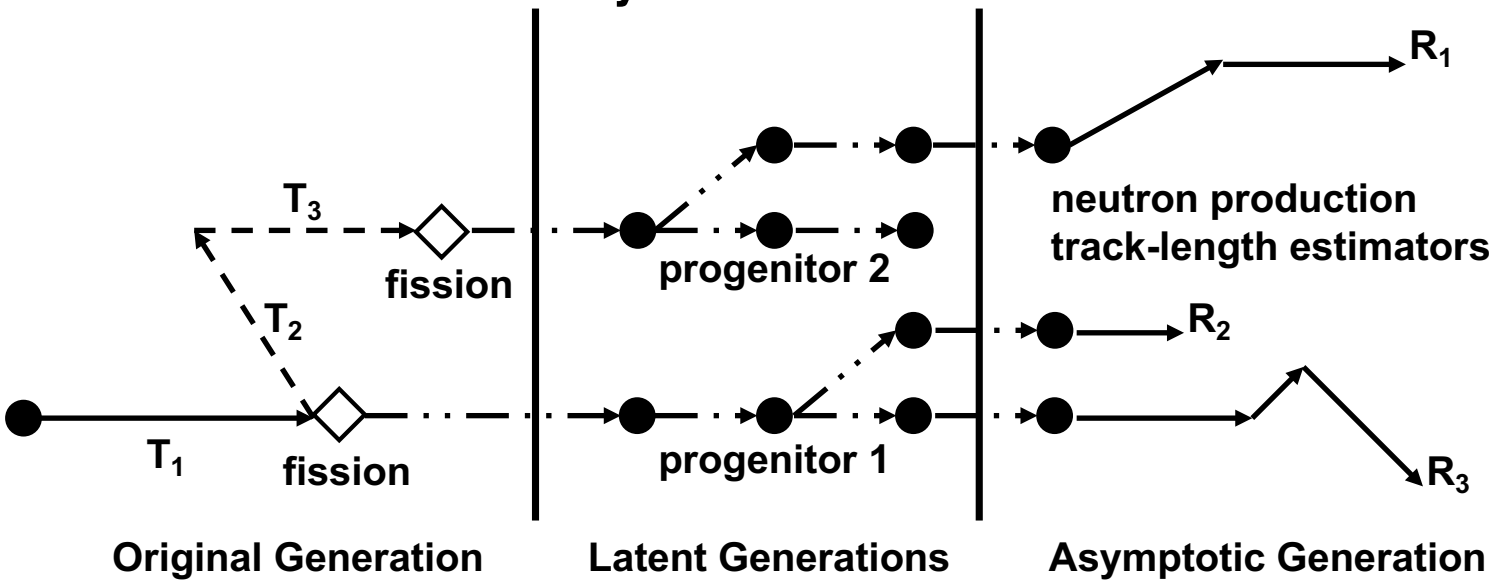
- **Note: Neutron balance is enforced in the *k*-eigenvalue problem regardless of true criticality.**

Monte Carlo Adjoint Calculations

- **The adjoint transport equation looks very similar to the forward transport equation**
- **Think of following particles backwards from detector (k_{eff}) to source…**
  - **Requires inverting the scattering laws**
    - Complicated for continuous-energy physics
    - Possible for multigroup physics
- **Until recently, estimating quantities requiring the adjoint function in continuous-energy Monte Carlo codes was not possible**
- **The iterated fission probability method is based on estimating the adjoint or importance weighting for continuous-energy Monte Carlo tallies during a normal forward calculation**

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.**



Point Kinetics Parameters

Point-Kinetics Equations

- **Time-dependence in a criticality accident scenario or in an insertion or removal of reactivity in a reactor can be modeled with the point-kinetics equations:**

$$\frac{dn}{dt} = \frac{\rho - \beta_{\text{eff}}}{\Lambda} n(t) + \sum_i \lambda_i C_i(t)$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{\Lambda} n(t) - \lambda_i C_i(t)$$

- **Need to know/solve for compute various parameters:**

$$\Lambda = \frac{\left\langle \psi^\dagger, \frac{1}{v}\psi \right\rangle}{\left\langle \psi^\dagger, F\psi \right\rangle} \qquad \beta_{\text{eff}} = \frac{\left\langle \psi^\dagger, B\psi \right\rangle}{\left\langle \psi^\dagger, F\psi \right\rangle}$$

- **Use the iterated fission probability method to estimate the adjoint weights during the forward calculation**

Example – Need for Adjoint-Weighting

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

unweighted

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

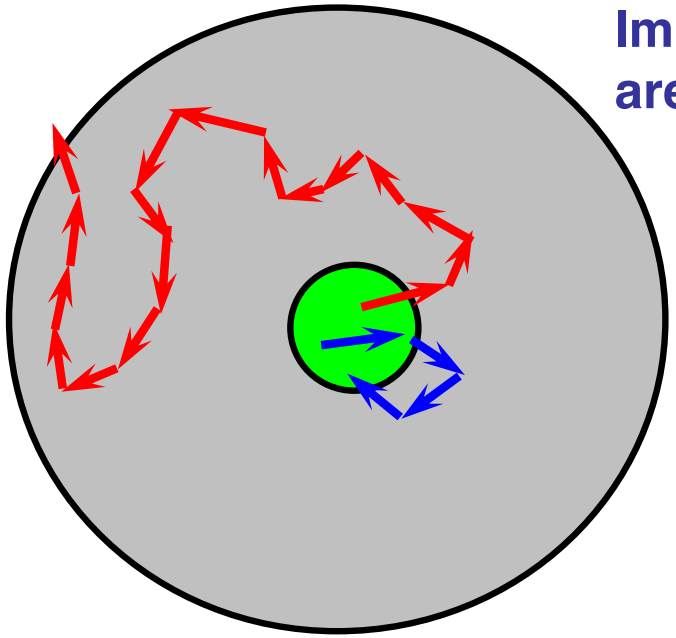
adjoint-weighted

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

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

Important neutrons are often short-lived

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



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

MCNP Implementation

- **Point-kinetics parameter calculations activated by the criticality calculation options (kopts) card:**

  kopts blocksize={N} kinetics={yes/no} precursor={yes/no}

- **Keywords:**
  - **blocksize: number of generations in each block (default = 10)**
  - **kinetics: whether to compute point-kinetics parameters (default = no)**
  - **precursor: whether to compute detailed delayed precursor information (default = no)**

Exercise 1: Godiva

- Copy g1.txt to gpk.txt.

- Add the kopts card to compute the point kinetics parameters:
  - The size of the blocks should be 5.
  - Calculate both the kinetics parameters and detailed precursor information

- Run the problem with:
  - 10000 histories per cycle
  - 10 inactive cycles, 250 active cycles

- Examine the output file

Godiva Results

- **Search for the following block in the output file:**

```
 the estimated adjoint-weighted point reactor kinetics parameters are:

                          estimate       std. dev.
              gen. time          5.66454        0.03265     (nsec)
            rossi-alpha    -1.17596E-03    5.26661E-05     (/nsec)
               beta-eff          0.00666        0.00030
```

- **Compare this with the prompt removal time:**

```
the final combined (col/abs/tl) prompt removal lifetime = 6.2535E-09 seconds
```

Exercise 2: Flattop

- Copy flatpk.txt from the SOLUTIONS directory.
- Inspect the input file geometry and KOPTS and KCODE cards.
- Run the problem.
- Examine the output file.

Flattop Results

- **Point kinetics parameters:**

```
 the estimated adjoint-weighted point reactor kinetics parameters are:

                          estimate      std. dev.

            gen. time        17.51114        0.16524    (nsec)
         rossi-alpha    -4.15749E-04    1.93956E-05    (/nsec)
            beta-eff         0.00728        0.00033
```

- **Prompt removal time:**

```
the final combined (col/abs/tl) prompt removal lifetime = 6.7080E-08 seconds
```

Observations

- **The prompt removal time does not match the neutron generation time**
  - **For Godiva, it is about 10% different.**
  - **For Flattop, with its reflector, it is almost a factor of 4.**

- **The effective delayed neutron fraction is also much higher in Flattop**
  - **Probably because of fast fission in U-238, which has a high delayed neutron fraction.**

Nuclear Data Sensitivities

Motivation

- 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

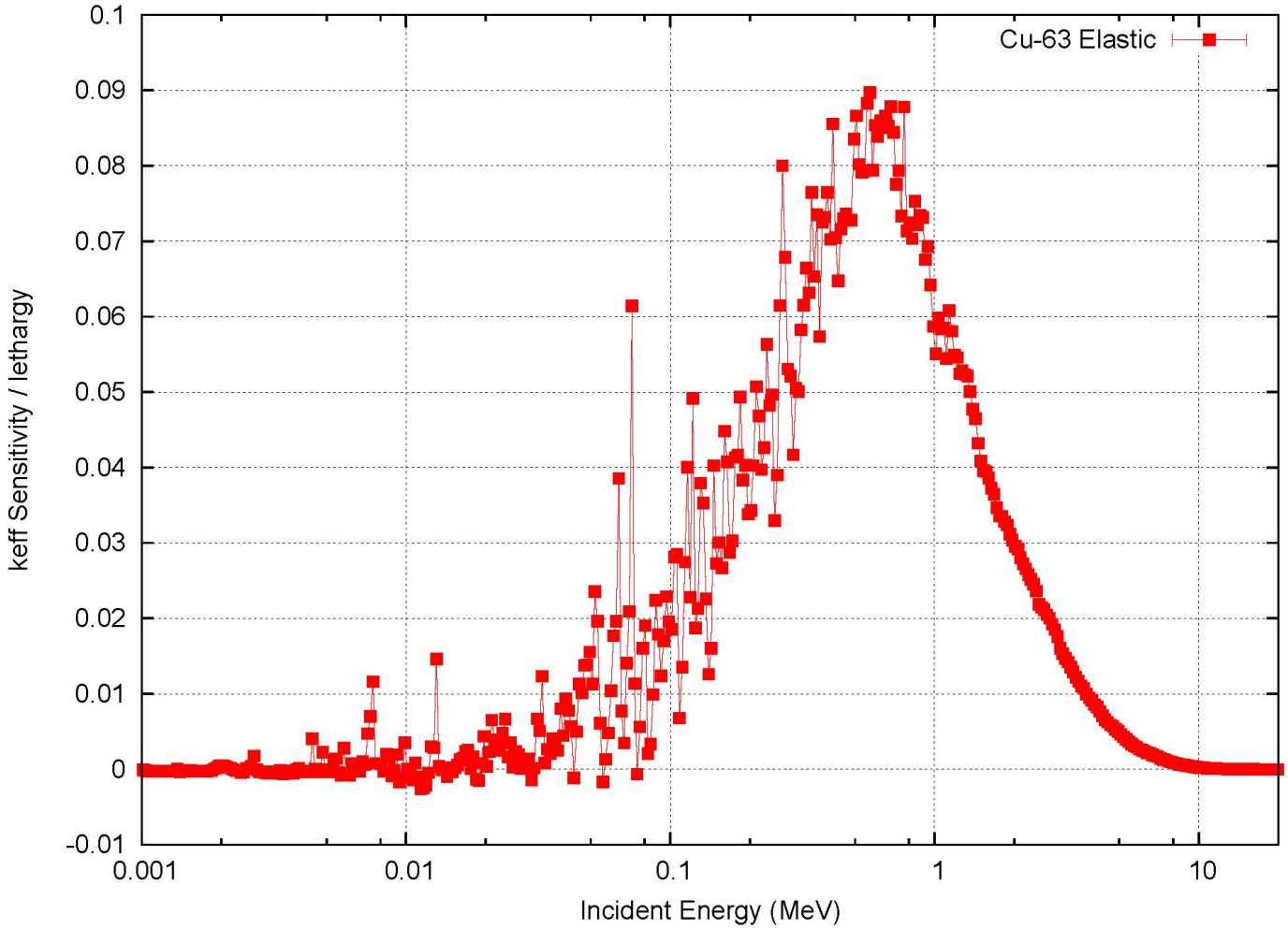
- This may be expressed using perturbation theory:

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

- 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.

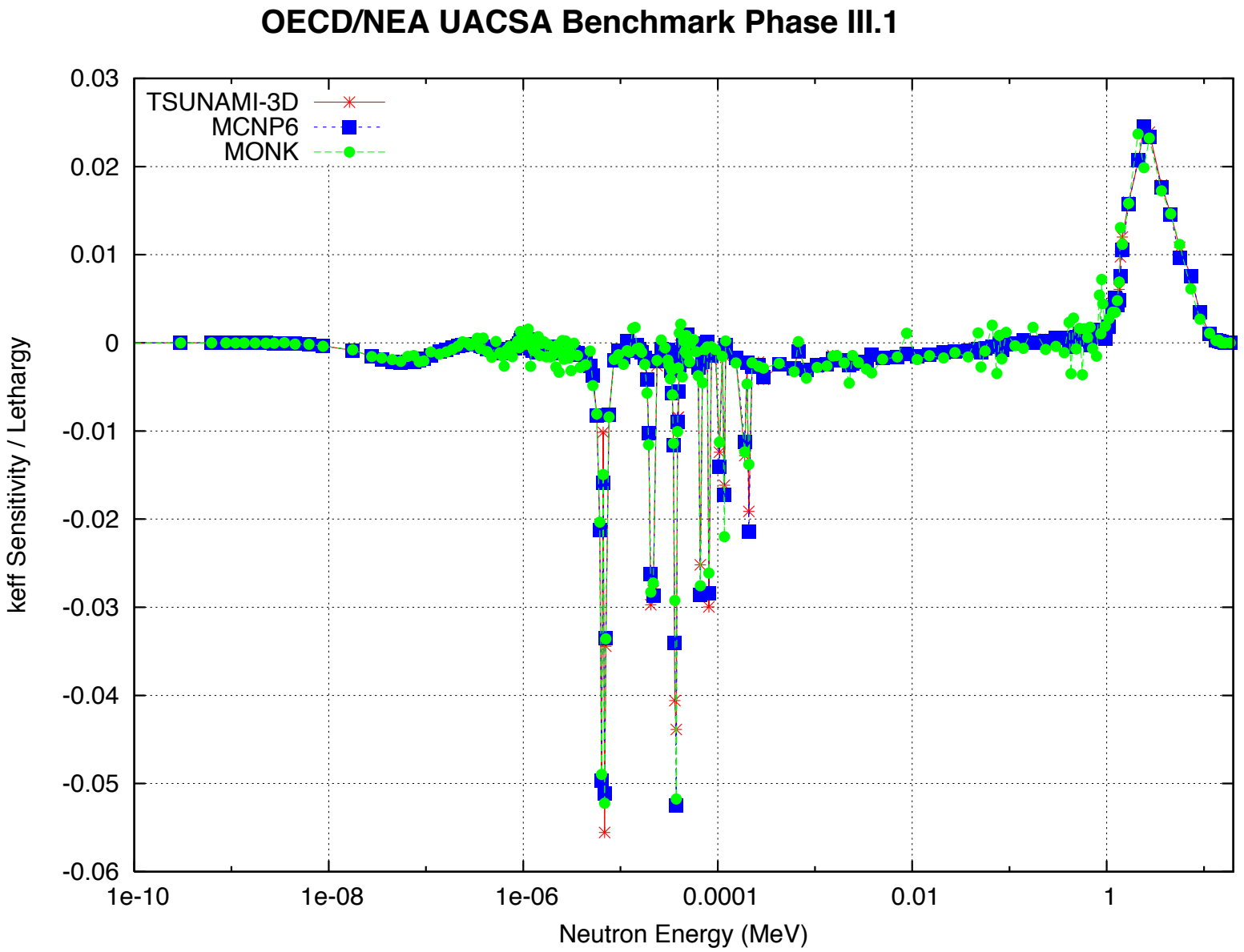
Example Sensitivity Coefficient Profile

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



Example Sensitivity Coefficient Profile

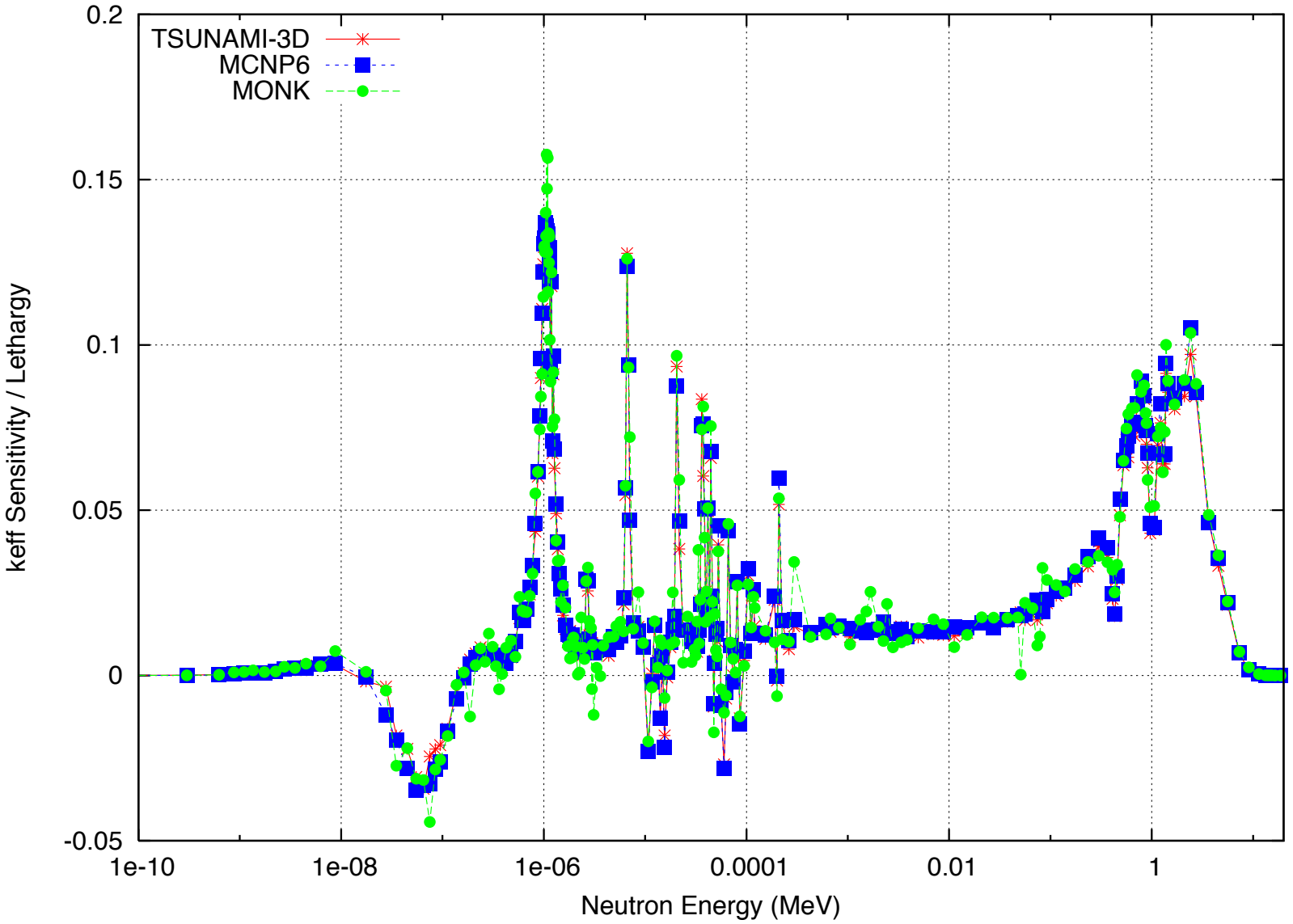
U-238: total cross-section sensitivity



Example Sensitivity Coefficient Profile

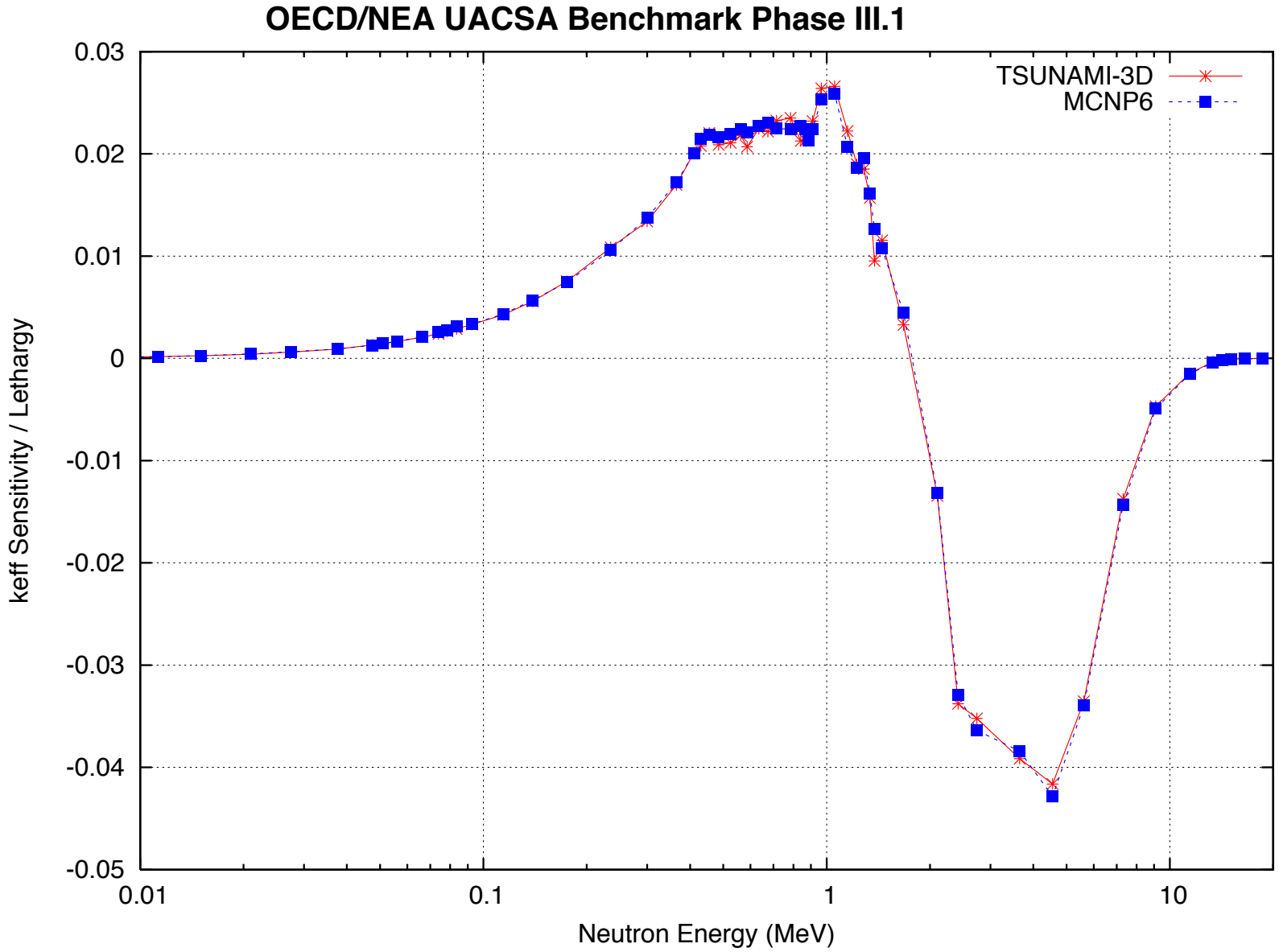
H-1: elastic scattering cross-section sensitivity

OECD/NEA UACSA Benchmark Phase III.1



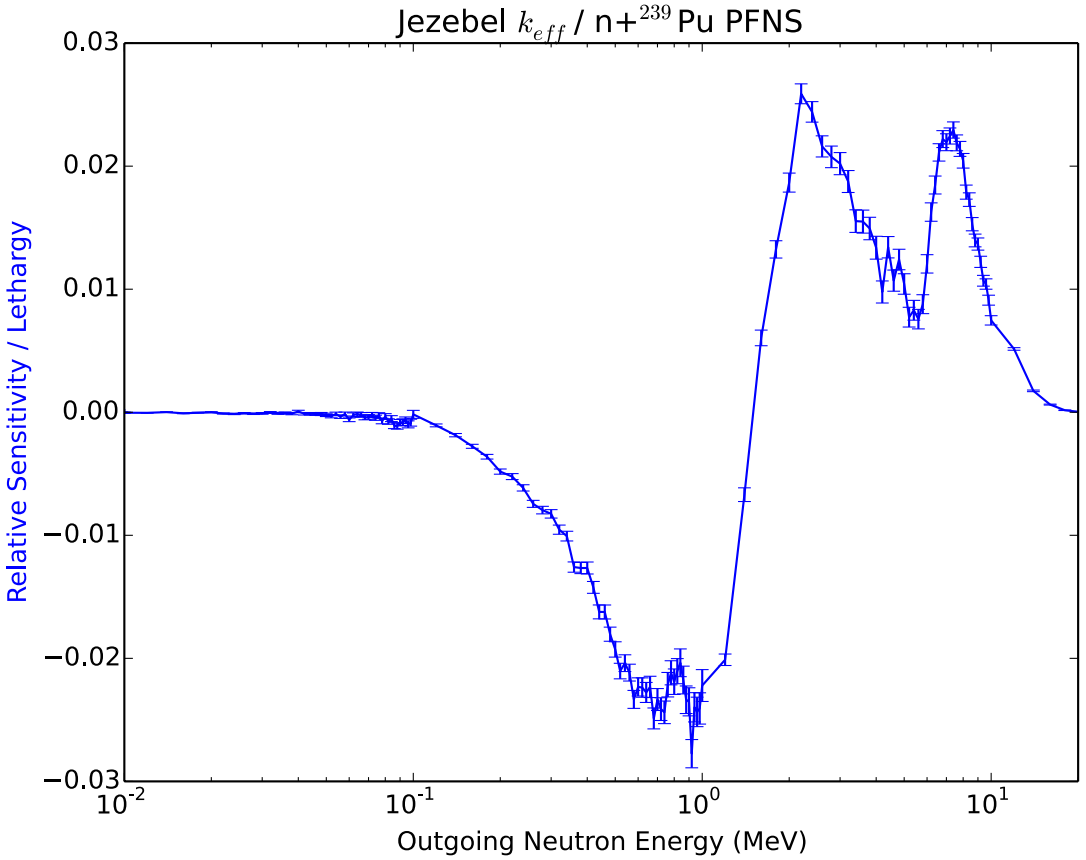
Example Sensitivity Coefficient Profile

Pu-239: fission chi(E) sensitivity

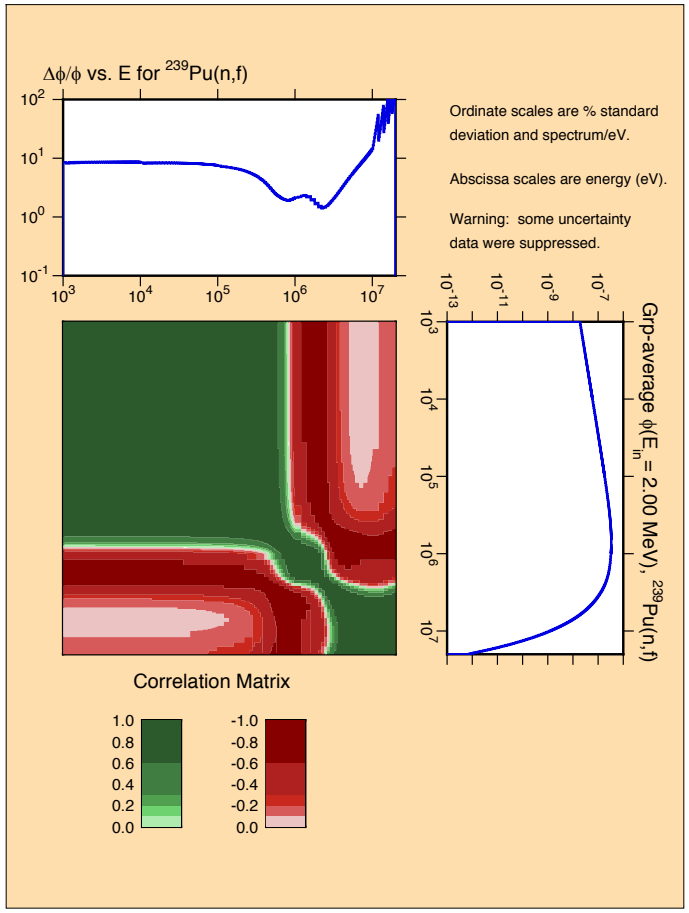


Example – Need for Sensitivities

Example using "sandwich rule" (linear error propagation), ^{239}Pu PFNS impact on k_{eff}



Nuclear Data Uncertainties



$$\sigma_k^2 = \vec{S}\,\mathbf{C}_X\,\vec{S}^T, \qquad \frac{\sigma_k}{k} \cong 0.160\%$$

← Uncertainty in k_{eff} due to ^{239}Pu PFNS only!

MCNP Implementation

- 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

MCNP Implementation

- **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*.**

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

- **See the list of reaction identifiers for tally multipliers**

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
```

Exercise 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

- Run the problem and analyze output.

Exercise 1: Keff Sensitivity Cards

```
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
```

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

Exercise 1: Discussion

- **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**

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
- Run the problem and analyze output.

Exercise 2: Keff Sensitivity Cards

```
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
```

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

Exercise 2: Discussion

- 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.

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.

- Run the problem and analyze output.

Exercise 3: Keff Sensitivity Cards

```
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
```

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

Exercise 3: Discussion

- 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.

KSEN for Secondary Distributions

(OPTIONAL)

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.**

Constrained Chi Sensitivity Example

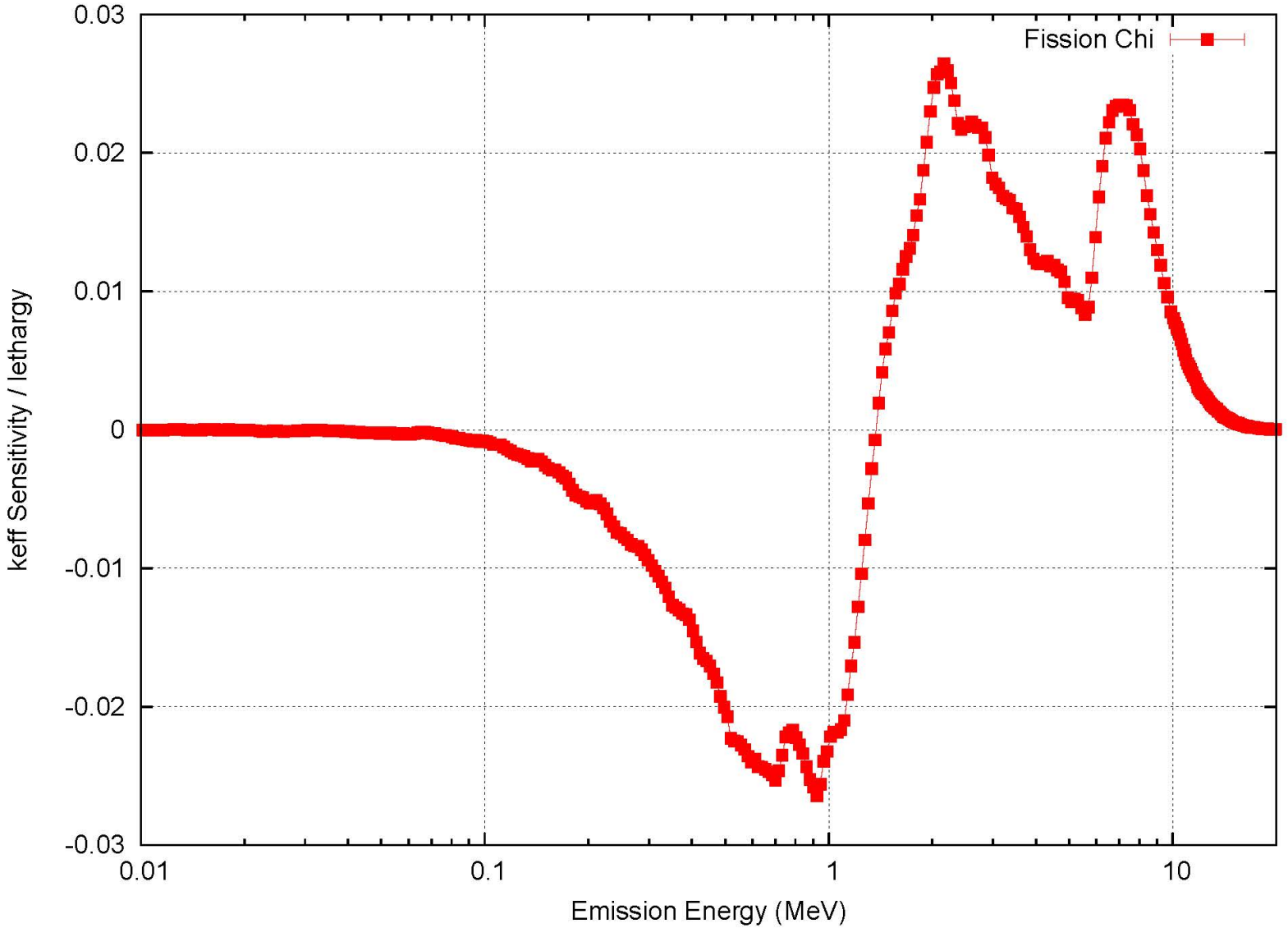
- **KSEN card of Pu-239 chi sensitivity:**

```
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):**



Physics & Nuclear Data

Nuclear data libraries

S(α,β) thermal neutron scattering data

Continuous energy vs. multigroup

Use of nuclear data in MCNP

Photon Transport

Plotting data with MCNP

<u>Optional</u>

Unresolved resonance treatment

Temperature-specific libraries & MAKXSF

Nuclear Data Libraries

Nuclear Interaction Data

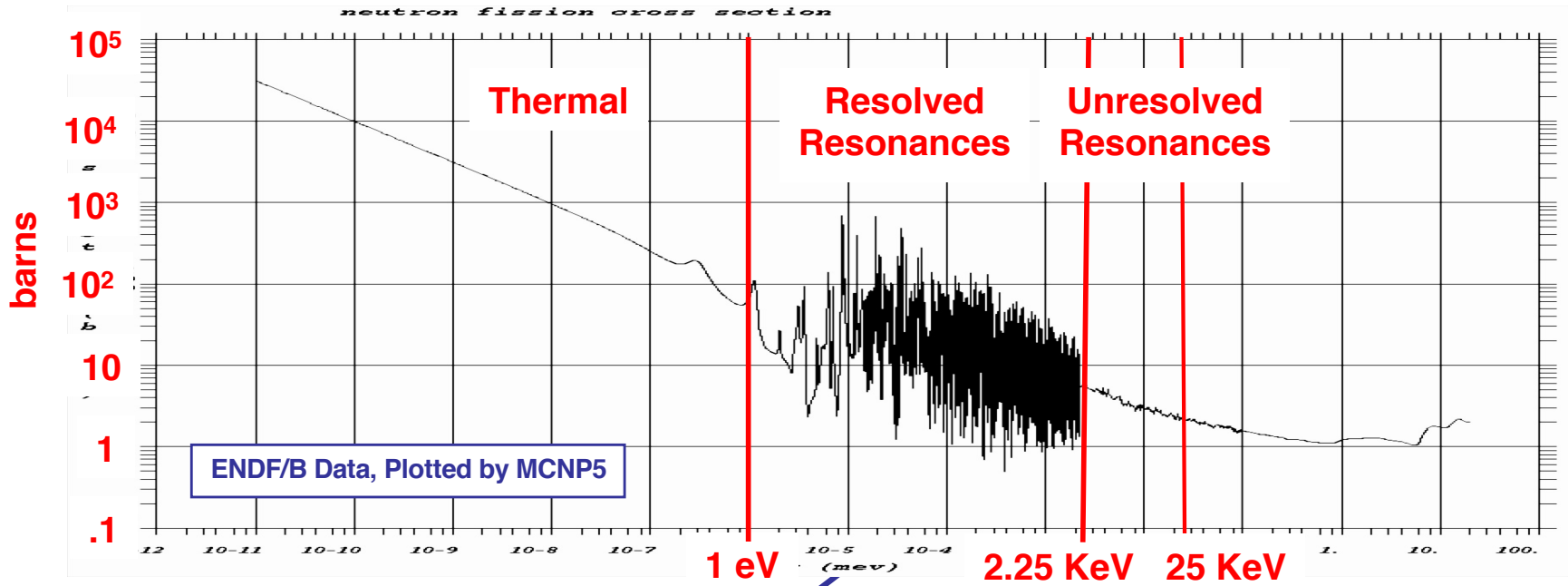
- "Nuclear" data involves interactions of incident particles with the nucleus.
  - Data libraries include cross-section and scattering data with interpolation laws, various parameters, etc., derived from both experiments and theory
  - Typically there are "ladders" of (E_J,σ_J) pairs, but many other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

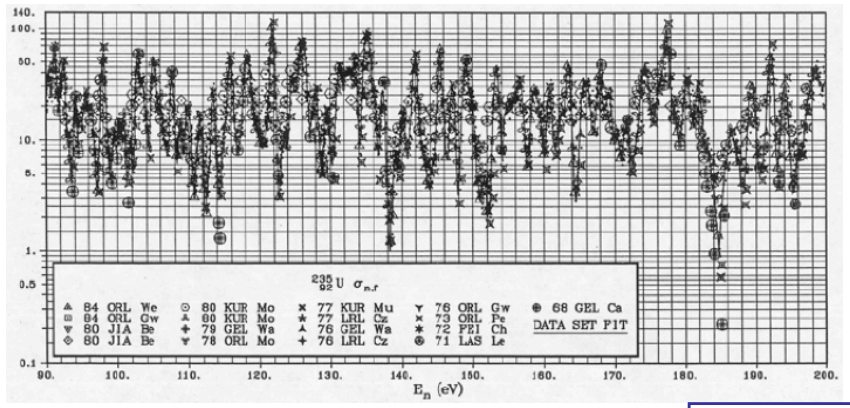
- MCNP will read
  - Neutron Data Files
  - Photonuclear Data Files
  - Proton Data Files

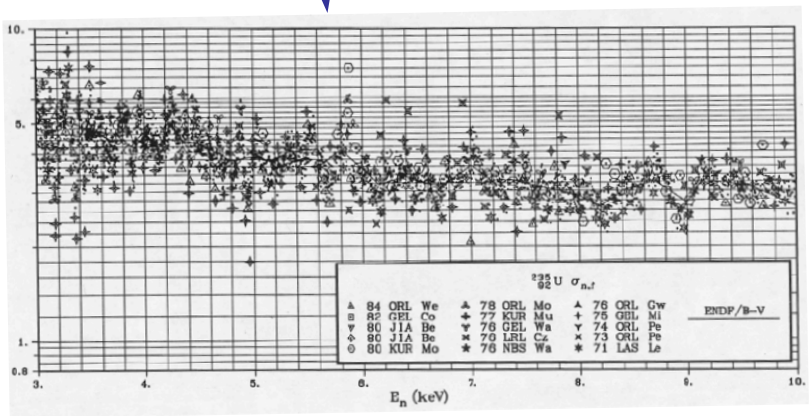
- Results obtained from a calculation depend upon both the code and the nuclear data it employs

- The "Mix and Match" capability extends nuclear data capability in MCNP6
  - Isotope Mixing: If an isotope is missing from the libraries, data from another isotope may be substituted
  - Energy Matching: If the particle energy is above the library energy limit, MCNP will switch to physics models to calculate interaction probabilities.

U^{235} Fission Cross-section



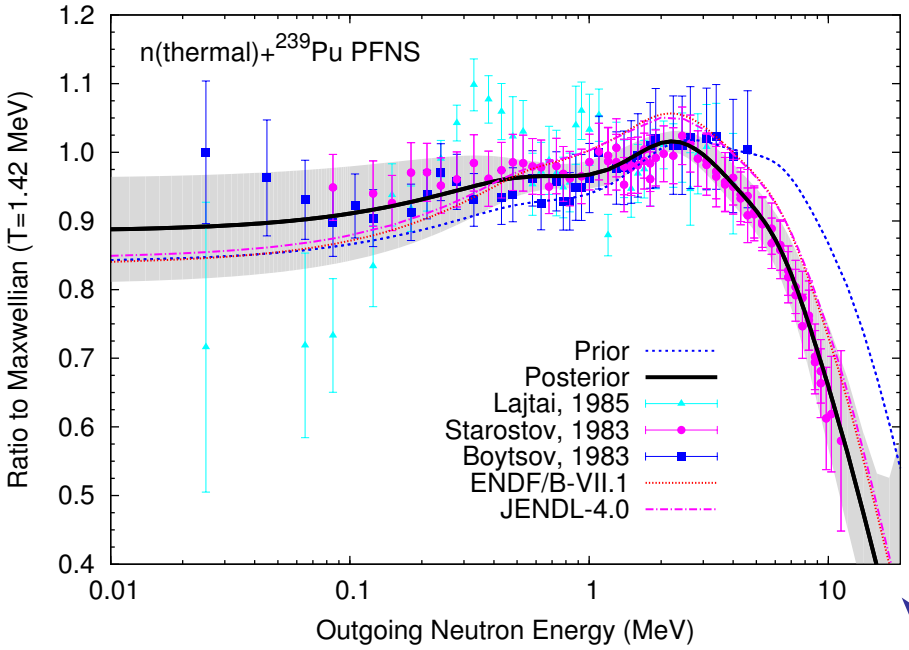


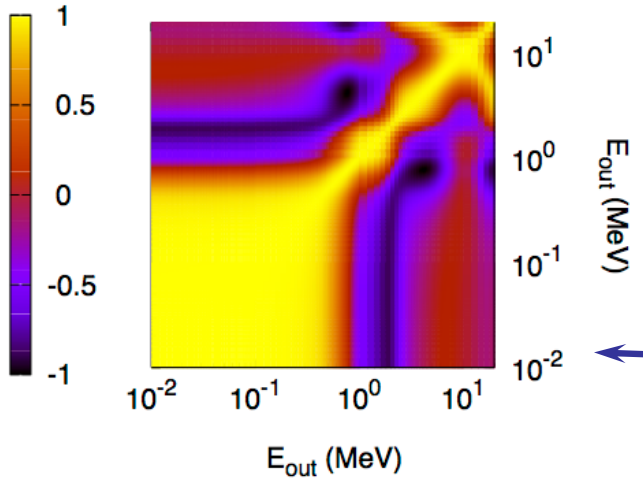


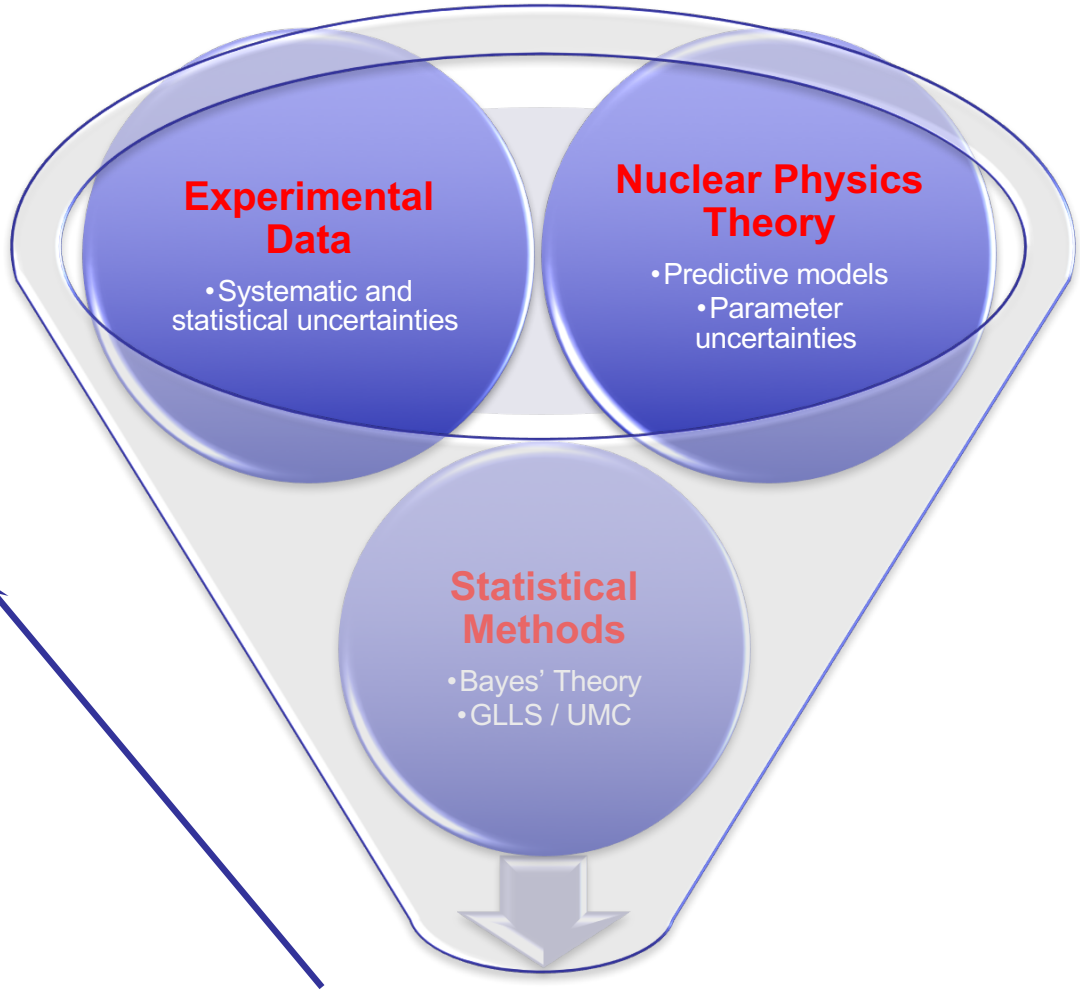
Experimental Data used by CSWEG

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
- **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

Data Libraries for MCNP

- **Data in the ENDF/B data libraries must be processed into formats that MCNP can use -- called ACE libraries (A Compact ENDF/B)**
  - **Type 1 ACE libraries - ASCII text**
  - **Type 2 ACE libraries - binary**
  - **MAKXSF code can convert & manipulate ACE libraries**
- **The NJOY code is used to process ENDF/B data & produce ACE libraries for MCNP**

  R. E. MacFarlane and D. W. Muir, "The NJOY Nuclear Data Processing System, Version 91," Los Alamos National Laboratory report LA-12740-M (October 1994)
- **Listing of Available ACE Data Tables, LA-UR-13-21822 on MCNP web & MCNP6 release has a listing of all the currently available ACE libraries & nuclides**
- **xsdir file provides MCNP link between data identifiers and contents of data library, or xsdir_mcnp6.1 for MCNP6.1**

Nuclear Data Identifiers in MCNP

- **General form of "ZAID" identifiers for ACE library cross-section data files is: ZZAAA.NNK, where**

ZZ = Atomic number

AAA = Mass number (000 = elemental)

NN = Library identifier

80-86 = ENDF/B-VII.1, 20-27 = ENDF/B-VII.1, thermal
70-74 = ENDF/B-VII.0, 10-17 = ENDF/B-VII.0, thermal
66 = ENDF/B-VI.6, 60 = ENDF/B-VI.2, 50 = ENDF/B-V

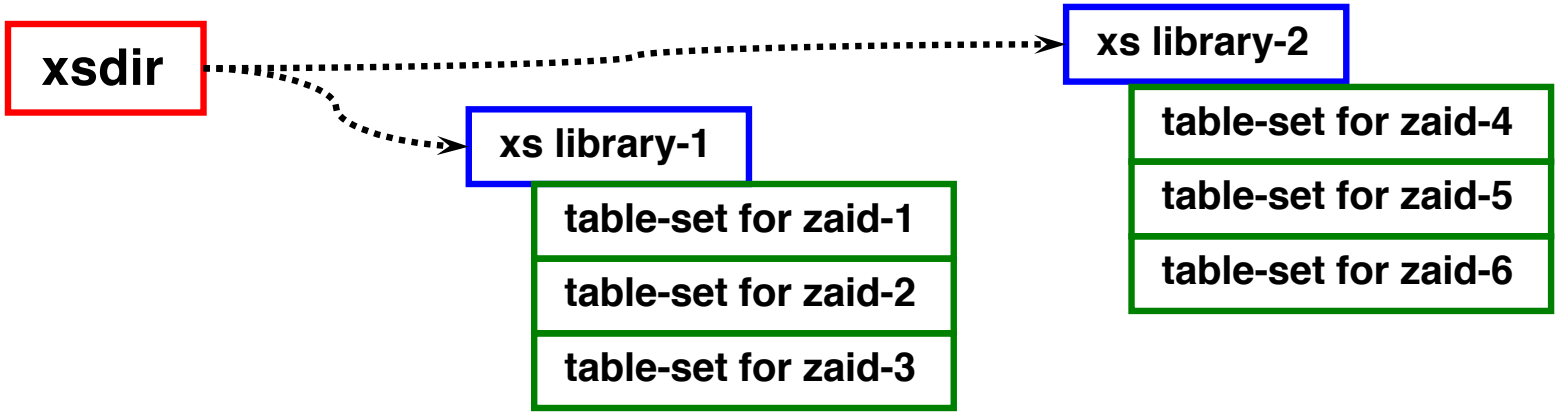
K = type of data:

c = continuous-energy neutron
t = thermal neutron scattering law S(α,β)
p = photons
m = multigroup neutron
… etc.

Examples:

1001.80c	continuous-energy data for hydrogen, ENDF/B-VII.1
92235.80c	continuous-energy data for ^{235}U, 293.6 K, ENDF/B-VII.1
92235.83c	continuous-energy data for ^{235}U, 1200 K, ENDF/B-VII.1

MCNP Data



- xsdir = table of contents for cross-section libraries
- Default xsdir file: mcnp5: `$DATAPATH/xsdir`
  mcnp6: `$DATAPATH/xsdir_mcnp6.1`
- See Listing of Available ACE Data Tables, LA-UR-13-21822 for a table of available nuclide datasets
- If a nuclide is specified as just "ZZAAA" (no library identifier), the first match found in the xsdir file is used

Structure of xsdir File

```
atomic weight ratios
   0001  1.000000   0001  1.000000
   1000  0.99931697 1001   0.99916733   1002   1.99679968   1003   2.99013997
                   1004   3.99320563   1005   4.99205575   1006   5.99301364
                   1007   6.99216250
   2000  3.96821760 2003   2.99012018   2004   3.96821897   2005   4.96916622
  . . . . .
118000  290.695815 118293 290.69581525
03/04/2013
directory
1001.80c 0.999167 xdata/endf71x/H/1001.710nc 0 1 4 17969 0 0 2.5301E-08
1001.81c 0.999167 xdata/endf71x/H/1001.711nc 0 1 4 17969 0 0 5.1704E-08
1001.82c 0.999167 xdata/endf71x/H/1001.712nc 0 1 4 17969 0 0 7.7556E-08
1001.83c 0.999167 xdata/endf71x/H/1001.713nc 0 1 4 17969 0 0 1.0341E-07
1001.84c 0.999167 xdata/endf71x/H/1001.714nc 0 1 4 17969 0 0 2.1543E-07
1001.85c 0.999167 xdata/endf71x/H/1001.715nc 0 1 4 17969 0 0 8.6174E-12
1001.86c 0.999167 xdata/endf71x/H/1001.716nc 0 1 4 17969 0 0 2.1543E-08
1002.80c 1.9968 xda
  . . . . .
92235.80c 233.0248 xdata/endf71x/U/92235.710nc 0 1 4 837481 0 0 2.5301E-08 +
          ptable
92235.81c 233.0248 xdata/endf71x/U/92235.711nc 0 1 4 694543 0 0 5.1704E-08 +
          ptable
  . . . . .
```

First match for 1001 (H-1) = default if .nnC not supplied

First match for 92235 (U-235) = default if .nnC not supplied

Neutron Libraries for MCNP

- **The MCNP data libraries are contained in the MCNP RSICC package.**
- **Continuous-energy neutron libraries for MCNP:**
  - **endf71x/*/*** **ENDF/B-VII.1**
  - **endf70 a-k** **ENDF/B-VII.0**

Older libraries, retained for historical reasons:

- **t16_2003** **preliminary ENDF/B-VII, for a few isotopes**
- **acti** **ENDF/B-VI.8**
- **endf66** **ENDF/B-VI.6**
- **endf60** **ENDF/B-VI.2**
- **endf62mt** **A multi-temperature Neutron Data Library**
- **ures** **ENDF/B-VI Neutron Library with Probability Tables**
- **endf6dn** **ENDF/B-VI Neutron Library with Delayed Neutron Tables**
- **newxs** **LANL based evaluations**
- **rmccs** **ENDF/B-V and LANL based evaluations**
- **rmccsa** **ENDF/B-V and LANL**
- **endf5p** **ENDF/B-V based**
- **endf5u** **ENDF/B-V based**
- **misc5xs** **Contains a number of previously released smaller libraries**
- **kidman** **Fission product evaluations**
- **100xs** **LANL based**
- **endl92** **1992 ENDL library from LLNL**
- **endl85** **1985 ENDL library**
- **endf5mt** **Multi-temperature data previously released as eprixs and u600k**

ZAIDs for MCNP Neutron Libraries - ENDF/B-V

```
 1001  1002  1003
 2003  2004
 3006  3007
 4009
 5010  5011
 6000  6012
 7014  7015
 8016
 9019
11023
12000
13027
14000
15031
16032
17000
18000
19000
20000
22000
23000
24000
25055
26000
27059
28000
29000
31000
35079 35081
36078 36080 36082 36083
      36084 36086
37085 37087
39089
40000
40093
41093
42000 42095
43099
44101 44103
45103 45105
46105 46108
47000 47107 47109
48000
53127 53135
54131 54135
55133 55135
56138
59141
60143 60145 60147 60148
61147 61148 61149
62147 62149 62150 62151
      62152
63151 63152 63153 63154
      63155
64152 64154 64155 64156
      64157 64158 64160
67165
69169
72000
73181
74000 74182 74183 74184
      74186
75185 75187
77000
79197
82000
83209
90232
91233
92233 92234 92235 92236
      92237 92238
93237
94238 94239 94240 94241
      94242
95241 95242 95243
96242 96244
```

Red = element

ZAIDs for MCNP Neutron Libraries - ENDF/B-VI

```
 1001  1002  1003
 2003  2004
 3006  3007
 4009
 5010  5011
 6000
 7014  7015
 8016  8017
 9019
11023
12000
13027
14000 14028 14029 14030
15031
16000 16032
17000 17035 17037
19000
20000
21045
22000
23000
24050 24052 24053 24054
25055
26054 26056 26057 26058
27059 28058 28060 28061
      28062 28064
29063 29065
31000
36078 36080 36082 36083
      36084 36086
37085 37087
39089
40000 40090 40091 40092
      40094 40096
41093
42000
43099
45103
46102 46104 46105 46106
      46108 46110
47107 47109
48106 48108 48110 48111
      48112 48113 48114
      48116
49000
53127 53129
54124 54126 54128 54129
      54130 54131 54132
      54134 54136
55133 55134 55135 55136
      55137
56138
62147 62149
63151 63152 63153 63154
      63155
64152 64154 64155 64156
      64157 64158 64160
67165
71175 71176
72000 72174 72176 72177
      72178 72179 72180
73181 73182
74182 74183 74184 74186
75185 75187
77191 77193
79197
82206 82207 82208
83209
90230 90232
91231 91233
92232 92233 92234 92235
      92236 92237 92238
      92239 92240 92241
93237 93239
94236 94237 94238 94239
      94240 94241 94242
      94243 94244
95241 95242 95243
96241 96242 96243 96244
      96245 96246 96247
      96248
97249
98249 98250 98251 98252
```

Red = element

ZAIDs for MCNP Neutron Libraries - ENDF/B-VII.0

1001 1002 1003
2003 2004
3006 3007
4009
5010 5011
6000
7014 7015
8016 8017
9019
11022 11023
12024 12025 12026
13027 14028 14029 14030
15031
16032 16033 16034 16036
17035 17037
18036 18038 18040
19039 19040 19041
20040 20042 20043 20044
20046 20048
21045 22046 22047 22048
22049 22050
23000
24050 24052 24053 24054
25055 26054 26056 26057
26058
27058 27458 27059
28058 28059 28060 28061
28062 28064
29063 29065
30000
31069 31071
32070 32072 32073 32074
32076
33074 33075
34074 34076 34077 34078
34079 34080 34082
35079 35081
36078 36080 36082 36083
36084 36085 36086

37085 37086 37087
38084 38086 38087 38088
38089 38090
39089 39090 39091
40090 40091 40092 40093
40094 40095 40096
41093 41094 41095
42092 42094 42095 42096
42097 42098 42099
42100
43099
44096 44098 44099 44100
44101 44102 44103
44104 44105 44106
45103 45105
46102 46104 46105 46106
46107 46108 46110
47107 47109 47510 47111
48106 48108 48110 48111
48112 48113 48114
48515 48116
49113 49115
50112 50113 50114 50115
50116 50117 50118
50119 50120 50122
50123 50124 50125
50126
51121 51123 51124 51125
51126
52120 52122 52123 52124
52125 52126 52527
52128 52529 52130
52132
53127 53129 53130 53131
53135
54123 54124 54126 54128
54129 54130 54131
54132 54133 54134
54135 54136

55133 55134 55135 55136
55137
56130 56132 56133 56134
56135 56136 56137
56138 56140
57138 57139 57140
58136 58138 58139 58140
58141 58142 58143
58144
59141 59142 59143
60142 60143 60144 60145
60146 60147 60148
60150
61147 61148 61548 61149
61151
62144 62147 62148 62149
62150 62151 62152
62153 62154
63151 63152 63153 63154
63155 63156 63157
64152 64153 64154 64155
64156 64157 64158
64160
65159 65160
66156 66158 66160 66161
66162 66163 66164
67165 67566
68162 68164 68166 68167
68168 68170
71175 71176
72174 72176 72177 72178
72179 72180
73181 73182 74182 74183
74184 74186
75185 75187
77191 77193
79197
80196 80198 80199 80200
80201 80202 80204

82204 82206 82207 82208
83209
88223 88224 88225 88226
89225 89226 89227
90227 90228 90229 90230
90232 90233 90234
91231 91232 91233
92232 92233 92234 92235
92236 92237 92238
92239 92240 92241
93235 93236 93237 93238
93239
94236 94237 94238 94239
94240 94241 94242
94243 94244 94246
95241 95642 95242 95243
95244 95644
96241 96242 96243 96244
96245 96246 96247
96248 96249 96250
97249 97250
98249 98250 98251 98252
98254
99254 99255
100255

Red = element

ZAIDs for MCNP Neutron Libraries - ENDF/B-VII.1

1001 1002 1003
2003 2004
3006
4007 4009
5010 5011
6000
7014 7015
8016 8017
9019
11022 11023
12024 12025 12026
13027
14028 14029 14030
15031
16032 16033 16034 16036
17035 17037
18036 18038 18040
19039 19040 19041
20040 20042 20043 20044
20046 20048
21045
22046 22047 22048 22049
22050
23050 23051
24050 24052 24053 24054
25055
26054 26056 26057 26058
27058 27059 27458
28058 28059 28060 28061
28062 28064
29063 29065
30064 30065 30066 30067
30068 3007 30070
31069 31071
32070 32072 32073 32074
32076
33074 33075
34074 34076 34077 34078
34079 34080 34082
35079 35081

36078 36080 36082 36083
36084 36085 36086
37085 37086 37087
38084 38086 38087 38088
38089 38090
39089 39090 39091
40090 40091 40092 40093
40094 40095 40096
41093 41094 41095
42092 42094 42095 42096
42097 42098 42099
42100
43099
44096 44098 44099 44100
44101 44102 44103
44104 44105 44106
45103 45105
46102 46104 46105 46106
46107 46108 46110
47107 47109 47111 47610
48106 48108 48110 48111
48112 48113 48114
48116 48515
49113 49115
50112 50113 50114 50115
50116 50117 50118
50119 50120 50122
50123 50124 50125
50126
51121 51123 51124 51125
51126
52120 52122 52123 52124
52125 52126 52128
52130 52132 52529
52627
53127 53129 53130 53131
53135
54123 54124 54126 54128
54129 54130 54131
54132 54133 54134

54135 54136
55133 55134 55135 55136
55137
56130 56132 56133 56134
56135 56136 56137
56138 56140
57138 57139 57140
58136 58138 58139 58140
58141 58142 58143
58144
59141 59142 59143
60142 60143 60144 60145
60146 60147 60148
60150
61147 61148 61149 61151
61648 62144 62147
62148 62149 62150
62151 62152 62153
62154
63151 63152 63153 63154
63155 63156 63157
64152 64153 64154 64155
64156 64157 64158
64160
65159 65160
66156 66158 66160 66161
66162 66163 66164
67165 67566
68162 68164 68166 68167
68168 68170
69168 69169 69170
71175 71176
72174 72176 72177 72178
72179 72180
73180 73181 73182
74180 74182 74183 74184
74186
75185 75187
77191 77193
79197

80196 80198 80199 80200
80201 80202 80204
81203 81205
82204 82206 82207 82208
83209
88223 88224 88225 88226
89225 89226 89227
90227 90228 90229 90230
90231 90232 90233
90234
91229 91230 91231 91232
91233
92230 92231 92232 92233
92234 92235 92236
92237 92238 92239
92240 92241
93234 93235 93236 93237
93238 93239
94236 94237 94238 94239
94240 94241 94242
94243 94244 94246
95240 95241 95242 95243
95244 95642 95644
96240 96241 96242 96243
96244 96245 96246
96247 96248 96249
96250
97245 97246 97247 97248
97249 97250
98246 98248 98249 98250
98251 98252 98253
98254
99251 99252 99253 99254
99255 99754
100255

Red = element

S(α,β) Thermal Neutron Scattering Data

Thermal Scattering – S(α,β) Data

- **At low energies (E < 9 eV), neutron scattering interactions are influenced by chemical binding, temperature, molecular effects, …**
  - Important for light nuclei (moderators)
  - MCNP libraries include thermal scattering laws, **S(α,β) libraries**, for water, heavy water, polyethylene, methane, benzene, graphite, beryllium, zirc-hydride, etc.
  - Include thermal scattering law(s) for every moderator nuclide in any problem where neutrons reach energies of 9 eV or less, using an **MTn** card
- **SAB2002**
  - ENDF/B-VI-based S(α,β) data, released in 2002
  - Data for 15 combinations of nuclides and materials
  - Typical temperature ranges are from 294 K to 1200 K, in increments of 200
  - Data typically tabulated at 16 angles and 64 energies for each temperature
  - Data are provided at ~ 20 K for a limited number of nuclides
- **ENDF70SAB (discrete), ENDF71SaB (continuous)**
  - ENDF/B-VII - based S(α,β) data, ca. 2008
  - Many more nuclide - material combinations:

    al27, be, be-o, benz, dortho, dpara, fe56, grph, h-zr, hortho, hpara, hwtr, lmeth, lwtr, o-be, o2-u, poly, smeth, u-o2, zr-h
  - Many more temperatures, data every 50 K or 100 K
  - See **Listing of Available ACE Data Tables, LA-UR-13-21822**

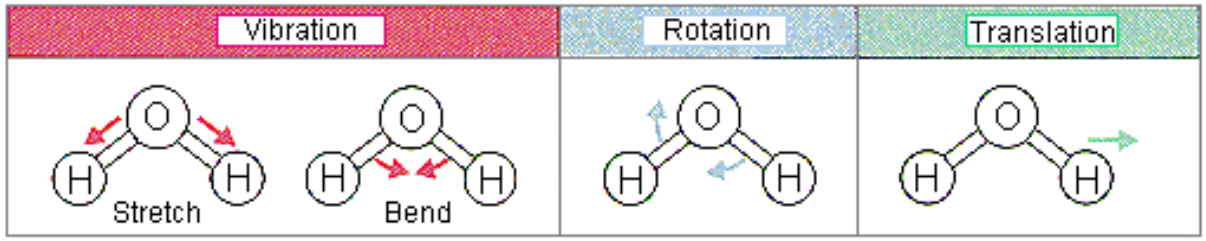
Neutron S(α,β) Thermal Scattering Libraries

ENDF/B-V	ENDF/B-VI	ENDF/B-VII.0	ENDF/B-VII.1		
tmccs	sab2002	endf70sab	ENDF71SaB		
discrete	discrete	discrete	continuous		
be	be	al27	al27		
benz	benz	be	be	be-o	be/o
beo	beo	be/o	benz		
grph	dortho	benz	dortho		
h/zr	dpara	dortho	dpara		
hwtr	grph	dpara	fe56		
lwtr	h/zr	fe56	grph		
poly	hortho	grph	h-zr	h/zr	
zr/h	hpara	h/zr	hortho		
	hwtr	hortho	hpara		
	lmeth	hpara	hwtr		
	lwtr	hwtr	lmeth		
	poly	lmeth	lwtr		
	smeth	lwtr	o-be	o/be	
	zr/h	o/be	o2-u	o2/u	
		o2/u	poly		
		poly	sio2		
		smeth	smeth		
		u/o2	u-o2	u/o2	
		zr/h	zr-h	zr/h	

Thermal Neutron Scattering

- **Moderator materials contain light isotopes (H, D, He, Be, Li, C)**
  - **Water, heavy water, poly, concrete, etc.**
  - **Fast neutrons colliding with moderator lose lots of energy**
  - **Systems with moderator material:**
    - Large thermal neutron flux
    - Fission cross-sections are very large at thermal energies
    - Significant fraction of fissions caused by thermal neutrons (maybe all!)

