OAK RIDGE National Laboratory

AGENDA

MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

SCALE Training: SCALE Nuclide Inventory and Radiation Shielding for Spent Fuel Storage and Transportation

September 29th - October 2nd, 2025, NRC Professional Development Center, Rockville, MD

Training summary: This 4-day course will provide intermediate level hands-on SCALE training on spent nuclear fuel depletion and shielding calculations, with real-world example problems involving spent fuel storage and transportation systems. It will first cover the use of ORIGEN for isotopic depletion, decay and radiation source-terms calculation, and the use of the ORIGAMI tool for quick calculation of spent fuel sources. The course also provides training on MAVRIC-Shift, including building complex 3D models (materials and compositions); using a connection to the ORIGEN libraries to model simple radioactive sources; importing complex ORIGEN sources; and calculating neutron fluxes. Additionally, the advanced variance reduction tools for deep penetration problems, CADIS and FW-CADIS, that are the foundation of MAVRIC will be covered. This class uses the Fulcrum user interface for interactive model setup, visualization, computation, and output review.

Note:
Previous experience with the SCALE/KENO-VI geometry is recommended. This course will require an active SCALE 6.3 license in order to participate.

Instructors: Cihangir Celik, Steve Skutnik, Lisa Fassino

OAK RIDGE National Laboratory

AGENDA

MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

Day 1: Monday, September 29th, 2025

Time		Session
ORIGEN for decay calculations and OPUS Post-processing		
8:00 am	12:00 pm	(01) Welcome, Introductions, Logistics, Course Objectives (02) Overview of Depletion/Decay Sequences and Relevant Files (03) Introduction to ORIGEN o Simple Decay Tutorial (04) OPUS Post-Processing Utility for ORIGEN f71 Files (05) ORIGEN Radiation Emission Calculations Tutorial (06) ORIGEN Reactor Libraries
Lunch break (12:00 pm – 1:00 pm)		
ORIGAMI for easy spent fuel calculations		
1:00 pm	5:00 pm	(07) ORIGAMI Introduction: Express Form Input for Rapid Spent Fuel Calculations o Tutorial: PWR Decay Heat o Tutorial: PWR Nuclide Inventory o Tutorial: LEU/LEU+ HBU Decay Heat Comparison o Workshop: PWR/BWR Decay Heat Comparison o Workshop: PWR/BWR Nuclide Inventory Comparison (08) ORIGAMI In-Depth: Detailed Input for Rapid Spent Fuel Calculations for 0-D and 1-D (Through slide 53) o Tutorial: BWR Axial Gradient o Workshop: Forsmark BWR Decay Heat o Workshop: Calvert Cliffs PWR Nuclide Inventory o Workshop: Ringhals PWR Decay Heat Axial Variation Day 1 Retrospective

Day 2: Tuesday, September 30th, 2025

Time		Session
Review of the standard composition input format and KENO-VI/Shift Geometry		
8:00 am	12:00 pm	(09) Standard Composition Review and Exercises (10) KENO-VI Geometry Review and Exercises
Lunch break (12:00 pm – 1:00 pm)		
MAVRIC basics		
1:00 pm	5:00 pm	(11) MAVRIC Basics o Tutorial: Iron Sphere o Workshop: Neutron Howitzer o Workshop: Graphite Shielding Measurements Day 2 Retrospective

OAK RIDGE National Laboratory

AGENDA

MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

Day 3: Wednesday, October 1st, 2025

Time		Session
MAVRIC basics and CADIS for variance reduction		
8:00 am	12:00 pm	(11) MAVRIC Basics - Continued (12) MAVRIC CADIS o Tutorial: Spent Fuel Cask Dose Rates o Workshop: Graphite Shielding Measurements Revisited
Lunch break (12:00 pm – 1:00 pm)		
MAVRIC FW-CADIS for tally optimization		
1:00 pm	5:00 pm	(13) MAVRIC FW-CADIS o Workshop: Storage Cask Farm Variance Reduction Day 3 Retrospective

Day 4: Thursday, October 2nd, 2025

Time		Session
Review of MAVRIC Exercises, Q&A, and Best Practices Discussion		
8:00 am	12:00 pm	(11) – (13) For Review
Lunch break (12:00 pm – 1:00 pm)		
ORIGAMI Beyond 1-D, SFR Demo, ORIGEN/ORIGAMI Review, Q&A, and Discussion		
1:00 pm	5:00 pm	(8) ORIGAMI In-Depth: Beyond 1-D (Slide 54+) o Workshop: SFR Demo – Using ORIGAMI with a custom library (2) – (8) For Review

**The morning and afternoon sessions will include 10-15 minutes breaks.*



OAK RIDGE National Laboratory

September 29th, 2025

SCALE Nuclide Inventory and Radiation Shielding for Spent Fuel Storage and Transportation

NRC Professional Development Center, Rockville, MD
September 29th - October 2nd, 2025

U.S. DEPARTMENT OF ENERGY

ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

Overview

Summary:

- This is a 4-day, hands-on SCALE training course for spent nuclear fuel depletion and shielding calculations

Experience Level:

- Intermediate - previous experience with SCALE and KENO-VI geometry is recommended.

Relevant SCALE sequences:

- **ORIGEN** (Depletion, decay, and activation)
- **ORIGAMI** (ORIGEN Assembly Isotopics for spent fuel inventories)
- **MAVRIC** (Fixed-source radiation transport with automated variance reduction)

Instructors



Cihangir Celik

Senior R&D Staff
MAVRIC Product Owner
celikc@ornl.gov



Steve Skutnik

R&D Staff
ORIGAMI Product Owner
skutnikse@ornl.gov



Lisa Reed Fassino

R&D Associate
VADER Product Owner
reedke@ornl.gov

Daily Schedule

Day 1

AM:
Debrief on SCALE 6.3.3 and learning ORIGEN basics

PM:
Use ORIGAMI for spent fuel calculations

Day 2

AM:
Review SCALE composition and KENO VI geometry definitions

PM:
Learn MAVRIC basics and how to feed data from ORIGAMI into MAVRIC

Day 3

AM:
MAVRIC and MAVRIC CADIS exercises

PM:
MAVRIC and MAVRIC FW-CADIS exercises

Day 4

AM:
Finish and review exercises

PM:
Review, discuss best practices and perform ORIGAMI SFR demo

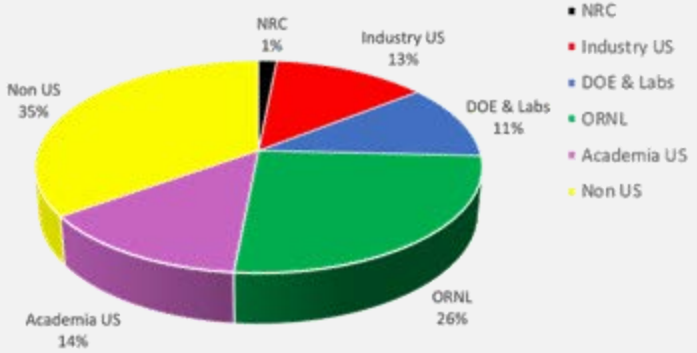
SCALE Resources

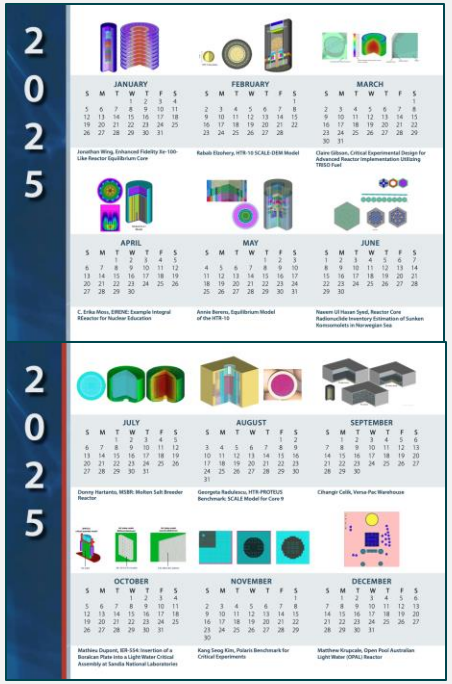
- Reach the SCALE team at **scalehelp@ornl.gov**
  - Request access to SCALE Users' Group google forum via **scalehelp**
- Large collection of learning resources from training, primers, and workshops
  - **https://www.ornl.gov/scale/learning**
- **SCALE 6.3 validation reports** are available:
  - Reactor Physics: https://doi.org/10.2172/2583839
  - Criticality Safety: https://doi.org/10.2172/2575336
  - Spent Nuclear Fuel: https://doi.org/10.2172/2587031
  - Radiation Shielding: https://doi.org/10.2172/2462890

Archives of publicly available models, reports, and journal papers for users can be found here:
https://code.ornl.gov/scale/analysis
https://www.ornl.gov/scale/references

2025 Annual SCALE Users' Group Workshop Recap

132 attendees from 20 countries





- **12 Tutorials offered**
- **13 Model contest entries**
- **Other sessions: Invited talks, open mic, upcoming feature preview, SCALE team highlights**

Slides available at:
https://www.ornl.gov/content/scale-users-group-annual-workshop

Next Users' Group Workshop: June 8th-12th, 2026

SCALE Team

Leadership

Director
Will Wieselquist

Production Coordinator
Rob Lefebvre

User Support Coordinator
Germina Procop

Logistics Coordinator
Sandra Poarch

Advisors
Matt Jessee
NEAMS Reactor Physics Lead

Tara Pandya
Section Head, Nucl. Appl. and Methods

Ugur Mertyurek
NRC Sponsor Representative

ORNL/NRC Projects Office

NRC Program Manager	NRC Business Ops
Bruce Bevard	Lindsey Aloisi

ORNL/NCSP Management

NCSP Execution Manager	ORNL Site Manager
Doug Bowen	Alex Lang

Product Owners

AMPX	Jordan McDonnell
CSAS	Kursat Bekar
DATA	Jesse Brown
FULCRUM	Rob Lefebvre
MAVRIC	Cihangir Celik
OMNIBUS	Seth Johnson
ORIGAMI	Steve Skutnik
ORIGEN	B.K. Jeon
POLARIS	Matt Jessee
SAMPLER	Ugur Mertyurek
STDCOMP	Rob Lefebvre*
TITAN	Tarek Ghaddar
TRITON	Rike Bostelmann
TSUNAMI	Jordan McDonnell*
VADER	Lisa Fassino
XSPROC	Kang Seog Kim

*Interim owner

Development/Testing/Analysis

Iyad Al-Qasir
Arzu Alpan
Goran Arbanas
Mehdi Asgari
Jin Whan Bae
Kursat Bekar
John Batson, III
Elliot Biondo
Rike Bostelmann
Jesse Brown
Chris Chapman
Cihangir Celik
Eva Davidson
Greg Davidson
Mathieu Dupont
Rabab Elzohery
Tom Evans
Lisa Fassino
Tarek Ghaddar
Travis Greene
Will Gurecky
Steven Hamilton
Donny Hartanto
Andrew Holcomb
Jianwei Hu
BK Jeon
Seth Johnson
Kang Seog Kim
Alex Lang
Brandon Langley
Rob Lefebvre
BJ Marshall
Jordan McDonnell
Ugur Mertyurek
Walid Metwally
Tara Pandya
Douglas Peplow
Marco Pigni
Germina Procop
Georgeta Radulescu
Kemal Ramic
Yves Robert
Kat Royston
Jose S. Perez
Alex Shaw
Steve Skutnik
Ben Taylor
Ben Coalson
Doro Wiarda
Will Wieselquist
Blake Wilkerson

OAK RIDGE National Laboratory

Overview of Depletion/Decay Sequences and Relevant Files

SCALE 6.3.2

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals

- Introduce users to the SCALE sequences used for depletion and decay workflows
- Illustrate how SCALE modules/sequences are used together to perform depletion and decay calculations

OAK RIDGE National Laboratory

Nomenclature refresher



Irradiated nuclear fuel

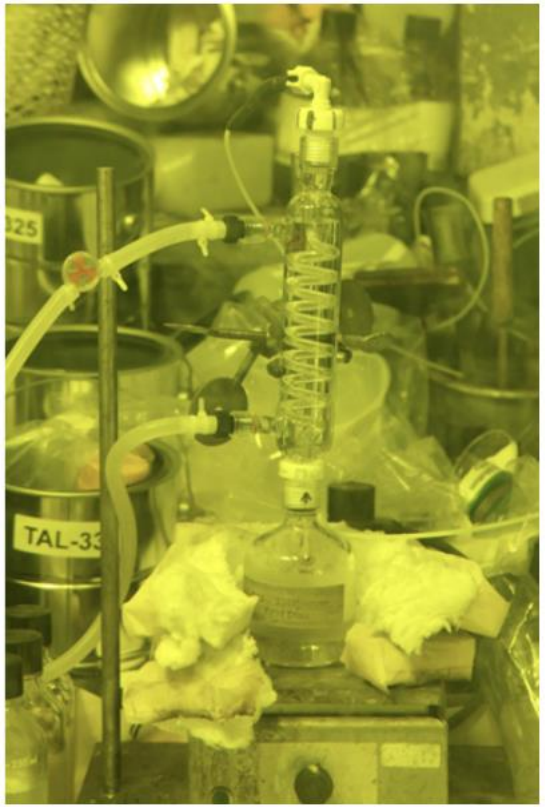
- Fresh fuel inserted in a reactor (e.g., U-based) undergoes nuclear transmutations, leading to production of higher actinides (e.g., Pu) and fission products (FPs)
- What happens to the fuel material irradiated in the reactor?
  - Irradiation changes the fuel material composition
  - U is lost through transmutation (mostly fission)
  - Pu is produced through neutron interactions (transmutation) of U
  - FPs are generated by fission of U, Pu
- Relevant metrics for spent fuel characterization
  - Nuclide inventories
  - Decay heat
  - Radioactivity
  - Radiation emissions (neutron, gamma sources)

Primary parameters for irradiated nuclear fuel

- Burnup
  - Measure of the integral number of fission events occurring in fuel during irradiation
  - Quantifies the amount of energy produced during irradiation
  - Unit: MWd/t or %FIMA
  - Burnup (MWd/t) = power (MW) × irradiation time (days)/initial heavy metal mass (metric ton)
  - %FIMA = (number of) fissions per initial metal atom, in percent
- Cooling time
  - Duration between the date the irradiated fuel is discharged from the reactor and a reference time of interest
  - Units: years, days
  - Highly relevant for decay heat and radiation emissions calculations
- Initial enrichment
  - Measure of the fissile mass in fresh fuel
  - Units: % ^{235}U/ U total for U-based fuel

Primary irradiated fuel parameters: Burnup

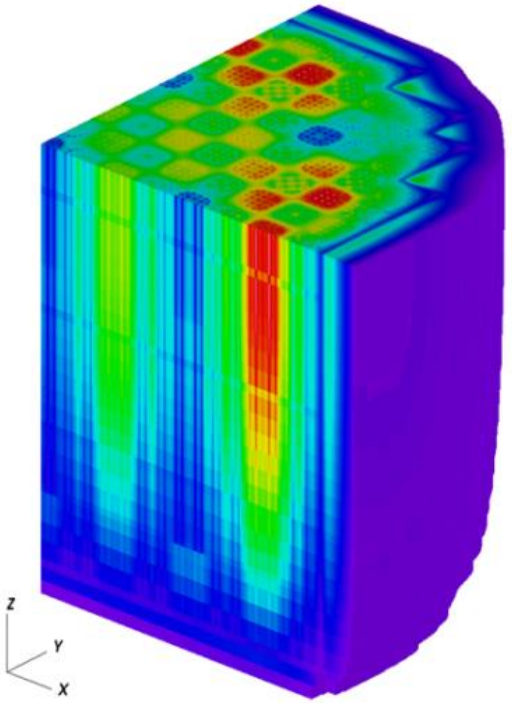
- Burnup estimates by destructive assay (DA) experiments
  - Based on burnup indicator fission products (e.g., ^{148}Nd) as a measure of total number of fissions in fuel
  - Requires cutting and dissolving fuel for radiochemical analysis (RCA)
  - Burnup $\propto$ ^{148}Nd (nuclide concentration)
  - Accuracy in burnup estimation based on ^{148}Nd generally $< 3\%$
- Burnup estimates by non-destructive assay (NDA) experiments
  - Based on neutrons and/or gamma measurements
  - Burnup $\propto$ ^{137}Cs (gamma)
  - Burnup^4 $\propto$ ^{244}Cm (neutron)
  - Requires knowledge of enrichment, cooling time, and reactor operating history
  - Codes are generally used to correlate NDA with burnup
  - Accuracy in burnup estimation based on NDA ~10%



Hot cell used for DA
ORNL/TM-2022/2788

Computer codes for fuel irradiation simulations

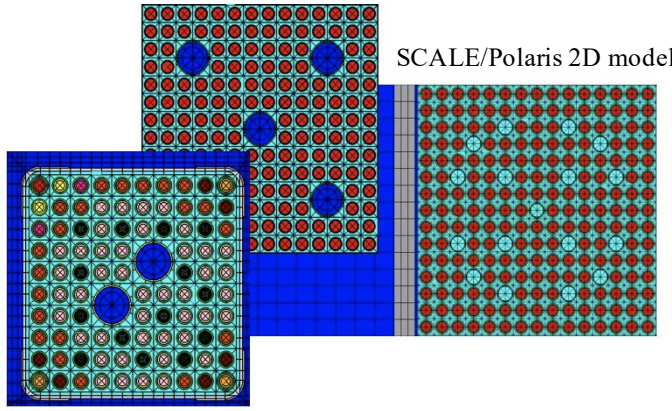
- Reactor core simulators (3D)
  - Simulate the entire reactor core to calculate the spatial distribution of power (and burnup) of each fuel element in the reactor
  - Used by reactor operators for operation and fuel management
  - Generally, require 3D modeling capabilities, significant computing power, and long computing time
  - Examples: VERA, CASMO/SIMULATE
- Depletion/burnup and decay codes (0D)
  - Simulate a single fuel element
  - Can calculate the complete set of nuclide contents in irradiated fuel
  - Can calculate radiation emissions used in spent fuel verification
  - Do not directly consider the spatial and energy dependence in the neutronic environment (however, this can be accounted for using the approach for generating cross section data)
  - Are very fast (seconds) compared to core simulators
  - Examples: ORIGEN, CINDER, FISPIN



CASLU-2015-0206-000, CASL (June 2016)

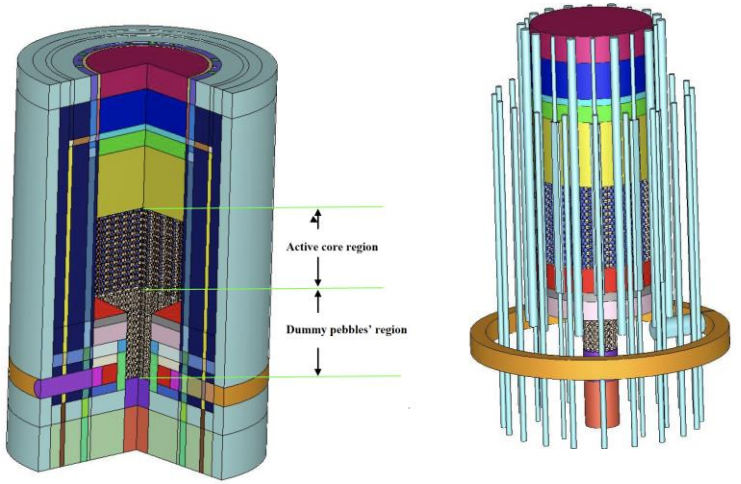
Computer codes for fuel irradiation simulations (cont.)

- Lattice physics computational tools (1D, 2D or 3D)
  - Can simulate a fuel element (assembly) to calculate the spatial distribution of power (and burnup) of each fuel pin in an assembly (e.g., for LWR)
  - May simulate a set of fuel assemblies or even the entire reactor for small configurations (e.g., small research reactors), depending on the problem
  - Generally used to obtain burnup-dependent lattice physics parameters for use in core simulators (e.g., nodal cross sections)
  - Generally require 2D or 3D modeling capabilities, moderate computing power, and longer computing time (compared to 0D depletion codes)
  - Iteratively couple a 1D, 2D, or 3D neutron transport solver with a 0D depletion/burnup solver (e.g., ORIGEN)
  - Examples: SCALE/TRITON, SCALE/Polaris, Serpent



SCALE/Polaris 2D models

Wide range of LWR geometry support



SCALE/KENO-VI 3D model for HTR-10 pebble bed reactor

(ORNL/TM-2020/1500/v3)

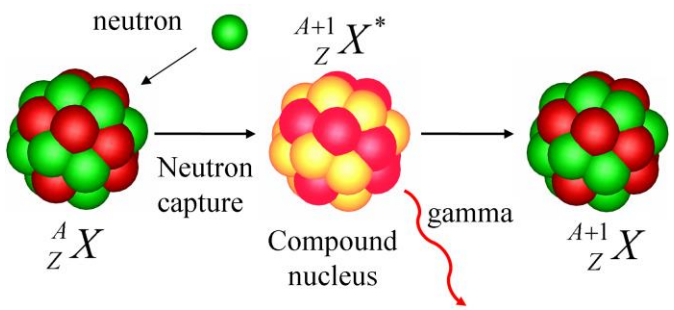
OAK RIDGE National Laboratory

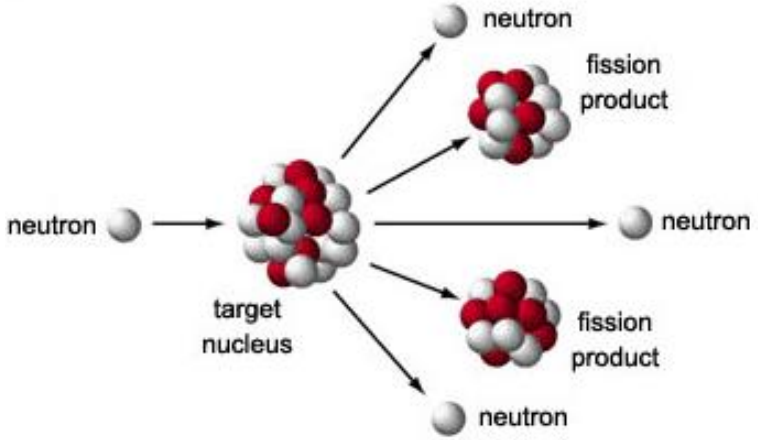
ORIGEN is the foundation of depletion capabilities in SCALE

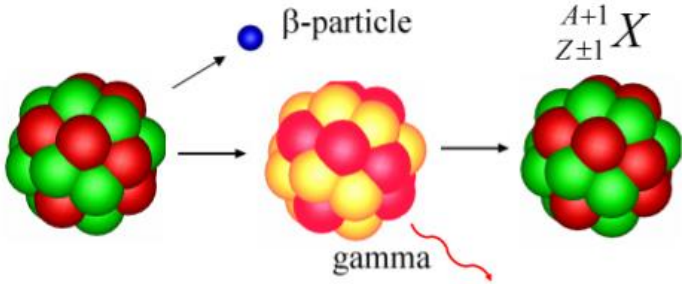


What is ORIGEN ?

- Oak Ridge Isotope Generation code in SCALE
- Irradiation and decay simulation code
- Explicit simulation of all pathways from neutron transmutation, fission, and decay



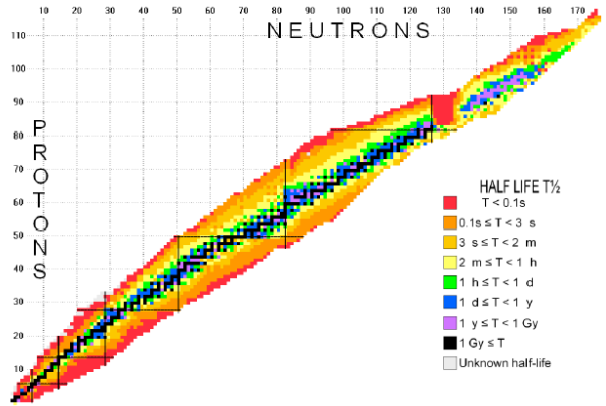




ORIGEN highlights

$$\dot{\boldsymbol{N}}(t) = \overline{\overline{\boldsymbol{A}}} \cdot \boldsymbol{N}(t)$$

- How does it work?
  - Solves the time-dependent transmutation equations over large time scales
  - Does not directly treat space and energy variables
  - Primary nuclear data needed: 1-group cross sections, fission yield data, decay data
  - Accurate problem-dependent cross section generation and accurate nuclear data are the secret
  - Based on state-of-the art nuclear data and cross section processing methodologies
  - Fast computing time (seconds)
- Where is ORIGEN present in SCALE?
  - Standalone depletion/decay code
  - Engine of ORIGAMI tool
  - Depletion solver in SCALE lattice physics codes (TRITON, Polaris)



ORIGEN family in SCALE

- ORIGEN
  - Main engine: solves depletion, decay, activation, and feed/removal problems, as well as radiation emission calculations (0D code)
- ORIGAMI (ORIGEN Assembly Isotopics)
  - Provides the capability to easily perform depletion and decay calculations with ORIGEN for LWR fuel assemblies
- OPUS (ORIGEN Post-processing Utility for SCALE)
  - Utility to perform post-processing and analysis of ORIGEN results contained in ORIGEN concentrations files, including sorting, ranking, and unit conversion
- ARP (Automatic Rapid Processing)
  - Cross section interpolator – interpolates on a set of pre-generated ORIGEN libraries (see next slide) to create a new ORIGEN library at different values of the interpolation parameters (e.g., burnup)
- OBIWAN (ORIGEN Binary Interrogation Without a SCALE Input)
  - Command-line utility for viewing, manipulating, and converting ORIGEN binary files F71, F33 (see next slide)

Special files: F71 and F33

- ORIGEN concentration file (F71)
  - Stores all nuclide inventory (always present) and neutron/gamma sources (optional)
  - Content can be visualized with Fulcrum and interrogated with OBIWAN
  - Consistent usage across the ORIGEN family
- ORIGEN library file (F33)
  - Special "library" that contains all the information needed to "reconstruct" a depletion calculation (details to be presented later this week)
  - Can be generated with TRITON or Polaris
  - Integral part of the ORNL ORIGEN-ARP method (more details in ORIGAMI presentation)
  - Consistent usage across ORIGEN, ORIGAMI, and ARP
  - Content can be interrogated with OBIWAN (see Section 5.2 of SCALE manual)

OAK RIDGE National Laboratory

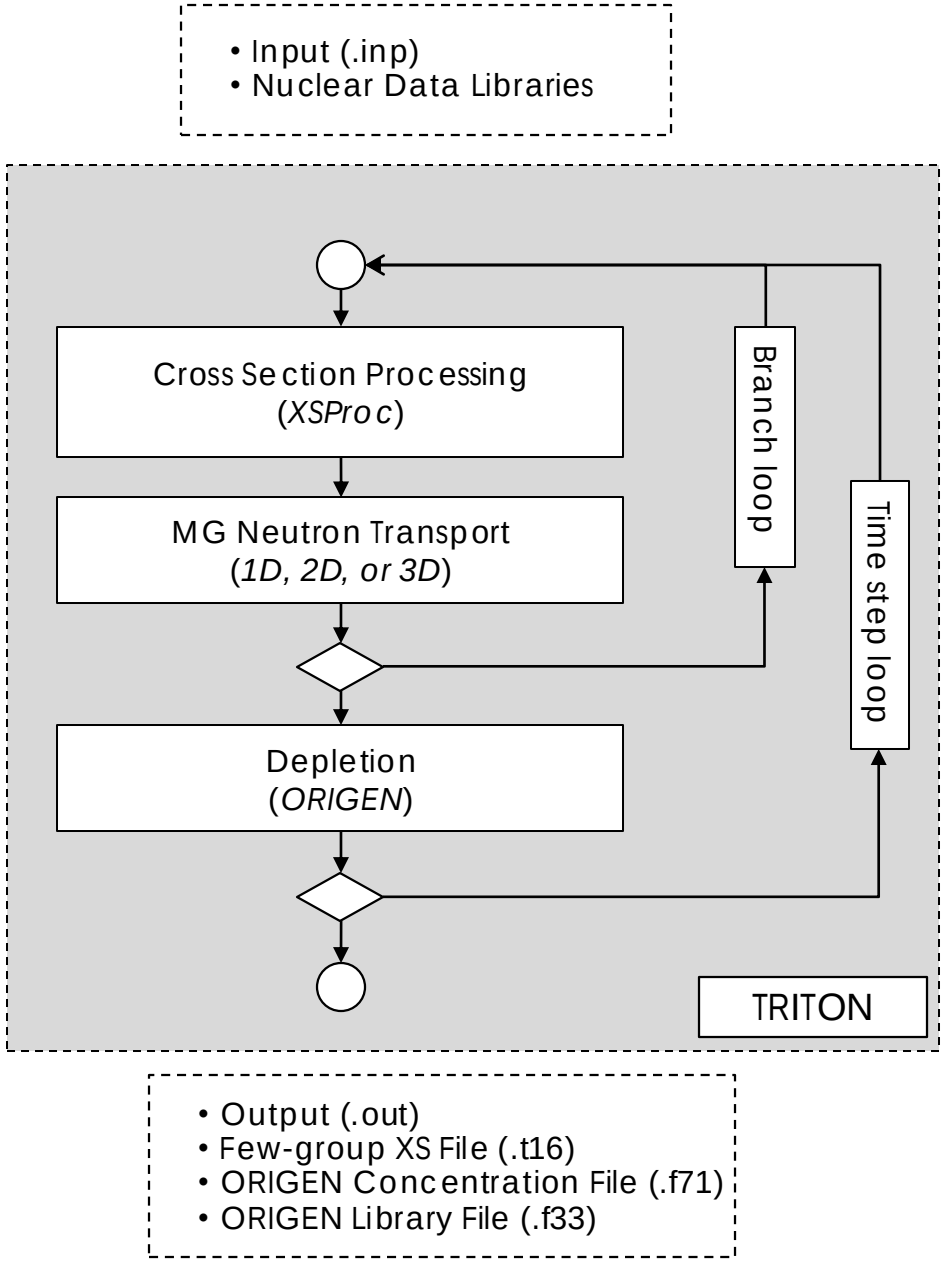
Depletion sequences in SCALE



TRITON depletion sequence

TRITON: SCALE's modular, general-geometry, reactor physics sequence

- Cross section processing: XSPROC
  - Wide range of self-shielding options
  - Same module used in criticality safety and shielding sequences
- Multigroup neutron transport:
  - 1D SN (XSDRNPM)
  - 2D ESC-SN (NEWT)
  - 3D Monte Carlo (KENO-V.a, KENO-VI, Shift)
- Transport-to-depletion coupling
  - Normalizes power/flux levels
  - Prepares transition matrices for ORIGEN
  - Manages time-stepping (predictor-corrector)
- Depletion (ORIGEN)
- TRITON in SCALE 6.2+ also supports 3D CE depletion



Basic capabilities for TRITON depletion

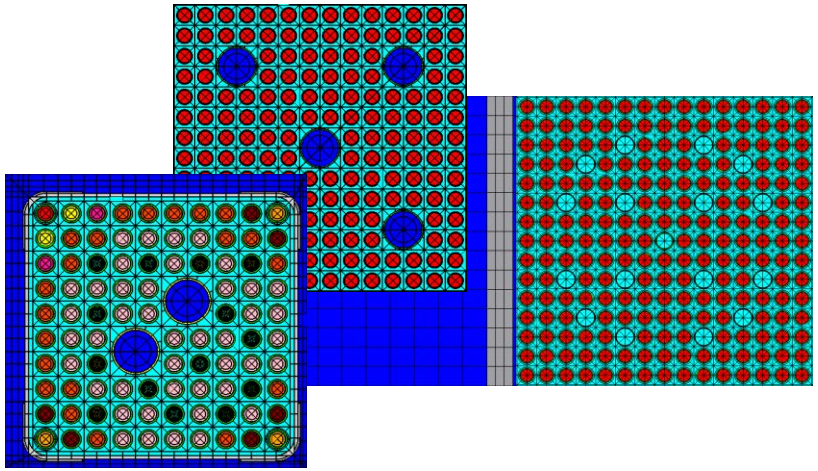
- Calculates evolution of nuclide concentrations during fuel depletion (irradiation, decay)
  - TRITON-NEWT: 2D general system depletion (available in SCALE 5.1 and later)
  - TRITON-KENO: 3D general system depletion (available in SCALE 5.1 and later)
  - TRITON-Shift: 3D general system depletion (new in SCALE 6.3)
- Uses ORIGEN as depletion/decay solver
- Calculates 1-group cross sections for ORIGEN during the depletion simulation
  - Integral part of special ORIGEN+ARP strategy for fast spent fuel nuclide inventory and source term calculations (in ORIGAMI)

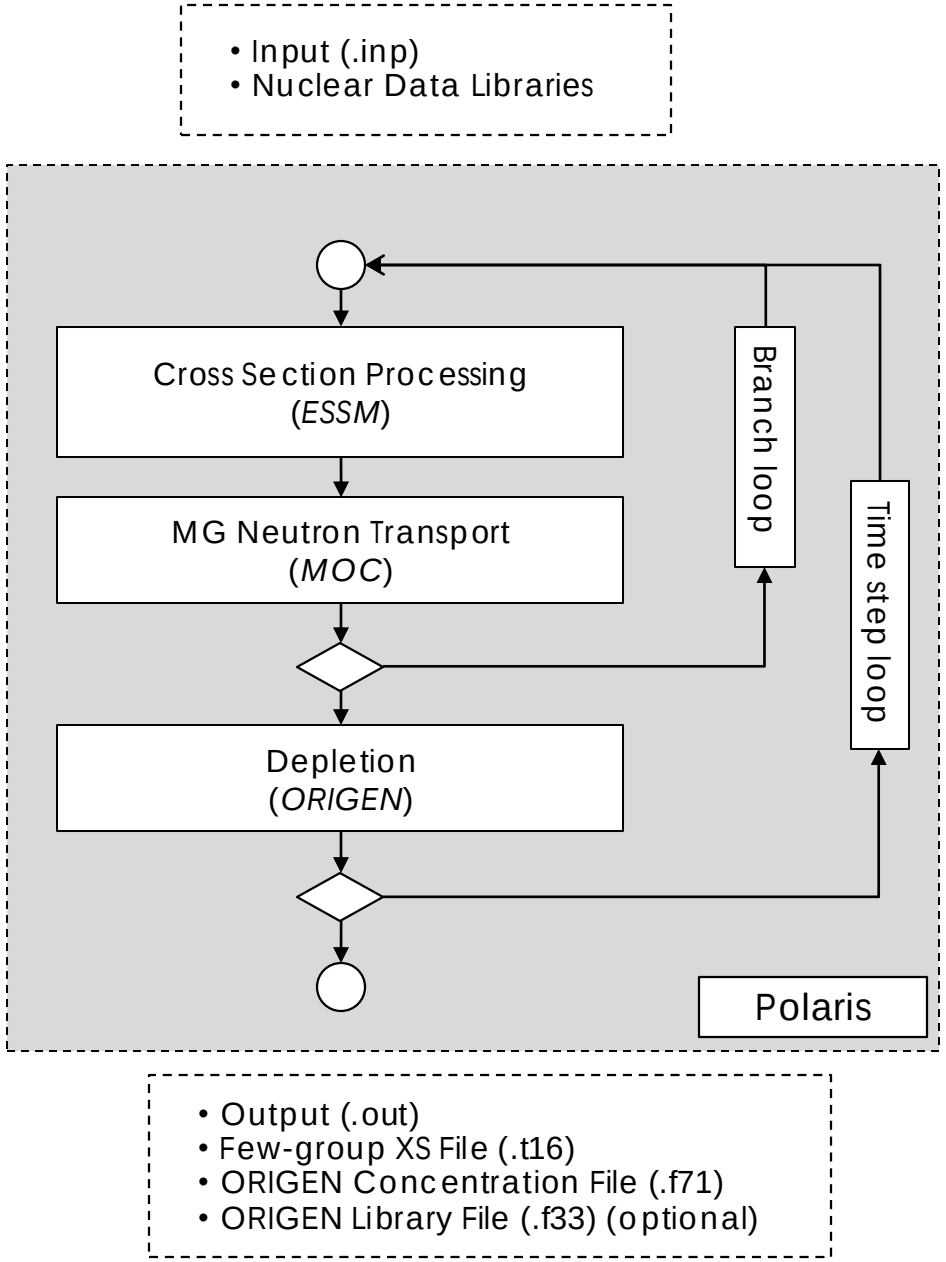
F33 File
- 1-group cross sections vs. burnup
- decay constants
- decay heat Q-values

F71 File
- inventory (moles per nuclide)
- gamma/neutron spectra (optional)

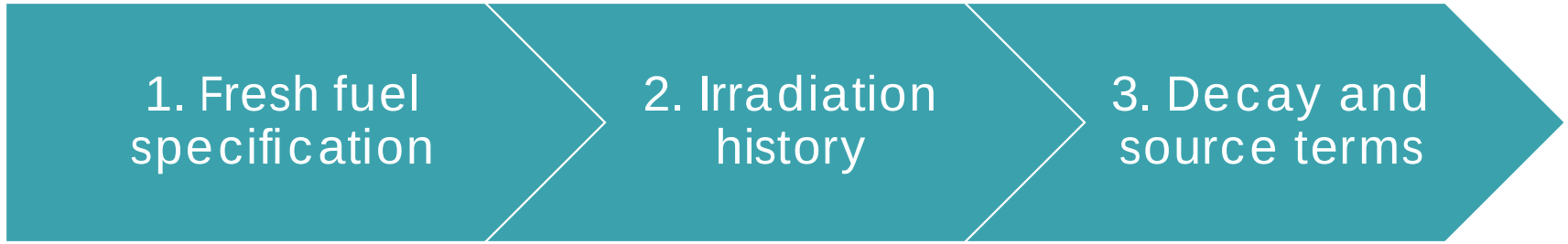
Polaris lattice physics sequence

- SCALE's fast, easy-to-use 2D lattice physics capability for LWR analysis
- Simple input
  - Assembly geometry
  - Material definitions
  - Range of system conditions
- Wide range of LWR geometry support





Typical fuel depletion analysis with SCALE



1. Fresh fuel composition/design are specified in the SCALE input file
   - TRITON (any general geometry)
   - Polaris (LWR geometry only)
   - ORIGAMI (geometry assumed based on the pre-generated ORIGEN reactor library file, F33, that is specified in input)
2. Irradiation history is specified in the SCALE input file
   - Power vs. time
   - State (void, temperature) vs. time (in TRITON and Polaris)
3. Decay with ORIGEN to calculate inventories, decay heat, and neutron/gamma sources as a function of desired cooling time
   - Read discharge nuclide inventory from F71 file and decay to a cooling time of interest

Relevant output from within TRITON and Polaris

- ORIGEN concentrations file (F71)
  - TRITON, Polaris, and ORIGAMI depletion automatically generates F71
  - ORIGEN calculations
    - User must provide energy bins for neutron and gamma emission calculations and explicitly request the emission calculations; also, must explicitly request saving information to a new F71
- ORIGEN library file (F33)
  - TRITON automatically generates F33 files
    - One F33 is generated per each depleted mixture in the problem; one combined F33 is also generated, representative of the entire system that is modeled

      basename.mixNNNNNN.f33, where NNNNNN is the material number and BASENAME is the root name of the input file (BASENAME.inp)

      basename.system.f33
  - Polaris generates F33 per request in the input file (*opt DEPL* card)
    - One F33 can be generated per each depleted mixture and for the entire system

      filename.hname.XXX.f33, where XXX is the name of the material (or system), filename is the root name of the input file, and hname is the name for the history

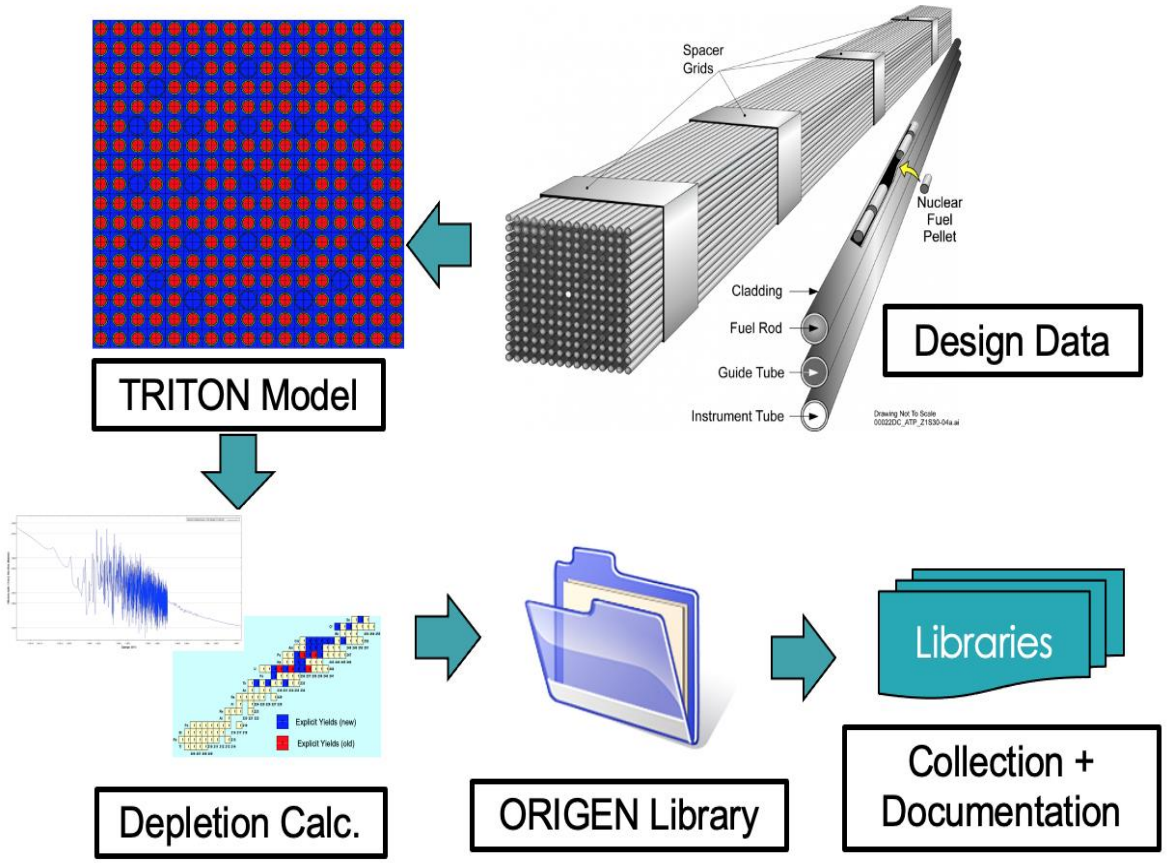
OAK RIDGE National Laboratory

Fast depletion calculations with SCALE

Typical approach in a nutshell



Generate ORIGEN reactor libraries (F33) with TRITON or Polaris



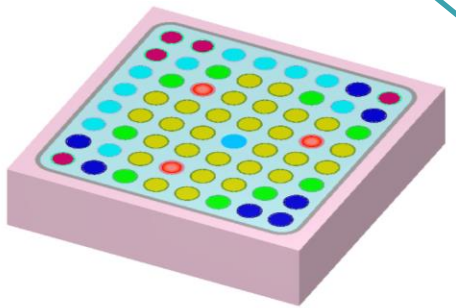
We'll learn how to generate reactor libraries later this week

Calculate source terms with ORIGAMI

using an ORIGEN reactor library and assembly data as input



ORIGEN reactor library

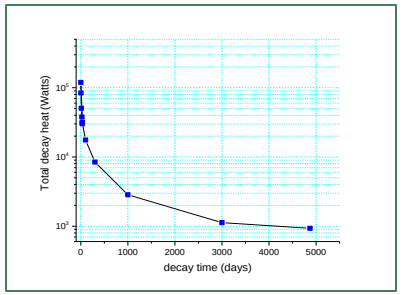


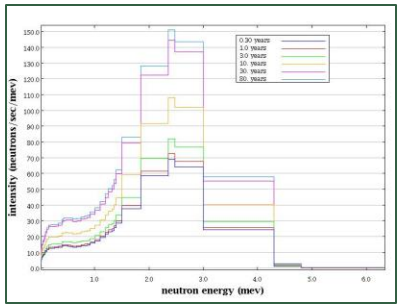
Assembly data
irradiation/decay
history data

ORIGAMI

(ORIGEN)

Nuclide inventories
Decay heat
Activities
Sources and spectra

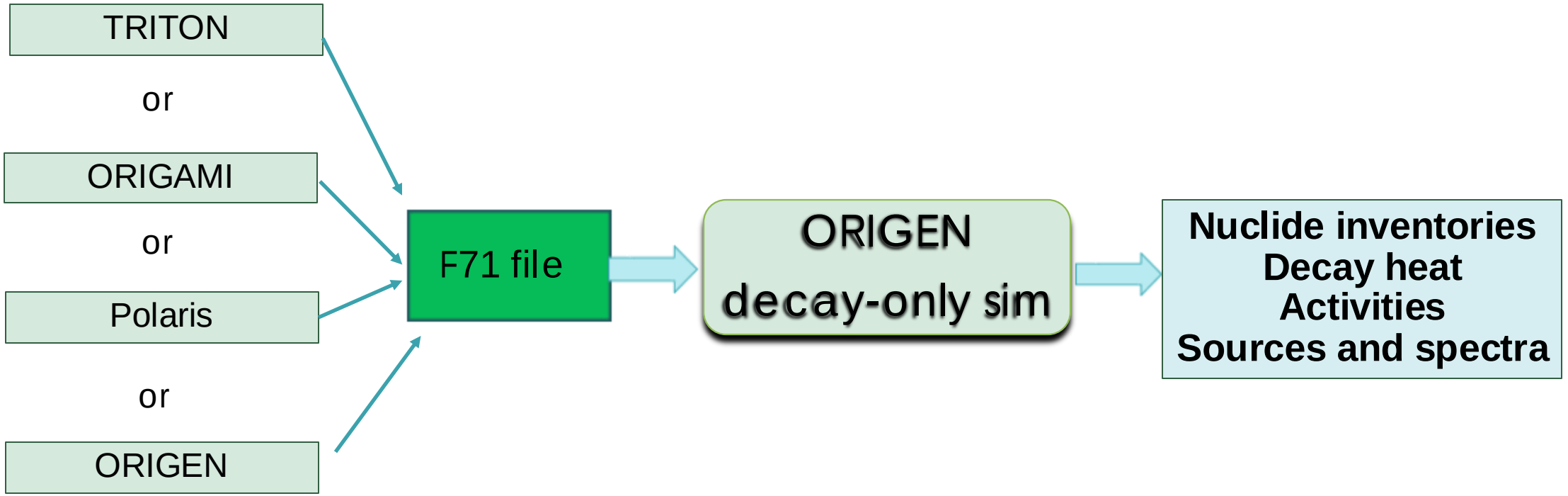




We'll learn hands-on how to use ORIGAMI later this week

Source terms calculation via ORIGEN decay

using available nuclide inventory from an F71 file as input



We'll learn hands-on how to perform ORIGEN decays later today and throughout this week

Summary

- ORIGEN is the foundation of all depletion capabilities in SCALE
  - Used as standalone or under the hood in ORIGAMI, TRITON, and Polaris sequences
- There are two highly relevant files for depletion, which are consistently used across all SCALE depletion sequences
  - ORIGEN nuclide concentration file (F71)
  - ORIGEN reactor library file (F33)
- Best approaches and practices on how to use the depletion capabilities may vary, depending on the application of interest
  - An example is presented for source term calculations (once the F71 file is available, the solution is only an ORIGEN-decay away)

OAK RIDGE National Laboratory

Introduction to ORIGEN

SCALE 6.3.2



ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals



Gain insight into ORIGEN's purpose and main capabilities



Become familiarized with generating and running ORIGEN input for a simple decay problem using Fulcrum

OAK RIDGE National Laboratory

What is ORIGEN?



Overview of ORIGEN

- Oak Ridge Isotope Generation code in SCALE
- Irradiation and decay simulation code
- Explicit simulation of 2,237 nuclides in database
  - 176 actinides
  - 1,151 fission products
  - 910 structural activation nuclides
- Explicit simulation of all pathways from neutron transmutation, fission, and decay
  - Includes all nuclides with half-lives > 1 ms
  - Accurately represents the evaluated nuclear data
  - Many physics codes track a small subset of isotopes

ORIGEN's purpose:
Solve for the nuclide inventory as a function of time: $N_i(t)$

- From the fundamental nuclide inventory result, it is a "unit conversion" to produce decay heat, activity, mass, etc.
- Applications
  - Spent nuclear fuel characterization
    - Nuclide concentrations (different units)
    - Activities
    - Decay heat
    - Radiation emission rates and spectra (neutron and gamma)
  - Source terms for accident analyses (operating reactors, spent fuel handling, storage)
  - Structural material activation (in-core, ex-core)
  - Fuel cycle analysis (material feed and removal processing)

ORIGEN enables a comprehensive isotopic characterization of fuel over a large time scale, including repository analysis.

ORIGEN is a high-performance ODE solver

General ordinary differential equation (ODE) solved for production and loss rate of a nuclide

Rate of change in nuclide concentration = Production rate of nuclide − Loss rate of nuclide

Rate of change in nuclide i concentration = Decay into nuclide i by other nuclides j + Production of nuclide i from irradiation of other nuclides k − Loss of nuclide i through decay, irradiation, or other means

$$\frac{dN_i}{dt} = \sum_{j=1}^{m} l_{ij}\lambda_j N_j + \bar{\Phi}\sum_{k=1}^{m} f_{ik}\sigma_k N_k - (\lambda_i + \bar{\Phi}\sigma_i + r_i)N_i + F_i \quad (i = 1, \ldots m)$$

Representation of ORIGEN input data

The transmutation equation can be written in matrix form:

$$\frac{d\boldsymbol{N}}{dt} = \overline{\overline{\boldsymbol{A}}} \cdot \boldsymbol{N}(t) = \left(\bar{\boldsymbol{\lambda}} + \boldsymbol{\phi}(\boldsymbol{t})\bar{\boldsymbol{\sigma}}(\boldsymbol{t})\right)\boldsymbol{N}(t)$$

ORIGEN (the =origen sequence) solves for a single material/mixture; the user is responsible for defining the following data:

- $\boldsymbol{N}(0)$ the initial nuclide vector
- $\overline{\overline{\boldsymbol{A}}}(\boldsymbol{t})$ the transition rate matrix (stepwise constant in time)

which has components:

- $\bar{\boldsymbol{\lambda}}$ the decay matrix (constant in time)
- $\bar{\boldsymbol{\sigma}}(\boldsymbol{t})$ the neutron reaction matrix (stepwise constant in time, changing as the flux spectrum changes)
- $\boldsymbol{\phi}(\boldsymbol{t})$ the flux magnitude (stepwise constant, changing as the flux/power level changes)

Matrix exponential method (MATREX solver)

- The transmutation equation can be written in matrix form (for the homogeneous equation):

$$\dot{\boldsymbol{N}}(t) = \overline{\overline{\boldsymbol{A}}} \cdot \boldsymbol{N}(t) \qquad \text{where } \dot{\boldsymbol{N}}(t) = \frac{d\boldsymbol{N}}{dt}$$

- Solution for the nuclide vector has the form

$$\boldsymbol{N}(t) = \mathbf{N}(0) \cdot \exp\left[\overline{\overline{\boldsymbol{A}}}\, t\right]$$

where

$\mathbf{N}(0)$ = vector of initial atom densities

$\overline{\overline{\boldsymbol{A}}}$ = transition matrix containing rate coefficients (s^{-1}) for radioactive decay, neutron capture, fission, etc.

$$\overline{\overline{\boldsymbol{A}}} = \boldsymbol{\lambda} + \overline{\Phi}\boldsymbol{\sigma}$$

Matrix exponential solution (MATREX solver)

- Solves the coupled differential equations that result from tracking the matrix of > 2,200 nuclides
- Method: extremely fast for large systems – it solves the matrix in ~10^{-2} seconds
- Requirements
  - Initial nuclide concentrations (user input)
  - Neutron flux (user input, derived from input power)
  - Irradiation time (user input)
  - Nuclear decay data (decay constants λ, etc.)
  - Neutron cross sections (probabilities of interaction)

Hybrid matrix exponential/linear chains (MATREX solver)

- Exponential function $\exp\left[\bar{\bar{A}}\, t\right]$ is represented as the series expansion

$$e^{\left[\bar{\bar{A}}\, t\right]} = \sum_{m=0}^{\infty} \frac{\left(\bar{\bar{A}}\, t\right)^m}{m!}$$

- Short lived transitions are difficult to solve due to round off

  e.g. $e^{-100} = 1 - 100 + 100^2/2! - 100^3/3! + \ldots = 3.72 \times 10^{-44}$

- To prevent loss of numerical accuracy, short-lived nuclides are removed from the matrix, matrix elements are adjusted, and matrix is compressed
- Removed nuclides are solved outside the matrix exponential, using Bateman chains solved by a Gauss-Seidel iterative technique

$$N_i = N_i(0)e^{-d_i t} + \sum_{k=1}^{i-1} N_k(0) \prod_{n=k}^{i-1} \left(\frac{a_{n+1,n}}{d_n}\right) \left[\sum_{j=k}^{i-1} d_j \left(\frac{e^{-d_j t} - e^{-d_i t}}{d_i - d_j}\right)\right] \prod_{\substack{n=k \\ n\neq j}}^{i-1} \left(\frac{d_n}{d_n - d_j}\right)$$

Chebyshev rational approximation method (CRAM solver)

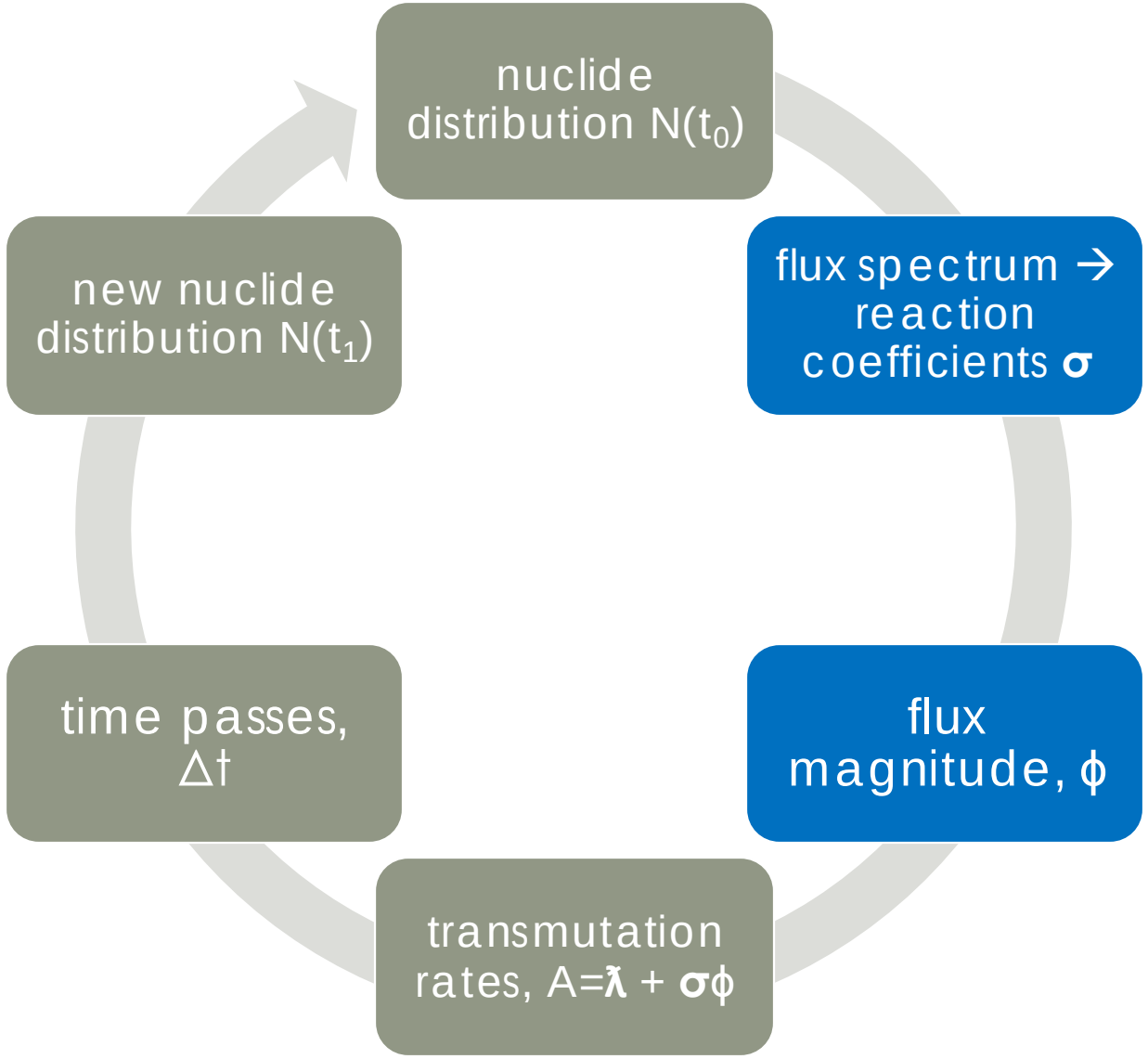
- Matrix exponential method based on Chebyshev rational approximation of the exponential function

$$e^{\overline{\overline{\boldsymbol{A}}}\,t} = a_0 + 2\,\mathbf{Re}\left[\sum_{i=1}^{k/2} a_i(\overline{\overline{\boldsymbol{A}}}\,t + \theta_i I)^{-1}\right]$$

- Fast and accurate, able to handle large systems of nuclides
- Overall accuracy almost independent of step lengths
- Adjoint solution implemented (*Annals of Nuclear Energy* 85, p.68, 2015)

Key to accurate simulations is accurate input data

- Little numerical error in the solvers
- Decay data λ are well known and constant in time
- **Reaction coefficient data σ** depends on:
  - Nuclear cross section and yield data
  - Fuel type
  - Enrichment
  - Burnup
  - Assembly design
  - Fuel temperatures
  - Moderator properties
  - Control rod/blade exposure, etc.
- Flux magnitude data ϕ depend on operating history



On the ORIGEN species



ORIGEN (1973)

- ORIGEN-SANDIA
- ORIGEN-RA (Argonne)
- RATAF (NRC)
- KORIGEN (Karlsruhe)
- ORIGEN-JR (JAEA)

DOE OCRWM

ORIGEN2 (1980)

- ORIGEN2.1
- ORIGEN2.2
- ORIGEN-JENDL (JAEA)
- Monteburns, VESTA
- MCODE-MIT
- MVP-BURN
- MOCUP





NRC
DOE

ORIGEN-S (1980)

- SCALE0
- SCALE1
- SCALE2
- SCALE3, 3.1
- SCALE4, 4.1 - 4.4
- ORIGEN-ARP GUI
- SCALE5, 5.1
- SCALE6, 6.1, 6.2

ORIGEN SCALE 6.3 (2023)

- MCU
- ORIGEN-X-GRS
- OCTOPUS (ECN)
- Monteburns
- MCNP5 (SCK-CEN)
- MCODE

Notes

- DOE halted support for ORIGEN2 in 1989
- DOE and NRC continue to support development of ORIGEN in SCALE

Comparison of ORIGEN branches

	ORIGEN (1982–present)	ORIGEN2 (1980-1992)
Modern nuclear data	✓	✗
Neutron spectra	✓	✗
User-specified gamma and neutron energy groups	✓	✗
Active development	✓	✗
QA configuration control	✓	✗
Validation	Extensive validation	None in over 20 years

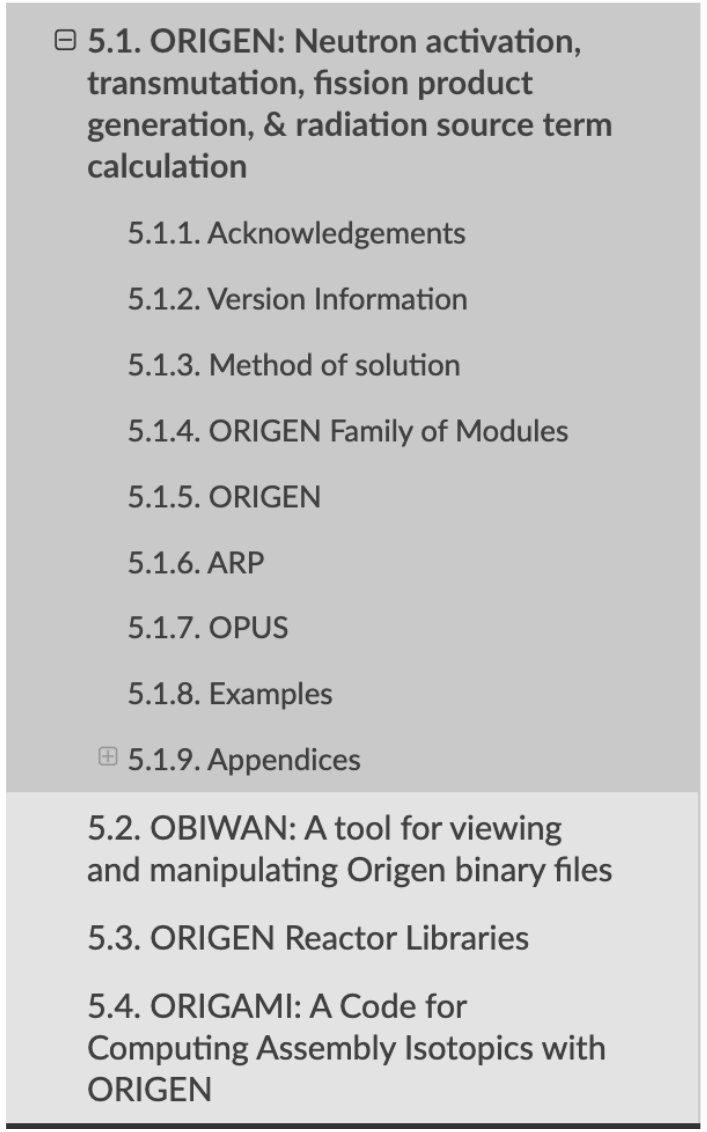
Evolution of ORIGEN capabilities

- SCALE 5.1 (2006)
  - Included graphical user interface, new ORIGEN libraries for different reactor types
- SCALE 6.0 (2009)
  - Included new base cross section libraries (ENDF/B-VI) and updated gamma ray libraries; retained all previous reactor libraries
- SCALE 6.1 (2011)
  - Included ENDF/B-VII decay data, expanded nuclides, energy-dependent fission yields
  - Replaced the old 3-group libraries with 238-group data from JEFF/A-3.0
- SCALE 6.2 (2016)
  - Completely rewritten modular ORIGEN source code, dynamic memory allocations
  - New CRAM solver; rewritten, streamlined input format
  - Integrated alpha and beta sources and spectra
  - ORIGEN API to allow embedding depletion calculation in other codes
- SCALE 6.3 (2023)
  - Modernized library construction integrated into the ORIGEN sequence instead of externally via COUPLE
  - Sensitivity analysis capability
  - OBIWAN command utility for viewing and manipulating ORIGEN binary files

ORIGEN Documentation

- Section 5.1 of the SCALE 6.3 manual
- Examples show some of the commonly used capabilities

https://scale-manual.ornl.gov/origen/origen-examples.html



https://scale-manual.ornl.gov/origen/

References on ORIGEN methods and data

I. C. Gauld, G. Radulescu, G. Ilas, B. D. Murphy, M. L. Williams, D. Wiarda, *Isotopic Depletion and Decay Methods and Analysis Capabilities in SCALE*, Nuclear Technology , 174, 2, 169 (2011).

A. Isotalo and W. Wieselquist, *A Method for Including External Feed in Depletion Calculations with CRAM and Implementation into ORIGEN*, Annals of Nuclear Energy, vol. 85, p.68-77 (2015).

M. T. Pigni, M. W. Francis, and I. C. Gauld, *Investigation of inconsistent ENDF/B-VII.1 independent and cumulative fission product yields with proposed revisions*, Nuclear Data Sheets 123, 231 (2015).

I. C. Gauld, M. T. Pigni, and G. Ilas, *Validation and testing of ENDF/B-VII decay data,* Nuclear Data Sheets 120, p.33-36 (2014).

G. Ilas, J. R. Burns, B. D. Hiscox, and U. Mertyurek, *SCALE 6.2.4 Validation: Reactor Physics*, ORNL/TM-2020/1500v3, UT-Battelle, LLC, Oak Ridge National Laboratory (November 2022). Web, doi:10.2172/1902818.

G. Ilas and B. Hiscox, "*Validation of SCALE 6.2.4 and ENDF/B-VII.1 data libraries for nuclide inventory analysis in PWR used fuel*", Transactions of the American Nuclear Society, vol. 124, p.552-554 (2021). https://www.ans.org/pubs/transactions/article-49653

G. Ilas and J. Burns, "SCALE 6.2.4 Validation for Light Water Reactor Decay Heat Analysis", Nuclear Technology (2021). https://doi.org/10.1080/00295450.2021.1935165

G. Ilas, I. C. Gauld, and H. Liljenfeldt, *Validation of ORIGEN for LWR used fuel decay heat analysis with SCALE*, Nuclear Engineering and Design, vol.273, p. 58-67 (2014)

G. Ilas, I. C. Gauld, and G. Radulescu, *Validation of new depletion capabilities and ENDF/B-VII data libraries in SCALE*, Annals of Nuclear Energy, vol. 46, p.43-55 (2012).

Publicly-available documents can be accessed via OSTI. See https://www.ornl.gov/scale/references Depletion and Decay

OAK RIDGE National Laboratory

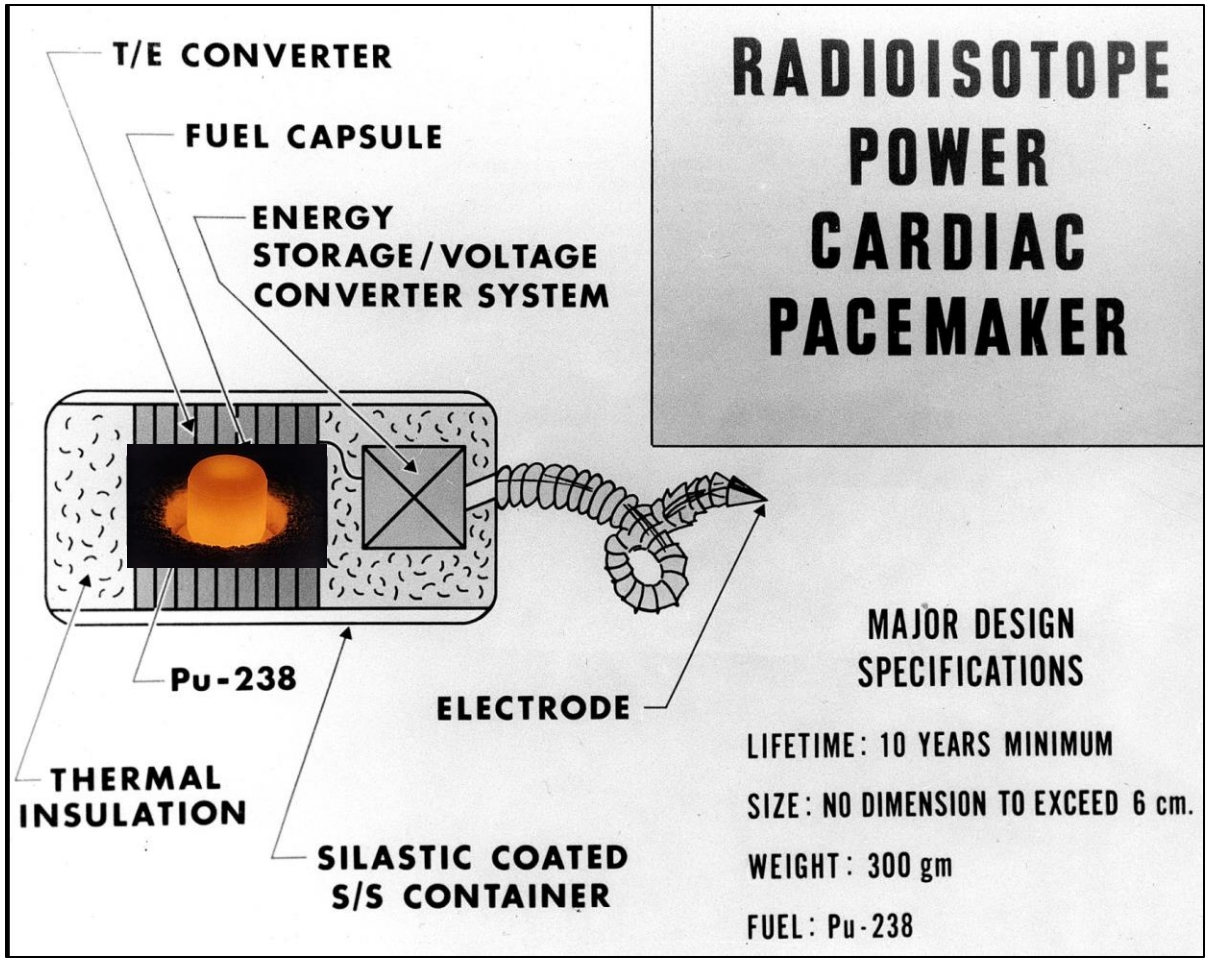
ORIGEN decay calculations

- Hands-on introduction to ORIGEN decay input data and input generation and validation using Fulcrum



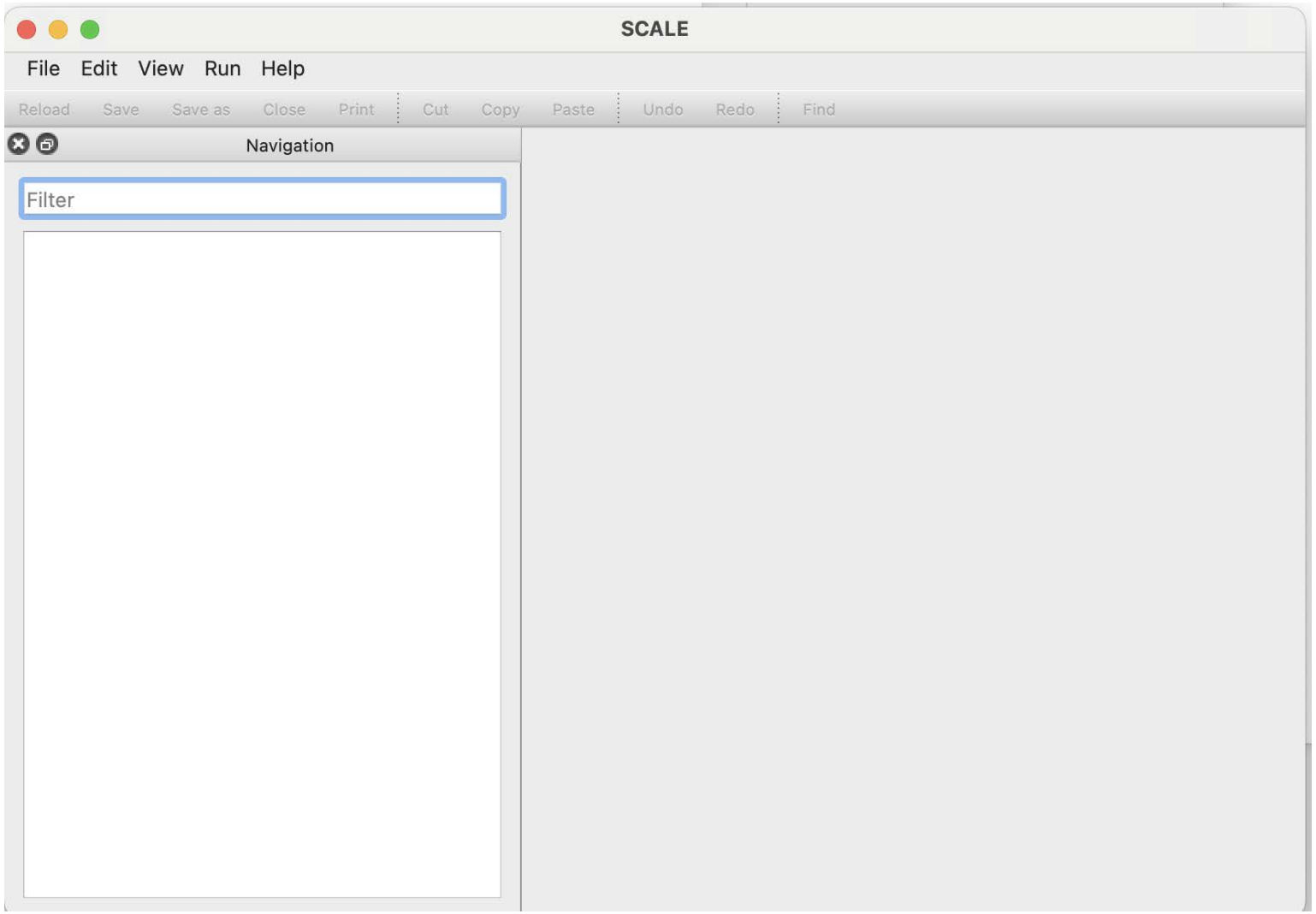
A simple decay heat calculation

- Problem
  - Determine the 100-year decay heat curve for a radioisotope-powered pacemaker powered by 1 mg ^{238}Pu
- Approach
  - Clean slate: start with an empty file and use the autocomplete capability in Fulcrum
  - Name your input "i_heart_pu238.inp"
  - Include save=yes
  - Open associated F71 file in Fulcrum



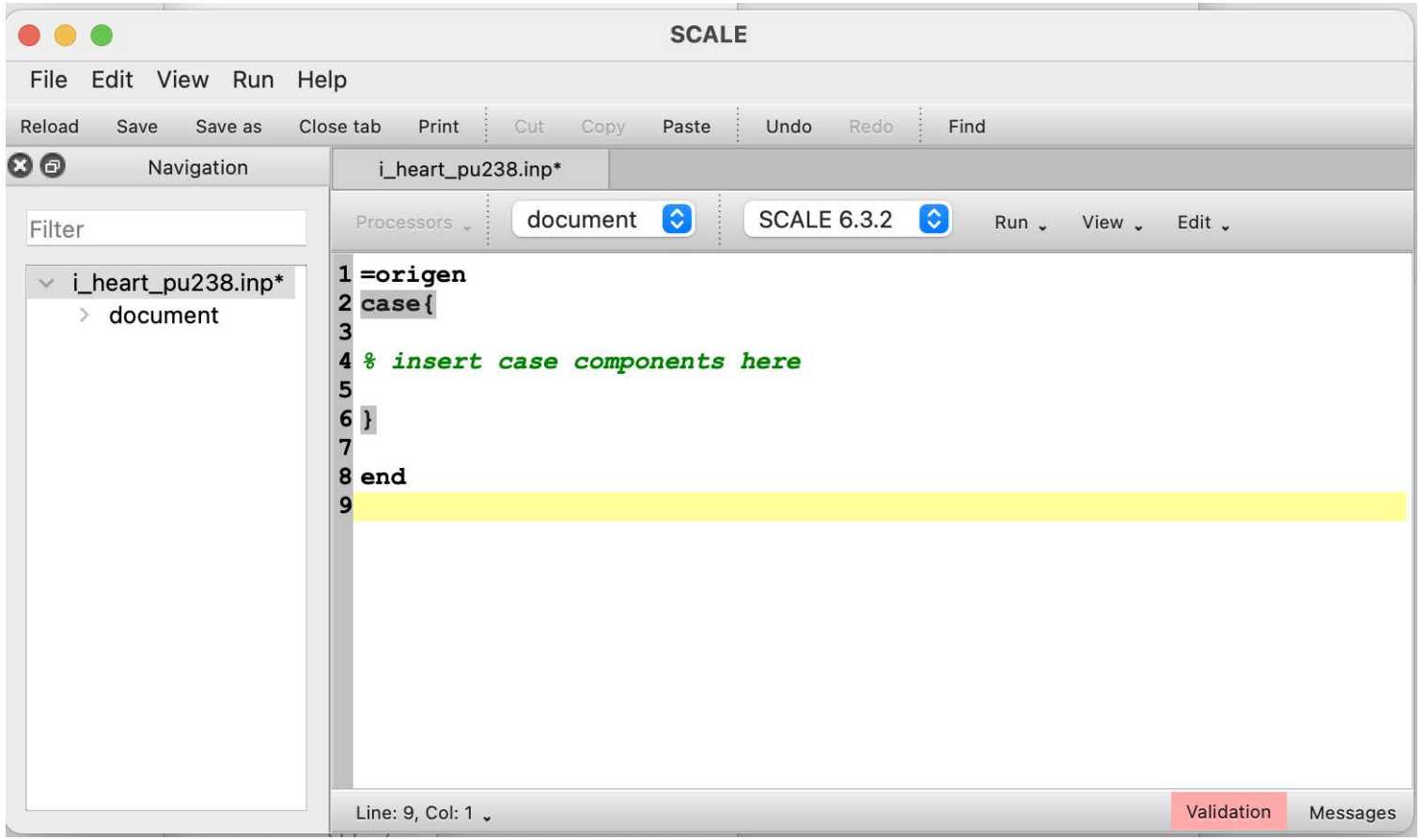
Radioisotope-powered cardiac pacemaker developed by the Atomic Energy Commission, circa 1967.

Simple decay calculation: Open Fulcrum



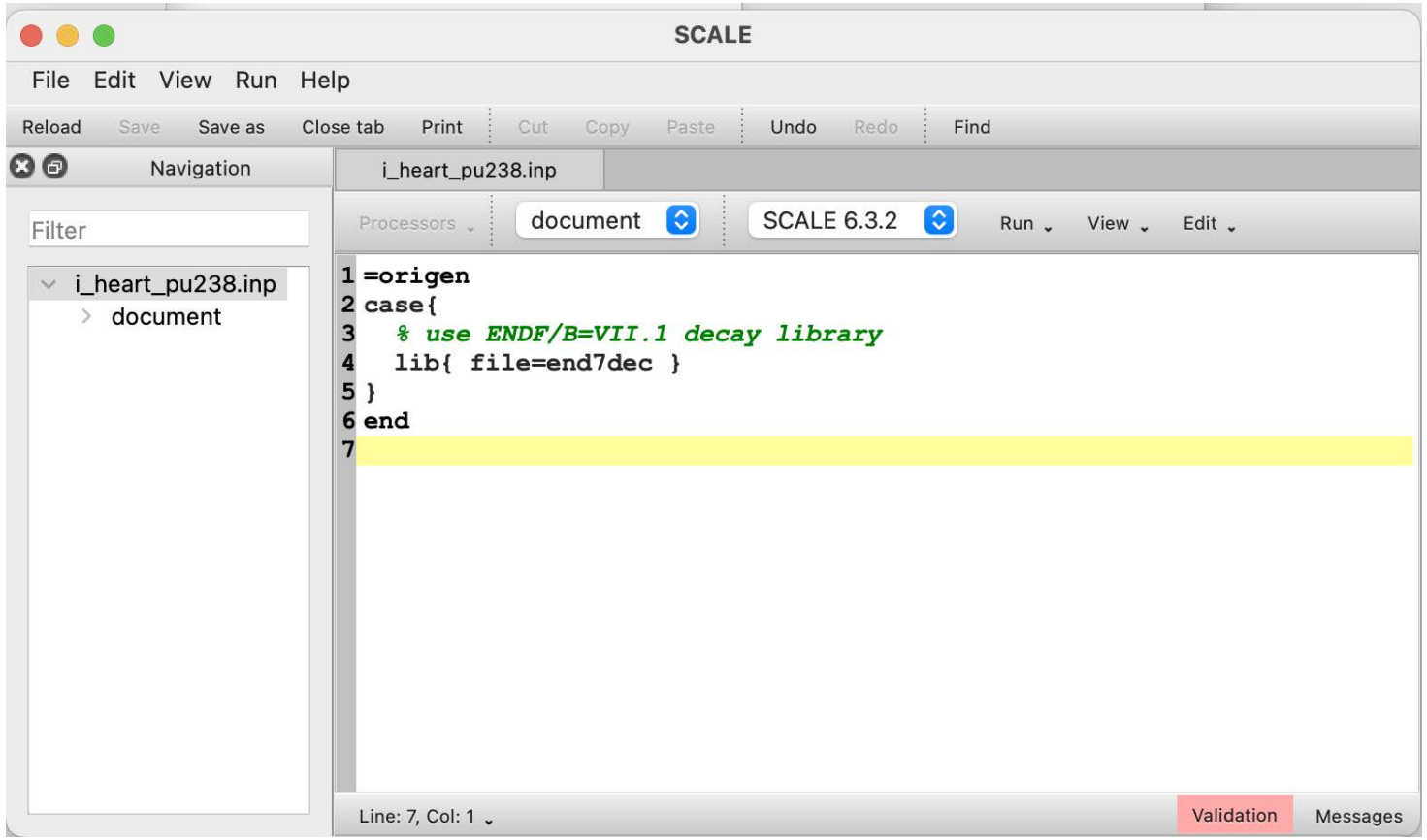
- Shortcut to Fulcrum is usually in the SCALE folder on Desktop (on Windows computer)

Simple decay calculation: Construct a case block



"case" is always the problem "unit" for each ORIGEN calculation

Simple decay calculation: Add a lib block

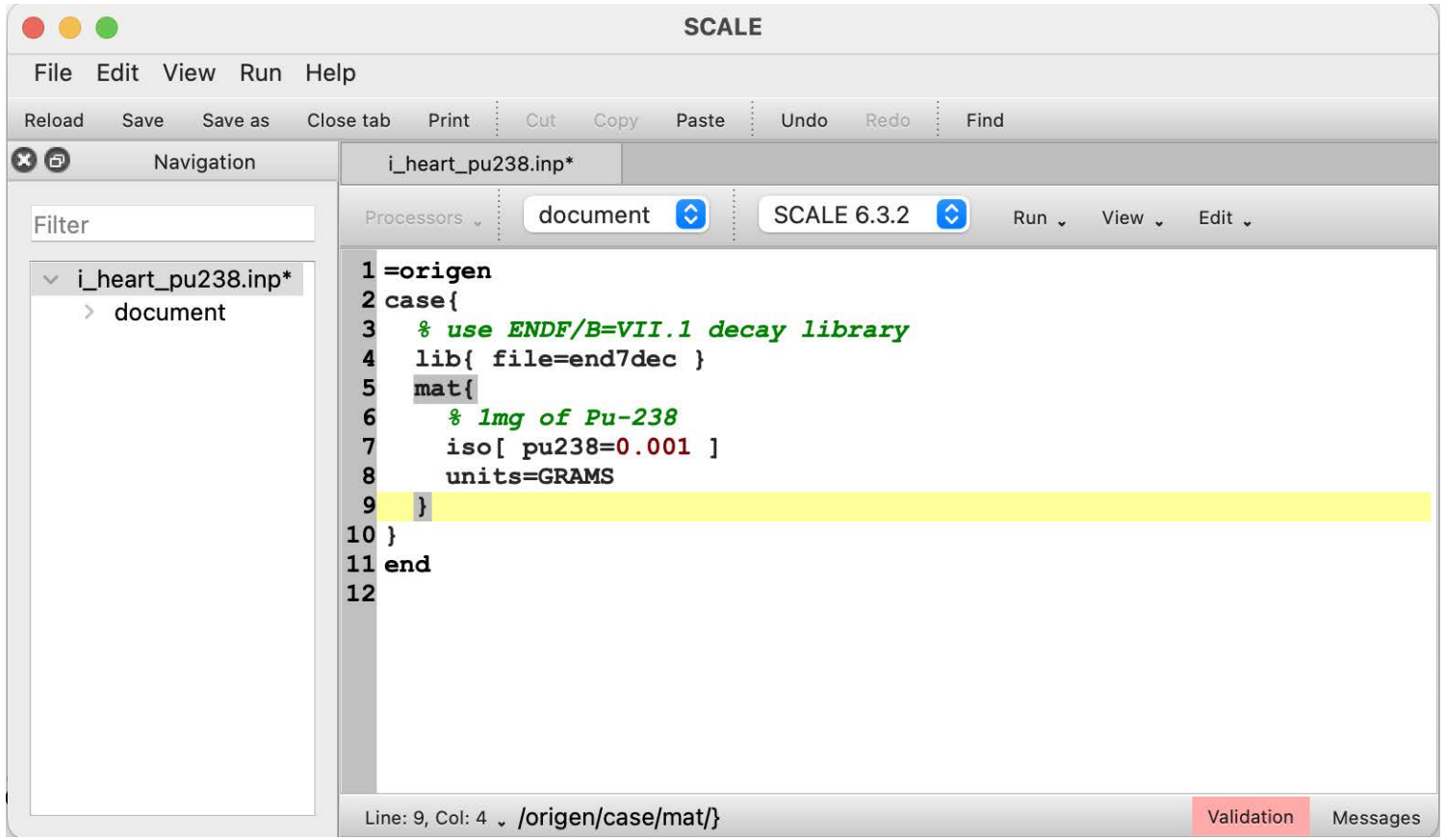


- Syntax is:

  ```
  lib{ file=filename
  pos=file_position }
  ```

- Use the special "end7dec" library, which is a decay-only library

Simple decay calculation: Add a mat block for ^{238}Pu

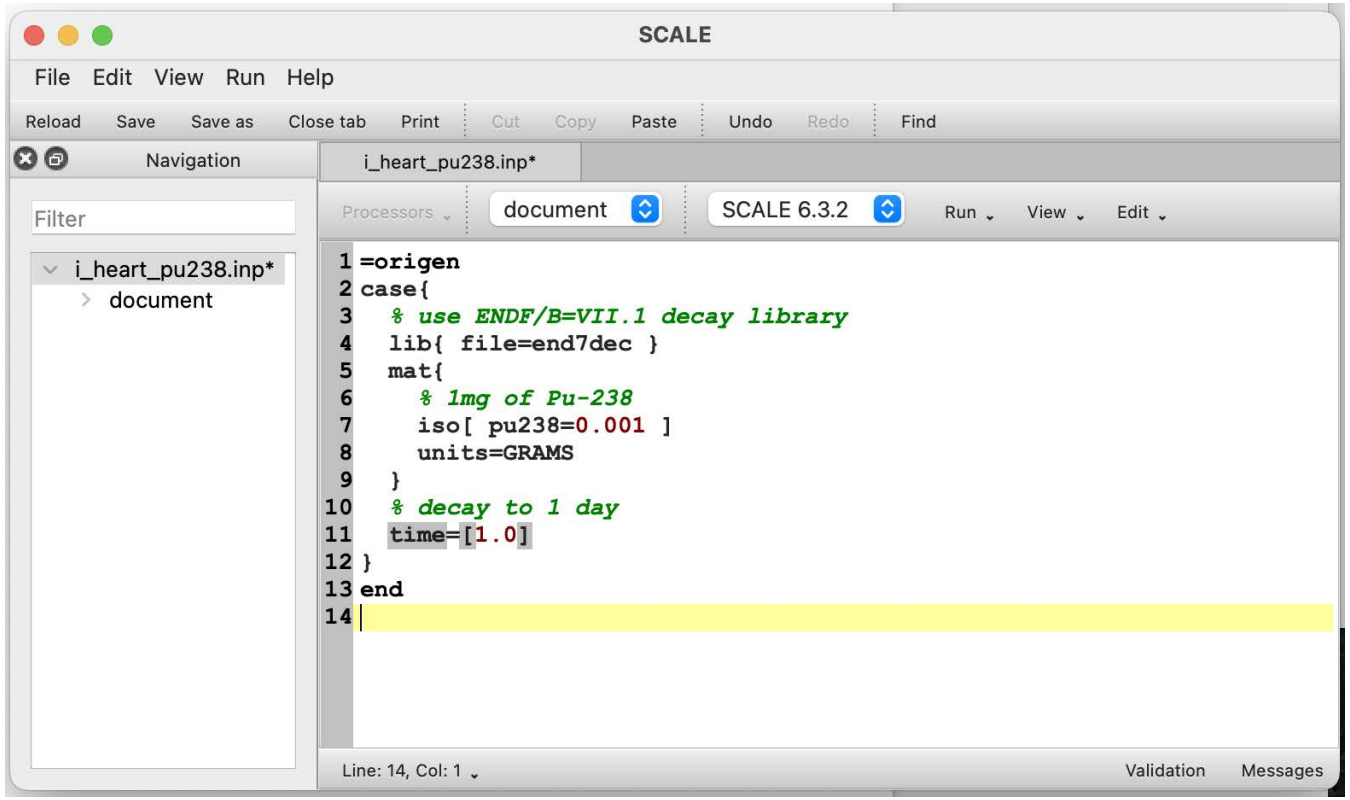


Material block has two components:

- **iso=[list of key/value pairs]**
- **units=** Concentration units for values in iso array (e.g., GRAMS)
- Default concentration units are MOLES

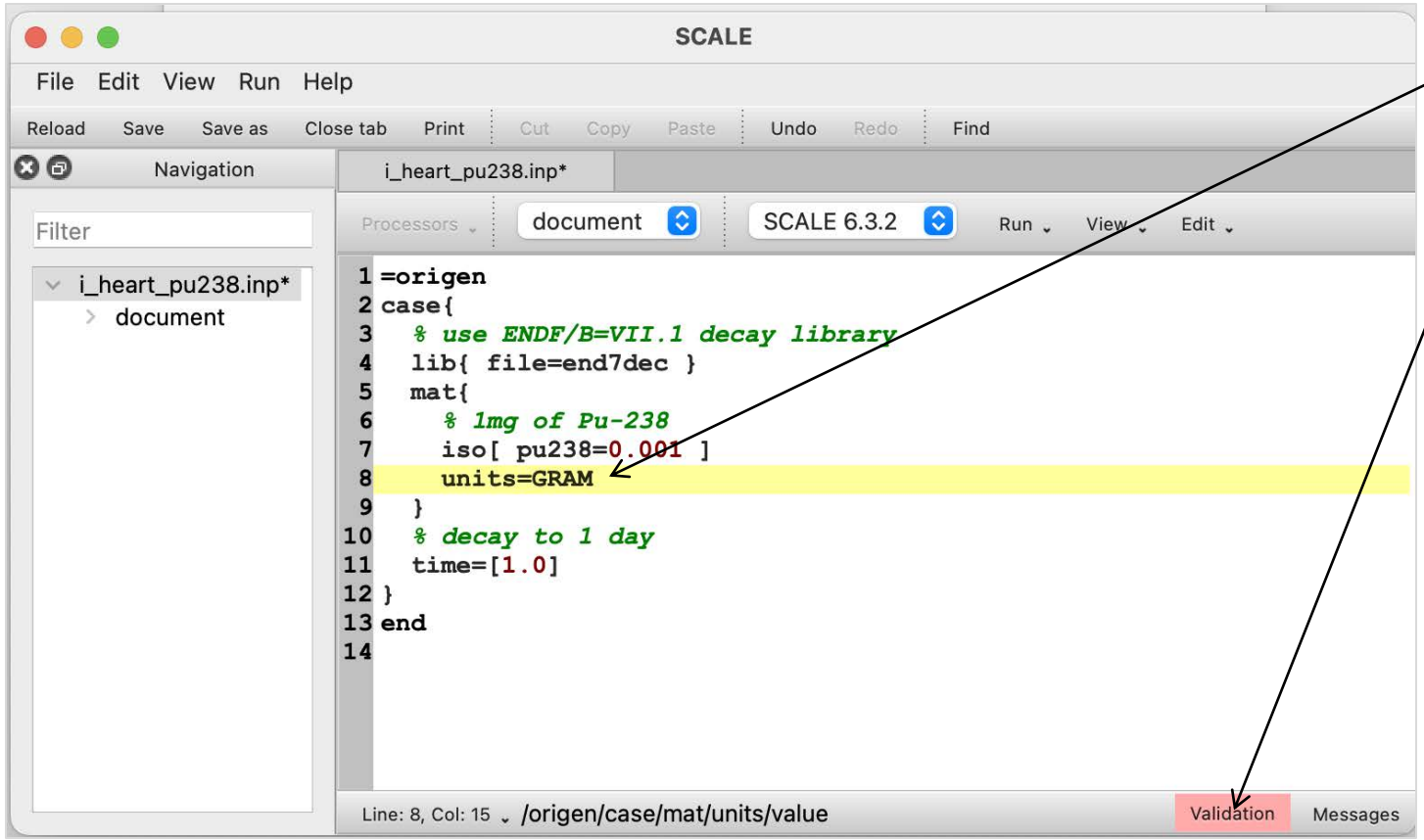
Add 1 mg (0.001) of Pu-238

Simple decay calculation: Add a time block



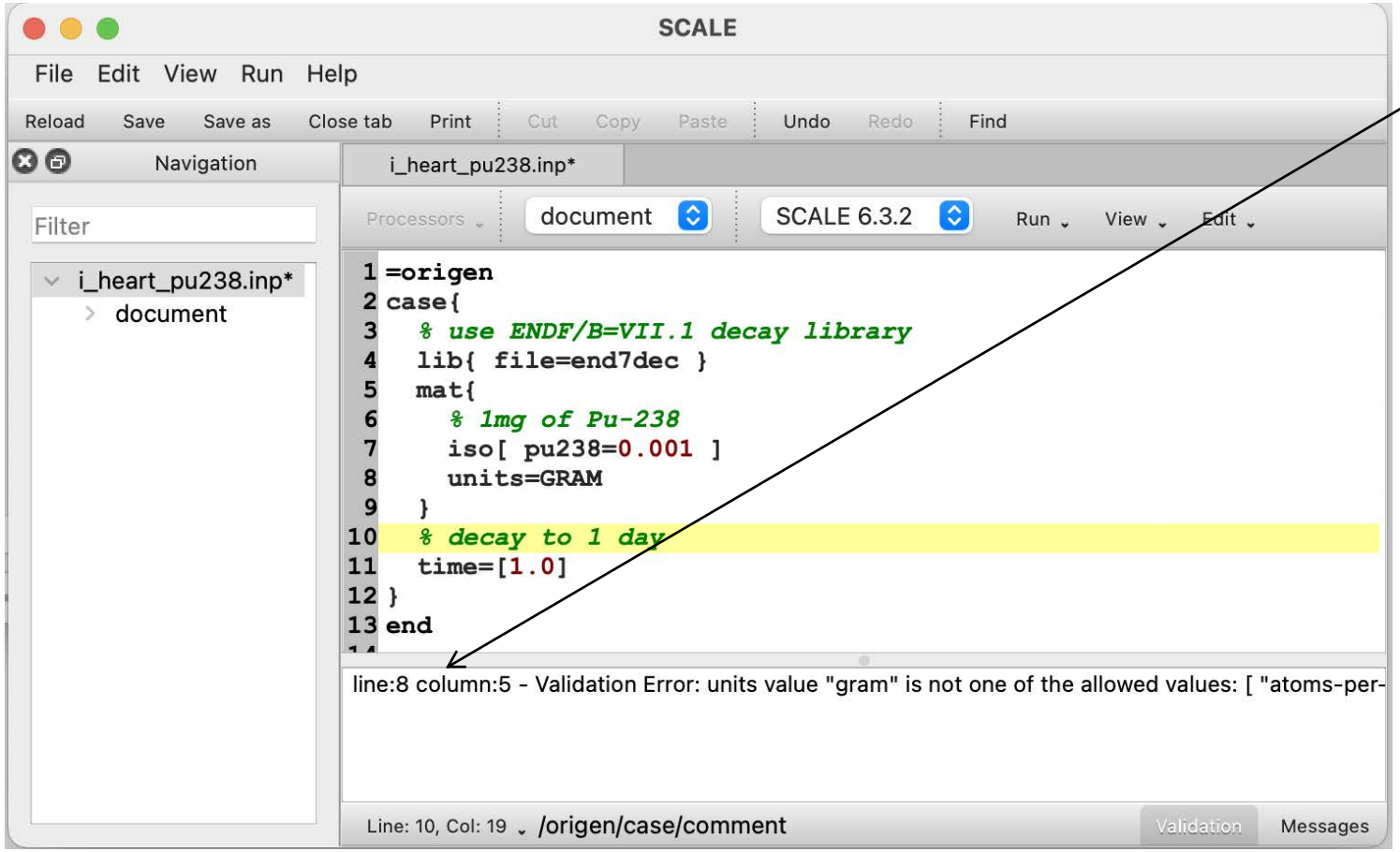
- For now, we'll just do 1 day
- Can use shortcut notation
  t=[1.0]
  (default units are days)
- We'll go back and add the full time in a moment
- Our problem is ready to run!

Simple decay calculation: Validate the ORIGEN input



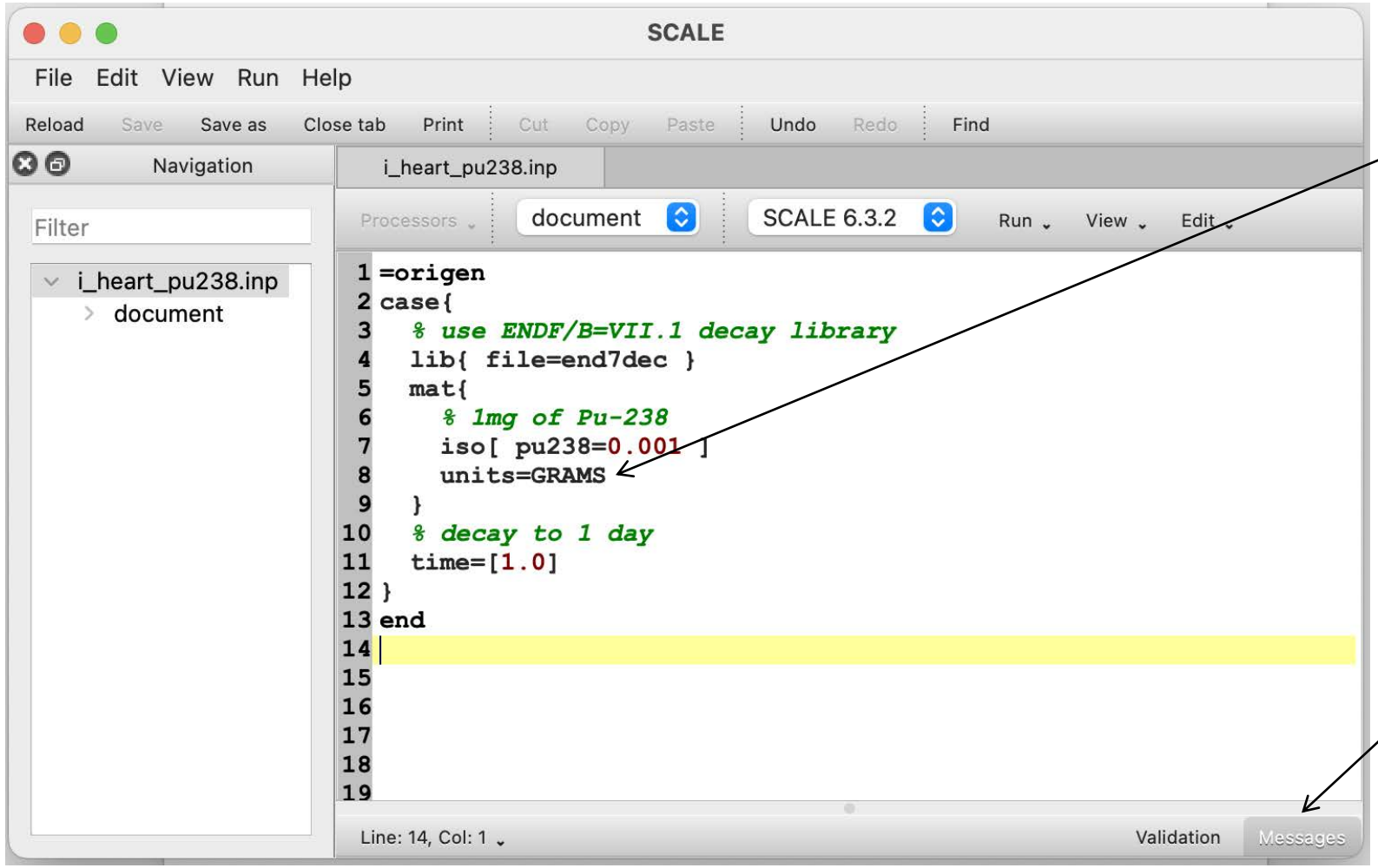
- Remove "S" from "GRAMS"
- Notice Validation button in bottom right (red shade)
- Click on button to get validation error details

Simple decay calculation: Find validation errors



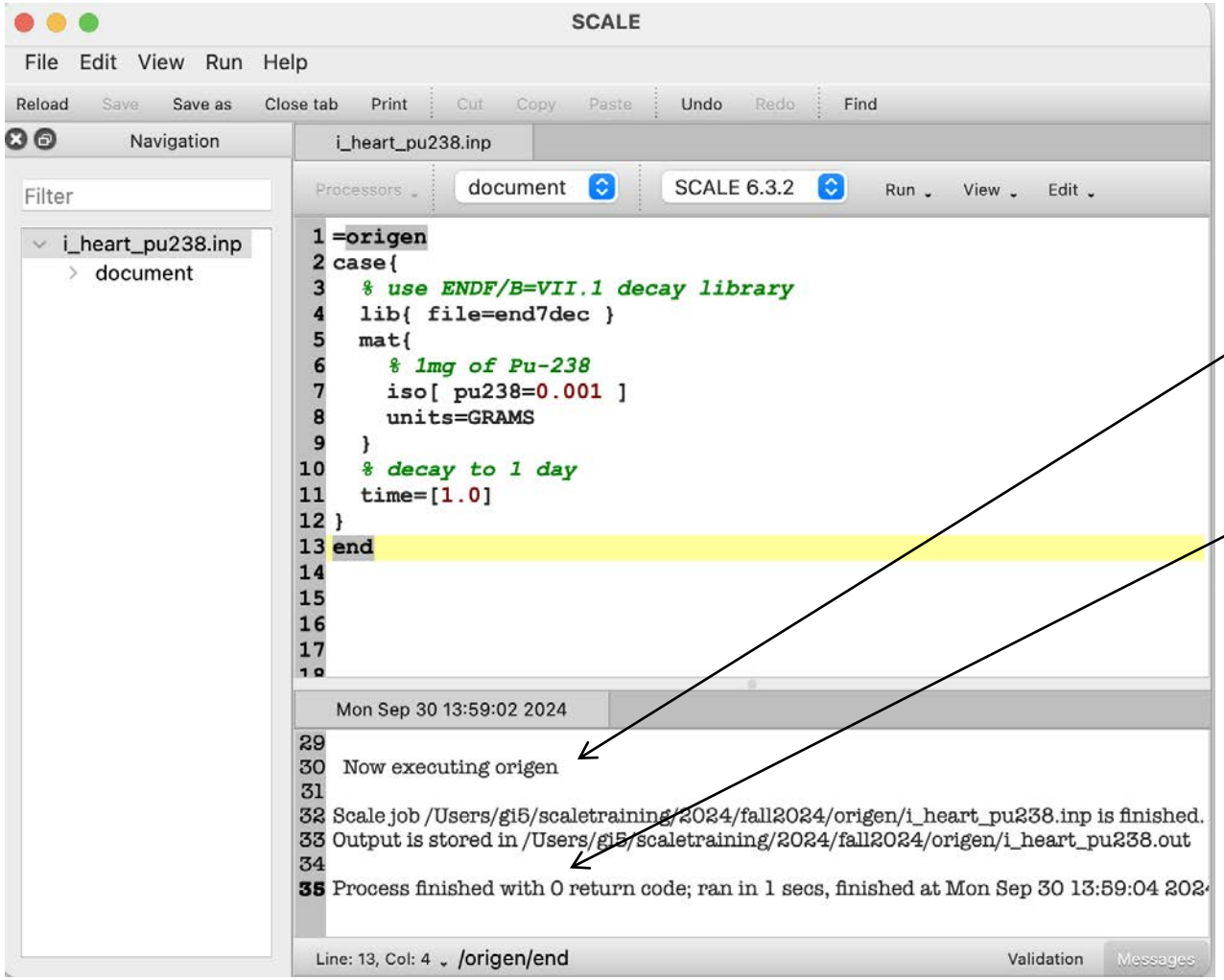
- Validation box shows LINE:COLUMN and then a message
- Clicking the message moves the cursor there

Simple decay calculation: Run calculation



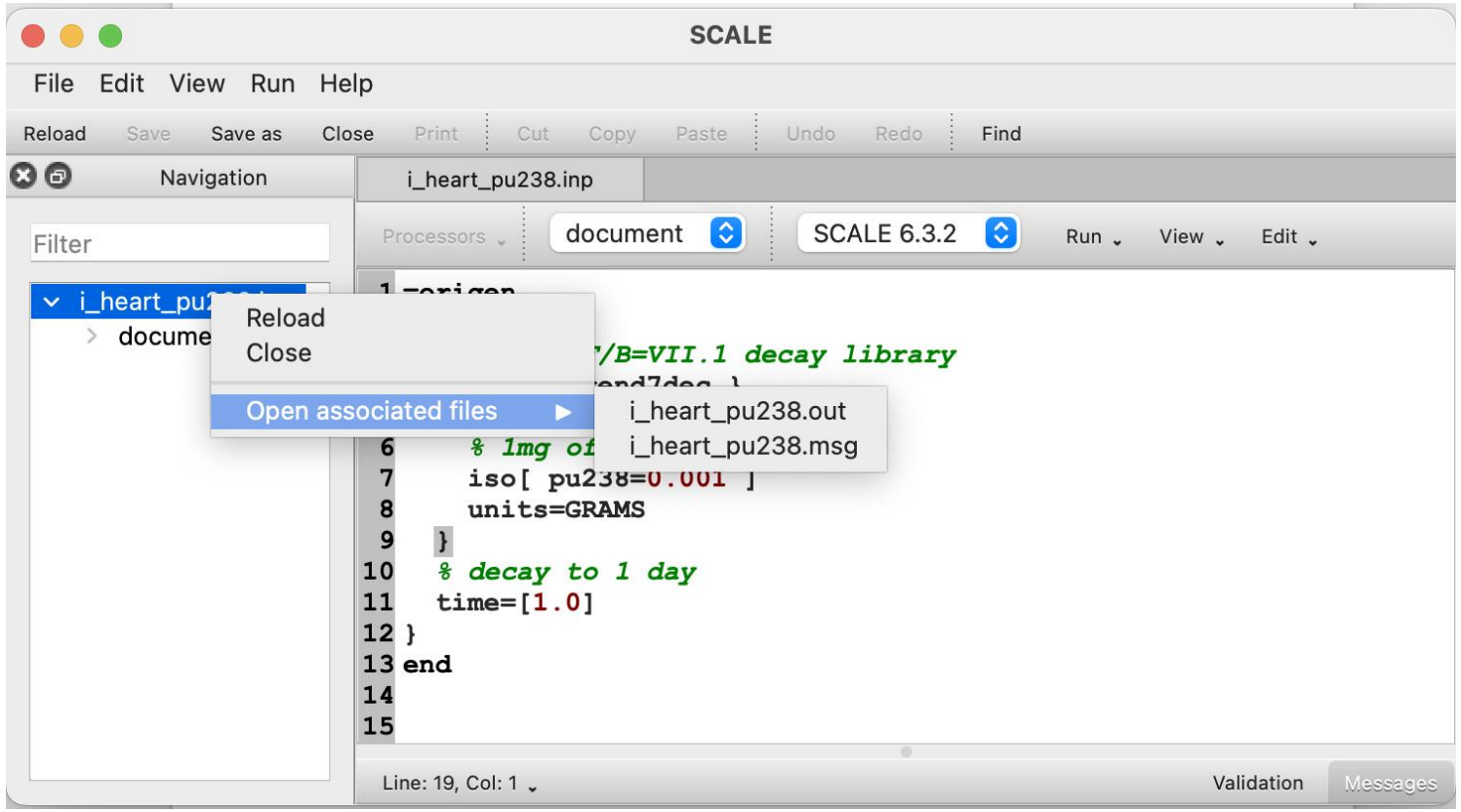
- Correct "GRAM" to GRAMS" and save file by click on "Save"
- Click the "Run" button
- Notice shaded Messages button
- Click Messages to see code running

Simple decay calculation: Examine messages



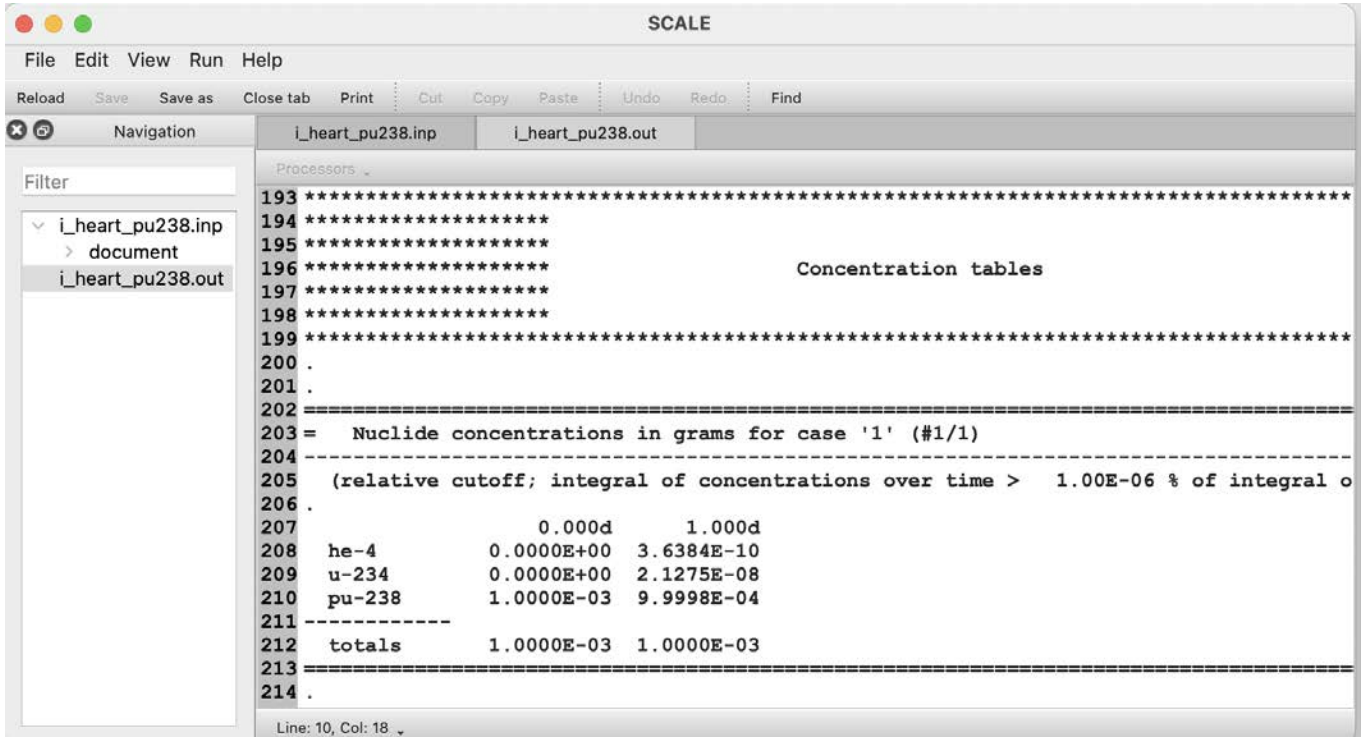
- See sequences executed
- A "0 return code" means success

Simple decay calculation: Open output file



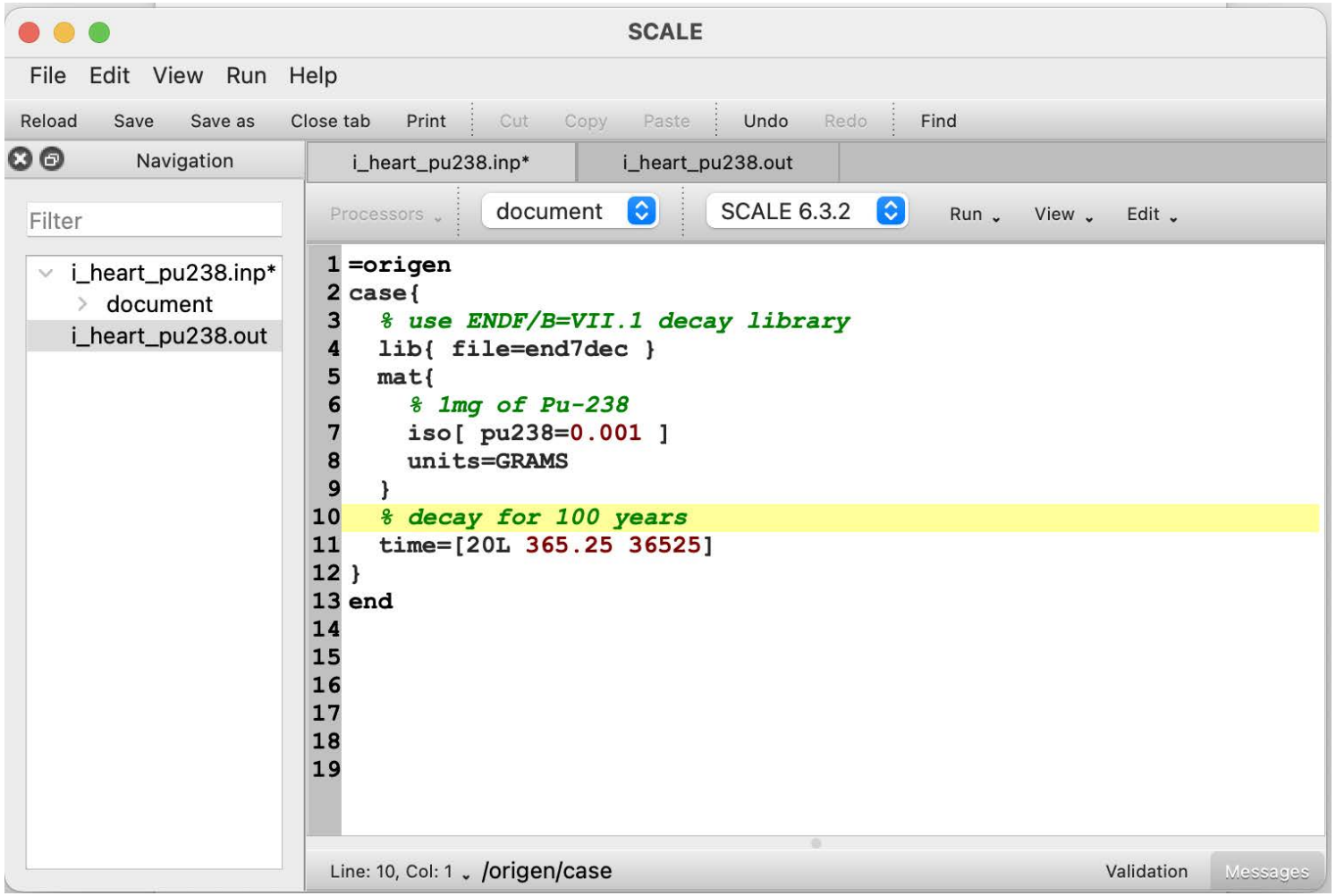
- Easiest way is to right-click on file name in the navigation pane and "Open associated files" → i_heart_pu238.out

Simple decay calculation: Examine output file



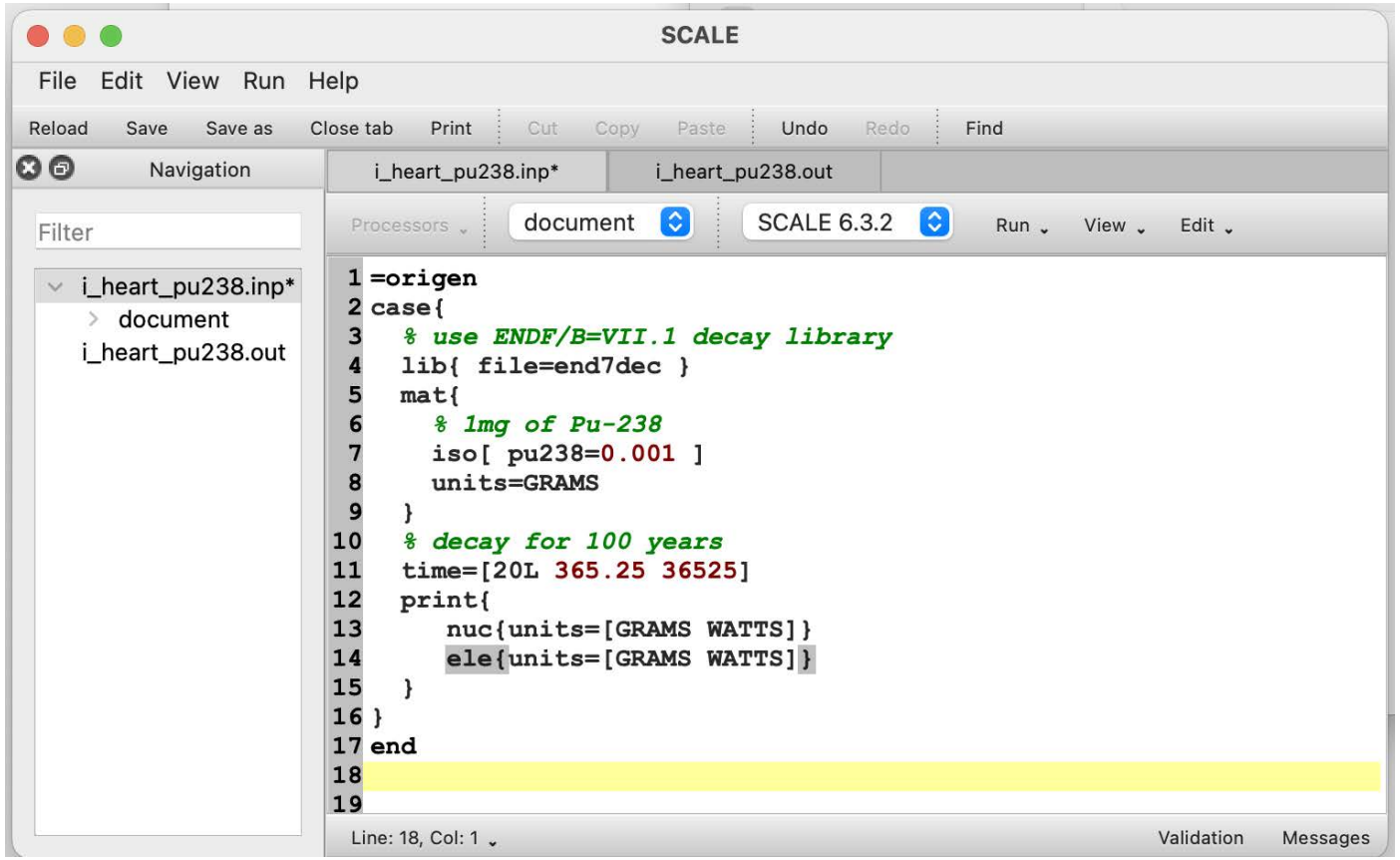
- "Concentration tables" are printed for each case
- By default, only grams are printed
- Nothing really happens to ^{238}Pu in a day…
  - Half-life is 87.7 years

Simple decay calculation: Add decay time to 100 years



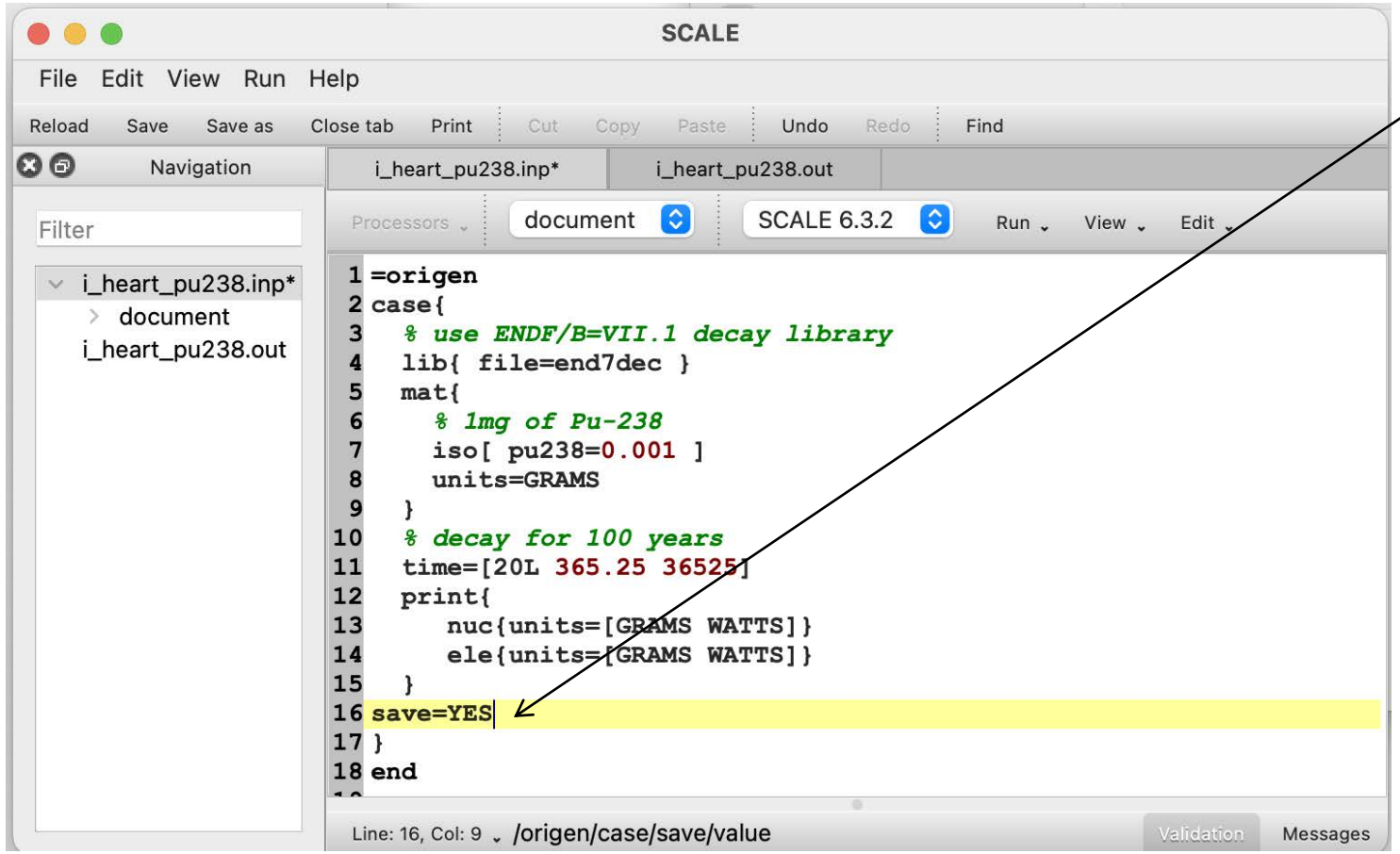
- 100 years is 36,525 days
  - Keeping it in days now for simplicity
- Input shortcut:
  - 20L = 20 logarithmically-spaced intervals between 365.25 and 36,525 days

Simple decay calculations:
Examine activity and elemental breakdowns



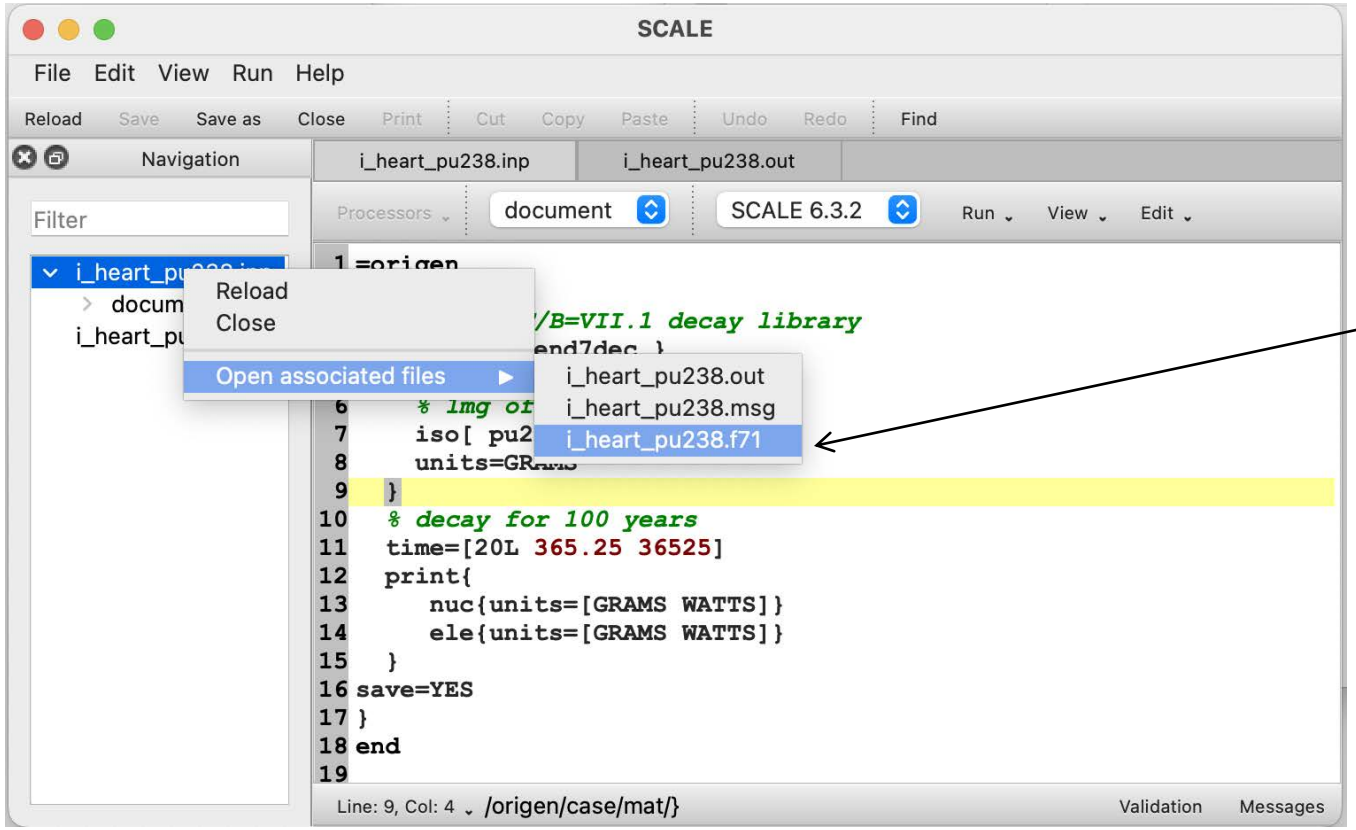
- Use the **`print`** block
- **`print`** block contains nuclide (**`nuc`**) and element (**`ele`**) blocks
- Blocks on the left print activity/mass by nuclide and element
- Add printout of decay heat (WATTS)

Simple decay calculation:
Save ORIGEN binary concentration file (f71)



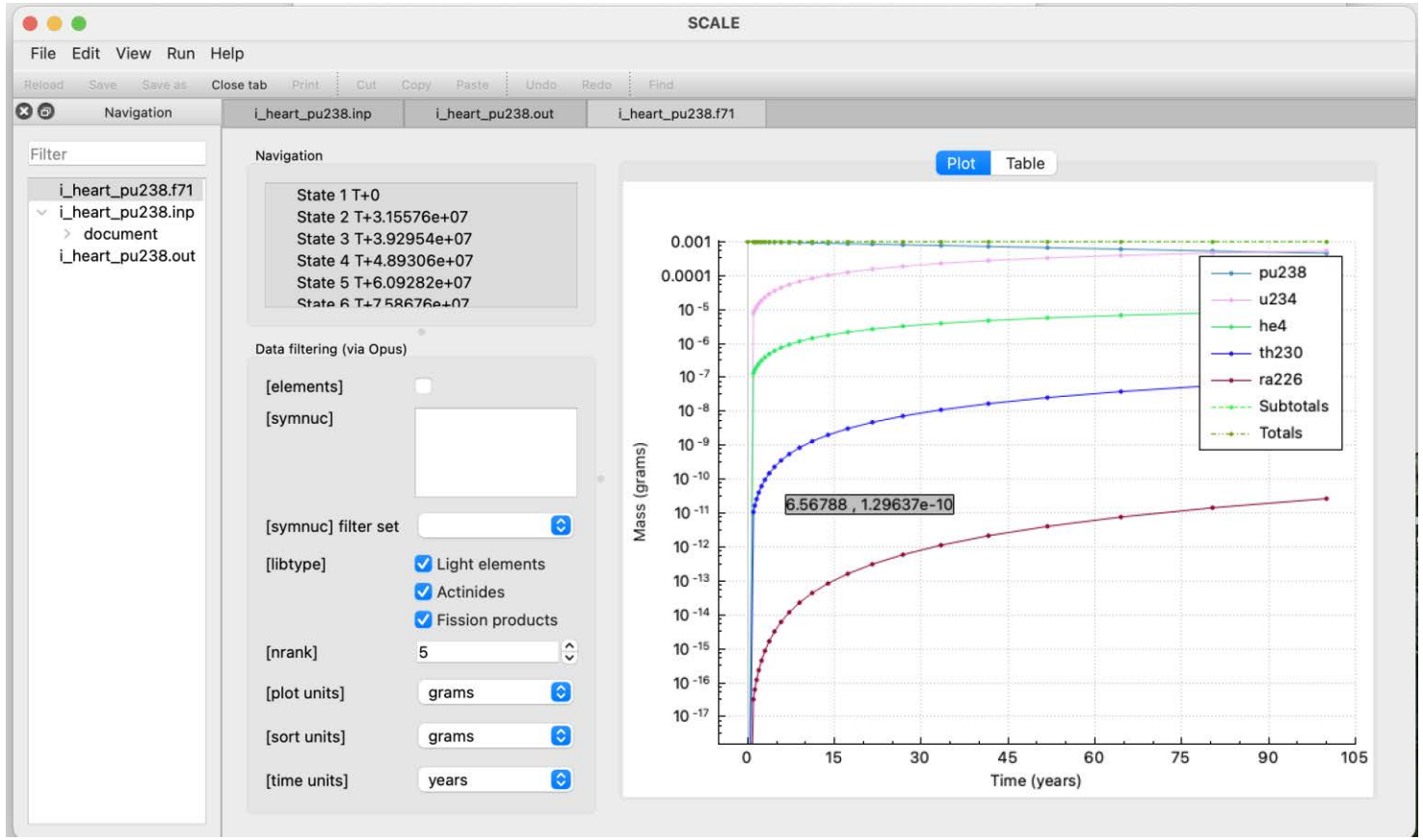
- Use the **save=yes** to generate binary concentration file
- The **save** block gives more control over the exact data to output
- Run input

Simple decay calculations: Open f71 file



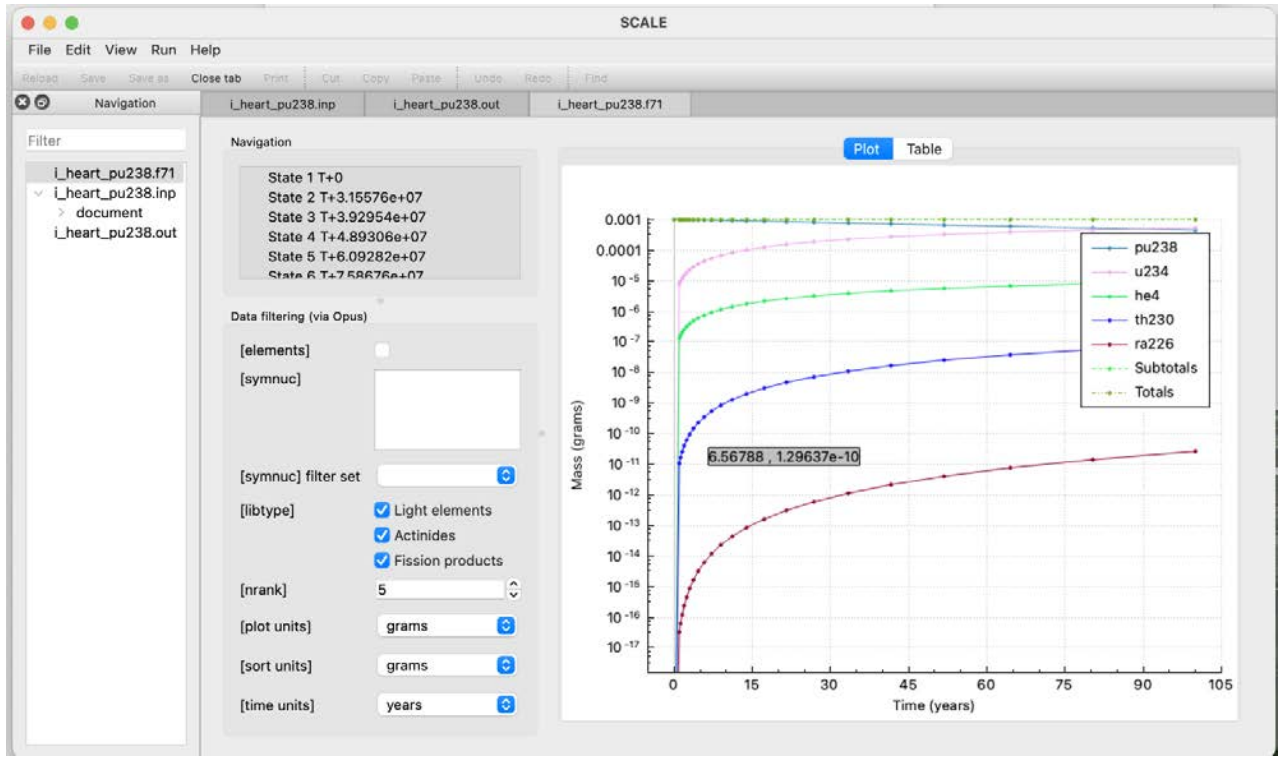
- Right-click on i_heart_pu238.inp
- Open associated files →"i_heart_pu238.f71"

Simple decay calculation:
Use f71 viewer to view concentration plot (in grams)

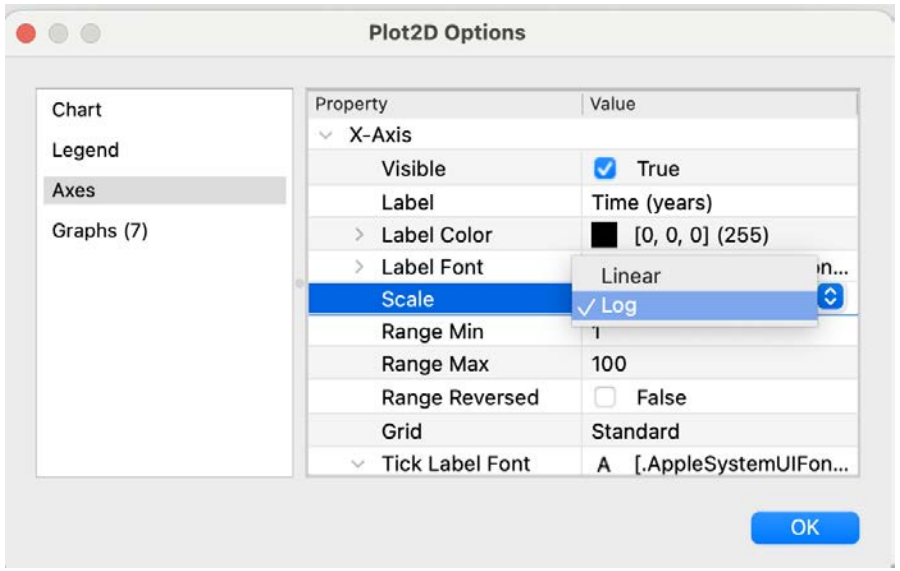


- First select all states (should be 23) any of the following ways
  - Left-click and drag down
  - Hold SHIFT and select range
  - CTRL-A
- Change [plot units] and [sort units] to "grams"

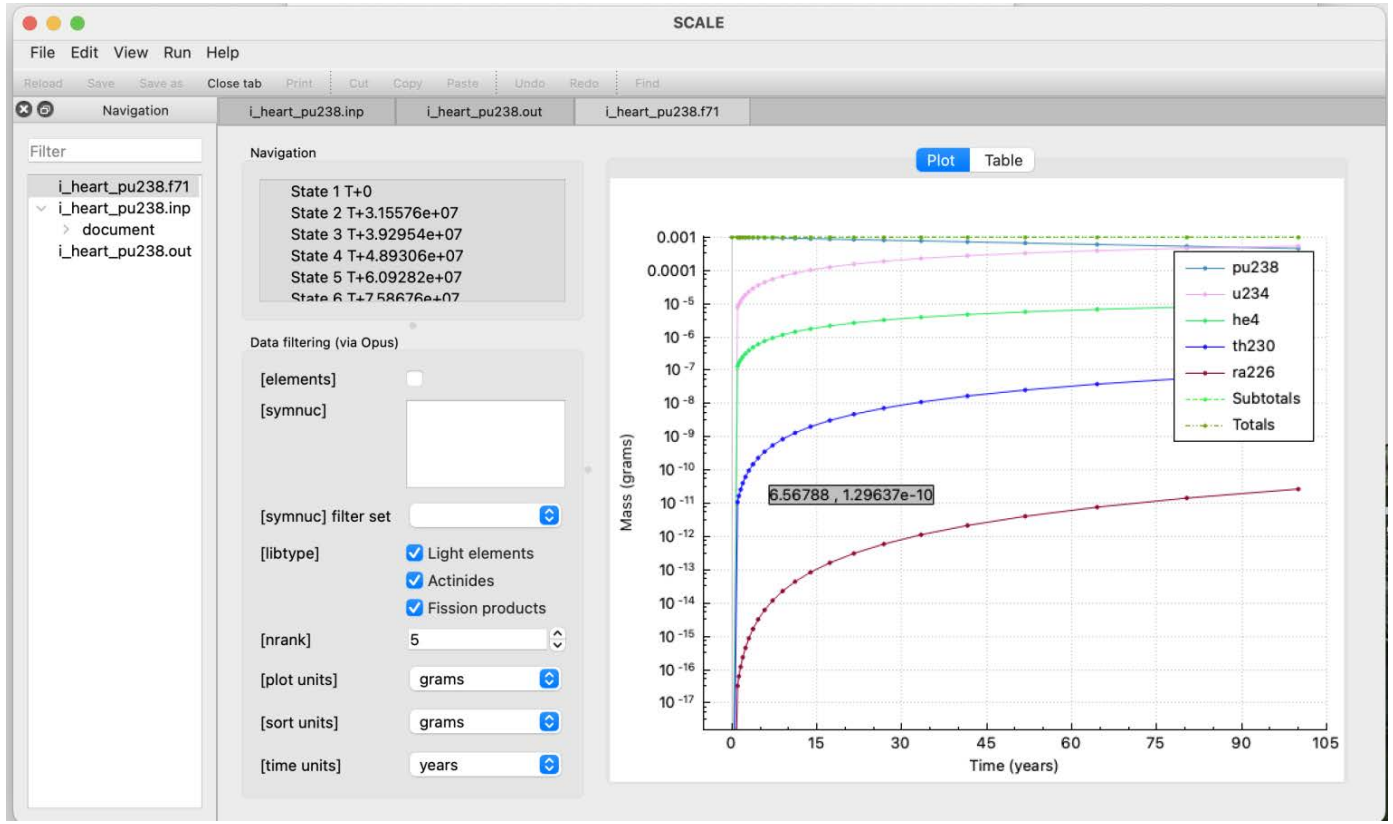
Simple decay calculation:
Can change x-axis time scale to Log or Linear



- Right click anywhere on plot and select "Plot Options"
- Navigate to "Axes→X-Axis→Scale" and change to "Log" or "Linear"

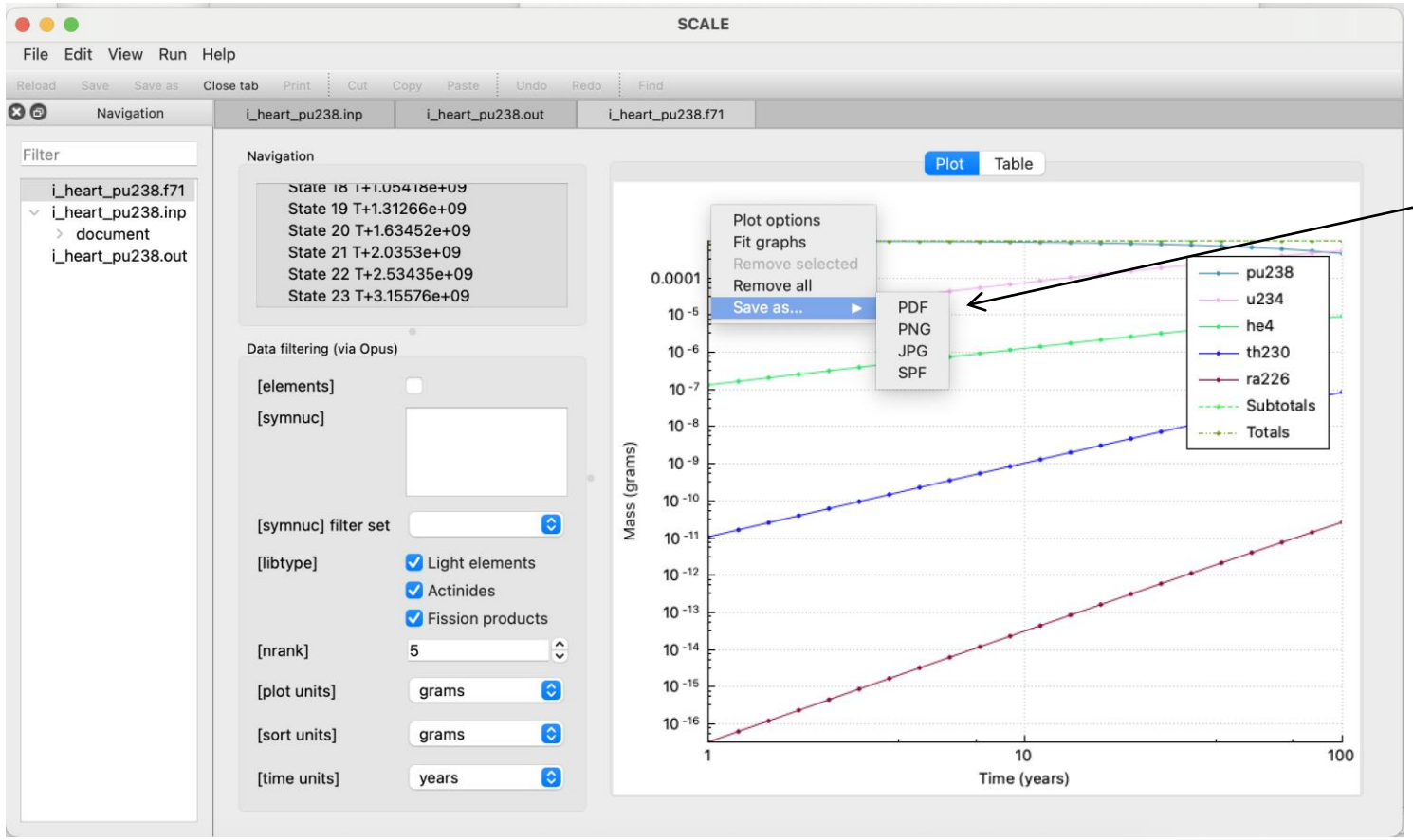


Simple decay calculation:
Experiment with different filtering options



- **`nrank`** is the number of things to show
- **`sort units`** is the units to use in the ranking
- **`plot units`** is the units to show on the plot
- **`elements`** toggles whether elements or nuclides are shown
- The plot to the left shows the concentration in grams of the top 5 nuclides or elements

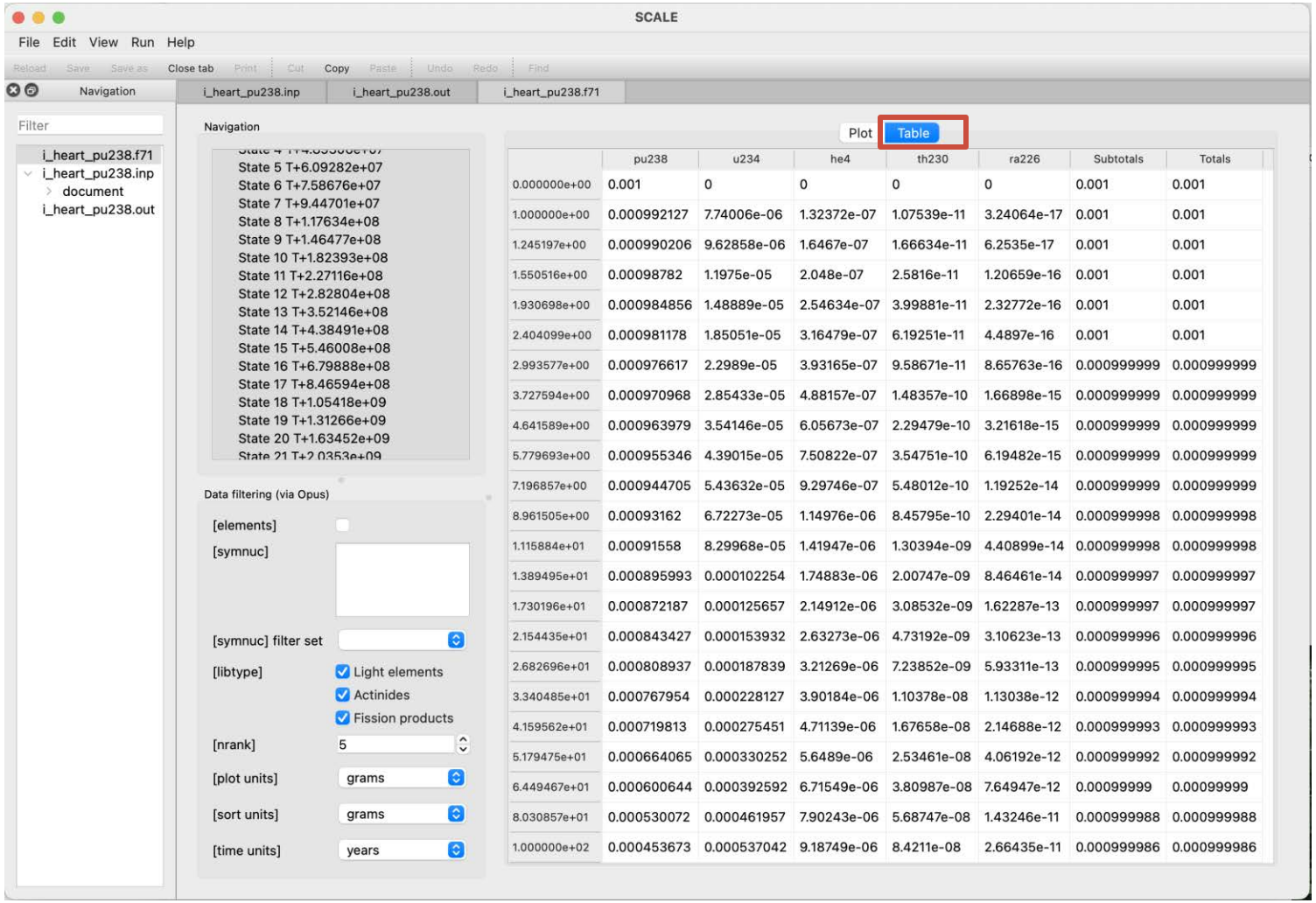
Simple decay calculation: Export plot as image



- Right-click on plot and select **"Save as..."** and choose file type
- PDF option uses vector graphics—good choice for embedding in a report

Simple decay calculation:

Export table for use into another program



- Click on the Table Tab
- Click on the top left square to highlight all data
- CTRL-C to copy contents
- Can open the other program (e.g., MS Excel) and CTRL-V to paste contents

Solution

```
=origen
case{
   lib{ file=end7dec }
   mat{
      iso[ pu238=1e-3 ]
      units=GRAMS
   }
   time=[ 20L 1 36525 ]
   print{ nuc{ units=[grams watts] } }
   save=yes
}
end
```

Simple decay calculation: Summary

- Input file structure is simple for simple decay calculations
  - one ORIGEN case block
  - use "end7dec" in lib block
  - short material specification (mat block), e.g.
    - mat{units=grams iso[u238=1.0]} % single nuclide
    - mat{units=moles iso[k=1.0]} % single element
  - rest is printing in .out file, f71 save options
- Logarithmic interpolation is convenient to create time array, e.g.

  time=[5L 1 10]
- For long decay times, make sure the time "grid" is refined sufficiently
  e.g., try both **time=[5L 1 10]**

  and

  time=[10L 1 10]

  and see if your results change significantly

OAK RIDGE National Laboratory

Post-processing ORIGEN F71 files

With OPUS

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

OPUS features

- OPUS: ORIGEN-S Post-processing Utility for SCALE
- Quickly and easily extract requested information from an ORIGEN calculation
- Rank most dominant nuclides or elements
- Request specific nuclides or elements
- Include totals for different nuclide groups, such as actinides, fission products, and/or activation products
- Gamma, alpha, beta, and neutron spectra
- Text output data file can be read directly for plotting
- Operate on .f71 concentration file saved by ORIGEN (not the output file)

OPUS options (plot setup)

DATA=
(DEFAULT: "ft71f001")
ORIGEN concentrations file (f71)

TYPARAMS=
type of parameter to be plotted

- NUCLIDES — individual nuclides
- ELEMENTS — individual elements
- ASPECTRUM — alpha spectrum
- BSPECTRUM — beta spectrum
- GSPECTRUM — gamma spectrum
- NSPECTRUM — total neutron spectrum
- SFSPEC — spontaneous fission neutron spectrum
- ANSPEC — neutron spectrum from (alpha,n) reactions only
- DNSPEC — delayed neutron spectrum

VOLUME=
material volume (cm^3), when using units of g/cm^3 or at/barn-cm

TIME=
unit of time to be plotted

- SECONDS
- MINUTES
- HOURS
- DAYS
- YEARS

More details:

See Section 5.1.7 of the SCALE manual

OPUS Options (response units and nuclide selection)

UNITS=

GRAMS	mass in grams
GATOMS	gram atoms (default unit) or moles
KILOGRAMS	mass in kilograms
ATOMS	atoms/barn-cm (requires VOLUME)
WPELEM	weight % of element (specified in SYMNUC)
GPERCM	grams/cm^3 (requires VOLUME)
CURIES	radioactivity in Ci
BECQUERELS	radioactivity in Bq
INTENSITY	spectra in particles/s/MeV
PARTICLES	spectra in particles/s
WATTS	total decay heat (W)
GAMWATTS	gamma-ray decay heat (W)
ABSORPTIONS	absorption/removal reaction rate
CAPTURES	capture reaction rate
FISSIONS	fission reaction rate
AIRM	toxicity, m^3 of air to dilute to RCG
H2OM	toxicity, m^3 of water to dilute to RCG

LIBTYPE=

LITE (light nuclide/activation)
ACT (actinide sub-library)
FISS (fission product sub-library)
FISSACT (ACT+FISS)
ALL (all nuclides across sub-libraries)

SYMNUC=

symbolic notation of requested nuclides or elements

SORT=

option to sort user-requested nuclides by their importance (YES, NO)

RESPONSE=

optional user-requested response conversion factors

NRANK=

number of nuclides/elements

OPUS options (data set selection)

NPOSITION=
list of all requested position numbers of data in ascending order (data in f71)

MINPOSITION=
minimum position number, used to request a range of positions

MAXPOSITION=
maximum position number, used to request a range of positions

TMIN=
minimum time for selection of positions (units as for TIME)

TMAX=
maximum time for selection of positions (units as for TIME)

Example inputs

```
=opus
data= "my.f71"
typarams=gspectrum
units=intensity
npos=20 end
title="gamma spectrum at 1 year"
end
```

Gamma spectra for data saved at position #20 of "my.f71"

```
=opus
units=watts
title="decay heat"
end
```

Total decay heat (automatically selects decay time steps if none are specified)

```
=opus
units=grams time=days
minpos=1 maxpos=10
symnuc=cm-242 cm-244 cm-245 cm-246 end
title="cm mass as function of burnup"
end
```

Nuclide inventory (grams) for selected curium isotopes for saved positions #1–10

Example inputs

```
=opus
units=grams
time=year
typarms=elements
tmin=1 tmax=2
title="spent fuel at 33 gwd/mthm"
xlabel="cooling time (years)"
symnuc=u pu np am cm end
end
```

Elemental inventories (grams) for selected actinides for all time steps between 1–2 years decay time

```
=opus
units=wpel
libtyp=act
symnuc=u pu end
end
```

Weight percent of uranium and plutonium isotopes during decay

```
=opus
typarams=nspe
npos=5 end
title="neutron spectrum at 1 year"
end
```

Neutron spectrum for data saved on position #5

OPUS sample plot file

Radioactivity (Ci) at 5 years of cooling

```
pwr spent fuel - 33 gwd/mtu
cooling time (years)
Bq
nuclide
9 23
                2.738e-02 1.095e-01 2.738e-01 6.571e-01 1.000e+00 2.000e+00 3.000e+00 4.000e+00 5.000e+00
         ag110m 9.793e+14 9.010e+14 7.628e+14 5.172e+14 3.654e+14 1.326e+14 4.811e+13 1.746e+13 6.336e+12
          am241 5.106e+08 5.914e+08 7.520e+08 1.122e+09 1.446e+09 2.362e+09 3.232e+09 4.060e+09 4.847e+09
         ba137m 1.358e+18 1.355e+18 1.350e+18 1.338e+18 1.327e+18 1.297e+18 1.268e+18 1.239e+18 1.211e+18
          cf252 7.976e+10 7.806e+10 7.477e+10 6.762e+10 6.181e+10 4.756e+10 3.660e+10 2.816e+10 2.167e+10
          cm242 1.374e+15 1.210e+15 9.373e+14 5.167e+14 3.034e+14 6.426e+13 1.370e+13 3.013e+12 7.516e+11
          cm244 3.272e+13 3.261e+13 3.241e+13 3.194e+13 3.152e+13 3.034e+13 2.920e+13 2.810e+13 2.705e+13
          cm246 4.308e+08 4.308e+08 4.308e+08 4.308e+08 4.308e+08 4.307e+08 4.306e+08 4.306e+08 4.305e+08
          cm248 1.361e+04 1.361e+04 1.361e+04 1.361e+04 1.361e+04 1.361e+04 1.361e+04 1.361e+04 1.362e+04
           co60 4.496e+14 4.447e+14 4.352e+14 4.138e+14 3.956e+14 3.468e+14 3.041e+14 2.666e+14 2.338e+14
          cs134 1.473e+15 1.433e+15 1.356e+15 1.192e+15 1.063e+15 7.597e+14 5.431e+14 3.882e+14 2.776e+14
          eu152 1.522e+10 1.515e+10 1.503e+10 1.474e+10 1.448e+10 1.376e+10 1.307e+10 1.242e+10 1.180e+10
          eu154 4.422e+13 4.393e+13 4.335e+13 4.203e+13 4.089e+13 3.772e+13 3.480e+13 3.211e+13 2.962e+13
           nb94 3.509e+05 3.509e+05 3.509e+05 3.509e+05 3.509e+05 3.508e+05 3.508e+05 3.508e+05 3.508e+05
          pr144 1.142e+21 1.061e+21 9.171e+20 6.524e+20 4.810e+20 1.978e+20 8.135e+19 3.345e+19 1.376e+19
          pu238 5.347e+10 5.413e+10 5.519e+10 5.675e+10 5.748e+10 5.801e+10 5.776e+10 5.735e+10 5.691e+10
          pu239 1.508e+07 1.509e+07 1.509e+07 1.509e+07 1.509e+07 1.509e+07 1.509e+07 1.509e+07 1.509e+07
          pu240 3.229e+08 3.230e+08 3.230e+08 3.231e+08 3.233e+08 3.236e+08 3.239e+08 3.242e+08 3.245e+08
         f424m2 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
          rh106 9.833e+21 9.298e+21 8.314e+21 6.404e+21 5.070e+21 2.566e+21 1.299e+21 6.574e+20 3.327e+20
           u238 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
            y90 4.971e+15 4.942e+15 4.920e+15 4.875e+15 4.835e+15 4.720e+15 4.608e+15 4.498e+15 4.391e+15
       subtotal 1.098e+22 1.036e+22 9.232e+21 7.058e+21 5.553e+21 2.765e+21 1.381e+21 6.920e+20 3.477e+20
          total 1.098e+22 1.036e+22 9.232e+21 7.058e+21 5.553e+21 2.765e+21 1.381e+21 6.920e+20 3.477e+20
```

OPUS sample plot file: Neutron spectrum

The neutron and gamma spectra (particles/s) is divided by the energy group width (MeV) for plotting.