- **Thermal neutron physics** 1×10^{-5} **eV < E < 9 eV**
  - **Neutron energy comparable to chemical binding effects, gives rise to incoherent inelastic scatter**



  - **Neutron wavelength comparable to atomic spacing**
    - In solids, may need **coherent elastic scatter** (Bragg) from crystals
    - In liquids & gases, may need **incoherent elastic scatter**

S(α,β) Thermal Neutron Scattering Data

- **S(α,β) data is used to model the physics for**
  - **Inelastic scatter (chemical binding, temperature, etc.)**
  - **Elastic scatter for some solids & liquids**
- **S(α,β) data is contained in special ACE files for MCNP**

α and β are dimensionless quantities representing:

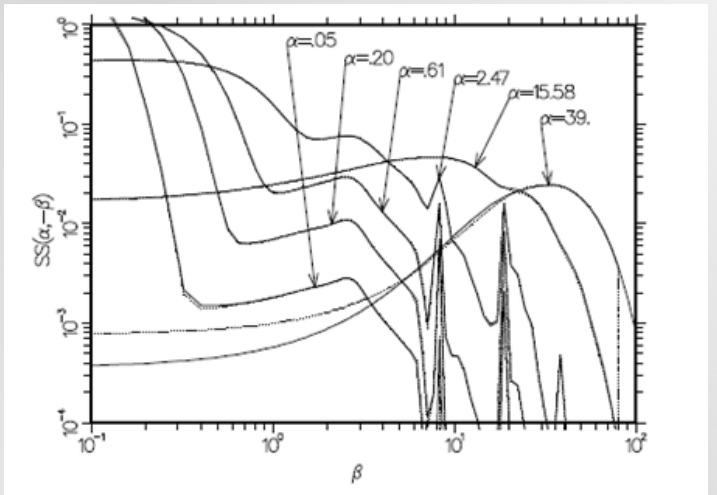
α: momentum transfer

$$\alpha = \frac{E + E' - 2\mu\sqrt{EE'}}{A_0 kT}$$

β: energy transfer

$$\beta = \frac{E' - E}{kT}$$

$$\sigma(E \to E', \mu, T) = \frac{\sigma_b}{2kT}\sqrt{\frac{E'}{E}}e^{-\beta/2}S(\alpha,\beta,T)$$



where:

E, E': pre- and post-collision energy
μ: cosine of the scattering angle
σ_b: bound atom scattering cross section
k: Boltzmann constant
T: temperature
S(α,β,T): symmetric form of the scattering law

S(α,β) Thermal Neutron Scattering Data

- **When to NOT use S(α,β) data:**
  - **Fast & intermediate systems, % thermal fissions small (< 10% ?)**
  - **Whenever no significant amount of moderator material**
    - Very thin coatings, very thin reflectors, paint, varnish, trace impurities
  - **Heavy isotopes - U, Zr, Fe, Al (anything heavier than O^{16})**

- **When to use S(α,β) data:**
  - **Thermal systems, significant % fissions from thermal neutrons**
  - **Solutions, sizable reflectors, concrete, hands, ….**
  - **Suggested:**
    - light water: **lwtr**
    - heavy water: **hwtr**
    - polyethylene: **poly**
    - concrete: **lwtr** (for H in concrete)
    - zirc-hydride: **h-zr**
    - oil **benz**
    - Be metal: **be** (for thermal systems)
    - Be oxide: **be-o** (for thermal systems)
    - Graphite: **grph**

S(α,β) Thermal Neutron Scattering Data

Things to consider:

- **Always used for thermal systems:** lwtr, hwtr, grph, poly, h-zr
- **Some S(α,β) datasets are only rarely used:** be-o, be, sio2, benz
- **Some S(α,β) datasets are almost never used:** u-o2, o2-u, zr-h, o-be
- **Some S(α,β) datasets were developed for specific research & experimental use (eg, ultra-cold neutron scatter experiments):** hortho, dortho, hpara, dpara, lmeth, smeth, al27, fe56
- **Cement:** $2\ Ca_3\ Si\ O_5 + 7\ H_2O \rightarrow 3(CaO)\cdot 2(SiO_2)\cdot 4(H_2O)(gel) + 3Ca(OH)_2$
  Usually just use lwtr (for H)

S(α,β) - Examples

- **Reactor fuel pin, 3.1% enriched UO2, with clad & water**
  - using lwtr (for H in water) k = 1.44853 +- 0.00005
  - using lwtr + o2-u (for O in UO2) k = 1.44853 +- 0.00005

  Can ignore S(a,b) for O, must include for H

- **pu-met-fast-018-001**
  - using S(a,b) for be: k = 0.99944 +- 0.00005
  - no S(a,b) k = 0.99942 +- 0.00005

  For fast spectrum systems, S(a,b) makes no difference

- **pu-comp-mixed-001-001**
  - using S(a,b) for lwtr, sio2, fe56: k = 1.02464 +- 0.00008
  - using S(a,b) for lwtr, sio2 only: k = 1.02463 +- 0.00008
  - using S(a,b) for lwtr only: k = 1.02458 +- 0.00008

- **pu-met-fast-041-001**
  - not using S(a,b) k = 1.00573 +- 0.00007
  - using S(a,b) for lwtr k = 1.00582 +- 0.00005

  0 % thermal fissions .....

Continuous Energy
vs.
Multigroup

Continuous-energy vs Multigroup

Continuous-energy (c)

- Cross-sections tabulated at "sufficient number" of energies.
- Secondary distributions (energy,angle) represented (almost) to same fidelity.
- Each reaction is represented separately.
- Application independent (almost)
- Allow transport codes to utilize nuclear data in as much detail as required.

Multigroup (m)

- Cross-sections averaged into groups.
- Energy distributions represented by P0 scattering matrix.
- Angular distributions represented by PN scattering matrices. Can lead to difficulties with negative xsecs.
- All reactions (except fission) combined.
- Can be used with deterministic transport codes.
- Compact
- Can give accurate results to problems for which you know the answer.

ZAIDs for MCNP Neutron Libraries – MGXSNP

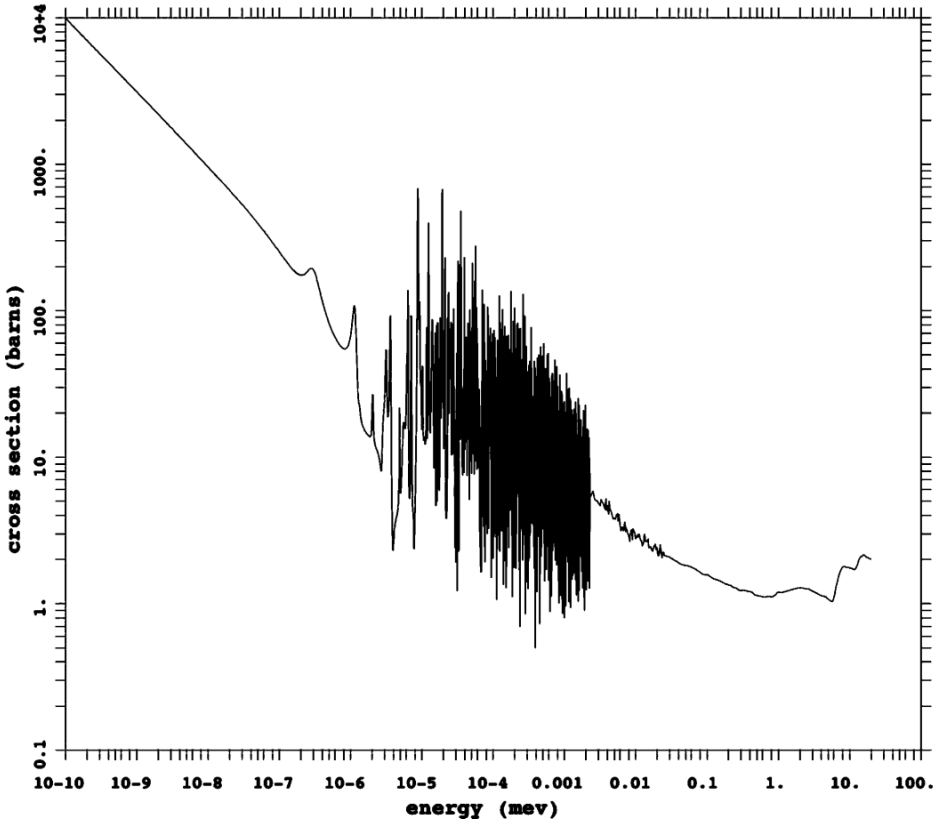
```
 1001   1002   1003
 2003   2004
 3006   3007
 4007   4009
 5010   5011
 6000   6012
 7014   7015
 8016
 9019
11023
12000
13027
14000
15031
16032
17000
18000
19000
20000
22000
23000
24000
25055
26000
27059
28000
29000
31000  33075
36078  36080  36082  36083
       36084  36086
40000
41093
42000
45103  45117
46119
47000  47107  47109
48000
50120  50998  50999
54000
56138
63000  63151  63153
64000
67165
73181
74000  74182  74183  74184
       74186
75185  75187
78000
79197
82000
83209
90232
91233
92233  92234  92235  92236
       92237  92238  92239
93237
94238  94239  94240  94241
       94242
95241  95242  95243
96242  96244
```

MGXSNP

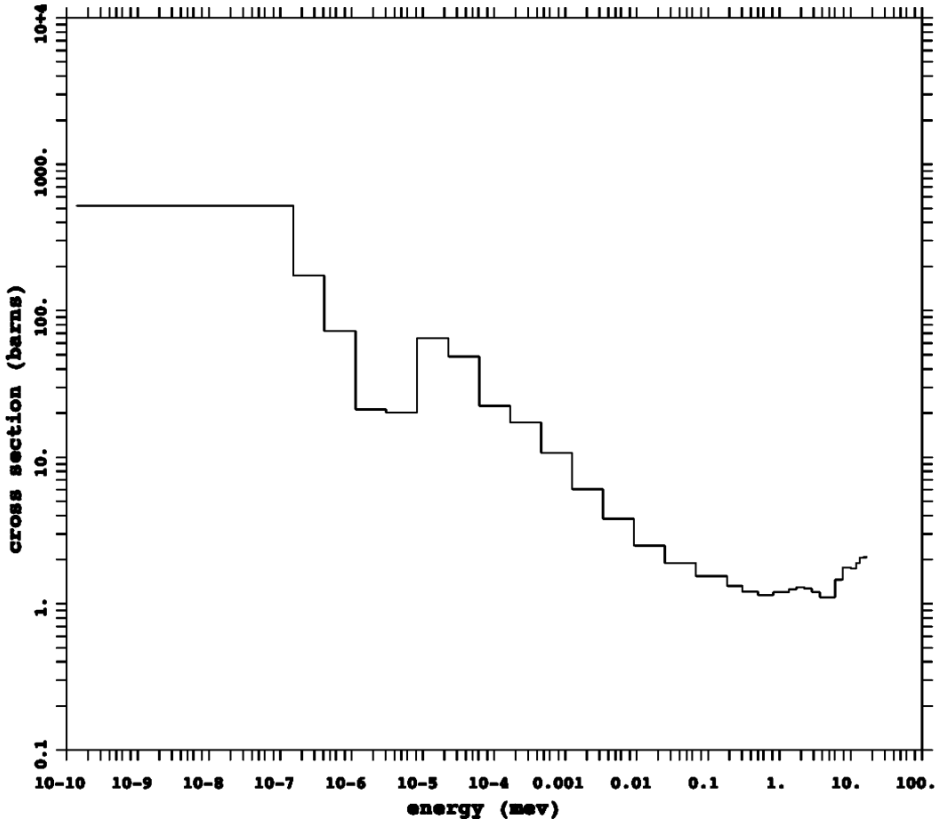
- **Very old multigroup cross-section library for MCNP – should generally not be used for any serious work**
- **MGXSNP is comprised of 30-group neutron and 12-group photon data primarily based on ENDF/B-V for 95 nuclides.**
- **The MCNP-compatible multigroup data library was produced from the original Sn multigroup libraries MENDF5 and MENDF5G using the code CRSRD in April 1987.**
- **All cross-sections are for room temperature, 300K, with P0 through P4 scattering**
- **The original neutron data library MENDF5 was produced using the "TD-Division Weight Function," also called "CLAW" by the processing code NJOY. This weight function is a combination of a Maxwellian thermal + 1/E + fission + fusion peak at 14.0 MeV.**
- **The data library contains no upscatter groups or self-shielding, and is most applicable for fast systems.**

U^{235} Fission – Continuous vs Multigroup

Continuous-energy
92235.80c Fission



Multigroup
92235.50m Fission

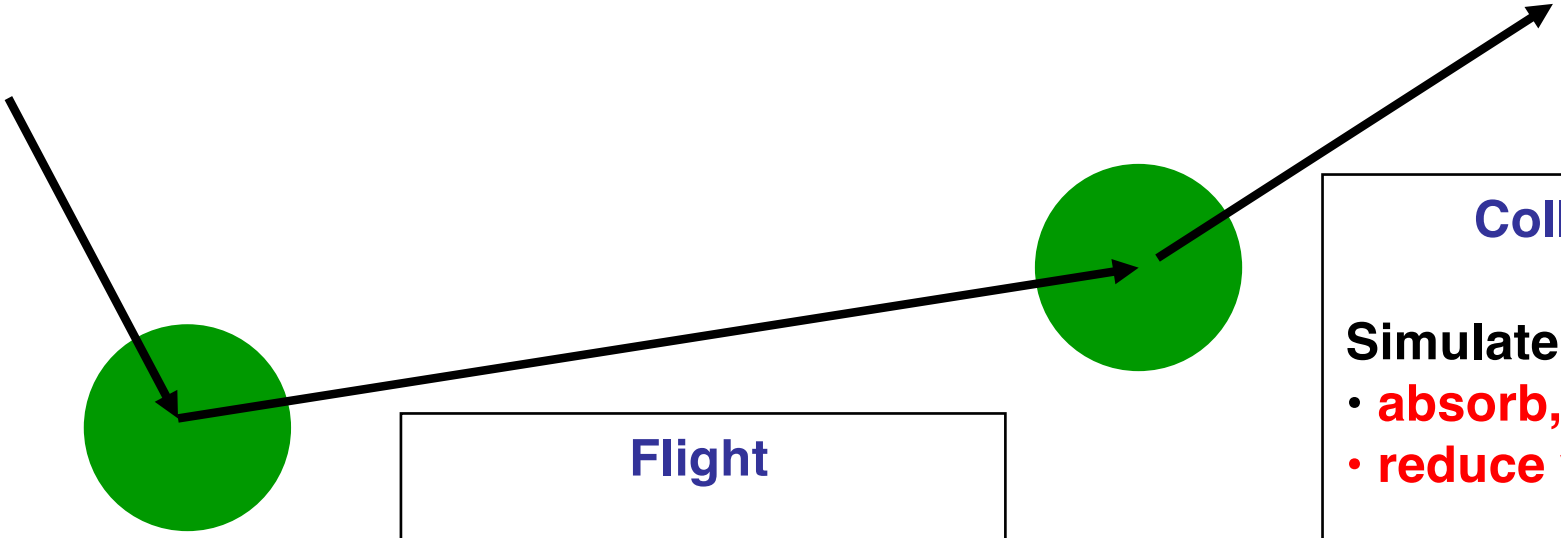


Example – Simplified Godiva HEU Sphere

Continuous, .80c data	`k = 1.0019 (4)`
Multigroup, .50m data	`k = 0.9947 (4)`

Use of Nuclear Data in MCNP

Using the Nuclear Data



Flight

Random sampling using Σ_T & exponential distribution:

- Free-flight distance to next collision, s

- For protons, ionization, multiple scattering & energy straggling are also taken into account

Collision

Simulate absorption:

- absorb, or
- reduce weight

Random sampling using nuclear data:

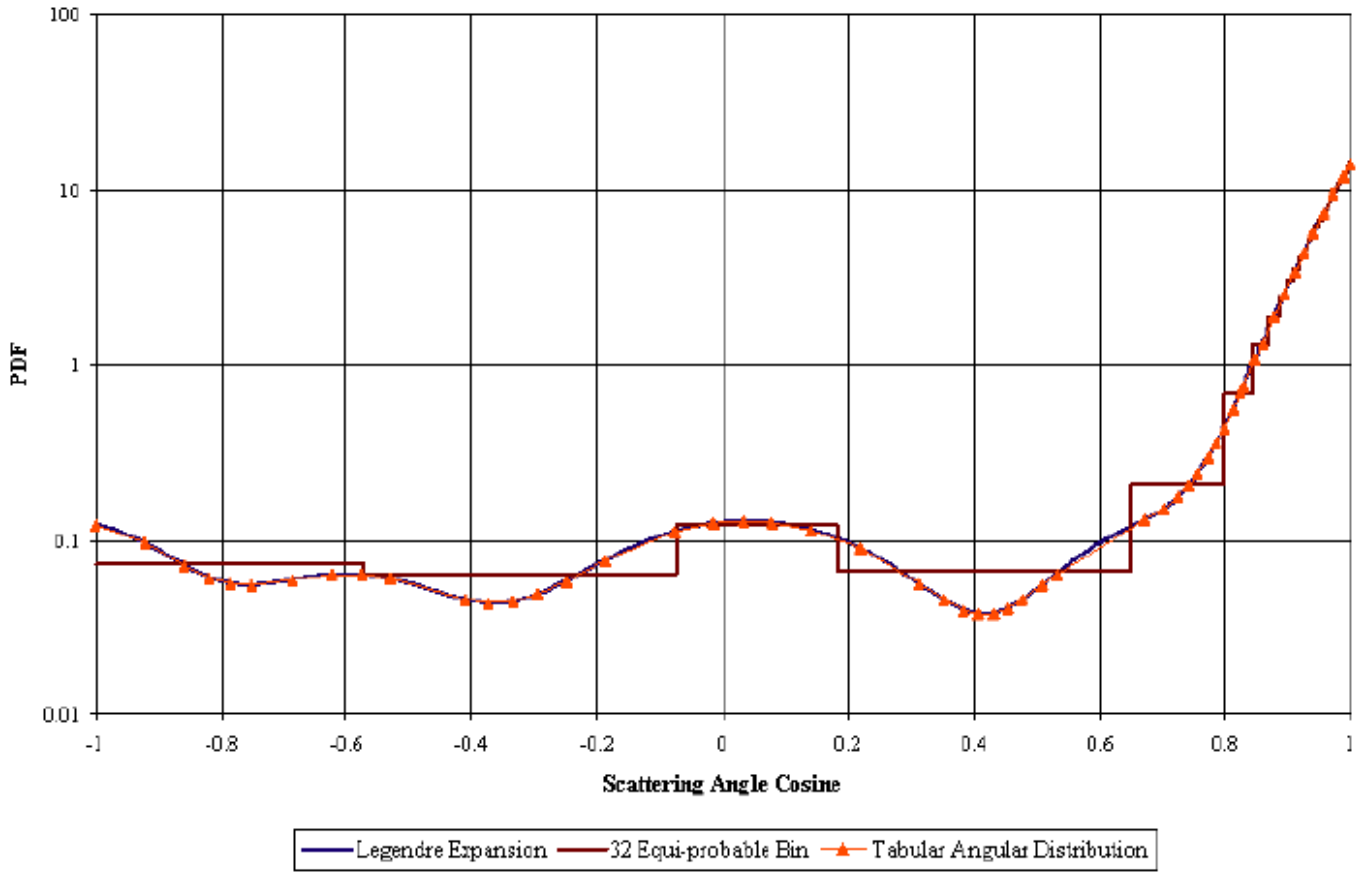
- Collision isotope
- Reaction type
- Exit E' & Ω'
- Secondary particles

For details & much more information, see the optional notes in this lecture & also the reference on the CD:

Fundamentals of Monte Carlo Particle Transport

Scattered Particle Angular Distributions

- **32 equiprobable cosine bins (Old: mcnp4, mcnp5, mcnpx, mcnp6)**
- **Tabulated distribution (New: mcnp4c, mcnp5, mcnpx, mcnp6)**
  - Arbitrary cosine grid
  - Interpolation between points can be either uniform or linear
  - More accurate representation of evaluated data



TOTNU card

- Data libraries include ν (number of neutrons produced from fission) for each fissionable nuclide, tabulated by the incident neutron energy
- Library tables of ν include both prompt & delayed neutron emission data
- Value of ν used in an MCNP calculation is determined by TOTNU card

TOTNU --> $\nu = \nu_{total} = \nu_p + \nu_d$

TOTNU NO --> $\nu = \nu_p$

- For MCNP6, TOTNU is the default for all problems.
- To use only prompt neutrons, specify TOTNU NO

PHYS:N Card

PHYS:N EMAX EMCNF IUNR xxx xxx xxx COILF CUTN . . .

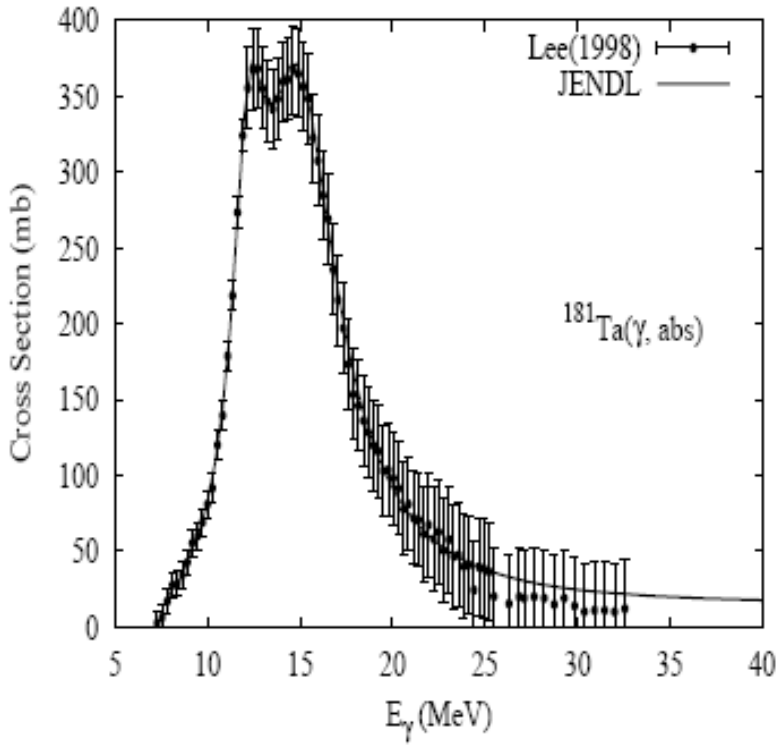
- **EMAX** default: 100 MeV
  Upper limit for neutron energy (MeV)
- **EMCNF** default: 0
  Implicit capture above, analog capture below (MeV)
- **IUNR** default: 0
  0 = Use probability tables for unresolved resonance range
  1 = Don't use probability tables
- **xxx – obsolete options**

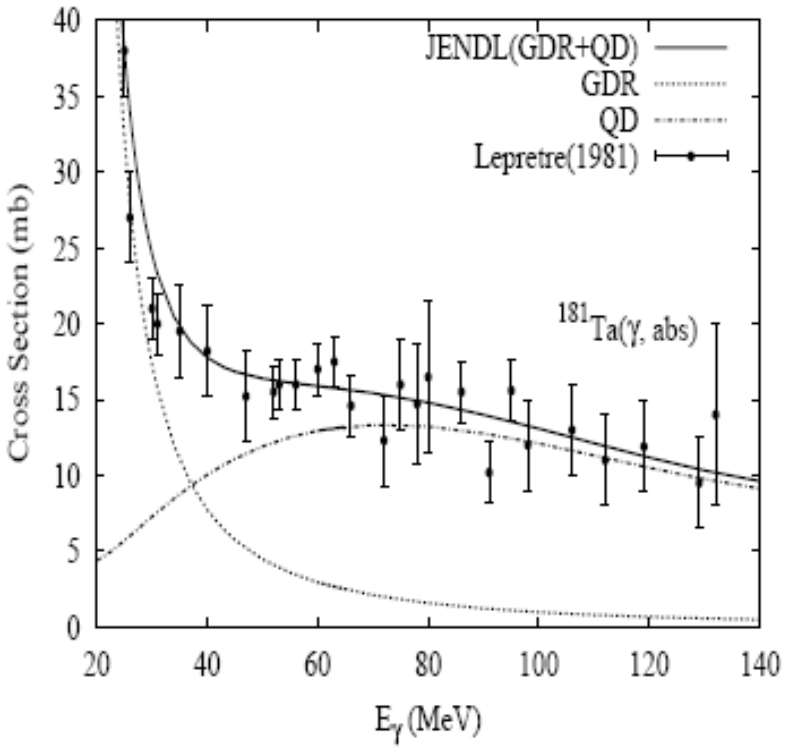
- **COILF (mcnp6)** default: 0
  Light-ion & heavy-ion recoil treatment, see MCNP6 manual
- **CUTN (mcnp6)** default: -1
  >0 = Use models above this energy, data tables below
  -1 = use mix & match treatment for models & data tables
  >emax = no models, only use data tables

Photon Transport

Photonuclear Libraries for MCNP

- **ENDF/B-VII has 157 isotopic photonuclear libraries**
  - **Primarily taken from the IAEA Photonuclear Data Library**
    `http://www-nds.iaea.org/photonuclear/`
  - **Maximum energies and secondary particle descriptions vary greatly**
  - **Generally treat the Giant Dipole Resonance and Quasi-Deuteron Resonance**
- **Photonuclear interactions must be turned on with the *ispn* entry on the PHYS:p card**





Photons

- **Transporting MCNP photons is similar to transporting MCNP neutrons in many ways. The selection of isotope, selection of reaction, secondary particle bank are all the same.**

- **However:**

  – **Photons interact (mostly) with atoms, not nuclei.**
  (Photonuclear interactions have already been discussed.)

  – **Photoatomic interactions include:**
    - Absorption by photo-electric effect
    - Scattering (coherent and incoherent)
    - Pair production

  – **Photon interactions produce electrons. Transporting electrons is computationally very expensive.**

Photons

When transporting photons (p on MODE card) the user has to decide:

- **Should electrons be transported also?**
  - Yes: Use this input data card: mode p e
  - No: Use this input data card: mode p
- **If electrons are not transported, the Thick-Target Bremsstrahlung (TTB) option can approximately account for the electron physics**
  - Don't create & transport electrons
  - Model the photons produced, as if those electrons had been there & slowed down (assumes infinite or large region)
  - This is on by default, can be changed on phys:p card
- **MCNP has two photon physics treatments:**
  - Simple – used only at high energies, E > 100 MeV
  - Detailed – used at all lower energies, E < 100 MeV

Photon Physics Treatments

- **Simple Physics Treatment**
  - Appropriate for high-energy photons and free electrons.
  - Ignores coherent scatter. Only Compton scattering with free electrons
  - Absorption by photoelectric effect is done with implicit capture. No fluorescent photons are produced.
  - Used above the energy EMCPF on the PHYS:p card. By default, this energy is 100 MeV.

- **Detailed Physics Treatment**
  - Appropriate for most problems. Essential for high-Z materials and deep penetration problems.
  - Includes coherent scatter with Form Factors
  - The incoherent scattering cross-section (Klein-Nishina) is modified by a form-factor specific to the atom the photon is interacting with.
  - Absorption by photoelectric effect is done with analog (explicit) capture. Zero, one or two fluorescent photons, or an Auger electron may be produced
  - Used below the energy EMCPF on the PHYS:p card.

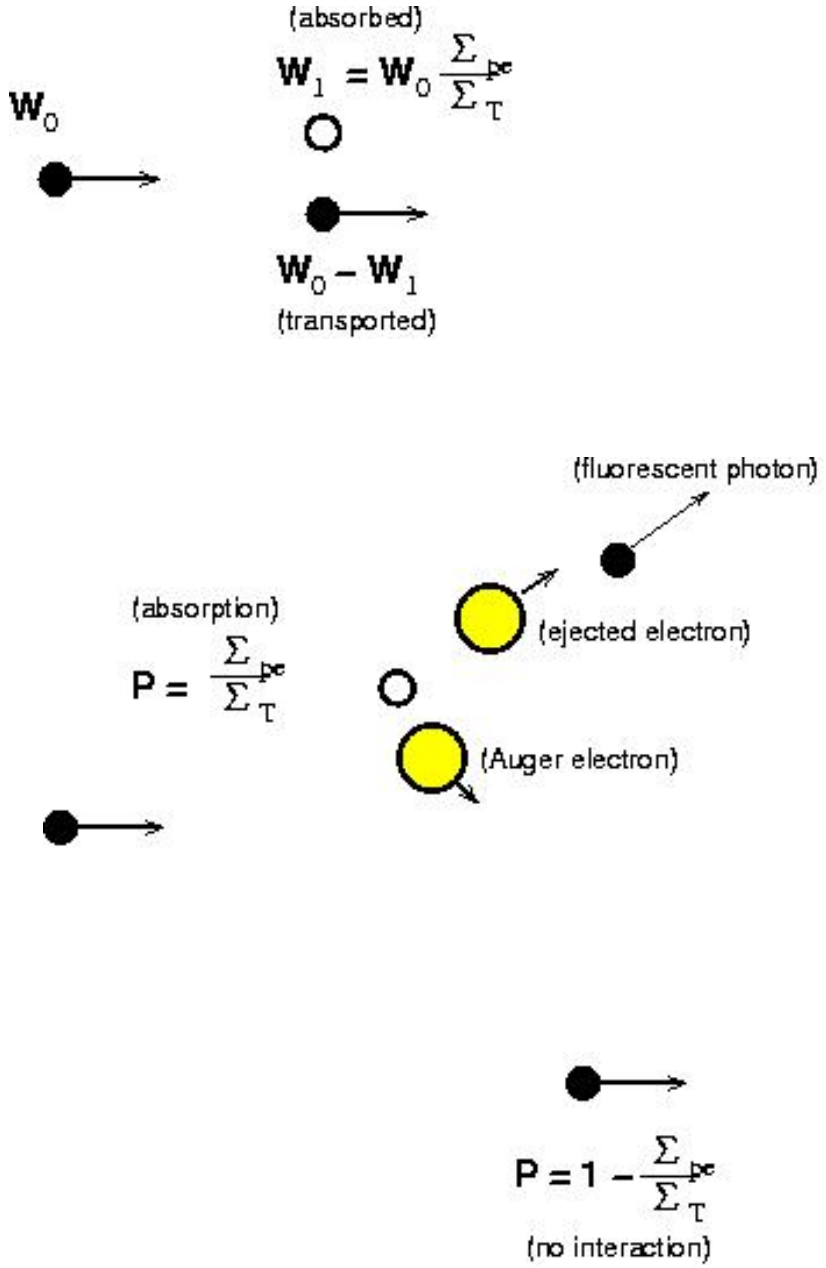
Photoelectric Absorption

Simple physics treatment:

- Photoelectric absorption is accounted for by weight reduction
- The (reduced-weight) photon then experiences either Compton scatter or pair production

Detailed physics treatment:

- If a photoelectric event is selected, the incident photon is absorbed, and secondary particles can be produced for transport
- Otherwise, Thomson or Compton scattering or pair production occurs



Secondary Electrons

- Coherent scattering involves no energy loss and thus can't produce electrons.

- All other reactions can produce electrons. The user has three options for what to do with the electrons:
  - Transport the electrons `mode p e`
  - TTB approximation `mode p`
  - Ignore the electrons `mode p & phys:p j 1`

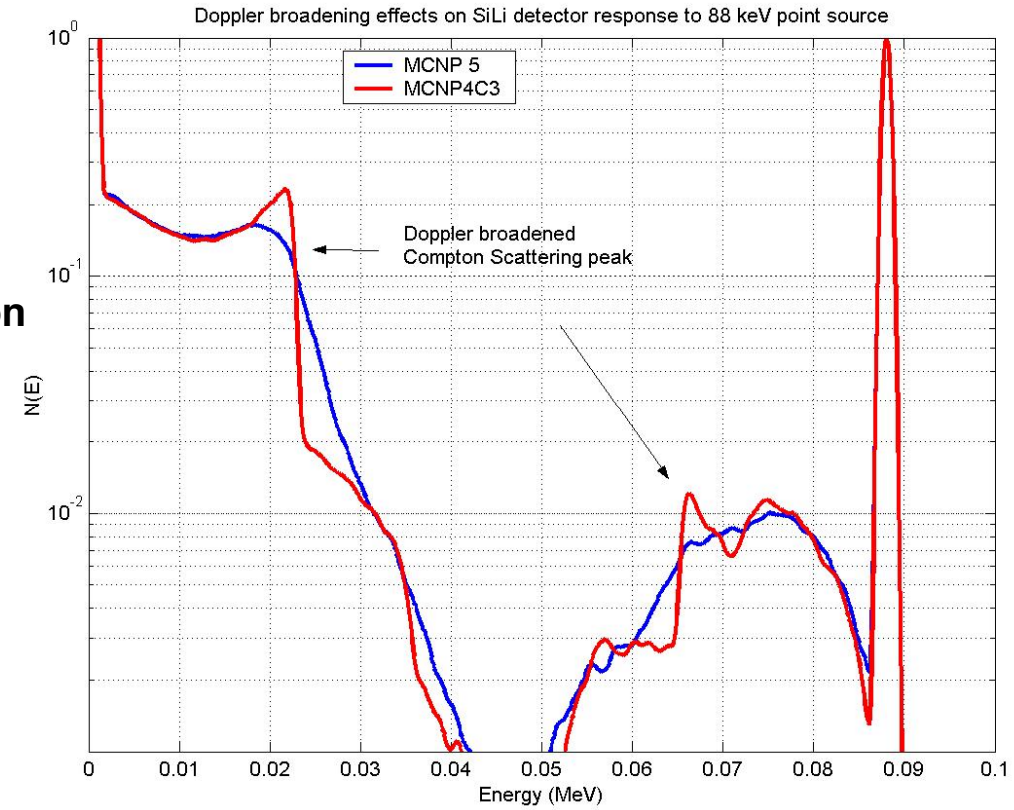
- Using TTB, if an electron is generated, its Bremsstrahlung photons are created but the electron is terminated.

- Results for demo Problem, Chapter 5, MCNP Manual, running on SGI 2000 with 104,000 particles:

MODE p, TTB off	0.10 cpu minutes
MODE p, TTB on	0.14 cpu minutes
MODE p e	27.28 cpu minutes

Photon Doppler Broadening

- **Incoherent photon scattering can occur with a bound electron and generate a Compton electron and a scattered photon.**
  - The electron binding effect becomes increasingly important for incident photon energies less than 1 MeV.
- **The angular distribution of the scattered photon appears as a reduction of the total scattering cross section in the forward direction.**
  - This effect has been accounted for in MCNP by modifying the Klein-Nishina cross section with a form factor.
- **The energy distribution of the scattered photon appears as a broadening of the exit energy spectrum due to the pre-collision momentum of the electron.**
  - This second effect is the definition of Doppler energy broadening for incoherent photon scattering and is in MCNP5 & MCNP6.



Blue – with photon Doppler broadening
Red – without

PHYS:p Card

PHYS:p EMCPF IDES NOCOH PNINT NODOP

EMCPF simple physics above EMCPF (default: 100 MeV)

IDES = 0 = TTB on or electron transport (default)
= 1 = turn off secondary electron production altogether

NOCOH = 0/1 = turn coherent scattering on/off (default: 0)

PNINT = -1 = analog photonuclear interactions on
= 0 = photonuclear physics off (default)
≥ 1 = biased photonuclear interactions on

NODOP = 0/1 = Doppler broadening on/off (default on)

MCNP Photoatomic Data Libraries

- **Older libraries:**
  - **MCPLIB - Based largely on ENDF/B-IV, ZAIDs end in .01p**
  - **MCPLIB02 - Based on EPDL, ZAIDs end in .02p**
  - **MCPLIB03 - Data from MCPLIB02 updated to include Doppler energy broadening data, ZAIDs end in .03p**
- **MCPLIB84**
  - **Based on ENDF/B-VI Release 8 (EPDL97)**
  - **ZAIDs end in .84p**
  - **Data provided for Z=1 to Z=100**
  - **Includes Compton Doppler energy broadening data**
  - **Updated fluorescence data**
  - **Current default**

- **Future plans for photoatomic/electron data:**
  - **Update libraries with new shell-wise data**
  - **Improve atomic relaxation sampling**

For photoatomic data, MCNP converts isotopic ZAIDs on M cards to elemental ZAIDS before searching for cross-section tables.

Plotting Data With MCNP

Plotting Cross-Sections with MCNP

- To plot cross-sections, MCNP needs an input file. The input file should have an M card with the ZAID of the cross-section you want to plot.

- Copy file SOLUTIONS\g1.txt to WORK\g1.txt
- Invoke the cross-section plotter:

```
mcnp   i=g1.txt    ixz
```

i – read input file

x – read cross-sections

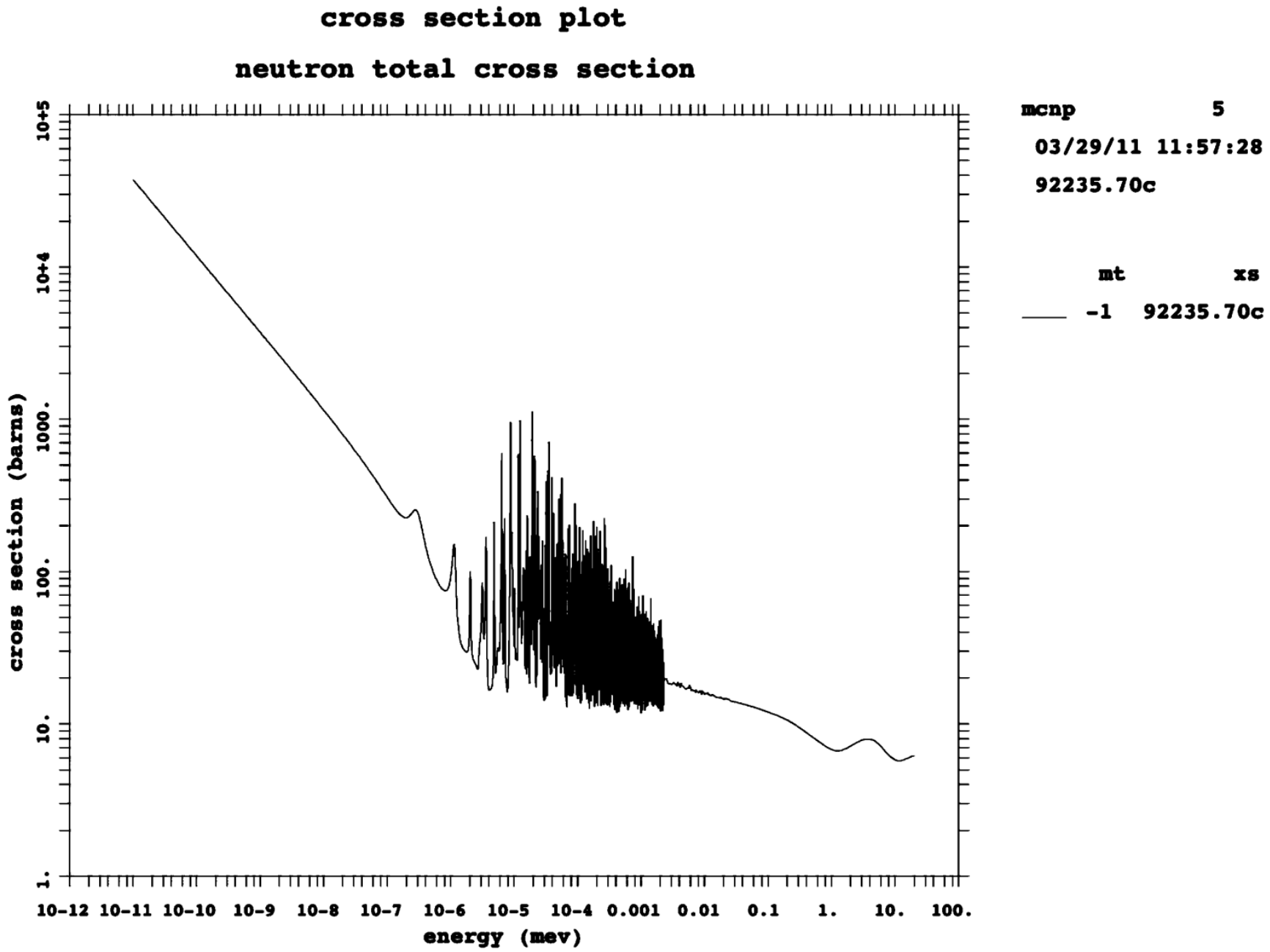
z – invoke the tally/cross-section plotter

Cross-section Plotting

- **Use mcnp to process input & plot cross-sections**

start:	bash:	`mcnp i=g1.txt ixz`
get help:	mcplot>	`xs ?`
get zaid list:	mcplot>	`xs 0`
pick U235:	mcplot>	`xs 92235.80c`
get mt list:	mcplot>	`mt 0`
plots: total	mcplot>	`mt -1`
capture	mcplot>	`mt -2`
elastic	mcplot>	`mt -3`
fission	mcplot>	`mt -6`
exit:	mcplot>	`end`

Cross-section Plotting



Cross-section Plotting

MCNP reaction types for plotting:

Type	Cont.Energy neutrons	Multigroup neutrons	Photons	Cont.Energy proton	Cont.Energy photonuclear
-1	total	total	incoh. scat	total	total
-2	capture	fission	coh. scat	nonelastic	nonelastic
-3	elastic scat	nu	photoelectric	elastic	elastic
-4	heating	chi	pair product	heating	heating
-5	photon prod.	capture	total	production	production
-6	fission	stop.power	heating		
-7	nu	mom.transfer			
-8	Q				

ENDF/B MT numbers for plotting

1	total	total
2	elastic scatter	nonelastic
18	fission	
101	capture	
102	(n,gamma)	

See Listing of Available ACE Data Tables, LA-UR-13-21822

Cross-section Plotting

- **Comparing 2 different neutron cross-section sets**

Edit the file g1.txt:

add to one of the material cards

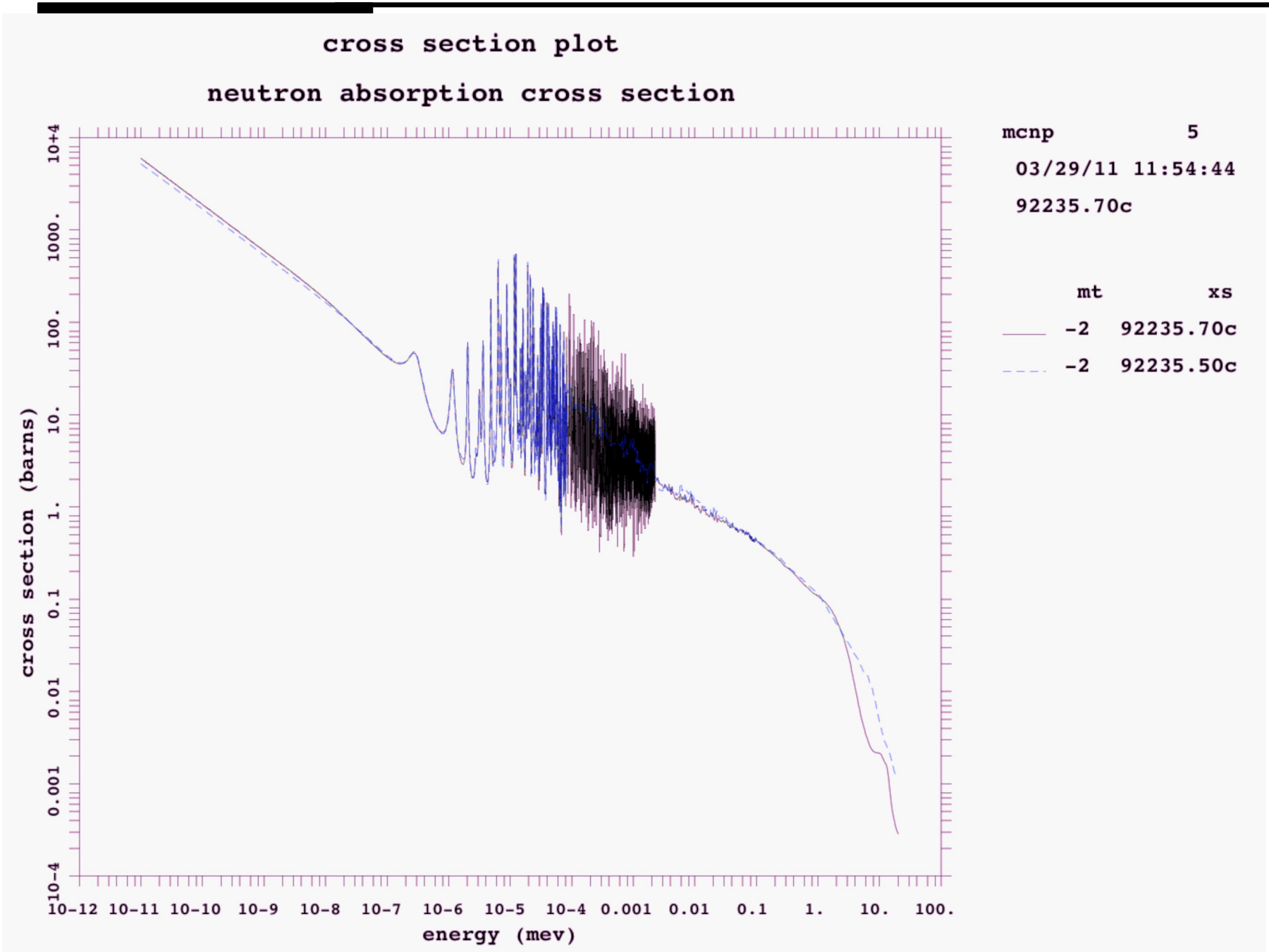
92235.50c -1.e-10

Run mcnp again to plot cross-sections:

```
mcnp   i=g1.txt   ixz
mcplot>  xs 92235.80c mt -2   coplot   xs 92235.50c mt -2
mcplot>    … try some other plots …
mcplot> end
```

note: second plot is done over the first …

Cross-section Plotting



Cross-section Plotting - Photons

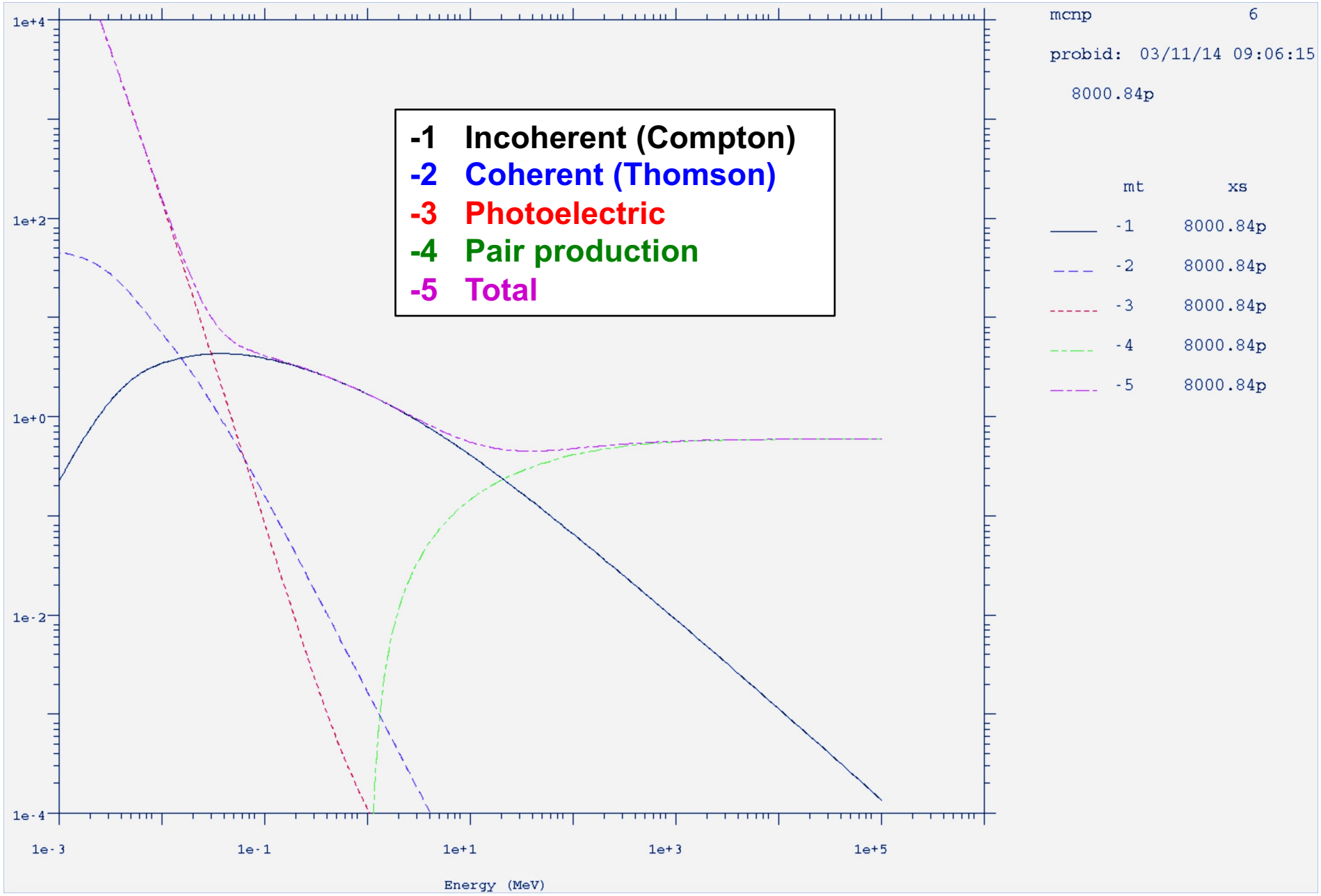
- **Plotting photon cross sections**

MT	FM	Description
501	-5	Total
504	-1	Incoherent (Compton)
502	-2	Coherent (Thomson)
522	-3	Photoelectric
516	-4	Pair production
301	-6	Heating

Copy the file ..\SOLUTIONS\source1.txt to the WORK folder

```
mcnp6   i=source1.txt   ixz
mcplot> xs 8000.84p
mcplot> ylims  1e-4  1e4
mcplot> mt -1  coplot  mt -2  coplot &
mcplot> mt -3  coplot  mt -4  coplot &
mcplot> mt -5
mcplot> end
```

Cross-section Plotting - Photons

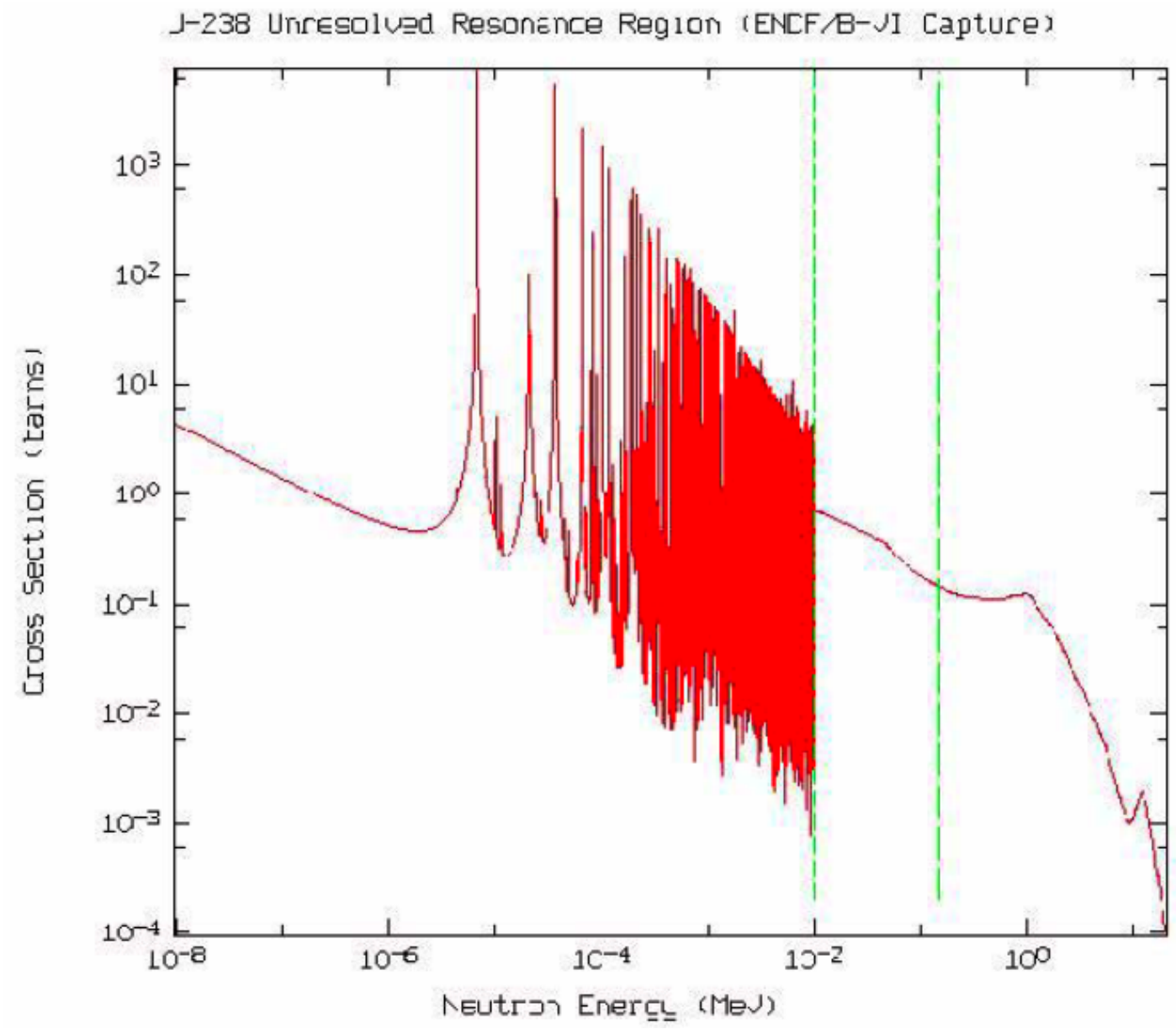


Treatment of Unresolved Resonance Region

-

Optional Material

U-238 Unresolved Resonance Region (ENDF/B-VI Capture)