This conversion (to units of particles/s –MeV) produces intensities that are more easily comparable across different energy groups and energy group structures.

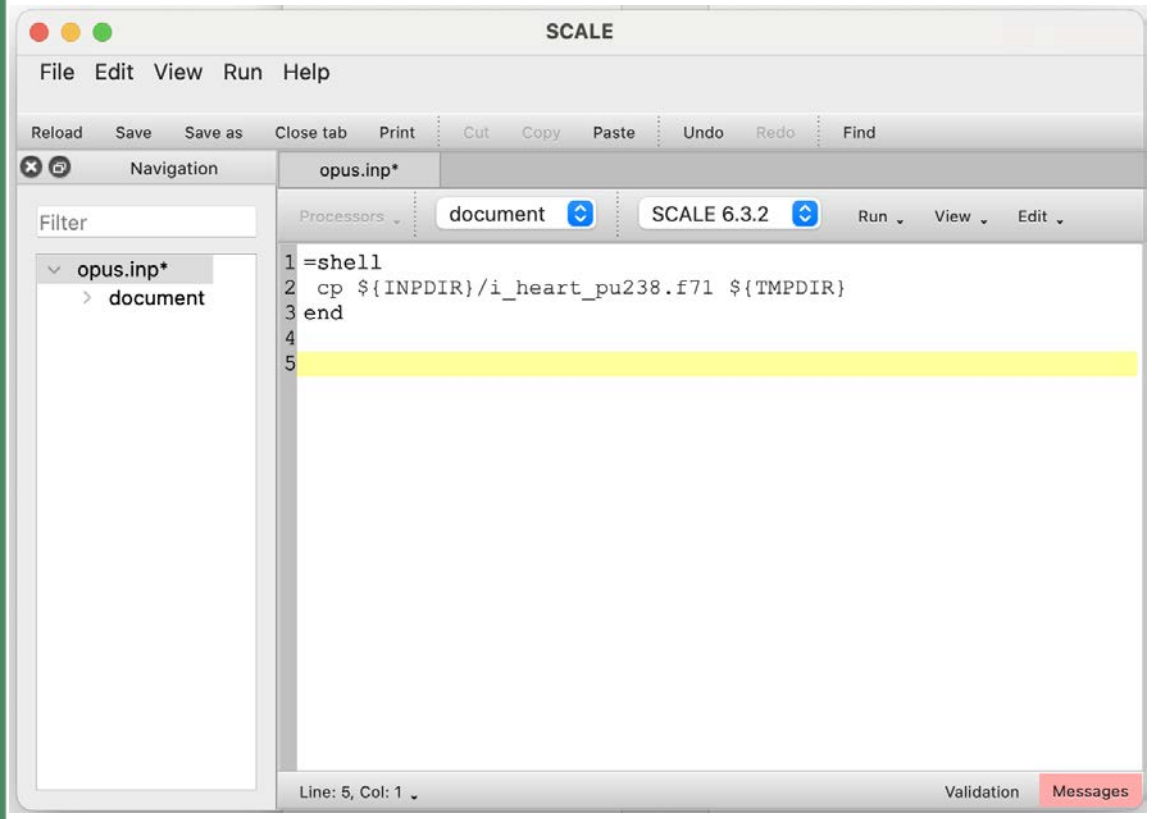
```
energy (mev)
1/(s.MeV)
spec
1 38
1.15741e-05 days
   1.000e-02   1.142e+09
   4.500e-02   1.142e+09
   4.500e-02   1.763e+08
   1.000e-01   1.763e+08
   1.000e-01   1.929e+07
   2.000e-01   1.929e+07
   2.000e-01   9.576e+05
   3.000e-01   9.576e+05
   3.000e-01   1.242e+06
   4.000e-01   1.242e+06
   4.000e-01   3.971e+04
   6.000e-01   3.971e+04
   6.000e-01   1.379e+04
   8.000e-01   1.379e+04
   8.000e-01   5.198e+05
   1.000e+00   5.198e+05
   1.000e+00   7.071e+09
   1.330e+00   7.071e+09
   1.330e+00   1.997e+09
   1.660e+00   1.997e+09
   1.660e+00   0.000e+00
   2.000e+00   0.000e+00
   2.000e+00   3.410e+04
   2.500e+00   3.410e+04
   2.500e+00   2.696e+01
   3.000e+00   2.696e+01
   3.000e+00   0.000e+00
   4.000e+00   0.000e+00
   4.000e+00   0.000e+00
   5.000e+00   0.000e+00
   5.000e+00   0.000e+00
   6.500e+00   0.000e+00
   6.500e+00   0.000e+00
   8.000e+00   0.000e+00
   8.000e+00   0.000e+00
   1.000e+01   0.000e+00
   1.000e+01   0.000e+00
   2.000e+01   0.000e+00
```

OAK RIDGE National Laboratory

Tutorial: Post-processing F71 files with OPUS

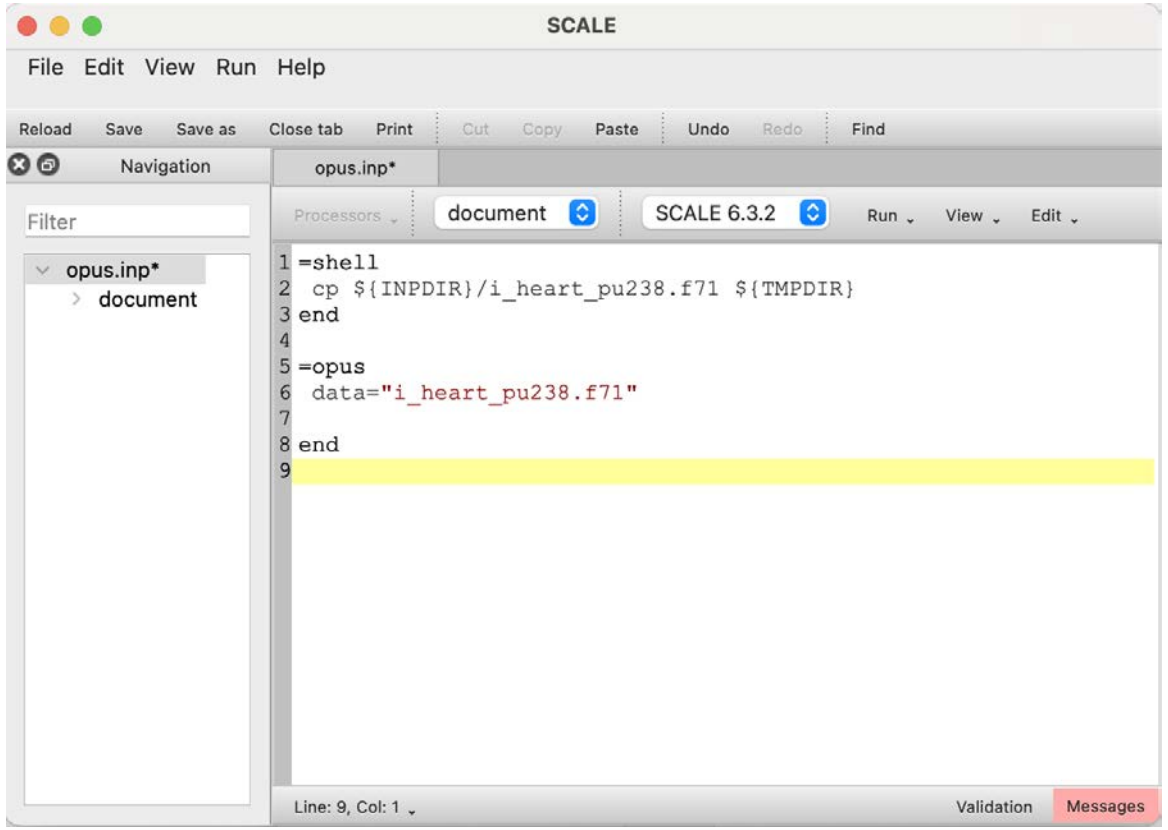


Copying a file into the SCALE temporary directory using shell



- The shell command offers a platform-independent way to manipulate files
  - A limited set of file manipulation utilities is also provided by SHELL (e.g., GREP, AWK)
- Here, we'll copy the F71 from our Pu-238 decay exercise from our input directory ${INPDIR} to our working temporary directory ${TMPDIR}
  - The shell module automatically expands the environment variables ${INPDIR} and ${TMPDIR}

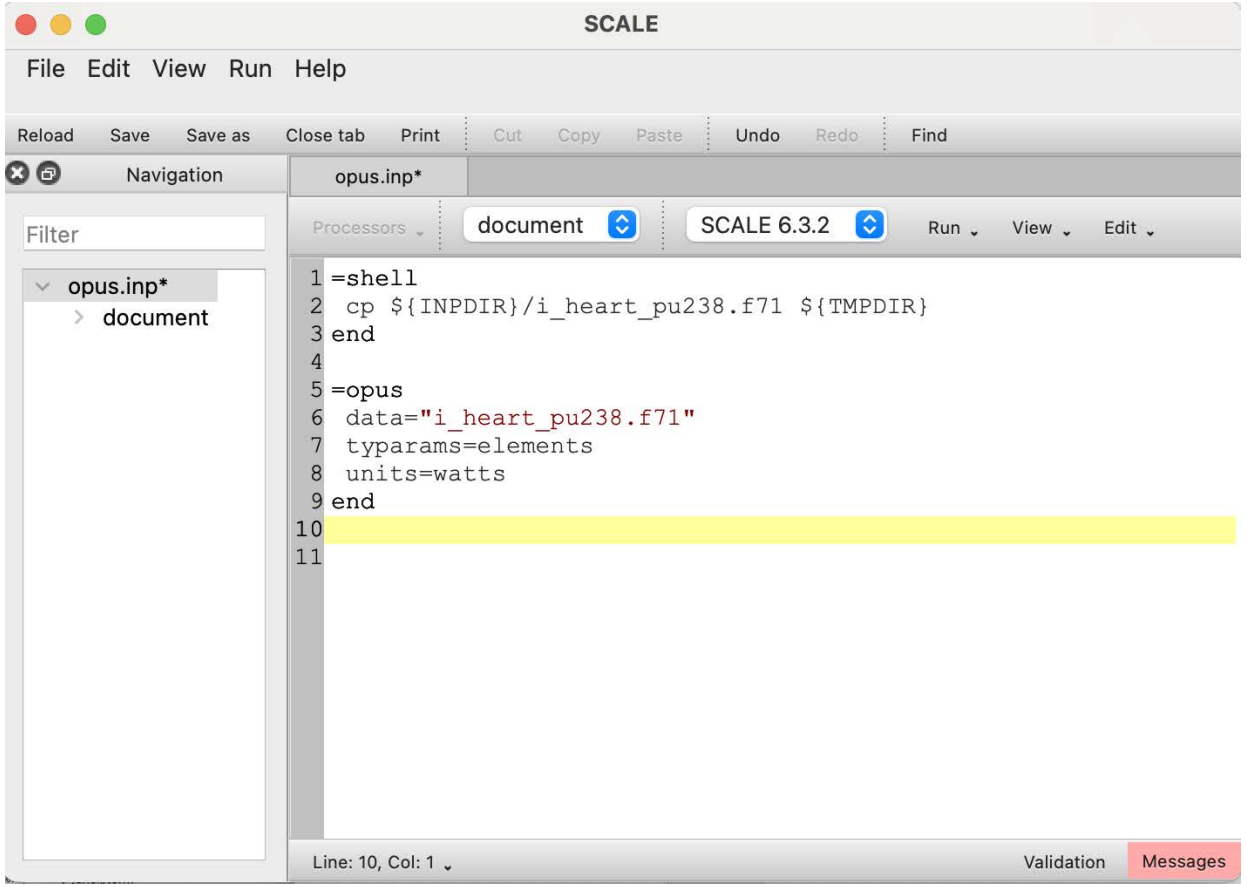
Working with our F71 file using OPUS



- Read data from the file you just copied into the temporary working directory:

```
data="i_heart_pu238.f71"
```

Getting decay heat by element



- Specify quantities to be output as by elements:

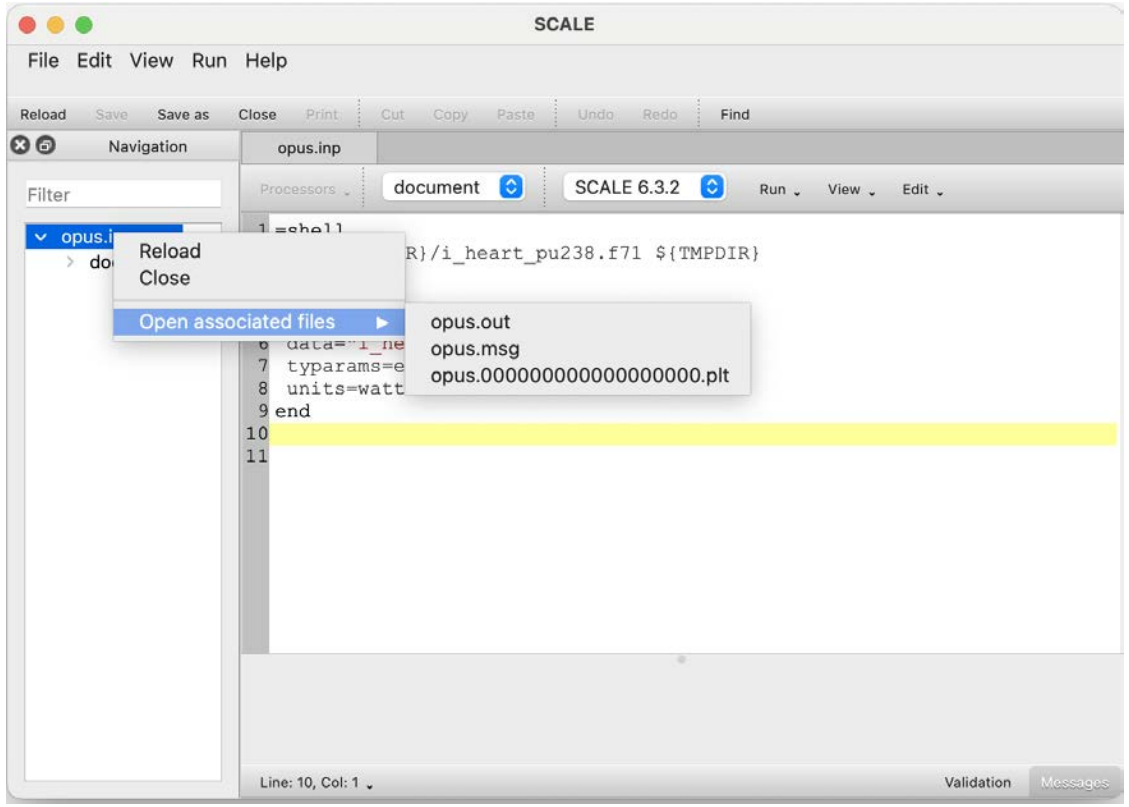
  `typarams=elements`

- Specify output units of Watts (total decay heat)

  `units=watts`

- Then hit "run"

Open the generated OPUS plot file

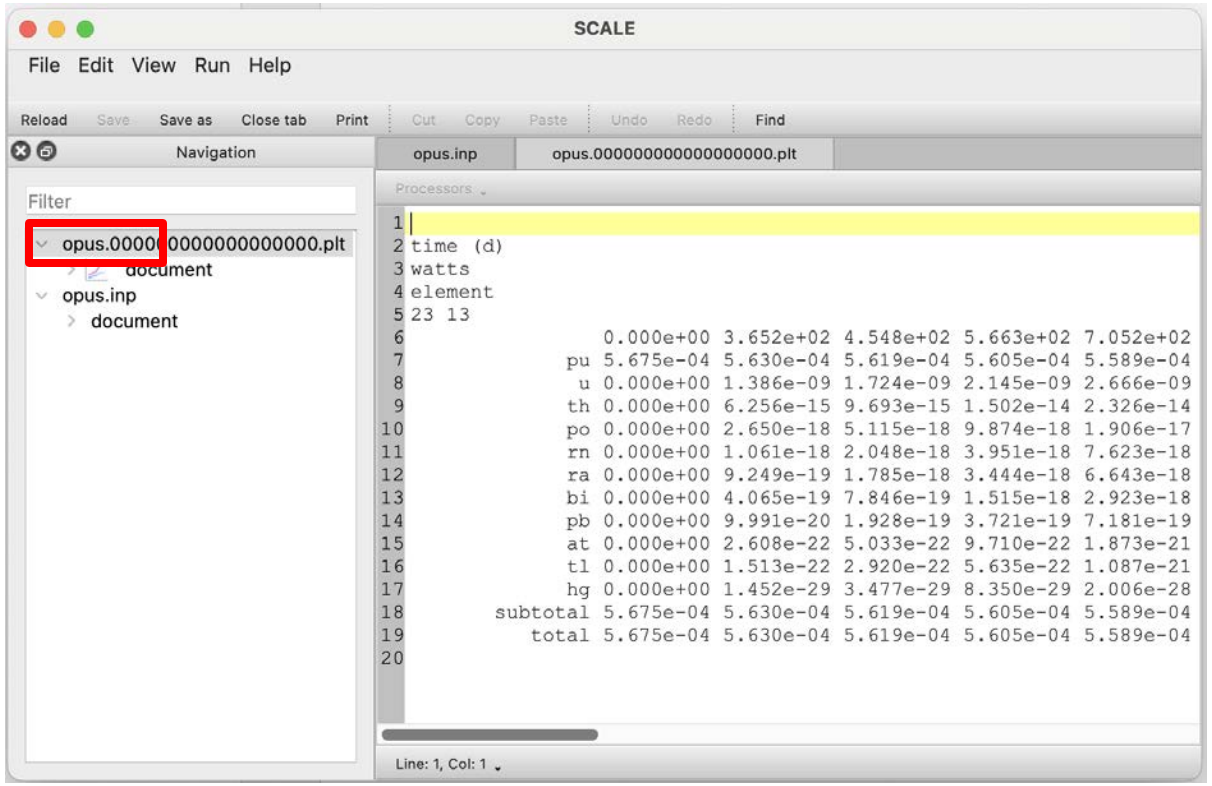


- Right-click on your input in the navigation menu
- Assuming your input ran correctly, you should have a file with the following name:

  `${BASENAME}.00000000000000000.plt`

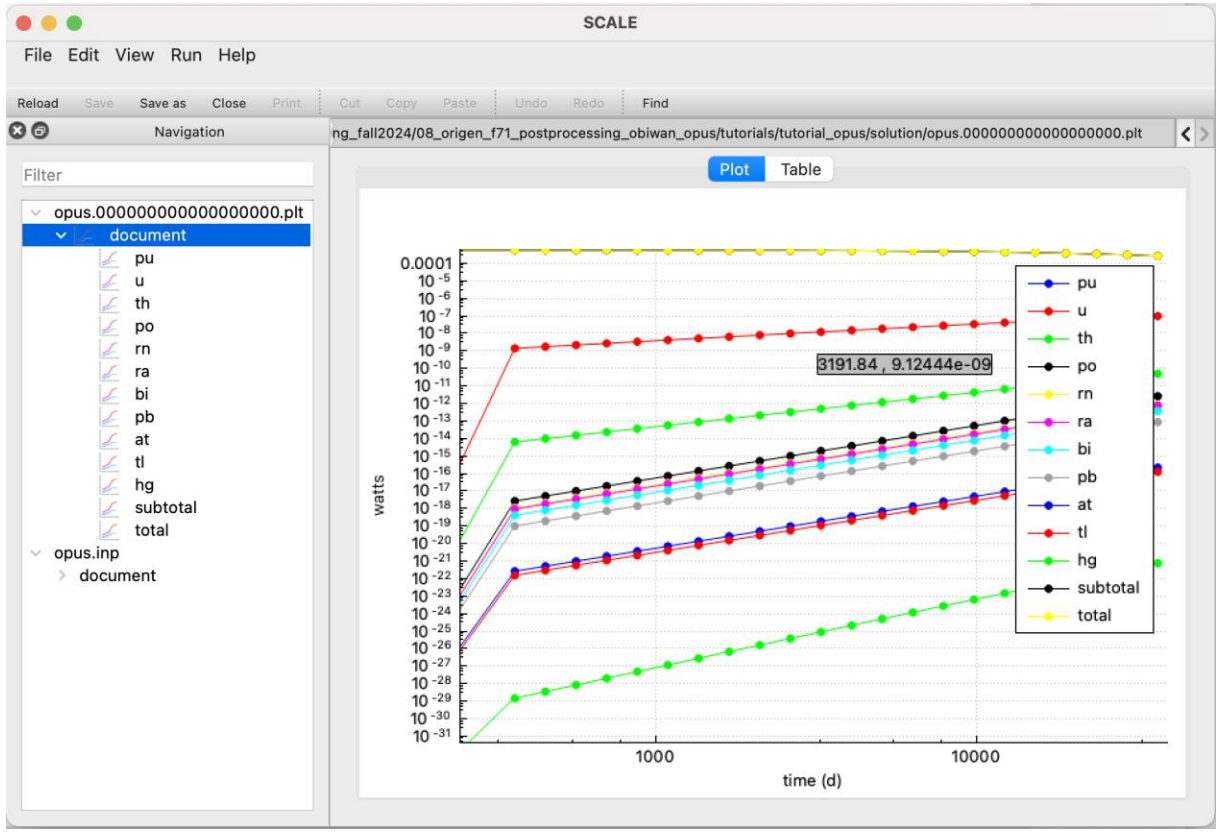
  where `${BASENAME}` is the root name of your input file

Examining the OPUS plot file



- The plot file contains the decay heat for each time interval (in units of Watts) by element
- This column-separated format is useful for additional post-processing using Python (or copying into Excel)
- But if we also double-click "document" under the navigation panel…

OPUS and the Fulcrum plot window



- Fulcrum natively "understands" how to process OPUS plt files
- While we selected a bare minimum of options here, OPUS can be used to screen for specific nuclides, timesteps, and time intervals
- OPUS can also be used to plot quantities such as radiation emission spectra

OAK RIDGE National Laboratory

Radiation emission calculations in ORIGEN



ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals



Introduce users to the radiation emission capabilities in ORIGEN (gamma, neutron, alpha, and beta) and their associated options



Demonstrate how OPUS can be used to visualize decay spectra calculated by ORIGEN

Input component: Emission calculations

```
=origen
bounds{
    neutron=[20e6 1e3 1 1e-3] %eV
}
case{
    …
    print{
       neutron{
           spectra=yes
       }
    }
    neutron=yes
    %equivalent to
    %neutron{
    %   alphan_medium=CASE
    %   alphan_step=LAST
    %   alphan_bins=200
    %   alphan_cutoff=0.0
    %}
}
…
end
```

- Emission calculations have 3 parts (neutron as example)
  - define bounds in eV: bounds→neutron
  - enable calculation/set parameters: case→neutron
  - ask for printout: case→print→neutron
- Neutron and gamma emission includes a charged particle slowing down component
  - (alpha,n) component for neutron emission
  - β+, β- Bremsstrahlung for gamma emission
- Alpha and beta emissions are *pure sources*, i.e., no slowing down!

Input component: Neutron

```
=origen
bounds{
    neutron=[1e6 1e3 1] %2-group with 1MeV, 1keV, 1eV
    %read neutron bounds from file instead
    %neutron="scale.rev04.xn252v7.1"
}
case{
    …
    neutron{
        alphan_medium=CASE %0/UO2 for UO2
                           %1/BOROSILICATE
                           %  for Borosilicate glass
                           %2/CASE for case-specific
                           %(DEFAULT)
        alphan_bins=200    %use 200 bins for
                           %alpha slowing down
        alphan_cutoff=0.0  %cutoff for alpha,n
        alphan_step=LAST   %step index in this case
    } %alternatively neutron=yes
    print{
        neutron{
            summary=yes
            spectra=no
            detailed=no
        }
    }
}
end
```

- Neutron emission calculation has 3 components
  - (alpha,n) neutrons
  - spontaneous fission neutrons
  - delayed neutrons
- (alpha,n) requires a charged particle slowing down calculation→dependent on medium
  - medium: UO_2, Borosilicate glass, problem-dependent
  - Problem-dependent compositions taken from a single step (alphan_step)
  - Isotopes with atom fraction or an activity below cutoff are not considered

Input component: Gamma

```
=origen
bounds{
    gamma=[1e6 1e3 1] %2-group with 1MeV, 1keV, 1eV
    %read gamma bounds from file instead
    %gamma="scale.rev10.xn27g19v7.0"
}
case{
    …
    gamma{
        sublib=ALL      %LT, FP, or AC – single sublib
                           %ALL – all sub-libraries
        brem_medium=UO2 %assume Bremsstrahlung in UO2
        immediate=yes   %load lines for immediate gamma/x-rays
        spont=yes       %include photons from spontaneous fission
                           %and alpha,n reactions
        continuum=yes %expand continuum data "lines" into proper continuua
        conserve_line_energy=no %conserve line energy instead of line intensity
        adjust_for_missing=no   %adjust photon intensity for missing data
        split_near_boundary=no  %lines within 3% of a group boundary are split
                                %into two bins
    }
    print{
        % default gamma options
        gamma{
            summary=yes
            spectra=no
            principal_step=NONE  %step index to calculate (NONE to suppress)
            principal_cutoff=2   %principal emitter cutoff in percent
            unbinned_warning=no  %print warning when line not binned
        }
    }
}
end
```

- Gamma emission calculation has 2 types of spectral data
  - line data
  - continuum data (e.g., bremsstrahlung)
- Bremsstrahlung requires charged particle slowing down
  - Pre-tabulated as a function of beta energy for UO_2 & H_2O media
  - brem_medium: UO_2, H_2O (no problem-dependent option like for neutron)
- Default calculation options generally fine except
  - brem_medium=H2O/UO2/NONE
  - If no option is appropriate, sensitivity can be assessed by trying each one

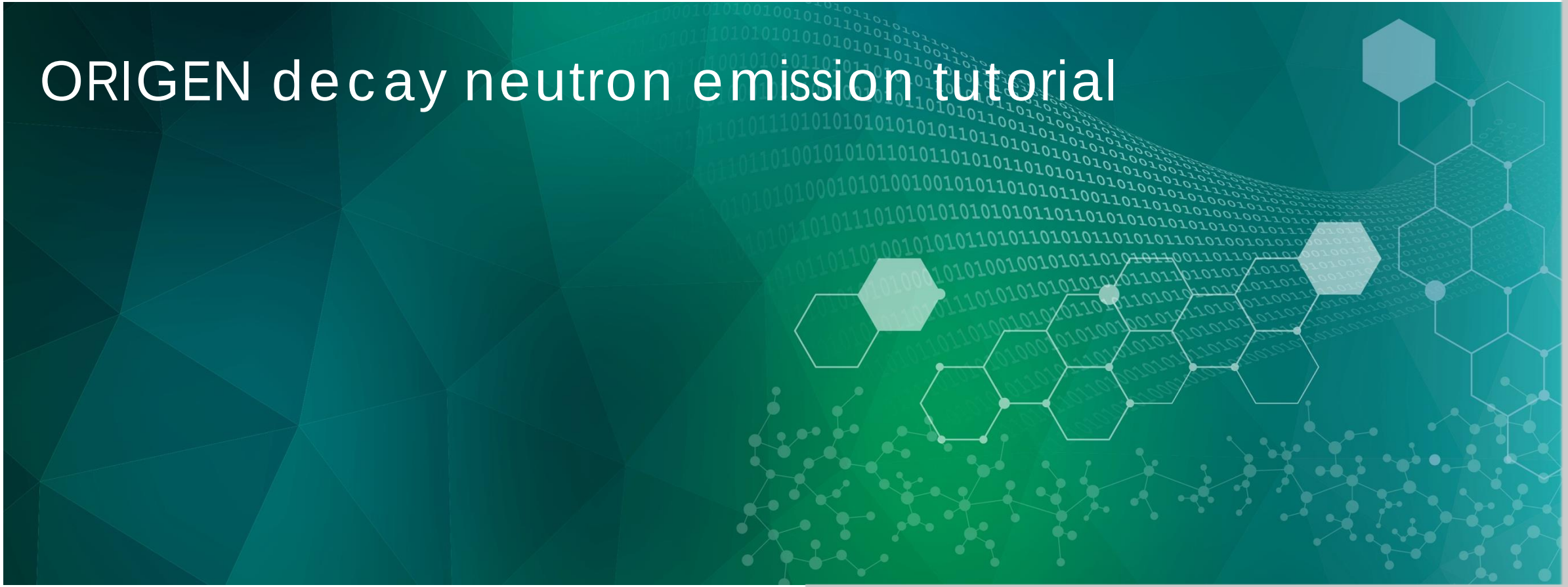
Input component: Alpha/beta

```
=origen
bounds{
    alpha=[1e6 1e3 1]
    beta=[1e6 1e3 1]
}
case{
    …
     beta{
        sublibs=ALL  %NONE – no calculation
                     %LT, FP, or AC – single sublib
                     %ALL – all sub-libraries
    } %or beta=yes
    alpha=yes %or alpha{}
    print{
        alpha{
            summary=yes
            spectra=no
        }
        beta{
            summary=yes
            spectra=no
            principal_step=NONE %step index to calculate
                                %principal emitters
            principal_cutoff=2  %principal emitter cutoff
(%)
        }
    }
}
end
```

- Alpha/beta bounds
  - `alpha/beta="/path/to/library"` not allowed
  - only direct eV input
- Calculation blocks
  - alpha block has no options
  - beta block only sets sublibs
- Remember alpha/beta are sources with no slowing down physics

OAK RIDGE National Laboratory

ORIGEN decay neutron emission tutorial



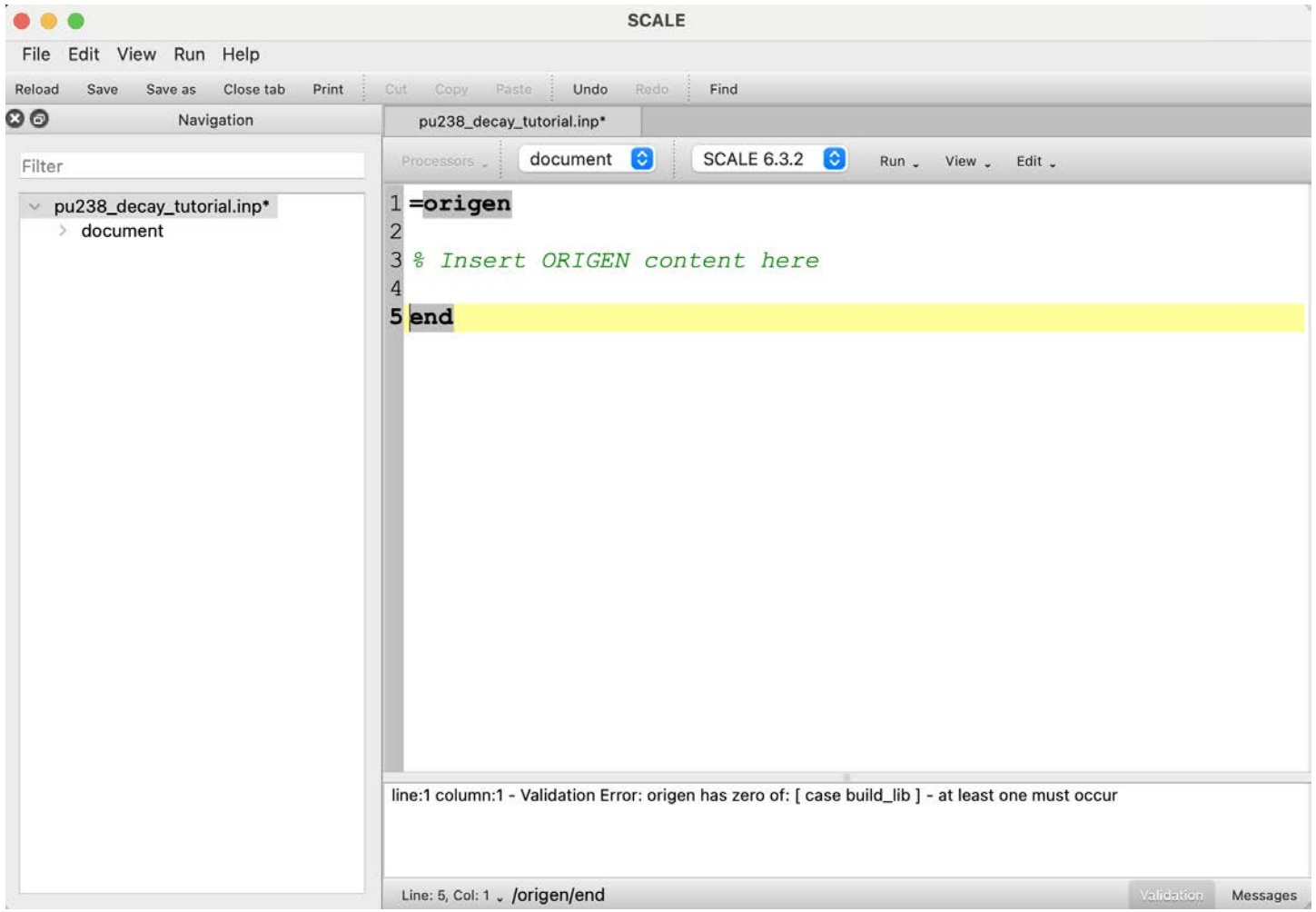
ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Overview

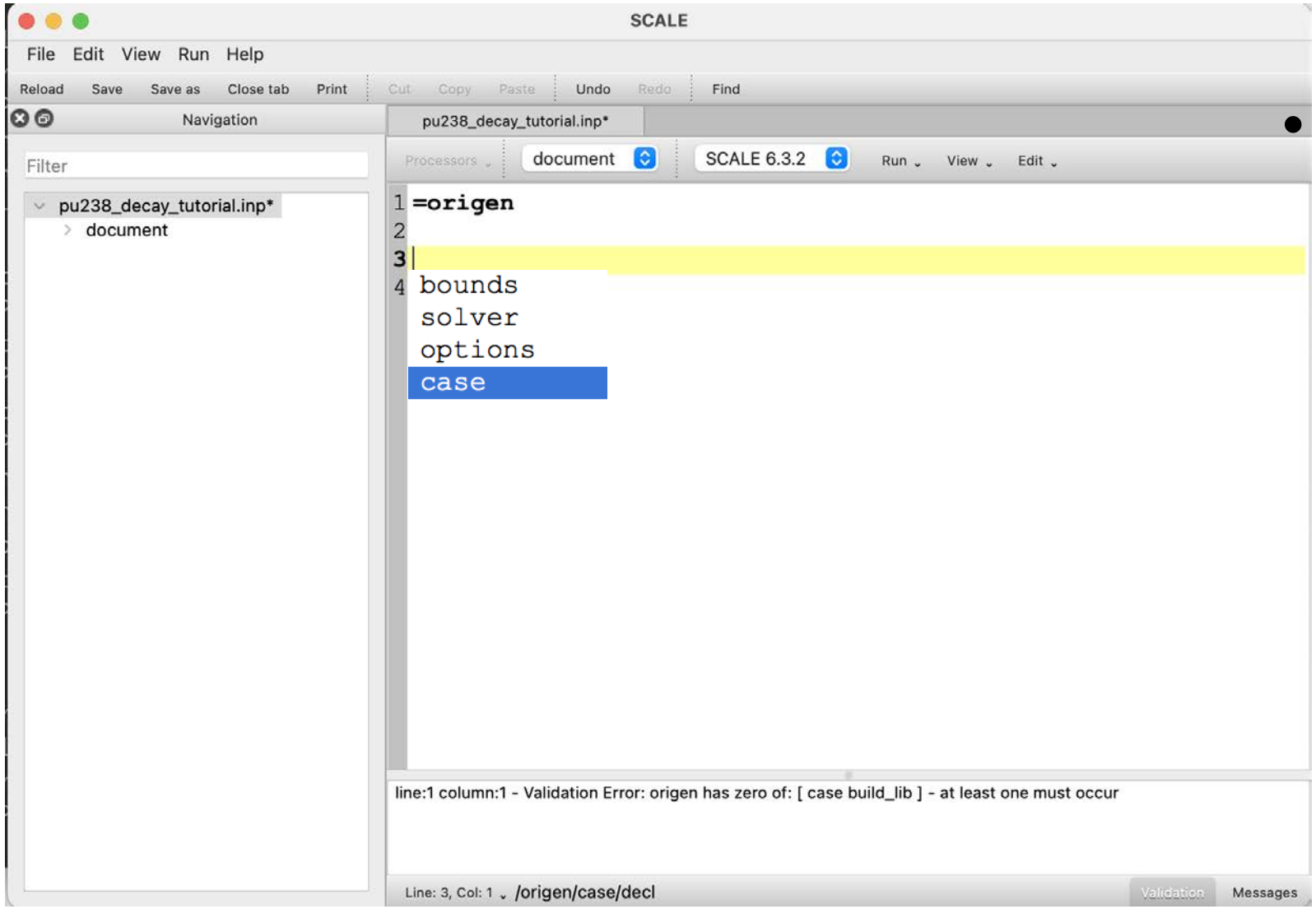
- Decay calculation with neutron emission
  - 1 g ^{238}Pu
  - 5 years decay
  - enable neutron calculation
  - compare (α,n) options problem specific to oxide matrix

Clean slate



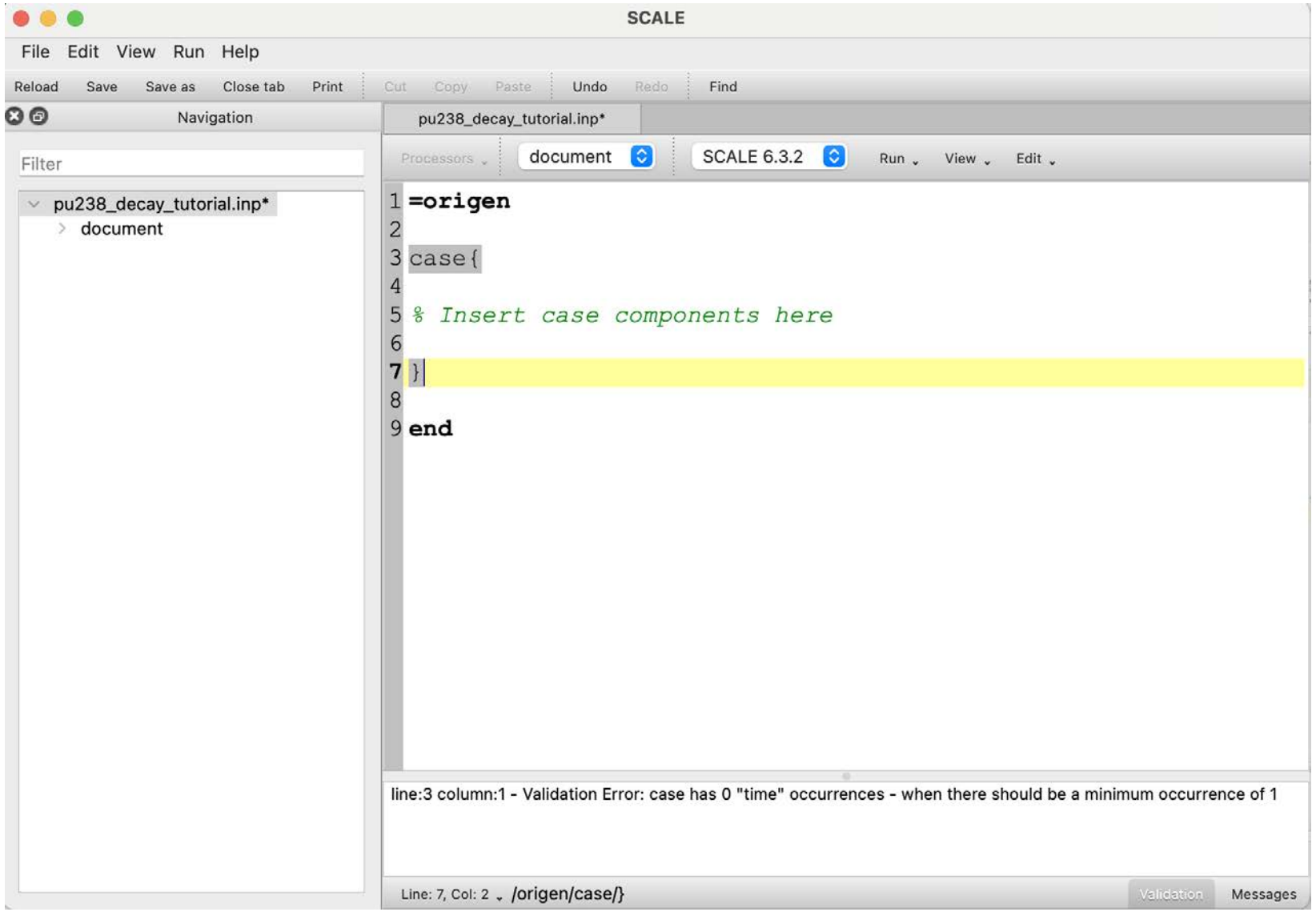
- Start with empty file
- For this tutorial, either
  - type commands as I do it
  - use CTRL+SPACE to autocomplete
- Click on "Validation" button to expand the validation window

Start with a case



- Create a case
  - CTRL+SPACE
  - Down arrow to case

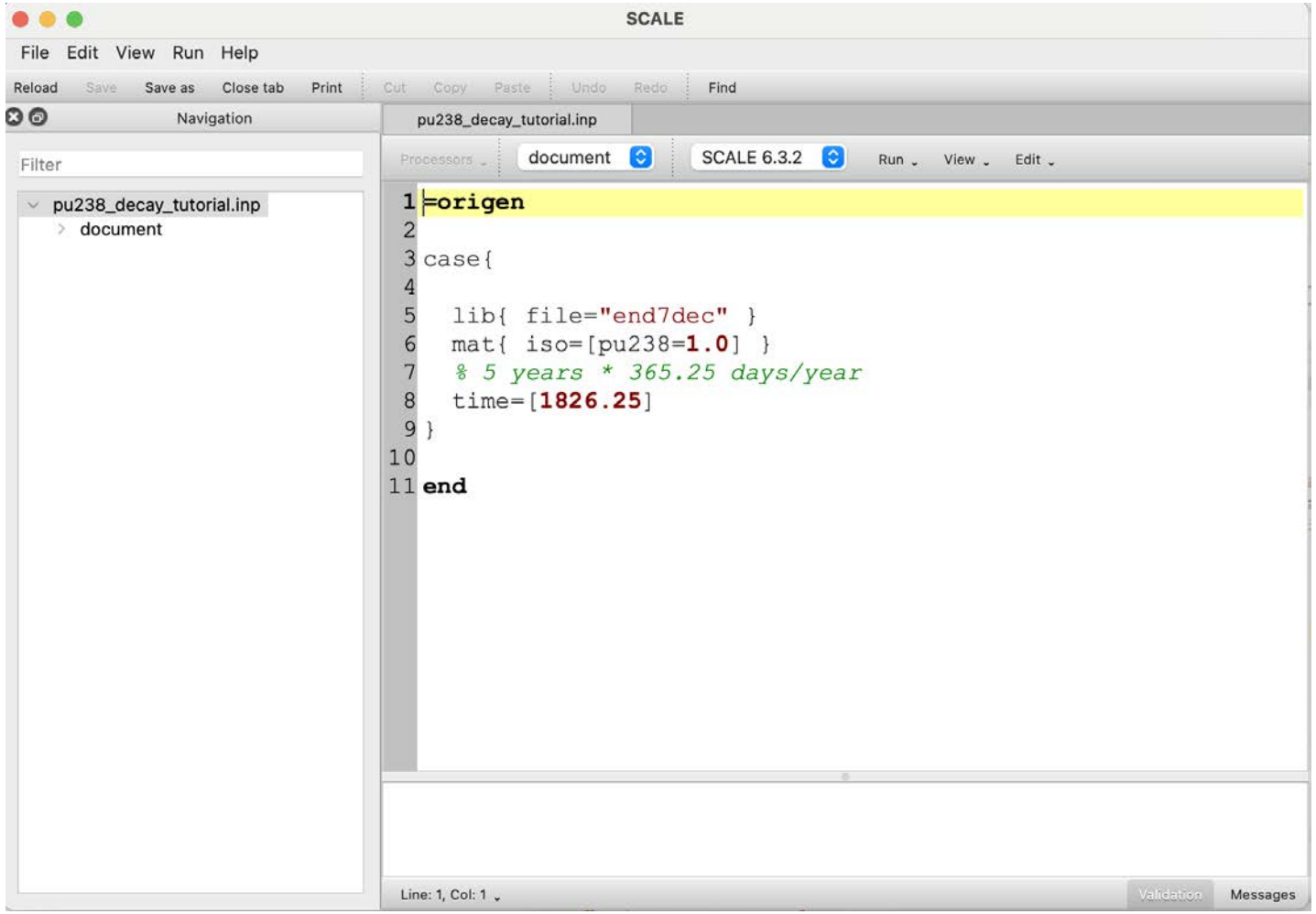
12 Possibilities in a Case



CTRL+SPACE inside case gives you

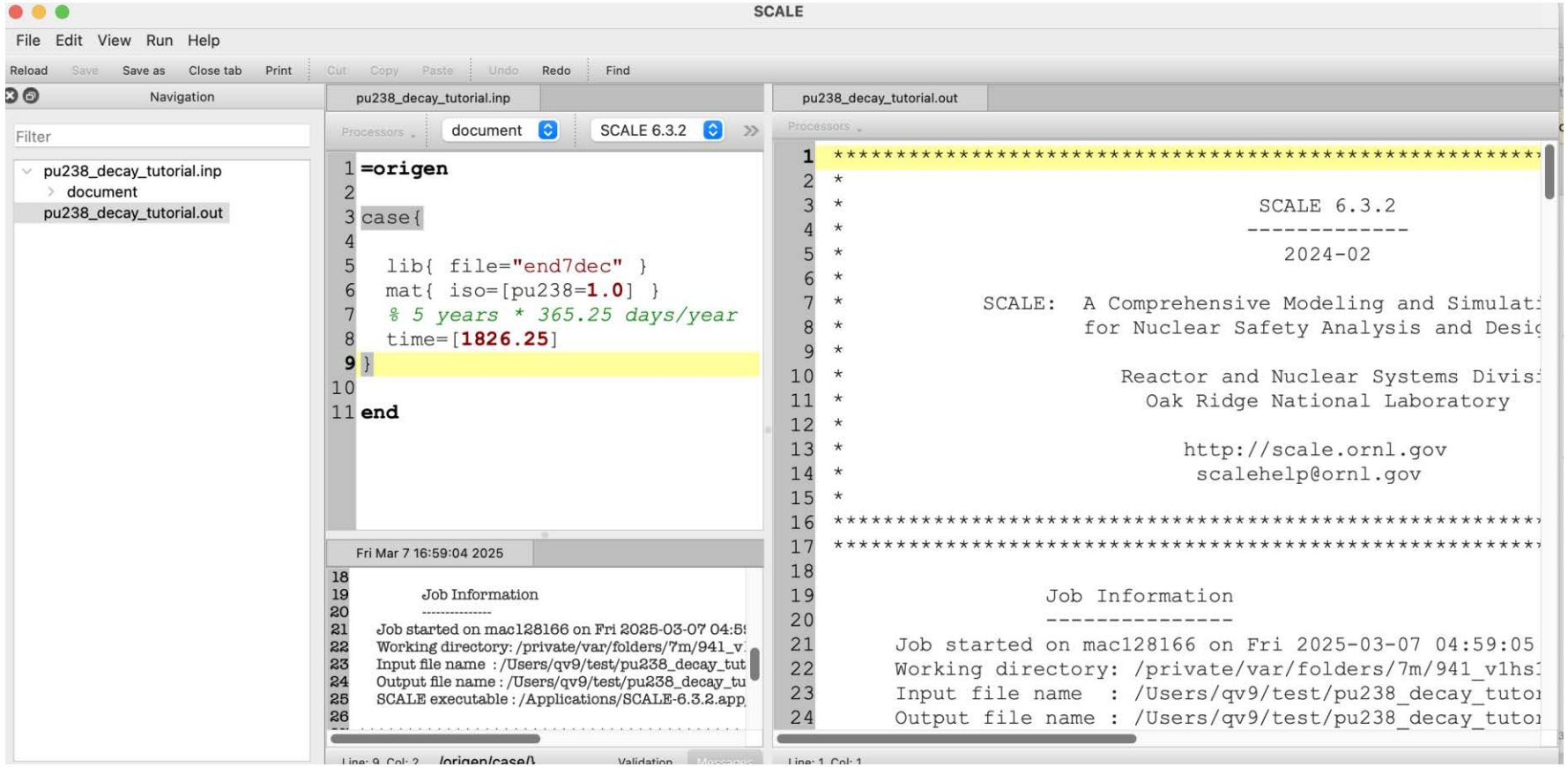
- title
- lib
- time
- flux
- power
- print
- mat
- save
- neutron
- gamma
- beta
- processing

Three are used almost every time



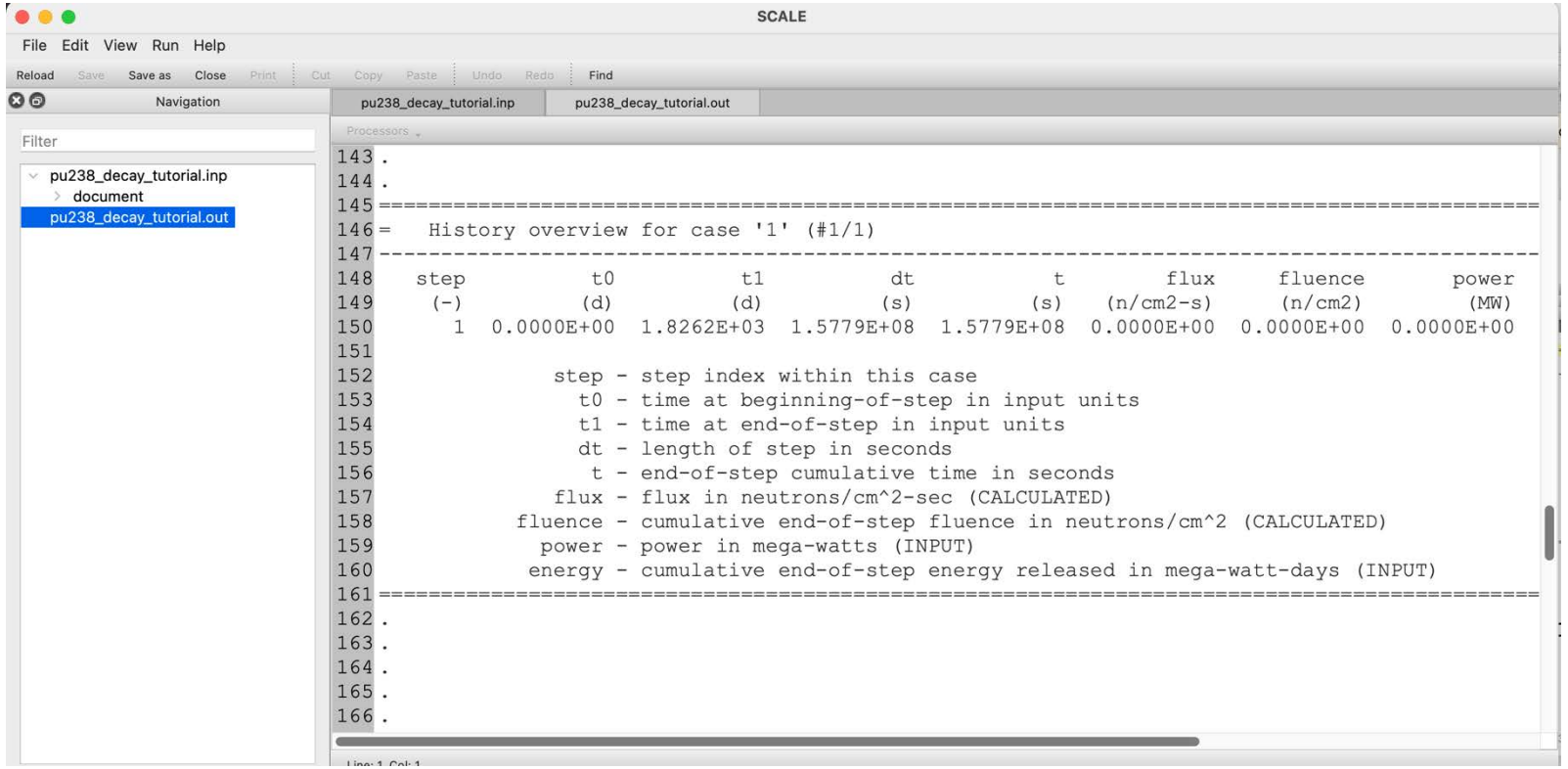
- Let's fill in the big 3
  - lib
    for decay, just use "end7dec"
  - mat
    iso array with default units
  - time
    time array with default units
- No validation errors!
- Click "Run"

Side-by-side for input/output



- Right-click in Navigation on *.inp and open *.out & *.msg
- Drag *.out TAB to right side of Fulcrum to have split screen input/output

Find the history table



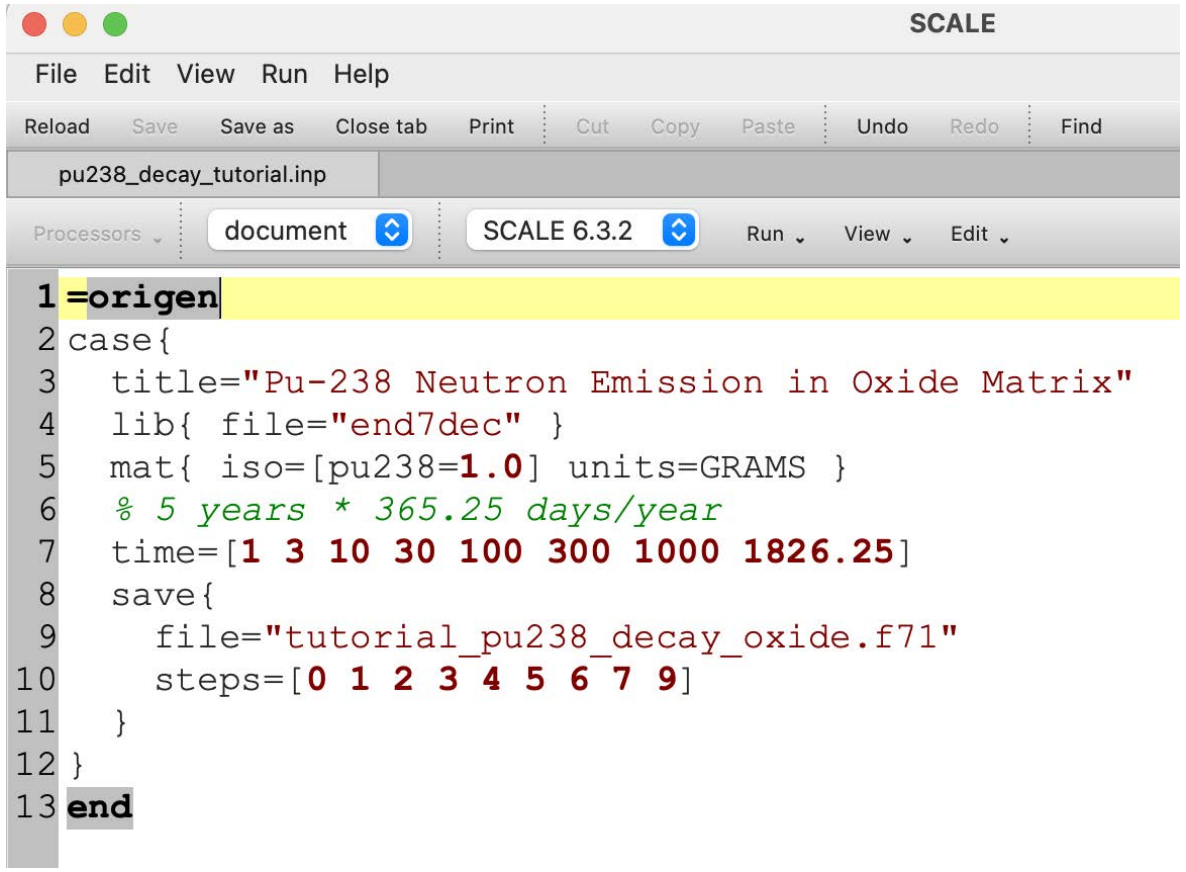
- SCALE output is very wide! (~120 columns)
- History overview is very valuable for double checking operating history entry (time/power/flux)
  - Use "Find" and search for "History overview"

Find the concentration table

```
.
**************************************************************************************************************
**********                                                                                          **********
**********                                                                                          **********
**********                          Concentration tables                                            **********
**********                                                                                          **********
**********                                                                                          **********
**************************************************************************************************************
.
.
==============================================================================================================
=   Nuclide concentrations in grams for case '1' (#1/1)
--------------------------------------------------------------------------------------------------------------
  (relative cutoff; integral of concentrations over time >   1.00E-06 % of integral of all concentrations over time)
.
                    0.000d   1826.250d
  he-4         0.0000E+00  1.5509E-01
  th-230       0.0000E+00  6.3330E-05
  u-234        0.0000E+00  9.0686E+00
  pu-238       2.3805E+02  2.2883E+02
------------
  totals       2.3805E+02  2.3805E+02
==============================================================================================================
```

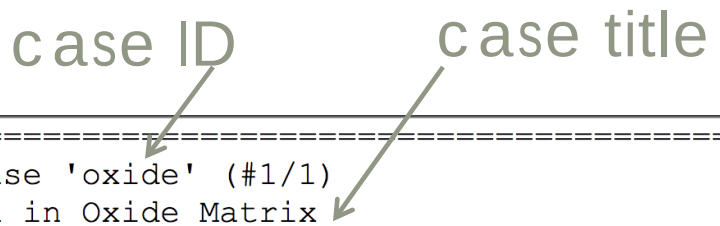
- Default output table is by nuclide in grams
- Verify 1 mole of Pu-238 = 238.05 grams
- Mass conserved, 0.15509 grams He generated

Add more detail



- Let's go back and make the following changes
  - add a title for the case
  - add an ID for the case (oxide)
  - change amount to 1 gram of Pu-238 (units=GRAMS)
  - add more times explicitly (no array shortcuts) using "rule of 3s" and start with 1 day
  - save an f71 file with all steps explicitly (no array shortcuts) called "tutorial_pu238_decay_oxide.f71"
- Inspect new output file
  - history
  - default grams output
  - compare He with 1 time step vs. rule of 3 s
    - 1 step: 6.5152E-04 g
    - Rule of 3s: 6.5152E-04 g

Verify in the output




```
146 ================================================================================================
147 =   History overview for case 'oxide' (#1/1)
148 =   Pu-238 Neutron Emission in Oxide Matrix
149 ------------------------------------------------------------------------------------------------
150    step          t0          t1          dt           t        flux     fluence       power      energy
151     (-)         (d)         (d)         (s)         (s)   (n/cm2-s)     (n/cm2)        (MW)       (MWd)
152       1  0.0000E+00  1.0000E+00  8.6400E+04  8.6400E+04  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
153       2  1.0000E+00  3.0000E+00  1.7280E+05  2.5920E+05  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
154       3  3.0000E+00  1.0000E+01  6.0480E+05  8.6400E+05  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
155       4  1.0000E+01  3.0000E+01  1.7280E+06  2.5920E+06  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
156       5  3.0000E+01  1.0000E+02  6.0480E+06  8.6400E+06  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
157       6  1.0000E+02  3.0000E+02  1.7280E+07  2.5920E+07  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
158       7  3.0000E+02  1.0000E+03  6.0480E+07  8.6400E+07  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
159       8  1.0000E+03  1.8262E+03  7.1388E+07  1.5779E+08  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
160
161              step - step index within this case
162                t0 - time at beginning-of-step in input units
163                t1 - time at end-of-step in input units
164                dt - length of step in seconds
165                 t - end-of-step cumulative time in seconds
166              flux - flux in neutrons/cm^2-sec (CALCULATED)
167           fluence - cumulative end-of-step fluence in neutrons/cm^2 (CALCULATED)
168             power - power in mega-watts (INPUT)
169            energy - cumulative end-of-step energy released in mega-watt-days (INPUT)
170 ================================================================================================
171 .
```

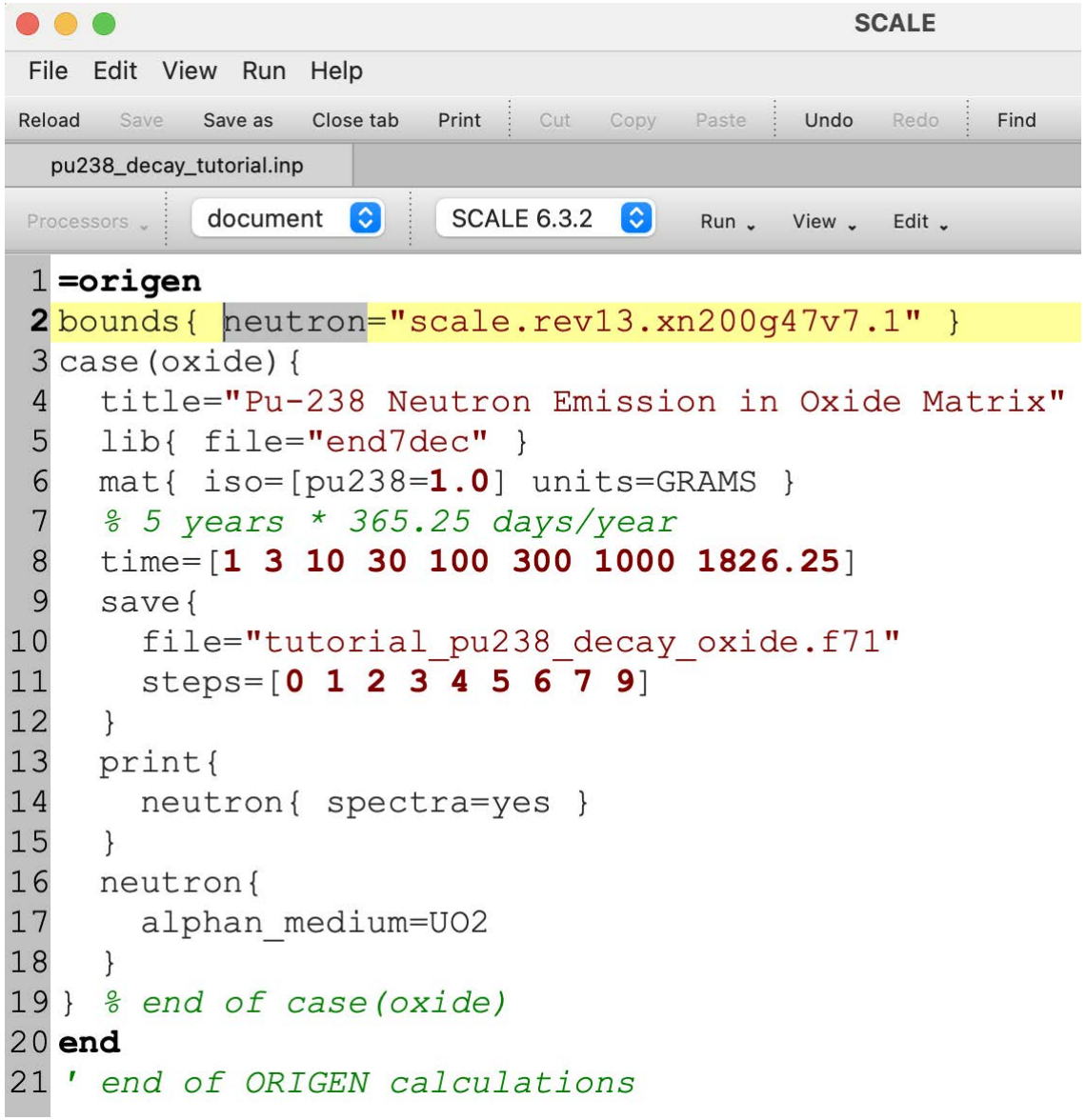
Verify the F71 is being written

```
**********************************************************************************************************
**********************                                                                        ***********
**********************                                                                        ***********
**********************                    Written F71 File Information                        ***********
**********************                                                                        ***********
**********************                                                                        ***********
**********************************************************************************************************
.
.
 index        time         power          flux       fluence        burnup libpos    case    step DCGNAB
     1  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       0 DC----
     2  8.64000e+04  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       1 DC----
     3  2.59200e+05  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       2 DC----
     4  8.64000e+05  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       3 DC----
     5  2.59200e+06  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       4 DC----
     6  8.64000e+06  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       5 DC----
     7  2.59200e+07  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       6 DC----
     8  8.64000e+07  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       7 DC----
     9  1.57788e+08  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00      1       1       8 DC----
    D - state definition present
    C - concentrations present
    G - gamma emission spectra present
    N - neutron emission spectra present
    A - alpha emission spectra present
    B - beta emission spectra present
```

available positions

content at each position

Add the neutron calculation




```
=origen
bounds{ neutron="scale.rev13.xn200g47v7.1" }
case(oxide){
  title="Pu-238 Neutron Emission in Oxide Matrix"
  lib{ file="end7dec" }
  mat{ iso=[pu238=1.0] units=GRAMS }
  % 5 years * 365.25 days/year
  time=[1 3 10 30 100 300 1000 1826.25]
  save{
    file="tutorial_pu238_decay_oxide.f71"
    steps=[0 1 2 3 4 5 6 7 9]
  }
  print{
    neutron{ spectra=yes }
  }
  neutron{
    alphan_medium=UO2
  }
} % end of case(oxide)
end
' end of ORIGEN calculations
```

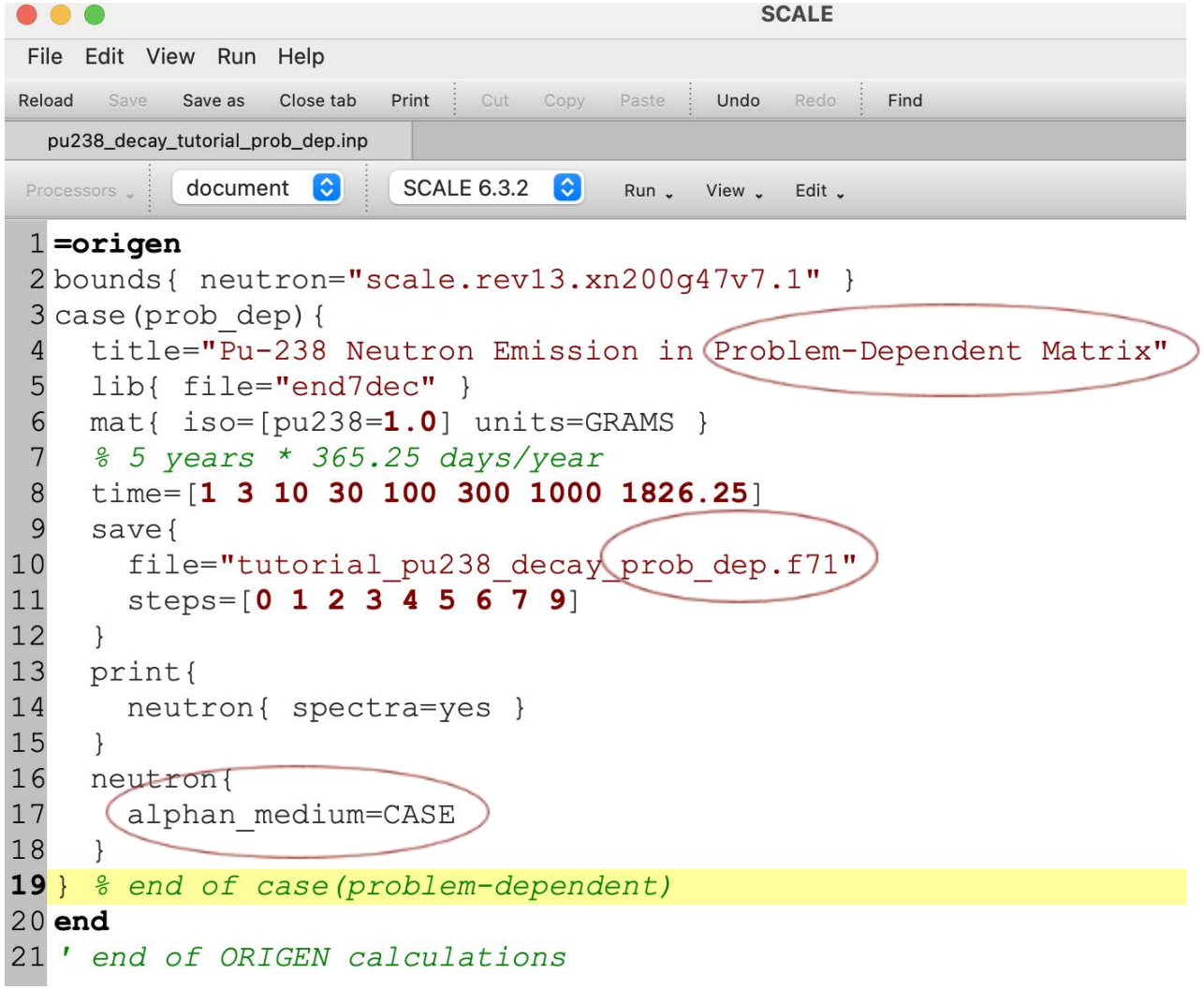
- Now let's add neutron calculation
  - case→neutron→alphan_medium=UO2
  - case→print→neutron→spectra=yes
  - bounds→neutron="scale.rev13.xn200g47v7.1"
- Inspect new output file
  - History
  - Default grams output
  - Compare He with 1 time step vs. rule of 3's
    - 1 step: 6.5152E-04 g
    - **Rule of 3's:** 6.5152E-04 g
  - Verify F71 now has neutron data

Verify neutron emission data on F71

```
1242 ********************************************************************************************************
1243 ********************
1244 ********************
1245 ********************                          Written F71 File Information
1246 ********************
1247 ********************
1248 ********************************************************************************************************
1249 .
1250 .
1251  index          time         power          flux       fluence        burnup libpos    case    step DCGNAB
1252      1   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       0 DC-N--
1253      2   8.64000e+04   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       1 DC-N--
1254      3   2.59200e+05   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       2 DC-N--
1255      4   8.64000e+05   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       3 DC-N--
1256      5   2.59200e+06   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       4 DC-N--
1257      6   8.64000e+06   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       5 DC-N--
1258      7   2.59200e+07   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       6 DC-N--
1259      8   8.64000e+07   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       7 DC-N--
1260      9   1.57788e+08   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00      1       1       8 DC-N--
1261     D - state definition present
1262     C - concentrations present
1263     G - gamma emission spectra present
1264     N - neutron emission spectra present
1265     A - alpha emission spectra present
1266     B - beta emission spectra present
```

neutron data available

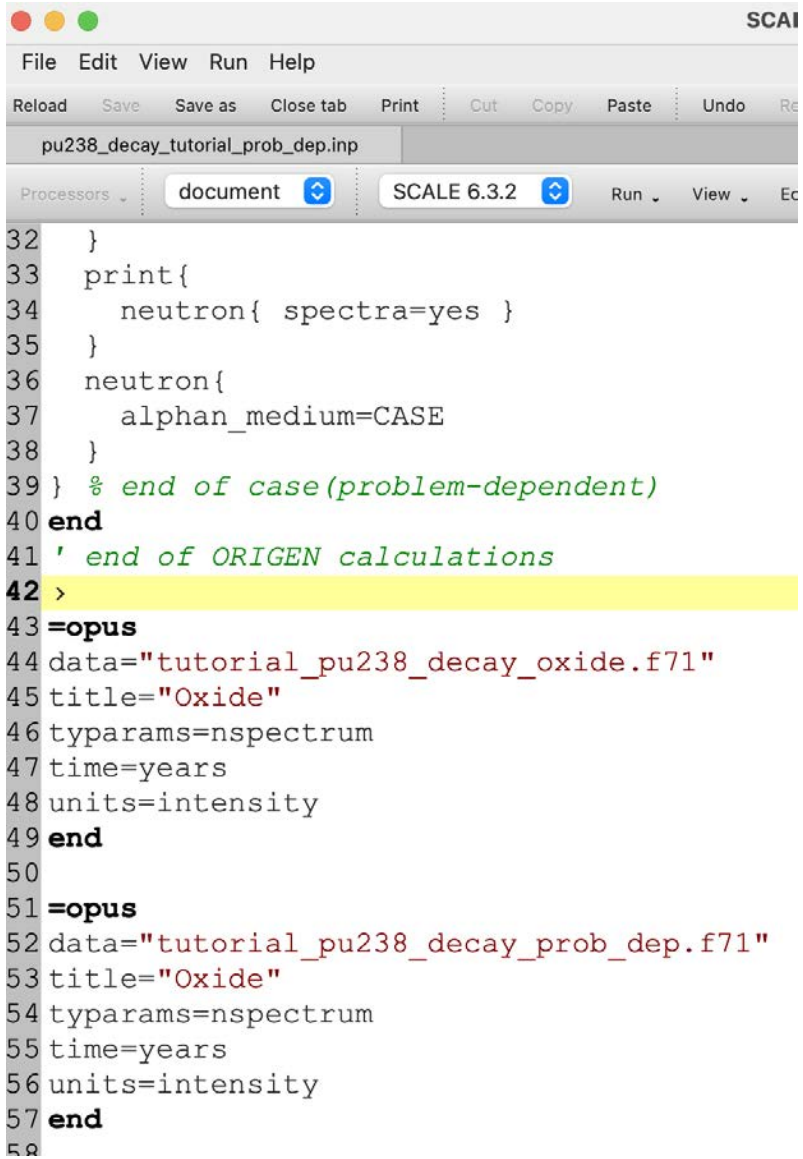
Create a second case with different medium




```
=origen
bounds{ neutron="scale.rev13.xn200g47v7.1" }
case(prob_dep){
  title="Pu-238 Neutron Emission in Problem-Dependent Matrix"
  lib{ file="end7dec" }
  mat{ iso=[pu238=1.0] units=GRAMS }
  % 5 years * 365.25 days/year
  time=[1 3 10 30 100 300 1000 1826.25]
  save{
    file="tutorial_pu238_decay_prob_dep.f71"
    steps=[0 1 2 3 4 5 6 7 9]
  }
  print{
    neutron{ spectra=yes }
  }
  neutron{
    alphan_medium=CASE
  }
} % end of case(problem-dependent)
end
' end of ORIGEN calculations
```

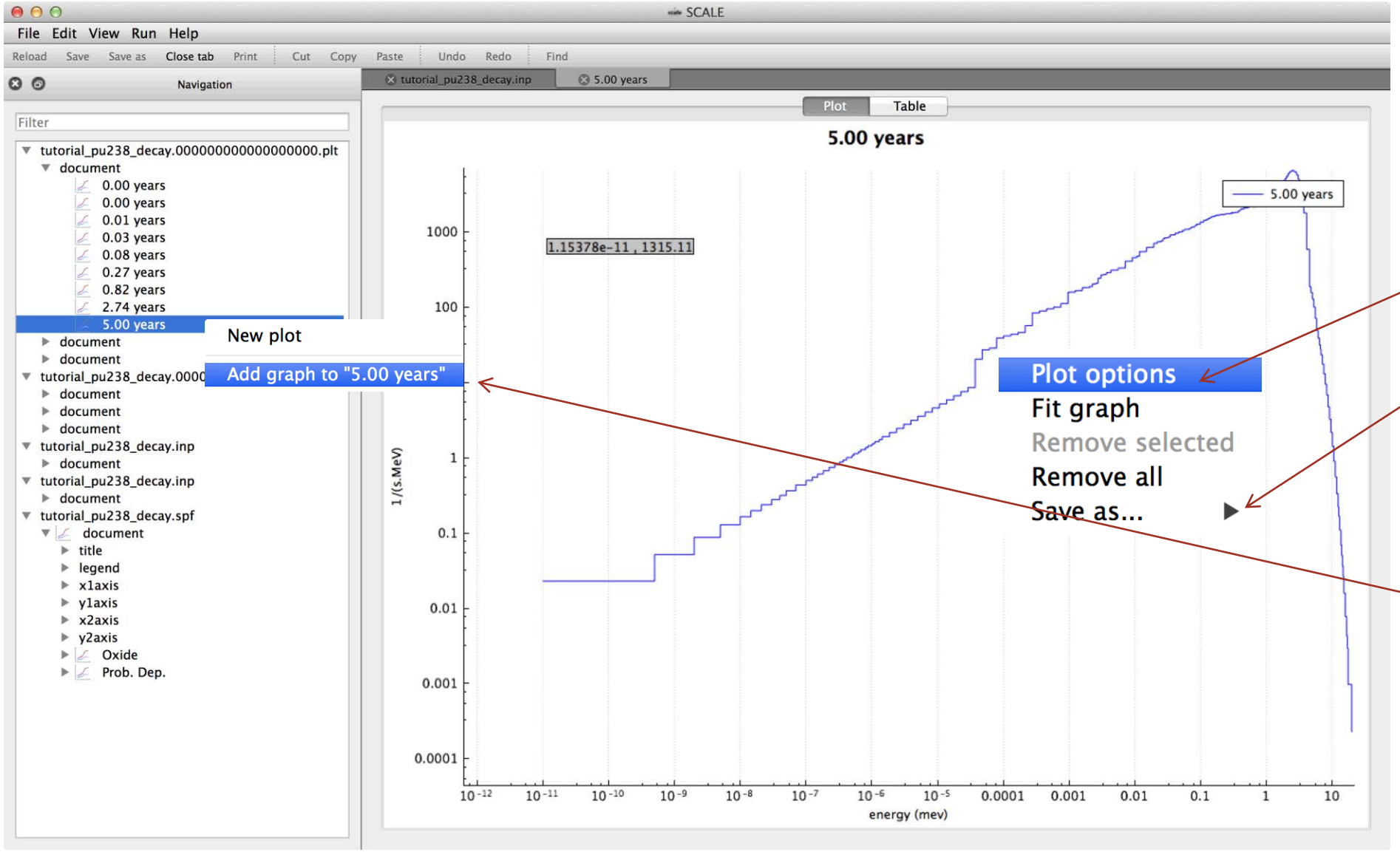
- Now let's add a second case!
  - copy-paste (CTRL-C / CTRL-V)
- Create independent case "case(probdep)" with problem-dependent medium
  - case→title
  - case→neutron→alphan_medium=CASE
  - case→save→file= "tutorial_pu238_decay_probdep.f71"
- Run it, examine output
  - find output tables for each case
  - find F71 summary for each case

Add OPUS



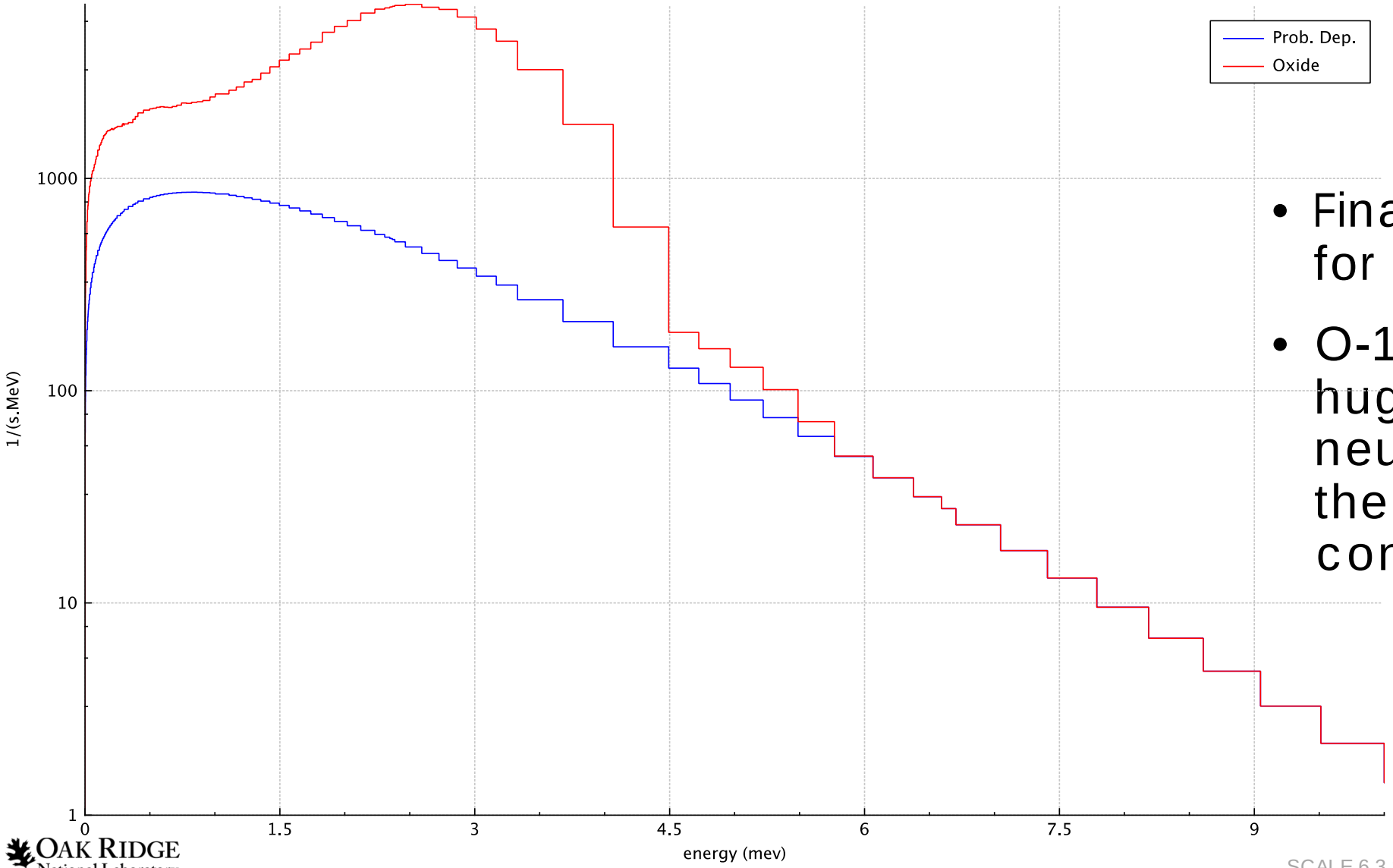
- Could copy/paste spectra from output file to Excel and make a comparison plot
- Here, we'll use OPUS and Fulcrum
  - Create one OPUS case for each f71 file
  - Rerun the calculation

Plot manipulation



- Can right-click on plot window
  - Change axes, legends, etc.
  - Export to image file
- Can right-click on plottable in nav to add to another plot

Final PDF



- Final PDF saved image for comparison
- O-17 and O-18 make huge difference in neutron source due to their (alpha,n) contribution

Useful shortcuts

- `save=yes` instead of a save block, saves the f71 with all time steps
- `neutron=yes` instead of a neutron block, enables a default neutron calculation (still need bounds, though)
- `gamma=yes` instead of a gamma block, enables a default gamma calculation (still need bounds, though)

Questions?



OAK RIDGE National Laboratory

ORIGEN Reactor Libraries

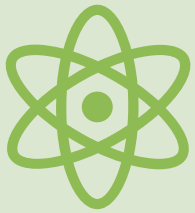
SCALE 6.3.2



ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals



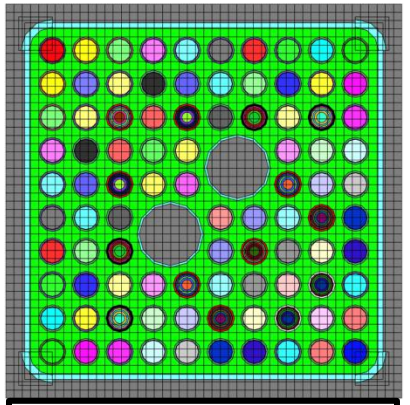
Understand ORIGEN reactor libraries

- What they are (what they physically represent)
- How they are used in depletion calculations
- How they are created using SCALE

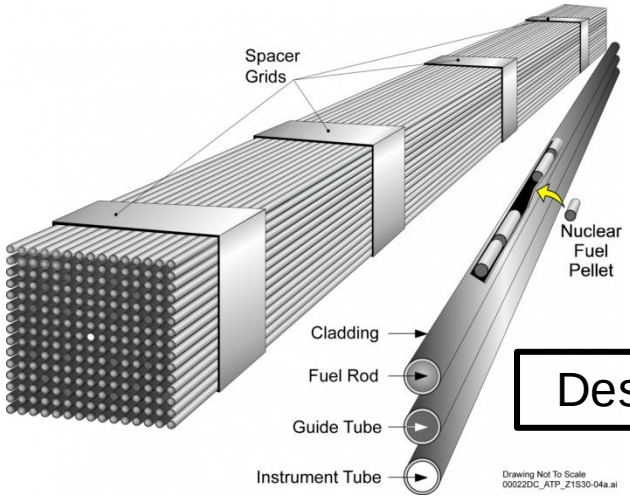


Familiarize users with the available pre-generated ORIGEN reactor libraries provided with SCALE

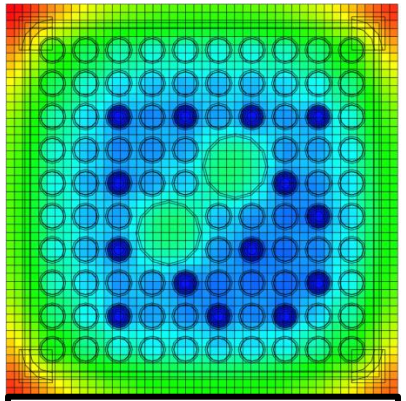
Generating ORIGEN reactor libraries



TRITON Model



Design Data



Depletion Calc.

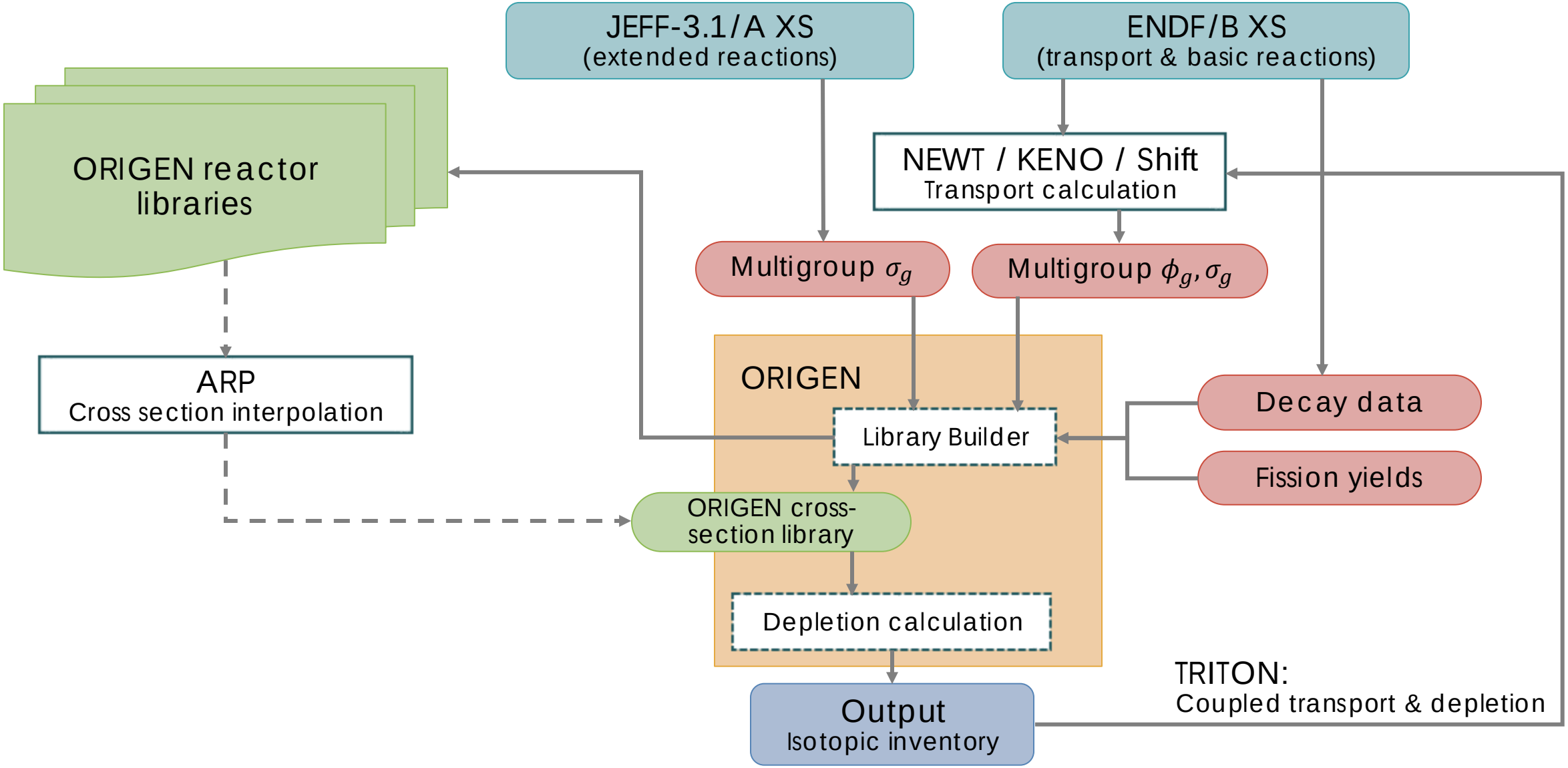


ORIGEN Library

Libraries

Collection +
Documentation

How ORIGEN uses nuclear data



What are ORIGEN cross section vs. ORIGEN reactor libraries?

ORIGEN cross section library

One F33 file containing burnup-dependent 1-group cross sections for one specific assembly design.

Note common terminology:
ORIGEN cross section library
= ORIGEN library
= ORIGEN 1-group library
= F33 file

ORIGEN reactor library

A set of ORIGEN cross section libraries for a specific reactor type and fuel assembly configuration, parametrized to, for example, the following:

- Enrichment
- Moderator density
- Pu content, ^{239}Pu values (MOX fuel)

Example

`g10_e40w03.f33`

One file corresponding to a specific BWR GE 10×10 fuel assembly model: 4.0 wt.% ^{235}U enrichment and 0.3 g/cm^3 coolant density

Example

```
g10_e05w01.f33
g10_e05w03.f33
g10_e05w07.f33
g10_e05w09.f33
g10_e15w01.f33
g10_e15w03.f33
…
```

```
…
g10_e80w07.f33
g10_e80w09.f33
g10_e85w01.f33
g10_e85w05.f33
g10_e85w07.f33
g10_e85w09.f33
```

A set of files for a BWR GE 10×10 fuel assembly for different combinations of enrichment and moderator density

What are these libraries used for?

ORIGEN cross section library

One F33 file containing burnup-dependent 1-group cross sections for one specific assembly design.

Note common terminology:
ORIGEN cross section library
= ORIGEN library
= ORIGEN 1-group library
= F33 file

ORIGEN reactor library

A set of ORIGEN cross section libraries for a specific reactor type and fuel assembly configuration, parametrized to, for example, the following:

- Enrichment
- Moderator density
- Pu content, ^{239}Pu values (MOX fuel)

ORIGEN

Used by ORIGEN to run a single depletion calculation of a specific problem to obtain the time-dependent inventory for this exact problem.

ORIGAMI

Used by ORIGAMI to rapidly run a depletion calculation and generate inventory for varying parameters.

User input: reactor library name together with desired enrichment/moderator density and depletion history
ORIGAMI calculation: interpolation of cross sections from pre-generated libraries followed by depletion to generate inventory

Available ORIGEN reactor libraries

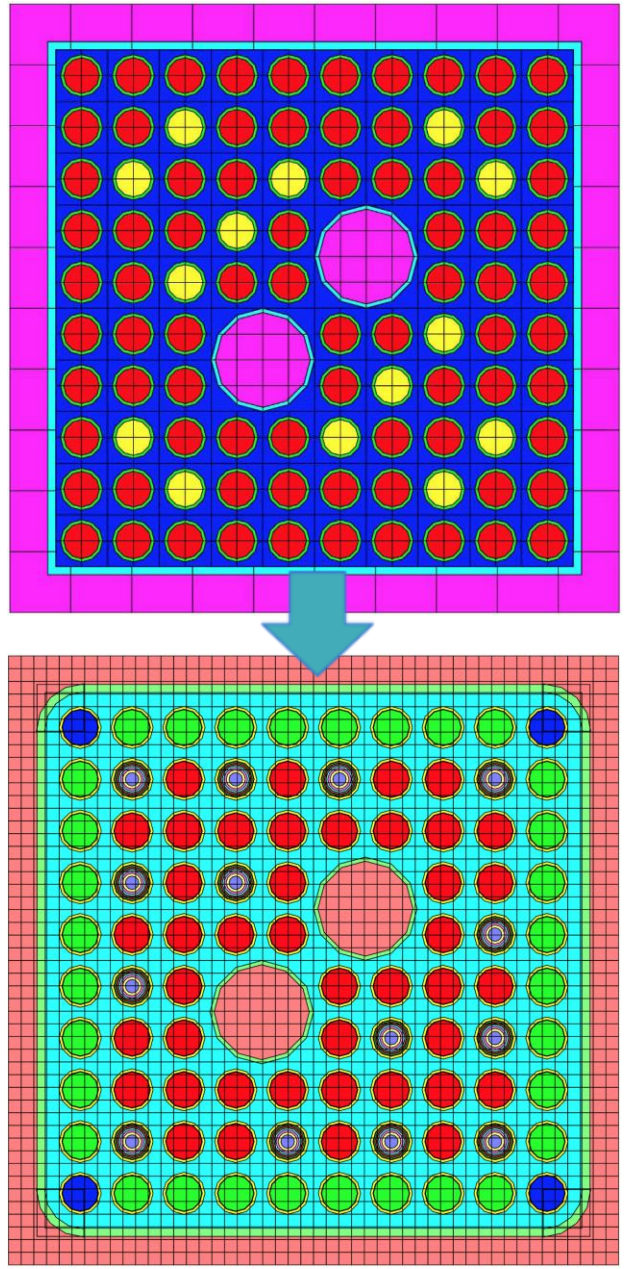


- Available with SCALE 6.3 for:
  - Pressurized water reactors (PWRs)
  - Boiling water reactors (BWRs)
  - Mixed-oxide (MOX) for typical BWRs and PWRs
  - Russian VVER reactors
  - Russian RBMK reactors
  - Canadian CANDU reactors
  - UK advanced gas reactors (AGRs)
  - MAGNOX reactors
  - IRT reactors

- Other reactor libraries can be generated by ORNL or by the user, provided sufficient design information is available

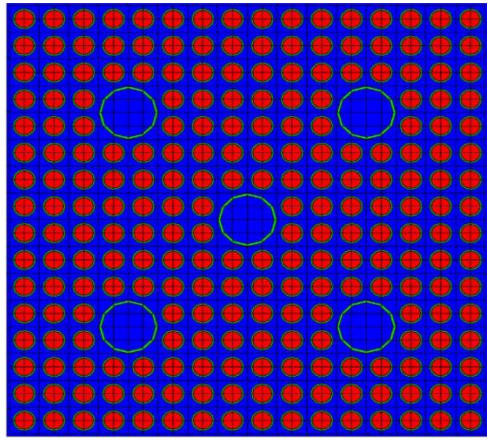
ORIGEN reactor libraries – SCALE 6.3

- Automation tools allow reactor libraries to be generated with each release of SCALE
- Generation of reactor libraries is based on ORNL modeling best practices
  - 252-group ENDF/B-VII.1 cross section library for the neutron transport solution
  - CENTRM cross section processing for the NEWT transport solver
  - Specialized Dancoff factors for BWR corner and edge pins
  - Adequate spatial solution grid
  - Sources of information provided
  - MOX assembly models with typical radial zoning
- New in SCALE-6.3
  - Extended enrichments (up to 8.5 wt % ^{235}U) and burnups (up to 82,500 MWd/MTU) for all PWR and BWR libraries

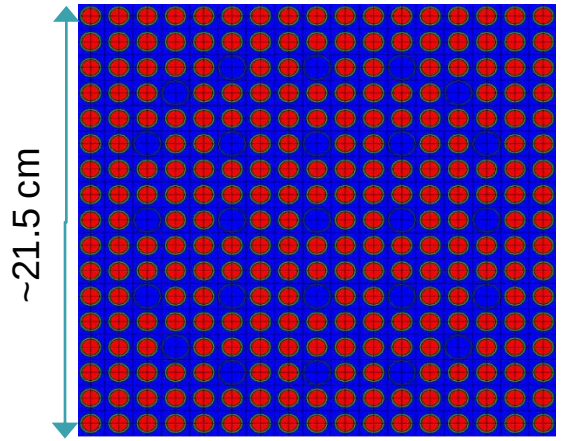


ORIGEN reactor libraries for PWRs

- PWRs operate under high pressure so that water does not boil
- Moderator (water) density is relatively constant in the reactor, ~0.7 g/cm^3
- Coolant contains boron to control the reactor, 0–2500 ppm, depending on the time in the cycle
- Most reactors in the United States use 17×17 fuel assemblies



CE 16x16



W 17x17

ORIGEN PWR libraries in SCALE 6.3



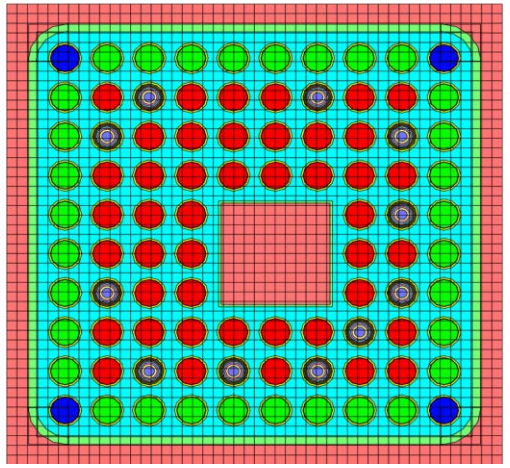
Assembly Design	Library Name	Enrichment (wt% ^{235}U)	Density (g/cc)
Westinghouse CE 14×14	ce14x14	0.5 – 8.5	0.7332
Westinghouse CE 16×16	ce16x16	0.5 – 8.5	0.7100
Westinghouse 14×14	w14x14	0.5 – 8.5	0.7264
Siemens 14×14	s14x14	0.5 – 8.5	0.7283
Siemens 18×18	s18x18	0.5 – 8.5	0.7283
Westinghouse 15×15	w15x15	0.5 – 8.5	0.7135
Westinghouse 17×17	w17x17	0.5 – 8.5	0.6700
Westinghouse 17×17 OFA	w17x17_ofa	0.5 – 8.5	0.6700

These libraries are based on 2D TRITON models (ENDF/B-VII.1 cross sections) with burnups to 82.5 GWd/MTU.

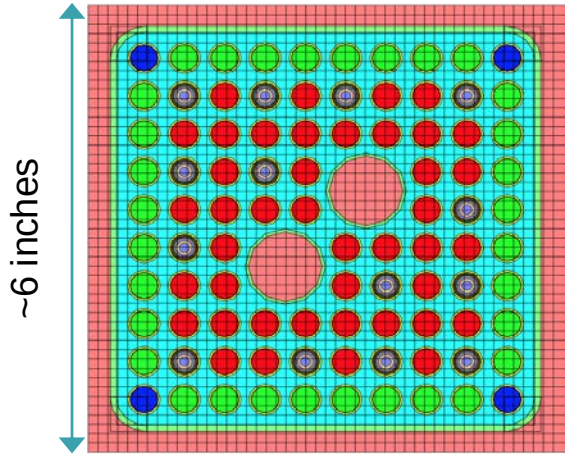
ORIGEN reactor libraries for BWRs



- BWRs operate at lower pressures (compared to PWRs)
- Water inside the channel boils, whereas water outside the channel remains at a relatively constant density
- Moderator density changes have a significant effect on the cross sections and, thus, fuel nuclide inventory
- BWRs typically contain many burnable poison rods (Gd_2O_3)
- Fuel enrichment and burnable absorber content can vary radially and axially
- GE14 (10×10) is the most common type of fuel in the United States; however, there are significantly more variations than are found in typical PWRs



ATRIUM 10



GE 10x10

ORIGEN BWR libraries in SCALE 6.3



Assembly Design	Library Name	Enrichment (wt% U-235)	Density (g/cc)
GE 7×7	ge7x7-0	0.5 – 8.5	0.1 – 0.9
GE 8×8	ge8x8-1, ge8x8-2	0.5 – 8.5	0.1 – 0.9
ABB 8×8	abb8x8-1	0.5 – 8.5	0.1 – 0.9
GE 9×9	ge9x9-2	0.5 – 8.5	0.1 – 0.9
GE 10×10	ge10x10-8	0.5 – 8.5	0.1 – 0.9
ATRIUM-9	atrium9x9-9	0.5 – 8.5	0.1 – 0.9
ATRIUM-10	atrium10x10-9	0.5 – 8.5	0.1 – 0.9
SVEA-64	svea64-1	0.5 – 8.5	0.1 – 0.9
SVEA-100	svea100-0	0.5 – 8.5	0.1 – 0.9
SVEA-96	svea96-1	0.5 – 8.5	0.1 – 0.9

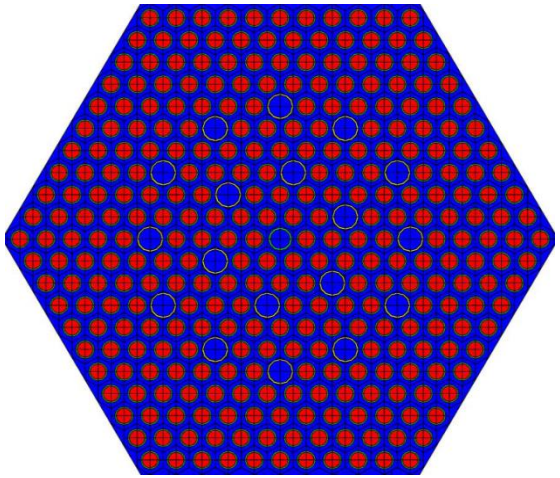
These libraries are based on 2-D TRITON models (ENDF/B-VII.1 cross sections) with burnups to 82.5 GWd/MTU.

ORIGEN VVER libraries in SCALE 6.3

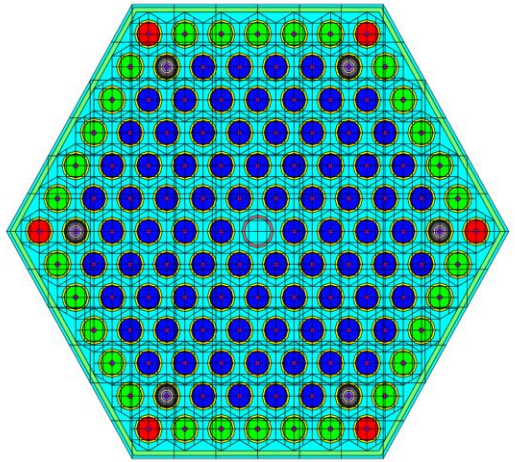
Libraries

Assembly Design	Library Name	Enrichment (wt% U-235)	Density (g/cc)
VVER-440 flat	vver440	1.6 – 3.6	0.73
VVER-440 profiled	vver440_3.82	3.82 average	0.73
VVER-440 profiled	vver440_4.25	4.25 average	0.73
VVER-440 profiled	vver440_4.38	4.38 average	0.73
VVER-1000	vver1000	0.5 – 8.5	0.7145

These libraries are based on 2D TRITON models (ENDF/B-VII.1 cross sections) with burnups to 82.5 GWd/MTU.

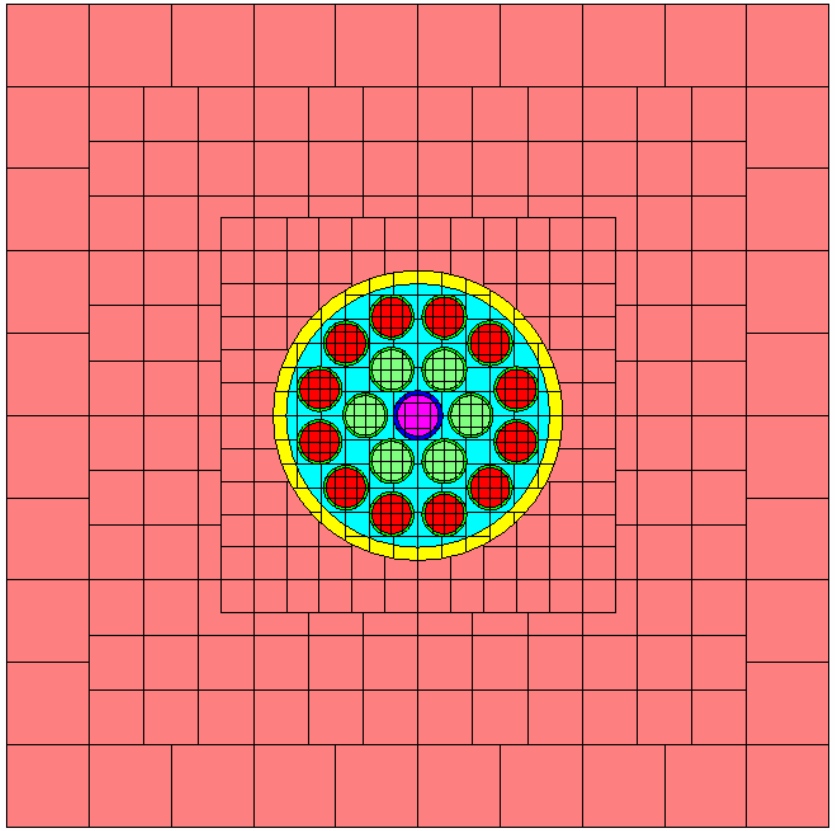


VVER-1000



VVER-440

ORIGEN RBMK libraries in SCALE 6.3



- Libraries generated based on design data representative of the Chernobyl Unit 4 reactor
- Enrichment: 1.8, 2.0, 2.2, 2.6, and 3.0 wt% ^{235}U
- Coolant density: 0.15, 0.28, 0.41, 0.54, 0.67, and 0.80 g/cm3
- Burnup: 0 to 24.375 GWd/MTU

ORIGEN reactor libraries for MOX fuel

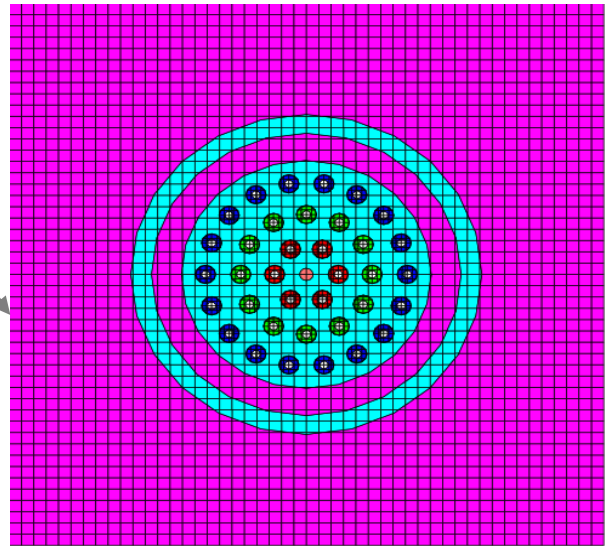


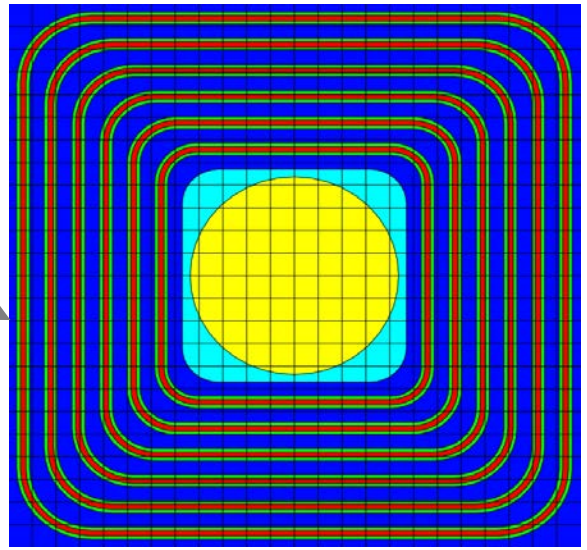
- MOX fuel PWR and BWR assemblies
  - PWR 14×14, 15×15, 16×16, 17×17, 18×18
  - BWR 8×8, 9×9, 10×10
  - Variable moderator density for all BWR MOX designs

Reactor Type	Lattice Type	No. of MOX Rods	No. of UO_2/Gd Rods	Fuel Pitch (cm)	Pellet diameter (mm)	Clad diameter (cm)	Assembly pitch (cm)
PWR	18 x 18	300	-	1.27	8.05	9.5	23
	17 x 17	264	-	1.26	8.19	9.5	21.5
	16 x 16	236	-	1.43	9.11	10.75	23
	15 x 15	205	-	1.43	9.13	10.75	21.6
	14 x 14	179	-	1.41	9.29	10.72	19.8
BWR	10 x 10 - 9	81	10	1.3	8.7	10.05	15.2
	9 x 9 - 9	60	12	1.43	9.5	10.8	15.2
	9 x 9 - 1	68	12	1.43	9.5	10.8	15.2
	8 x 8 - 2	62	14	1.63	10.3	12.3	15.2

Other ORIGEN reactor libraries in SCALE 6.3

- CANDU – natural UO_2, heavy water
- MAGNOX – natural U metal
- AGR
- IRT – research reactor fuel





OAK RIDGE National Laboratory

The future format of ORIGEN reactor libraries:

"Archive" format for multi-dimensional libraries

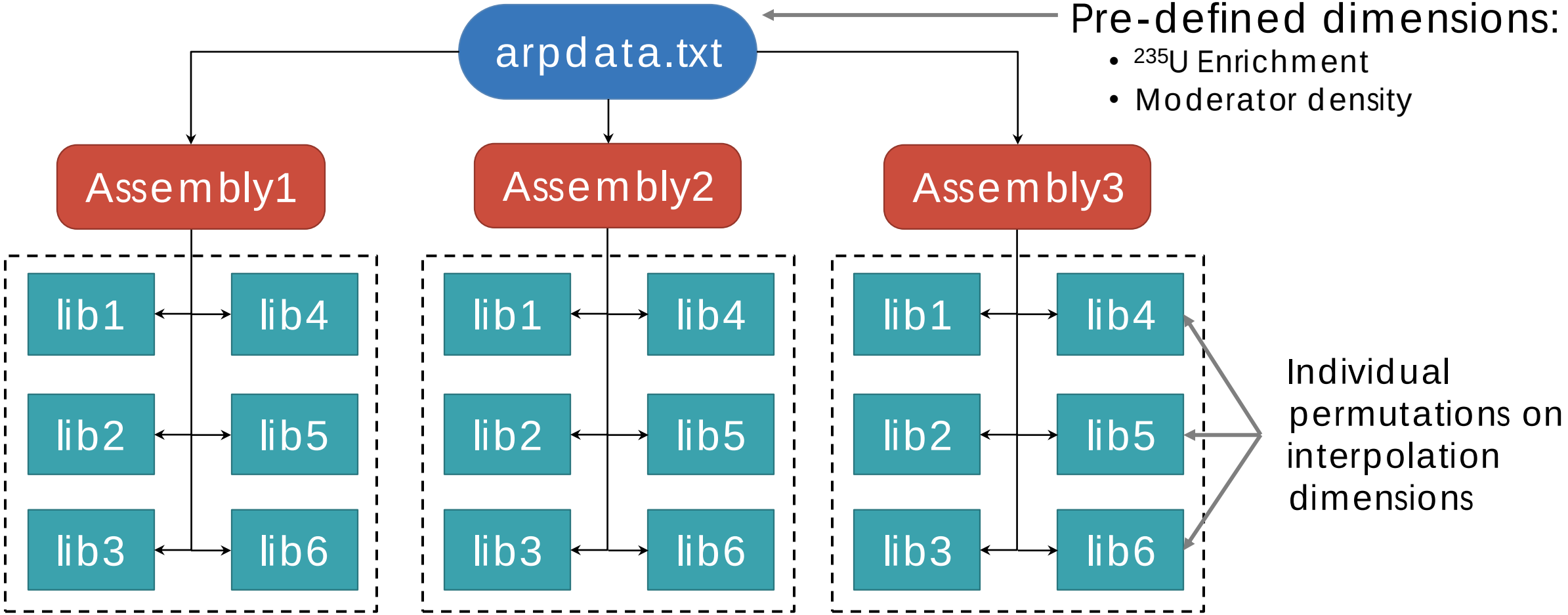


ORIGEN "Archive" format

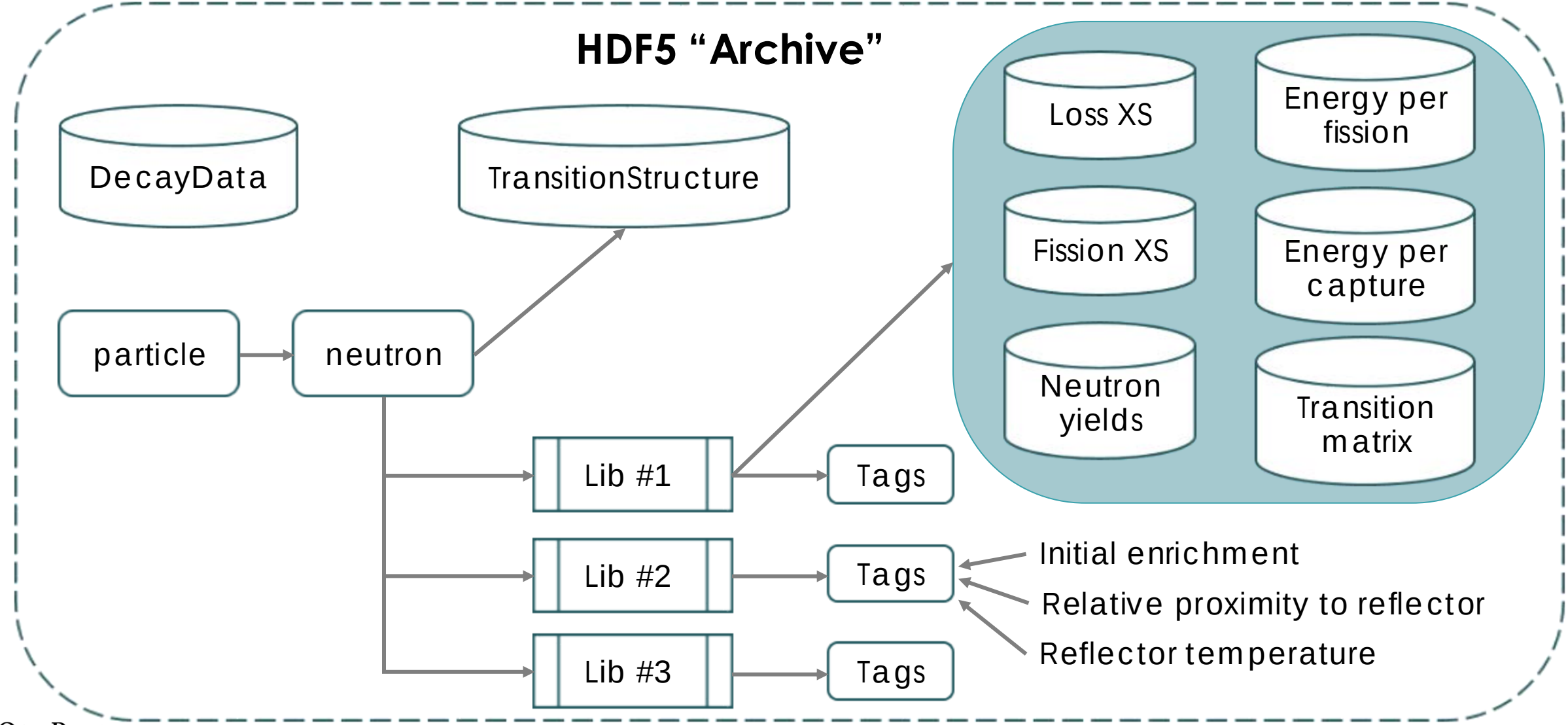
- ORIGEN and ORIGAMI in SCALE 7.0 will rely on HDF5-formatted "Archives" with descriptive interpolation data written on the file
- Flexible Archive data format allows for run-time specification & checking of interpolation parameters
- OBIWAN in SCALE 6.3.1 already supports conversion of libraries to the "archive" format
- Users can convert existing LWR libraries in `arpdata.txt` for use with ORIGAMI

```
obiwan convert –format=hdf5 –alias=[lib_alias][path_to_arpdata.txt]
```

ORIGEN reactor library interpolation relies on an ASCII database with hard-coded interpolation dimensions

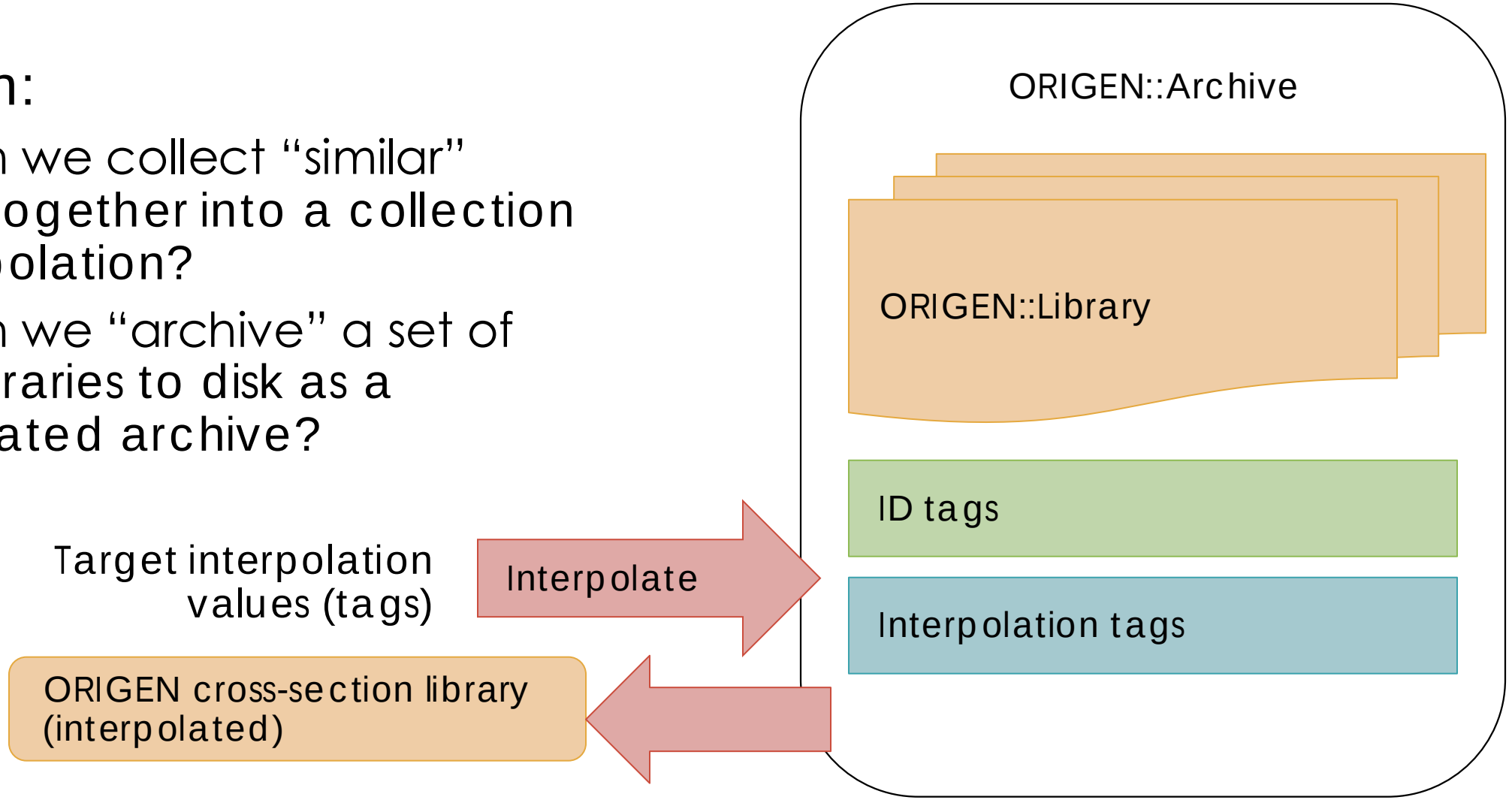


New HDF5-based "Archive" format is designed to accommodate arbitrary interpolation dimensions



New ORIGEN "Archive" format will pair with HDF5 to facilitate consolidated storage & interpolation

- Motivation:
  - How can we collect "similar" libraries together into a collection for interpolation?
  - How can we "archive" a set of similar libraries to disk as a consolidated archive?



OAK RIDGE National Laboratory

ORIGAMI Introduction

for rapid fuel depletion calculations

SCALE 6.3.2

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals

- Introduce users to the ORIGAMI interface to ORIGEN
- Demonstrate how it can be used to perform easy spent fuel calculations for simple irradiation histories using the UO2 express form
- Understand the similarities and differences in the ORIGAMI workflow compared to using ARP+ORIGEN

OAK RIDGE National Laboratory

Introduction to ORIGAMI



What is ORIGAMI?

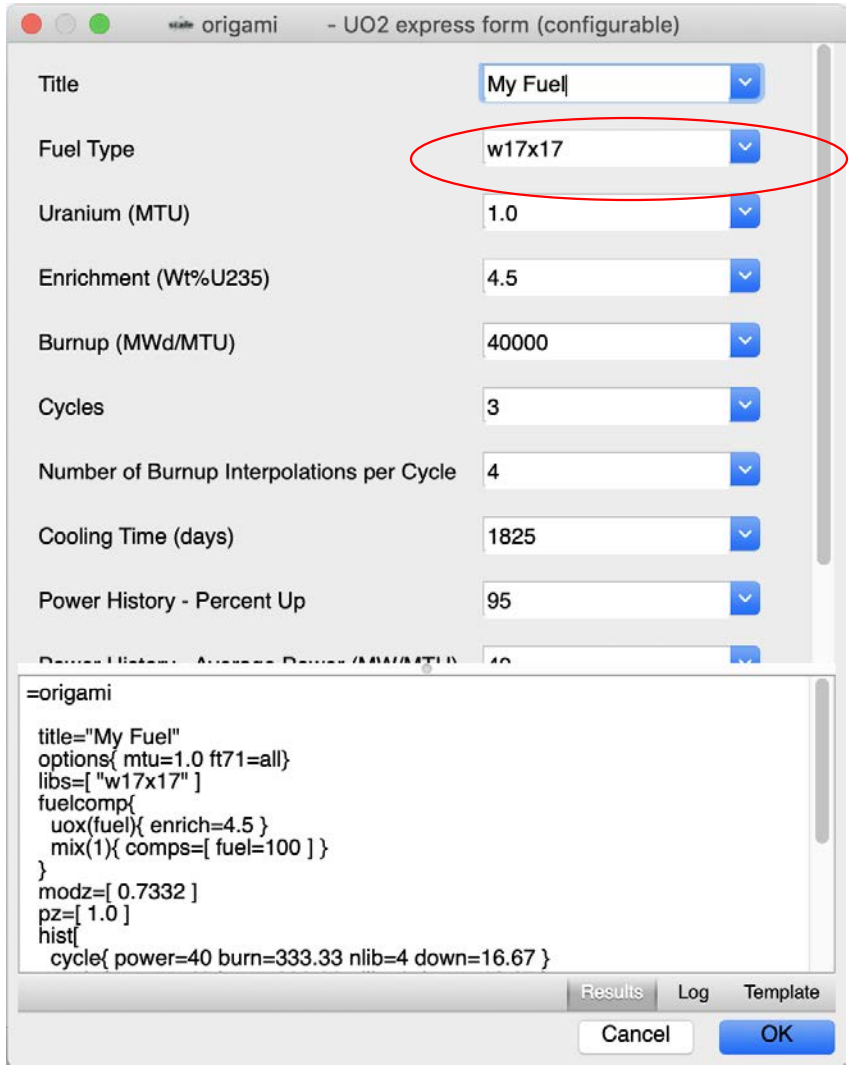
More details:

Section 5.4 of the SCALE manual

- ORIGen AsseMbly Isotopics
- Modern tool for easy depletion/decay calculations for LWR fuel assemblies, using ORIGEN with pre-generated ORIGEN reactor libraries
- Provides detailed nuclide inventories and radiation sources for LWR assemblies at desired complexity levels – 0D, 1D, 2D, 3D
- Calculations are performed based on the ORIGEN-ARP methodology
  - ORIGEN to solve the depletion/decay problem
  - ARP to create a specific library by interpolating on available ORIGEN reactor libraries

ORIGAMI enables rapid fuel inventory calculations

- The ORIGAMI sequence was released in SCALE 6.2 (2016) as modern approach to using ORIGEN reactor libraries
- ORNL has developed the ORIGEN+ARP methodology with the ORIGEN code to rapidly generate nuclide inventories using ORIGEN reactor libraries
- ORIGEN use of fundamental nuclear data allows the following metrics to be calculated from nuclide inventory
  - Mass
  - Decay heat
  - Activity
  - Gamma emission
  - Neutron emission



Why use ORIGAMI?

- Get rapid answers to common questions, such as

  What U/Cs/Pu content could I expect in a PWR fuel?

  a. assuming constant power?
  b. during a power maneuver?
  c. after 1 cycle of irradiation?
  d. after 100 days of cooling?
  e. after 100 years of cooling?
  f. at 50 GWd/MTU burnup?
  g. at 3.55% enrichment?

Each answer requires a <10 second calc. on a single CPU

Why is speed important? All back-end analysis can use this approach: dry storage casks, on-site storage, discharge inventory analysis, transportation packages

- Up-front work required
  - Physics analysis of the reactor system to understand the changes that impact neutron flux spectrum in the fuel (e.g., moderator density in BWR)
  - Run many CPU-hours of coupled transport+depletion cases to generate a database of 1-group cross sections σ which can be interpolated to a specific state of interest (ORIGEN reactor library)
  - These generated libraries can be used later (in ORIGAMI) to determine nuclide inventory and reaction rates:
    $RR(t) = \sigma(t)\, N(t)\, \phi(t)$
  - Why do it this way?
    If σ is insensitive to decay time, power level, then questions b through g can be answered based on a single TRITON depletion pre-calculation!

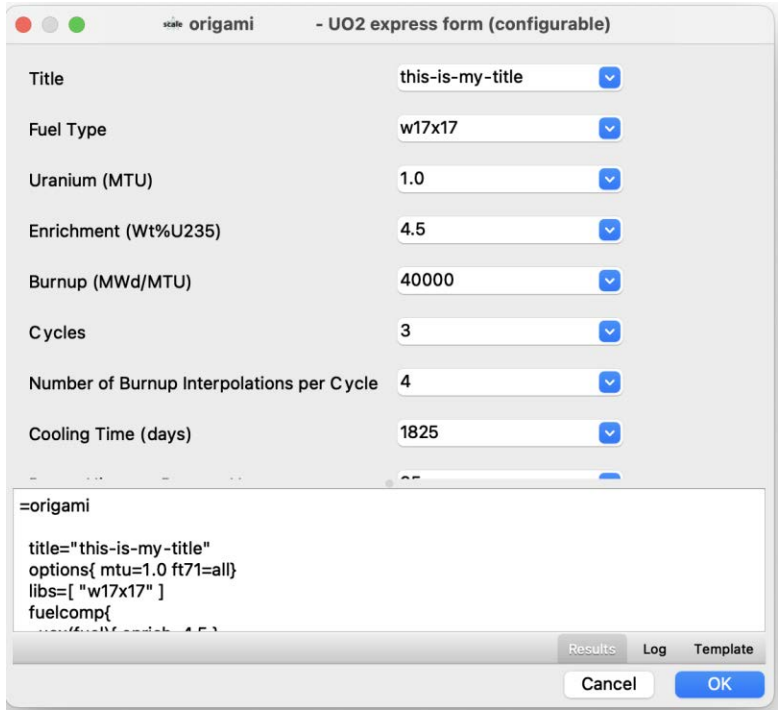
ORIGAMI capabilities highlights

With a narrower scope, many convenient features

- Short, concise interface to ORIGEN+ARP for easy spent fuel calculations
- Good default parameter settings for LWR UO_2
- Allows axial and radial variations to be investigated (e.g., axial decay heat in a BWR assembly)
- Produces F71 files automatically
  - Can be readily viewed in Fulcrum
  - *Used to seed an ORIGEN "restart" decay calculation
  - Can be post-processed with OPUS
- Can produce nuclide compositions
  - SCALE StdCmp format for nuclides important to burnup credit (e.g., for use in SCALE criticality models of transportation casks)
  - MCNP material cards format (e.g., for NDA detector models simulations)

ORIGAMI has two modes for user input

UO2 Express form configurable

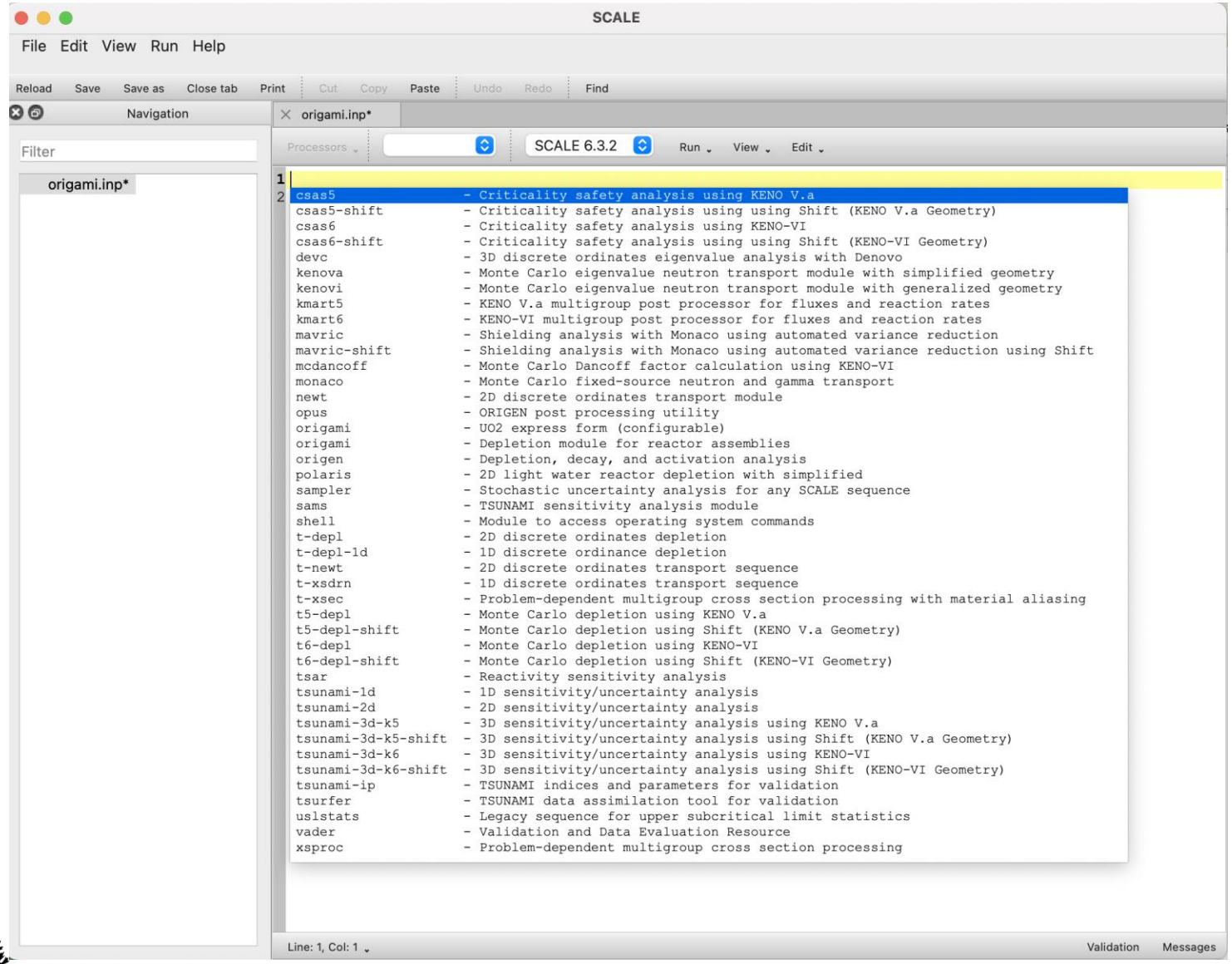


Detailed input

keyword-input with the SCALE Object Notation (SON) syntax

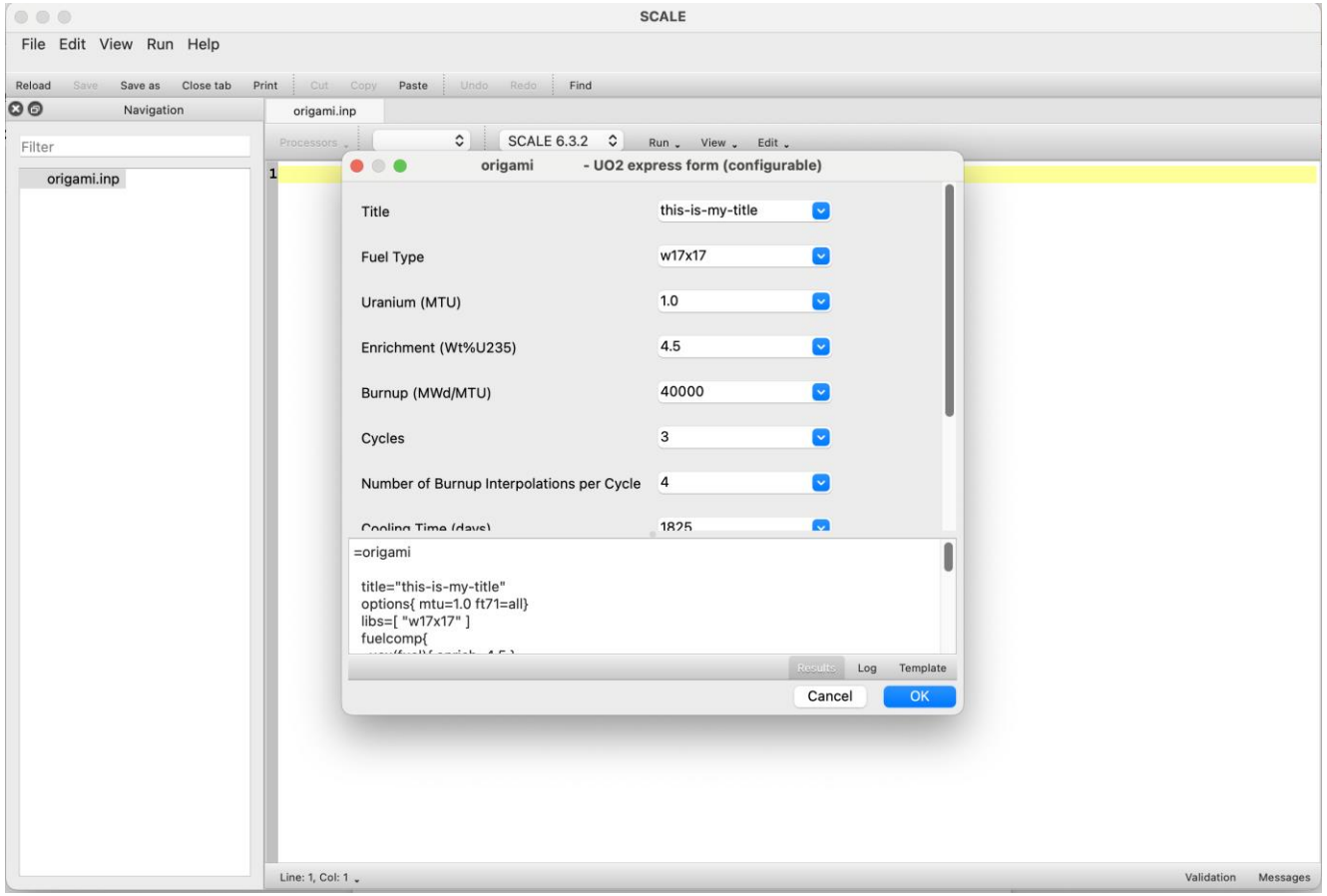
```
=origami
title='fully lumped assembly model'
options{ mtu=0.38 ft71=all }
libs=[ ce14x14 ]
fuelcomp{
    stdcomp(fuel){
    base=uo2 iso[92234=0.02848 92235=3.2
                 92236=0.01472 92238=96.7568] }
    mix(1){ comps[fuel=100] }
}
nonfuel=[ cr=3.366 mn=0.1525 fe=6.309 co=0.0302
        ni=2.366 zr=516.3 sn=8.412 gd=2.860 ]
hist[
    cycle{ power=40 burn=284 nlib=4 down=54 }
    cycle{ power=38.6 burn=300 nlib=4 down=28 }
    cycle{ power=25.2 burn=250 nlib=3 down=30 }
]
print{
    nuc {
    sublibs=[ac fp]
    units=[grams moles]
    total=no }
}
end
```

ORIGAMI "Express Form" – How to start ?



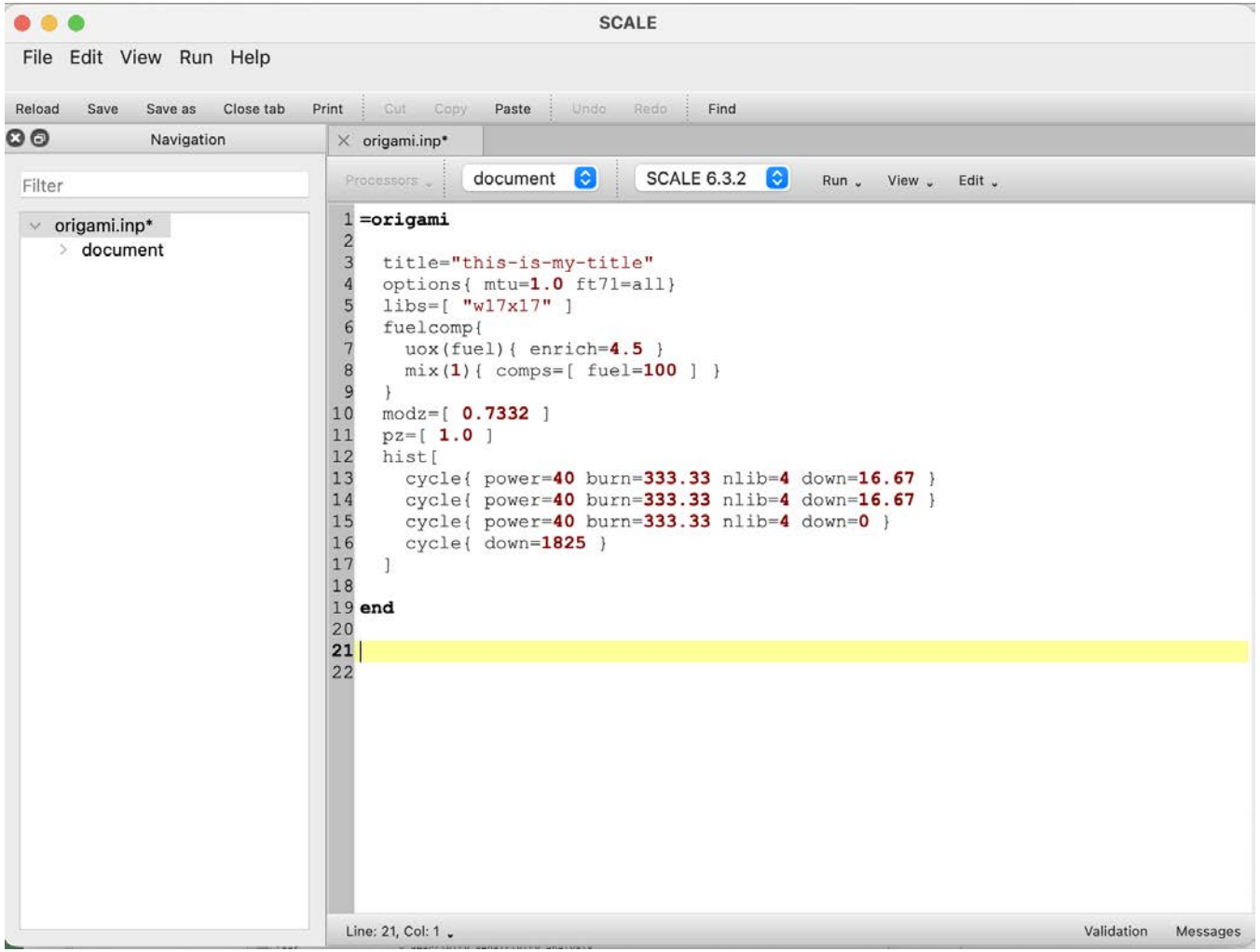
- Create empty file (origami.inp)
- CTRL+SPACE inside empty file
- Choose "origami – UO2 express form (configurable)

ORIGAMI "Express Form" – How to use ?