Treatment of Unresolved Resonance Region

- Self-shielding effects in the unresolved resonance range can be accurately represented by a probability-table treatment ("stratified sampling")
- Probability-table treatment is statistical in nature
- Probability-table treatment samples cross sections from energy-dependent distributions within the unresolved energy range rather than using smooth, infinitely-dilute values

Unresolved Resonance Ranges for Uranium & Plutonium Isotopes

Isotope	Unresolved Resonance Range (keV)	
	Lower Limit	Upper Limit
^{233}U	0.06	10
^{234}U	1.5	100
^{235}U	2.25	25
^{236}U	1.5	100
^{238}U	10	149.03

Isotope	Unresolved Resonance Range (keV)	
	Lower Limit	Upper Limit
^{239}Pu	2.5	30
^{240}Pu	5.7	40
^{241}Pu	0.3	40.2
^{242}Pu	0.986	10

Reactivity Effect of Probability-table Treatment

Benchmark Name	Type	Δk_{PT}
Jezebel	Pu	0.0002 ± 0.0008
Jezebel-240	Pu	0.0001 ± 0.0008
VERA-11A	Pu	-0.0005 ± 0.0009
HISS/HPG	Pu	0.0002 ± 0.0008
PNL-2	Pu	0.0012 ± 0.0015

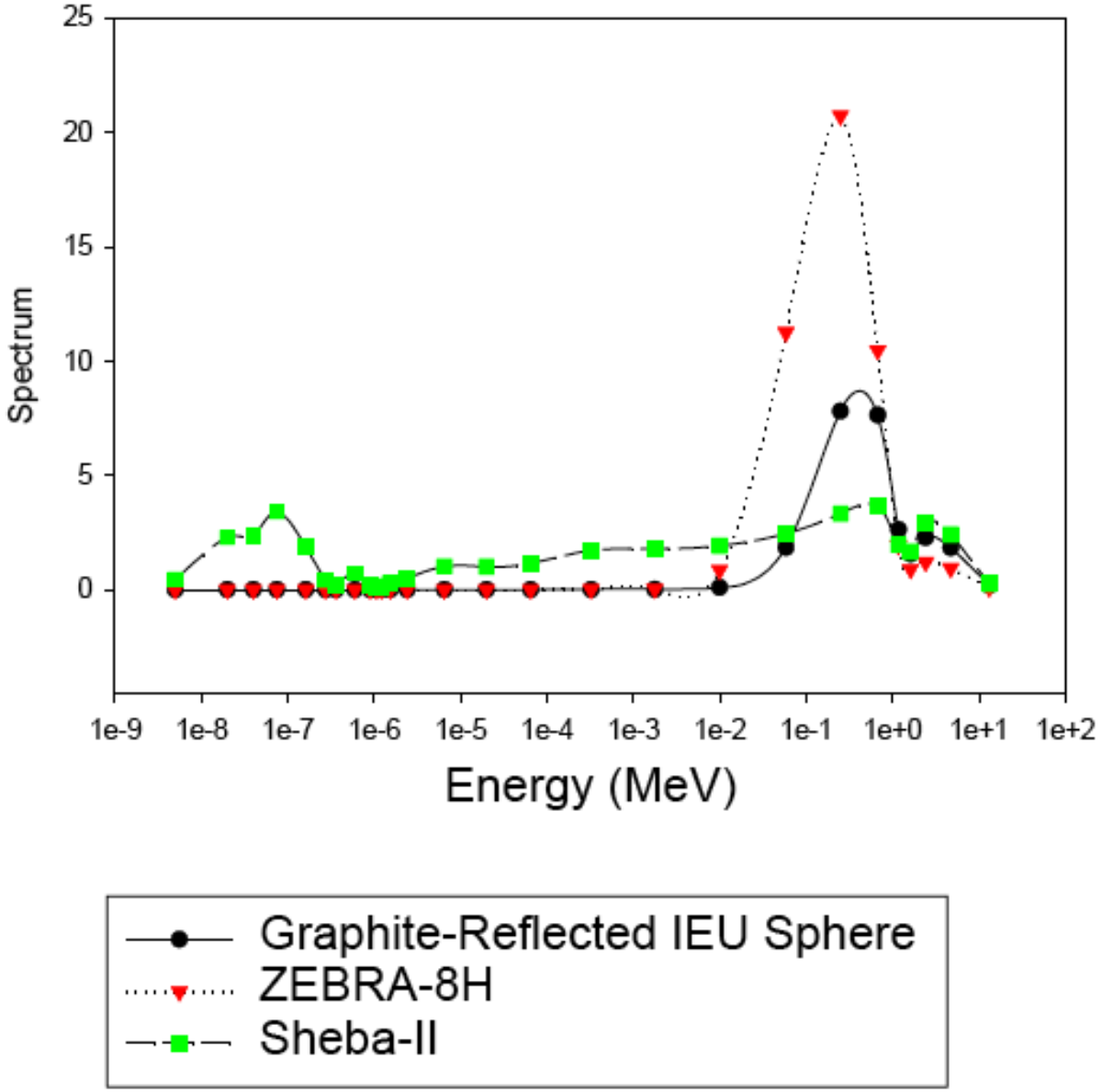
Benchmark Name	Type	Δk_{PT}
Jezebel-23	^{233}U	0.0002 ± 0.0008
Flattop-23	^{233}U	0.0007 ± 0.0009
^{233}U + W	^{233}U	-0.0001 ± 0.0009
^{233}U + Be	^{233}U	-0.0006 ± 0.0009
Falstaff-1	^{233}U	-0.0013 ± 0.0015
ORNL-9	^{233}U	0.0006 ± 0.0008

Reactivity Effect of Probability-table Treatment

Benchmark Name	Type	Δk_{PT}
Godiva	HEU	-0.0002 ± 0.0008
ZPR-9-34	HEU	-0.0005 ± 0.0009
VERA-1B	HEU	-0.0003 ± 0.0009
ZEUS	HEU	-0.0002 ± 0.0011
UH_3 Case 4	HEU	-0.0004 ± 0.0011
HISS/HUG	HEU	0.0007 ± 0.0007
ORNL-4	HEU	-0.0008 ± 0.0008

Benchmark Name	Type	Δk_{PT}
GRIEUS	IEU	0.0004 ± 0.0008
ZPR-III-2	IEU	0.0024 ± 0.0008
ZPR-III-6F	IEU	0.0025 ± 0.0009
ZPR-III-12	IEU	0.0044 ± 0.0008
ZEBRA-2	IEU	0.0052 ± 0.0008
ZPR-III-11	IEU	0.0024 ± 0.0006
BIG TEN	IEU	0.0048 ± 0.0007
ZEBRA-8H	IEU	0.0115 ± 0.0006
SHEBA-II	LEU	0.0009 ± 0.0011

Comparison of Spectra



Conclusions: Probability-table Treatment

- **Probability-table treatment produces**
  - **No significant changes for fast or thermal systems**
  - **Negligible reactivity changes for HEU benchmarks**
  - **Substantial increases in reactivity for systems with large amounts of ^{238}U and intermediate spectra**
- **Probability-table treatment is required to prevent nonconservative results for systems with intermediate spectra and large amounts of ^{238}U (or, probably, other fertile isotopes)**

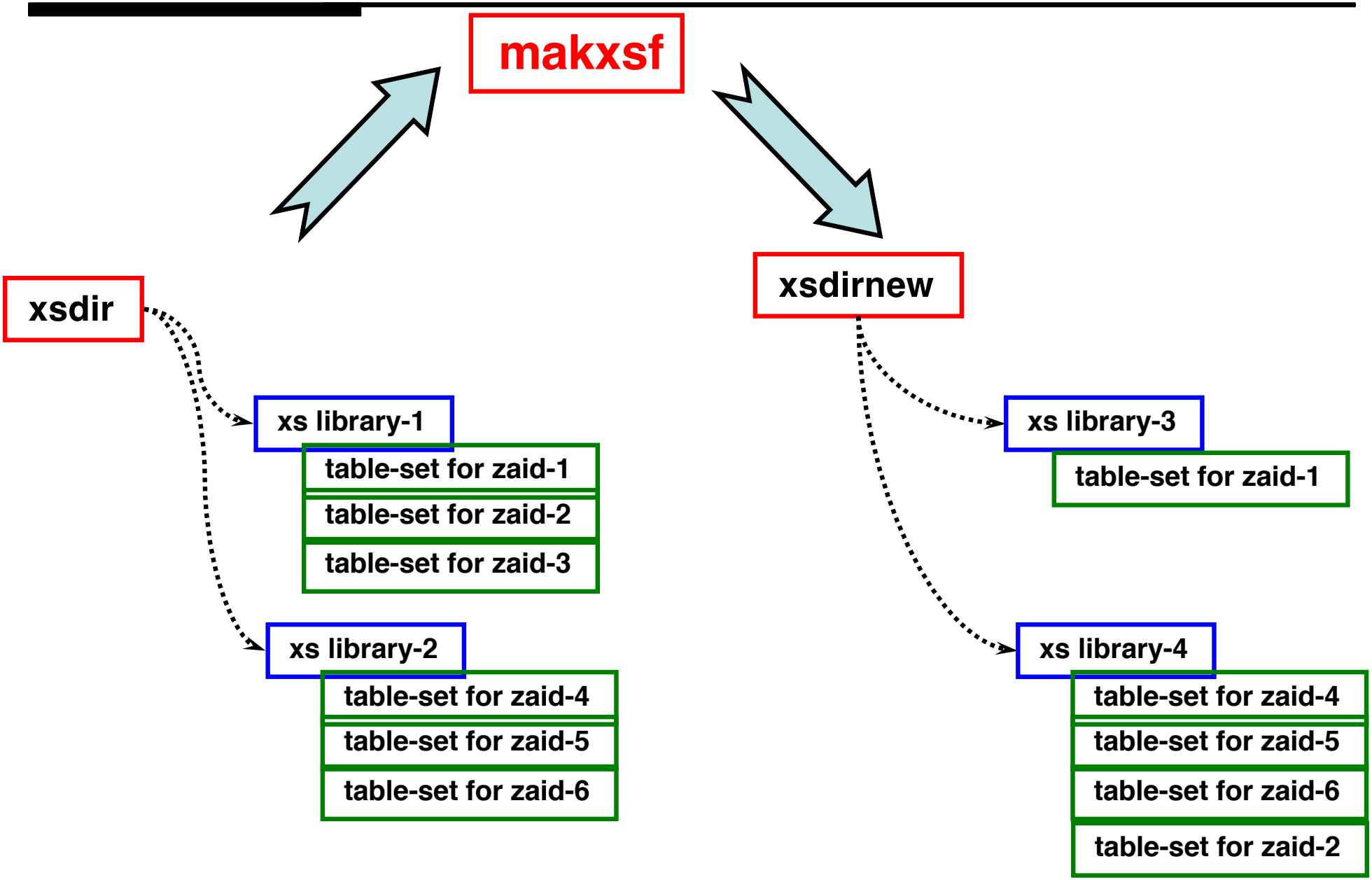
makxsf Code with Doppler Broadening for MCNP

Optional Material

makxsf Code

- **Convert cross-section libraries:**
  - From Type-1 (ASCII) to Type-2 (binary)
  - From Type-2 (binary) to Type-1 (ASCII)
- **Copy entire libraries to new files, Type-1 or Type-2**
- **Copy selected nuclide table-sets to new libraries, Type-1 or Type-2**
- **Create nuclide table-sets at new temperatures** [new]
  - Doppler broaden resolved resonance data to higher T
  - Interpolate unresolved resonance probability tables to new T
  - Interpolate S(alpha,beta) thermal data to new T
- **Create new xsdir file which includes all of the above changes**

makxsf


makxsf
xsdir
xsdirnew
xs library-1
table-set for zaid-1
table-set for zaid-2
table-set for zaid-3
xs library-2
table-set for zaid-4
table-set for zaid-5
table-set for zaid-6
xs library-3
table-set for zaid-1
xs library-4
table-set for zaid-4
table-set for zaid-5
table-set for zaid-6
table-set for zaid-2

Input / Output Files

- **Input: specs file**
  - **Free format**
  - **Blank or tab field delimiters [new]**
  - **80-character records**
  - **End line in "&" or "+" to continue [new]**
  - **Comment lines start with "#", "!", "$", "!" [new]**

- **Output:**
  - **Information previously went to "tprint" file [old]**
    - Abort if "tprint" file already existed [annoying]
  - **Information now printed to screen output [new]**

specs File

Line 0 - optional - old xsdir path **[new]**

datapath=/dir/containing/xsdir

Line 1 - names of old & new xsdir files

old_xsdir new_xsdir

The sets of lines below may be repeated

[for type 1, omit new-lenrec and new-itemrec]

copy all of an old xs-library to a new xs-library

old-xs-lib new-xs-lib new-type [new-lenrec new-itmrec]
[new-route, or blank]

copy selected xs-tables to a new xs-library

new-xs-lib new-type [new-lenrec new-itmrec]
[new-route, or blank]
list of zaid's, one per line:
zaid **table is just copied**
zaid-new T zaid-low zaid-high **[new] doppler broaden, interp Sab/ptable**
blank line

Sample specs File

```
#
# test for new makxsf
#
# old-xsdir  new-xsdir
  testdir1    testdir2
#
# copy selected xs-tables to testlib2, type2
 testlib2   2
 /home/fbrown/mcnp6_new/MCNP6/Testing/xsec_data/testlib2
   1001.60c
   1001.50m
   …
   7014.50d
                              [blank line]
#
# copy some more to testlib3, type 1
 testlib3  1
 /home/fbrown/mcnp6_new/MCNP6/Testing/xsec_data/testlib3
   7000.01g
   7014.50m
   8016.40c
   …
   82000.50m
                              [blank line]
#
# copy some more to testlib4, type 2
 testlib4  2
 /home/fbrown/mcnp6_new/MCNP6/Testing/xsec_data/testlib4
   92235.50d
   92000.01g
   92235.50m
   92238.40c
   92238.50m
   lwtr.01t
   …
# for doppler broad, T can be MeV or degrees-K
   92235.00c  540  92235.69c  92235.68c
#
# if higher T does not exist or not needed, use 0
   8016.01c   540  8016.69c   0
                              [blank line]
#
# done
```

Doppler & Interpolation Routines

- **Taken from "doppler" code by MacFarlane, taken from NJOY**
  - **References:**
    - R.E. MacFarlane & P. Talou, "DOPPLER: A Utility Code for Preparing Customized Temperature-Dependent Data Libraries for the MCNP Monte Carlo Transport Code", unpublished (Oct 3, 2003)
    - R.E. MacFarlane & D.W. Muir, "The NJOY Nuclear Data Processing System, Version 91", LA-12740-M (1994).

- **Doppler**
  - **Doppler broaden the resolved resonance data to new (higher) temperature**
  - **Temperatures can be specified in degrees-K or in MeV**
  - **Only need base cross-section, at lower temperature**

- **Interpolation**
  - **For S(alpha,beta) thermal data or unresolved resonance probability table data**
  - **Must have existing datasets at BOTH lower & higher temperature**

Testing

- **For MCNP5/6, testing of Type-2 cross-sections was changed**
  - **Use makxsf to convert testlib1 (Type-1) into 3 separate xs-libraries: testlib2 (Type-2), testlib3 (Type-1), testlib4 (Type-2)**
  - **Run the REGRESSION test set**
- **Compared every word of various table-sets generated by original DOPPLER code & new makxsf**
  - **Matched exactly, except for Zr/H S(a,b) data (due to bug in DOPPLER)**
- **Visually compared xsec plots of Doppler-broadened U235 xsecs to standard library xsecs**
- **For V&V of Doppler & interpolation routines, reran Mosteller's Kritz benchmarks**

Testing - Kritz Benchmarks

- **Mosteller, MacFarlane, Little, White, "Analysis of Hot and Cold Kritz Benchmarks With MCNP5 and Temperature-specific Nuclear Data Libraries", LA-UR-03-7071 (2003).**
  - Separate nuclear data sets were generated at 245 C using DOPPLER and NJOY
  - Basic data were taken from ENDF/B-VI Release 6 (ENDF66)
  - MCNP5 calculations were performed for the hot 2-D benchmarks, and the results were compared
  - Each calculation employed 550 generations with 10,000 neutron histories per generation
  - Results from first 50 generations were discarded, giving 5,000,000 active histories for each case

Case	Library	keff	Δk (vs NJOY)
Kritz:2-1	NJOY	0.9914 ± 0.0003	—
	DOPPLER	0.9911 ± 0.0003	-0.0003 ± 0.0004
	new MAKXSF	0.9913 ± 0.0003	-0.0001 ± 0.0004
Kritz:2-13	NJOY	0.9944 ± 0.0003	—
	DOPPLER	0.9942 ± 0.0003	-0.0002 ± 0.0004
	new MAKXSF	0.9940 ± 0.0003	-0.0004 ± 0.0004
Kritz:2-19	NJOY	1.0005 ± 0.0003	—
	DOPPLER	1.0009 ± 0.0003	0.0004 ± 0.0004
	new MAKXSF	1.0004 ± 0.0003	-0.0001 ± 0.0004

Monte Carlo Parameter Studies & Uncertainty Analyses With MCNP6

Introduction
mcnp_pstudy
Examples
Usage
Statistics
Examples

Frequent Questions

How are calculated results affected by:

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

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

Brute force approach:

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

mcnp_pstudy

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

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

mcnp_pstudy

- **Completely automates the setup/running/collection for parameter studies & total uncertainty analyses**
  - Painless for users
  - 1 input file & run command can spawn 100s or 1000s of jobs
  - Fast & easy way to become the #1 user on a system
    (Added bonus: make lots of new friends in computer ops & program management.)
- **Ideal for Linux clusters & parallel ASC computers:**
  - Can run many independent concurrent jobs, serial or parallel
  - Faster turnaround: Easier to get many single-cpu jobs through the queues, rather than wait for scheduling a big parallel job
  - Clusters always have some idle nodes

mcnp_pstudy

- **mcnp_pstudy is written in *perl***
  - 640 lines of perl (plus 210 lines of comments)
  - Would have taken many thousands of lines of Fortran or C
- **Portable to any computer system**
  - Tested on Unix, Linux, Mac OS X, Windows
  - For Windows PCs, need to execute under the Cygwin shell
- **Can be modified easily if needed**
  - To add extra features
  - To accommodate local computer configuration
    - Node naming conventions for parallel cluster
    - Batch queueing system for cluster
    - Names & configuration of disk file systems (ie, local or shared)
    - Location of MCNP6 and MCNP6.mpi

Examples

MCNP input for simple Godiva calculation

```
gdv
c
1   100 -18.74   -1    imp:n=1
2   0             1    imp:n=0

1      so 8.741

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

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

```
gdv-A
C @@@  RADIUS = 8.500  8.741  8.750
1   100 -18.74   -1    imp:n=1
2   0             1    imp:n=0

1      so  RADIUS

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

Basics

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

```
c  @@@
```

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

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

- **Parameter definitions have the form**

```
c  @@@   P =    value or list
c  @@@   P = (  arithmetic-expression  )
```

- **Constraints have the form**

```
c  @@@   CONSTRAINT = (  expression  )
```

- **Control directives have the form**

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

Parameter Definition

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

```
C   @@@     P1   =   value
```

- **List of values**

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

- **List of N random samples from a Uniform probability density**

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

- **List of N random samples from a Normal probability density**

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

- **List of N random samples from a Lognormal probability density**

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

- **List of N random samples from a Beta probability density**

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

Parameter Definition

- **Arithmetic expression**

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

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

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

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

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

Parameter Definition

- **Examples**

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

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

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

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

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

Parameter Definition

Random Sampling of Parameters

- uniform probability density, each sample is obtained as

  P = *xmin* + (*xmax-xmin*)*rand()

- normal probability density, each sample is obtained using the Box-Muller scheme

  P = *ave* + *dev* * sqrt(-2*log(rand()) * sin(2*pi*rand())

- lognormal probability density, each sample is obtained as

  P = exp(*ave* + *dev* * sqrt(-2*log(rand()) * sin(2*pi*rand()))

- beta probability density, each sample is obtained as

  x = product of a RNs, y = product of b RNs

  P = *log(x) / (log(x)+log(y))*

Arithmetic Expressions & Constraints

- Evaluated within perl, using the *eval* function
- Must conform to perl rules for arithmetic

Parameter Expansion

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

Example:

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

Outer:

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

Inner:

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

Constraint Conditions

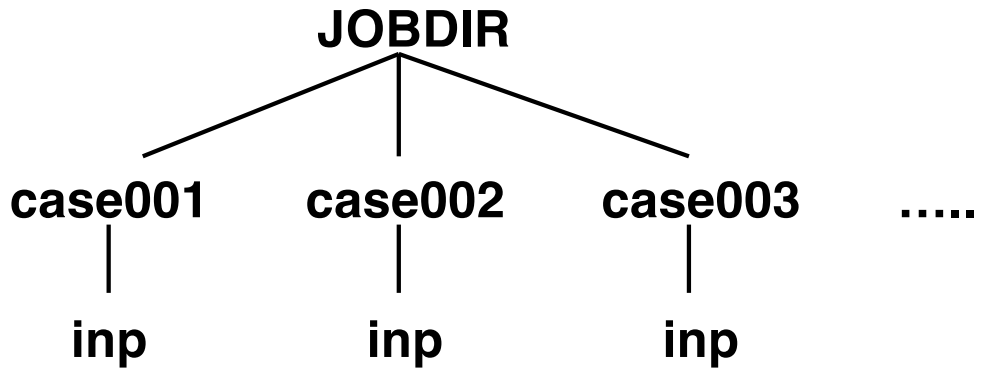
- After all parameters are defined & expanded, constraint conditions are evaluated
- Constraints involve comparison operators (>, <, >=, <=, ==, !=) or logical operators (&& (and), || (or), ! (not)), and may involve arithmetic or functions
- Constraints must evaluate to True or False
- If a any constraint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

Example

```
C pick a random direction
C @@@  ANGLE = ( 6.2831853 * rand(1) )
C @@@  UUU   = ( cos(ANGLE) )
C @@@  VVV   = ( sin(ANGLE) )
C
C same, using CONSTRAINT to implement rejection scheme
c @@@  RN1 = ( 2.*rand(1) - 1. )
C @@@  RN2 = ( 2.*rand(1) - 1. )
C @@@  CONSTRAINT = ( RN1**2 + RN2**2  <  1.0 )
C @@@  UUU = ( RN1 / srqt(RN1**2 + RN2**2) )
C @@@  VVV = ( RN2 / sqrt(RN1**2 + RN2**2) )
```

Creating INP Files & Job Directories

- **Directory structure for MCNP5 jobs**



  - Unix filesystem conventions followed

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

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

Job Options

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

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

  - Within the INP file

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

- **Common options**

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

Running or Submitting Jobs

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

Examples:

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


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


bash:  mcnp_pstudy -inner -i inp01 -setup -submit
    … wait till all jobs complete…
bash:  mcnp_pstudy -inner -i inp01   -collect
```

Creating Input Files ONLY

- **To bypass the creation of job directories, and running/submitting problems:**
  - A special command line option is available: **-inponly**
  - Invoking this option performs the parsing & setup of the input files for each case, but the resulting mcnp input files are placed in the current directory with default names of the form

    `inp_case001, inp_case002, etc.`
  - Using `-case study01a -inponly` would result in files with names

    `inp_study01a001, inp_study01a002, etc.`
  - Other options `-run, -submit` cannot be used if `-inponly` is present
  - The option `-whisper` can be used, and is equivalent to `-inponly`

Combining Results

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

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

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

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

$$\sigma^2_{\bar{X}} \approx \sigma^2_{\bar{X},\ \text{Monte Carlo}} + \sigma^2_{\bar{X},\ \text{Initial Conditions}}$$

Varies as 1/M ~ Constant

Examples

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

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

1     so  URAD

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

Vary fuel density & mass independently, for 50 cases

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

1    so  URAD

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

Examples

Table 1. Results from varying parameters in the Godiva problem

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

Applications

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

Conclusions

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

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

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

Practical Examples from Criticality Safety

Examples

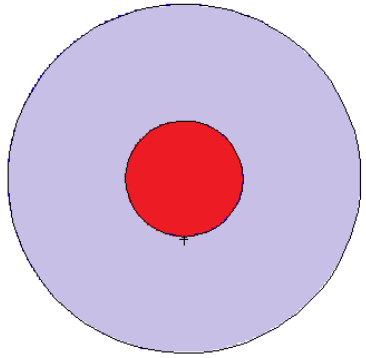
- **wval4:** 4.5 kg Pu Sphere, Ta-reflected with varying reflector thickness
- **wval1:** 4.5 kg Pu Ingot, solid cylinder with varying H/D
- **wval2:** 4.5 kg Pu Ring, hollow cylinder with varying H & R_{in}

Example

wval4,
4.5 kg Pu Sphere,
Ta-reflected

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

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



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

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

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

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

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

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

  1 so   3.7857584
  2 so   11.405758

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

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

  1 so    R_INNER
  2 so    R_OUTER

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

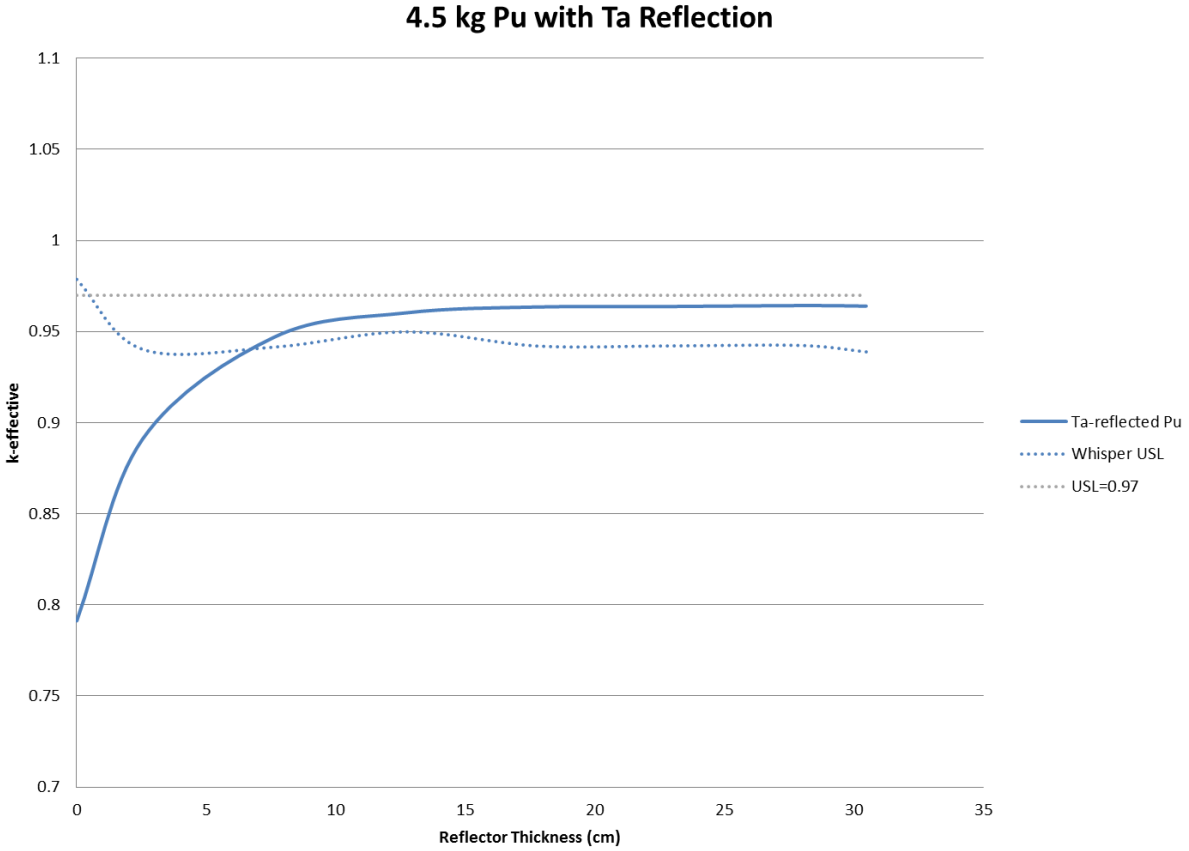
Example wval4: 4.5 kg Pu Sphere, Ta-reflected (3)

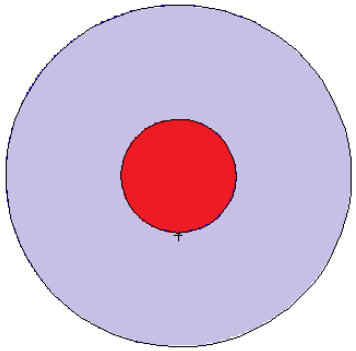
wval4, thick=7.62
mcnp6 i=wval4.txt

k = 0.94638 (41)

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

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





Example

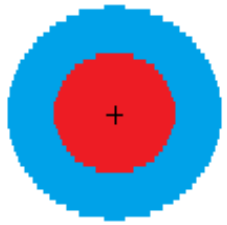
wval1,

4.5 kg Pu Ingot,

varying H/D

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

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



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

    mcnp6 i=wval1.txt
  - **Copy wval1.txt to wval1p.txt, then insert directives for mcnp_pstudy**
    - Define list for HD:

      `c @@@ HD = 0.5 1.0 1.5 2.0 2.5 3.0`
    - For a given H/D, compute Pu radius, then other dimensions

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

    - Use parameters for dimensions & location of KSRC point

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

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

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

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

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

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

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

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

wval1, H/D = 1

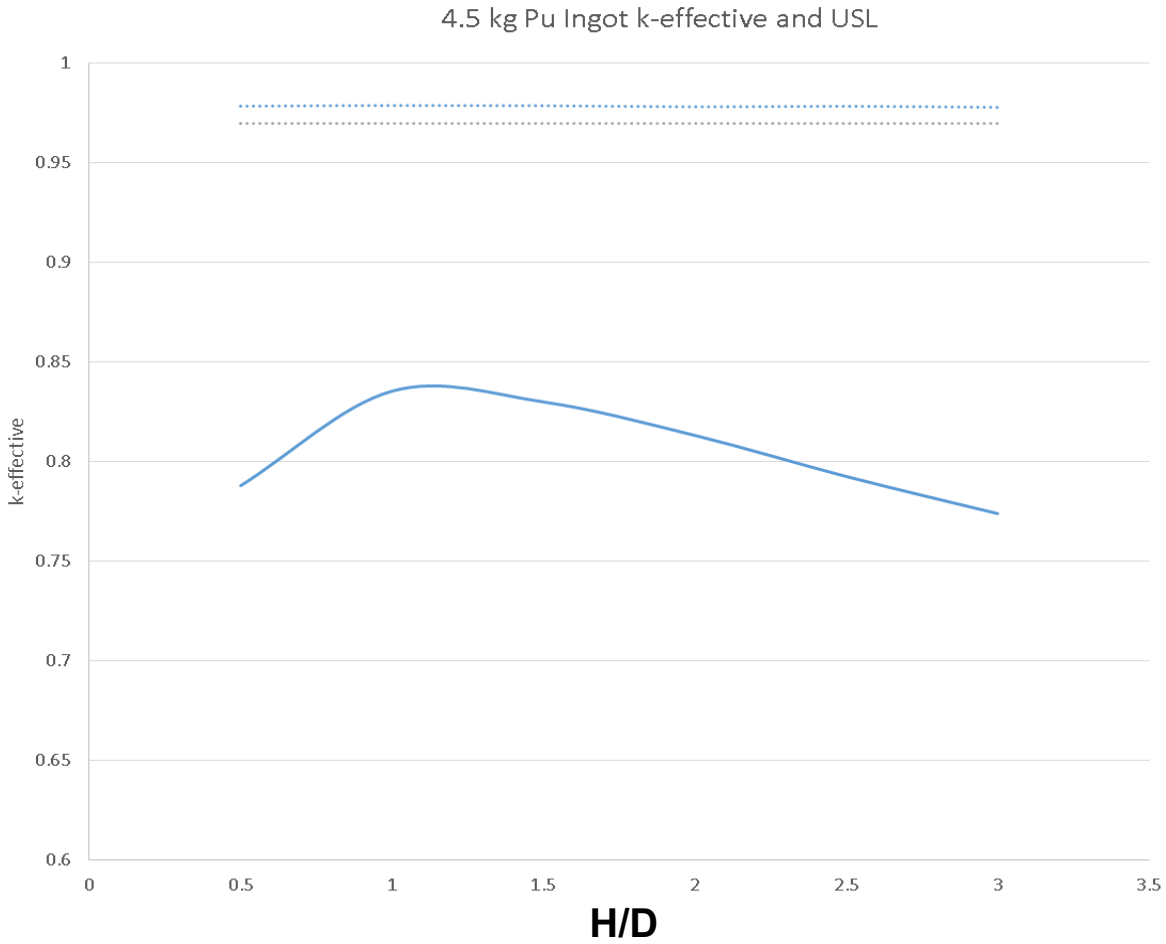
```
mcnp6 i=wval1.txt
```

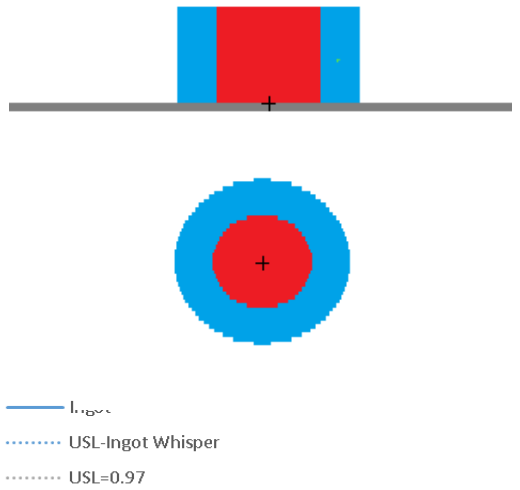
k = 0.83491 (41)

wval1p, varying H/D

```
mcnp_pstudy -i wval1p.txt -setup -run
```

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



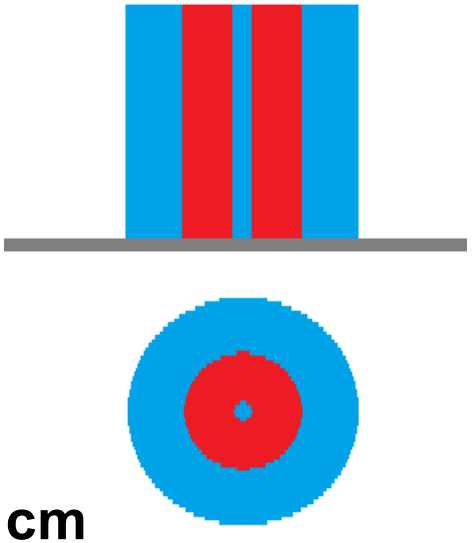


Example

wval2,
4.5 kg Pu Annulus,
varying H & R_{in}

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

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



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

    mcnp6 i=wval2.txt
  - Copy wval2.txt to wval2p.txt, then insert directives for mcnp_pstudy
    - Define list for solid HD:

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

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

Solid cylinder

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

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

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

Hollow cylinder

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

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

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

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

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

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

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

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

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

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

wval2

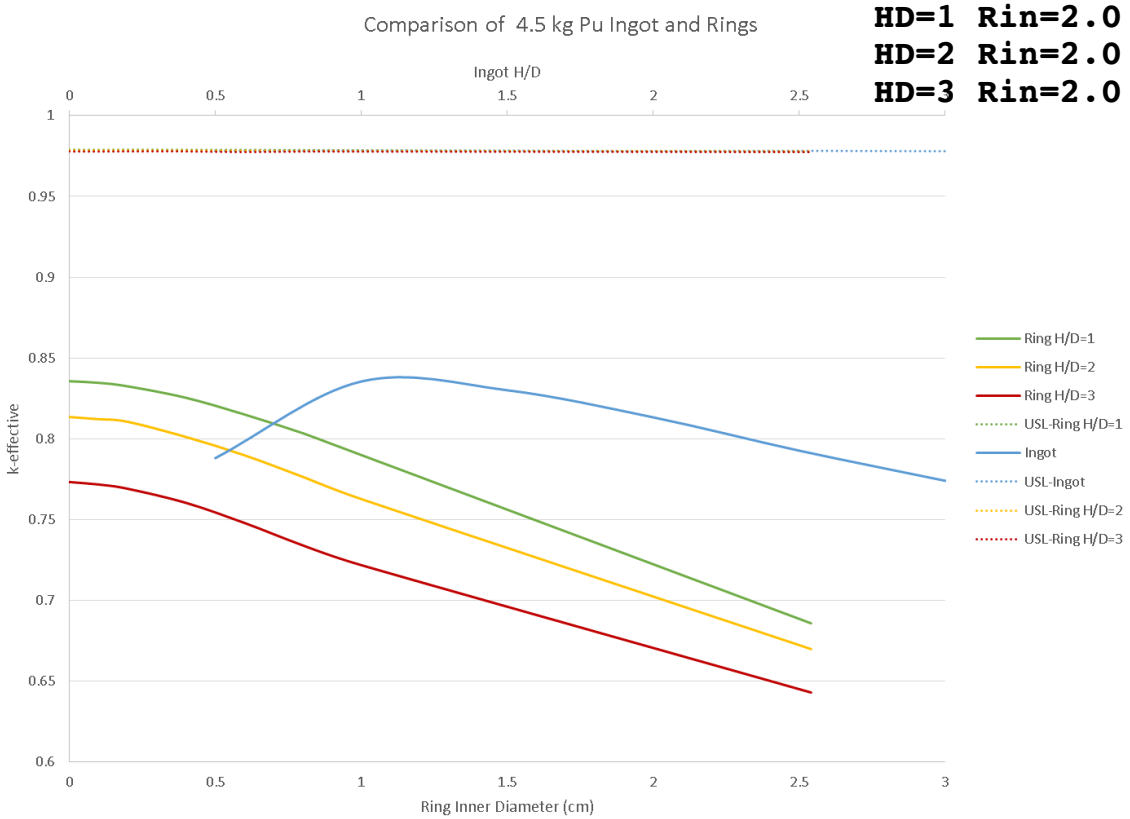
mcnp6 i=wval2.txt

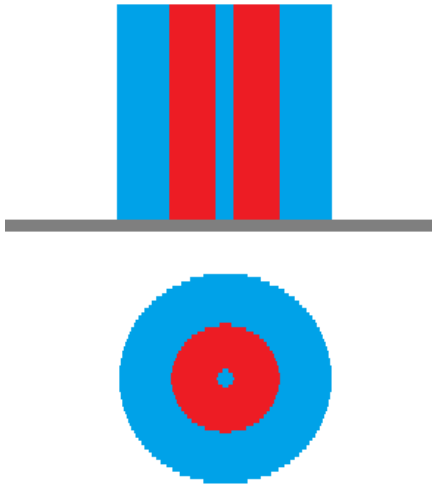
k = 0.83413 (42)

wval2p, varying H & R_{in}

mcnp_pstudy -i wval2p.txt -setup -run

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





Advanced Topics

Tied parameters

Concurrent jobs

Parameter Expansion (1)

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

Example:

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

Outer: all combinations, 16 cases

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

Inner: 2 cases

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

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

Parameter Expansion (2)

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

Example:

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

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

Example:

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

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

Example:

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

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

Example:

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

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

Parameter Expansion (3)

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

Example:

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

Cases:

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

Example:

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

Cases:

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

Example:

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

Cases:

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

Example:

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

Cases:

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

Concurrent Jobs (1)

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

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

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

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

Nuclear Criticality Safety
-
Validation - I

Background

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

Orders, Standards, Guides for NCS

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

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

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

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

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

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

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

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

MCNP Verification & Validation Suites

Verification Suites

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

Validation Suites

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

Background

Establishing Subcriticality

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

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

Validation: Definitions (1)

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

Validation: Definitions (2)

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

Excerpts from ANSI/ANS - 8.24-2007

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

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

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

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

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

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

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

(1) a review of the benchmark applicability;
(2) a review of the input files and output files to ensure accurate modeling and adequate convergence;
(3) a review of the methodology, and its use, for determining bias, bias uncertainty, and margins;
(4) concurrence with the validation applicability.

Upper Subcritical Limit

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

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

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

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

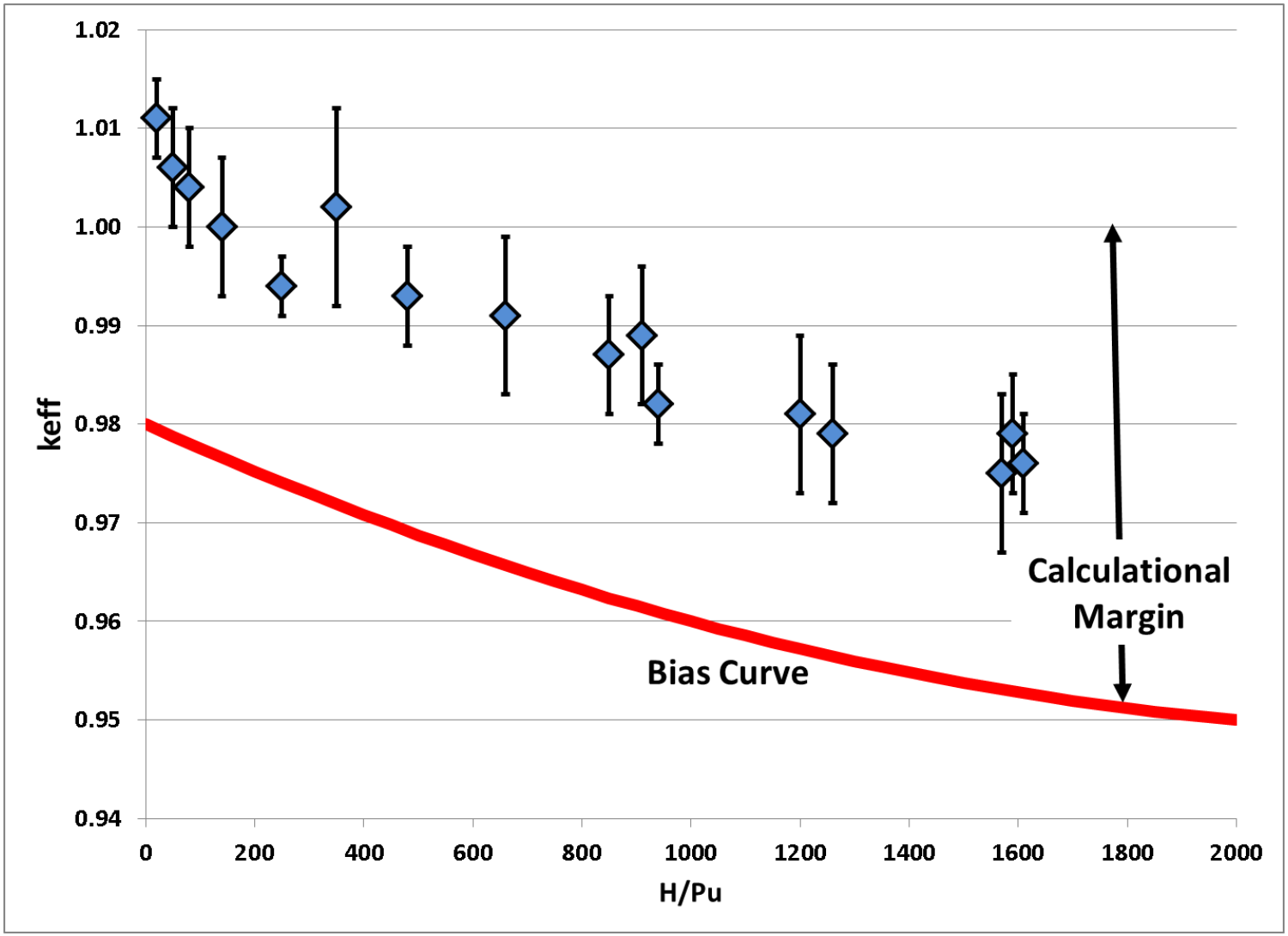
- The bias and bias uncertainty are at some confidence level, typically 95% or 99%.
  - These confidence intervals may be derived from a normal distribution, but the normality of the bias data must be justified.
  - Alternatively, the confidence intervals can be set using non-parametric methods.

Calculational Margin

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

Calculational Margin Example

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



Margin of Subcriticality

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

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

Potential Bias Cause

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

Selection of Benchmarks

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

Selection of Benchmarks

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

How do NCS analysts develop engineering judgement?

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

Selection of Benchmarks

- **Identify the parameters that correlate experiments to the system or process being analyzed in the criticality safety evaluation (ANSI/ANS-8.24 5.1)**

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

Selection of Benchmarks

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

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

- **Benchmark models prepared by outside organizations should be evaluated for appropriateness, completeness & accuracy** (ANSI/ANS-8.24 5.9)
  - **ICSBEP handbook cautions against using their input files without review**
  - **Modeling techniques used may not be adequately similar to that used in the criticality safety evaluation models**

Calculating Bias and Bias Uncertainty

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

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

Results Distribution

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

S/U Analysis

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

- **Techniques:**

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

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

S/U Analysis

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

S/U Analysis

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

Comparison of Validation Approaches (Simplified)

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

Documentation and Independent Technical Review

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

Nuclear Criticality Safety Validation – II
-
Using MCNP & Whisper

Los Alamos NATIONAL LABORATORY EST. 1943

NCSP NUCLEAR CRITICALITY SAFETY PROGRAM

Topics

- Whisper
  - Summary, methodology, status
  - Whisper-1.1.0 update
  - Sensitivity profiles
  - Covariance data
  - Correlation coefficients
  - USLs & Validation
- whisper_mcnp
  - Usage, files, output
- whisper_usl
  - Usage, files, output
- Whisper.out
- Conclusions & Discussion

Whisper – Summary

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

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

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

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

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

Whisper Methodology for Validation & USLs

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

- **Using Whisper**

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

  Run Whisper:

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

  ② Bias + bias uncertainty from Extreme Value Theory
  - Based on most-similar Benchmarks selected

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

MCNP6 & Whisper Status

- **MCNP releases by RSICC**

  MCNP6.1 – 2013, production version

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

  MCNP6.2 – 2017, 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"

Introduction (1)

Whisper? Who cares?

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

Introduction (2)

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

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

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

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

Traditional methods provide a check on S/U methods

S/U approach to automated benchmark selection is quantitative, physics-based, & repeatable. Provides a check on traditional selection

Traditional methods use $MOS_{data+code}$ of 2-5%.
Quantitative, physics-based, repeatable $MOS_{data+code}$ from S/U usually smaller

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

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

Whisper-1.1.0 Update

- **Whisper code** **1.0.0 → 1.1.0 update**
  - **Robust numerics, to avoid memory problems on Mac & PC**
    - Explicit threaded loops, to replace many instances of F90 matrix operators
    - Replaced old Linpack coding by modern inline Fortran
    - Additional threading for some slow sections of code
    - No change to any results
  - **Methods**
    - Chi-square & benchmark rejection changed from based on dk to dk/k
    - Some very minor diffs in list of rejected benchmarks
    - For USL, the list of benchmarks used is sorted by weight (or Ck)
  - **Files**
    - up to 256-character filenames
    - printed list of all files in use, full pathnames
    - TOC files now 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-1.1.0 Update

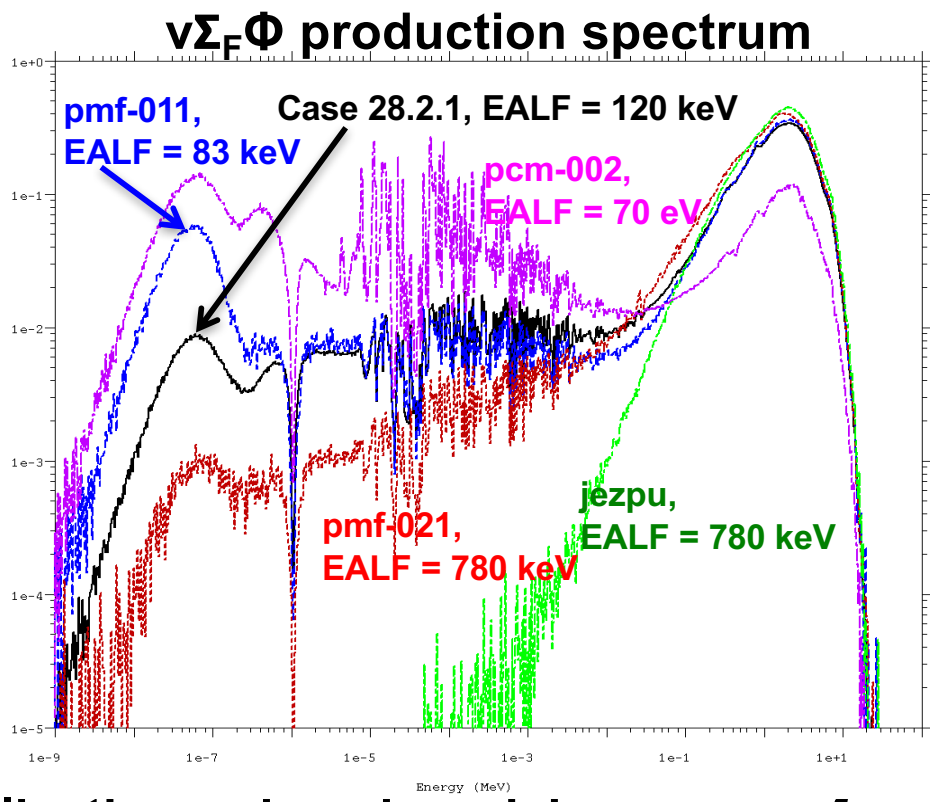
- **Whisper support** **1.0.0 → 1.1.0 update**
  - **Build & test procedures completely revised, to be similar to mcnp6**
  - **Previous C-shell scripts replaced by portable perl scripts**

    whimcnp → whisper_mcnp.pl

    ww → whisper_usl.pl
- **Whisper files** **1.0.0 → 1.1.0 update**
  - **Benchmarks**
    - Updated 27 files, added 15 new files
    - Serious error: pu-comp-mixed-001-005 [ncs]
    - Minor error: mix-met-fast-005-001 [Kahler]
    - Trivial MT card fix: heu-sol-therm-001-001 [Kahler]
    - Tweaks to dimensions: pu-met-fast- 003-103, 018-001, 019-001,
      (typical Δk ~ 0 - 0.02 %) 023-001, 024-001, 027-001, 041-001
      pu-sol-therm-012- 001…013
      pu-comp-mixed-001- 001..004
    - Added benchmarks: pu-met-fast-042- 001…015 [ncs]
  - **Reran 42 benchmarks**
    - new sensitivity profiles
    - new BenchmarkTOC.dat & ExcludedBenchmarks.dat
    - new adjusted covariance data files

Neutronics of Applications & Benchmarks

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



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

Sensitivity Profiles for Criticality Safety

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

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

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

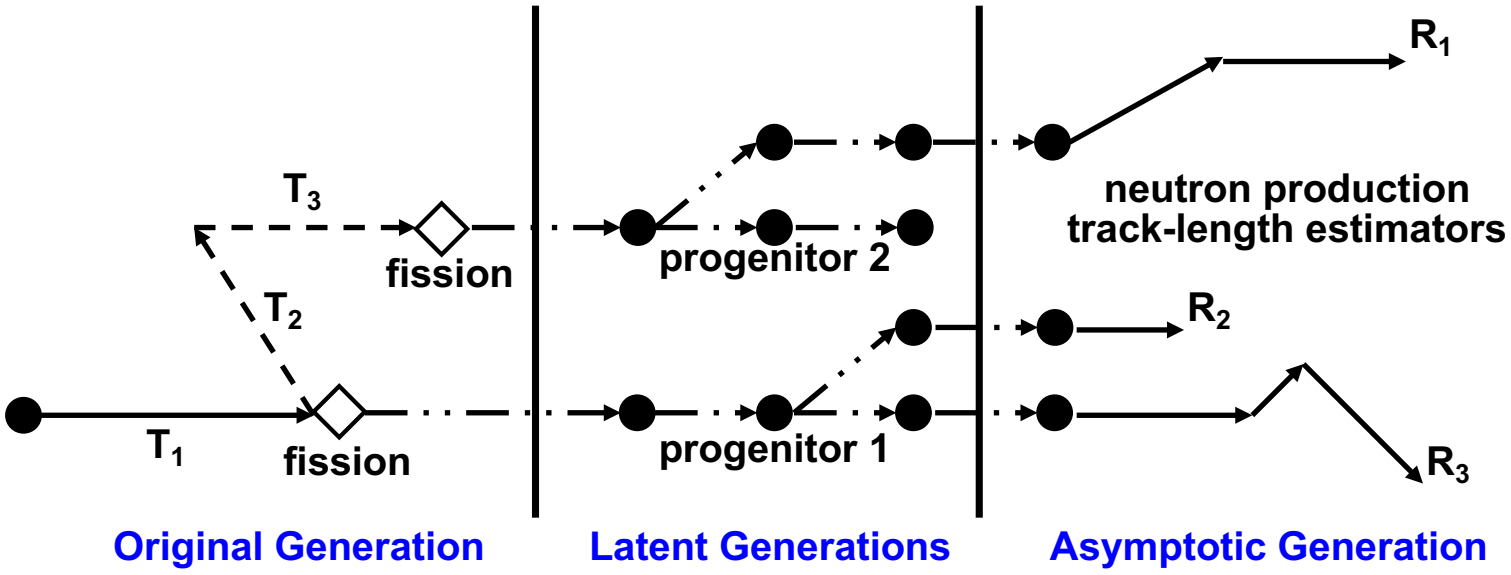
$$S_{k,x} = \frac{x}{k}\frac{dk}{dx} = -\frac{\left\langle \psi^{\dagger}, \left(\Sigma_x - \mathrm{S}_x - k^{-1}\mathrm{F}_x\right)\psi \right\rangle}{\left\langle \psi^{\dagger}, k^{-1}\mathrm{F}\psi \right\rangle}$$

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

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

Sensitivity Profiles – Adjoint Weighting

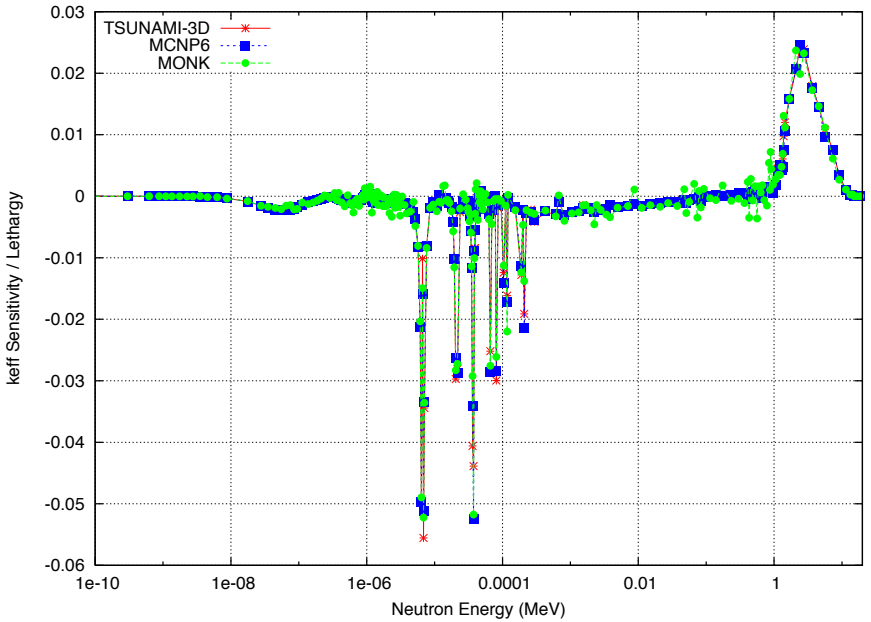
- **Using the Iterated Fission Probability method, MCNP6 can compute adjoint-weighted integrals of any quantity.**
- **MCNP breaks active cycles into consecutive *blocks*:**
  - **Tally scores are collected in original generation, & progenitor neutrons tagged**
  - **All subsequent progeny within the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



Sensitivity Profiles - Examples

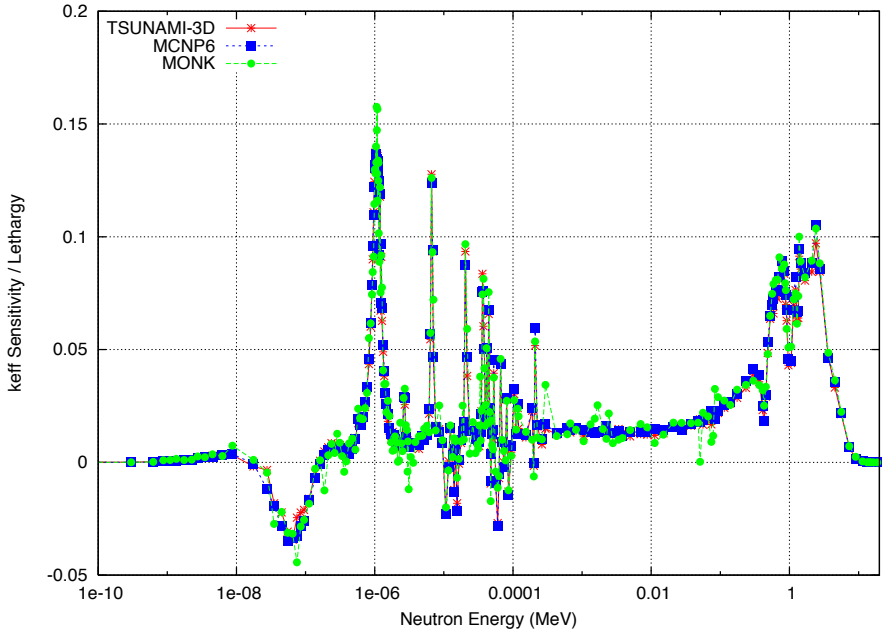
U-238: total cross-section sensitivity

OECD/NEA UACSA Benchmark Phase III.1



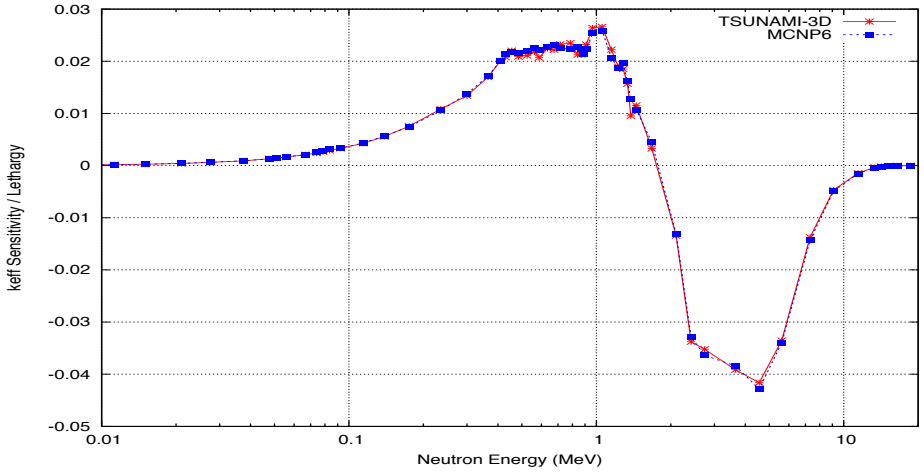
H-1: elastic scattering cross-section sensitivity

OECD/NEA UACSA Benchmark Phase III.1



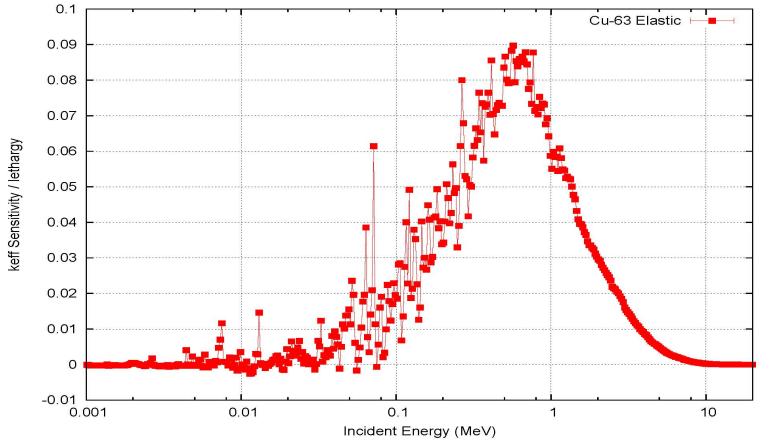
Pu-239: fission chi(E) sensitivity

OECD/NEA UACSA Benchmark Phase III.1



Cu-63: Elastic Scattering Sensitivity

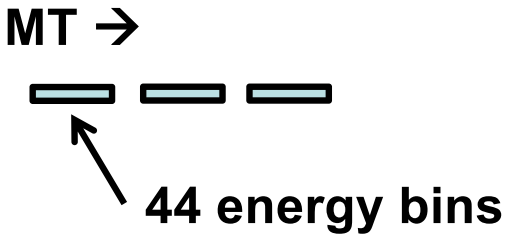
Copper-Reflected Zeus experiment:



Sensitivity Profiles (Vectors)

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

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



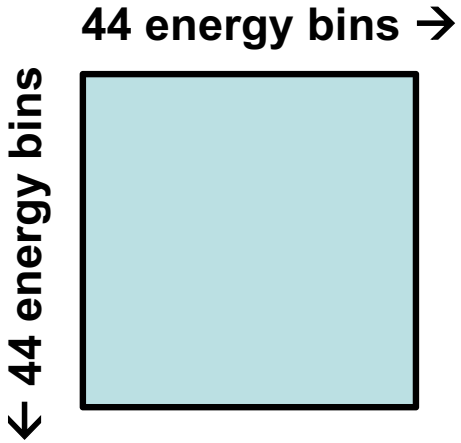
where x is the cross-section for a particular isotope, reaction (MT), & energy bin

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

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

Cross-section Covariance Data

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



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

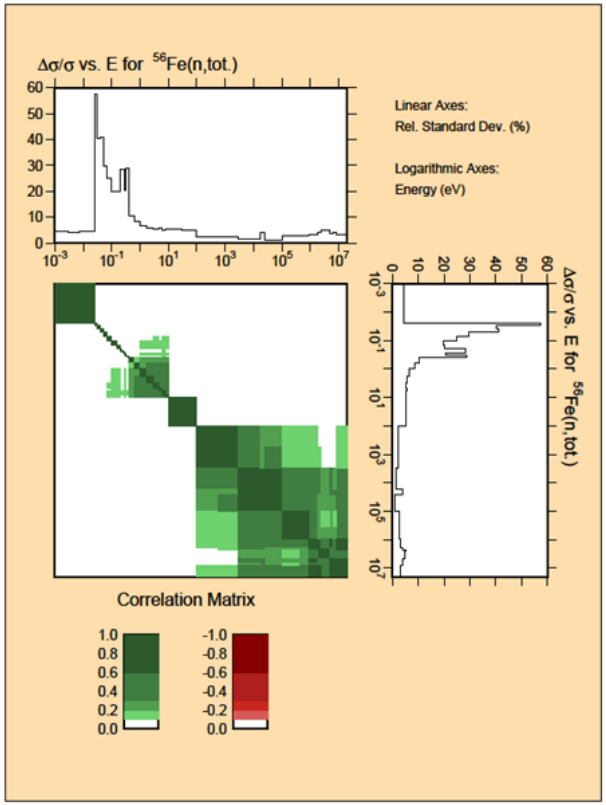


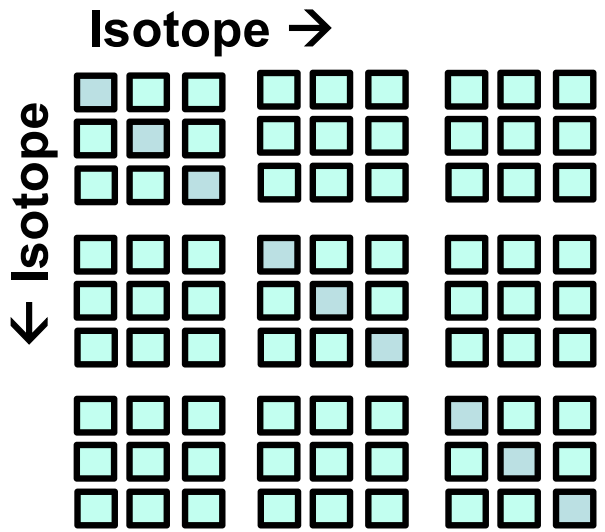
FIG. 9: A typical NJOY-generated plot of ENDF/B-VII.0 data downloaded from the National Nuclear Data Center, BNL, USA.

Cross-section Covariance Data

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

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

C_{xx} =



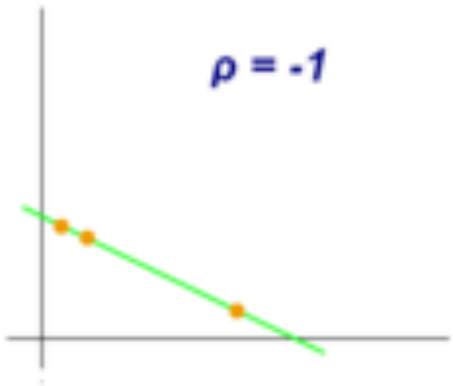
Correlation Coefficient

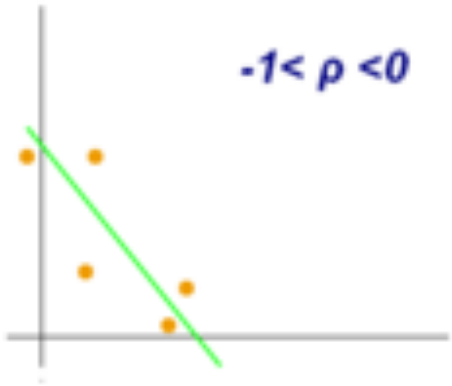
- **Correlation coefficient**
  - **Pearson product-moment correlation coefficient, *r* or *ρ***
  - **A measure of the linear correlation between variables X & Y**

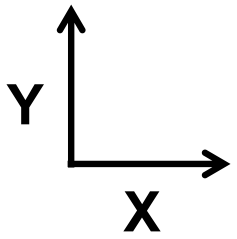
$\rho = +1$ total positive correlation

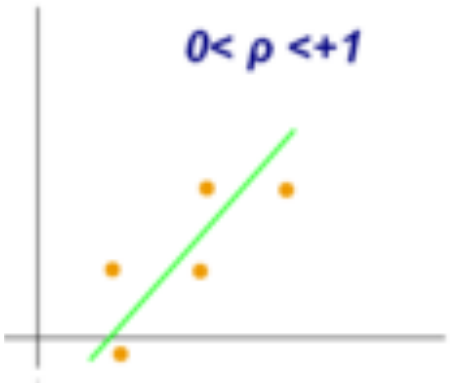
$\rho = -1$ total negative correlation

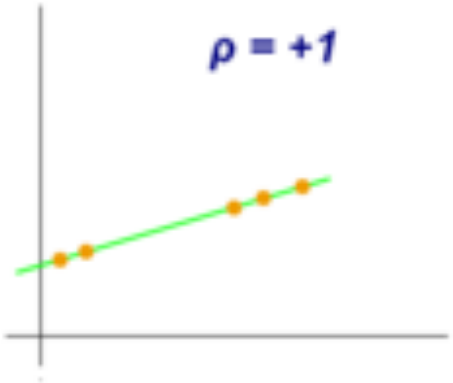
$\rho = 0$ no correlation

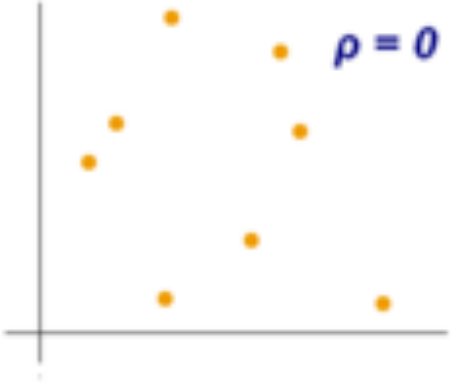












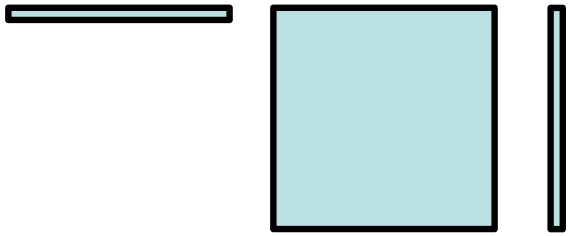
Variance in Keff & Correlation Between Problems

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

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

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

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



= 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**

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

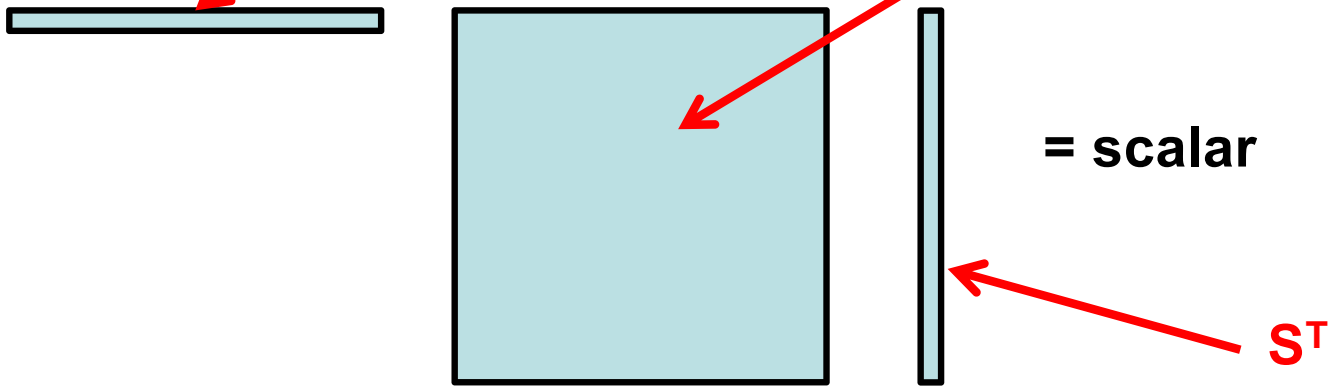
$$Cov_k(A,B) = \vec{S}_A \bar{C}_{xx} \vec{S}_B^T$$

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

Problem-dependent sensitivity vector, S.
Based on flux spectrum, adjoint spectum, nuclear data, problem isotopes, geometry, temperature
Size = G x MT x NI

Nuclear Data Covariances

Size= (G x MT x NI)^2



Upper Subcritical Limit

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

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

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

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

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

Validation

To determine USL for applications

- **Run MCNP6 for applications**
  - Traditional: k_{eff} only
  - S/U-based: k_{eff} & sensitivity profiles
- **Select benchmarks similar to applications**
  - Traditional: Expert judgment
  - S/U-based: Select benchmarks with highest C_k's
- **Statistical analysis**
  - Standard statistical methods, determine bias & bias-uncertainty using the set of selected benchmarks
- **Determine appropriate $MOS_{data,code,applicability}$**
  - Traditional: Expert judgment, usually 2% or 5%, more if warranted
  - S/U-based: Use GLLS to estimate MOS_{data}, code-expert for MOS_{code}
- **Determine USL**
  - Is $k_{application} + 2\sigma < USL$?

Whisper

Whisper Methodology

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

- **Whisper – Determine Benchmark c_k's**
  - For each benchmark B_J, determine $c_k^{(J)}$ correlation coefficient between A & B_J

- **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**
  - Margin for nuclear data uncertainties
  - Margin for unknown code errors

Using Whisper for Validation

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

- **To use Whisper for validation:**

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

  ② **Run Whisper, using the whisper_usl script**

Whisper-1.1.0 – Batch Job

To try it, on Moonlight HPC front end:

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

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

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

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

cd WTEST

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

- **Submit batch job file**

```
msub   job.txt
```

Whisper-1.1.0 – Interactive

To try it, on Moonlight HPC:

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

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

  - **csh, tcsh:**

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

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

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

Using whisper_mcnp (1)

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

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

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

  - **Creates files & dirs:**

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

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

Using whisper_mcnp (2)

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

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

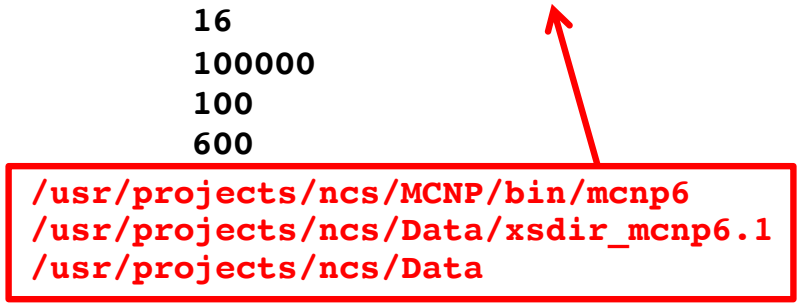
  - **Note that there are large numbers of neutrons/cycle & cycles for the KCODE input. While it may be tempting to reduce these to get shorter runs, that is discouraged since it is important to achieve reasonable statistical uncertainties on the sensitivity profiles for a large number of reactions, isotopes,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

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

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

whisper_mcnp.pl - Usage

```
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.

Nuclear Criticality Safety
-
Validation - III
-
Using Whisper
Examples for NCS Analysts

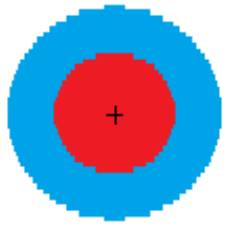
Examples using Whisper

- **Pyrochemical Processing**
  - Example 1: Typical computational model: ingot
  - Example 2: Geometry: Annular
  - Example 3: Material: Pu-NaCl
  - Example 4: Reflection: Ta
  - Example 5: Moderation: Oil
- **General Studies**
  - Example 6: “Revisiting a Practical Application of the Single-Parameter-Subcritical-Mass Limit for Plutonium Metal with Whisper”
  - Example 7: Critical-mass curves and USL-mass curves comparison

Example 1
–
4.5 kg Pu Ingot, varying H/D

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

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



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

    mcnp6 i=wval1.txt
  - **Copy wval1.txt to wval1p.txt, then insert directives for mcnp_pstudy**
    - Define list for HD:

      `c @@@ HD = 0.5 1.0 1.5 2.0 2.5 3.0`
    - For a given H/D, compute Pu radius, then other dimensions

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

    - Use parameters for dimensions & location of KSRC point

Example 1 - wval1: 4.5 kg Pu Ingot, varying H/D (2)

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

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

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

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

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

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

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

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

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

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

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

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

```
whisper_usl.pl
```

use whisper_usl to run Whisper & determine USL for each case

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

wval1, H/D = 1

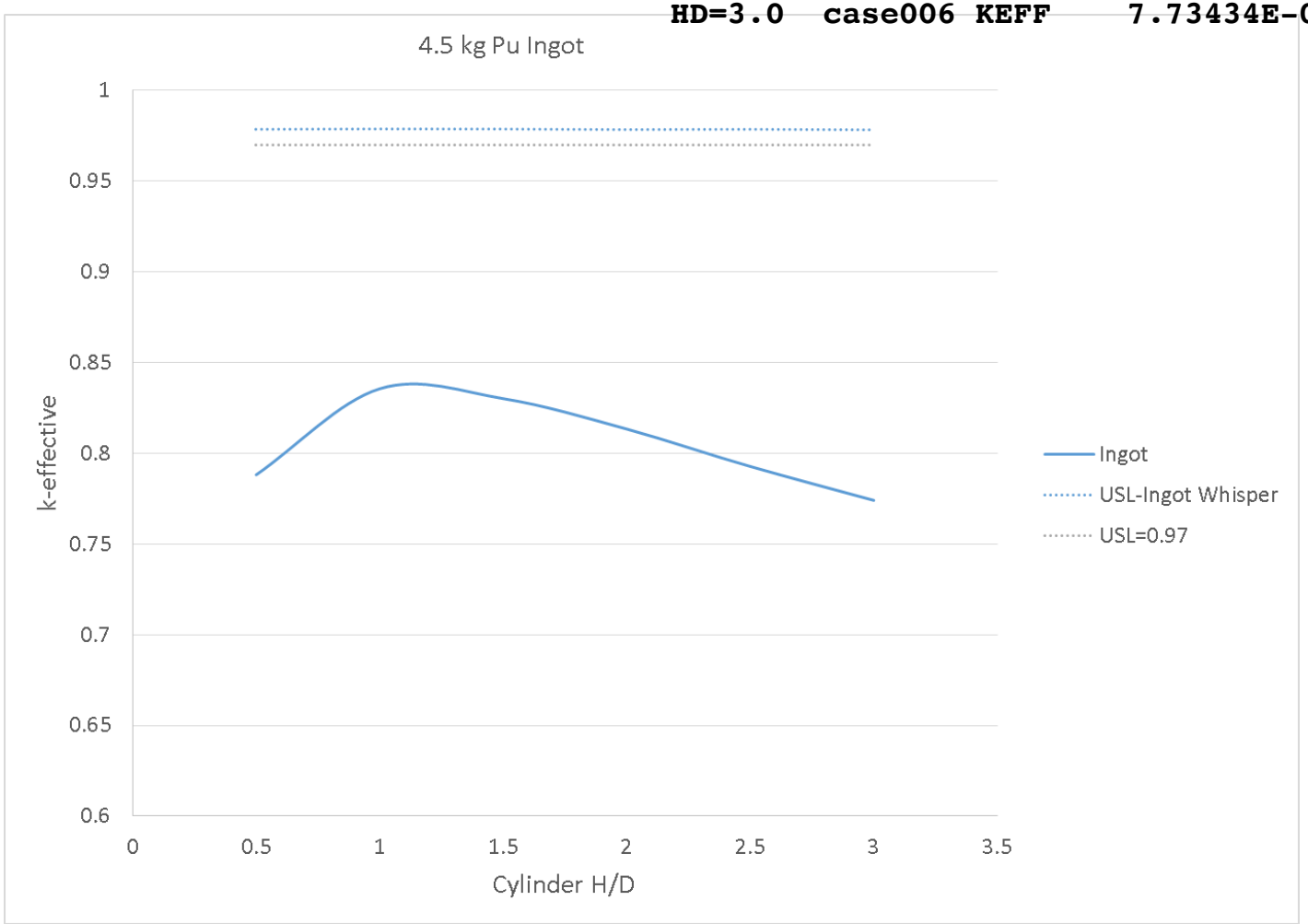
```
mcnp6 i=wval1.txt
```

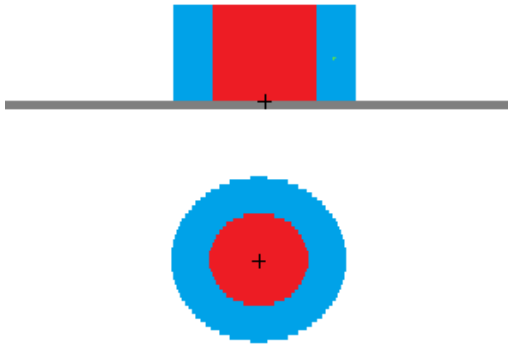
k = 0.83491 (41)

wval1p, varying H/D

```
mcnp_pstudy -i wval1p.txt -setup -run
```

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





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

MCNP6-Whisper Results

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

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

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

Traditional Validation Results:

USL = 0.99-MOS-AoA = 0.97 - AoA

Example 2
–
4.5 kg Pu Annulus, varying H & R_{in}

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

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

From a typical traditional validation report

Parameter	Area of Applicability
Fissile Material	^{239}Pu
Fissile Material Form	Pu Metal, PuO_2, and $Pu(NO_3)_4$
H/^{239}Pu	$0 \leq H/^{239}Pu \leq 2807$
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 the benchmarks that are relevant (similar) to the ring?**

5.3 Metallic units

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

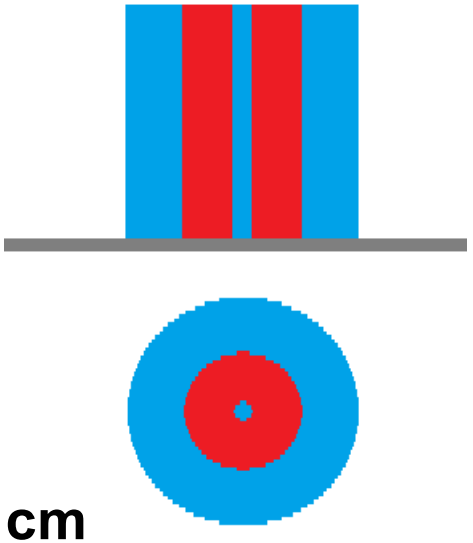
Table 3 – Single-parameter subcritical limits for metal units

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

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 - wval2: 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 - wval2: 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 - wval2: 4.5 kg Pu Annulus, varying H & R_{in} (4)

```
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
```

worm file (1 of 3) for parameter studies

```
wval2_worm:  4500 g Pu metal, various H/D
c
c  pu container1 definition:
c   pu mass     <pu_mas=4500>
c   pu density  <pu_den=19.86>
c   pu volume   <pu_vol=pu_mas/pu_den>
c   pu height   <pu_hei=6.608> [height of ingot H/D=1]
c   inner rad   <in_rad=bit,0.1,0.2,0.4,0.6,0.8,1,2.54>
c   ring vol    <ring_vol=pu_vol+(pi*pu_hei*in_rad^2)>
c   radius      <ring_rad=(ring_vol/pi/pu_hei)^(1/2)>
c
c  reflector definition:
c   reflector thickness = <r_thk = 1*in>
c   reflector radius    = <r_rad = ring_rad+r_thk|3.6>
c   reflector height    = <r_hei=pu_hei>
c
1 3 -1.0          -1       imp:n=1
2 1 -19.86        +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 <pu_hei|3.6> <in_rad|3.6>
 2 rcc 0 0 0 0 0 <pu_hei|3.6> <ring_rad|3.6>
11 rcc 0 0 0 0 0 <r_hei|3.6> <r_rad|3.6>
20 rcc 0 0 <-1*in|3.6> 0 0 <36*in|3.6> <36*in|3.6>
30 rcc 0 0 <-0.25*in|3.6> 0 0 <0.25*in|3.6> <30*in|3.6>

kcode 10000 1.0 150 500
ksrc   <in_rad+0.5> 0 <pu_hei/2|3.6>
c
..... etc.
```

2 more files like this, for H/D=2 & H/D=3

Example 2 - wval2: 4.5 kg Pu Annulus, varying H & R_{in} (5)

wval2

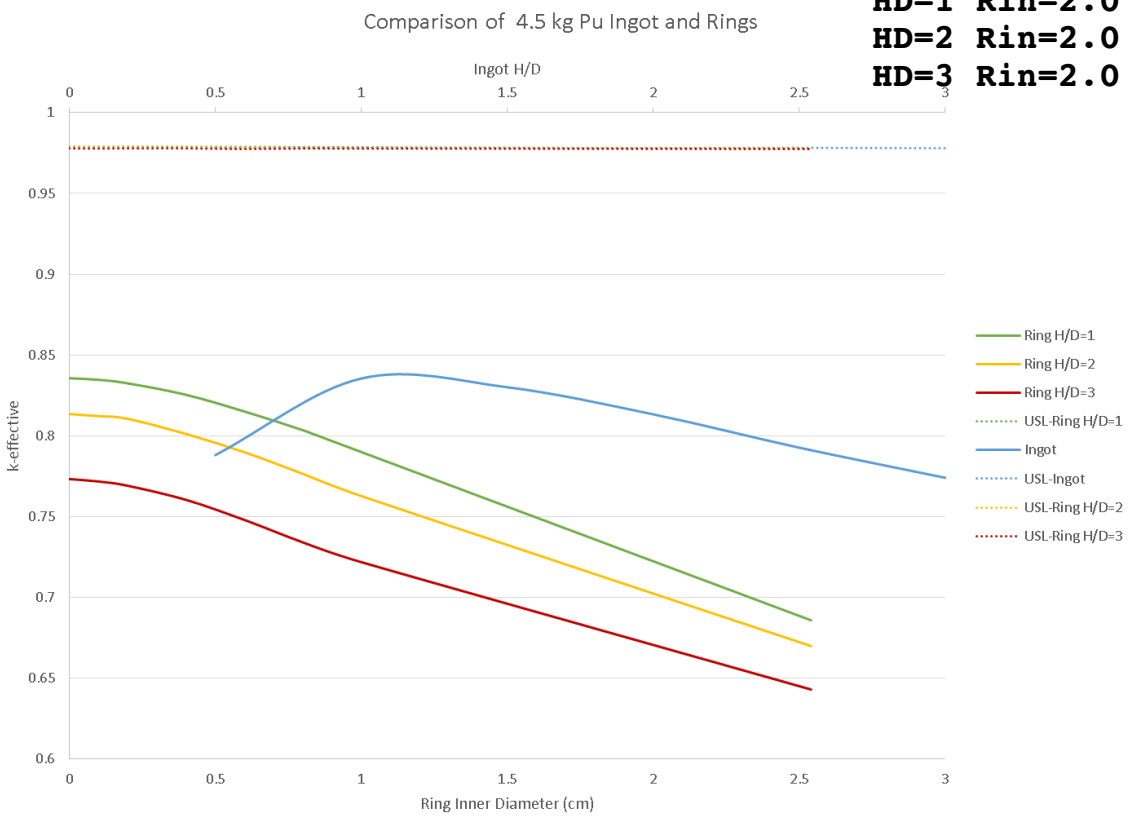
```
mcnp6 i=wval2.txt
```

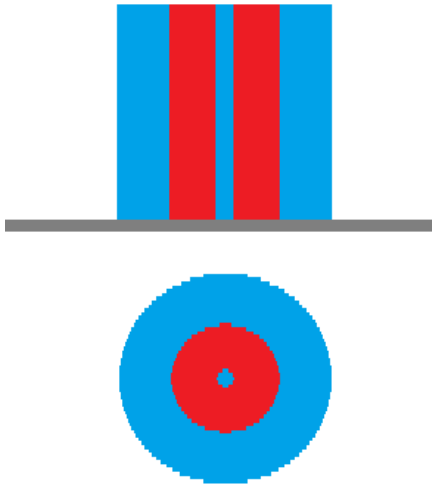
k = 0.83413 (42)

wval2p, varying H & R_{in}

```
mcnp_pstudy -i wval2p.txt -setup -run
```

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





Example 2 - wval2: 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

Traditional Validation Results:

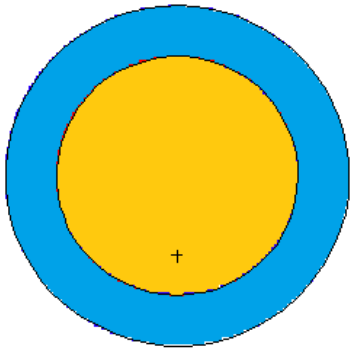
USL = 0.99-MOS-AoA = 0.97 - AoA

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 – wval3: 4.5 kg Pu-NaCl Mixture (1)

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



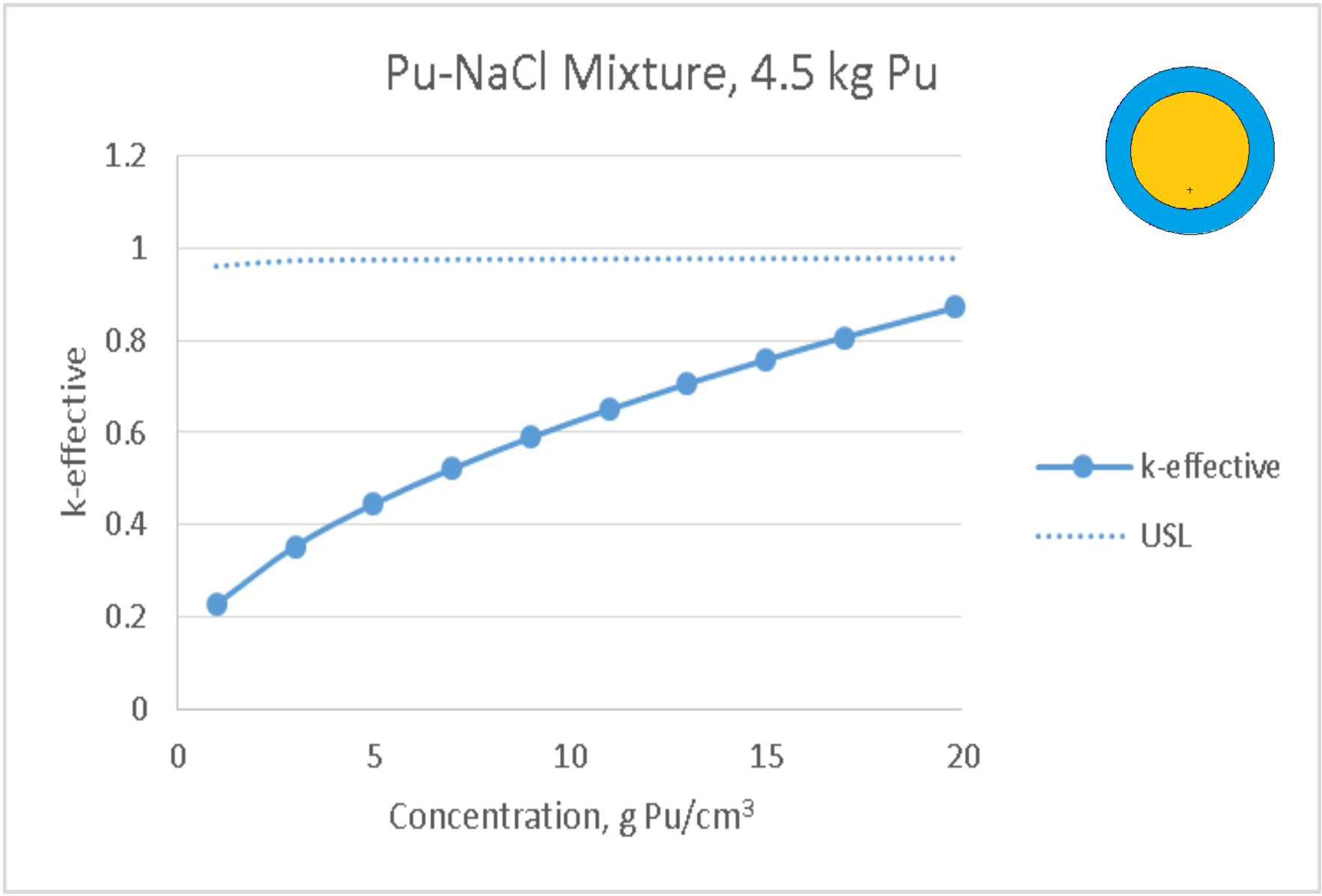
Example 3 – wval3: 4.5 kg Pu-NaCl Mixture (2)

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

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

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

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

 1  so   RADIUS
 2  so   OUTER_H2O

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

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

MCNP6-Whisper Results



Example 3 – wval3: 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

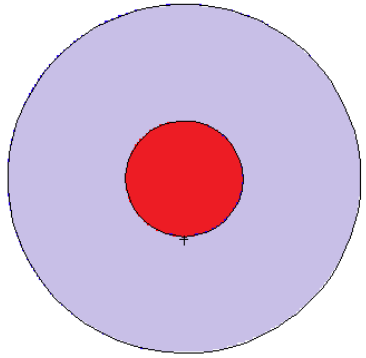
Parameter	Area of Applicability
Fissile Material	^{239}Pu
Fissile Material Form	Pu Metal, PuO_2, and $Pu(NO_3)_4$
$H/^{239}Pu$	$0 \leq H/^{239}Pu \leq 2807$
Average Neutron Energy Causing Fission (MeV)	$0.003 \leq ANECF \leq 1.935$
^{240}Pu	0 to 42.9 wt% ^{240}Pu
Moderating Materials	none, water, graphite, 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 wval4: 4.5 kg Pu Sphere, Ta-reflected (1)

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



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

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

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

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

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

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

  1 so   3.7857584
  2 so   11.405758

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

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

  1 so    R_INNER
  2 so    R_OUTER

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

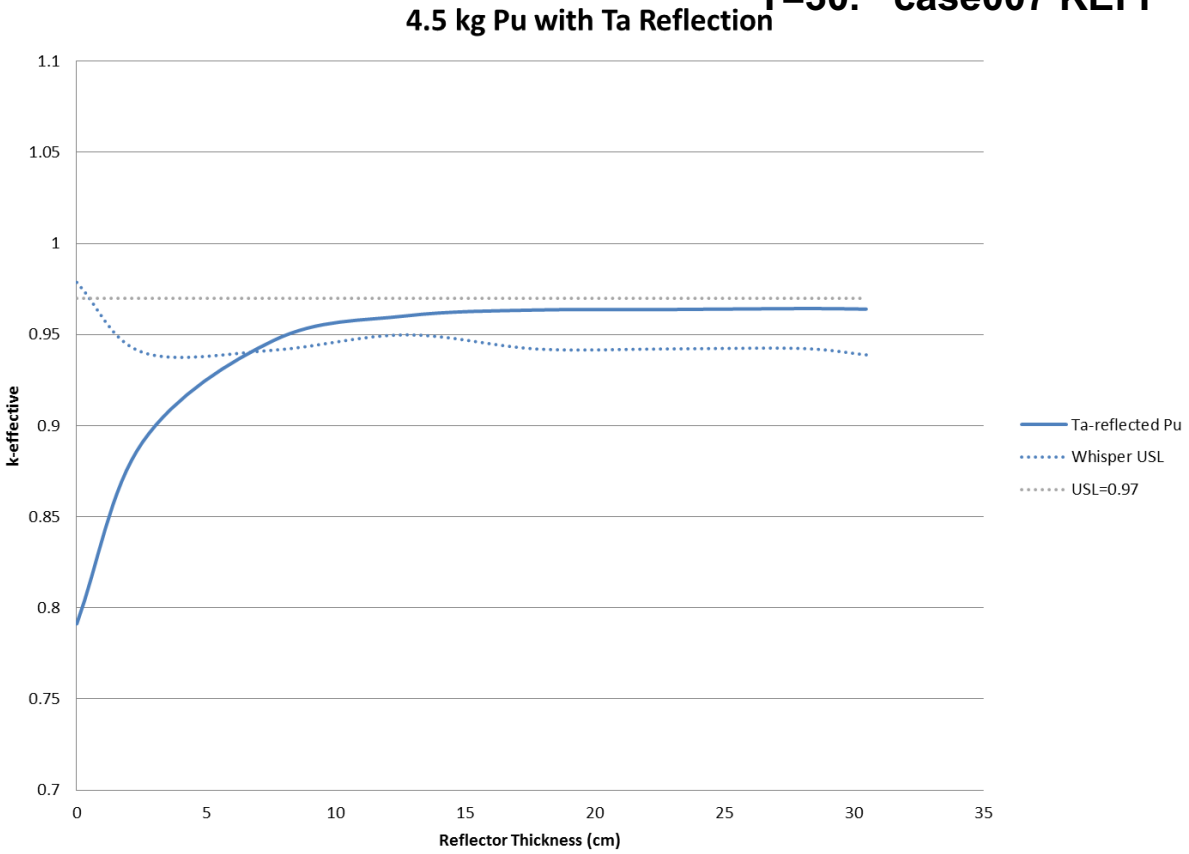
Example wval4: 4.5 kg Pu Sphere, Ta-reflected (3)

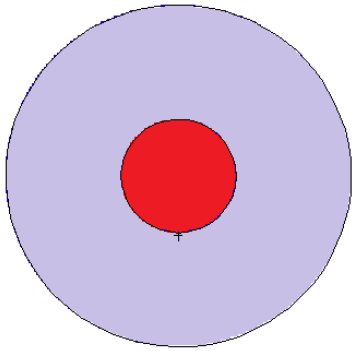
wval4, thick=7.62
mcnp6 i=wval4.txt

k = 0.94638 (41)

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

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





Example 4: Ta-reflected Pu

MCNP6 and Whisper Results

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

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

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

Trouble ! Benchmarks are not very similar to application

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

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

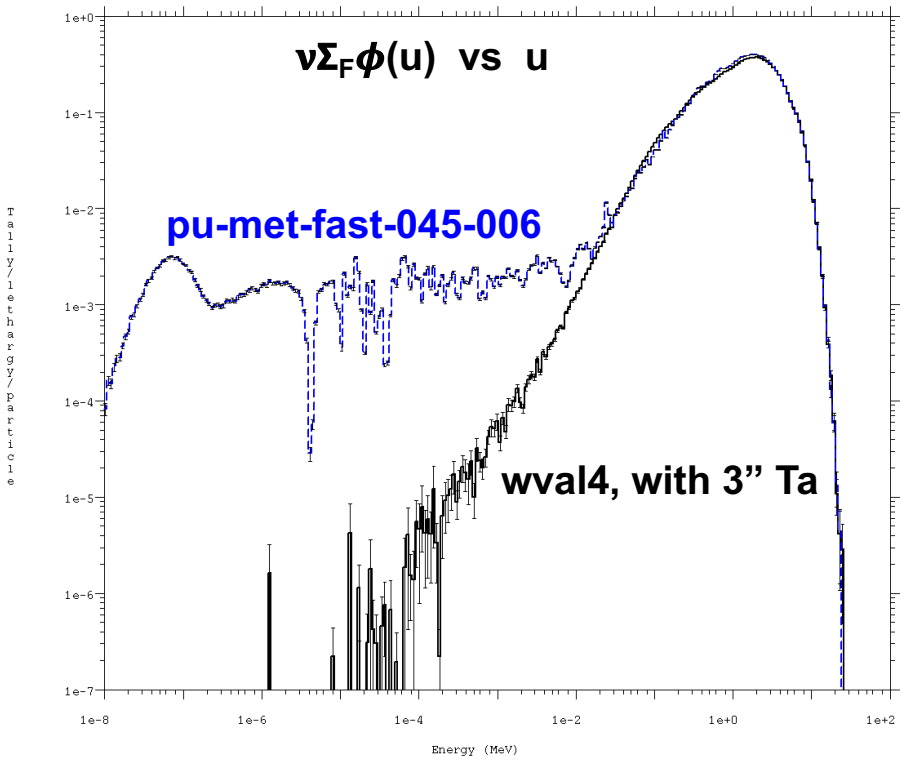
- **None of the benchmarks appear to have the same neutronics as the application**
  - **Largest C_k in the Whisper example output is 0.64 – very low**
  - **Guidance from ORNL Scale/Tsunami developers:**

$0.95 < C_k \rightarrow$ great

$0.90 < C_k < 0.95 \rightarrow$ good

$C_k < 0.90 \rightarrow$ not so good

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



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

Example 5
–
4.5 kg Pu Sphere, Oil moderated

Example 5: Oil-Moderated Pu

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

From a typical traditional validation report

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

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

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

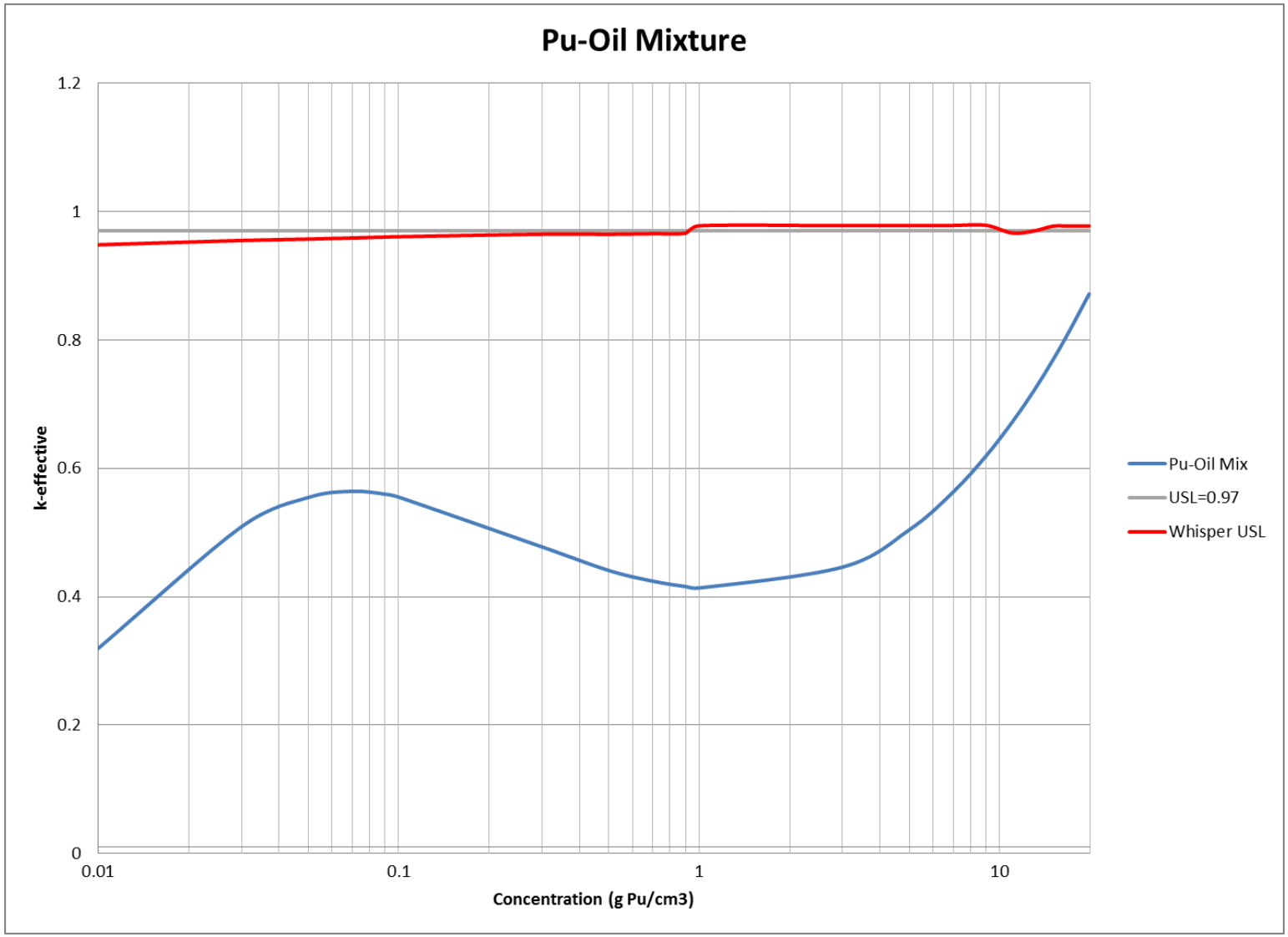
Example 5: Oil-Moderated Pu

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

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

1   so     10.2417609488294
2   so     12.7817609488294

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

Example 5: Oil-Moderated Pu

- **MCNP6 and Whisper Results**



Example 5: Oil-Moderated Pu

MCNP6 and Whisper Results

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

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

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

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

.....

Traditional Validation Results:

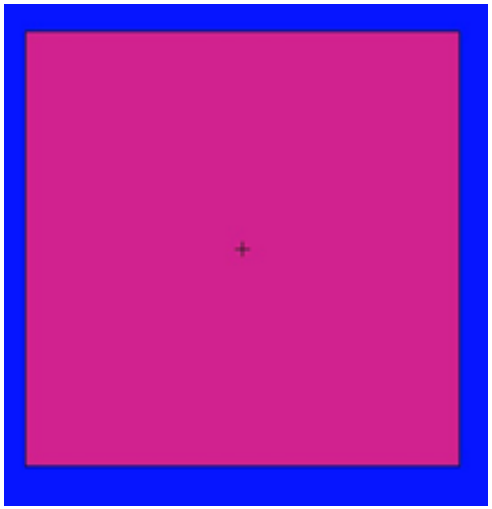
USL = 0.99-MOS-AoA = 0.97 - AoA

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

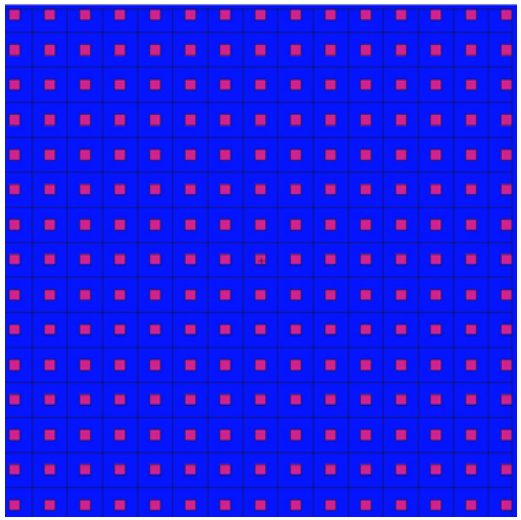
Example 6: Revisiting a Practical Application of the SPSL for Pu Metal

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

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



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



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

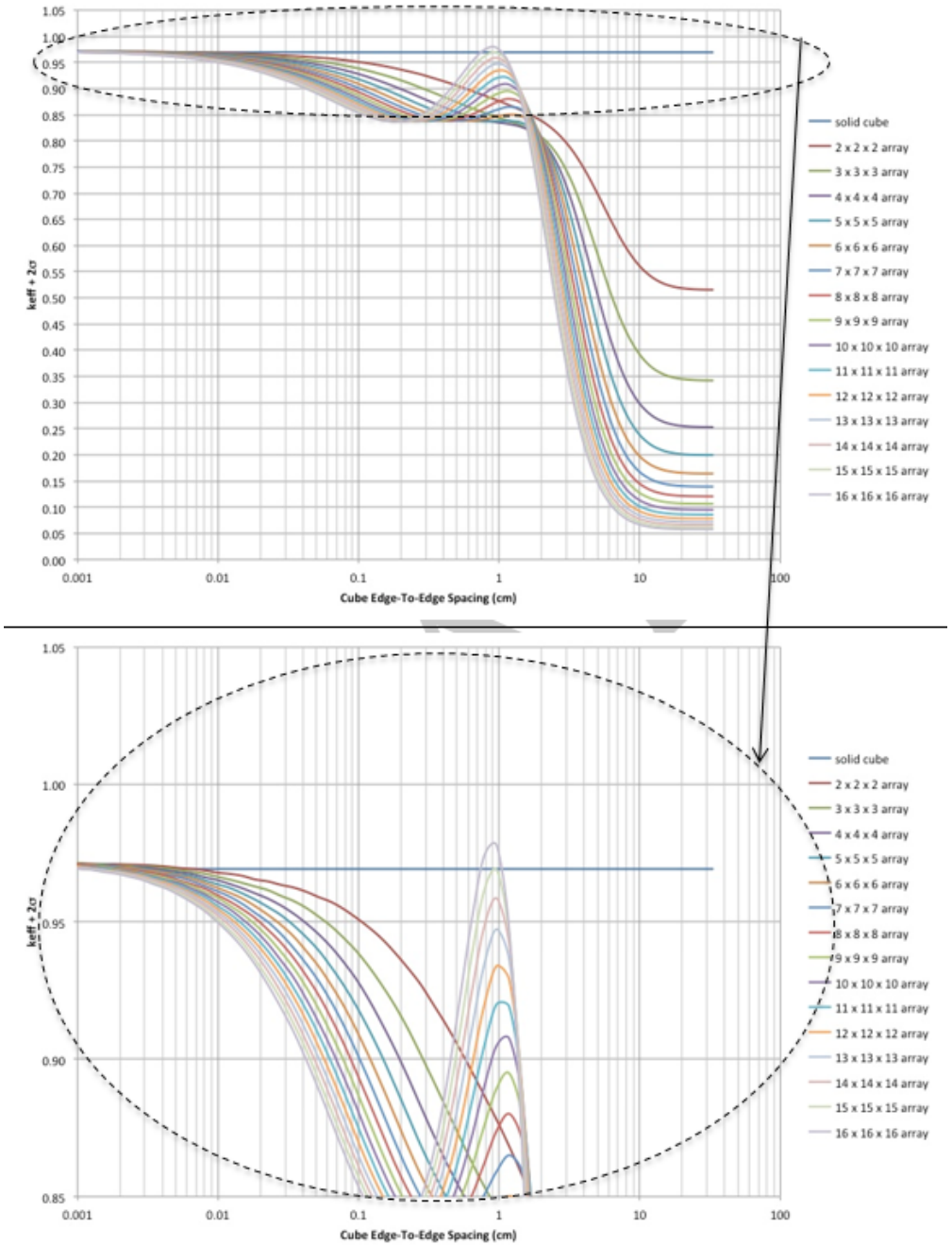
Example 6: Revisiting a Practical Application of the SPSL for Pu Metal

5.3 Metallic units

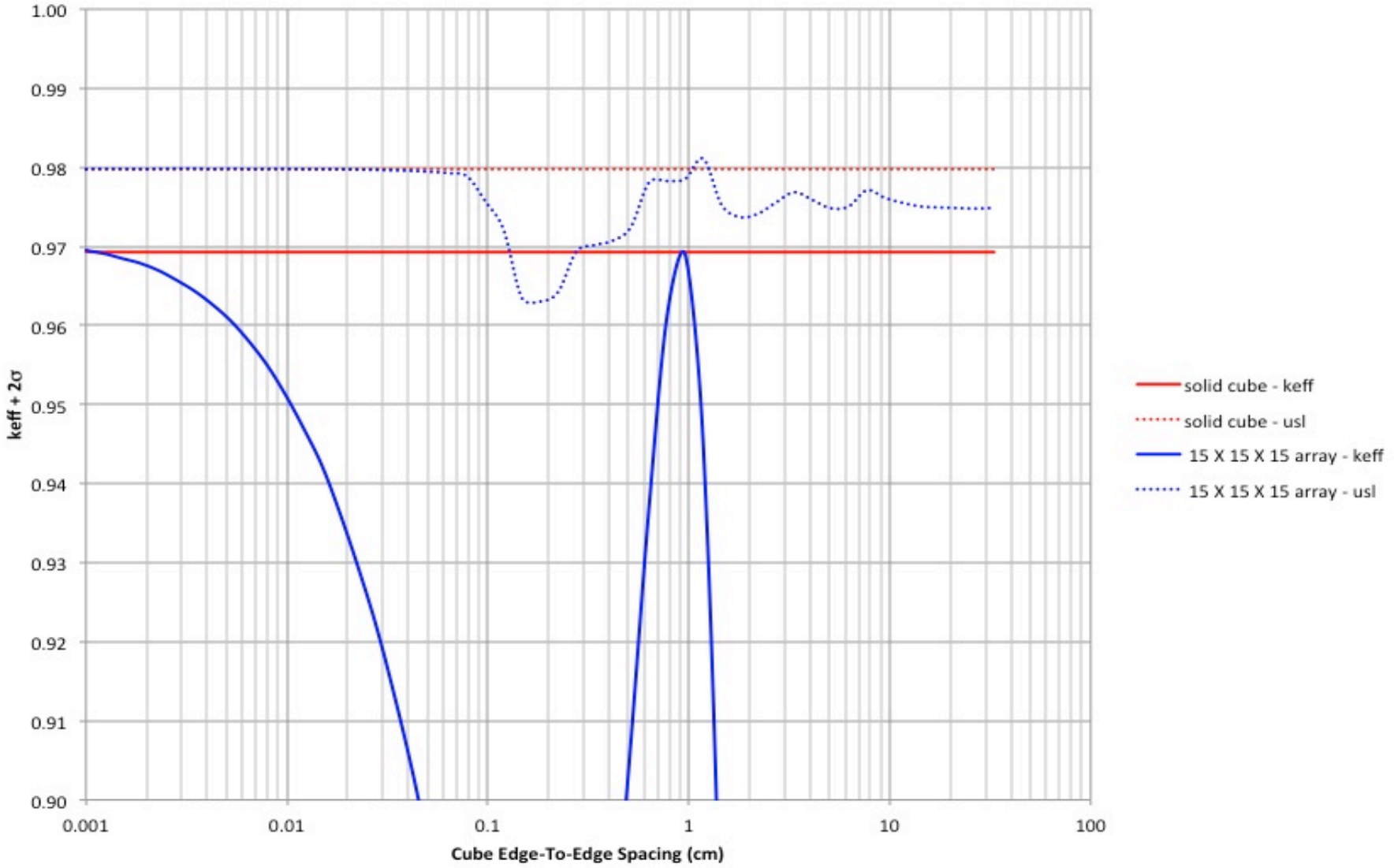
The enrichment subcritical limit for uranium and the mass subcritical limits given in Table 3 apply to a single piece having no concave surfaces.