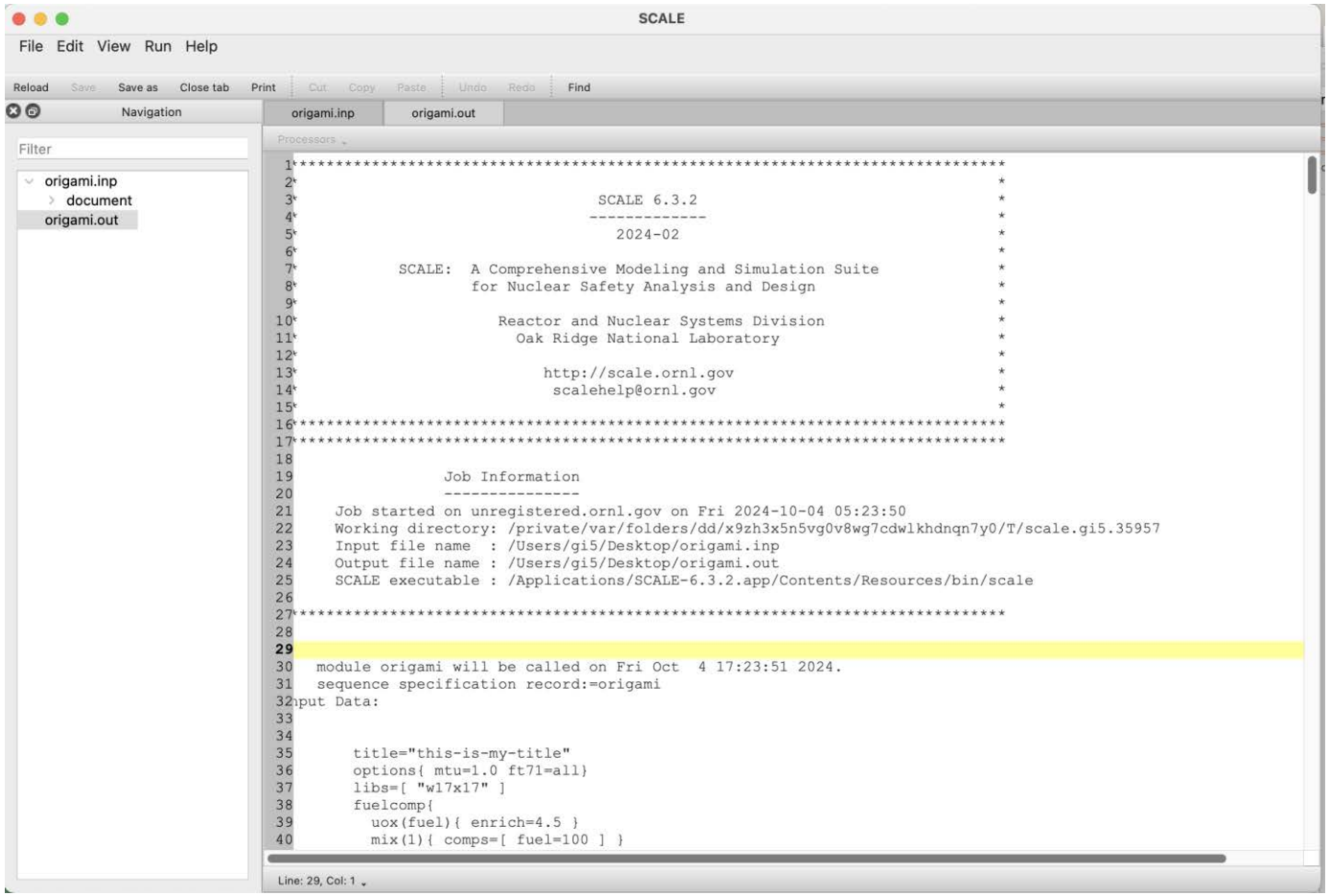
- Configurable window appears
- Click "OK"

ORIGAMI "Express Form" – Input behind the form



- This is a valid ORIGAMI input!
- Click Run

ORIGAMI "Express Form" – Execution

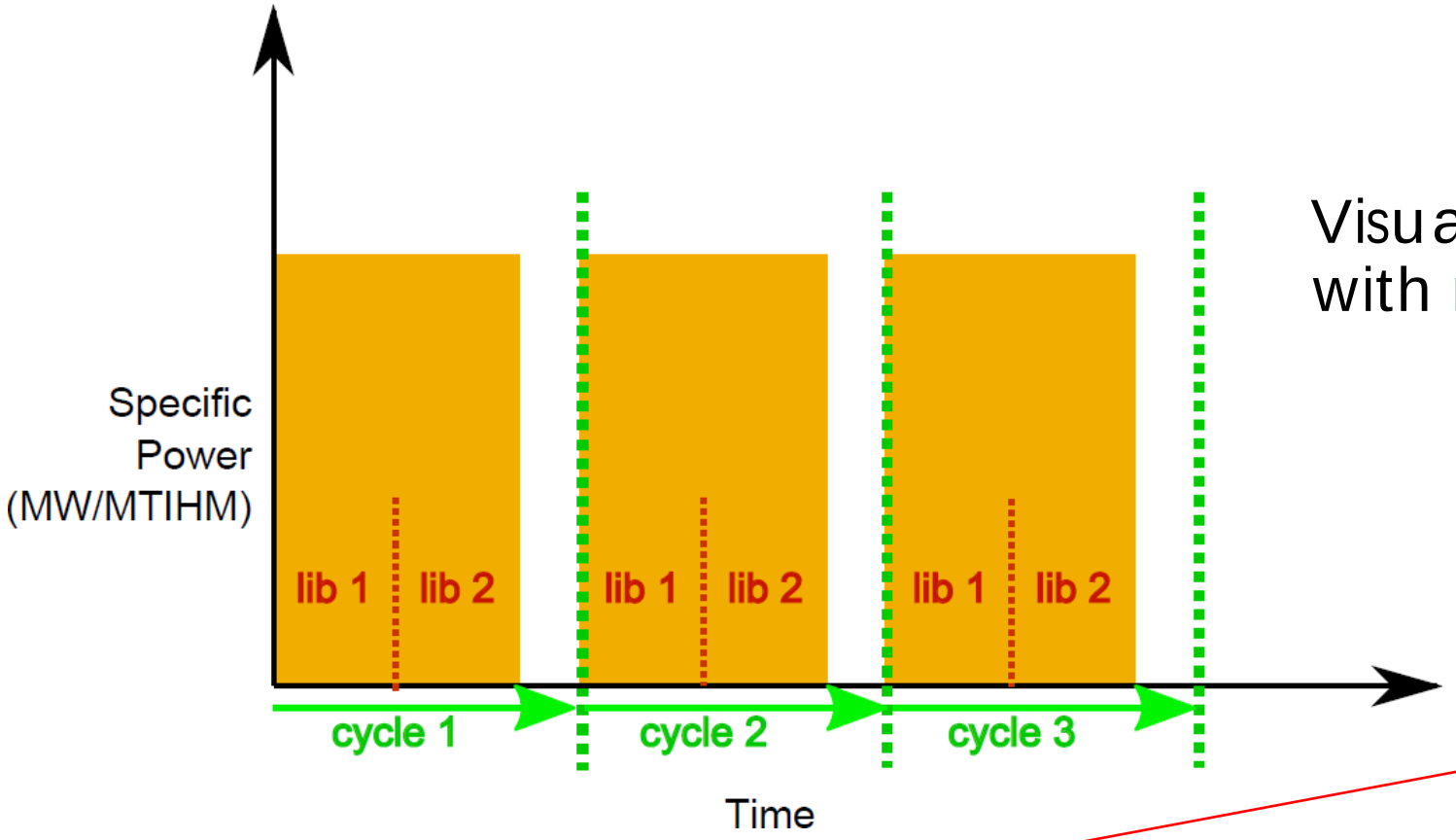


ORIGAMI takes a little while…

- Performs 10 substeps by default per depletion / decay step
- nlib=4 would require 4 cases in ORIGEN
- The input on previous page involves (4+4+4+1)*10 = 130 depletion solves

Controlling cross section interpolation

nlib (libraries per cycle)



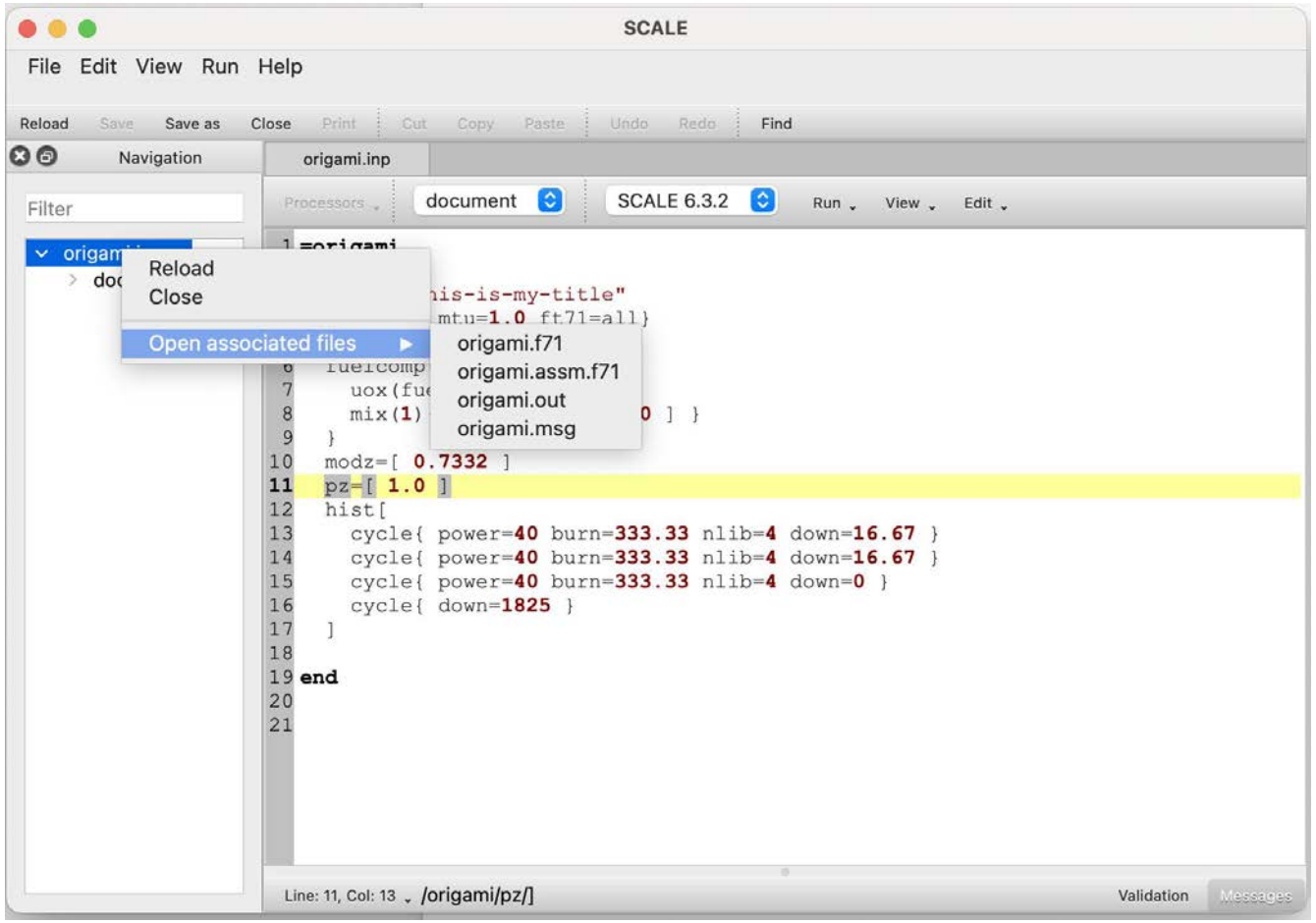
Visualization of 3 cycles with nlib=2

in express form

in generated input

```
Number of Burnup Interpolations per Cycle 4
cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
```

ORIGAMI "Express Form" – F71 files produced



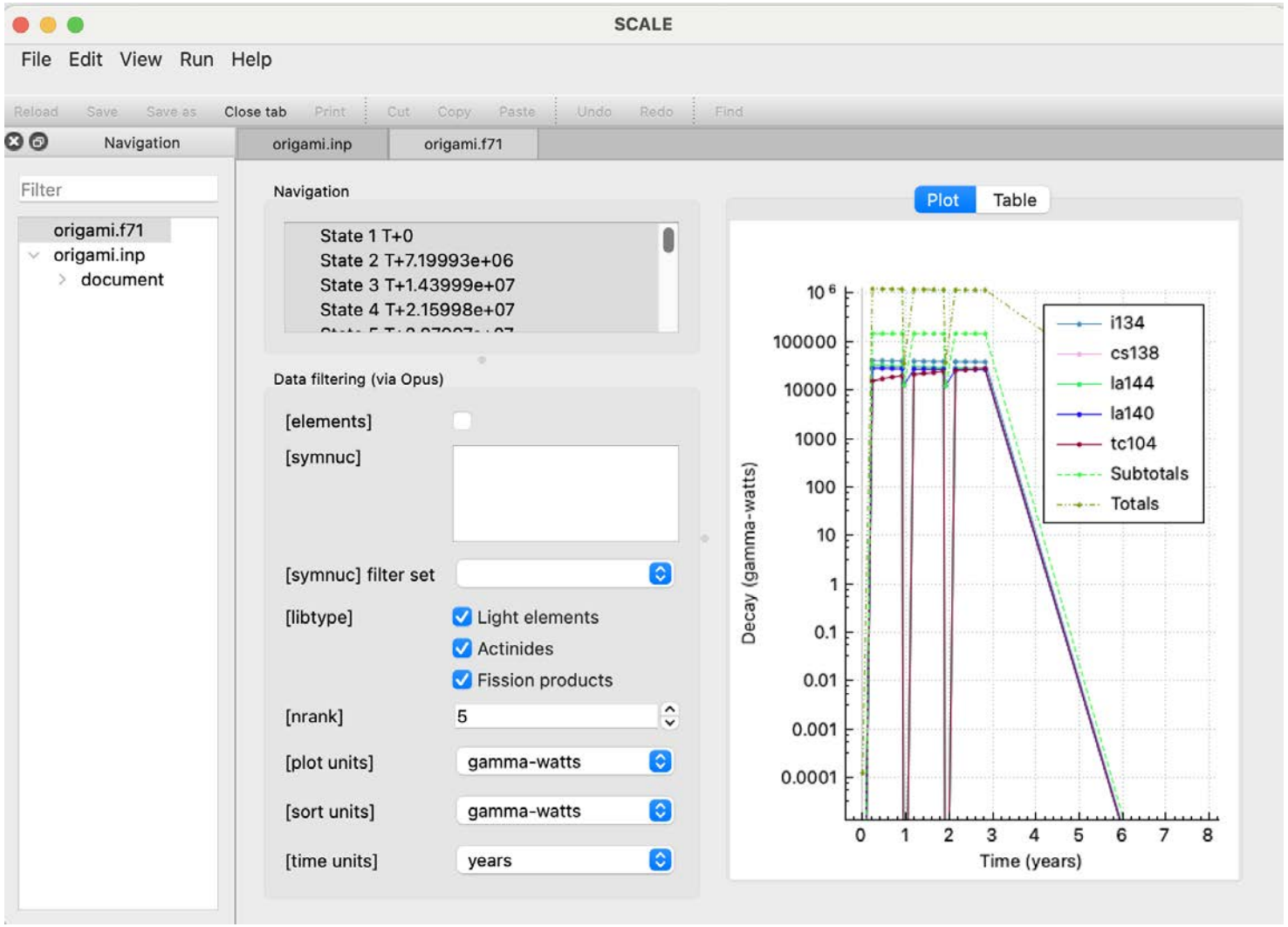
- ORIGAMI produces by default two F71 files:

  ${basename}.f71

  ${basename}.assm.f71

- How to open .f71
  - Right-click on origami.inp in navigation pane
  - Open associated files
  - Select origami.f71

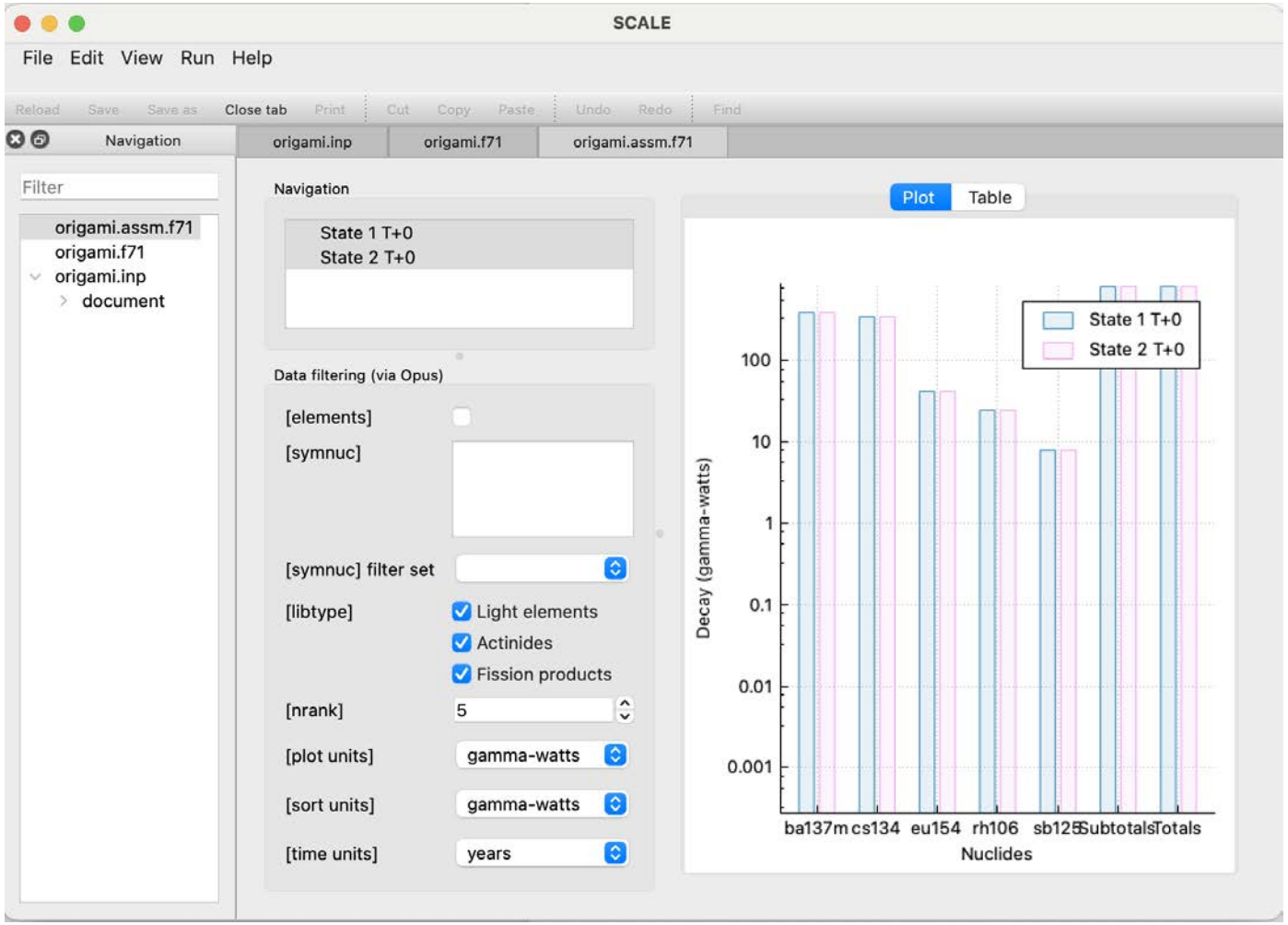
ORIGAMI "Express Form" – F71 files coverage



- ${basename}.f71

Has data for all times in the problem

ORIGAMI "Express Form" – F71 files coverage



- ${basename}.assm.f71 has data only for final time in the problem
- Both States 1 and 2 correspond to cooling time input in the form

ORIGAMI-UOX Express: Ringhals-2 Sample F32

- Open the ORIGAMI UOX Express form
- Populate the average cycle history parameters and run
- Use the w15x15 library

- 3.197 wt. % enrichment
- 436.99 kgU
- 50,962 MWd/MTU burnup
- 5,860 days cooling time
- Average power: 27.91 MW/MTU

Cycle	Burnup (MWd/MTU)	Cycle length (d)	Down time (d)	Avg. power (MW/MTU)
1	10,553	290	80	36.39
2	21,162	317	84	33.47
3	29,553	273	91	30.74
4	37,314	320	161	24.25
5	43,943	297	71	22.32
6	50,962	329	—	21.33

How accurate is ORIGAMI for calculating decay heat?

Measured power: 692 W

Method	Decay power (W)	C/E – 1 (%)
ORIGAMI (average history)	720.8	4.2
ORIGAMI (detailed history)	715.3	3.4
ORIGAMI (avg. with custom library)	706.7	2.1
ORIGAMI (detailed with custom library)	701.5	1.4

Generic reactor data library (w15x15)

Design-specific (F32) reactor data library

Notes on the axial effect

- Depletion is non-linear with respect to cross sections and decay constants, i.e. expected/average values

$$\left\langle e^{-\lambda t} \right\rangle \neq e^{-\langle \lambda \rangle t}$$
$$\langle \lambda t + c \rangle = \langle \lambda \rangle t + \langle c \rangle$$

- Therefore, assembly-average decay heat calculated from a single "average" ORIGAMI simulation is NOT the same as the average of either
  - Axial-dependent simulation
  - Pin-by-pin simulation
- Luckily the effect appears to be 1-2%

Although detailed axial calculations may not be practical for routine decay heat analysis, and this effect is relatively small (the magnitude in many cases being smaller than the experimental uncertainties), the user needs to be aware of the bias introduced in assembly models that use average data instead of detailed axial data. Importantly, it can be seen from Figure 6.17 that there is no single average value for the assembly burnup (or coolant density) that will yield results that are equivalent to a detailed axial calculation.

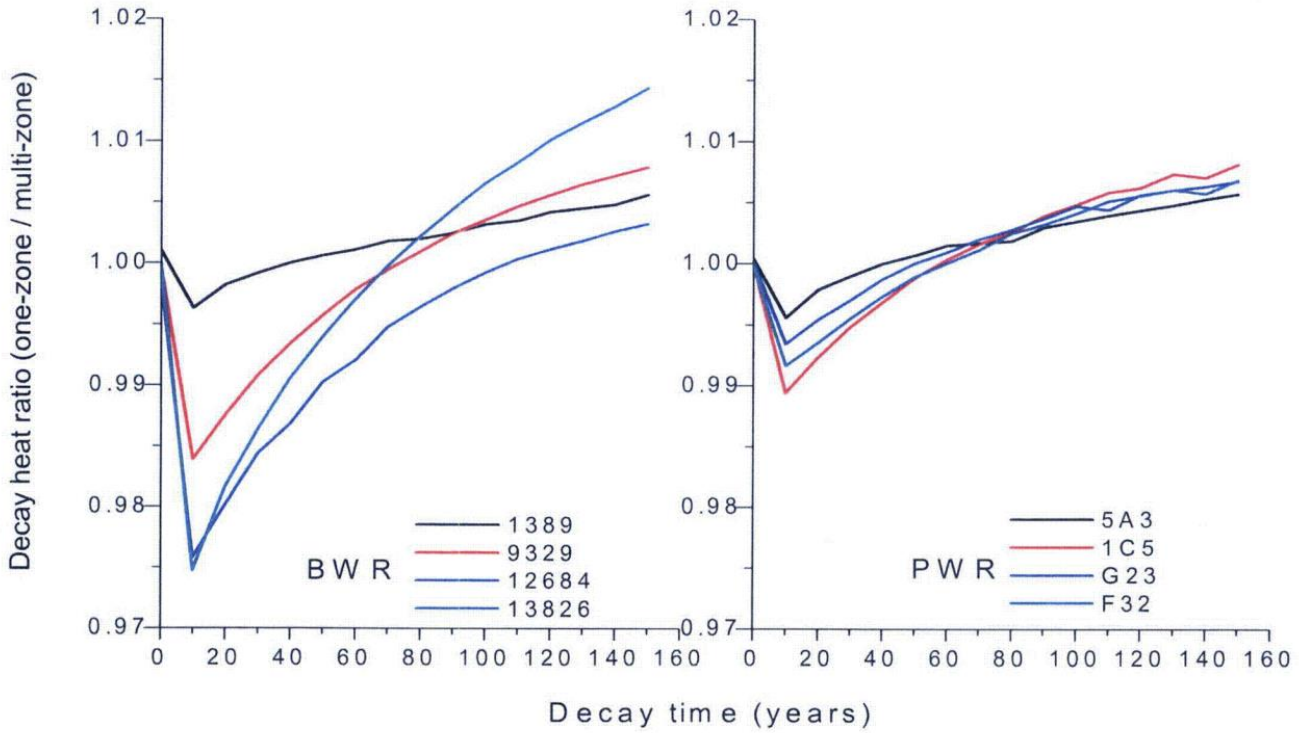


Figure 6.17 Ratio of decay heat calculated using assembly average parameters to that obtained using explicit axial calculations for selected Swedish BWR and PWR assemblies

Steps to fast inventory calcs with ORIGAMI

Developer Steps

1. Sensitivity studies are performed with TRITON (or Polaris if LWR) to understand different conditions important for representing the neutron energy spectrum as a function of burnup/time in a particular fuel system. Examples for LWRs:
   - UO_2 PWR: assembly design, enrichment, burnup
   - UO_2 BWR: assembly design, enrichment, void fraction, burnup
   - MOX PWR/BWR: same as UO_2 but Pu fraction + Pu fissile fraction instead of enrichment
2. ORIGEN Reactor Libraries are generated on a grid, e.g. for a specific assembly design, 10 enrichments, 3 moderator densities, 30 burnups → 10x3x30 grid of 1-group cross section data
3. ORIGEN Reactor Libraries are "registered" in a special text file called arpdata.txt

Runtime Steps

4. Based on user input, ORIGAMI finds the specified ORIGEN reactor library and interpolates non-burnup dimensions, e.g. 4.3% enrichment, 0.735 g/cc moderator density. The result is a burnup-dependent set of 1-group cross section data, A(bu)
5. ORIGAMI then interpolates this A(bu) data on burnup according to the user-provided power, P(t) and repeatedly requests the ORIGEN kernel solver to compute the end-of-step nuclide inventory $n(bu_{i+1})$ based on $n(bu_i)$, $A(bu_{i+1/2})$
6. Finally ORIGAMI writes the f71 with requested nuclide inventory results

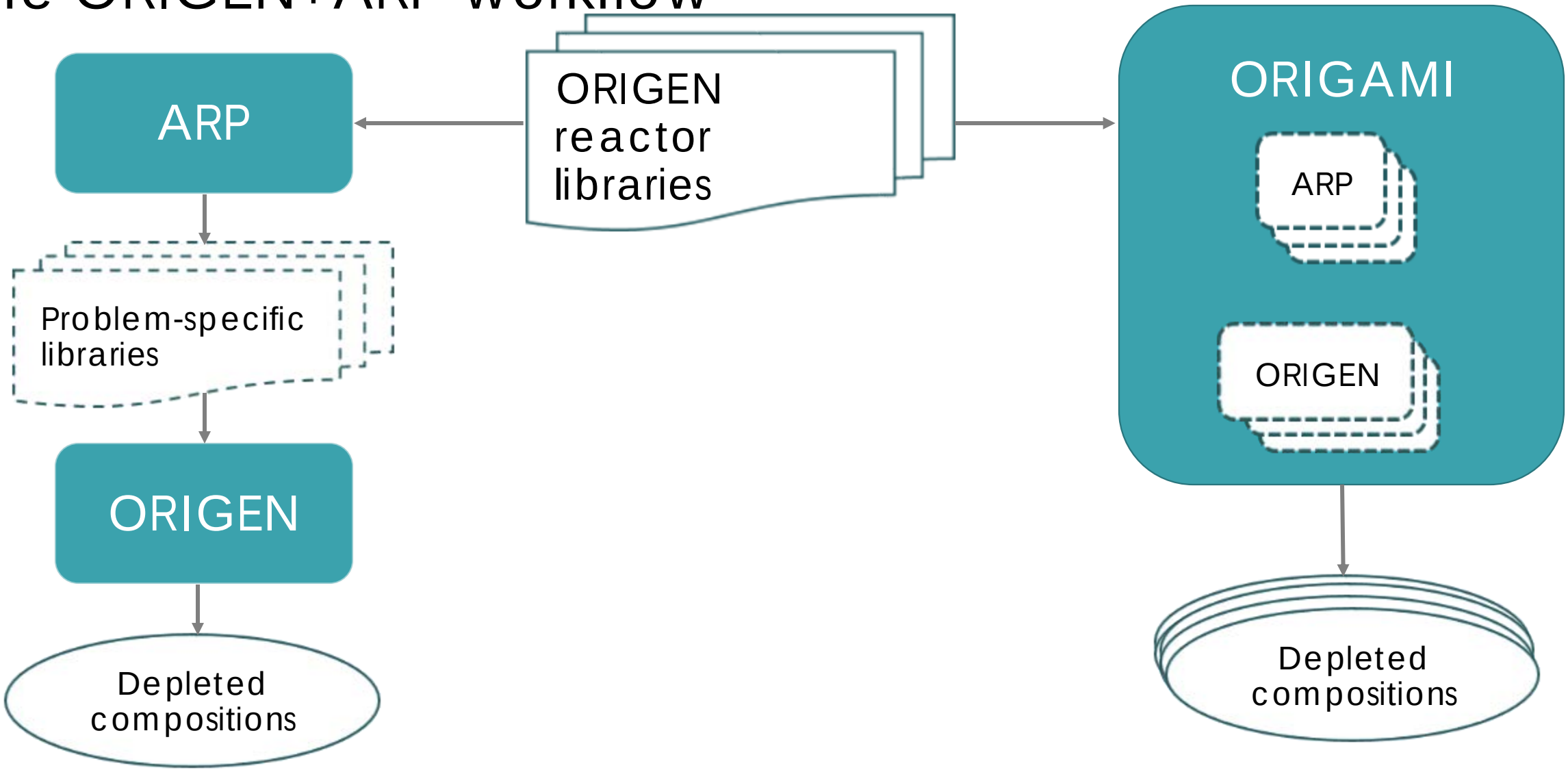
OAK RIDGE National Laboratory

ORIGAMI vs ORIGEN+ARP

Selecting the right tool for your workflow



ORIGAMI is a more specialized and compact version of the ORIGEN+ARP workflow



Which tool is right for me?

ARP + ORIGEN

- Depletion of one or a few regions
- Complex irradiation histories and/or advanced processing
- Any fuel type (incl. MOX, CANDU, etc.)

ORIGAMI

- Depletion of many regions / compositions (several depletion problems)
- Relatively simple irradiation histories
- Enables follow-on analysis using tools such as KENO and MCNP
- UOX only (presently)

OAK RIDGE National Laboratory

Tutorial:

Nuclide inventory calculation with ORIGAMI UO2 Express



Problem: Nuclide inventory in typical PWR fuel at 10 years cooling

- Required results:
  - Concentrations in g/MTU of important actinides and fission products (see table with nuclide list)
  - Total amount of Pu in g/MTU
  - Weight percentage of Pu-238, Pu-239, Pu-240 and Pu-241 relative to total Pu
- Input assumptions:
  - W17x17 reactor library
  - Initial fuel enrichment: 5.0% ^{235}U
  - Cooling time: 10 years
  - Number of reactor cycles: 3
  - Keep all other input default values in the form

Nuclide	g/MTU
U-235	
Pu-239	
Pu-240	
Cs-137	
Pu-241	
Am-241	
Pu-238	
Cm-244	
Gd-155	
Sm-149	
Cs-134	

Solution: Nuclide inventory in typical PWR fuel at 10 years cooling

Nuclide	g/MTU
U-235	16413.8
Pu-238	193.6
Pu-239	6504.6
Pu-240	2266.4
Pu-241	908.0
Am-241	603.2
Cm-244	18.3
Cs-134	5.1
Cs-137	1171.4
Sm-149	4.7
Gd-155	7.0

Nuclide	wt% Pu
Pu-238	1.9
Pu-239	63.0
Pu-240	22.0
Pu-241	8.8
Pu-242	4.3

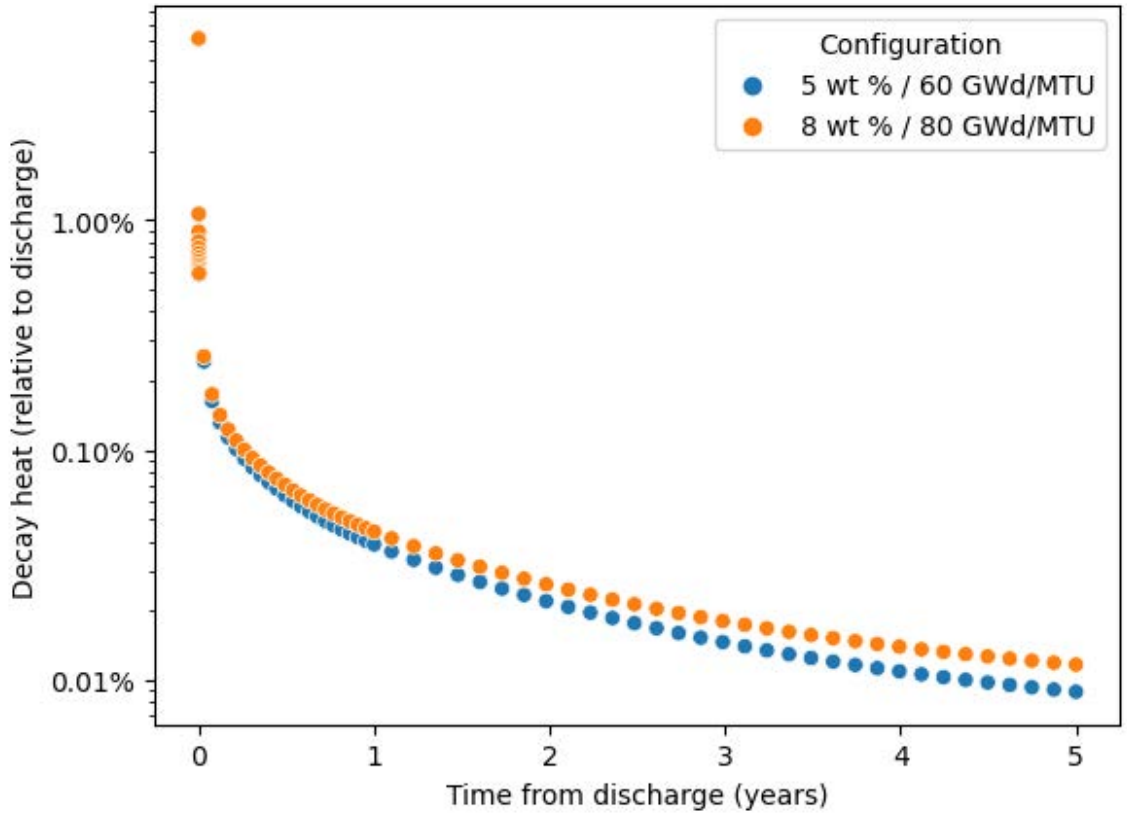
OAK RIDGE National Laboratory

Tutorial: Compare decay heat of LEU to LEU+/HBU



Goals

- With ORIGAMI, generate spent fuel inventory for W17x17 for two variants producing two f71 files
  - 40 MW/MTU, 5 wt% @60 GWd/MTU, w5-60.discharge.f71
  - 40 MW/MTU, 8 wt% @80 GWd/MTU, w8-80.discharge.f71
- Plot decay heat up to five years as % of full power
  - Import each f71 at discharge into a simple ORIGEN decay input producing two more f71 files
    - w5-60.decay.f71
    - w8-80.decay.f71
  - Load these f71 files in Fulcrum, change units to decay heat, copy/paste data from Table view into spreadsheet, reformat as needed



OAK RIDGE National Laboratory

Workshop:

Fast spent fuel calculations with ORIGAMI UO2 Express

Compare calculated key spent fuel metrics (activity, decay heat, nuclide concentrations) of typical PWR and BWR typical fuel



OAK RIDGE National Laboratory

Description of problems



Problem 1a: Decay heat and activity of PWR spent fuel (W17x17)at 10-year cooling time

- Input data:
  - 3.5% U-235 enrichment
  - 3 irradiation cycles
  - 45,000 MWd/MTU burnup
  - 10-year cooling after discharge
  - 90% uptime
  - 32 MW/MTU average power
- Results required:
  - Total decay heat (W/MTU)
  - Gamma component of decay heat (W/MTU)
  - Activity (Bq/MTU)

- Strategy:
  - Use ORIGAMI Express with "w17x17" reactor library; use defaults for non-specified input values
  - Can extract results using Fulcrum with F71 file generated by ORIGAMI, or by editing the generated ORIGAMI input to add OPUS blocks at the end of the input and rerun the case

Problem 1b: Decay heat and activity of BWR spent fuel (GE 10x10) at 10-year cooling time

- Input data:
  - 3.5% U-235 enrichment
  - 3 irradiation cycles
  - 45,000 MWd/MTU burnup
  - 10-year cooling after discharge
  - 90% uptime
  - 25 MW/MTU average power
  - 0.45 g/cm^3 moderator density
- Results required:
  - Total decay heat (W/MTU)
  - Gamma component of decay heat (W/MTU)
  - Activity (Bq/MTU)
- Strategy:
  - Use ORIGAMI Express with "ge10x10-8" reactor library; use defaults for non-specified input values
  - Can extract results using Fulcrum with F71 file generated by ORIGAMI, or by editing the generated ORIGAMI input to add OPUS blocks at the end of the input and rerun the case

Problem 1c: Compare decay heat and activity of typical PWR and BWR spent fuel at 45 GWd/MTU burnup and 10-year cooling

- Input data:
  - PWR assembly initial U load: 460 kg
  - BWR assembly initial U load: 180 kg
- Results required:
  - Total decay heat: difference in Watt and % (per MTU, and per assembly)
  - Gamma decay heat: difference in Watt and % (per MTU, and per assembly)
  - Activity: difference in Bq and % (per MTU, and per assembly)
- Strategy:
  - Use inputs and results from Problem 1a and Problem 1b

Problem 2a:
Nuclide inventories in once-, twice-, thrice-burned PWR fuel

Calculate mass data of selected U, Pu, Nd isotopes for a typical PWR W17x17 fuel irradiated in the reactor for 1, 2, and 3 cycles

- Input data:
  - Enrichment 4.5 wt% ^{235}U
  - Power 30 MW/MTU
  - Reactor library w17x17
  - Burnup 15, 30, and 45 GWd/MTU for 1-, 2-, and 3-cycles irradiation (3 cases)
  - Keep default values for everything else
- Strategy:
  - Start with ORIGAMI express form
  - Can extract the data using Fulcrum for the F71 file generated by ORIGAMI, or by editing the generated input to add OPUS blocks at the end of the input and rerun the case

- Results required:
  - U-235 mass (kg/MTU) at discharge
  - Percentage of initial U-235 that was burned during the irradiation
  - Pu mass (kg/MTU) at 0- and 10-year cooling time
  - Percentage of fissile Pu at 0- and 10-year cooling time (Pu-239+Pu-241)/Pu
  - Nd-148 mass (g/MTU) at discharge [note this is a burnup indicator fission product]

Problem 2b:
Nuclide inventories in once-, twice-, thrice-burned BWR fuel

Calculate mass data of selected U, Pu, Nd isotopes for a typical BWR 8x8 fuel irradiated in the reactor for 1, 2, and 3 cycles

- Input data:
  - Enrichment 4.5 wt% ^{235}U
  - Power 30 MW/MTU
  - Moderator density 0.45 g/cm^3
  - Burnup 15, 30, and 45 GWd/MTU for 1-, 2-, and 3-cycles irradiation (3 cases)
  - Assembly type ge8x8-1
  - Keep default values for everything else
- Strategy:
  - Start with ORIGAMI express form
  - Can extract data using Fulcrum with F71 file generated by ORIGAMI, or by editing the generated input to add OPUS blocks at the end of the input and rerun the case
- Results required:
  - U-235 mass (kg/MTU) at discharge
  - Percentage of initial U-235 that was burned during the irradiation
  - Pu mass (kg/MTU) at 0- and 10-year cooling time
  - Percentage of fissile Pu at 0- and 10-year cooling time (Pu-239+Pu-241)/Pu
  - Nd-148 mass (g/MTU) at discharge [note this is a burnup indicator fission product]

Problem 2c:
Nuclide inventories in once-, twice-, thrice-burned LWR fuel

Compare mass data of selected U, Pu, Nd isotopes for a typical PWR 17x17 fuel and a BWR 8x8 fuel irradiated in the reactor for 1, 2, and 3 cycles

- Input data:
  - For PWR fuel, as for Problem 2a
  - For BWR fuel, as for Problem 2b
- Strategy:
  - Use inputs or results from Problem 2a and Problem 2b

- Results required:
  - Compare U-235 mass (kg/MTU) at discharge
  - Compare percentage of initial U-235 that was burned during the irradiation
  - Compare Pu mass (kg/MTU) at 10-year cooling time
  - Compare percentage of fissile Pu at 10-year cooling time (Pu-239+Pu-241)/Pu
  - Compare Nd-148 mass (g/MTU) at discharge

OAK RIDGE National Laboratory

Solutions



Problem 1a: Decay heat and activity of PWR spent fuel (W17x17)at 10-year cooling time

- Total decay heat:

  1.650e+03 W/MTU at 10-year cooling

- Gamma component of decay heat:

  4.934e+02 W/MTU (29.9% of total) at 10-year cooling

- Activity:

  1.941E+16 Bq at 10-year cooling

Problem 1b: Decay heat and activity of BWR spent fuel (GE 10x10) at 10-year cooling time

- Total decay heat:
  1.586e+03 W/MTU at 10-year cooling
- Gamma component of decay heat:
  4.736e+02 W/MTU (29.9% of total) at 10-year cooling
- Activity:
  1.843E+16 Bq at 10-year cooling

Problem 1c: Compare decay heat and activity of typical PWR and BWR spent fuel at 35 GWd/MTU burnup and 10-year cooling

- Total decay heat (PWR vs. BWR) at 10-year cooling
  - 64.0 W/MTU (3.9%)
  - 473.7 W/assembly (62.4%)
- Gamma component of decay heat (PWR vs. BWR) at 10-year cooling
  - 19.8 W/MTU (4.0%)
  - 141.7 W/assembly (62.4%)
- Activity (PWR vs. BWR) at 10-year cooling
  - 9.8E+14 Bq/MTU (5.0%)
  - 5.6E+15 Bq/assembly (62.8%)

Problem 2a: U-235 and Nd-148 mass at discharge (PWR)

Discharge burnup	Cooling time = 10-year	
(GWd/MTU)	U-235	Nd-148
15	29.63 kg/MTU (34.2% depleted)	169.8 g/MTU
30	18.63 kg/MTU (58.6% depleted)	336.7 g/MTU
45	10.89 kg/MTU (75.9% depleted)	501.4 g/MTU

Problem 2a: Pu mass and percentage fissile (PWR)

Discharge burnup	Cooling time (years)	
(GWd/MTU)	0	10
15	5.60 kg/MTU (85.3% fissile)	5.51 kg/MTU (85.1% fissile)
30	9.14 kg/MTU (76.6% fissile)	8.78 kg/MTU (75.6% fissile)
45	11.49 kg/MTU (69.0% fissile)	10.94 kg/MTU (67.3% fissile)

Problem 2b: U-235 and Nd-148 mass at discharge (BWR)

Discharge burnup	Cooling time = 10-year	
(GWd/MTU)	U-235	Nd-148
15	29.45 kg/MTU (34.6% depleted)	170.2 g/MTU
30	18.03 kg/MTU (59.9% depleted)	337.7 g/MTU
45	9.99 kg/MTU (77.8% depleted)	503.0 g/MTU

Problem 2b: Pu mass and percentage fissile (BWR)

Discharge burnup	Cooling time (years)	
(GWd/MTU)	0	10
15	5.05 kg/MTU (84.5% fissile)	4.97 kg/MTU (84.3% fissile)
30	8.13 kg/MTU (74.4% fissile)	7.84 kg/MTU (73.5% fissile)
45	10.15 kg/MTU (65.5 fissile)	9.72 kg/MTU (64.2% fissile)

Problem 2c: U-235 and Nd-148 mass at discharge (PWR vs BWR)

Discharge burnup	Ratio (PWR vs BWR) at 10-year cooling	
(GWd/MTU)	U-235	Nd-148
15	1.006	0.998
30	1.033	0.997
45	1.090	0.997

Problem 2c: Pu mass and percentage fissile (PWR vs BWR)

Discharge burnup	Ratio PWR vs BWR at 10-year cooling	
(GWd/MTU)	Pu	Pu % fissile content
15	1.109	1.010
30	1.120	1.029
45	1.125	1.055

OAK RIDGE National Laboratory

ORIGAMI Detailed Input for Rapid Spent Fuel Calculations

SCALE 6.3.2

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals



Gain experience creating from scratch an ORIGAMI input for fuel assembly depletion calculation



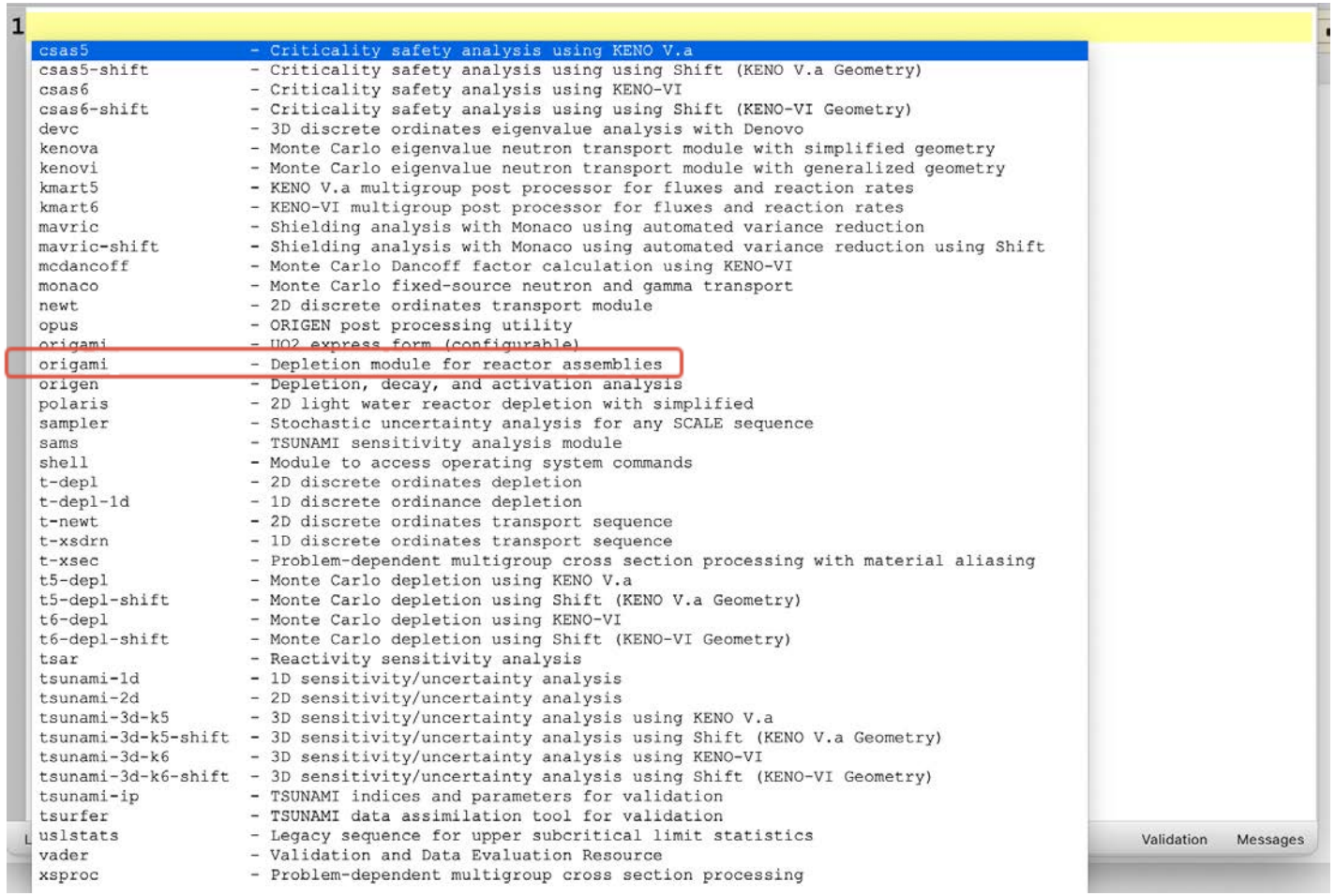
Become familiarized with the full set of ORIGAMI features, including axial and radial moderator density and power gradients

OAK RIDGE National Laboratory

Create ORIGAMI input from scratch

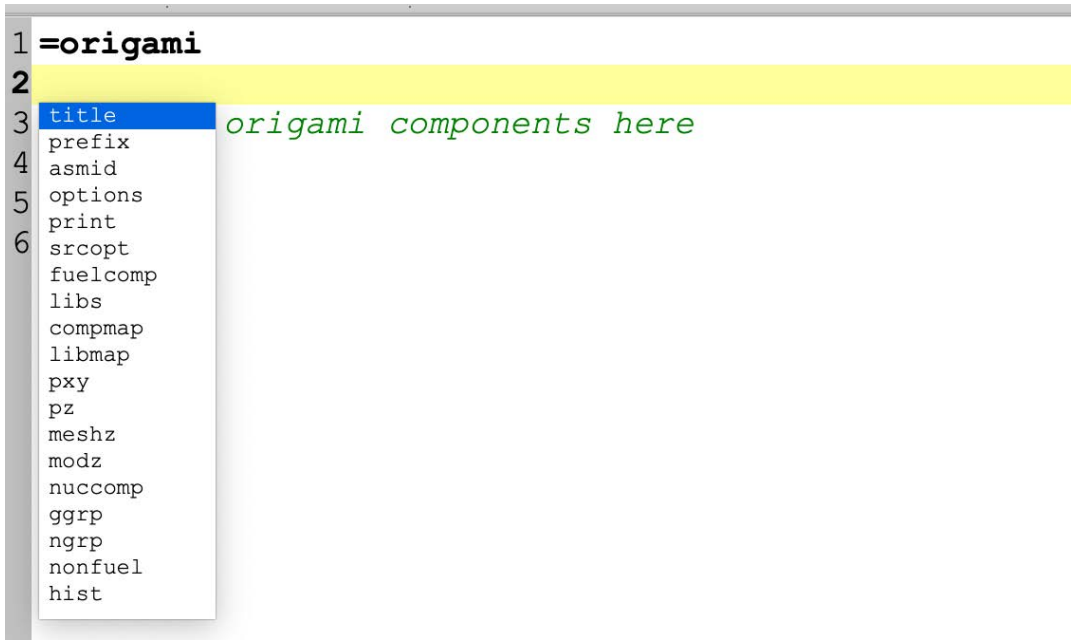


ORIGAMI input from scratch – Create new input file



- CTRL + SPACE
- Select origami
  (not UO2 express form)
- Has autocomplete capability

ORIGAMI input from scratch – Select ORIGEN reactor library



- Put marker inside of origami sequence
- CTRL + SPACE
- Select libs (keep default for specs)

Note: default reactor library is *w17x17* for Westinghouse 17×17 PWR fuel assembly

ORIGAMI input from scratch – Input fuel composition

```
1 =origami
2 libs=[ w17x17 ]
3 fuelcomp{
4
5   uox
6   stdcomp
7   mix
8 end
9
```

```
origami components here
```

- Put marker inside of origami sequence
- CTRL + SPACE
- Select fuelcomp

- Put marker inside of fuelcomp sequence
- CTRL + SPACE
- Select uox (keep default for specs)

Note: default enrichment value for uox is 5% u235

ORIGAMI input from scratch – Input fuel mixture

```
1  =origami
2  libs=[ w17x17 ]
3  fuelcomp{
4  uox(fuel){ enrich=5.0 }
5
6
7                origami components here
8
9  end
10
```

```
uox
stdcomp
mix
```

- Put marker inside of fuelcomp sequence
- CTRL + SPACE
- Select mix (keep default specs)

Note: default for mix is 100% fuel

ORIGAMI input from scratch – Input irradiation history

```
1  =origami
2  libs=[ w17x17 ]
3  fuelcomp{
4  uox(fuel){ enrich=5.0 }
5  mix(1){ comps[ fuel=100.0 ] }
6  }
7  hist[
8
9  cycle          t cycles here
10
11 ]
12
13 % insert origami components here
14
15 end
```

- Put marker outside fuelcomp but inside origami
- CTRL + SPACE
- Select hist

- Put marker inside hist
- CTRL + SPACE
- Select cycle

Note: default for cycle is power 40MW/MTIHM, irradiation time 365 days, downtime 31 days

ORIGAMI input from scratch – First detailed input

```
1 =origami
2 libs=[ w17x17 ]
3 fuelcomp{
4 uox(fuel){ enrich=5.0 }
5 mix(1){ comps[ fuel=100.0 ] }
6 }
7 hist[
8 cycle{ power=40 burn=365 down=31 }
9
10 % insert cycles here
11
12 ]
13
14 % insert origami components here
15
16 end
17
```

Final input should look something like this.

Or cleaned up:

```
1 =origami
2 libs=[ w17x17 ]
3 fuelcomp{
4   uox(fuel){ enrich=5.0 }
5   mix(1){ comps[ fuel=100.0 ] }
6 }
7 hist[ cycle{ power=40 burn=365 down=31 } ]
8 end
```

File is ready to run

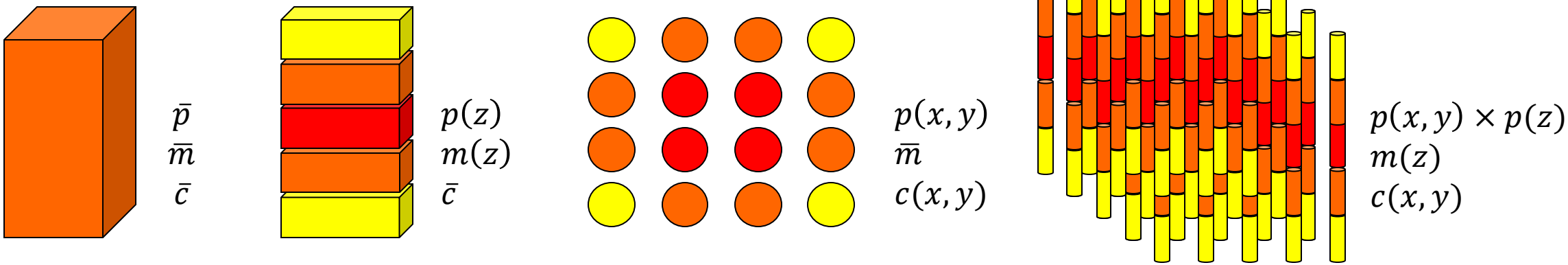
Results will be per MTU

```
29
30 Now depleting axial zone: 001, pin: 01-01
31
32
33 Scale job C:/SCALE-6.2/henrik/temp/origami.inp is finished.
34 Output is stored in C:\SCALE-6.2\henrik\temp\origami.out
35
36 Process finished with 0 return code; ran in 3 secs, finished at Wed Oct 19 10:24:53 2016
```

ORIGAMI spatial variation modeling

- Complexity level needed depends on the application and spent fuel metric of interest
  - 0D is generally adequate for assembly-average decay heat, activity, and nuclide inventory
  - 1D is needed if axial distributions are desired (e.g., for BWR), whereas 2D and 3D may be of interest to specific safeguards applications

power shape → axial & radial → $p(z) \times p(x,y)$
moderator density → axial only → $m(z)$
fuel composition → radial only → $c(x,y)$



0D assembly | 1D assembly | 2D assembly | 3D assembly

ORIGAMI input description: Main blocks

```
=origami
  title="this-is-my-title"
  options{ mtu=1.0 ft71=all }
  libs=[ w17x17 ]
  fuelcomp{
    uox(fuel){ enrich=4.5 }
    mix(1){ comps[ fuel=100.0 ] }
  }
  modz=[ 0.7332 ]
  pz=[ 1.0 ]
  hist[
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ down=1825 }
  ]
end
```

- title descriptive title
- options various global options
- libs ORIGEN reactor library(ies) to use
- fuelcomp fuel mixtures and compositions (single mixture problems use 1)
- modz axial moderator density
- pz axial power shape
- hist operating history

ORIGAMI Input: Fuel composition

```
=origami
  title="this-is-my-title"
  options{ mtu=1.0 ft71=all }
  libs=[ w17x17 ]
  fuelcomp{
    uox(fuel){ enrich=4.5 }
    mix(1){ comps[ fuel=100.0 ] }
  }
  modz=[ 0.7332 ]
  pz=[ 1.0 ]
  hist[
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ down=1825 }
  ]
end
```

- uox**(name){...}**

  creates a SCALE StdCmp UO_2 mixture

- mix(N)

  N mixture index

  comps array of composition names and wt %

SCALE StdCmp has many common standard compositions (zirc4, gd2o3, etc.)

ORIGAMI Input: ORIGEN reactor data libraries

```
=origami
  title="this-is-my-title"
  options{ mtu=1.0 ft71=all }
  libs=[ w17x17 ]
  fuelcomp{
    uox(fuel){ enrich=4.5 }
    mix(1){ comps[ fuel=100.0 ] }
  }
  modz=[ 0.7332 ]
  pz=[ 1.0 ]
  hist[
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ down=1825 }
  ]
end
```

ORIGEN reactor library names in arpdata.txt

libs=["w17x17"]

- declares library 1 to be "w17x17" (Westinghouse 17x17 PWR assembly)
- could be any reactor library
- a single entry is typical

ORIGAMI Input: Power history

```
=origami
  title="this-is-my-title"
  options{ mtu=1.0 ft71=all }
  libs=[ w17x17 ]
  fuelcomp{
    uox(fuel){ enrich=4.5 }
    mix(1){ comps[ fuel=100.0 ] }
  }
  modz=[ 0.7332 ]
  pz=[ 1.0 ]
  hist[
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ down=1825 }
  ]
end
```

Operating history

- cycle

  power assembly-average specific power in MW/MTIHM

  burn time in days at power

  down time in days of decay

  nlib number of library interpolation points per cycle

ORIGAMI Input: Moderator density (modz) & axial power profile (pz)

```
=origami
  title="this-is-my-title"
  options{ mtu=1.0 ft71=all }
  libs=[ w17x17 ]
  fuelcomp{
    uox(fuel){ enrich=4.5 }
    mix(1){ comps[ fuel=100.0 ] }
  }
  modz=[ 0.7332 ]
  pz=[ 1.0 ]
  hist[
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ down=1825 }
  ]
end
```

Axial power and moderator distribution

- modz

  density in g/cm^3 of moderator
  NOTE: not all reactor libraries have density variations

- pz

  axial power shape (unitless)

  NOTE: entered input is renormalized by default

Axial distribution: When is it useful ?

```
=origami

  title="W17x17 Assembly wtih Axial Distribution"
  options{ mtu=1.0 ft71=all }
  libs=[ "w17x17" ]
  fuelcomp{
    uox(fuel){ enrich=4.5 }
    mix(1){ comps=[ fuel=100 ] }
  }

  meshz=[0    50    100   200   300    350]
  modz =[ 0.73  0.72  0.71   0.70  0.69   ]
  pz   =[ 0.80  0.90  1.00   0.85  0.75   ]

  hist[
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=0 }
    cycle{ down=1825 }
  ]

end
```

- to calculate axial decay heat in fuel assemblies
- to evaluate impact of moderator density variations on nuclide inventories, decay heat, or radiation emissions
- to model fork detector for NDA

meshz keyword useful for specifying non-uniform zone sizes (i.e., zone boundaries in cm)

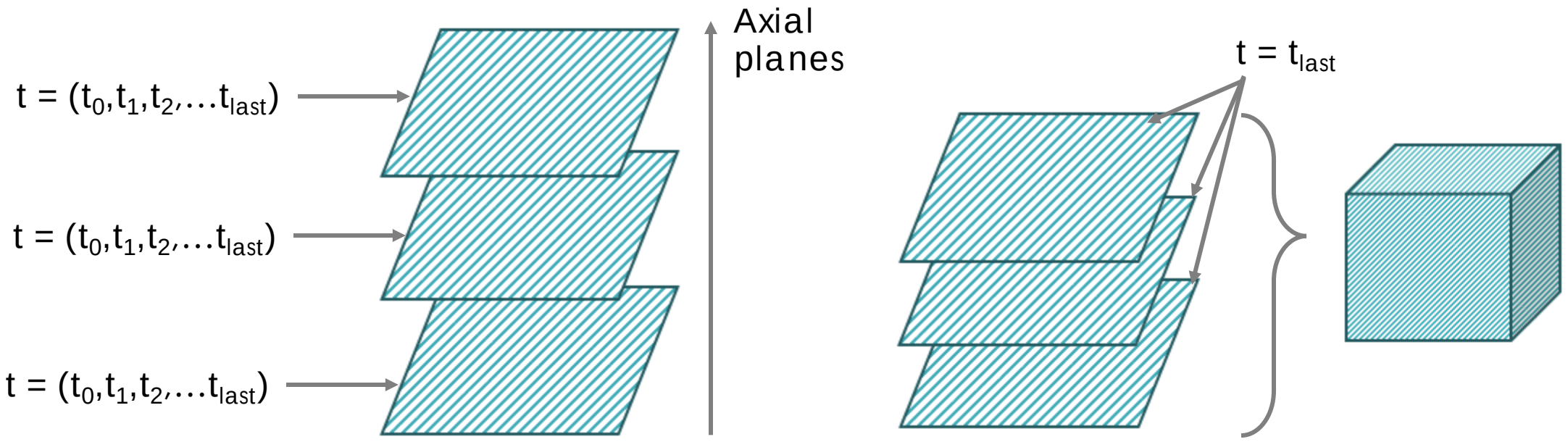
ORIGAMI F71 files generated for models considering axial distributions of burnup and moderator density

${basename}.f71

- Stores concentrations for each saved step for every depleted zone, for each timestep saved

${basename}.assm.f71

- Stores concentrations for each axial zone and the total lumped per assembly, at the last time step



ORIGAMI input: Other options

```
=origami
  title="this-is-my-title"
  options{ mtu=1.0 ft71=all }
  libs=[ w17x17 ]
  fuelcomp{
    uox(fuel){ enrich=4.5 }
    mix(1){ comps[ fuel=100.0 ] }
  }
  modz=[ 0.7332 ]
  pz=[ 1.0 ]
  hist[
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ power=40 burn=333.33 nlib=4 down=16.67 }
    cycle{ down=1825 }
  ]
end
```

Miscellaneous input

- mtu
  - metric tons of initial heavy metal (not just U)
  - DEFAULT: 1 MTU
- ft71
  - flag to write f71 file (last/cycle/all)
- other options
  - stdcomp write SCALE StdComp blocks
  - mcnp write MCNP materials cards

OAK RIDGE National Laboratory

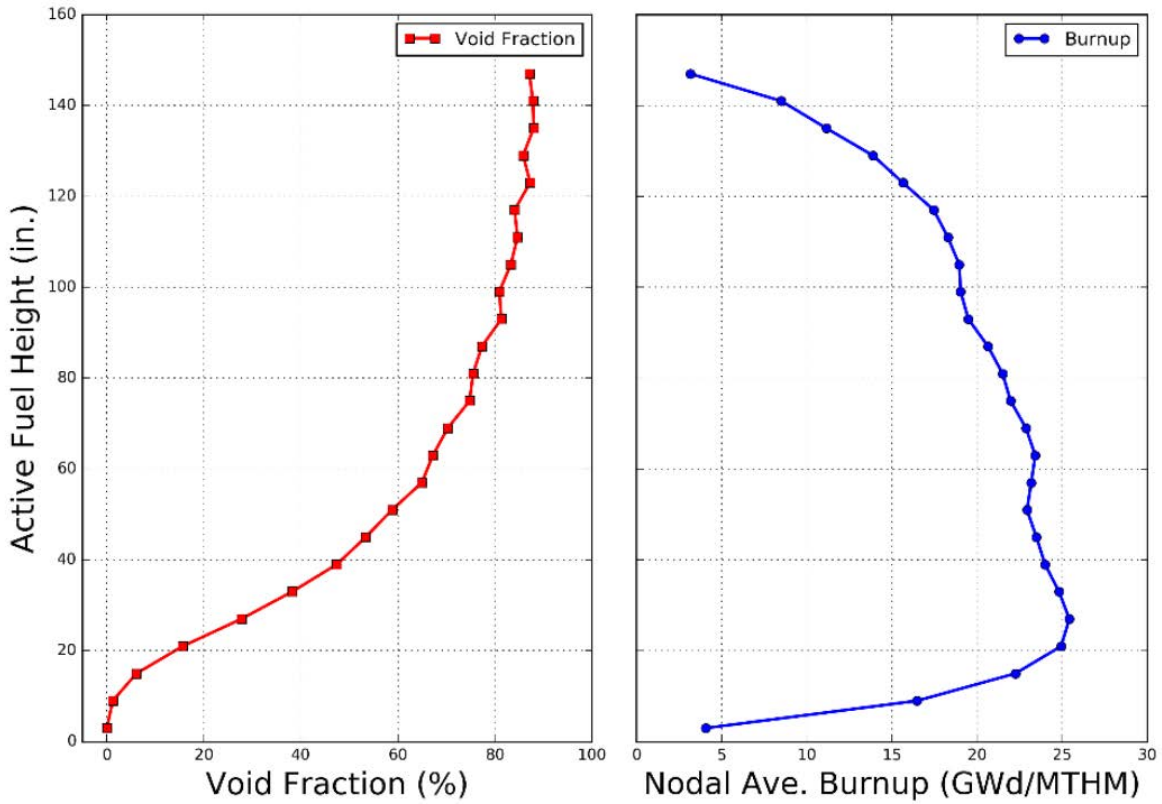
Tutorial:

Modeling axial variation in burnup and coolant density for a BWR assembly



Problem: Model a BWR assembly using 0D and 1D models

- Given a GE14 BWR assembly (ge10×10-8), what is the relative impact of assuming an assembly-average (0D) burnup and coolant density instead of axial burnup and coolant density profiles (1D) at discharge and 10-year cooling time on
  - Assembly Pu content
  - Assembly content for Pu isotopes
  - Assembly decay heat
  - Assembly total neutron source
- Evaluate the impact using two models
  - Typical 0D ORIGAMI calculation (UOX Express)
  - 1D ORIGAMI detailed input calculation



Void fraction and discharge burnup profile for a representative BWR assembly (from NUREG/CR-7240) https://www.nrc.gov/reading-rm/doc-collections/nuregs/contract/cr7240/index.html

Part 1: 0D approximation

- Irradiation history and assembly data
  - Discharge burnup: 45.2 GWd/MTU
  - Assume uniform specific power (21.836 MW/MTU)
  - 3 irradiation cycles, 690 days per cycle, 30 days downtime between cycles
  - Library: ge10x10-8
  - Enrichment: 4.5% ^{235}U
  - Assembly-average coolant density: 0.3084 g/cm^3
  - Use nlib=2 (for speedup)
- Evaluate at 10 years cooling time

Start with the UOX express form!

Part 2: 1D approximation

- Irradiation history and assembly data
  - Discharge burnup: 45.2 GWd/MTU
    assume uniform specific power (21.836 MW/t)
  - 3 irradiation cycles, 690 days per cycle, 30 days downtime between cycles
  - Library: ge10x10-8
  - Enrichment: 4.5% ^{235}U
  - Axial power: use provided data
  - Axial coolant density: use provided data
  - Axial node height: use provided data
  - Use nlib=2 (for speedup)
- Evaluate at 10 years cooling time

Axial node height (cm)	Axial node coolant density (g/cc)	Axial node power (relative)
30.48	0.750	0.549
45.72	0.710	1.189
91.44	0.557	1.340
106.68	0.370	1.283
121.92	0.330	1.256
137.16	0.297	1.226
182.88	0.270	1.238
213.36	0.210	1.162
228.60	0.200	1.104
259.08	0.170	1.030
304.80	0.147	0.975
381.00	0.124	0.559

Hint: We can reuse (expand upon) our prior input!

OAK RIDGE National Laboratory

Tutorial solution:

Modeling axial variation in burnup and coolant density for a BWR assembly



ORIGAMI: 0D input

Decay heat:

- Discharge: 1.343 MW/MTU
- 10 years: 1.590 kW/MTU

Plutonium content (kg/MTU)

	Discharge	10 years
^{239}Pu	6.22	6.27
^{240}Pu	2.78	2.79
^{241}Pu	1.53	0.94
^{238}Pu	0.31	0.31
^{242}Pu	0.64	0.64
Total Pu	11.48	10.96

Neutron source (n/s/MTU)

	10 years
(alpha,n)	1.022E+07
SF	4.822E+08
Total	4.924E+08

^{244}Cm (g/MTU)

- Discharge: 62.8
- 10 years: 42.8

ORIGAMI: 1D input

Decay heat:

- Discharge: 1.340 MW/MTU
- 10 years: 1.643 kW/MTU

Plutonium content (kg/MTU)

	Discharge	10 years
^{239}Pu	6.24	6.30
^{240}Pu	2.70	2.73
^{241}Pu	1.43	0.89
^{238}Pu	0.33	0.32
^{242}Pu	0.69	0.69
Total Pu	11.40	10.92

Neutron source (n/s/MTU)

	10 years
(alpha,n)	1.186E+07
SF	6.969E+08
Total	7.087E+08

^{244}Cm (g/MTU)

- Discharge: 90.4
- 10 years: 61.7

How large is the axial modeling effect for BWRs?

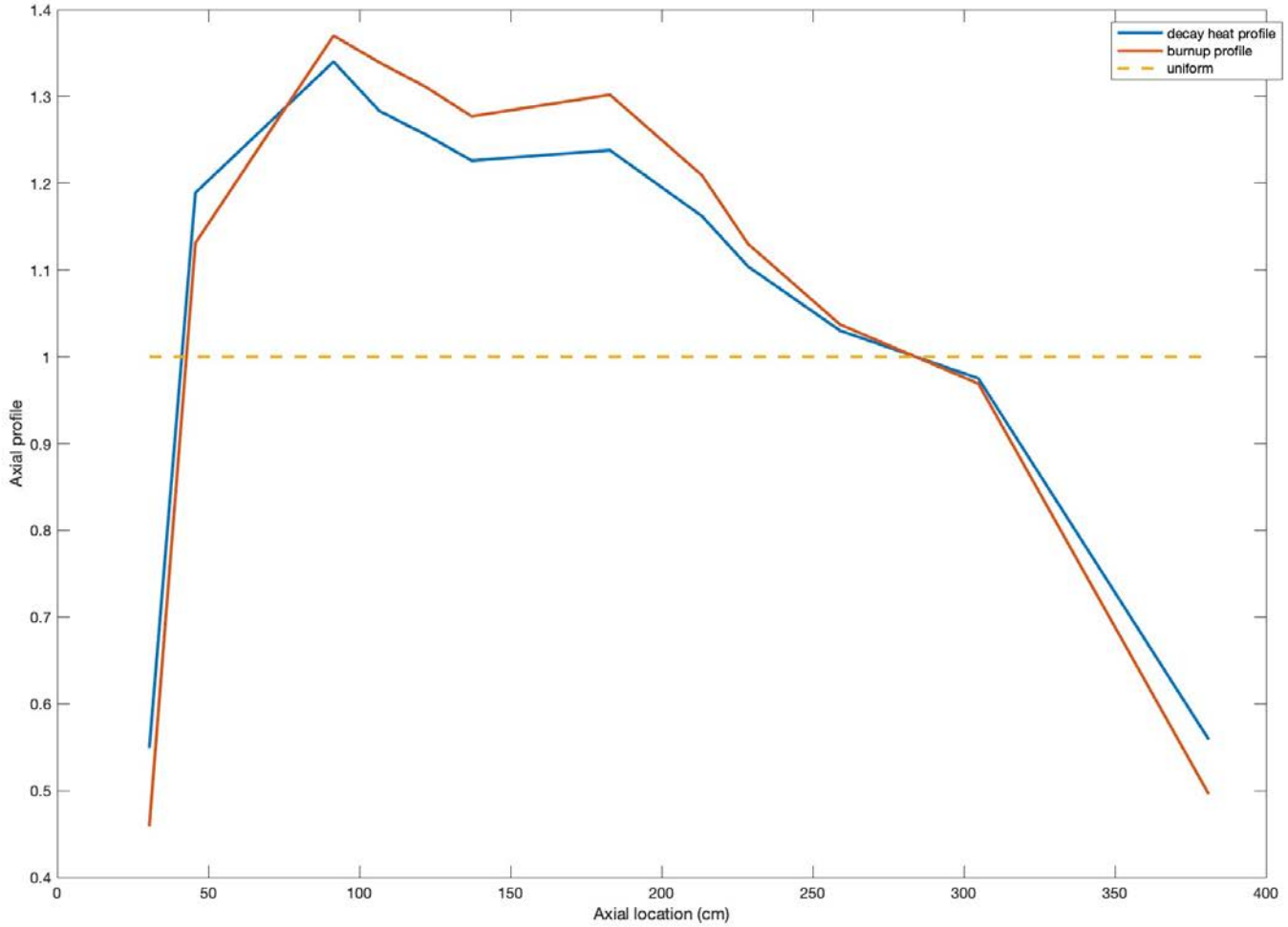
- No change in assembly total Pu @ 10 years
- 3.3% increase in assembly decay heat @ 10 years
- 45% increase in total neutron source @ 10 years

  (45% increase in ^{244}Cm content @ 10 years)

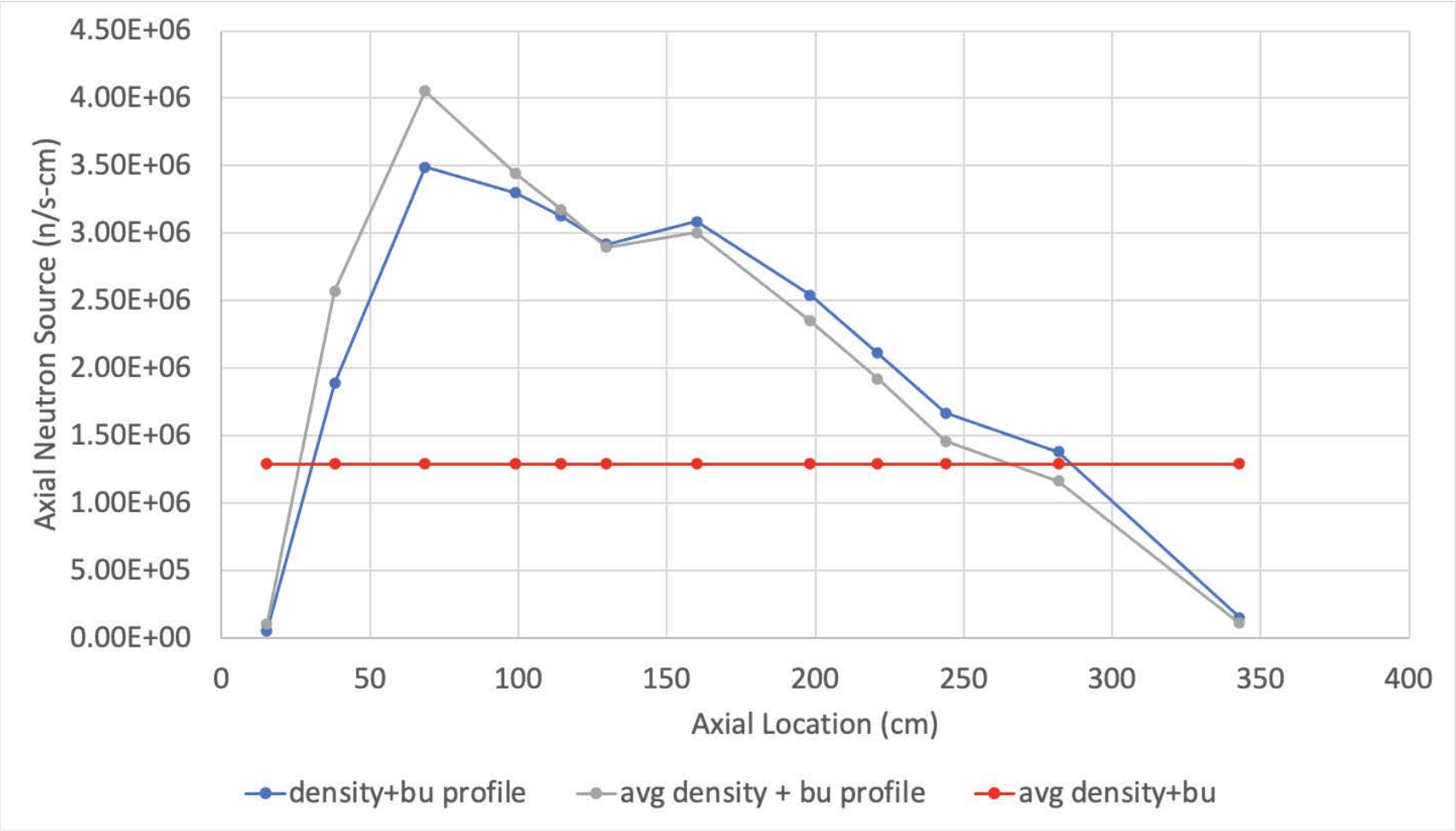
Conclusion?

Best-estimate models for BWR neutron source calculations benefit from including axial information

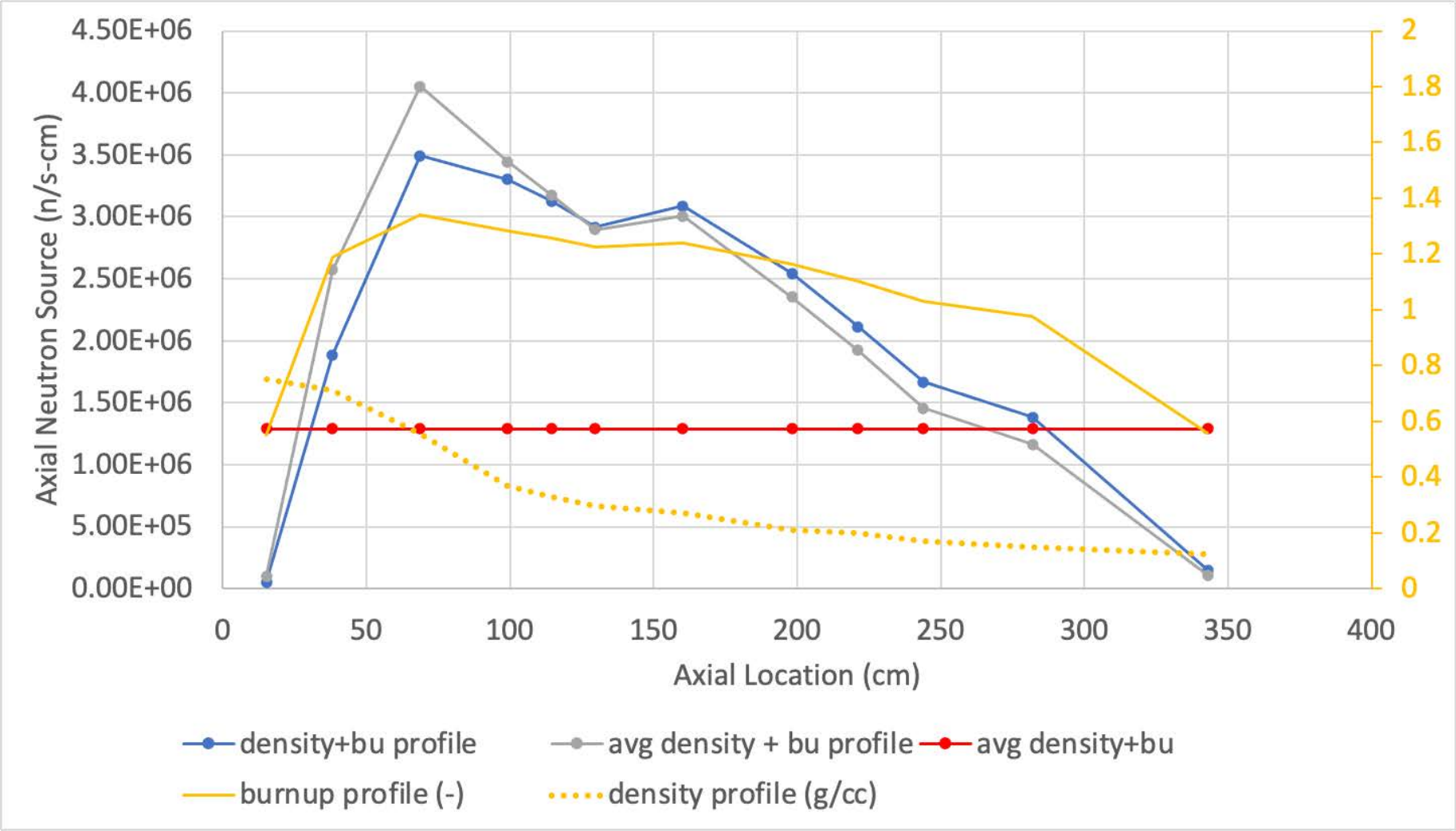
Impact on the decay heat: Axial decay heat profile vs. axial burnup profile



Impact of modeling assumptions on the neutron source



Impact of modeling assumptions on the neutron source with burnup and coolant density profiles



OAK RIDGE National Laboratory

Frequently used input



Global-level options: The "options" block

Controls several calculation options, including

- Problem mass basis / normalization (mtu); default: 1.0
- Non-fuel mass fraction (fracnf)
- Axial power normalization (relnorm)
- Output file generation
  - ft71 (binary), output (ascii)
- Auxiliary output file generation
  - mcnp, stdcomp (KENO), decayheat
- Calculation options
  - ORIGEN solver substeps: nburn, ndecay
  - ORIGEN solver: cram, matrex

Mass normalization: mtu=

- Syntax: mtu=[float]
- Default: mtu=1.0
- Will normalize mass across all zones to value in metric tons of heavy metal (not just uranium)
  - Assumes constant density across all axial / radial locations
  - Will apportion mass per axial zone by volume for non-uniform axial grid
- Used to convert to power in MW

Output control: ft71= and output=

- Syntax:

  ft71=[all / cycle / last / none]

  output=[all / cycle / last / none]

all: Print / save concentrations at the end of each irradiation AND decay step (if present) for each cycle

cycle: Print / save concentrations at the last timestep (irradiation or decay) for each cycle

last: Print / save concentrations at the last timestep (irradiation or decay) of the entire history

Non-fuel components: nonfuel

```
=origami
  title="ORIGAMI depletion and decay 2x2 fuel array"
  options{ mtu=1.0 ft71=all stdcomp=yes }
  libs=[ w17x17 w15x15 ]
  libmap=[ 1 2
           2 1 ]
  fuelcomp{
    uox(fuel){ enrich=5.0 }
    mix(1){ comps[ fuel=100.0 ] }
    uox(fuel2){ enrich=4.0 }
    mix(2){ comps[ fuel2=100.0 ] }
  }
  compmap=[ 2 1
            1 2 ]
  pxy=[ 0.80 1.00
        1.00 0.80 ]
  pz=    [ 0.20 0.50 0.80 1.00 1.10 1.15 1.20 0.90 0.60 0.30 ]
  modz= [ 0.75 0.74 0.73 0.72 0.71 0.70 0.69 0.68 0.67 0.66 ]
  meshz=[ 0   10   50  100  150  200  300  350  360  370 ]
  nonfuel[ zirc4=200.0 co=0.02 ]
```

- Nonfuel is given in kg per MTU
  - Give value pairs of standard compositions (i.e., zirc4), nuclides (i.e., zr-94) or elements (zr)
  - Can be used to account for activation of cladding, spacers, fuel channels, control rods, etc.
- Optional:
  - fracnf= in options block
  - Relative abundances of non-fuel components renormalized to [fracnf] kg per MTU

Print options

```
=origami
  title="ORIGAMI depletion and decay 2x2 fuel array"
  options{ mtu=1.0 ft71=all stdcomp=yes }
  libs=[ w17x17 w15x15 ]
  libmap=[ 1 2
           2 1 ]
  fuelcomp{
    uox(fuel){ enrich=5.0 }
    mix(1){ comps[ fuel=100.0 ] }
    uox(fuel2){ enrich=4.0 }
    mix(2){ comps[ fuel2=100.0 ] }
  }
  compmap=[ 2 1
            1 2 ]
  pxy=[ 0.80 1.00
        1.00 0.80 ]
  pz=   [ 0.20 0.50 0.80 1.00 1.10 1.15 1.20 0.90 0.60 0.30 ]
  modz= [ 0.75 0.74 0.73 0.72 0.71 0.70 0.69 0.68 0.67 0.66 ]
  meshz=[ 0   10   50  100  150  200  300  350  360  370 ]
  nonfuel[ zirc4=200.0 co=0.02 ]
  print{
    nuc{
      sublibs=[lt ac fp]
      total=yes
      units=[ GRAMS CURIES WATTS ]
    }
  }
```

print - sets what is shown in the output for last time step

- select between nuclides (nuc) or elements (ele)
- select sublibs; can be light elements (lt), actinides (ac), fission products (fp) or all
- total=yes will print the total concentration for each output unit type

Special option for axial meshes (in options)

- Axial power normalization (relnorm) in options block
  - Yes (default): shape factors treated as relative modifiers of power
  - No: shape factors treated as absolute power fractions per axial zone

OAK RIDGE National Laboratory

Workshop:

ORIGAMI detailed input for rapid LWR spent fuel calculations



Problem 1: Decay heat benchmark for BWR assembly

Determine the decay heat for BWR fuel assembly ID=13847 irradiated in Forsmark Unit 3 and compare to the reported measured decay heat; estimate the impact of cladding and structural materials activation on the calculated assembly decay heat

- Input data (source NUREG/CR-6972)
  - SVEA 10×10 assembly design
  - Moderator density: 0.3816 g/cm^3
  - Enrichment: 2.77 % ^{235}U
  - Assume initial U contains U-235 and U-238 only
  - Assembly initial U loading 180.67 kg
  - Light element loading (accounts for clad and spacers activation): Table 1
  - Irradiation history data: Table 2
  - Cycle powers: Table 3
  - Reported measured decay heat: 169.6 W
  - Cooling time (from discharge to date of measurement): 4871 days

Problem 1: Decay heat benchmark for BWR assembly (cont.)

- Results required:
  - Calculated decay heat for assembly (W) using all provided input data
  - Calculated-to-measured (C/M) ratio
  - Difference between calculation and measurement C-M in W
  - Impact of cladding and structural materials activation on calculated decay heat (W) for the assembly
- Strategy:
  - Use SVEA100-0 reactor library
  - Optionally, can calculate decay heat for 1 MTU basis, then scale to assembly initial U loading
  - For estimating the impact of cladding and structural materials activation, perform calculation with or without the light element loading included

Problem 1: Decay heat benchmark for BWR assembly (cont.)

Table 1. Assembly light element content

Element	kg/MTU
C	1.882E-03
N	6.033E-03
Al	3.708E-02
Si	1.389E-02
S	3.321E-04
Ti	1.160E-01
Cr	6.958E-01
Mn	3.244E-02
Fe	3.154E-01
Co	3.017E-02
Ni	3.356E+00
Cu	2.325E-03
Nb	4.173E-02
O	1.384E+02
Zr	3.819E+02
Sn	2.712E+00

Table 2. Assembly 13847 irradiation history

Operational Data		Cycle 2	Cycle 3	Cycle 4	Cycle 5
Startup date		9/2/1986	8/9/1987	9/5/1988	7/13/1989
Shutdown date		7/10/1987	8/12/1988	6/9/1989	7/13/1990
Operating days		311	369	277	365
Downtime days		30	24	34	---
Assembly ID	Design	Assembly burnup (MWd/MTU)			
13847	SVEA-100	7857	17217	23779	31275

Problem 1: Decay heat benchmark for BWR assembly (cont.)

Table 3. Assembly 13847 power data

Cycle number	Sample power (MW/MTU)
2	25.26
3	25.37
4	23.69
5	20.54

Problem 2: Nuclide inventory benchmark for PWR spent fuel

Determine the nuclide inventory for Calvert Cliffs PWR spent fuel sample D047_MKP109_P and compare to experimental nuclide concentrations reported for this sample

- Input data (source NUREG/CR-6968 report and SFCOMPO database):
  - CE 14×14 assembly design
  - Enrichment: 3.038% U-235
  - Initial U isotopic composition (wt%): 0.027 U-234, 0.014 U-236, 96.921 U-238
  - Sample irradiation history data: Table 3
  - Measured nuclide inventory data: Table 4
  - Cooling time (from discharge to measurement date): Table 4 footnote
  - Note on sample identifier: this sample was selected from fuel rod MKP109 of fuel assembly D047

Problem 2: Nuclide inventory benchmark for PWR spent fuel (cont.)

- Results required:
  - Calculated (C) nuclide concentrations (g/MTU) for the nuclides listed in Table 2
  - Calculated-to-measured (C/M) concentration ratios for the nuclides listed in Table 2
- Strategy:
  - Use ce14x14 reactor library
  - Multiple approaches to resolve the problem, based on ORIGAMI only, or adding to it ORIGEN and/or OPUS

Problem 2: Nuclide inventory benchmark for PWR spent fuel (cont.)

Table 3. Irradiation history data for Calvert Cliffs sample D047_MKP109_P

Reactor cycle #	Start date (month/day/year)	End date (month/day/year)	Duration (days)	Down (days)	Sample cumulative burnup (GWd/MTU)
2	3/22/77	1/22/78	306	71	9.52
3	4/3/78	4/20/79	382	81	21.93
4	7/10/79	10/18/80	466	85	34.14
5	1/11/81	4/17/82	461		44.34

Problem 2: Nuclide inventory benchmark for PWR spent fuel (cont.)

Table 4. Experimental data for Calvert Cliffs sample D047_MKP109_P

Nuclide	g/g U initial	Exp. uncert. (%)
u234	1.361E-04	1.6
u235	4.016E-03	1.6
u236	4.186E-03	1.6
u238	9.358E-01	1.6
pu238	3.049E-04	1.6
pu239	4.943E-03	1.6
pu240	2.885E-03	1.6
pu241	1.157E-03	1.6
pu242	9.530E-04	1.6
np237	5.338E-04	1.9
am241	4.331E-04	4.9
sr90	5.425E-04	5.7
tc99	8.944E-04	3.5
cs133	1.407E-03	0.0
cs135	4.874E-04	14.0
cs137	1.424E-03	3.5

Nuclide	g/g U initial	Exp. uncert. (%)
nd143	8.656E-04	1.0
nd144	1.864E-03	1.0
nd145	8.440E-04	1.0
nd146	9.416E-04	1.0
nd148	4.855E-04	1.0
nd150	2.360E-04	2.0
sm147	3.081E-04	3.8
sm148	2.246E-04	3.8
sm150	3.638E-04	3.7
sm151	1.072E-05	5.1
sm152	1.325E-04	4.2
sm154	5.640E-05	3.8
eu153	1.612E-04	3.4
gd155	9.959E-06	3.9

* measurement data reported at 1870 d after discharge for all nuclides except for Sm, Eu, Gd; for the latter, cooling time is 4656 d.

Problem 3: Decay heat axial variation for PWR assembly benchmark

Determine the impact of modeling axial variation of burnup on the total assembly decay heat for the decay heat benchmark for PWR assembly F32 irradiated in Ringhals 2 reactor

- Input data (sources NUREG/CR-6972 and NUREG/CR-6971):
  - 15×15 assembly design
  - Enrichment: 3.197% ^{235}U
  - Initial U contains 0.02 wt% U-234 and 0.04 wt% U-236
  - Light element loading is neglected here
  - Discharge burnup 50.962 GWd/MTU
  - Assembly initial U loading 436.99 kg
  - Assembly irradiation history data: Table 5
  - Fuel rod active length 365.8 cm
  - Axial assembly power provided for 24 axial nodes of equal heights: Table 6
  - Reported assembly measured decay heat: 692.0 W
  - Cooling time (from discharge to measurement date): 5860 days

Problem 3: Decay heat axial variation for PWR assembly benchmark

- Results required:
  - Calculated decay heat for assembly (W) with and without considering assembly axial power
  - Calculated-to-measured (C/M) assembly decay heat ratio with and without considering assembly axial power
  - Difference between calculation and measurement C-E in W for assembly decay heat with and without considering assembly axial power
  - Calculated assembly axial decay heat (in 24 axial nodes)
  - Calculated assembly axial decay heat profile
  - Comparison of assembly axial decay heat profile and axial power (burnup) profile
- Strategy:
  - Use the w15x15 reactor library
  - Optional, can calculate decay heat for 1 MTU basis, then scale to assembly initial U loading
  - Perform two calculations using a 0D model and a 1D model
  - Calculate assembly decay heat profile by dividing the axial node decay heat by the assembly-average (over all nodes) decay heat value

Problem 3: Decay heat axial variation for PWR assembly benchmark

Table 5. Axial power profile for assembly F32

Axial node #	Axial node power (relative)
1	0.631
2	0.871
3	0.989
4	1.037
5	1.055
6	1.061
7	1.062
8	1.061
9	1.060
10	1.059
11	1.058
12	1.057
13	1.056
14	1.055
15	1.054
16	1.054
17	1.053
18	1.053
19	1.051
20	1.047
21	1.033
22	0.994
23	0.889
24	0.661

Cycle #	Duration (days)	Down (days)	Assembly power (MW/MTU)
1	290	80	36.390
2	317	84	33.467
3	273	778	30.736
4	320	63	24.253
5	297	54	22.320
6	329		21.334

OAK RIDGE National Laboratory

Workshop solutions:

ORIGAMI detailed input for rapid LWR spent fuel calculations



Problem 1: Decay heat benchmark for BWR assembly (input)

```
=origami
title="Forsmark BWR Deacy Heat Benchmark"
options{ mtu=0.18067 ft71=all} % 180.67 kgU mass basis
libs=[ "svea100-0" ]
nonfuel=[ % nonfuel in kg/MTU mass basis
   C=1.882E-03   N=6.033E-03 Al=3.708E-02 Si=1.389E-02
   S=3.321E-04  Ti=1.160E-01 Cr=6.958E-01 Mn=3.244E-02
   Fe=3.154E-01 Co=3.017E-02 Ni=3.356E+00 Cu=2.325E-03
   Nb=4.173E-02 O=1.384E+02  Zr=3.819E+02 Sn=2.712E+00
]
fuelcomp{
   mix(1){ stdcomp(fuel){ base=uo2 iso[ 92235=2.77 92238=97.23 ] } }  % 2.77% U-235 + U-238 only
}
modz=[ 0.3816 ] % 0.3816 g/cm3 average moderator density
pz=[ 1.0 ] % 1 axial zone
hist[
   cycle{ power=25.26 burn=311 nlib=4 down=30 }
   cycle{ power=25.37 burn=369 nlib=4 down=24 }
   cycle{ power=23.69 burn=277 nlib=4 down=34 }
   cycle{ power=20.54 burn=365 nlib=4 down=4871 }
]
end
```

Problem 1: Decay heat benchmark for BWR assembly (results)

- Calculated (C) assembly decay heat at 4871 days after discharge
  - 168.322 W (fuel only)
  - 169.207 W (fuel + structural material)
- Measured (M) decay heat: 169.6 W (per assembly)
- Difference calculation-measurement
  - C-M = −1.3 W (fuel); -0.393 W (fuel + structural material)
  - C/M = 0.992 (fuel); 0.998 (fuel + structural material)

Problem 2: Nuclide inventory benchmark for PWR spent fuel

Nuclide	Measured g/gU initial	Calculated g/gU initial	C/M
u234	1.361E-04	1.280E-04	0.940
u235	4.016E-03	3.619E-03	0.901
u236	4.186E-03	4.287E-03	1.024
u238	9.358E-01	9.349E-01	0.999
pu238	3.049E-04	2.933E-04	0.962
pu239	4.943E-03	4.832E-03	0.978
pu240	2.885E-03	2.839E-03	0.984
pu241	1.157E-03	1.432E-03	1.238
pu242	9.530E-04	9.527E-04	1.000
np237	5.338E-04	5.209E-04	0.976
am241	4.331E-04	6.266E-05	0.145
sr90	5.425E-04	6.363E-04	1.173
tc99	8.944E-04	1.043E-03	1.166
cs133	1.407E-03	1.432E-03	1.018
cs135	4.874E-04	4.870E-04	0.999
cs137	1.424E-03	1.584E-03	1.112

Nuclide	Measured g/gU initial	Calculated g/gU initial	C/M
nd143	8.656E-04	8.465E-04	0.978
nd144	1.864E-03	1.642E-03	0.881
nd145	8.440E-04	8.420E-04	0.998
nd146	9.416E-04	9.502E-04	1.009
nd148	4.855E-04	4.936E-04	1.017
nd150	2.360E-04	2.454E-04	1.040
sm147	3.081E-04	2.570E-04	0.834
sm148	2.246E-04	2.120E-04	0.944
sm150	3.638E-04	3.766E-04	1.035
sm151	1.072E-05	1.047E-05	0.977
sm152	1.325E-04	1.314E-04	0.992
sm154	5.640E-05	5.414E-05	0.960
eu153	1.612E-04	1.556E-04	0.965
gd155	9.959E-06	6.385E-06	0.641

Problem 3: Decay heat axial variation for PWR assembly benchmark

- Assembly calculated decay heat (W)
  - 0D model: 719.5
  - 1D model: 724.3
- Calculated - Measured (W)
  - 0D model: 27.5
  - 1D model: 32.3
- Calculated/Measured
  - 0D model: 1.040
  - 1D model: 1.047

Axial node #	Axial power profile	Axial decay heat profile
1	0.631	0.552
2	0.871	0.825
3	0.989	0.978
4	1.037	1.045
5	1.055	1.070
6	1.061	1.079
7	1.062	1.080
8	1.061	1.079
9	1.060	1.077
10	1.059	1.076
11	1.058	1.074
12	1.057	1.073
13	1.056	1.072
14	1.055	1.070
15	1.054	1.069
16	1.054	1.069
17	1.053	1.067
18	1.053	1.067
19	1.051	1.064
20	1.047	1.059
21	1.033	1.039
22	0.994	0.985
23	0.889	0.847
24	0.661	0.584

OAK RIDGE National Laboratory

Beyond 1D: Radial modeling with ORIGAMI

- Powers, compositions, and complex mixtures



Beyond UOX: Using mixture compositions

- ORIGAMI is built on the SCALE standard composition engine
  - Mixtures, therefore, work like they do in other tools such as TRITON
- The uox= input is internally an *alias* for the more general syntax:

Creates a named standard composition (StdComp) based on UO_2

```
uox(fuel) {
  enrich=5.0
}
```

```
stdcomp(fuel) {
  base=uo2 iso[ 92234=0.0 92235=5.0 92236=0.0 92238=95.0 ]
}
mix(1){ comps[ fuel=100.0 ]
```

Each mixture has a unique numeric identifier (used by compMap)

Each mixture made up of one or more named compositions from StdComp

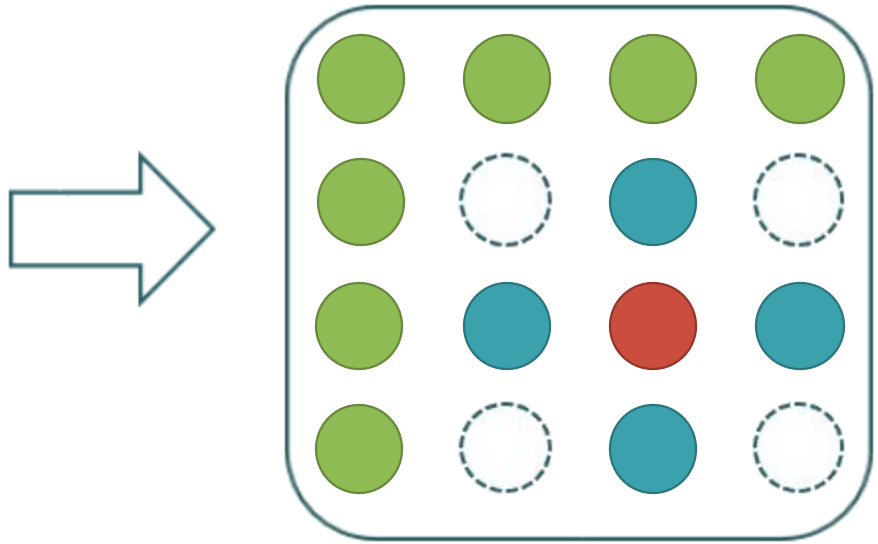
Creating complex mixtures for ORIGAMI using standard compositions

```
fuelcomp{
  stdcomp(fuel_5wt) {
    base=uo2 iso[ 92234=0.0 92235=5.0 92236=0.0 92238=95.0   ]
  }
  stdcomp(fuel_3wt) {
    base=uo2 iso[ 92234=0.0 92235=3.0 92236=0.0 92238=97.0   ]
  }
  mix(1){ comps[ fuel_5wt=100.0 ]  % 100% UO2 @ 5.0% enrichment
  mix(2){ comps[ fuel_3wt=100.0 ]  % 100% UO2 @ 3.0% enrichment
  mix(3){ comps[ fuel_3wt=97.0 Gd2O3=3.0 ] } % 97% UO2 @ 3.0% enrichment + 3% Gadolinia
…
}
```

Creating a heterogeneous array of pins: compmap

- Entries in compmap array correspond to fuel mixture numbers defined in the fuelcomp{ } block
- Entry format identical to TRITON / KENO arrays
  - Start at bottom row, left side => bottom-right
  - Proceed left-to-right along rows
  - "0" entry indicates a void (no material / fuel)

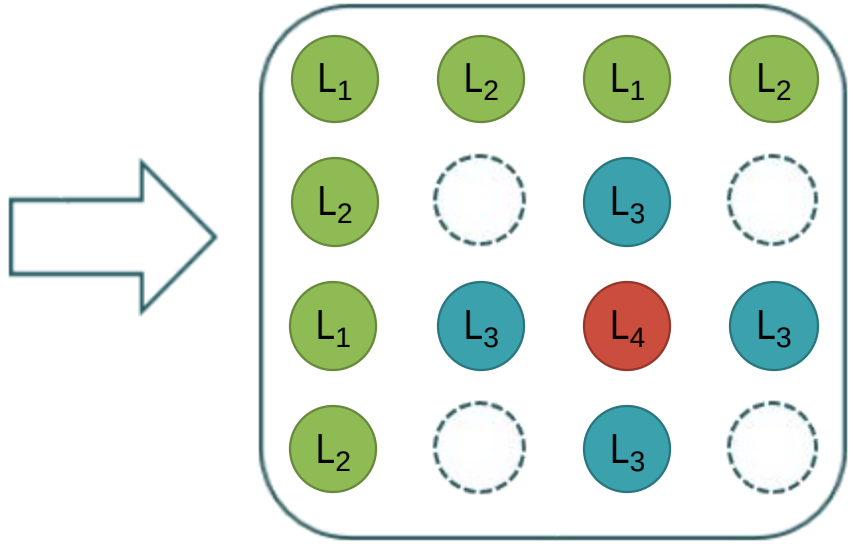
```
compmap=
      [ 1  0  2  0
        1  2  3  2
        1  0  2  0
        1  1  1  1 ]
```



Defining different radial libraries: libmap

- Entries in **libmap** array correspond to index of entries defined in libs[ ] array (starting from 1)
- Same mapping syntax as compmap[ ]
  - Default is libmap=[1], which will assign the same library to all pins
  - Caveat: if libmap is explicitly specified, must be consistent with compmap
    - "0" entry implies no pin at location

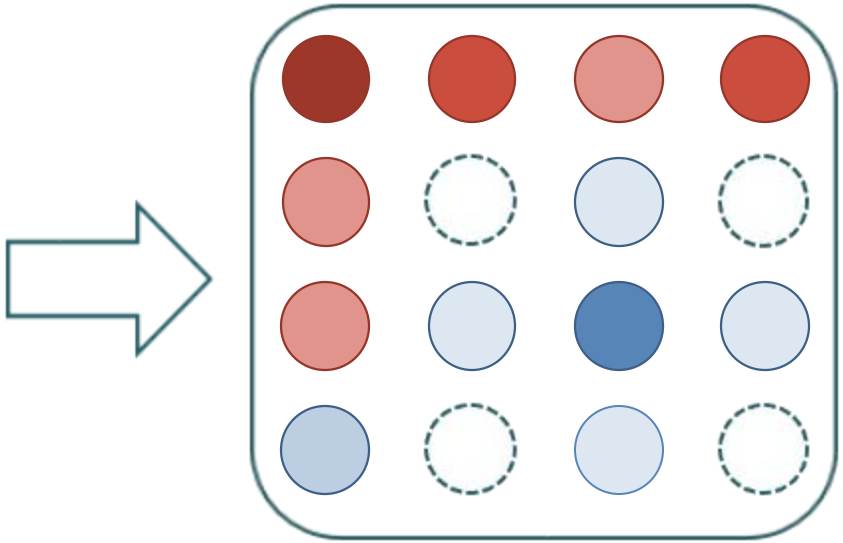
```
libmap=
      [ 2  0  3  0
        1  3  4  3
        2  0  3  0
        1  2  1  2 ]
```



Introducing radial power gradients: pxy

- Radial power gradients are specified using the pxy array
- Imposes a relative power tilt
  - Mean of pxy automatically normalized to 1.0 for all fuel pins, preserving total assembly power

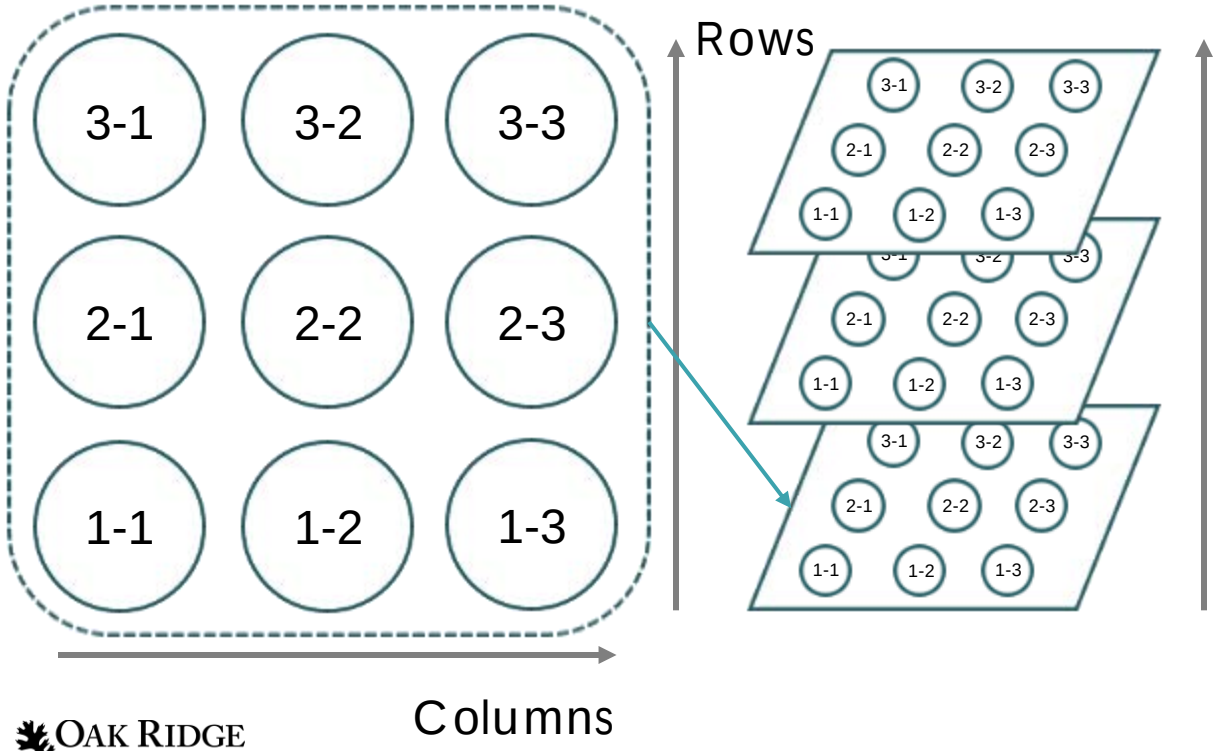
```
pxy= [ 0.75 0.0  0.80  0.0
       1.1  0.8  0.65  0.8
       1.1  0.0  0.8   0.0
       1.2  1.15 1.1   1.15 ]
```



F71 binary concentrations file, revisited

Now that we've added radial concentration information in addition to axial concentrations, how do we navigate the F71 file?

```
${basename}.f71
```



Concentrations now stored in sequential order by:

1. Time step (t_0, t_1, t_2, …)
2. Pin (1-1, 1-2, 1-3, 2-1…)
3. Axial zone (1, 2… N_{ax})

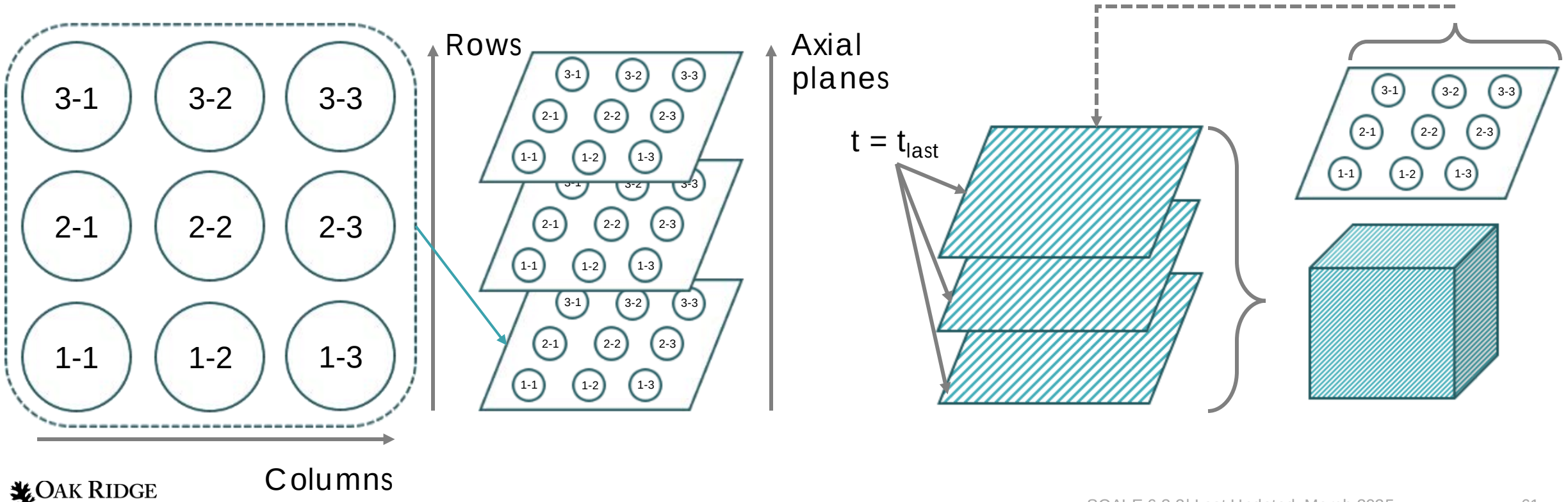
ORIGAMI radial and axial modeling: F71 files structures

${basename}.f71

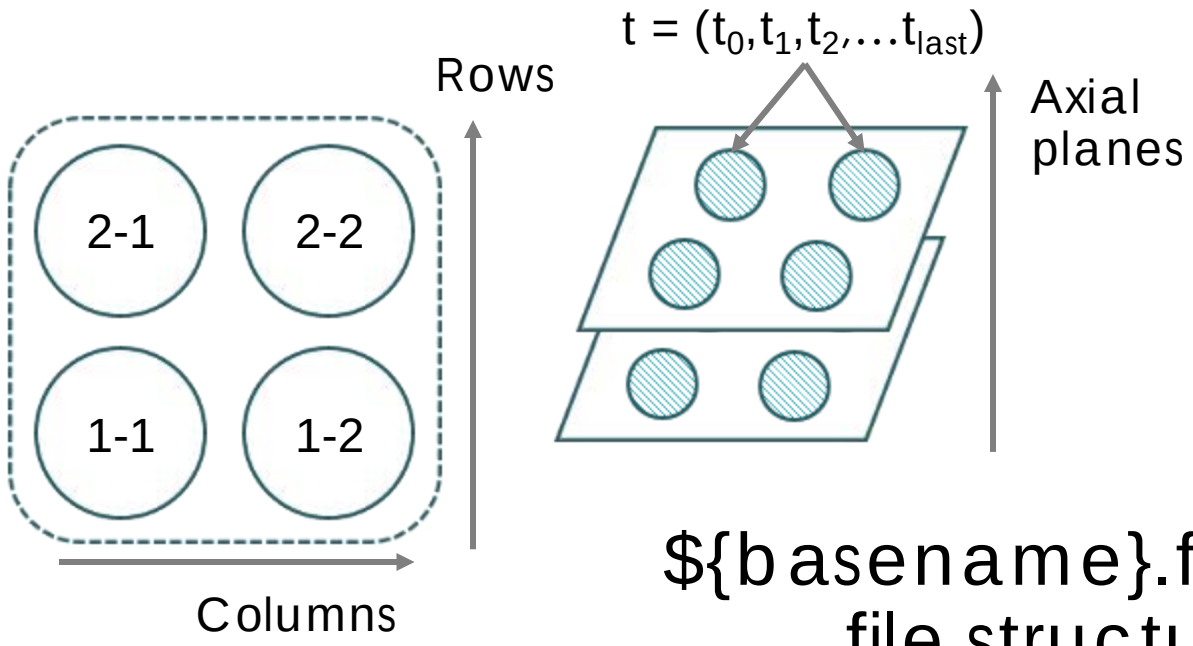
- Stores concentrations for each saved step for every depleted zone

${basename}.assm.f71

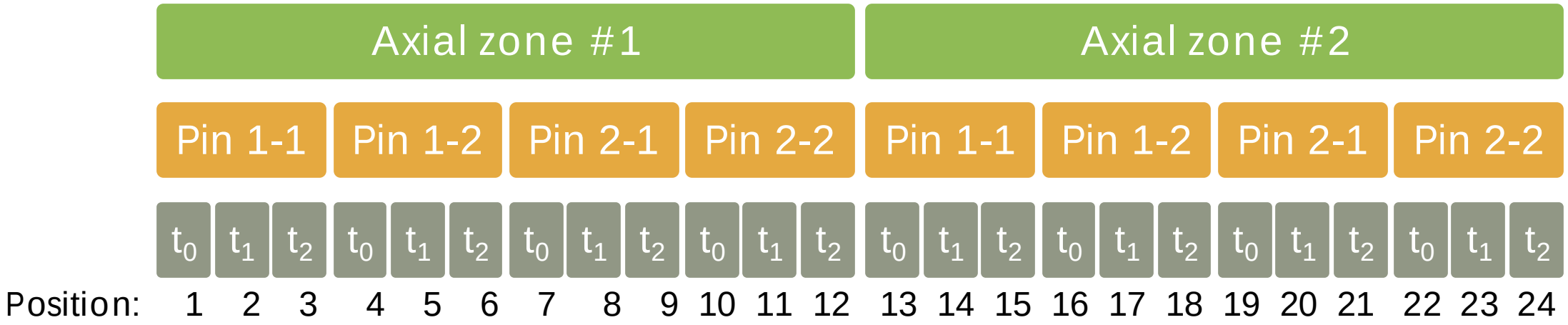
- Stores concentrations for each lumped axial zone and the total lumped assembly at the last saved timestep



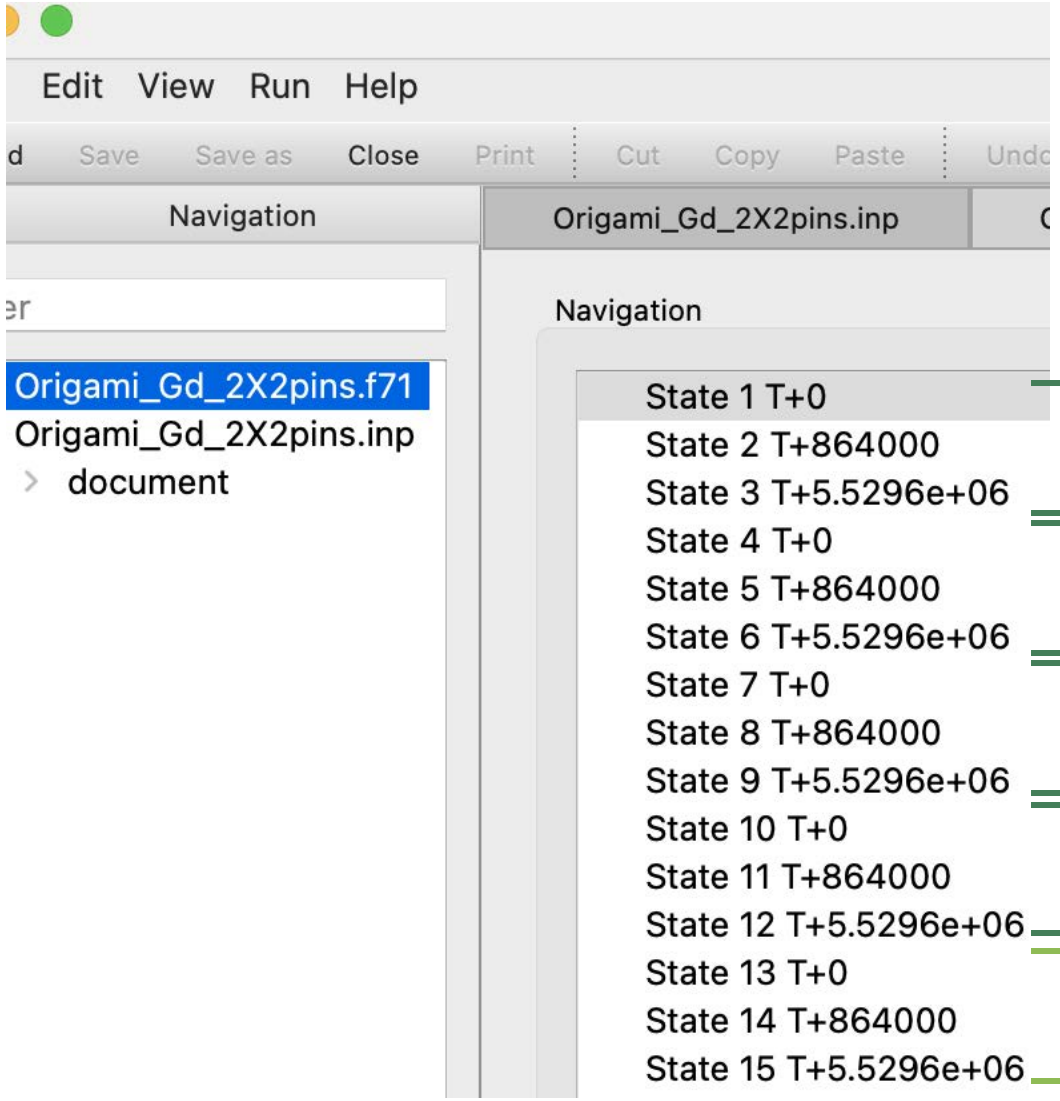
Parsing ORIGAMI concentrations files: pin-by-pin



${basename}.f71 file structure



Making sense of the F71 file: Individual pins



Axial zone #1, Pin 1-1
- t=0 (initial)
- t=10 days (end of irradiation)
- t=64 days (+ 50 days decay)

Axial zone #1, Pin 1-2

Axial zone #1, Pin 2-1

Axial zone #1, Pin 2-2

Axial zone #2, Pin 1-1

OAK RIDGE National Laboratory

Tutorial:

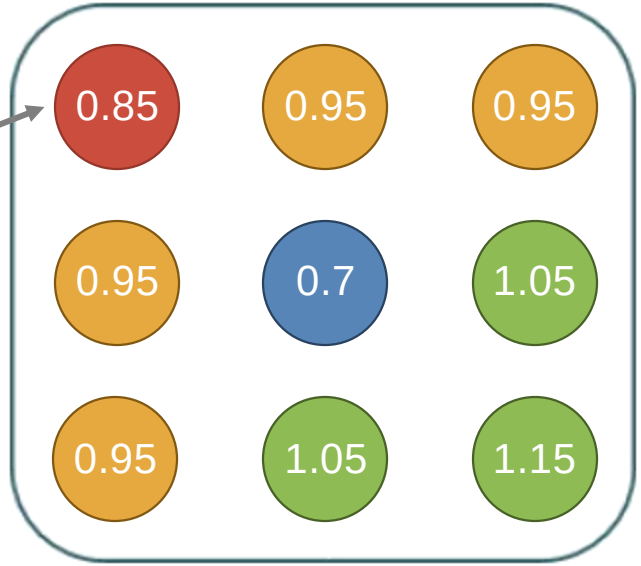
Gd-bearing BWR assembly: Effect on decay heat and Pu inventory



Tutorial: BWR assembly with Gd rods

- Calculate the decay heat and plutonium content for each fuel pin of a corner region for a Gd-bearing BWR assembly. Use two sets of ORIGEN libraries:
  - Same library for all pins (ge10x10-8)
  - Pinwise libraries (provided)
- How do your results change with the library used?

Numbers indicate relative power shaping factor



- 1.8% ^{235}U / 100 % UO_2
- 2.6% ^{235}U / 100% UO_2
- 3.0% ^{235}U / 100% UO_2
- 5.0% ^{235}U / 94% UO_2 + 6% Gd_2O_3

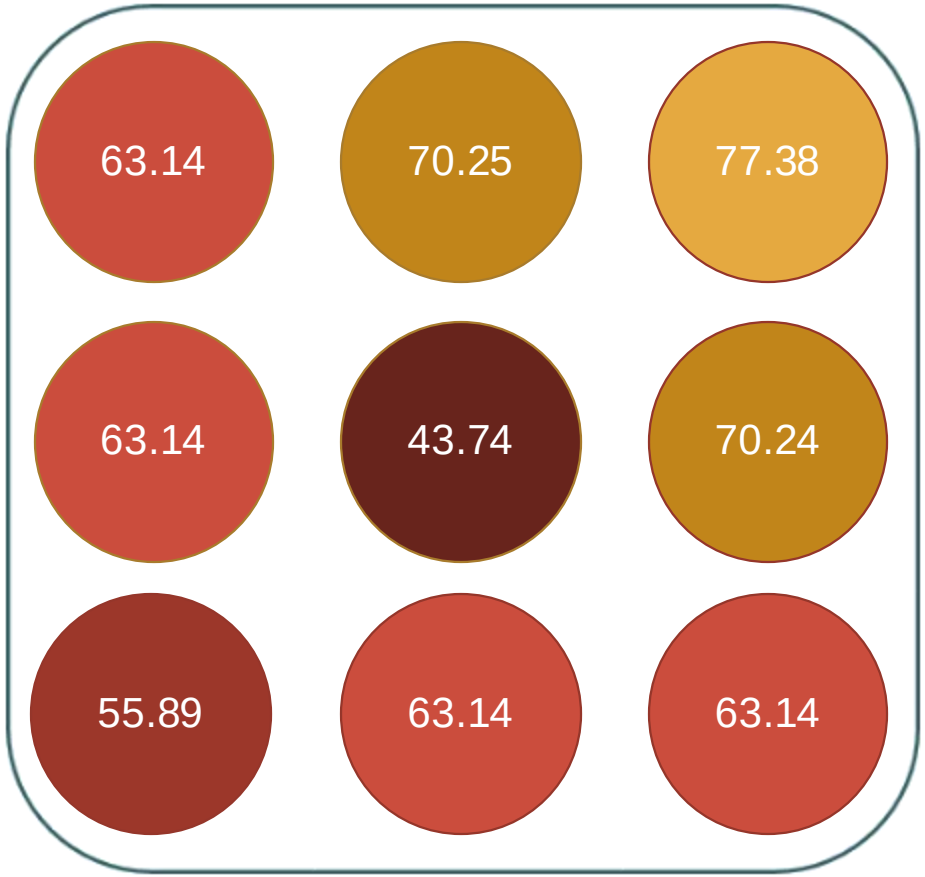
Tutorial: BWR assembly with Gd rods

- Input data:
  - 1 cycle irradiation, 24-month
  - Specific power: 25 MW/MTU
  - Cooling time: 10 years
  - Fuel compositions (wt %) are shown on previous page
  - Pinwise libraries and associated arpdata.txt file provided
  - Assume coolant density of 0.6 g/cc
- Required results (for each pin):
  - Decay heat (W)
  - Pu content (wt% Pu-239 / Pu, Pu mass)

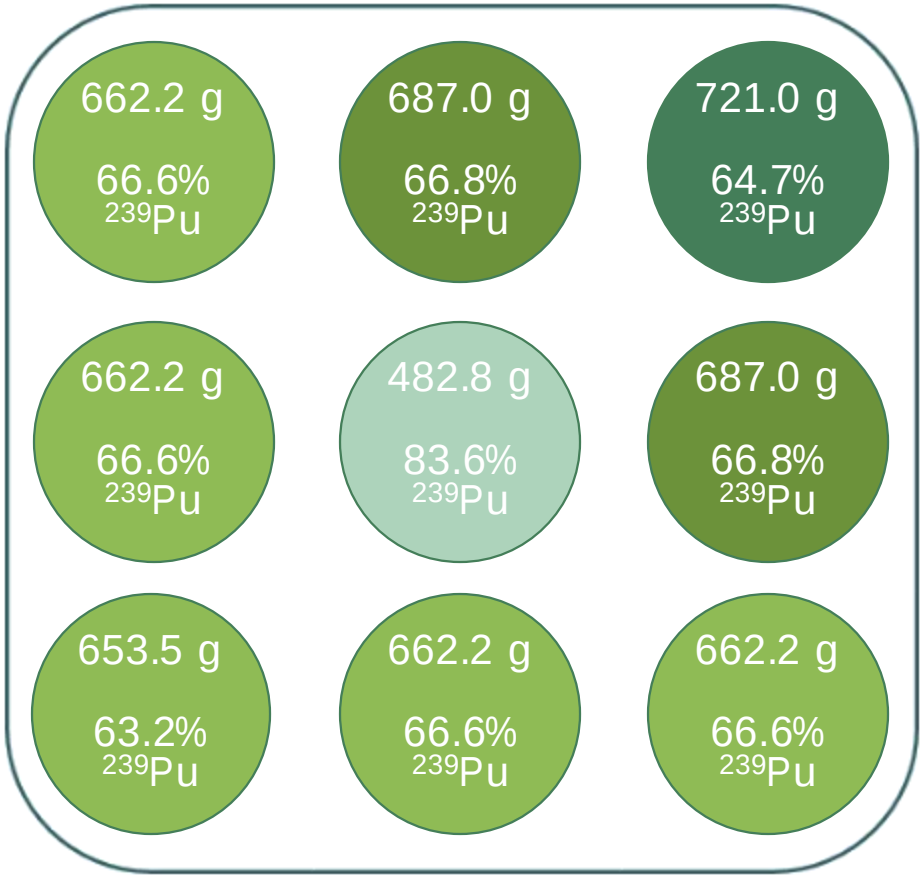
- Strategy:
  - Note that data Compositions are element wt. %; 6% of Gd pin mass is Gd_2O_3
  - Use a shell block at the beginning of the input to provide information about the custom arpdata.txt and pinwise libraries

Solution: BWR assembly with Gd rods (same library in all pins)

Decay heat (W) @ 10 years cooling

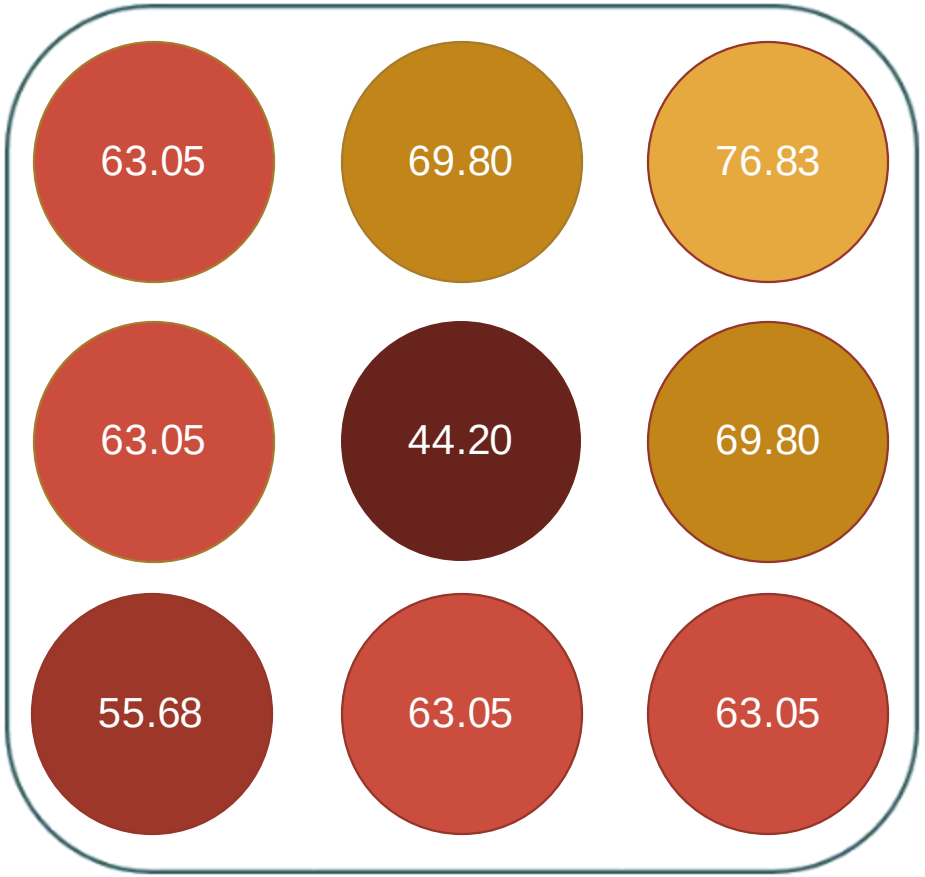


Plutonium content @ 10 years cooling



Solution: Relative change in decay heat using pin-specific libraries (compared to generic library)

Decay heat (W) @ 10 years cooling with pinwise libraries

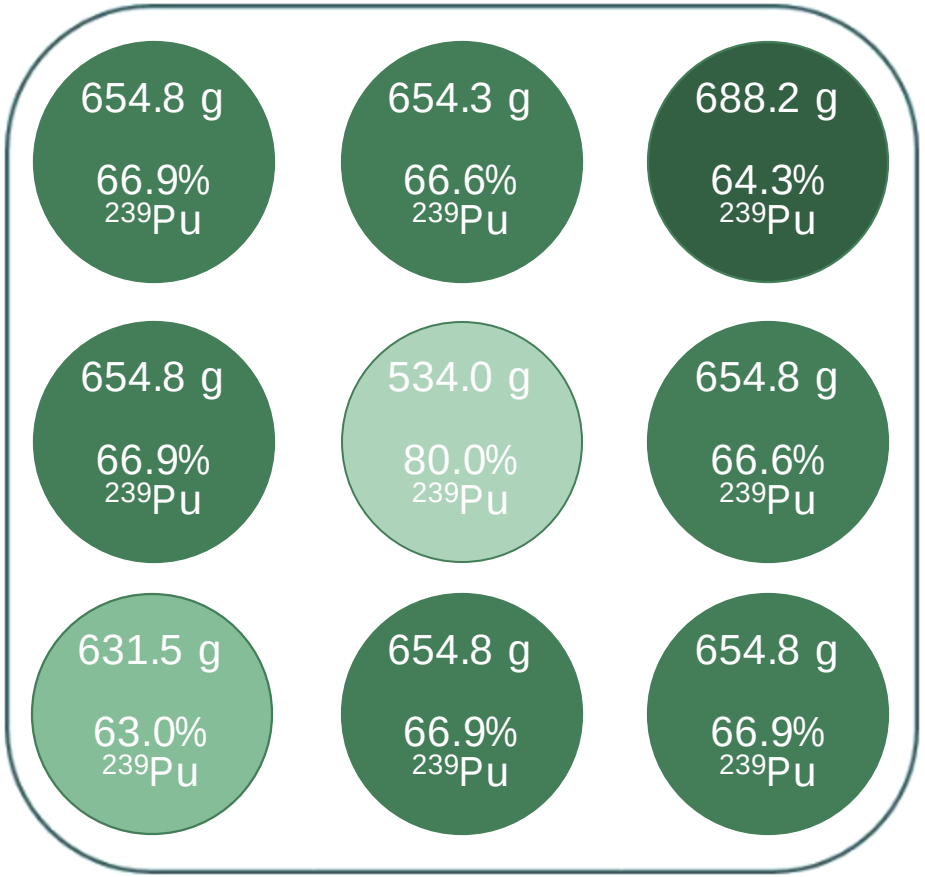


Decay heat (pinwise vs. generic library)

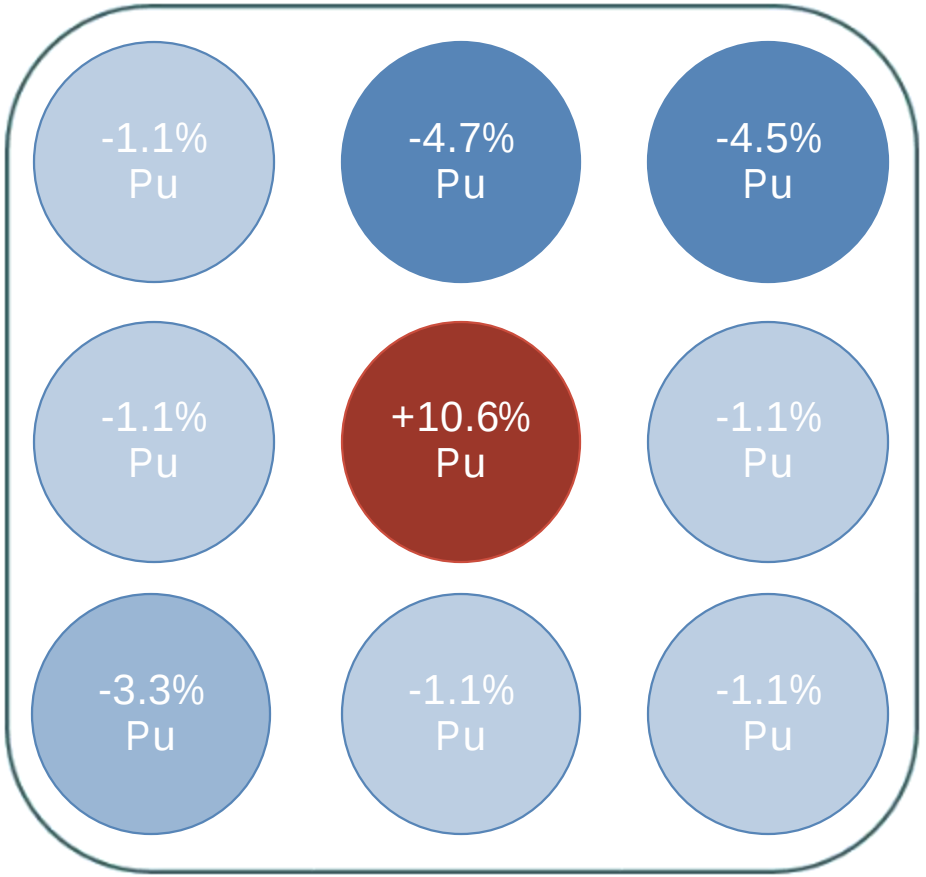


Solution: Relative change in plutonium using pin-specific libraries (compared to generic library)

Plutonium content @ 10 years cooling with pinwise libraries



Plutonium content (pinwise vs. generic library)



OAK RIDGE National Laboratory

Workshop: Using ORIGAMI with a custom library – SFR Demo



Advanced burner reactor with HALEU fuel

- Consider a fast reactor fueled by 16.5% enriched U-Zr fuel, burned to a discharge burnup of 150 GWd/MTU
  - Use the provided "ABTR_haleu" library & arpdata.txt
    - Copy into ${TMPDIR} using shell before your ORIGAMI input
  - Assume a mixture of 90% U, 10% Zr with a density of 11.3726 g/cm^3
  - Assume a specific power of 46.21523 MW/MTU
- Calculate the decay heat per MTU from 1 year to 10,000 years post-discharge
- Compare your time-dependent decay heat to an 5.0% enriched UO_2 LWR assembly irradiated to 45 GWd/MTU
  - Assume 40 MW/MTU specific power, w17x17 library

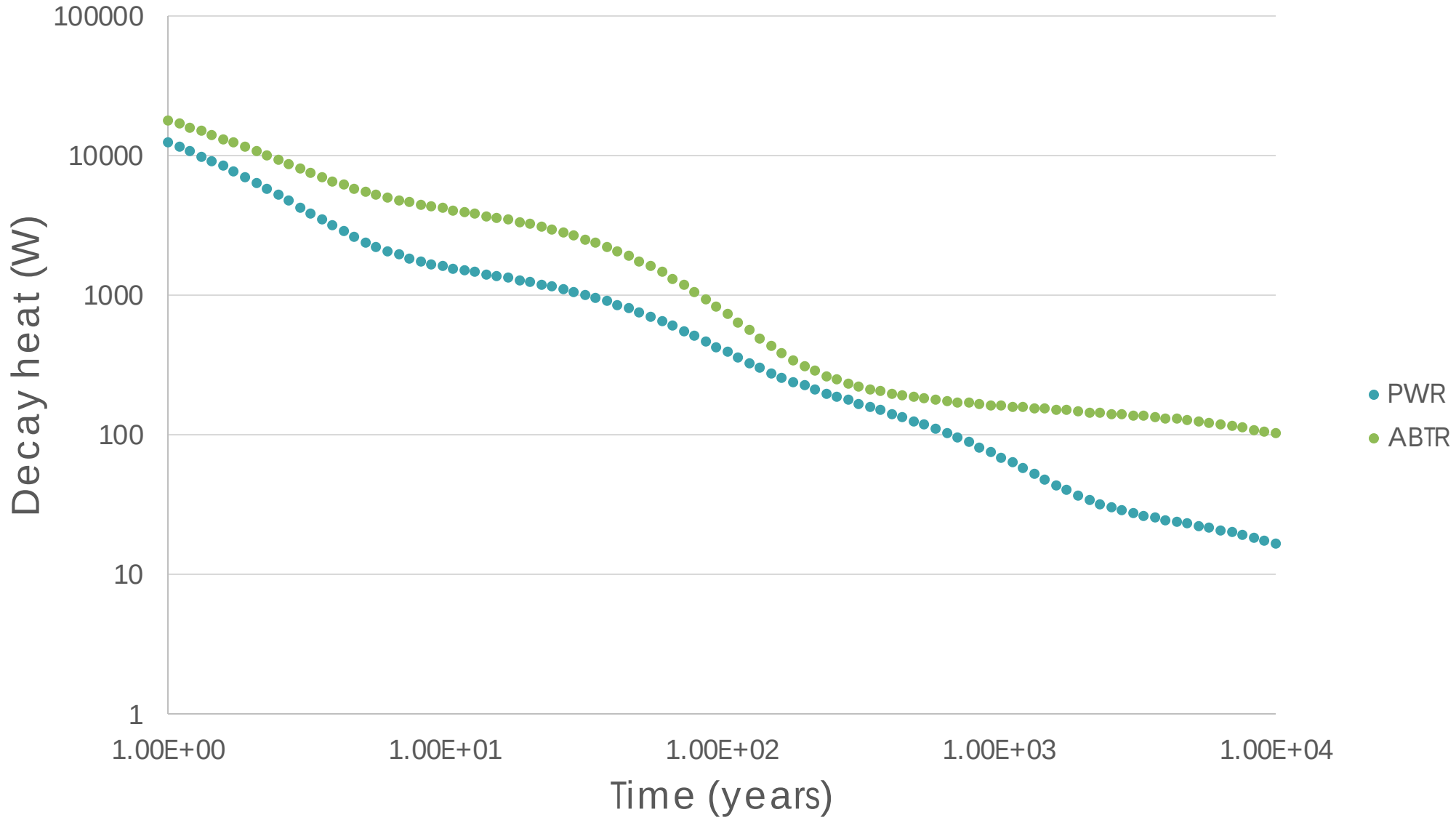
Tip: Use a log-log scale for decay heat comparisons over very long time intervals

OAK RIDGE National Laboratory

Solution



Decay heat comparison: PWR vs. ABTR



OAK RIDGE National Laboratory

Options for generating files for follow-on analyses



Options for generating data files for follow-on analysis

- Within options block
  - stdcomp: Generates SCALE standard composition files (*.compBlock) for axial zone compositions, for burnup credit nuclides (yes/no; default: no)
  - fdens: Sets fuel density (used for conversion to atom density for MCNP & SCALE standard composition outputs; default:10.4 g/cm^3)
  - temper: Sets fuel temperature in K (used for SCALE standard composition outputs; default: 293 K)
  - decayheat: Generates text output (*.AxialDecayHeat) of decay heat in Watts by axial zone (yes/no; default: no)
  - mcnp: Generates text output of material and source cards in MCNP input format (yes/no; default: no)

Nuclides included in .compBlock file

- Light elements: ^{16}O
- Actinides

  ^{234}U, ^{235}U, ^{236}U, ^{238}U, ^{237}Np, ^{238}Pu, ^{239}Pu, ^{240}Pu, ^{241}Pu, ^{242}Pu, ^{241}Am, ^{243}Am
- Fission products

  ^{95}Mo, ^{99}Tc, ^{101}Ru, ^{103}Rh, ^{109}Ag, ^{133}Cs, ^{143}Nd, ^{145}Nd, ^{147}Sm, ^{149}Sm, ^{150}Sm, ^{151}Sm, ^{152}Sm, ^{151}Eu, ^{153}Eu, ^{155}Gd

Gamma/neutron source options: srcopt block

- ORIGAMI allows for a limited subset of ORIGEN's source printing & calculation options (key/value pairs, defaults highlighted)

brem_medium: medium used for bremsstrahlung calculation (gamma)
- **uo2** (uranium oxide matrix)
- h2o (water)
- none (no bremsstrahlung calculation)

alphan_medium: medium used for (a,n) neutron calculation (neutron)
- uo2 (uranium oxide matrix)
- boroslicate (borosilicate glass)
- **case** (problem-specific mixture)

sublib: calculate / print gamma source terms by sublibrary
- lt (light elements)
- fp (fission products)
- ac (actinides)
- **all** (all nuclides)

print: print gamma/neutron source term to text output
- yes: generate text output files (*.AxialGammaSpec, *.AxialNeutronSpec) of gamma/neutron sources by axial zone
- **no:** sources only output to ft71 binary output

Examples of generated files for MCNP follow-on analyses

*.MCNP_matls.inp

```
C  Axial zone: 01, Pin: 001
C  Zone mass (grams):  9.073507E+04
m101       2004  -2.000317E-11
           8016  -1.182644E-01
           8017  -4.787782E-05
          82207  -3.165821E-18
          88223  -5.555578E-19
          88226  -4.426406E-15
          89227  -3.933551E-16
          90227  -9.115049E-19
          90230  -9.799348E-10
          90232  -5.310030E-12
          90234  -1.222845E-11
          91231  -3.824823E-11
          92234  -3.531152E-04
          92235  -3.967597E-02
          92236  -1.825094E-04
          92238  -8.414761E-01
C  Axial zone: 02, Pin: 001
C  Zone mass (grams):  9.073421E+04
m102       1001  -3.043445E-10
…
```

*.MCNP_gamma.inp

```
C  Gamma source for axial zone 01, pin 001
C  Total intensity (g/sec):  2.7917E+09
SI101     H       1.0000E-08  5.0000E-01  1.0000E+00  2.0000E+00
      1.0000E+01
SP101     D  0.0  9.8774E-01  8.1040E-03  4.1530E-03  1.4567E-06
C  Gamma source for axial zone 02, pin 001
C  Total intensity (g/sec):  1.0395E+15
SI102     H       1.0000E-08  5.0000E-01  1.0000E+00  2.0000E+00
      1.0000E+01
…
```

*.MCNP_neutron.inp

```
C  Neutron source for axial zone 01, pin 001
C  Total intensity (n/sec):  1.1489E+03
SI101     H       1.0000E-08  1.0000E-05  1.0000E-03  1.0000E-02
      1.0000E-01  1.0000E+00  2.0000E+01
SP101     D  0.0  1.3872E-08  1.4103E-05  4.3280E-04  1.3448E-02
      3.0902E-01  6.7708E-01
C  Neutron source for axial zone 02, pin 001
C  Total intensity (n/sec):  2.0845E+04
SI102     H       1.0000E-08  1.0000E-05  1.0000E-03  1.0000E-02
      1.0000E-01  1.0000E+00  2.0000E+01
…
```

Example of .compBlock file for SCALE follow-on analyses

```
o-16     1001  0  4.6291E-02   293.0 end
u-234    1001  0  6.6649E-06   293.0 end
u-235    1001  0  3.8451E-04   293.0 end
u-236    1001  0  1.4447E-04   293.0 end
u-238    1001  0  2.1426E-02   293.0 end
np-237   1001  0  1.3378E-05   293.0 end
pu-238   1001  0  4.4772E-06   293.0 end
pu-239   1001  0  1.4923E-04   293.0 end
pu-240   1001  0  5.1931E-05   293.0 end
pu-241   1001  0  2.0601E-05   293.0 end
pu-242   1001  0  1.0086E-05   293.0 end
am-241   1001  0  1.3683E-05   293.0 end
am-243   1001  0  1.8190E-06   293.0 end
mo-95    1001  0  5.5250E-05   293.0 end
tc-99    1001  0  5.5607E-05   293.0 end
ru-101   1001  0  5.1086E-05   293.0 end
rh-103   1001  0  3.0394E-05   293.0 end
ag-109   1001  0  3.8543E-06   293.0 end
cs-133   1001  0  5.7415E-05   293.0 end
nd-143   1001  0  4.1092E-05   293.0 end
nd-145   1001  0  3.2112E-05   293.0 end
sm-147   1001  0  1.1819E-05   293.0 end
sm-149   1001  0  1.7224E-07   293.0 end
sm-150   1001  0  1.2831E-05   293.0 end
sm-151   1001  0  5.3905E-07   293.0 end
sm-152   1001  0  4.3198E-06   293.0 end
eu-151   1001  0  4.3855E-08   293.0 end
eu-153   1001  0  4.6953E-06   293.0 end
gd-155   1001  0  2.4911E-07   293.0 end
```

OAK RIDGE National Laboratory

Nuclear Data Libraries and Composition Descriptions

SCALE 6.3.2

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

OAK RIDGE National Laboratory

Nuclear data libraries



SCALE's nuclear data libraries

- The AMPX code system is used to prepare (almost*) all of SCALE's nuclear data libraries
- The key nuclear data libraries are as follows:
  - Continuous-energy (CE) cross section libraries
  - Multi-group (MG) cross section libraries
  - Multi-group covariance libraries (uncertainty and correlation data)
  - Other important libraries: Fission yields, decay data, activation, etc.
- SCALE 6.3 libraries were prepared based on evaluated nuclear data libraries: ENDF/B-VII.1 and ENDF/B-VIII.0
- All data libraries are stored in one directory: "data" within the SCALE installation folder
- SCALE is searching for libraries via the ${DATA} environmental variable

*There are a couple of new, small HDF5 libraries that were generated differently

Relevant cross section libraries for this training

- "v7.1" indicates ENDF/B-VII.1; "v8.0" ENDF/B-VIII.0
- CE cross section libraries:
  - *ce_v7.1_endf.xml* (old-style xml file)
  - *ce_v8.0_endf.h5* (new HDF5 format)
- MG cross section libraries:
  - *Fine-group thermal: scale.rev05.xn252v7.1*
    scale.rev01.xn252v8.0
  - *Broad-group LWR: scale.rev04.xn56v7.1*
  - *Fine-group SFR: scale.rev00.xn302v7.1, v8.0-302*
  - *Very fine-group: v7.1-1597, v8.0-1597*
  - *And many more coupled neutron–gamma libraries*
- The CE and MG libraries can be called via their explicit file name or via their alias (see later when we talk about input file development)

Typical SCALE inputs reference a library "alias" like v7.1-252 or ce_v7.1.

All library aliases are defined in the file FileNameAliases.txt in the SCALE installation directory.

Cross section libraries and their use

- CE cross section libraries:
  - Point-wise tabulated values of cross sections on fine energy mesh
  - Used by CENTRM for self-shielding and by KENO/Shift for neutron transport in CE mode
- MG cross section libraries:
  - Libraries contain flux-weighted averages of point-wise data over energy group
  - Problem-independent MG libraries are distributed with SCALE
  - Self-shielding calculation required to convert problem-independent into problem-dependent MG library for neutron transport calculations

Self-shielding must be requested through the "celldata" block, which enables XSProc calculation

Comments on CE vs. MG

- CE
  - No need for cell data
  - Accurate representation of cross sections
  - Accurate calculation of energy distribution of integral quantities such as flux and reaction rates
  - Longer execution times in Monte Carlo modules for equivalent uncertainty
- MG
  - Generally shorter overall execution times (problem dependent)
    - Always shorter times in Monte Carlo modules for equivalent uncertainty
  - Accurate resonance self-shielding is not always possible

OAK RIDGE National Laboratory

Basic standard compositions



Materials in the COMPOSITION block

- Materials or mixtures for many of SCALE's sequences (CSAS, TRITON, MAVRIC, XSProc, etc.) are defined in the COMPOSITION block
- There is a range of ways to define a mixture
- These different types have been developed to allow simple user inputs for a range of typical material descriptions
- Selecting the "right" input type can dramatically simplify user effort and the input—and it can enable easier error detection
- A mixing table listing the densities of individual nuclides in atoms/barn-cm for each mixture can be generated directly in Fulcrum to double-check mixture definitions
  - Click drop-down next to "Run", then "Mixing Table"

Standard composition types

- Basic standard compositions
  - Individual nuclides
  - Elements with user-defined isotopic distribution
  - Some frequently used compounds/alloys (UO_2, SS304, others)
  - Always check here first – these are the easiest!
- Compounds
  - User-specified compounds (atoms per molecule/compound)
- Weight percent mixtures
  - User-specified weight percent mixtures
- Fissile solutions
  - Nitrate and fluoride solutions of U, Pu, and Th as well as associated excess acid

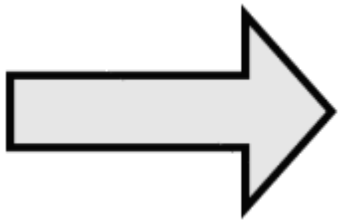
OAK RIDGE National Laboratory

Basic standard compositions



The standard composition library

- Contained in the data directory: scale.rev40.sclib
- This library contains information on isotopes, elements, compounds, alloys and mixtures:
  - SCALE IDs
  - Natural abundances of elements
  - Composition of alloys and mixtures (e.g., UO_2 or SS304)
  - Theoretical densities
  - Masses (in amu)
- The library is read and written with the compoz module
- The content is also displayed in the SCALE manual in section "Standard composition library"



Besides Fulcrum's autocomplete windows, this is where you can find out which assumptions SCALE makes for the different compounds, alloys, and mixtures.