Table 3 – Single-parameter subcritical limits for metal units

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

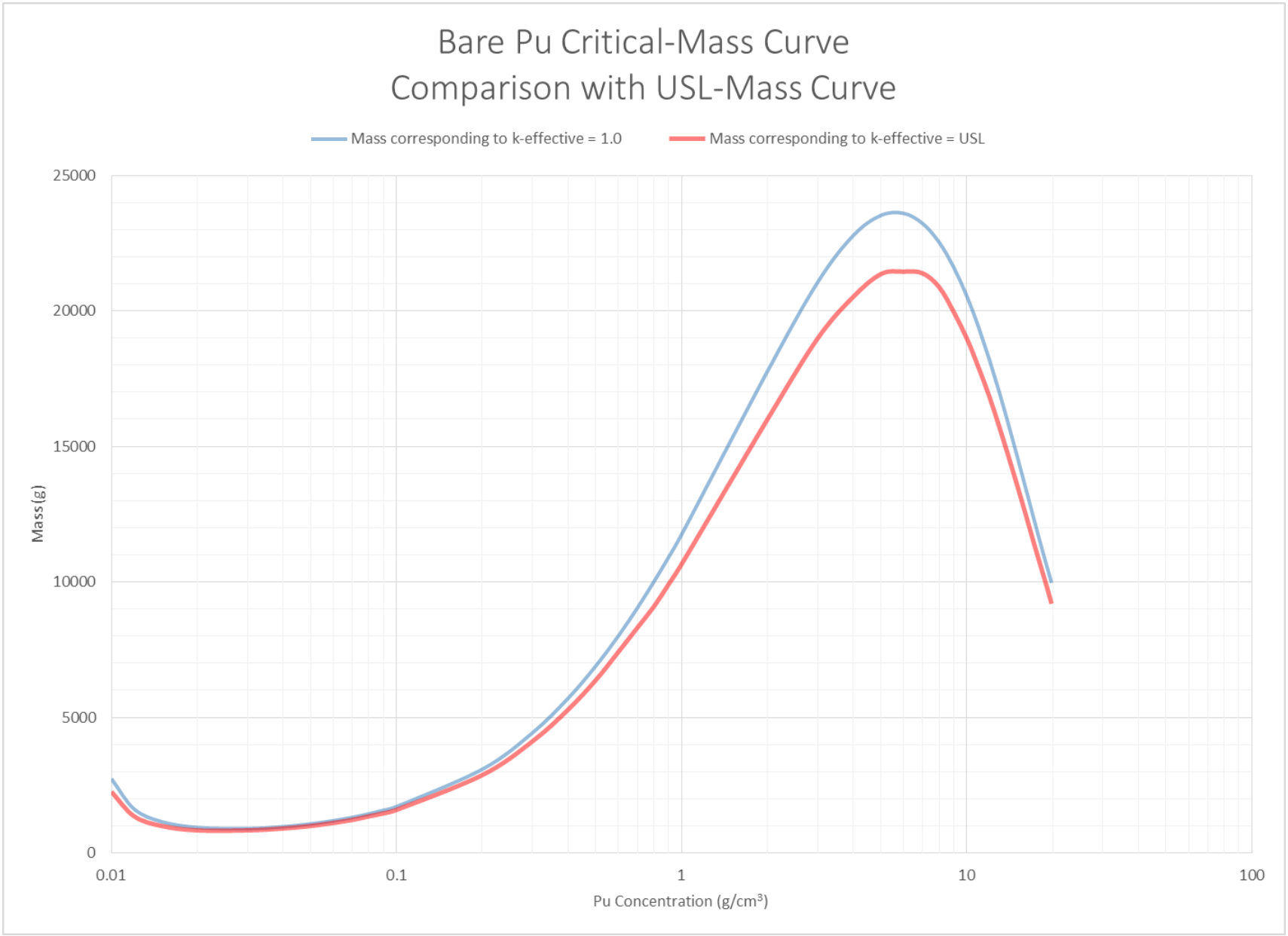


Example 6: Revisiting a Practical Application of the SPSL for Pu Metal



Example 7
–
Critical Mass & USL Curves

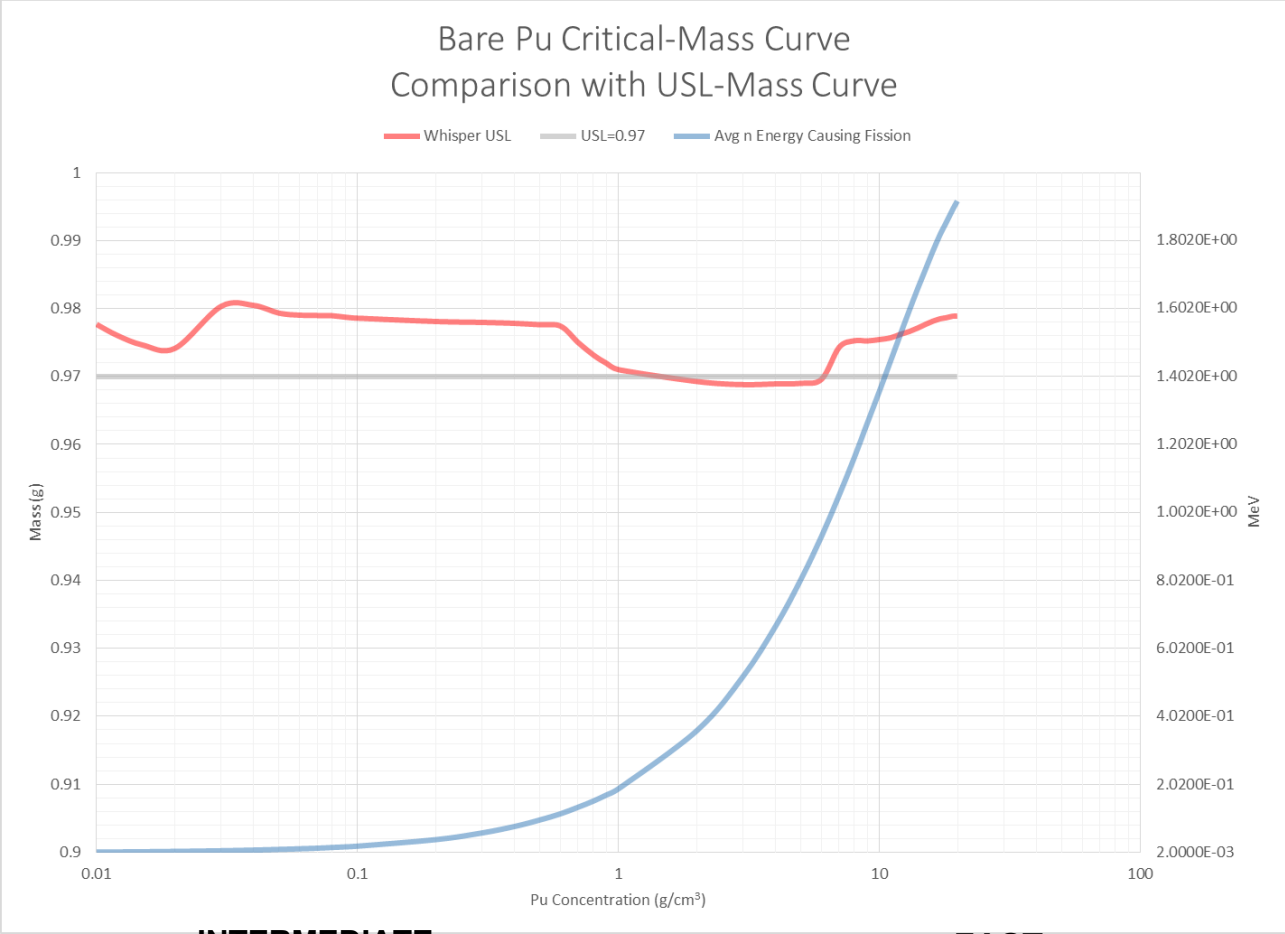
Example 7: Critical-Mass and USL-Mass Curves



Example 7: Critical-Mass and USL-Mass Curves

[ANSI/ANS-8.24 7.2]

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



THERMAL

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

INTERMEDIATE

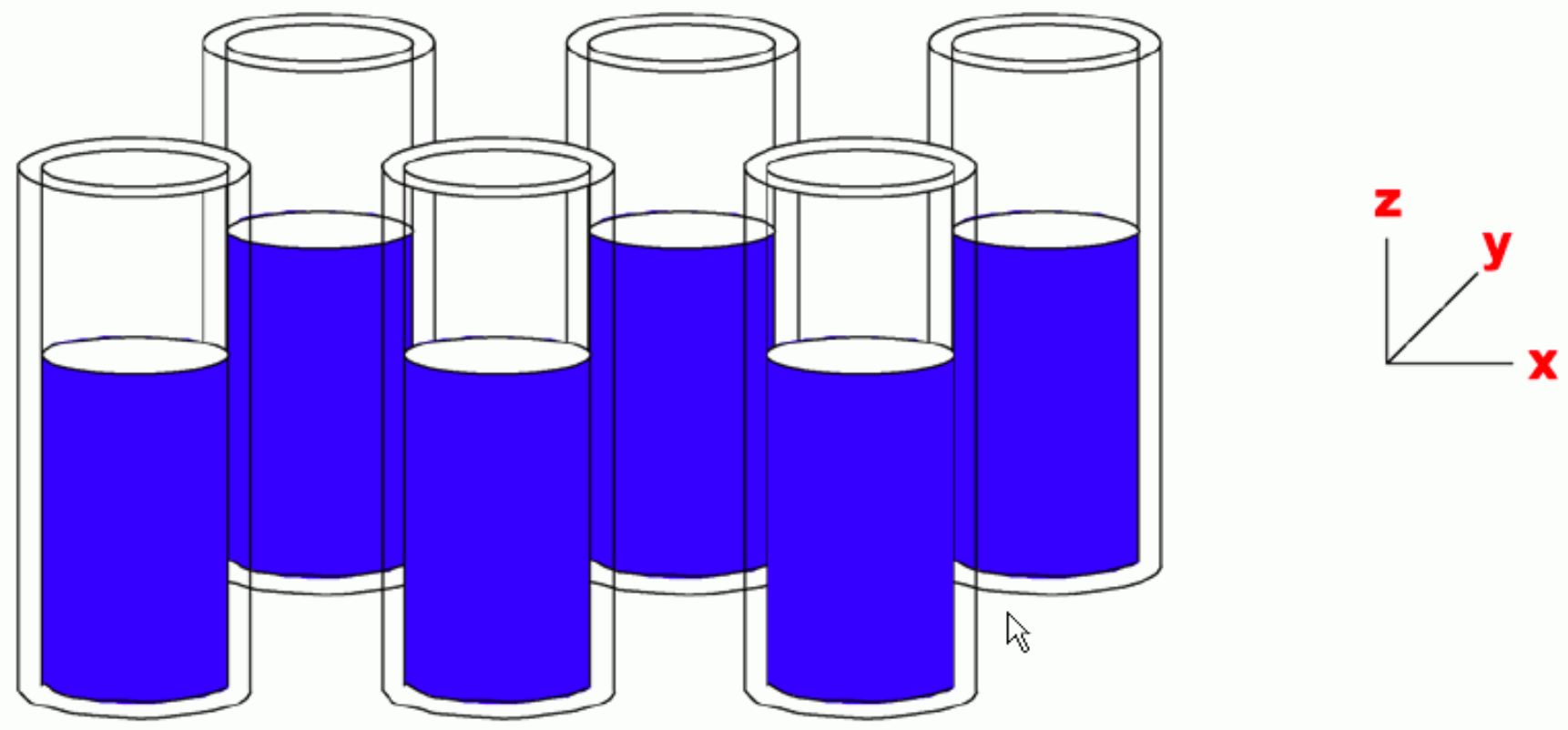
- **Average neutron energy causing fission: 0.519 MeV**
- **% of fissions caused by neutrons: 18%; 55%; 27%**
- **Bias+bias uncertainty: 0.02197**
- **Nuclear data uncertainty: 0.00162**
- **USL = 0.96881**

FAST

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

Case Study #1

3 x 2 Lattice, From Chapter 5 of Crit Primer



Example Problem - puc1 (1)

Plutonium Nitrate Solution
In a Cylindrical tank
--

Sample input file: puc1.txt

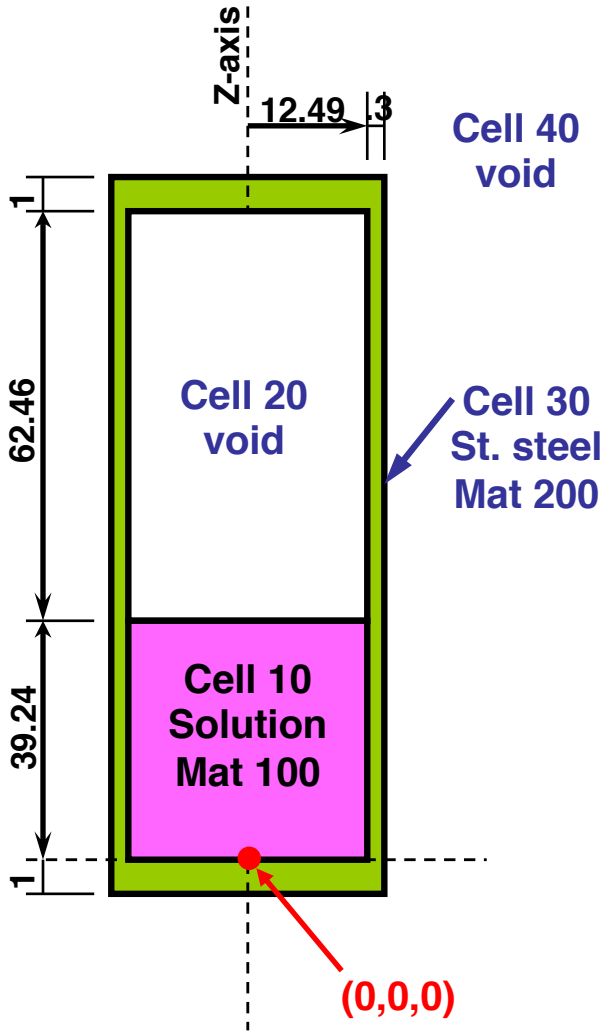
Example Problem - puc1 (2)

Cell 10:

Radius = 12. 49 cm
Height = 39.24 cm
Density = 9.9270e-2 atoms/b-cm

Cell 30:

Tank thickness = 0.3 cm
Bottom thickness = 1.0 cm
Top thickness = 1.0 cm
Inside height = 101.7 cm
Density = 8.6360e-2 atoms/b-cm



Note: Dimensions & material specifications taken from the example used in Section 5.3 of the MCNP Criticality Primer

Example Problem - puc1 (3)

Material 100

```
rho = 9.927e-2

 1001   6.0070e-2
 8016   3.6540e-2
 7014   2.3699e-3
94239   2.7682e-4
94240   1.2214e-5
94241   8.3390e-7

94242   4.5800e-8
```

Material 200

```
rho = 8.636e-2

24050   7.1866e-4
24052   1.3859e-2
24053   1.5715e-3
24054   3.9117e-4
26054   3.7005e-3
26056   5.8090e-2
26057   1.3415e-3
26058   1.7853e-4
28058   4.4318e-3
28060   1.7071e-3
28061   7.4207e-5
28062   2.3661e-4
28064   6.0256e-5
```

Material 100 - Pu-nitrate solution

Material 200 - stainless-steel, for container

Void - outside container, and
inside container above solution

To save typing, just set up the problem geometry (MCNP cells & surfaces), and then:

- Cut & paste DATA cards from file puc1_data_cards.txt in SOLUTIONS folder
- Includes KCODE, KSRC, M100, MT100, M200 cards
- mt100 lwtr card invokes S(α,β) thermal scattering treatment for hydrogen in material 100

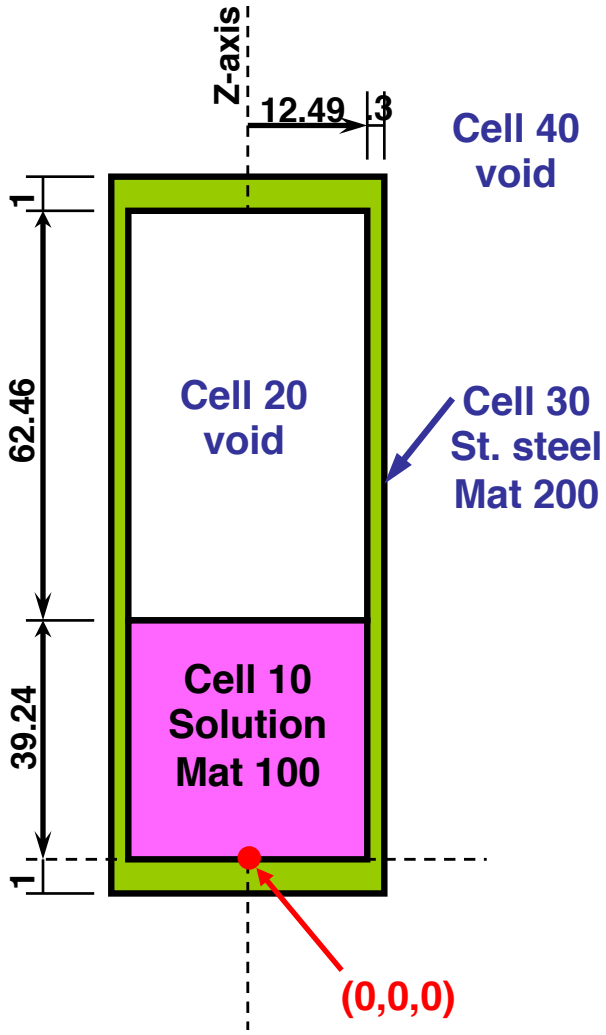
Example Problem - puc1 (4)

Construct a single tank

- Cell 10 – material 100
- Cell 20 – void
- Cell 30 – material 200
- Cell 40 – void

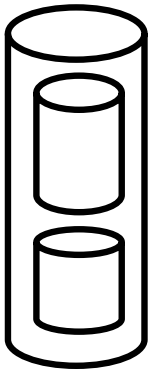
- Use ksrc, center of cell 10
- Use 1000 neutrons/cycle
- Discard 25 cycles, run 100 total
- Don't forget imp:n

- Edit file puc1.txt
- Plot
- Compute keff

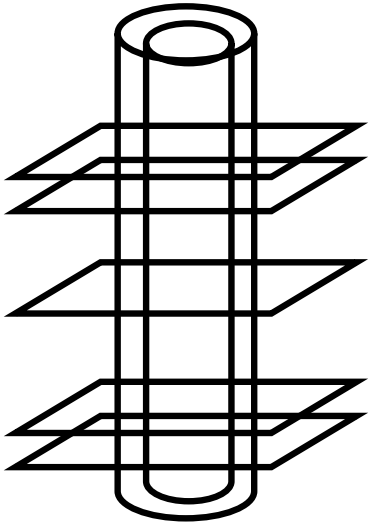


Example Problem - puc1 (5)

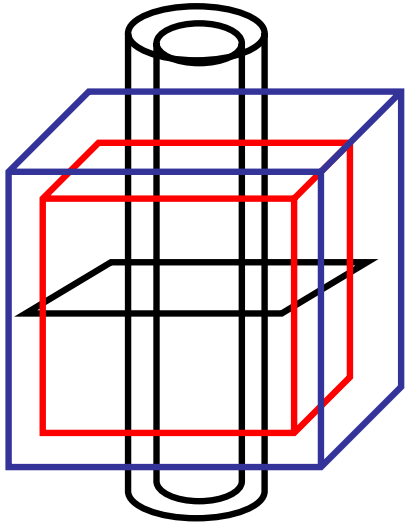
Possible geometry setups



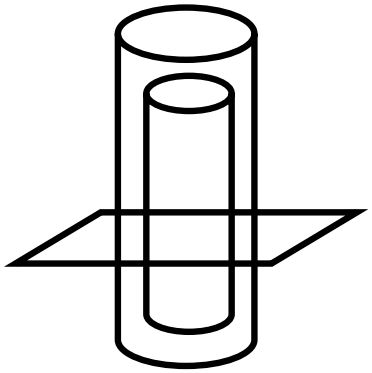
3 RCC bodies



2 infinite z-cylinders
+ 5 planes



2 infinite z-cylinders
+ 2 RPP bodies
+ 1 plane



2 RCC bodies
+ plane

Do it this way
(you'll see why later …)

Example Problem - puc1 (6)

```
puc1 - single cylinder
c    CELL CARDS
10   100    9.9270e-2     -1 -3     imp:n=1       $ solution
20   0                    -1  3     imp:n=1       $ void above solution
30   200    8.6360e-2      1 -2     imp:n=1       $ can
40   0                     2        imp:n=0       $ outer void

c    SURFACE CARDS
1    RCC   0. 0.   0.    0. 0. 101.7    12.49      $ inner can
2    RCC   0. 0. -1.    0. 0. 103.7    12.79      $ outer can
3    pz    39.24                                  $ solution height

c      DATA CARDS (from puc1_data_cards.txt)
kcode    1000   1.0   25   100 $ criticality calc
ksrc     0. 0. 19.62           $ source guess
c
m100          . . .            $ solution material
mt100 lwtr                     $ use h2o  S(a,b)
m200          . . .            $ can material
```

Result:
keff = 0.88778 ± 0.00363

Example puc2 (1)

Problem puc2

Use UNIVERSE for solution & inner void
&
FILL the steel can with that universe

Example puc2 (2)

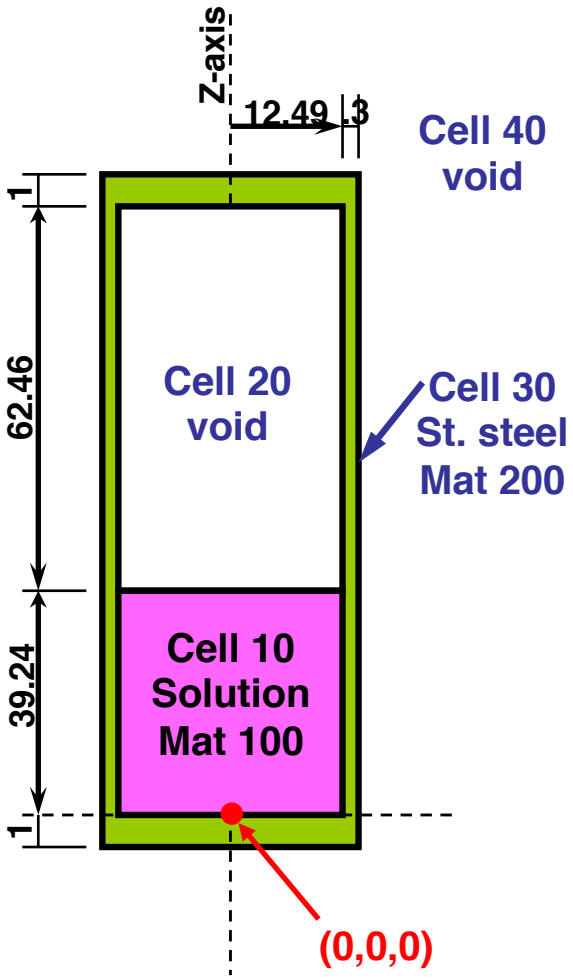
- **Copy file puc1.txt to puc2.txt**

- **Edit file puc2.txt**
  - **Modify definitions of cells 10 & 20:**
    - Identify cells 10 & 20 as **being in universe 1**
    - Remove surface 1 from their definitions
    - Both cells are infinite (in universe 1)
  - **Define a cell (25) for the interior of the container**
    - Bounded by the inner surface of the container (1)
    - **Fill it with universe 1**
    - Don't forget to add imp:n=1

- **Run the problem, with**

```
kcode  1000  1.0   25   100
```

- **Note that the answer is identical to the previous run**



Example puc2 (3)

```
puc2 - single cylinder, using universe & fill
c
C    ----- universe 1 -----
10   100   9.9270e-2        -3        u=1   imp:n=1    $ infinite solution
20   0                       3        u=1   imp:n=1    $ infinite void
C    ----- real world -----
25   0                    -1 fill=1   imp:n=1    $ inside container, filled
30   200   8.6360e-2       1 -2       imp:n=1    $ container
40   0                     2          imp:n=0    $ exterior

1      RCC    0. 0.  0.     0. 0. 101.7      12.49   $ inner
2      RCC    0. 0. -1.     0. 0. 103.7      12.79   $ outer
3      pz     39.24                                  $ solution height

C DATA CARDS from puc1_data_cards.txt
kcode    1000 1.0 25 100
ksrc     0. 0. 19.62
m100     . . .
mt100    . . .
m200     . . .
```

Result:

keff = 0.88778 ± 0.00363

Example puc2 (4)

- **Universe 1 is infinite**
  - Infinite void above surface 3, infinite solution below surface 3
  - Because cells 10 & 20 are infinite, MCNP can't compute their volume, & uses Volume=0 in the output
- **Cell 25 is filled with universe 1**
  - Universe 1 is clipped by the container cell (cell 25)
  - The container (cell 25) must be completely filled by the embedded universe (of course it is, since universe 1 is infinite…)
- **Plotting**
  - "XY" plot
  - See what happens when "Level" is changed -- Level 0, Level 1 (must click on "Redraw" to refresh the plot after changing Level)
- **Results**
  - Same as previous runs
  - Not always true when you use universe/fill - might have different roundoff …

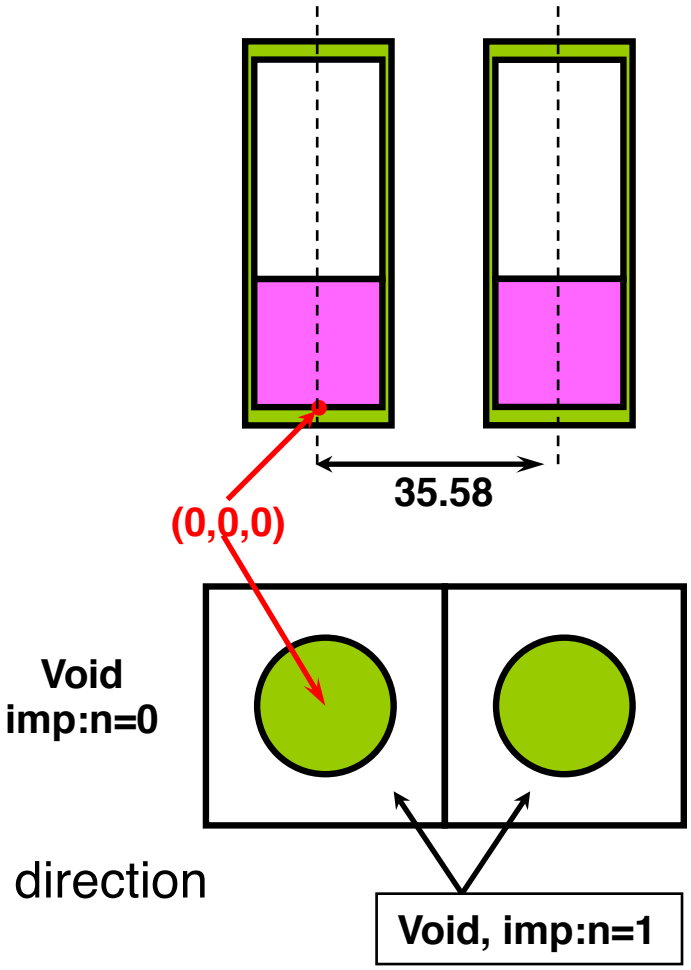
Example puc3 (1)

Problem puc3

Two Cans of Solution
Using ‘Like m But’ and TRCL

Example puc3 (2)

- **Two cans, 35.58 cm separation between centers**
- **Copy file puc2 to puc3**
- **Edit file puc3**
  - **Identify cells 25, 30, 40 as being in universe 2 (change the importance of cell 40 to imp:n=1)**
  - **Define cell 50 & surface 4**
    - A box around the first can (cell 30)
    - Use RPP, x,y in range (-17.79,17.79), z in range (-1,102.7)
    - FILL cell 50 with universe 2
    - Importance = 1
  - **Define cell 60**
    - Same as cell 50, but translated 35.58 cm in +x direction
  - **Define cell 99**
    - Void, importance = 0, outside of 50 & 60
  - **Add another source point to KSRC, at (35.58, 0., 19.62)**



Example puc3 (3)

```
puc3 - TWO cylinders
C ----- universe 1, infinite solution & void -----
10  100     9.9270e-2  -3          u=1  imp:n=1   $ infinite solution
20  0                   3          u=1  imp:n=1   $ infinite void
C ----- universe 2, filled can & infinite exterior -----
25  0                  -1 fill=1   u=2  imp:n=1   $ inside of can, filled
30  200     8.6360e-2   1 -2       u=2  imp:n=1   $ can
40  0                   2          u=2  imp:n=1   $ infinite exterior
C ----- real world, 2 boxes (contining cans) & infinite exterior -----
50  0                  -4 fill=2        imp:n=1   $ 1st box at origin, with can
60  like 50 but  trcl=(35.58  0.  0.)             $ 2nd box shifted, with can
99  0                   #50 #60         imp:n=0   $ exterior to both boxes

1       RCC   0. 0.  0.    0. 0. 101.7     12.49
2       RCC   0. 0. -1.    0. 0. 103.7     12.79
3       pz    39.24
4       RPP  -17.79 17.79   -17.79 17.79   -1. 102.7

C DATA CARDS from puc1_data_cards.txt
kcode   1000 1.0 25 100
ksrc    0. 0. 19.62        35.58 0. 19.62
m100    . . .
mt100   . . .
m200    . . .
```

Result:
keff = 0.91260 ± 0.00381

Example puc3 (4)

- **Universe 1 is infinite**
  - Same as before, but now appears in 2 different places
  - Clipped by surface 1 when if FILLs cell 25
- **Universe 2 is infinite**
  - Can (containing universe 1) & exterior void
  - Embedded in Cell 50, and also in cell 60
- **Plotting**
  - "XY" plot, "ZX" plot
  - See what happens when "Level" is changed -- Level 0, Level 1, Level 2 (must click on "Redraw" to refresh the plot after changing Level)
  - Note surface 60004 -- what happened to other translated surfaces?

    (See output file for info on identical surfaces…)
  - Click on "MBODY On" - note the surface "facets" (internal label for body surfaces)
- **Results**
  - Higher Keff, as expected

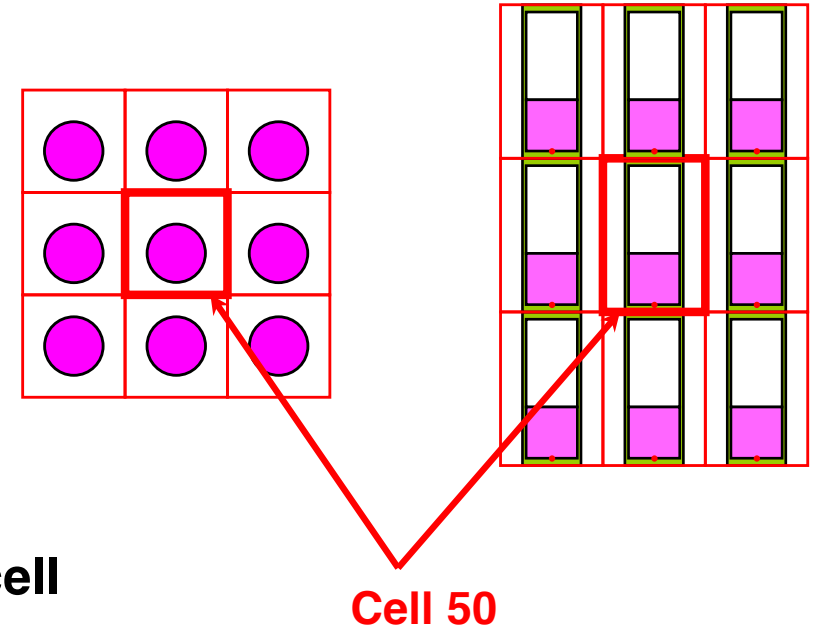
Example puc4 (1)

Problem puc4

Infinite Lattice of Cans
(3D Lattice)

Example puc4 (2)

Define the center lattice cell, filled with universe 2



- **Copy file puc3 to puc4**
- **Edit file puc4**
  - **Delete cells 60 & 99**
  - **Declare Cell 50 to be the center lattice cell in a hexahedral (box) lattice, LAT=1**
  - **Add this data card (turns off entropy calc for infinite lattice):**

```
hsrc  1 -1.e10 1.e10  1 -1.e10 1.e10  1 -1.e10 1.e10
```

  - **Remove the second point in KSRC**
  - **Plot:** `mcnp6 i=puc4 ip`
  - **Compute keff:** `mcnp6 i=puc4`

Example puc4 (3)

```
puc4 - infinite 3D lattice of cans
10   100   9.9270e-2   -3          u=1     imp:n=1   $ infinite solution
20   0                  3          u=1     imp:n=1   $ infinite void
25   0                 -1 fill=1   u=2     imp:n=1   $ inside can, filled
30   200   8.6360e-2    1 -2       u=2     imp:n=1   $ can
40   0                  2          u=2     imp:n=1   $ infinite exterior
50   0                 -4 fill=2   lat=1   imp:n=1   $ center lattice cell

1    RCC    0. 0.  0.     0. 0. 101.7      12.49
2    RCC    0. 0. -1.     0. 0. 103.7      12.79
3    pz     39.24
4    RPP   -17.79 17.79   -17.79 17.79    -1. 102.7   $ box to hold can

C DATA CARDS from puc1_data_cards.txt
kcode    1000 1.0 25 100
ksrc     0. 0. 19.62
m100     . . .
mt100    . . .
m200     . . .
hsrc     1 -1.e10 1.e10   1 -1.e10 1.e10   1 -1.e10 1.e10
```

Result:
keff = 1.61058 ± 0.00194

Example puc4 (4)

- **Keff is pretty large**
  - Infinite lattice, no leakage, no absorbers, …
- **Note that lattices are:**
  - Defined by creating the center cell & flagging it with LAT=1
  - Filled with 1 or more universes
  - Infinite in extent
- **How do you get a finite lattice?**

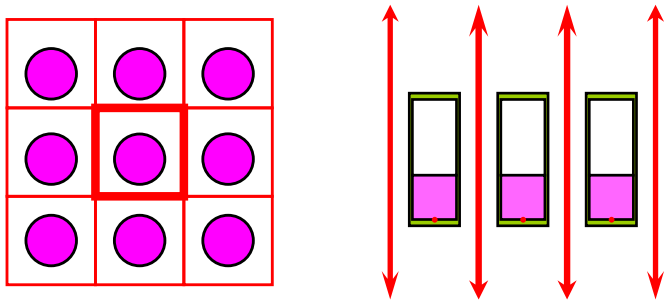
1. Make an infinite lattice, then give it a universe number
2. Create a container cell to hold some portion of the lattice
3. Then fill that container cell with the lattice universe
   - Infinite lattice is clipped (truncated) by the container cell boundaries
   - Lattice elements outside the container can never be reached

Example puc5 (1)

Problem puc5

Infinite Lattice of Cans
(2D Lattice)

Example puc5 (2)



Define a 2D lattice (x-y plane) of cans

- Copy file puc4.txt to puc5.txt
- Edit file puc5.txt
  - Change surface 4 (RPP body that defines center cell)
    - Make the RPP infinite in the z-direction
      - Use "0. 0." for the z top/bottom
      - This tells MCNP that the RPP is ininite in z-direction (no top/bottom)
  - Need to consider neutrons above & below the cans
    - Want void with imp:n=1 between cans
    - Want void with imp:n=0 above & below cans
      (to prevent neutrons from streaming forever…)
    - Need to add Cell 45 to universe 2, above can & below can, imp:n=0
      Could do this with 2 extra surfaces or use existing macrobody facets
  - Plot: mcnp6 i=puc5.txt ip
  - Compute keff: mcnp6 i=puc5.txt

Example puc5 (3)

```
puc5 - infinite lattice of cans, 2D  (USING SURFACES)
10   100  9.9270e-2   -3          u=1    imp:n=1   $ infinite solution
20   0                 3          u=1    imp:n=1   $ infinite void
25   0                -1 fill=1   u=2    imp:n=1   $ contents of can, filled
30   200  8.6360e-2    1 -2       u=2    imp:n=1   $ can
40   0                 2 -5   6   u=2    imp:n=1   $ void, beside can
45   0                    5 :-6   u=2    imp:n=0   $ void, above & below can
50   0                -4 fill=2   lat=1  imp:n=1   $ infinite 2D lattice of cans

1    RCC   0. 0.  0.    0. 0. 101.7     12.49
2    RCC   0. 0. -1.    0. 0. 103.7     12.79
3    pz    39.24
4    RPP  -17.79 17.79   -17.79 17.79   0. 0.     $ box, infinite in z-dir
5    pz    102.7                                  $ at top of can
6    pz     -1.0                                  $ at bottom of can

C DATA CARDS from puc1_data_cards.txt
kcode   1000 1.0 25 100
ksrc    0. 0. 19.62
m100    . . .
mt100   . . .
m200    . . .
hsrc    1 -1.e10 1.e10   1 -1.e10 1.e10   1 -1.e10 1.e10
```

Result:
keff = 1.15362 ± 0.00339

Example puc5 (4)

```
puc5m - infinite lattice of cans, 2D  (USING MACROBODY FACETS)
10       100      9.9270e-2   -3     u=1              imp:n=1
20       0                     3     u=1              imp:n=1
25       0                    -1 fill=1      u=2      imp:n=1
30       200      8.6360e-2    1 -2           u=2      imp:n=1
40       0                     2 -2.2 -2.3    u=2      imp:n=1
45       0                        2.2: 2.3    u=2      imp:n=0
50       0                    -4 fill=2  lat=1         imp:n=1

1        RCC    0. 0.  0.     0. 0. 101.7      12.49
2        RCC    0. 0. -1.     0. 0. 103.7      12.79
3        pz     39.24
4        RPP   -17.79 17.79   -17.79 17.79     0. 0.

C DATA CARDS from puc1_data_cards.txt
kcode    1000 1.0 25 100
ksrc     0. 0. 19.62
m100     . . .
mt100    . . .
m200     . . .
hsrc     1 -1.e10 1.e10   1 -1.e10 1.e10   1 -1.e10 1.e10
```

Result:
keff = 1.15362 ± 0.00339

Example puc5 (5)

- **Keff is more reasonable**
  - **Infinite lattice in 2D, leakage in Z, no absorbers, …**
  - **Same result for both puc5 & puc5m**
- **File puc5 - straightforward**
  - **Extra cell & surfaces to define voids in between cans with imp:n=1, and above/below cans with imp:n=0**
- **File puc5m**
  - **Similar to puc5, but no extra surfaces needed**
  - **Use existing top of can (facet 2.2) and bottom of can (facet 2.3)**
  - **Major source of confusion: the sign (sense) for facet 2.3**
    - For macrobodies, MCNP internally translates the body definition into a collection of surfaces: infinite cylinder (2.1), top plane (2.2), bottom plane (2.3)
    - **By definition, MCNP considers inside the body to have negative sense, & outside the body to have positive sense**
    - **MCNP alters the surface definitions to match the body sense convention**, hence inside the body has negative sense wrt surface 2.3 and outside the body has positive sense wrt surface 2.3, opposite to the normal surface sense conventions for 2.3

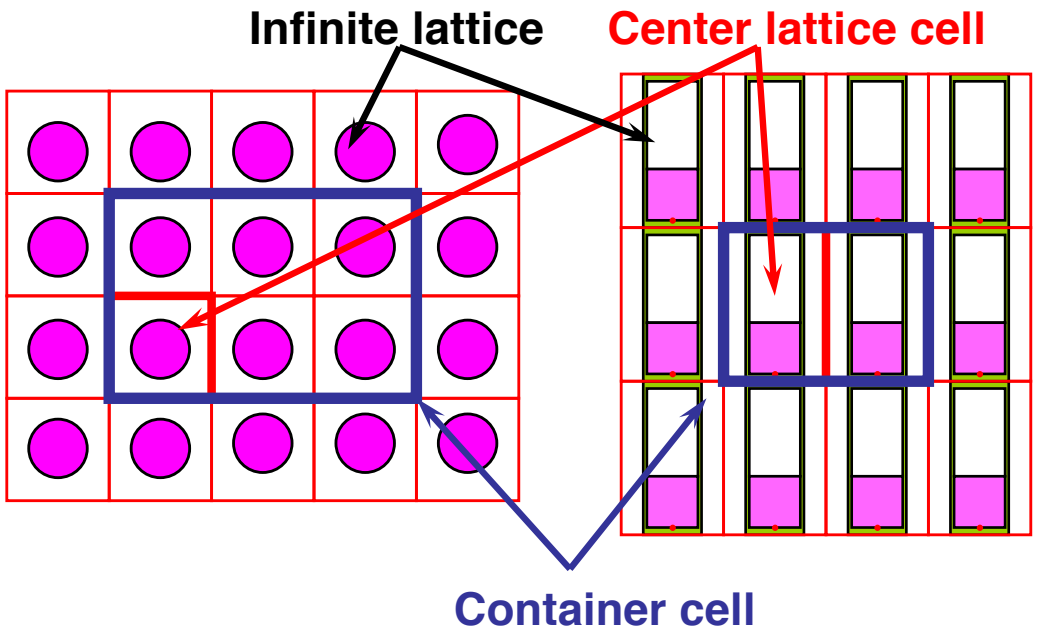
Example puc6 (1)

Problem puc6

Finite 2x3 Lattice of Cans

Example puc6 (2)

Start with infinite 3D lattice (Problem puc4) & create a 3x2 array of cans



- **Copy file puc4.txt to puc6.txt**
- **Edit file puc6.txt**
  - **Declare cell 50 as universe 3**
  - **Define the container cell 60, and an RPP body sized to hold 2x3 array of cell 50**
  - **Fill cell 60 with universe 3**
  - **Define cell 99, outside container, imp:n=0**
  - **Change KCODE card to:** `kcode 20000 1.0 25 100`
  - **Modify the HSRC card (optional):**

```
hsrc  3 -17.79 88.95   2 -17.79 53.37   1 -1. 102.7
```

  - **Plot:** mcnp5 i=puc6.txt ip
  - **Compute keff:** mcnp5 i=puc6.txt

Example puc6 (3)

```
puc6 3 x 2 array of cans
10   100   9.9270e-2   -3          u=1          imp:n=1    $ infinite solution
20   0                  3          u=1          imp:n=1    $ infinite void
25   0                 -1 fill=1   u=2          imp:n=1    $ contents of can, filled
30   200   8.6360e-2    1 -2       u=2          imp:n=1    $ can
40   0                  2          u=2          imp:n=1    $ outside of can, infinite
50   0                 -4 fill=2   lat=1 u=3    imp:n=1    $ infinite 3D lattice
60   0                 -5 fill=3                imp:n=1    $ container, fill by lattice
99   0                  5                       imp:n=0    $ outside container

1        RCC    0. 0.  0.     0. 0. 101.7      12.49
2        RCC    0. 0. -1.     0. 0. 103.7      12.79
3        pz     39.24
4        RPP   -17.79 17.79   -17.79 17.79     -1. 102.7
5        RPP   -17.79 88.95   -17.79 53.37     -1. 102.7  $ container, holds 3x2

C DATA CARDS from puc1_data_cards.txt
kcode    20000 1.0 25 100
ksrc     0. 0. 19.62
m100     . . .
mt100    . . .
m200     . . .
hsrc     1 -17.79 88.95   2 -17.79 53.37   1 -1. 102.7
```

Result:
keff = 0.98736 ± 0.00094

Example puc6 (4)

- **Keff is close to 1.0 ….**

- **Note that lattices are:**
  - **Defined by creating the center cell & flagging it with LAT=1**
  - **Filled with 1 or more universes**
  - **Infinite in extent**

- **To get a finite lattice:**

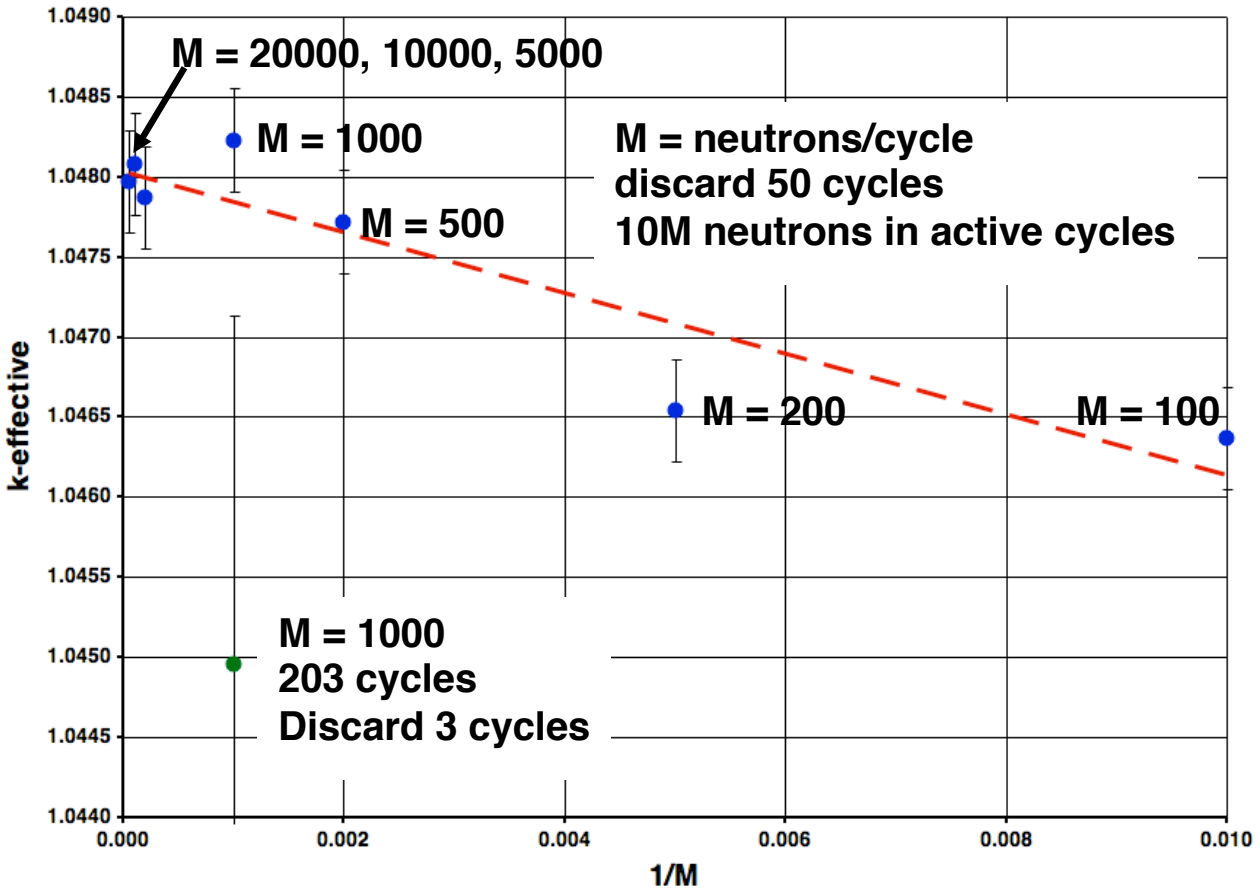
1. **Make an infinite lattice, then give it a universe number**

2. **Create a container cell to hold some portion of the lattice**

3. **Fill the container cell with the lattice universe**
   - **Infinite lattice is clipped (truncated) by the container cell boundaries**
   - **Lattice elements outside the container can never be reached**

Comments - puc6

- As noted in the Theory Lecture #5 - Eigenvalue Calculations - III, there will be a bias (error) in K-effective if too few neutrons/cycle are used

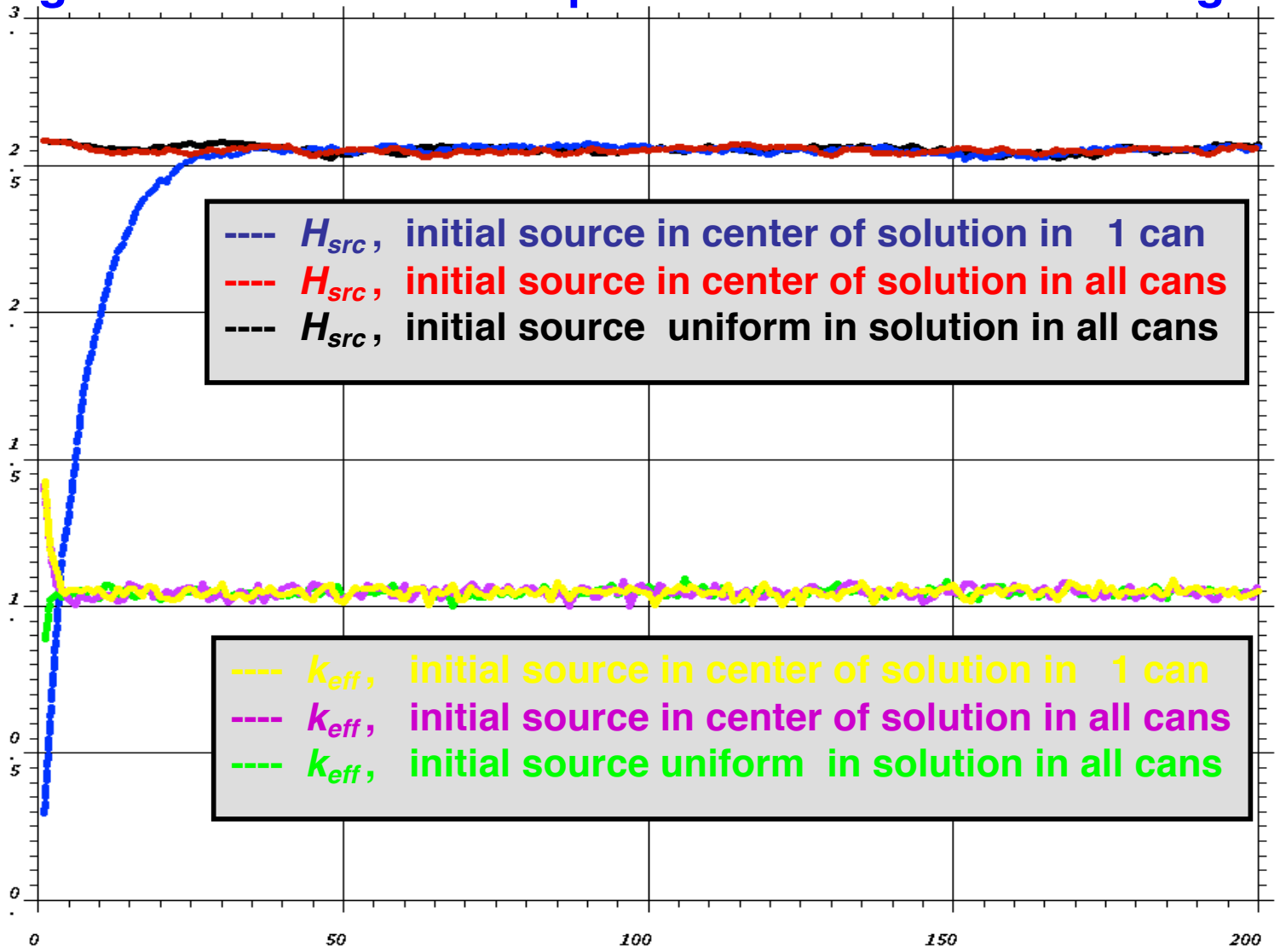


- Note: Bias in green point due to using Keno default - discard 3 cycles, 203 cycles total

ENDF/B-VI+T16 data used for above calculations

Comments - puc6

- As noted in the Theory Lecture #5 - Eigenvalue Calculations - III, convergence of the source depends on the initial source guess:



ENDF/B-VI+T16 data used for above calculations

Summary of Results

Single can	puc1	keff = 0.88778 ± 0.00363
Two cans	puc3	keff = 0.91260 ± 0.00381
3x2 array of cans	puc6	keff = 0.98736 ± 0.00094
Infinite lattice, 2D	puc5	keff = 1.15362 ± 0.00339
Infinite lattice, 3D	puc4	keff = 1.61058 ± 0.00194

Next …

- Vary the solution height in the cans
- Random location of cans & random solution height
- Test other representations for the initial source distribution
- Put a source point in each tank
- Etc.
- Note: Problems puc7, puc8, puc9, puc10, puc11 were run using different cross-section data

Example

Problem puc7

Parameter Study - Vary the Solution Height in the 2x3 Lattice of Cans

Problem puc7

- **Want to run the problem with a number of different solution heights (with same solution height in each of the 6 cans)**
  - Easy to do - just change the number on surface 3 (z-plane)
  - Could edit the file, run a case, edit the file, run another case, etc.
  - A simpler way - use the mcnp_pstudy utility to setup & run all the cases

- **Copy file puc6 to puc7 & edit file puc7**
  - Add parameters for mcnp_pstudy

```
c @@@ HEIGHT =  5 10 15 20 25 30 35 40 45 50 \
c @@@           55 60 65 70 75 80 85 90 95 100
c @@@ SRCZ    = ( .5 * HEIGHT )
```

  - Modify cards for Surface 3 & KSRC

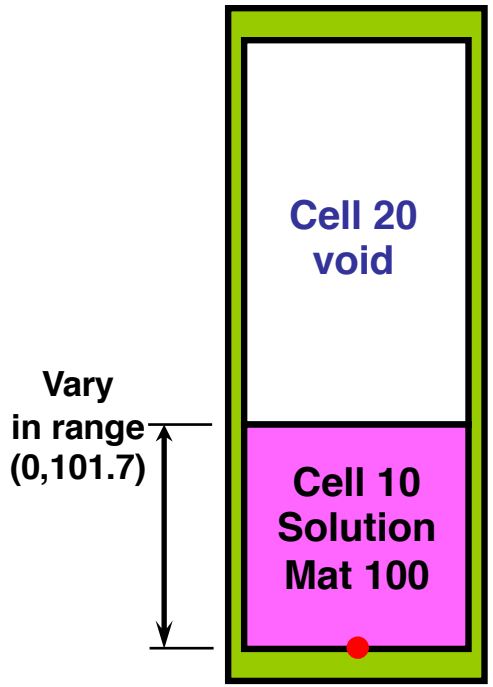
```
3        pz     HEIGHT

ksrc     0. 0. SRCZ
```

  - Add PRDMP card (to produce mctal files)

```
prdmp  0  0  1  1  0
```

  - Modify KCODE for more histories

```
kcode  1000  1.0   50   350
```



- **Run:** `mcnp_pstudy -i puc7 -setup -run`

Problem puc7

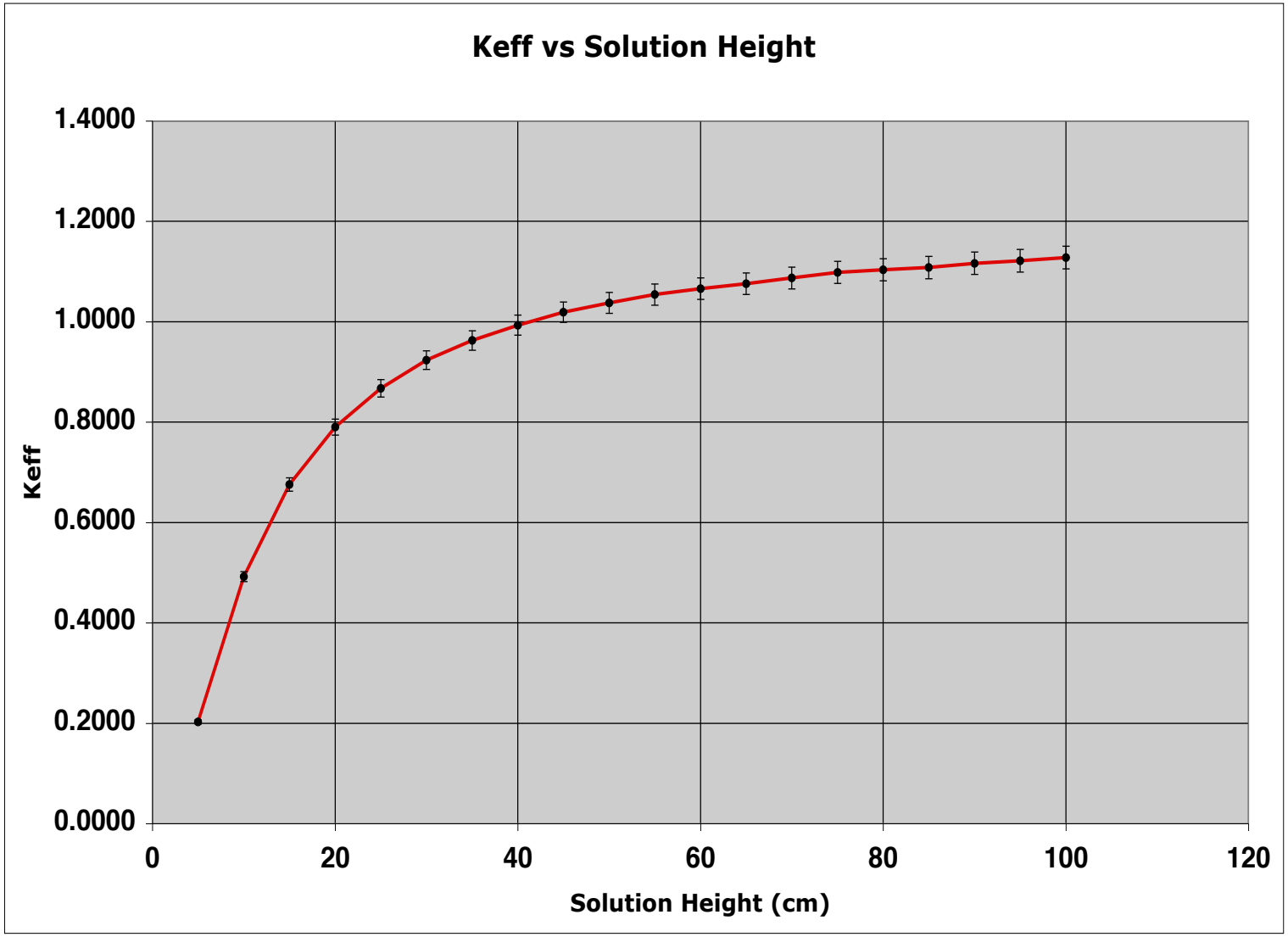
```
puc7 3 x 2 array of cans -- varing solution height
c
c @@@ HEIGHT =  5 10 15 20 25 30 35 40 45 50 \
c @@@          55 60 65 70 75 80 85 90 95 100
c @@@ SRCZ   = ( .5 * HEIGHT )
c
10      100     9.9270e-2   -3    u=1              imp:n=1
20      0                    3    u=1              imp:n=1
25      0                   -1 fill=1  u=2         imp:n=1
30      200     8.6360e-2    1 -2      u=2         imp:n=1
40      0                    2         u=2         imp:n=1
50      0                   -4 fill=2  lat=1 u=3   imp:n=1
60      0                   -5 fill=3              imp:n=1
99      0                    5                     imp:n=0

1       RCC   0. 0.  0.    0. 0. 101.7     12.49
2       RCC   0. 0. -1.    0. 0. 103.7     12.79
3       pz    HEIGHT
4       RPP  -17.79 17.79   -17.79 17.79   -1. 102.7
5       RPP  -17.79 88.95   -17.79 53.37   -1. 102.7

C DATA CARDS from puc1_data_cards.txt
kcode   1000 1.0 50  350
ksrc    0. 0. SRCZ
prdmp   0  0  1  1  0
hsrc    3 -17.79 88.95   2 -17.79 53.37   1 -1. 102.7
m100    . . .
mt100   . . .
m200    . . .
```

Problem puc7

- **Results - Keff vs solution height**

Hgt	Keff
5	0.203
10	0.492
15	0.676
20	0.790
25	0.867
30	0.923
35	0.962
40	0.993
45	1.019
50	1.037
55	1.054
60	1.066
65	1.076
70	1.087
75	1.099
80	1.104
85	1.108
90	1.116
95	1.121
100	1.128

$\sigma \approx .002$



- **Examine "case" directories & log.txt file**

Example

Problems
puc8, puc9, puc10, puc11

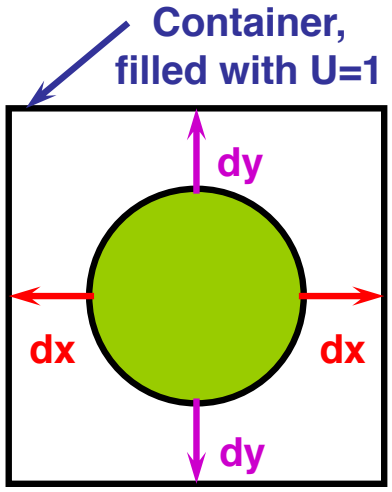
Stochastic Geometry
Randomly Vary the Solution Height & the Location of Cans

Stochastic Geometry

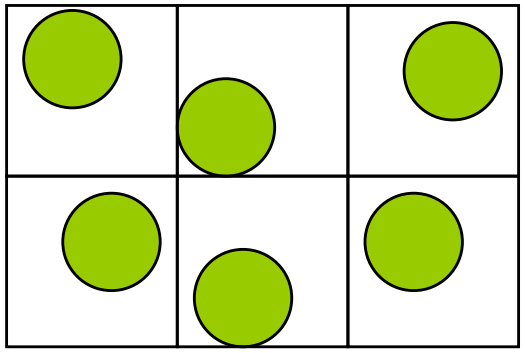
- What if the location of each of the 6 cans is not precisely known?
  What if the solution height in each of the 6 cans is not precisely known?
- MCNP5 can randomly vary the geometry on-the-fly while tracking particles
  - Limitation: random displacements in x,y,z of up to 2 universes
  - Input card:

```
uran   u1   dx1   dy1   dz1   u2   dx2   dy2   dz2
```

where u_1, u_2 = universe numbers
dx,dy,dz = maximum displacements in ±x, ±y, ±z,



One instantiation for lattice filled with universe:
Random displacement each time a particle enters a stochastic universe



Stochastic Geometry

- **For random location of each can (individually)**
  - Universe 2 fills the lattice elements defined by Cell 50

    Diameter of can = 2 x 12.79 = 25.38 cm

    Width of Cell 50 = 2 x 17.79 = 35.38 cm

    ➜ Could displace a can ±5 cm in x or y without clipping by container cell

- **For random solution height (in each can individually)**
  - Any change in solution height (location of z-plane 3) results in gain or loss in total mass of solution
  - Varying the solution height does not preserve mass

- **For this exercise:**

  ➜ Randomly vary the x & y can locations up to ±5 cm

  ➜ Set the solution height = 20 cm

  ➜ Randomly vary the solution height up to ±5 cm in z

Stochastic Geometry

- **Copy file puc6 to puc8**
  - Case puc8 will be a base case - no variations in can location or solution hgt, but more cycles & neutrons/cycle
  - Modify Surface 3 & KSRC

```
3   pz   20.

ksrc  0. 0. 10.

kcode   5000  1.0  50   1050
```

- **Copy puc8 to: puc9, puc10, puc11**
  - puc9: use URAN to vary the solution height randomly up to ±5 cm
  - puc10: use URAN to vary x-y can locations randomly up to ±5 cm
  - puc11: use URAN to vary both solution height & x-y up to ±5 cm

Problem puc8

```
puc8  3 x 2 array of cans
10       100      9.9270e-2   -3     u=1              imp:n=1
20       0                     3     u=1              imp:n=1
25       0                    -1 fill=1  u=2          imp:n=1
30       200      8.6360e-2    1 -2      u=2          imp:n=1
40       0                     2        u=2          imp:n=1
50       0                    -4 fill=2  lat=1 u=3    imp:n=1
60       0                    -5 fill=3               imp:n=1
99       0                     5                      imp:n=0

1        RCC    0. 0.  0.     0. 0. 101.7      12.49
2        RCC    0. 0. -1.     0. 0. 103.7      12.79
3        pz     20.
4        RPP   -17.79 17.79    -17.79 17.79    -1. 102.7
5        RPP   -17.79 88.95    -17.79 53.37    -1. 102.7

C DATA CARDS from puc1_data_cards.txt
kcode    5000 1.0 50  1050
hsrc     3 -17.79 88.95   2 -17.79 53.37   1 -1. 102.7
ksrc     0. 0. 10.
m100     . . .
mt100    . . .
m200     . . .
```

Problem puc9

```
puc9  3 x 2 array of cans -- vary solution height randomly
10      100     9.9270e-2   -3    u=1              imp:n=1
20      0                    3    u=1              imp:n=1
25      0                   -1 fill=1  u=2         imp:n=1
30      200     8.6360e-2    1 -2      u=2         imp:n=1
40      0                    2         u=2         imp:n=1
50      0                   -4 fill=2  lat=1 u=3   imp:n=1
60      0                   -5 fill=3              imp:n=1
99      0                    5                     imp:n=0

1       RCC    0. 0.  0.    0. 0. 101.7     12.49
2       RCC    0. 0. -1.    0. 0. 103.7     12.79
3       pz     20.
4       RPP  -17.79 17.79   -17.79 17.79   -1. 102.7
5       RPP  -17.79 88.95   -17.79 53.37   -1. 102.7

uran    1  0. 0. 5.0        $ randomly vary position of universe 1, +- 5 cm in z
C DATA CARDS from puc1_data_cards.txt
kcode   5000 1.0 50  1050
hsrc    3 -17.79 88.95   2 -17.79 53.37   1 -1. 102.7
ksrc    0. 0. 10.
m100    . . .
mt100   . . .
m200    . . .
```

Problem puc10

```
puc10    3 x 2 array of cans -- vary can x-y positions randomly
10       100      9.9270e-2   -3    u=1              imp:n=1
20       0                     3    u=1              imp:n=1
25       0                    -1 fill=1  u=2         imp:n=1
30       200      8.6360e-2    1 -2      u=2         imp:n=1
40       0                     2        u=2         imp:n=1
50       0                    -4 fill=2  lat=1 u=3   imp:n=1
60       0                    -5 fill=3              imp:n=1
99       0                     5                     imp:n=0

1        RCC    0. 0.  0.     0. 0. 101.7     12.49
2        RCC    0. 0. -1.     0. 0. 103.7     12.79
3        pz     20.
4        RPP   -17.79 17.79   -17.79 17.79    -1. 102.7
5        RPP   -17.79 88.95   -17.79 53.37    -1. 102.7

uran    2  5. 5. 0.          $ randomly vary x-y position of universe 2, +- 5 cm
C DATA CARDS from puc1_data_cards.txt
kcode    5000 1.0 50  1050
hsrc     3 -17.79 88.95   2 -17.79 53.37   1 -1. 102.7
ksrc     0. 0. 10.
m100     . . .
mt100    . . .
m200     . . .
```

Problem puc11

```
puc11 3 x 2 array of cans -- vary solution height & can x-y positions
10       100      9.9270e-2   -3     u=1                imp:n=1
20       0                     3     u=1                imp:n=1
25       0                    -1 fill=1  u=2             imp:n=1
30       200      8.6360e-2    1 -2      u=2             imp:n=1
40       0                     2        u=2             imp:n=1
50       0                    -4 fill=2  lat=1 u=3       imp:n=1
60       0                    -5 fill=3                  imp:n=1
99       0                     5                         imp:n=0

1        RCC    0. 0.  0.     0. 0. 101.7      12.49
2        RCC    0. 0. -1.     0. 0. 103.7      12.79
3        pz     20.
4        RPP   -17.79 17.79    -17.79 17.79    -1. 102.7
5        RPP   -17.79 88.95    -17.79 53.37    -1. 102.7

uran   2  5. 5. 0.   1 0. 0. 5.    $ randomly vary position (univ 2) &
                                    $ height (univ 1)  independently
C DATA CARDS from puc1_data_cards.txt
kcode    5000 1.0 50  1050
hsrc     3 -17.79 88.95   2 -17.79 53.37   1 -1. 102.7
ksrc     0. 0. 10.
m100     . . .
mt100    . . .
m200     . . .
```

Stochastic Geometry - Results

	Keff (uncert in last digit)
• puc8 - base case	.7911 (4)
• puc9 - vary solution height	.8078 (4)
• puc10 - vary x-y can position	.7945 (4)
• puc11 - vary both pos & hgt	.8105 (4)

- Deltas from base case, due to random geometry:
  - Vary can x-y positions only Δk = + .0034 (6)
  - Vary solution heights Δk = + .0167 (6)
  - Vary both (independently) Δk = + .0194 (6)

Stochastic Geometry - Comments

- Note that in puc11, the solution heights & x-y can locations are randomly varied independently for each of the 6 cans, each time a neutron enters a new lattice element
- It would take an extraordinary number of separate Monte Carlo runs with different geometry replicas to duplicate these results
- Not aware of any other MC code that can do this type of calculation
- Lots of other possible applications for the stochastic geometry:
  - Fuel kernels in VHTR & HTGR (done)
  - Lung tissue modeling for medical physics
  - Bubbles & voids in reactor coolant
  - Vibrating control rods
  - Many criticality safety problems (junk in cans, solutions, …..)

Case Study #2

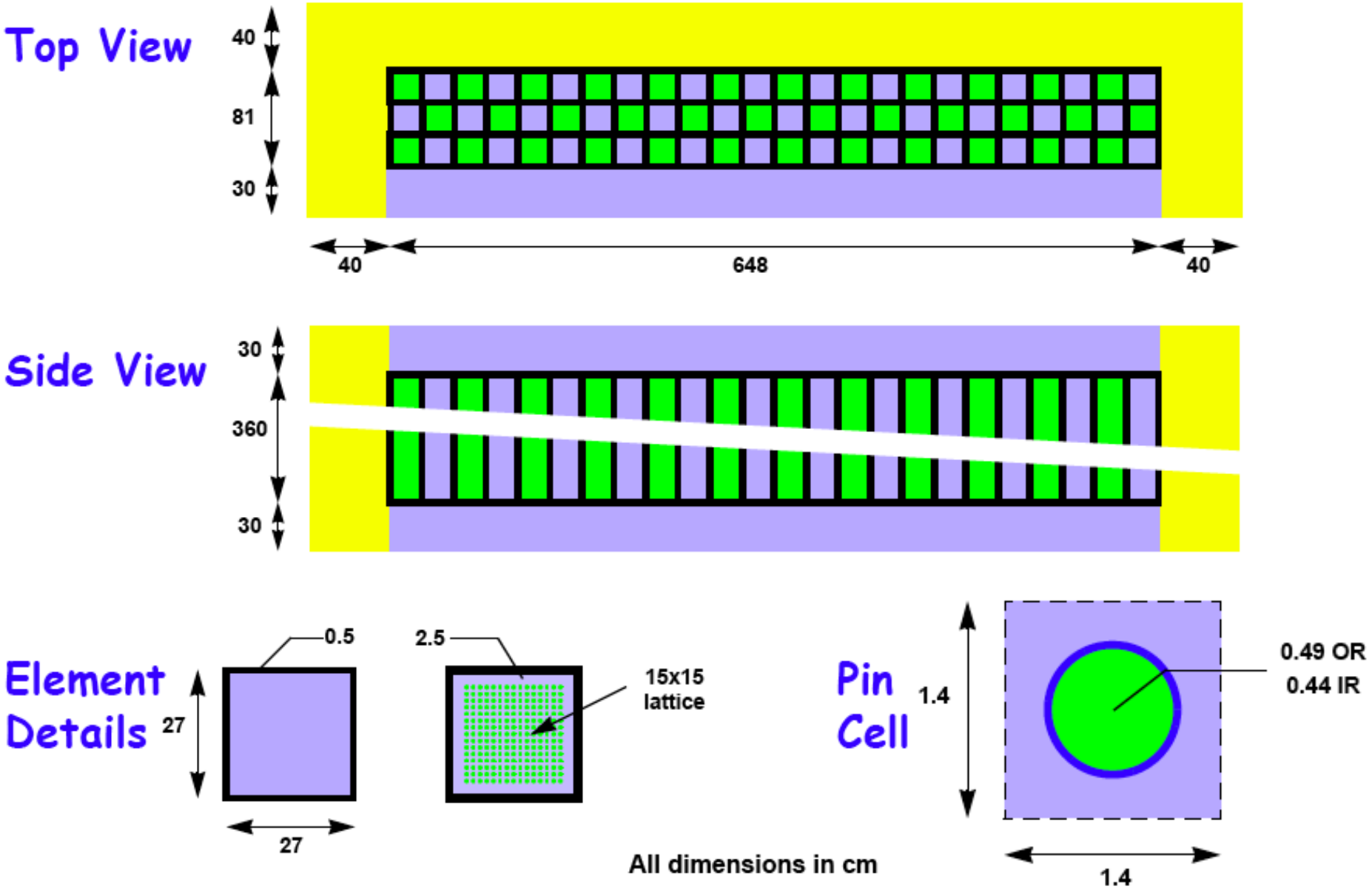
OECD/NEA Source Convergence Benchmark

Fuel Storage Vault - Checkerboard Storage of Assemblies

- Benchmark problem defined by OECD Nuclear Energy Agency "Expert Group on Monte Carlo Source Convergence"
- Very large & difficult problem, easy to get wrong answers
- Set up problem in small stages:
  (1) Pin cell
  (2) Infinite lattice of pin cells
  (3) Fuel assembly
  (4) Fuel assembly with various external geometry
  (5) Entire fuel vault

Calculations based on ENDF/B-VII.0 data libraries

Fuel Storage Vault



Materials

Fuel: rho = 0.06925613 atoms/b-cm

U238	92238	2.2380e-2
O	8016	4.6054e-2
U235	92235	8.2213e-4

Cladding: rho = 0.042910 atoms/b-cm

Zr	40000	4.2910e-2

Water: rho = 0.100059 atoms/b-cm

O	8016	3.3353e-2
H	1001	6.6706e-2, with S(α,β) lwtr

Iron: rho = 0.083770 atoms/b-cm

Fe	26000	8.3770e-2

Concrete: rho = 0.0725757 atoms/b-cm

H	1001	5.5437e-3, with S(α,β) lwtr
C	6000	6.9793e-2
Si	14000	7.7106e-3
Ca	20000	8.9591e-3
O	8016	4.3383e-2

Problem fv1 - Pin Cell

- **Fuel (m100):** rho = 0.06925613 atoms/b-cr

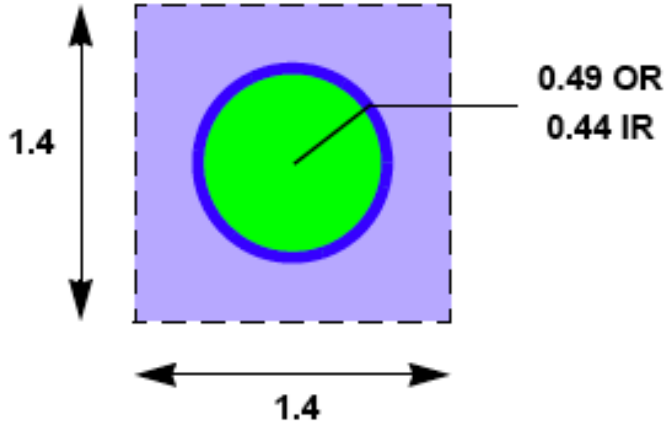
U238	92238	2.2380e-2
O	8016	4.6054e-2
U235	92235	8.2213e-4

- **Cladding (m200):** rho = 0.042910 atoms/b-cm

Zr	40000	4.2910e-2

- **Water (m300):** rho = 0.100059 atoms/b-cm

O	8016	3.3353e-2
H	1001	6.6706e-2, with S(α,β) lwtr



(1)Create & edit file "fv1"

(2)Setup the geometry so that (0,0,0) is at center of fuel pin, at z midpoint of pin

(3)Above/below pin cell - 30 cm of water, then vacuum (ignore end-caps)

(4) Use macrobodies

cylinder:	RCC	x0 y0 z0 h1 h2 h3 r
box:	RPP	x1 x2 y1 y2 z1 z2

(5) Add these data cards:

```
kcode   1000   .1   10   50
ksrc    0.0  0.0  0.0
mt300   lwtr
```

Problem fv1

- File fv1

```
Fuel vault - pin cell
c  CELLS
10   100    0.06925613     -1          $ fuel
20   200    0.042910        1 -2       $ clad
30   300    0.100059        2 -3       $ water
40     0                     3         $ exterior

c SURFACES
1   RCC    0. 0. -180.      0. 0. 360.      0.44
2   RCC    0. 0. -180.      0. 0. 360.      0.49
3   RPP    -.7 .7    -.7 .7    -210. 210.

c DATA
Kcode  1000   .1   10   50
ksrc   0.0   0.0   0.0
imp:n    1   1   1   0
c
m100   92238 2.2380e-2    92235 8.2213e-4    8016 4.6054e-2 $ fuel
m200   40000 4.2910e-2                                     $ clad
m300   1001 6.6706e-2    8016 3.3353e-2                    $ water
mt300    lwtr
```

- Execution

```
mcnp6 i=fv1  ip
mcnp6  i=fv1
```

Result:

$K_{eff} = 0.01216 \pm 0.00011$

Comments - fv1

- **Hmmm, runs fast & Keff ~ .01357 ± .00011 ?????**
  - **Problem was setup with vacuum boundary -- imp:n 0 for cell 40**
  - **Should use reflecting boundary condition**

- **Rerun with reflective boundary condition:**

  - **Copy file fv1 to file fv1a, then edit:**
    - **Add * before surface # on surface card for surface 3**:

```
*3    RPP    -.7  .7     -.7  .7     -210.  210.
```

    - Change KCODE card to:

```
kcode   1000   1.0    10    50
```

rm out* runtp* srct*
mcnp6 i=fv1a

  - **Final Keff ~ 1.45296 +- .00239**

Problem fv1a

```
Fuel vault - pin cell
c  CELLS
10   100    0.06925613     -1          $ fuel
20   200    0.042910        1 -2       $ clad
30   300    0.100059        2 -3       $ water
40     0                     3         $ exterior

c SURFACES
1    RCC    0. 0. -180.      0. 0. 360.      0.44
2    RCC    0. 0. -180.      0. 0. 360.      0.49
*3   RPP    -.7 .7     -.7 .7     -210. 210.

c DATA
Kcode   1000   1.0   10   50
ksrc   0.0   0.0   0.0
imp:n    1   1   1   0
c
m100   92238 2.2380e-2    92235 8.2213e-4    8016 4.6054e-2 $ fuel
m200   40000 4.2910e-2                                     $ clad
m300   1001 6.6706e-2    8016 3.3353e-2                    $ water
mt300    lwtr
```

Result:

$K_{eff} = 1.45296 \pm 0.00239$

Comments - fv1a

- **Cell 40 is not needed**
  - Could delete it (remember to update imp:n card)

- **Examine the output file, fv1a**
  - No warning messages concerning results or convergence
  - Look for "the box" -- final Keff results
  - Note that the number of source points in each cycle does not have to be exactly 1000
  - the number varies, but the total weight of the source particles is 1000.

- **Data card: mt300 lwtr**
  - Specifies that S(α,β) data should be used for the thermal scattering cross-sections
  - This treatment accounts for molecular binding effects on thermal collisions
  - Should always specify S(α,β) treatment for water in thermal systems

Comments - fv1a

- **Plotting results**
  - **Invoke "mcplot" to view results:** **mcnp6 r=runtpe z**
  - **Plot k(col/abs/trk len):** **mcplot> kcode 16**

- **Other useful plots:**
  - **Individual cycle results:**

k(col)	**mcplot> kcode 1**
k(abs)	**mcplot> kcode 2**
k(trk)	**mcplot> kcode 3**
Shannon entropy	**mcplot> kcode 6**

  - **Cumulative results:**

k(col)	**mcplot> kcode 11**
k(abs)	**mcplot> kcode 12**
k(trk)	**mcplot> kcode 13**
k(col/abs/trk)	**mcplot> kcode 16**
k(col/abs/trk) by cycles skipped:	**mcplot> kcode 17**

- **Multiple plots:**

  mcplot> kcode 1 **coplot** kcode 2 **coplot** kcode 3

- **Exit mcplot:**

  mcplot> **end**

Problem fv2

Pin Cell Using UNIVERSE & FILL

Problem fv2 - Pin Cell Using U & FILL

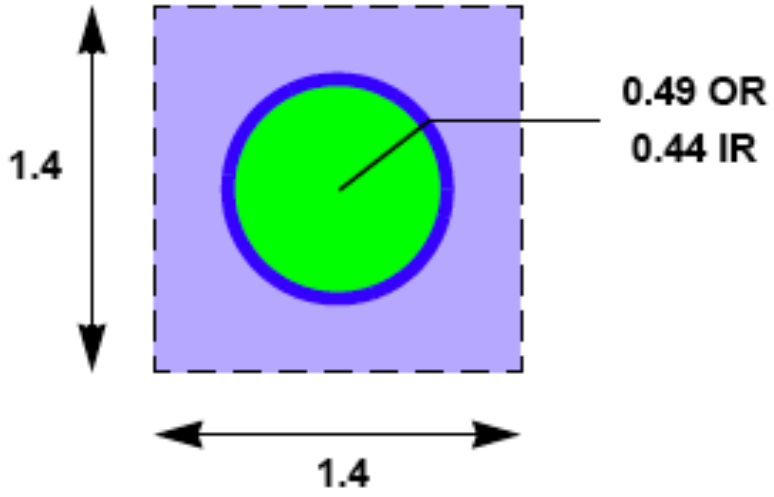
Modify Problem fv1a

- Copy fv1a to file fv2, then edit

- Universe 1: fuel rod, clad, & infinite moderator
- Create the bounding box
- Fill the bounding box with Universe 1
- Add the HSRC card:

```
hsrc   4  -.5  .5    4 -.5 .5    4 -180. 180.
```

```
cp   fv1a   fv2
[edit fv2]
mcnp5 i=fv2 ip
mcnp5 i=fv2
```



Solution - Problem fv2

File fv2

```
Problem fv2  -  pin cell - using universe & fill
c     CELLS
10    100   0.06925613   -1        u=1          $ fuel
20    200   0.042910       1 -2    u=1          $ clad
30    300   0.100059       2       u=1          $ water
40    0                   -3      fill=1

c     SURFACES
1     RCC   0. 0. -180.    0. 0. 360.    0.44
2     RCC   0. 0. -180.    0. 0. 360.    0.49
*3    RPP   -.7 .7   -.7 .7     -210. 210.

c     DATA
kcode    1000     1.0   10   50
hsrc    4 -.5 .5     4 -.5 .5     4 -180. 180.
ksrc     0 0 0
imp:n     1 1 1 1
c
m100    92238 2.2380e-2    92235 8.2213e-4   8016 4.6054e-2   $ fuel
m200    40000 4.2910e-2                                      $ clad
m300     1001 6.6706e-2     8016 3.3353e-2                   $ water
mt300    lwtr
```

Result:

$K_{eff} = 1.45296 \pm 0.00239$

Problem fv2 - Comments

- **No cell is needed outside of 40**
  - Neutrons can never reach it, due to reflecting BC on macrobody 3
  - OK to include exterior region
- **Plotting**
  - "XY" plot
  - See what happens when "Level" is changed -- Level 0, Level 1
  - Note that Cell 30 (water) is infinite -- MCNP can't compute volume, uses Vol=0
- **Results**
  - Should be same as previous runs

Problem fv3

Infinite Lattice of Fuel Pins

Problem fv3 - Infinite Lattice of Fuel Pins

- Copy file fv2 to fv3, then edit file fv3

- Convert to infinite lattice (remove reflecting BC)

- Change HSRC card

  HSRC 1 -1.e20 1.e20 1 -1.e20 1.e20 1 -1.e20 1.e20

  This effectively turns off the source entropy checking

```
cp fv2 fv3
[edit fv3]
mcnp6 i=fv3 ip
mcnp6 i=fv3
```

Problem fv3 - Comments

- **Plotting**
  - Change Extent to (5, 5), "XY" plot
  - See what happens when "Level" is changed -- Level 0, Level 1, level -
  - Note that Cell 30 (water) is infinite -- MCNP can't compute volume, uses Vol=0

- **Results**
  - Should be same - within statistics - to previous run with reflecting boundary condition, but not identical

- **fill=1 lat=1**
  - Note that fill=1 means "fill this cell with universe #1"
  - Note that lat=1 means "lattice is rectangular"

Solution - Problem fv3

File fv3

```
Problem fv3  -  infinite lattice of pin cells
c     CELLS
10    100   0.06925613   -1       u=1          $ fuel
20    200   0.042910      1 -2    u=1          $ clad
30    300   0.100059      2       u=1          $ water
40    0                  -3      fill=1   lat=1

c     SURFACES
1     RCC   0. 0. -180.    0. 0. 360.    0.44
2     RCC   0. 0. -180.    0. 0. 360.    0.49
3     RPP   -.7 .7   -.7 .7    -210. 210.     $ infinite in z

c     DATA
kcode    1000     1.0   10   50
hsrc   1 -1.e20 1.e20    1 -1.e20 1.e20    1 -1.e20 1.e20
ksrc     0 0 0
imp:n     1 1 1 1
c
m100    92238 2.2380e-2    92235 8.2213e-4   8016 4.6054e-2   $ fuel
m200    40000 4.2910e-2                                      $ clad
m300     1001 6.6706e-2     8016 3.3353e-2                   $ water
mt300    lwtr
```

Result:

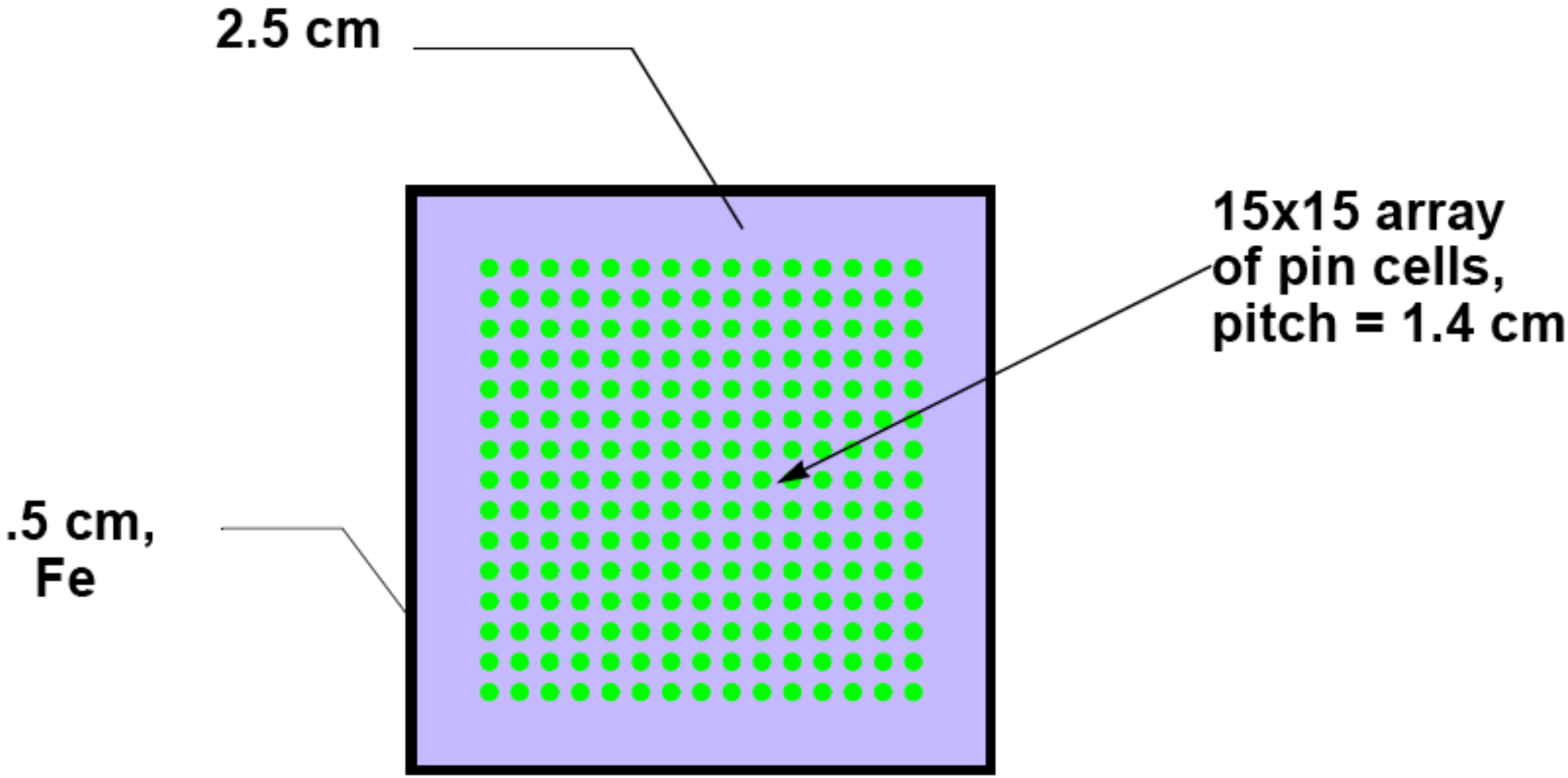
$K_{eff} = 1.44683 \pm 0.00232$

Problem fv4

Fuel Assembly

Problem fv4 - Fuel Assembly

- **Fuel Element - 15x15 Array of Pin Cells**



Problem fv4 - Fuel Assembly

- Copy file fv3 to file fv4, then edit file fv4
- Create 15x15 array:
  - Create infinite lattice of unit cells, origin at center of central pin cell
  - Create box to hold 15x15 array, & fill it with the lattice
  - Add surrounding water & iron

    Fe: 26000 8.3770e-2 atoms/b-cm
- Delete HSRC card
- Use SDEF to define a uniform source in the lattice region

  (see "Basics" lecture for SDEF recipe)
- Assume zero-importance region outside of assembly

```
cp fv3 fv4
[edit fv4]
mcnp6 i=fv4 ip
mcnp6 i=fv4
```

Problem fv4 - Solution

```
Problem fv4  -  Fuel assembly
c     CELLS
10     100   0.06925613   -1        u=1          $ fuel
20     200   0.042910      1 -2     u=1          $ clad
30     300   0.100059      2        u=1          $ water
40     0                  -3      fill=1   lat=1   u=2
50     0                  -4      fill=2        $ lattice region
60     300   0.100059      4 -5                  $ outer water
70     400   0.083770      5 -6                  $ outer Fe
80     0                   6                    $ zero-import region

c     SURFACES
1     RCC   0. 0. -180.    0. 0. 360.    0.44
2     RCC   0. 0. -180.    0. 0. 360.    0.49
3     RPP   -.7 .7         -.7 .7      -210. 210.
4     RPP   -10.5 10.5   -10.5 10.5    -210. 210.
5     RPP   -13.0 13.0   -13.0 13.0    -210. 210.
6     RPP   -13.5 13.5   -13.5 13.5    -210. 210.
```

Problem fv4 - Solution

```
c     DATA
kcode    1000     1.0   10   50
sdef     x=d1 y=d2 z=d3
si1      -10.5 10.5
sp1         0     1
si2      -10.5 10.5
sp2         0     1
si3      -180.   180.
sp3         0     1
imp:n     1 1 1 1    1 1 1 0
c
m100    92238 2.2380e-2    92235 8.2213e-4   8016 4.6054e-2   $ fuel
m200    40000 4.2910e-2                                      $ clad
m300     1001 6.6706e-2     8016 3.3353e-2                   $ water
mt300    lwtr
m400    26000 8.3770e-2                                      $ Fe
```

Result:

$K_{eff} = 0.78907 \pm 0.00357$

Problem fv4 - Comments

- **Results (with vacuum boundary)**

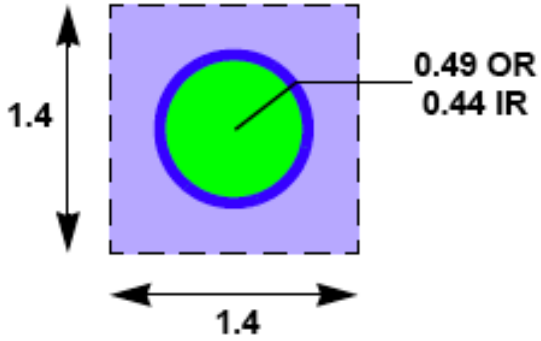
  Keff ~ .789 +- .004

  – **isolated fuel assembly, vacuum boundary conditions**

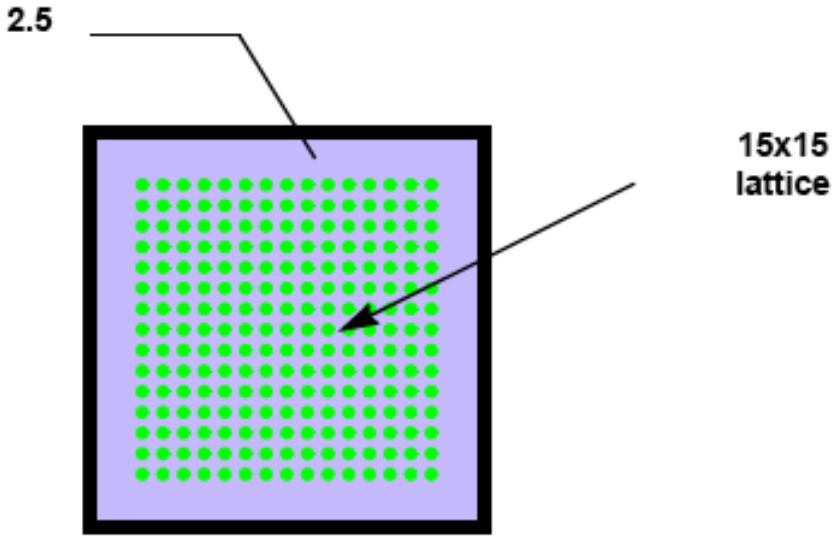
- **Initial source**
  – **Only used for first cycle**
  – **Generally, use simplest source that covers all fuel regions**

Results Obtained So Far

- Pin Cell
  - **Vacuum BC**
  - **Reflective BC**
  - **Infinite lattice**

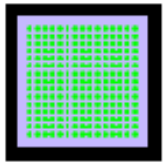


- Fuel Assembly
  - **vacuum BC**

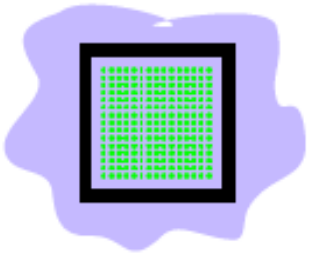


Additional Results Desired

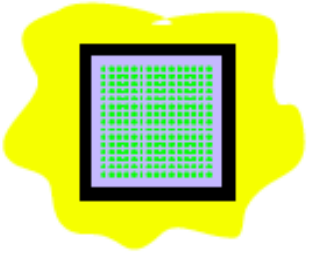
Fuel Element with Reflecting Boundary



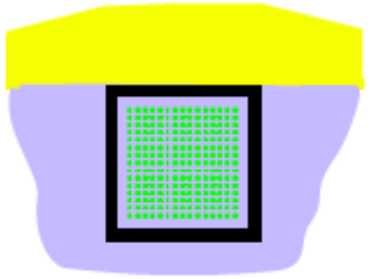
Single Fuel Element,
surrounded by water



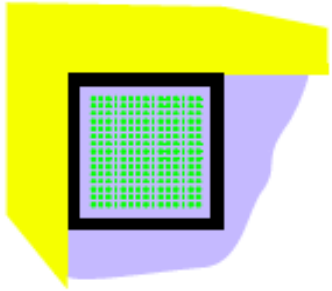
Single Fuel Element,
surrounded by concrete



Single Fuel Element,
water 3 sides, concrete 1 side



Single Fuel Element,
water 2 sides, concrete 2 sides



Problem fv5 - Infinite Array of Fuel Elements

- **Change boundary condition to reflecting**
  - Copy file fv4 to file fv5, then edit fv5 & add '*' before surface 6

```
cp fv4 fv5
[edit fv5]
```

  - Try to plot - looks OK, but error message to screen & output file

```
mcnp6 i=fv5 ip
 surface      3.5 and surface      4.5 are the same.      4.5 will be deleted.
 surface      3.5 and surface      5.5 are the same.      5.5 will be deleted.
 surface      3.5 and surface      6.5 are the same.      6.5 will be deleted.
 fatal error.  boundary condition on identical surface.

 surface      3.6 and surface      4.6 are the same.      4.6 will be deleted.
 surface      3.6 and surface      5.6 are the same.      5.6 will be deleted.
 surface      3.6 and surface      6.6 are the same.      6.6 will be deleted.
 fatal error.  boundary condition on identical surface.
```

  - Change surface 6, so that it doesn't coincide with surfaces 4 & 5

```
*6    RPP  -13.5 13.5  -13.5 13.5   -210.01  210.01
```

Plot again - no error message. Then run mcnp6.

Problem fv5

```
Problem fv5  -  Fuel assembly - reflecting BC
c
c     CELLS
10     100   0.06925613   -1        u=1          $ fuel
20     200   0.042910      1 -2     u=1          $ clad
30     300   0.100059      2        u=1          $ water
40     0                  -3      fill=1   lat=1   u=2
50     0                  -4      fill=2         $ lattice region
60     300   0.100059      4 -5                   $ outer water
70     400   0.083770      5 -6                   $ outer Fe
80     0                   6                     $ zero-import region

c     SURFACES
1     RCC   0. 0. -180.    0. 0. 360.    0.44
2     RCC   0. 0. -180.    0. 0. 360.    0.49
3     RPP   -.7 .7        -.7 .7        -210. 210.
4     RPP   -10.5 10.5    -10.5 10.5    -210. 210.
5     RPP   -13.0 13.0    -13.0 13.0    -210. 210.
*6    RPP   -13.5 13.5    -13.5 13.5    -210.01 210.01
```

Problem fv5 - Comments

```
c     DATA
kcode    1000     1.0   10   50
sdef     x=d1 y=d2 z=d3
si1      -10.5 10.5
sp1         0     1
si2      -10.5 10.5
sp2         0     1
si3      -180. 180.
sp3         0     1
imp:n     1 1 1 1    1 1 1 0
c
m100    92238 2.2380e-2    92235 8.2213e-4   8016 4.6054e-2   $ fuel
m200    40000 4.2910e-2                                      $ clad
m300     1001 6.6706e-2     8016 3.3353e-2                   $ water
mt300    lwtr
m400    26000 8.3770e-2                                      $ Fe
```

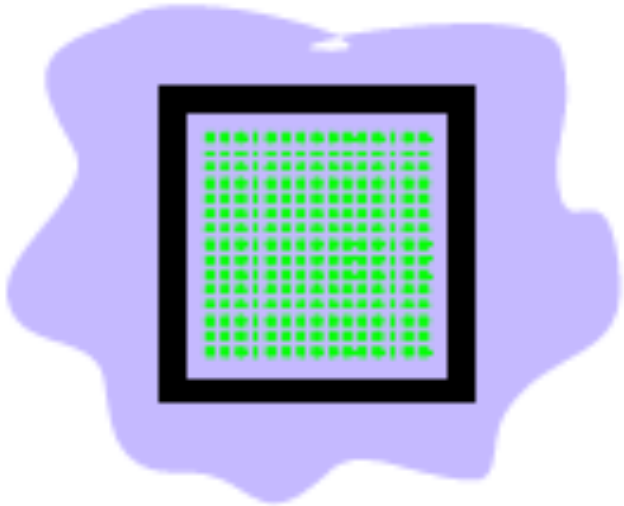
Result:

$K_{eff} = 1.11616 \pm 0.00327$

Problem fv6 - Fuel Element Surrounded by Water

- **Modify fv4**
  - Put water arround fuel element
  - Put a large box around everything, +- 1000, with imp:n 0

```
cp fv4 fv6
[edit fv6]
mcnp5 i=fv6 ip
mcnp5 i=fv6
```



Problem fv6

```
Problem fv6  -  Fuel assembly in water
c
c    CELLS
10    100  0.06925613  -1       u=1          $ fuel
20    200  0.042910     1 -2    u=1          $ clad
30    300  0.100059     2       u=1          $ water
40    0                -3      fill=1   lat=1   u=2
50    0                -4      fill=2       $ lattice region
60    300  0.100059     4 -5                 $ outer water
70    400  0.083770     5 -6                 $ outer Fe
80    300  0.100059     6  -7                $ surrounding water
90    0                7

c    SURFACES
1    RCC  0. 0. -180.   0. 0. 360.   0.44
2    RCC  0. 0. -180.   0. 0. 360.   0.49
3    RPP  -.7 .7        -.7 .7       -210. 210.
4    RPP  -10.5  10.5   -10.5 10.5   -210. 210.
5    RPP  -13.0  13.0   -13.0 13.0   -210. 210.
6    RPP  -13.5  13.5   -13.5 13.5   -210. 210.
7    RPP   -1000 1000  -1000 1000   -1000 1000
```

Problem fv6 - Comments

```
c    DATA
imp:n    1 1 1 1 1 1 1 1 0
c
kcode   1000    1.0  10  50
sdef    x=d1 y=d2 z=d3
si1     -10.5 10.5
sp1        0    1
si2     -10.5 10.5
sp2        0    1
si3     -180. 180.
sp3        0    1
c
m100   92238 2.2380e-2   92235 8.2213e-4  8016 4.6054e-2  $ fuel
m200   40000 4.2910e-2                                   $ clad
m300    1001 6.6706e-2    8016 3.3353e-2                 $ water
mt300   lwtr
m400   26000 8.3770e-2                                   $ Fe
```

Result:

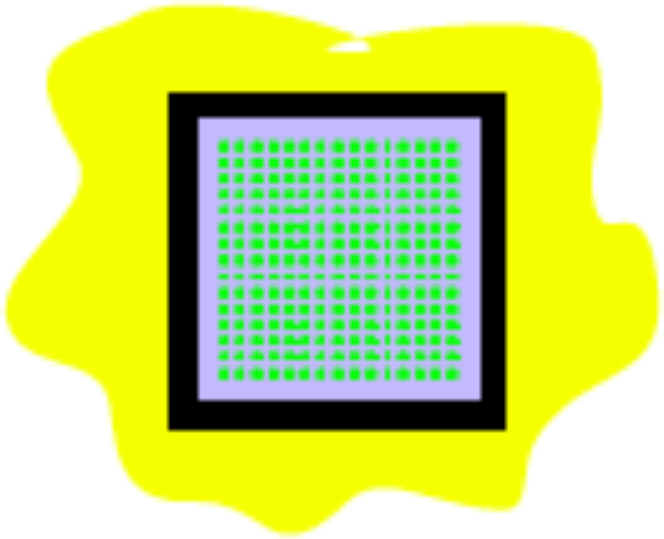
$K_{eff} = 0.87221 \pm 0.00426$

Problem fv7 - Fuel Element Surrounded by Concrete

- Modify fv6
  - **Replace surronging water by concrete**

rho =0.0725757

H	**1001**	**5.5437e-3, with S(α,β) lwtr**
C	**6000**	**6.9793e-3**
Si	**14000**	**7.7106e-3**
Ca	**20000**	**8.9591e-3**
O	**8016**	**4.3383e-2**



```
cp fv6 fv7
[edit fv7]
mcnp5 i=fv7 ip
mcnp5 i=fv7
```

Problem fv7

```
Problem fv7  -  Fuel assembly in concrete
c
c    CELLS
10    100  0.06925613  -1       u=1          $ fuel
20    200  0.042910     1 -2    u=1          $ clad
30    300  0.100059     2       u=1          $ water
40    0                -3      fill=1   lat=1   u=2
50    0                -4      fill=2
60    300  0.100059     4 -5                 $ outer water
70    400  0.083770     5 -6                 $ outer Fe
80    500  0.0725757    6  -7                $ surrounding concrete
90    0                7

c    SURFACES
1    RCC  0. 0. -180.   0. 0. 360.   0.44
2    RCC  0. 0. -180.   0. 0. 360.   0.49
3    RPP  -.7 .7        -.7 .7      -180. 180.
4    RPP   -10.5 10.5   -10.5 10.5  -210. 210.
5    RPP   -13.0 13.0   -13.0 13.0  -210. 210.
6    RPP   -13.5 13.5   -13.5 13.5  -210. 210.
7    RPP   -1000 1000   -1000 1000   -1000 1000
```

Problem fv7 - Comments

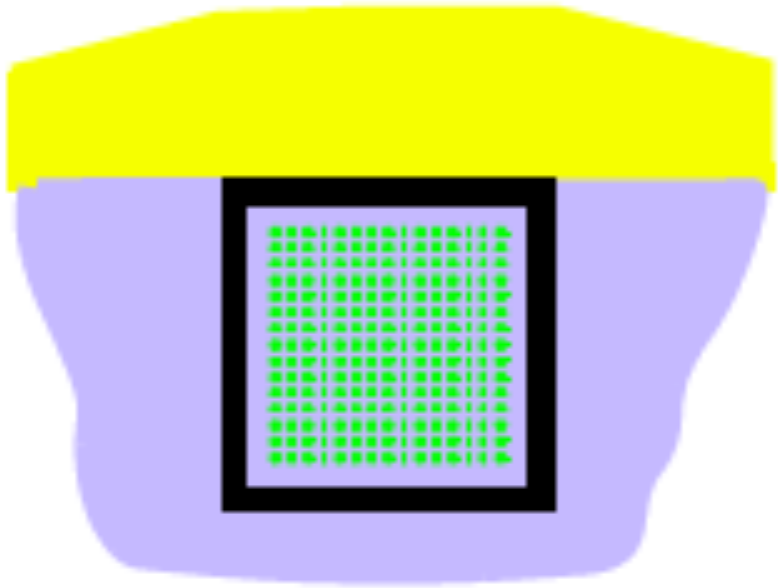
```
c     DATA
imp:n     1 1 1 1 1 1 1 1 0
kcode    1000     1.0   10   50
sdef     x=d1 y=d2 z=d3
si1      -10.5 10.5
sp1         0     1
si2      -10.5 10.5
sp2         0     1
si3      -180. 180.
sp3         0     1
m100    92238 2.2380e-2    92235 8.2213e-4   8016 4.6054e-2   $ fuel
m200    40000 4.2910e-2                                      $ clad
m300     1001 6.6706e-2     8016 3.3353e-2                   $ water
mt300    lwtr
m400    26000 8.3770e-2                                      $ Fe
m500     1001 5.5437e-3   6000 6.9793e-3 14000 7.7106e-3     $ concrete
        20000 8.9591e-3   8016 4.3383e-2
mt500    lwtr
```

Result:
$K_{eff} = 0.91111 \pm 0.00474$

Problem fv8 - Assy, Concrete on 1 Side, Water on 3 Sides

- **Modify fv7**

```
cp fv7 fv8
[edit fv8]
mcnp6 i=fv8 ip
mcnp6 i=fv8
```



Problem fv8

```
Problem fv8  -  Fuel assembly, cnocrete 1 side, water 3 sides
c    CELLS
10    100  0.06925613  -1       u=1          $ fuel
20    200  0.042910     1 -2    u=1          $ clad
30    300  0.100059     2       u=1          $ water
40    0                -3      fill=1   lat=1  u=2
50    0                -4      fill=2
60    300  0.100059     4 -5                 $ outer water
70    400  0.083770     5 -6                 $ outer Fe
80    500  0.0725757    -7          $ surrounding concrete
90    300  0.100059     6 -8        $ outer water
100   0                 7  8

c    SURFACES
1    RCC  0. 0. -180.   0. 0. 360.   0.44
2    RCC  0. 0. -180.   0. 0. 360.   0.49
3    RPP  -.7 .7        -.7 .7      -180. 180.
4    RPP   -10.5 10.5   -10.5 10.5  -210. 210.
5    RPP   -13.0 13.0   -13.0 13.0  -210. 210.
6    RPP   -13.5 13.5   -13.5 13.5  -210. 210.
7    RPP   -1000 1000    13.5 1000   -1000 1000
8    RPP   -1000 1000  -1000 13.5   -1000 1000
9    RPP   -1000 1000  -1000 1000   -1000 1000
```

Problem fv8 - Comments

```
c     DATA
imp:n     1 1 1 1 1 1 1 1 1 0
kcode    1000     1.0   10   50
sdef     x=d1 y=d2 z=d3
si1      -10.5 10.5
sp1         0     1
si2      -10.5 10.5
sp2         0     1
si3      -180. 180.
sp3         0     1
m100    92238 2.2380e-2    92235 8.2213e-4   8016 4.6054e-2   $ fuel
m200    40000 4.2910e-2                                      $ clad
m300     1001 6.6706e-2     8016 3.3353e-2                   $ water
mt300    lwtr
m400    26000 8.3770e-2                                      $ Fe
m500     1001 5.5437e-3   6000 6.9793e-3 14000 7.7106e-3     $ concrete
        20000 8.9591e-3   8016 4.3383e-2
mt500    lwtr
```

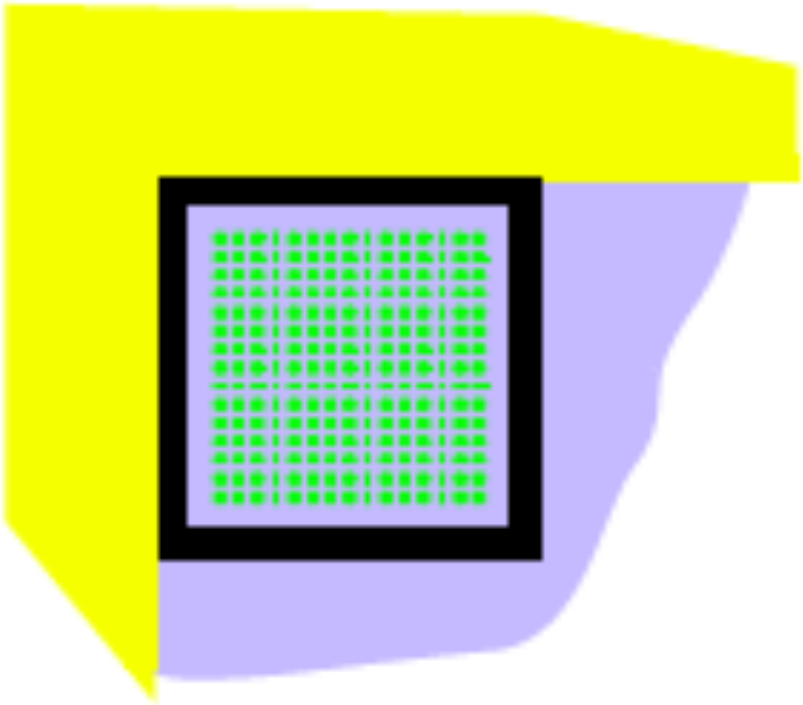
Result:

$K_{eff} = 0.87127 \pm 0.00335$

Problem fv9 - Concrete on 2 Sides, Water on 2 Sides

- Modify fv8

```
cp fv8 fv9
[edit fv9]
mcnp i=fv9 ip
mcnp n=fv9
```



Problem fv9

```
Problem fv9  -  Fuel assembly, concrete 2 sides, water 2 sides
c
c    CELLS
10    100  0.06925613  -1       u=1          $ fuel
20    200  0.042910     1 -2    u=1          $ clad
30    300  0.100059     2       u=1          $ water
40    0                -3      fill=1   lat=1   u=2
50    0                -4      fill=2
60    300  0.100059     4 -5                 $ outer water
70    400  0.083770     5 -6                 $ outer Fe
80    500  0.0725757    7 -8       $ surrounding concrete
90    300  0.100059     6 -7       $ outer water
100   0                 8

c    SURFACES
1    RCC  0. 0. -180.   0. 0. 360.   0.44
2    RCC  0. 0. -180.   0. 0. 360.   0.49
3    RPP  -.7 .7        -.7 .7      -180. 180.
4    RPP   -10.5 10.5   -10.5 10.5  -210. 210.
5    RPP   -13.0 13.0   -13.0 13.0  -210. 210.
6    RPP   -13.5 13.5   -13.5 13.5  -210. 210.
7    RPP   -13.5 1000   -1000 13.5  -1000 1000
8    RPP   -1000 1000   -1000 1000  -1000 1000
```

Problem fv9 - Comments

```
c    DATA
imp:n    1 1 1 1 1 1 1 1 1 0
kcode   1000    1.0  10  50
sdef    x=d1 y=d2 z=d3
si1     -10.5 10.5
sp1        0    1
si2     -10.5 10.5
sp2        0    1
si3     -180. 180.
sp3        0    1
m100   92238 2.2380e-2   92235 8.2213e-4  8016 4.6054e-2  $ fuel
m200   40000 4.2910e-2                                   $ clad
m300    1001 6.6706e-2    8016 3.3353e-2                 $ water
mt300   lwtr
m400   26000 8.3770e-2                                   $ Fe
m500    1001 5.5437e-3  6000 6.9793e-3 14000 7.7106e-3   $ concrete
       20000 8.9591e-3  8016 4.3383e-2
mt500   lwtr
```

Result:

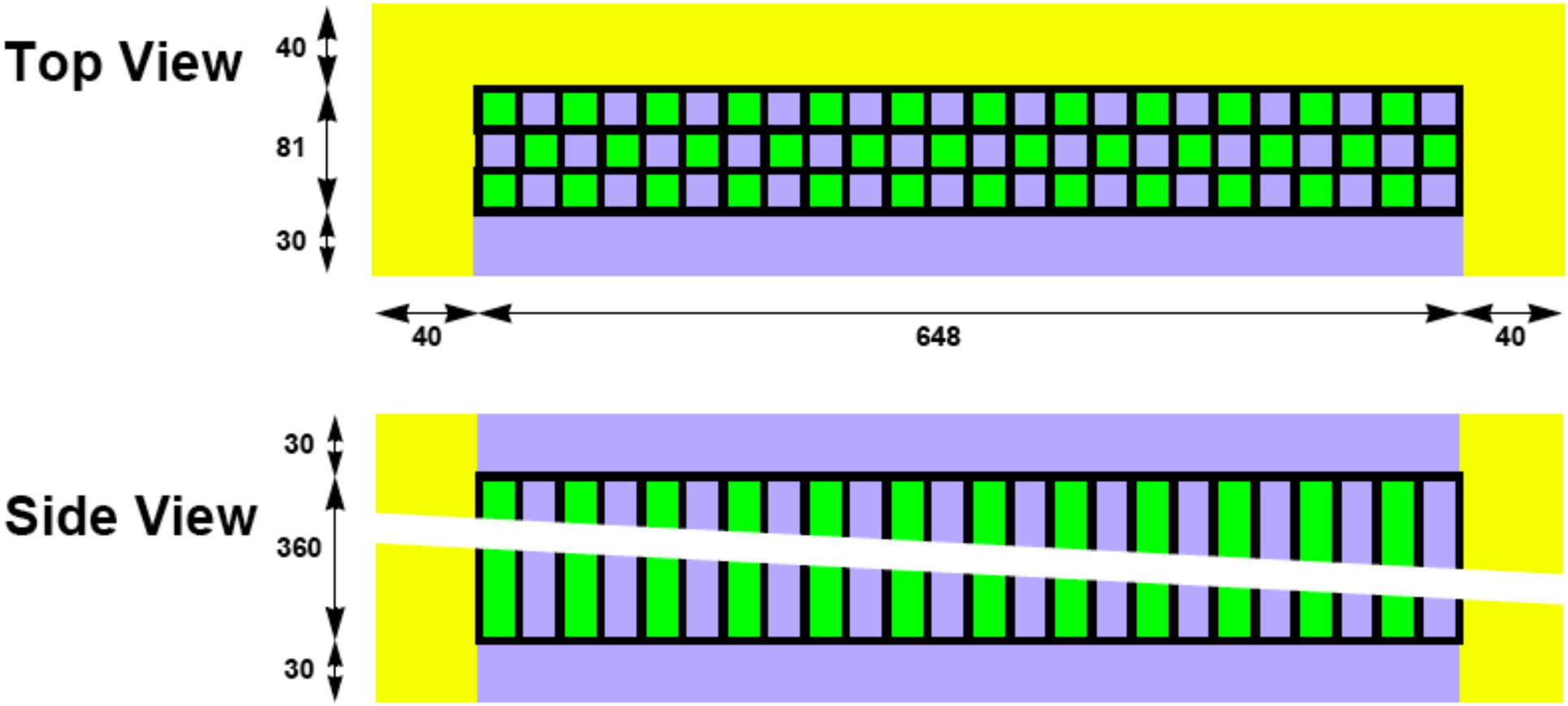
$K_{eff} = 0.89327 \pm 0.00485$

Summary So Far

[using ENDF/B-VII.0 cross-sections]

- **Pin Cell**
  - vacuum BC .01216 ± .00011
  - reflective BC 1.45296 ± .00239
  - infinite lattice 1.45296 ± .00239
- **Fuel Element**
  - vacuum BC .78907 ± .00357
  - water 4 sides .87221 ± .00426
  - water 3 sides, concrete 1 side .87127 ± .00335
  - water 2 sides, concrete 2 sides .89327 ± .00485
  - concrete 4 sides .91111 ± .00474
  - infinite lattice 1.11616 ± .00327

Problem fvf - Fuel Vault, Full Model



Problem fvf

- **Copy fv9 to fvf, then edit**
  - **create "water element" -- Fe box, with water inside**
  - **create 24x3 lattice, filled with alternating fuel & water elements**
  - **add outer regions**
  - **source...**

Problem fvf

```
Problem fvf  -  Fuel storage vault
c    CELLS
10    100   0.06925613  -1       u=1          $ fuel
20    200   0.042910     1 -2    u=1          $ clad
30    300   0.100059     2       u=1          $ water
c    =====> fuel lattice, infinite array of pins in water
40    0                 -3      fill=1   lat=1  u=2
c    =====> fuel element
50    0                 -4      fill=2  u=3   $ fuel lattice
60    300   0.100059     4 -5            u=3   $ water gap
70    400   0.083770     5               u=3   $ Fe
c    =====> water element
80    300   0.100059    -5               u=4   $ water
90    400   0.083770     5               u=4   $ Fe
c    =====> element lattice, infinite
100   0                 -6       u=5  lat=1   fill= 0:23 0:2  0:0
       3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
       4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3
       3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
c    =====> full model
110   0                 -7         fill=5     $ lattice of elements
120   300   0.100059    -8 7                  $ outside water
130   500   0.0725757   -9 8                  $ outside concrete
140   0                  9
```

Problem fvf

```
c     SURFACES
1     RCC  0. 0. -180.    0. 0. 360.    0.44
2     RCC  0. 0. -180.    0. 0. 360.    0.49
3     RPP  -.7 .7         -.7 .7       -180. 180.
4     RPP   -10.5 10.5    -10.5 10.5   -210. 210.
5     RPP   -13.0 13.0    -13.0 13.0   -210. 210.
6     RPP   -13.5 13.5    -13.5 13.5   -210. 210.
7     RPP   -13.5 634.5   -13.5 67.5   -180. 180.
8     RPP   -13.5 634.5   -43.5 67.5   -210. 210.
9     RPP   -53.5 674.5   -43.5 107.5  -210. 210.
```

Problem fvf

```
c    DATA
imp:n    1 12r 0
c
kcode   1000    1.0  10  50
hsrc  24 -10.5 631.5   3 -10.5 64.5   1 -180. 180.
sdef    x=d1 y=d2 z=d3
c    =====> source in element (0 0 0)
c    si1     -10.5 10.5
c    si2     -10.5 10.5
c    =====> source in all elements
c    si1     -10.5 631.5
c    si2     -10.5 64.5
c    =====> source in element (0 2 0)
si1     -10.5 10.5
si2      40.5 64.5
c
sp1       0    1
sp2       0    1
si3     -180. 180.
sp3       0    1
```

Problem fvf

```
c
c    =====> material cards
m100   92238 2.2380e-2   92235 8.2213e-4  8016 4.6054e-2  $ fuel
m200   40000 4.2910e-2                                    $ clad
m300    1001 6.6706e-2    8016 3.3353e-2                  $ water
mt300   lwtr
m400   26000 8.3770e-2                                    $ Fe
m500    1001 5.5437e-3  6000 6.9793e-3 14000 7.7106e-3    $ concrete
       20000 8.9591e-3  8016 4.3383e-2
mt500   lwtr
```

Problem fvf - Comments

- **Results**
  - source in (0 0 0) element Keff ~ .88836 +- .00428
  - source in (0 2 0) element .89375 +- .00461
  - uniform source in elements .88618 +- .00383
  - long run, 100 M histories .89068 +- .00001

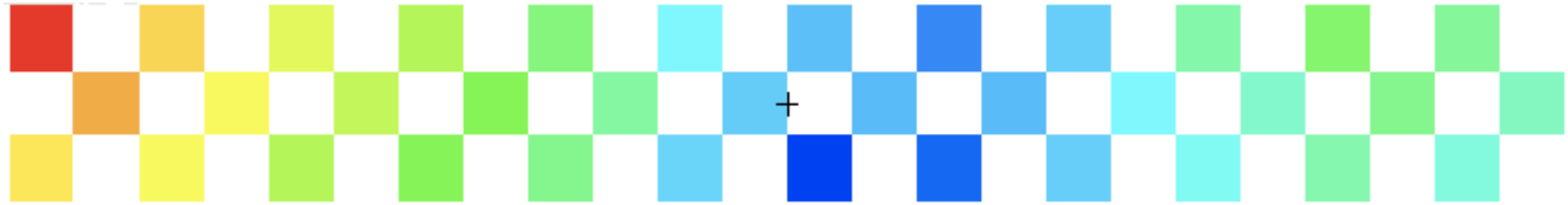
Based on ENDF/B-VII.0 nuclear libraries

Problem fvf - Fission Distribution

- **Converged fission distribution, from MCNP reference case (90 M histories):**

37084	11171	4239	1494	871	615	354	413	235	120	13	4
15225	5659	2253	958	587	426	321	246	143	50	4	7
7972	5226	2118	891	466	264	237	173	108	43	9	3

(normalized to sum = 100000)



Case Study #3

PWR Benchmark Example, with Edits of Assembly Powers

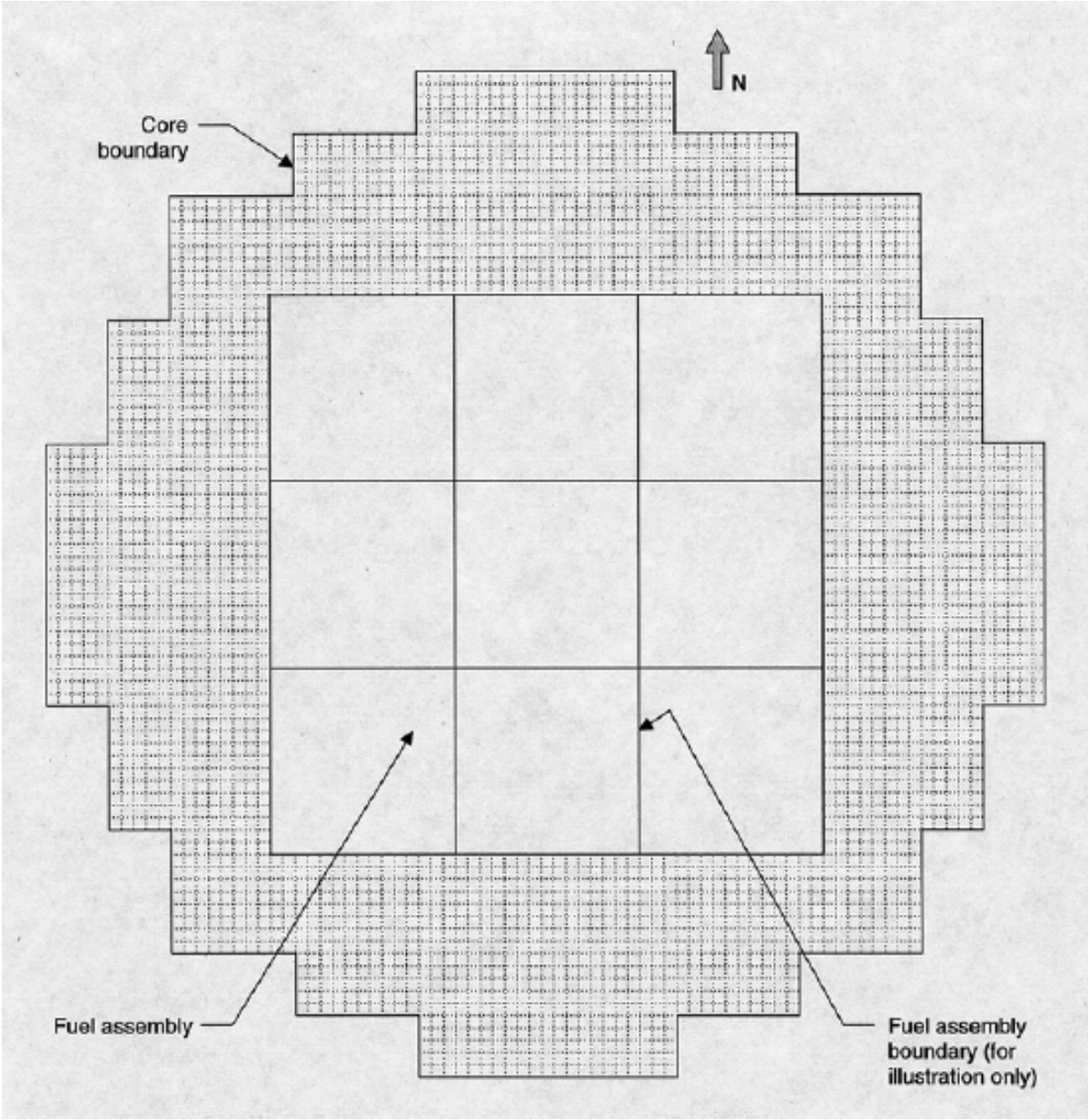
(Using tallies in repeated structures)