Basic standard composition input data

- Standard composition name
- Mixture number
- Theoretical density (optional)
  - Keyworded: DEN=
- Density multiplier (optional)
- Temperature (K) (optional)
- Isotopic distribution (optional)
- end

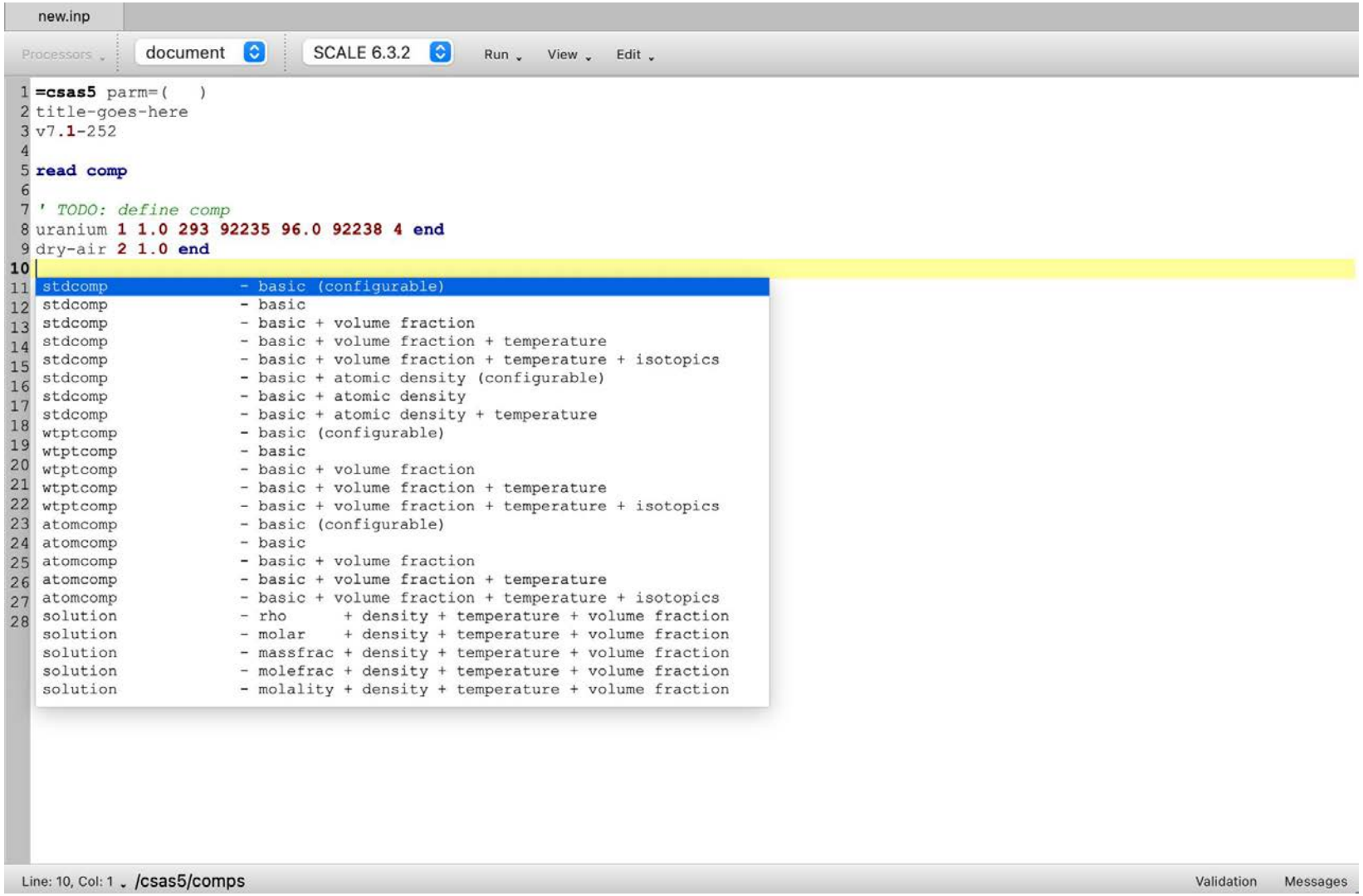
You can stop once you've supplied everything you need, but you must provide the END!

```
uranium 1 1.0 293 92235 96.0 92238 4 end
dry-air 2 end
h2o     3 den=1.0 1.0 597 end
```

Basic standard composition

- Fulcrum provides autocomplete capability
  - Ctrl-Space opens drop down
- Select "stdcomp" option with data you want to provide
  - Replace example data with appropriate data
  - OR chose "configurable" and fill in fields

Autocomplete choices

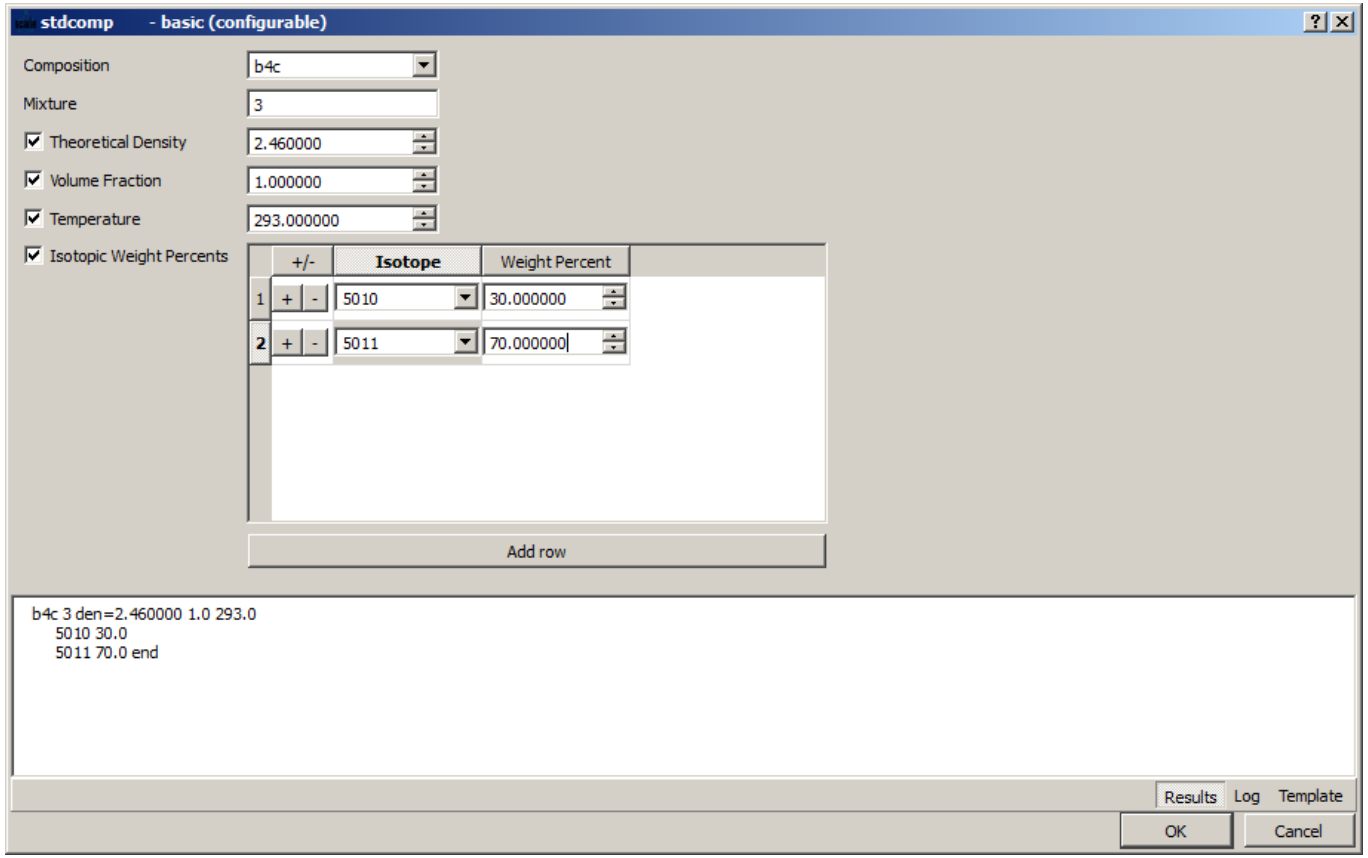


Autocomplete example

- Need to specify: B_4C as mixture 3: ρ=2.46 g/cm^3, 30 wt% ^{10}B
- Autocomplete provides:
  - `uo2 1 1.0 293 92234 0.5 92235 5.0 92238 94.5 end`
- Change to:
  - `b4c 3 DEN=2.46 1.0 293 5010 30 5011 70 end`

Configurable example

- Same mixture 3, B_4C: ρ=2.46 g/cm^3, 30 wt% ^{10}B



Simple use of density multiplier

- UO_2 pellets pressed to 96% of theoretical density (TD)
  - UO2 1 den=10.5216 1 293 end
  - UO2 1 0.96 293 end
  - den= is keyword construct
  - 0.96 is multiplied by the default TD (*or* *supplied density*) to determine density in mixture

Density and density multiplier

- Theoretical density for each material has a default value that can be changed with DEN=
- Variable name for density multiplier is VF (volume fraction), though it isn't always a volume fraction
- Density for each component is determined as TD times density multiplier
- Examples:

```
h2o 1 0.7 293 end [default ρ=0.9982]
h2o 2 den=0.69874 1 293 end
```

Mixing basic compositions with density multipliers

- Multiple basic compositions can be used in the same mixture by providing the same mixture number
- The amount of each basic composition in the overall mixture can be controlled with the density multiplier for each component
- Must be done correctly – be careful

Example for mixing basic compositions

- Consider a mixture of 80 vol% water and 20 vol% UO_2
- Input would look like:

```
h2o          2 0.8 300   end
uo2          2 0.2 300   end
```

- Mixture 2 total mass density is calculated as:

$$\rho_{H_2O} = 0.9982 \times 0.8 = 0.79856$$
$$\rho_{UO_2} = 10.96 \times 0.2 = 2.192$$
$$\rho_{MIX=2} = 0.79856 + 2.192 = 2.99056$$

- The total mass density for each mixture is provided in the mixing table

Another approach

- Assume we know the mass density and the mass fractions
  - $\rho = 2.99056\ g/cm^3$
  - H2O wt% = 26.703
  - UO2 wt% = 73.297
- Input looks like:

```
h2o          1 den=2.99056 0.26703 300   end
uo2          1 den=2.99056 0.73297 300   end
```

- Now the density multiplier is a mass fraction!

Input for example

```
h2o          1 den=2.99056 0.26703 300   end
uo2          1 den=2.99056 0.73297 300   end

h2o          2 0.8 300   end
uo2          2 0.2 300   end
```

	Mixture 1	Mixture 2
Mass density (g/cm^3)	2.9906	2.9906
H-1 (a/b-cm)	5.338292e-002	5.338230e-002
O-16 (a/b-cm)	3.638303e-002	3.638276e-002
U-234 (a/b-cm)	2.640060e-007	2.640071e-007
U-235 (a/b-cm)	3.521722e-005	3.521737e-005
U-238 (a/b-cm)	4.853083e-003	4.853104e-003

^{2}H, ^{17}O and ^{18}O are also reported—more on that later!

Entering number densities

- Density multiplier of 0 is a flag for SCALE that the next input is a number density in atoms/barn-cm

```
h             1 0 0.0667531 300   end

pu-238        1001  0  4.747e-07  293.000  end
pu-239        1001  0  1.068e-04  293.000  end
pu-240        1001  0  1.577e-05  293.000  end
pu-241        1001  0  5.626e-06  293.000  end
pu-242        1001  0  6.187e-07  293.000  end
rh-103        1001  0  1.104e-05  293.000  end
```

Caution on default densities

- The default density for all isotopes is 1.0 g/cm^3 and, thus, likely not helpful

- The default density for all elements that are gases at room temperature is also 1.0 g/cm^3, which is also unlikely to be helpful

Example elemental data

ID	Symbol	Name	Mass (amu)	Density	Isotopic Distribution	Atom %
6000	c	CARBON	12.0107	2.1	6012	98.93
					6013	1.07
24000	cr	CHROMIUM	51.9961	7.2	24050	4.345
					24052	83.789
					24053	9.501
					24054	2.365
1000	h	HYDROGEN	1.0079	1.0	1001	99.9885
					1002	0.0115
26000	fe	IRON	55.845	7.86	26054	5.845
					26056	91.745
					26057	2.119
					26058	0.282
92000	u	URANIUM	238.0289	19.05	92234	0.0054
					92235	0.7204
					92238	99.2742

From Table 7.2.2 Isotopes and their natural abundances

Historical comments on isotopes

- SCALE 6.3 expands all elements to isotopes according to their natural abundance
  - If a nuclear data library does not have a cross section, then it is removed
  - O-18 is an exception, where the ENDF/B-VII.1 cross section is treated as zero
- Earlier treatment of H, O created H-1 and O-16, respectively
- Earlier treatment of Pu
  - In earlier versions, SCALE has allowed Pu to be specified without an isotopic description, and it was assumed 100% Pu-239
  - This is no longer allowed; any material in the standard comp library including Pu (PuO_2, PuC, $Pu(NO_3)_4$, PuF_4, PuN) needs a user-specified isotopic distribution
  - Pu isn't recognized (it has no natural abundance)
    - Plutoniumalp/plutoniumdlt or isotopes can be used
    - Configurable window does not provide 94AAA options for isotopic composition
    - Type nuclide IDs (e.g., 94239) into fields directly

Example compound data

Name	Description	Density	ID	Atoms Per Molecule
al2o3	Aluminum oxide	3.97	13000	2
			8000	3
b4c	Boron carbide: B4C; natural Isotopic distribution obtained by default	2.52	5000	4
			6000	1
balsa	Balsa wood: $C_6H_{10}O_5$	0.125	6000	6
			1000	10
			8000	5
benzene	Benzene with a S(α,β) thermal kernel	0.8765	5006000	6
			6001001	6

From Table 7.2.4 Compounds

Example alloy/mixture data

Name	Description	Density	Nuclide	Weight %
CARBONSTEEL	Carbon steel	7.8212	26000 6000	99.0 1.0
REG-CONCRETE	Regulatory concrete (developed for U.S. NRC)	2.3	26000 1000 13000 20000 8000 14000 11000	1.4 1.0 3.4 4.4 53.2 33.7 2.9
SS304S	Stainless steel-304 (using standard nuclides instead of special weighted nuclides)	7.94	6000 14000 15000 24000 25000 26000 28000	0.08 1.0 0.045 19.0 2.0 68.375 9.5

From Table 7.2.5 Alloys/mixtures

Void

- Mixture 0 is always void
  - Does not need to be specified in the composition block
- Void has no cross section – it is simply empty space
- Void can be specified in geometry data as necessary
- Volumes within the model that are void must be specified – void is NOT assumed in undefined space
- It can be used as many times as necessary in unit cell data specifications

OAK RIDGE National Laboratory

Thermal scattering kernels



Thermal scattering kernels

- Cross sections of several nuclides are generated with thermal scattering kernels (TSL data)
- These can have a significant impact on calculated results
- The selection of TSL data is done in SCALE by choosing the correct, corresponding nuclide identifier
- Some examples:
  - H, Be, D, Fe, O, C cross sections are available with various TSLs
  - Carbon:
    - Free-gas, i.e. carbon cross section data: *c*
    - Carbon cross section data + graphite TSL data: *c-graphite*
  - Hydrogen:
    - Free-gas, i.e. hydrogen cross section data: *hfreegas*
    - Hydrogen cross section data + TSL data is available for various options: *h* (h in h2o), *h-benzene, h-cryo_ortho, h-cryo_para, h-liquid_ch4, h-poly, h-solid_ch4, h-zrh2*
- See STDCMP section of the manual for available cross sections

H-poly data thermal scattering

- SCALE 6.2 and earlier has incorrect h-poly due to an error in ENDF
  - Note that the default *h-1* is tied to thermal scattering for H_2O and was processed correctly
- Error in scattering can cause k_{eff} mispredictions of 1% or more
  - Patch available for SCALE 6.2 CE data from https://code.ornl.gov/scale/sse/sdl-2021-024/-/releases
- With new ENDF/B-VIII.0 TSLs, this error also applies to h-lucite and si-*
- Assessment has shown that as long as CENTRM is used in XSProc (instead of BONAMI), the error is significantly reduced compared to 6.2
- CE h-poly, h-lucite, and si-* data are fixed in SCALE 6.3.1 data patch
- MG h-poly and h-lucite data are fixed in SCALE 6.3.2 data patch

OAK RIDGE National Laboratory

New data / data relevant for advanced reactors



Graphite

- Graphite = carbon cross section data + graphite TSL data
- ENDF/B-VII.1:
  - Carbon available only as element (*last element for which this was the case!*): c
  - Graphite TSL data only available for graphite as perfect crystal, i.e., 0% porosity: c-graphite or graphite
- ENDF/B-VIII.0:
  - Carbon isotopic data: c-12, c-13, c*
  - Graphite TSL data available for graphite at different porosities:
    - 0%: c12-graphite, c13-graphite, c-graphite*
    - 10%: c12-10_graph, c13-10_graph, c-10_graph*
    - 30%: c12-30_graph, c13-30_graph, c-30_graph*

We recommend the use of 0% porosity data. See K. Ramic, et al. (2025), Carbon, 244, 120619, 10.1016/j.carbon.2025.120619

- Note: Graphite data with 20% porosity planned to be included in future ENDF/B releases

* Note: Shortcut "c" still works with ENDF/B-VIII.0, but will be expanded internally to a combination of c-12 and c-13 data according to their natural abundances.

New TSL data in ENDF/B-VIII.0

- ENDF/B-VII.1: SiO_2, ZrH_2
- ENDF/B-VIII.0: SiO_2, SiC, ZrH_2, YH_2

OAK RIDGE National Laboratory

Exercise: Basic standard composition



Exercises

- We'll do one example together for each group of exercises
- Keep your inputs somewhere, as many of the example specifications are drawn from the exercises you'll be working on later
- We'll create models using the T-XSEC sequence with the "v7-56" library

Problem 1 – together

- Uranium metal
  - Density = 18.74 g/cm^3
  - Mixture temperature: 293 K
  - Enrichment:
    - 1.0202 w/o ^{234}U
    - 93.7112 w/o ^{235}U
    - 5.2686 w/o ^{238}U

Problem 2

- Aluminum metal
  - Density = 2.702 g/cm^3
- Uranium oxide
  - 95% theoretical density
  - Mixture temperature: 293 K
  - Enrichment:
    - 4.95 w/o ^{235}U
    - 95.05 w/o ^{238}U

Problem 3

- Homogenous mixture of UF_4 and paraffin
  - Mixture density 3.8655 g/cm^3
  - UF_4 – 88.62 wt%
    - Enrichment:
      - 2.0 w/o ^{235}U
      - 98.0 w/o ^{238}U
  - Paraffin – 11.38 wt%
  - Mixture temperature 27 °C

Remember, the density multiplier functions as a *mass* fraction if the overall mixture density is specified.

Problem 3a

- Homogenous mixture of UF_4 and paraffin
  - UF_4 – 51.13 vol%
    - Enrichment:
      - 2.0 w/o ^{235}U
      - 98.0 w/o ^{238}U
  - Paraffin – 48.87 vol%
  - Mixture temperature 27 °C

Remember, the density multiplier can function as a volume fraction.

OAK RIDGE National Laboratory

Exercise solutions: Basic standard compositions



Problem 1

```
u            1 den=18.74 1 293
                                92235 93.7112
                                92238 5.2686
                                92234 1.0202   end
```

Problem 2

```
al          2 end
uo2         3 0.95 293
                                92235 4.95
                                92238 95.05   end
```

Problem 3

```
uf4          4 den=3.8655 0.8862 300.15
                                 92235 2
                                 92238 98   end
paraffin     4 den=3.8655 0.1138 300.15   end
```

Problem 3a

```
uf4          4 0.5113 300.15
                              92235 2
                              92238 98   end
paraffin     4 0.4887 300.15   end
```

OAK RIDGE National Laboratory

Compounds



Atom compounds

- atom*name* – "atom" prefix is required as a flag to SCALE
- Mixture number
- Theoretical density
- Number of elements
- Nuclide ID and number of atoms
- Density multiplier
- Temperature (K) (optional)
- Isotopic distribution (optional)
- end

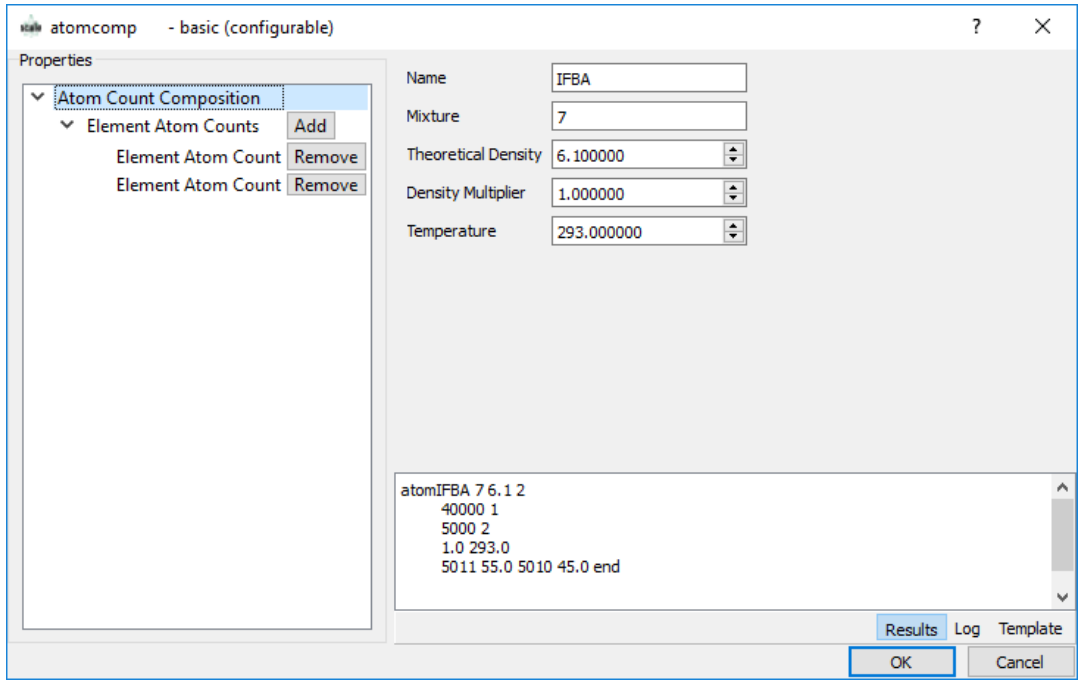
IFBA – ZrB_2 at 6.1 g/cm^3 (45 wt% ^{10}B)

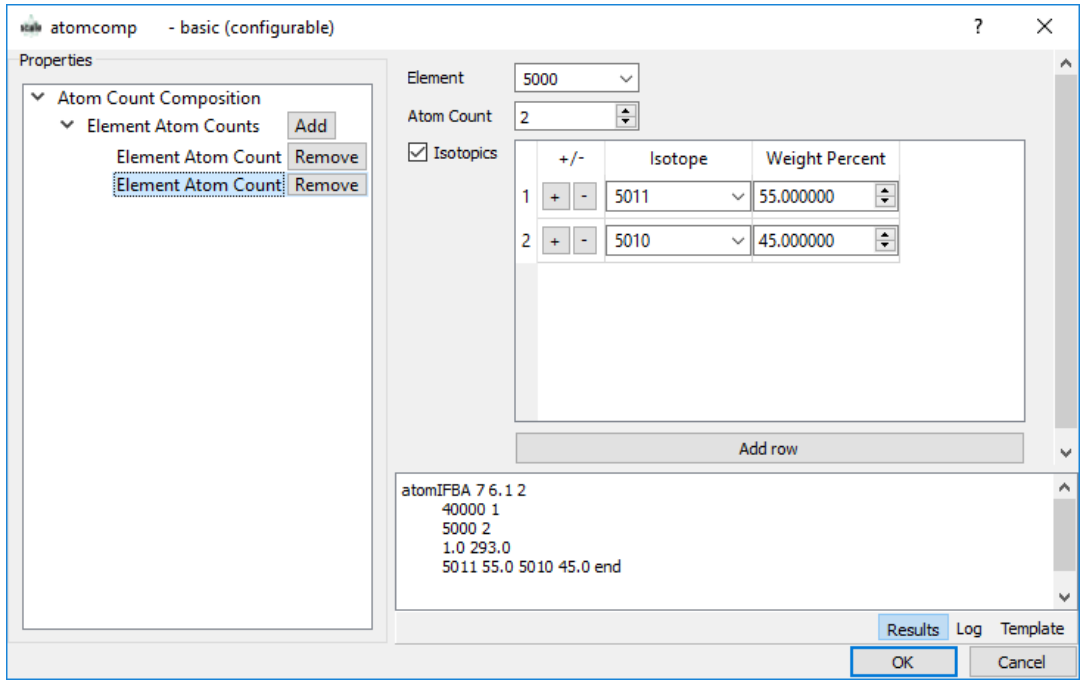
```
atomIFBA 7 6.1 2
     40000 1
      5000 2
     1.0 293
     5010 45 5011 55
     end
```

Integers required

Atom compounds

- Autocomplete and configurable also work for atom compounds (atomcomp)
- Default atomcomp is "atomuo2"





OAK RIDGE National Laboratory

Weight percent mixture



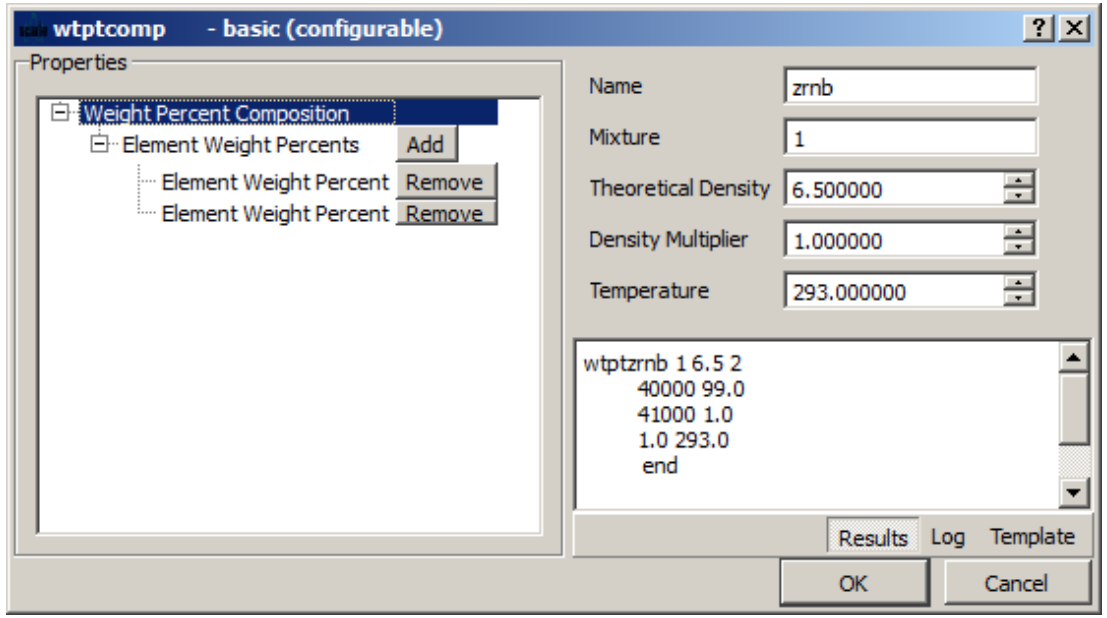
Weight percent mixture

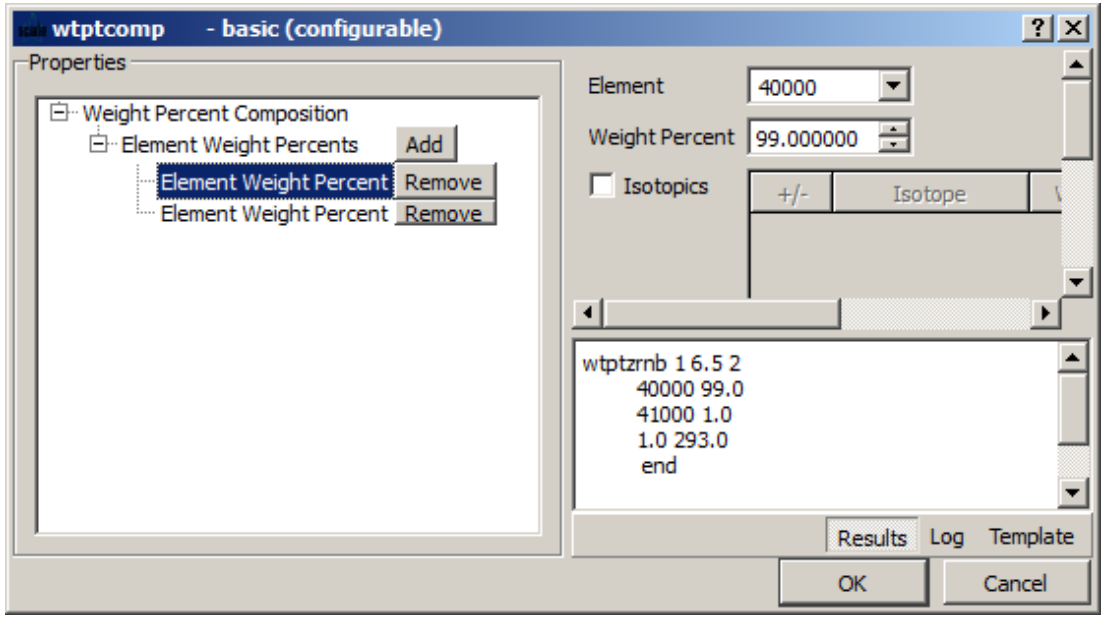
- wtpt*name* – "wtpt" prefix is required as a flag to SCALE
- Mixture number
- Theoretical density
- Number of elements
- Nuclide ID and weight percent
- Density multiplier
- Temperature (K) (optional)
- Isotopic distribution (optional)
- end

```
wtptzrnb 1 6.5 2
     40000 99.0
     41000 1.0
     1.0 293
     end
```

Weight percent mixture

- Autocomplete and configurable also work for weight percent mixtures (wtptcomp)
- Default wtptcomp is “wtptuo2”





OAK RIDGE National Laboratory

Exercise: Compound and weight percent mixture



Problem 1 – together!

- Sodium bicarbonate ($NaHCO_3$)
  - Temperature 22 °C
  - Density = 2.1 g/cm^3

Problem 2

- Calcium fluoride (CaF_2)
  - Temperature 610 °F (approximately 594 K)
  - Density = 3.18 g/cm^3
  - Enrichment:
    - 96 w/o ^{40}Ca
    - 4 w/o ^{48}Ca

Problem 3

- Concrete
  - Overall density 2.32 g/cm^3
  - Constituent elements (w/o)
    - 48.49 w/o O (8000)
    - 23.31 w/o Ca (20000)
    - 15.5 w/o Si (14000)
    - 5.52 w/o C (6000)
    - 2.17 w/o Al (13000)
    - 1.37 w/o K (19000)
    - 1.25 w/o Mg (12000)
    - 1.01 w/o Fe (26000)
    - 0.75 w/o H (1000)
    - 0.63 w/o Na (11000)
  - Material temperature 293 K

OAK RIDGE National Laboratory

Exercise solutions: Compound and weight percent mixture



Problem 1

```
atomnahco3 1 2.1 4
       11000 1
       1000 1
       6000 1
       8000 3
              1.0 295.15
end
```

Problem 2

```
atomcaf2          7  3.18  2
                                    20000 1
                                    9000 2
                         1.0 594.26
                                    20040 96
                                    20048  4   end
```

Problem 3

```
wtptconc          6   2.32   10
                                      26000 1.01
                                       1000 0.75
                                       6000 5.52
                                       8000 48.49
                                      12000 1.25
                                      13000 2.17
                                      14000 15.5
                                      19000 1.37
                                      11000 0.63
                                      20000 23.31
                          1.0 293     end
```

Concrete is a common material, but no two concretes are the same. Without an assay of the particular concrete relevant to your safety case, studies should be performed to identify a conservative composition. Magnason's concrete is a fictitious composite intended to bound all real concretes. See also recent work by Richard Taylor and EPRI on generic concrete compositions.

OAK RIDGE National Laboratory

Fissile solutions



Fissile solution components

Component Name		Nuclides in component
$UO_2(NO_3)_2$	Nitrate solutions	92000 8000 7000
$Th(NO_3)_4$		90000 8000 7000
$Pu(NO_3)_4$		94000 8000 7000
HNO_3		1001 7000 8000
UO_2F_2	Fluoride solutions	92000 9019 8000
ThF_4		90000 9019 8000
PuF_4		94000 9019 8000
HF		1001 9019

From Table 7.2.6. Table of available solution components

Fissile solution input

- Most complicated composition input
- Multiple components are allowed/used in each solution
  - One or two salts and potentially excess acid
  - HNO_3 for nitrate solutions and HF for fluoride solutions
- *solution … end solution* construct within composition block
- Many inputs keyworded within solution block

Fissile solution input

```
SOLUTION MIX=5 RHO[UO2(NO3)2]=225 92235 91 92238 8 92234 0.8 92236 0.2
MOLAR[HNO3]=0.6 TEMPERATURE=293 END SOLUTION
```

- MIX=mixture number – only mixture type with keyword
- RHO[]=mass of fissile component per liter of solution, in this case gU/L, gPu/L, or gTh/L
  - Followed by enrichment of fissile component
- MOLAR[]=excess acid molarity
- TEMPERATURE=temperature (K)
  - Temperature must be > 273.15 K → Don't freeze the solution!

Fissile solution input (cont.)

- Overall solution density can also be input
- SCALE can also solve for the density if "?" is entered: `density=?`
- Isotopic distributions can follow any other specification
  - For example, uranium enrichment on previous slide
- One or two fuel salts are allowed
  - Excess acid can be input as third component if needed

OAK RIDGE National Laboratory

Exercise: Fissile solutions



Problem 1 – together!

- UO_2F_2 solution
  - Specific gravity 2.0289
    - Assume water density at lab conditions is exactly 1 g/cm^3
  - 907.01 gm U/l
  - Enrichment:
    - U-235 5.0 %
    - U-238 95.0 %

Problem 2

- Uranyl nitrate solution
  - Unknown density – let SCALE calculate it
  - 334.77 gU/l
  - Excess nitric acid 0.521 M
  - Enrichment:
    - 1.022 w/o ^{234}U
    - 93.172 w/o ^{235}U
    - 0.434 w/o ^{236}U
    - 5.372 w/o ^{238}U

OAK RIDGE National Laboratory

Exercise solutions: Fissile solutions



Problem 1

```
solution mix=1 rho[uo2f2]=907.01 92235 5.0 92238 95
               density=2.0289 temp=293 volfrac=1.0
         end solution
```

Problem 2

```
solution   mix=5  rho[uo2(no3)2]=334.77 92234 1.022
                                        92235 93.172
                                        92236 0.434
                                        92238 5.372
                  molar[hno3]=0.521
                  density=? temperature=293 vol_frac=1
 end solution
```

OAK RIDGE National Laboratory

Composition specification summary



SUMMARY

- SCALE includes several methods for defining mixtures for use in models
- Many of the mixture specification options are tailored to ease input based on typical real-world material descriptions
- SCALE sequences automatically perform number density calculations for mixture definitions of other types
- The mixing table should be used to double-check the input

```
read comp
h2o 1 den=2.99056 0.26703 300 end
uo2 1 den=2.99056 0.73297 300 end
'
h2o 2 0.8 300 end
uo2 2 0.2 300 end
'
b4c 3 den=2.460000 1.0 293.0
      5010 30.0
      5011 70.0 end
'
atomIFBA 4 6.1 2
         40000 1
         5000 2
         1.0 293.0
         5010 45.0 5011 55.0 end
'
wtptzrnb 5 6.5 2
         40000 99.0
         41000 1.0
         1.0 293.0 end
'
solution mix=6 rho[uo2f2]=1.0 92235 4.5
92238 95.5
     density=1.0 temp=300 volfrac=1.0
end solution
end comp
```

OAK RIDGE National Laboratory

KENO-VI geometry

SCALE 6.3.2

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Outline

- Sub-Module 1: KENO-VI geometry basics
  - Exercises: Lady Godiva, Green Block, Rocky Flats
- Sub-Module 2: Geometry modification data
  - Exercises: Y30, Hopper, Cross
- Sub-Module 3: HOLEs in KENO-VI
  - Exercise: 2c8
- Sub-Module 4: Arrays in KENO-VI
  - Exercises: 2c8, Pins in a Tank, 4A-4M, GOSLAR cask, CASTOR cask
- Sub-Module 5: Randomgeom block
  - Exercise: HTGR pebble

Learning Objectives

- Describe the KENO-VI geometry syntax and logic
- Apply geometry modifications in KENO inputs
- Apply holes and arrays for repetitive and advanced geometries
- Apply KENO and new Shift capabilities

OAK RIDGE National Laboratory

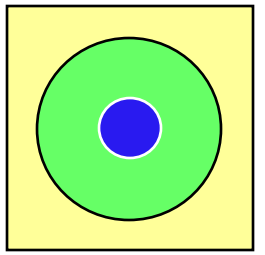
Sub-module 1

- KENO-VI geometry basics

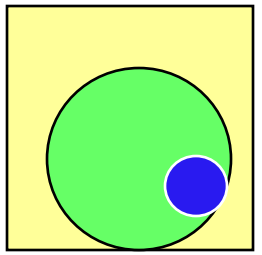


KENO-VI geometry basics

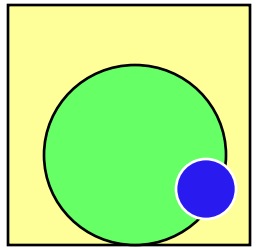
1. *Volumes are built in sections called UNITS*. Each UNIT is independent of all other units and has its own coordinate system.
2. *Units are built using REGIONS*. Regions are made using available KENO-VI geometry shapes. The unit boundary fully encloses all defined regions in the unit; it may truncate regions in the unit.
3. *Regions may share boundaries and may intersect.* (Not allowed in KENO V.a.)
4. *Regions may be rotated.* (Not allowed in KENO V.a.)



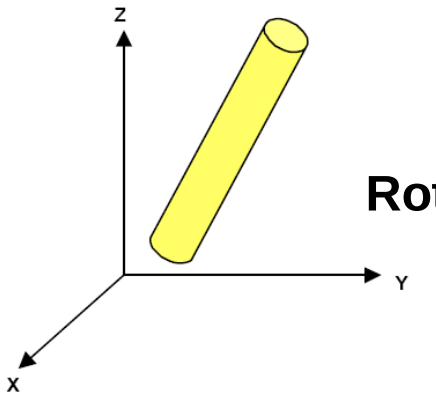
Regions in a unit



Shared boundaries



Intersecting regions



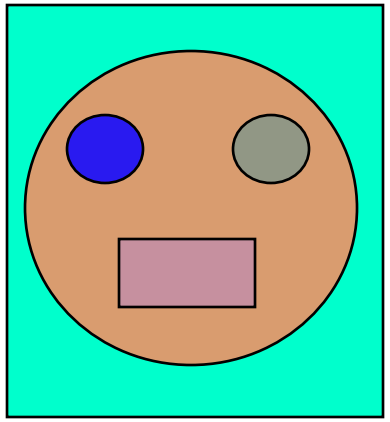
Rotated cylinder

KENO-VI geometry basics (cont.)

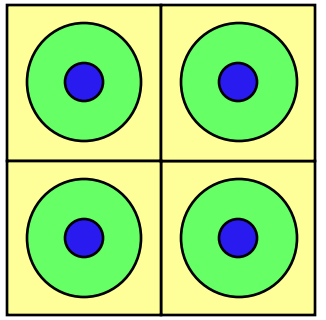
5. *A HOLE is used to place a unit within a region of a different unit*. The HOLE must be completely contained within the unit and may not intersect other holes or nested arrays. As many HOLEs as required may be placed in a unit.

6. *An ARRAY is an ordered stack of units*. The touching faces of adjacent units in an array must be the *same shape* and *size*. Multiple arrays may be placed directly in a UNIT, and arrays can also be placed within a unit using HOLEs. The BOUNDARY record of each unit must be appropriate for the selected array type.

7. *A GLOBAL UNIT that encloses the entire system must be specified.* All geometry data used in a problem are correlated by the global unit coordinate system.



Unit Containing HOLES



Array of units

KENO-VI input records

- Three types of records are used to define the volume, position, and material contents of the regions in a unit
  - Geometry record
  - Contents record
  - Boundary record

- Each unit may have as many Geometry and Contents records as needed but must have one and only one Boundary record

Unit geometry keywords

- GLOBAL
  - Specifies global/outer-most unit in problem
  - Precedes the keyword UNIT
- UNIT
  - Specifies a new independent set of geometries and materials
- COM=
  - Specifies a comment record, one per unit
  - Must begin and end with the same delimiter

Example input layout for a UNIT

```
GLOBAL UNIT 1
COM=!This is a comment field!
Geometry Records
Media Records
Boundary Record*
```

*It is a best practice (and more or less required for Fulcrum) to put the boundary record at the end of the unit.

Geometry record

- Specifies the volumes that make up the unit
- Comprises the following components:
  - Geometry keyword
  - Geometry record label
  - Geometry boundary definitions
  - Geometry modification data

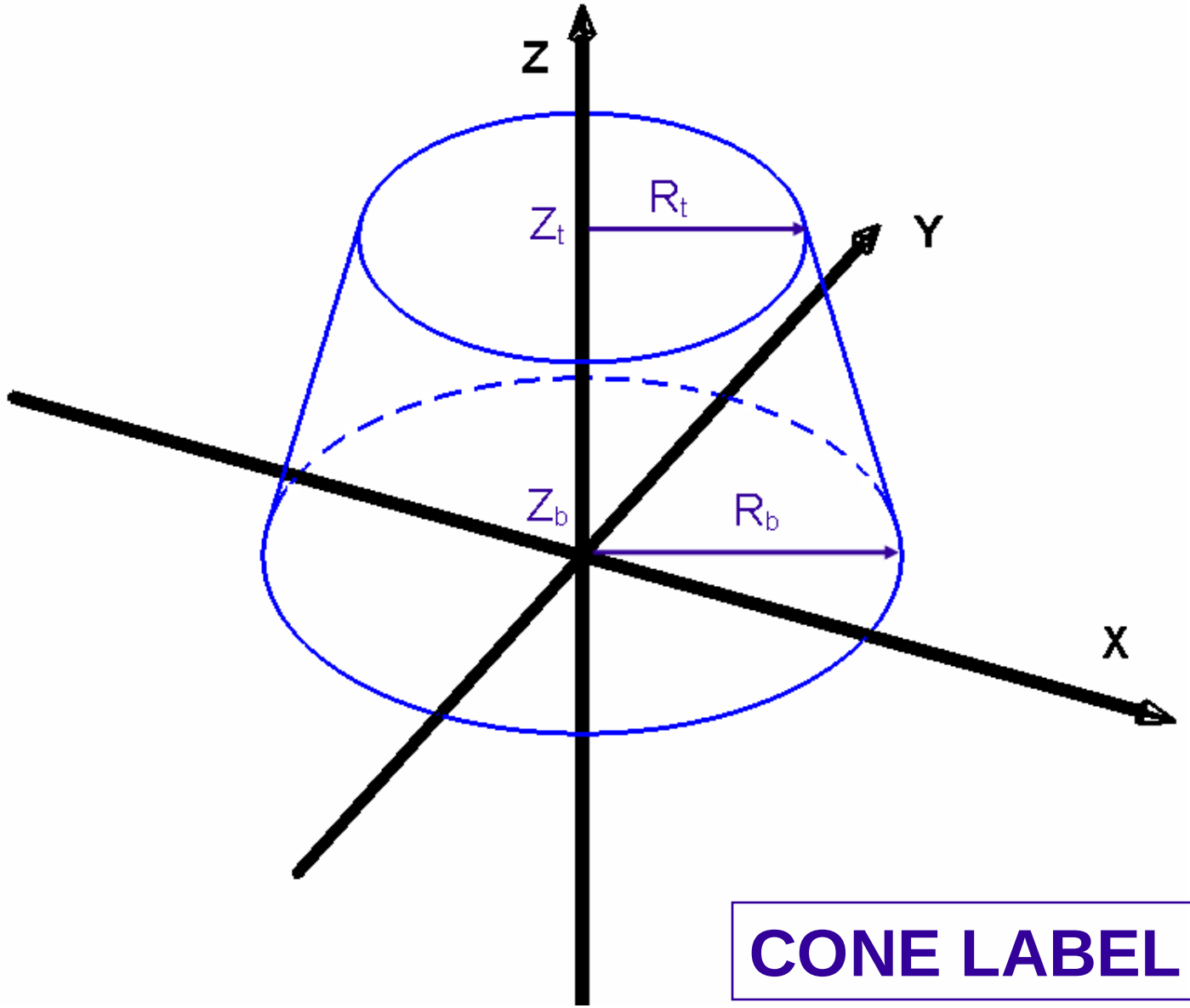
KENO-VI geometry input

KENO-VI Geometry Keyword	Geometry Record Input
CONE	Label, Top and Bottom Radius, Top and Bottom Z-Positions
CUBOID	Label, +X, -X, +Y, -Y, +Z, -Z, Positions
CYLINDER (also XCLYINDER or YCYLINDER)	Label, Radius, Top and Bottom Z-Position
DODECAHEDRON	Label, Inner Radius
ECYLINDER	Label, X-Radius, Y-Radius, Top and Bottom Z-Positions
ELLIPSOID	Label, X-Radius, Y-Radius, Z-Radius
HEXPRISM, RHEXPRISM	Label, Inner Radius, Top and Bottom Z-Positions
HOPPER	Label, Top Half Length along X & Y axis, Bottom Half length along X & Y axis, Top and Bottom Z-Positions

KENO-VI geometry input (cont.)

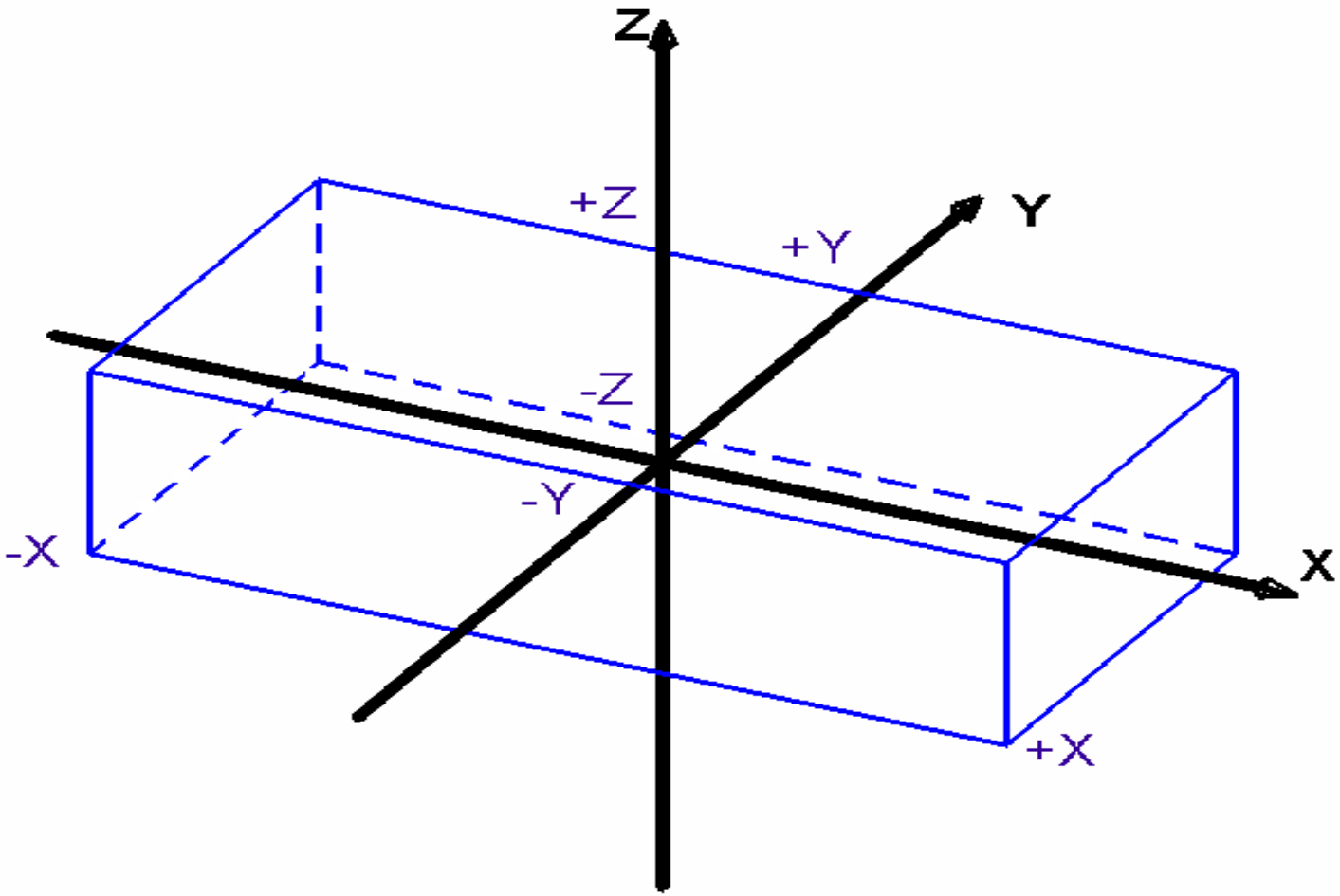
KENO-VI Geometry Keyword	Geometry Record Input
PARALLELPIPED (PPIPED)	Label, Distance along X, Y, and Z axes Angles between the X, Y, and Z faces and axes
PENTAGON	Label, Radius, Top and Bottom Z-Position
PLANE	Label, Non-Zero Coefficients of the Quadratic
QUADRATIC	Label, Non-Zero Coefficients of the Quadratic
RHOMBOID	Label, Length of a Side, Angle between the Y-Edge of the Base and the Y-axis
SPHERE	Label, Radius
WEDGE	Label, Length of triangle base along X-axis X & Y position of the triangle top Length of the Wedge along the Z-axis
XPPLANE, YPPLANE, ZPPLANE	Label, Max. and Min. positions on axis
RING	Inner and outer radii, +Z –Z positions

Cone



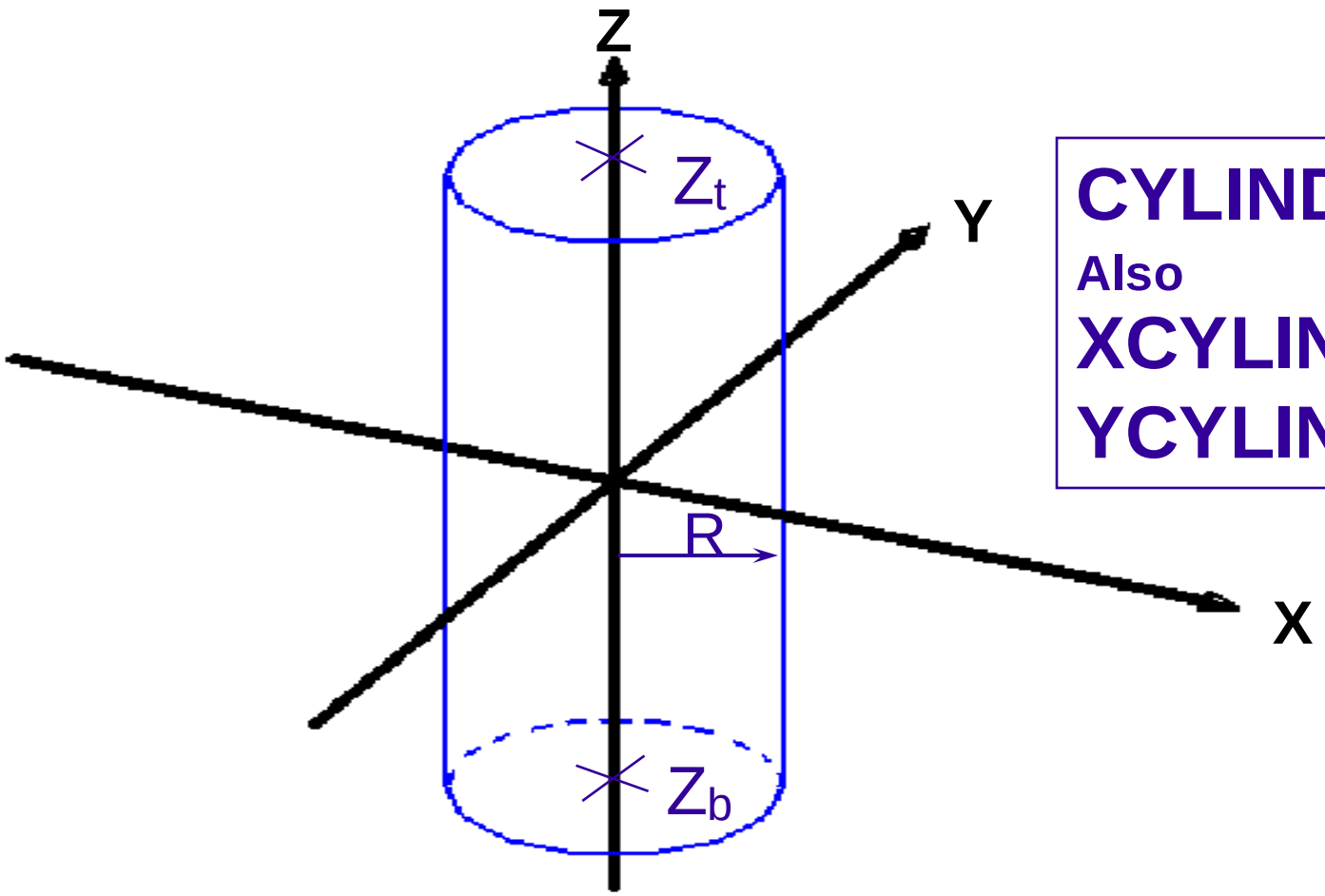
CONE LABEL R_t Z_t R_b Z_b

Cuboid



CUBOID LABEL +X -X +Y -Y +Z -Z

Cylinder



CYLINDER LABEL R Z_t Z_b

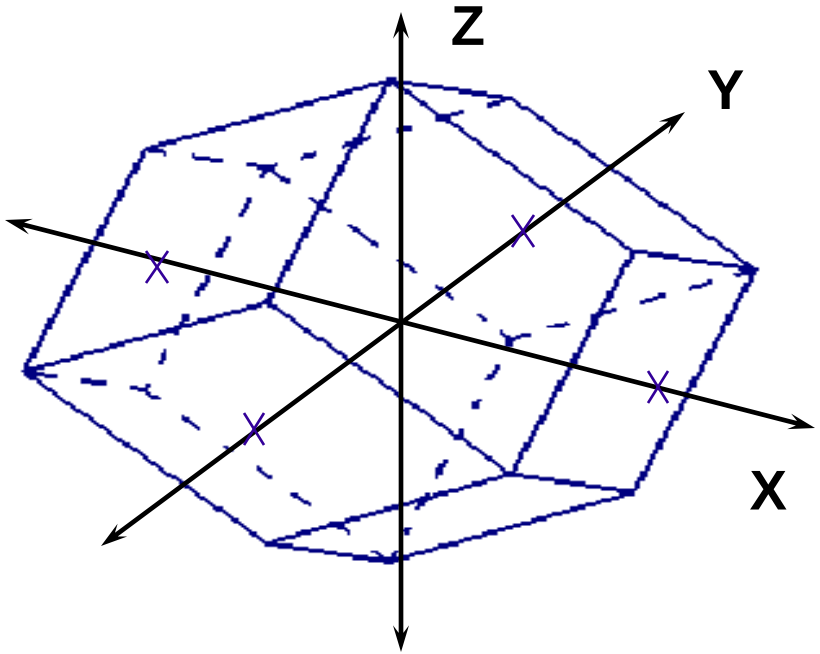
Also

XCYLINDER LABEL R Xt Xb

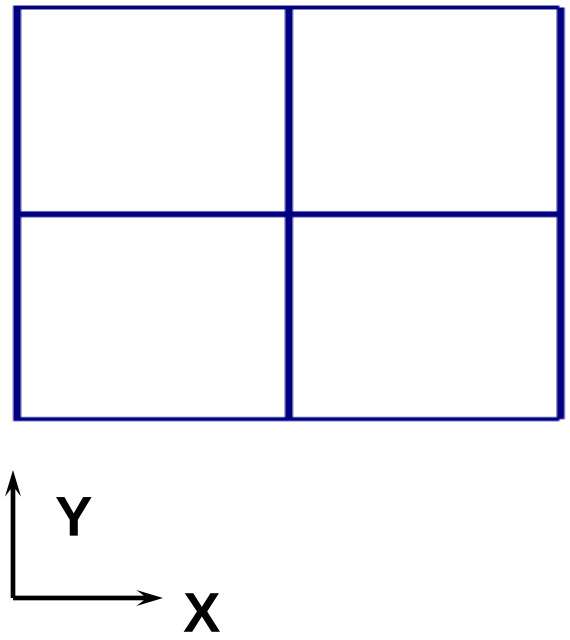
YCYLINDER LABEL R Yt Yb

Dodecahedron

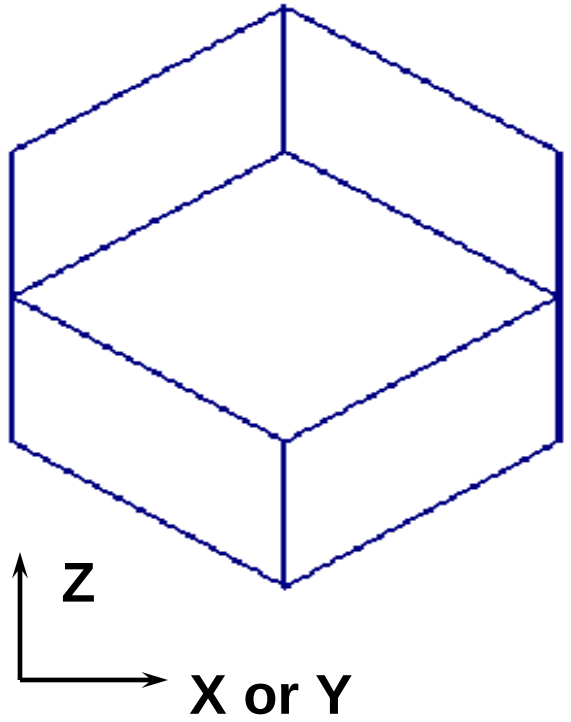
OVERALL VIEW



TOP VIEW

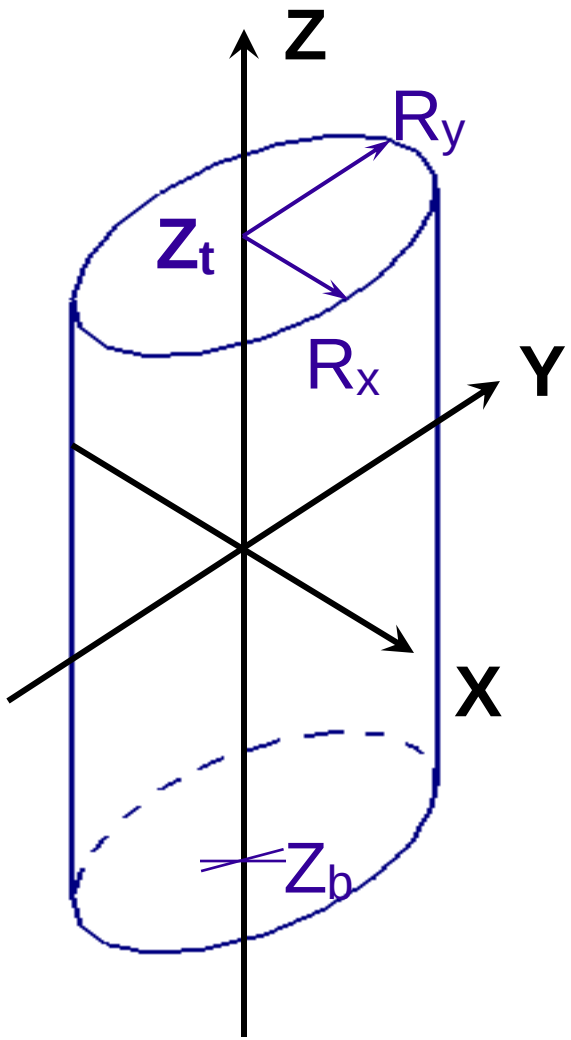


SIDE VIEW



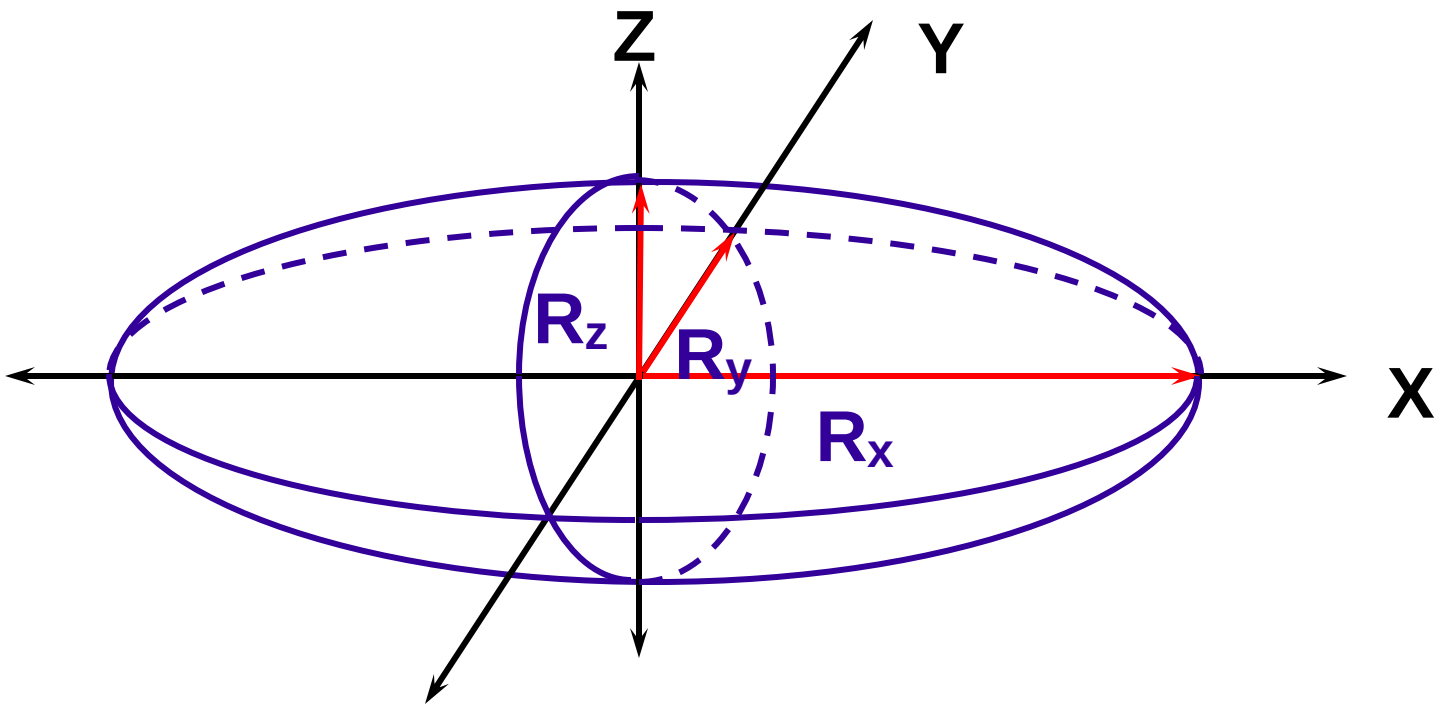
DODECAHEDRON LABEL R
(R is the Inscribed Radius)

Elliptical cylinder



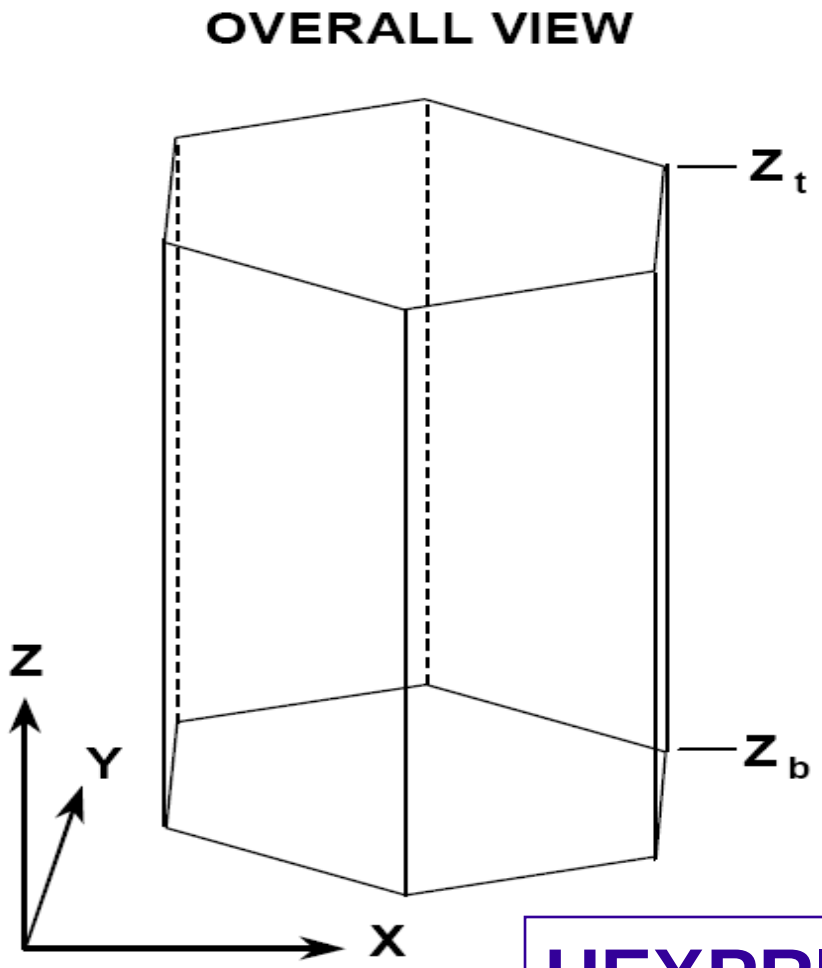
ECYLINDER LABEL R_x R_y Z_t Z_b

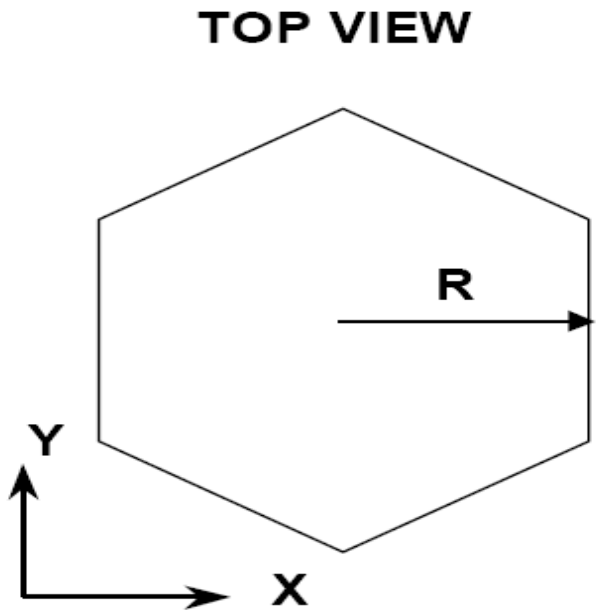
Ellipsoid



ELLIPSOID LABEL R_x R_y R_z

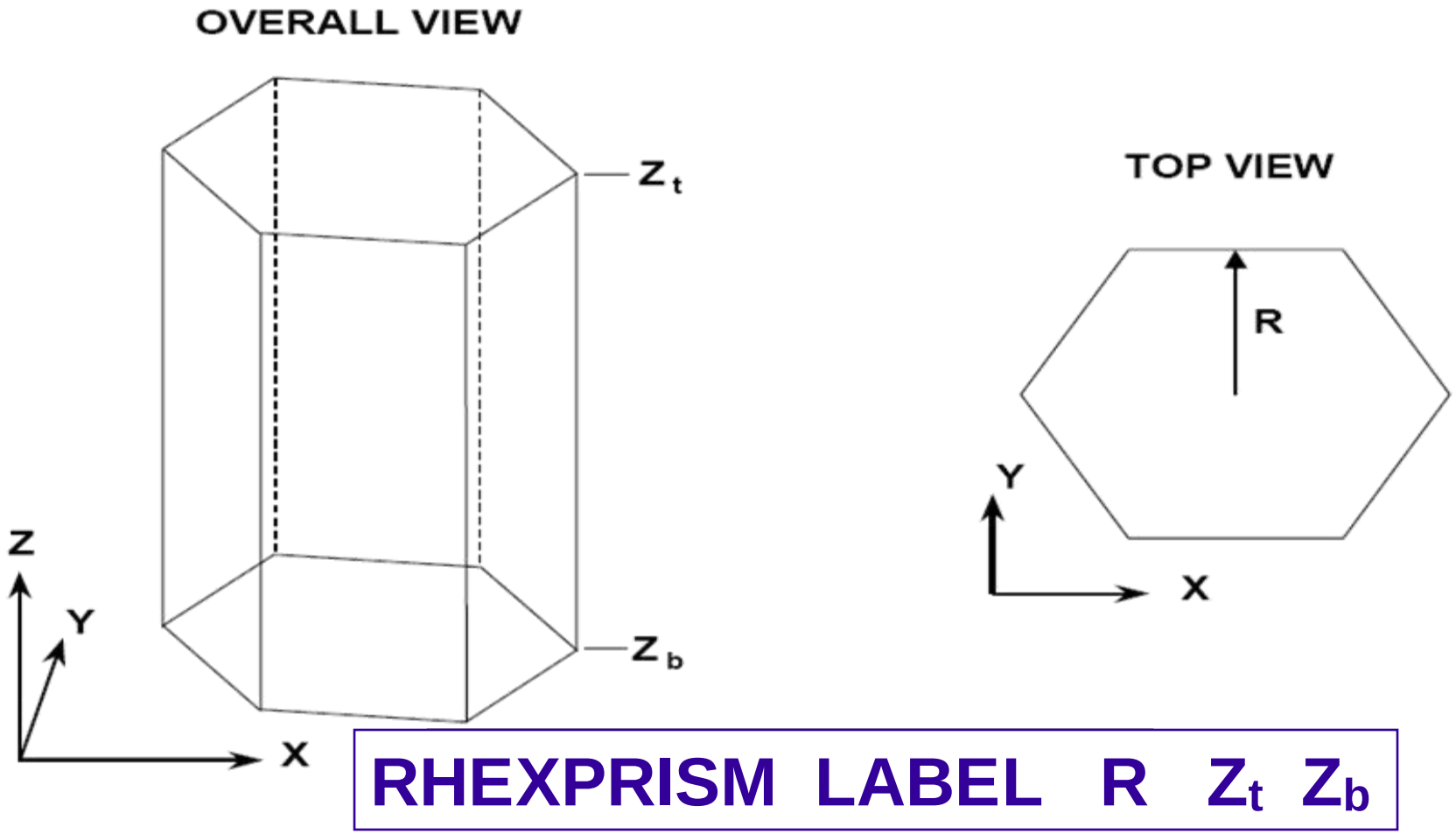
Hexagonal prism





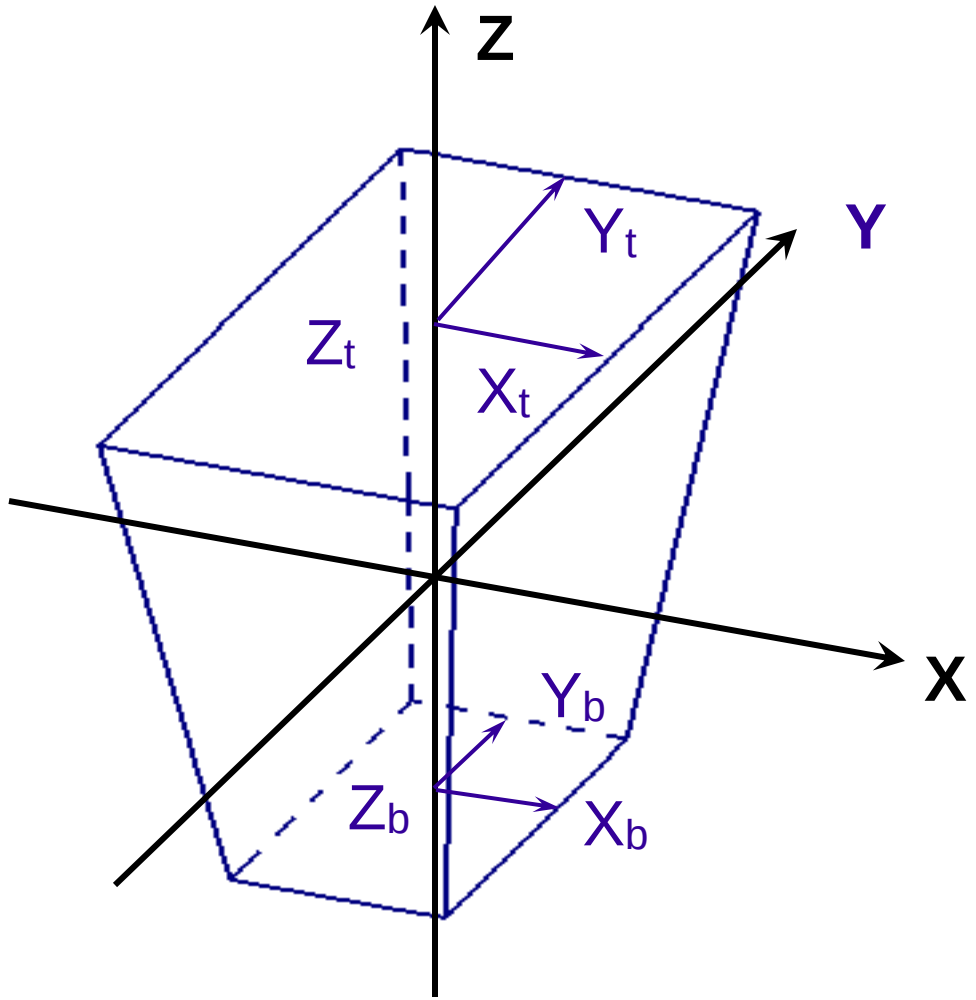
HEXPRISM LABEL R Z_t Z_b

Rotated hexprism



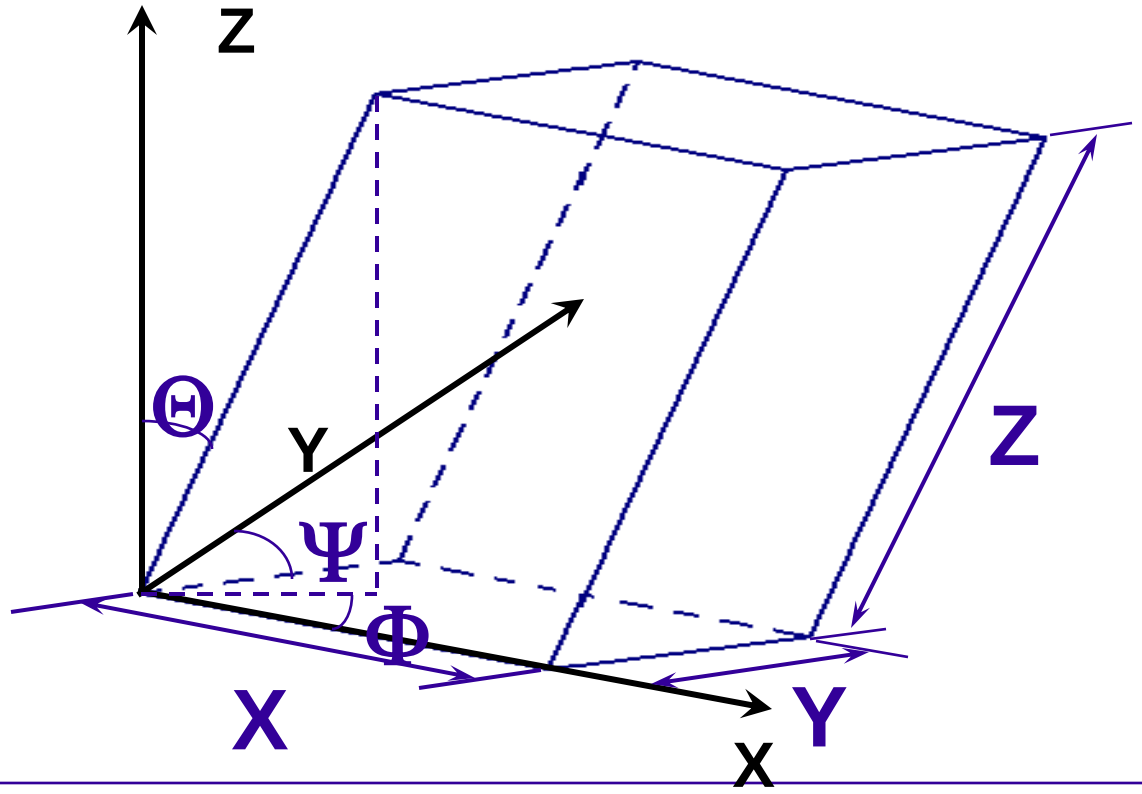
RHEXPRISM LABEL R Z_t Z_b

Hopper



HOPPER LABEL X_t Y_t Z_t X_b Y_b Z_b

Parallelepiped

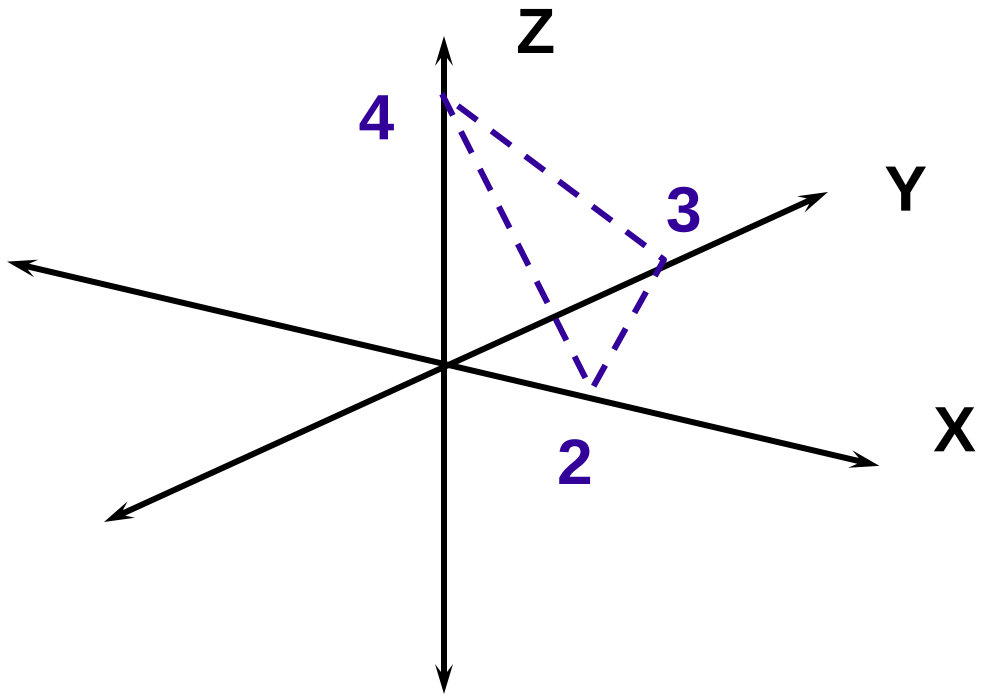


PPIPED LABEL X Y Z Ψ Θ Φ

or

PARALLELEPIPED LABEL X Y Z Ψ Θ Φ

Plane

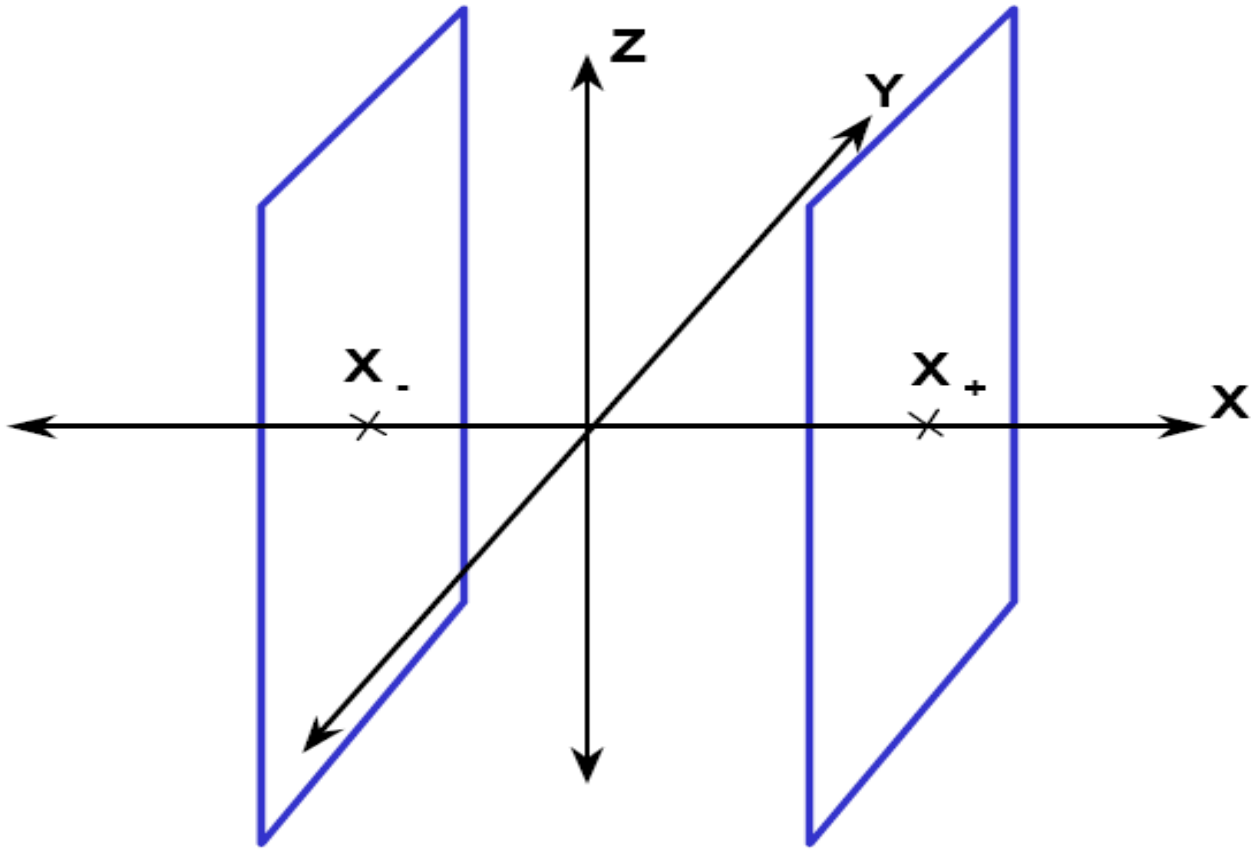


PLANE LABEL XPL=a YPL=b ZPL=c CON=d

$$aX + bY + cZ + d = 0$$

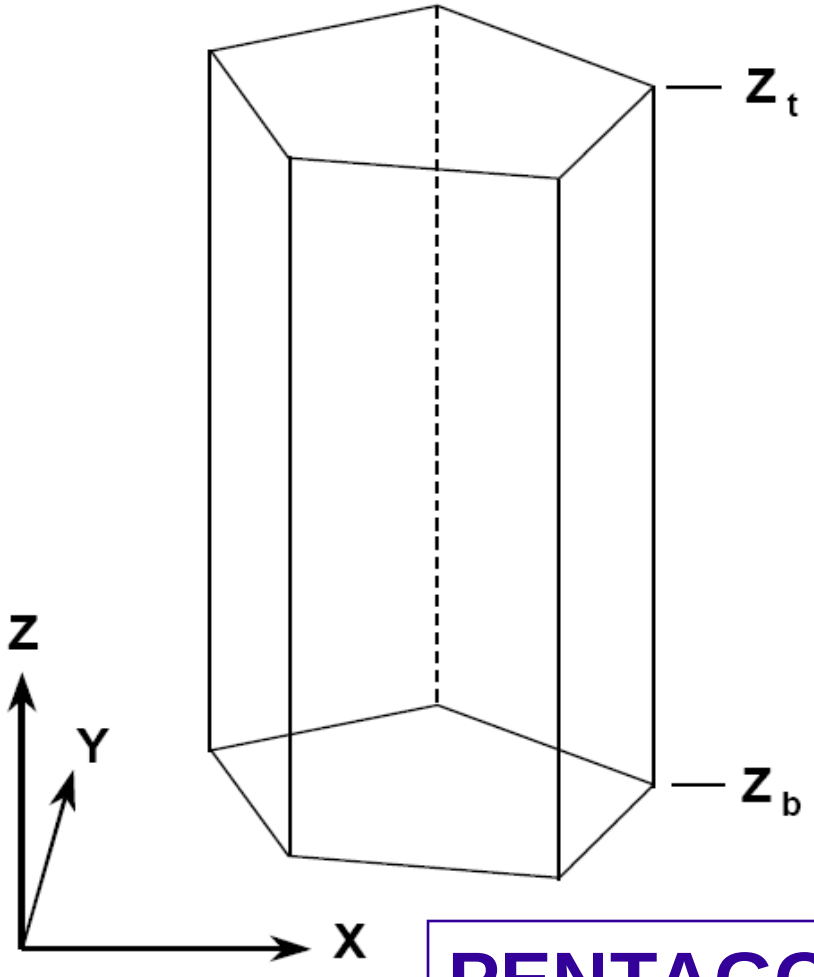
$$6X + 4Y + 3Z - 12 = 0$$

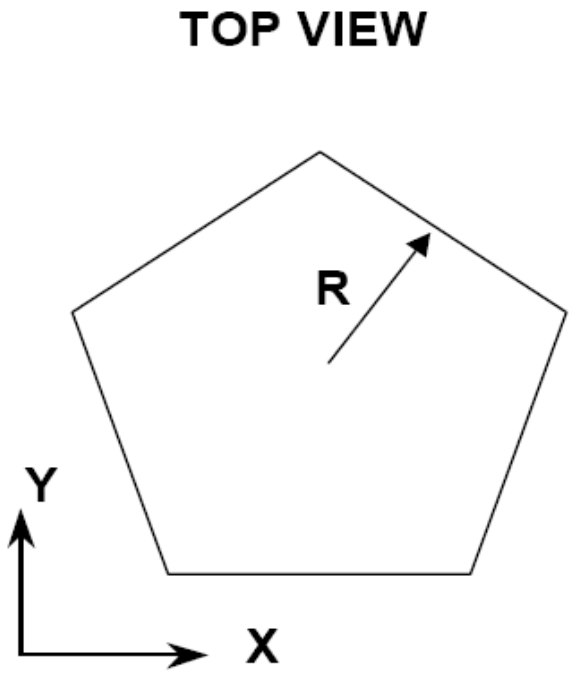
Paired planes




```
XPPLANE   LABEL  X+  X-
Also
YPPLANE   LABEL  Y+  Y-
ZPPLANE   LABEL  Z+  Z-
```

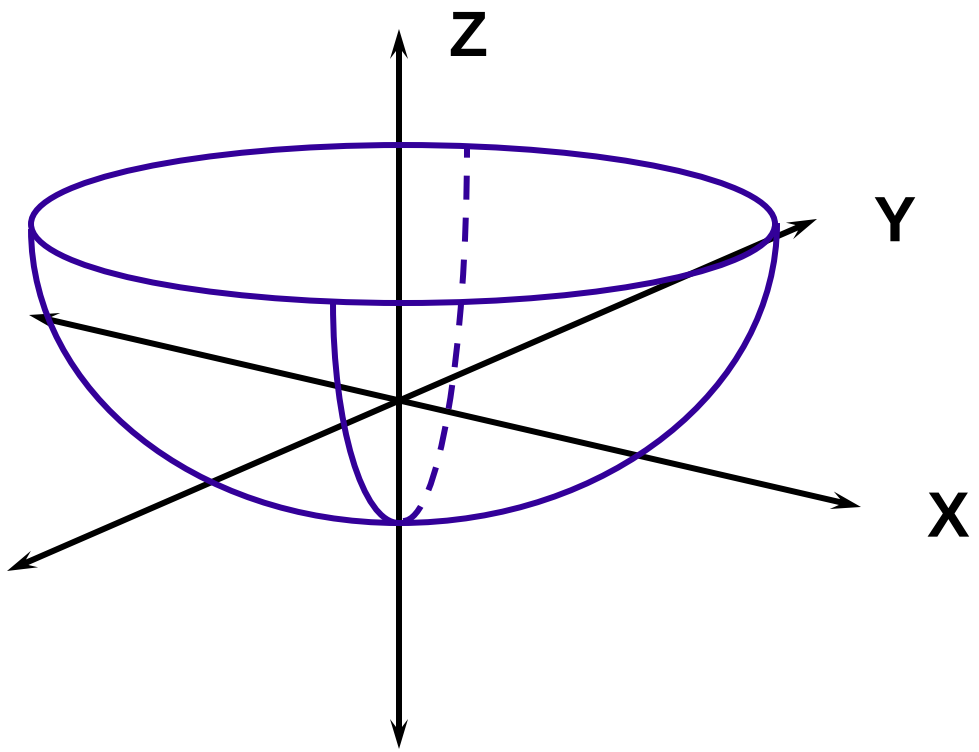
Pentagonal prism





PENTAGON LABEL R Z_t Z_b

Quadratic surface

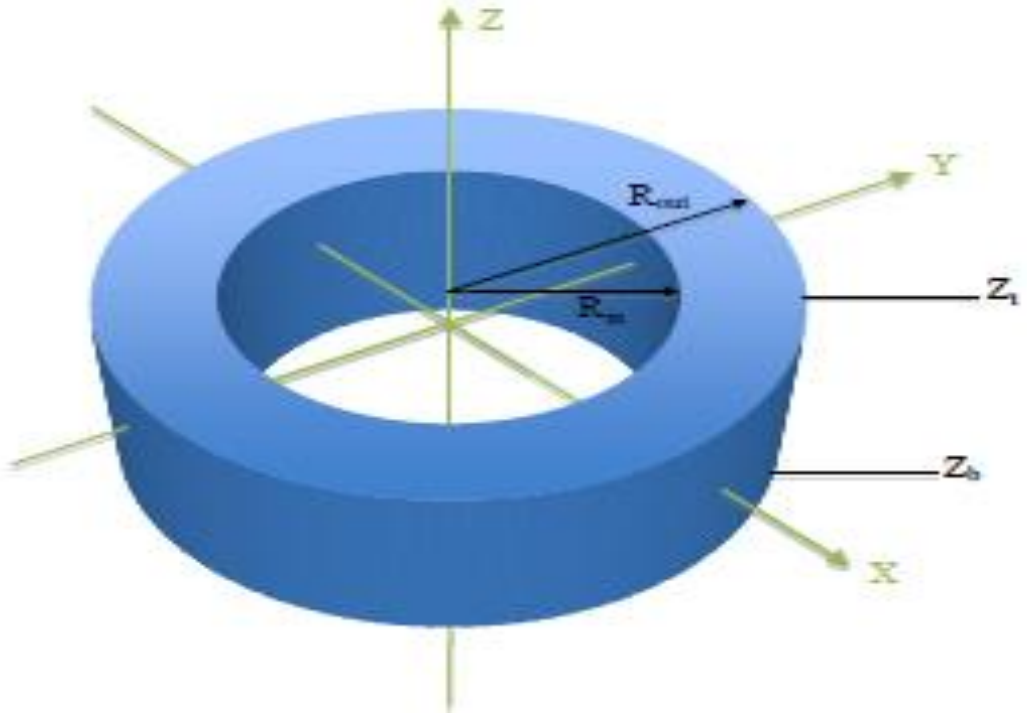


QUADRATIC LABEL AQU=a BQU=b CQU=c DQU=d EQU=e FQU=f GQU=g HQU=h IQU=I JQU=j

$$aX^2 + bY^2 + cZ^2 + dXY + eXZ + fYZ + gX + hY + iZ + J = 0$$

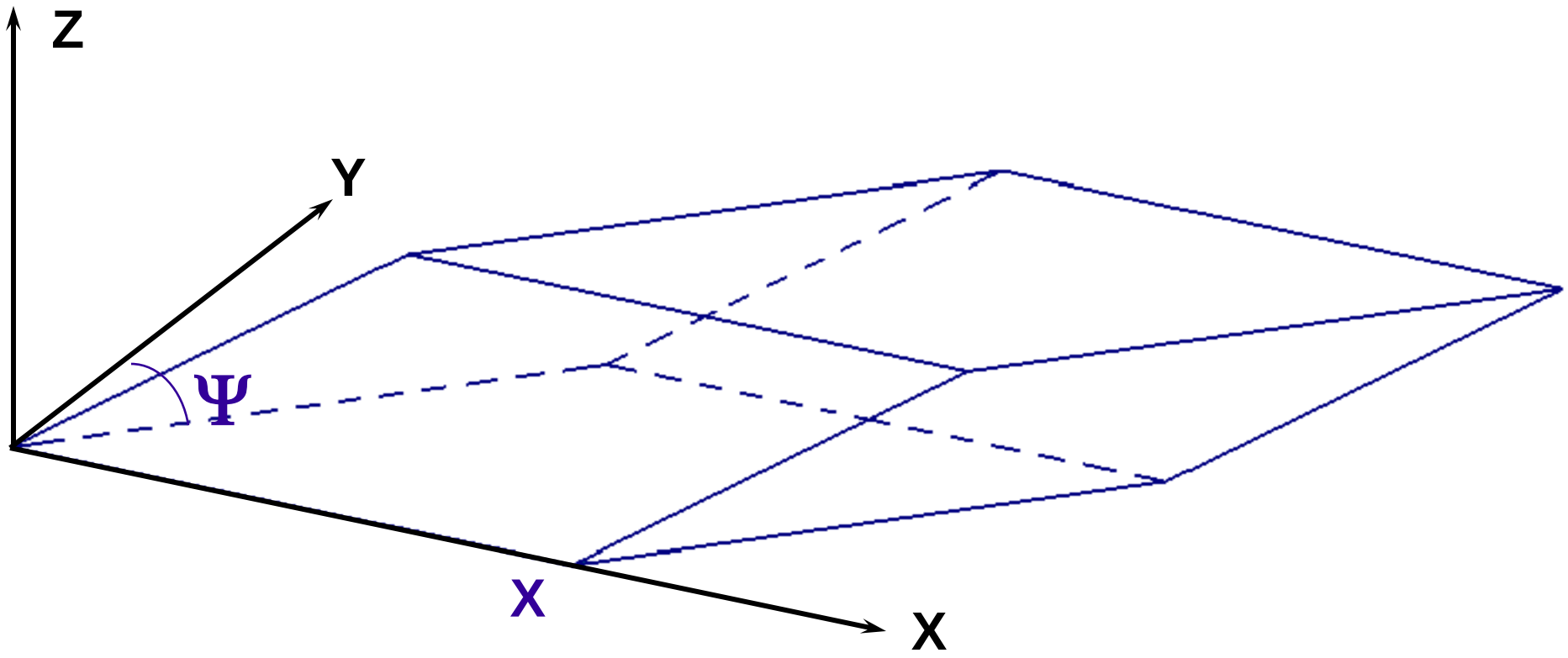
(Only non-zero values need to be entered)

Ring



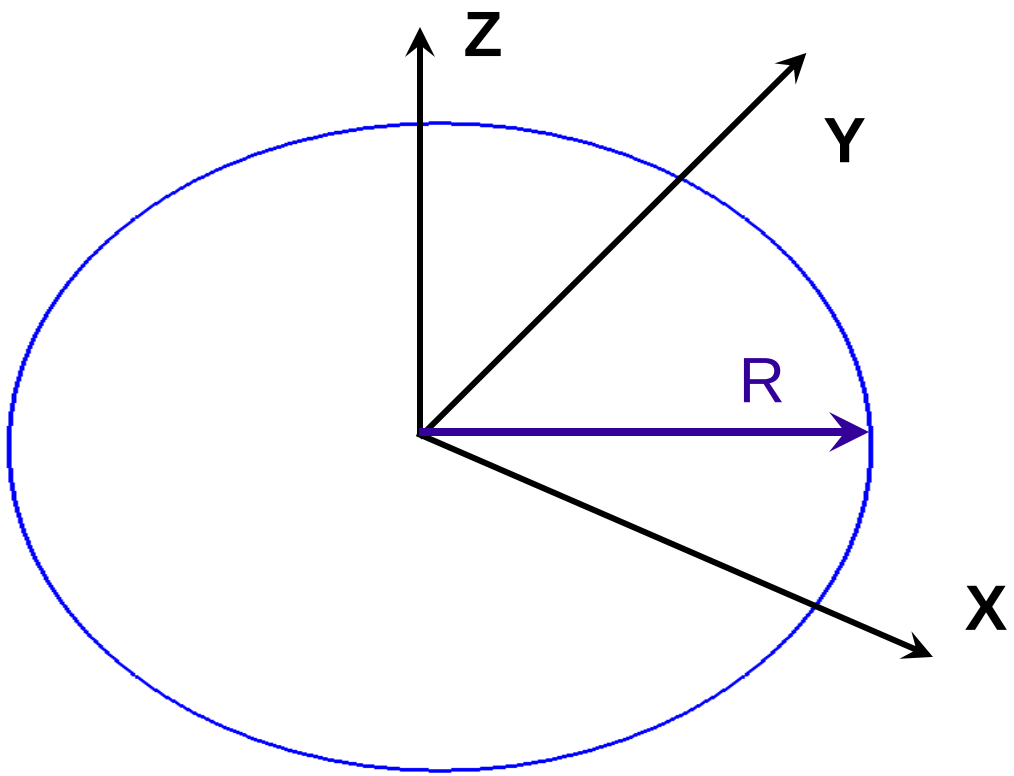
RING LABEL R_{in} R_{out} Z_t Z_b

Rhomboid



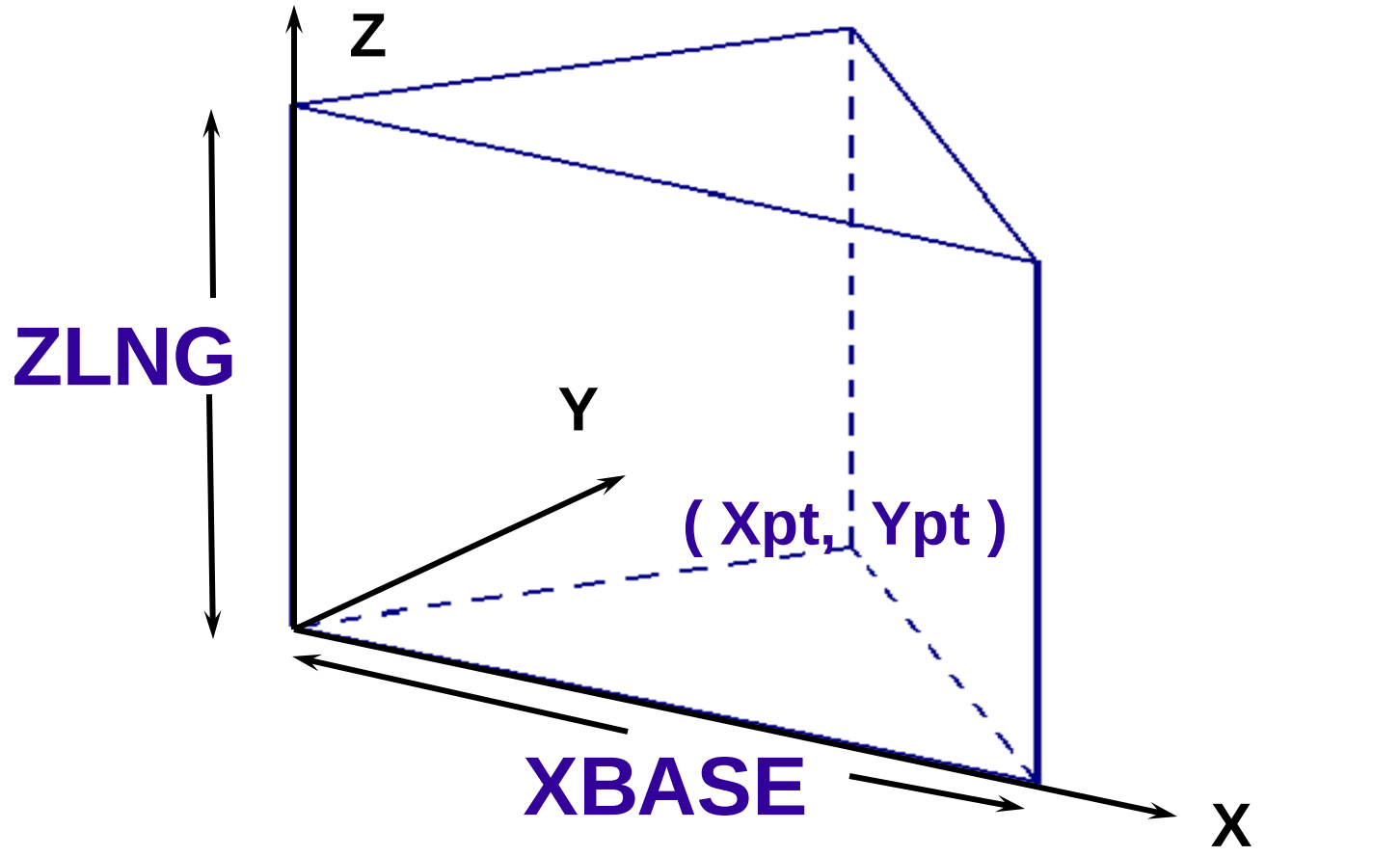
RHOMBOID LABEL X Ψ

Sphere



SPHERE LABEL R

Wedge



WEDGE LABEL XBASE Xpt Ypt ZLNG

OAK RIDGE National Laboratory

Questions on geometry records?



Contents record

- Specifies the contents of a volume defined relative to the Geometry records in that unit

- Three types of Contents records:

1. Media – Places a mixture in the specified volume
2. Hole – Places a previously defined unit at the specified position
3. Array – Places an array in the specified volume

Contents record types

Keyword	Contents Record Input
MEDIA	Mixture Number Bias ID Number Region Definition Vector
ARRAY	Array Number Region Definition Vector Keyword PLACE followed by the X, Y, Z position of the unit used to place the array and the X, Y, Z coordinates of the unit with respect to the origin of the outer region
HOLE	Unit number Keywords ORIGIN and ROTATE, if desired

Will be discussed in a dedicated lecture

Will be discussed in a dedicated lecture

Contents record types (cont.)

- Comprises the following components
  - Contents keyword (Array, Hole, Media)
  - ID number (Array #, Unit #, Mixture #)
  - Bias ID number (Media only)
  - Region definition vector (Array, Media only)
  - PLACE (Array only)
  - ORIGIN and/or ROTATE (Hole only)
- Examples:

```
media 4 1  20 -15
hole 1 origin x=5 y=5 z=10 rotate a1=45
array 6  10  PLACE 7 7 1  0 0 0
```

Media Contents record

- Places mixture into a region defined in the current unit
- Comprises the following components:
  - MEDIA keyword
  - Mixture ID number
  - Bias ID number (essentially always 1)
    - Biased tracking not recommended for use in KENO
    - Historically only available for some reflector materials
    - Bias weights may not be accurate anymore, leading to wrong answers
  - Region definition vector
  - VOL keyword
    - Necessary for region-averaged tallies, not eigenvalue

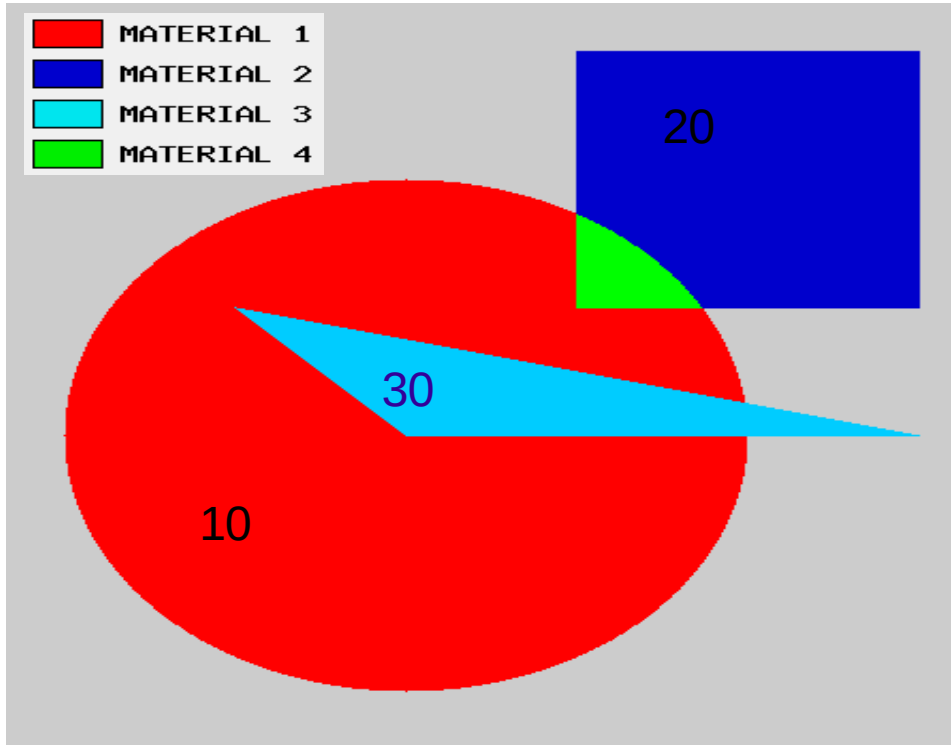
Region definition vector

- Composed of Geometry Record Labels used to specify the volume and location of a region in a unit
- The sign associated with each geometry label in the region definition vector specifies the relative location of the region in the unit using Boolean logic
  - Inside Geometry: Positive Label (AND)
  - Outside Geometry: Negative Label (NOT)
  - *NO* union operator

Region definition vector examples

```
media 1 1  10 -20 -30
media 2 1  20 -10
media 4 1  20 10
media 3 1  30
```

Region Definition Vectors

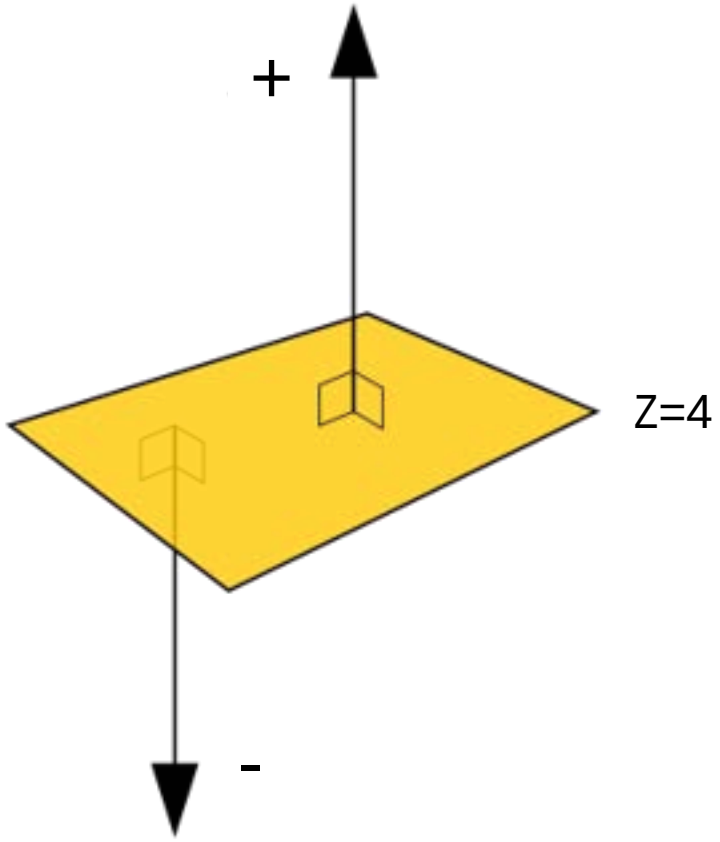


Region definition vector for planes

- How can a particle be inside a plane?
- The positive side (inside) of the plane is the side that the normal points to or where $aX + bY + cZ + d > 0$
  - Positive label: $aX + bY + cZ + d > 0$
  - Negative label: $aX + bY + cZ + d < 0$

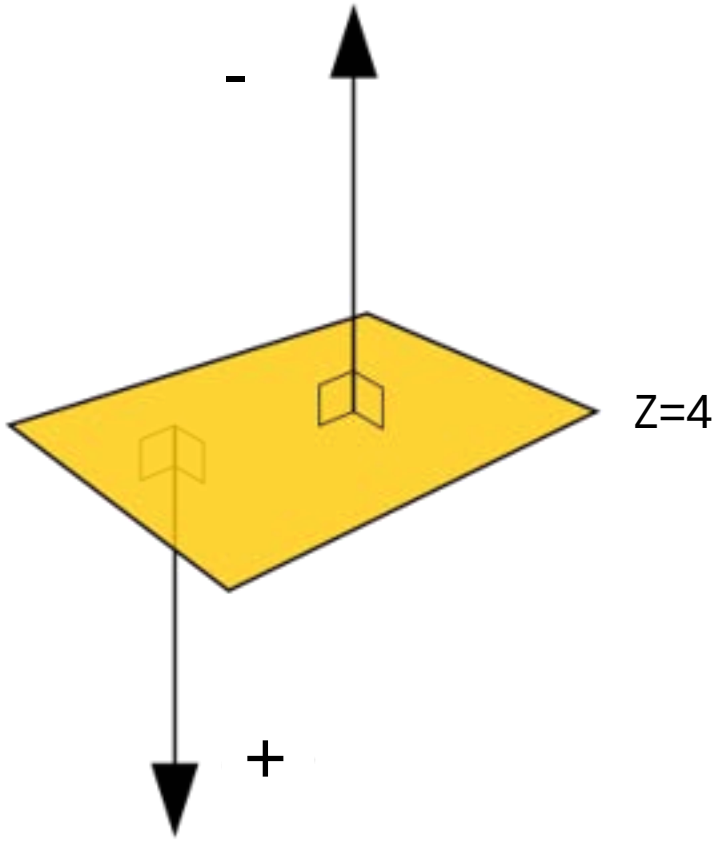
Example for plane at Z=4

- Option 1: Set ZPL = 1, CON = -4
  - 0X + 0Y + 1Z + (-4) > 0, or Z = 4
  - This makes volume above plane inside (+)
  - A point on Z = 5 is + relative to the plane
  - This is intuitive, but the constant has the opposite sign of the plane coordinate (CON=-4)
- No "best" choice; pick the lesser evil for you



Example for plane at Z=4

- Option 2: Set ZPL = -1, CON = 4
  - 0X + 0Y + (-1)Z + 4 > 0, or Z = 4
  - This makes the volume above the plane outside (-)
  - A point on Z = 5 is - relative to the plane
  - This is counterintuitive, but the constant has the same sign as the plane coordinate (CON=4)
- No "best" choice; pick the lesser evil for you



OAK RIDGE National Laboratory

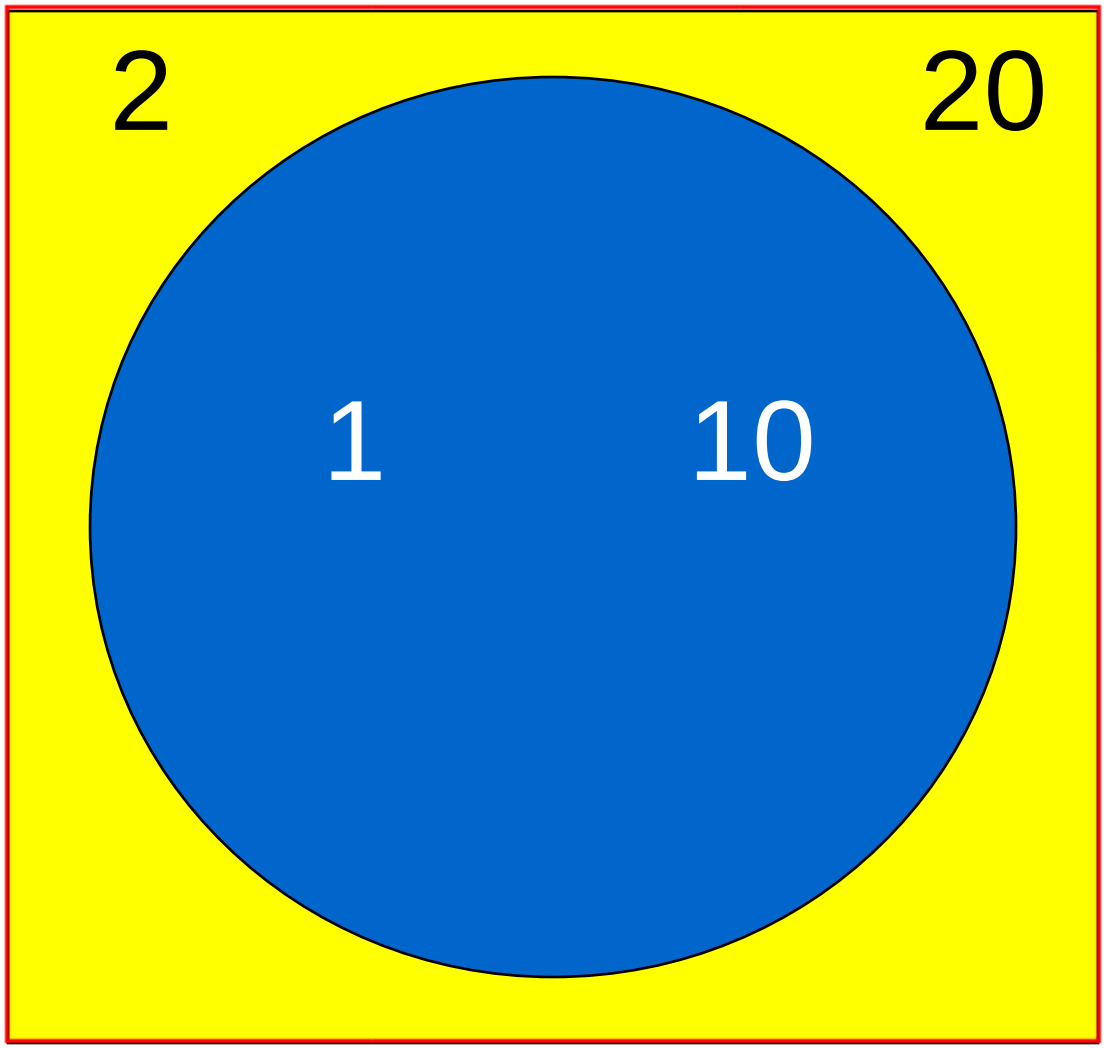
Questions on contents records?
Aside from HOLEs and ARRAYs, which are later…



Boundary record

- Specifies the overall volume of a unit
  - One and only one Boundary record is required for each unit
  - Only the volume defined in the region definition vector of the unit's boundary record is contained within the unit
  - Other specified volumes outside the boundary record's region definition vector are not included in the unit
- Composed of a Region definition vector

Single unit example




```
read geometry
global unit  1
   sphere  10  10.0
   cuboid  20  6p12.0
   media  1  1  10
   media  2  1  20  -10
boundary  20
end geometry
```

Free-form input in geometry definitions

Count Field	Option Field	Operand Field	Comment
i	r, *, $	j	Stores j in the next i positions and, if in an array, increments the array position by position i.
i	p	j	Stores j in the next i positions, alternating the sign each time.
i	z		Stores zero in the next i positions.

Examples:

3R2.5 is equivalent to: 2.5 2.5 2.5

4P3 is equivalent to: 3 -3 3 -3

5Z is equivalent to: 5R0 *or* 0 0 0 0 0

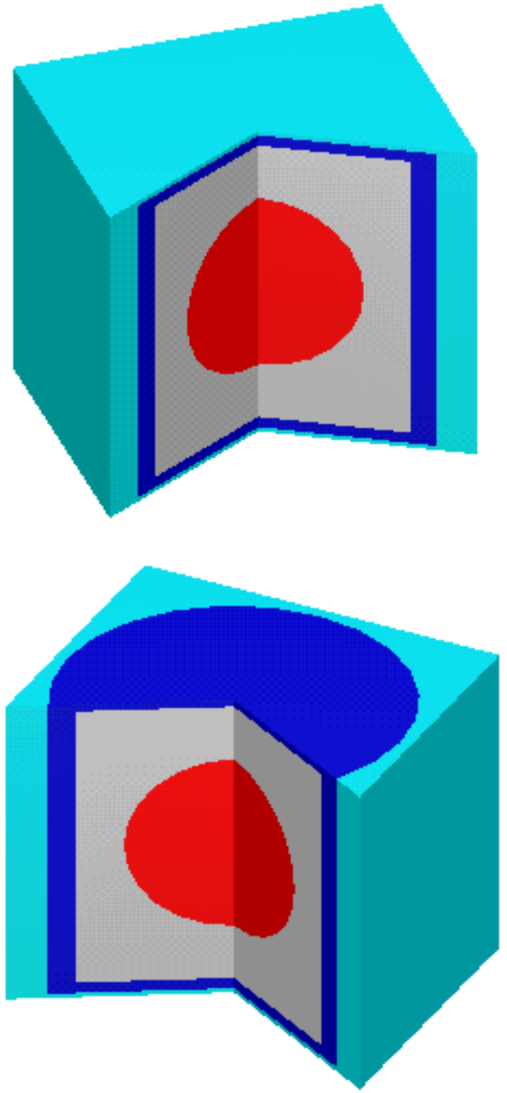
Building a unit in KENO-VI

```
unit 1
com='simple unit example'
sphere 10  4
cylinder 20  6  6.5  -6.5
cylinder 30  7  7  -7
cuboid  40  6p7.2
media 1 1 10
media 2 1 20 -10
media 3 1 30 -20
media 4 1 40 -30
boundary 40
```

Geometry Record

Contents Record

Boundary Record

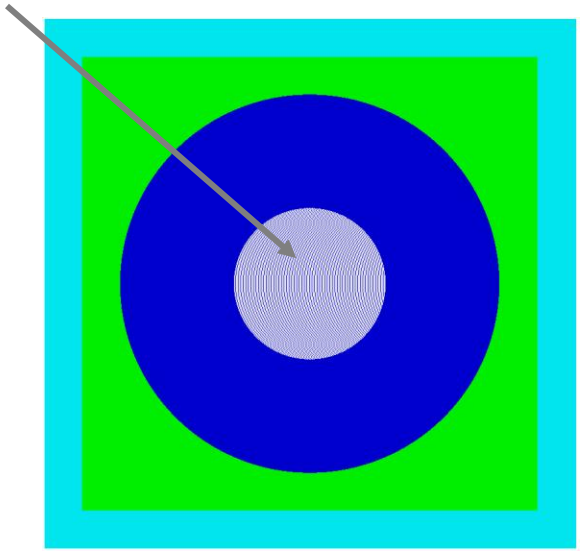


Error message for RDV problem: csas6

```
***** error ***** keno message number k6-106 follows:
 posit error --- unit 1 has multiply defined space at
 x = -1.65087E+00   y = -8.89616E-01   z = -3.63992E-01
 sector inside
    1      T
    2      F
    3      F
    4      T
```

- Sector number is contents record number – just counted
- T/F is true/false for inside or outside of sector

Doubly defined space in 6.2.4 Fulcrum



Input

Contents Record #	
	`global unit 1`
	`sphere 10 5`
	`cuboid 20 6p6`
	`cuboid 30 6p7`
	`sphere 40 2`
	`'`
1	`media 1 1 10`
2	`media 3 1 20 -10`
3	`media 2 1 30 -20`
4	`media 0 1 40`
	`'`
	`boundary 30`

Sphere 40 is smaller than (and completely inside) sphere 10. This is doubly defined space; thus, the particle is inside 1 and 4 at the same time.

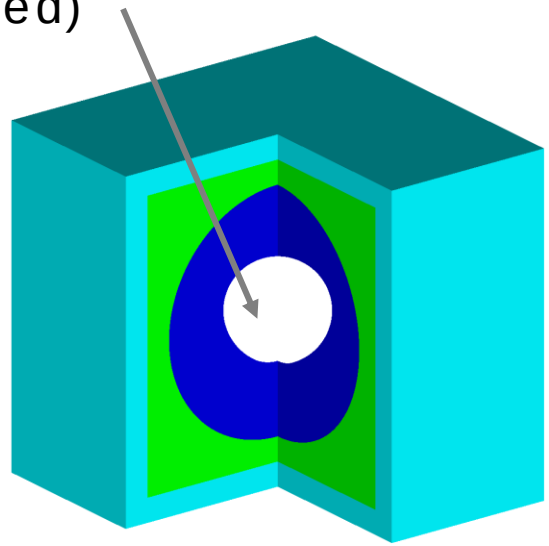
Error message for RDV problem: csas6-Shift

```
!!! Geometry error in particle 0:11: Multiple overlapping cells found at -1.88819
0.47528 0.456992 along 0.90074 -0.296867 0.317076 in 'unit 1': inside '1.4' (<inp>:65,
local cell ID 4) and '1.1' (<inp>:59, local cell ID 1)
!!! Geometry error in particle 0:19: Multiple overlapping cells found at -1.28204
1.16206 1.00299 along 0.882639 -0.422375 0.20627 in 'unit 1': inside '1.4' (<inp>:65,
local cell ID 4) and '1.1' (<inp>:59, local cell ID 1)
```

- Shift provides location of overlapping cell instance
- Provides unit regions that overlap (cell ID 1 and cell ID 4) as well as input location of overlapping media instance (line 65 and 59)

Doubly defined space in 6.3.2 Fulcrum:

- Less obvious than in 6.2.4
- Hard to distinguish from void (if expected)



Input

Contents Record #	
	global unit 1
	sphere 10 5
	cuboid 20 6p6
	cuboid 30 6p7
	sphere 40 2
	'
1	media 1 1 10
2	media 3 1 20 -10
3	media 2 1 30 -20
4	media 0 1 40
	'
	boundary 30

Sphere 40 is smaller than (and completely inside) sphere 10. This is doubly defined space; thus, the particle is inside 1 and 4 at the same time.

OAK RIDGE National Laboratory

Volume block



Volumes in Monte Carlo models

- Region volumes are not needed in CSAS/KENO-VI or CSAS/Shift for criticality calculation
  - However, region/mixture averaged tallies, i.e., reaction, flux, and others, require region volumes
- Volumes are always required in TRITON/KENO-VI and TRITON/Shift depletion calculations
  - If your fuel volumes are incorrect, then the power generated from the fuel is incorrect, and your whole depletion calculation is wrong
- Options:
  - Specify volume via "vol=" after each MEDIA record
    - Example: **`media 10 1 3 -4 vol=2.0e-2`**
  - Add VOLUME block to enable volume calculation (next slide)
- CSAS skips the volume calculation if not enabled
- TRITON enables a volume calculation if no volume block is specified and if volumes are missing
- Units for volume definitions are cm^3

Volume calculation via VOLUME block

- Use the VOLUME block to calculate region and mixture volumes
- Volume calculation is performed before each neutron transport calculation, and also if **`parm=(check)`**
- KENO-VI:
  - Monte Carlo integration (point sampling):

    `type=random nbatches=500 npoints=1000`
  - Trapezoidal integration (ray-tracing):

    `type=trace nrays=100000`
- Shift:
  - Ray tracer: **`type=trace nrays=100000`**
- Check volumes and corresponding errors in output

See details on VOLUME block in SCALE manual section 8.1.3.13

Volume calculation with Fulcrum

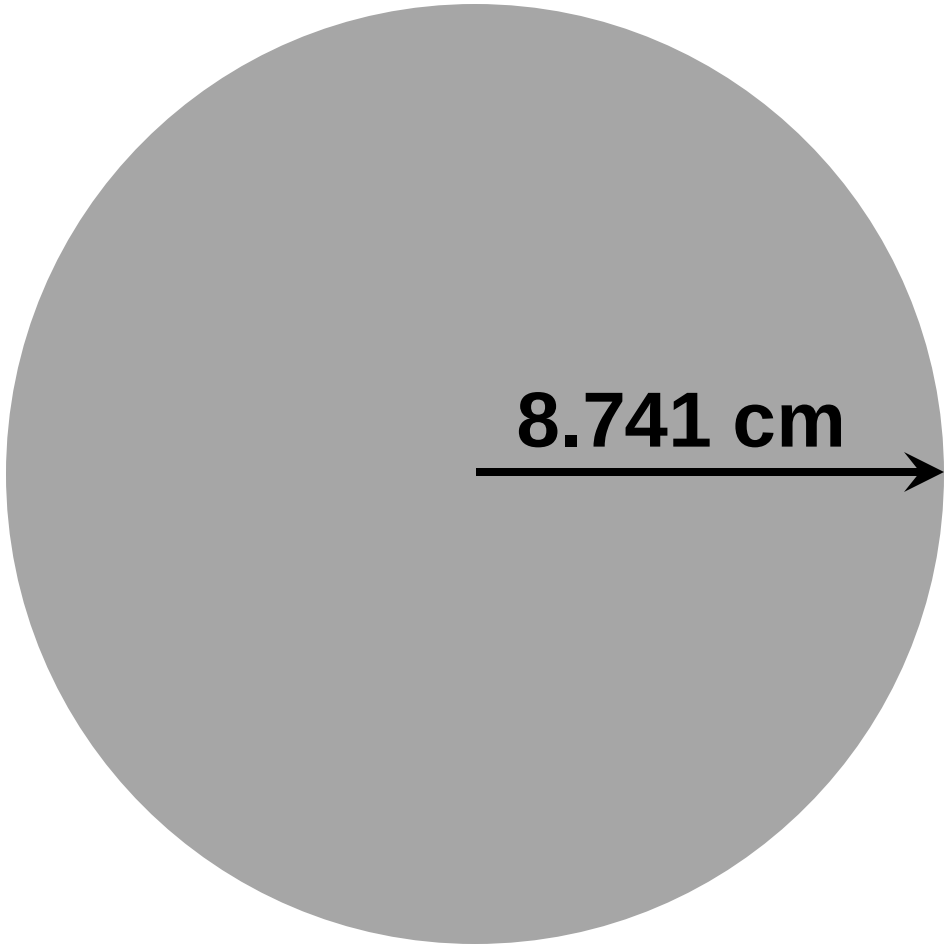
- Use Fulcrum to calculate volumes for simple geometries
- Hit arrow next to "Run" and select "Calculate volumes"
- Message window will contain input with volumes added for each MEDIA record
- Note:
  - Behavior of Fulcrum is slightly different between SCALE versions: addition of *all* vol= records, replacement of existing vol= records, etc.
  - This only works for simple geometries
  - Uncertainty of volumes is not reported
  - Recommended only to double-check your own calculated volumes

OAK RIDGE National Laboratory

Class exercise: Lady Godiva



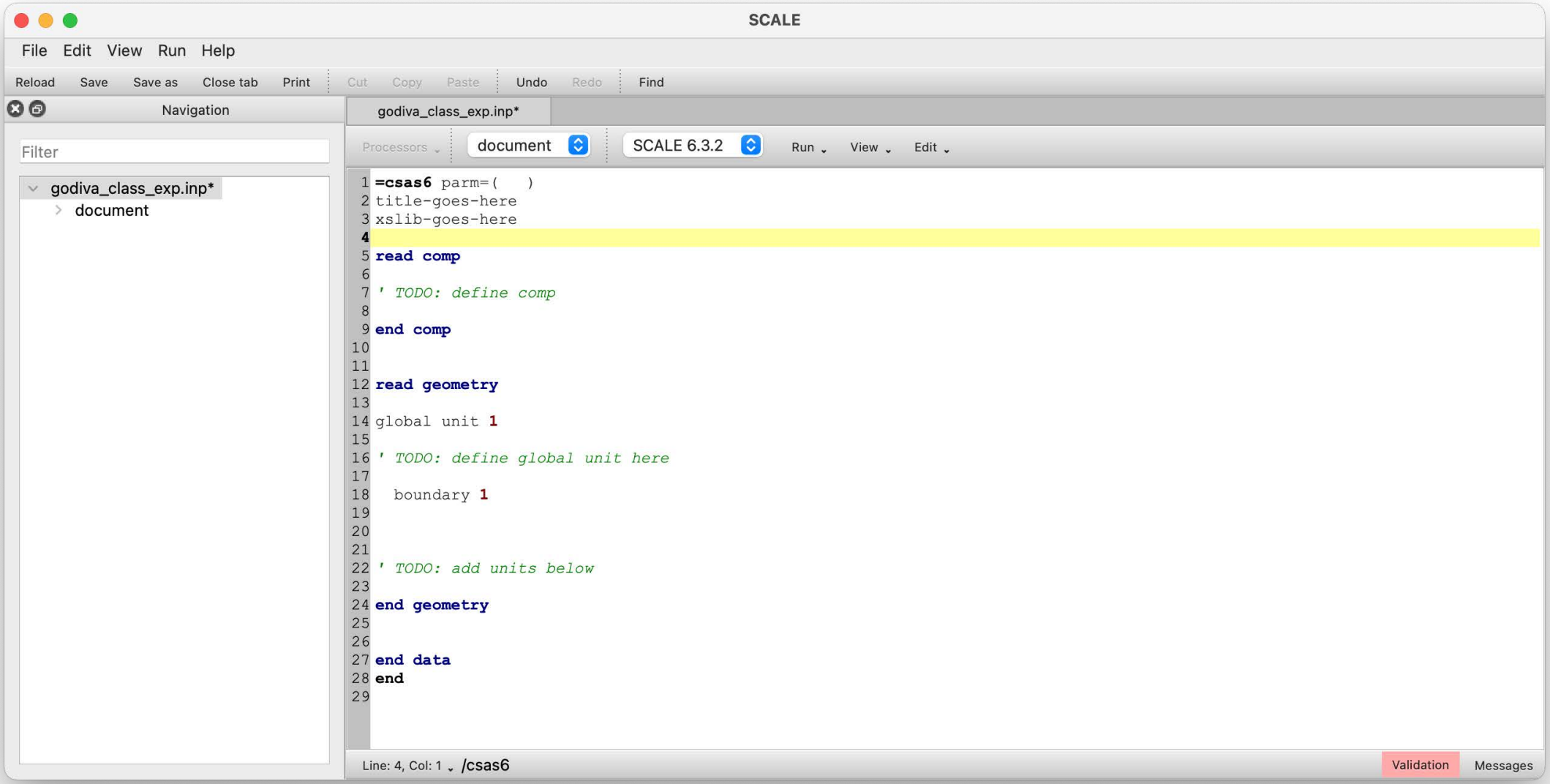
Lady Godiva



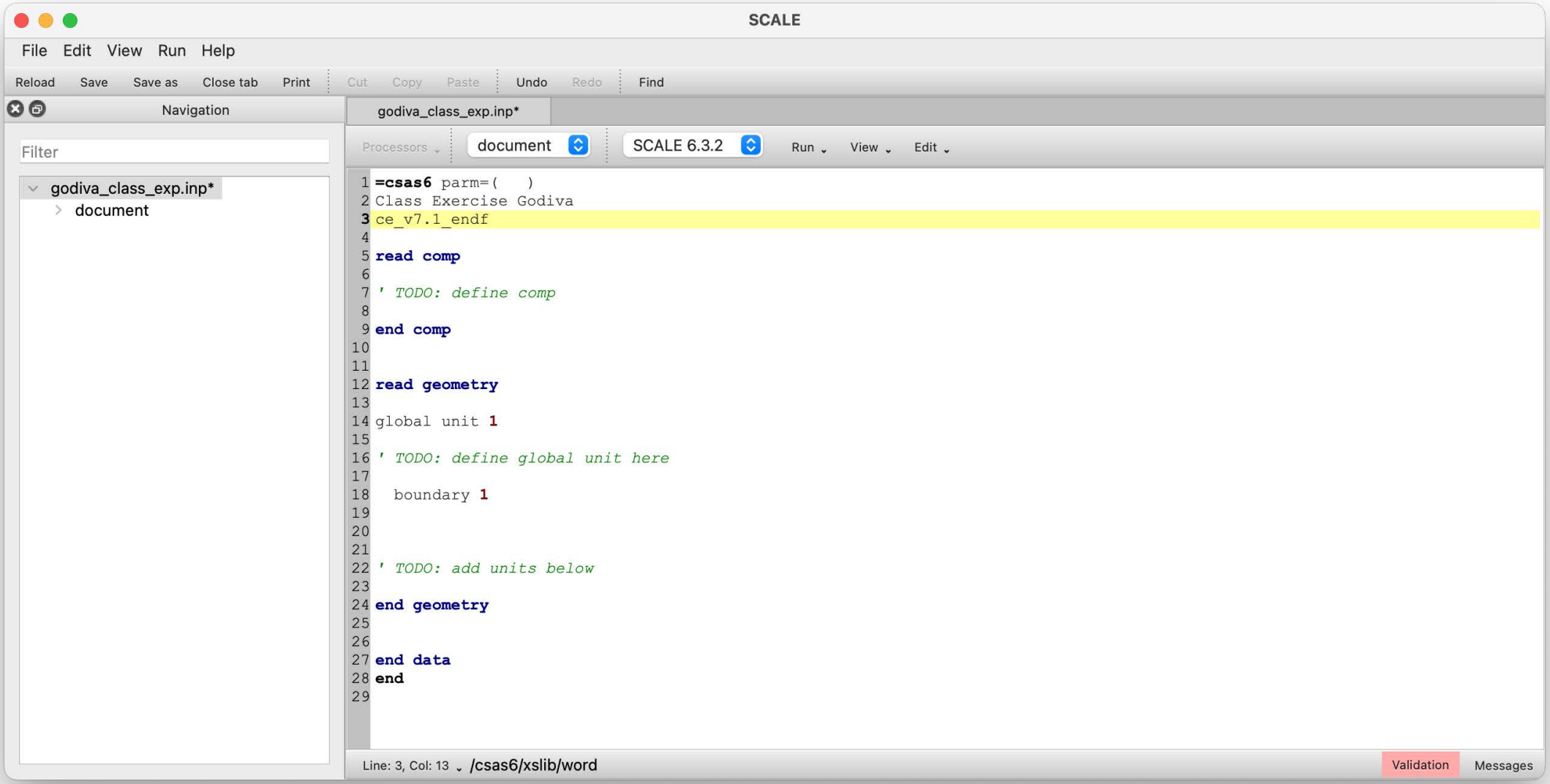
Bare Uranium Metal Sphere
Density 18.74 g/cc

Enrichment:	**U-235**	**93.7112 wt%**
	U-238	**5.2686 wt%**
	U-234	**1.0202 wt%**

Input skeleton



General problem data



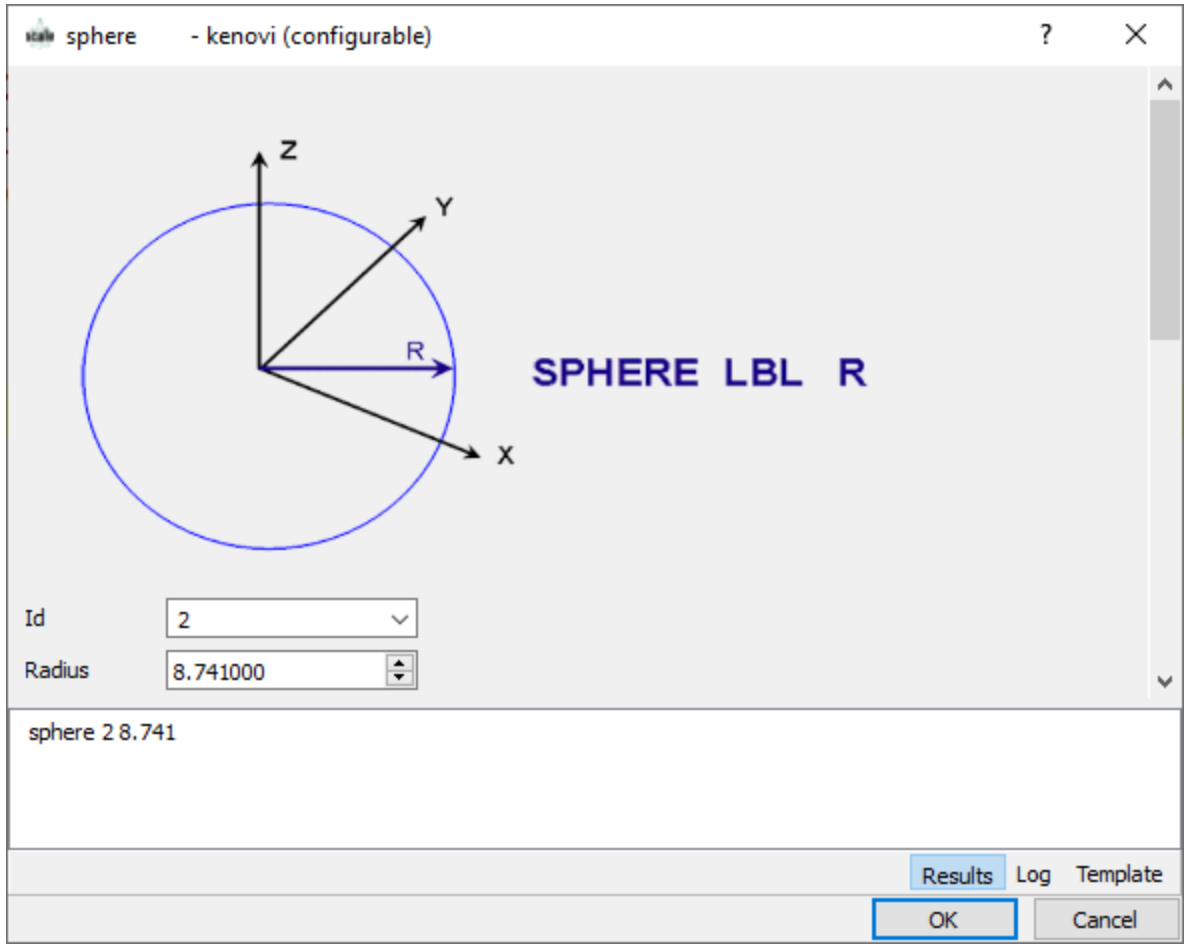
Create U metal composition

```
read comp

 u 10 den=18.74 1 293 92234 1.0202
                      92235 93.7112
                      92238 5.2686 end
end comp
```

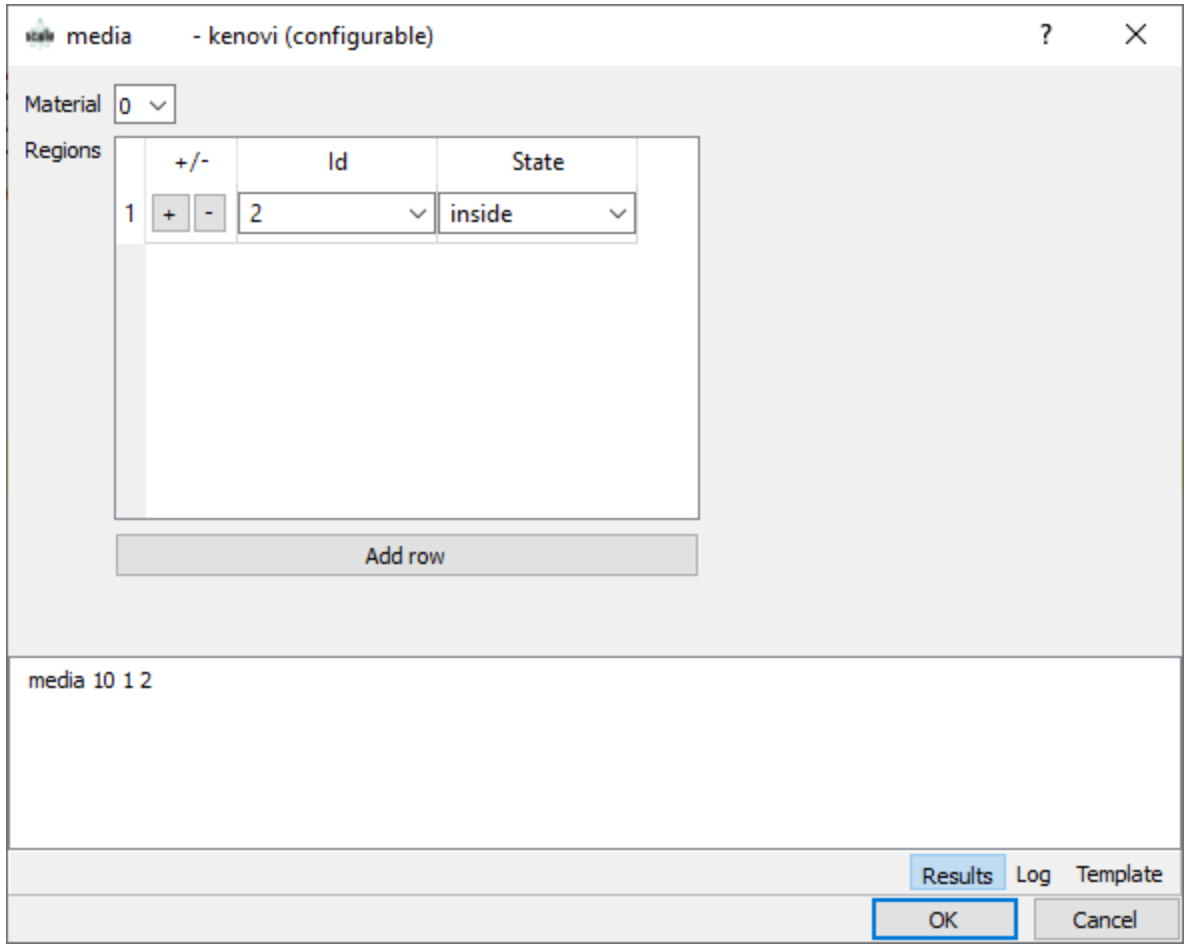
Create sphere geometry record

```
11 read geometry
12 global unit 1
13 com='the whole thing'
14   sphere 2 8.741
```



Create Media record

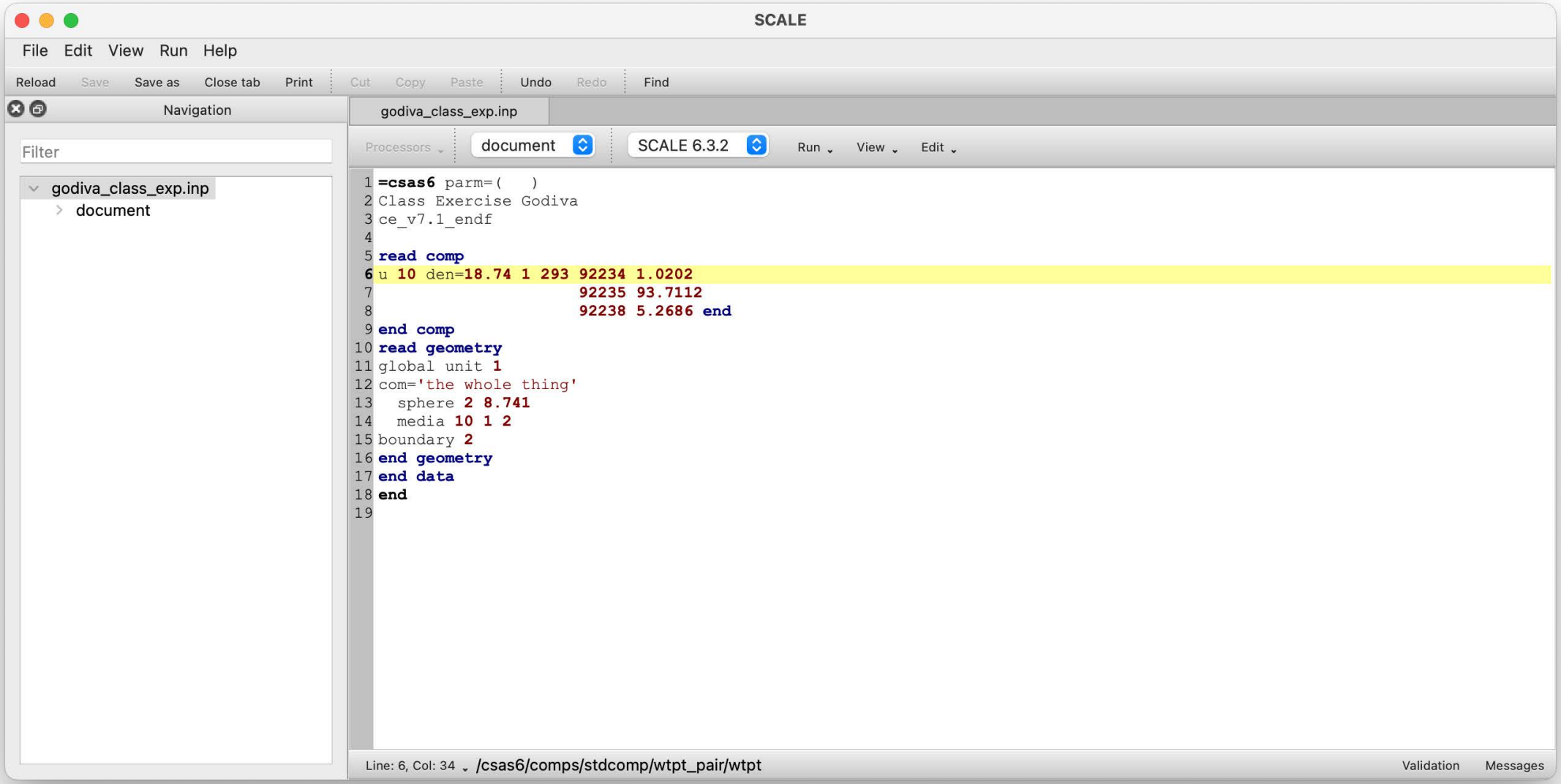
```
11 read geometry
12 global unit 1
13 com='the whole thing'
14   sphere 2 8.741
15   media 10 1 2
```



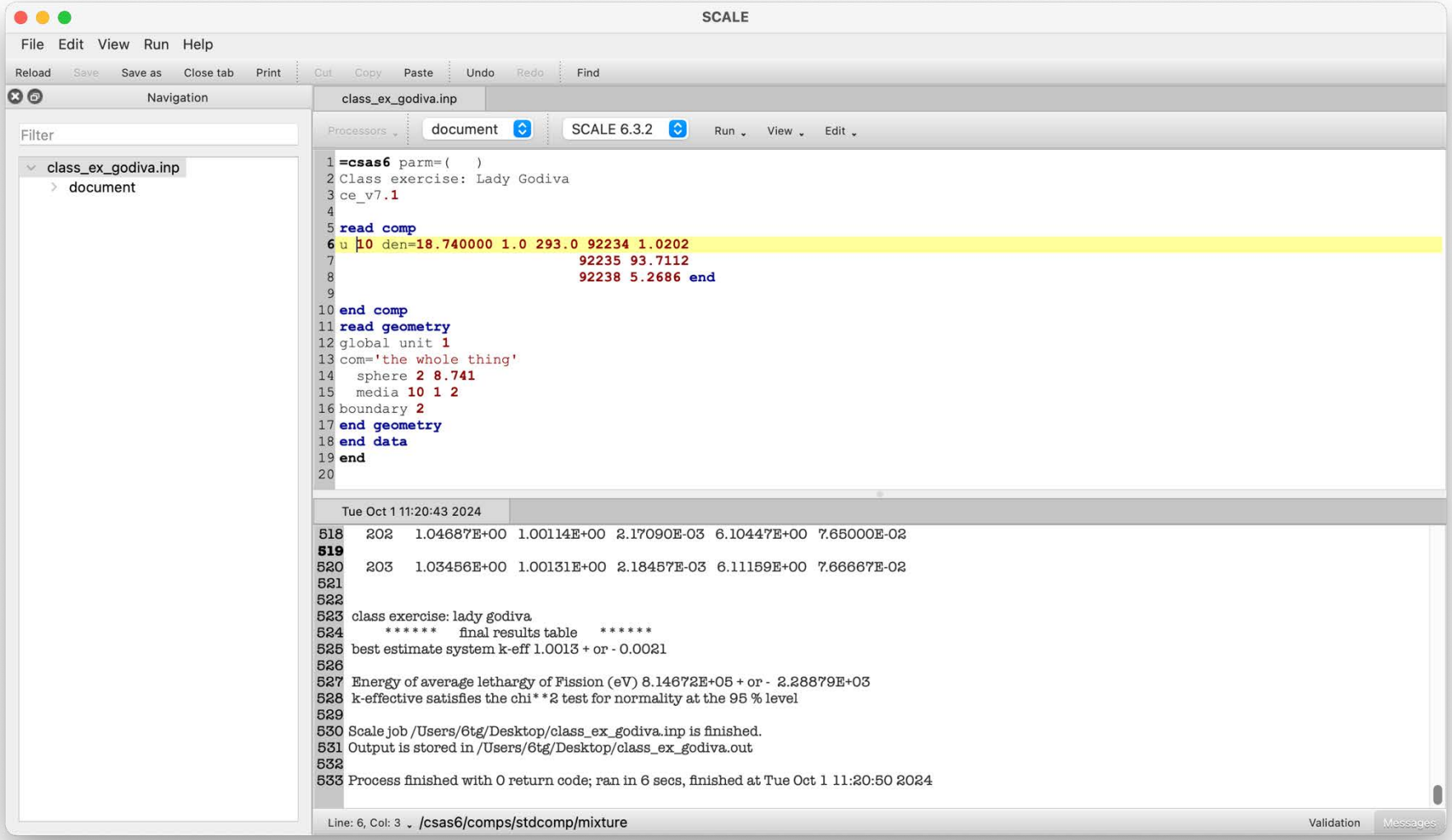
Create Boundary record

```
11 read geometry
12 global unit 1
13 com='the whole thing'
14   sphere 2 8.741
15   media 10 1 2
16 boundary 2
17 end geometry
```

Entire problem description



Save and run



Best estimate k_{eff}:
1.0013 ± 0.0021

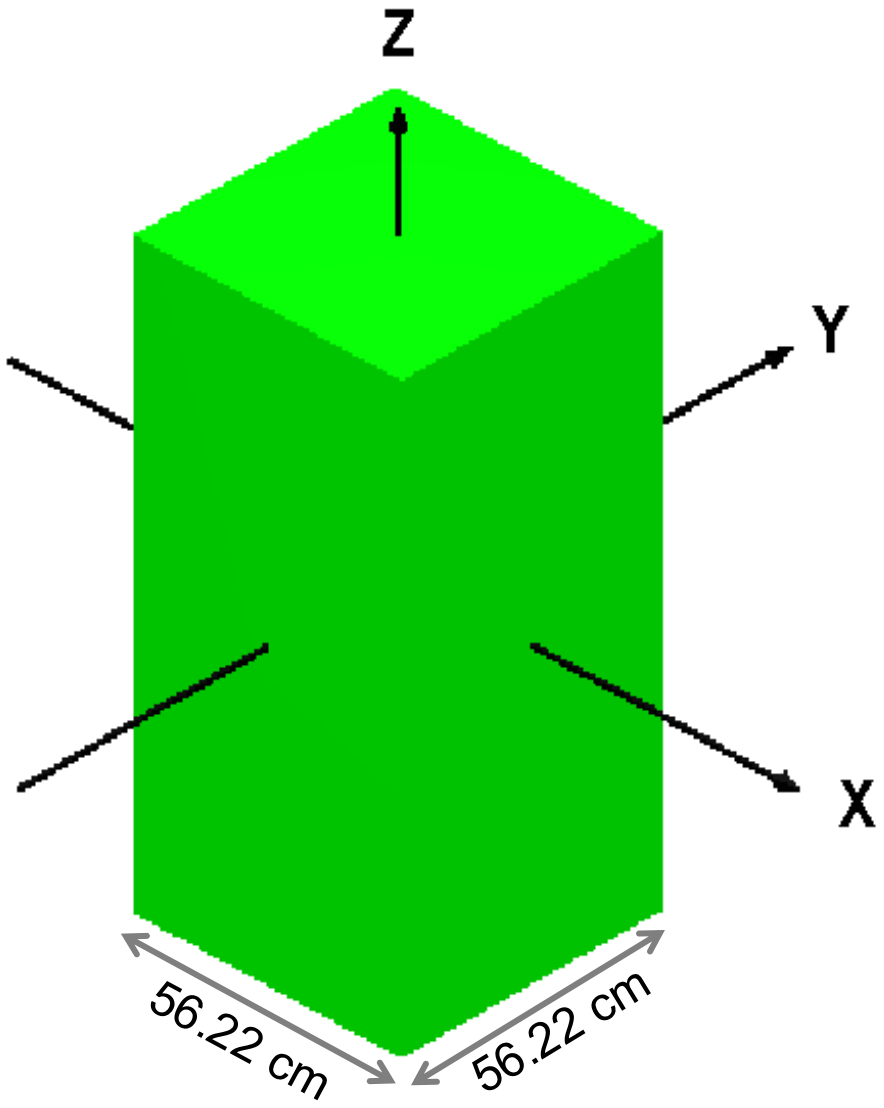
OAK RIDGE National Laboratory

Exercises for sub-module 1

- Simple KENO-VI geometry exercises
  - Set 1: Green block and Rocky Flats



Problem: Green block



Material

Temperature = 293 K

UF_4 – 51.13 vol%

U-235 2.0 wt%

U-238 98.0 wt%

Paraffin – 48.87 vol%

Hint: remember that if DEN is not the same for both components of a mixture, the density multiplier works as a *volume* fraction.