B&W Core XI, Loading 2

- Experiment simulates the core of a PWR with 15x15 Assemblies
- Core contains contains 4,808 fuel pins (2.459 ± 0.002 wt.%)
- Core is slightly supercritical, with 1335.5 PPM of boron
- Specifications are taken from International Handbook of Evaluated Criticality Safety Benchmark Experiments (LEU-COMP-THERM-008, case 2)
- Comparison of direct and repeated-structures calculations

	Direct	Repeated Structures
Lines of Input:	3,468	164
Relative Execution Time:	1.00	1.42

Core XI Layout



Assembly Arrangement for Loading 2

					W				W					
			W								W			
		W			W				W			W		
							W							
		W			W				W			W		
			W								W			
					W				W					

W = Water Hole

Input for Repeated-structures Calculation

```
B&W Criticals Core XI Load 2 1335.5 PPM Axially Uniform Octant RFD Edit
c LEU-COMP-THERM-008 ENDF/B-VI
1  1 0.068525  -13     u=1                          $ Fuel Pin
2  2 0.055323   13 -14 u=1                          $ Cladding
3  3 0.10018    14     u=1                          $ Water
4  0           -16 15 -18 17 u=2 lat=1 fill=1      $ Pin Cell
c  15x15 Assembly Lattice
5  3 0.10018 -16 15 -18 17 u=3 lat=1  fill=-7:7 -7:7 0:0
              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
              1 1 1 1 1 3 1 1 1 3 1 1 1 1 1
              1 1 1 3 1 1 1 1 1 1 1 3 1 1 1
              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
              1 1 3 1 1 3 1 1 1 3 1 1 3 1 1
              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
              1 1 1 1 1 1 1 3 1 1 1 1 1 1 1
              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
              1 1 3 1 1 3 1 1 1 3 1 1 3 1 1
              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
              1 1 1 3 1 1 1 1 1 1 1 3 1 1 1
              1 1 1 1 1 3 1 1 1 3 1 1 1 1 1
              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Input (cont'd)

```
6  0                    -20 19 -22 21 u=4 lat=1
            fill=-2:2 -2:2 0:0
                         3 3 3 3 3
                         3 3 3 3 3
                         3 3 3 3 3
                         3 3 3 3 3
                         3 3 3 3 3
7   0                    1 -2 3 4 -6       fill=4   $ Inner Core
8   0                    1 -2 3 6 -9 -10   fill=2   $ Buffer Zone
9   0                    1 -2 6 -8 10 -11  fill=2   $ Buffer Zone
10  0                    1 -2 6 -7 11 -12  fill=2   $ Buffer Zone
11  0                    1 -2 4 -7 12      fill=2   $ Buffer Zone
12  3 0.10018            1 -2 3 -5 9 -10             $ Reflector
13  3 0.10018            1 -2 -5 8 10 -11            $ Reflector
14  3 0.10018            1 -2 4 -5 7 11              $ Reflector
15 0                     -1:2:-3:-4:5

c Problem Boundaries
1   pz  -81.662                  $ Bottom
2   pz   81.662                  $ Top
*3  py   0.0                     $ Front Edge
*4  p   1.0 -1.0 0.0 0.0         $ Diagonal
5   cz   76.200                  $ Tank IR
```

Input (cont'd)

```
c Interior Boundaries
6   px  36.80470        $ Inner Core
7   px  49.89070        $ Buffer
8   px  58.06950        $ Buffer
9   px  66.24830        $ Buffer
10  py  17.17550        $ Buffer
11  py  33.53310        $ Buffer
12  py  36.80470        $ Inr Core
c Pin Cell Boundaries
13  cz  0.514858        $ Pellet OR
14  cz  0.602996        $ Cladding OR
15  px  -0.81788        $ Left Edge
16  px  0.81788         $ Right Edge
17  py  -0.81788        $ Front Edge
18  py  0.81788         $ Back Edge
c Assembly Boundaries
19  px  -12.26820       $ Left Edge
20  px   12.26820       $ Right Edge
21  py  -12.26820       $ Front Edge
22  py   12.26820       $ Back Edge
mode n
kcode 10000  1.0  50  550
imp:n 1.0 13r 0.0
```

Input (cont'd)

```
sdef x=d1 y=d2 z=d3
si1 12.5 36.0
Sp1   0 1
si2 0.1 12.5
sp2   0 1
si3 -81.6 81.6
sp3   0 1
c Edit fast and thermal fluxes in pins of central assembly
f4:n            (1<(5[1 0 0])<6[0 0 0]<7) (1<(5[2 0 0])<6[0 0 0]<7)
                (1<(5[3 0 0])<6[0 0 0]<7) (1<(5[4 0 0])<6[0 0 0]<7)
                (1<(5[5 0 0])<6[0 0 0]<7) (1<(5[6 0 0])<6[0 0 0]<7)
                (1<(5[7 0 0])<6[0 0 0]<7)
                (1<(5[1 1 0])<6[0 0 0]<7) (1<(5[2 1 0])<6[0 0 0]<7)
                (1<(5[3 1 0])<6[0 0 0]<7) (1<(5[4 1 0])<6[0 0 0]<7)
                (1<(5[5 1 0])<6[0 0 0]<7) (1<(5[6 1 0])<6[0 0 0]<7)
                (1<(5[7 1 0])<6[0 0 0]<7)
                (1<(5[3 2 0])<6[0 0 0]<7) (1<(5[4 2 0])<6[0 0 0]<7)
                (1<(5[6 2 0])<6[0 0 0]<7) (1<(5[7 2 0])<6[0 0 0]<7)
                (1<(5[3 3 0])<6[0 0 0]<7) (1<(5[4 3 0])<6[0 0 0]<7)
                (1<(5[5 3 0])<6[0 0 0]<7) (1<(5[6 3 0])<6[0 0 0]<7)
                (1<(5[7 3 0])<6[0 0 0]<7)
                (1<(5[5 4 0])<6[0 0 0]<7) (1<(5[6 4 0])<6[0 0 0]<7)
                (1<(5[7 4 0])<6[0 0 0]<7)
                (1<(5[5 5 0])<6[0 0 0]<7) (1<(5[6 5 0])<6[0 0 0]<7)
```

Input (cont'd)

```
                (1<(5[7 5 0])<6[0 0 0]<7)
                (1<(5[6 6 0])<6[0 0 0]<7) (1<(5[7 6 0])<6[0 0 0]<7)
                (1<(5[7 7 0])<6[0 0 0]<7) t
Sd4             68.006  68.006  68.006  68.006  68.006  68.006  68.006
                68.006 136.011 136.011 136.011 136.011 136.011 136.011
                               136.011 136.011         136.011 136.011
                                68.006 136.011 136.011 136.011 136.011
                                                136.011 136.011 136.011
                                                 68.006 136.011 136.011
                                                        68.006 136.011
                                                               68.006

                3536.292
c Edit fission distribution in central assembly
f14:n  (1<(5[1 0 0])<6[0 0 0]<7) (1<(5[2 0 0])<6[0 0 0]<7)
                (1<(5[3 0 0])<6[0 0 0]<7) (1<(5[4 0 0])<6[0 0 0]<7)
                (1<(5[5 0 0])<6[0 0 0]<7) (1<(5[6 0 0])<6[0 0 0]<7)
                (1<(5[7 0 0])<6[0 0 0]<7)
                (1<(5[1 1 0])<6[0 0 0]<7) (1<(5[2 1 0])<6[0 0 0]<7)
                (1<(5[3 1 0])<6[0 0 0]<7) (1<(5[4 1 0])<6[0 0 0]<7)
                (1<(5[5 1 0])<6[0 0 0]<7) (1<(5[6 1 0])<6[0 0 0]<7)
                (1<(5[7 1 0])<6[0 0 0]<7)
                (1<(5[3 2 0])<6[0 0 0]<7) (1<(5[4 2 0])<6[0 0 0]<7)
                (1<(5[6 2 0])<6[0 0 0]<7) (1<(5[7 2 0])<6[0 0 0]<7)
                (1<(5[3 3 0])<6[0 0 0]<7) (1<(5[4 3 0])<6[0 0 0]<7)
```

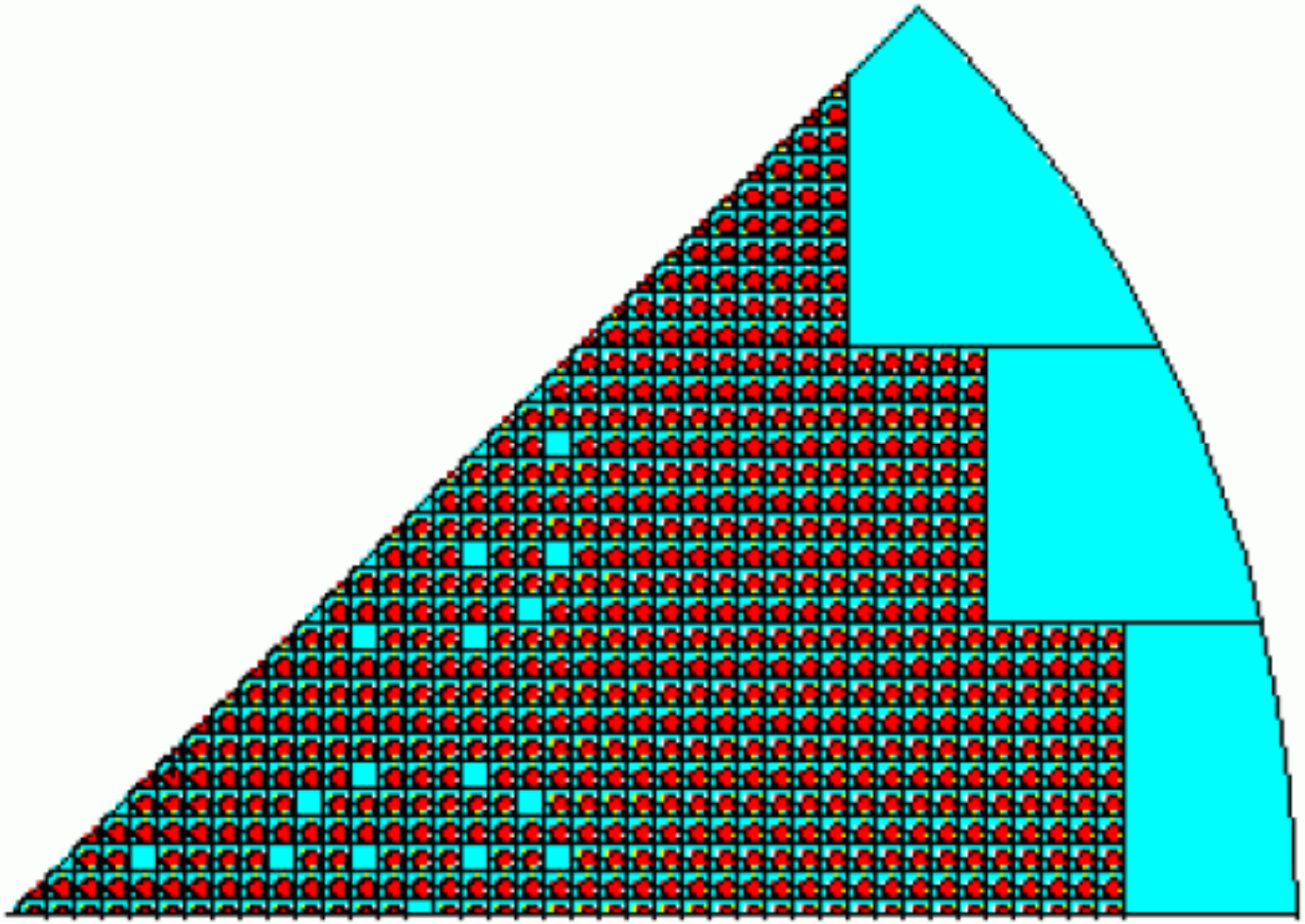
Input (cont'd)

```
                (1<(5[5 3 0])<6[0 0 0]<7) (1<(5[6 3 0])<6[0 0 0]<7)
                (1<(5[7 3 0])<6[0 0 0]<7)
                (1<(5[5 4 0])<6[0 0 0]<7) (1<(5[6 4 0])<6[0 0 0]<7)
                (1<(5[7 4 0])<6[0 0 0]<7)
                (1<(5[5 5 0])<6[0 0 0]<7) (1<(5[6 5 0])<6[0 0 0]<7)
                (1<(5[7 5 0])<6[0 0 0]<7)
                (1<(5[6 6 0])<6[0 0 0]<7) (1<(5[7 6 0])<6[0 0 0]<7)
                (1<(5[7 7 0])<6[0 0 0]<7) t
fm14 -1.0 1 -6
sd14 68.006  68.006  68.006  68.006  68.006  68.006  68.006
     68.006 136.011 136.011 136.011 136.011 136.011 136.011
                    136.011 136.011         136.011 136.011
                     68.006 136.011 136.011 136.011 136.011
                                    136.011 136.011 136.011
                                     68.006 136.011 136.011
                                            68.006 136.011
                                                    68.006
     3536.292
c Fuel (2.459 w/o with B-10 for impurities)
m1    5010.66c 2.6055e-7
      8016.62c 4.5683e-2
      92234.66c 4.5689e-6   92235.66c 5.6868e-4
      92238.66c 2.2268e-2
```

Input (cont'd)

```
c Aluminum 6061 cladding
m2    12000.62c 6.2072e-4
      13027.62c 5.3985e-2
      14028.66c 2.9726e-4    14029.62c 1.5051e-5
      14030.66c 9.9130e-6
      22000.62c 4.7263e-5
      24050.62c 2.5214e-6    24052.62c 4.8622e-5
      24053.62c 5.5128e-6    24054.62c 1.3724e-6
      25055.62c 4.1191e-5
      26054.62c 1.1157e-5    26056.62c 1.7344e-4
      26057.62c 3.9711e-6    26058.62c 5.2948e-7
      29063.62c 4.1054e-5    29065.62c 1.8299e-5
c Water with 1335.5 PPM
m3    1001.62c 6.6737e-2
      8016.62c 3.3369e-2
      5010.66c 1.4821e-5     5011.66c 5.9657e-5
mt3 lwtr.60t
totnu
prdmp j 275
Print

end of input
```

MCNP Model of Symmetric Octant



Fission Distribution

Central assembly of B&W Core XI, Configuration 2 (5M active histories)

Water Hole	1.107 ± 0.002 1.108 ± 0.012 1.111 ± 0.012	1.026 ± 0.006 1.041 ± 0.011 1.058 ± 0.011	1.000 ± 0.001 1.020 ± 0.010 1.015 ± 0.010	1.025 ± 0.007 1.031 ± 0.011 1.012 ± 0.010	1.026 ± 0.003 1.006 ± 0.010 1.023 ± 0.010	0.980 ± 0.021 0.992 ± 0.010 0.991 ± 0.010	0.983 ± 0.008 0.972 ± 0.010 0.986 ± 0.010
	1.068 ± 0.002 1.079 ± 0.011 1.079 ± 0.011	1.075 ± 0.000 1.103 ± 0.008 1.088 ± 0.008	1.036 ± 0.007 1.052 ± 0.008 1.052 ± 0.008	1.047 ± 0.004 1.058 ± 0.008 1.052 ± 0.008	1.098 ± 0.006 1.081 ± 0.008 1.082 ± 0.008	1.026 ± 0.023 1.022 ± 0.008 1.014 ± 0.008	1.003 ± 0.031 0.975 ± 0.007 0.994 ± 0.008
		Water Hole	1.116 ± 0.012 1.131 ± 0.008 1.119 ± 0.008	1.118 ± 0.011 1.138 ± 0.008 1.116 ± 0.008	Water Hole	1.070 ± 0.010 1.087 ± 0.008 1.077 ± 0.008	0.961 ± 0.010 0.994 ± 0.007 0.987 ± 0.007
			1.091 ± 0.009 1.107 ± 0.011 1.095 ± 0.011	1.145 ± 0.008 1.134 ± 0.008 1.138 ± 0.008	1.133 ± 0.010 1.119 ± 0.008 1.116 ± 0.008	1.032 ± 0.026 1.001 ± 0.007 1.027 ± 0.008	0.924 ± 0.006 0.973 ± 0.007 0.979 ± 0.007
				Water Hole	1.109 ± 0.007 1.092 ± 0.008 1.096 ± 0.008	1.007 ± 0.014 0.988 ± 0.007 0.998 ± 0.007	0.974 ± 0.026 0.954 ± 0.007 0.961 ± 0.007
					1.015 ± 0.002 1.025 ± 0.011 1.014 ± 0.010	0.973 ± 0.023 0.962 ± 0.007 0.960 ± 0.007	0.971 ± 0.012 0.945 ± 0.007 0.952 ± 0.007
						0.970 ± 0.006 0.957 ± 0.010 0.939 ± 0.010	0.950 ± 0.005 0.933 ± 0.007 0.944 ± 0.007
						Measured Direct Repeated Structures	0.920 ± 0.013 0.933 ± 0.010 0.930 ± 0.010

RMS Differences between Sample Means

Direct: 0.021
Repeated Structures: 0.019

keff

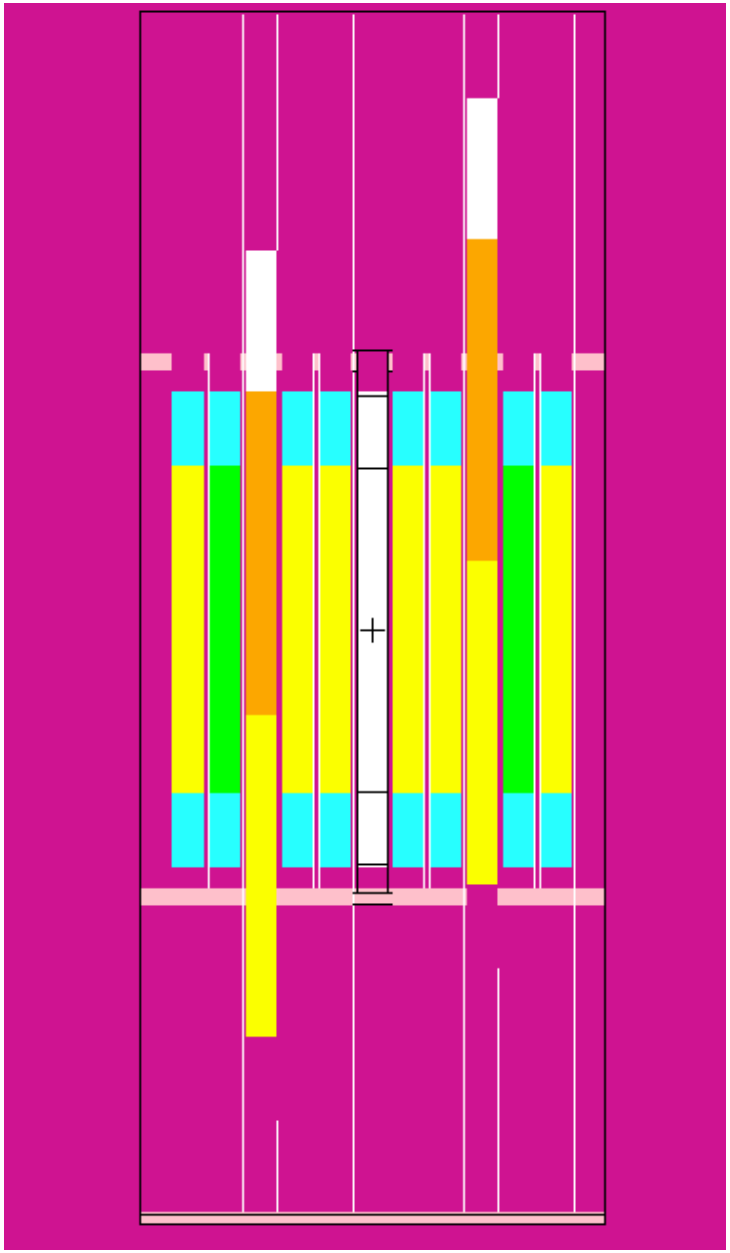
Measured: 1.0007 ± 0.0006
Direct: 0.9970 ± 0.0003
Repeated Structures: 0.9977 ± 0.0003

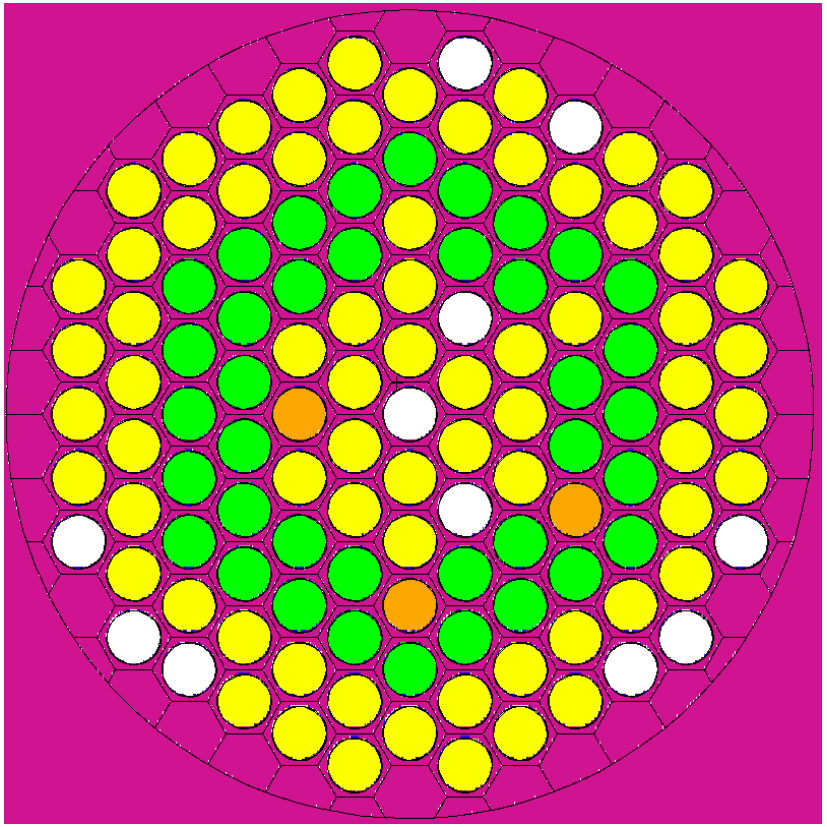
Case Study #4

--

Hexagonal Geometry, Simplified TRIGA Reactor

Simplified TRIGA Reactor

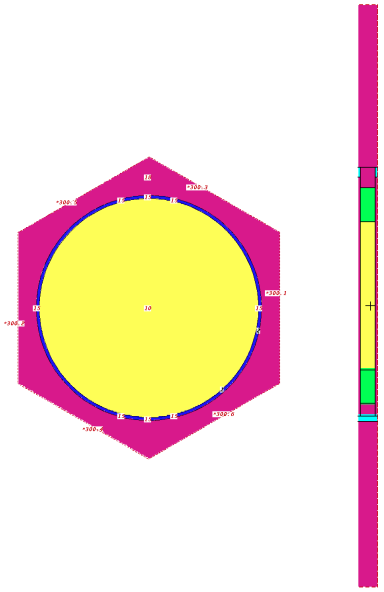




Based on reference:
ORNL-TM-13656, June 1998
Triga type OAEP Thai-Research-Reactor

Simplified TRIGA Reactor

Basic approach:

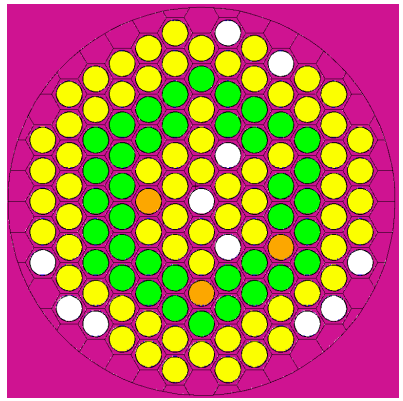


1. Set up & run unit fuel cell

- Universe 1: Fuel rod, grid plates, water exterior
- Create hexagonal unit cell with reflecting boundaries
- Fill hexagonal cell with Universe 1

2. Set up & run core region

- Create Universe 2 - LEU version of Universe 1
- Create Universe 3 - like U=1, but void filled, no fuel (instrument tube,…)
- Create Universe 4 - empty position, water filled - no rod
- Create Universe 5 - control rod, movable, use 5 different transformations
- Create hexagonal unit cell lattice
- Fill hexagonal lattice with Universes 1-5



3. Add core externals & run

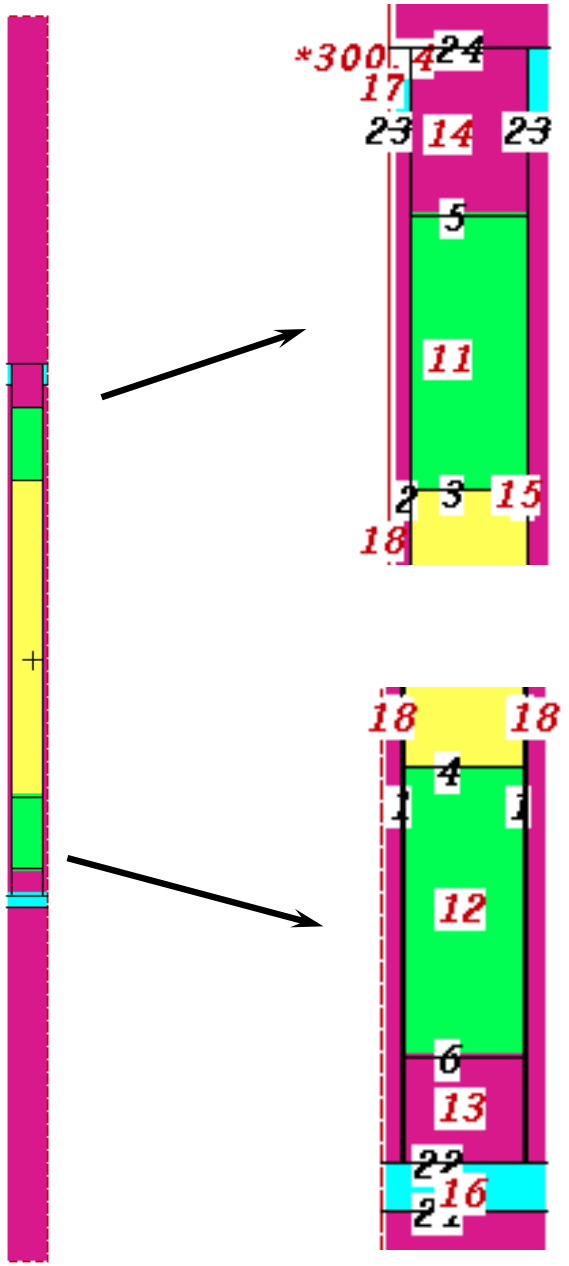
Example triga1 (1)

Problem triga1

Hexagonal Unit Cell, Filled with Fuel Rod

Example Problem - triga1 (2)

- **Fuel rod**
  - **Fuel region**
    - Radius = 1.815 cm
    - Height = 38.1 cm
  - **Top & bottom graphite plugs**
    - Radius = 1.815 cm
    - Height = 8.7 cm
  - **Clad - SS304**
    - Inner/outer radii = 1.815 cm, 1.865 cm
  - **Bottom plate - Aluminum**
    - Thickness = 1.5 cm
    - Position = 31 cm below core midplane
    - Holes for control rods
  - **Grid plate - Aluminum**
    - Thickness = 2.5 cm
    - Position = 30.5 cm above core midplane
    - Holes for fuel & control rods



Example Problem - triga1 (3)

- **Materials**

Light water (density = 1.00 g/cm^3)	
H	$6.7000 \cdot 10^{-2}$
O	$3.3500 \cdot 10^{-2}$

Stainles steel SS304 (density = 7.87 g/cm^3)	
Cr	$1.4161 \cdot 10^{-2}$
Mn	$8.3717 \cdot 10^{-4}$
Fe	$5.6089 \cdot 10^{-2}$
Ni	$1.0046 \cdot 10^{-2}$
Mo	$2.0929 \cdot 10^{-3}$

Aluminum 6061 (density = 2.70g/cm^3)	
Mg	$6.0887 \cdot 10^{-4}$
Al	$5.9305 \cdot 10^{-2}$
Si	$3.5127 \cdot 10^{-4}$
Ti	$1.7164 \cdot 10^{-5}$
Cr	$5.0597 \cdot 10^{-5}$
Mn	$1.4965 \cdot 10^{-5}$
Fe	$1.0010 \cdot 10^{-4}$
Cu	$8.7251 \cdot 10^{-5}$

Standard fuel (density = 6.11g/cm^3)	
H	$5.5318 \cdot 10^{-2}$
Zr	$3.5849 \cdot 10^{-2}$
U-235	$2.4800 \cdot 10^{-4}$
U-238	$9.8550 \cdot 10^{-4}$

LEU fuel (density = 6.51g/cm^3)	
H	$5.0440 \cdot 10^{-2}$
Zr	$3.4295 \cdot 10^{-2}$
~~Hf~~	~~$6.3186 \cdot 10^{-5}$~~
U-235	$6.3860 \cdot 10^{-4}$
U-238	$2.5700 \cdot 10^{-2}$

Er 3.835e-5
Original model substituted Hf (no Er xsecs)

Graphite (density = 1.28g/cm^3)	
C	$6.4240 \cdot 10^{-2}$

Boron carbide (density = 2.50g/cm^3)	
B-10	$2.1780 \cdot 10^{-2}$
B-11	$8.7131 \cdot 10^{-2}$
C	$2.7228 \cdot 10^{-2}$

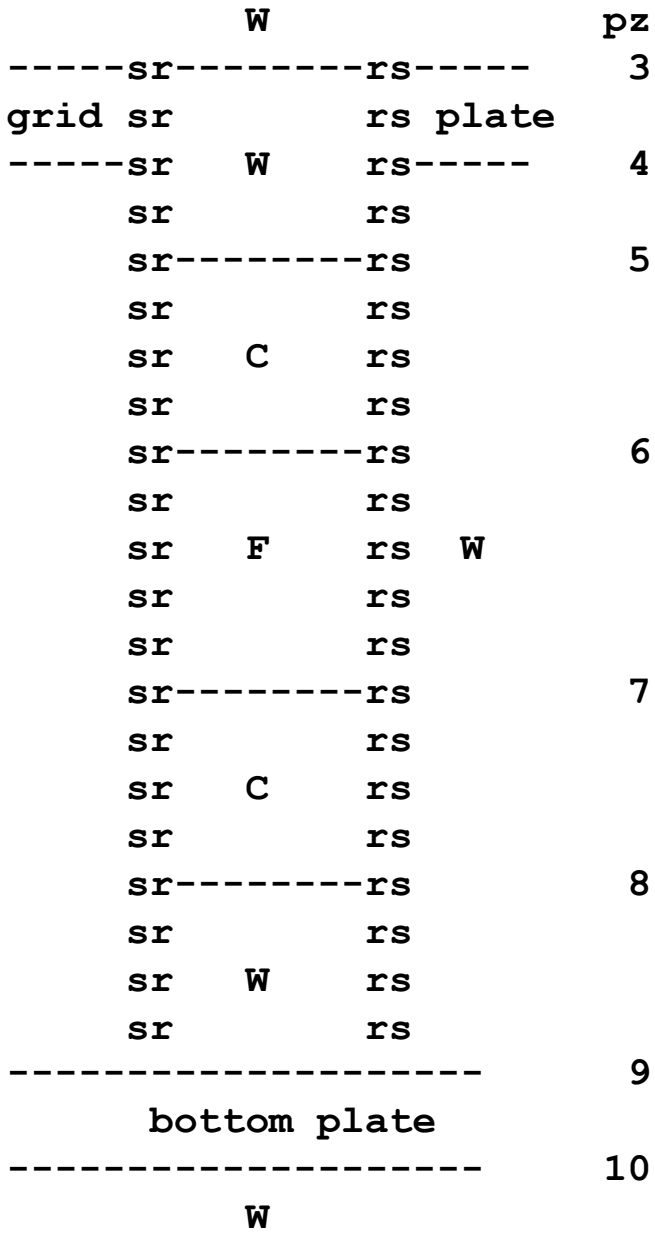
Example Problem - triga1 (4)

- **Materials**
  - **Want to use ENDF/B-VII.0 cross-section data**
  - **ENDF/B-VII.0 does not have elemental cross-sections**
  - **Convert Zr, Cr, Mn, Fe, Ni, Mo, Mg, Si, Ti, Cu, Er into individual isotopes using natural abundances [very tedious]**
  - **Fuel at 700K in report, use 600K for this problem (dataset temperature)**
  - **Other materials at room temperature, 293.6 K**
  - **Use S(α,β) data for:**
    - H in zirc hydride at 600K: h/zr.13t
    - H in water at 293.6K: lwtr.10t
    - C in graphite at 293.6K: grph.10t

- **File triga_materials.txt contains MCNP input for materials**
  - **M1000 - water**
  - **M2000 - SS304 cladding**
  - **M3000 - standard fuel**
  - **M4000 - LEU fuel**
  - **M5000 - graphite**
  - **M6000 - boron carbide**
  - **M7000 - aluminum**

Example Problem - triga1 (5)

- **Geometry**



W = water
C = graphite plug
F = fuel
r = clad tube, inner (sur=1)
s = clad tube, outer (sur=2)

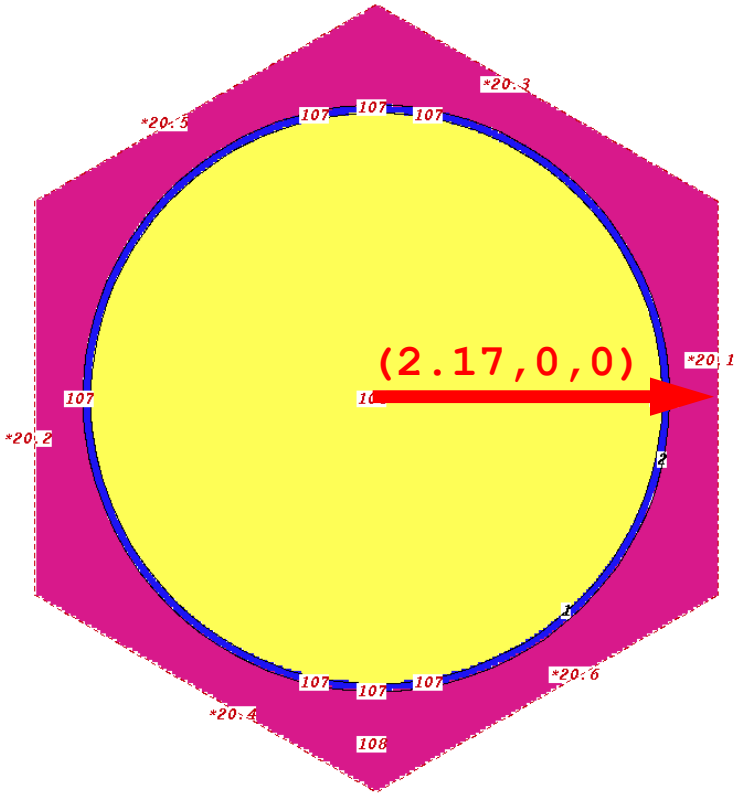
3	pz	33.00
4	pz	30.50
5	pz	27.75
6	pz	19.05
7	pz	-19.05
8	pz	-27.75
9	pz	-31.00
10	pz	-32.50

Example Problem - triga1 (6)

- **Geometry - hexagonal prism for unit cell**

```
20   RHP    bx by bz    h1 h2 h3    d1 d2 d3
```

bx,by,bz = center of base
h1,h2,h3 = axis vector, magnitude=height
d1,d2,d3 = vector from center of base to middle of first side




```
*20 RHP   0 0 -75    0 0 150    2.17 0
```

```
* = reflecting surface
```

Example Problem - triga1 (7)

```
triga1 - TRIGA unit cell, reflecting boundaries
c
c ------------------------------------------- CELL CARDS --------
 100 1000 -1.00    +3                u=1 imp:n=1.0   $ water above
 101 7000 -2.70    -3  +4  +2        u=1 imp:n=1.0   $ upper grid plate
c
c =====> fuel rod
 102 1000 -1.00    -1  -3  +5        u=1 imp:n=1.0   $ water above upper plug
 103 5000 -1.28    -1  -5  +6        u=1 imp:n=1.0   $ upper graphite plug
 104 3000 -6.109   -1  -6  +7        u=1 imp:n=1.0   $ fuel, std
 105 5000 -1.28    -1  -7  +8        u=1 imp:n=1.0   $ lower graphite plug
 106 1000 -1.00    -1  -8  +9        u=1 imp:n=1.0   $ water below lower plug
c
 107 2000 -7.87    +1  -2  -3  +9    u=1 imp:n=1.0   $ ss304 cladding
 108 1000 -1.00    +2  -4  +9        u=1 imp:n=1.0   $ water outside rod
c
 109 7000 -2.70    -9  +10           u=1 imp:n=1.0   $ lower grid plate
 110 1000 -1.00    -10               u=1 imp:n=1.0   $ water below
c
c =====> hex unit cell
 199    0          -20     fill=1       imp:n=1.0
c ------------------------------------------------------------------
```

Example Problem - triga1 (8)

```
c ------------------------------------------------ SURFACE CARDS ------
 1    cz     1.815     $ clad inner radius, fuel outer radius
 2    cz     1.865     $ clad outer radius
 c
 3    pz    33.00      $ upper grid plate - top
 4    pz    30.50      $ upper grid plate - bottom
 5    pz    27.75      $ upper graphite plug - top
 6    pz    19.05      $ top of fuel
 7    pz   -19.05      $ bottom of fuel
 8    pz   -27.75      $ bottom of graphite plug
 9    pz   -31.00      $ lower grid plate - top
10    pz   -32.50      $ lower grid plate - bottom
c
c hexagonal cell,  pitch = 4.34, reflecting BC
*20   rhp    0. 0. -75.     0. 0. 150.     2.17   0   0
c ------------------------------------------------------------------
```

Example Problem - triga1 (9)

```
C Material cards from
   triga_materials.txt
C
m1000  $=====> light water
         1001.70c  0.66667
         8016.70c  0.33333
mt1000   lwtr.10t
c
m2000  $=====> cladding (SS304)
        25055.70c  8.3717e-4
        26054.70c  3.27840e-03
        26056.70c  5.14639e-02
        26057.70c  1.18853e-03
        26058.70c  1.58171e-04
        24050.70c  6.15295e-04
        24052.70c  1.18654e-02
        24053.70c  1.34544e-03
        24054.70c  3.34908e-04
        28058.70c  6.83901e-03
        28060.70c  2.63437e-03
        28061.70c  1.14514e-04
        28062.70c  3.65122e-04
        28064.70c  9.29858e-05
        42092.70c  3.09121e-04
        42094.70c  1.93175e-04
        42095.70c  3.32771e-04
        42096.70c  3.49096e-04
        42097.70c  2.00081e-04
        42098.70c  5.06273e-04
        42100.70c  2.02383e-04
```

```
m3000  $=====> standard fuel
         1001.71c  5.5318e-02
        40090.71c  1.84443e-02
        40091.71c  4.02226e-03
        40092.71c  6.14810e-03
        40094.71c  6.23056e-03
        40096.71c  1.00377e-03
        92235.71c  2.4800e-04
        92238.71c  9.8550e-04
mt3000   h/zr.13t
c
m4000  $=====> leu fuel
         1001.71c  5.0440e-2
        40090.71c  1.76448e-02
        40091.71c  3.84790e-03
        40092.71c  5.88159e-03
        40094.71c  5.96047e-03
        40096.71c  9.60260e-04
        68162.71c  5.3306e-08
        68164.71c  6.1398e-07
        68166.71c  1.2848e-05
        68167.71c  8.7702e-06
        68168.71c  1.0346e-05
        68170.71c  5.7180e-06
        92235.71c  6.3860e-4
        92238.71c  2.5700e-3
mt4000   h/zr.13t
```

Example Problem - triga1 (10)

```
c
m5000  $=====> graphite
         6000.70c 6.4240-2
mt5000   grph.10t
c
```

```
m7000  $=====> aluminum 6061
         12024.70c  4.80944e-04
         12025.70c  6.08867e-05
         12026.70c  6.70363e-05
         13027.70c  5.93050e-2
         14028.70c  3.23954e-04
         14029.70c  1.64571e-05
         14030.70c  1.08614e-05
         22046.70c  1.41601e-06
         22047.70c  1.27698e-06
         22048.70c  1.26531e-05
         22049.70c  9.28556e-07
         22050.70c  8.89080e-07
         24050.70c  2.19844e-06
         24052.70c  4.23948e-05
         24053.70c  4.80723e-06
         24054.70c  1.19662e-06
         25055.70c  1.49649e-5
         26054.70c  5.85114e-06
         26056.70c  9.18503e-05
         26057.70c  2.12122e-06
         26058.70c  2.82296e-07
         29063.70c  6.03338e-05
         29065.70c  2.69168e-05
```

Example Problem - triga1 (11)

```
c
c ---------------------------------------------- OTHER CARDS ------
c =====> controls
kcode  5000  1.0  25  125
hsrc  3 -1.815 1.815  3 -1.815 1.815   11 -19.05 19.05
c
c =====> cell temperatures:  fuel= 600K,  other= 293.6K
tmp   2.5301e-8  3r  5.1704e-8   2.5301e-8 5r
c
c =====> source guess
sdef  pos 0 0 0   axs=0 0 1   rad=d1   ext=d2   erg=d3
si1   0 1.815
si2  -19.05 19.05
sp2   0 1
sp3  -3
c
c =====> mesh tally for detailed fission rate
c fmesh4:n  geom=xyz origin= -1.815 -1.815 -19.05
c           imesh= 1.815  iints=25
c           jmesh= 1.815  jints=25
c           kmesh= 19.05  kints=1
c fm4  -1. 0  -6
c ----------------------------------------------------------------
```

Result:

keff = 1.26116 ± 0.00079

Example triga2 (1)

Problem triga2

Hexagonal Lattice,
Filled with Fuel, Control Rods, …

Example Problem - triga2 (2)

```
trr-1/m1 -MCNP INPUT DECK FOR THE THAI RESEARCH REACTOR
c
c  +-----------------------------------------------+
c  | base input file from:  ORNL-TM-13656, June 1998 |
c  +-----------------------------------------------+
c
c Model of the Triga type OAEP Thai-Research-Reactor
c
c created March 31, 1998 by Jabo Tang & Franz Gallmeier ORNL, USA
c for the Office of Atomic Energy for Peace, Thailand
c
c This model includes:
c   -standard fuel rods (fresh fuel)
c   -LEU fuel rods (fresh fuel)
c   -vacant rods
c   -five control rods positioned independently by TR-cards
c    (fresh fuel & fresh boron carbide)
c   -the bottom and top grid plates c the safety plate
c   -substantial amount of light water about the core
c   -set up for keff calculations
```

Example Problem - triga2 (3)

```
C ------------------------------------------------ CELL CARDS --------
 100 1000 -1.00    +3                          u=1 imp:n=1.0   $ water above
 101 7000 -2.70    -3  +4  +2                  u=1 imp:n=1.0   $ grid plate
c =====> fuel rod
 102 1000 -1.00    -1  -3  +5                  u=1 imp:n=1.0   $ water above upper plug
 103 5000 -1.28    -1  -5  +6                  u=1 imp:n=1.0   $ upper graphite plug
 104 3000 -6.109   -1  -6  +7                  u=1 imp:n=1.0   $ fuel, std
 105 5000 -1.28    -1  -7  +8                  u=1 imp:n=1.0   $ lower graphite plug
 106 1000 -1.00    -1  -8  +9                  u=1 imp:n=1.0   $ water below lower plug
 107 2000 -7.87    +1  -2  -3  +9              u=1 imp:n=1.0   $ ss304 cladding
 108 1000 -1.00    +2  -4  +9                  u=1 imp:n=1.0   $ water outside rod
 109 7000 -2.70    -9  +10                     u=1 imp:n=1.0   $ lower grid plate
 110 1000 -1.00    -10                         u=1 imp:n=1.0   $ water below
c =====> leu fuel rod
 200  like 100  but                           u=2 imp:n=1.0
 201  like 101  but                           u=2 imp:n=1.0
 202  like 102  but                           u=2 imp:n=1.0
 203  like 103  but                           u=2 imp:n=1.0
 204  like 104  but mat=4000 rho=-6.508       u=2 imp:n=1.0
 205  like 105  but                           u=2 imp:n=1.0
 206  like 106  but                           u=2 imp:n=1.0
 207  like 107  but                           u=2 imp:n=1.0
 208  like 108  but                           u=2 imp:n=1.0
 209  like 109  but                           u=2 imp:n=1.0
 210  like 110  but                           u=2 imp:n=1.0
```

Example Problem - triga2 (4)

```
c =====> vacant rods (ct, rabbit, nd, iit)
 300  like 100  but                          u=3 imp:n=1.0
 301  like 101  but                          u=3 imp:n=1.0
 302  like 102  but                          u=3 imp:n=1.0
 303  like 103  but  mat=0                   u=3 imp:n=1.0
 304  like 104  but  mat=0                   u=3 imp:n=1.0
 305  like 105  but  mat=0                   u=3 imp:n=1.0
 306  like 106  but                          u=3 imp:n=1.0
 307  like 107  but                          u=3 imp:n=1.0
 308  like 108  but                          u=3 imp:n=1.0
 309  like 109  but                          u=3 imp:n=1.0
 310  like 110  but                          u=3 imp:n=1.0
c
c =====> empty position
 400  like 100  but                          u=4 imp:n=1.0   $ water above
 401 7000 -2.70    -3  +4                    u=4 imp:n=1.0   $ upper grid plate
 402 1000 -1.00    -4  +9                    u=4 imp:n=1.0   $ water below grid plate
 409  like 109  but                          u=4 imp:n=1.0   $ lower grid plate
 410  like 110  but                          u=4 imp:n=1.0   $ water below
```

Example Problem - triga2 (5)

```
c =====> control rod (movable by tr1,2,3,4,5)
 500 1000 -1.00    +11                      u=50 imp:n=1.0   $ water above
 501 0             -1 -11  +6               u=50 imp:n=1.0   $ void
 502 6000 -2.50    -1 -6   +7               u=50 imp:n=1.0   $ B4C
 503 3000 -6.109   -1 -7   +12              u=50 imp:n=1.0   $ fuel, std
 504 1000 -1.00    -1 -12  +13              u=50 imp:n=1.0   $ water below
 505 1000 -1.00    -13                      u=50 imp:n=1.0
 506 2000 -7.87    +1 -11 +13               u=50 imp:n=1.0   $ ss304 cladding
c =====> rod 1, movable by tr1
 510 1000 -1.00    +3      +2               u=5  imp:n=1.0   $ water above
 511 7000 -2.70    -3  +4  +2               u=5  imp:n=1.0   $ grid plate
 512 1000 -1.00    +2  -4  +9               u=5  imp:n=1.0   $ water outside
  rod
 513 7000 -2.70    -9  +10    +2            u=5  imp:n=1.0   $ lower grid plate
 514 1000 -1.00    -10        +2            u=5  imp:n=1.0   $ water below
 515 0             -2    fill=50(1)         u=5  imp:n=1.0   $ rod inserted,
  tr1
c =====> rod 2, movable by tr2
 520 like 510  but                          u=6  imp:n=1.0
 521 like 511  but                          u=6  imp:n=1.0
 522 like 512  but                          u=6  imp:n=1.0
 523 like 513  but                          u=6  imp:n=1.0
 524 like 514  but                          u=6  imp:n=1.0
 525 like 515  but       fill=50(2)         u=6  imp:n=1.0
```

Example Problem - triga2 (6)

```
c =====> rod 3, movable by tr3
 530 like 510  but                          u=7  imp:n=1.0
 531 like 511  but                          u=7  imp:n=1.0
 532 like 512  but                          u=7  imp:n=1.0
 533 like 513  but                          u=7  imp:n=1.0
 534 like 514  but                          u=7  imp:n=1.0
 535 like 515  but       fill=50(3)         u=7  imp:n=1.0
c =====> rod 4, movable by tr4
 540 like 510  but                          u=8  imp:n=1.0
 541 like 511  but                          u=8  imp:n=1.0
 542 like 512  but                          u=8  imp:n=1.0
 543 like 513  but                          u=8  imp:n=1.0
 544 like 514  but                          u=8  imp:n=1.0
 545 like 515  but       fill=50(4)         u=8  imp:n=1.0
c =====> rod 5, movable by tr5
 550 like 510  but                          u=9  imp:n=1.0
 551 like 511  but                          u=9  imp:n=1.0
 552 like 512  but                          u=9  imp:n=1.0
 553 like 513  but                          u=9  imp:n=1.0
 554 like 514  but                          u=9  imp:n=1.0
 555 like 515  but       fill=50(5)         u=9  imp:n=1.0
```

Example Problem - triga2 (7)

```
c core configuration:
600  0          -20                          u=99 imp:n=1.0
      lat=2  fill= -7:7 -7:7 0:0
      4 4 4 4 4 4 4 4 4 4 4 4 4 4 4                    $ bottom row, (:,-7,0)
       4 4 4 4 4 4 4 4 1 1 1 3 3 4 4
        4 4 4 4 4 4 1 1 1 1 1 1 3 4
         4 4 4 4 4 1 1 2 2 2 2 2 1 1 4
          4 4 4 4 1 1 2 6 2 2 7 2 1 1 4
           4 4 4 3 1 2 2 1 3 1 2 2 1 1 4
            4 4 3 1 2 2 1 1 1 1 2 2 1 1 4
             4 4 1 2 2 1 1 3 1 1 8 2 1 4 4          $ center,      (:,0,0)
              4 3 1 2 2 5 1 1 3 2 2 1 1 4 4
               4 1 1 2 2 1 1 1 2 2 1 1 4 4 4
                4 1 1 2 2 2 2 9 2 1 3 4 4 4 4
                 4 1 1 2 2 2 2 2 1 1 4 4 4 4 4
                  4 1 1 1 1 1 1 1 3 4 4 4 4 4 4
                   4 4 1 1 1 1 1 4 4 4 4 4 4 4 4
                    4 4 4 4 4 4 4 4 4 4 4 4 4 4 4   $ top row,     (:,+7,0)
c
610  0            -30          fill=99          imp:n=1.0 $ core region
620  1000 -1.00   -40  +30                     imp:n=1.0 $ water about the
  core
999  0            +40                         imp:n=0.0 $ system boundary
c ------------------------------------------------------------------------
```

Example Problem - triga2 (8)

```
c ------------------------------------------------ SURFACE CARDS ------
 1   cz    1.815     $ clad inner radius, fuel outer radius
 2   cz    1.865     $ clad outer radius
 c
 3   pz   33.00      $ upper grid plate - top
 4   pz   30.50      $ upper grid plate - bottom
 5   pz   27.75      $ upper graphite plug - top
 6   pz   19.05      $ top of fuel
 7   pz  -19.05      $ bottom of fuel
 8   pz  -27.75      $ bottom of graphite plug
 9   pz  -31.00      $ lower grid plate - top
10   pz  -32.50      $ lower grid plate - bottom
11   pz   36.0
12   pz  -57.15
13   pz  -67.0
c
20   rhp  0. 0. -75.   0. 0. 150.   2.17 0. 0.  $ hexagonal cell,  pitch = 4.34
30   rcc  0. 0. -69.   0. 0. 142.   27.5        $ core region
40   rcc  0. 0. -75.   0. 0. 150.   54.         $ water surrounding the core
c
c 30     rcc  0. 0. -70.   0. 0. 1.     27.5   $ safety plate
c general control rod surface cards
c 51     pz -120.0      $ lower extension of geometry
c 52     pz  100.0      $ upper extension of geometry
c ------------------------------------------------------------------
```

Example Problem - triga2 (9)

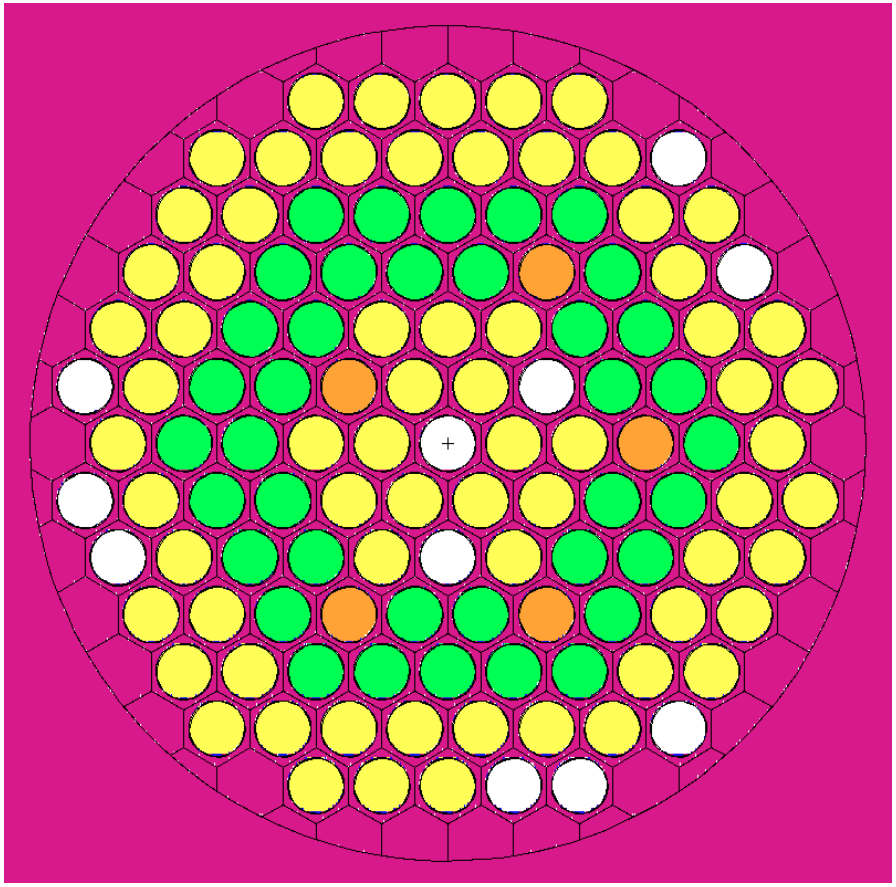
```
c ------------------------------------------ TRANSFORMATION CARDS ---
c
c for positioning the control rods
c (the positive third entry gives the CR position,
c where the value 0.0 describes the fully inserted CR position)
c
tr1    0    0    +1.0       $ CR1 -> universe u=5
tr2    0    0    +9.0       $ CR2 -> universe u=6
tr3    0    0   +18.0       $ CR3 -> universe u=7
tr4    0    0   +36.0       $ CR4 -> universe u=8
tr5    0    0   +27.0       $ CR5 -> universe u=9
c ------------------------------------------------------------------
c
c ------------------------------------------------ MATERIAL CARDS ----
m1000  $=====> light water (h2o) density= 1.00 g/cc, 293.6 K
```

. . . . . From file triga_materials.txt

Example Problem - triga2 (10)