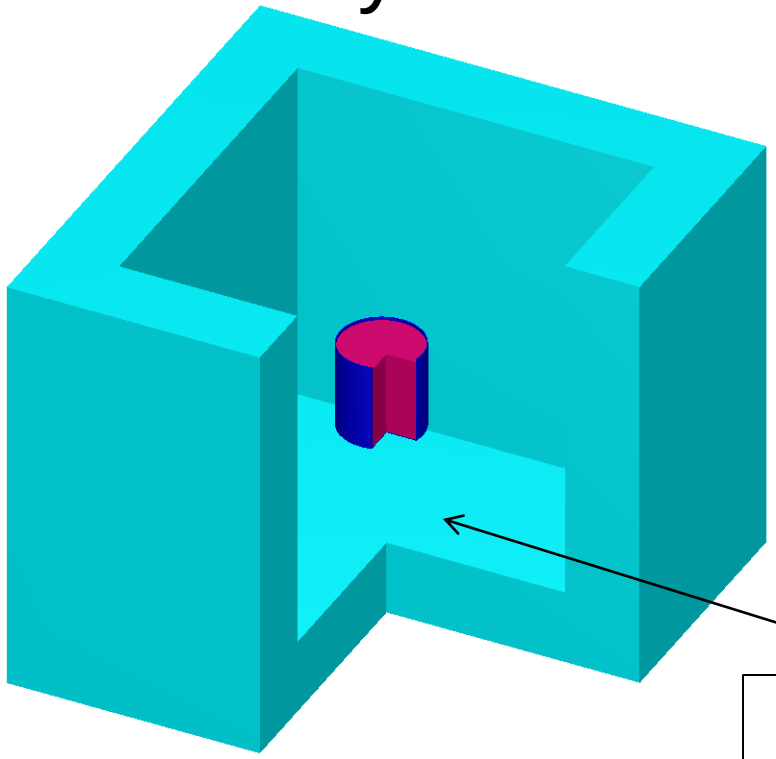
Dimensions

X : 56.22 cm

Y : 56.22 cm

Z : 122.47 cm

Problem: Rocky Flats



Room filled with void

Al Tank

14.005 cm inner radius, solution height 28.6 cm
0.32 wall and bottom thickness
30.0 cm total tank height
Distance ***from the*** **tank inside bottom center**
to the 25.7 cm thick Concrete walls, floor, and ceiling
+ X: 64.6 cm -X: -57.4 cm
+ Y: 57.4 cm -Y: -64.8 cm
+ Z: 80.0 cm -Z: -41.1 cm

Materials: Room Temperature

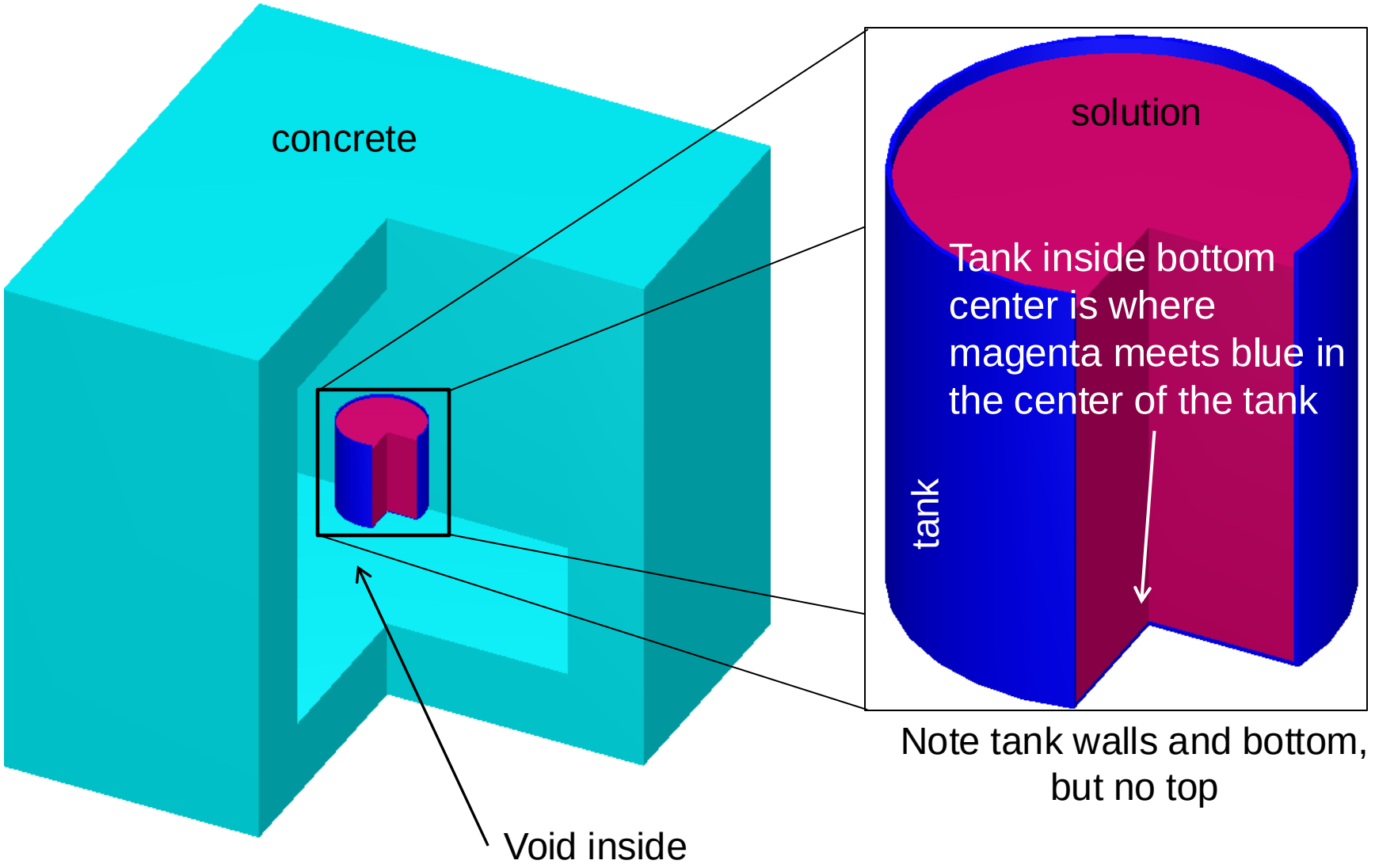
uo2(no3)2 Solution default density
g U/l: 334.77
Acid Molarity: 0.521
U-234: 1.022 wt%
U-235: 93.172 wt%
U-236: 0.434 wt%
U-238: 5.372 wt%

Al density = 2.702 g/cc

Concrete density = 2.32 g/cc

oxygen:	48.49 wt%
calcium:	23.31 wt%
silicon:	15.5 wt%
carbon:	5.52 wt%
aluminum:	2.17 wt%
potassium:	1.37 wt%
magnesium:	1.25 wt%
iron:	1.01 wt%
hydrogen:	0.75 wt%
sodium:	0.63 wt%

Problem: Rocky Flats (cont.)



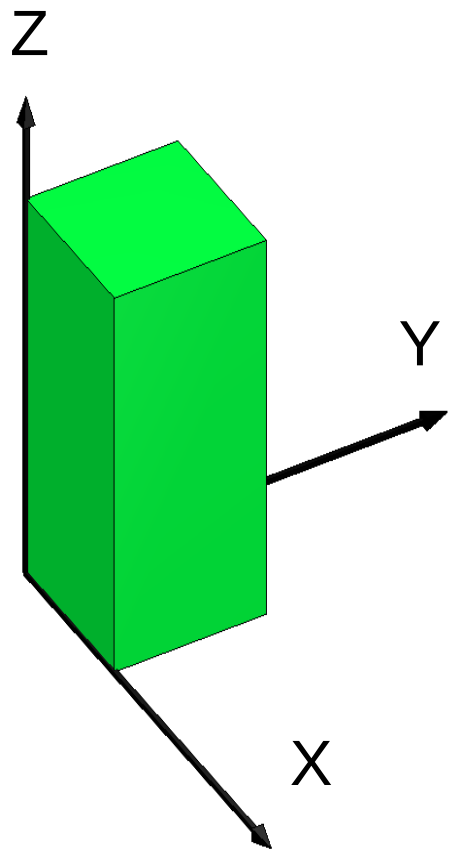
Calculated k-eff values

	KENO-VI	Shift
Green block	1.0070 +/- 0.0017	1.000393 +/- 0.002452
Rocky Flats	1.0060 +/- 0.0025	1.001669 +/- 0.004153

Your results may differ based on the options and computer used

Solution: Green block

```
=csas6
raffety and mihalczo u(2)f4-2 unreflected green block
ce_v7.1_endf
read composition
 uf4          1 0.5113 293  92235 2  92238 98   end
 paraffin     1 0.4887 293   end
end composition
read geometry
global unit 1
 cuboid 1    56.22 0.0   56.22  0.0   122.47  0.0
 media 1 1 1
 boundary 1
end geometry
end data
end
```

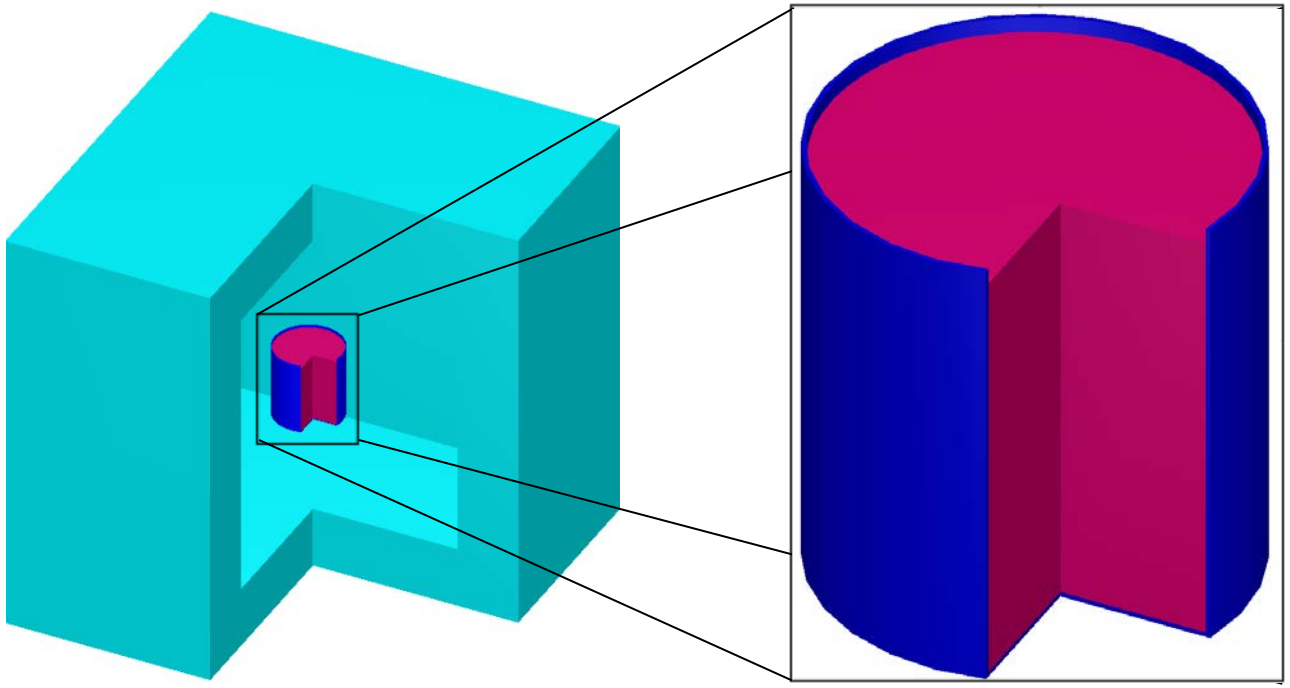


Solution: Rocky Flats

```
=csas6
rocky flats uo2(no3)2 soln 334.77 g u/l 28.01 cm diam conc refl
ce_v7.1_endf
read composition
 solution   mix=1  rho[uo2(no3)2]=334.77 92234 1.022
                                         92235 93.172
                                         92236 0.434
                                         92238 5.372
                molar[hno3]=0.521
                density=? vol_frac=1
  end solution
 al          2  end
 wtptconcrete    3  2.32  10
   8000 48.49
   20000 23.31
   14000 15.5
   6000 5.52
   13027 2.17
   19000 1.37
   12000 1.25
   26000 1.01
   1000 0.75
   11000 0.63  end
end composition
```

Solution: Rocky Flats (cont.)

```
read geometry
global unit 1
com='global unit 1'
 cylinder   5   14.005      29.68   28.6
 cylinder 10   14.005      28.6       0
 cylinder 20   14.325      29.68    -0.32
 cuboid 30     64.6    -57.4     57.4    -64.8     80.0    -41.1
 cuboid 40     90.3    -83.1     83.1    -90.5    105.7    -66.8
 media 0 1 5
 media 1 1 10
 media 2 1 20 -10 -5
 media 0 1 30 -20
 media 3 1 40 -30
 boundary 40
end geometry
end data
end
```



Summary of Module 1

- KENO-VI rules and restrictions
- Geometry region descriptions and definitions
- Godiva example, Green Block and Rocky Flats exercises

OAK RIDGE National Laboratory

Sub-module 2

- Geometry modification data



Geometry modification data

- These options modify the previous geometric volume and are always performed sequentially, in the order listed below:
  1. Chord: Truncates the volume with a plane perpendicular to a major axis
  2. Rotate: Rotates the volume
  3. Origin: Translates the volume origin, thus translating the volume

Note that the order matters: If all three geometry modifiers are present, first "chord" is performed, then "rotate", and then "origin". Use Fulcrum to double-check your modifications.

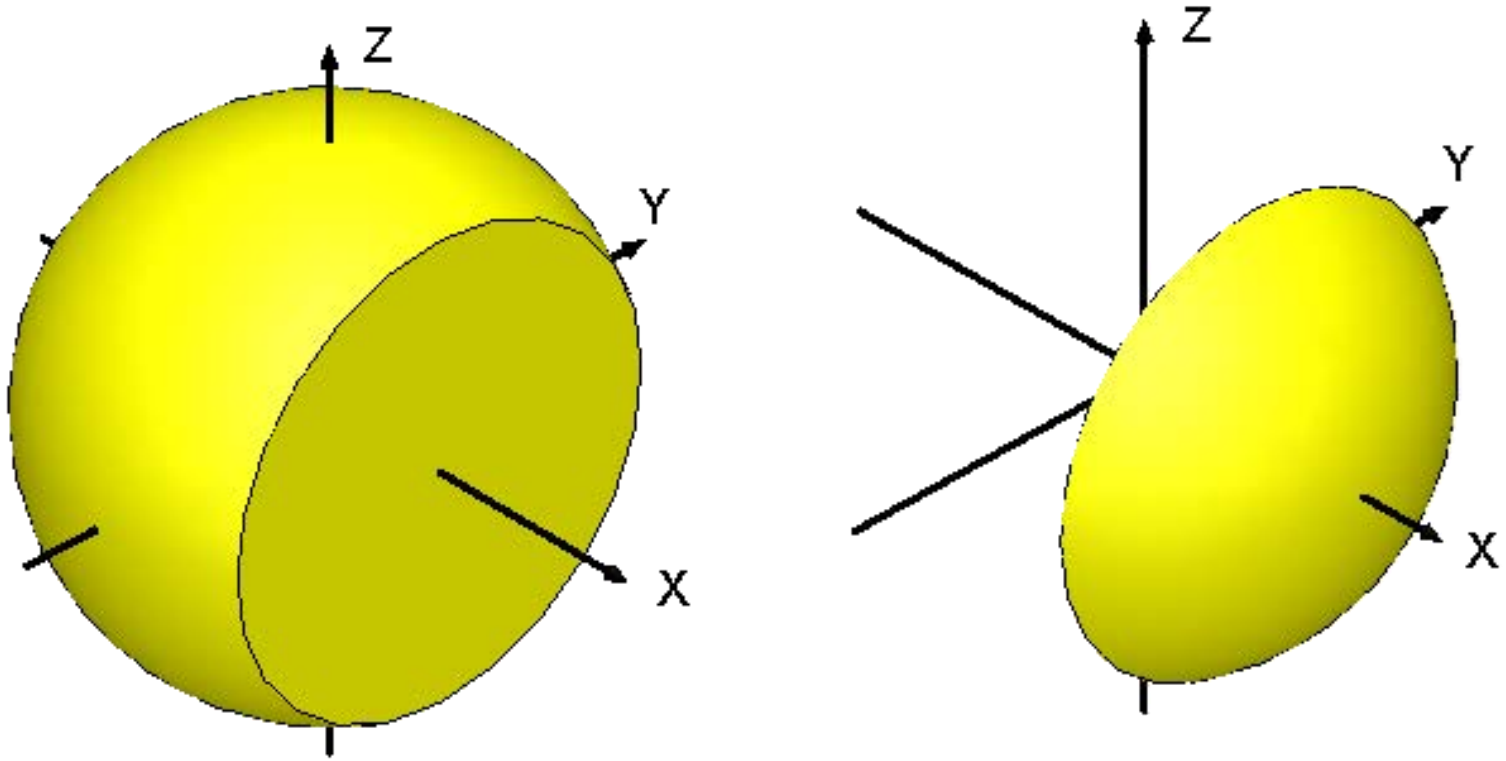
Geometry modification keywords

Keyword	Input Values
CHORD	+X=, -X=, +Y=, -Y=, +Z=, -Z=
ROTATE	A1=, A2=, A3=
ORIGIN	X=, Y=, Z=

CHORD data

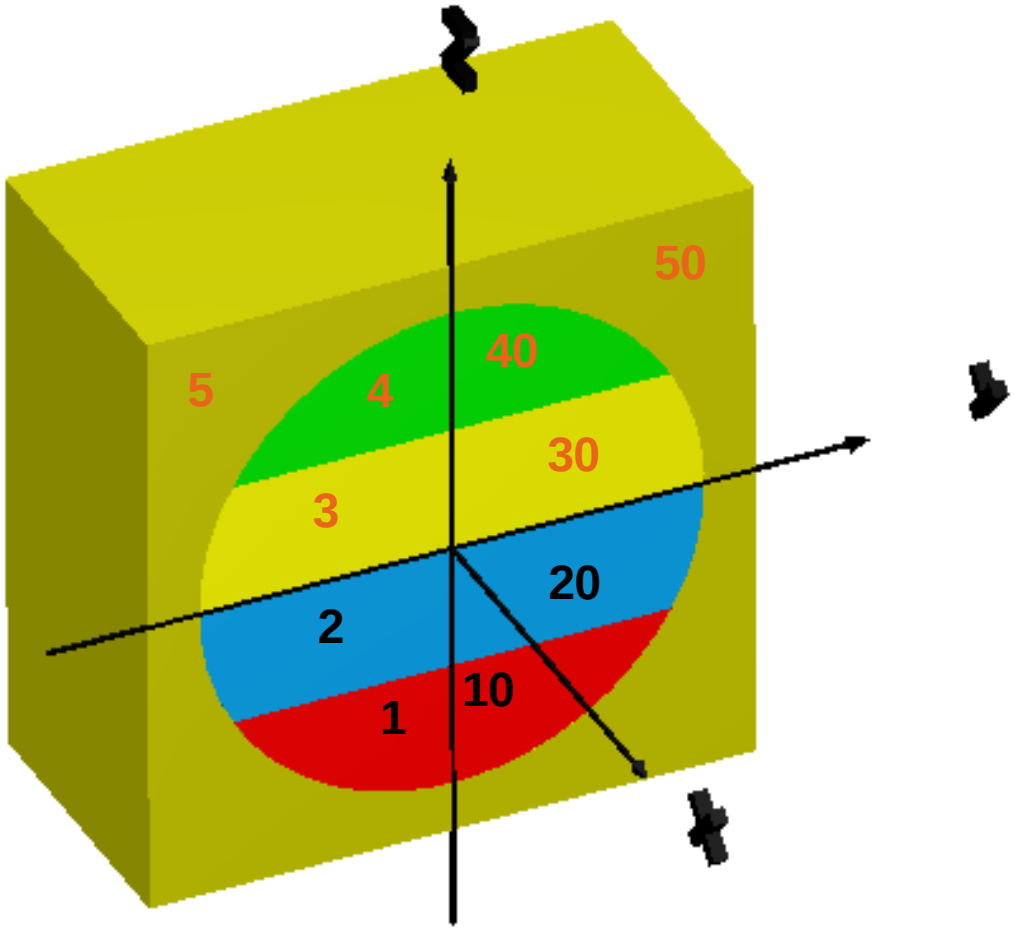
- Truncates a geometric volume with a plane perpendicular to a major axis
- Applies to the Geometry record that immediately precedes it
- Contains the following components
  - CHORD keyword
  - Subordinate keywords
    - +X=, -X=, +Y=, -Y=, +Z=, -Z=
  - Unlike KENO V.a, the subordinate keyword determines which side will be kept

CHORD examples



SPHERE 1 10.0 CHORD -X=5.0 SPHERE 2 10.0 CHORD +X=5.0

Sphere cut into four regions using CHORDs




```
global unit 1
sphere  10  10.0  chord -z=-5.0
sphere  20  10.0  chord +z=-5.0
                      chord -z=0.0
sphere  30  10.0  chord +z=0.0
                      chord -z=5.0
sphere  40  10.0  chord +z=5.0
cuboid  50  6p12.0
media  1  1  10
media  2  1  20
media  3  1  30
media  4  1  40
media  5  1  50  -10 -20 -30 -40
boundary  50
```

OAK RIDGE National Laboratory

Questions about CHORD?

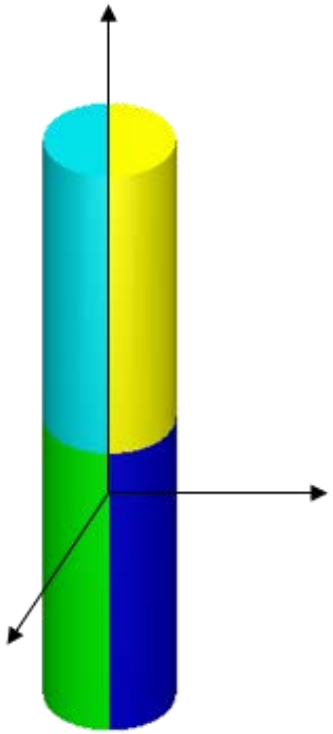


ROTATE Data

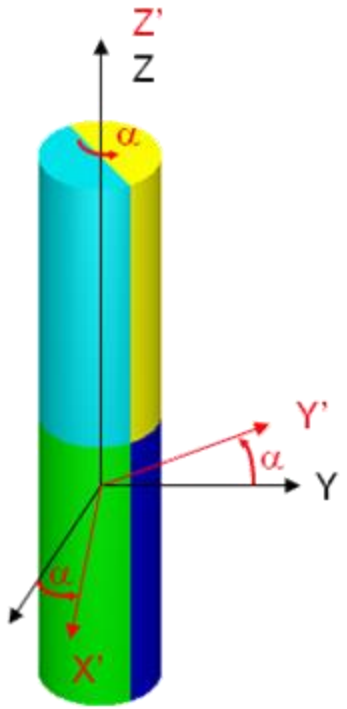
- Rotates a volume about the origin using the Euler angle x-convention
- Applies to the Geometry or HOLE record that immediately precedes it
- Contains the following components
  - ROTATE keyword
  - Subordinate keywords

    A1=, A2=, A3=

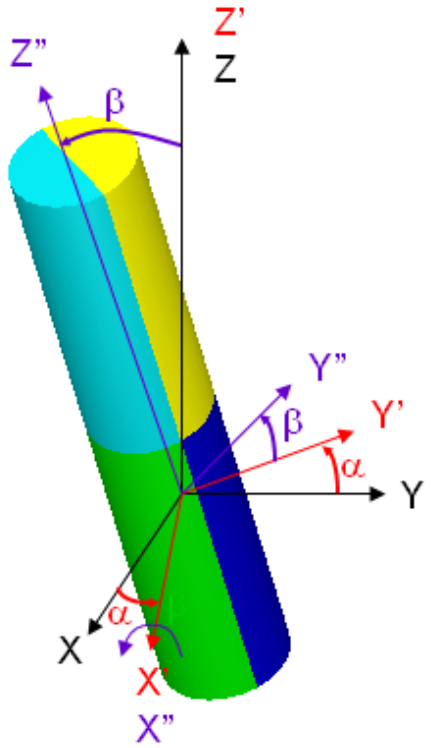
CYLINDER Label R Zt Zb ROTATE A1=α A2=β A3=γ



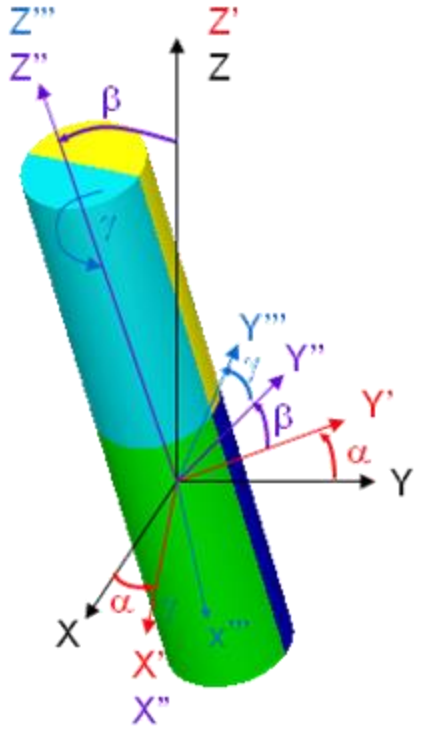
Cylinder before rotation



First rotation, A1 = α is counter-clockwise about the Z axis



Second rotation, A2 = β is counter-clockwise about the X' axis



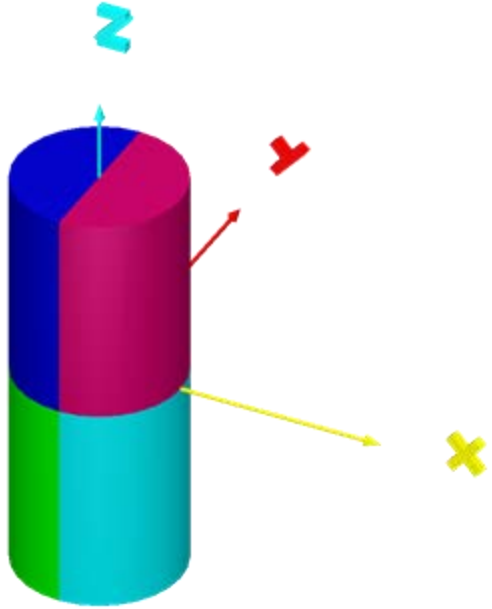
Third rotation, A3 = γ is counter-clockwise about the Z'' axis

Right-handed coordinate system: thumb as axis pointing positive, fingers rotate in positive direction.

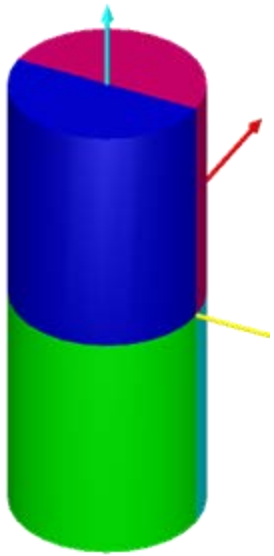
Another way of thinking about it…

CYLINDER Label R Zt Zb ROTATE A1=α A2=β A3=γ

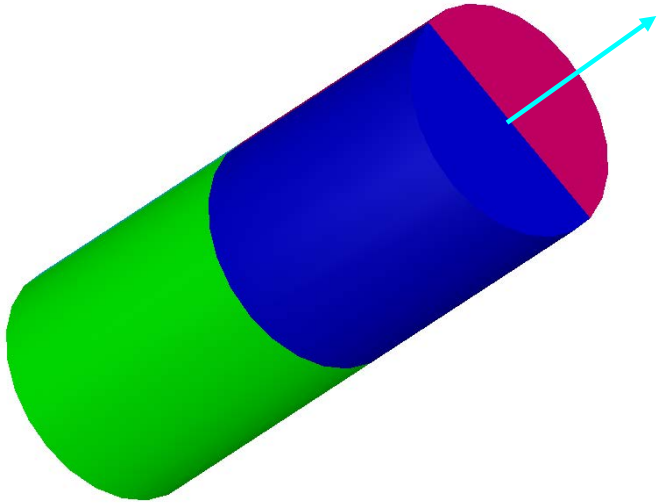
- A1 rotates about Z to position the next axis of rotation for A2
- A2 rotates about X' and positions next axis of rotation for A3



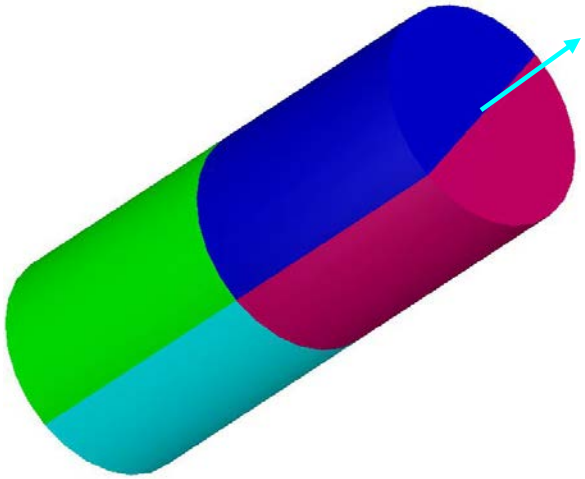
Initial object



A1=90°
A2 is now about red axis

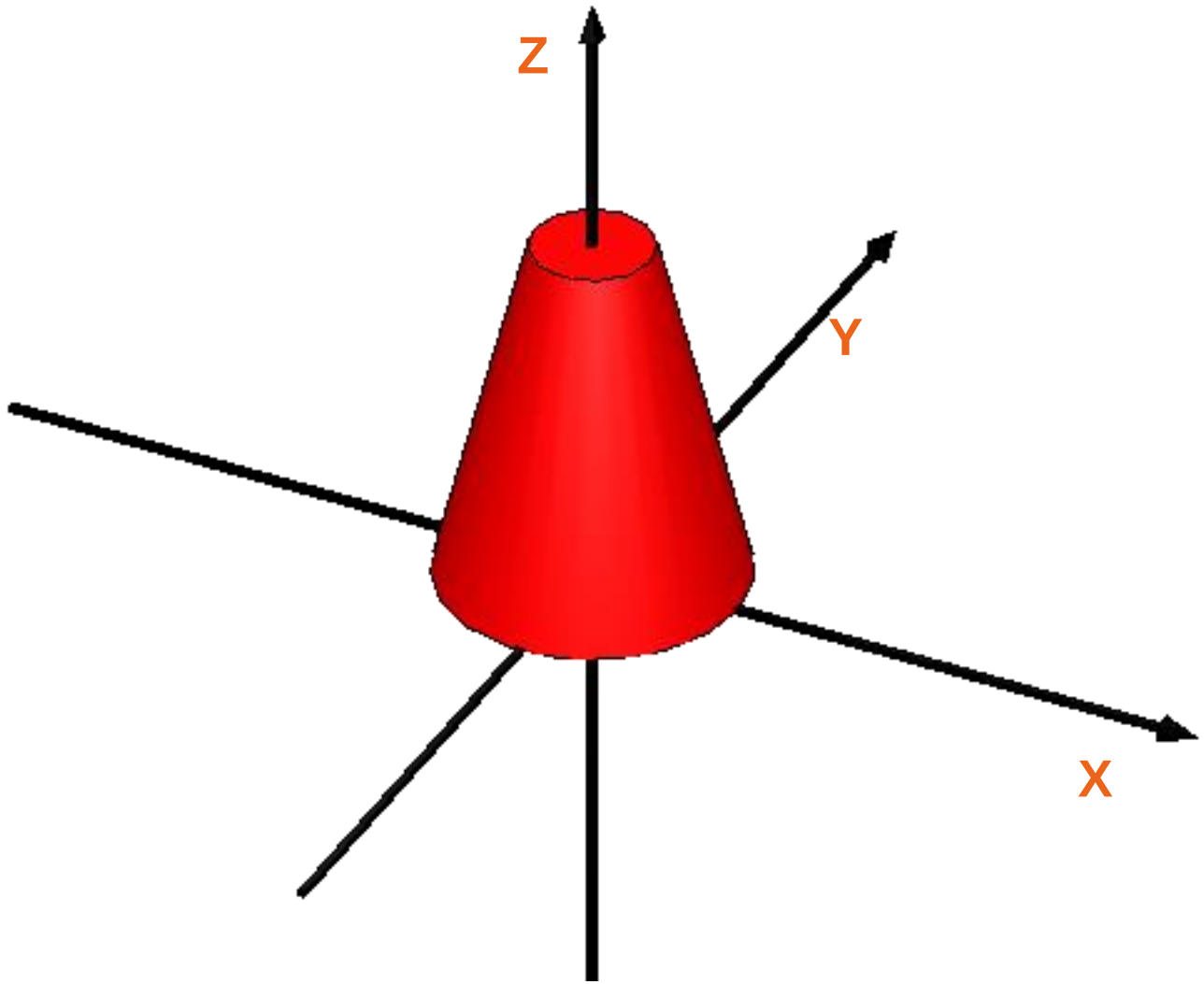


A1=90° A2=45° A3 is now about axis out the top



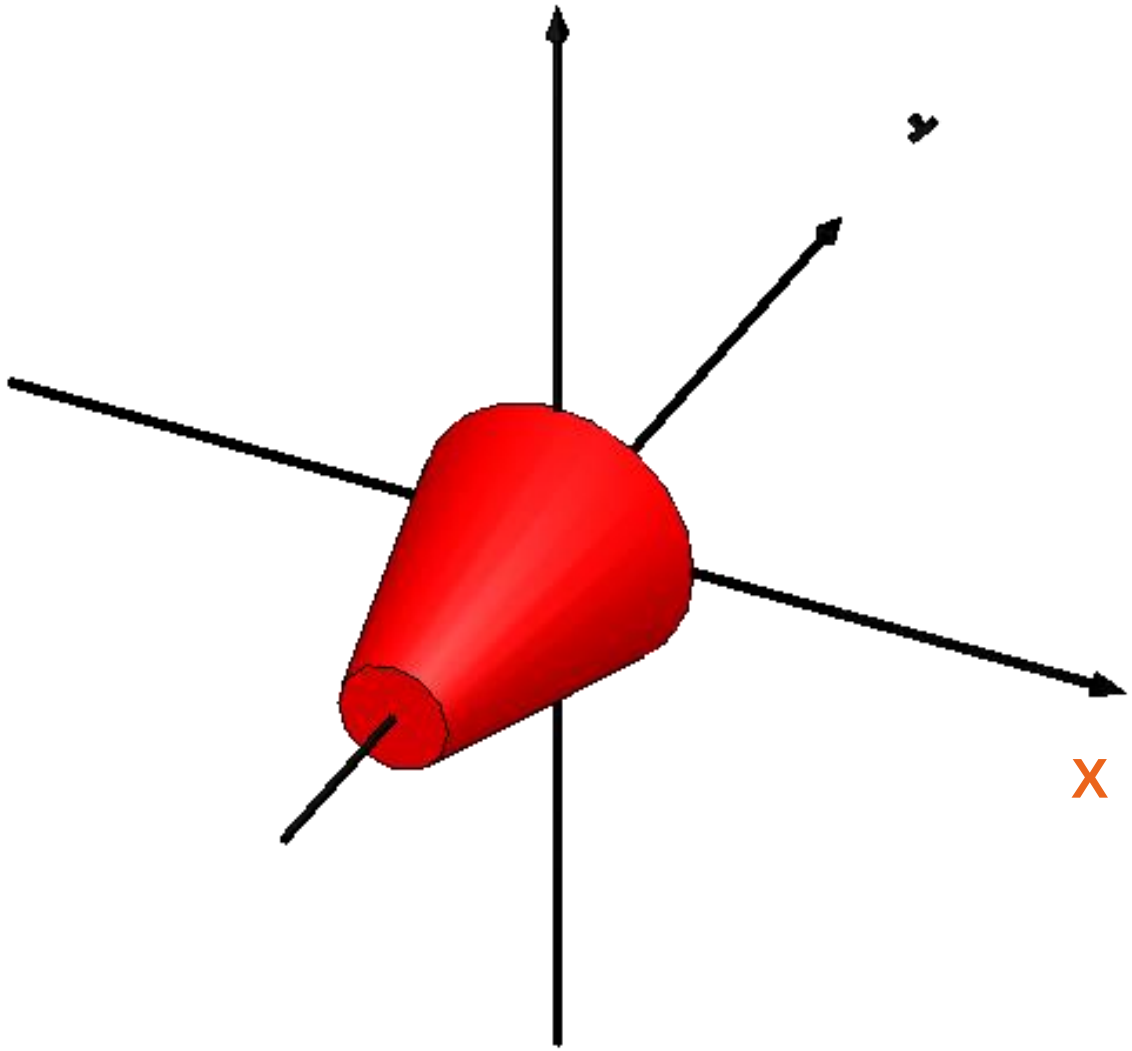
A1=90° A2=45° A3=-90°

Unrotated cone



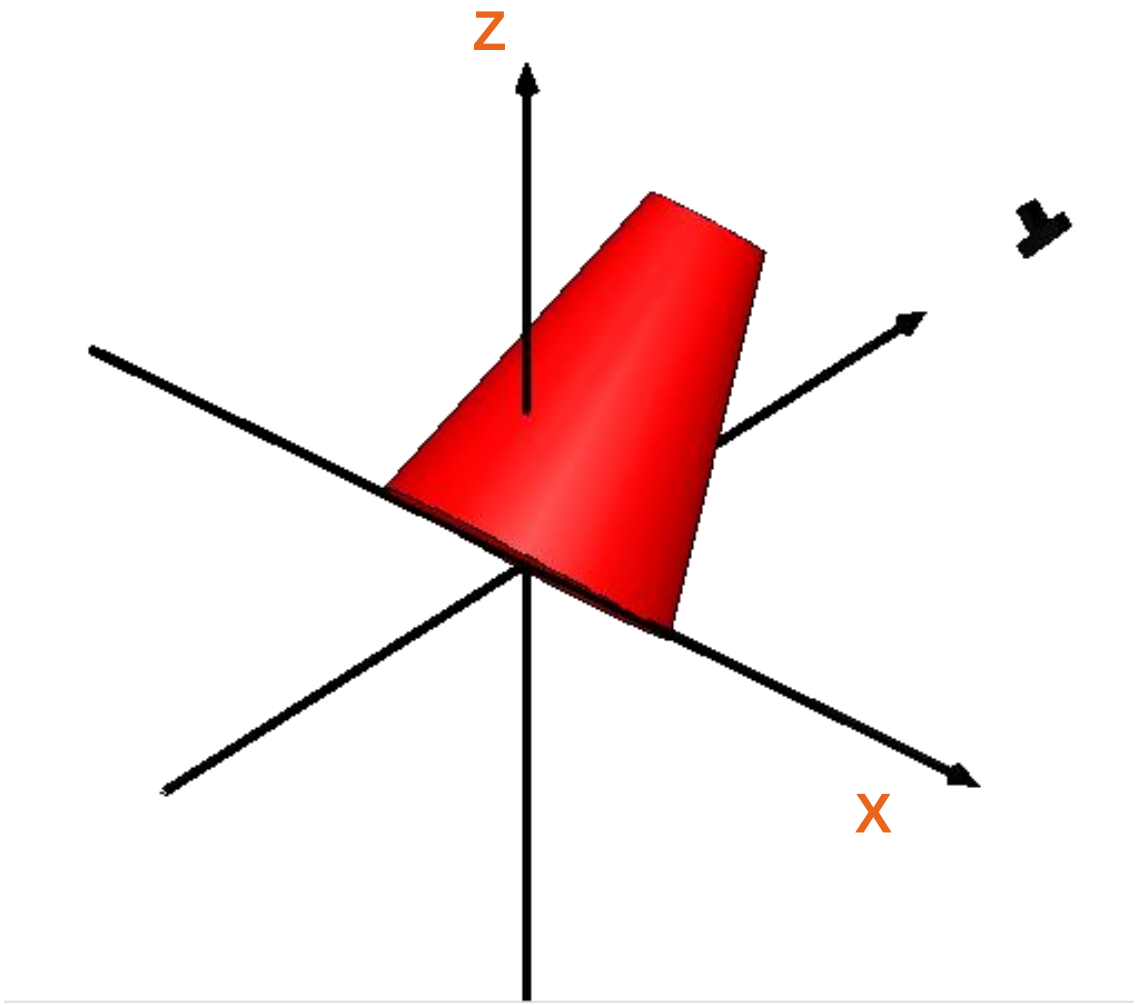

```
read geometry
global unit 1
com='cone'
 cone 1    2   12  5   0
 media 1 1 1
 boundary 1
end geometry
```

Cone rotated to Y-axis



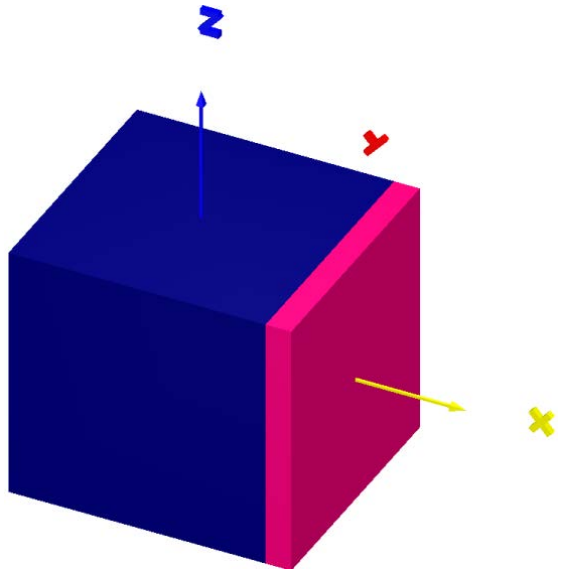

```
read geometry
global unit 1
com='y cone'
 cone 1  2 12  5 0
      rotate a2=90
 media 1 1 1
 boundary 1
end geometry
```

45 ° rotation around X-axis

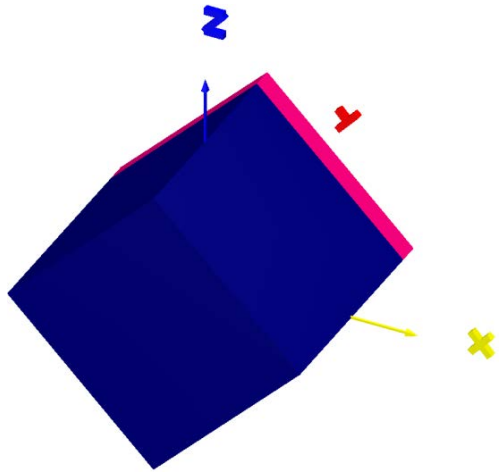



```
read geometry
global unit 1
com='45 deg rotation'
 cone 1  2 12  5 0
      rotate a2=-45
 media 1 1 1
 boundary 1
end geometry
```

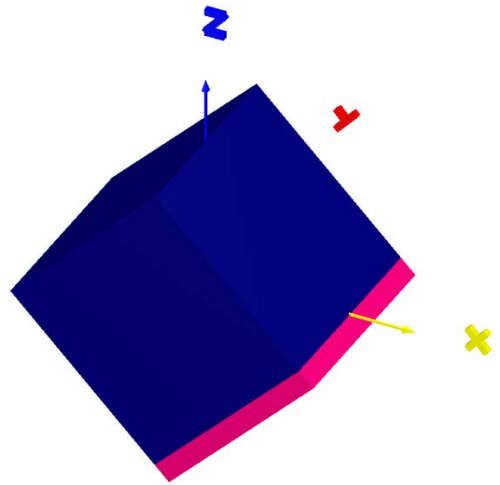
Cuboid examples



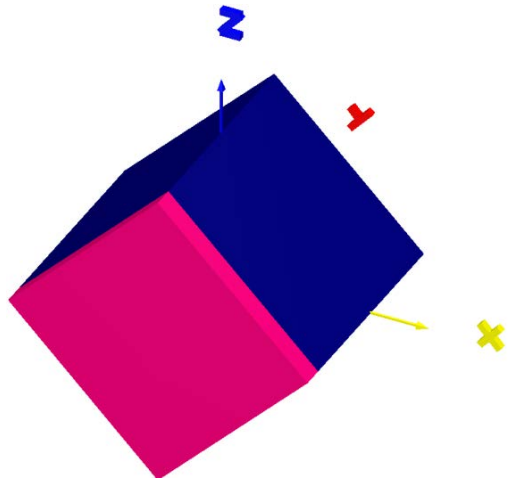
Initial object



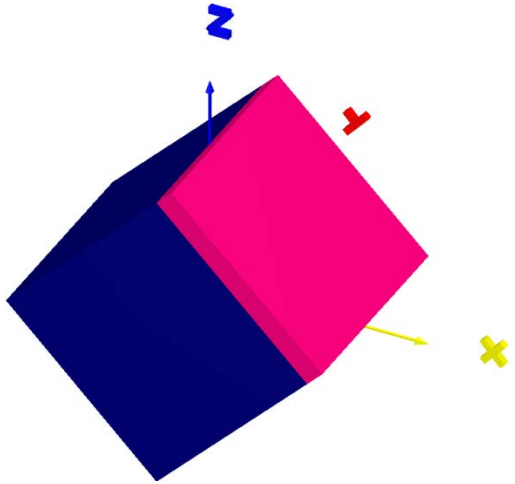
A1=90 A2=45 A3=0



A1=90 A2=45 A3=-90



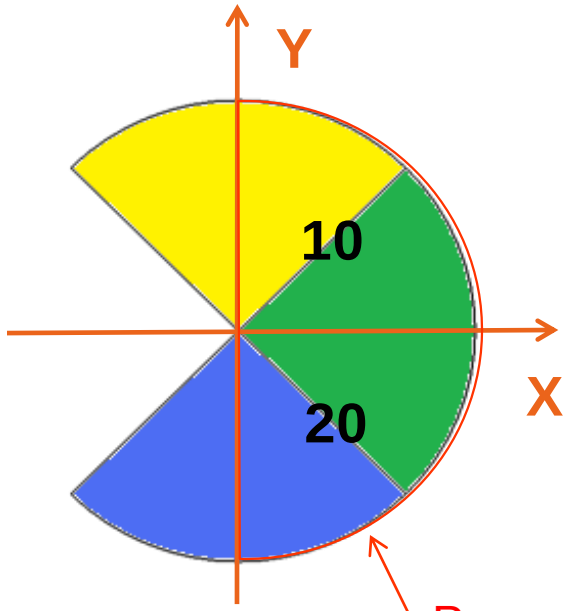
A1=-90 A2=45 A3=0

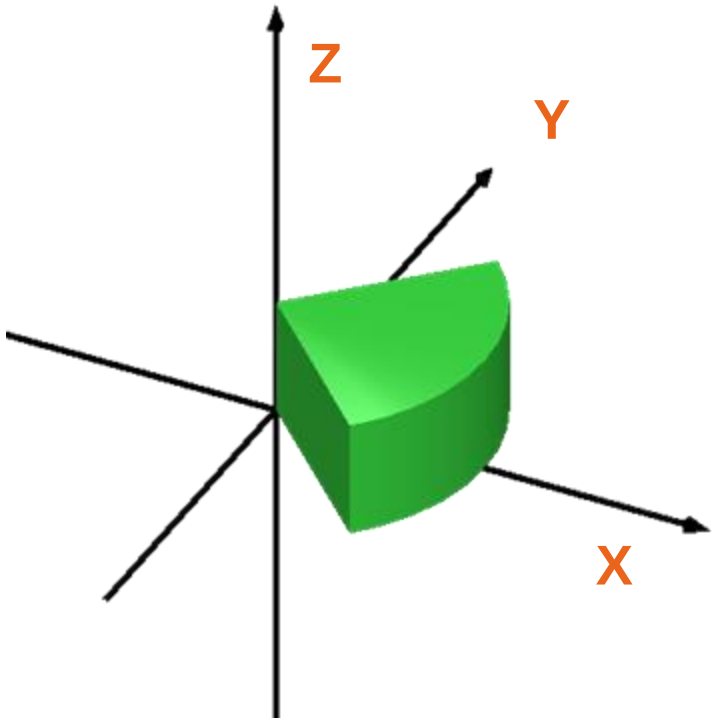


A1=-90 A2=45 A3=90

Rotations and CHORDs

```
unit 1
com='1/4 cylinder'
 cylinder 10  9.0 5.0  0  chord +x=0   rotate  a1=45
 cylinder 20  9.0 5.0  0  chord +x=0   rotate  a1=-45
 cuboid 30   4p9.0 5.0 0
 media 1 1 10 20
 media 0 1 10 -20
 media 0 1 20 -10
 media 0 1 30 -10 -20
 boundary 30
```





OAK RIDGE National Laboratory

Questions about ROTATE?

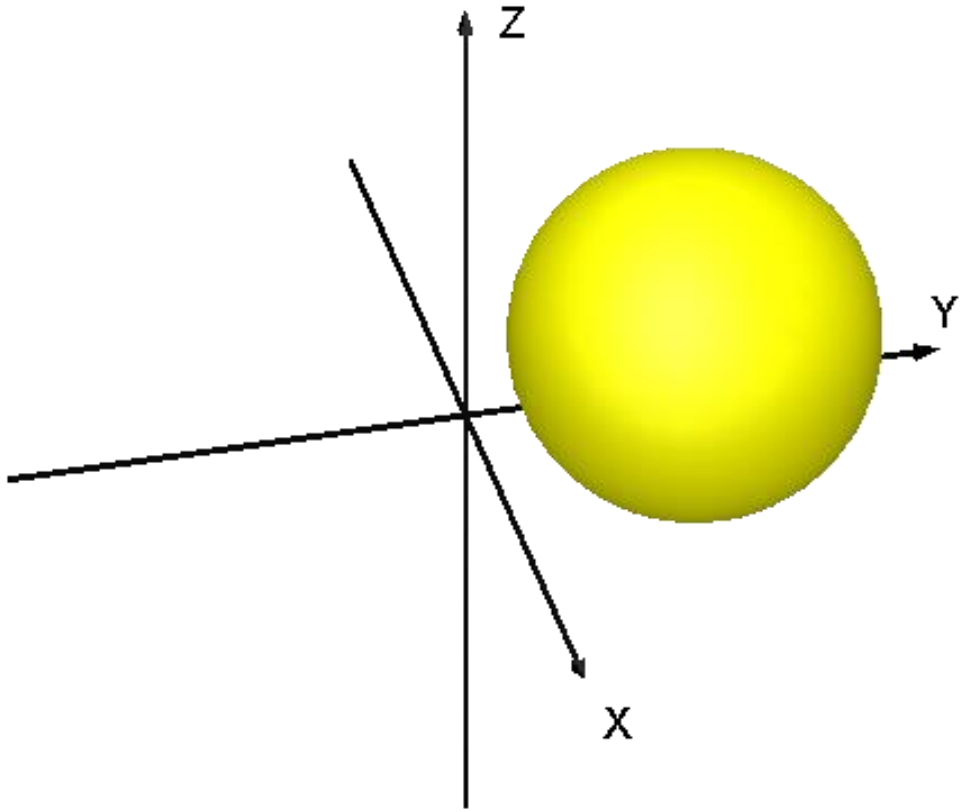


ORIGIN data

- Translates the origin of a volume to a new specified position
  - Only non-zero data need to be entered
- Applies to the Geometry or Hole record that immediately precedes it
- Contains the following components:
  - ORIGIN keyword
  - Subordinate keywords

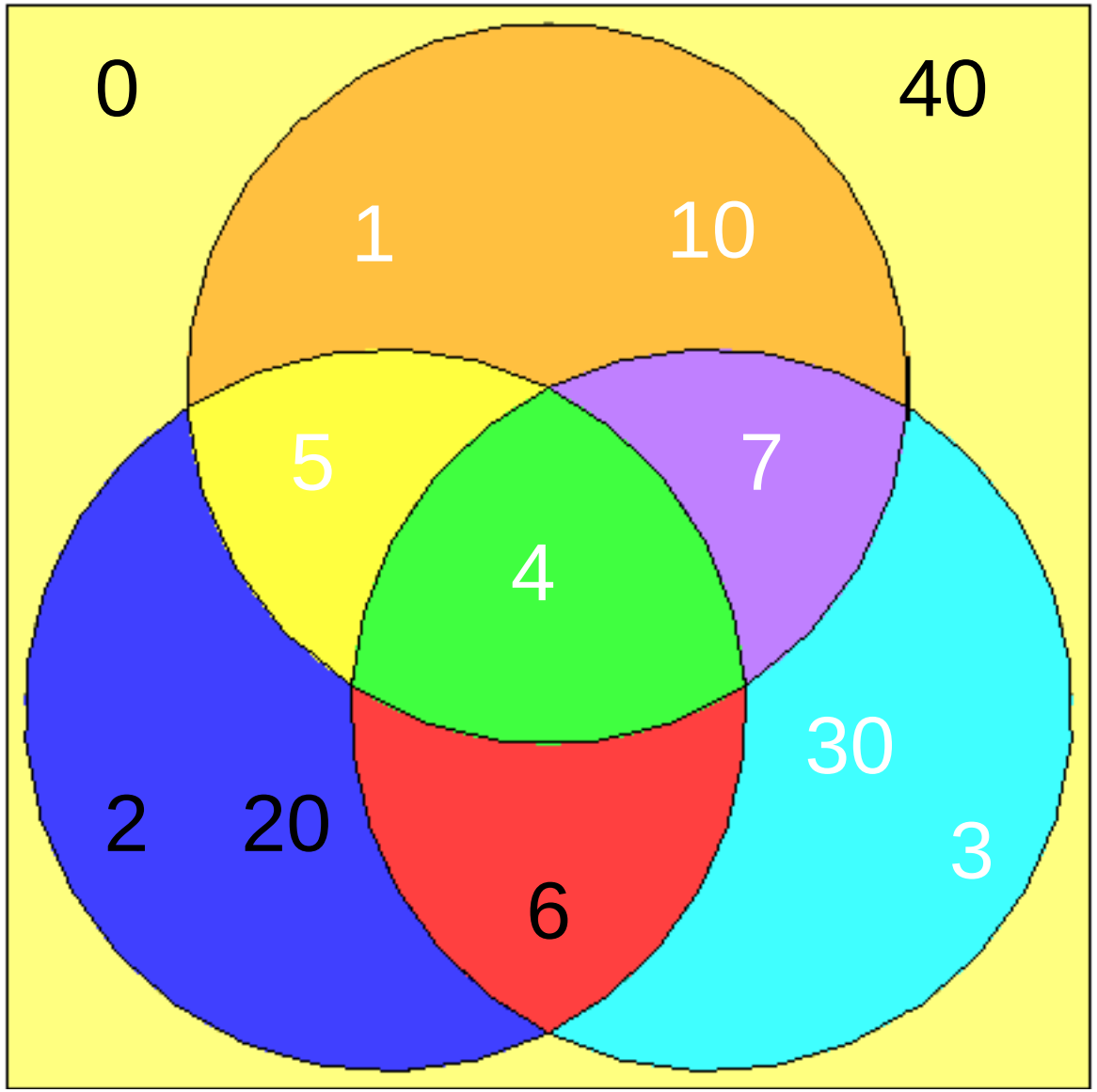
    X=, Y=, Z=

ORIGIN example



SPHERE 1 10.0 ORIGIN X=10.0 Y=10.0 Z=10.0

Three intersecting spheres




```
global unit 1
sphere  10  10.0  origin z=4.5
sphere  20  10.0  origin y=-4.5  z=-4.5
sphere  30  10.0  origin y=4.5  z=-4.5
cuboid  40  6p15.0
media  1  1  10  -20  -30
media  2  1  20  -10  -30
media  3  1  30  -10  -20
media  4  1  10   20   30
media  5  1  10   20  -30
media  6  1  20   30  -10
media  7  1  10   30  -20
media  0  1  40  -10  -20  -30
boundary  40
```

OAK RIDGE National Laboratory

Questions about ORIGIN?



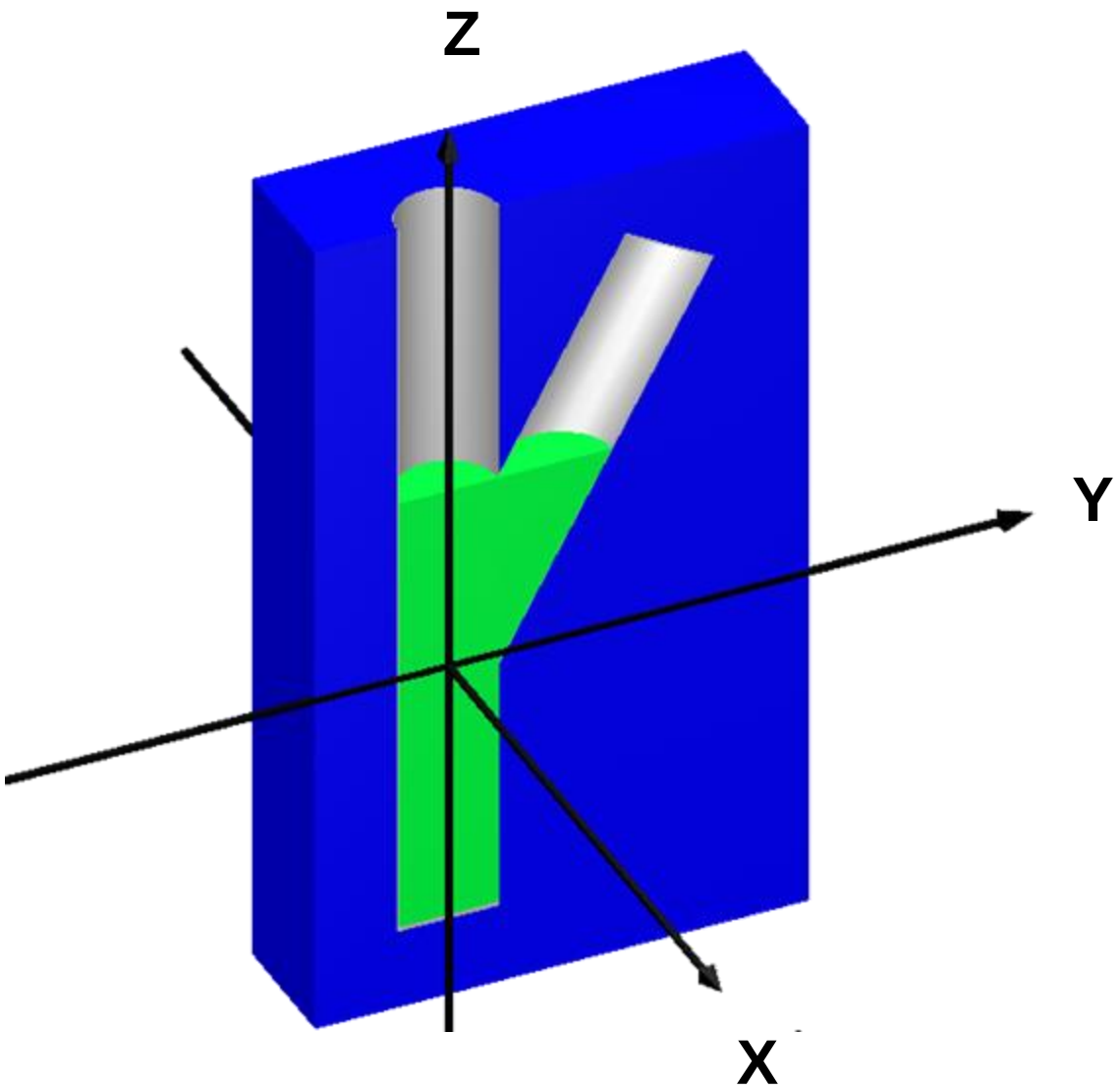
OAK RIDGE National Laboratory

Demo: Geometry modification data

- Y30



Y30



UO_2F_2 solution

specific gravity 2.0289
907.01 g U/l
Enrichment:
U-235 5.0 %
U-238 95.0 %

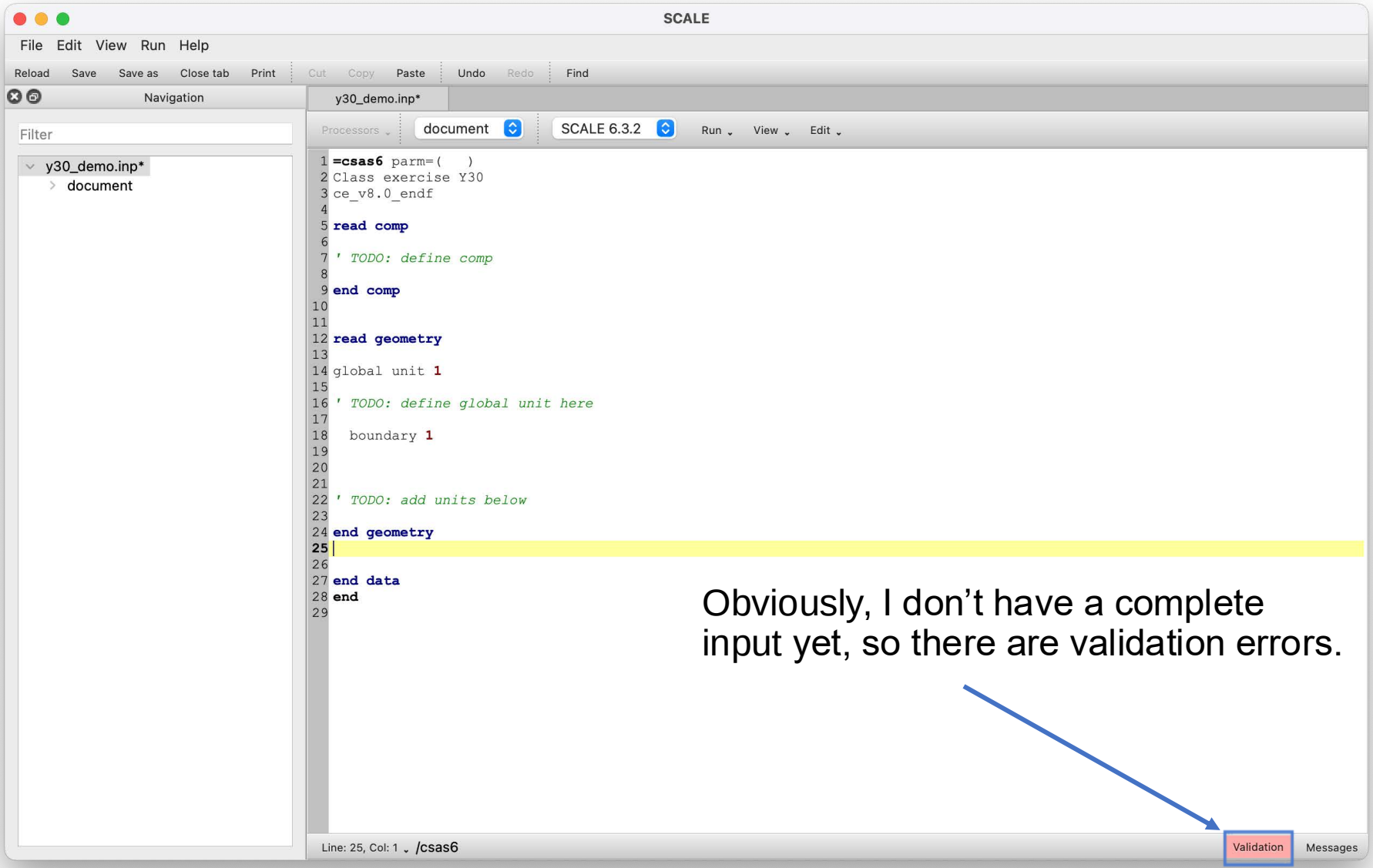
Aluminum Y-Pipe

(Dimensions taken from center of Y intersection)
13.95 cm inner radius
14.11 cm outer radius
1.30 cm bottom thickness
Side leg at 29.26° angle from vertical
Top of vertical pipe: 135.4 cm
Bottom of vertical pipe: -76.7 cm
Side leg length: 126.04 cm
Side leg sealed top: 0.64 cm thick
UO_2F_2 Solution Height: 52.8 cm

Water Reflector

+X = 37.0 cm; -X = -37.0 cm
+Y = 100.0 cm; -Y = -37.0 cm
+Z = 135.4 cm; -Z = -99.6 cm

General problem data



Create UO_2F_2 solution

```
5 read composition
6  solution mix=1 rho[uo2f2]=907.01 92235 5 92238 95
7           density=2.0289 temperature=293 vol_frac=1
8   end solution
9
```

Create Al and H_2O compositions

```
 8  al      2   1   293   end
 9  h2o     3   1   293   end
10 end composition
11
```

Create Geometry records

1. Inside of vertical pipe
2. Vertical pipe wall
3. Inside of angled pipe
4. Angled pipe wall
5. Z-plane for level of solution
6. Cuboid for water reflector

```
14 global unit 1
15  cylinder 10   13.95   135.4 -75.4
16  cylinder 20   14.11   135.4 -76.7
17  cylinder 30   13.95   125.4  0     rotate a2=-29.26
18  cylinder 40   14.11   126.04 0     rotate a2=-29.26
19  plane    50   zpl=-1    con=52.8
20  cuboid   60   37 -37   100 -37   135.4 -99.6
21
```

Create Media records

1. Solution in vertical pipe
2. Vertical pipe wall
3. Solution in angled pipe
4. Angled pipe wall
5. Void in vertical pipe
6. Void in angled pipe
7. Water reflector

```
21 '
22  media 1 1   10 50
23  media 2 1   20 -10 -30
24  media 1 1   30 -10 50
25  media 2 1   40 -30 -20
26  media 0 1   10 -50
27  media 0 1   30 -50 -10
28  media 3 1   60 -40 -20
29
```

Create Boundary record

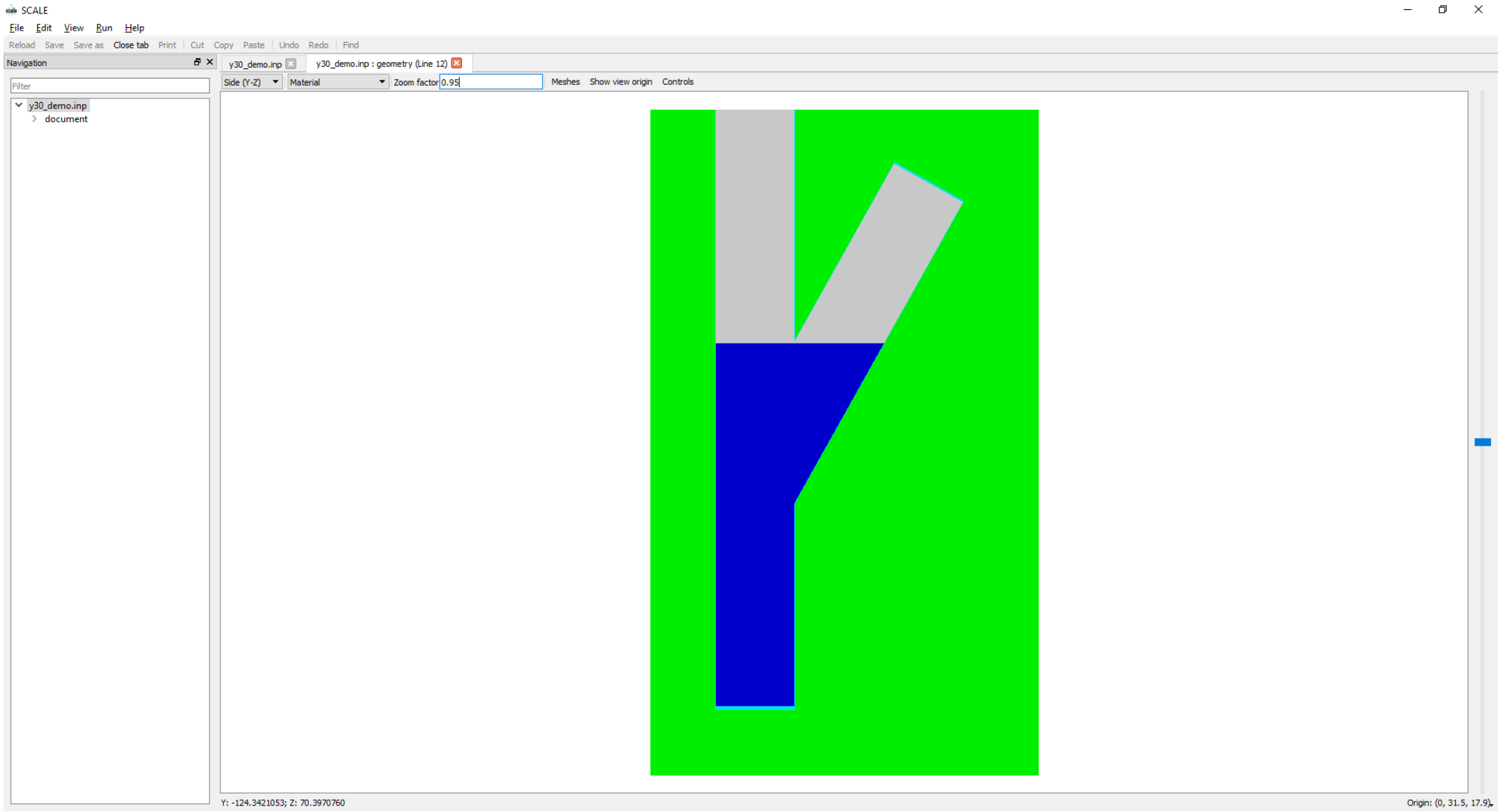
```
read geometry

global unit 1
 cylinder 10  13.95  135.4 -75.4
 cylinder 20  14.11  135.4 -76.7
 cylinder 30  13.95  125.4  0     rotate a2=-29.26
 cylinder 40  14.11  126.04 0     rotate a2=-29.26
 plane    50  zpl=-1   con=52.8
 cuboid   60  37 -37  100 -37  135.4 -99.6
'
 media 1 1  10 50
 media 2 1  20 -10 -30
 media 1 1  30 -10 50
 media 2 1  40 -30 -20
 media 0 1  10 -50
 media 0 1  30 -50 -10
 media 3 1  60 -40 -20
'
boundary 60
end geometry
```

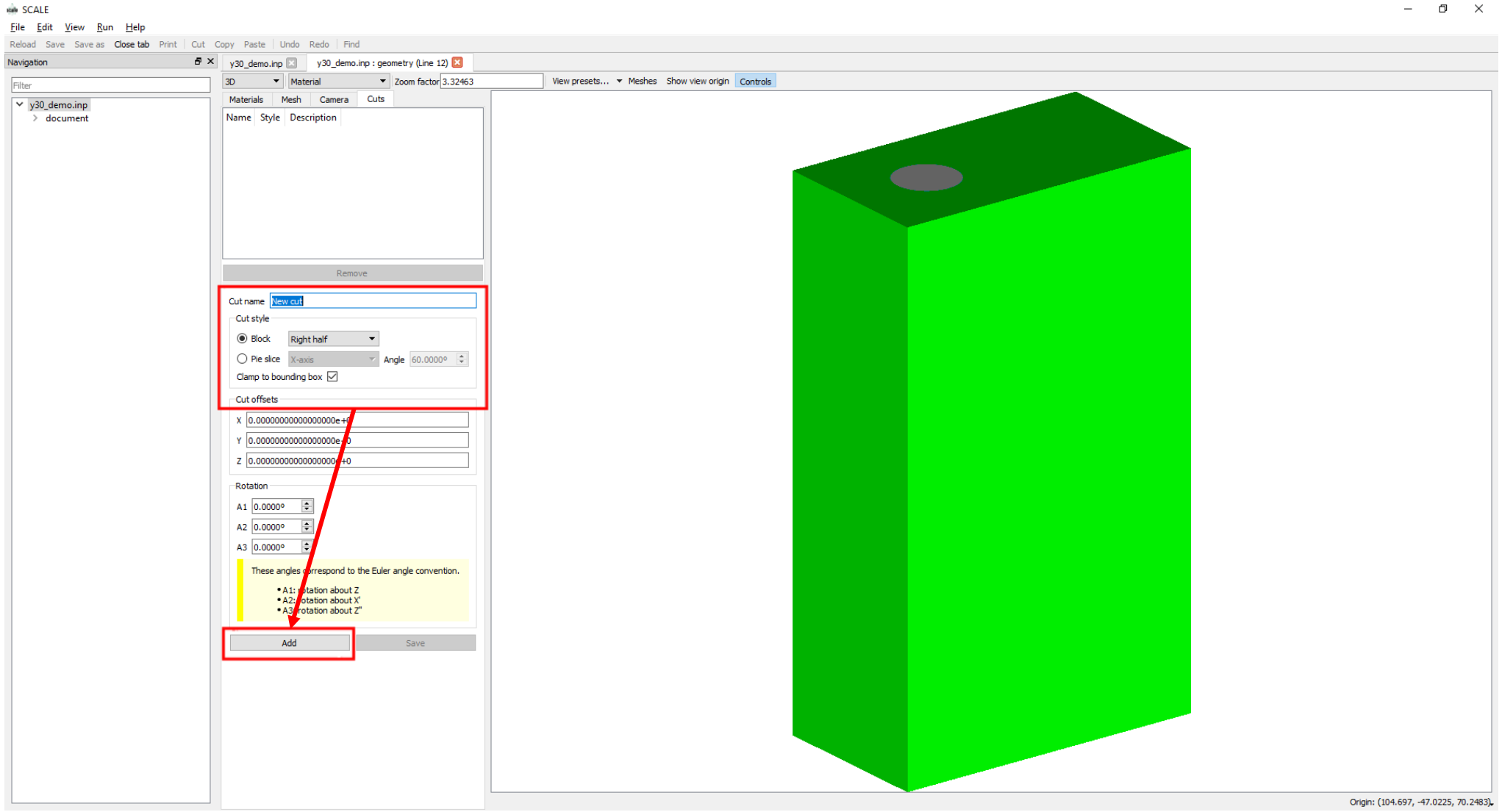
End Geometry block and input

```
read geom
global unit 1
cylinder 10 13.95 135.4 -75.4
cylinder 20 14.11 135.4 -76.7
cylinder 30 13.95 125.4 0 rotate a2=-29.26
cylinder 40 14.11 126.04 0 rotate a2=-29.26
plane 50 zpl=-1 con=52.8
cuboid 60 37 -37 100 -37 135.4 -99.6
'
media 1 1 10 50
media 2 1 20 -10 -30
media 1 1 30 -10 50
media 2 1 40 -30 -20
media 0 1 10 -50
media 0 1 30 -50
media 3 1 60 -40 -20
'
boundary 60
end geom
end data
end
```

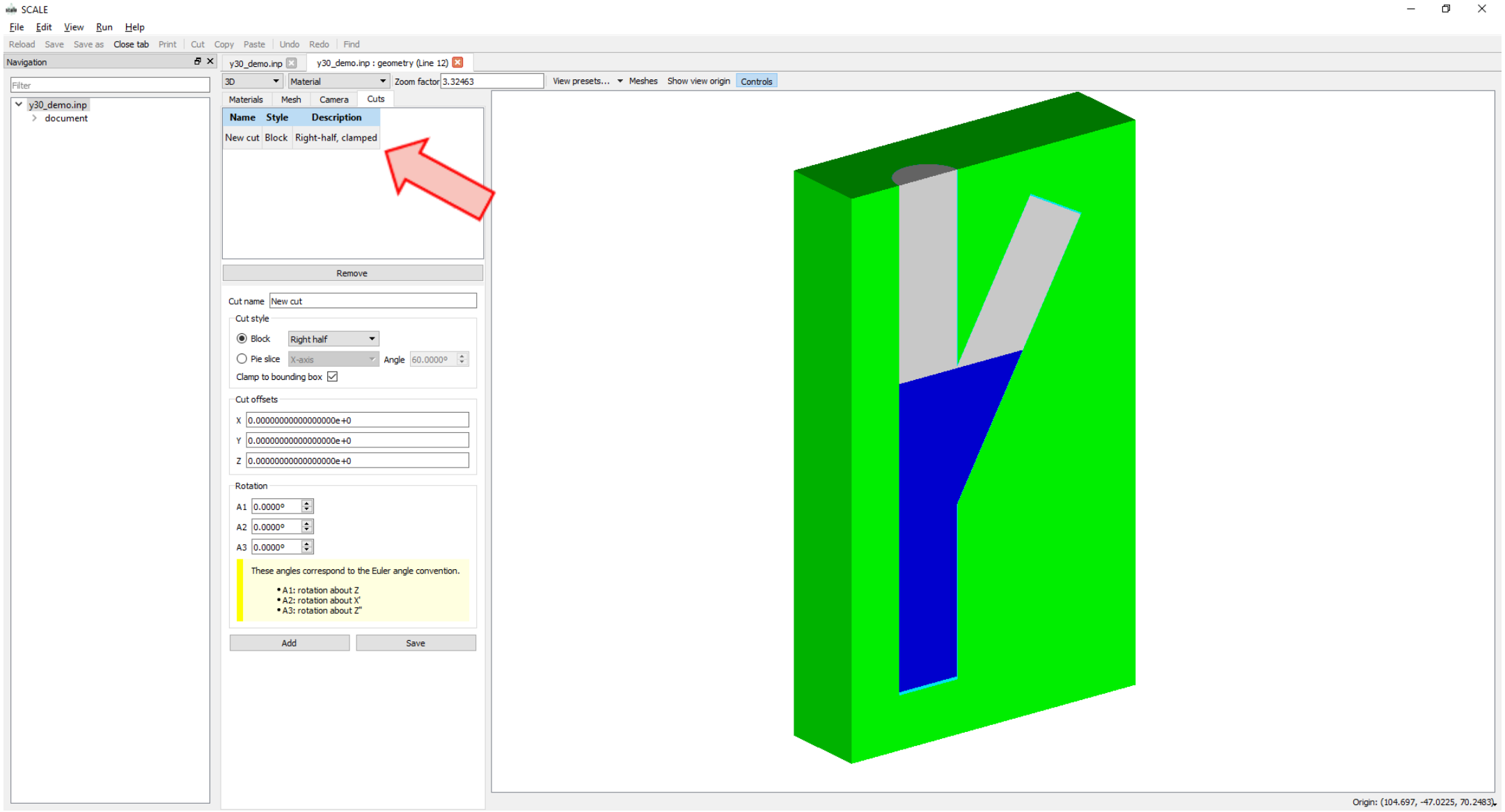
Visualize in Fulcrum: 2D (side)



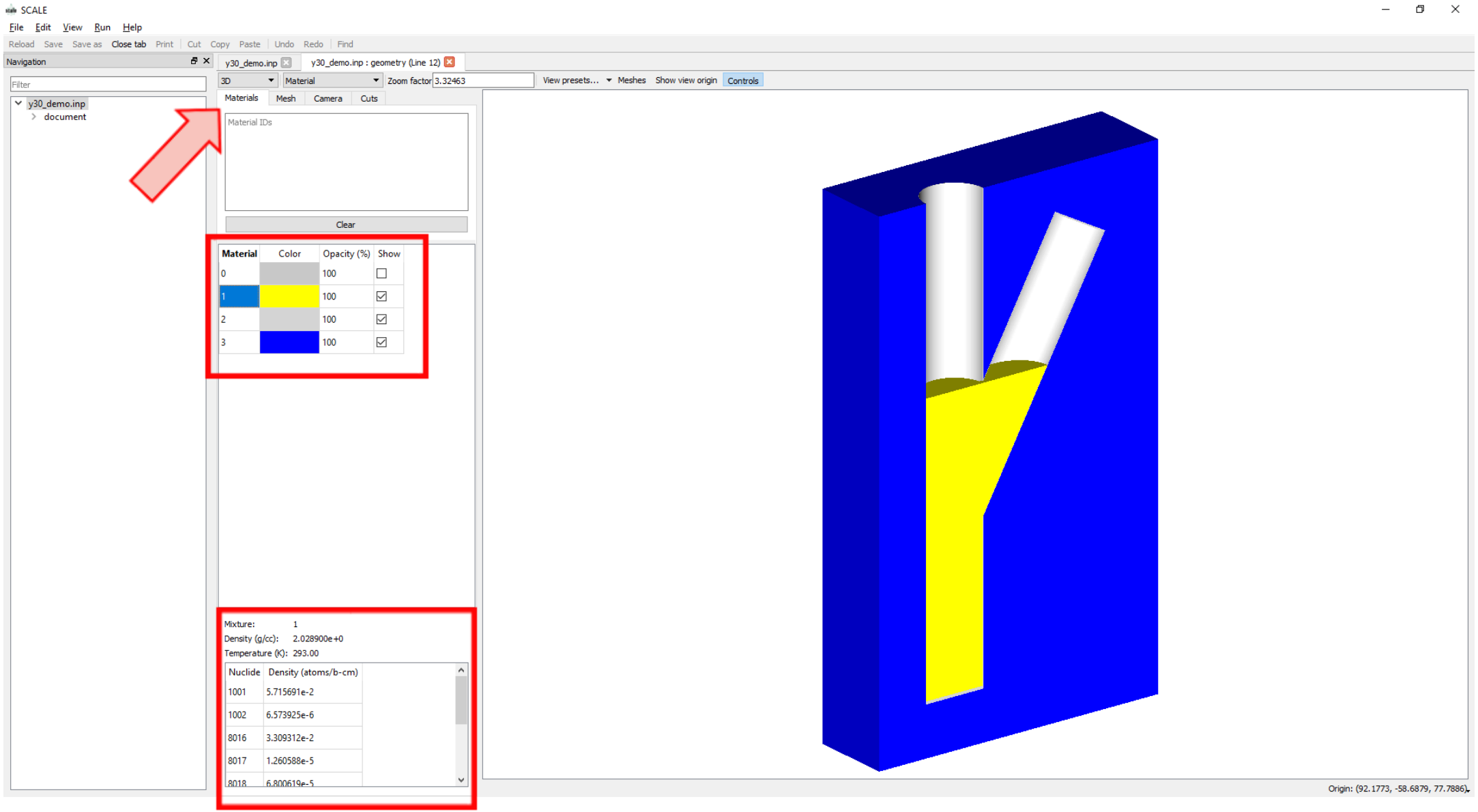
Visualize in Fulcrum: 3D



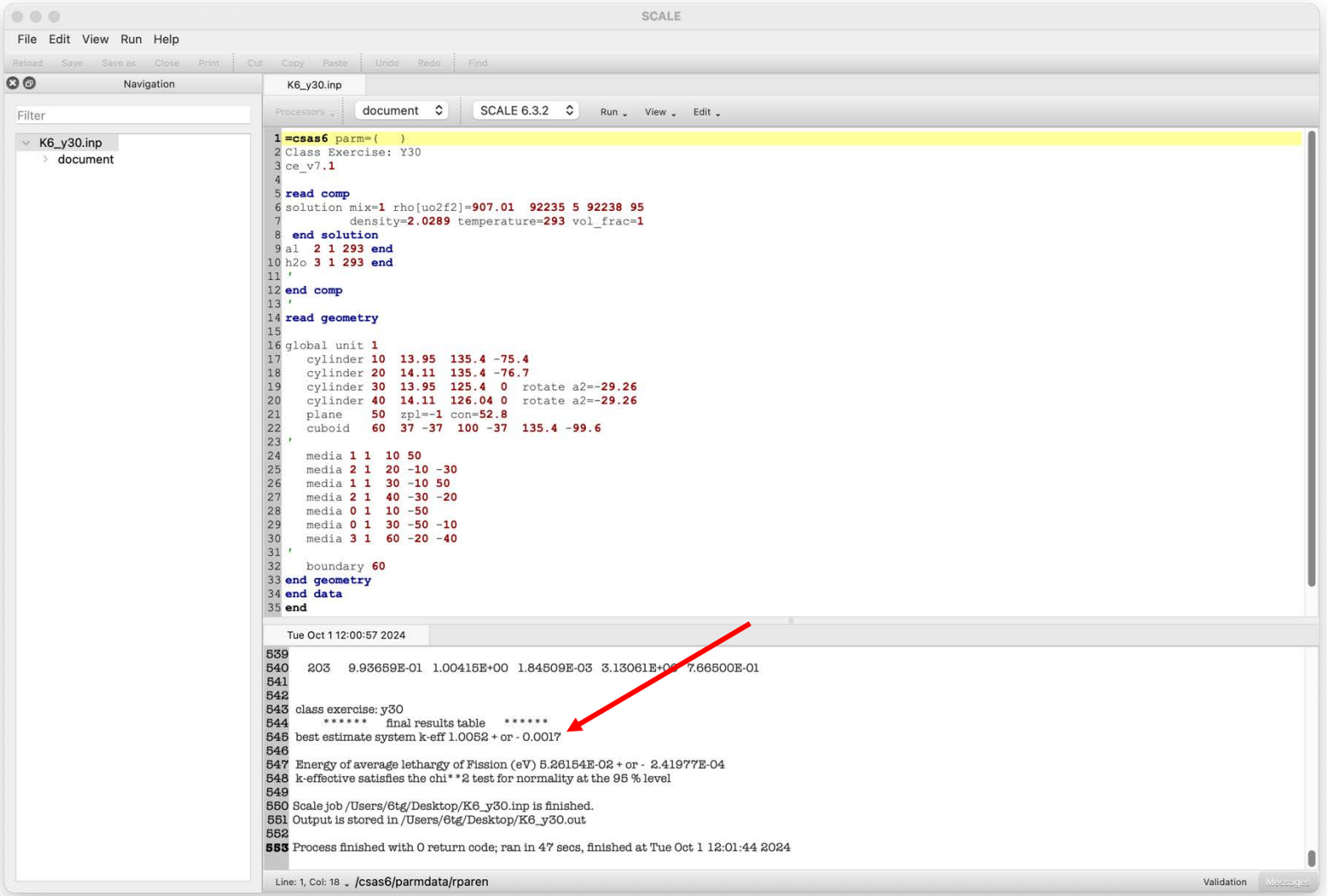
Visualize in Fulcrum: 3D Part 2



Visualize in Fulcrum: 3D Part 3



Run!


SCALE
File Edit View Run Help
Navigation
K6_y30.inp
document
SCALE 6.3.2
Run
View
Edit
=csas6 parm=( )
Class Exercise: Y30
ce_v7.1
read comp
solution mix=1 rho[uo2f2]=907.01 92235 5 92238 95
density=2.0289 temperature=293 vol_frac=1
end solution
al 2 1 293 end
h2o 3 1 293 end
'
end comp
'
read geometry
global unit 1
cylinder 10 13.95 135.4 -75.4
cylinder 20 14.11 135.4 -76.7
cylinder 30 13.95 125.4 0 rotate a2=-29.26
cylinder 40 14.11 126.04 0 rotate a2=-29.26
plane 50 zpl=-1 con=52.8
cuboid 60 37 -37 100 -37 135.4 -99.6
'
media 1 1 10 50
media 2 1 20 -10 -30
media 1 1 30 -10 50
media 2 1 40 -30 -20
media 0 1 10 -50
media 0 1 30 -50 -10
media 3 1 60 -20 -40
'
boundary 60
end geometry
end data
end
Tue Oct 1 12:00:57 2024
203 9.93659E-01 1.00415E+00 1.84509E-03 3.13061E+00 7.66500E-01
class exercise: y30
****** final results table ******
best estimate system k-eff 1.0052 + or - 0.0017
Energy of average lethargy of Fission (eV) 5.26154E-02 + or - 2.41977E-04
k-effective satisfies the chi**2 test for normality at the 95 % level
Scale job /Users/6tg/Desktop/K6_y30.inp is finished.
Output is stored in /Users/6tg/Desktop/K6_y30.out
Process finished with 0 return code; ran in 47 secs, finished at Tue Oct 1 12:01:44 2024
Line: 1, Col: 18
/csas6/parmdata/rparen
Validation

OAK RIDGE National Laboratory

Exercises for sub-module 2

- KENO-VI geometry exercises for geometry modification data
  - Set 1: Hopper and cross

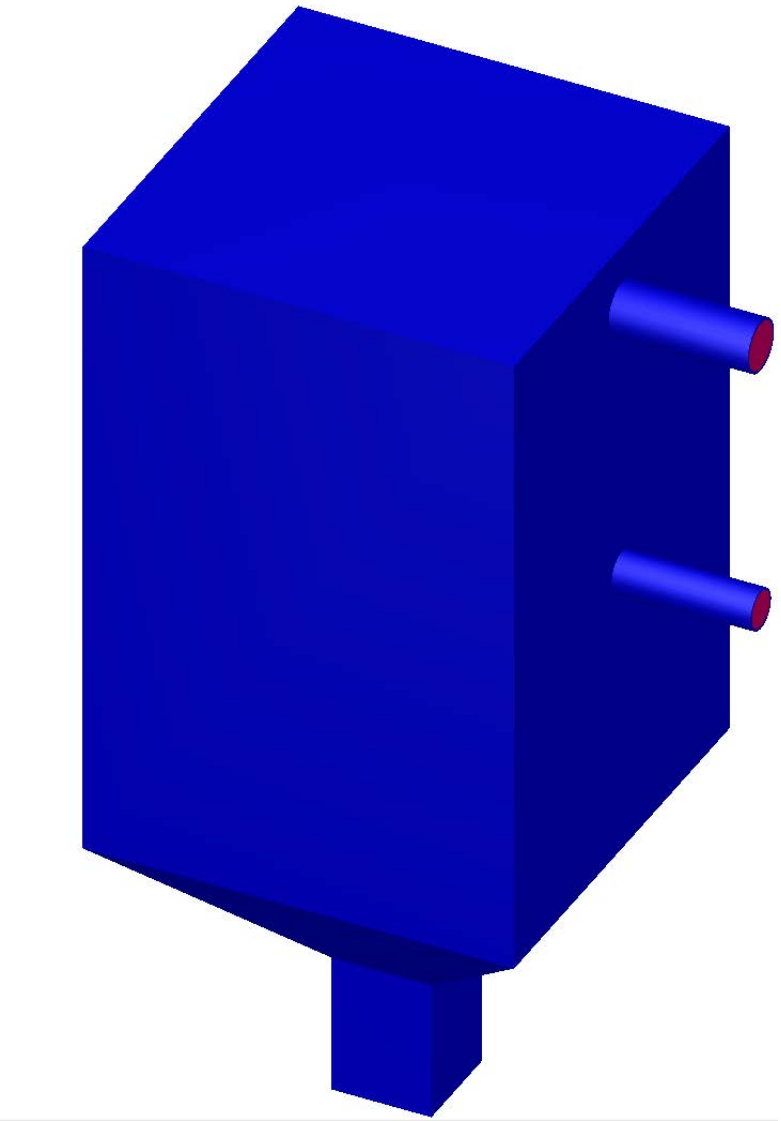


Hopper filled with UO_2 powder

Top section of hopper
SS304
Inside dimensions:
91.44 x 91.44
138.43-cm-tall
Top and **walls**: 0.635-cm-thick

Reducer section of hopper
SS304 – 0.635-cm-thick
Top inside : 91.44 x 91.44
Bottom inside: 20.32 x 20.32
Length: 35.8775

Bottom section of hopper
SS304 – 0.635-cm-thick
Inside: 20.32 x 20.32
30.48-cm-tall

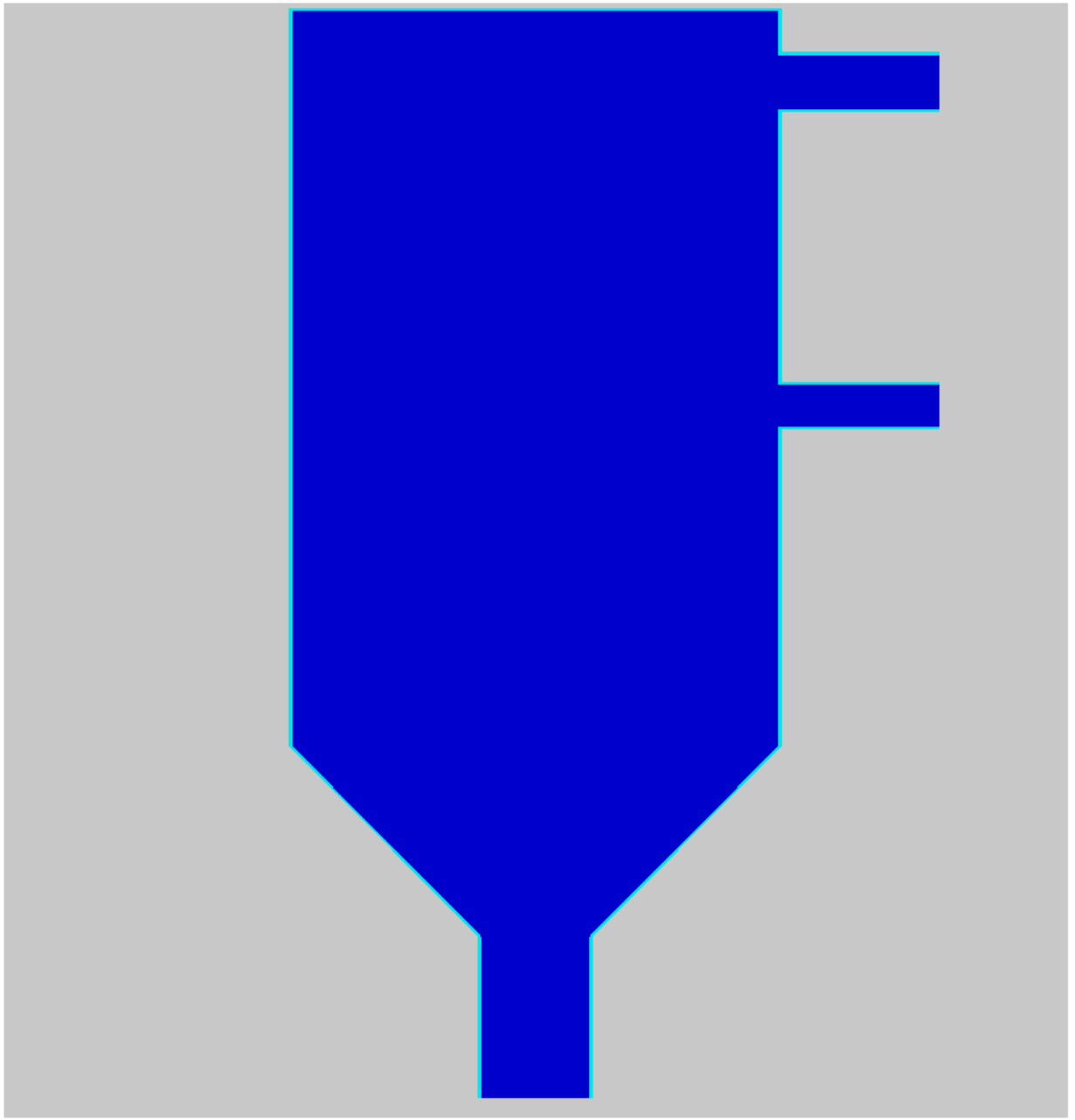


Top side pipe
SS304
Outer diameter: 11.43
Wall thickness: 0.60198
Center is 13.97 cm from the top of the top hopper section
Length: 30.48 cm

Bottom side pipe
SS304
Outer diameter: 8.89
Wall thickness: 0.54864
Center is 60.96 cm from center of top pipe
Length: 30.48 cm

UO_2 Powder
1.5 wt% U-235
98.5 wt% U-238
Density: 2.1 g/cc

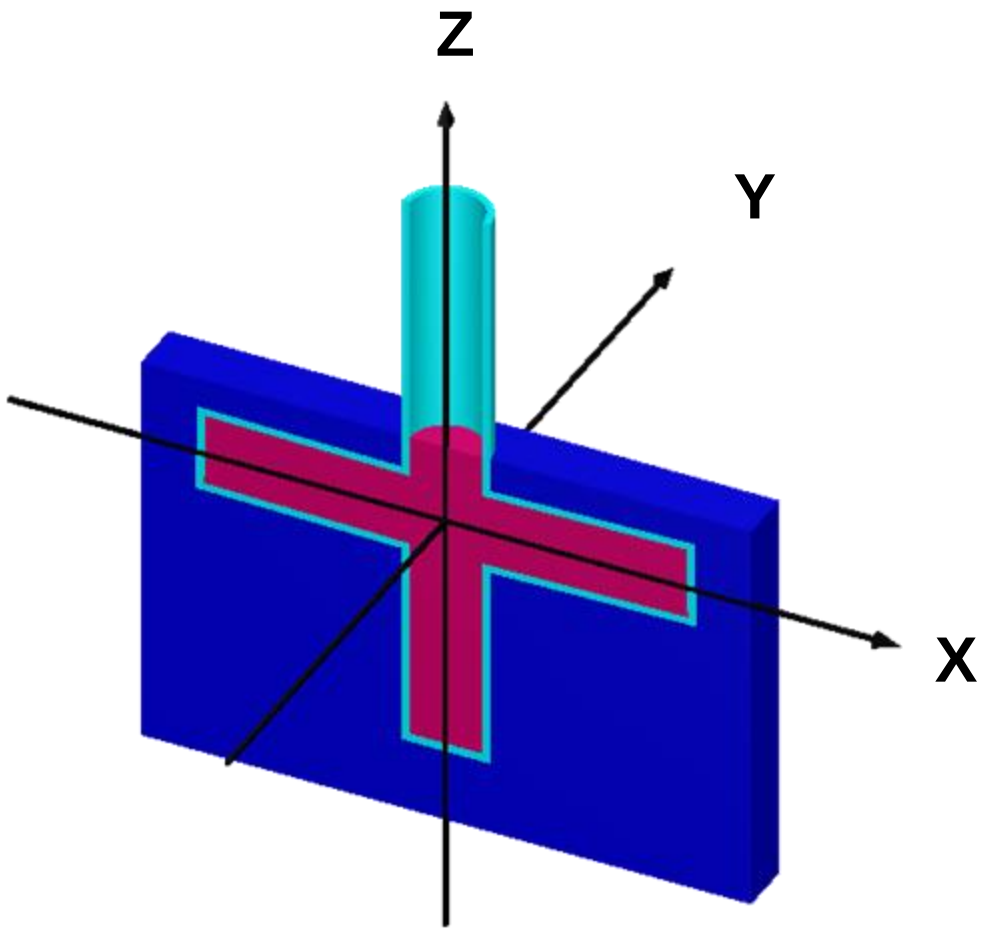
Side view



Notes:

- No internal obstructions to powder flow
- Steel around the outside except for the bottom surface and outer side of both pipes

Cross



UO_2F_2 solution
specific gravity 2.0150
896.1 g U/l
Enrichment:
U-235 5.0 %
U-238 95.0 %

Plexiglas cross
(Dimensions taken from center of cross intersection)
13.335 cm inside radius
3.17 cm thick ends
2.855 cm thick walls
2 side and bottom legs are 91.44 cm long from center to outer surface
Side legs at 89 degree angle from vertical
Top leg is 121.92 cm long
UO_2F_2 solution height 28.93 cm
Water reflector thickness
+X = 111.74 cm; -X = -111.74 cm
+Y = 20.64 cm; -Y = -20.63 cm
+Z = 29.03 cm; -Z = -118.428 cm

Calculated k_{eff} values

	KENO-VI	Shift
Hopper	0.16369 +/- 0.00051	0.163801 +/- 0.000596
Cross	1.0065 +/- 0.0018	1.006680 +/- 0.002980

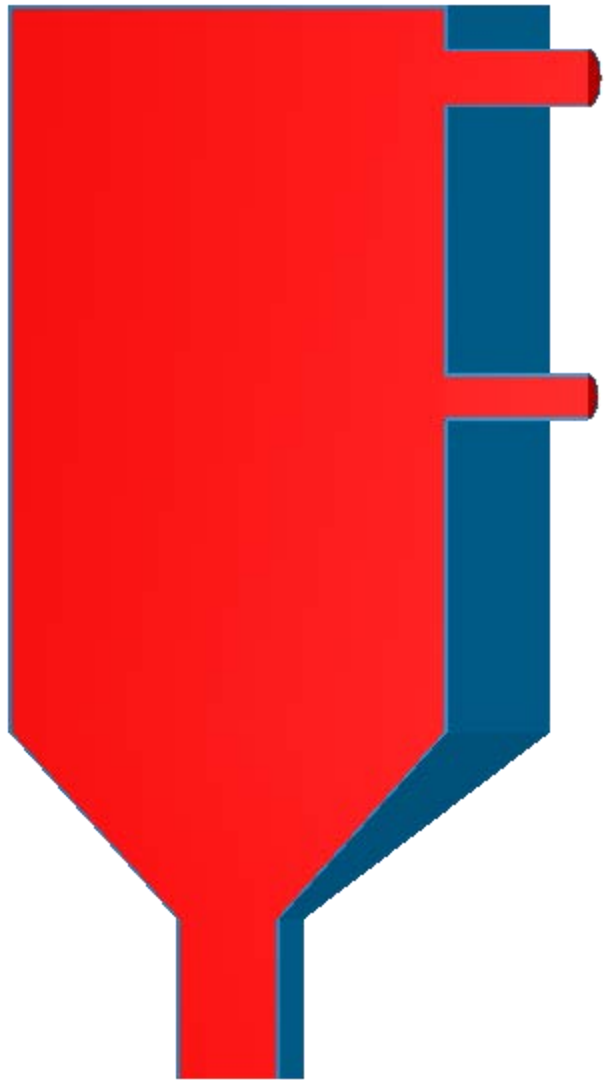
Your results may differ based on the options and computer used

Hopper with UO_2 powder

```
=csas6
hopper
ce_v7.1_endf
read composition
 uo2          1 den=2.1 1 293    92235 1.5    92238 98.5   end
 ss304        2 end
end composition
```

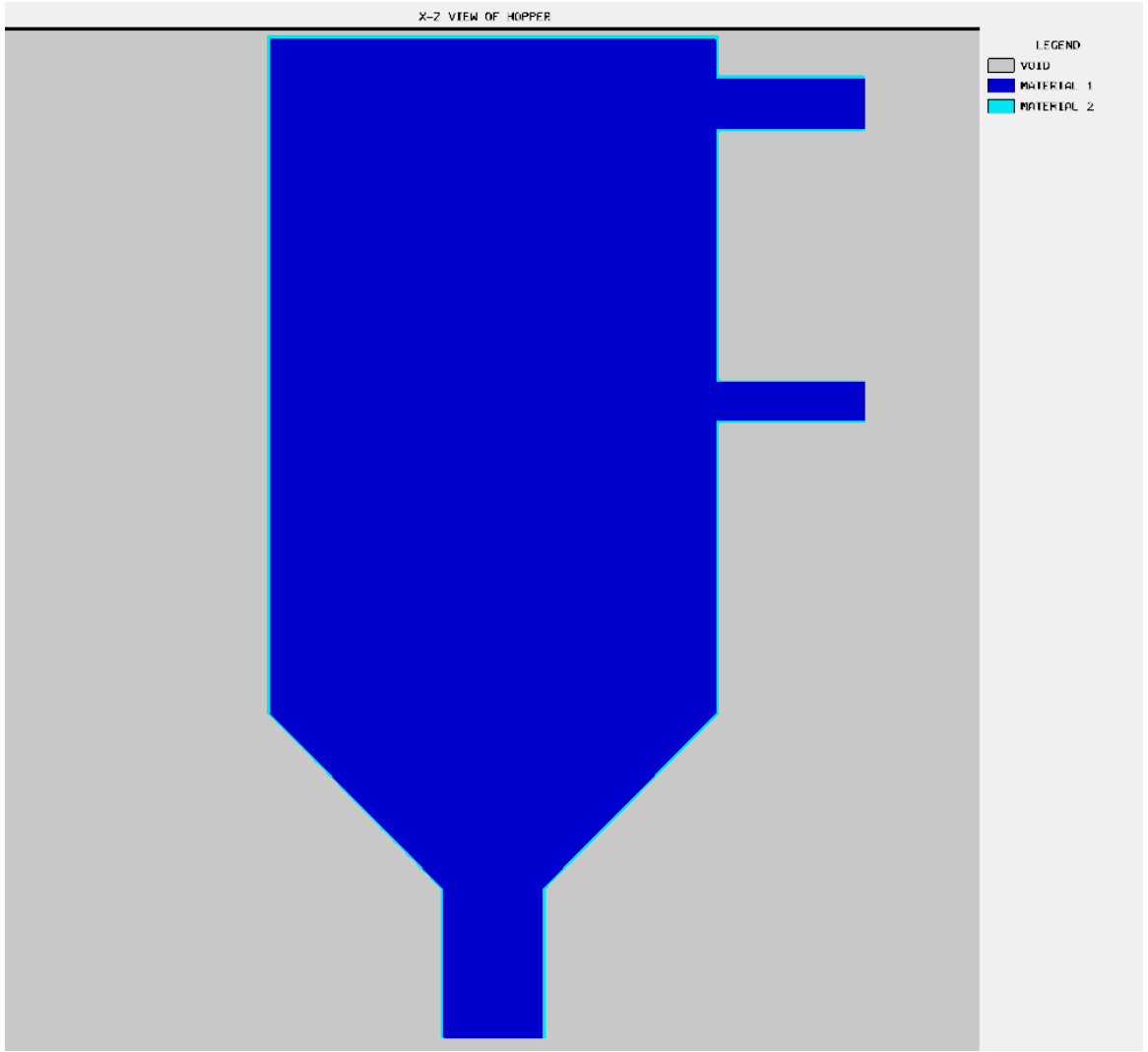
Hopper with UO_2 powder (cont.)

```
read geometry
global unit 1
com='hopper in 3 sections with 2 pipes'
 cuboid 1    45.72   -45.72    45.72   -45.72   138.43         0
 cuboid 2   46.355  -46.355   46.355  -46.355  139.065         0
 hopper 11    45.72    45.72         0    10.16    10.16  -35.8775
 hopper 12   46.355   46.355       0   10.795   10.795  -35.8775
 cuboid 21    10.16    -10.16    10.16   -10.16  -35.8775  -66.3575
 cuboid 22   10.795  -10.795  10.795  -10.795  -35.8775  -66.3575
 xcylinder 31    5.113     76.2    45.72   origin  z=125.095
 xcylinder 32    5.715     76.2    45.72   origin  z=125.095
 xcylinder 41    3.896     76.2    45.72   origin  z=64.135
 xcylinder 42    4.444     76.2    45.72   origin  z=64.135
 cuboid 50      100     -100      100     -100      140      -70
```



Hopper with UO_2 powder (cont.)

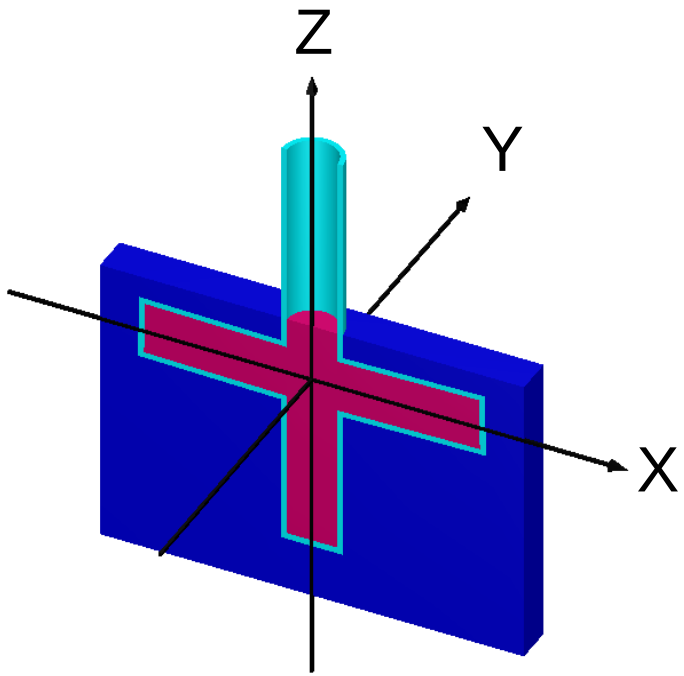
```
 media 1 1 1
 media 1 1 11
 media 1 1 21
 media 1 1 31
 media 1 1 41
 media 2 1 2 -1 -32 -42
 media 2 1 12 -11
 media 2 1 22 -21
 media 2 1 32 -31
 media 2 1 42 -41
 media 0 1 50 -2 -12 -22 -32 -42
 boundary 50
end geometry
end data
end
```



Cross

```
=csas6
kenovi 89 deg cross 5% uo2f2 soln, plexiglass pipes, h2o refl.
ce_v7.1_endf
read composition
 solution   mix=1  rho[uo2f2]=896.1   92235 5   92238 95
                  density=2.015
  end solution
 h2o          2 end
 plexiglass   3 end
end composition
```

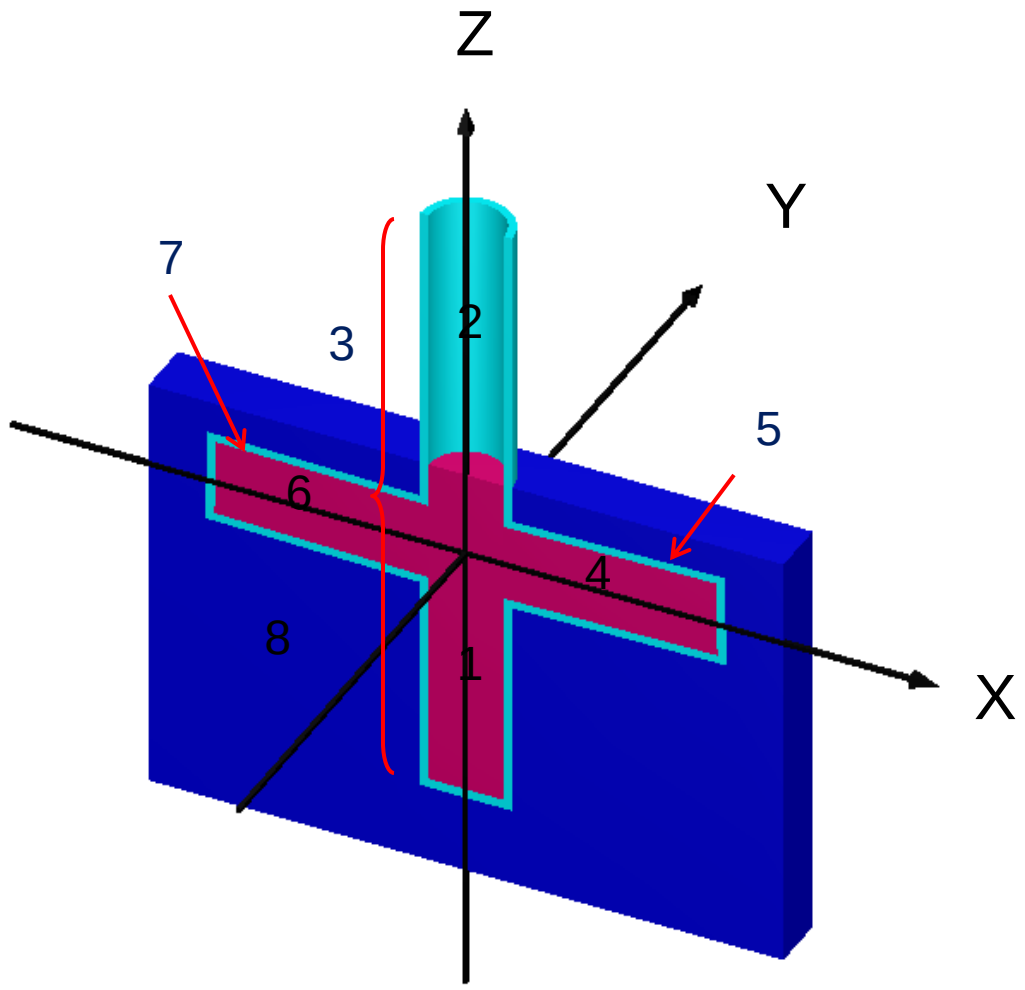
Cross (cont.)




```
read geometry
global unit 1
 cylinder 10   13.335    28.93   -88.27
 cylinder 20   13.335   121.92   -88.27
 cylinder 30   16.19    121.92   -91.44
 cylinder 40   13.335    88.27      0   rotate  a1=90  a2=89 a3=0
 cylinder 50   16.19     91.44      0   rotate  a1=90  a2=89 a3=0
 cylinder 60   13.335    88.27      0   rotate  a1=-90 a2=89 a3=0
 cylinder 70   16.19     91.44      0   rotate  a1=-90 a2=89 a3=0
 cuboid 80   111.74  -111.74    20.64   -20.64    29.03  -118.428
 cuboid 90   111.74  -111.74    20.64   -20.64   121.92  -118.428
```

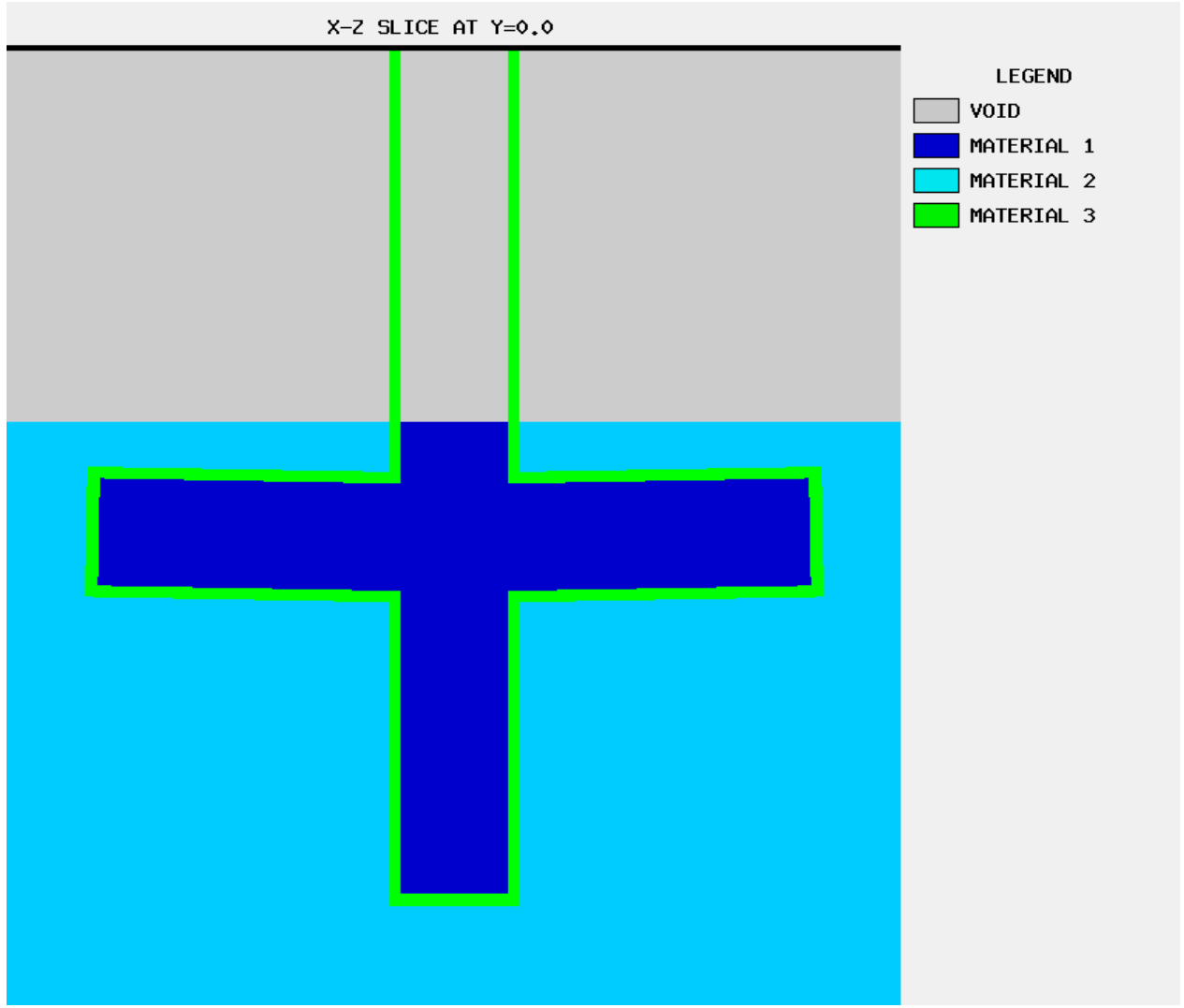
Cross (cont.)

```
  media 1 1 10
  media 0 1 20 -10
  media 3 1 30 -20 -50 -70
  media 1 1 40 -20
  media 3 1 50 -40 -20
  media 1 1 60 -20
  media 3 1 70 -60 -20 -50
  media 2 1 80 -30 -50 -70
  media 0 1 90 -30 -80
  boundary 90
end geometry
```



Cross (cont.)

```
read plot
 ttl=' x-z slice at y=0.0 '
 xul=-111  yul=0  zul=121
 xlr=111     ylr=0  zlr=-118
 nax=640  typ=xz   end
end plot
end data
end
```



Summary of Module 2

- Overview of modification options and syntax
- Geometry modifications are always performed sequentially; first "chord", then "rotate", and then "origin“
- Y30 example, Hopper and Cross exercises

OAK RIDGE National Laboratory

Sub-module 3

- HOLEs in KENO-VI



Contents record types

- Specifies the contents of a volume defined relative to the Geometry records in that unit
- Three types of contents records
  1. Media: places a mixture in the specified volume
  2. Hole: places a previously defined unit at the specified position
  3. Array: places an array in the specified volume

Contents records

Keyword	Contents Record Input
MEDIA	Mixture Number Bias ID Number Region Definition Vector Discussed previously in the geometry lecture
HOLE	Unit number Keywords ORIGIN and ROTATE, if desired
ARRAY	Array Number Region Definition Vector Will be discussed in the next lecture Keyword PLACE followed by the X, Y, Z position of the unit used to place the array and the X, Y, Z coordinates of the unit with respect to the origin of the outer region

HOLE contents record

- Used to place a unit into a different unit

- Consists of the following components:
  - HOLE Keyword
  - Unit ID number placed in HOLE
  - Modification data (ORIGIN, ROTATE) used to place the origin of the unit being inserted to the desired position and orientation

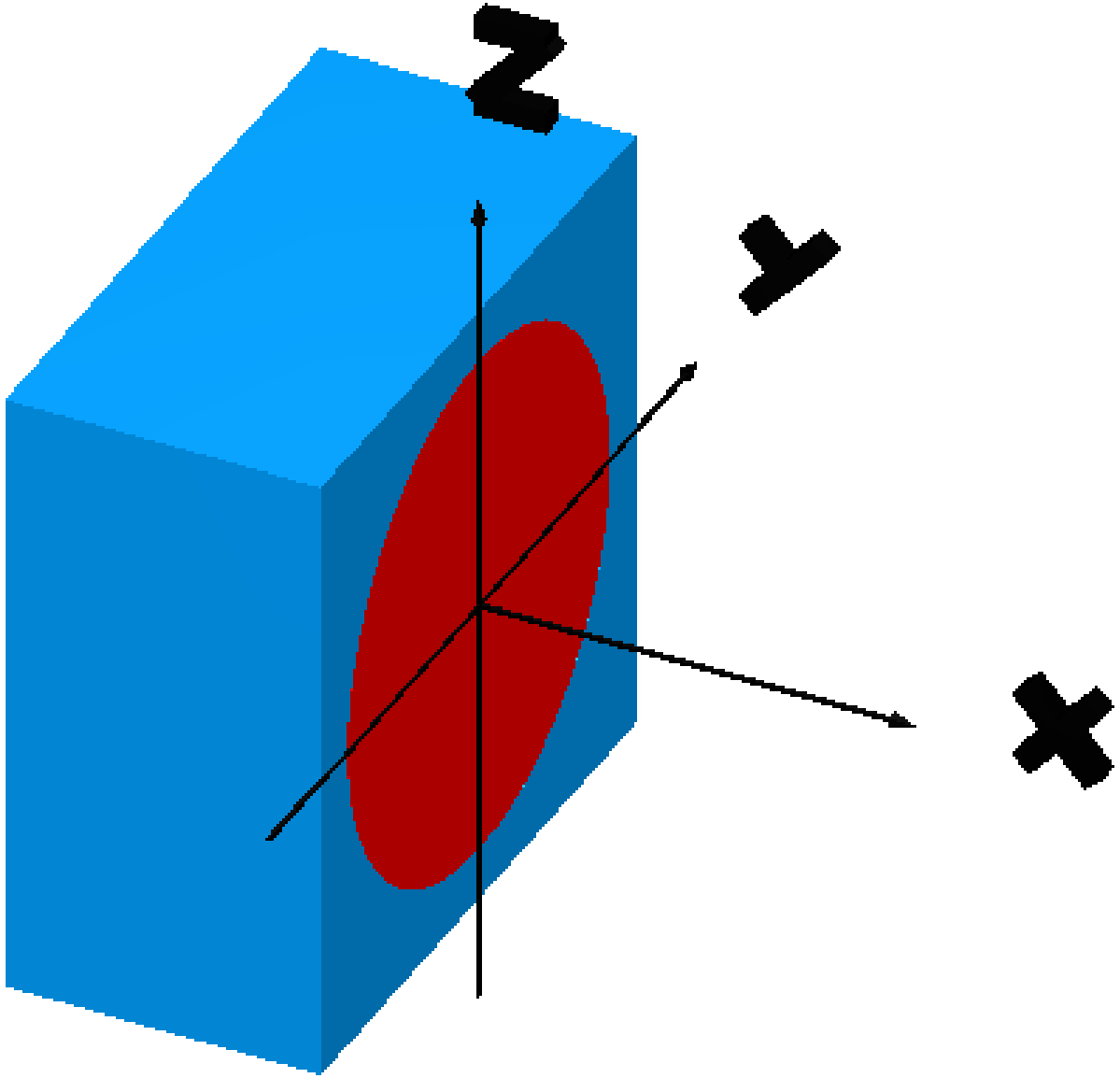
Advantages and disadvantages of HOLEs

- Advantages
  - Allows for better modularization of input
  - Can simplify region definition vectors for units
  - Can decrease run time
- Disadvantages
  - Arrays are a better option for repeated structures
  - Depending on viewpoint, adds complexity to input
    - More units, but each unit is simpler

Restrictions on holes

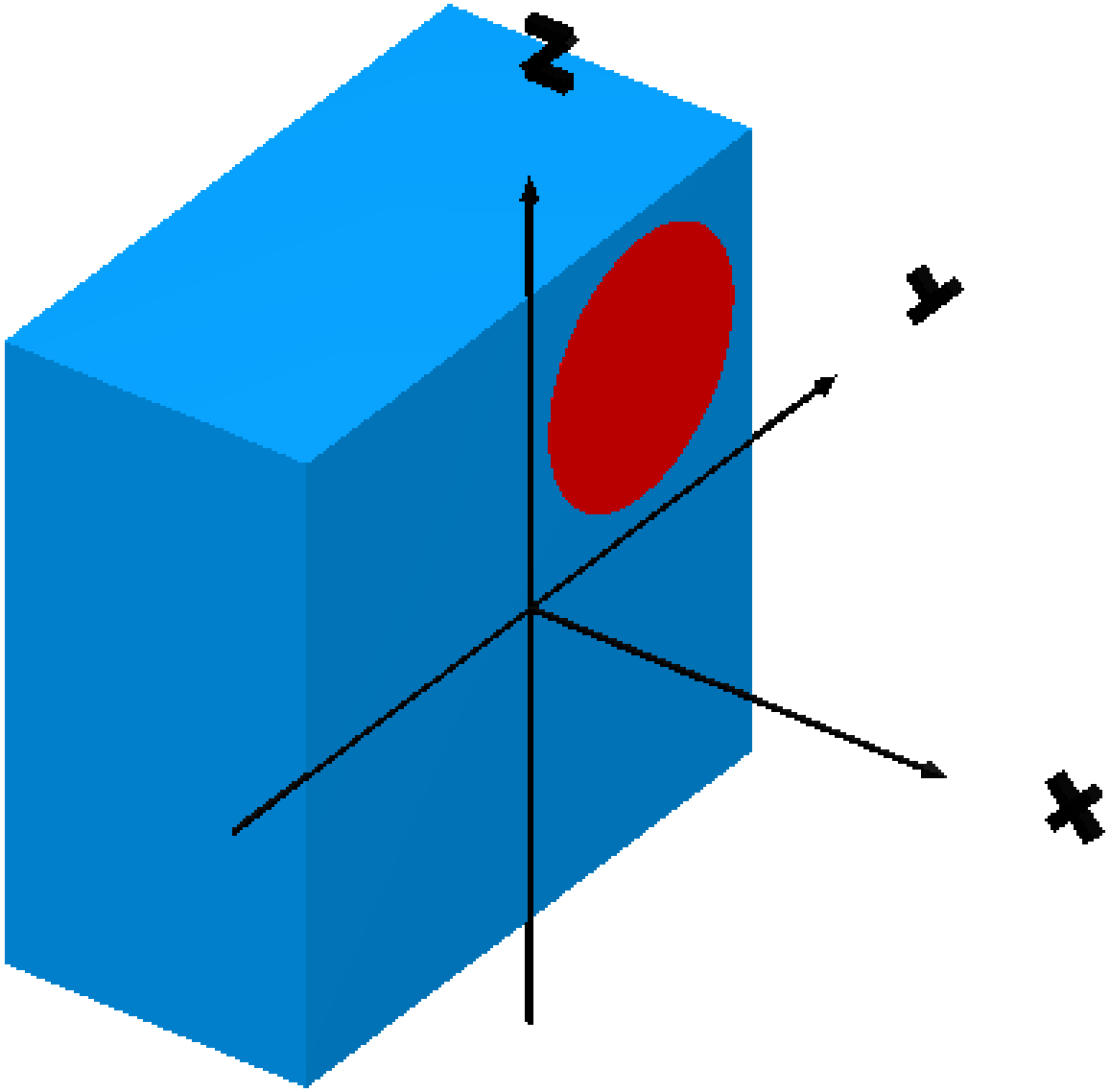
- Holes cannot intersect other holes
  - Two holes may share boundaries
- Holes cannot intersect the boundary of the unit into which they are placed
- Holes may not be truncated by an array boundary
  - We'll come back to this in the next lecture on arrays…
- All geometry labels on the boundary record of the unit being inserted as the hole must be positive
  - The hole must be *inside* the hole boundary
  - This is the reason for the RING shape

Simple HOLE example



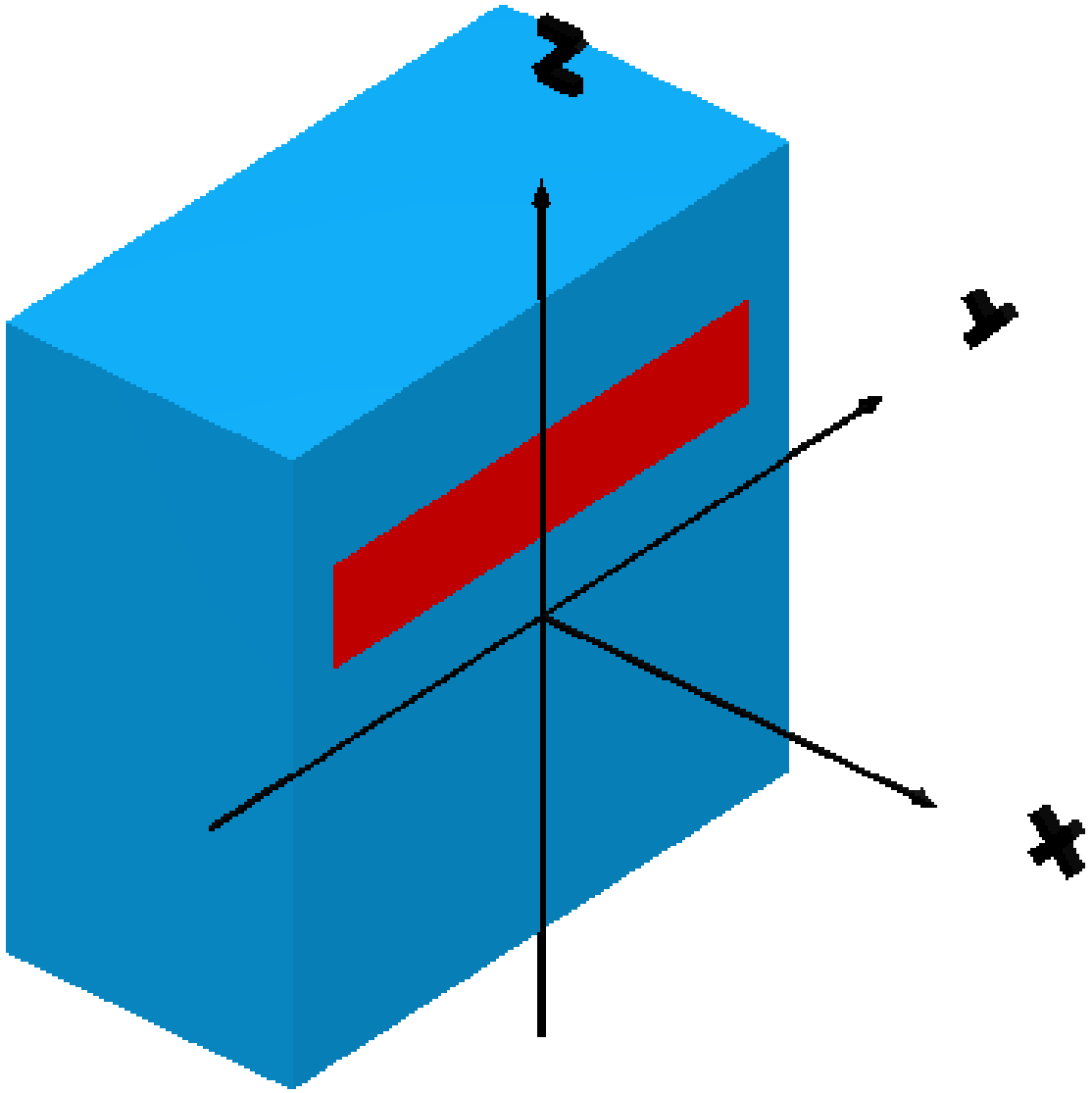

```
read geometry
unit 1
  sphere      10  10.0
  media        1  1  10
boundary  10
`
global unit 2
  cuboid  10  6p12.0
  media    2  1  10
  hole       1
boundary 10
end geometry
```

HOLE example with origin



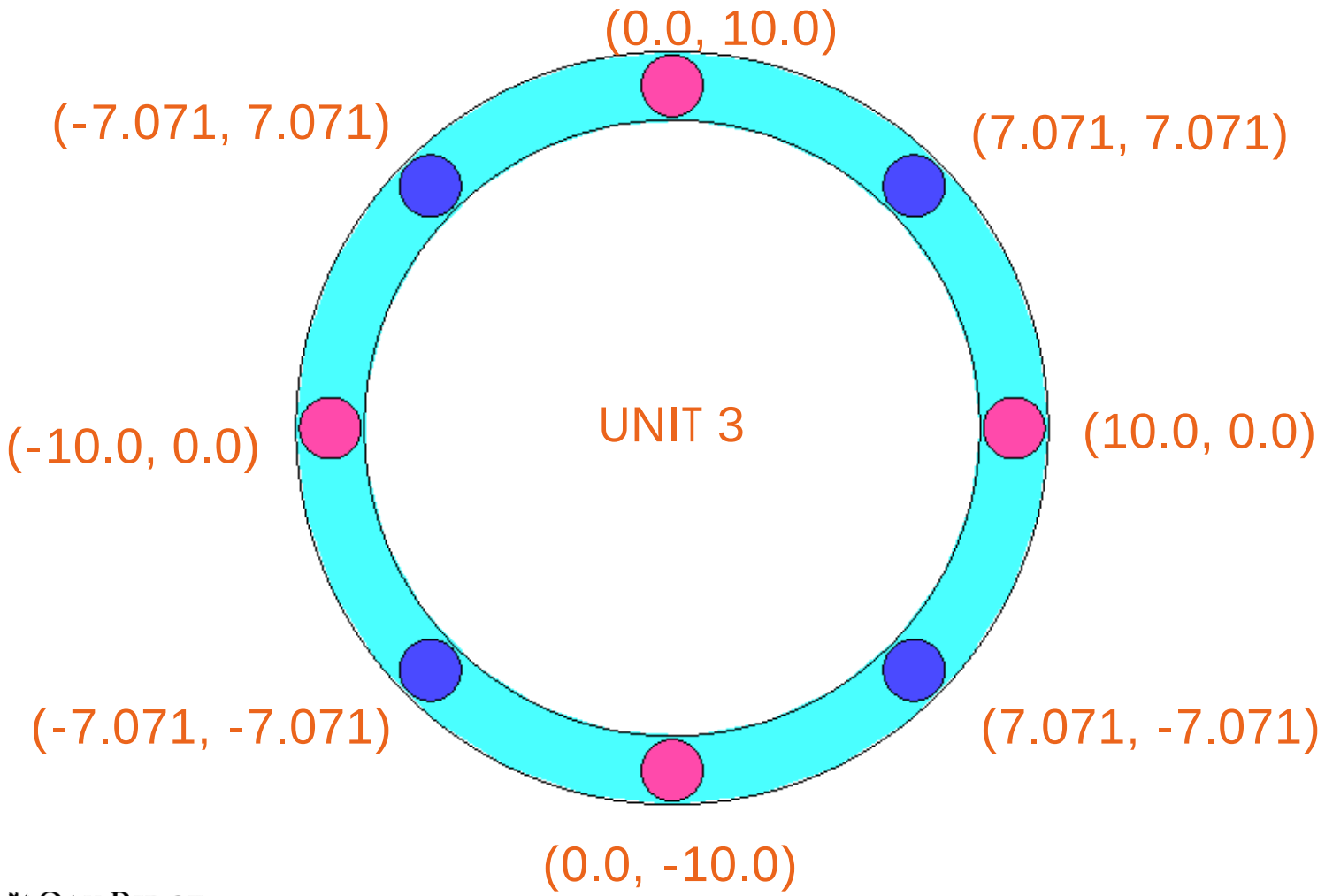

```
read geometry
unit 1
  sphere  10  5.0
  media    1  1  10
boundary  10
`
global unit 2
  cuboid  10  6p12.0
  media    2  1  10
  hole  1  origin y=6.0  z=6.0
boundary 10
end geometry
```

HOLE example with rotation




```
read geometry
unit 1
  cylinder  10  2.0  2p10.0
  media     1  1  10
boundary  10
`
global unit 2
  cuboid  10  6p12.0
  media    2  1  10
  hole  1  origin  z=5.0 rotate a2=90
boundary 10
end geometry
```

HOLE placement

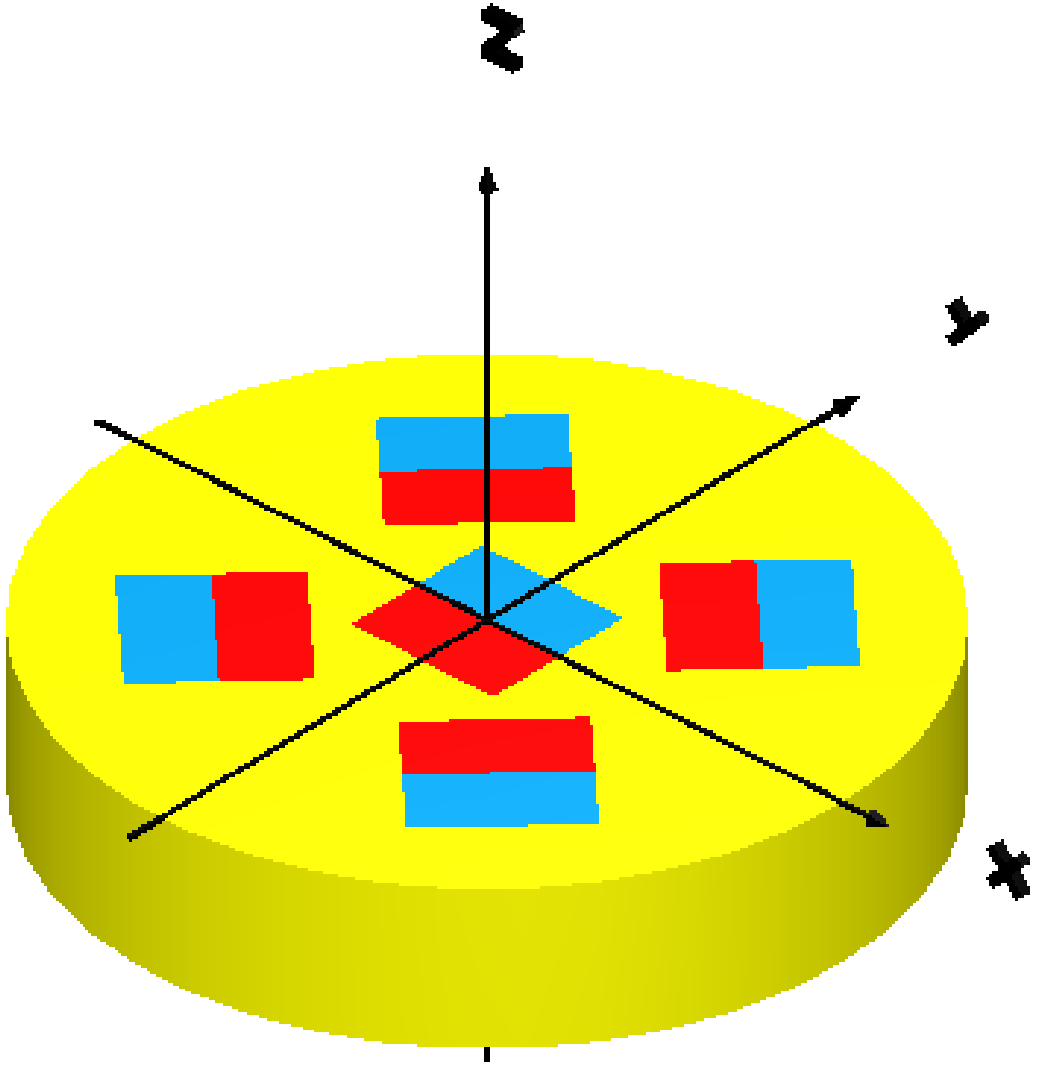



```
unit 1
  cylinder 10  0.9 20.0 0.0
  media  1  1  10
boundary  10

unit 2
  cylinder 10  0.9 20.0 0.0
  media  2  1  10
boundary  10

global unit 3
  cylinder  10  9.0  20.0  0.0
  cylinder  20 11.0  20.0  0.0
  media  0  1  10
  media  3  1  20 -10
  hole 1 origin x=-10.0  y=0.0
  hole 2 origin x=-7.071  y=7.071
  hole 1 origin x=0.0  y=10.0
  hole 2 origin x=7.071  y=7.071
  hole 1 origin x=10.0  y=0.0
  hole 2 origin x=7.071  y=-7.071
  hole 1 origin x=0.0  y=-10.0
  hole 2 origin x=-7.071  y=-7.071
boundary  20
```

HOLEs with modification data




```
read geometry
unit 1
  cuboid 10 2p6.0 0.0 -6.0 2p10.0
  cuboid 20 2p6.0 6.0 -6.0 2p10.0
  media  1  1  10
  media  2  1  20 -10
boundary  20
global unit 2
  cylinder  10  30.0  2p12.0
  media  3  1  10
  hole 1
  hole 1 origin x=12.0  y=12.0 rotate a1=-45
  hole 1 origin x=12.0  y=-12.0 rotate a1=-135
  hole 1 origin x=-12.0 y=-12.0 rotate a1=135
  hole 1 origin x=-12.0 y=12.0  rotate a1=45
boundary 10
end geometry
```

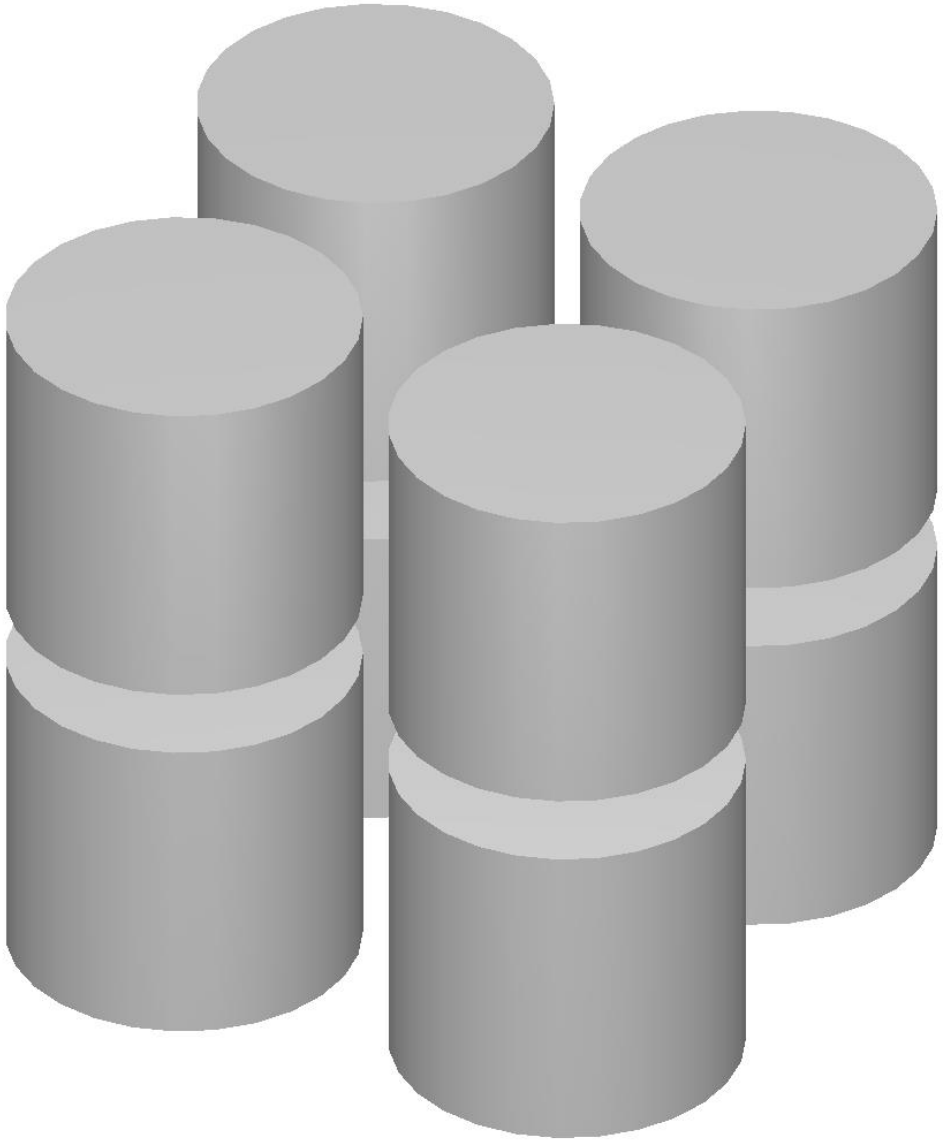
OAK RIDGE National Laboratory

Demo: HOLEs

- 2c8 – Take 1



2c8



8 Uranium Cylinders

5.748 cm radius

10.765 cm length

13.74 cm center to center spacing in X and Y dimensions

13.01 cm center to center spacing in Z dimension

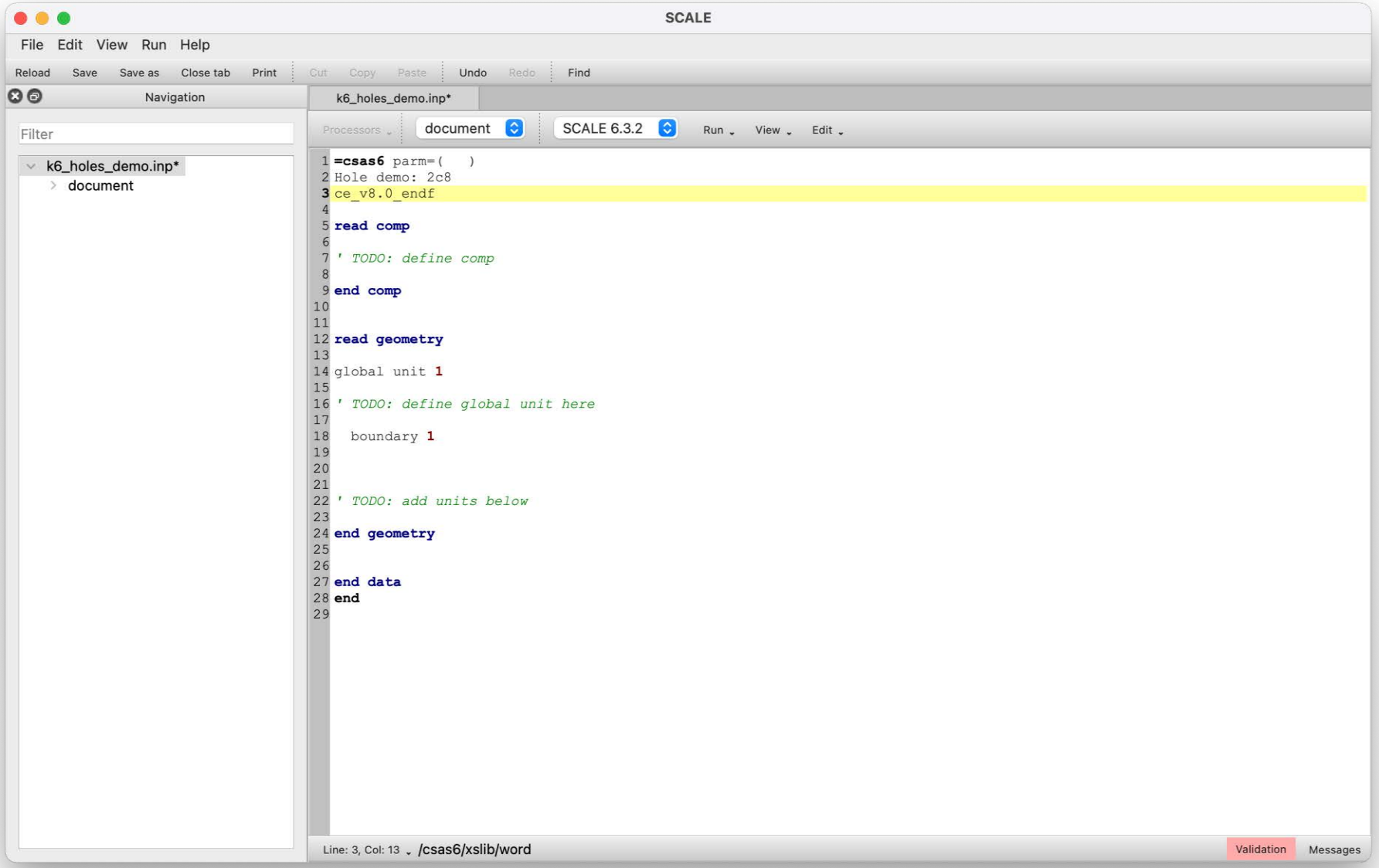
Uranium Metal

Density = 18.764 g / cc

Enrichment:

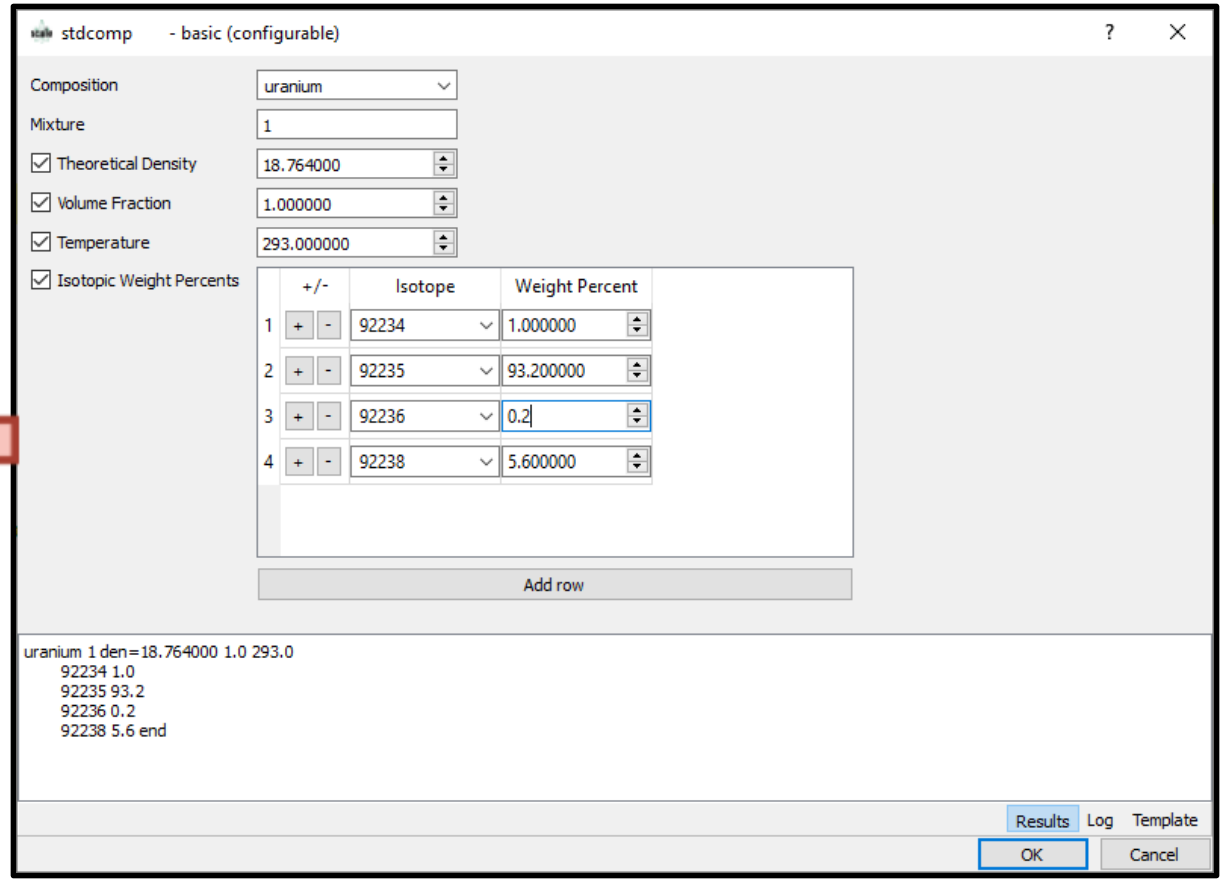
93.2%	U-235
5.6%	U-238
1.0%	U-234
0.2%	U-236

General problem data



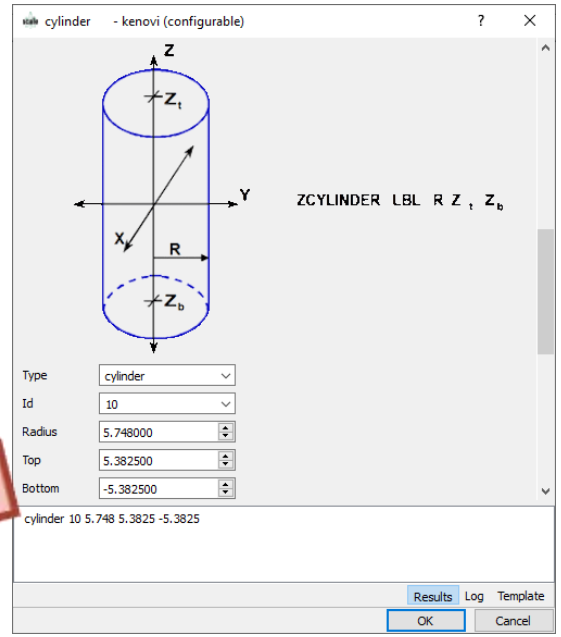
Composition data

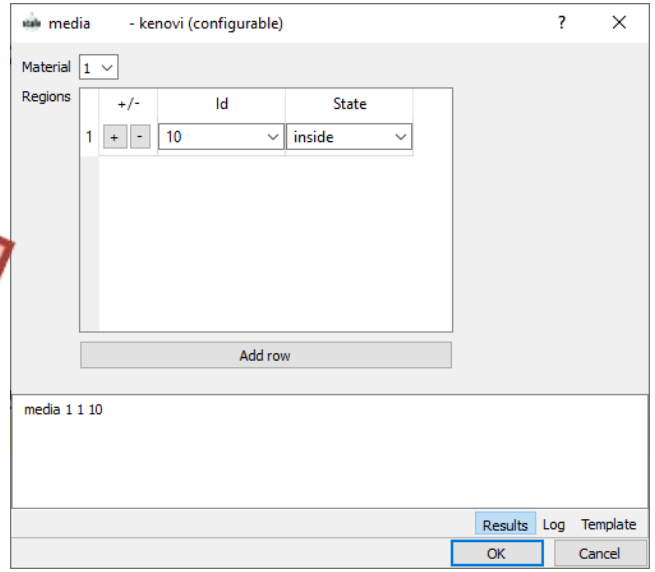
```
read comp
uranium 1 den=18.764000 1.0 293.0
        92234 1.0
        92235 93.2
        92236 0.2
        92238 5.6 end
end comp
```



Unit 1: Metal casting

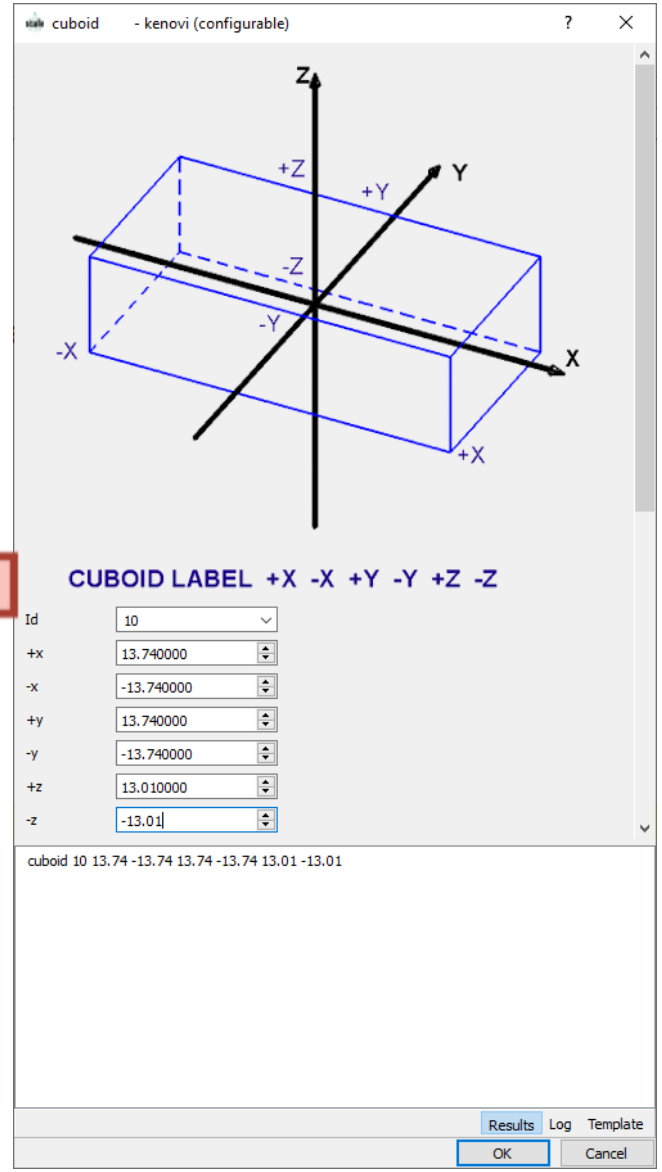
```
15 unit 1
16   cylinder 10 5.748 5.3825 -5.3825
17   media 1 1 10
18 boundary 10
```





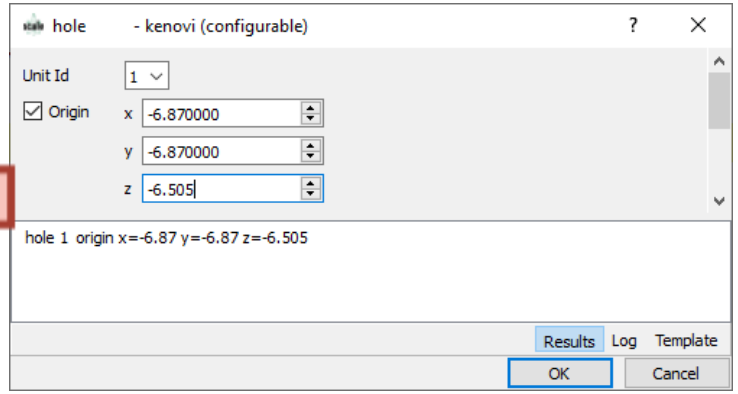
Global unit 2: Void space

```
19 global unit 2
20   cuboid 10 13.74 -13.74 13.74 -13.74 13.01 -13.01
21   media 0 1  10
22
23 boundary 10
```



Global unit 2: Add first hole

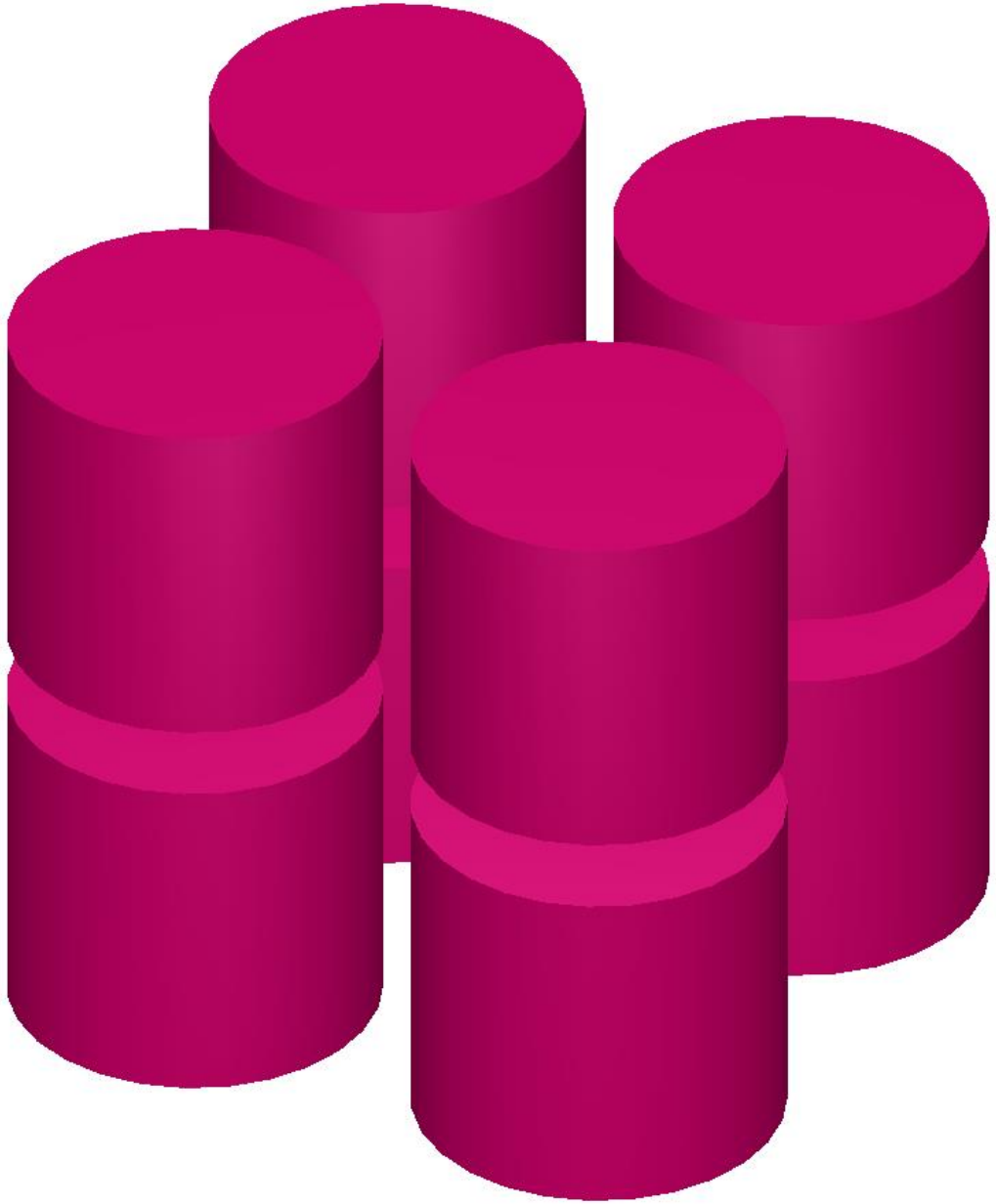
```
19 global unit 2
20   cuboid 10 13.74 -13.74 13.74 -13.74 13.01 -13.01
21   media 0 1  10
22   hole 1  origin x=-6.87 y=-6.87 z=-6.505
23
24 boundary 10
```



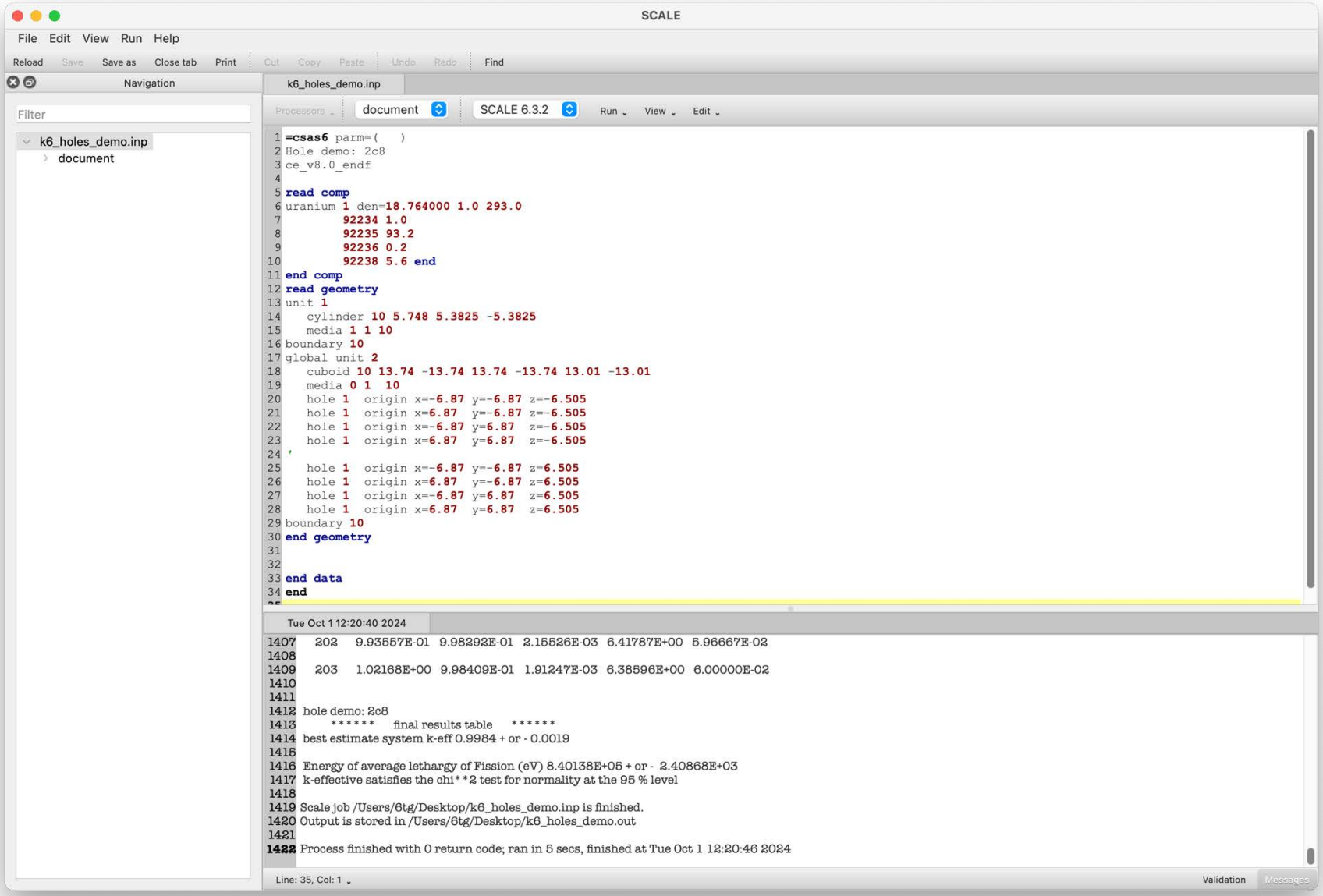
Global unit 2: Add remaining holes

```
19 global unit 2
20   cuboid 10 13.74 -13.74 13.74 -13.74 13.01 -13.01
21   media 0 1  10
22   hole 1  origin x=-6.87 y=-6.87 z=-6.505
23   hole 1  origin x=6.87  y=-6.87 z=-6.505
24   hole 1  origin x=-6.87 y=6.87  z=-6.505
25   hole 1  origin x=6.87  y=6.87  z=-6.505
26 '
27   hole 1  origin x=-6.87 y=-6.87 z=6.505
28   hole 1  origin x=6.87  y=-6.87 z=6.505
29   hole 1  origin x=-6.87 y=6.87  z=6.505
30   hole 1  origin x=6.87  y=6.87  z=6.505
31 boundary 10
```

3D rendering



Click run:



k-eff: 0.9984 +/- 0.0019

Summary of Module 3

- Overview of KENO VI holes
- Can simplify geometry construction
- Consists of the following components:
  - HOLE Keyword
  - Unit ID number placed in HOLE
  - Modification data (ORIGIN, ROTATE) used to place the origin of the unit being inserted to the desired position and orientation

OAK RIDGE National Laboratory

Sub-module 4

- Arrays in KENO-VI



Contents record types

- Specifies the contents of a volume defined relative to the Geometry records in that unit
- Three types of Contents records
  1. Media – Places a mixture in the specified volume
  2. Hole – Places a previously defined unit at the specified position
  3. Array – Places an array in the specified volume

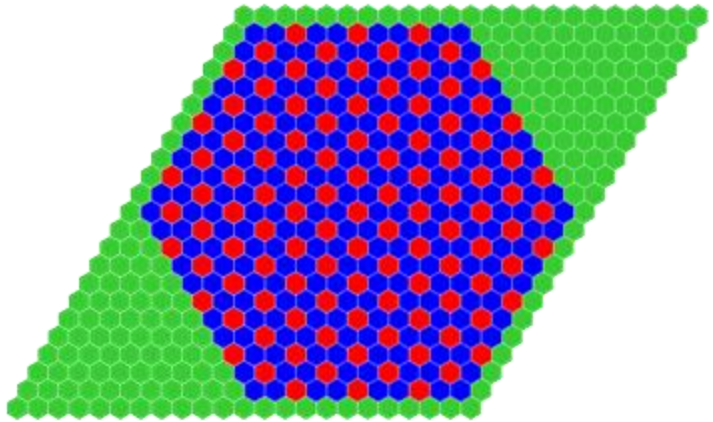
Contents records

Keyword	Contents Record Input
MEDIA	Mixture Number Bias ID Number Region Definition Vector
HOLE	Unit number Keywords ORIGIN and ROTATE, if desired
ARRAY	Array Number Region Definition Vector Keyword PLACE followed by the X, Y, Z position of the unit used to place the array and the X, Y, Z coordinates of the unit with respect to the origin of the outer region

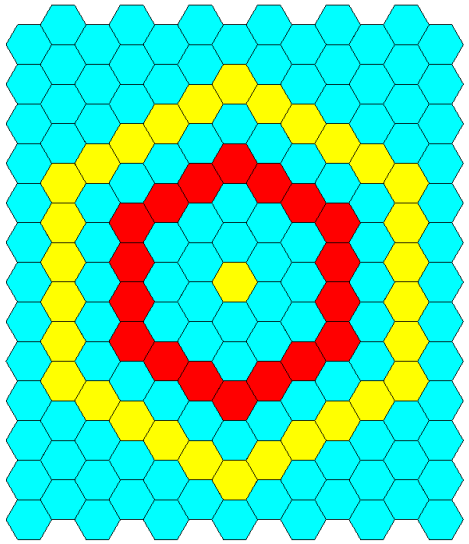
Discussed in previous geometry lecture

Discussed in previous holes lecture

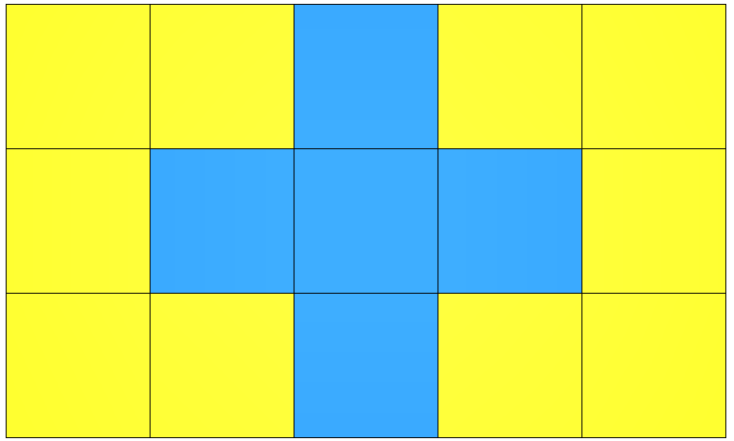
KENO-VI array types



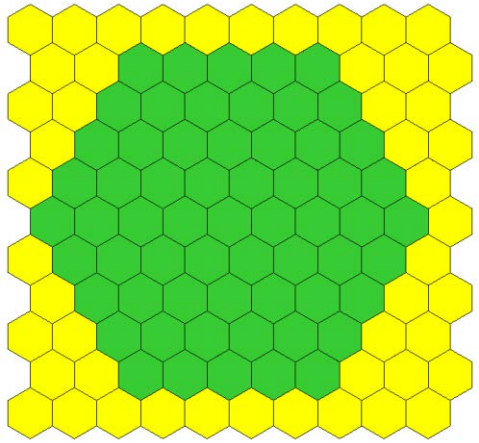
Hexagonal (or Triangular)



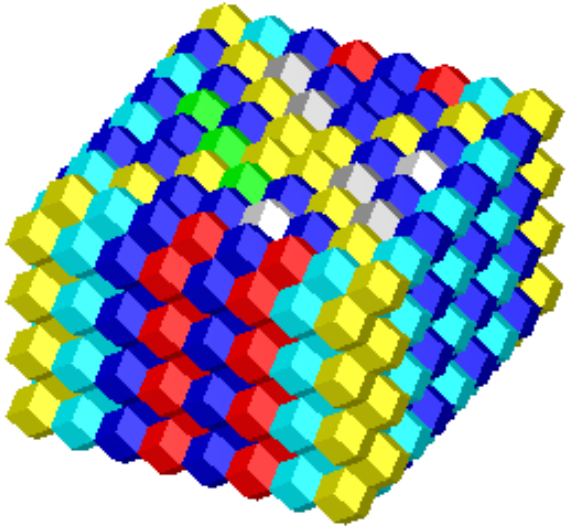
Rotated hex



Rectangular



Standard hex



Dodecahedral

Basic flow of using arrays in KENO-VI

1. Build the units you need to construct the array
2. Specify the array in the array block
   - All adjacent faces must have the same dimensions
3. Place the array within another unit in the problem geometry in the geometry block
   - Position origin of specified unit in the array at specific point in new unit

Array parameters

Keyword	Default	Type of Data
ARA=	1	Number defining the array
TYP=	Square	Square (cuboidal)
		Hexagonal (triangular)
		Shexagonal
		Rhexagonal
		Dodecahedral
COM=	none	*delimiter* comment *delimiter* 132 character maximum

Array parameters

Keyword	Default	Type of Data
NUX=	1	No. of units in the X direction
NUY=	1	No. of units in the Y direction
NUZ=	1	No. of units in the Z direction

FILL data

- Fills the array with the unit numbers in the specified order
- Begins with keyword “fill”
- Ends with keywords “end fill”
- Fills the array by incrementing
  - X from 1 to NUX
  - then Y from 1 to NUY
  - then Z from 1 to NUZ

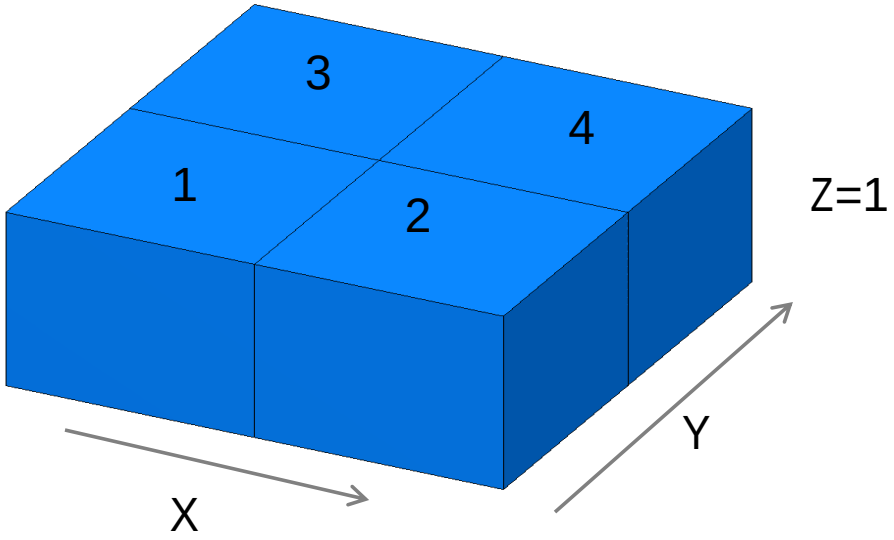
FILL data

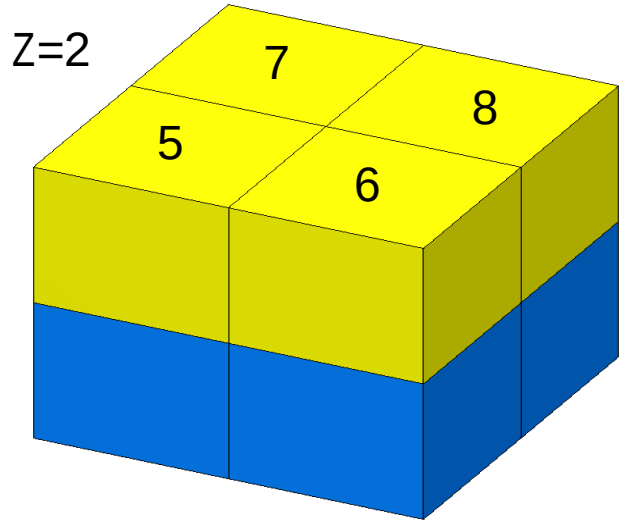
- For a 2 × 2 × 2 array, the order of the units stored in the array using the FILL is:

(1,1,1) (2,1,1) (1,2,1) (2,2,1)

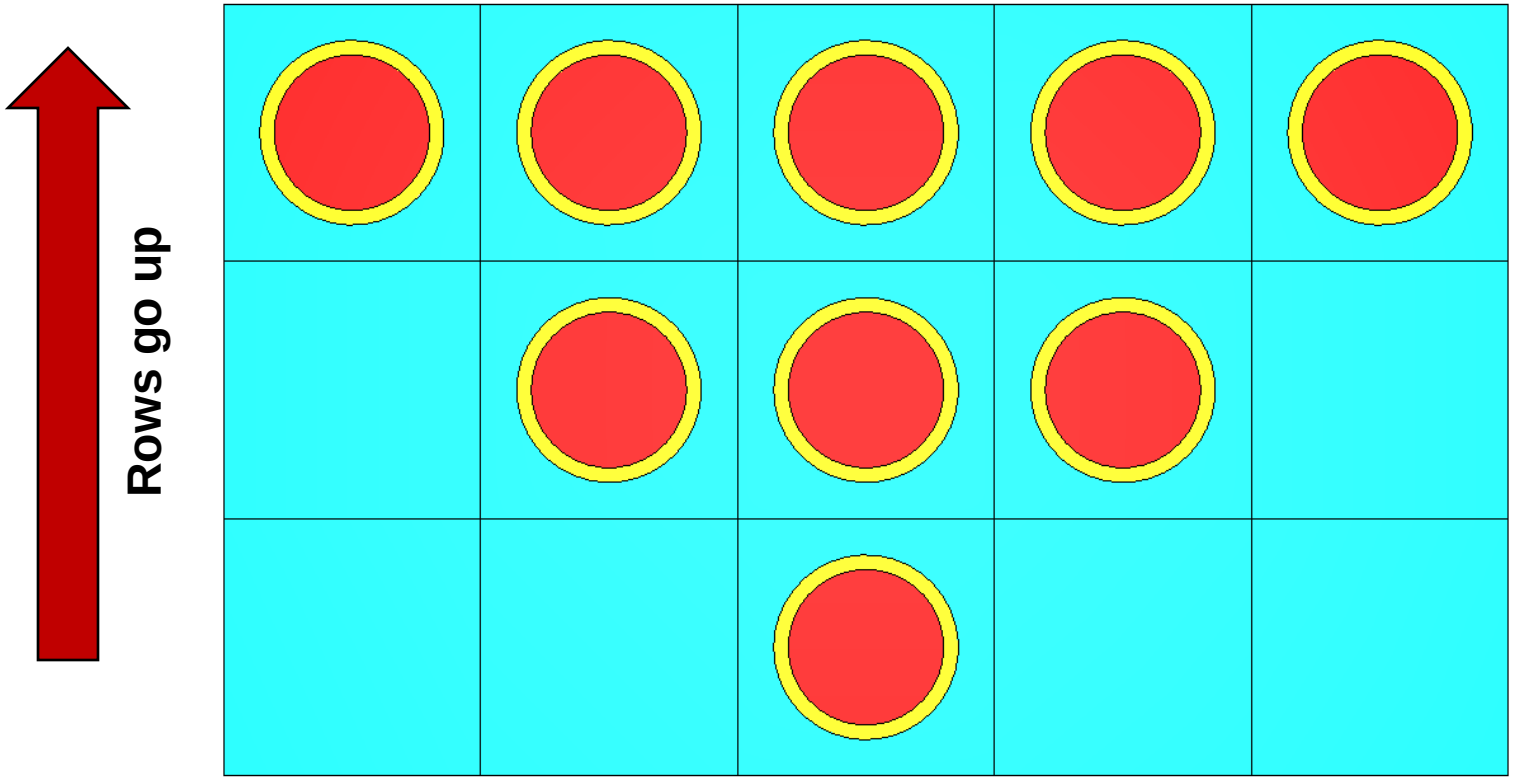
(1,1,2) (2,1,2) (1,2,2) (2,2,2)

```
fill
   1 1 1 1
   2 2 2 2
end fill
```

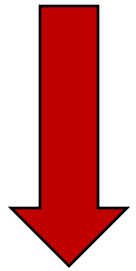




Rectangular array




```
read array
ara=1 nux=5 nuy=3
typ=square
  fill
   2  2  1  2  2
   2  1  1  1  2
   1  1  1  1  1
   end fill
end array
```



Rows go down

FIDO-like shortcuts for FILL

Count Field	Option Field	Operand Field	Comment
		j	Stores *j* in current array position
	f	j	Fills the remainder of the array with *j* starting with the current position
i	r or *	j	Repeats entry *j* a total of *i* times
i	q	j	Repeats the previous *j* entries *i* times

Refer to Table 8.1.19 for a complete list of options

Examples:

```
fill 5 5 5 5 5 end fill          is identical with  fill f5          end fill
fill 10 10 5 5 5 end fill        is identical with  fill 2r10 3r5    end fill
fill 1 2 3 1 2 3 1 2 3 end fill  is identical with  fill 1 2 3 2q3 end fill
```

ARRAY Contents record

- Used to place an array into a region defined in the current unit
- Consists of the following components:
  - ARRAY keyword
  - Array ID number
  - Region definition vector
  - Keyword PLACE followed by data to position the array in the region

Array PLACE data

- Data following the keyword PLACE used to position the array in a region consist of
  - Element i, j, k in the array used to establish the location of the array
  - the X, Y, Z coordinates of the origin of the unit in that array element, in the coordinate system of the unit into which the array is being placed

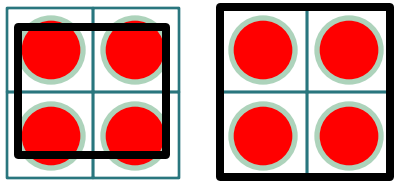
In other words:

PLACE the origin of the unit that is the ijkth element in the array at position (X, Y, Z) in the current unit coordinate system

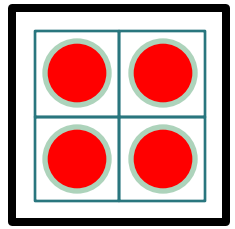
Restrictions on arrays

- Array boundaries cannot intersect holes
  - A hole entirely inside of an array is OK
- Array boundary MAY intersect boundary of lower-level units (elements) within array
- Array must AT LEAST fill the region into which it is PLACED
  - Region can be smaller than array
  - Region cannot be larger than array
    - Different treatment from TRITON/NEWT

Either of these is OK

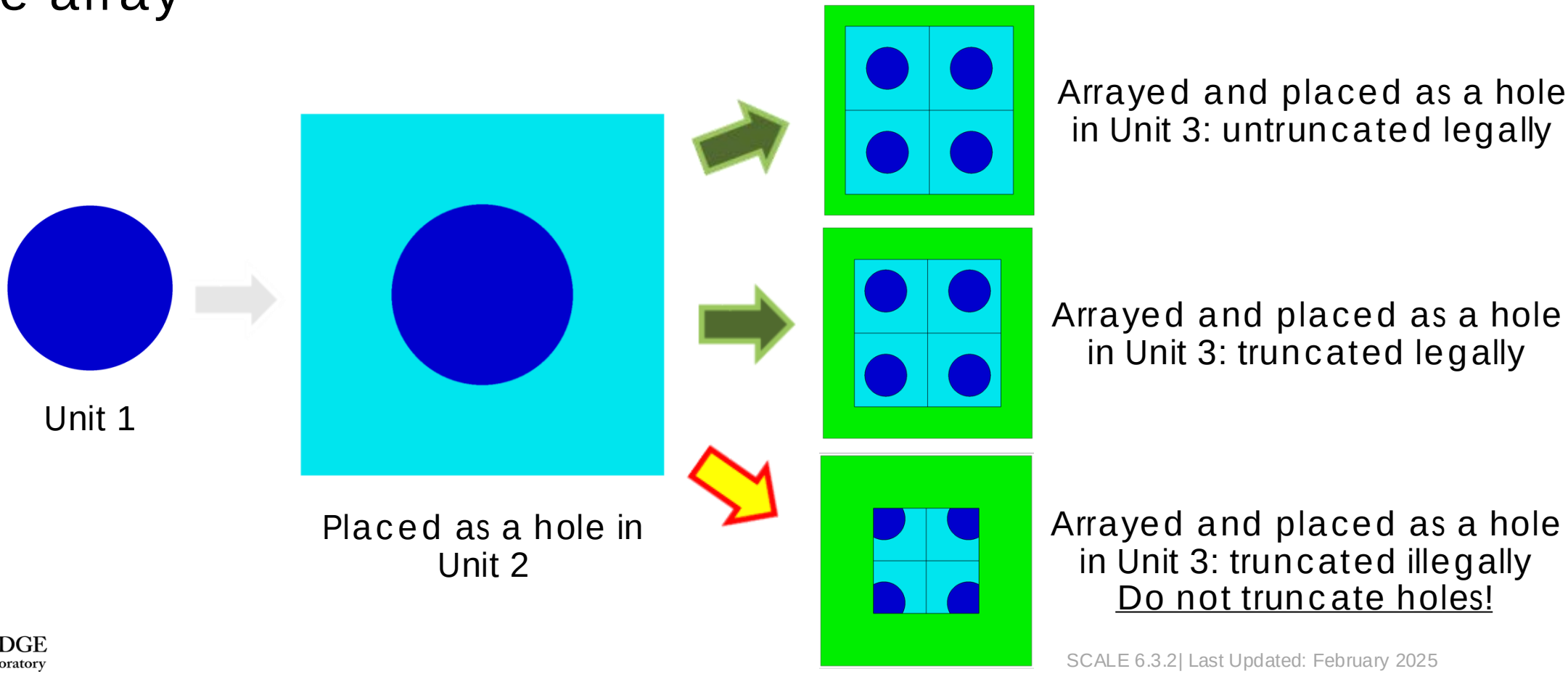


This is not

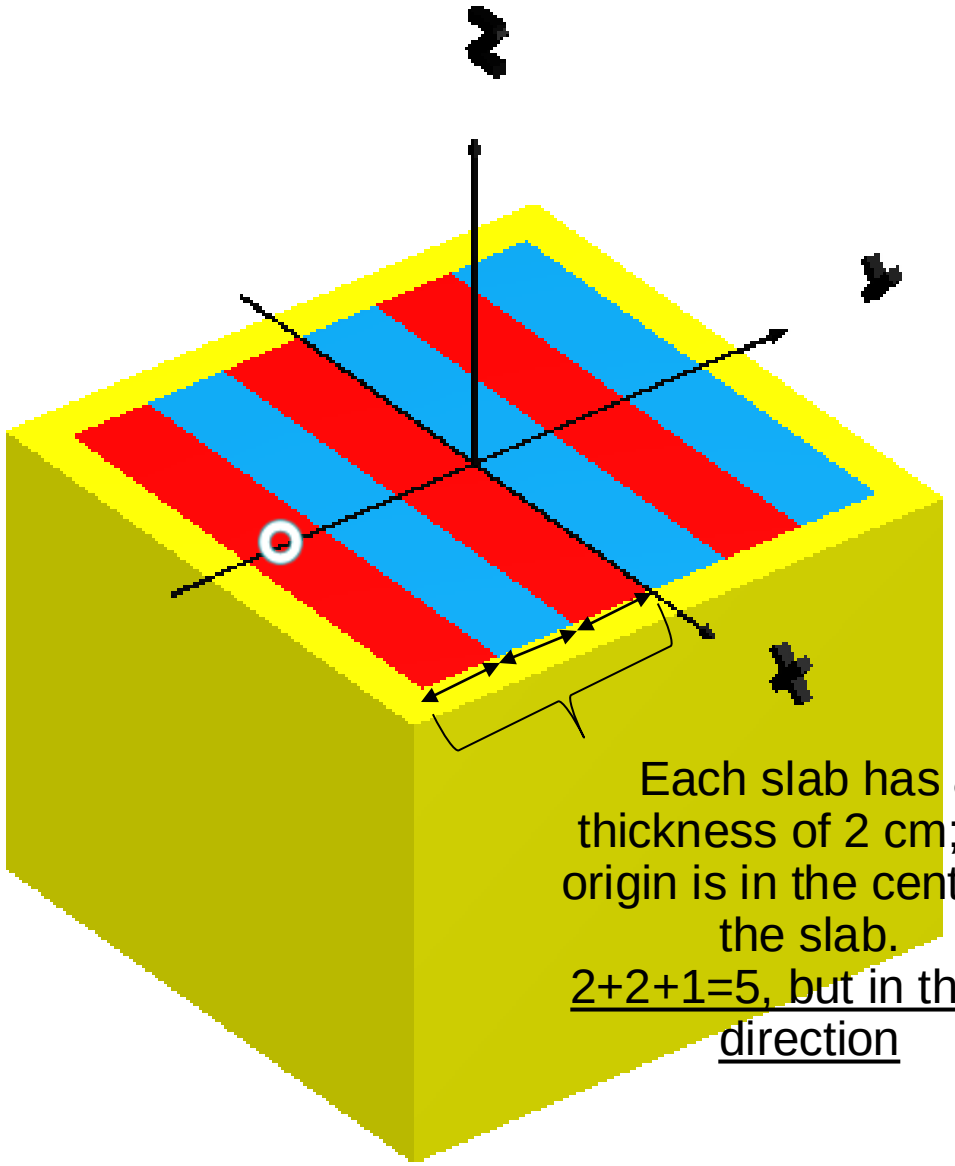


Restrictions on arrays related to holes

- Array boundary cannot truncate a hole contained in a unit in the array



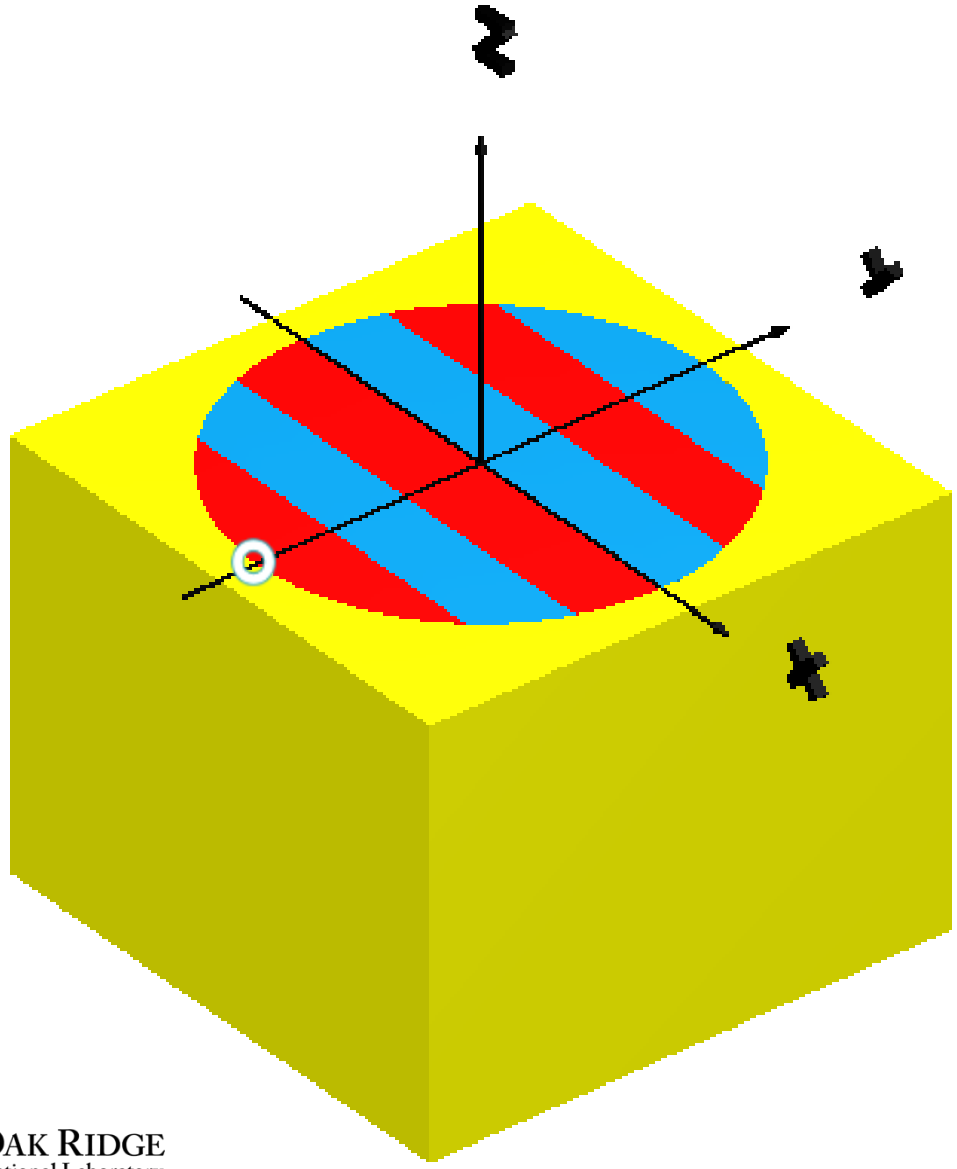
1D array



Each slab has a thickness of 2 cm; the origin is in the center of the slab.
2+2+1=5, but in the –y direction

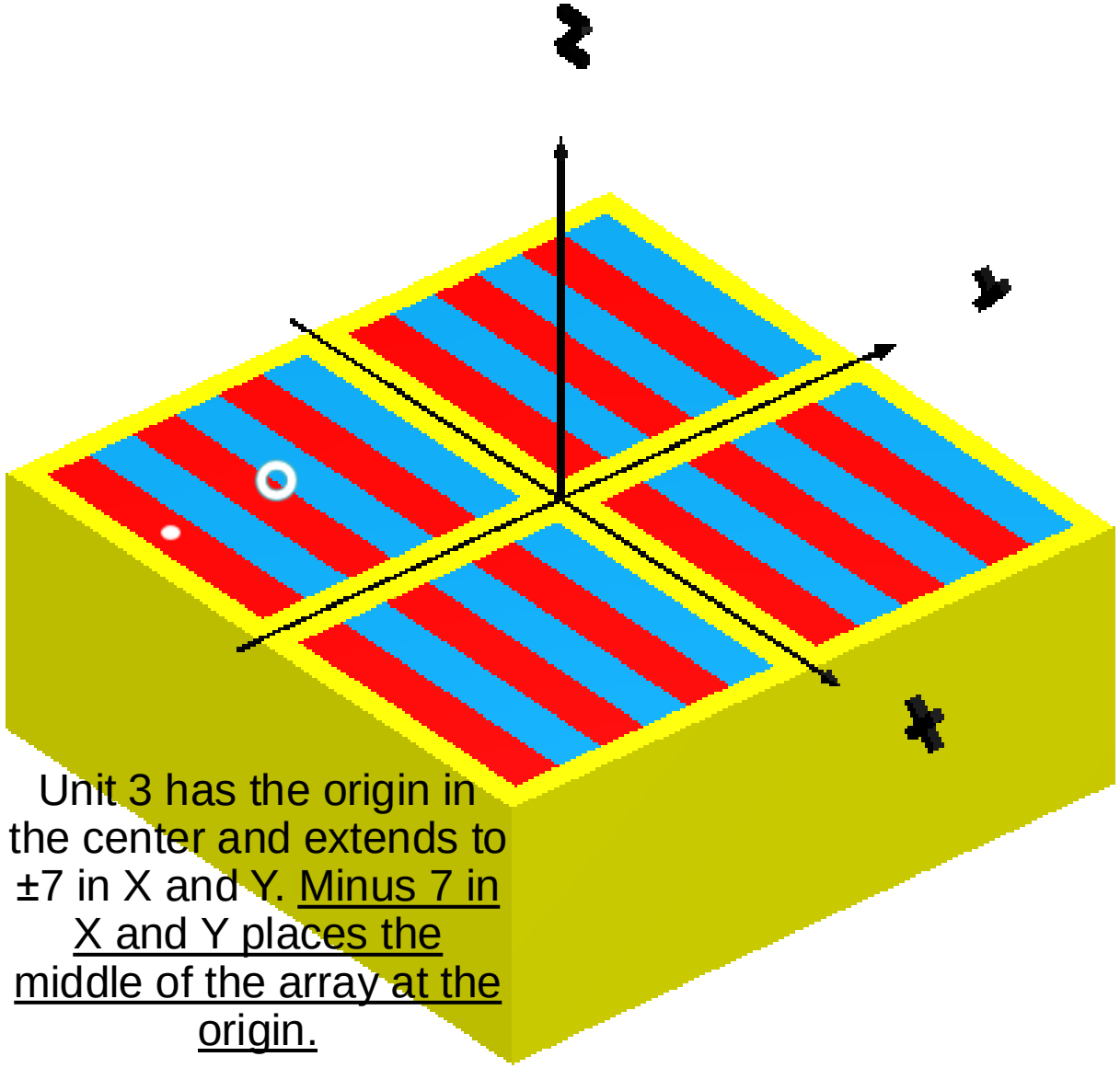
```
read geometry
unit 1
  cuboid 10 2p6.0  1.0 -1.0  2p10.0
  media  1  1  10
boundary  10
unit 2
  cuboid 10 2p6.0  1.0 -1.0  2p10.0
  media  2  1  10
boundary  10
global unit 3
  cuboid 10 4p6.0 2p10.0
  cuboid 20 4p7.0 2p11.0
  array 1 10 place 1 1 1 0.0 -5.0 0.0
  media 3 1 20 -10
boundary  20
end geometry
read array
ara=1 nux=1 nuy=6 nuz=1
  fill 1 2 1 2 1 2 end fill
end array
```

1D array with cylindrical boundary



```
read geometry
unit 1
  cuboid 10 2p6.0 2p1.0 2p10.0
  media  1  1  10
boundary  10
unit 2
  cuboid 10 2p6.0 2p1.0 2p10.0
  media  2  1  10
boundary  10
global unit 3
  cylinder 10 6.0 2p10.0
  cuboid 20 4p7.0 2p11.0
  array 1 10 place 1 1 1 0.0 -5.0 0.0
  media 3 1 20 -10
boundary  20
end geometry
read array
ara=1 nux=1 nuy=6 nuz=1
  fill 1 2 1 2 1 2 end fill
end array
```

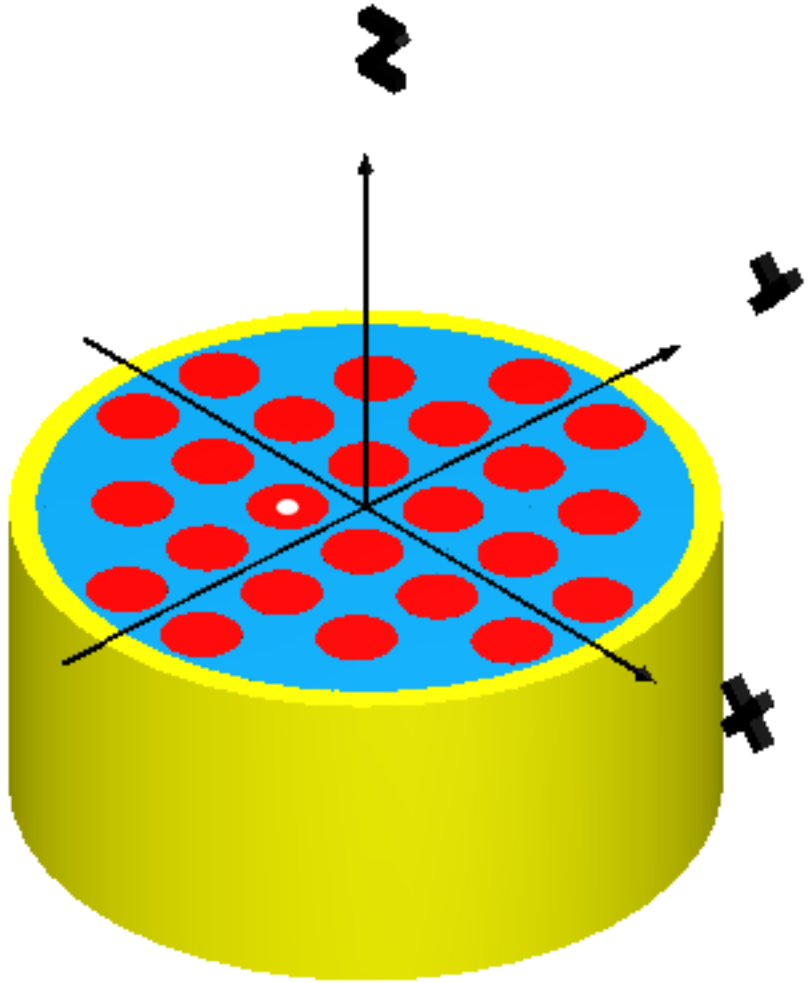
Array of arrays



Unit 3 has the origin in the center and extends to ±7 in X and Y. Minus 7 in X and Y places the middle of the array at the origin.

```
read geometry
unit 1  cuboid 10 2p6.0 2p1.0 2p10.0
  media  1  1  10  boundary  10
  unit 2  cuboid 10 2p6.0 2p1.0 2p10.0
  media  2  1  10  boundary  10
unit 3
  cuboid 10 4p6.0 2p10.0
  cuboid 20 4p7.0 2p11.0
  array 1 10 place 1 1 1 0.0 -5.0 0.0
  media 3 1 20 -10  boundary  20
global unit 4
  cuboid 10 4p14.0 2p11.0
  array 2 10 place 1 1 1 -7.0 -7.0 0.0
boundary 10
end geometry
read array
ara=1 nux=1 nuy=6 nuz=1
  fill 1 2 1 2 1 2 end fill
ara=2 nux=2 nuy=2 nuz=1
  fill f3 end fill  end array
```

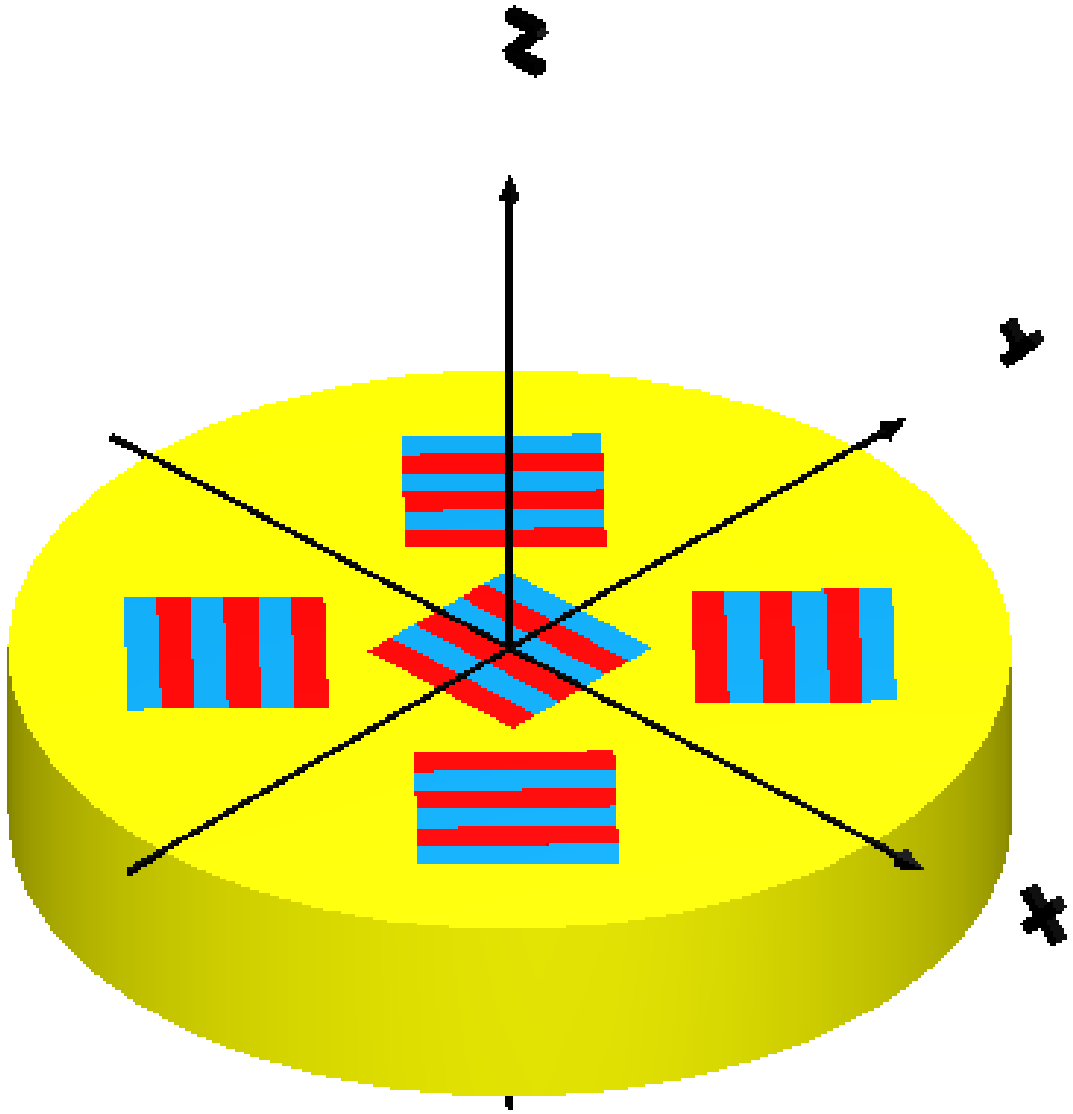
Rectangular array with cylindrical boundary




```
read geometry
unit 1
  cylinder 10 0.75 2p5.0
  cuboid 20 4p1.0 2p5.0
  media  1  1  10
  media  2  1  20 -10
boundary  20
unit 2
  cuboid 10 4p1.0 2p5.0
  media  2  1  10
boundary  10
global unit 3
  cylinder 10 6.0 2p5.0
  cylinder 20 6.5 2p6.0
  array 1 10 place 3 3 1 -0.5 -0.5 0.0
  media 3 1 20 -10
boundary  20
end geometry
```

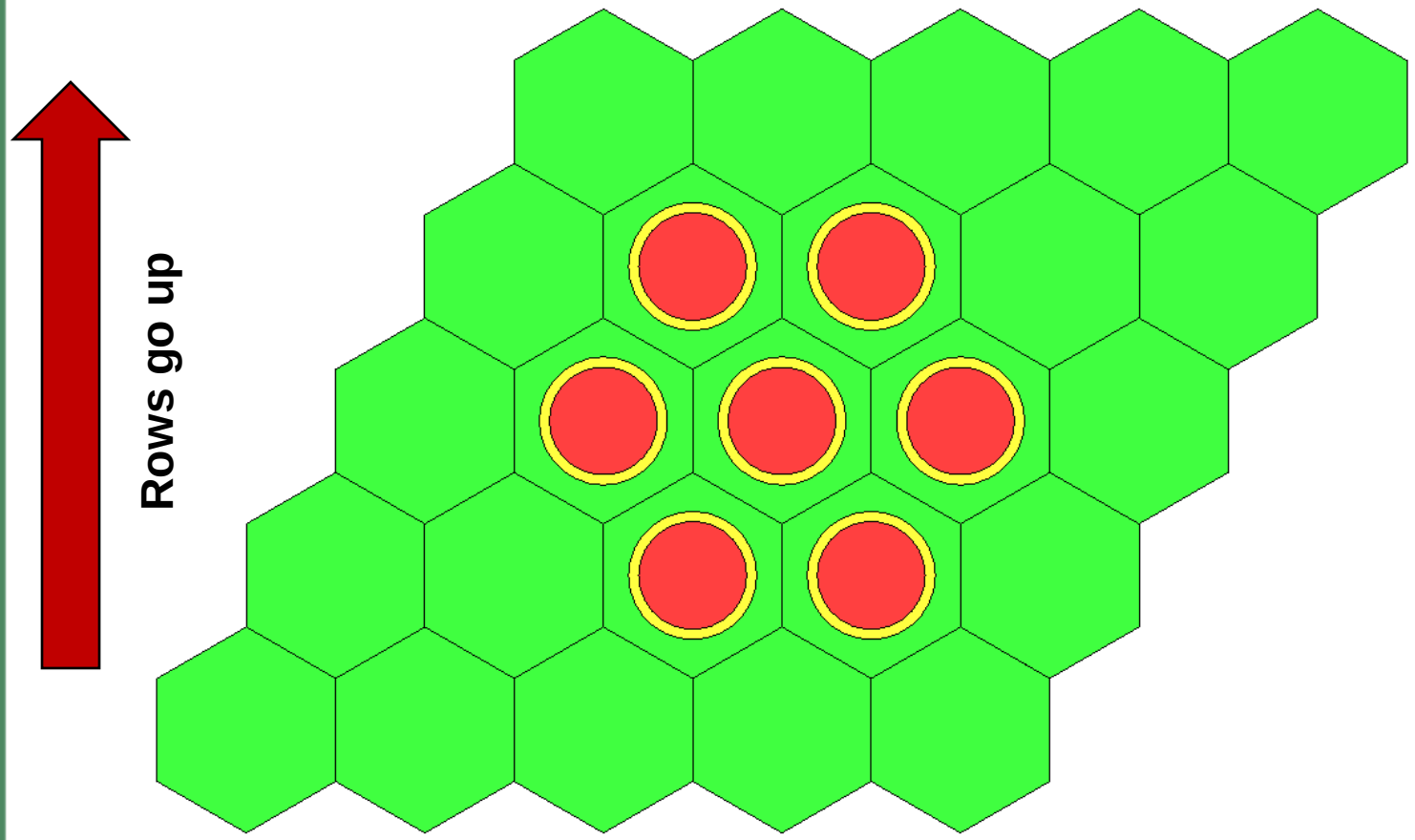
```
read array
ara=1 nux=6 nuy=6 nuz=1
fill 2 2 1 1 2 2
     2 1 1 1 1 2
     1 1 1 1 1 1
     1 1 1 1 1 1
     2 1 1 1 1 2
     2 2 1 1 2 2 end fill
end array
```

Placing arrays with holes and rotation




```
read geometry
unit 1   cuboid 10 2p6.0 2p1.0 2p10.0
  media  1  1  10   boundary  10
unit 2   cuboid 10 2p6.0 2p1.0 2p10.0
  media  2  1  10   boundary  10
unit 3   cuboid 10 2p6.0 2p6.0 2p10.0
  array 1 10 place 1 1 1 0.0 -5.0 0.0
boundary  10
global unit 4
  cylinder  10  30.0  2p12.0
  media  3  1  10
  hole 3
  hole 3 origin x=12.0  y=12.0  rotate a1=-45
  hole 3 origin x=12.0  y=-12.0 rotate a1=-135
  hole 3 origin x=-12.0 y=-12.0 rotate a1=135
  hole 3 origin x=-12.0 y=12.0  rotate a1=45
boundary 10  end geometry
read array   ara=1 nux=1 nuy=6 nuz=1
  fill 1 2 1 2 1 2 end fill   end array
```

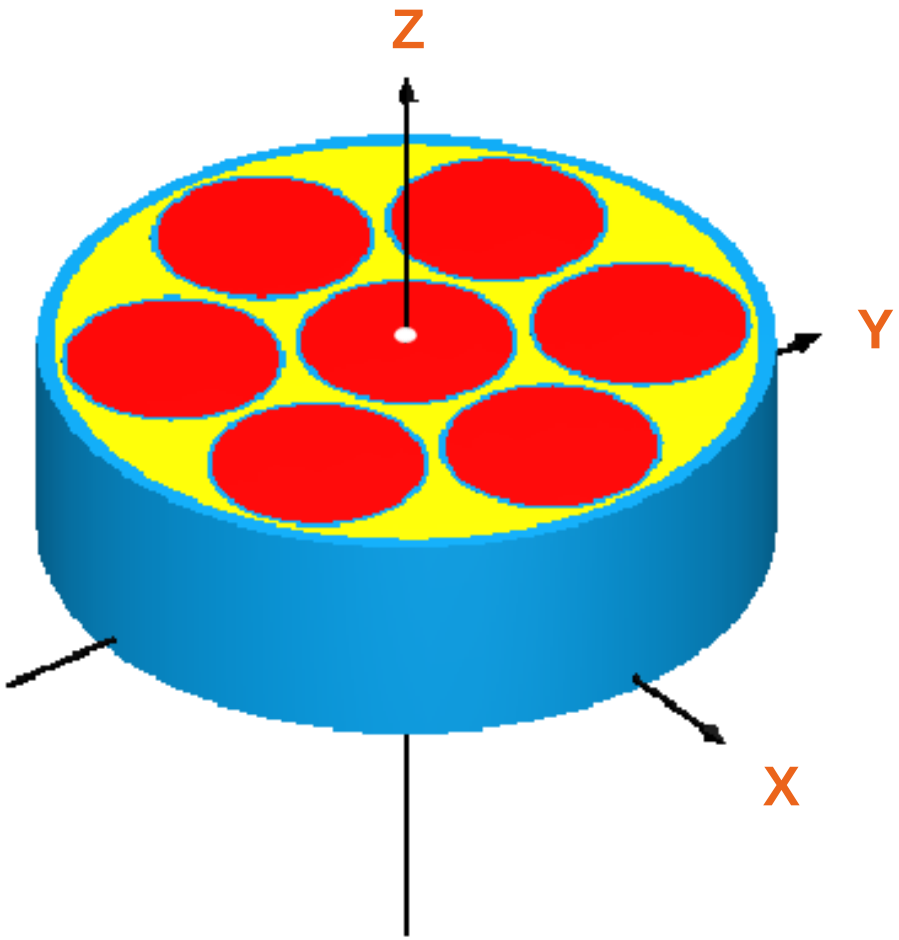
Hexagonal array




```
read array
ara=1 nux=5 nuy=5 nuz=1
typ=hexagonal
 fill
   2   2   2   2   2
     2   2   1   1   2
       2   1   1   1   2
         2   1   1   2   2
           2   2   2   2   2
 end fill
end array
```

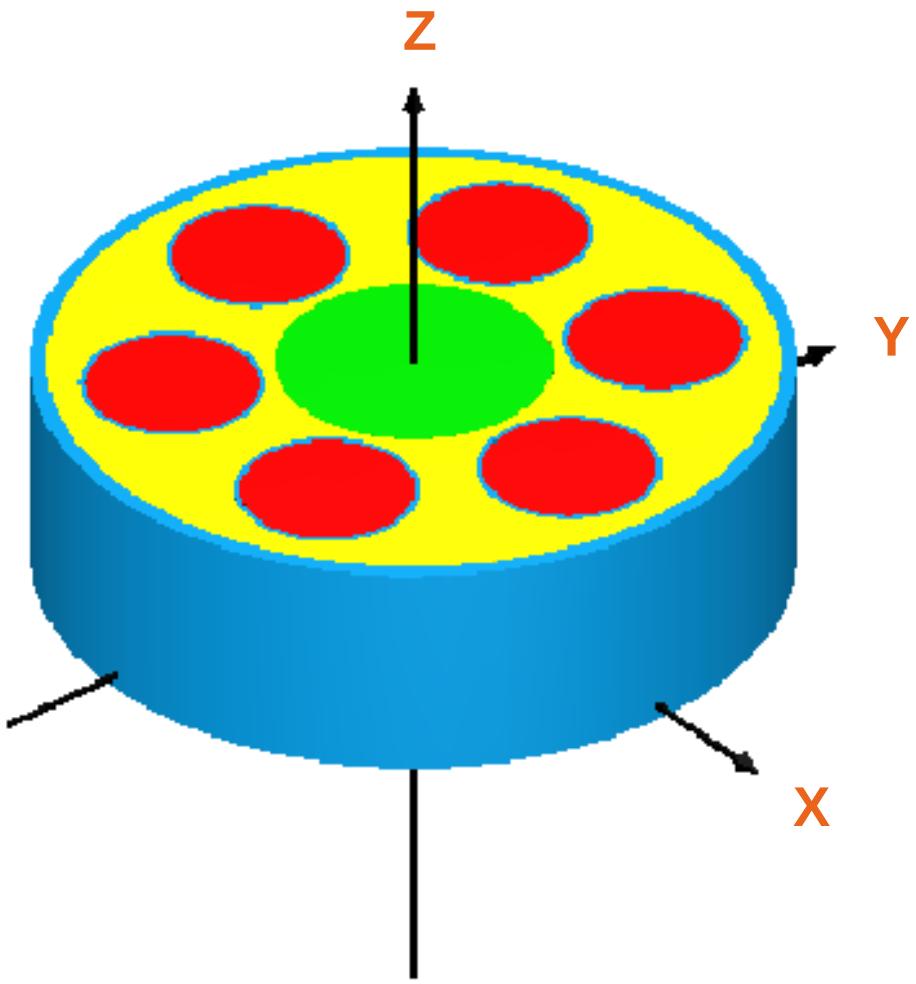
Rows go down

Hexagonal array with cylindrical boundary



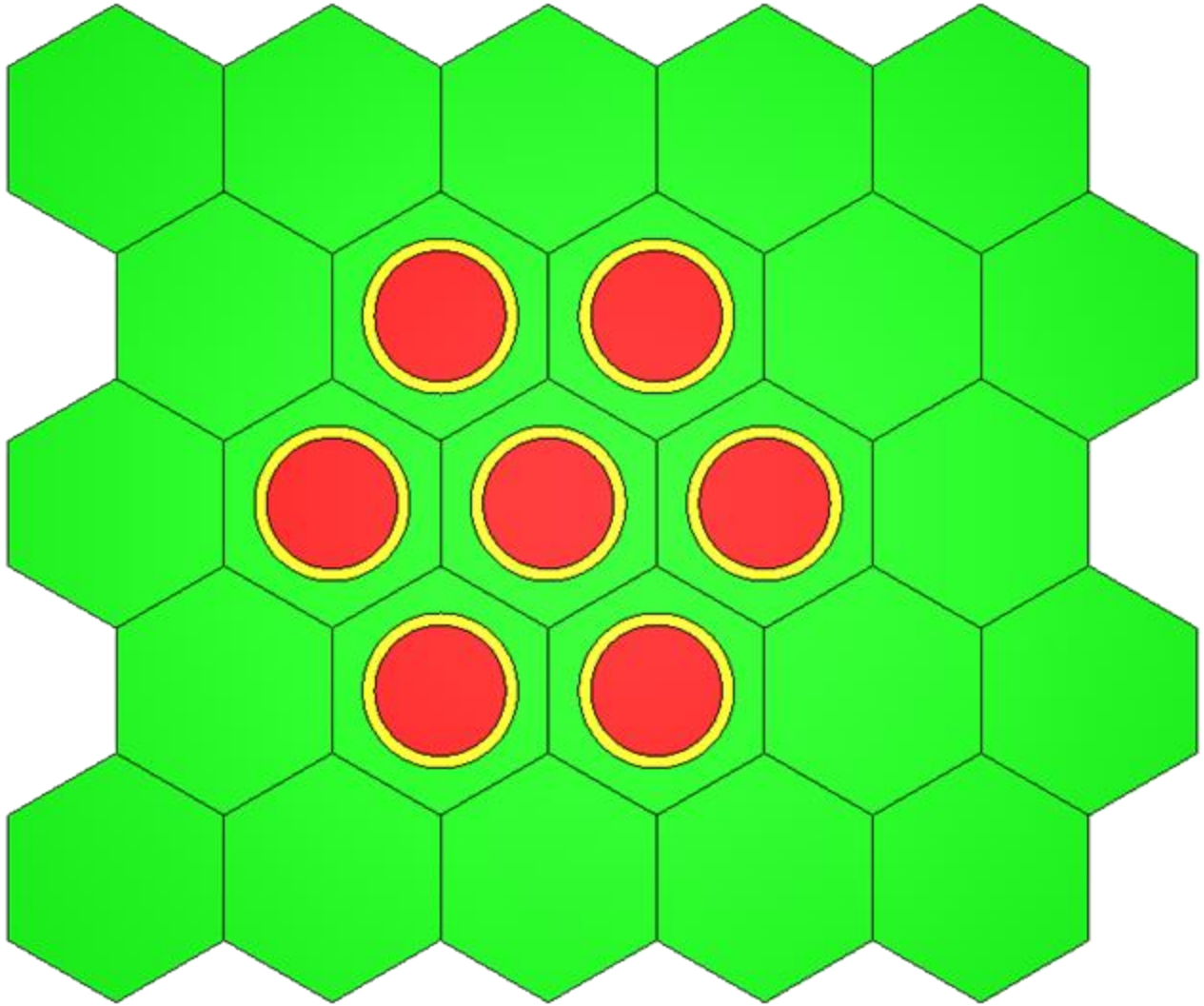

```
read geometry
unit 1   cylinder 10  9.5 18.3  0.0
  cylinder 20 10.0 18.3 -0.15
  hexprism 30 10.5 18.3 -0.15
  media 1 1 10
  media 2 1 20 -10
  media 3 1 30 -20 -10   boundary  30
unit 2   hexprism 10 10.5 18.3 -0.15
  media  3  1  10   boundary  10
global unit 3
  cylinder 10 31.5 18.3 -0.15
  cylinder 20 33.0 18.3 -1.45
  array 1 10 place 3 3 1  0 0 0
  media 2 1 20 -10
boundary  20
end geometry
read array ara=1 typ=triangular
nux=5 nuy=5 nuz=1
fill  2   2   2   2   2
      2   2   1   1   2
        2   1   1   1   2
          2   1   1   2   2
            2   2   2   2   2  end fill
end array
```

Hexagonal array with inner and outer boundaries



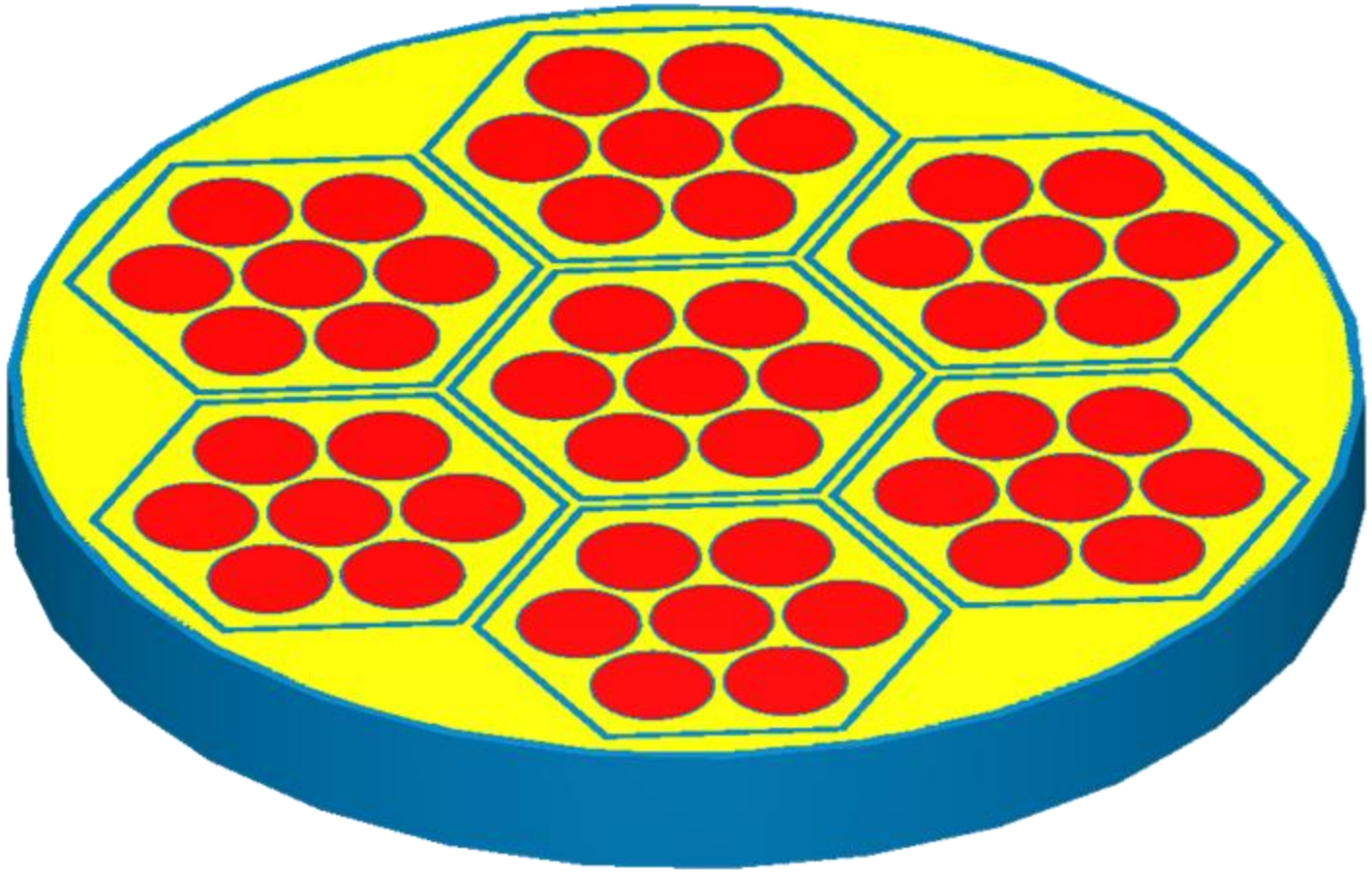

```
read geometry
unit 1   cylinder 10  7.5 18.3  0.0
   cylinder 20  8.0 18.3 -0.15
   hexprism 30 10.5 18.3 -0.15
   media 1 1 10
   media 2 1 20 -10
   media 3 1 30 -20 -10   boundary  30
unit 2   hexprism 10 10.5 18.3 -0.15
   media  3  1  10   boundary  10
global unit 3
   cylinder 10 31.5 18.3 -0.15
   cylinder 20 33.0 18.3 -1.45
   cylinder 30 12.0 18.3 -0.15
   array 1 10 -30 place 3 3 1 3r0.0
   media 2 1 20 -10
   media 4 1 30   boundary  20
end geometry
read array ara=1 typ=triangular
nux=5 nuy=5 nuz=1
fill   2    2    2    2    2
         2    2    1    1    2
           2    1    1    1    2
             2    1    1    2    2
               2    2    2    2    2  end fill
end array
```

Standard hexagonal array



```
ara=2 nux=5 nuy=5 nuz=1
typ=shexagonal
 fill
   2   2   2   2   2
     2   1   1   2   2
   2   1   1   1   2
     2   1   1   2   2
   2   2   2   2   2
 end fill
end array
```

Hexagonal array of arrays



For more of the same, review this demo later…

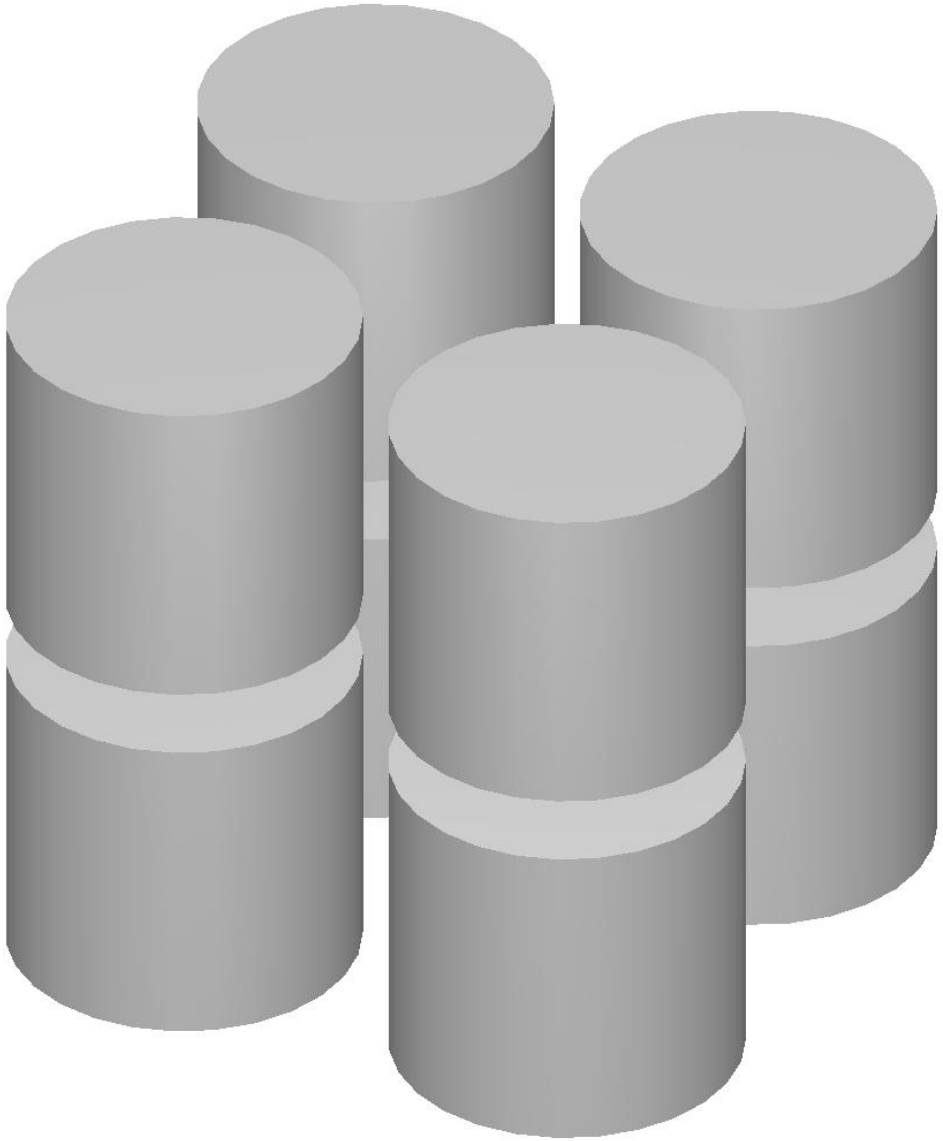
OAK RIDGE National Laboratory

Demo: Array

- 2c8 – Take 2



2c8



8 Uranium Cylinders

5.748 cm radius

10.765 cm length

13.74 cm center to center spacing in X and Y dimensions

13.01 cm center to center spacing in Z dimension

Uranium Metal

Density = 18.764 g / cc

Enrichment:

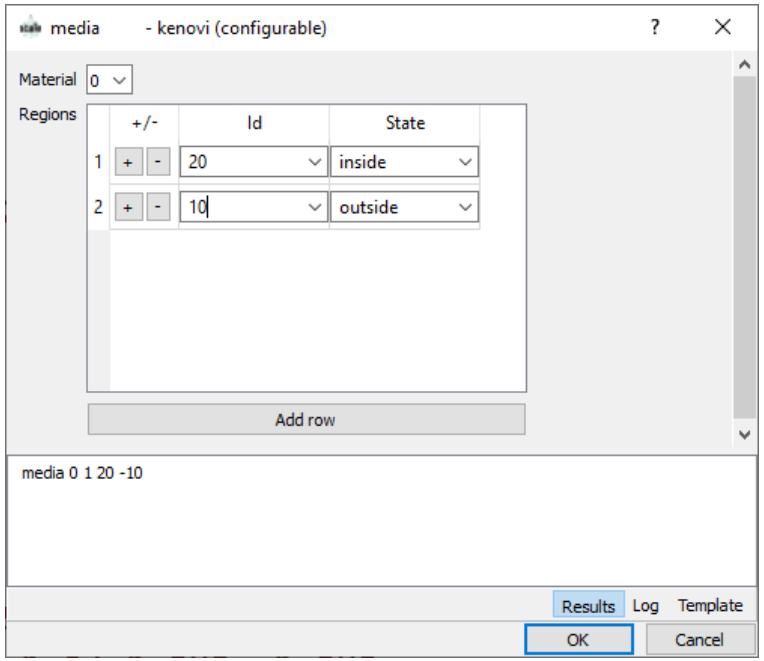
93.2%	U-235
5.6%	U-238
1.0%	U-234
0.2%	U-236

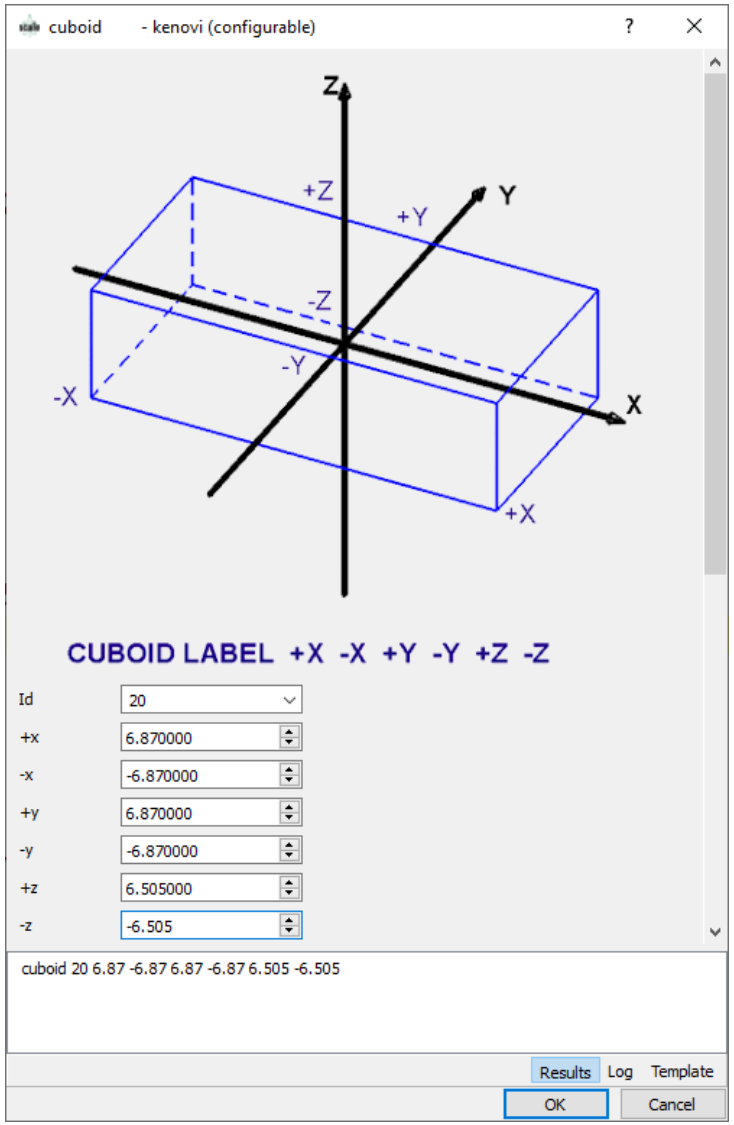
Process for this demo

1. Start from hole demo input
2. Modify unit 1 to have a void-filled cuboid boundary
3. Create array
4. Place array in Global Unit 2

Unit 1: Add cuboid for boundary

```
15 unit 1
16   cylinder 10 5.748 5.3825 -5.3825
17   cuboid 20 6.87 -6.87 6.87 -6.87 6.505 -6.505
18 '
19   media 1 1 10
20   media 0 1 20 -10
21 boundary 20
```





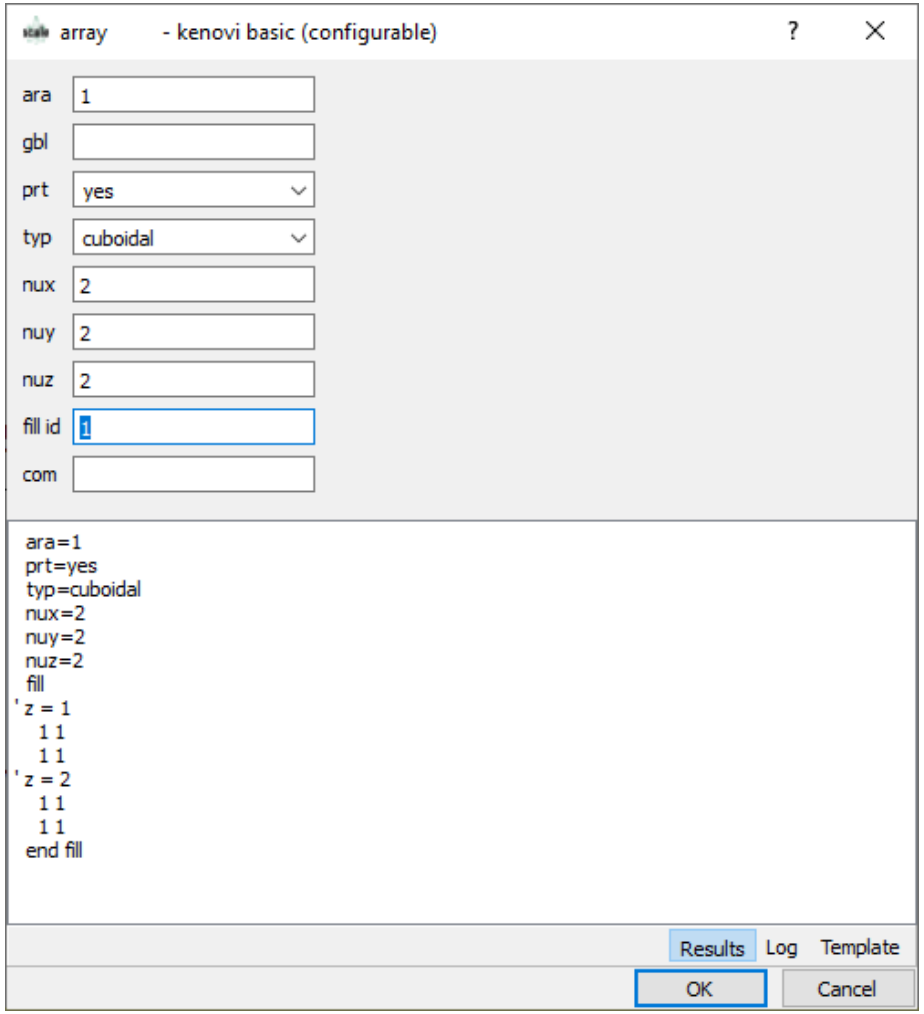
Add array block, then create array

```
'
  media 1 1 10
        0 1 20 -10
        y 20
        unit 2
        d 10 13.74 -13.74 13.74 -13.74 13.01 -13.01

        y 10
        metry

end data
end
```

cells
parameters
biasing
start
arrays
xlds
mixt
volume
energy
grid
bounds
plots
reactions



Array input

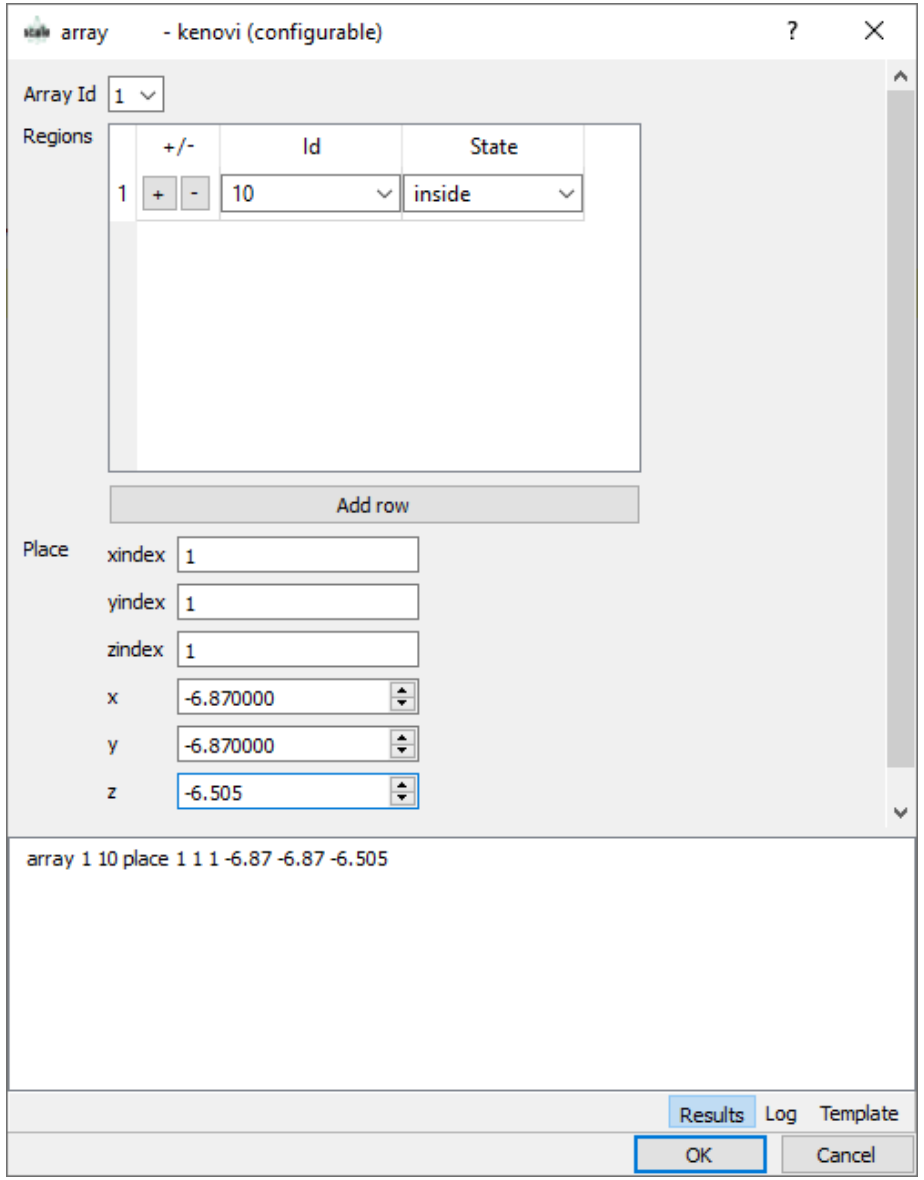
```
27 read array
28   ara=1
29   prt=yes
30   typ=cuboidal
31   nux=2
32   nuy=2
33   nuz=2
34   fill
35 ' z = 1
36      1 1
37      1 1
38 ' z = 2
39      1 1
40      1 1
41   end fill
42 end array
```

Place array in global unit

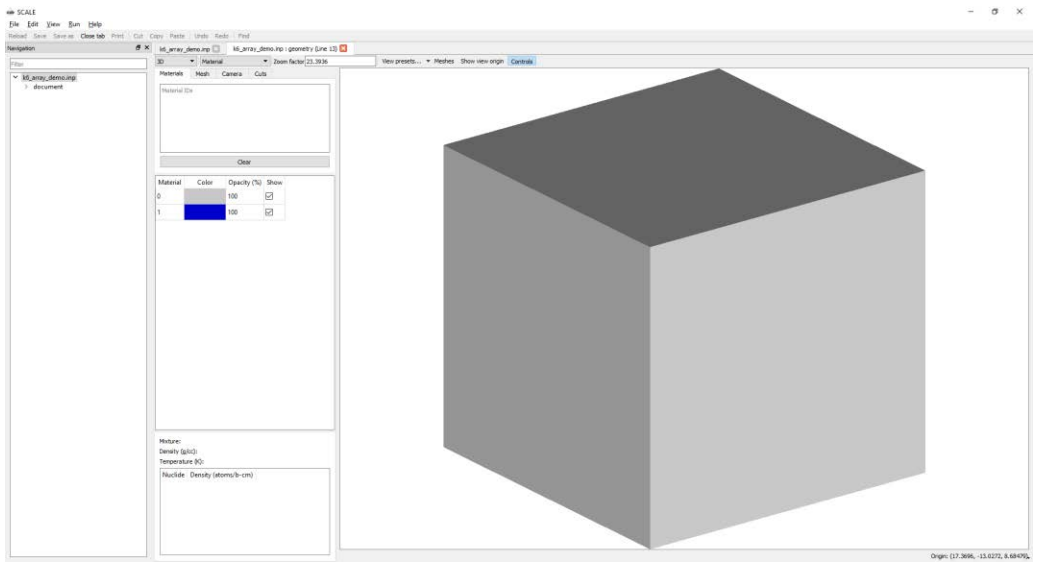
```
22 global unit 2
23   cuboid 10 13.74 -13.74 13.74 -13.74 13.01 -13.01
24   array 1 10 place 1 1 1 -6.87 -6.87 -6.505
25 boundary 10
```

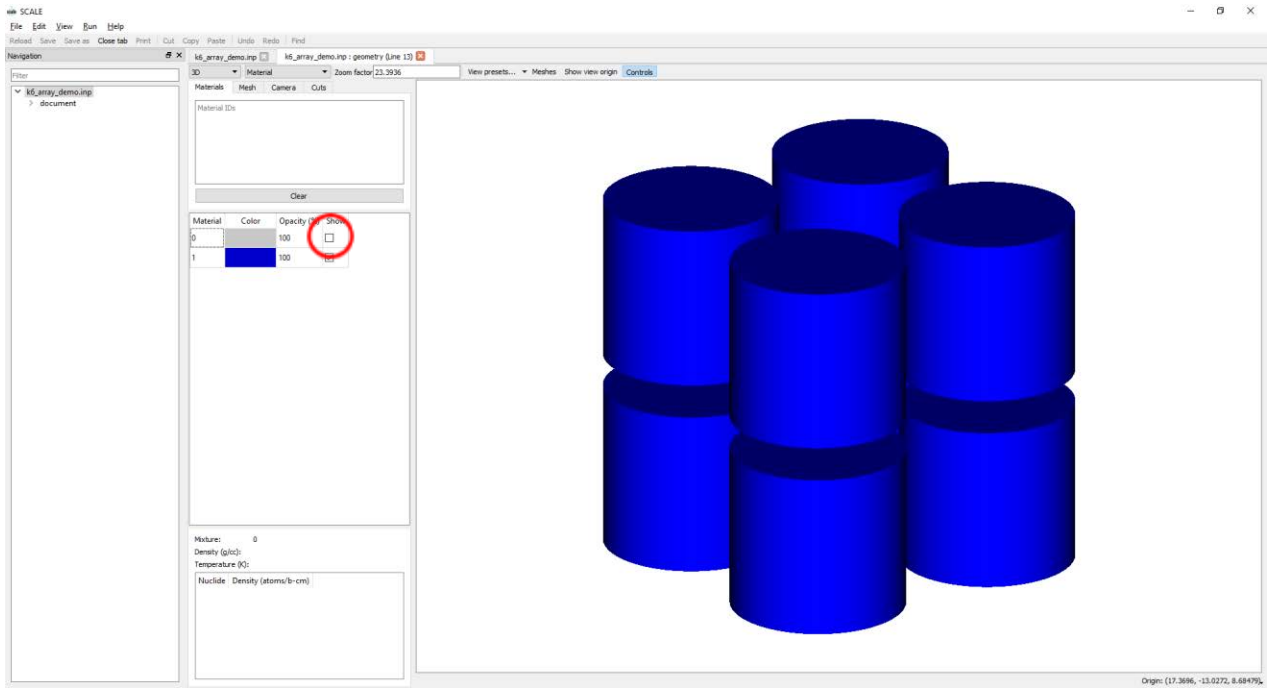
Array element as reference to position array

Location of origin of referenced element in the array in the current unit

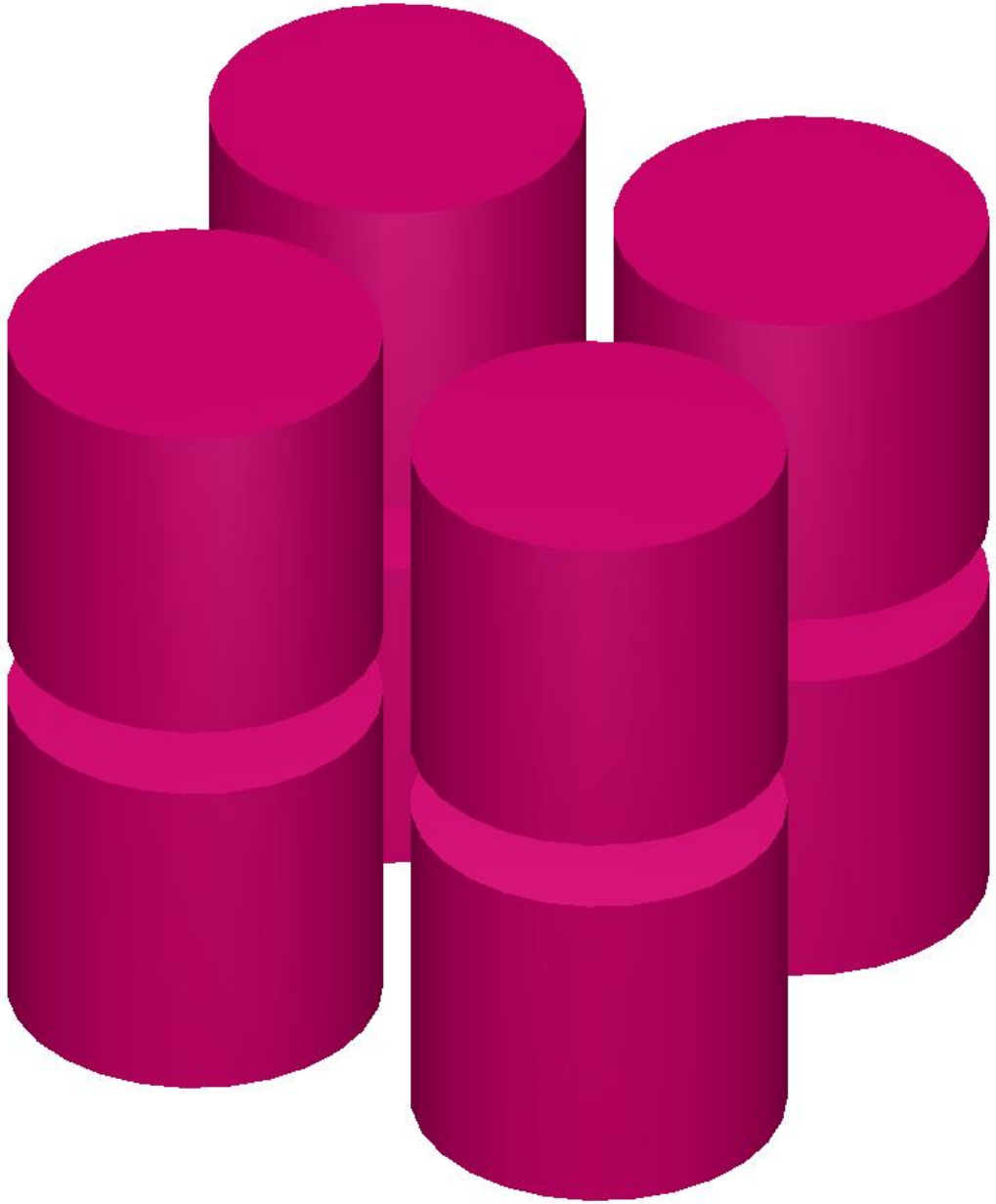


3D renderings in Fulcrum

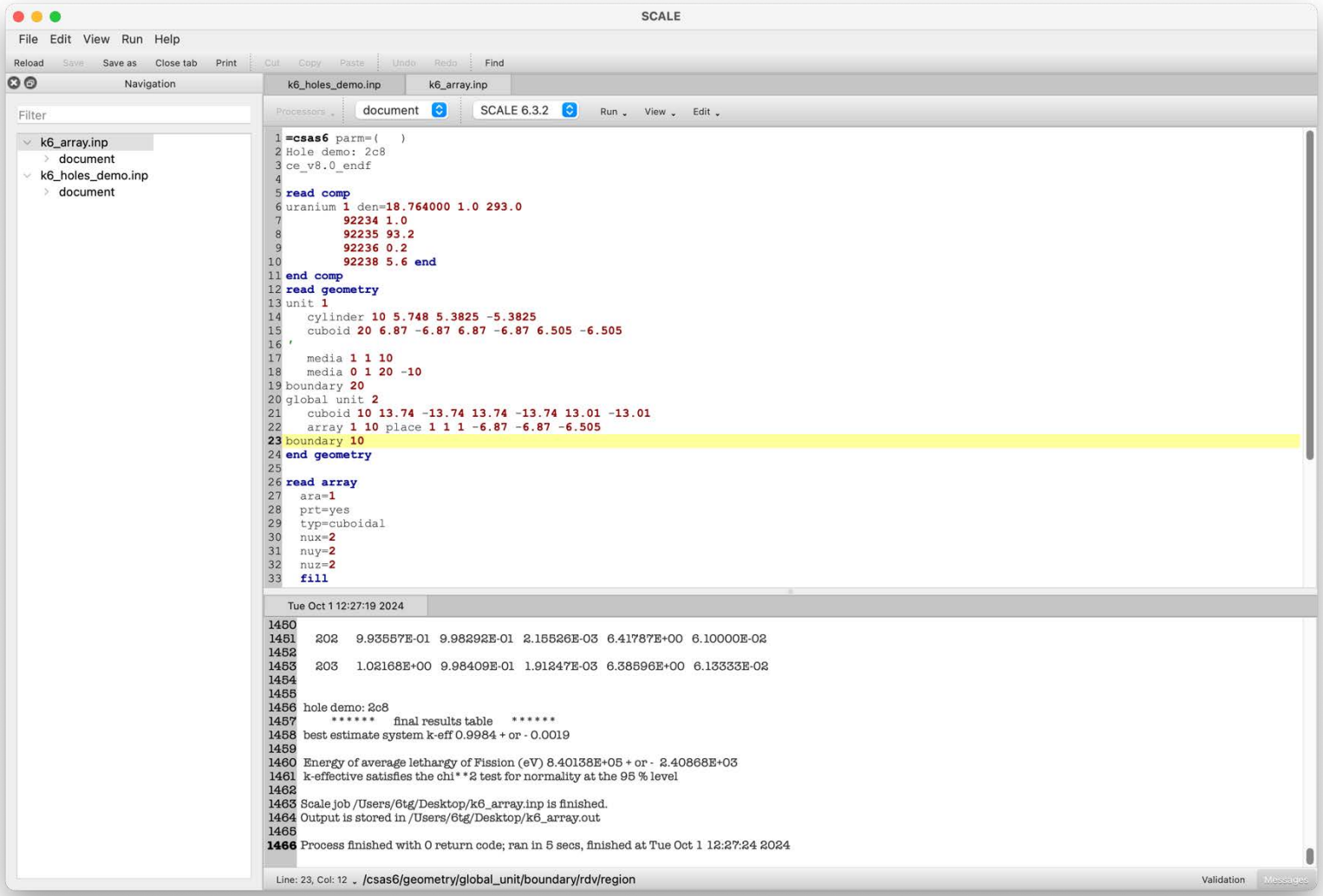




3D rendering in KENO3D



Click run



k-eff: 0.9984 +/- 0.0019

OAK RIDGE National Laboratory

Exercises for sub-module 4

- Advanced KENO-VI geometry exercises
  - Set 1: Pins in tank, array of cylinders, GOSLAR cask, CASTOR cask



Problem: Pins in a tank

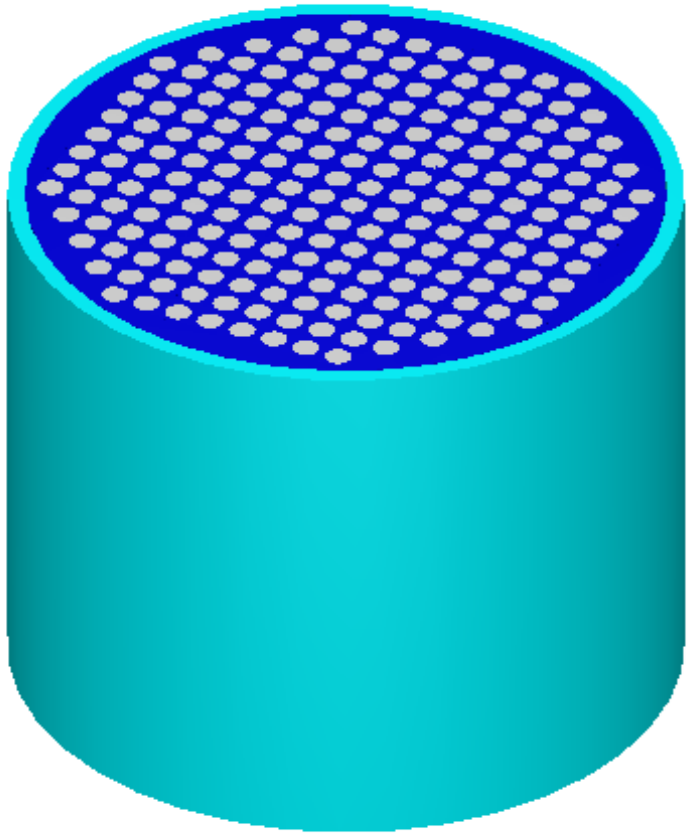


Figure shows tank cut in half (in Z) without the reflector

RODS

0.75 cm OD X 30.0 cm long

Pitch: 1.0 cm

Fuel: 20 wt% enriched U metal default density

(No ^{234}U or ^{236}U in fuel)

TANK

18 cm ID

60 cm Inside Height

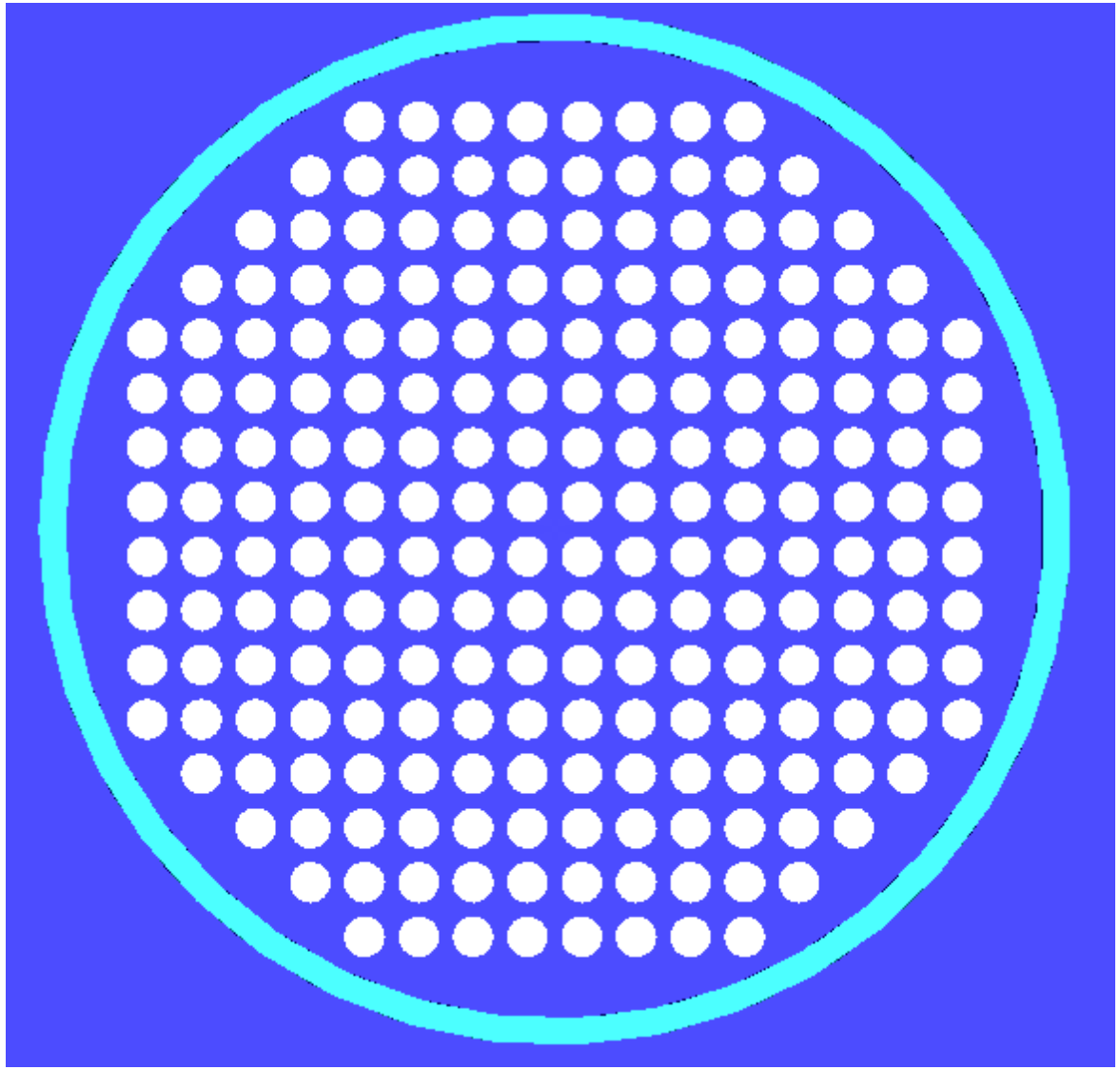
Wall & Bottom 0.5 cm Aluminum

Tank is filled to 45 cm with H_2O

Tank is reflected on the bottom and sides with 30 cm H_2O

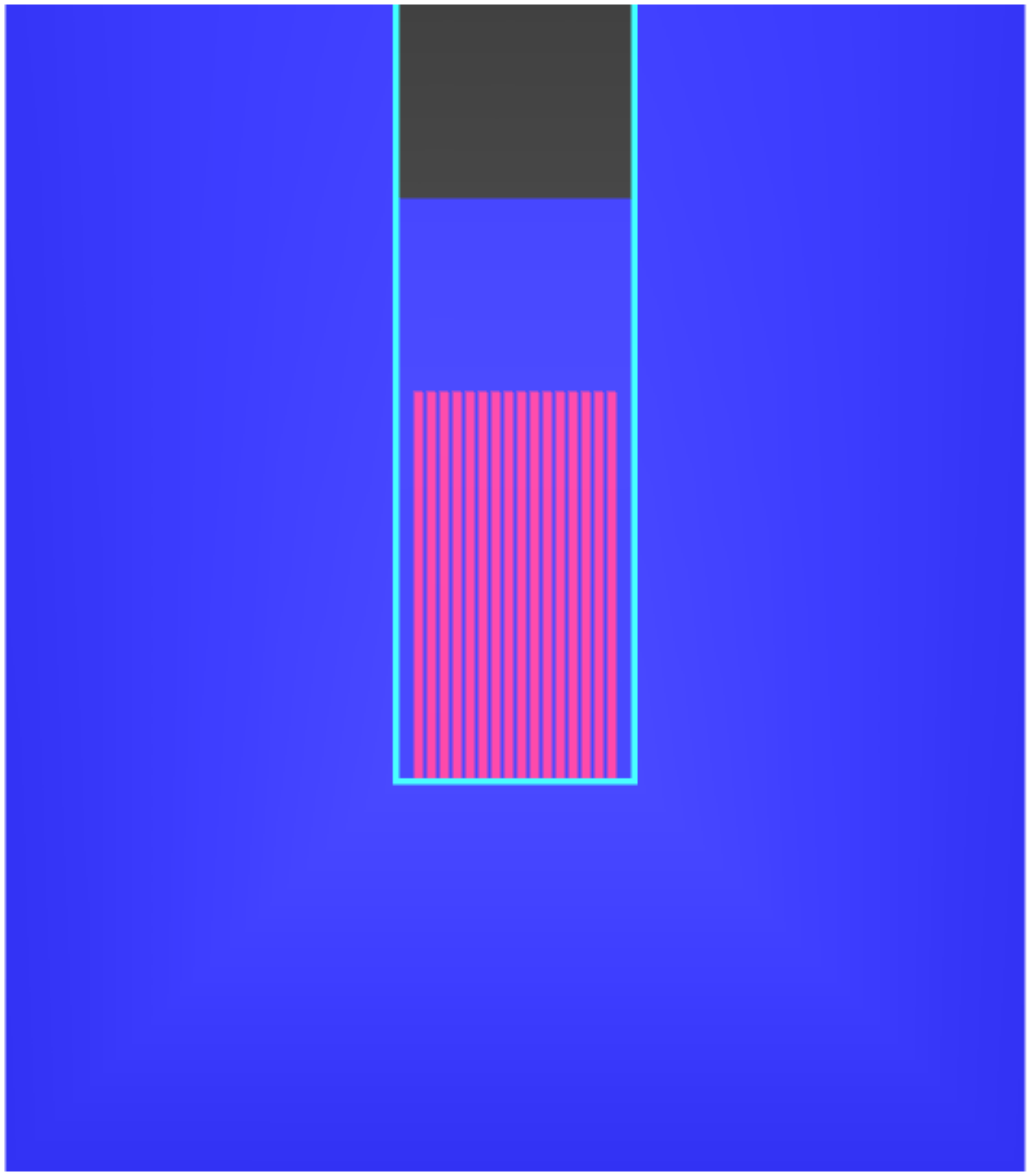
Fuel rods sit on the tank bottom

Problem: Pins in a tank (radial view)

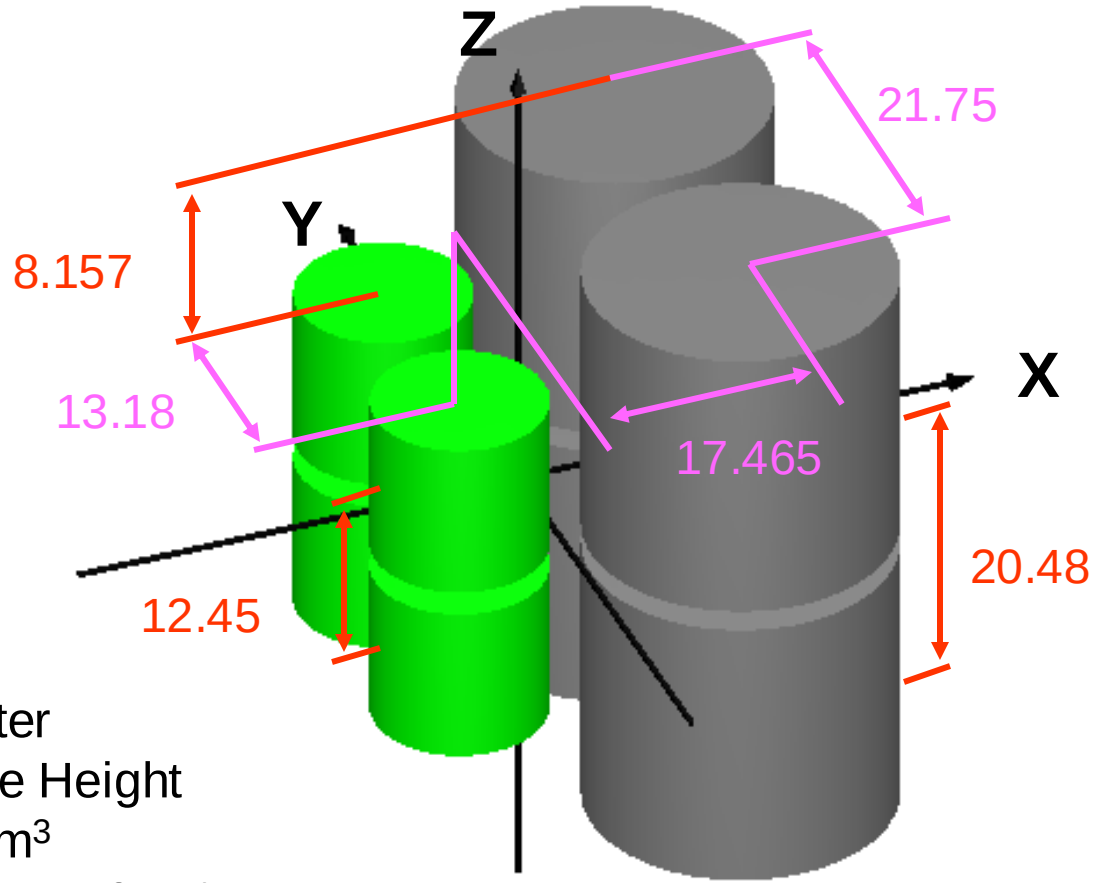


Reflector is not shown completely

Problem: Pins in a tank (axial view)



Problem: 4 aqueous – 4 metal (4A-4M) array



Metal Cylinders

11.496 cm Diameter
10.765 cm Outside Height
Density 18.76 g/cm^3
Enrichment: U-234 1.0 wt%
U-235 93.2 wt%
U-236 0.2 wt%
U-238 5.6 wt%

$UO_2(NO_3)_2$ Solution

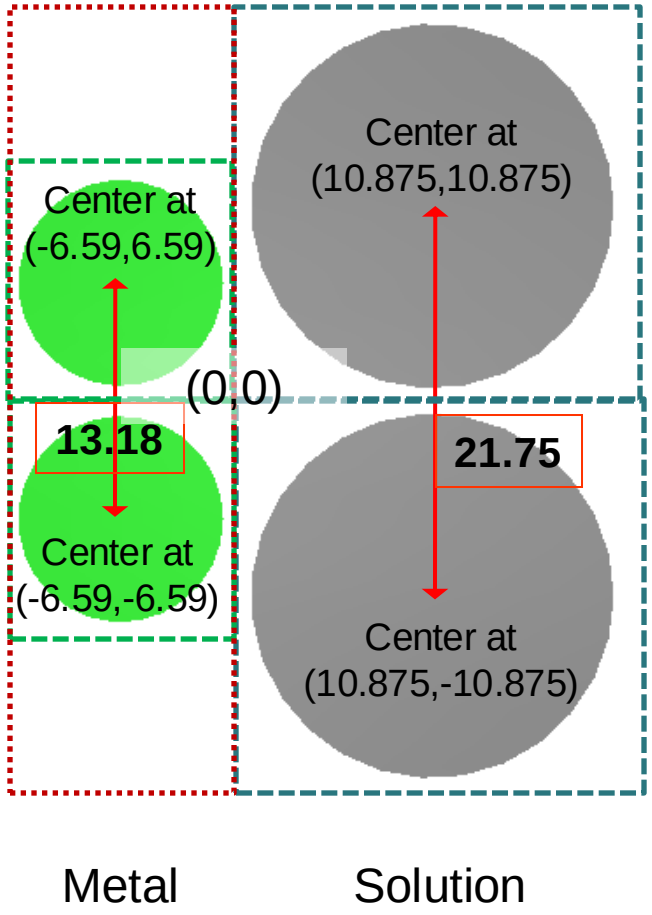
415 g U/l Solution
Let SCALE calculate density
0.39 mg excess NO_3 / g solution
NO_3 Molecular weight 61.9895
(0.009783 M excess HNO_3)
Enrichment:
U-234 1.0 wt%
U-235 92.6 wt%
U-236 0.5 wt%
U-238 5.9 wt%

Plexiglass Containers

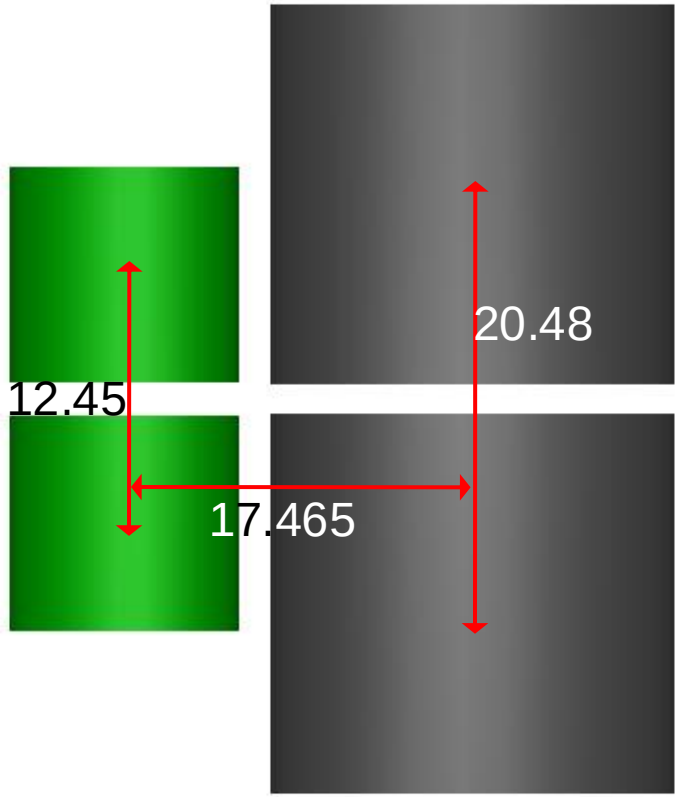
19.05 cm ID
19.05 cm Outside Height
0.635 cm thick wall & ends
Filled with UO_2 $(NO_3)_2$ Solution

4A-4M Array

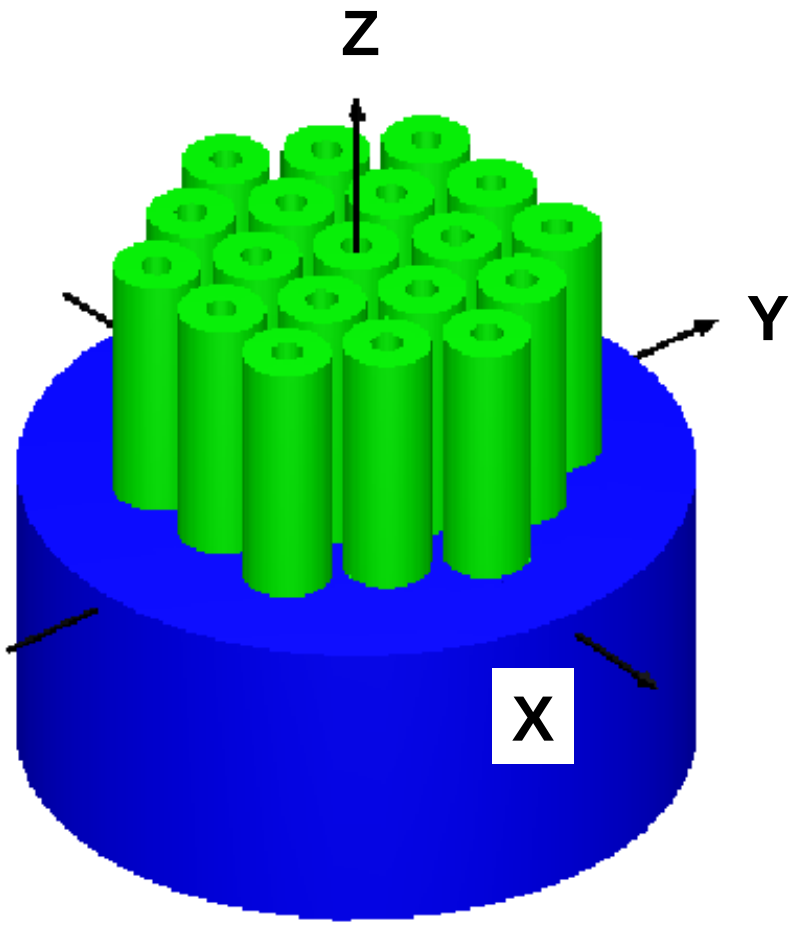
X-Y View



X-Z View



Problem: Hexagonal array of annular rods



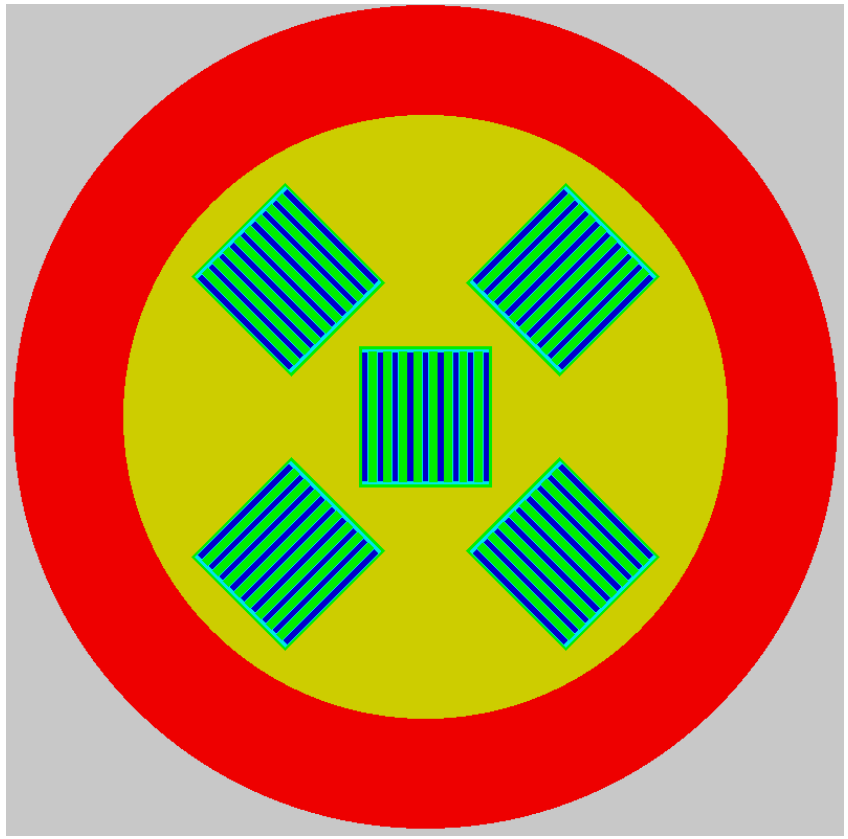
Hexagonal Array of Rods

19 rods in triangular pitched array
20.828 cm pitch (center-to-center)
Water flooded
18 cm below bottom of rods
70.0 cm radius from center
47.1 cm above bottom of rods

Annular Rods

3.302 cm inner radius
9.144 cm outer radius
101.6 cm in length
Uranium Metal
density = 19.0 g / cc
Enrichment:
U-235 1.950 wt %
U-238 98.042 wt %
U-234 0.002 wt %
U-236 0.006 wt %

Problem: GOSLAR cask



U-Al Fuel Plates
density = 3.78 g/cc
93.36 % Al
6.64 % U
Enrichment:
U-235: 92.85 wt %
U-238: 7.15 wt %

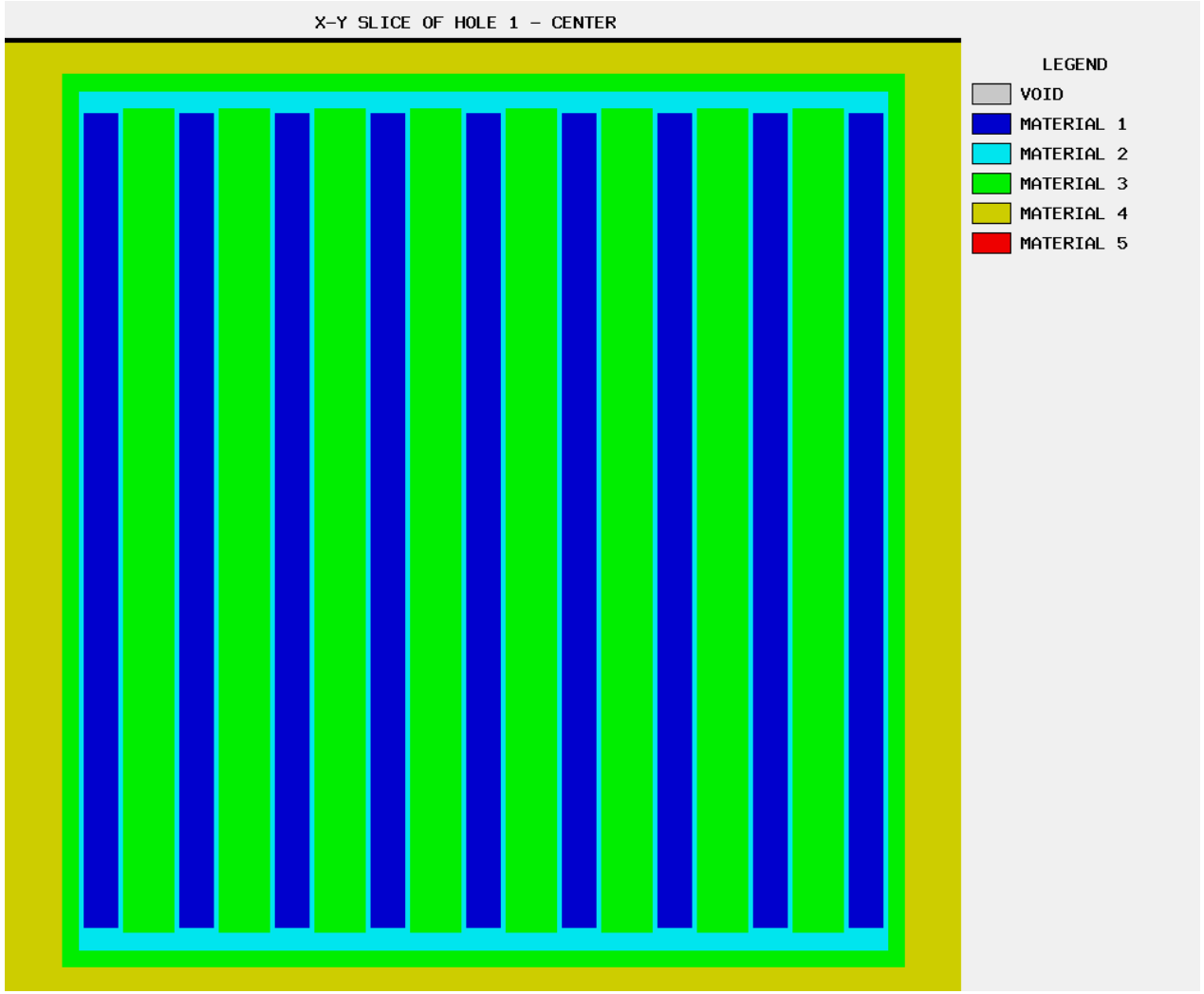
Fuel Plate Dimensions:
0.4 cm thickness
9.4 cm width
60.0 cm length
0.05 cm Al cladding on all sides
Al density = 2.7 g/cc

Fuel Assembly
contains 9 fuel plates
0.6 cm H2O gap between plates
0.2 cm thick Al plate connects
fuel plates along edge
0.2 cm H2O on X & Y surfaces

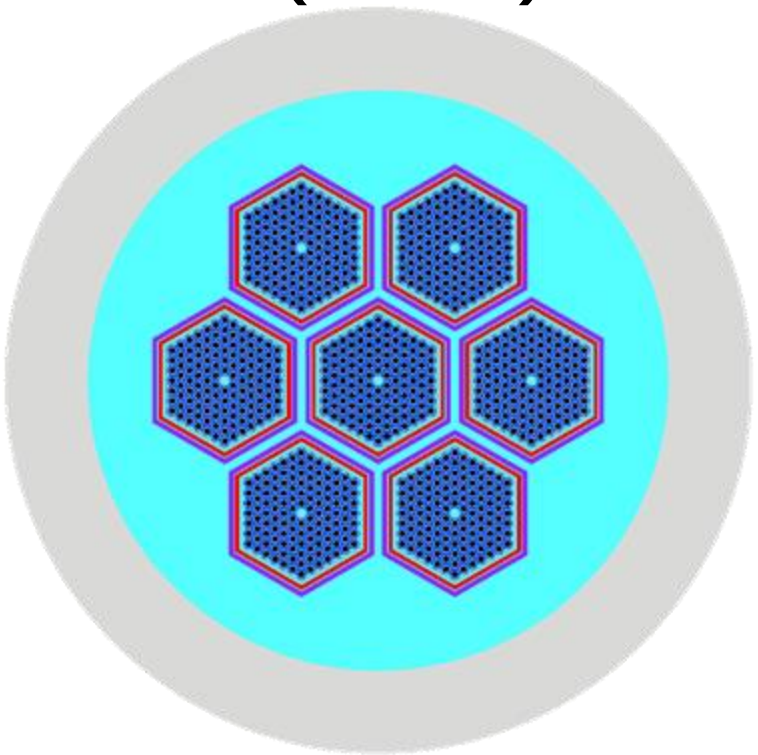
SS304 Fuel Cask
30.0cm radius x 100 cm length (outer)
22 cm radius x 80 cm length (inner)
ss304 density = 7.92 g/cc
Between assemblies: Al at 0.27 g/cc
1 center assembly and
4 outer assemblies rotated 45°
Offset from Center of Cask to the
Center of each Fuel Assembly
X=0.0, Y=0.0, Z=0.0
X=10.0, Y=10.0, Z=0.0
X=10.0, Y=-10.0, Z=0.0
X=-10.0, Y=-10.0, Z=0.0
X=-10.0, Y=10.0, Z=0.0

Not a typo

Problem: GOSLAR cask: Fuel assembly



Problem:(Mini) CASTOR cask



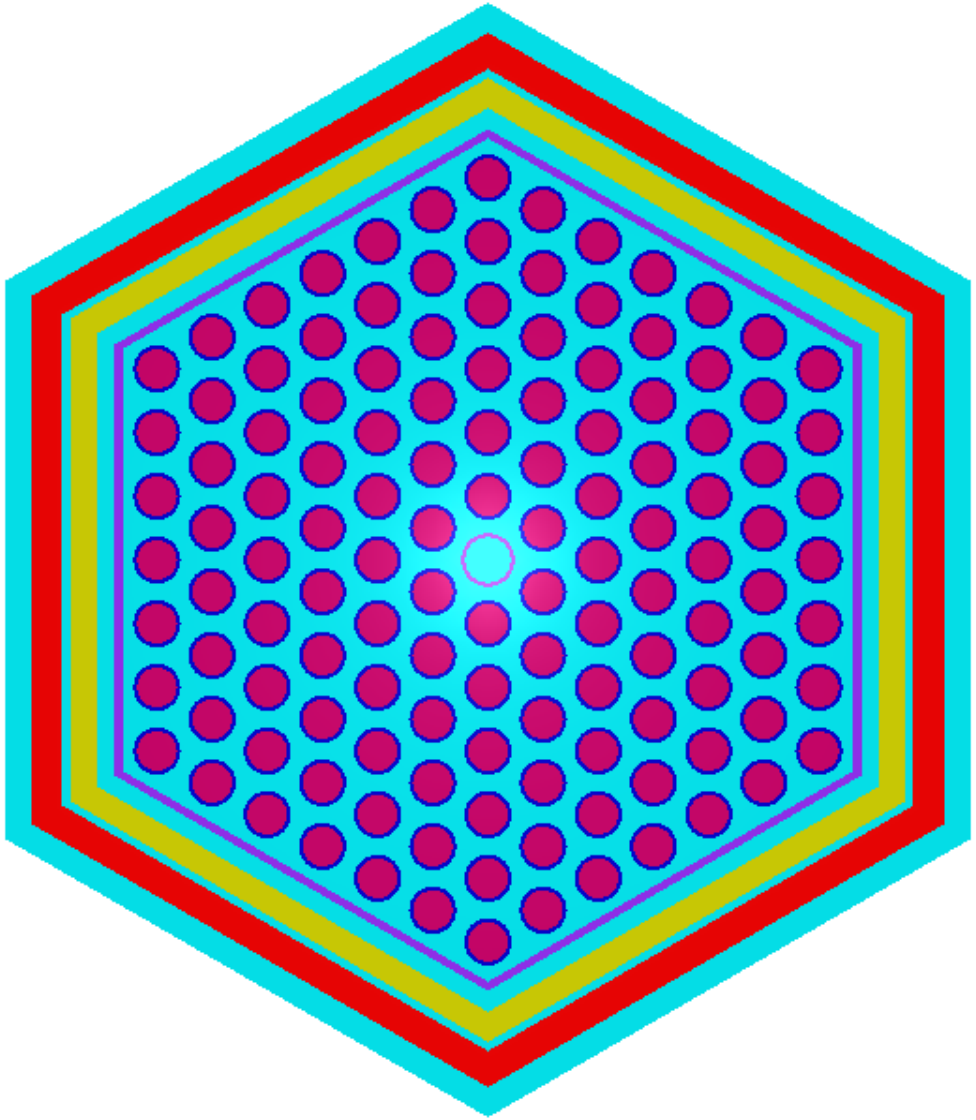
Instrument Tube
0.44 cm inner radius
0.515 cm outer radius
242.0 cm length
Zr density = 6.4369 g / cc
H_2O filled

UO2 Fuel Rods
3.6 % U-235 96.4 % U-238 UO2
density = 10.192 g / cc
0.38 cm Fuel radius
242.0 cm Fuel length
0.075 cm thick Zr clad
Zr density = 6.088 g / cc

Fuel Assembly
Water Flooded
1.22 cm pitch
126 fuel rods + 1 Instrument Tube
Central position is Instrument Tube
7.0 cm from center, 0.15 cm thick Zr plate
Zr density = 6.4369 g /cc
7.5 cm from center, 0.5 cm thick ss304-b plate
ss304-b density = 8.3157 g /cc
0.938 wt % natural boron
8.15 cm from center, 0.6 cm thick Al plate
Al density = 2.79 g / cc
18.5 cm Assembly Pitch

SS304 Fuel Cask
water flooded
contains 7 fuel assemblies
5 cm water above and below fuel
1.0 cm spacing between fuel assemblies
Inner radius = 35.0 cm
Wall, Top, and Bottom thickness = 10.0 cm
Cask outside length = 272.0 cm
SS304 density = 7.92 g / cc
Fuel positioned in center of cask

Problem: CASTOR cask: Fuel assembly in cell



Calculated k-eff values

KENO-VI	k_{eff}	Std Dev	Run-time (s)	Comparison (MG-CE)
Pins in a tank (CE)	0.8666	0.0017	262.02	0.0019 ± 0.0031 (0.6 σ)
Pins in a tank (MG)	0.8685	0.0026	140.22	
4A-4M (CE)	1.0032	0.0022	38.51	0.0000 ± 0.0033 (0.0 σ)
4A-4M (MG)	1.0032	0.0024	43.73	

Your results may differ based on the options and computer used

Calculated k-eff values

Shift	k_{eff}	Std Dev	Run-time (s)	Comparison (MG-CE)
Pins in a tank (CE)	0.866392	0.002661	213.60	0.000622 ± 0.004272 (0.14 σ)
Pins in a tank (MG)	0.867014	0.003342	149.81	
4A-4M (CE)	1.001237	0.003230	32.97	-0.004115 ± 0.003957 (1.04 σ)
4A-4M (MG)	0.997122	0.002286	42.21	

Your results may differ based on the options and computer used

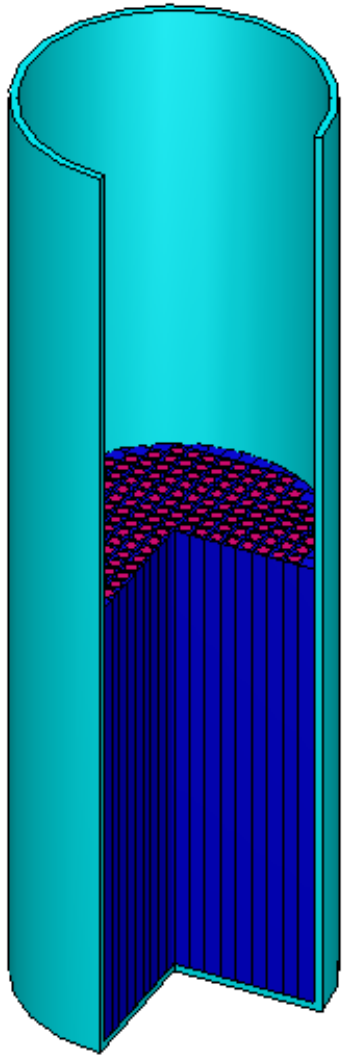
Calculated k-eff values

	KENO-VI	Shift
Pins in a tank (CE)	0.8666 +/- 0.0017	0.866392 +/- 0.002661
Pins in a tank (MG)	0.8685 +/- 0.0026	0.867014 +/- 0.003342
4A-4M (CE)	1.0032 +/- 0.0022	1.001237 +/- 0.003230
4A-4M (MG)	1.0032 +/- 0.0024	0.997122 +/- 0.002286

Your results may differ based on the options and computer used

Solution: Pins in a tank

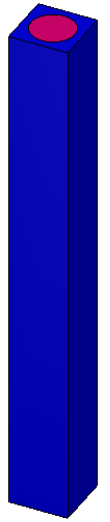
```
=csas6
tank of 20% u rods in aluminum tank
ce_v7.1_endf
read composition
 uranium     1 1 293
                                  92235 20
                                  92238 80   end
 h2o         2   end
 al          3   end
 h2o         4   end
end composition
read celldata
  latticecell squarepitch fuelr=0.375 1 hpitch=0.5 2 end
end celldata
read parameter
 fdn=yes
end parameter
```

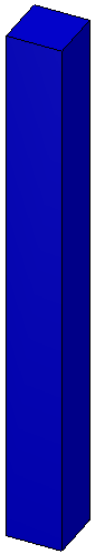


Solution: Pins in a tank (cont.)

```
read geometry
unit 1
 cylinder 10     0.375         15        -15
 cuboid 20       0.5      -0.5       0.5      -0.5         15        -15
 media 1 1 10
 media 2 1 20 -10
 boundary 20

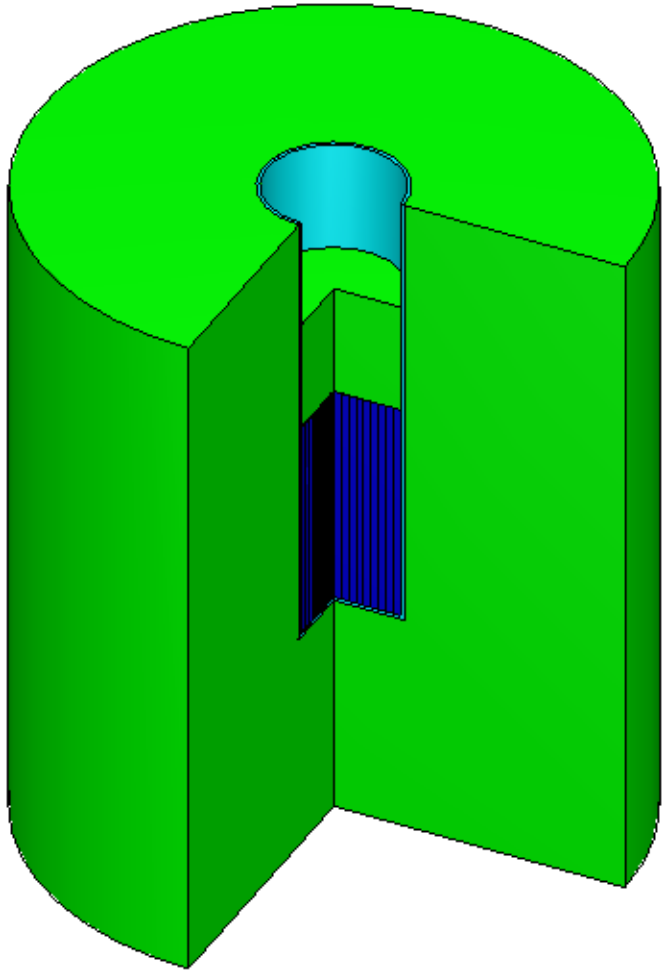
unit 2
 cuboid 20       0.5      -0.5       0.5      -0.5         15        -15
 media 2 1 20
 boundary 20
```





Solution: Pins in a tank (cont.)

```
global unit 3
 cylinder 10         9        15       -15
 cylinder 20         9        30       -15
 cylinder 30         9        45       -15
 cylinder 40       9.5        45     -15.5
 array 1 10  place 10 10 1  -0.5 -0.5 0
 cylinder 50      39.5        45     -45.5
 media 4 1 20 -10
 media 0 1 30 -20
 media 3 1 40 -30
 media 4 1 50 -40
 boundary 50
end geometry
read array
ara=1 nux=20 nuy=20 nuz=1 typ=square
```

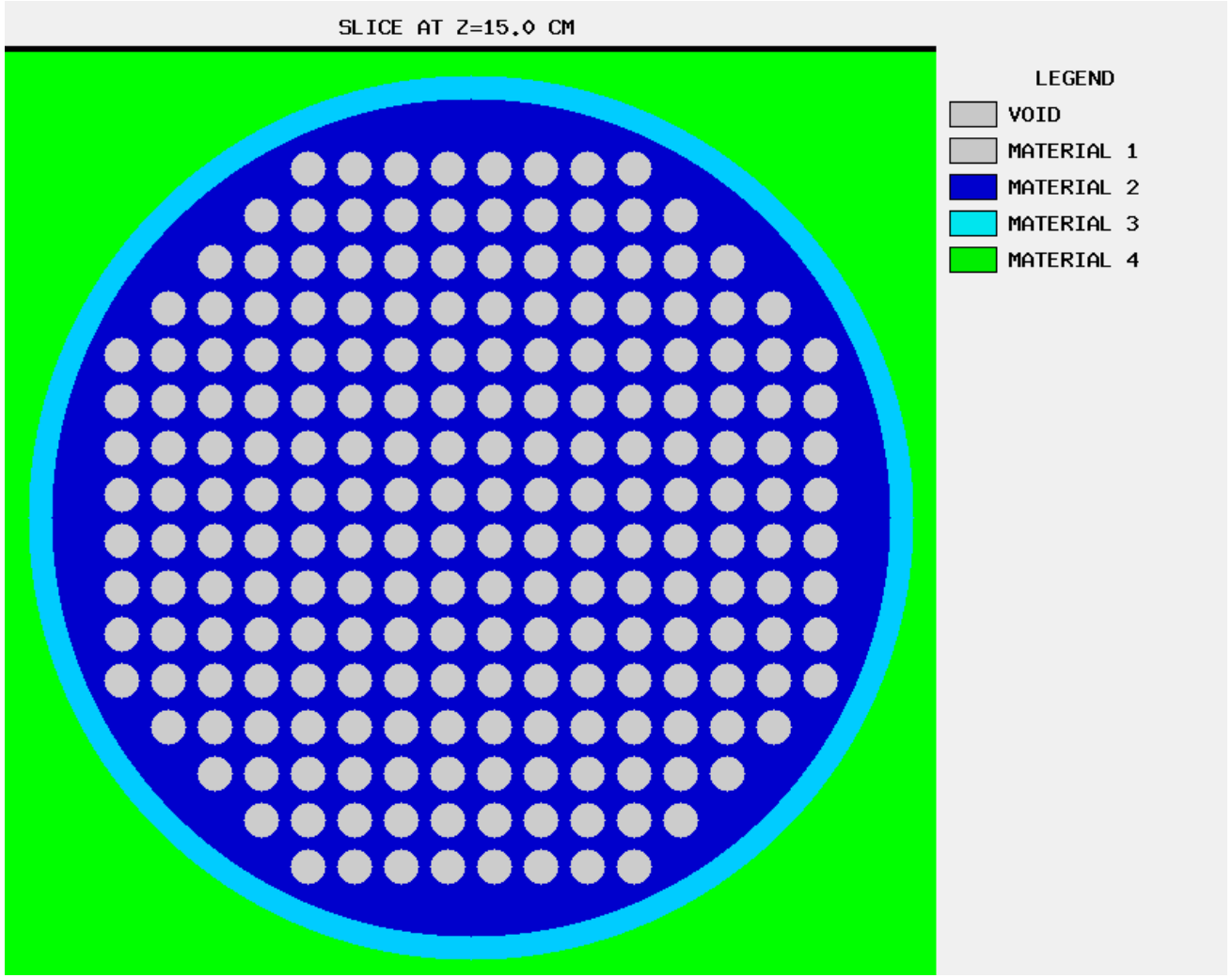


Solution: Pins in a tank (cont.)

```
fill
    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
    2    2    2    2    2    2    1    1    1    1    1    1    1    1    2    2    2    2    2    2
    2    2    2    2    2    1    1    1    1    1    1    1    1    1    1    2    2    2    2    2
    2    2    2    2    1    1    1    1    1    1    1    1    1    1    1    1    2    2    2    2
    2    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2    2
    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2
    2    2    2    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2    2
    2    2    2    2    1    1    1    1    1    1    1    1    1    1    1    1    2    2    2    2
    2    2    2    2    2    1    1    1    1    1    1    1    1    1    1    2    2    2    2    2
    2    2    2    2    2    2    1    1    1    1    1    1    1    1    2    2    2    2    2    2
    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
end fill
end array
```

Solution: Pins in a tank (cont.)

```
read plot
 ttl='slice at z=15.0 cm '
 xul=-10  yul=10  zul=0
 xlr=10    ylr=-10  zlr=0
 nax=640
 clr=1 200 200 200
     2 0 0 205
     3 0 229 238
     4 0 238 0
 end color
 typ=xy end
end plot
end data
end
```



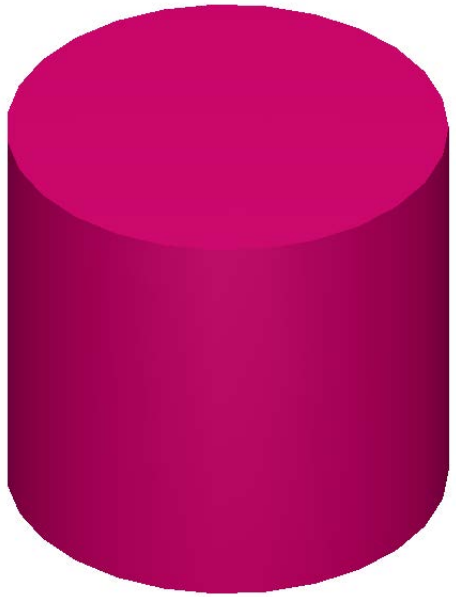
Solution: 4A-4M array

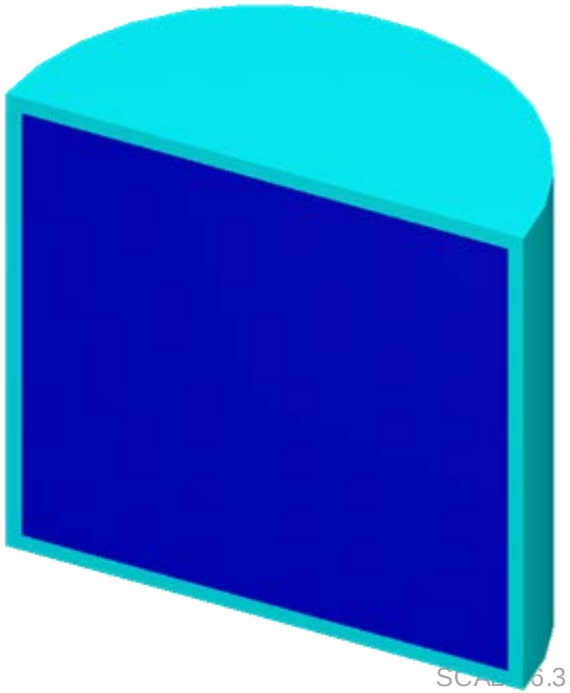
```
=csas5 parm=( )
Class exercise: 4a-4m
ce_v7.1


read comp
u 1 den=18.76 1.0 293 92234 1.0 92235 93.2 92236 0.2 92238 5.6 end
solution mix=2
   rho[uo2(no3)2]=415.0 92234 1.0 92235 92.6 92236 0.5 92238 5.9
   molar[hno3]=0.009783
   density=1.555 temp=293 volfrac=1.0
end solution
plexiglas 3 end
end comp
```

Solution: 4A-4M array (cont.)

```
unit 1
com='metal cylinders'
  cylinder 10 5.748 5.3825 -5.3825
  cuboid 20 4p6.59 2p6.225
'
  media 1 1 10
  media 0 1 20 -10
boundary 20
unit 2
com='solution cylinders'
  cylinder 20 9.525 8.889 -8.889
  cylinder 30 10.16 9.525 -9.525
  cuboid 40 4p10.875 10.24 -10.24
'
  media 2 1 20
  media 3 1 30 -20
  media 0 1 40 -30
boundary 40
```

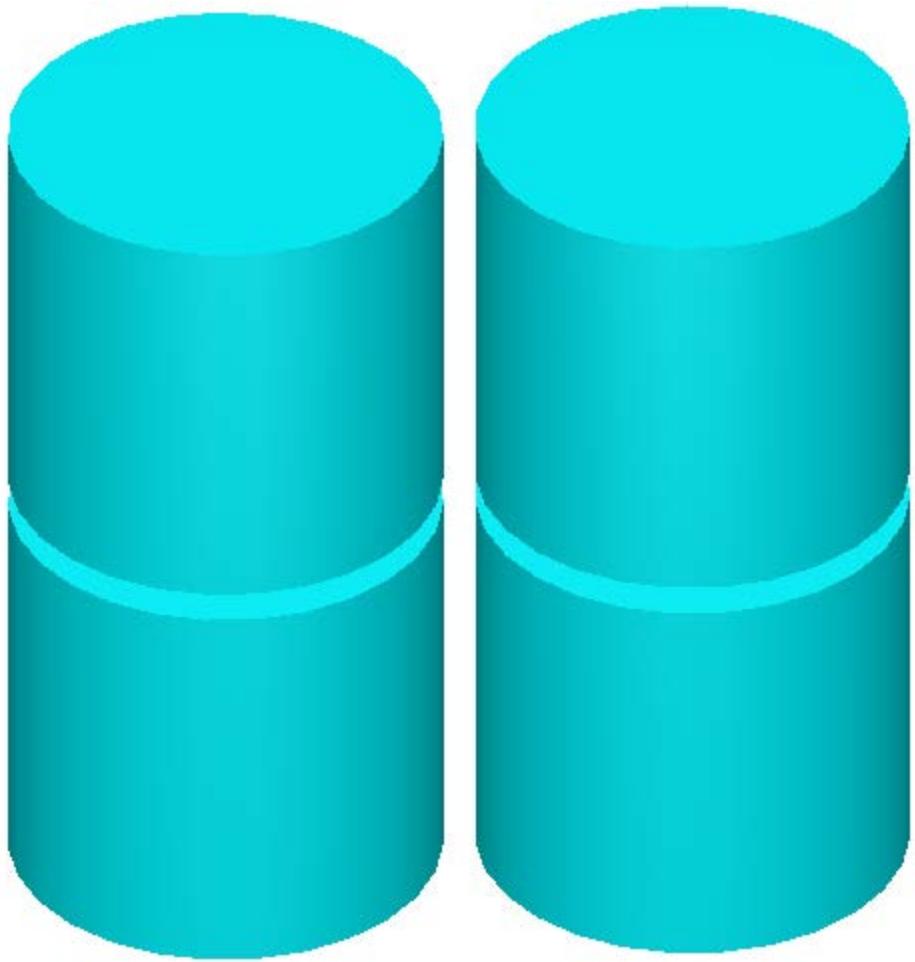




Solution: 4A-4M array (cont.)

```
unit 3
com='array of solution tanks’
  cuboid 10 2p10.875 2p21.75 2p20.48
  array 2 10 place 1 1 1 0 -10.875 -10.24
boundary 10

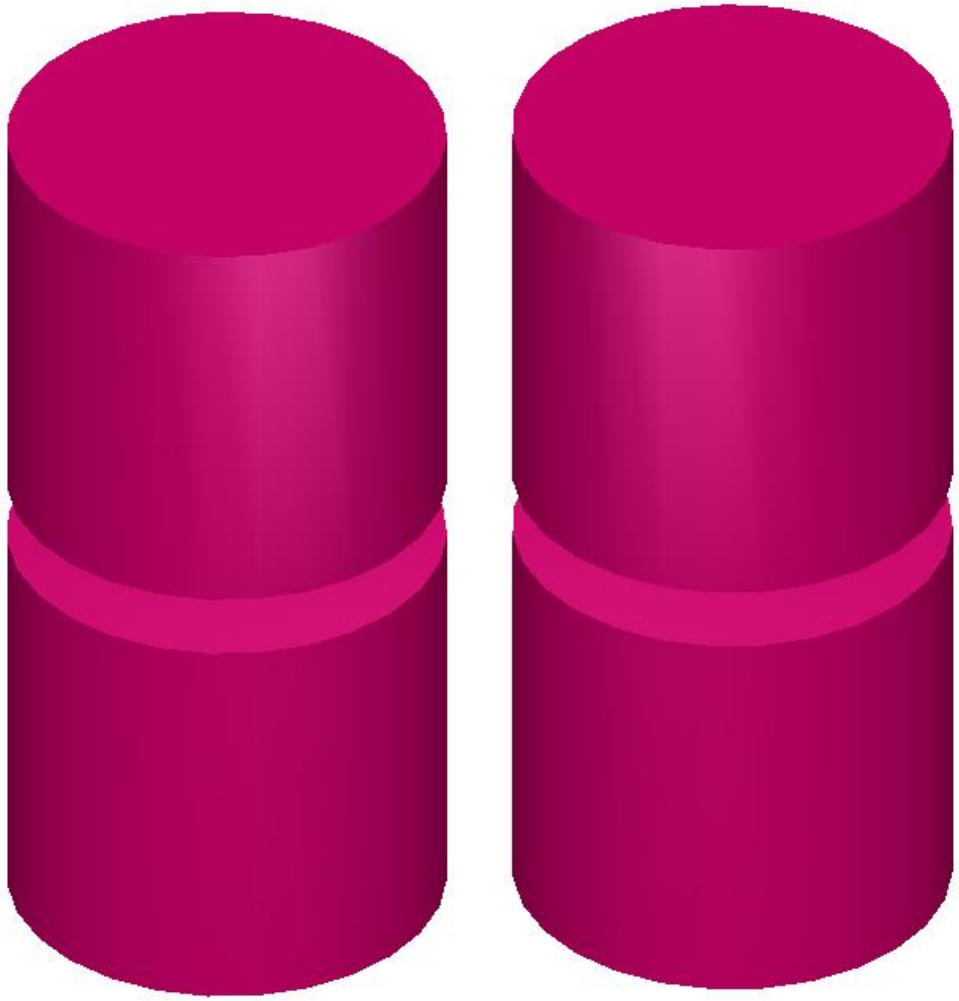

          ara=2
          prt=yes
          nux=1
          nuy=2
          nuz=2
          com="Array of solution tanks"
          fill
          ' z = 1
          2
          2
          ' z = 2
          2
          2
          end fill
```



Solution: 4A-4M array (cont.)

```
unit 4
com='array of metal cylinders'
  cuboid 10 2p6.59 2p13.18 2p12.45
  cuboid 20 6.59 -6.59 21.75 -21.75 20.48 -20.48
  array 1 10 place 1 1 1 0 -6.59 -6.225
  media 0 1 20 -10
boundary 20
```

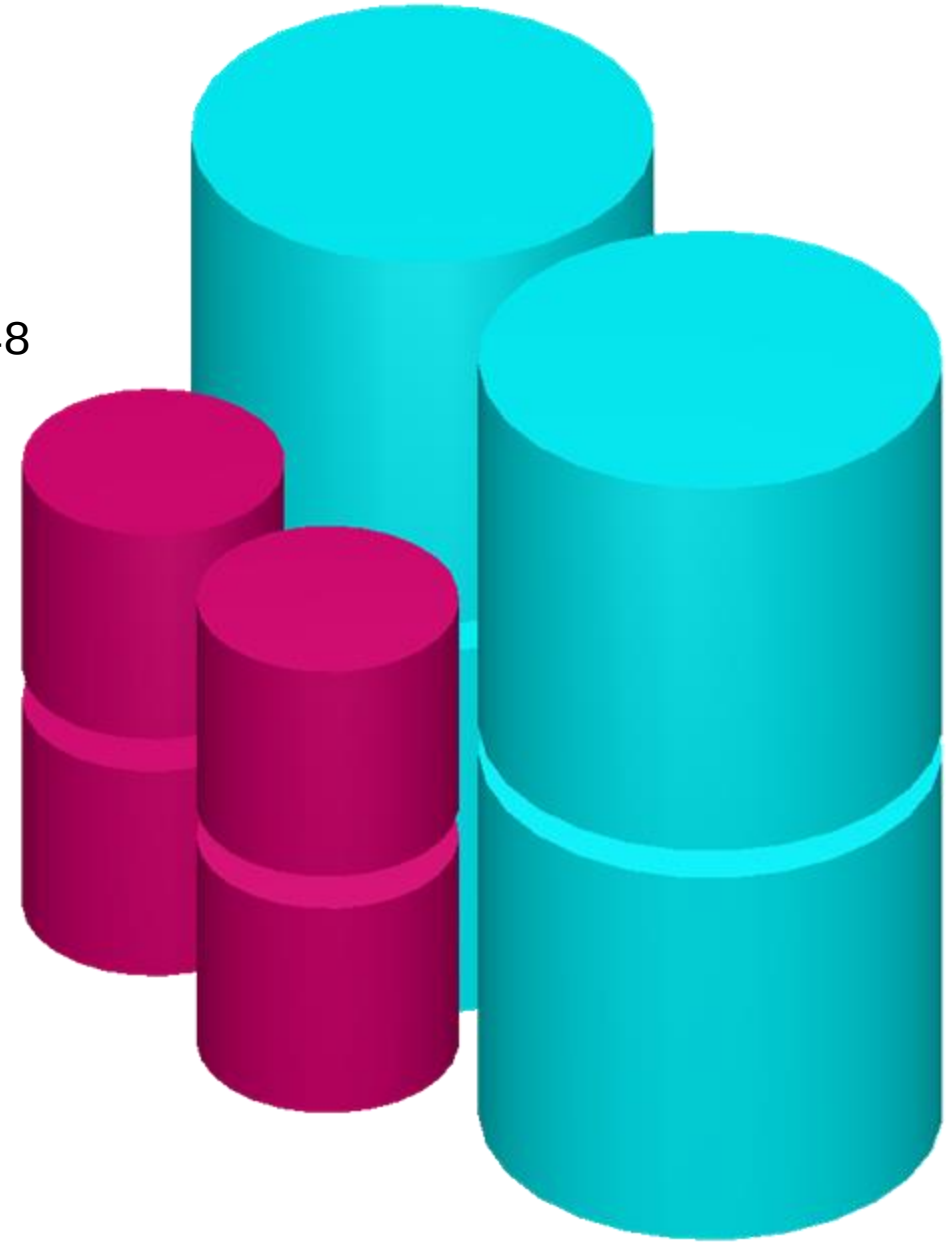
```
ara=1 nux=1 nuy=2 nuz=2
com='array of metal cylinders'
fill 1 1
     1 1 end fill
```



Solution: 4A-4M array (cont.)

```
global unit 5
com='global, containing array of arrays'
  cuboid 10 21.75 -13.18 21.75 -21.75 20.48 -20.48
  array 3 10 place 1 1 1 -6.59 0 0
boundary 10
```

```
ara=3 nux=2 nuy=1 nuz=1
com='array of arrays'
fill 4 3 end fill
```



Calculated k-eff results

KENO-VI	k_{eff}	std. dev.	Run-time (s)	Comparison (MG-CE)
Annular hex array – ce_v7.1	1.0021	0.0018	183.29	-0.0004 ± 0.0028 (0.14 σ)
Annular hex array – v7.1-252	1.0017	0.0022	109.68	
Goslar cask – ce_v7.1	0.4859	0.0018	117.41	0.0071 ± 0.0024 (2.95 σ)
Goslar cask – v7.1-252	0.4930	0.0016	73.62	
Castor cask – ce_v7.1	0.6286	0.0020	197.18	-0.0039 ± 0.0026 (1.52 σ)
Castor cask – v7.1-252	0.6247	0.0016	97.87	

Your results may differ based on the options, geometry, and computer used

Calculated k-eff results

Shift	k_{eff}	std. dev.	Run-time (s)	Comparison (MG-CE)
Annular hex array – ce_v7.1	0.999171	0.002285	117.29	0.000249 ± 0.003054 (0.08 σ)
Annular hex array – v7.1-252	0.999420	0.002026	100.85	
Goslar cask – ce_v7.1	0.485404	0.002076	147.32	0.005503 ± 0.003252 (1.69 σ)
Goslar cask – v7.1-252	0.490907	0.002503	97.80	
Castor cask – ce_v7.1	0.623442	0.002489	205.93	0.002345 ± 0.003216 (0.73 σ)
Castor cask – v7.1-252	0.625787	0.002036	162.35	

Your results may differ based on the options, geometry, and computer used

Calculated k-eff values

	KENO V.a	Shift
Annular hex array – ce_v7.1	1.0021 +/- 0.0018	0.999171 +/- 0.002285
Annular hex array – v7.1-252	1.0017 +/- 0.0022	0.999420 +/- 0.002026
Goslar cask – ce_v7.1	0.4859 +/- 0.0018	0.485404 +/- 0.002076
Goslar cask – v7.1-252	0.4930 +/- 0.0016	0.490907 +/- 0.002503
Castor cask – ce_v7.1	0.6286 +/- 0.0020	0.623442 +/- 0.002489
Castor cask – v7.1-252	0.6247 +/- 0.0016	0.625787 +/- 0.002036

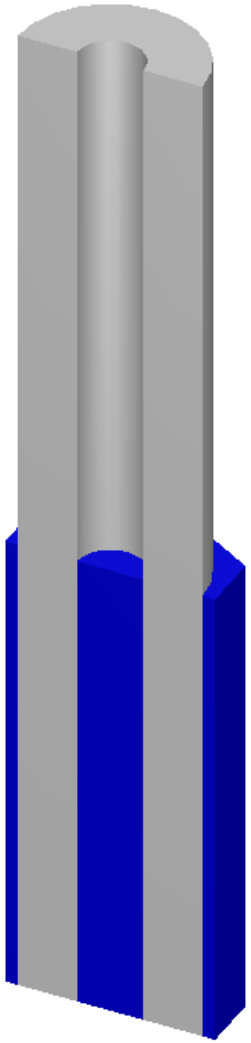
Your results may differ based on the options and computer used

Solution: Hexagonal Array - CE

```
=csas6
critical triangular pitched array of annular rods
ce_v7.1_endf
read composition
 uranium     1 den=19 1 293
                                92234 0.002
                                92235 1.95
                                92236 0.006
                                92238 98.042   end
 h2o         2 1 293   end
end composition
read parameter
 htm=yes
end parameter
```

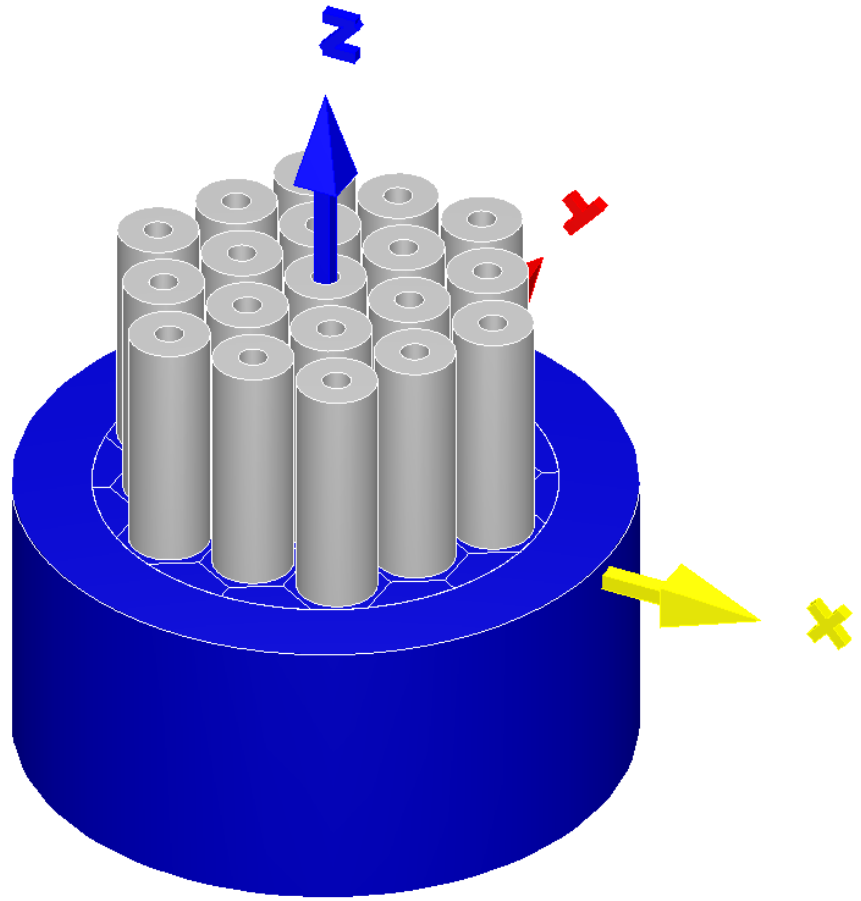
Solution: Hexagonal Array – CE (cont.)

```
read geometry
unit 1
com="flooded annular rod"
 cylinder 10    3.302     47.1          0
 cylinder 20    9.144     47.1          0
 hexprism 30   10.414     47.1          0
 cylinder 40    3.302     101.6          0
 cylinder 50    9.144     101.6          0
 hexprism 60   10.414     101.6          0
 media 2 1 10
 media 1 1 20 -10
 media 2 1 30 -20 -10
 media 0 1 40 -10
 media 1 1 50 -40 -20
 media 0 1 60 -50 -30
 boundary 60
```



Solution: Hexagonal Array – CE (cont.)