```
c ---------------------------------------------- OTHER CARDS ------
c =====> controls
kcode  5000  1.0  25  125
hsrc  9 -27.5 27.5   9 -27.5 27.5    9 -19.05 19.05
c
c =====> temperatures:  fuel= 600K,  other= 293.6K
tmp   2.5301e-8  3r  5.1704e-8
      2.5301e-8  9r  5.1704e-8
      2.5301e-8 62r
c
c =====> source guess
sdef  pos 0 0 0   axs=0 0 1   rad=d1   ext=d2   erg=d3
si1   0 28.
si2  -19.05 19.05
sp2   0 1
sp3  -3
c
c =====> mesh tally for detailed fission rate
c fmesh4:n  geom=xyz origin= -28. -28. -69.
c           imesh= 28.  iints=400
c           jmesh= 28.  jints=400
c           kmesh= 73.  kints=1
c fm4  -1. 0  -6
c ----------------------------------------------
```

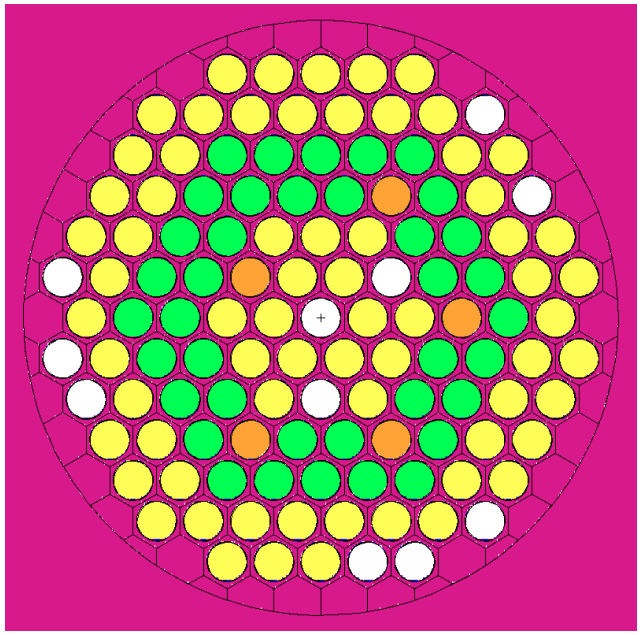
Example Problem - triga2 (11)



z=19

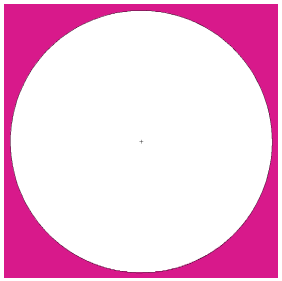
Fuel, standard
Fuel, LEU
Void tubes
Control rods

Example Problem - triga2 (12)

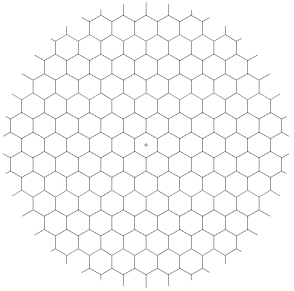


z=19, all levels

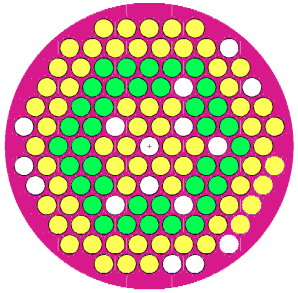
Fuel, standard
Fuel, LEU
Void tubes
Control rods



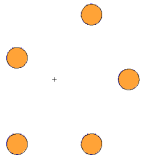
Level-1



Level-2

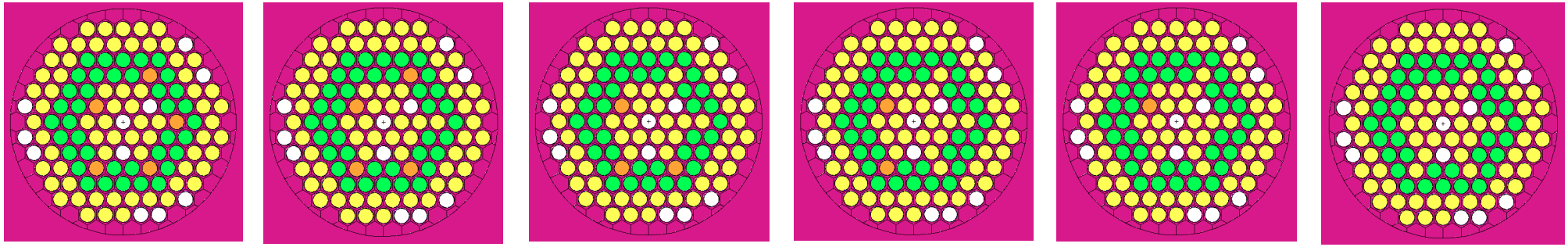


Level-3



Level-4

Example Problem - triga2 (13)



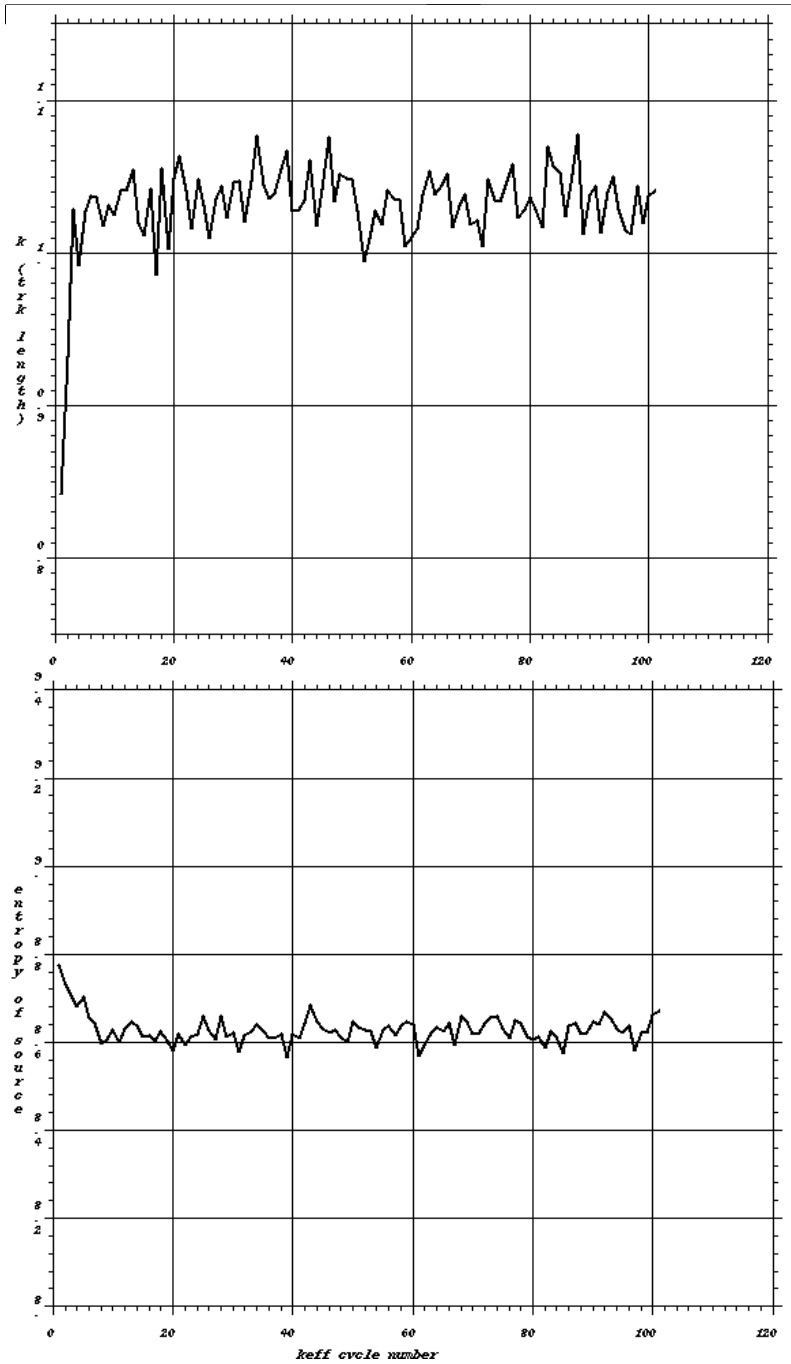
Z = 19 Z = 10 Z = 0 Z = -10 Z = -18 Z = -19

For this example:

CR-1	withdrawn	1 cm	bottom of B4C: z = -18.05 cm
CR-2	withdrawn	9 cm	bottom of B4C: z = -10.05 cm
CR-3	withdrawn	18 cm	bottom of B4C: z = -1.05 cm
CR-4	withdrawn	36 cm	bottom of B4C: z = 7.95 cm
CR-5	withdrawn	27 cm	bottom of B4C: z = 16.95 cm

Fuel is below B4C region in CR's

Example Problem - triga2 (14)



K_{path} vs cycle
converge at ~15

Result, for kcode 5000 1 25 125
keff = 1.03782 ± 0.00110

H_{src} vs cycle
converge at ~20

Conclusions

Next:

- **Add more 3D features - beam tubes, structure, shielding, …..**
- **Various runs to get control rod worths**
- **Try different loading patterns**
- **. . . . . .**

Case Study #5

Fissile Material Storage Vault
(Avoiding Mistakes in Loosely-Coupled Problems)

Objectives

- See how a simple mistake can lead to a very bad prediction of the system eigenvalue k.
- Develop an understanding of the numerical issues of applying Monte Carlo to loosely-coupled systems to show why this error occurred.
- Learn a workflow so that the chance of such incorrect results is minimized.

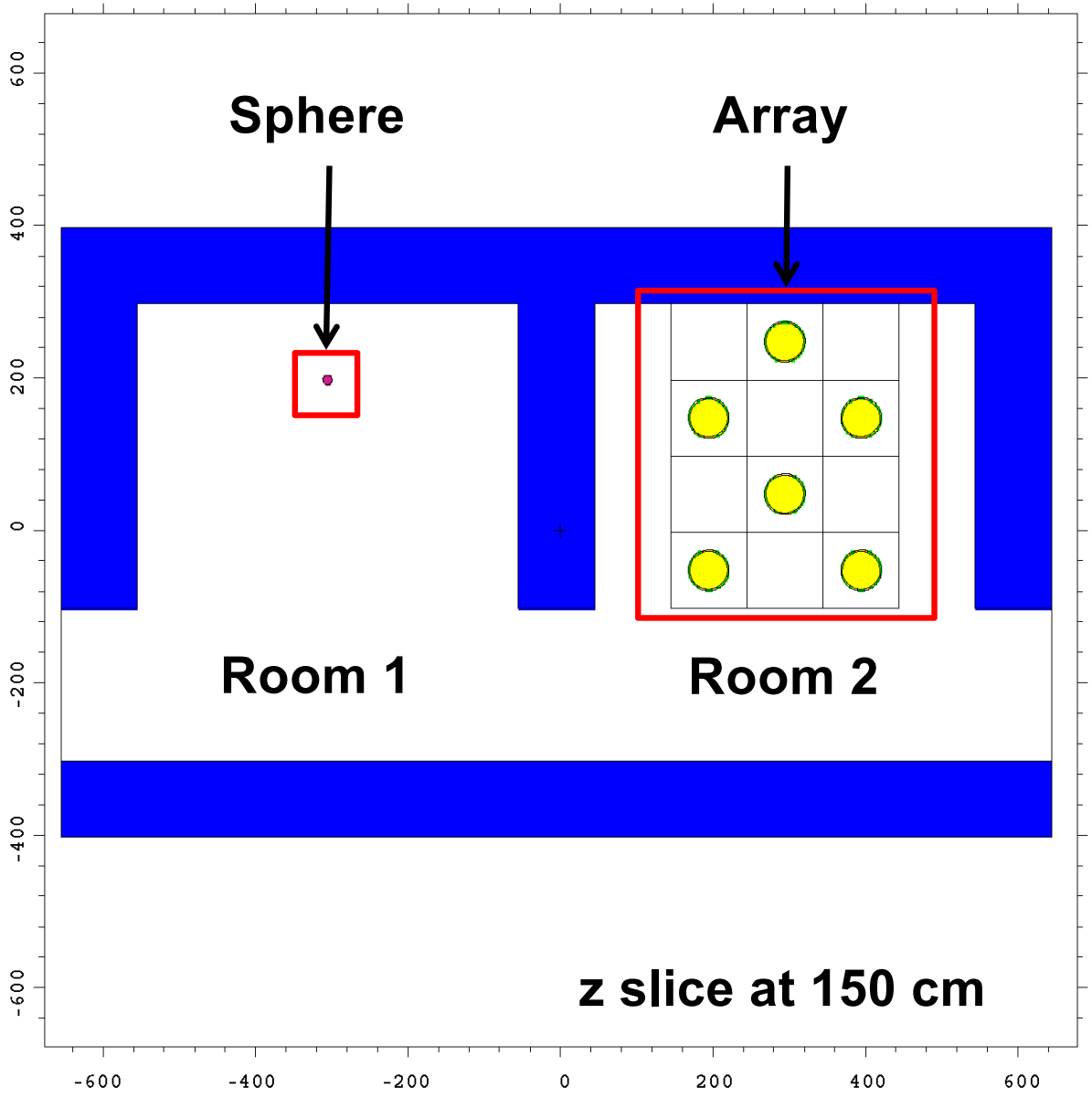
Concepts and Terminology

- **Coupling:** A measure of the relative amount of neutrons that transport from one zone to another.
- **Loosely Coupled:** A system that has two or more discrete, separate zones of fissile material such that a small fraction of neutrons transport between the regions.
- **Reactive:** The eigenvalue of an individual fissile zone where fission in all other zones is not considered.
- **Initiation:** The process whereby a stable neutron population is established in a zone.
- **Undersampling:** Failure to adequately sample all relevant regions of phase space.

Fissile Material Storage Vault

- **Consider the following arrangement:**
  - **A storage vault that consists of two rooms, each having fissile material**
  - **Room 1: Bare sphere of plutonium**
  - **Room 2: An array of cans of plutonium nitrate solution**
  - **Rooms are connected by a hallway, but otherwise separated by a meter of concrete**
- **First, however, calculate k of each region alone.**

System Diagram



Case 1: Plutonium Sphere Alone

- Copy vault01.txt from the solutions directory:
- This is the bare plutonium sphere in Room 1 alone.
- Plot the geometry to verify.
  - Plot the 'xy' view with a z slice through 150 cm ('pz 150')
  - Cell 100 is the plutonium sphere
- Set the random seed to 2*N – 1, where your N is assigned in class.
- Run the problem and report your result.

Case 2: Plutonium Nitrate Can Array Alone

- Copy vault02.txt from the solutions directory:
- This is the array of cans of plutonium nitrate solution in Room 2 alone.
- Plot the geometry to verify.
  - Plot the 'xy' view with a z slice through 150 cm ('pz 150')
  - Cell 200 is the plutonium nitrate solution
- Use the same random number seed, run the problem and report your result.

Results

- Case 1 should yield a result that is approximately critical within statistical confidence.
- Case 2 should yield a result that is subcritical.
- What can we say about the expected k of the combined system?

Fissile Material Storage Vault Properties

- **For physical reasons, the combined system must be supercritical.**
  - **Room 1 is more reactive than room 2.**

- **Relatively few neutrons will travel from one room to another.**
  - **The combined system is loosely coupled.**

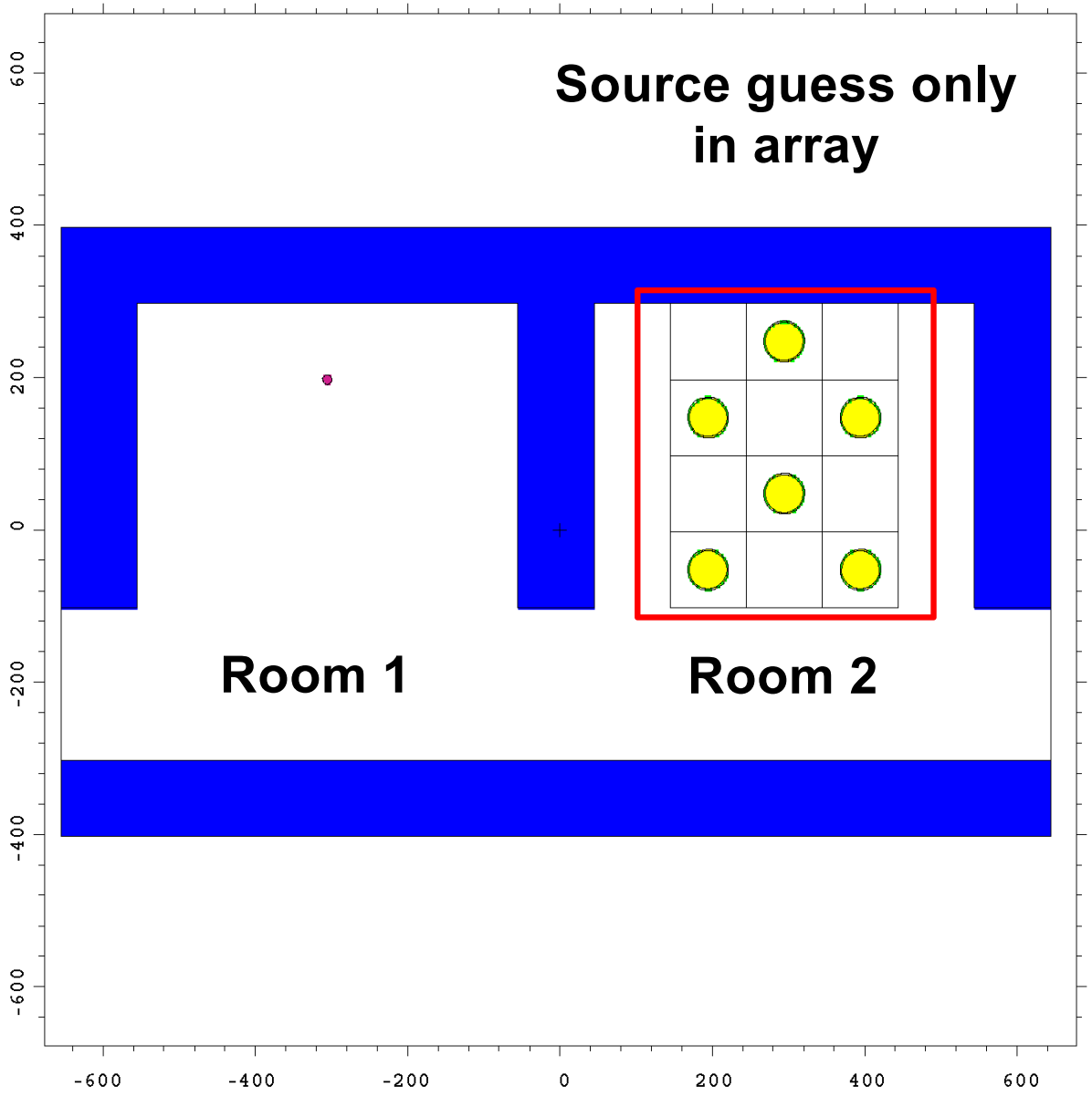
Full Vault Case

- Copy vault03.txt from the solutions directory:
- This is the combined system. Plot to verify.
- The number of cycles is larger. Take note of the source guess and the batch size.
- Using the same random number seed, run the problem and report your result.

Observation

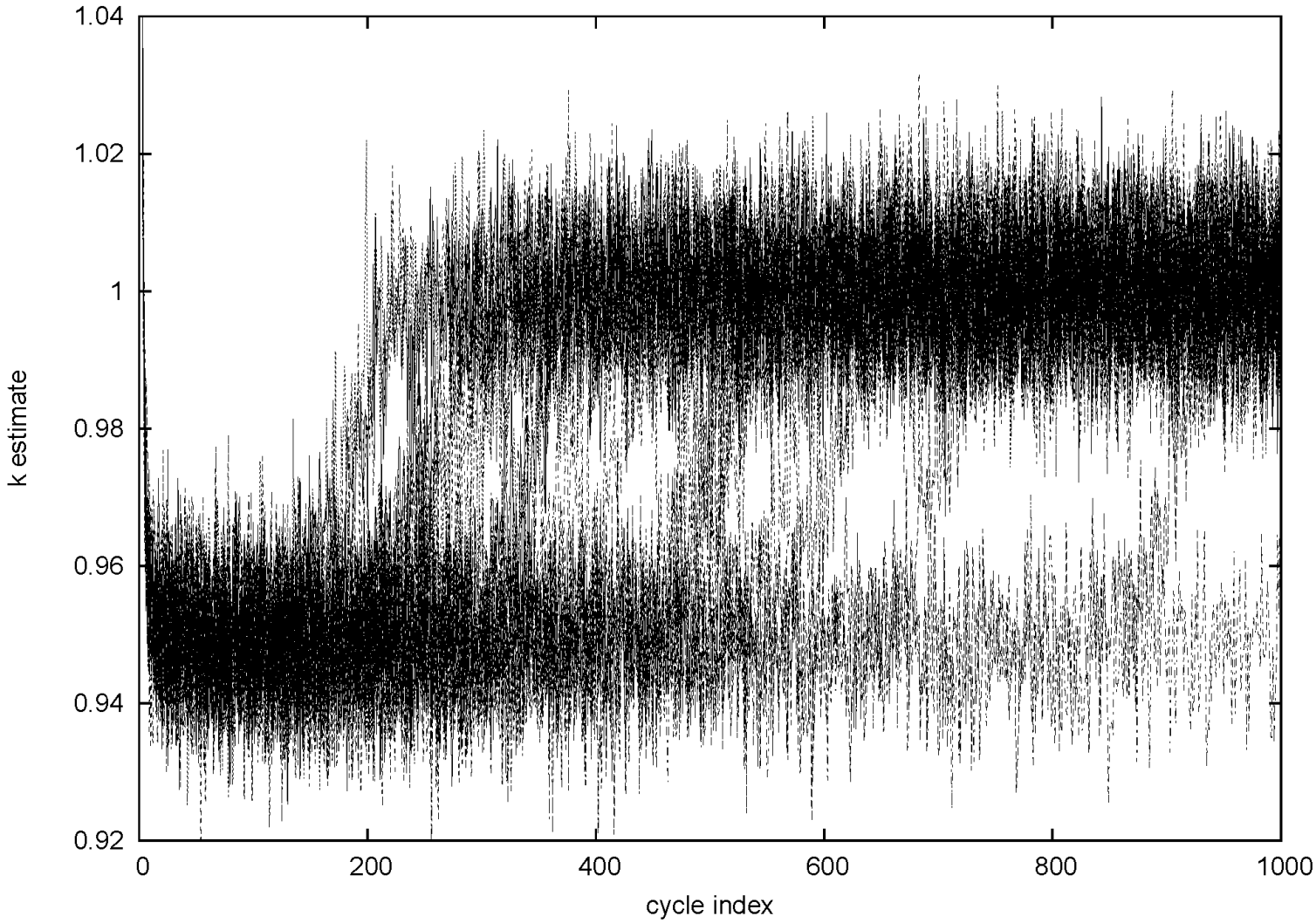
- Almost all independent calculations should yield an incorrect subcritical value of k.
- A few may yield the result near critical.
- We may have not detected any problem had we not run the two cases independently.
- Why the wide disparity?

Omission in the Source Guess



Cycle Eigenvalue for Random Trials

Notice the randomness in the convergence of k:



Investigation of Population Tables

- **Examine the population table in the output file.**

- **Good result:**

	cell	tracks entering	population	collisions	collisions * weight (per history)	number weighted energy	flux weighted energy	average track weight (relative)	average track mfp (cm)
1	100	98379	98362	203437	1.4586E+00	1.0857E+00	1.7989E+00	7.3169E-01	3.2274E+00
2	200	3025	2474	198160	1.3294E+00	1.4336E-04	4.9348E-01	7.6650E-01	1.3170E+00

- **Poor result (cell 100, the Pu sphere is undersampled):**

	cell	tracks entering	population	collisions	collisions * weight (per history)	number weighted energy	flux weighted energy	average track weight (relative)	average track mfp (cm)
1	100	8	8	4	1.9473E-05	1.2420E-03	1.3548E+00	7.4106E-01	2.6703E+00
2	200	118828	99866	9030639	6.3167E+01	1.6295E-04	5.5321E-01	7.9872E-01	1.4080E+00

The Issue

- **The sampling of the Pu sphere is poor in cases where k is too low.**

- **Source guess does not include the Pu sphere.**
  - **For k to be correct, initiation must occur in the Pu sphere.**
  - **This may only happen if a neutron, by chance alone, reaches the sphere and establishes a sustained fission chain.**

- **Ways to increase the chance of initiation:**
  - **Increase the batch size or number of inactive cycles.**
    - **Chance of reaching Pu sphere (coupling) is very small, works, but inefficient.**
    - **Larger batch size is good advice anyway for production runs.**
  - **Use a source guess that includes the Pu sphere.**
    - **By adding a point in the sphere, on average 1/7 of the starter neutrons will be sampled in the sphere. For a batch size of 1000, there will be about 140 chances for initiation, which should be enough.**

Full Vault Case with a Better Source Guess

- Copy vault03.txt to vault04.txt:
- Add the point at the center of the Pu sphere (350, 500, 150) to the ksrc card.
- Using your same random number seed as before, run the problem and report your result.

Discussion of Results

- **The new source guess should yield correct results consistently.**
  - **A suitable guess ensures that initiation will occur in the most reactive region assuming the batch size is large enough.**
    - 1000 is generally low for a batch size.
    - **Used here in the interest of time, for real work use 10000+.**

- **Analyze the population tables to check this is indeed the case.**

- **Develop a workflow to protect against undersampling.**
  - **Even if you know what you are doing, typos can happen!**

Establishing a Lower Bound on k

- In the transport equation, 1/k is a factor to balance neutron gains from fission and losses from other means.

- For physical reasons, the most reactive region alone sets the lower bound for k.
  - The system surrounded by vacuum will have some k.
  - Additional multiplication in other regions or from neutrons from other regions will increase the amount of fission and increase k.

- Identifying each region and running a criticality calculation of it alone is useful to determine if answers are reasonable.
  - If the calculated k of the composite system is less than that of the highest k of each individual component, you know something is wrong.

Deterministic and Stochastic Convergence

- **The neutron transport equation implicitly assumes average behavior is descriptive of the system.**
  - **In the limit of infinite particles per cycle, the convergence follows the deterministic neutron transport equation.**
  - **When the sampling is poor, the convergence is less predictable and is best described by more general equations of branching processes.**
  - **We require large enough batch sizes and adequate coverage of the problem so that the equations are approximately the deterministic ones with some small amount of noise.**
  - **When this is true, k each iteration can be predicted by some mean value for that iteration plus or minus some stochastic variation.**

- **Adequately sampling the most reactive region is an absolutely necessary (but not necessarily sufficient) condition for predictive convergence.**

Workflow for Loosely-Coupled Problems

1) Identify each discrete region of the problem.
2) Calculate k of each region where all the others are absent.
3) Find the maximum of these values; this is your lower bound on k.
4) Construct the composite system.
   - Ensure that each region has at least one source point.
   - Ensure that the batch size is large enough so that there will almost certainly be a sufficient amount of sampling in each region.
   - Double check your geometry, materials, and source guess.
5) Run the problem.
6) Analyze the results of k, heed any warnings, and examine the population tables.
   - The composite system k should be at least the lower bound.
   - The most reactive region should usually have most of the particles.
7) If anything appears out of the ordinary, recheck your input and run a few independent cases with differing random number seeds.

Criticality Accident Alarm System Calculations

Introduction & Objectives

- **Criticality Accident Alarm System (CAAS) calculations combine skills for needed for both criticality and shielding calculations.**

- **Objectives:**
  - **Generate a source file from a KCODE calculation**
  - **Define tallies to calculate energy deposition to a detector object**
  - **Employ variance reduction techniques to obtain statistically significant results**
  - **Understand requirements for statistical significance**

Exercise 1: Criticality Review

- **Copy caas1.txt from the SOLUTIONS directory to the working directory.**
  - **Inspect the input file and plot the geometry:**

```
mcnp6  i = caas1.txt  ip
```

  - **Run the problem:**

```
mcnp6  i = caas1.txt  o = caas1o.txt  srctp = ksource
```

  - **Make plots of keff and Shannon entropy versus cycle.**
  - **Save the source tape (srctp) file and name this ksource**

Surface Source Files

Surface Source Write

- Generated with the SSW or "Surface Source Write" card.
- Form: SSW CEL = C1 C2 …
  - The arguments after the CEL keyword is a list of cells to store fission source points for in a KCODE calculation.
  - Produces a file with default name wssa.
  - Additional options are available. See the MCNP Manual.

Surface Sources

- MCNP can generate a binary file called a "surface source" file.
  - Normally, this contains particle tracks that crossed a surface, which are run in a different calculation
- The "surface source" file may also contain fission source points from a KCODE calculation.
- The surface source file may be used as the source from a criticality accident.
- Note: Surface sources do not yet work with OMP threading.

Surface Source Considerations

- **Some considerations when deciding how many fission points to bank:**
  - **Define NSS as the number of particles banked.**
  - **A sufficient number of fission source points is needed describe a continuous neutron field – more is usually better.**
  - **When reading the file, MCNP allows a variable number of particles NPS to be used.**
  - **If NPS < NSS, then NPS particles are randomly selected with an increased starting weight per history.**
  - **If NPS > NSS, then NSS starting particles are started, but some will be duplicated randomly with lower starting weight per history. Note that the NPS used for normalizing tallies is the same.**

Exercise 2: Generating a Surface Source

- Copy caas1.txt to caas2.txt.
  - Delete the KSRC card. Use the source tape ksource as starting guess.
  - Modify KCODE card to use 20,000 neutrons per cycle, skip one cycle, and run 50 active cycles
  - Add SSW card for cell 100.
  - Run the problem, name the wssa file source.

```
mcnp6  i = caas2.txt  o = caas2o.txt  srctp = ksource
       wssa = source
```

  - Note: This will generate 1 million source points and will take a long time.

Surface Source Read

- **May read the surface source in a fixed source calculation.**
- **Form: SSR CEL = C1 C2 … WGT = W PSC = P**
  - **The arguments after the CEL keyword is a list of cells to use from the surface source file.**
  - **W is the intensity of the source (neutrons released from the burst).**
  - **PSC is the probability of scattering cosine. From fission this is isotropic and 0.5. This is needed for F5 tallies and DXTRAN.**
  - **Reads a file with default name rssa.**
  - **Additional options are available. See the MCNP Manual.**

Neutron Fission Treatment

- **Since fission was already treated in the KCODE calculation, it must be treated as capture.**

- **Form: NONU N1 N2 … N(NCEL)**

  – **Specifies a list of cells where fission is treated as capture**
    = 0, do not perform fission (treat as capture)
    = 1, perform fission
  – **Must list "0" for each cell in the problem. May also do this on the cell card.**
  – **If this is not done, the problem will run forever because it is supercritical. Even if it were subcritical, the answer would be wrong.**

Energy Deposition (F6) Tally

- **The neutron energy deposited in a cell may be obtained with an F6 tally.**

- **Form:** **F6:n C1 C2 …**

  – **Computes energy deposition for each cell listed on the card in MeV/gram.**
  – **Otherwise, very much like an F4 tally, i.e., may use FM cards, etc.**

Exercise 3: Reading the Surface Source

- Copy caas2.txt to caas3.txt.
  - Delete the SSW card and insert a SSR card.
  - The intensity of the burst is 1e15 fissions times 2.9 neutrons per fission.
  - Delete KCODE and insert an NPS card with 1e4 neutrons.
  - Insert a NONU card and turn off fission in all cells.
  - Create an energy deposition tally for cell 300. Convert the units of the tally to Gy or J/kg (1.602e-10 is the conversion factor from MeV/gram to J/kg).
  - Run the problem reading the file source.

```
mcnp6  i = caas3.txt  o = caas3o.txt  rssa = source
```

  - Analyze the tally output. Note that no neutrons scored.

Variance Reduction Concepts

Variance Reduction

- **Monte Carlo calculations can be very inefficient**
- **Variance reduction techniques increase the efficiency while preserving the average answer**
  - Mean is preserved
  - Variance (statistical uncertainty) is hopefully reduced
- **Techniques are a balancing act**
  - Some decrease the expected variance and increase the time per history
  - Some increase the expected variance and decrease the time per history
  - Coming out ahead is the art of variance reduction
- **Many increase the efficiency of calculating one quantity at the expense of another**

Monte Carlo Efficiency

- **Relative uncertainty scales as**

$$R = \frac{Const}{\sqrt{N}}$$

- **Variance reduction techniques do not change the rate of convergence, but change the constant**

- **Efficiency of a calculation defined by Figure of Merit:**

$$FOM = \frac{1}{R^2 T}$$

- **Constant for large N**
  - **Higher FOM means higher efficiency**

Variance Reduction Techniques

- **MCNP offers several variance reduction techniques:**
  - Implicit capture
  - Splitting
  - Russian Roulette
  - Weight Cutoff (Roulette)
  - DXTRAN Spheres
  - Energy Splitting
  - Forced Collisions
  - Weight Windows
  - Source Biasing
  - Exponential Transform

- **We will discuss many of these at a basic level**
  - Take our advanced variance reduction class for a more complete overview

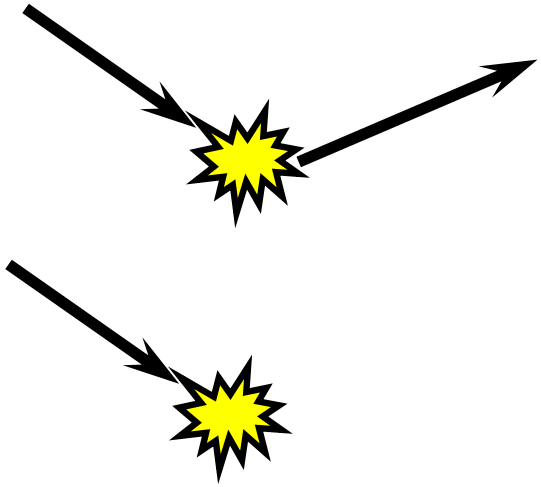
Particle Weight

- All particles carry a statistical weight as part of its state
- Tallies are multiplied by this statistical weight
- Variance reduction techniques adjust the statistical weight in ways that preserve all tally scores on average

Implicit Capture

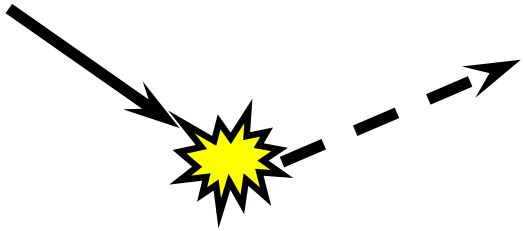
- **Implicit capture separates the particle into a captured and scattered component**
  - **By modifying weight, both capture and scattering are simulated simultaneously**

Real World



Scatter or Capture

Implicit Capture



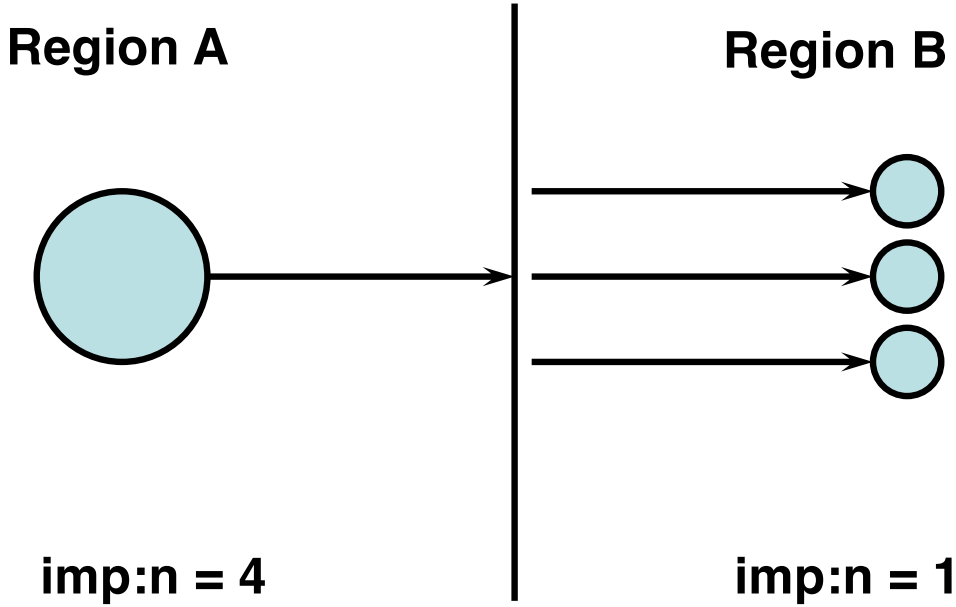
$$w_{\text{out}} = w_{\text{in}} \frac{\sigma_t - \sigma_c}{\sigma_t}$$

"Averages" Both Events

- **Implicit capture is on by default!**

Importance Splitting

- When a particle enters a region of higher importance, it is copied with each copy having a fraction of the original statistical weight



$$w_B = w_A \frac{4}{12} = \frac{w_A}{3}$$

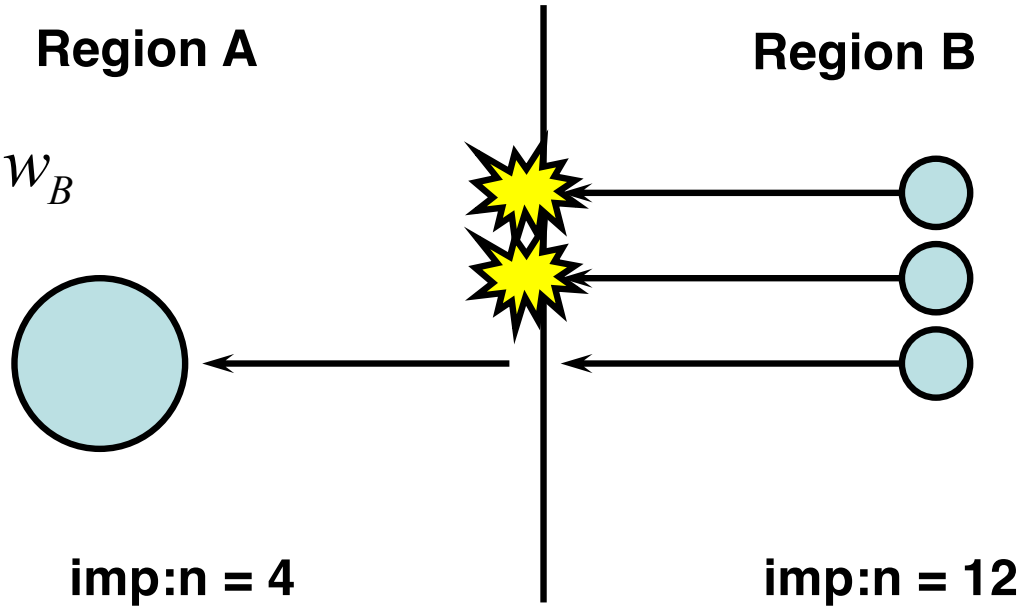
- Decreases the variance per history, but increases the time

- Fractional importances cause stochastic splitting
  - If region B has imp:n = 10, 2 or 3 neutrons produced half of the time each

Russian Roulette

- **When a particle enters a region of lower importance, it may be terminated a probability of the ratio of importances. If it survives, its weight is increased**

$$w_A = w_B \frac{12}{4} = 3w_B$$



- **Increases the variance per history, but decreases the time**
  - **Useful to terminate tracks that only contribute seldom or little**

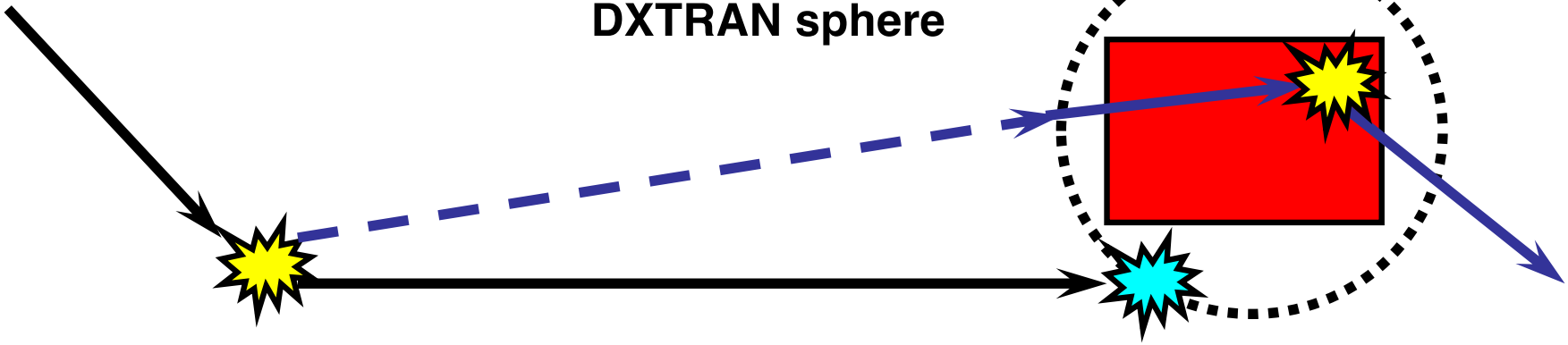
Weight Cutoff

- **Misnamed: should be Weight Roulette**
- **If weight is less than the weight cutoff:**
  - **Terminated with some probability**
  - **Surviving particles have their weight increased to preserve total weight on average**
- **Like Russian Roulette, culls low weight particles that contribute little to tallies**

DXTRAN Spheres

Increasing Efficiency: DXTRAN

- **Unfortunately, the most useful VR technique for this problem is the most difficult to understand.**
- **Scattering to a specific region may be improved with angular biasing created with DXTRAN spheres.**



Energy Deposition (F6) Tally

- **DXTRAN spheres cover a region of space. Particles are "pulled" to that region each collision.**

- **Form:** **DXT:n X0 Y0 Z0 Rin Rout WC1 WC2**

  - **First three entries are the coordinates for the center of the DXTRAN sphere.**
  - **Rin and Rout are the inner and outer radii of the sphere to refine angular biasing. Usually Rin = Rout.**
  - **WC1 and WC2 are weight cutoff parameters to roulette low weight particles. Important especially if multiple DXTRAN spheres are present.**

Exercise 4: DXTRAN Spheres

- **Copy caas3.txt to caas4.txt.**
  - **Place a DXTRAN sphere covering the detector cell. Use identically inner and outer radii. WC1 = 5e-6 and WC2 = 1e-6 to cull low-weight particles.**
  - **Run the problem:**

```
mcnp6  i = caas4.txt  o = caas4o.txt  rssa = source
```

  - **Analyze the tally results.**
  - **There should now be scores, but high variance:**

```
1tally        6        nps =       10131
 cell  300
                1.77551E-06 0.5064
```

Energy Splitting

Increasing Efficiency: Energy Splitting

- Neutrons with low energies can only deposit very little energy to the detector.
- Much time is wasted in this problem with scattering of thermal neutrons in concrete.
- Can solve this with Energy Splitting, which split/roulette particles once they enter a certain energy range.

Energy Splitting

- **Energy splitting card:**

- **Form:** **ESPLT:n I1 E1 … IN EN**

  - First a list of importances where the default importance is 1.
  - If > 1, then neutrons are split when they enter energy region, if < 1, neutrons are rouletted.
  - List of energy bin bounds in descending order.

Exercise 5: Energy Splitting

- Copy caas4.txt to caas5.txt.
  - Introduce energy splitting at 1 keV and 1 eV.
  - Neutrons that downscatter below 1 keV should be rouletted 4 to 1 (I1 = 0.25)
  - Neutrons that downscatter below 1 eV should be rouletted 10 to 1 (I2 = 0.1).
  - Run the problem and analyze the results.

```
mcnp6  i = caas5.txt  o = caas5o.txt  rssa = source
```

  - Problem should run faster, but variance should be higher. This is okay because less time is spent tracking unimportant particles.

DXTRAN Transmissions

- **Inspect DXTRAN table:**
  - Search for string 1dxtran

contributions by cell

	cell	misses	hits	weight per history	weight per hit
1	100	92093	3034	1.43293E+05	4.72471E+07
2	101	23	2	1.09021E+03	5.45316E+08
3	102	4765	242	8.82326E+03	3.64738E+07
4	200	0	2	9.66173E+06	4.83273E+12
5	210	332	42	3.48516E+04	8.30121E+08
6	220	2	0	0.00000E+00	0.00000E+00
8	240	1	0	0.00000E+00	0.00000E+00
18	900	77127	2376	6.37678E+07	2.68486E+10
	total	174343	5698	7.36176E+07	1.29249E+10

- **Cell 200, the room with the detector, has very high weight per hit and only two collisions**

Forced Collisions

Forced Collisions

- **Indicates that rare collisions with air in the room contribute most to the tally.**
- **Want to sample collisions more often, but at a lower weight.**
- **MCNP can do this with forced collisions.**
  - **Forced collisions decompose the particle into a collided and uncollided part.**
  - **The collided part plays the DXTRAN game and can contribute to the tally.**
  - **This is done on a per cell basis only (for now).**

Forced Collisions

- **Forced Collision Card:**

- **Form:** FCL:n F1 F2 ... F(ncel)

  – List of forced collision parameters Fi from 0 to 1, which is the probability of playing the forced collision game.
  – May also specify each one individually on the cell card.

Exercise 6: Forced Collisions

- Copy caas5.txt to caas6.txt.
  - Need more resolution for forced collisions.
  - Add 7 px planes at x = -275, -300, -350, -400, -500, -750, -1000.
  - Break cell 200 into 8 separate cells.
  - Update the NONU card accordingly.
  - Set FCL:n=1 for all air cells in the detector room, experiment room, and adjoining doorway.
  - Increase NPS to 1e5.
  - Run the problem.

```
mcnp6  i = caas6.txt  o = caas6o.txt  rssa = source
```

  - Analyze output and DXTRAN table.

Tally Results

- Compare results from caas5o.txt and caas6o.txt:

```
cell   300
                    3.26341E-06 0.6501

cell   300
                    1.66358E-05 0.4379
```

- Large difference in tally result (factor of 5 increase) indicates previous result was undersampled.

DXTRAN Tables

- **Many more collisions (over 1000) sampled:**

contributions by cell

	cell	misses	hits	weight per history	weight per hit
1	100	927522	2328	1.40548E+05	6.03962E+07
2	101	22274	176	2.70857E+00	1.53955E+04
3	102	49182	228	2.48928E+04	1.09221E+08
4	200	6	223	7.86902E+06	3.53007E+10
5	201	3	208	9.54134E+06	4.58895E+10
6	202	4	197	4.26732E+07	2.16699E+11
7	203	4	175	1.15652E+07	6.61123E+10
8	204	1	161	1.38518E+07	8.60694E+10
9	205	0	168	2.08741E+07	1.24299E+11
10	206	0	103	6.43411E+06	6.24911E+10
11	207	3	65	7.68397E+05	1.18260E+10
...					
25	900	867239	3717	4.80408E+08	1.29296E+11
	total	1928549	8301	5.94214E+08	7.16110E+10

DXTRAN Tables

- **Few transmissions still dominate most of the scoring:**

times average weight	transmissions	cumulative fraction of transmissions	weight transmitted per history	cumulative fraction of total weight
1.0000E-01	662	0.07975	3.73242E+05	0.00063
1.0000E+00	912	0.18962	2.46540E+06	0.00478
2.0000E+00	340	0.23057	3.05984E+06	0.00993
5.0000E+00	440	0.28358	9.13241E+06	0.02530
1.0000E+01	277	0.31695	1.27528E+07	0.04676
1.0000E+02	657	0.39610	1.23412E+08	0.25445
1.0000E+03	178	0.41754	2.39811E+08	0.65802
1.0000E+38	22	0.42019	2.02635E+08	0.99904
before dd roulette	4813	1.00000	5.73101E+05	1.00000

- **Weight windows are useful to help equalize this.**

Weight Windows

Weight Windows

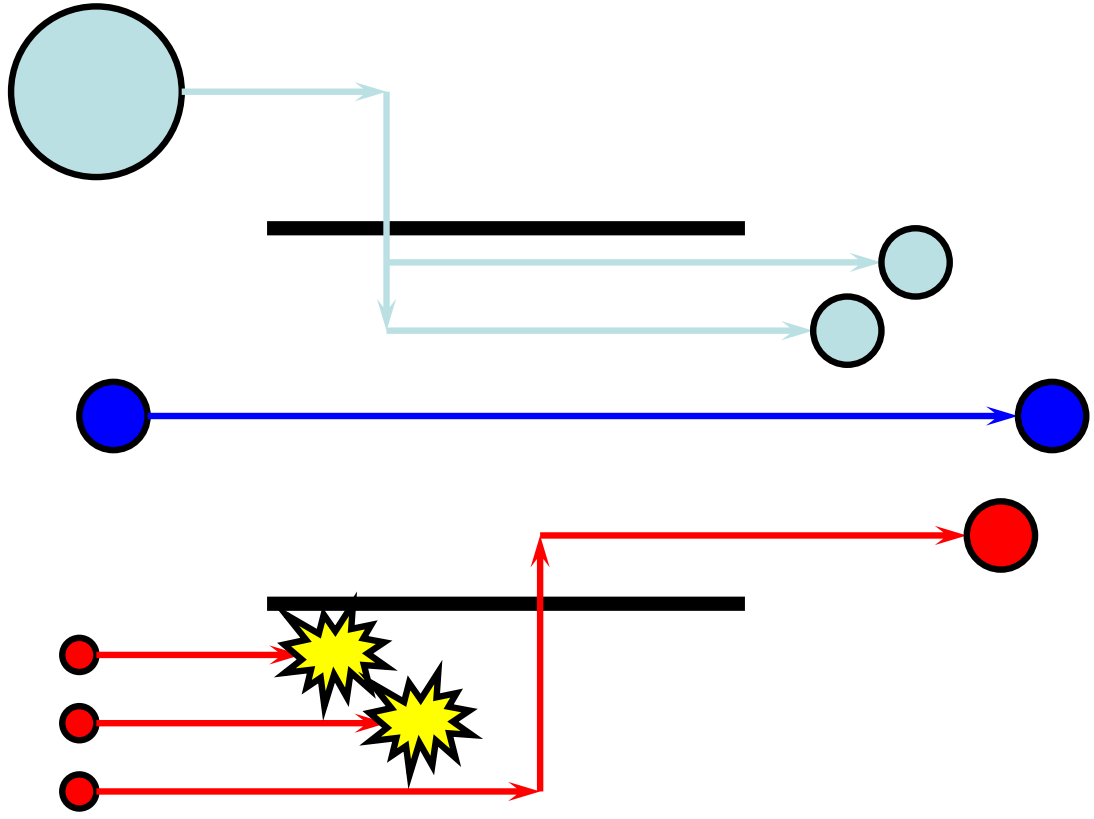
- Variation in particle weight leads to some variance in tally scores = higher uncertainty for a fixed number of histories

- Controlling the band of allowed weights in a region helps reduce this source of variance

Particles with weight above window split

Particles with weight inside window unaffected

Particles with weight below window undergo Russian Roulette



Weight Windows

- MCNP allows specification of the lower weight-window bounds as a function of space, energy, and time
  - Can be done by cell or (preferably) on a superimposed mesh
- The upper weight-window bound is determined by a constant global multiplier on the lower bound
  - Default is 5, can be changed only for the entire problem
- MCNP has an automated generator and can read lower weight-window bounds from external files
  - Results generated in a file called wwout
  - User specifies weight-window input file on the command line with:

```
wwinp = wwfilename
```

Weight-Window Generator

- **Weight-Window Generator card:**
- **Form: WWG TAL**
  - **Argument is the tally number for optimization**
- **Mesh-based weight windows needs MESH card:**
  - **See MCNP manual for description.**
  - **One will be provided for the exercise.**

Comment on Energy Splitting

- **The meaning of energy splitting changes when weight windows are present:**
  - **Without weight windows, particles are split/roulette based on the number on the ESPLT card.**
  - **With weight windows, the value of the weight window lower bounds are multiplied by the value on the ESPLT card. Splitting/rouletting is done only if outside the adjusted value of the weight window.**

Exercise 7: Weight Windows

- **Copy caas6.txt to caas7.txt.**
  - **Invoke the weight-window generator for detector tally.**
  - **Insert the following MESH card:**

```
mesh    geom = xyz   origin = -1300 -1900  -50   ref = 0 0 5
        imesh = -1250  -500  -250  -200   800   850  1850  1900
        iints =     1     5    10    10     4     1     1     1
        jmesh = -1850  -850  -800  -305  -295   200   250
        jints =     1     1     1     2     1     2     1
        kmesh =     0   290   300   310
        kints =     1     3     1     1
```

  - **Run the problem with wwout file named wwinp1:**

```
mcnp6  i = caas7.txt  o = caas7o.txt  rssa = source
       wwout = wwinp1
```

  - **Analyze output file results.**

Exercise 7: Weight Windows

- Copy caas7.txt to caas7a.txt.

- New set of weight windows are in the file generated.
  - To tell MCNP to get weight-window lower bounds from a file, insert the following weight-window parameter card:

```
WWP:N  4J  -1
```

  - Repeat the calculation with wwinp = wwinp1 on the command line and wwout as wwinp2.

```
mcnp6  i = caas7a.txt    o = caas7ao.txt  rssa = source
       wwinp = wwinp1   wwout = wwinp2
```

  - Inspect the tally results, statistical tests, and fluctuation charts.

Tally Fluctuation Chart

- **Tally fluctuation chart:**

tally 6

nps	mean	error	vov	slope	fom
8000	1.6000E-05	0.2739	0.5208	2.1	25
16000	2.7201E-05	0.4173	0.8890	2.2	5.1E+00
24000	2.6483E-05	0.3178	0.6172	2.0	5.7E+00
32000	2.5852E-05	0.2621	0.4737	1.9	6.5E+00
40000	2.6164E-05	0.2228	0.3641	1.9	7.4E+00
48000	3.3524E-05	0.3051	0.6111	1.8	3.3E+00
56000	3.4233E-05	0.2658	0.5271	1.8	3.8E+00
64000	3.4335E-05	0.2373	0.4793	1.8	4.0E+00
72000	3.2836E-05	0.2235	0.4578	1.8	4.0E+00
80000	3.6378E-05	0.2304	0.3184	1.8	3.4E+00
88000	3.4661E-05	0.2207	0.3140	1.8	3.4E+00
96000	3.2677E-05	0.2147	0.3136	1.8	3.4E+00
99761	3.2372E-05	0.2087	0.3122	1.8	3.4E+00

- **Uncertainties are still high, PDF slope < 2, indicating that there may be undersampling.**

DXTRAN Diagnostics

- **DXTRAN diagnostics show small number of histories contribute most to tally scores:**

times average weight	transmissions	cumulative fraction of transmissions	weight transmitted per history	cumulative fraction of total weight
1.0000E-01	130914	0.44569	1.44092E+08	0.06823
1.0000E+00	67387	0.67511	2.03444E+08	0.16456
2.0000E+00	4782	0.69139	7.24808E+07	0.19888
5.0000E+00	3102	0.70195	1.02035E+08	0.24719
1.0000E+01	1064	0.70557	7.84557E+07	0.28434
1.0000E+02	1030	0.70908	3.05051E+08	0.42879
1.0000E+03	92	0.70939	2.47417E+08	0.54594
1.0000E+38	9	0.70942	9.52540E+08	0.99697
before dd roulette	85351	1.00000	6.39980E+06	1.00000

- **Need to iterate again to get more sampling and better weight windows.**

Exercise 7: Weight Windows

- Copy caas7a.txt to caas7b.txt.
  - Double the number of neutron histories to 2e5.
  - Particles will start with lower weight (recall total weight preserved), but weight windows calibrated to old source weight.
  - To fix this, double the WGT entry on SSR card to 5.8e15.
  - However, must halve the multiplier on the FM card to make tally scores consistent.
  - Run again with new parameters and weight window file wwinp2 and wwout file wwinp3.

```
mcnp6  i = caas7b.txt    o = caas7bo.txt  rssa = source
       wwinp = wwinp2   wwout = wwinp3
```

  - Analyze results.

Tally Fluctuation Chart

- **Tally fluctuation chart:**

tally 6

nps	mean	error	vov	slope	fom
16000	6.0753E-05	0.3549	0.6794	1.7	6.6E+00
32000	5.1930E-05	0.2416	0.4032	1.8	7.3E+00
48000	6.4566E-05	0.1989	0.1694	1.7	7.1E+00
64000	5.6555E-05	0.1745	0.1562	1.8	7.1E+00
80000	5.3126E-05	0.1559	0.1337	1.8	7.2E+00
96000	5.0723E-05	0.1400	0.1204	1.8	7.5E+00
112000	4.9157E-05	0.1279	0.1073	2.0	7.7E+00
128000	5.2538E-05	0.1409	0.2044	2.1	5.6E+00
144000	5.1412E-05	0.1299	0.1935	2.2	5.9E+00
160000	5.0551E-05	0.1203	0.1844	2.3	6.2E+00
176000	4.8769E-05	0.1139	0.1805	2.4	6.2E+00
192000	5.0467E-05	0.1087	0.1451	2.4	6.3E+00
199772	5.4009E-05	0.1067	0.1170	2.4	6.3E+00

- **Improved uncertainty, VOV, and PDF slope.**

DXTRAN Diagnostics Table

- **DXTRAN diagnostics table shows far less contribution with very high weight**

times average weight	transmissions	cumulative fraction of transmissions	weight transmitted per history	cumulative fraction of total weight
1.0000E-01	224697	0.45391	5.28193E+08	0.09295
1.0000E+00	169851	0.79703	1.22091E+09	0.30779
2.0000E+00	16432	0.83023	5.44557E+08	0.40362
5.0000E+00	10588	0.85161	7.54294E+08	0.53635
1.0000E+01	3090	0.85786	5.00341E+08	0.62439
1.0000E+02	2375	0.86265	1.33853E+09	0.85993
1.0000E+03	152	0.86296	7.06564E+08	0.98427
1.0000E+38	2	0.86297	7.81290E+07	0.99802
before dd roulette	67835	1.00000	1.12798E+07	1.00000

- **Weight windows are working as expected. Try iterating again.**

Exercise 7: Weight Windows

- Copy caas7b.txt to caas7c.txt.

- Iterate again with more particles.
  - Double the number of neutron histories to 4e5.
  - Double the WGT entry on SSR card to 1.16e16.
  - Halve the multiplier on the FM card again.
  - Run again with new parameters and weight window file wwinp3 and wwout as wwinp4.

```
mcnp6  i = caas7c.txt    o = caas7co.txt  rssa = source
       wwinp = wwinp2   wwout = wwinp3
```

  - Analyze results.

Tally Fluctuation Chart

- **Tally fluctuation chart:**

tally 6

nps	mean	error	vov	slope	fom
32000	3.0224E-05	0.1918	0.1981	2.1	26
64000	4.1513E-05	0.2034	0.4089	1.8	11
96000	6.4146E-05	0.3190	0.7493	1.7	3.0E+00
128000	6.1763E-05	0.2533	0.6903	1.7	3.6E+00
160000	6.6198E-05	0.2046	0.5103	1.8	4.4E+00
192000	6.5499E-05	0.1759	0.4707	2.0	4.9E+00
224000	6.4277E-05	0.1566	0.4361	2.2	5.3E+00
256000	6.4818E-05	0.1406	0.3803	2.3	5.8E+00
288000	6.1463E-05	0.1325	0.3727	2.3	5.8E+00
320000	6.1318E-05	0.1212	0.3536	2.6	6.3E+00
352000	5.9053E-05	0.1154	0.3422	2.6	6.3E+00
384000	5.9649E-05	0.1135	0.2672	2.6	5.9E+00
401021	5.9550E-05	0.1099	0.2571	2.7	6.1E+00

- **Small improvement in PDF slope, higher VOV (might be from better sampling).**
- **Further improvements from weight windows unlikely.**

Exercise 7: Weight Windows

- Copy caas7c.txt to caas8.txt.

- Production run.
  - Increase NPS card to 1e7, adjust SSR weight and FM multipliers accordingly.
  - Remove WWG and MESH cards.
  - Run again with new parameters and weight window file wwinp4.

```
mcnp6  i = caas8.txt    o = caas8o.txt  rssa = source
       wwinp = wwinp4
```

  - Note NPS is still at about 1e6. Source particles sampled multiple times with different random number sequences.
  - Analyze final results to see if more sampling necessary.

Tally Fluctuation Chart

- **Tally fluctuation chart:**

tally 6

nps	mean	error	vov	slope	fom
64000	5.3344E-05	0.1089	0.0904	2.0	8.3E+00
128000	5.4596E-05	0.0982	0.1999	2.7	5.1E+00
192000	5.3603E-05	0.0754	0.1255	3.2	5.8E+00
256000	5.5552E-05	0.0682	0.0720	3.0	5.3E+00
320000	5.8662E-05	0.0807	0.2320	2.9	3.0E+00
384000	5.7549E-05	0.0713	0.1994	2.6	3.2E+00
448000	5.6034E-05	0.0639	0.1865	2.5	3.5E+00
512000	5.5265E-05	0.0577	0.1732	2.6	3.8E+00
576000	5.3481E-05	0.0535	0.1674	2.7	3.9E+00
640000	5.4066E-05	0.0502	0.1385	2.7	4.0E+00
704000	5.3421E-05	0.0471	0.1289	2.7	4.1E+00
768000	5.2963E-05	0.0445	0.1182	2.8	4.3E+00
832000	5.2434E-05	0.0420	0.1127	3.1	4.4E+00
896000	5.1383E-05	0.0401	0.1095	3.3	4.5E+00
960000	5.2090E-05	0.0386	0.0941	3.3	4.6E+00
1000385	5.2877E-05	0.0376	0.0836	3.3	4.6E+00

- **Good behavior in mean, uncertainty, VOV, and PDF slope. Well converged.**

Statistical Checks

- All but FOM trend check (probably least important) has passed.

results of 10 statistical checks for the estimated answer for the tally fluctuation chart (tfc) bin of tally 6

tfc bin	--mean--	---------relative error---------			----variance of the variance----			--figure of merit--		-pdf-
behavior	behavior	value	decrease	decrease rate	value	decrease	decrease rate	value	behavior	slope
desired	random	<0.10	yes	1/sqrt(nps)	<0.10	yes	1/nps	constant	random	>3.00
observed	random	0.04	yes	yes	0.08	yes	yes	constant	increase	3.27
passed?	yes	yes	yes	yes	yes	yes	yes	yes	no	yes

- From observed trends with the other calculations, results are probably reliable.
- Final answer: 5.28×10^{-5} Gy +/- 3.8%