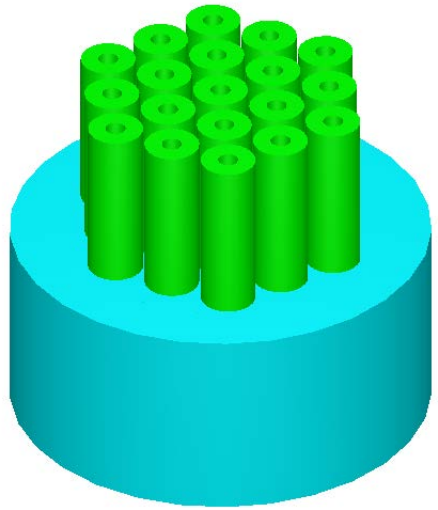
```
unit 2
 hexprism 10   10.414      47.1          0
 hexprism 20   10.414      101.6         0
 media 2 1 10
 media 0 1 20 -10
 boundary 20
global unit 5
com="7x7x1 hexprism array"
 cylinder 10        52     101.6          0
 cylinder 20        70      47.1        -18
 cylinder 30        70     101.6        -18
 array 1 10  place 4 4 1 0 0 0
 media 2 1 20 -10
 media 0 1 30 -20 -10
 boundary 30
end geometry
```



Solution: Hexagonal Array – CE (cont.)

```
read array
ara=1 nux=7 nuy=7 nuz=1 typ=shexagonal gbl=1
 fill
    2    2    2    2    2    2    2
    2    2    1    1    1    2    2
    2    2    1    1    1    1    2
    2    1    1    1    1    1    2
    2    2    1    1    1    1    2
    2    2    1    1    1    2    2
    2    2    2    2    2    2    2
  end fill
end array
end data
end
```

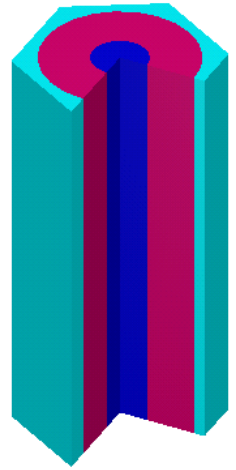
Solution: Hexagonal Array - MG

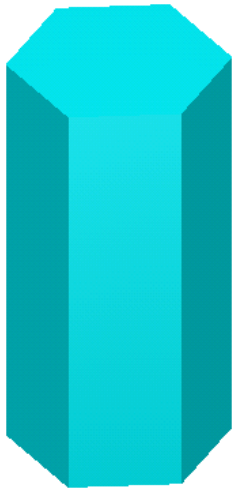


```
=csas6
critical triangular pitched array of annular rods
v7.1-252n
read composition
 uranium     1 den=19 1 293
              92235 1.95    92238 98.042   92234 0.002  92236 0.006   end
 h2o         2 1 293   end
 h2o         3 1 293   end
 uranium     4 den=19 1 293
              92235 1.95    92238 98.042   92234 0.002  92236 0.006   end
end composition
read celldata
  latticecell atriangpitch imodr=3.302 2 fuelr=9.144 1 hpitch=10.414 3 end
end celldata
```

Solution: Hexagonal Array – MG (Cont.)

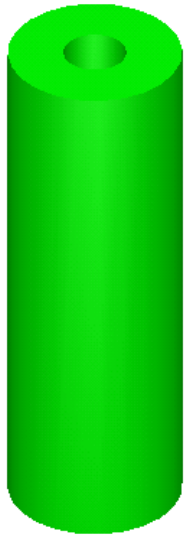
```
read geometry
unit 1
com='flooded annular rod'
 cylinder 10         3.302      47.1          0
 cylinder 20         9.144      47.1          0
 hexprism 30        10.414      47.1          0
 media 2 1 10
 media 1 1 20 -10
 media 3 1 30 -20
 boundary 30
unit 2
 hexprism 10    10.414      47.1          0
 media 3 1 10
 boundary 10
```

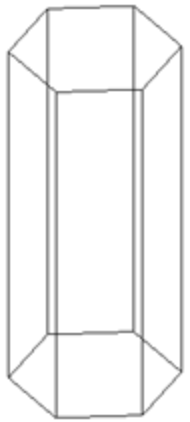




Solution: Hexagonal Array – MG (Cont.)

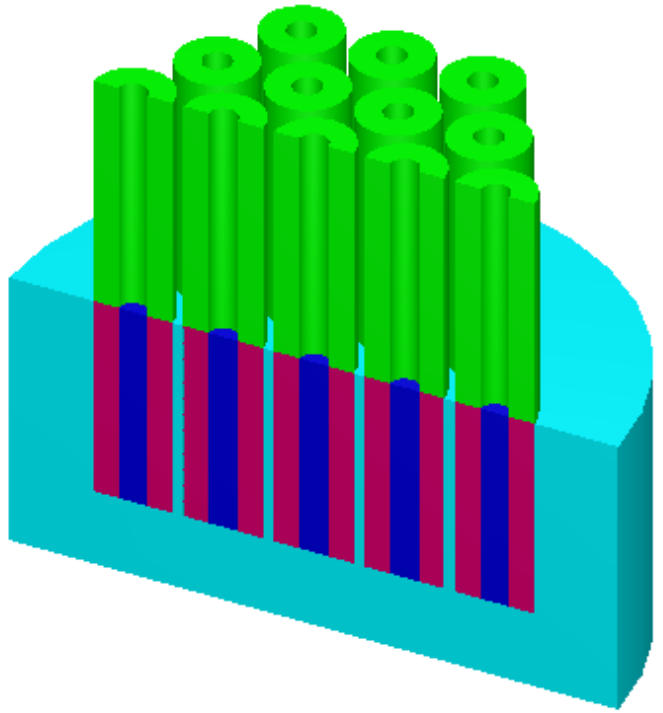
```
unit 3
com='void annular rod'
 cylinder 10        3.302     54.5         0
 cylinder 20        9.144     54.5         0
 hexprism 30       10.414     54.5         0
 media 0 1 10
 media 4 1 20 -10
 media 0 1 30 -20 -10
 boundary 30
unit 4
 hexprism 10   10.414     54.5         0
 media 0 1 10
 boundary 10
```





Solution: Hexagonal Array - MG (Cont.)

```
global unit 5
com='7x7x2 hexprism array'
 cylinder 10        52     101.6         0
 cylinder 20        70      47.1       -18
 cylinder 30        70     101.6       -18
 array 1 10  place 4 4 1 0 0 0
 media 3 1 20 -10
 media 0 1 30 -20
 boundary 30
end geometry
```



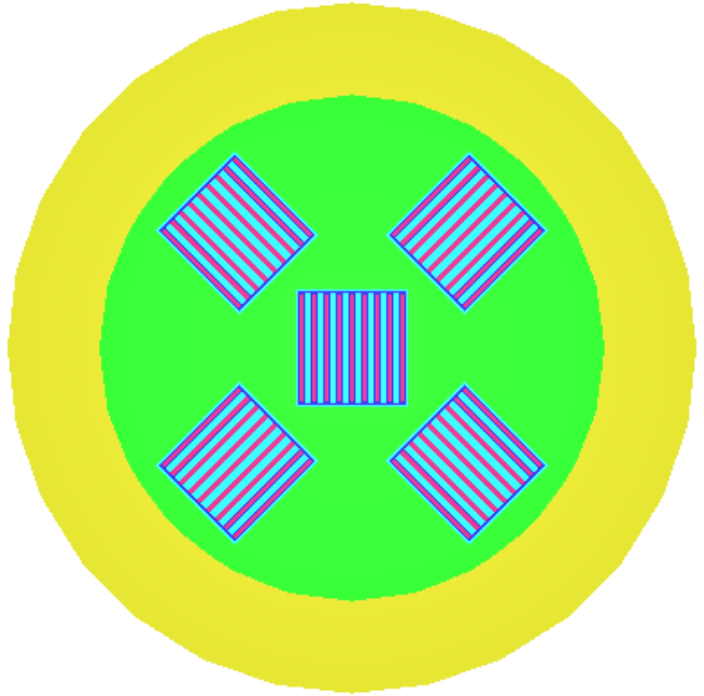
Solution: Hexagonal Array - MG (Cont.)

```
read array
ara=1 nux=7 nuy=7 nuz=2 typ=shexagonal
 fill
    2    2    2    2    2    2    2
       2    2    1    1    1    2    2
    2    2    1    1    1    1    2
       2    1    1    1    1    1    2
    2    2    1    1    1    1    2
       2    2    1    1    1    2    2
    2    2    2    2    2    2    2
```

Solution: Hexagonal Array - MG (Cont.)

```
      4     4     4     4     4     4     4
         4     4     3     3     3     4     4
      4     4     3     3     3     3     4
         4     3     3     3     3     3     4
      4     4     3     3     3     3     4
         4     4     3     3     3     4     4
      4     4     4     4     4     4     4
  end fill
end array
end data
end
```

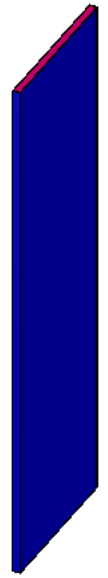
Solution: Goslar Cask

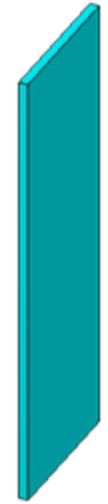


```
=csas6
baby goslar fuel cask  92.85 w% u-235
ce_v7.1_endf
read composition
 uranium     1 den=3.78 0.0664 293
                                92235 92.85
                                92238 7.15   end
 al          1 den=3.78 0.9336 293   end
 al          2 den=2.7 1 293   end
 h2o         3 1 293   end
 al          4 den=2.7 0.1 293   end
 ss304       5 1 293   end
end composition
read celldata
  latticecell symmslabcell fuelr=0.2 1 cladr=0.25 2 hpitch=0.55 3 end
end celldata
```

Solution: Goslar Cask (Cont.)

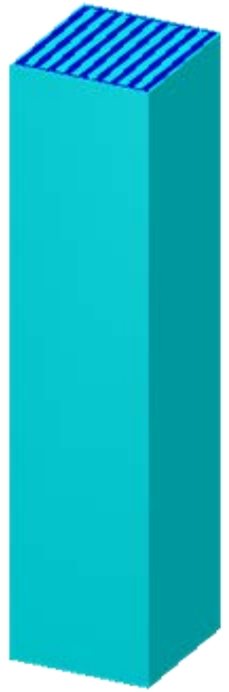
```
read geometry
unit 1
com='fuel slab'
 cuboid 10      0.2      -0.2       4.7       -4.7         30         -30
 cuboid 20     0.25     -0.25     4.75     -4.75     30.05    -30.05
 media 1 1 10
 media 2 1 20 -10
 boundary 20
unit 2
com='h2o slab'
 cuboid 10      0.3      -0.3      4.75     -4.75     30.05    -30.05
 media 3 1 10
 boundary 10
```





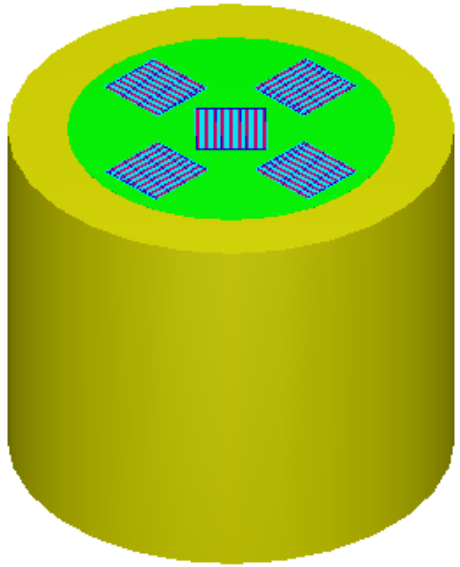
Solution: Goslar Cask (Cont.)

```
unit 3
com='single assembly'
 cuboid 10     4.65    -4.65     4.75    -4.75    30.05   -30.05
 cuboid 20     4.65    -4.65     4.95    -4.95    30.05   -30.05
 cuboid 30     4.85    -4.85     5.15    -5.15    30.05   -30.05
 array 1 10  place 9 1 1 0 0 0
 media 2 1 20 -10
 media 3 1 30 -20 -10
 boundary 30
```



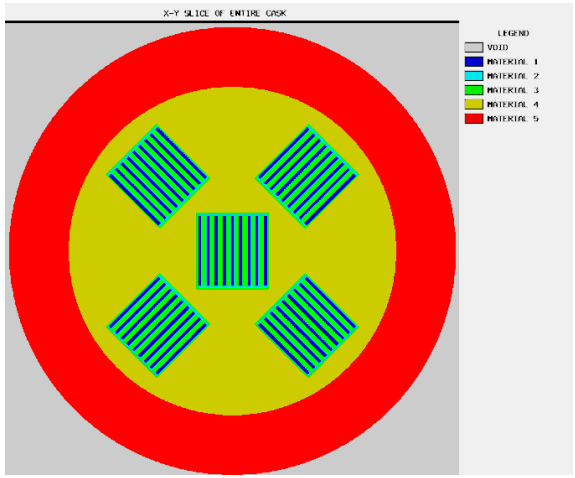
Solution: Goslar Cask (Cont.)

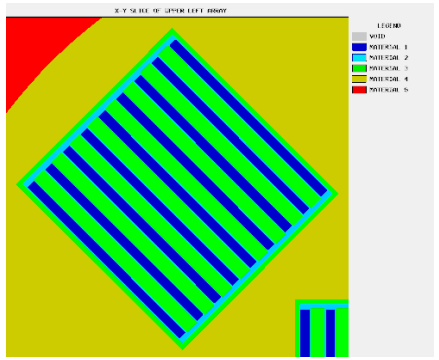
```
global unit 10
 cylinder 10       22       40      -40
 hole 3
 hole 3   origin  x=10 y=10 z=0     rotate  a1=-45 a2=0 a3=0
 hole 3   origin  x=10 y=-10 z=0    rotate  a1=-135 a2=0 a3=0
 hole 3   origin  x=-10 y=-10 z=0   rotate  a1=135 a2=0 a3=0
 hole 3   origin  x=-10 y=10 z=0    rotate  a1=45 a2=0 a3=0
 cylinder 20     30       50      -50
 cuboid 30       31      -31       31      -31       51      -51
 media 4 1 10
 media 5 1 20 -10
 media 0 1 30 -20
 boundary 30
end geometry
read array
ara=1 nux=17 nuy=1 nuz=1 typ=square
 fill    1  2  1  2  1  2  1  2  1  2  1  2  1  2  1  2  1
 end fill
end array
```

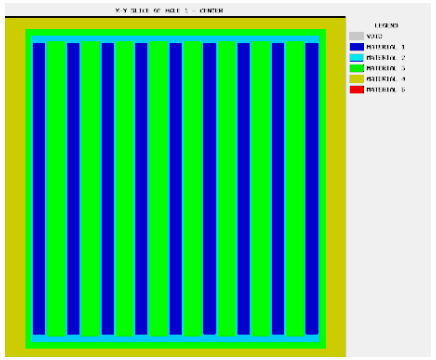


Solution: Goslar Cask (Cont.)

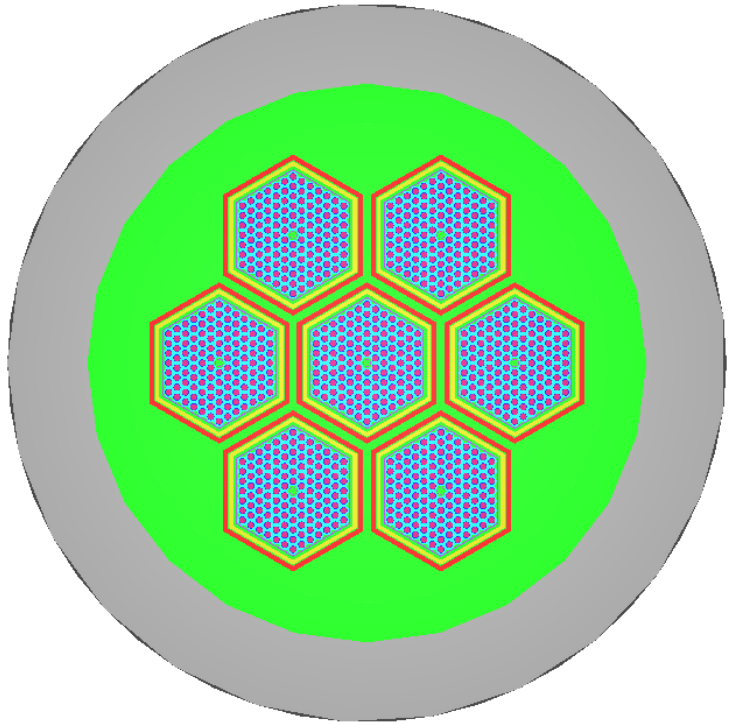
```
read plot
ttl='x-y slice of entire cask'
 xul=-30.5  yul=30.5  zul=0
 xlr=30.5    ylr=-30.5  zlr=0
 nax=800   uax=1 vdn=-1  end
ttl='x-y slice of hole 1 - center'
 xul=-5.5  yul=5.5  zul=0
 xlr=5.5   ylr=-5.5  zlr=0
 nax=800 uax=1 vdn=-1  end
ttl='x-y slice of upper left array'
 xul=-17.5 yul=17.5  zul=0
 xlr=-2.5    ylr=2.5     zlr=0
 nax=800  uax=1 vdn=-1  end
end plot
end data
end
```







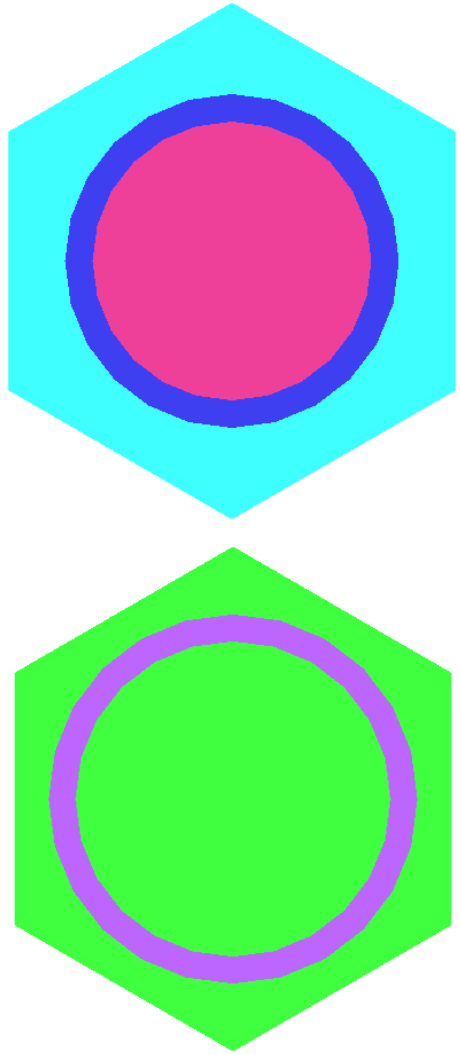
Solution: Castor Cask



```
=csas6
castor 440/84 - wet infinite lattice
ce_v7.1_endf
read composition
 uo2          1 den=10.192 1 293
                                  92235 3.6
                                  92238 96.4   end
 zirconium    2 den=6.088 1 293   end
 h2o          3 1 293   end
 h2o          4 1 293   end
 b            5 den=0.078 1 293   end
 ss304        5 den=8.2377 1 293   end
 al           6 den=2.7 1 293   end
 zirconium    7 den=6.4369 1 293   end
 ss304        8 1 293   end
end composition
read celldata
  latticecell triangpitch fuelr=0.38 1 cladr=0.455 2 hpitch=0.61 3 end
end celldata
```

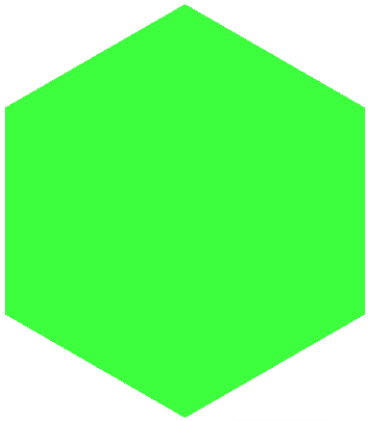
Solution: Castor Cask (Cont.)

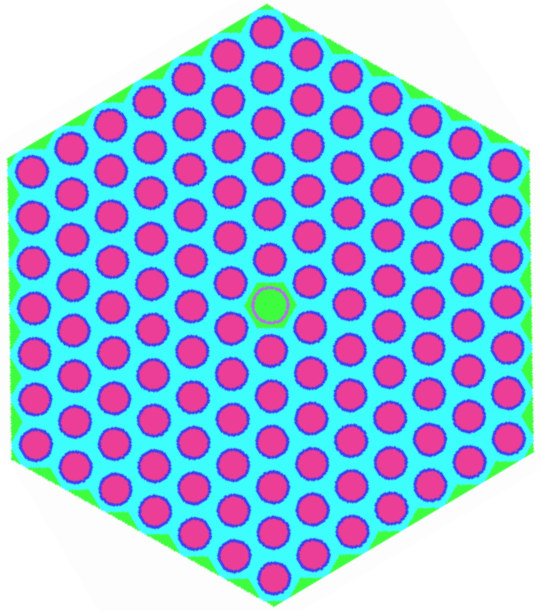
```
read geometry
unit 1
com='fuel pin'
 cylinder 10      0.38      242         0
 cylinder 20      0.455     242         0
 hexprism 30      0.61      242         0
 media 1 1 10
 media 2 1 20 -10
 media 3 1 30 -20
 boundary 30
unit 2
com='central instrumental tube'
 cylinder 10     0.44       242         0
 cylinder 20     0.515      242         0
 hexprism 30     0.61       242         0
 media 4 1 10
 media 7 1 20 -10
 media 4 1 30 -20
 boundary 30
```



Solution: Castor Cask (Cont.)

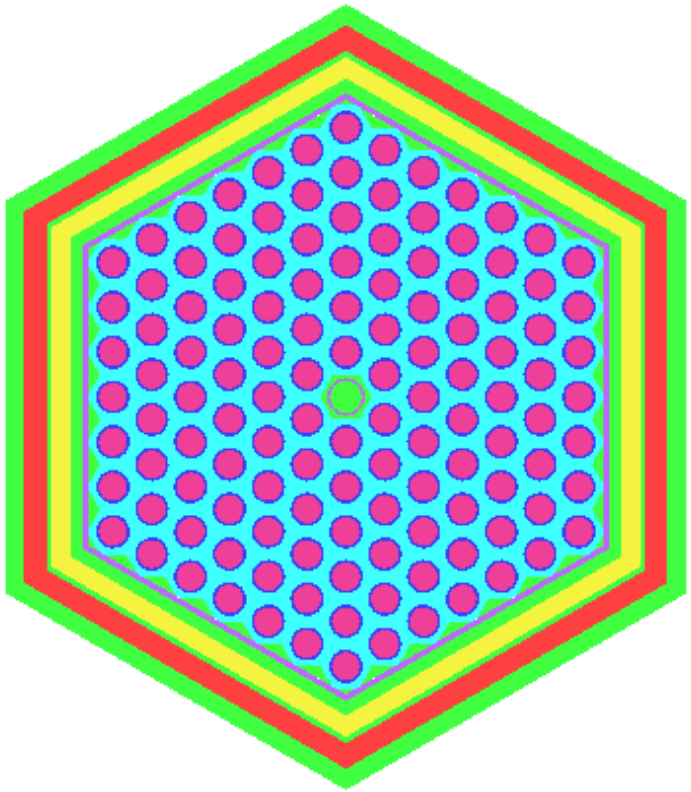
```
unit 3
com='empty cell used to fill array'
 hexprism 10     0.61      242         0
 media 4 1 10
 boundary 10
unit 4
com='hexagonal unit of the lattice'
 hexprism 10    7   242   0   rotate  a1=30 a2=0 a3=0
 array 1 10  place 8 8 1 0 0 0
 boundary 10
```





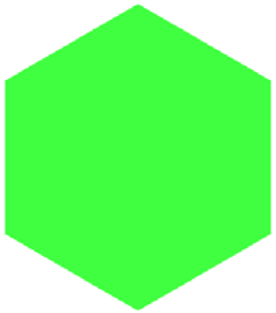
Solution: Castor Cask (Cont.)

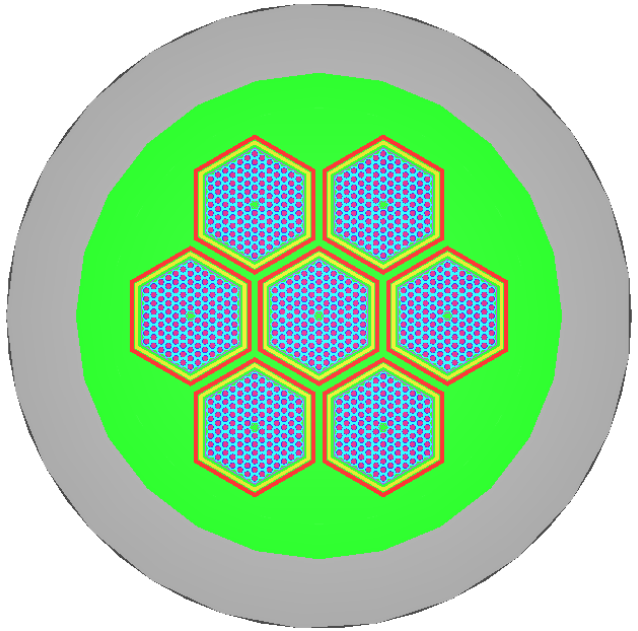
```
unit 5
hexprism 20      7.15      242          0
hexprism 30       7.5      242          0
hexprism 40         8      242          0
hexprism 50      8.15      242          0
hexprism 60      8.75      242          0
hexprism 70      9.25      242          0
hole 4   rotate  a1=-30 a2=0 a3=0
media 7 1 20
media 4 1 30 -20
media 5 1 40 -30
media 4 1 50 -40
media 6 1 60 -50
media 4 1 70 -60
boundary 70
```



Solution: Castor Cask (Cont.)

```
unit 6
 hexprism 10      9.25        242          0
 media 4 1 10
 boundary 10
global unit 7
 cylinder 10         30        242          0
 cylinder 20         35        247         -5
 cylinder 30         45        257        -15
 array 2 10  place 3 3 1 0 0 0
 media 4 1 20 -10
 media 8 1 30 -20
 boundary 30
end geometry
```





Solution: Castor Cask (Cont.)

```
read array
ara=1 nux=15 nuy=15 nuz=1 typ=triangular
 fill
    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
    3    3    3    3    3    3    3    1    1    1    1    1    1    1    3
    3    3    3    3    3    3    1    1    1    1    1    1    1    1    3
    3    3    3    3    3    1    1    1    1    1    1    1    1    1    3
    3    3    3    3    1    1    1    1    1    1    1    1    1    1    3
    3    3    3    1    1    1    1    1    1    1    1    1    1    1    3
    3    3    1    1    1    1    1    1    1    1    1    1    1    1    3
    3    1    1    1    1    1    1    2    1    1    1    1    1    1    3
    3    1    1    1    1    1    1    1    1    1    1    1    1    3    3
    3    1    1    1    1    1    1    1    1    1    1    1    3    3    3
    3    1    1    1    1    1    1    1    1    1    1    3    3    3    3
    3    1    1    1    1    1    1    1    1    1    3    3    3    3    3
    3    1    1    1    1    1    1    1    1    3    3    3    3    3    3
    3    1    1    1    1    1    1    1    3    3    3    3    3    3    3
    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
 end fill
```

Solution: Castor Cask (Cont.)

```
ara=2 nux=5 nuy=5 nuz=1 typ=triangular
 fill
    6    6    6    6    6
        6    6    5    5    6
           6    5    5    5    6
              6    5    5    6    6
                 6    6    6    6    6
 end fill
end array
read start
 nst=1
 xsm=-5  xsp=5  ysm=-5  ysp=5  zsm=0 zsp=242
  end start
end data
end
```

Summary of Module 4

- Overview of KENO VI arrays
- Can simplify geometry construction with repeated structures
- Consists of the following components:
  - Array Keyword
  - Array Type
  - Fill Statement

OAK RIDGE National Laboratory

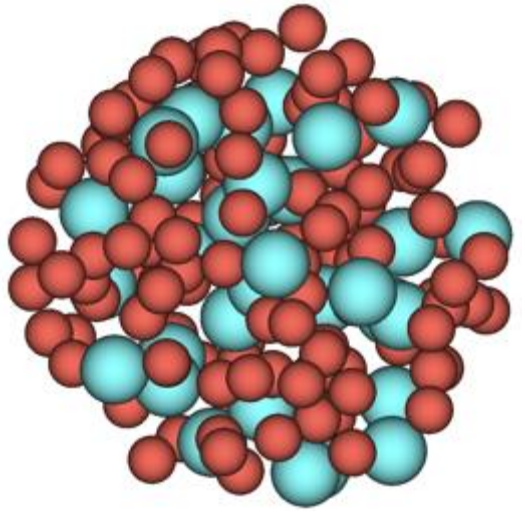
Sub-module 5

- Randomgeom block



Randomgeom block

- Block available with Shift (not with KENO)
- Enables straightforward generation of reference solutions for stochastic geometry (TRISO fuel particles, FCM, etc.)
- Current capabilities:
  - Automatic generation of random distribution of particles within a geometry
  - Supports multiple particle types with a geometry
  - Placement into spheres, cylinders, and cuboids
  - Visualization in Fulcrum
- Limitations:
  - Distribution of spherical particles
  - Can only be used with a single region
  - Maximum packing fraction of ~20%



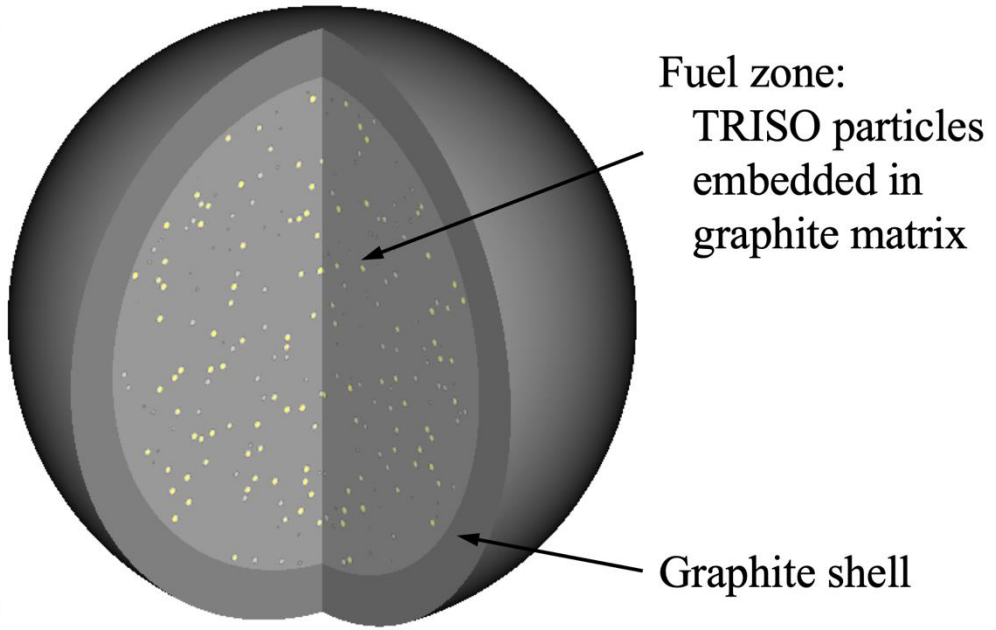
Random distribution of 2 particle types

Randomgeom block input

```
read geometry
unit 1 com='TRISO'
  sphere 1 2.50e-02
  sphere 2 3.40e-02
  …
  media 100 1 1
  media 101 1 2 -1
  …
  boundary 5
Global unit 10
com='pebble'
  sphere 1 2.5
  sphere 2 3.0
  cuboid 3 6p3.0
  media 105 1 1 randommix='trisos'
  media 106 1 2 -1
  media 300 1 3 -2
  boundary 3
end geometry
```

```
read randomgeom
  randommix = 'trisos'
    type=random
    units=1 end
    pfs=0.10 end
    clip=no
    seed=1111
  end randommix
end randomgeom
```

Fuel pebble with random distribution of TRISO particles



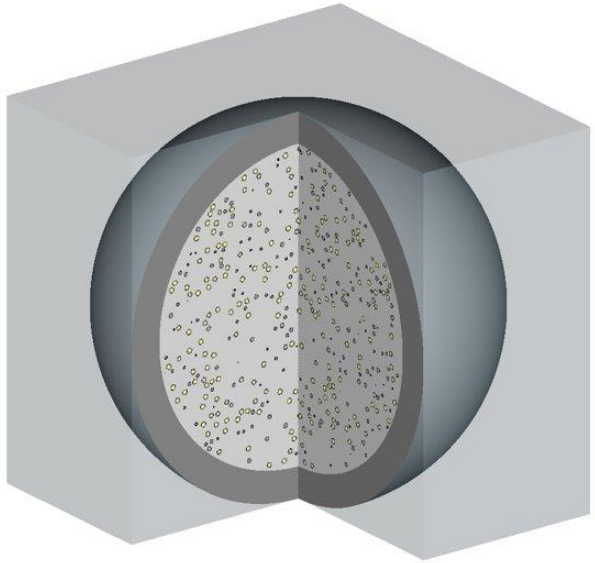
OAK RIDGE National Laboratory

Exercises for sub-module 5

- Randomgeom block



Problem: HTGR pebble



Goal: Develop HTGR fuel pebble model

1. Use "Advaced_Reactors/ htgr_pebble_ce_starter.inp" and save as "htgr_pebble_ce.inp"
2. Add geometry and randomgeom block based on details on the right
3. Run calculation and observe runtime

Pebble design details

Pebble outer radius: 3.0 cm
Thickness of fuel-free zone: 0.5 cm
Cube half pitch: 3.0 cm
Fuel kernel diameter: 500 μm
Kernel material: UO_2 with enrichment of 9.6%
Kernel density: 10.4 g/cm^3
Kernel coating material: C/C/SiC/C
Kernel coating thickness: 95/40/35/40 μm
Kernel coating densities: 1.05/1.90/3.18/1.90 g/cm^3
Matrix and shell materials: C
Matrix and shell densities: 1.74
Number of TRISOs per pebble: 15,000
Room temperature

Solution: HTGR pebble

```
' ==============================
'   Geometry
' ==============================
read geometry

' ================
'   TRISO particle
' ================
unit 1
 sphere  1  0.0250
 sphere  2  0.0345
 sphere  3  0.0385
 sphere  4  0.0420
 sphere  5  0.0460
 media  10  1  1
 media  11  1  2  -1
 media  12  1  3  -2
 media  13  1  4  -3
 media  14  1  5  -4
 boundary  5

' ================
'   fuel pebble
' ================
global unit 10
 sphere 10 2.5000
 sphere 11 3.0
 cuboid 12 6p3
 media  15  1  10  randommix='trisos'
 media  16  1  11  -10
 media  17  1  12  -11
 boundary 12

end geometry
```

- Geometry:
  - Convert TRISO layer thicknesses to radii
  - Pebble outer radius equals cuboid half length
- Randomgeom:
  - Determine packing fraction from 15,000 TRISO particles with outer radius 0.046 cm in a fuel zone of 2.5 cm radius
  - Define name and use name to fill TRISO particles into fuel zone within the geometry

```
read randomgeom
  randommix = 'trisos'
  type=random
  units=1 end
  pfs=0.09344256 end
  clip=no
  seed=1111
  end randommix
end randomgeom
```

Summary of Module 5

- Introduction to Shift randomgeom capability
- Baseline random geometry capability
  - Limited in volume fraction
  - Limited in allowable shapes
- TRISO pebble example

OAK RIDGE National Laboratory

KENO-VI geometry verification



Tips to verify model

- Use Fulcrum to develop your model and address validation errors
- Visually inspect your geometry with Fulcrum
- Run your geometry using sequence parameter "parm=(check)"
- Calculate volumes of your regions and mixtures and compare them to the expectation
- Common geometry issues:
  - Undefined space
  - Doubly defined space
  - Array incorrectly placed
  - Units in array are of inconsistent outer shape
  - Overlapping holes

OAK RIDGE National Laboratory

Multigroup Cross Section Processing

SCALE 6.3

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning Goals

- Discuss resonance self-shielding treatments in SCALE
- Identify situations in which to apply self-shielding methods in SCALE
- Understand methods to create problem-dependent resonance self-shielding of multigroup cross sections in SCALE

Outline

- Sub-module 1: Multigroup processing theory
- Sub-module 2: Multigroup processing input
- Sub-module 3: Multigroup processing exercises
- Sub-module 4: Non-uniform lattices
- Sub-module 5: Non-uniform lattices exercises
- Sub-module 6: Double-heterogeneous systems
- Sub-module 7: Double-heterogeneous exercises
- Summary

OAK RIDGE National Laboratory

Sub-module 1

- Multigroup processing theory



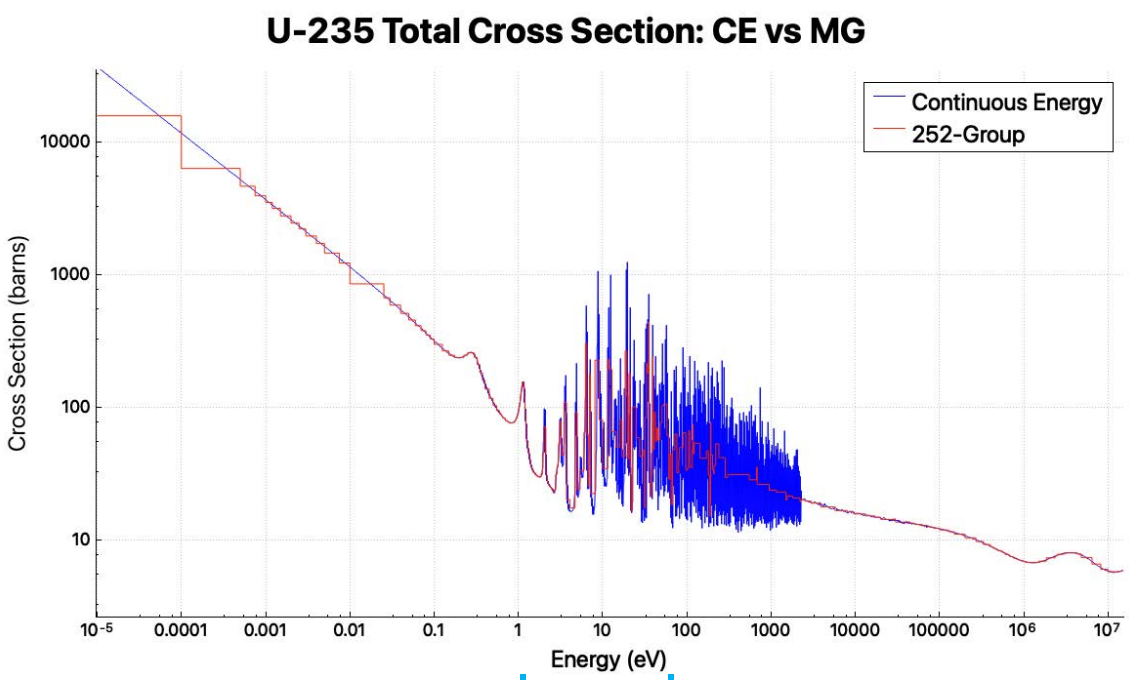
OAK RIDGE National Laboratory

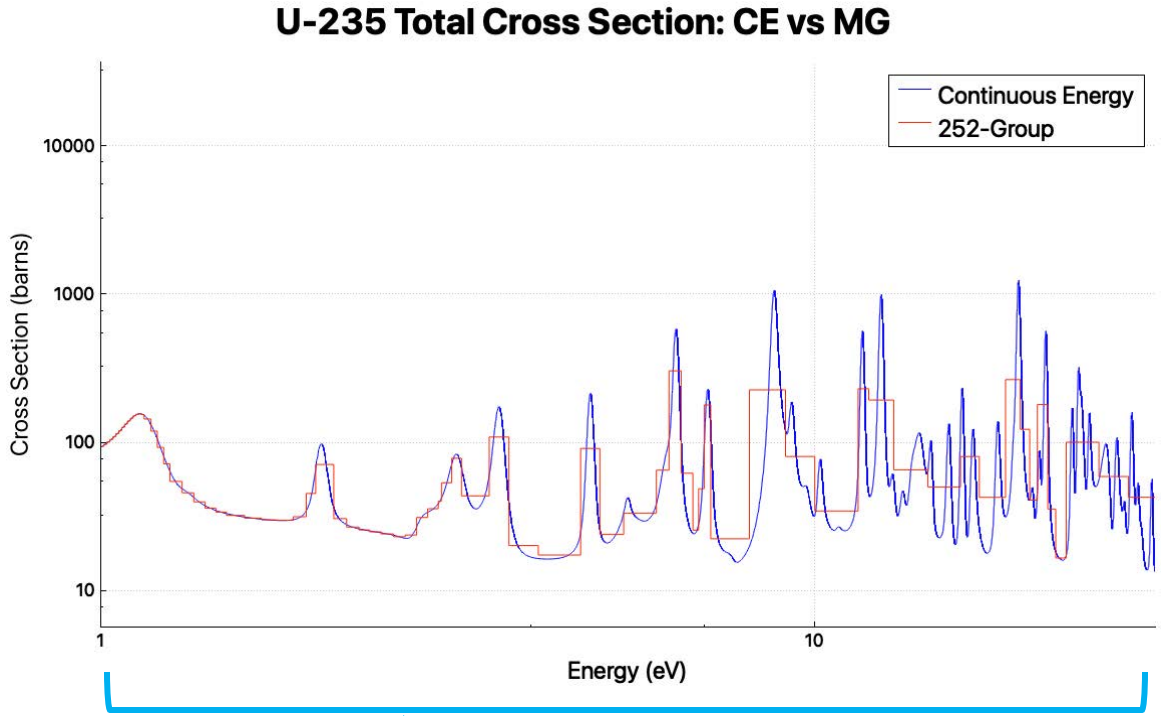
Multigroup problem-dependent resonance self-shielded cross section data



SCALE capabilities: Cross section processing

- Continuous energy and multigroup libraries are distributed with SCALE
- When using multigroup libraries, *cross section processing* is performed to account for resonance self-shielding effects





Types of nuclear data libraries in SCALE

- Pointwise (PW)
  - Tabulated values of (E_i,σ_i) on energy mesh
  - Used by CENTRM [for self-shielding] and by CE Monte Carlo transport
  - Cross section processing is <u>not needed</u> for CE data because pointwise data are used *without averaging*
- Multigroup (MG)
  - Flux-weighted <u>*averages*</u> of PW data over an energy group
  - Problem-independent MG libraries are distributed with SCALE
  - Self-shielding calculations during the cross section processing convert problem-*independent* into problem-*dependent* libraries for transport calculations

Interpretation of microscopic multigroup cross sections

Number density

Flux

$$R(E) = N\,\sigma(E) \cdot \Phi(E)$$

Reaction rate

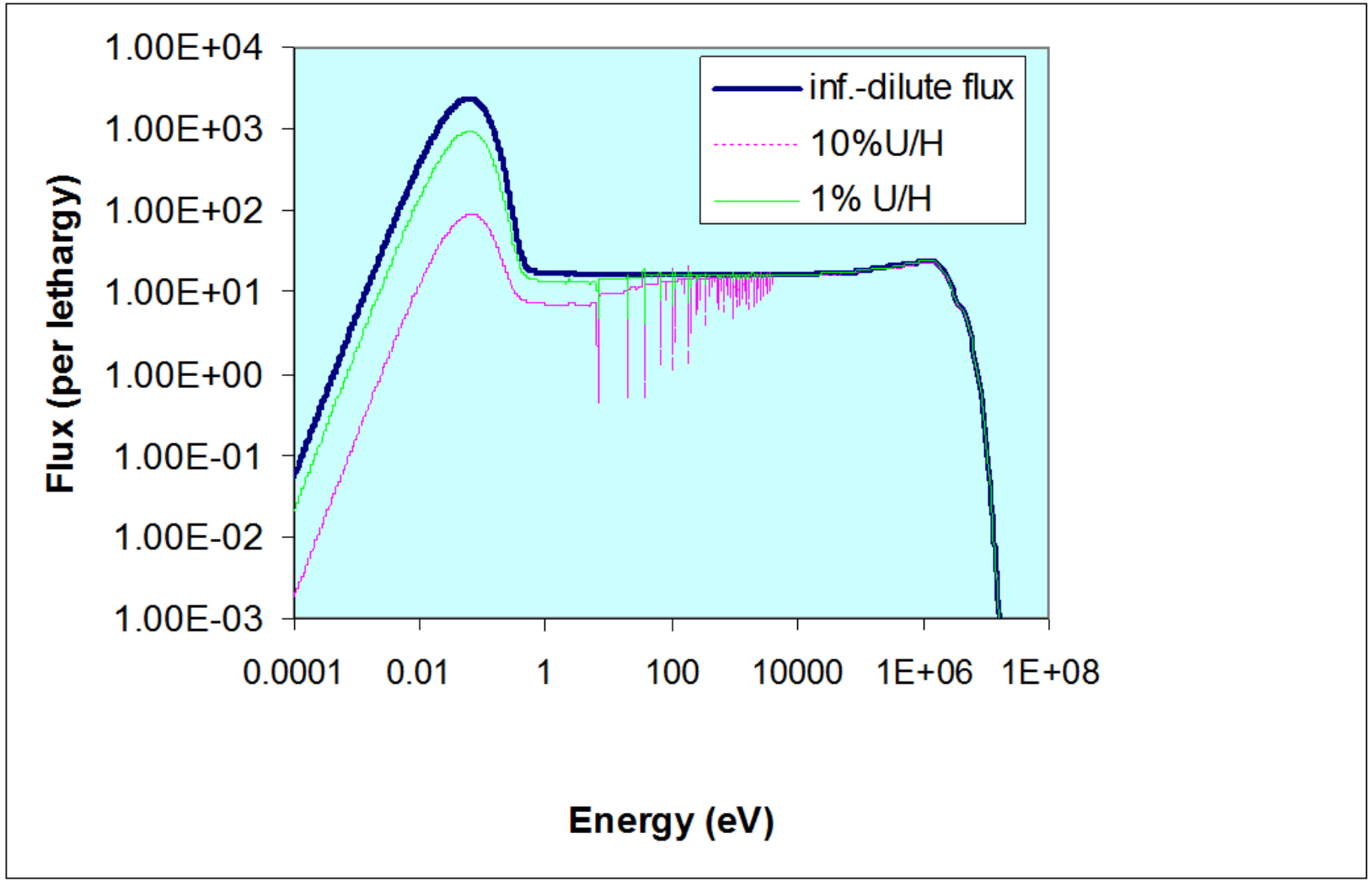
Micro XS

$$\sigma(E) = \frac{\left(\frac{R(E)}{N}\right)}{\Phi(E)}$$

Reaction rate per atom in target

Incident neutron flux with energy E

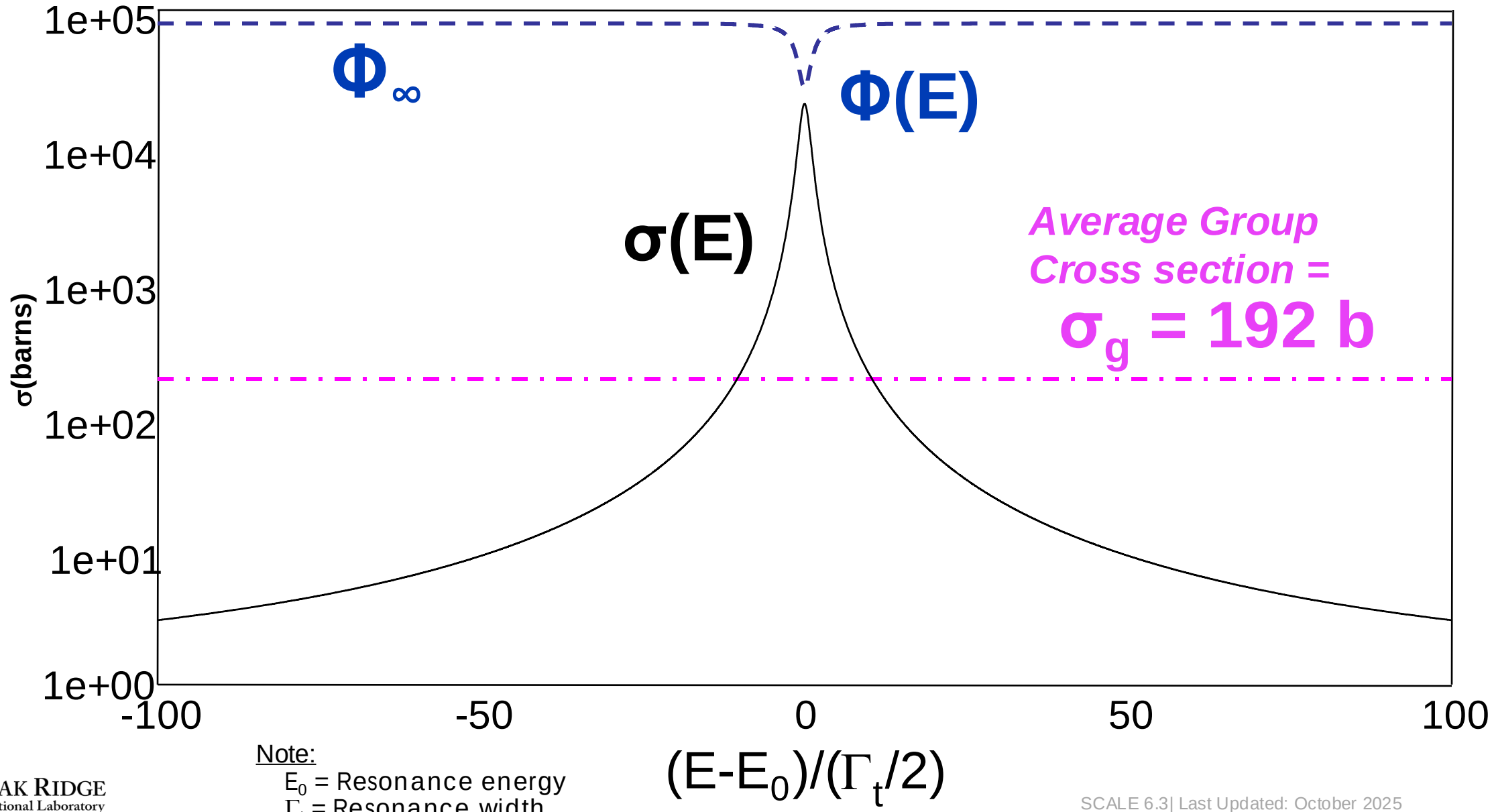
Impact of resonance absorption on neutron spectrum



Flux is directly affected by the amount of resonance absorption

Highly dilute absorber

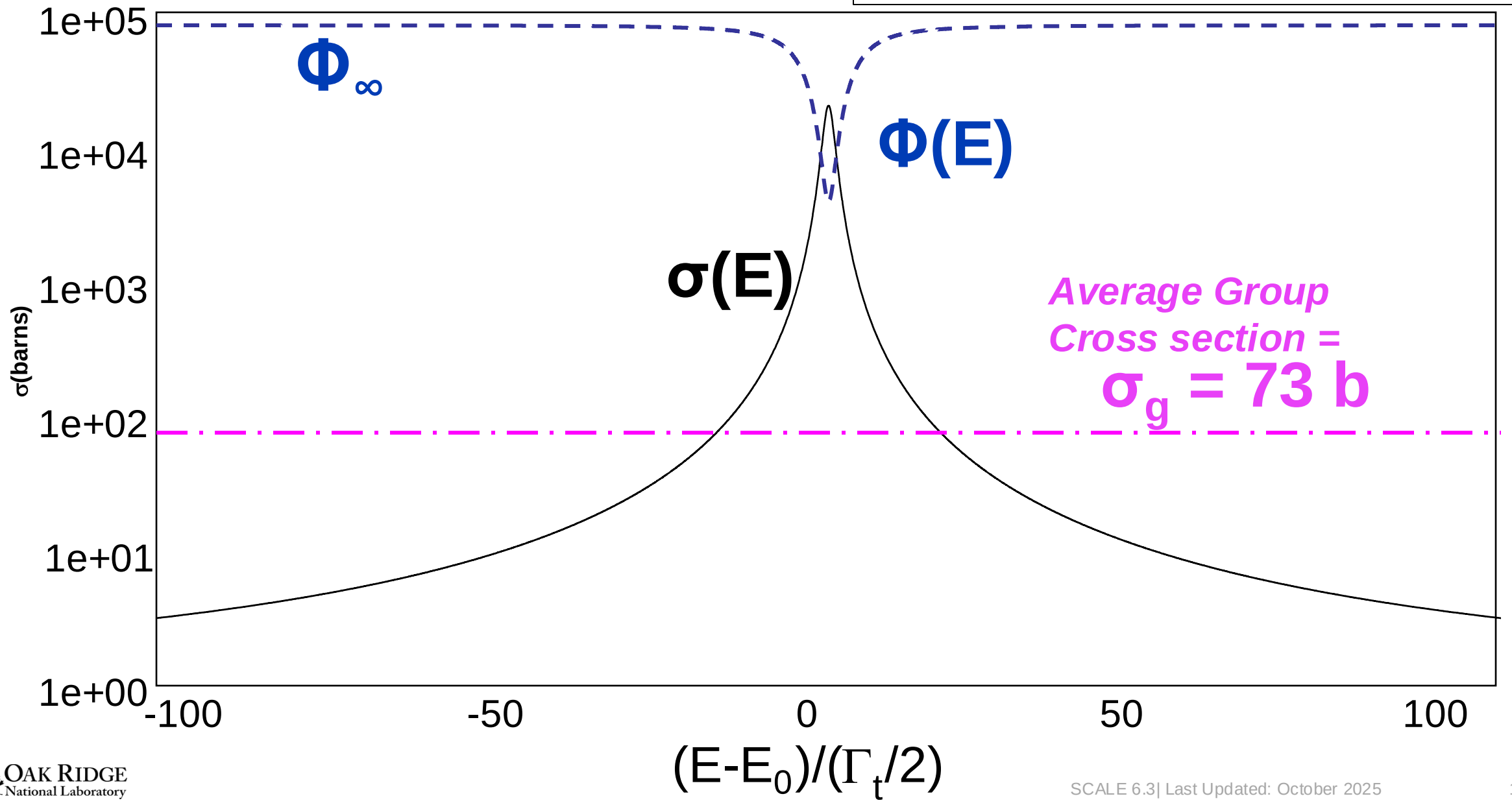
Minimal effect on flux and average group cross section



Note:
E_0 = Resonance energy
Γ_t = Resonance width

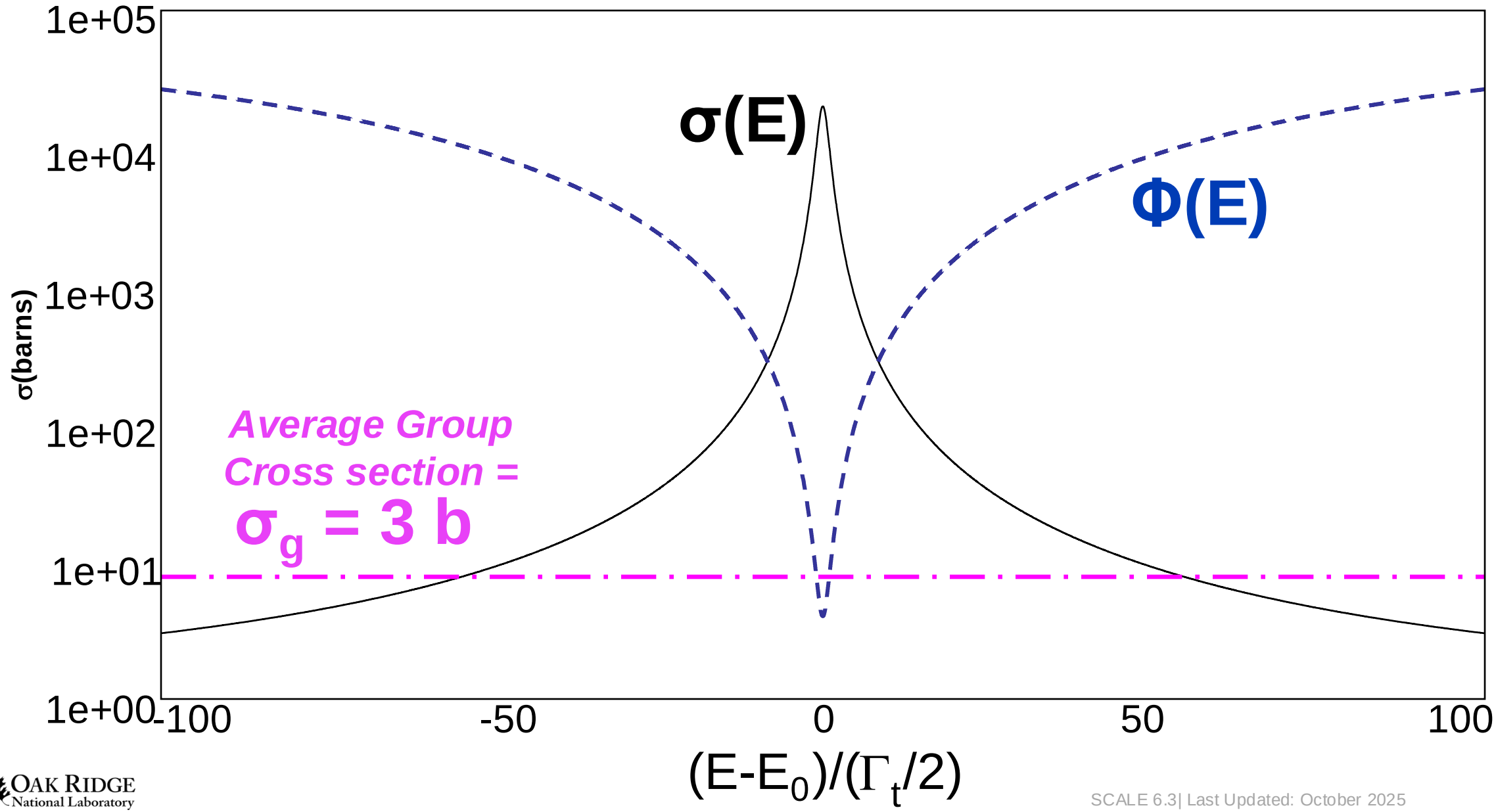
Moderately dilute absorber

Increased effect on flux causes average group cross section to be decreased compared to the highly dilute absorber

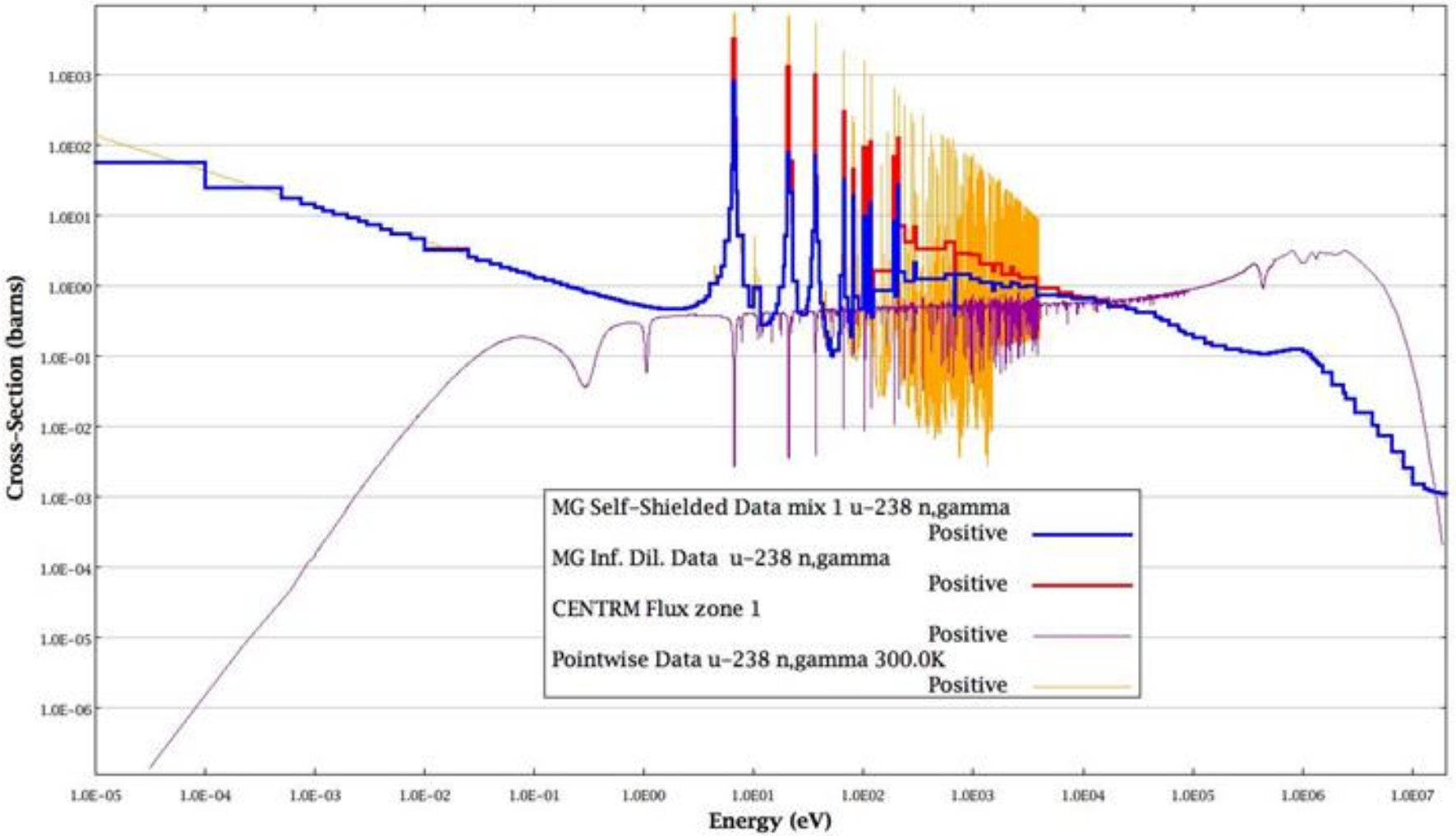


Undiluted absorber (pure absorber)

Dramatic effect on flux and average group cross section



Problem-dependent cross section preparation for MOX fuel



Why do we care?

- As the concentration of the resonance nuclide increases, self-shielding increases
- As the self-shielding increases, the microscopic multigroup cross section decreases
- If the absorption cross section decreases, k_{eff} is impacted

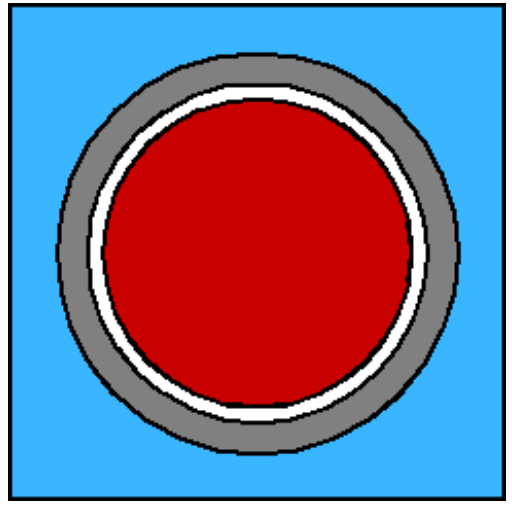
Example: LWR fuel pin eigenvalue result

1. $k_\infty = 1.26$

   with adequate cross section processing considering the LWR fuel pin lattice

2. $k_\infty = 1.32$

   assuming that each mixture is an independent, infinite homogeneous medium



LWR fuel pin

$\Delta k = 6{,}000$ pcm difference between calculations 2 and 1

In multigroup calculations, improper resonance self-shielding, i.e. incorrect cross section processing, will produce incorrect results!

OAK RIDGE National Laboratory

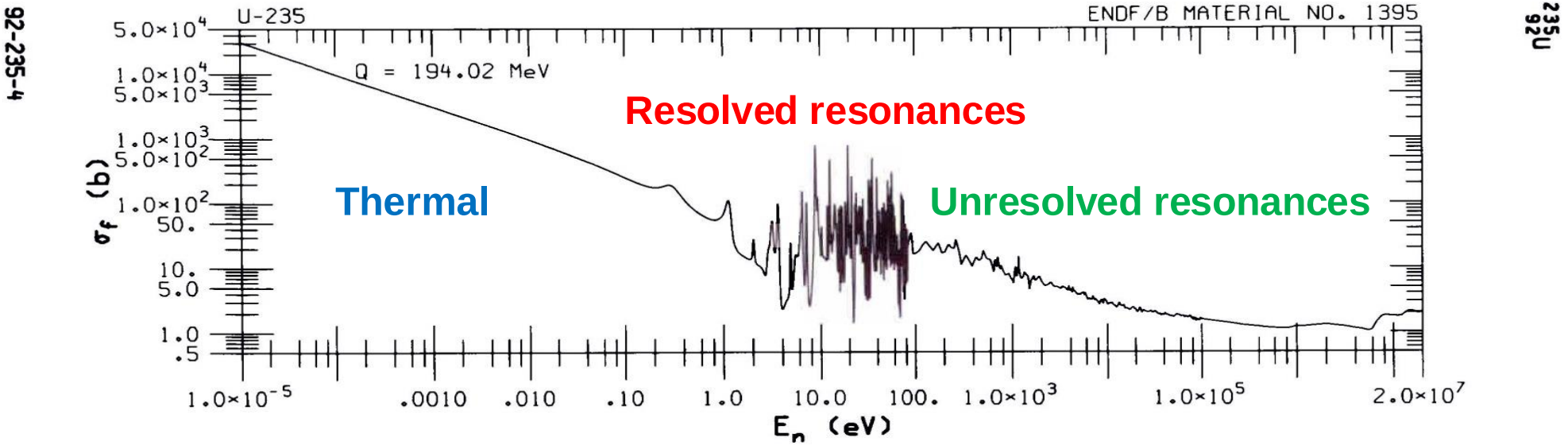
Problem-dependent resonance self-shielding methods in SCALE



SCALE approach to resonance self-shielded MG cross sections

Calculation of the neutron flux Φ(E) and self-shielded MG cross sections on 1D and 2D cells in three different neutron energy ranges:

- Thermal range: 0.00001 eV - 0.001 eV
- Resolved resonance region: 0.001 eV - 20 keV
- Unresolved resonance region: 20 keV - 20MeV



Self-shielding methods and modules in SCALE/XSProc

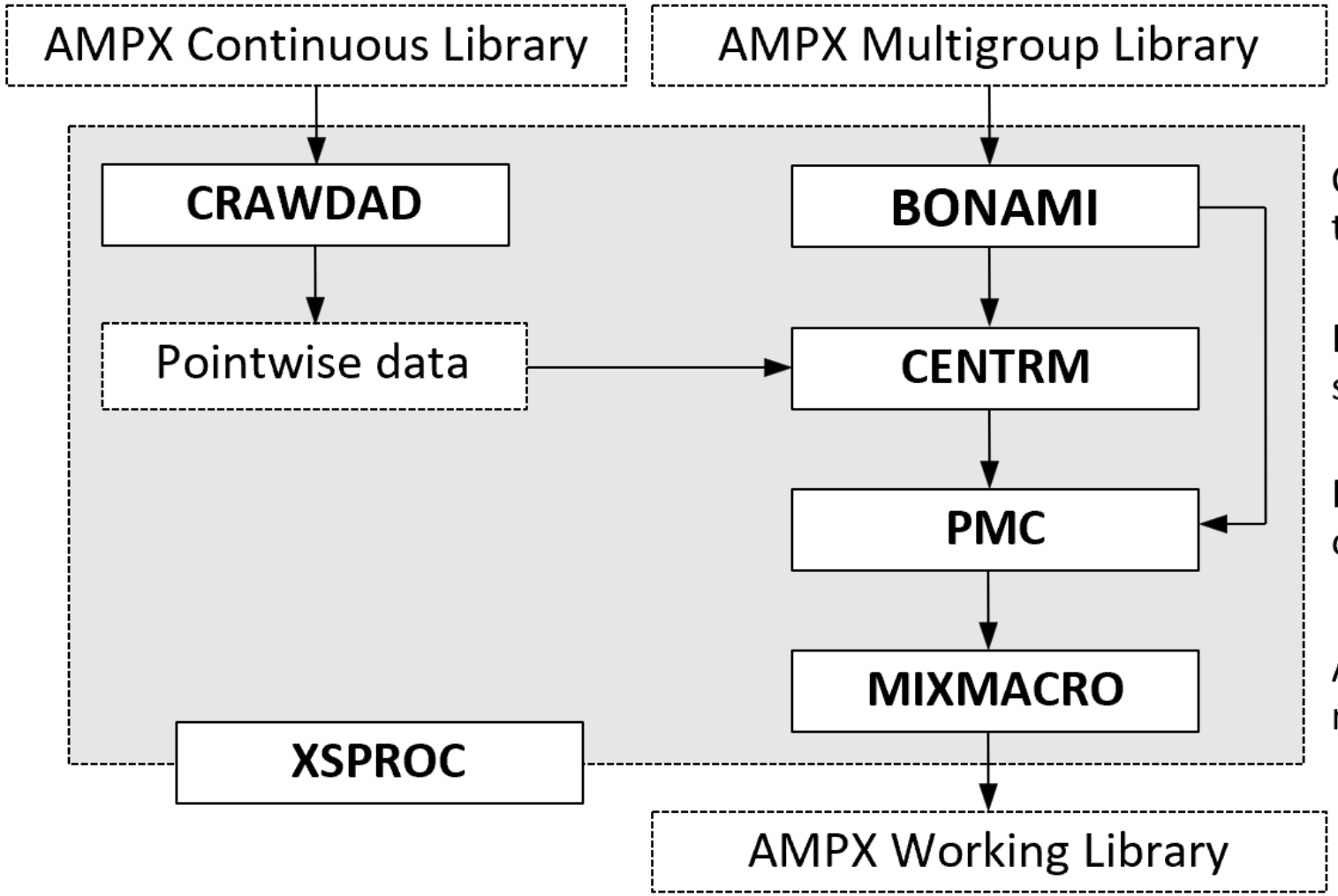
- XSProc/BONAMI:
  - Method: Bondarenko
  - Most approximate method
  - Performed over the entire energy range
  - Execution time very fast (seconds per cell)
- XSProc/CENTRM:
  - MG transport for the thermal, unresolved and fast energy range
  - Pointwise transport for the resolved energy range
  - Available methods: 1D Sn, 2D MOC, space-independent (0D) slowing down, and simplified two-region collision-probability method
  - Most rigorous models in both energy and geometry effects
  - Requires longer execution time (seconds to minutes per cell)

- Default:
  - Thermal, unresolved resonance, and fast energy range: BONAMI
  - Resolved resonance region: CENTRM 2D MOC (for 2d square pin cell lattices) or 1D Sn (for all other cells)

In practice, SCALE/XSProc performs a BONAMI calculation for the full energy range. Afterwards, a CENTRM calculation is performed and the self-shielded MG cross sections from CENTRM are used to replace BONAMI's MG cross sections in the resolved energy range.

BONAMI should only be used for the full energy range when testing models or capabilities. The default combination of CENTRM and BONAMI is required to obtain the most accurate self-shielded MG cross sections.

XSProc simplified flowchart



Cross section processing based on the Bondarenko method

Problem-dependent pointwise slowing down calculation

Flux-weighted self-shielded 1D cross-sections

Adjustment of the 2D scattering matrices

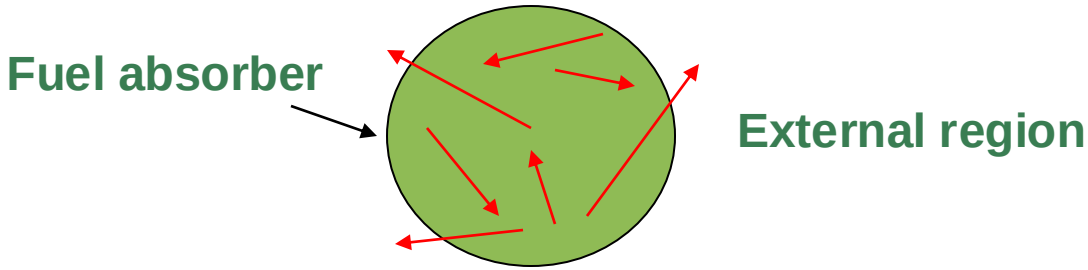
Degree of self-shielding depends upon system characteristics

- Homogeneous
  - Consists of uniform mixture of absorber (fuel) + moderator
  - Mean free path is << characteristic dimensions, OR
    >> characteristic dimensions
  - Resonance absorption depends mainly on relative amounts of absorber vs. moderator in composition
- Heterogeneous
  - Absorber (fuel) is "lumped" in pins, plates, etc. separated by moderator
  - Neutron mean free path is on the order of the characteristic dimension of lump (e.g., fuel pin lattice pitch)
  - Resonance absorption depends on geometry as well as material composition

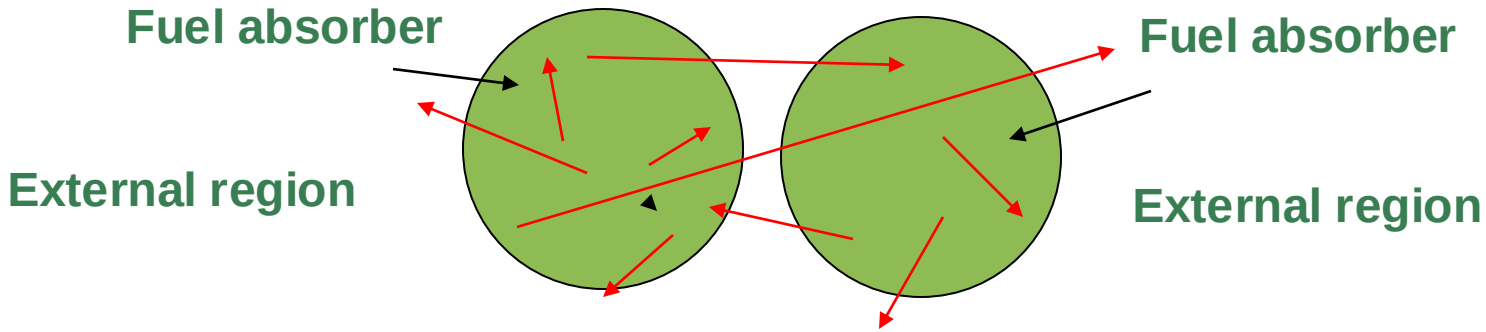
Solutions require escape probabilities from absorber region

A: absorber (fuel)
M: moderator

- Isolated lump
  - $P^{(A \rightarrow M)} \rightarrow P_0 \equiv$ isolated escape probability



- Multiple lumps (e.g., fuel lattice)
  - $P^{(A \rightarrow M)} \rightarrow P_{esc} \equiv$ collective escape probability from all absorber bodies

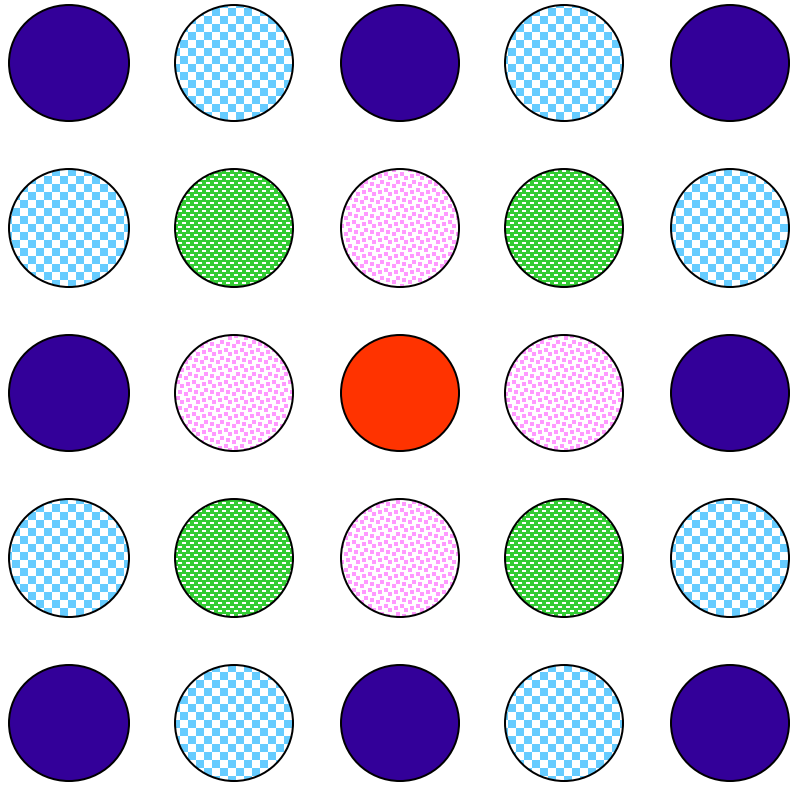


Solutions that require Dancoff factors

- In an isolated fuel pin, some of the neutrons that leave the pin without colliding may escape into the moderator
- In a fuel lattice, many of the neutrons enter a neighboring pin instead of reaching the moderator, where they may be absorbed in a resonance
- The Dancoff factor quantifies this effect: it is the probability that a neutron leaving one fuel lump sees another fuel lump rather than pure moderator
  - Dancoff factor ≈ 1.0: neutrons are likely to stream without a collision between fuel lumps
  - Dancoff factor ≈ 0.0: neutrons are likely to have a collision between fuel lumps
- The Dancoff factor is a function of moderator cross section, lump size, shape, pitch, etc.
- As part of the self-shielding calculation, SCALE/XSProc uses numerical integration to calculate the Dancoff factor for uniform arrangement of fuel pins (fuel lump) (e.g., lattices)
  - These automatically calculated Dancoff factors can be overwritten by the user

Dancoff factor calculation for a square pitched uniform lattice

- **Reference fuel lump**
- **Nearest fuel lumps**
- **2nd Nearest fuel lumps**
- **3rd Nearest fuel lumps**
- **Totally shadowed**

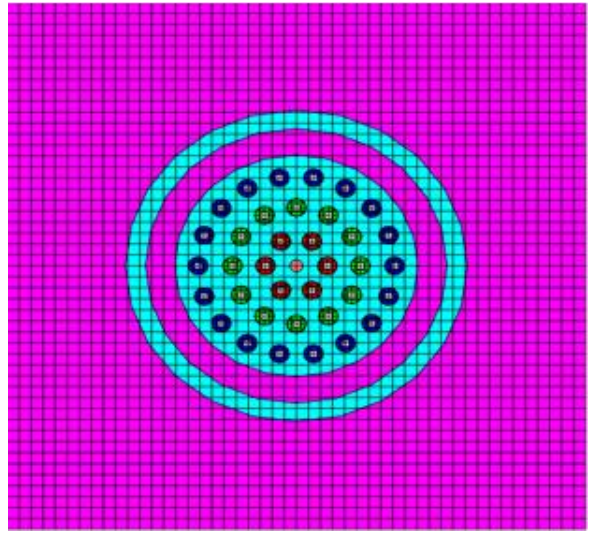


Analytic expressions are available for uniform lattices:
square pitch, triangular pitch, spherical square pitch, etc.

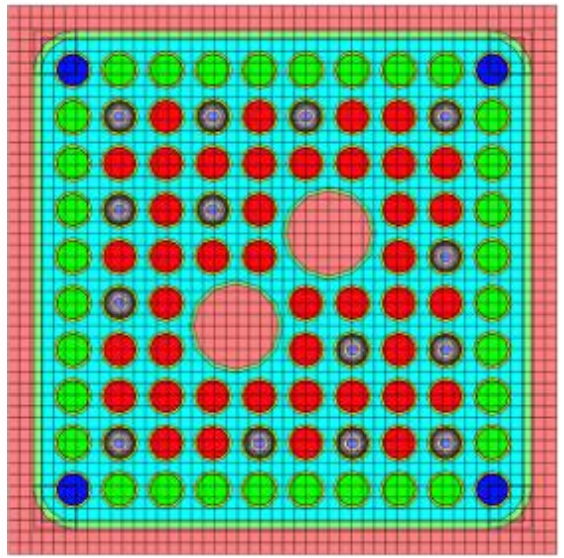
Special case 1: Non-uniform lattices

- Reactor cores deviate from a uniform lattice because they contain
  - Water holes
  - Burnable absorber rods
  - Control rods
  - Boundary interfaces (e.g., reflector)
- SCALE's default self-shielding models are based on the uniform lattice assumption because, by default:
  - CENTRM uses 1D cell with white/reflected boundary
  - BONAMI uses Dancoff factors computed for uniform lattice
- To improve the self-shielding calculation, the automatically calculated Dancoff factors for uniform lattices can be overwritten with Dancoff factors that were "manually" computed for the specific problem at hand
- Dancoff factors can be computed for each individual fuel pin of a model by running an auxiliary (separate) Monte Carlo calculation with the MCDANCOFF module (pronounced Mick-Dancoff, like McDonald's)

More on this later



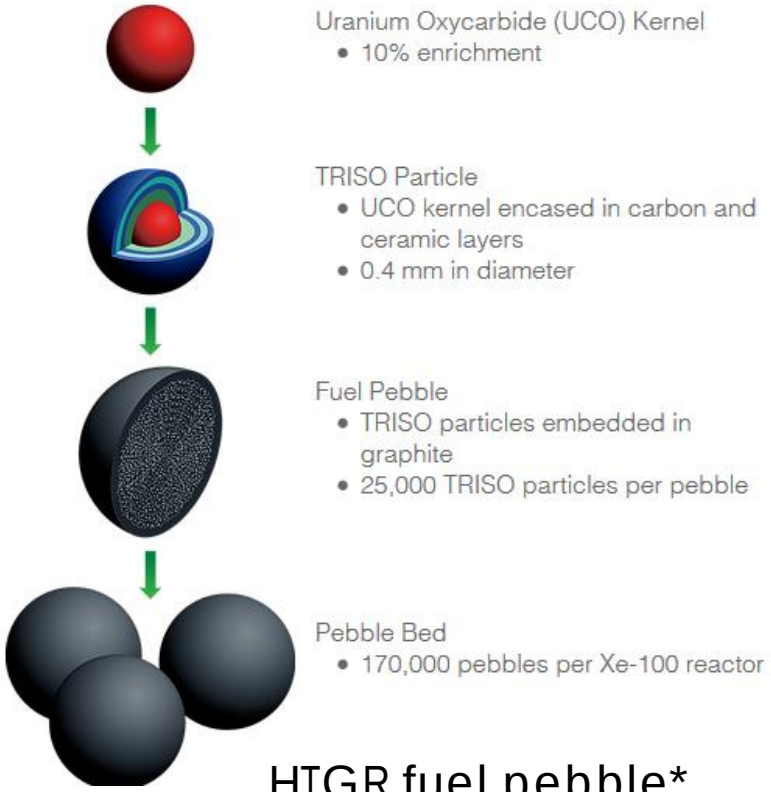
CANDU fuel element



BWR fuel element

Special case 2: Double-heterogeneous systems

- Double-heterogeneity:
  - Thousands of fuel particles (grains) are randomly dispersed in a graphite matrix → First heterogeneity
  - Hundreds or thousands of fuel elements (pebble, compact, plate) are distributed across the reactor core → Second heterogeneity
- Examples:
  - High-temperature gas-cooled reactors (HTGRs): pebbles, prismatic blocks
  - Fluoride salt-cooled high-temperature reactors (FHRs): pebbles, plates, etc.
  - TRISO-fueled heat pipe microreactors
  - Coarse mixed-oxide fuel in LWRs



HTGR fuel pebble*

*M. P. Baker, et al., *J. Nucl. Materials* 432, pp.395-406, 2013; X-Energy website

Special case 2: Double-heterogeneous self-shielding

- Self-shielding considerations:
  - Usual fuel-to-fuel element effect (Dancoff)
  - Grain heterogeneity can have a significant impact on resonance self-shielding
    - Grains are generally small (~mm) but still greater than mean free path at resonance energies
    - Causes spatial flux depression in grain (disadvantage factor)
- SCALE's unique approach for double-heterogeneous models:
  - Two subsequent self-shielding calculations: (1) fuel grains in graphite matrix, (2) lattice of pebbles/compacts/plates in core
  - Automated procedure

More on this later

OAK RIDGE National Laboratory

Sub-module 2

- Basic multigroup processing input



OAK RIDGE National Laboratory

Unit cell input specification in SCALE



Unit cell specification

- Provides information for cross section processing in the CELLDATA block
- Multiple unit cells may be specified in a single input
- Each mixture number may appear in only one region of only one unit cell
- Any mixture not specified in a unit cell is treated as an infinite homogeneous medium
- Additional unit cell data entered using MORE DATA and CENTRM DATA

Resonance processing of cross sections

- Unit cell types:
  1. Infinite homogeneous medium
  2. Lattice cell
  3. Multiregion
  4. Double-heterogeneous cell
- Other resonance processing options
  - Generalized Dancoff approach to treat non-uniform lattices
    - Users input Dancoff factors, and SCALE calculates Dancoff-equivalent unit cell

INFHOMMEDIUM

- For large homogeneous systems
- All materials treated as infinite system
- No corrections for geometrical effects
- No leakage corrections
- No lattice corrections

INFHOMMEDIUM Example

```
=xsproc
inf  homogeneous  media (93.2%) uranyl nitrate
v7.1-252
read comp
 solution   mix=1  rho[uo2(no3)2]=345 92235 93.2
     92238 6.8 molar[hno3]=0.5 density=? temperature=293
  end solution
 ss304        2 1 293 end
end comp
read celldata
 inf 1 end
 inf 2 end
end celldata
end
```

Optional – equivalent to default treatment

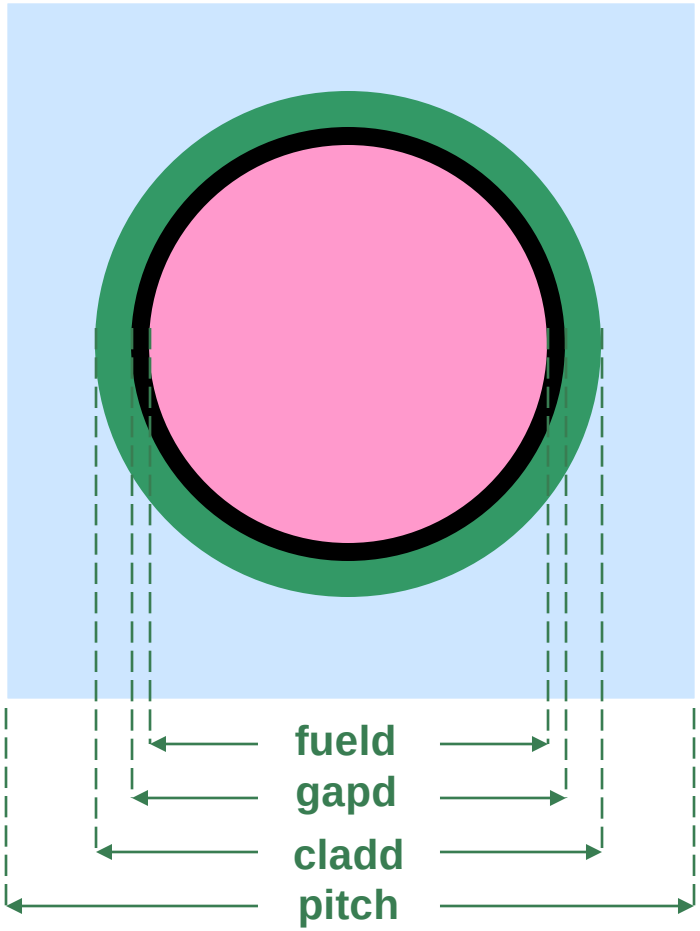
LATTICECELL

- For large arrays (lattices) of spheres, slabs, or rods
  - Small cells in terms of neutron mean free path
- Resonance processing based on cell geometry and dimensions
- Provides corrections for lattice structuring
- CENTRM solves SQUAREPITCH LATTICECELL problems with a MOC solver, all other problems with an SN solver

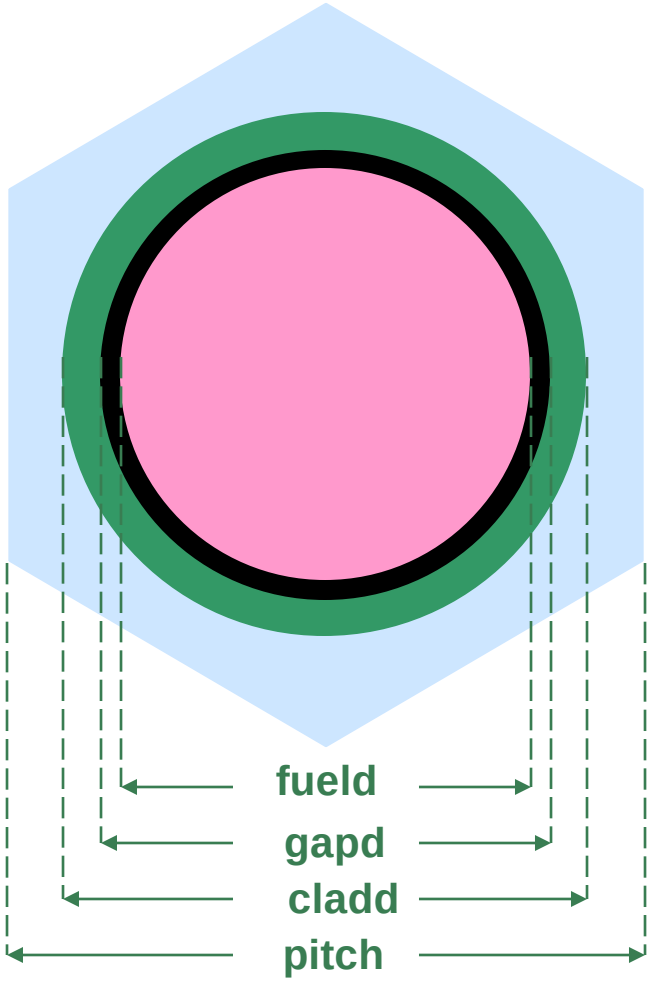
LATTICECELL types

Regular Cells	Annular Cells	Slab Cells
SQUAREPITCH	ASQUAREPITCH	SYMMSLABCELL
TRIANGPITCH	ATRIANGPITCH	ASYMSLABCELL
SPHSQUAREP	ASPHSQUAREP	
SPHTRIANGP	ASPHTRIANGP	

LATTICECELL – Geometry specifications

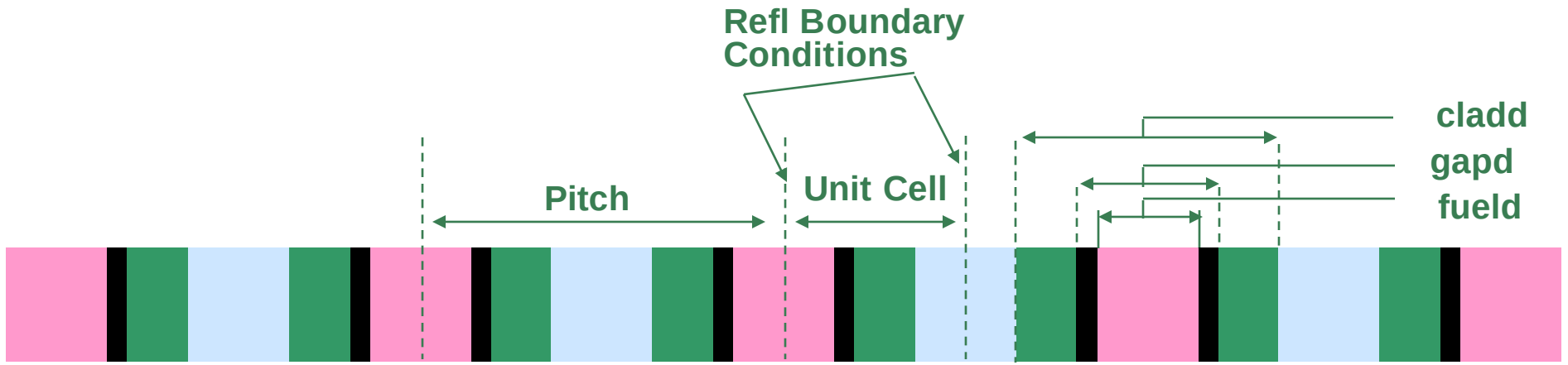


SQUAREPITCH and SPHSQUAREP unit cell material arrangement



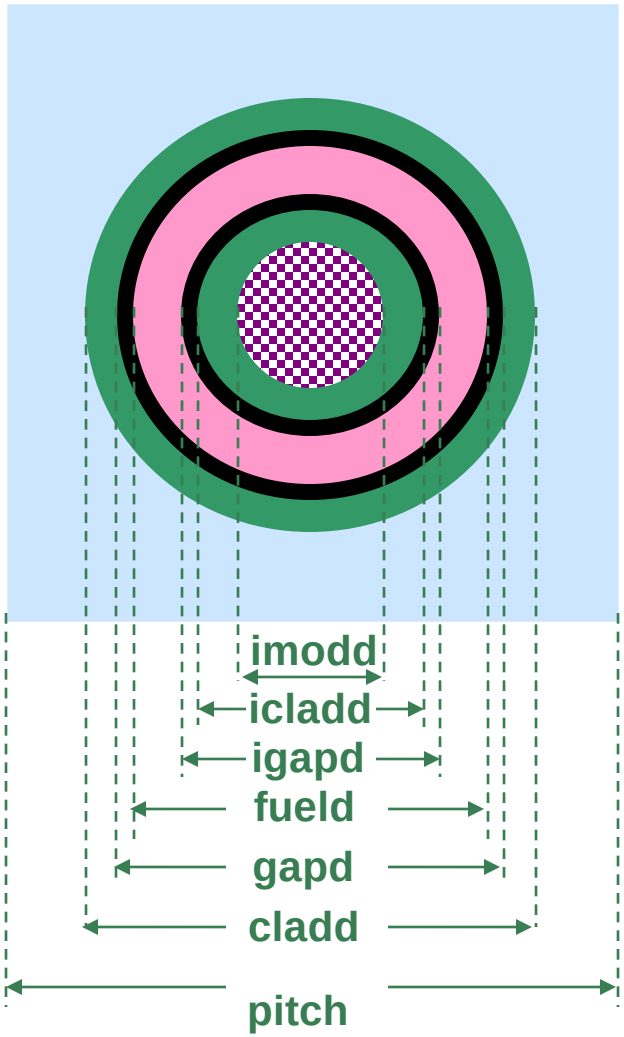
TRIANGPITCH and SPHTRIANGP unit cell material arrangement

LATTICECELL – Slab geometry

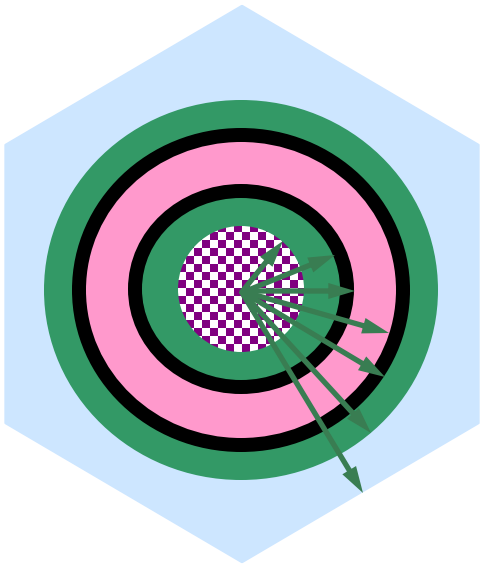


SYMMSLABCELL unit cell with left and right reflected boundary conditions material arrangement

LATTICECELL – Annular geometries



ASQUAREPITCH and ASPHSQUAREP unit cell material arrangement



imodr = internal moderator radius

icladr = internal clad radius

igapr = internal gap radius

fuelr = fuel radius

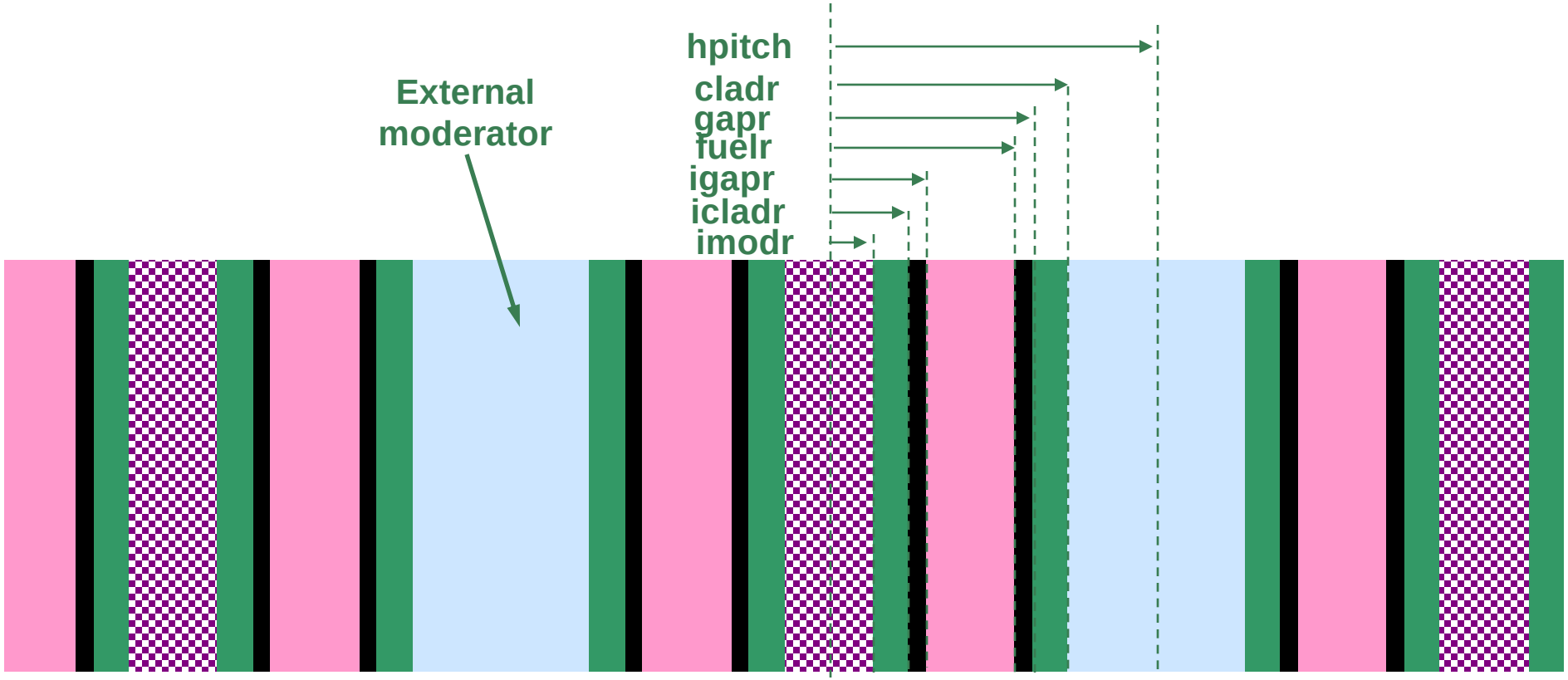
gapr = gap radius

cladr = clad radius

hpitch = half pitch

ATRIANGPITCH and ASPHTRIANGP unit cell material arrangement

LATTICECELL – asymmetric slab geometry



ASYMMSLABCELL unit cell with left and right reflected boundary conditions material arrangement

LATTICECELL examples

```
latticecell squarepitch  pitch 1.63576 3  fuelr 0.514858 1  cladr 0.602996 2 end
```

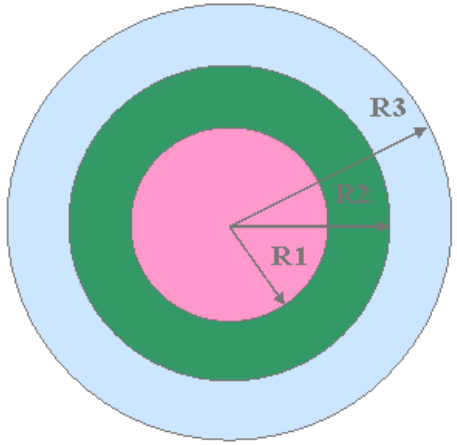
```
   latticecell triangpitch pitch=2.0320 7  fueld=1.2827 1  cladd=1.4351 4 end
```

```
      latticecell  squarepitch  pitch= 1.2941 3  fueld= 0.78440 101
                                cladd= 0.91440 2  gapd= 0.80020   4 end
```

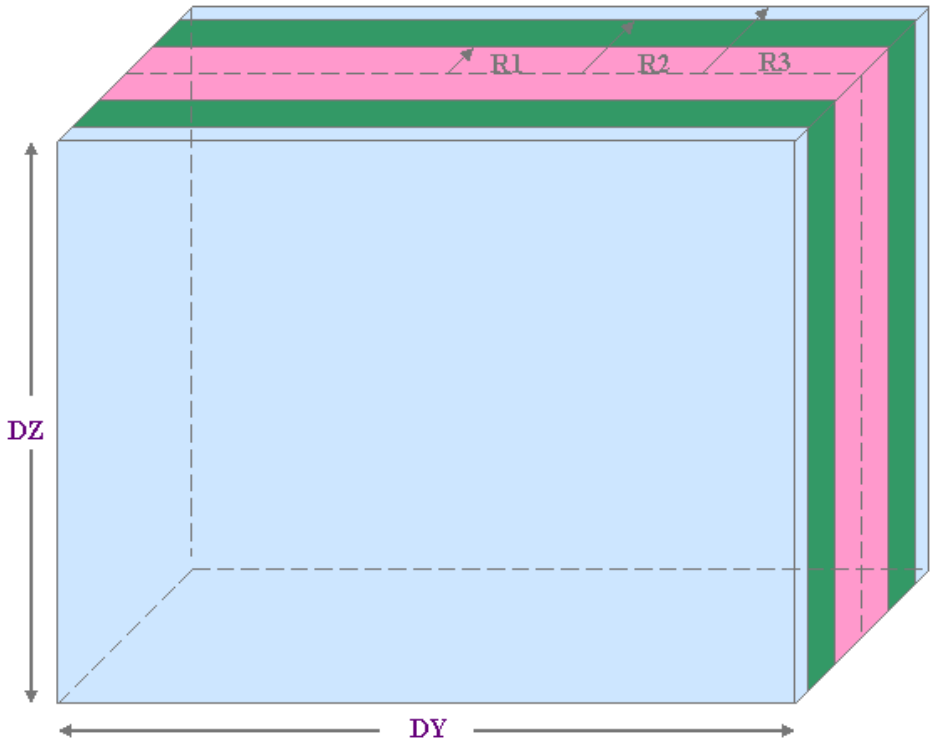
MULTIREGION

- For geometric configurations more complicated than LATTICECELL
- For large geometric regions with minimal geometry effects
- Resonance processing based on cell geometry and dimensions
- Lattice effects are approximated using white boundary condition
- Types: Slab, Cylindrical, Spherical

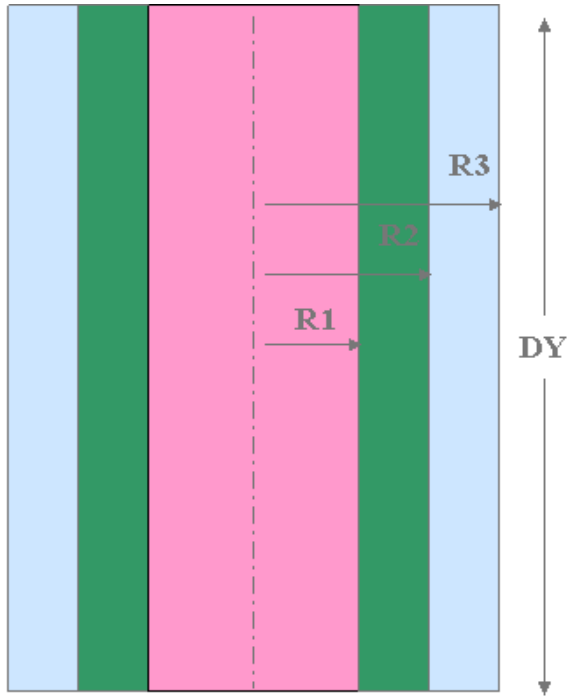
MULTIREGION geometry specifications



SPHERICAL, CYLINDRICAL, and BUCKLEDCYL
unit cell material arrangement



SLAB and BUCKLEDSLAB, unit cell material arrangement
DY & DZ (cm) are used for BUCKLEDSLAB
DY & DZ are entered only for non-infinite Y & Z dimensions



BUCKLEDCYL
DY is the buckled height in cm

MULTIREGION cell input

- Two parts to MULTIREGION cell input
  - Keywords
    - Geometry type, boundary conditions, etc.
  - Geometry
    - Mixtures and dimensions

MULTIREGION geometry input

Entry #	Data
1	Mixture (Zone 1)
2	Radius/distance from origin (Zone 1)
3	Mixture (Zone 2)
4	Radius/distance from origin (Zone 2)
5	Mixture (Zone 3)
6	Radius/distance from origin (Zone 3)
.	. . .
Last	END ZONE

MULTIREGION boundary conditions

- Spherical, Cylindrical
  - *left_bdy* (reflected only)
    - Reflected: this is the only valid boundary condition because the left boundary corresponds to the centerline of the cylinder or the center of the sphere
  - *right_bdy* (vacuum [default] or white)
    - White: this is the best option to approximate reflection at the outermost surface
- Slab
  - *left_bdy* (reflected, periodic, vacuum [default], or white)
  - *right_bdy* (reflected [default], periodic, vacuum, or white)

MULTIREGION Example

Example multiregion input:

```
multiregion cylindrical right_bdy=white end
    702
    0.1656051
    703
    0.2342010
    310
    0.730626 end zone
```

Identical input but visually clearer:

```
multiregion cylindrical right_bdy=white end
    702 0.1656051
    703 0.2342010
    310 0.730626 end zone
```

Sub-module 3

- Multigroup processing exercises
  - Criticality Safety: Problems 1–5
  - Reactor Physics:
    - LWR: Problems 6–8
    - Non-LWR: Problem 9

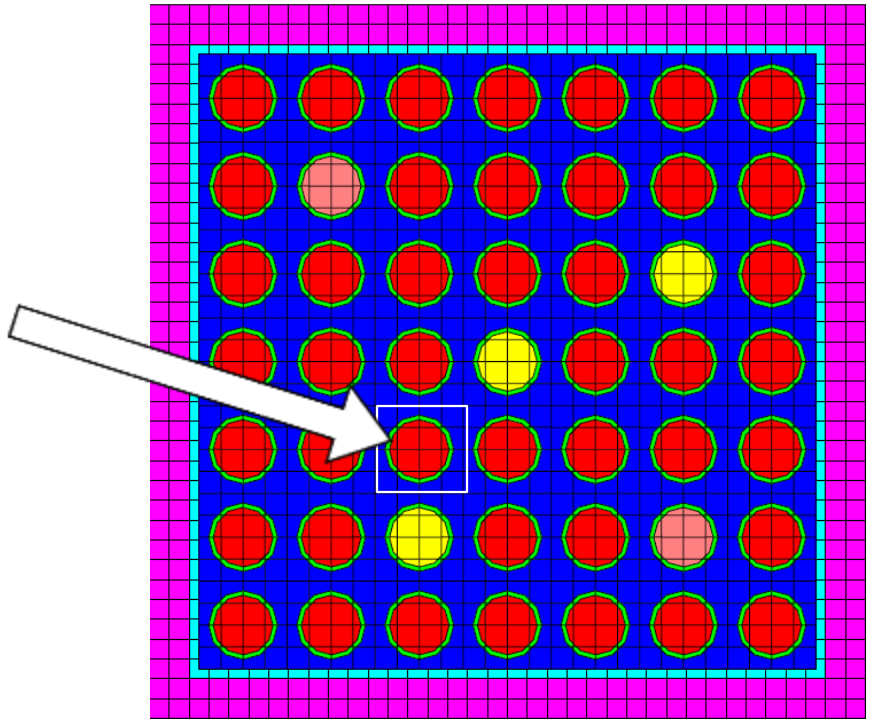


Objectives

- Increase familiarity with unit cell data
- Gain competence in selecting appropriate unit cell type
- Use Fulcrum to visualize cross sections
  - More in future lecture

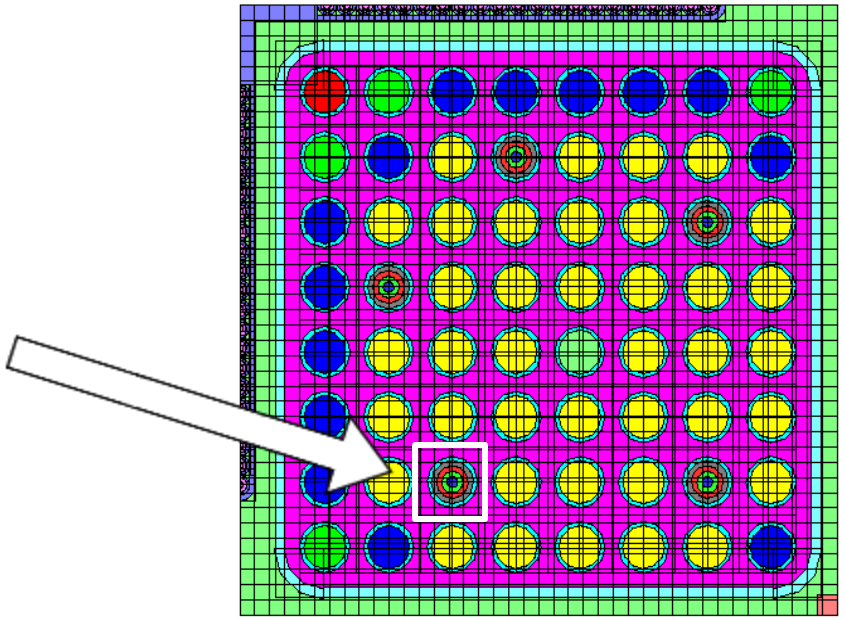
Problem 6: BWR pin cell

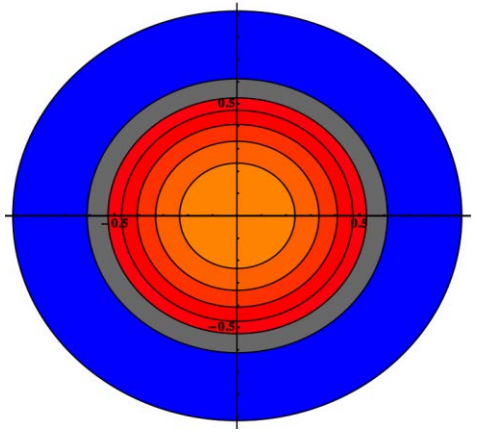
- General:
  - Title: BWR pin cell
  - Sequence: xsproc
  - Library: v7-56
  - Use: centrm
- Use materials from ReactorPhysicsMaterials.inp
  - Mixture 1, fuel
  - Mixture 20, clad
  - Mixture 30, moderator
  - No Gap material
- Cell data:
  - squarepitch lattice
  - Pitch=1.875 cm
  - Fuel Radius=0.621 cm
  - Clad Radius=0.715 cm



Problem 7: Gd-bearing BWR pin cell

- General:
  - Title: Gd-bearing BWR pin
  - Sequence: t-xsec
  - Library: v7-56
  - Use: centrm
- Use materials from ReactorPhysicsMaterials.inp
  - Fuel (mixtures 2-6): UO_2 (4% enr) 94%VF + Gd_2O_3 6%VF, 900 K, 10.5 g/cc
  - Zirc-2 Clad, default density, 600 K
  - H2O Coolant, 0.46 g/cc (i.e. ~40% void fraction), 540 K
- Dimensions:
  - Fuel radius: 0.5283 cm
  - Cladding radius: 0.6134 cm
  - Lattice pitch: 1.6256 cm

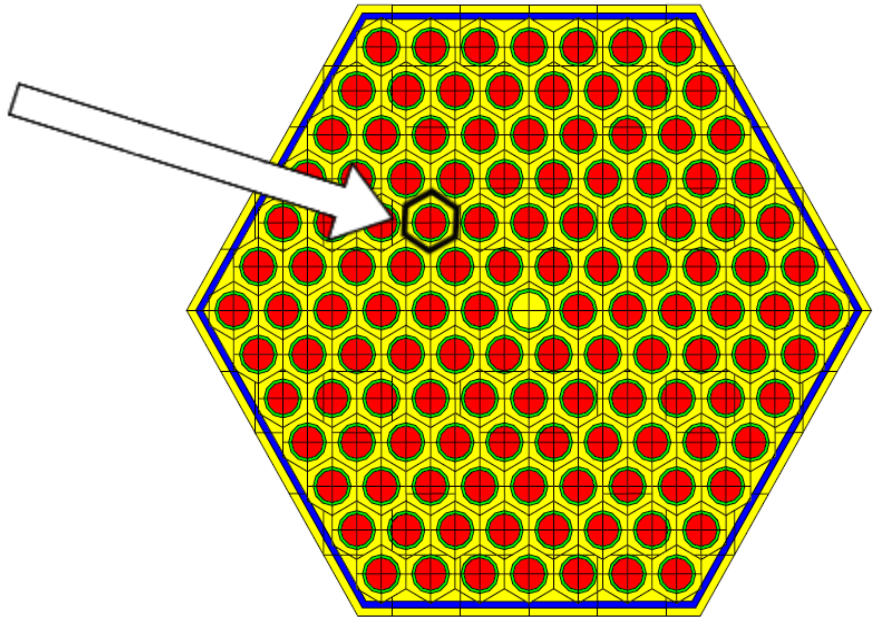




Help: Multiregion zones
- 5 equal-volume fuel zones: 0.23626, 0.33413, 0.40922, 0.47253, 0.5283 cm
- 1 cladding zone (R=0.6134)
- 1 moderator zone (R=0.91715)

Problem 8: VVER pin cell

- General data:
  - Title: VVER pin cell
  - Sequence: xsproc
  - Library: v7-56
  - Use: centrm
- Use materials from ReactorPhysicsMaterials.inp:
  - Fuel: UO_2(4% enr), 95% TD
  - Clad Alloy: Zirc-2, default density (Mixture 20)
  - Moderator: 900 ppm boron in H_2O, 0.75 g/cc (Mixture 30)
- Dimensions:
  - Fuel radius: 0.386 cm
  - Cladding radius: 0.455 cm
  - Lattice pitch: 1.22 cm
- Cell:
  - Use triangpitch cell option



Bonus exercise:

- Copy problem-dependent cross section library (watch out for bad characters in the copy string):

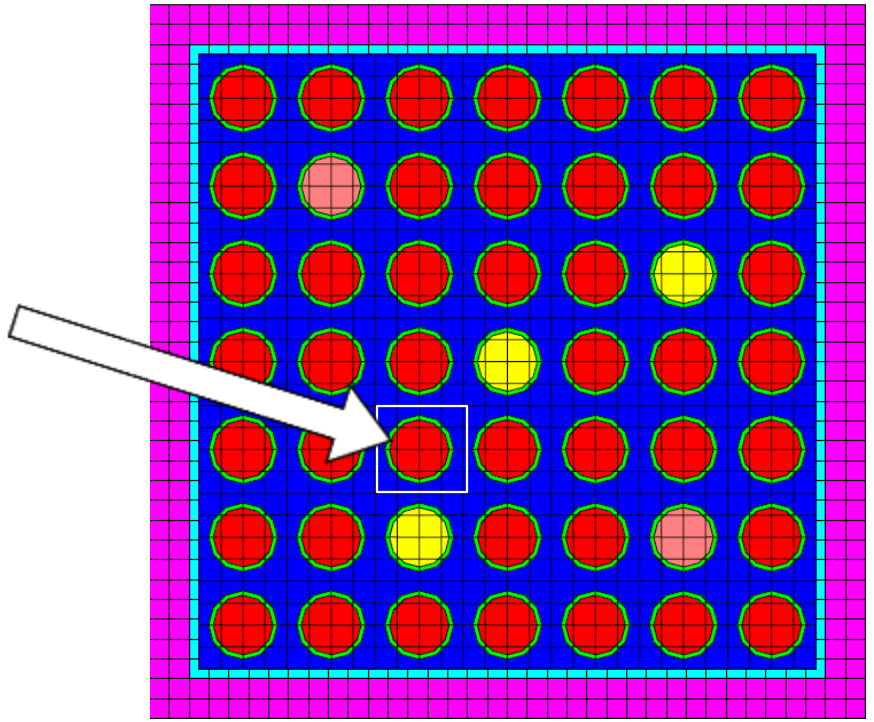
```
=shell
cp ${TMPDIR}/sysin.microLib ${OUTDIR}/${BASENAME}.xs
end
```

- Plot U-238 n,γ cross section: compare the data from SCALE's 56g library with the data from the copied problem-dependent library

SOLUTIONS

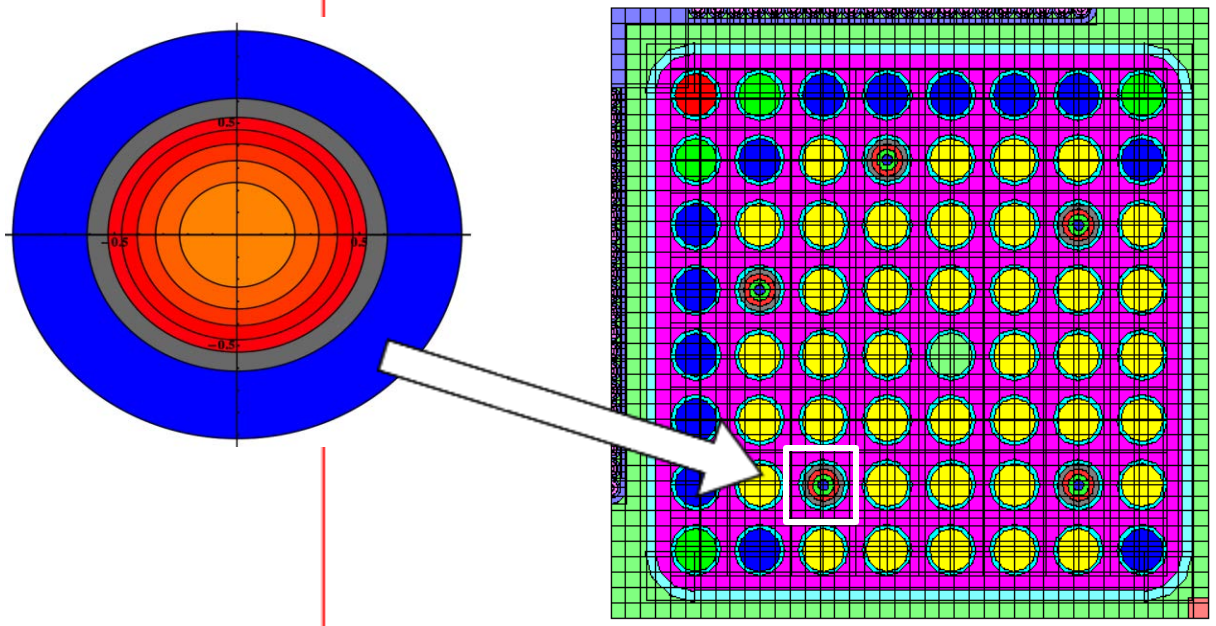
Solution 6: BWR pin cell

```
read celldata
  latticecell squarepitch
    fuelr=0.621 1
    cladr=0.715 20
    hpitch=0.9375 30  end
end celldata
```



Solution 7: Gd-bearing BWR pin cell

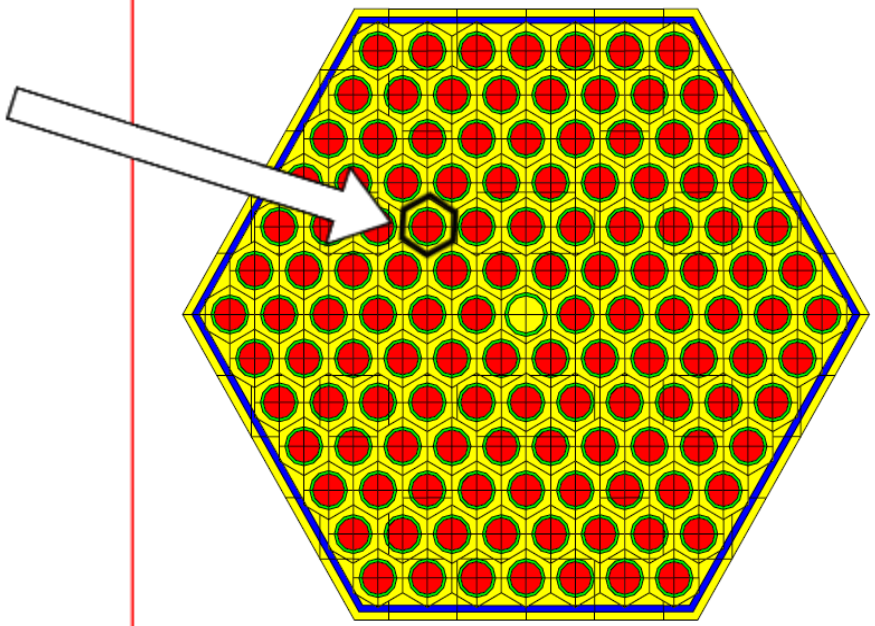
```
read celldata
  multiregion cylindrical
       left_bdy=reflected
       right_bdy=white  end
            2       0.23626
            3       0.33413
            4       0.40922
            5       0.47253
            6       0.5283
           20       0.6134
           30       0.91715
       end zone
  centrmdata id1=2 end centrmdata
end celldata
```



Solution 8: VVER pin cell

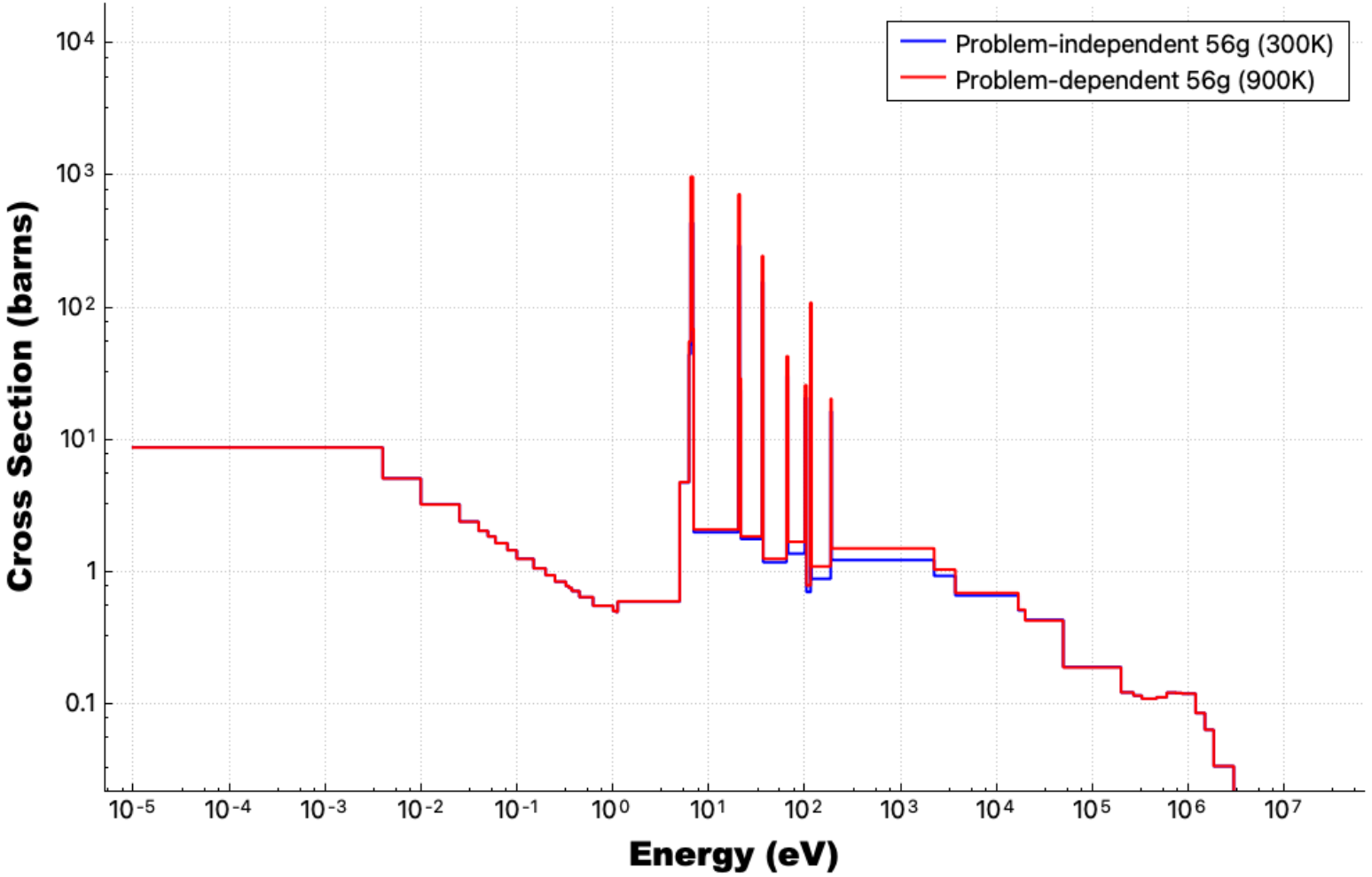
```
=xsproc parm=(centrm)
problem 8: VVER pin cell
v7-56
read composition
 uo2          1 0.95 900
                  92235 4.0
                  92238 96.0   end
 zirc2        20 1 600   end
 h2o          30 den=0.75 1.0000 540   end
 boron        30 den=0.75 0.0009 540   end
end composition
read celldata
  latticecell triangpitch fuelr=0.386 1 cladr=0.455 20
hpitch=0.61 30  end
end celldata
end

=shell
 cp "${TMPDIR}/sysin.microLib"  "${OUTDIR}/${BASENAME}.xs"
end
```



Solution 8: VVER pin cell

^{238}U neutron capture (n,γ) cross section



OAK RIDGE National Laboratory

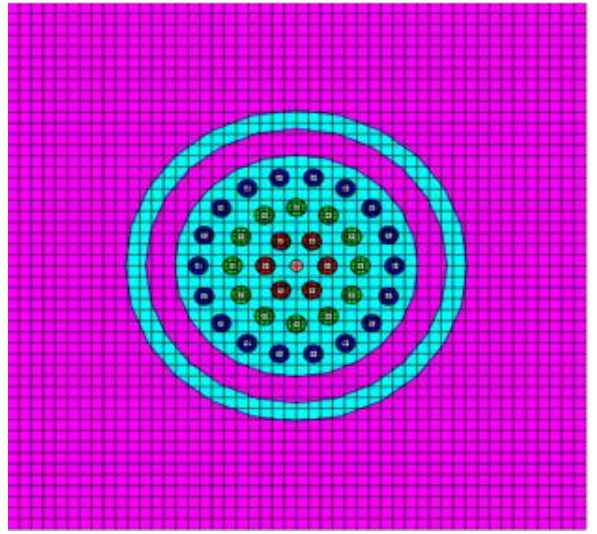
Sub-module 4

- Non-uniform lattices
  - Reactor Physics only

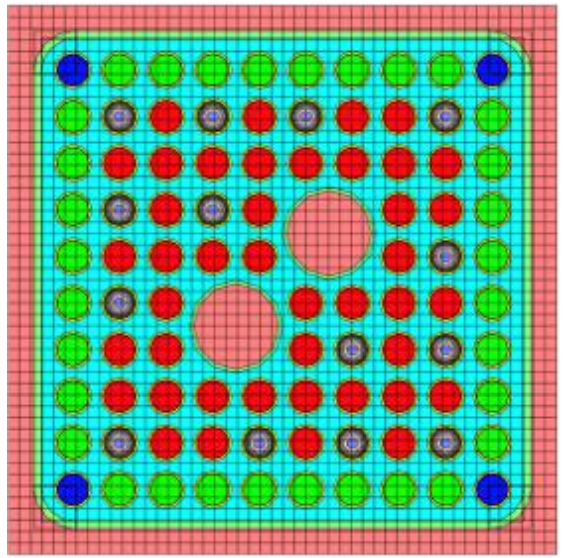


Non-uniform lattices

- Reactor cores deviate from a uniform lattice because they contain
  - Water holes
  - Burnable absorber rods
  - Control rods
  - Boundary interfaces (e.g., reflector)
- SCALE's default self-shielding models are based on the uniform lattice assumption because, by default:
  - CENTRM uses 1D cell with white/reflected boundary
  - BONAMI uses Dancoff factors computed for uniform lattice
- Solution: Calculate Dancoff factors and provide them in the SCALE input



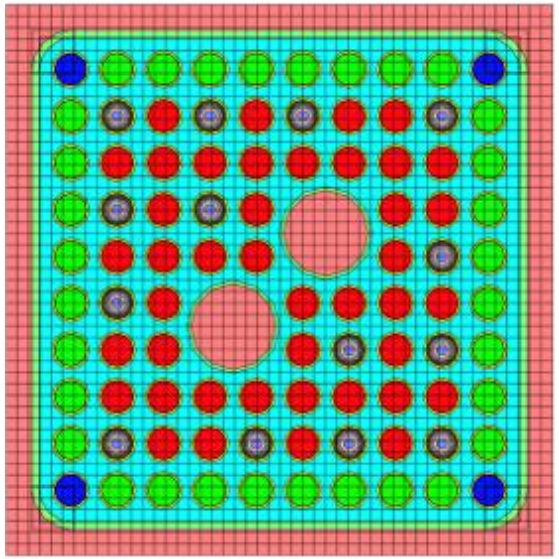
CANDU fuel element



BWR fuel element

Dancoff factor

- The Dancoff factor describes the probability that a neutron leaving one fuel lump sees another fuel lump, rather than pure moderator.
  - Dancoff factor ≈ 1.0: neutrons are likely to stream without a collision between fuel lumps
  - Dancoff factor ≈ 0.0: neutrons are likely to have a collision between fuel lumps
- The Dancoff factor is a function of moderator cross section, lump size, shape, pitch, etc.
- SCALE/XSProc calculates by default a Dancoff factor under the assumption that the fuel pin lies in an infinite lattice



BWR fuel element

- Example high-void BWR lattice:
  - Corner, edge, and water hole-adjacent pins receive significant moderation due to full-density water features
  - The assumption of an infinite lattice is not accurate for this assembly
- More accurate Dancoff factors for fuel pins can be obtained using SCALE module MCDANCOFF

MCDANCOFF

- MCDANCOFF is a modified version of KENO-VI that runs a fixed-source calculation to determine approximate Dancoff factors for user-specified regions in the model
- MCDANCOFF requires a KENO-VI geometry
- MCDANCOFF can calculate Dancoff factors for regions bounded by cylinders, spheres, or cuboids

MCDANCOFF input generation

1. Create CSAS-KENO-VI input of the model of interest
2. Change the sequence name to "mcdancoff"
3. Use the "xn01" cross section library
4. Add the START block to specify for which locations to calculate Dancoff factors

```
read start          ! Start
dancoff             ! Calculate a Dancoff factor
  array 1 4 2 1     ! Located in array 1, at array position (4,2,1)
  unit 10           ! Unit number 10 fills position (4,2,1)
  region 1          ! Calculate the Dancoff factor for region 1 of unit 10
end start           ! End reading Dancoff factors
```

Multiple Dancoff factors can be calculated on each run by adding extra dancoff statements.

MCDANCOFF output

- A *.danc file will be created for each material for which a Dancoff factor is requested

```
Unit 17 at global x -5.62500E+00  y -5.62500E+00  z  0.00000E+00
        index        nuclide          dancoff       deviation
            1          92235     1.41663E-01     2.24232E-03
            2          92234     1.41663E-01     2.24232E-03
            3          92236     1.41663E-01     2.24232E-03
            4          92238     1.41663E-01     2.24232E-03
            5           8016      9.59918E-01     3.68866E-04
```

This is the value to pick

- Save the Dancoff factor of a fuel material for use in TRITON

Application of Dancoff factors in CELLDATA

- The calculated Dancoff factors can be added to the CENTRM celldata by using the dan2pitch option

```
read celldata
  latticecell squarepitch  pitch=1.3     10
                           fuelr=0.44     5
                           gapr=0.45      20
                           cladr=0.5      30      end
  centrmdata dan2pitch=0.1417 end centrmdata
end celldata
```

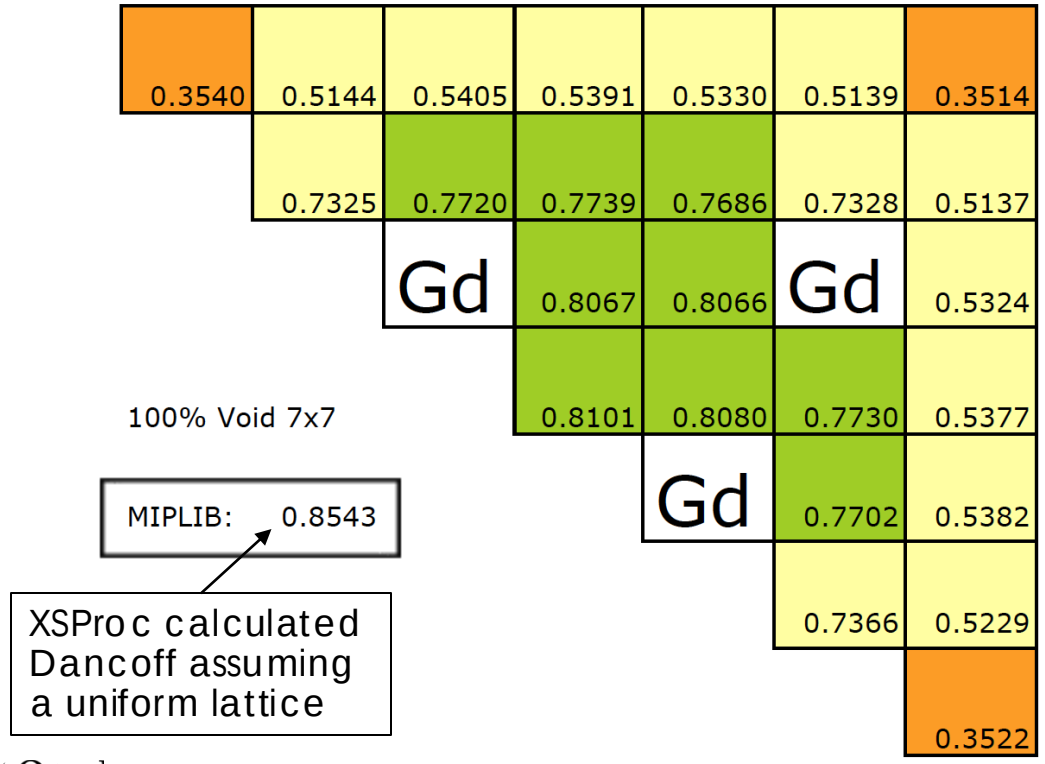
- XSProc reads the user-specified Dancoff factor to construct a Dancoff-equivalent unit cell, for which an infinite lattice calculation is performed

CENTRMDATA blocks apply only to the cell specified directly above, not to any other cell

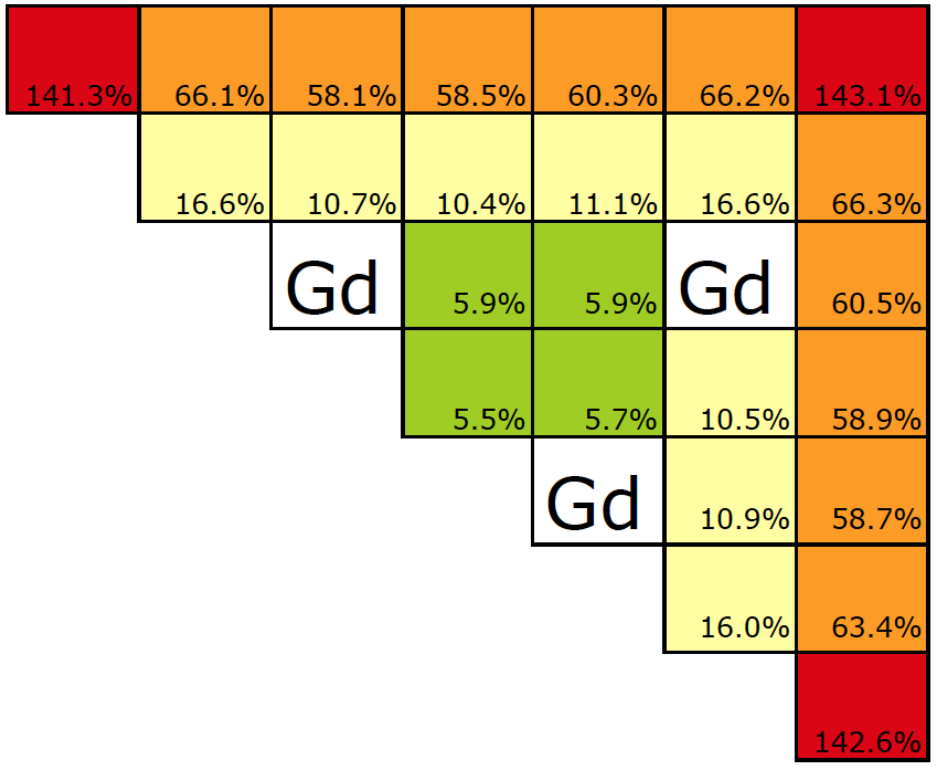
Dancoff factor example: High-void BWR lattice

- A Dancoff factor map visually shows the variation of Dancoff factors for a lattice
- High-void BWR lattices may have corner pins whose Dancoff factor is overpredicted by >100%, and corner pins whose Dancoff factor is overpredicted by >50%

Dancoff factors calculated with MCDANCOFF

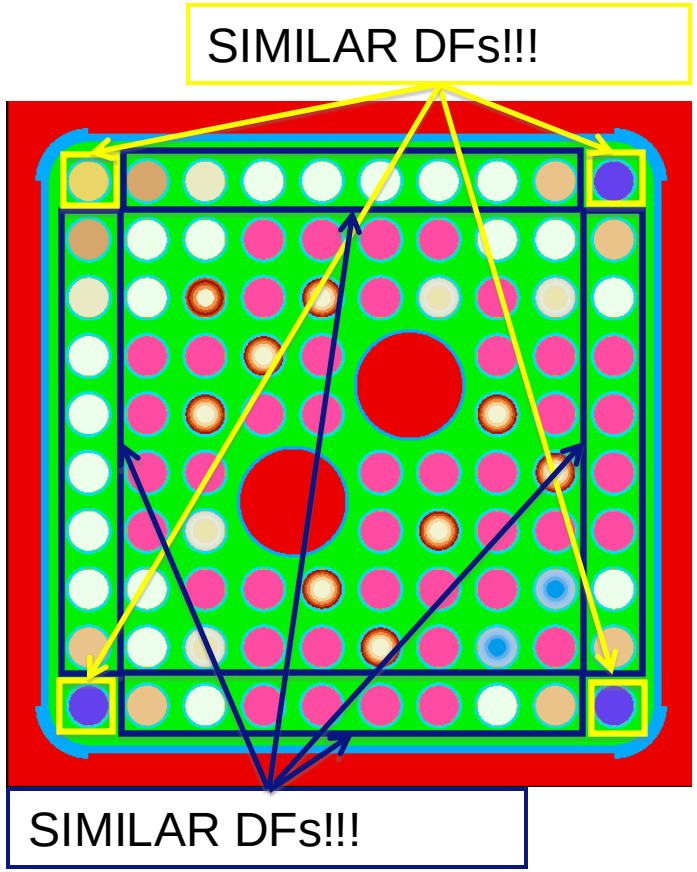


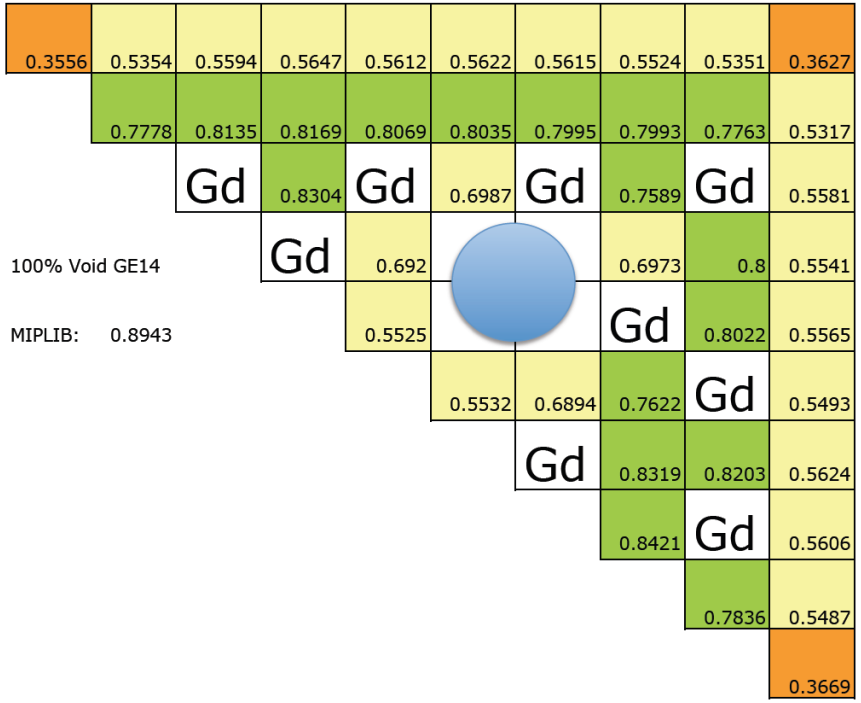
Difference to Dancoff factors calculated with XSProc assuming a uniform lattice



Fuel lumping strategy

- For self-shielding calculations, there are two levels of fuel lumping that should be used
  - Same enrichment
  - Similar Dancoff factor (in boxes below)





Fuel lumping strategy

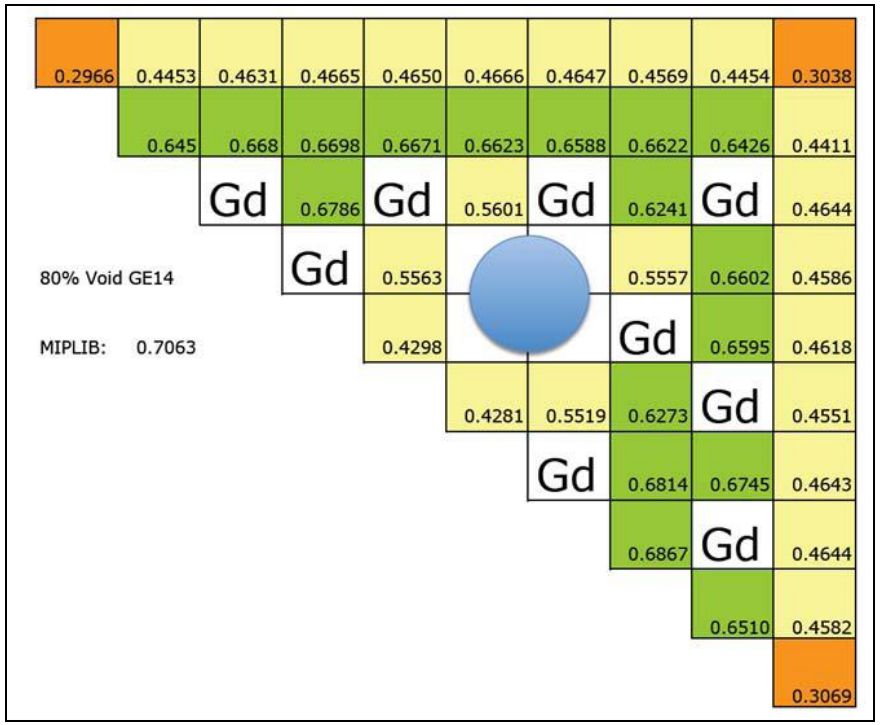
- Lump all same-enrichment pins into one CENTRM latticecell block
- Add another CENTRM latticecell block for each enrichment that has pins on the assembly edge or corner as well as pin in the assembly interior
- Apply the Dancoff factor calculated using MCDANCOFF

```
latticecell squarepitch  pitch=1.3          963
                         fuelr=0.44         440
                          gapr=0.45         961
                         cladr=0.50         962          end

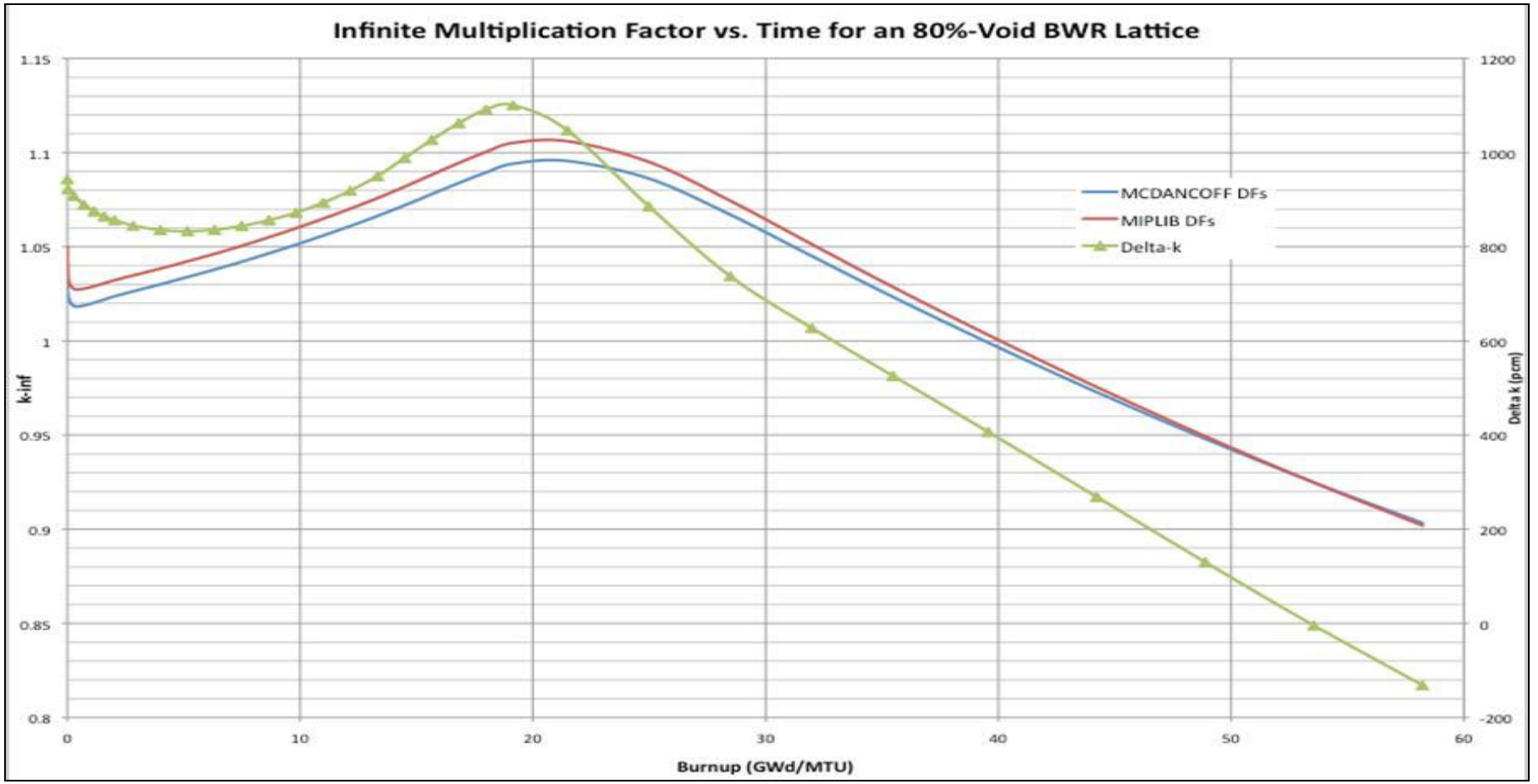
latticecell squarepitch  pitch=1.3          966
                         fuelr=0.44         449
                          gapr=0.45         964
                         cladr=0.50         965          end
centrmdata dan2pitch=0.5500  end centrmdata
```

Example: BWR fuel depletion calculations

- Heterogeneities exist in BWR bundle design
  - Radial and axial enrichment zoning
  - Placement and loading of gadolinia burnable absorber rods
  - Water rods/holes
- Some guidance provided in SCALE newsletter Number 42 (Summer/Fall 2010), pp. 6–8
  - https://www.ornl.gov/sites/default/files/2019-02/scale_42_fall2010.pdf
  - Dancoff factor map for non-Gd fuel pins in a BWR lattice: void fraction of 80% in the fuel channel and solid saturated water in the bypass and water holes



Example: BWR fuel depletion calculations



For an 80% Void GE14 lattice, the difference between using corner and edge Dancoff Factors and not using them can be as great as 1,000 pcm

OAK RIDGE National Laboratory

Sub-module 4

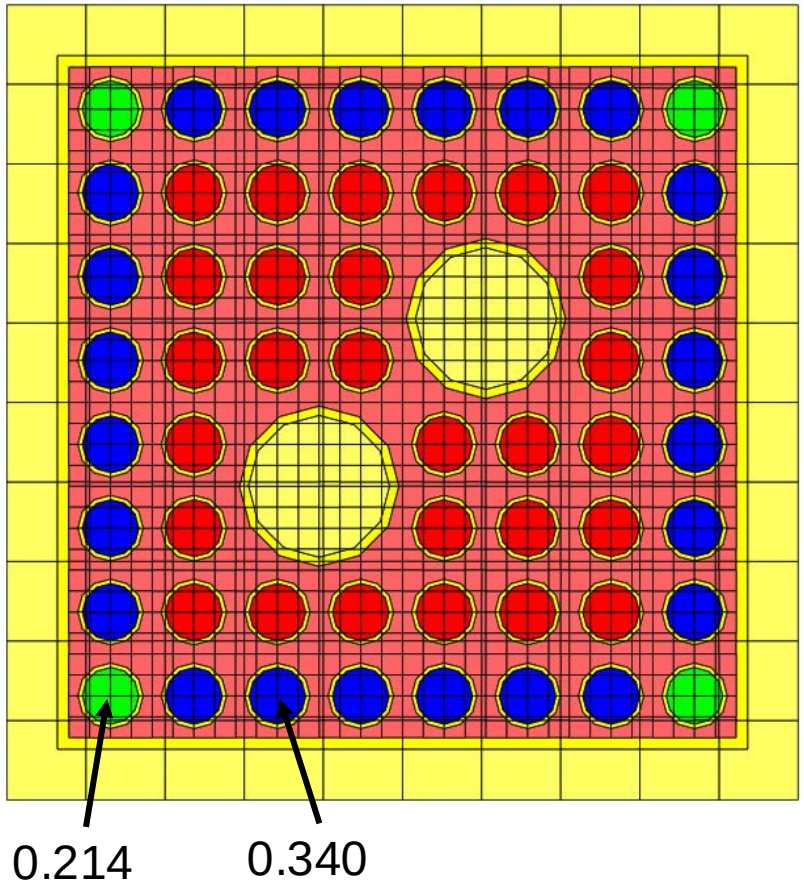
- Dancoff exercises
  - LWR: Problem 10
  - Non-LWR: Problem 11



Problem 10a: User-defined Dancoff Factors in TRITON

- Base run:
  - Open file "problem_10a_starter.inp" and save as "problem_10a.inp"
  - Note that this is a TRITON-NEWT 2D model
  - Run "problem_10a.inp"
  - Record k_{eff} and the automatically calculated Dancoff factor
- Run with Dancoff:
  - Copy "problem_10a.inp" to "problem_10a_dancoff.inp"
  - Add two new fuel, clad, and moderator mixtures
  - Add additional latticecells for the corner and edge fuel pins
    - Use the average Dancoff factor for the corner fuel pins and the edge fuel pins given in the image below
  - Use the new materials in the NEWT transport model for the corner and edge fuel pins
  - Run the model
- How do the obtained k_{eff} values compare?

NEWT model of the BWR fuel element

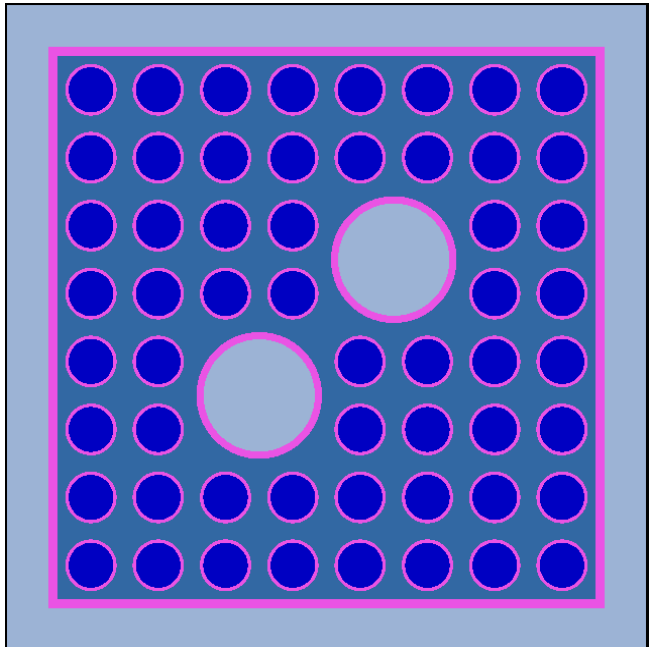


Dancoff factors calculated with McDancoff

Problem 10b: Dancoff Factors using MCDancoff

- Open file "problem_10b_starter.inp" and save as "problem_10b.inp"
- Note that this is a KENO 3D model corresponding to the NEWT 2D model from Problem 10a
  - All fuel pins can be modeled using 1 fuel pin definition
- Add a start block to calculate Dancoff factors for each fuel pin
- Use the following parameters:
  - gen=100
  - npg=500
  - nsk=0
  - run=yes
- Run the calculation and record the Dancoff factors for each pin
- How do the Dancoff factors calculated by McDancoff compare to the XSProc-calculated Dancoff factors?
- Could we do anything else to make the solution more accurate?

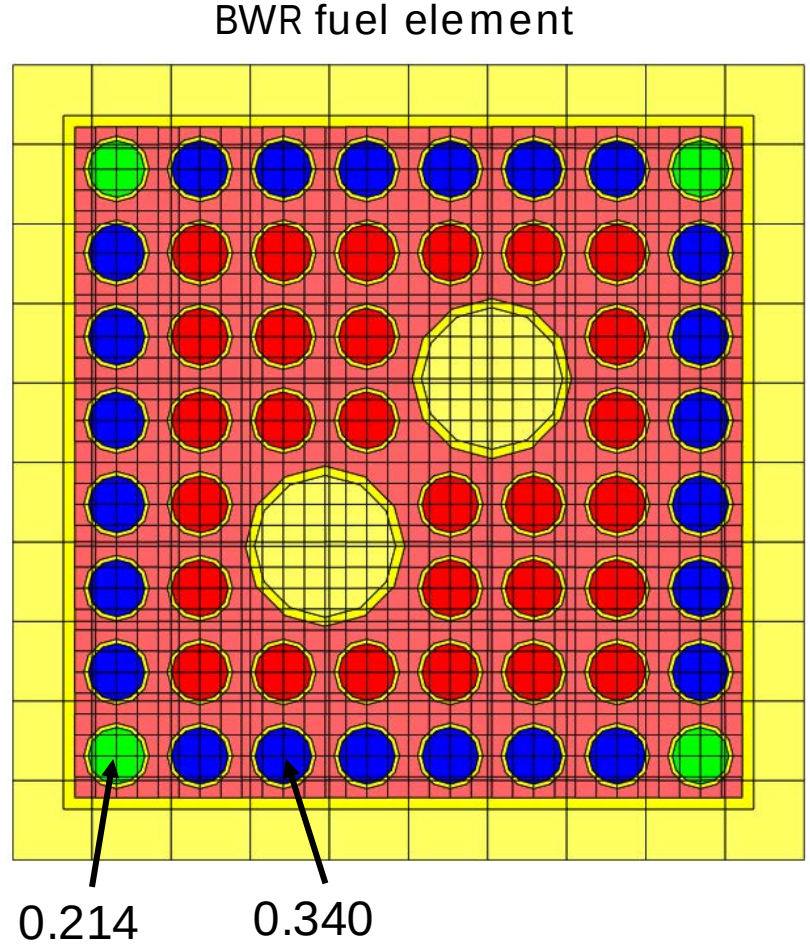
KENO model of the BWR fuel element



SOLUTIONS

Problem 10a: User-defined Dancoff Factors in TRITON

- XSProc-calculated Dancoff factor: 0.595767
  - Search for "dancoff calculations for cell 1"
- k_{eff} comparison:
  - Base run: 1.43104
  - Run with Dancoff: 1.42384
  - Δk = 720 pcm

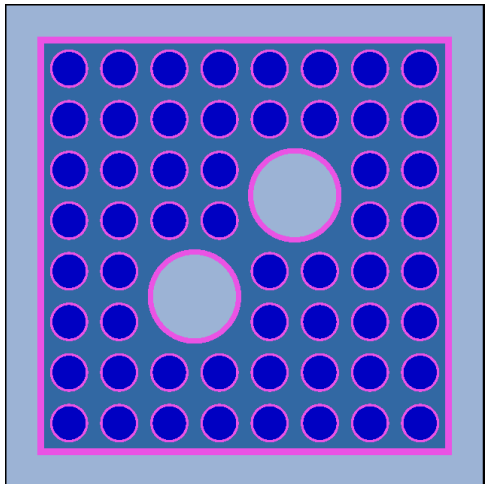


Dancoff factors calculated with McDancoff

Problem 10b: Dancoff Factors using MCDancoff

- XSProc Dancoff factor:
  - 0.596
- MCDancoff Dancoff factors:
  - Corner-average: 0.213
  - Side-average: 0.339
  - Average of all others: 0.446
- How do the Dancoff factors compare?
  - XSProc overestimates Dancoff factor → underpredicts the probability that neutrons have a collision between fuel lumps
- Could we do anything else to make the solution more accurate?
  - Apply Dancoff factors to all fuel pins (while benefiting from the model's symmetry)
  - Run MCDancoff with more neutron histories to improve convergence
  - Update other NEWT parameters (SN order, spatial grid, etc.)

```
read start
dancoff
array 1 1 1 1
unit 1
region 1
dancoff
array 1 2 1 1
unit 1
region 1
…
dancoff
array 1 7 8 1
unit 1
region 1
dancoff
array 1 8 8 1
unit 1
region 1
end start
```



BWR fuel element

0.211	0.328	0.338	0.338	0.340	0.347	0.339	0.214
0.331	0.453	0.412	0.409	0.471	0.513	0.506	0.338
0.340	0.408	0.420	0.523			0.514	0.348
0.342	0.409	0.317	0.420			0.470	0.341
0.338	0.469			0.420	0.318	0.410	0.338
0.346	0.515			0.524	0.418	0.409	0.340
0.338	0.506	0.512	0.469	0.411	0.410	0.452	0.331
0.217	0.341	0.348	0.340	0.340	0.339	0.330	0.212

Dancoff factors

OAK RIDGE National Laboratory

Sub-module 5

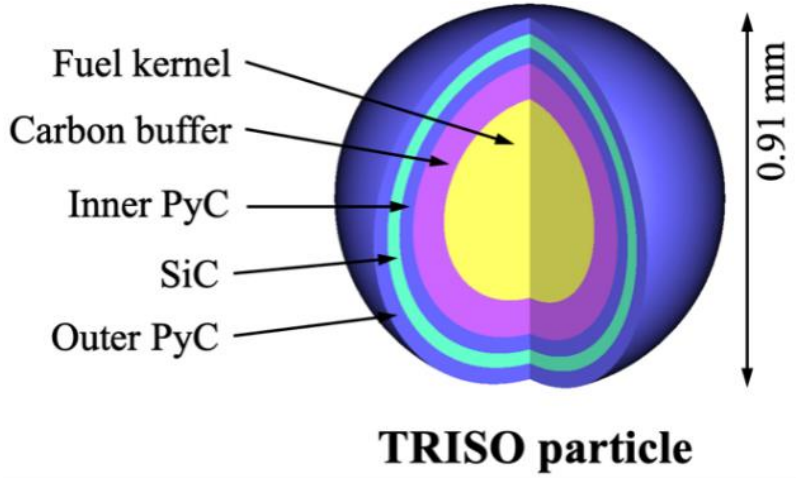
- Double-heterogeneous systems



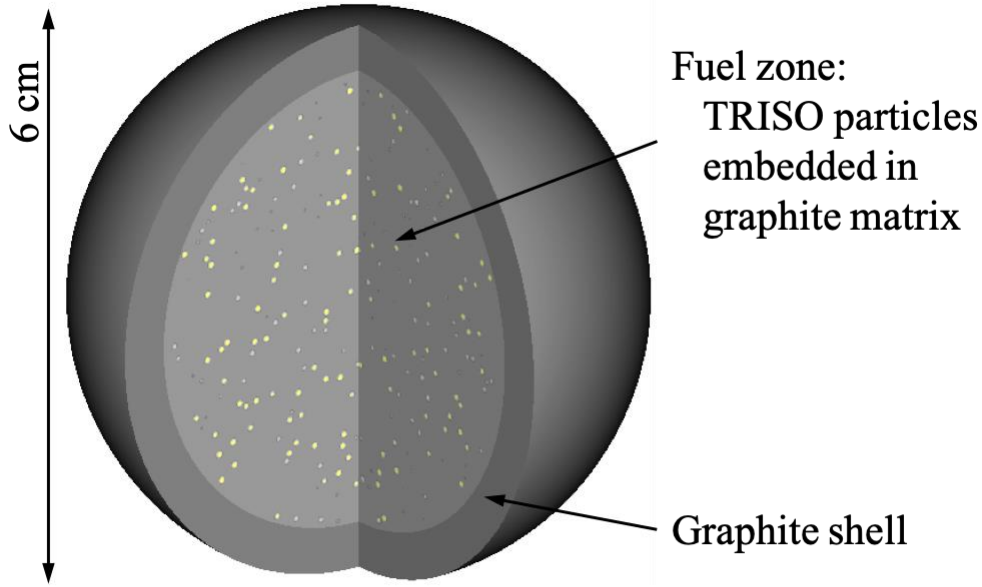
Double-heterogeneous systems

Double-heterogeneous systems with spherical particles in a spherical, cylindrical, or plate fuel component:

- 1st level: Thousands of fuel particles (grains) are randomly dispersed in a graphite matrix
- 2nd level: Hundreds or thousands of fuel elements (pebble, compact, plate) are distributed across the reactor core (regular or irregular)



TRISO particle

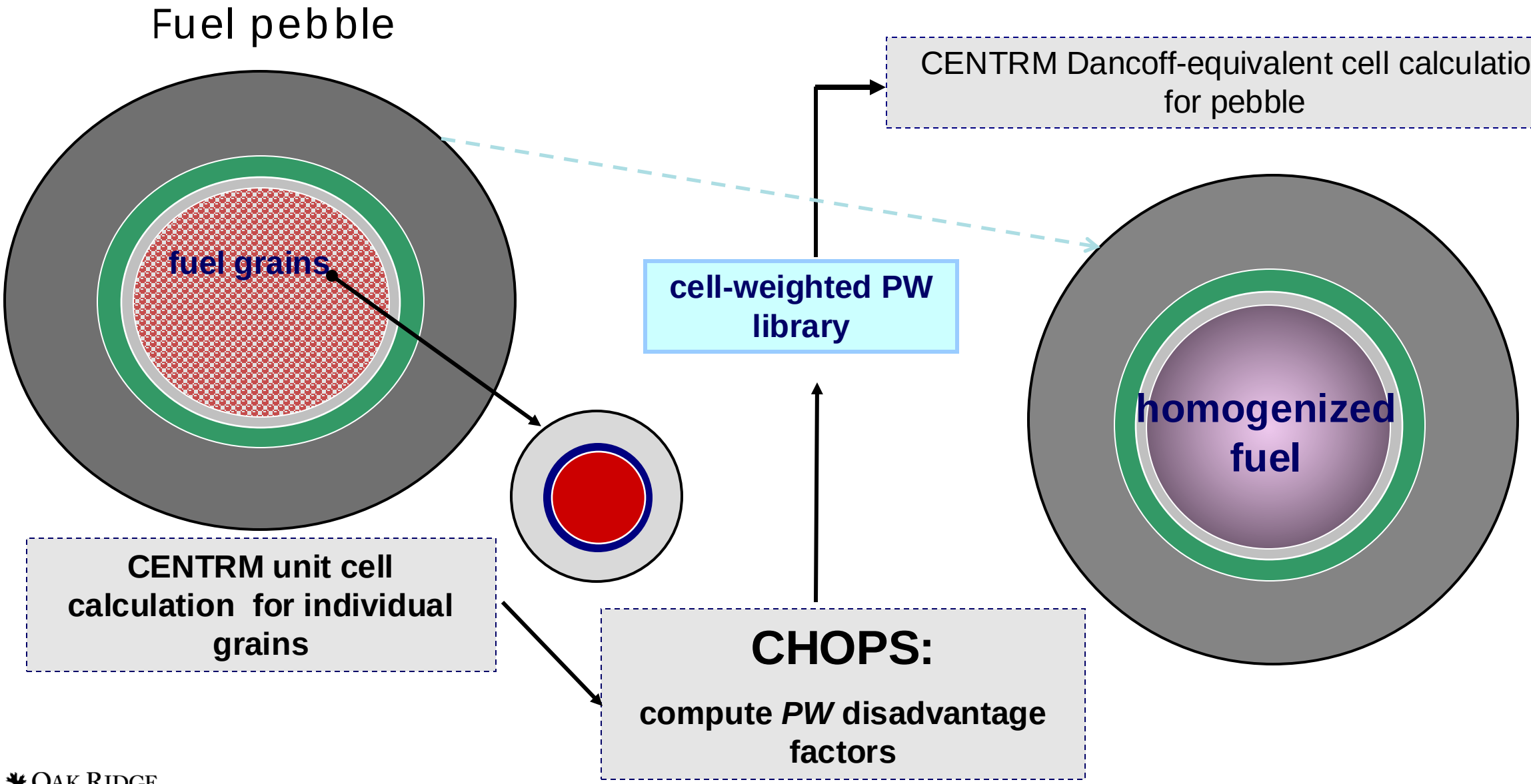


Fuel pebble

Double-heterogeneous self-shielding

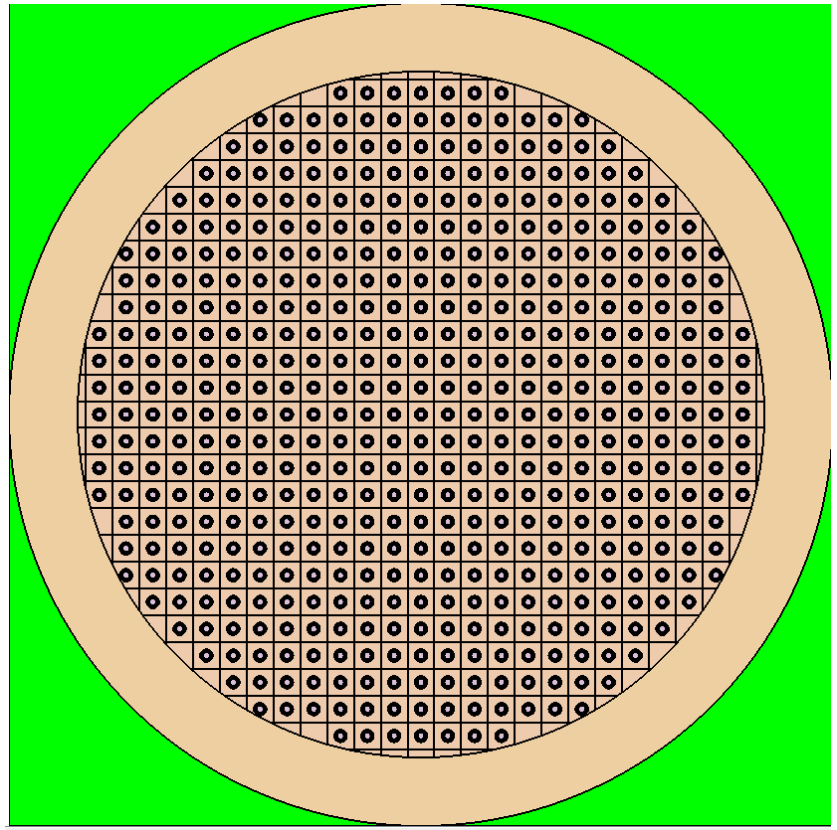
- Self-shielding considerations:
  - Usual fuel-to-fuel element effect (Dancoff)
  - Grain heterogeneity can have a significant impact on resonance self-shielding
    - Grains are generally small (~mm), but still greater than mean free path at resonance energies
    - Causes spatial flux depression in grain (disadvantage factor)
- SCALE's unique approach for double-heterogeneous models:
  - Two subsequent self-shielding calculations: (1) fuel grains in graphite matrix, (2) lattice of pebbles/compacts/plates in core
  - Automated procedure

Double-heterogeneous self-shielding



Example calculation of a double-heterogeneous system

- Infinitely reflected fuel pebble:
  - 17 wt.% enriched UO_2 TRISO particles with 0.25 mm radius
  - 2.5 cm outer radius fuel zone
  - 3 cm outer pebble radius
  - Infinite array of pebbles in a square lattice with He around the pebbles
  - KENO-VI CE calculation
  - KENO-VI MG calculations with different self-shielding options



Effect of double-heterogeneity and self-shielding

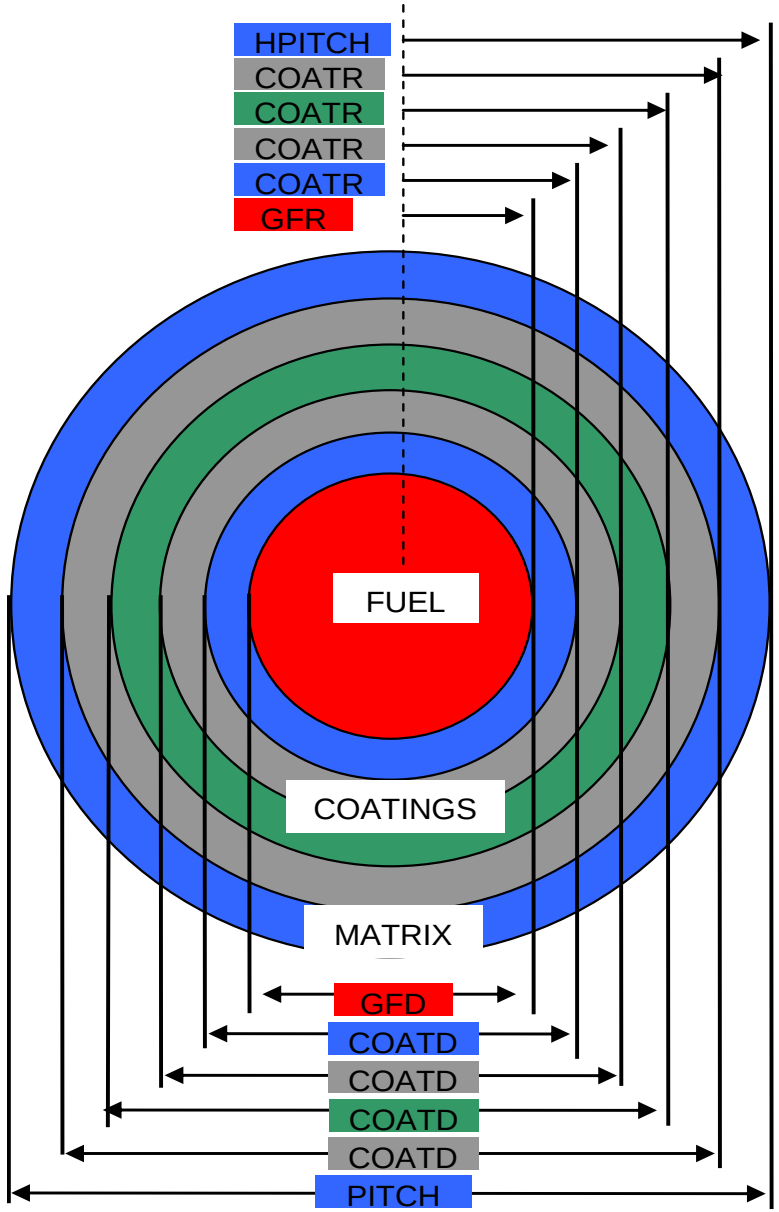
Example comparison of single fuel pebble calculations:

Model		k_{inf}	Δk [pcm]	Runtime [CPU-hours]
CE	particle lattice	1.6745 ± 0.0001	(ref)	150
MG	volume homogenized fuel region (no flux weighting)	1.5650 ± 0.0001	-10,950	24
MG	particle lattice geometry, infinite hom. medium for self-shielding	1.7768 ± 0.0001	10,230	123
MG	particle lattice geometry, multiregion for self-shielding	1.6380 ± 0.0001	-3,650	121
MG	double-het self-shielding	1.6742 ± 0.0001	-30	24

SCALE's doublehet self-shielding method results in great accuracy of MG calculations, while achieving short computation times!

Doublehet input

- First level
  - Describes smaller (inner) geometry, usually grain (particle)
- Second level
  - Describes larger (outer) fuel component
  - 2nd level keywords similar to single-heterogeneous celldata options
  - No fuel mixture entered for fuel element → homogenized fuel/moderator/clad mixture from first level is used automatically



First level: Fuel particle input

Example: UO_2-fueled pebble – composition

```
=csasmg
HTGR fuel pebble
v7-252
read comp
u-238    1 0 2.12877e-2 end
u-235    1 0 1.92585e-3 end
o        1 0 4.64272e-2 end
c        2 0 5.26449e-2 end
c        3 0 9.52621e-2 end
c        4 0 4.77240e-2 end
si       4 0 4.77240e-2 end
c        5 0 9.52621e-2 end
graphite 6 0 8.77414e-2 end
c        7 0 8.77414e-2 end
he-4     8 0 2.65156e-5 end
end comp
```

Compositions of:

- Fuel grain
- All particle layers
- Graphite matrix
- Pebble shell
- Coolant

Example: UO_2-fueled pebble – doublehet celldata

```
read celldata
doublehet  fuelmix=10 end
    gfr=0.0250 1
  coatr=0.0285 2
  coatr=0.0325 3
  coatr=0.0365 4
  coatr=0.0455 5
  numpar=15000
  matrix=6  end grain
pebble
  fuelr=2.5
  cladr=3.0   7
  hpitch=3.0  8
  right_bdy=white
  left_bdy=reflected end
end celldata
```

Doublehet identifier and fuel mixture number

Fuel particle definition

Fuel component description (here: pebble)

Doublehet keywords for 1st level

Keyword	Description
DOUBLEHET	First keyword for each double-heterogeneity cell
right_bdy = keyword	Right boundary condition (**vacuum** or **white**)
fuelmix = mixture#	Homogenized fuel mixture number
END	Required keyword
gfr* = dimension mixture#	Grain fuel radius
coatr*,** = dimension mixture#	Coating radius
matrix = mixture#	Mixture number for the matrix material
numpar = # ***or*** **vf** = fraction	Number of grains of this type or volume fraction of grain type in fuel element
END GRAIN	End of input for 1st level

***substitute "d" for "r" to use diameter**
****substitute "t" for "r" to use thickness**

Doublehet keywords for 2nd level

Keyword*	Description
Fuel element type	**PEBBLE** or **ROD** or **SLAB**
Lattice cell type	**SQUAREPITCH** or **TRIANGPITCH** for **ROD** **SPHSQUAREP** or **SPHTRIANGP** for **PEBBLE** **SYMSLABCELL** for **SLAB** *Annular versions supported for all options*
fuel*r** = dimension	Fuel outer radius
gap*r** = dimension mixture#	Gap outer radius
clad*r** = dimension mixture#	Clad outer radius
h*pitch* = dimension mixture#	Half-pitch (based on center-to-center spacing)
fuelh = dimension	Fuel height for rod or slab
fuelw = dimension	Fuel width for slab
right_bdy = keyword	Right boundary condition (**vacuum** or **white**)
fuelmix = mixture#	Mixture number to be used in fuel element geometry
END	End of input for 2nd level

***substitute "d" for "r" to use diameter**
****substitute "pitch" for "hpitch" for full pitch**

OAK RIDGE National Laboratory

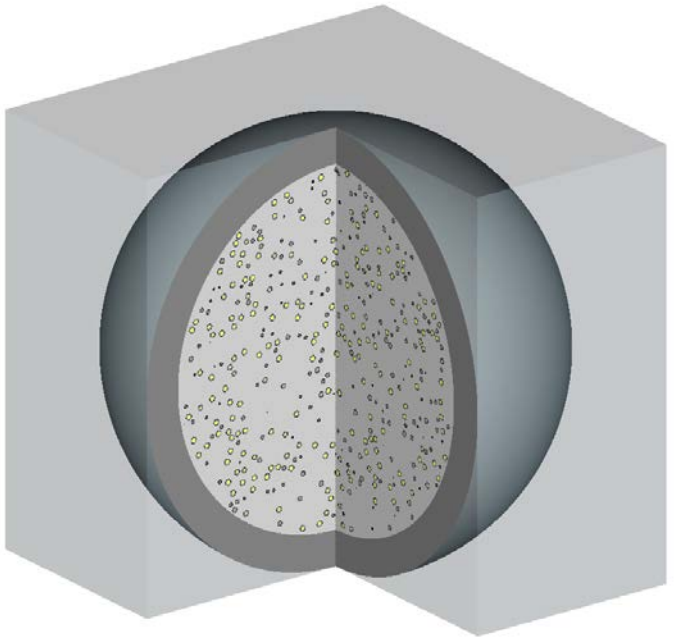
Sub-module 6

- Double-heterogeneous exercises
  - HTGR fuel pebble
  - HTGR prismatic fuel element



Problem 13: HTGR pebble

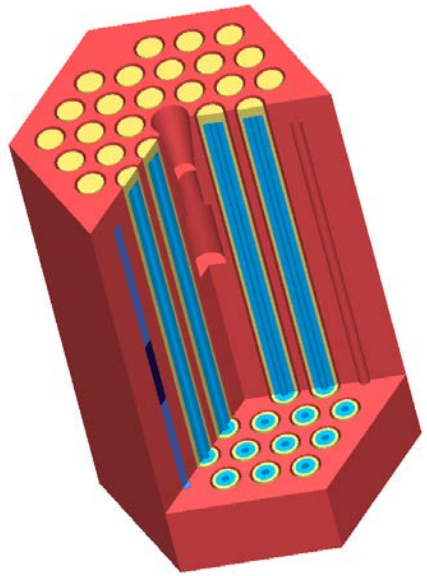
1. Open file "problem_13_starter.inp" and save as "problem_13.inp"
   a) Note that this is a CSAS-Shift 3D model of a pebble in a cube with reflective boundary conditions
2. Add doublehet cell to celldata block
   a) Use the geometrical details on the right
   b) Find the right materials in the composition block
3. Run MG calculation
4. Compare total runtime, transport runtime, and k_{eff} results between your MG calculation and the provided CE calculation



Fuel kernel OR	0.0250 cm
Porous carbon OR	0.0345 cm
IPyC OR	0.0385 cm
SiC OR	0.0420 cm
OPyC OR	0.0460 cm
Pebble fuel zone OR	2.5 cm
Pebble shell OR	3 cm
Cube half pitch	3 cm

OR = Outer Radius

Problem 14: HTGR prismatic fuel element



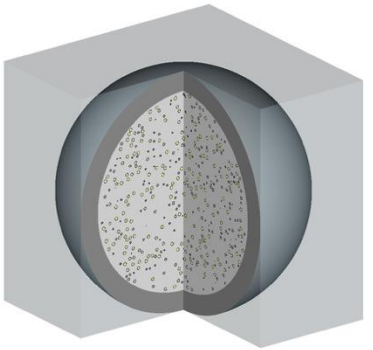
1. Open file "problem_14_starter.inp" and save as "problem_14.inp"
   a) Note that this is a CSAS 3D model of a stack of fuel compacts in a HTGR prismatic fuel element
2. Add doublehet cell to celldata block
   a) Choose the appropriate doublehet lattice type
   b) Use the geometrical details on the right and these additional details:
      - Set doublehet mixture as 10
      - Graphite matrix is mixture 18
      - Structural graphite is mixture 20
      - TRISO packing fraction = 30%
   c) Note Fulcrum does not list a gap by default, but you can add it as for any other cell
3. Run MG calculation

Fuel kernel OR (mix 7)	0.06 cm
Porous carbon OR (mix 13)	0.036 cm
IPyC OR (mix 14)	0.039 cm
SiC OR (mix 15)	0.0415 cm
OPyC OR (mix 16)	0.0460 cm
Fuel OR	1.3 cm
Gap OD (mix 19)	3.4 cm
Cladding OD (mix 21)	4.1 cm
Fuel height	56 cm
Pitch	5.15 cm

SOLUTIONS

Solution 13: HTGR pebble

Case	k_{eff}	Run-time
CE	1.53339 +/- 0.00402	516 minutes with SCALE 6.3.3 16 minutes with SCALE 7
MG 252G	1.53154 +/- 0.00228	3.2 minutes



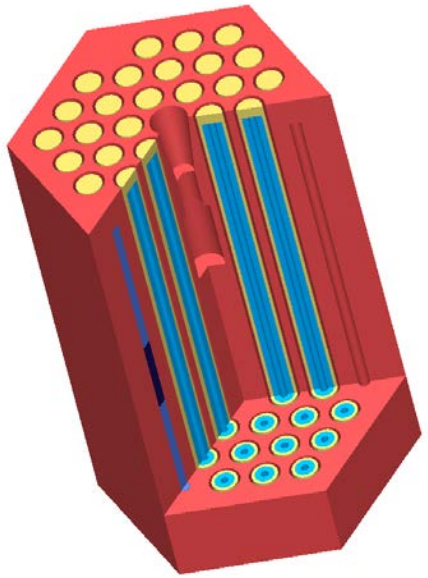
Takeaways:

- MG doublehet vs CE run-time differences
  - For this problem, it is reduced by 100x
  - Notice the "Final timing report" table in Shift-sequences that reports the timing for cross section processing (the self-shielding calculation) and the neutron transport calculation separately
- MG vs CE k_{eff} differences
  - ~185 pcm difference for this set of Monte Carlo histories
  - Expect good agreement for larger numbers of Monte Carlo particle histories
  - The 252-gr thermal library produces good results

```
read celldata
doublehet right_bdy=white fuelmix=20
end
  gfr=0.025    10
  coatr=0.0345 11
  coatr=0.0385 12
  coatr=0.0420 13
  coatr=0.0460 14
  matrix=15 numpar=15000
  end grain
 pebble sphsquarep
        right_bdy=white
        left_bdy=reflected
        hpitch=3.0 17
        fuelr=2.5
        cladr=3.0 16 end
end celldata
```

Solution 14: HTGR prismatic fuel element

```
read celldata
 doublehet fuelmix=10 end
    gfd=0.06       7
  coatr=0.036     13
  coatr=0.039     14
  coatr=0.0415    15
  coatr=0.046     16
 matrix=18
 vf=0.30 end grain
 rod triangpitch right_bdy=white left_bdy=reflected
 pitch=5.15 20 fuelr=1.3 fuelh=56 gapd=3.4 19 cladd=4.1 21 end
end celldata
```



OAK RIDGE National Laboratory

Sub-module 7

- Summary



Recommendations

Things to consider when selecting unit cell modeling treatment(s):

- Are multiple lumps of fuel present?
- Is a repeating array present?
- Are dimensions of the model similar to the neutron mean free path?
- Is a MULTIREGION treatment needed because of complexity?
- Don't forget that slab and annular cells are available

Self-shielding summary

- Resonance self-shielding is essential for accurate multigroup calculations
- SCALE includes several methods for self-shielding, including a rigorous 1D PW transport calculation
- SCALE sequences automatically perform self-shielding calculations based on celldata input provided by user
- SCALE includes a unique capability for self-shielding of double-heterogeneous systems, providing accurate results in a much-reduced runtime
- Dancoff factors must be used to account for nonuniform lattices
- **Use Fulcrum's autocompletion** to see all available self-shielding options and to get all keywords and boundary conditions
- Fulcrum can be used to visualize cross sections and the impacts of resonance self-shielding corrections

OAK RIDGE National Laboratory

Questions?



OAK RIDGE National Laboratory

MAVRIC/Monaco

SCALE 6.3.2



ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals

- Introduce attendees to fixed-source / shielding simulations
- Introduce attendees to variance reduction (VR) techniques
- Introduce attendees to the MAVRIC sequence
- Identify the information required to build a computational model of a fixed-source / shielding problem
- Introduce the attendees to the text-based MAVRIC/Monaco input
- Review the different MAVRIC/Monaco output files and plotting packages

OAK RIDGE National Laboratory

Shielding calculations: Monte Carlo basics



Learning goals

- Review the statistical nature of Monte Carlo simulations
- Review the strengths and weaknesses of Monte Carlo simulations for shielding problems
- Introduce attendees to VR techniques commonly available in Monte Carlo codes and identify which are available in MAVRIC/Monaco

Shielding calculations

- Calculation of radiation transport through materials
- Usually a fixed source; can involve multiplication
- Desired output – usually dose or dose equivalent rates
- Can be very deep penetration (fluxes may be attenuated by many orders of magnitude)
- Methods
  - Point kernel (historical)
  - Discrete ordinates
  - Monte Carlo

Monte Carlo simulation

- Simulates individual particles interacting (e.g., scattering or being absorbed) in materials by random sampling
- History: start with a source particle, follow it as it makes contributions to various responses (random walk)

- Final answer is an average of all histories

$$\bar{x} = \frac{1}{N}\sum x_i$$

- Final answer has statistical uncertainty; similar to detector counting statistics

$$\sigma_{\bar{x}} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum x_i^2 - \bar{x}^2}$$

Monte Carlo simulation (cont.)

- Strengths of MC
  - Straightforward physics
    - particle-based interactions
    - physics expressed as probability distribution functions (pdf)
  - Geometry at any level of detail – no meshing approximations
- Drawbacks of MC
  - Time: depends on the problem, number of results (tallies), level of geometry detail
- Difficult problems
  - Streaming
  - Highly scattering
  - Deep penetration

Examples of Monte Carlo (MC) codes

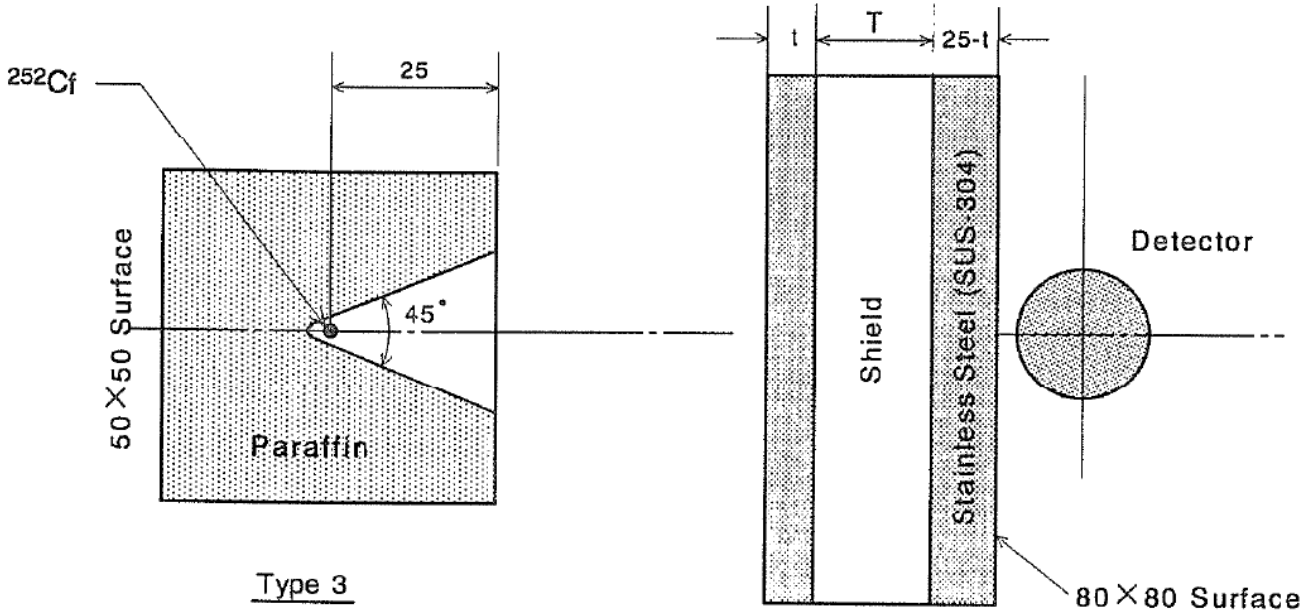
- Monaco/KENO/Shift – Oak Ridge National Laboratory
- MCNP/MCNPX – Los Alamos National Laboratory
- VIM – Argonne National Laboratory
- COG, TART – Lawrence Livermore National Laboratory
- ITS – Sandia National Laboratory
- Serpent – VTT Technical Research Centre of Finland
- OpenMC – Computational Reactor Physics Group at MIT
- EGS, EGSnrc – SLAC National Accelerator Laboratory
- FLUKA – The FLUKA team
- Geant – Geant4 Collaboration

Monte Carlo codes (cont.)

- Different geometry descriptions
- Different energy treatments (continuous versus multi-group)
- Different energy regimes
- Different particles (neutrals, charged, exotic)
- Different levels of user programming

Variance reduction (VR)

- Changes to the physical random walk to get more particles to add score to the result
- Scores are now weighted to account for the probability of this history occurring in an analog simulation

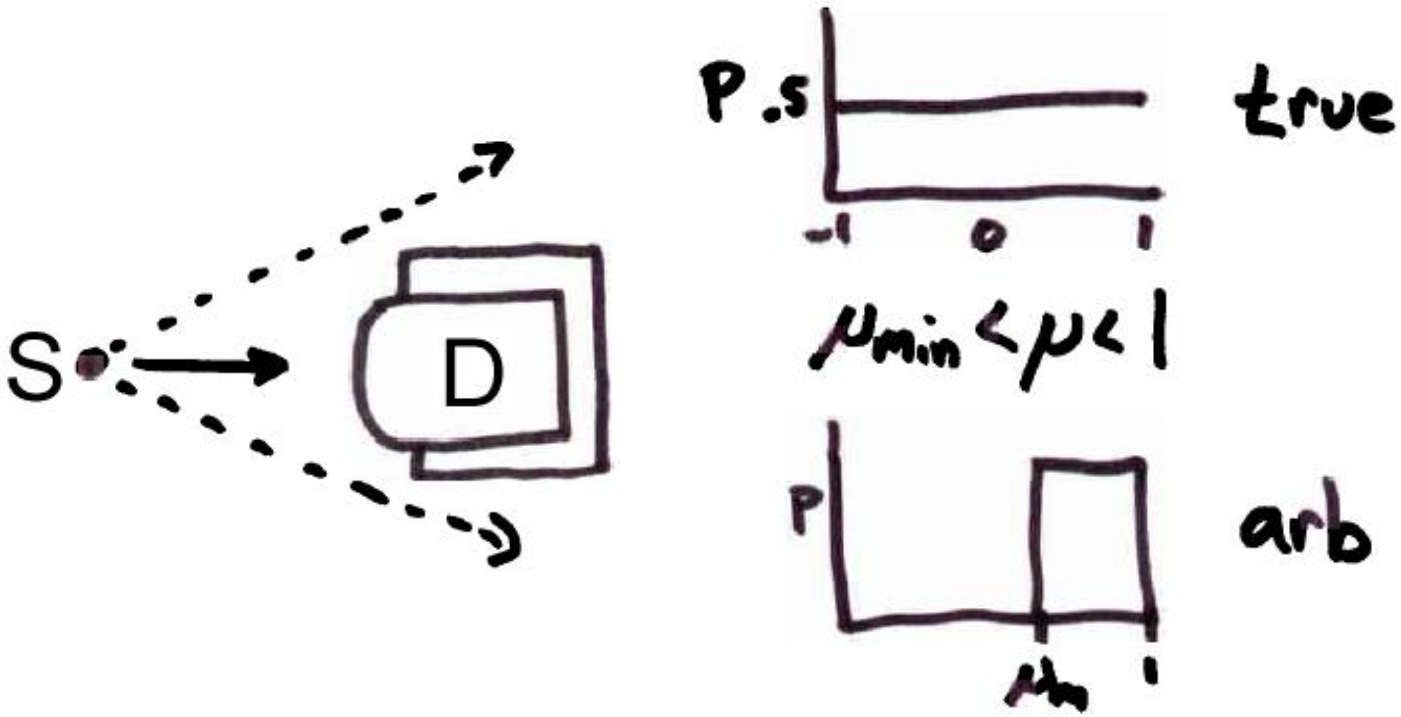


Ueki shielding measurements
K. Ueki et al.,
"Systematic Evaluation of Neutron Shielding Effects for Materials," Nuclear Science and Engineering, vol. 124, no. 3, pp. 455-464, 1996.

VR methods – forcing

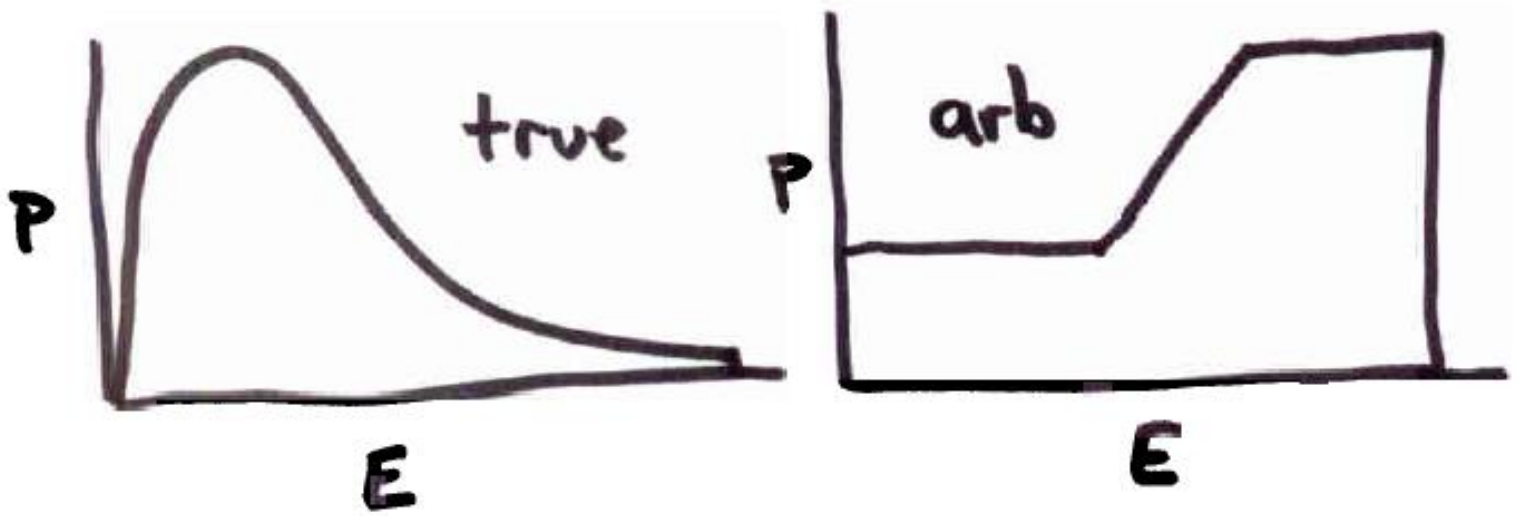
- Force a particle to do something when the alternative never leads to a score

Example – only start source particles in "direction of interest"



VR methods – source biasing

- Pick more source particles with desired energy, direction, and starting location
- Weight is true probability density function (pdf)/arbitrary pdf

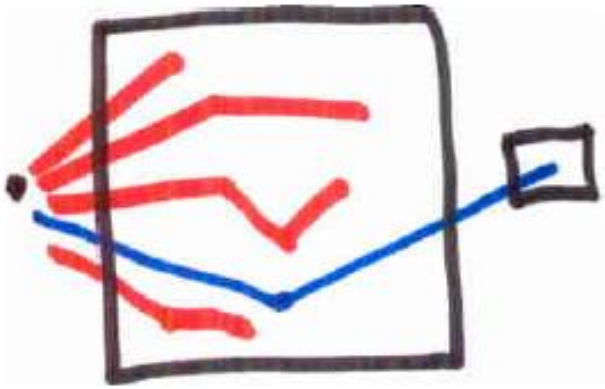


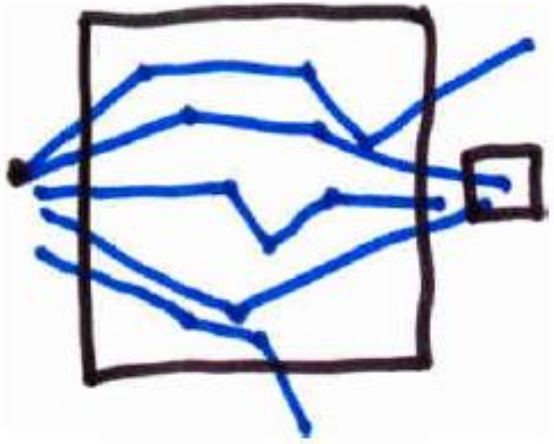
VR methods – Implicit capture

- Do not let particles get absorbed

$$w' = w\frac{\sigma_s}{\sigma_a + \sigma_s}$$

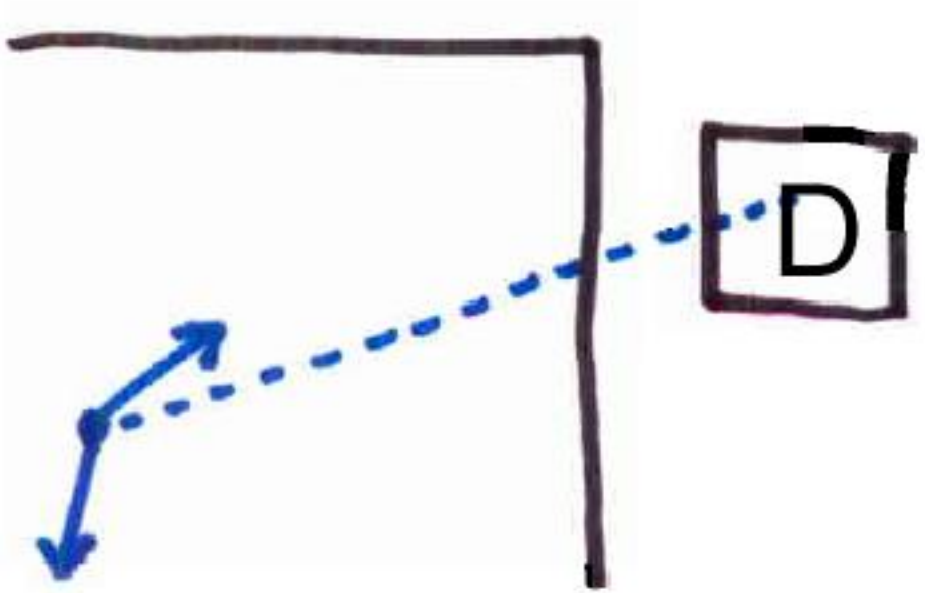
- Instead, reduce weight and keep tracking
- This requires roulette to handle very low weight particles: either remove them or increase their weights





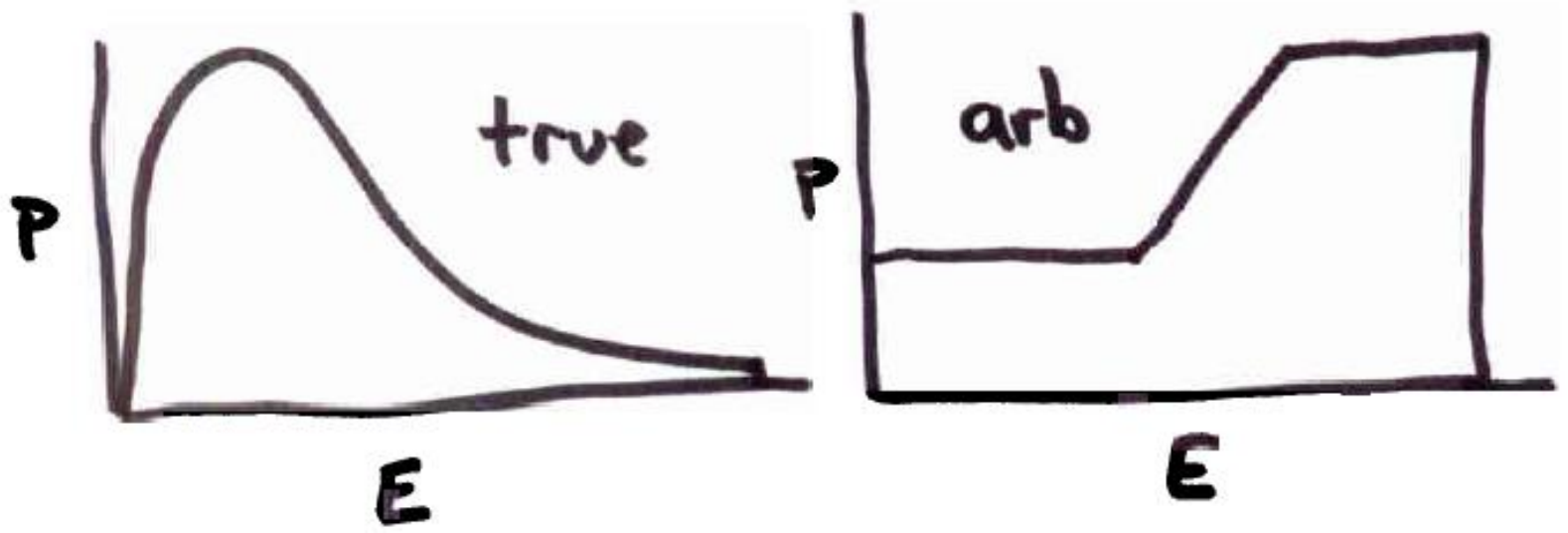
VR methods – path length stretching

- Also known as exponential transform
- If a particle is going toward the response area, then reduce the cross section (to increase the mean free path length) and reduce the particle weight



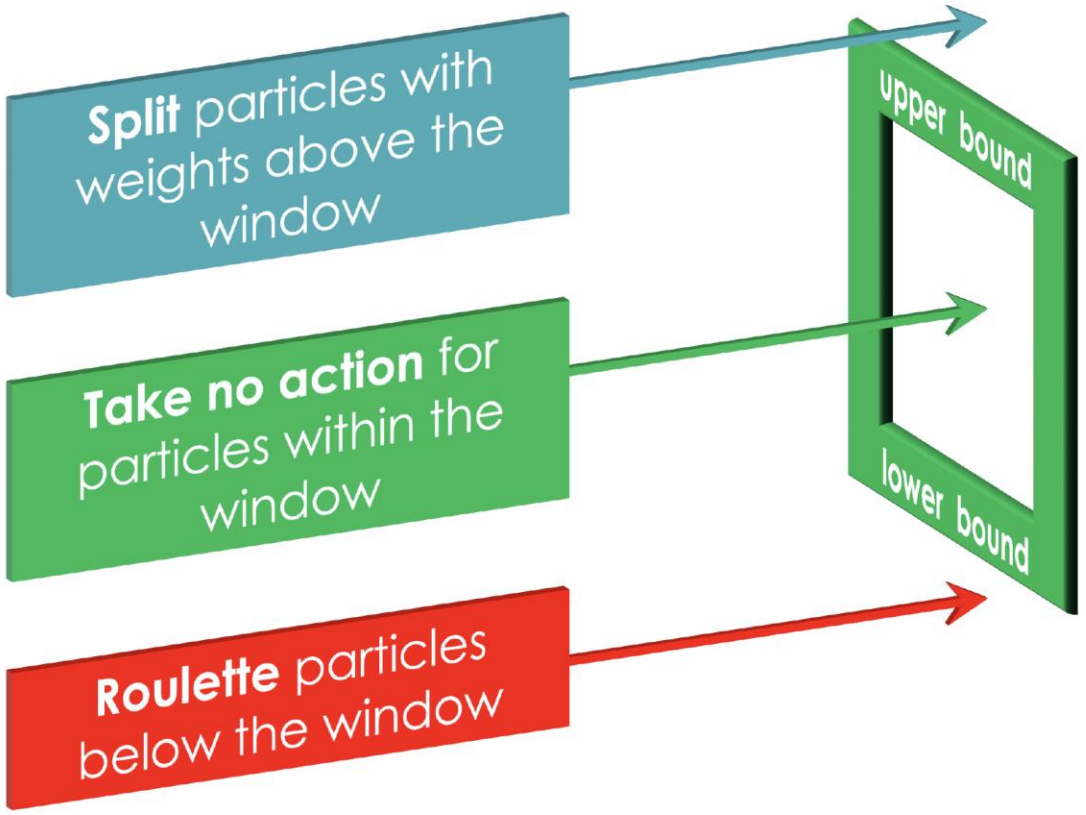
VR methods – PDF biasing

- Change certain PDFs (scattering angle, energy change, etc.) to get more particles to score
- Weight is true PDF/arbitrary PDF
- Must be careful not to use competing methods



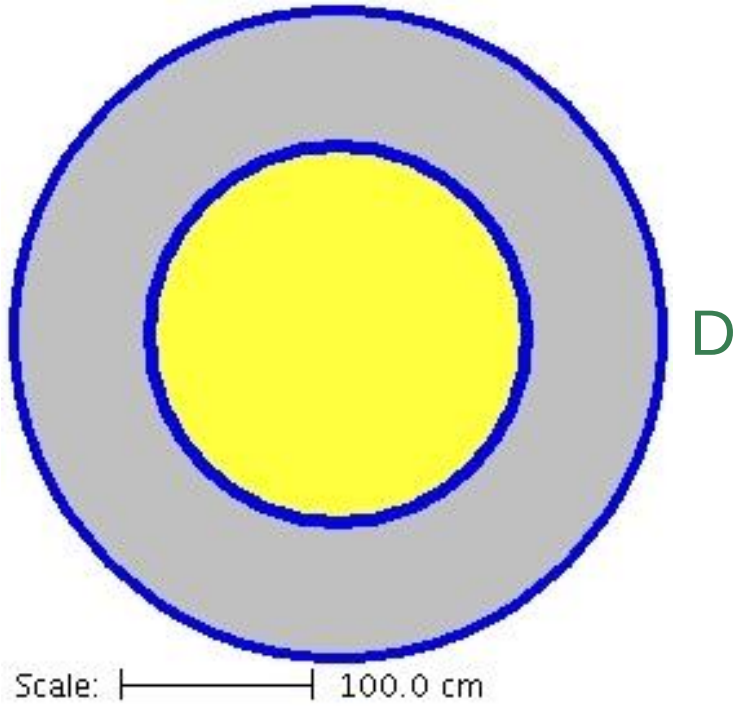
VR methods – weight windows

- Assign spatial and energy region importance
- As a particle crosses a boundary, the weight is compared to the weight target
- High-weight particles are 'split' into multiple lower-weight particles
- Low-weight particles are rouletted (either the history is terminated, or it is continued with an increased particle weight)

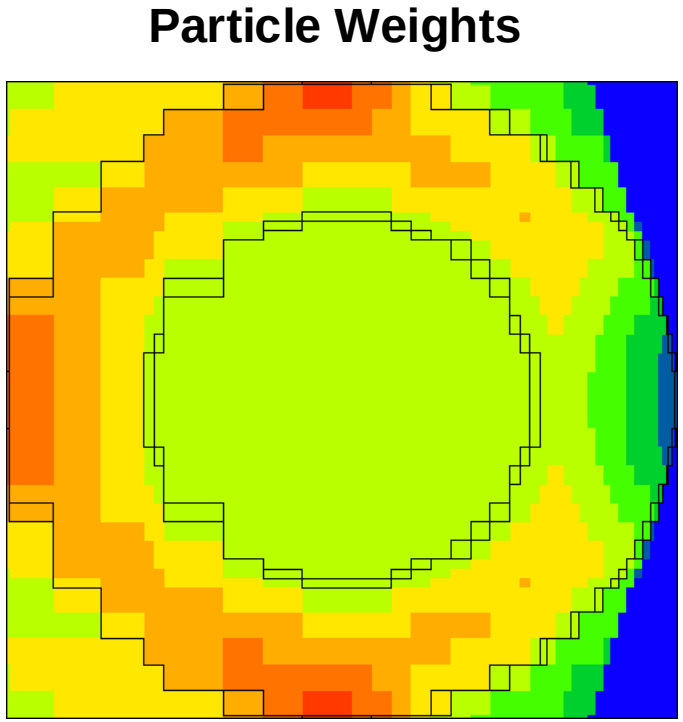


VR methods – importance map (adjoint)

- Uses the importance of a particle (i.e., space, energy, direction) to set weight windows: high importance → low target weight



D



VR methods – summary

- Requires knowledge about how the calculation will most likely proceed
- May require an iterative process to tune biasing
- "Bad" biasing can slow the rate of convergence
- Automated VR would be nice

Importance sampling

- If the exact importance is known, then we do not need a Monte Carlo simulation — we could just convolve the true source distribution with the adjoint function
- True importance (adjoint) is just as difficult to compute as the original problem
- Fortunately, an approximate importance map can be used as weight window targets to greatly accelerate the convergence of the forward MC simulation
- CADIS, Consistent Adjoint Driven Importance Sampling, creates an importance map and source biasing that work together

Importance sampling (cont.)

- MCNP – stochastic weight window generator
- ADVANTG – uses 3D TORT, CADIS methodology, and MCNP
- MAVRIC – uses 3D Denovo, CADIS methodology, and Monaco
- ADVANTG 3.0 – uses 3D Denovo, CADIS methodology, and MCNP

SCALE: Monaco/MAVRIC

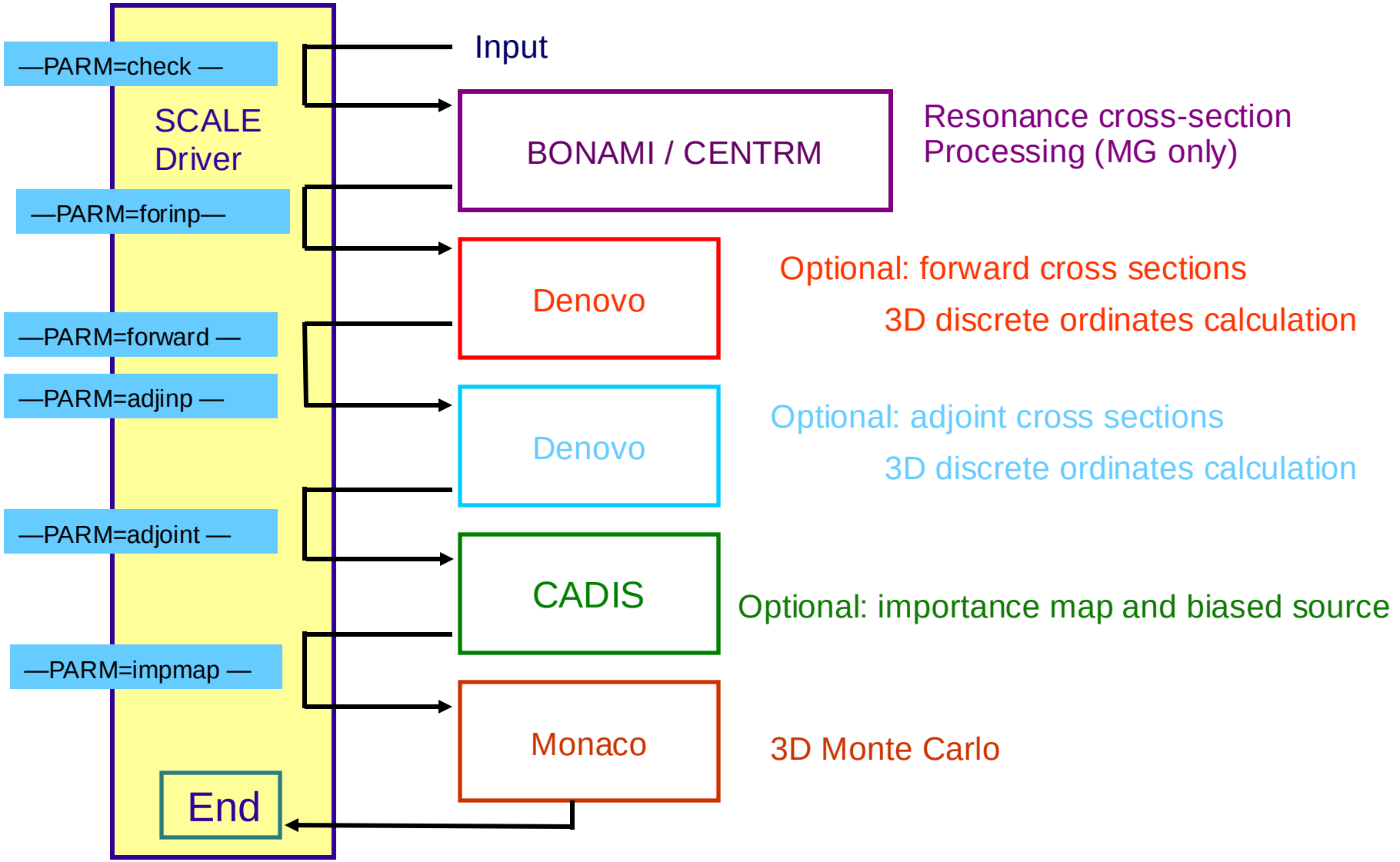
Introduced in SCALE 6 (released: Jan 2009)

- Monaco – multi-group Monte Carlo transport
  - Based on MORSE physics (predates SCALE and later included in SCALE)
  - SCALE General Geometry Package (SGGP), same as KENO-VI
  - Continuous-energy physics added for 6.2 (Jun 2016)
  - Parallel processing enabled for MAVRIC-Shift in 6.3 (Jan 2022)



- MAVRIC – Monaco with Automated Variance Reduction using Importance Calculations
  - Computes cross sections
  - Performs a Denovo adjoint calculation, forms an importance map and biased source distribution
  - Runs Monaco

Sequence MAVRIC calls Monaco



OAK RIDGE National Laboratory

Elements of a shielding calculation

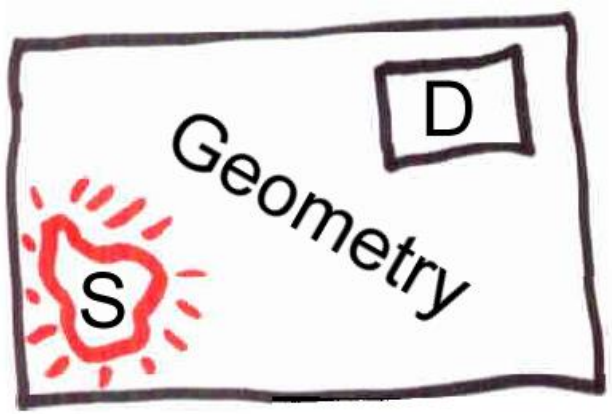


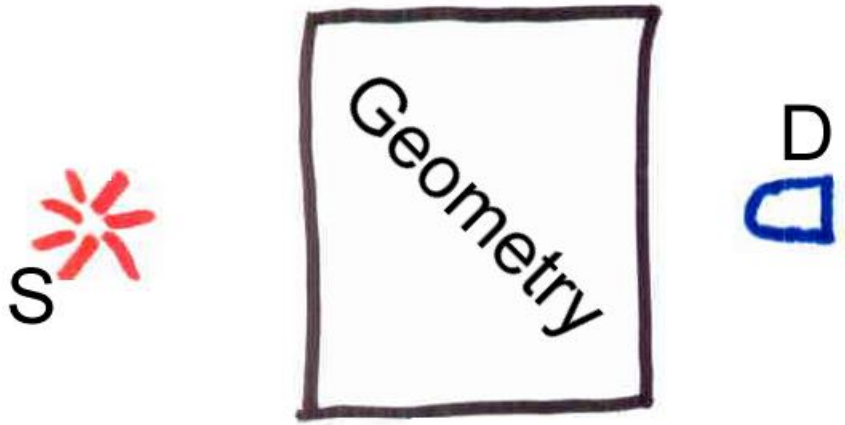
Learning goals

- Identify the information required to build a computational model of a fixed source / shielding problem

Elements of a shielding calculation

- Physical model
- Source
- Response
- Calculational parameters





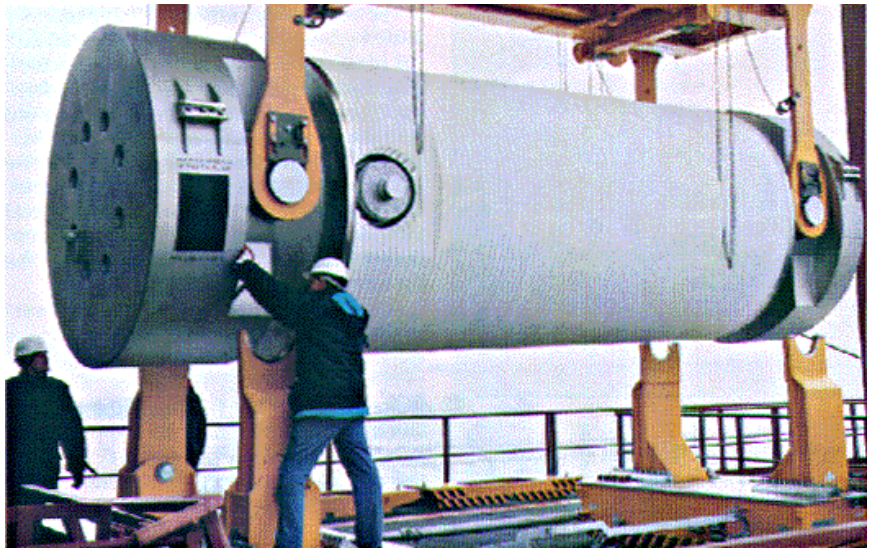
Physical model

- Geometry description
  - Boolean operation of solid and surface equations
  - CAD format – faceted surfaces
- Material compositions
  - Physical mixture: weight fractions, atomic fractions
  - Chemical formula
  - Density
  - Isotopic distribution of elements
- Interaction cross sections

Sources

- Particle type
- Spatial distribution
- Energy distribution
- Directional distribution



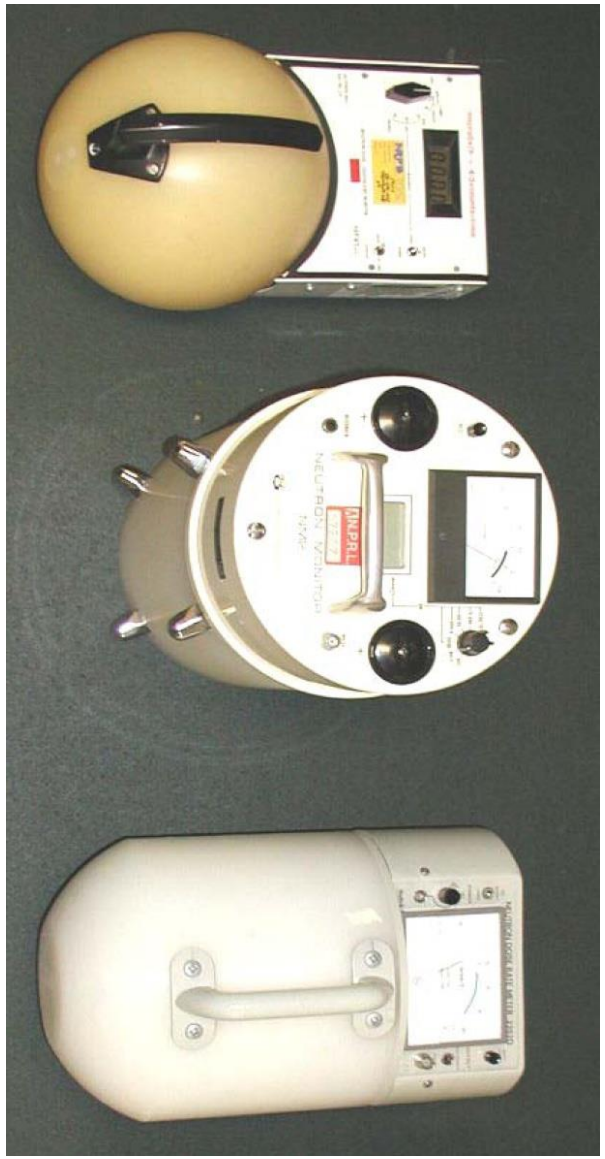


Example of a transportation container for radioactive materials
BMT, CargoHandBook, www.cargohandbook.com



Responses

- Type of response
  - At a point
  - For a geometric region
  - Superimposed mesh
- Quantity to calculate
  - Flux
  - Dose rate
  - Reaction rate
- Dimensionality
  - Total
  - Function of (x,y,z), E, Ω, etc.



Calculational parameters

- Minimum accuracy
- Maximum runtime
- Problem truncation
- VR

OAK RIDGE National Laboratory

MAVRIC Input



Learning goals

- Introduce attendees to the text-based MAVRIC/Monaco input file when using standard biasing techniques
  - Identify where in the MAVRIC/Monaco input file to enter information required to build a computational model of a shielding problem
  - Additionally, identify the MAVRIC/Monaco syntax used to enter information required to build a computational model of a shielding problem
- Review the different MAVRIC/Monaco output files and identify the location of different types of requested output
- Introduce attendees to the SCALE plotting packages applicable to MAVRIC/Monaco output and demonstrate how to create commonly used output plots

MAVRIC input

- Material compositions, CellData processing (XSProc)
- Geometry: Regions made of intersections of solids, arrays, volumes (same as KENO-VI)
- Definitions: Locations, Response Functions, Grid Geometries, Distributions
- Sources description: spatial, energy, direction
- Tallies: region tallies, point detectors, mesh tallies
- Basic calculational parameters
- VR parameters

MAVRIC input: Block format

```
=mavric    parm=keyword
Some title for this problem
v7.1-28n19g

read compositon
   ...
end composition
read celldata
   ...
end celldata

read geometry
   ...
end geometry
read array
   ...
end array
read volume
   ...
end volume
read plot
   ...
end plot

read definitions
   ...
end definitions
read sources
   ...
end sources
read tallies
   ...
end tallies

read parameters
   ...
end parameters
read biasing
   ...
end biasing
read importanceMap
   ...
end importanceMap

end data
end
```

MAVRIC input: Keyword input style

- Single values
  - `batches=1000`
  - `strength=6.7e15`
- Array values
  - `abscissa 27r1.0 19r0.0 end`
  - `yPlanes 15.0 20.0 25.0 50.0 end`
- Fixed-length parameter list
  - `cuboid 10 0  20 0  30 0`
  - `xShellCylinder 0 10  -35 35`
- Keywords without values
  - `firstCollision`

- Text Strings
  - `title="cool program, eh?"`
  - `sourceMeshFile="C:\My Documents\source.msm"`

SCALE is not case sensitive*

***except in places where it is**

MAVRIC input: Basics

Format:

=Sequence name

Problem title

SCALE Library name

- Denovo *always* uses this library; Monaco *might* use this library
- "library" or "ceLibrary" keywords in parameters block override this library for Monaco, which is useful for coarse group Denovo and fine group Monaco (mixing group structures for speed)

Example:

```
=mavric
Some long title for this particular
  problem
V7.1-28n19g
```

Always a multigroup library!

Material composition block

- Standard composition – from SCALE library
- Arbitrary compound – using any isotope from SCALE library
  - Atomic fractions (atom)
  - Weight percents (wtpt)
- Solution (soln) – mixture of SCALE library solutions

Composition block example

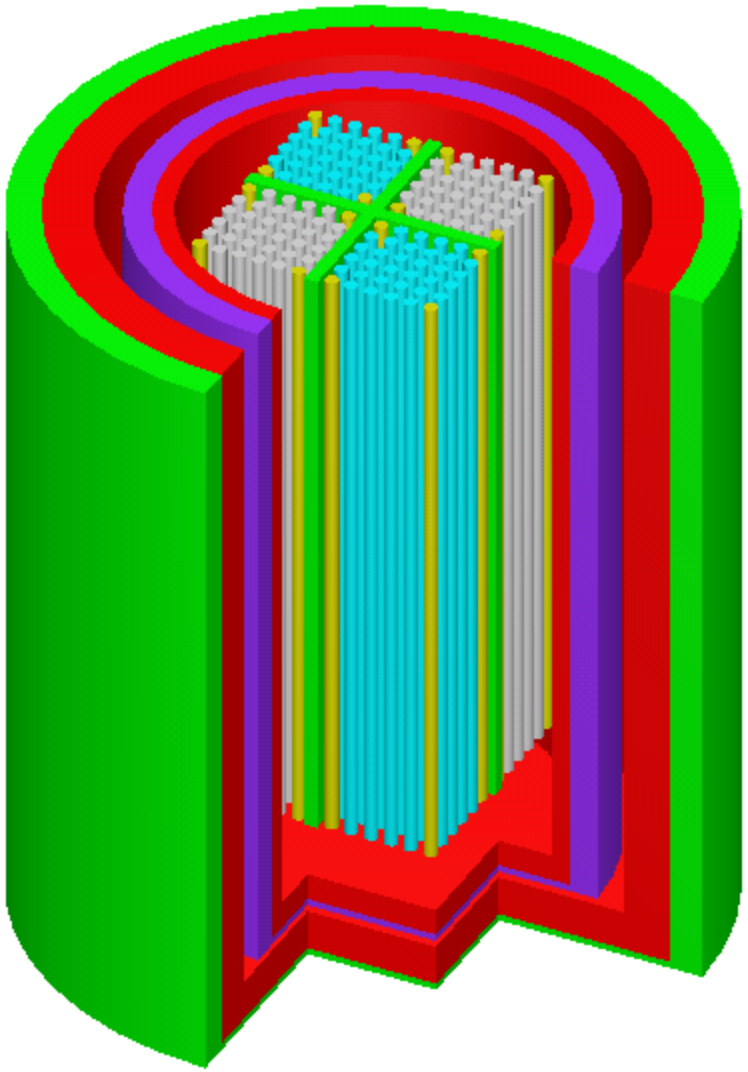
```
read composition
    para(h2o)   1 1.0 293   end

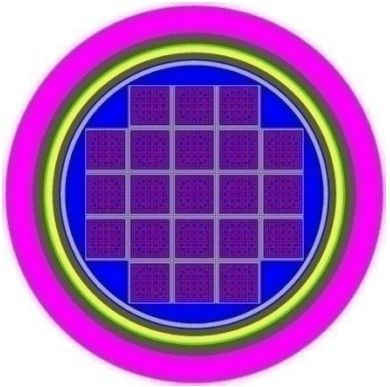
    wtptkrafton 2 1.7 16
         1000 4.21   3000 1.82    5000 7.12   6000 24.07
         7014 0.6    8016 28.38  12000 6.16  13000 3.7
        14000 0.46  16000 0.43   19000 0.52  22000 3.44
        26000 6.75  28000 4.98   56138 4.32  64000 3.04
              1 293   end
end composition
```

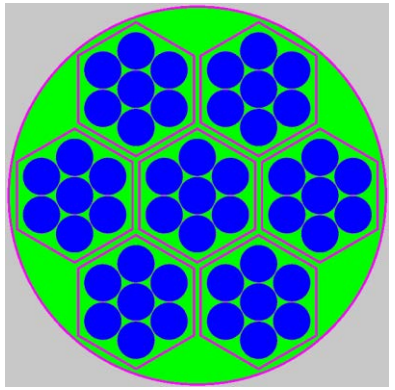
Physical geometry blocks

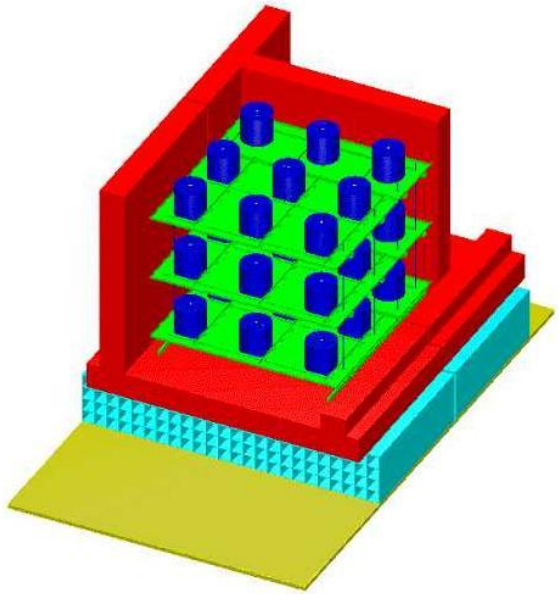
- Uses the SCALE General Geometry Package
  - Shared with KENO-VI
- Solids: cuboids, cylinders, spheres, cones, etc.
- Regions: intersections of solids
- Units: regions sharing the same coordinates
- Boundary: limits of a unit
- Holes/Arrays: embed other units into a given unit
- Global unit: top-level unit
- Blocks: Geometry, Array, Volume

Physical geometry visuals







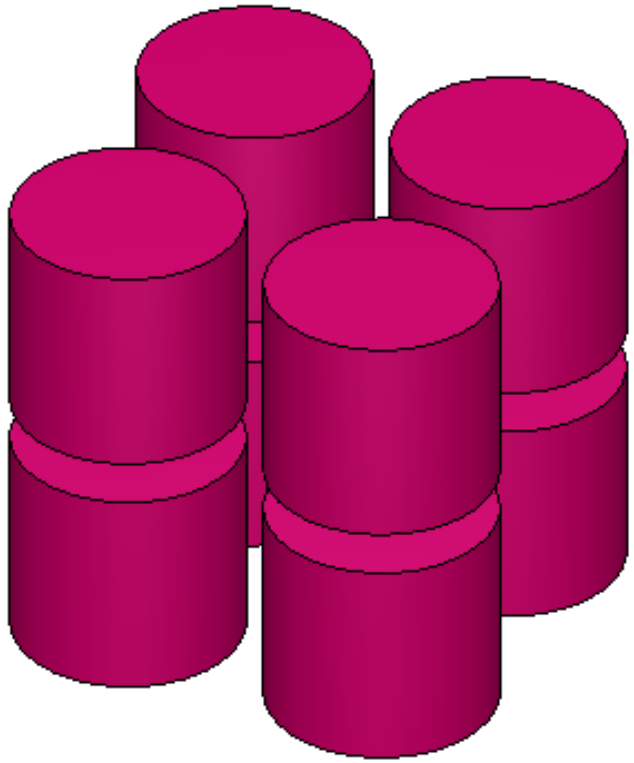


Physical geometry example

```
read geometry
  unit 1
    cylinder 10 5.748 5.3825 -5.3825
    cuboid   20 6.87 -6.87 6.87 -6.87 6.505 -6.505
    media 1 1 10
    media 0 1 20 -10
  boundary 20

  global unit 2
    cuboid 10 13.74 -13.74 13.74 -13.74 13.01 -13.01
    array 1 10  place 1 1 1 -6.87 -6.87 -6.505
  boundary 10
end geometry

read array
   ara=1 nux=2 nuy=2 nuz=2 typ=square gbl=1
   fill 1 1 1 1 1 1 1 1 end fill
end array
```



Definitions block

- Items used multiple times in other blocks
  - Locations
  - Responses
  - Grid geometries
  - Cylindrical geometries
  - Distributions
    - Spatial
    - Energy
    - Direction
  - EnergyBounds
  - TimeBounds

```
read definitions
    location  1
       ...  end location

    response 19
       ... end response

    gridGeometry 1
       ... end gridGeometry

    distribution 7
       ... end distribution

    energyBounds 2
       ... end energyBounds

    cylGeometry 2
       ... end cylGeom
end definitions
```

Location format

```
location id
    title="Optional title in double quotes"
    position x y z
end location
```

Light gray is an optional keyword
Red is a value to be entered

Examples:

```
location 1
    title="Radial detector"
    position 162.0 0.0 0.0
end location

location 47
    position 0.0 0.0 295.6
end location

location 2
    title="Corner detector"
    position 162.0 0.0 295.6
end location
```

Response formats

```
response id
   title="Standard multi-group response definition"
   neutron/photon
   values v1 v2 v3 … vG  end
end response
```

This method does not work when Denovo and Monaco use different group structures.

Or

```
response id
   title="Use a dose response from the cross section library"
   doseData=mt
end response
```

Or

```
response id
    title="Histogram response using energy bins (eV)"
    neutron/photon
    bounds E1 E2 E3 … En En+1  end -or- energyBoundsID=id
    values v1 v2 v3 … vn       end
end response
```

Or

```
response id
    title="Value/function pair using energy bins (eV)"
    neutron/photon
    bounds E1 E2 E3 … En  end -or- energyBoundsID=id
    values v1 v2 v3 … vn  end
end response
```

Response formats (cont.)

```
response id
   title="Cross section from the library"
   material=m  ZAID=zaid  MT=mt
end response
```

Or

```
response id
   title="Cross section from the library"
   material=m  MT=mt  macro
end response
```

Or

```
response id
   title="Cross section from the library"
   material=m  nuclide=Sy-XYZ  reaction=Rx_name
end response
```

With any response:

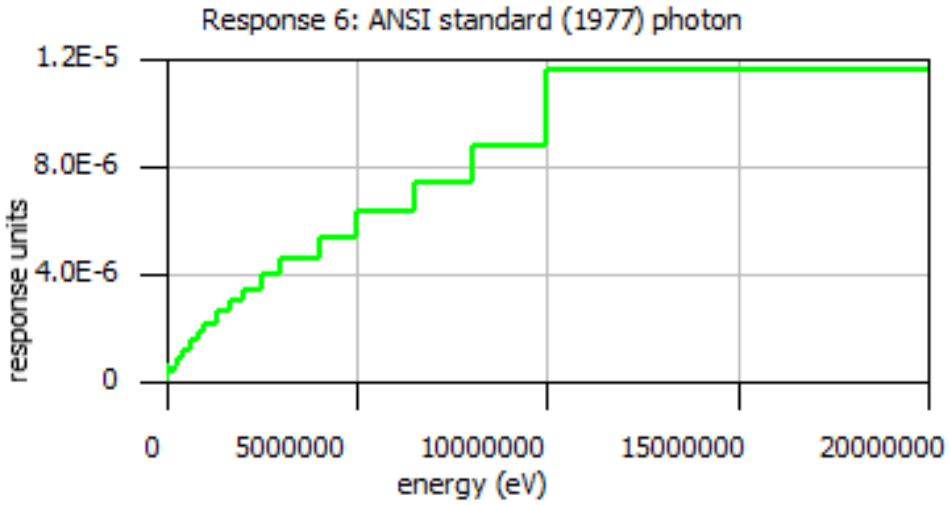
```
multiplier=m
makeChart
eHigh=E
eLow=E
```

Outside all responses:

```
multiplier=m
makeCharts
```

Response examples

```
response 19
   title="ANSI standard (1977) photon flux-to-dose-rate factors"
   values  27r0.0                  1.1620E-05  8.7446E-06  7.4597E-06
         6.3506E-06  5.3995E-06  4.6017E-06  3.9523E-06  3.4588E-06
         3.0131E-06  2.6200E-06  2.1944E-06  1.8270E-06  1.5149E-06
         1.1595E-06  8.7045E-07  6.2187E-07  3.7081E-07  2.6878E-07
         5.9327E-07  end
end response
```



Or

```
response 6
  doseData=9504
end response
```

Or

```
response 24
    photon
    bounds  10e6   8e6   6e6   4e6   3e6   2e6   1e6   0.5e6  end
    values      1.0   2.0   3.0   4.0   3.0   2.0   1.0        end
end response
```

Response examples (cont.)

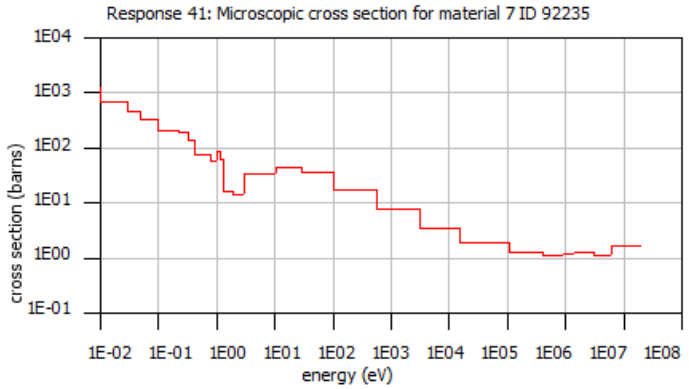
```
response 41
   title="Cross section from the library"
   material=7  ZAID=92235  MT=18
end response
```

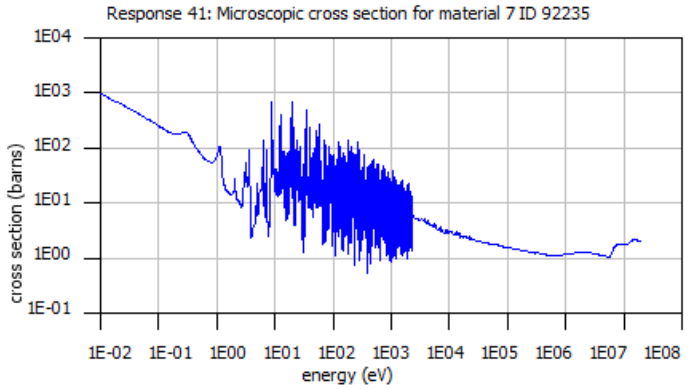
Or

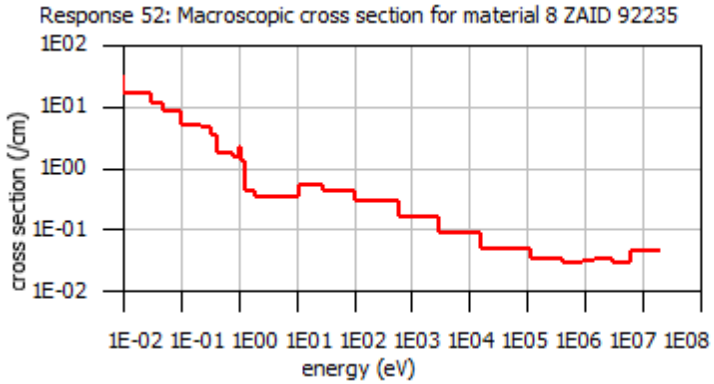
```
response 46
   title="Cross section from the library"
   material=7  MT=18  macro
end response
```

Or

```
response 52
   title="Cross section from the library"
   material=8  nuclide=U-235
     reaction=fission   macro
end response
```







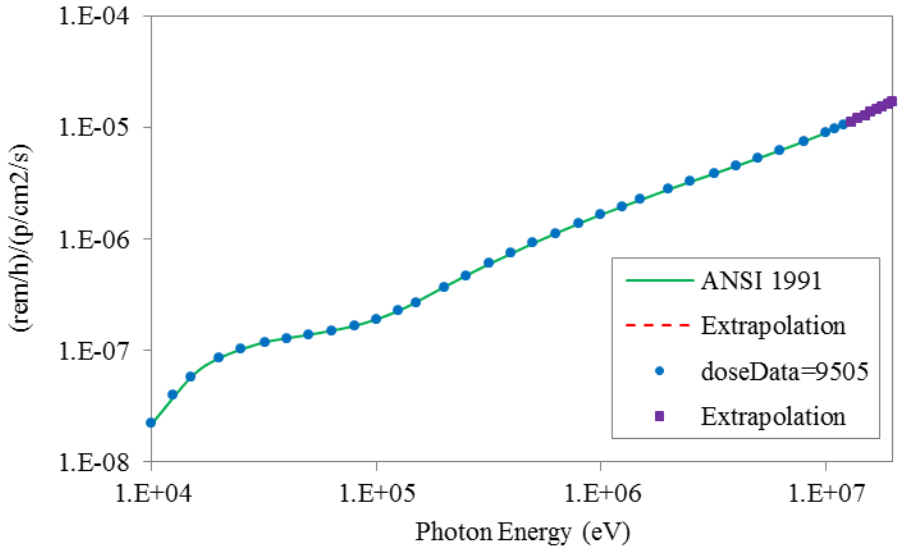
Keyword *doseData* (*specialDose* is *obsolete*)

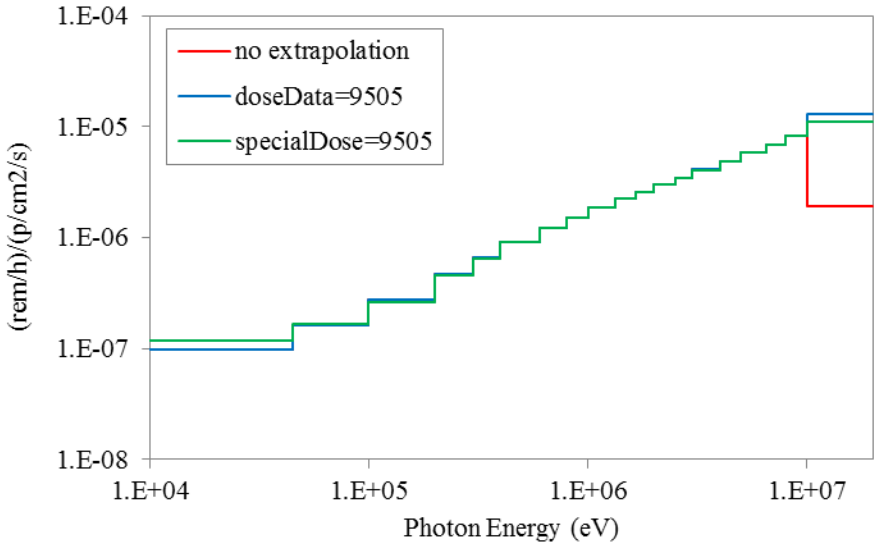
$$\dot{D} = \int f(E)\phi(E)dE$$

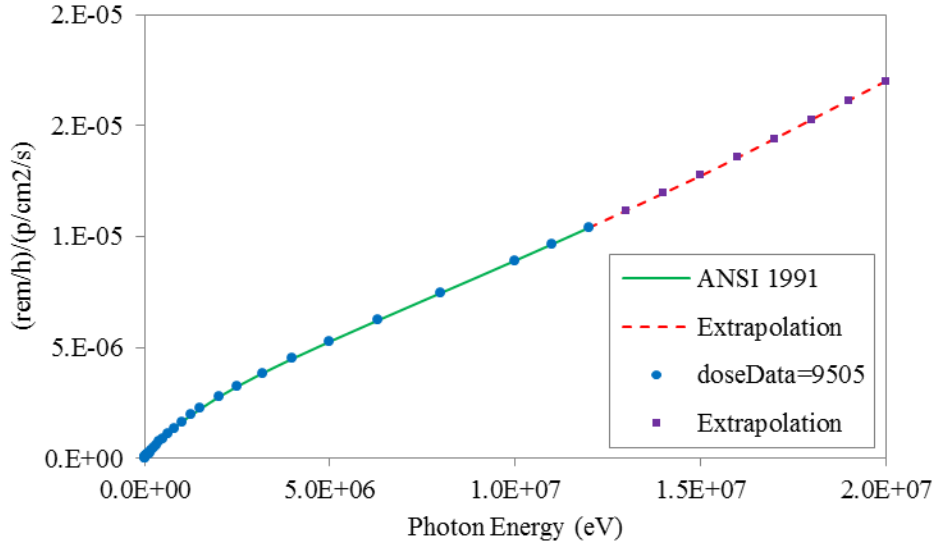
Response		Neutron MT	Neutron Units	Photon MT	Photon Units
Henderson conversion factors		9027	(rad/h)/(n/cm^2/s)	9502	(rad/h)/(p/cm^2/s)
Claiborne-Trubey conversion factors				9503	(rad/h)/(p/cm^2/s)
ANSI standard (1977) flux-to-dose-rate factors		9029	(rem/h)/(n/cm^2/s)	9504	(rem/h)/(p/cm^2/s)
ANSI standard (1991) flux-to-dose-rate factors		9031	(rem/h)/(n/cm^2/s)	9505	(rem/h)/(p/cm^2/s)
ICRU-44 Table B.3 (air) Kerma		9032	(Gy/h)/(n/cm^2/s)		
		9033	(rad/h)/(n/cm^2/s)		
ICRU-57 Table A.21 (air) Kerma				9506	(Gy/h)/(p/cm^2/s)
				9507	(rad/h)/(p/cm^2/s)
Ambient dose equivalent (ICRU-57)		9034	(Sv/h)/(n/cm^2/s)	9508	(Sv/h)/(p/cm^2/s)
		9035	(rem/h)/(n/cm^2/s)	9509	(rem/h)/(p/cm^2/s)
Effective dose (ICRU-57)		9036	(Sv/h)/(n/cm^2/s)	9510	(Sv/h)/(p/cm^2/s)
		9037	(rem/h)/(n/cm^2/s)	9511	(rem/h)/(p/cm^2/s)
Dosimetry for Criticality Accidents (ANSI/HPS N13.3-2013, Table B1 & B2)	Personal absorbed dose	8001	(Gy/h)/(n/cm^2/s)	8511	(Gy/h)/(p/cm^2/s)
	Ambient absorbed dose	8002	(Gy/h)/(n/cm^2/s)	8512	(Gy/h)/(p/cm^2/s)
	(air) Kerma			8501	(Gy/h)/(p/cm^2/s)

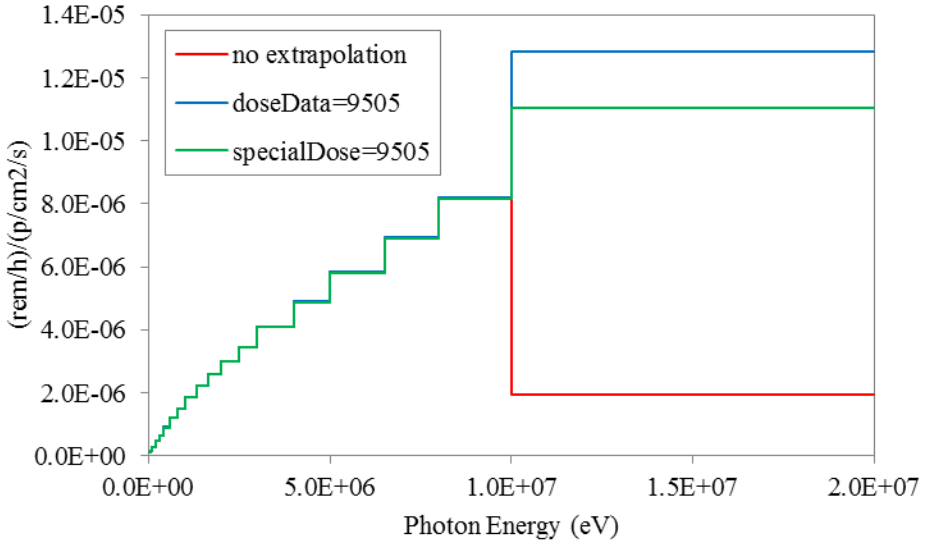
New! SCALE 6.3

Keyword doseData=9505









specialDose vs. *doseData*

keyword(s)	Multi-Group (MG)	Continuous Energy (CE)
specialDose=*MT*	uses pre-computed responses contained in the MG library: data points from reference, extrapolated to energy range of the library and collapsed to the library group structure using a weighting spectrum	this keyword is not allowed since there are no pre-computed dose responses contained within the CE libraries
doseData=*MT*	data points from reference, extrapolated over the range of 1.0d-11 to 20.0 MeV and collapsed to the library group structure without a weighting spectrum	data points from reference, extrapolated over the range of 1.0d-11 to 20.0 MeV
doseData=*MT* noExtrapolation	data points from reference and collapsed to the library group structure without a weighting spectrum	data points from reference using only the original energy range

Reaction rate MT values

$$R = \int \Sigma(E)\phi(E)dE$$

MT	Description
1	Total cross section
18	Total fission cross section
27	Absorption cross section (MT=18 and 101)
101	Neutron disappearance
102	(n,γ) radiative capture cross section
103	(n,p) cross section
104	(n,^{2}H) cross section
105	(n,^{3}H) cross section
106	(n,^{3}He) cross section
107	(n,^{4}He) cross section
1452	Product of ν times the fission cross section

MT	Description
501	Total photon interaction cross section
502	Photon coherent scattering
504	Photon incoherent scattering
516	Pair production, nuclear and electron field
518	Photofission (γ,f)
522	Photoelectric

Grid geometry format

List (x,y,z) planes in each dimension separately:

1. By specifying *n* uniform cells from the min to max
2. By listing all the plane values

```
gridGeometry id
    title="Optional title in double quotes"
    xLinear nx xmin xmax
    yPlanes y0 y1 y2 … yn  end
    zLinear nz zmin zmax
    make3dMap
end gridGeometry
```

Use *Linear and *Planes multiple times; duplicates will be removed.

Grid geometry example

```
gridGeometry 7
    title="large meshes in paraffin, 5 cm mesh for shield thicknesses"
    xLinear 5 -25 25
    xLinear 12 30 90
    xplanes 100 110 120 -30 end

    yplanes -50 -40 40 50 end
    yLinear 7 -35 35

    zplanes -50 -40 40 50 end
    zLinear 7 -35 35
end gridGeometry

make3dMaps
```

Cylindrical geometry format

List (r,θ,z) values in each dimension separately:

1. By specifying n uniform steps from the min to max
2. By listing all the values

```
cylGeometry id
    title="Optional title in double quotes"
    radii r0 r1 r2 … rn end          radiusLinear nr rmin rmax
    thetas θ0 θ1 θ2 … θn  end        thetaLinear nθ θmin θmax
    degrees θ0 θ1 θ2 … θn  end       degreeLinear nθ θmin θmax
    zPlanes z0 z1 z2 … zn  end       zLinear nz zmin zmax
    make3dMap
    zaxis u v w   xaxis u v w
end cylGeometry
```

Use *Linear and *Planes multiple times; duplicates will be removed.

For angle, theta is 0–2π; degrees are 0–360°.

Cylindrical geometry example

```
cylGeometry 12
    radiusLinear 20 100.0 168.0
    radiusLinear 10 168.0 368.0
    thetaLinear 12 0 6.283
    zLinear 25 255.2 -255.2
    zPlanes  -45.0 -40. -35.0 end
end cylGeometry

cylGeometry 14
    title="degenerate: emulate surface tally over partial angle range"
    radiusLinear 1 367.5 368.5
    degreeLinear 1 45 135
    zLinear 25 255.2 -255.2
    zaxis 0 0 1
    xaxis 0 -1 0
 end cylGeometry

 make3dMaps
```

For Tallies only –
not importance maps

Distribution formats

```
distribution id
    title="a binned histogram"
    abscissa x1 x2 x3 … xn xn+1 end
    -or-
    energyBoundsID=id
    truePDF    f1 f2 f3 … fn    end
end distribution
```

Or

```
distribution id
    title="value/function pairs"
    abscissa x1 x2 x3 … xn end
    -or-
    energyBoundsID=id
    truePDF  f1 f2 f3 … fn end
end distribution
```

Optional – with either type

```
distribution id
    biasedPDF  f1 f2 f3 … fn end
    importance  f1 f2 f3 … fn end
    runSampleTest
end distribution
```

Or

use keywords "trueCDF", "biasedCDF"

Additional keywords that can be used with binned histograms instead of the abscissa keyword:

```
neutronGroups
photonGroups
```

However, these keywords cannot be used when Denovo and Monaco use different group structures.

zDistributions to sample the axial dimension of a cylinder are relative to the bottom of the cylinder, i.e., bottom of cylinder at z = 0.

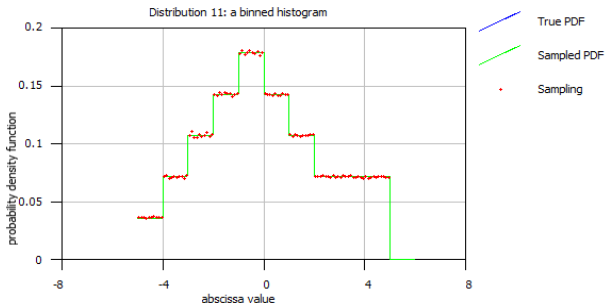
Distribution examples

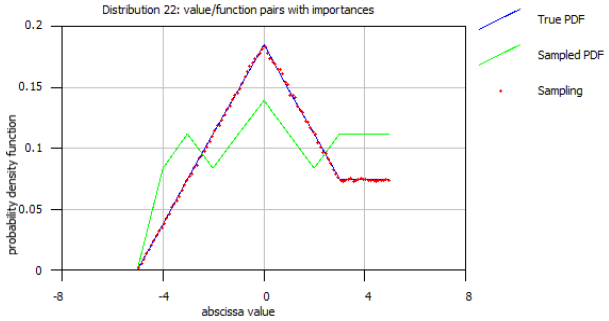
```
distribution 11
   title="a binned histogram"
   abscissa -5 -4 -3 -2 -1 0 1 2 3 4 5 end
   truePDF   1  2  3  4  5 4 3 2 2 2   end
end distribution

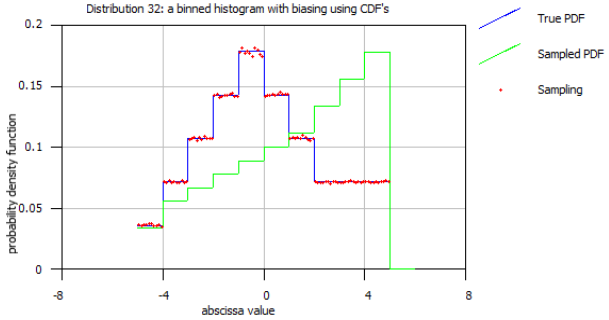

distribution 22
   title="value/function pairs with importances"
   abscissa   -5 -4 -3 -2 -1 0 1 2 3 4 5  end
   truePDF     0  1  2  3  4 5 4 3 2 2 2  end
   importance  4  3  2  1  1 1 1 1 2 2 2  end
end distribution


distribution 32
   title="a binned histogram with biasing using CDF's"
   abscissa -5 -4 -3 -2 -1  0  1  2  3  4 5 end
   trueCDF   1  3  6 10 15 19 22 24 26 28   end
   biasedPDF 3  5  6  7  8  9 10 12 14 16   end
end distribution

runSampleTests
```







Distribution formats

```
distribution id
    title="discrete probabilities"
    discrete E1 E2 E3 … En end
    truePDF  p1 p2 p3 … pn   end
end distribution
```

Example:

```
distribution 11
   title="Cobalt-60 discrete"
   discrete 1173228 1332492 end
   truePDF   0.9985 0.999826 end
end distribution
```

Optional

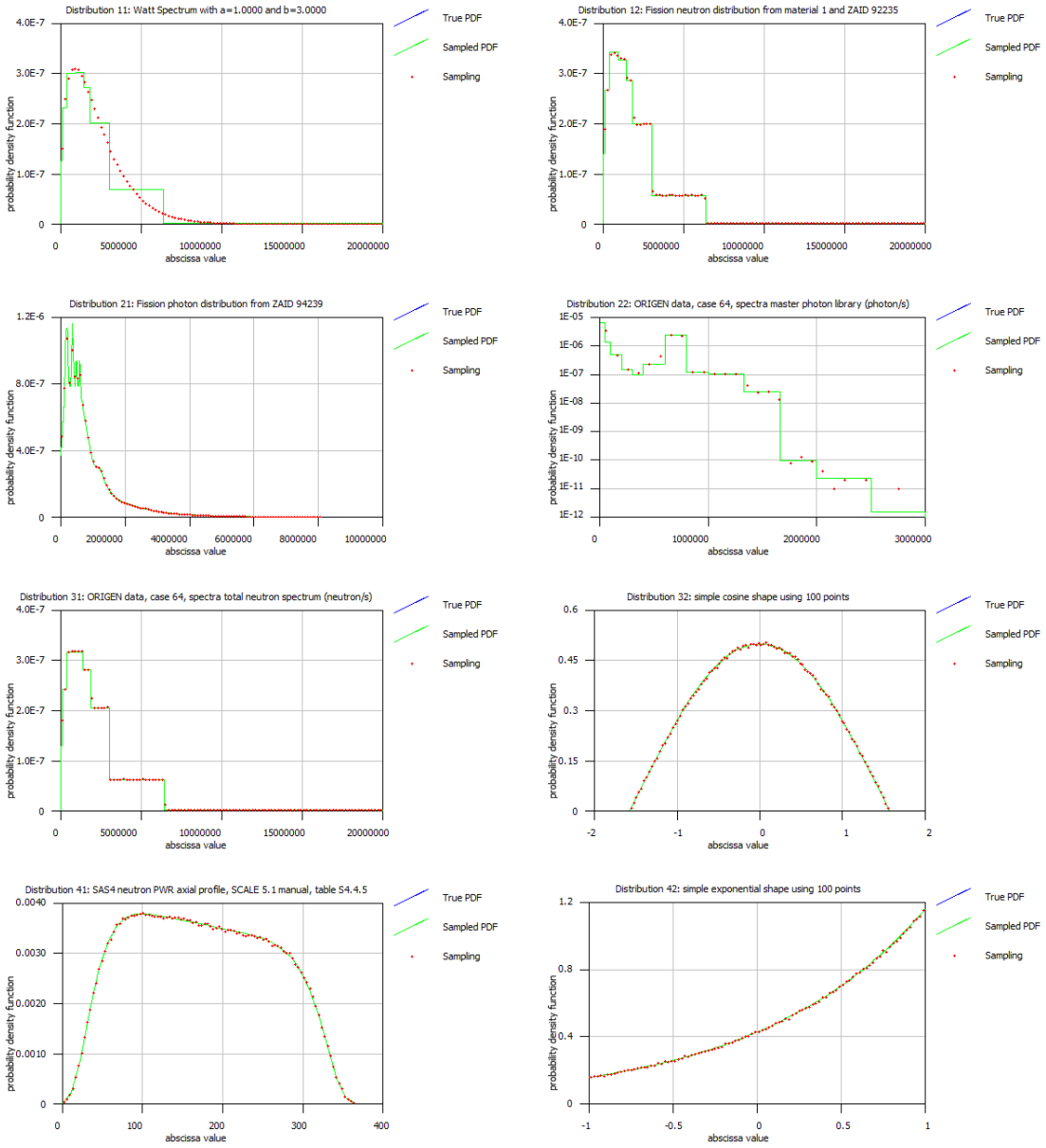
```
distribution id
    biasedPDF  p1 p2 p3 … pn end
    importance  f1 f2 f3 … fn end
    weight  w1 w2 w3 … wn end
    runSampleTest
end distribution
```

Or
use keywords “trueCDF”, “biasedCDF”

Built-in distributions

```
distribution id
    title="a built-in distribution"
    special="distributionName"
    parameters p1 p2 … pn  end
end distribution
```

Distribution	Parameters
"wattSpectrum"	*a b n*
"fissionNeutrons"	*m ZAID*
"fissionPhotons"	*ZAID*
"origensBinaryConcentrationFile"	*c s*
"cosine"	*n*
"pwrNeutronAxialProfile"	none
"pwrGammaAxialProfile"	none
"pwrNeutronAxialProfileReverse"	none
"pwrGammaAxialProfileReverse"	none
"exponential"	*a n*
"origensDiscreteGammas"	*z a m*



EnergyBounds/TimeBounds format

```
energyBounds id
    title="energy bin boundaries"
    bounds E1 E2 E3 … En end
    linear n Emin Emax
    logarithmic n Emin Emax
    tolerance=t
end energyBounds
```

Or

```
energyBounds id
    252n 238n 200n 56n 44n 27n
    47p 19p
end energyBounds
```

```
timeBounds id
    title="time bin boundaries"
    bounds t1 t2 t3 … tn end
    linear n tmin tmax
    logarithmic n tmin tmax
    tolerance=t
end timeBounds
```

Sources block

Each Monaco source uses separable distributions in space, energy, and direction.

```
read sources
    src id
        (source basics)
        (spatial distribution)
        (energy distribution)
        (directional distribution)
    end src
end sources
```



Multiple sources allowed

Source basics

Optional title and source strength in particles/sec.

```
read sources
    src id
        title="Optional title in double quotes"
        neutron/photon
        strength=S
        biasedStrength=B

          (spatial distribution)
          (energy distribution)
          (directional distribution)
    end src
end sources
```

Source spatial shape format

- Spatial – uniform density over a shape:

Keyword	Parameters	Possible degenerate cases
cuboid	x_{max} x_{min} y_{max} y_{min} z_{max} z_{min}	rectangular plane, line, point
xCylinder	r x_{max} x_{min}	circular plane, line, point
yCylinder	r y_{max} y_{min}	circular plane, line, point
zCylinder	r z_{max} z_{min}	circular plane, line, point
xShellCylinder	r_1 r_2 x_{max} x_{min}	cyl., planar annulus, cyl. surface, line, ring, point
yShellCylinder	r_1 r_2 y_{max} y_{min}	cyl., planar annulus, cyl. surface, line, ring, point
zShellCylinder	r_1 r_2 z_{max} z_{min}	cyl., planar annulus, cyl. surface, line, ring, point
sphere	r	point
shellSphere	r_1 r_2	sphere, spherical surface, point

  - Limited by unit, region, or material

    unit=u region=r or mixture=m
  - Optional translation: origin x=x y=y z=z
  - Optional rotation: axis u v w (cylinders)

    rotate $a_1=\theta_1$ $a_2=\theta_2$ $a_3=\theta_3$ (cylinders)

Source spatial distribution

- Cuboids

```
xDistributionID=id  xScaleDist
yDistributionID=id  yScaleDist
zDistributionID=id  zScaleDist
```

- Cylinders

```
rDistributionID=id  rScaleDist
zDistributionID=id  zScaleDist
```

- Spheres

```
rDistributionID=id  rScaleDist
```

zDistributions to sample the axial dimension of a cylinder are relative to the bottom of the cylinder, i.e., bottom of cylinder at z = 0

- or -

use *zScaleDist*

Source spatial distribution example

```
read sources
    src 1
        title="Uniform over cask cylinder but only in spent fuel (7)"
        strength=3.7e18
        neutrons
        zCylinder  95.0   228.6 -228.6
        zDistributionID=4 zScaleDist
        mixture=7
        (energy distribution)
        (directional distribution)
    end src
end sources
```

Source energy and direction

- Specify the energy distribution

```
eDistributionID=id
useNormConst  (do not use strengh= keyword)
```

- Specify the optional directional distribution (in μ)

```
direction u v w
dDistributionID=id
dScaleDist
```

Full source example

```
read definitions
    distribution 4 special="cosine" end distribution
    distribution 10
        special="wattSpectrum" parameters 1.0 3.0 end
    end distribution
    distribution 21
        abscissa -1.0   0.0   0.8    0.9    1.0 end
        truePDF      0.5   0.4   0.05   0.05    end
        biasedPDF    0.2   0.2   0.2    0.4     end
    end distribution
end definitions
read sources
    src 1
        title="Uniform over cask cylinder but only in spent fuel (7)"
        neutrons
        strength=3.7e18
        zCylinder  95.0   228.6 -228.6
        zDistributionID=4 zScaleDist
        mixture=7
        eDistributionID=10
        direction 0.0 0.0 1.0
        dDistributionID=21
   end src
end sources
```

MAVRIC input: Mesh file source

Use a Monaco Mesh Source (*.msm) file for the spatial and energy components of a source.

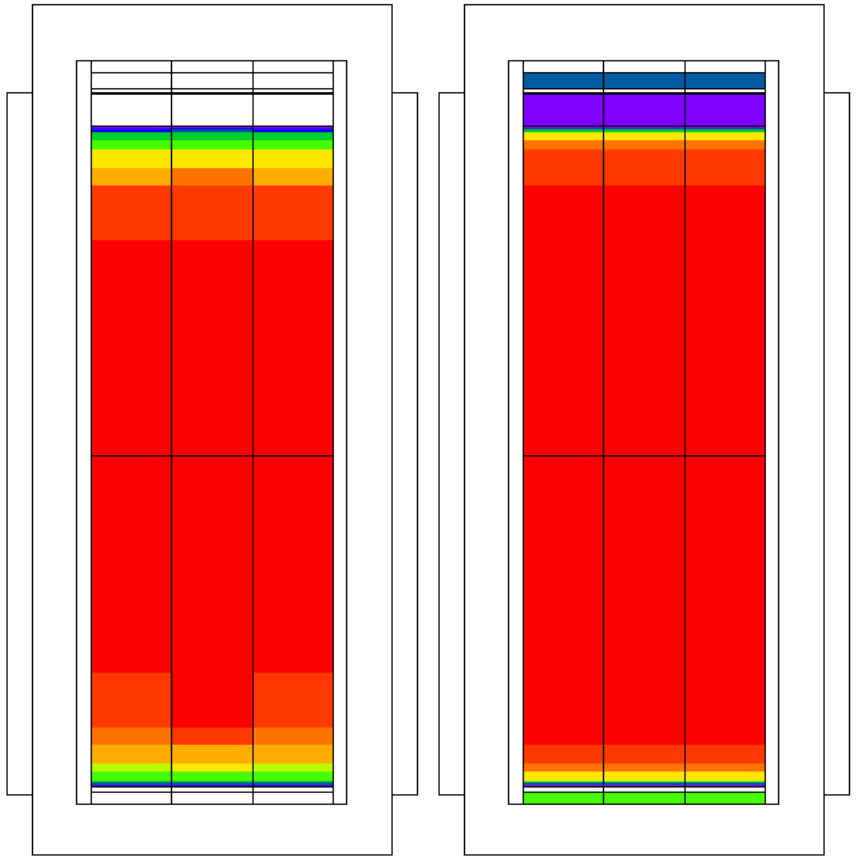
```
read sources
    src 1
        meshSourceFile=”c:\mydocu~1\previouslyMadeSource.msm”
        strength=S
        biasedStrength=B
        origin x=x y=y z=z
        direction u v w
        dDistributionID=id
        dScaleDist
        allowResampling
    end src
end sources
```

Mesh source saver

- Make a mesh-based version of the source
  - Viewable in Fulcrum

```
read sources
    src 1 … end src
    ...
    src 7 … end src
    meshSourceSaver
        gridGeometryID=1
        filename="meshed.msm"
        subCells=3
        sourceTrials=1000000
        reduce
        makeTotal

        nEnergyBoundsID=
        pEnergyBoundsID=
    end meshSourceSaver
end sources
```



Tallies in Monaco

- Point detectors
  - Estimates flux at a point
  - Can integrate flux with a response function(s)
  - Gives both uncollided and total response
  - Gives uncertainties
- Region tallies
  - Estimates flux over a region (or material)
  - Gives the track-length and collision density estimates
  - Can integrate flux with a response function(s)
- Mesh tallies
  - Gives the group fluxes and optional responses using a grid geometry

Tallies block

```
read tallies
    pointDetector  1
       ...
    end pointDetector

    regionTally 7
       ...
    end regionTally

    meshTally 1
       ...
    end meshTally

    regionTally 1
       ...
    end regionTally

    meshTally 2
       ...
    end meshTally
end tallies
```

Tally formats

```
pointDetector id
    title="optional title goes here"
    locationID=p
    minSampProb=m
end pointDetector

regionTally id
    title="optional title goes here"
    unit=u  region=r  -OR-  unit=u  mixture=m
end regionTally

meshTally id
    title="optional title goes here"
    gridGeometryID=g
    unit=u  region=r  -OR-  unit=u  mixture=m
    noGroupFluxes
    saveRespDetails
    weightless
end meshTally
```

Tally formats (cont.)

```
tallyType id
    title="optional title goes here"

    neutron/photon

    responseID=R
    responseIDs R1 R2 … Rn end

    multiplier=m

    energyBoundsID=i
    timeBoundsID=j
end tallyType

multiplier=m
```

Tally examples

```
read tallies
    pointDetector 7
        neutron
        locationID=2
        responseID=9029
    end pointDetector

    regionTally 8
        title="most important region in problem"
        photon   unit=2  region=8
        responseIDs 24 30 end
    end regionTally

    meshTally 1
        title="Mesh Tally with two responses"
        photon  gridGeometryID=1
        responseIDs 24 30 end
    end meshTally

'   convert responses from rem to mrem (doesn't change flux)
    multiplier=1000.0
end tallies
```

Parameters block

- Random seed = 16 character hexidecimal
- Cross section library used by Monaco can be set by *library="name"*
- Total histories = *batches*perBatch*

```
read parameters
    randomSeed=00003ecd7b4e3e8b
    perBatch=10000
    batches=100
    maxMinutes=1440.0

    fissionMult=0,1
    secondaryMult=0,1,2

    voidAllRegions

    library="name"  Or  ceLibrary="name"
end parameters
```

For Multi-Group
(default is all particles in library)

```
noNeutrons  noPhotons
```

For Continuous-Energy
(default is none)

```
neutrons   photons
```

Standard biasing block format

- Weight windows by region and path length stretching – everything is optional

```
read biasing
    windowRatio=R
    lowerWeights  w1 w2 w3 ... wn end
    targetWeights w1 w2 w3 ... wn end
    upperWeights  w1 w2 w3 ... wn end

    pathStretch p1 p2 p3 ... pn  end
    direction="globalZp"
end biasing
```

- Number of entries should match one of the following:
  - Number of regions
  - Number of groups
  - Regions*groups
- Number of entries equal to the number of groups or regions*groups does not work when Denovo and Monaco use different group structures

Standard biasing block examples

Weight windows by region and path length stretching

```
read biasing
    windowRatio=10.0
    targetWeights 46r1.0 46r2.0 46r3.0 46r1.0 46r2.0 46r3.0
                  46r1.0 46r2.0 46r3.0 46r1.0 46r2.0 46r3.0
                  46r1.0 46r2.0 46r3.0 end

    pathStretch   46r0.5 46r0.6 46r0.75 end
    direction="globalZp"
end biasing
```

For CE

```
    nEnergyBoundsID=i
    pEnergyBoundsID=j
```

Or, using a Monaco Importance Map File

```
read biasing
    meshImpMapFile="c:\mydocu~1\previouslyMadeImpMap.mim"
end biasing
```

MAVRIC input: After all blocks

```
end data
end
```

(no more data blocks to read)

(ends the "=mavric")

Known issues with file names in Denovo

- In general, SCALE can handle spaces in file names
- However, Denovo *cannot* handle spaces in file names, so do not include spaces in your MAVRIC input file names

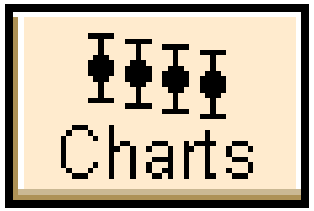
MAVRIC output

- Output from each module stitched together
  - *outputName*.out
- Diagnostic files – from definitions block
  - Grid geometry: *outputName*.grid*id*.3dmap
  - Cylindrical geom: *outputName*.cyl*id*.3dmap
  - Response: *outputName*.resp*id*.chart
  - Distribution: *outputName*.dist*id*.chart
- Plots – from plot block
  - *outputName*.plot*xxxx*.png



Fulcrum





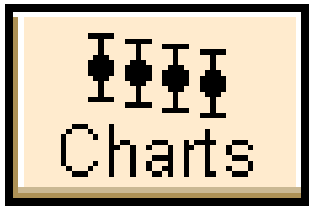
MAVRIC output

- Main output text file: *outputName*.out
  - Monaco part is at the end
  - Summary tables for region tallies and point detectors
- Separate files – group-wise results for flux and integrated results for each optional response
  - Mesh tallies: *outputName*.mt*id*.3dmap
  - Region tallies:
    - Table of results: *outputName*.rt*id*.txt
    - Convergence plot: *outputName*.rt*id*.chart
  - Point detector tallies:
    - Table of results: *outputName*.pd*id*.txt
    - Convergence plot: *outputName*.pd*id*.chart



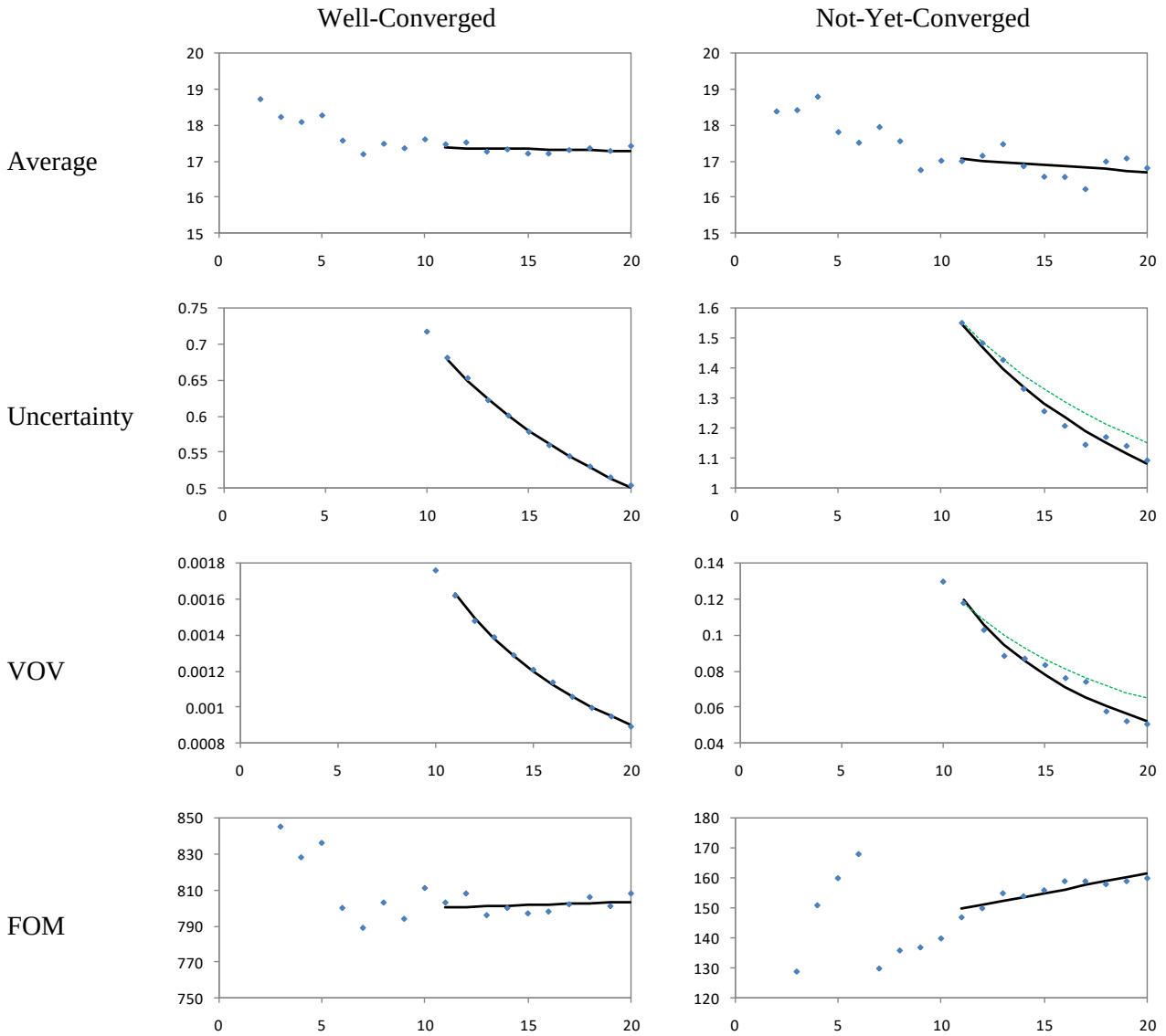
Fulcrum





MAVRIC output – Statistical checks

	Quantity/Test	Goal	Within
1.	mean relative slope of linear fit	= 0.00	0.10
2.	standard deviation exponent of power fit	= -0.50	$R^2 > 0.99$
3.	relative uncertainty final value	< 0.05	
4.	relative VOV exponent of power fit	= -1.00	$R^2 > 0.95$
5.	relative VOV final value	< 0.10	
6.	figure-of-merit relative slope of linear fit	= 0.00	0.10



MAVRIC output – Statistical checks

- Tally detail file (*outputName*.pd*id*.txt, *outputName*.rt*id*.txt)

Response 1 (using flux by track-length)

ANSI standard (1977) neutron flux-to-dose-rate factors (rem/h)/(neutrons/cm2/s)

Tally Values as the Simulation Progressed

batch	average value	standard deviation	relat uncert	rel VOV	FOM (/min)	stats check 1 2 3 4 5 6
1	7.55738E-05	7.07675E-08	0.00094	3.69E-06	1.11E+06	X X
2	7.55214E-05	5.00518E-08	0.00066	1.84E-06	1.10E+06	X X
3	7.55393E-05	4.08808E-08	0.00054	1.23E-06	1.10E+06	X X
4	7.55475E-05	3.54003E-08	0.00047	9.21E-07	1.10E+06	X X
5	7.55304E-05	3.16769E-08	0.00042	7.36E-07	1.09E+06	X X
6	7.55107E-05	2.89240E-08	0.00038	6.14E-07	1.09E+06	X X
7	7.55152E-05	2.67798E-08	0.00035	5.26E-07	1.09E+06	X X
8	7.55141E-05	2.50501E-08	0.00033	4.61E-07	1.09E+06	X X
9	7.55300E-05	2.36183E-08	0.00031	4.09E-07	1.09E+06	X X
10	7.55335E-05	2.24068E-08	0.00030	3.68E-07	1.09E+06	X X X X X X

Final Statistical Check (fits are over the last half of the simulation)

quantity	check		goal	actual	pass
1 mean	rel slope of linear fit	=	0.00	0.0008	yes
2 standard deviation	exponent of power fit	=	-0.50	-0.4998	yes
3 relative uncertainty	final value	<	0.05	0.0003	yes
4 relative VOV	exponent of power fit	=	-1.00	-0.9989	yes
5 relative VOV	final value	<	0.10	0.0000	yes
6 figure-of-merit (FOM)	rel slope of linear fit	=	0.00	-0.0029	yes

MAVRIC output – Statistical checks

- Tally summary (main text output file, *outputName*.out)

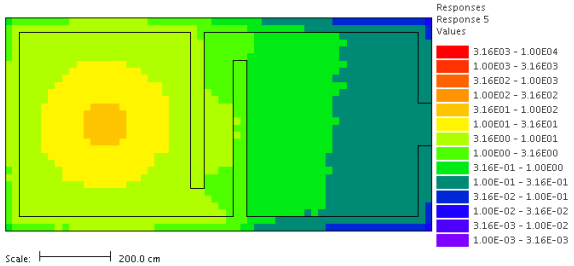
```
Region Tally 4.  example region tally

                      average      standard     relat      FOM    stat checks
    tally/quantity     value       deviation    uncert   (/min)   1 2 3 4 5 6
    ------------------  ----------  ----------  -------  --------  -----------
    neutron flux (tl)   9.75686E-01  5.47382E-05  0.00006  3.04E+07  X X X X X X
    photon flux (tl)    0.00000E+00
    neutron flux (cd)   0.00000E+00
    photon flux (cd)    0.00000E+00
    response 1          7.55335E-05  2.24068E-08  0.00030  1.09E+06  X X X X X X
    ------------------  ----------  ----------  -------  --------  ------------
```

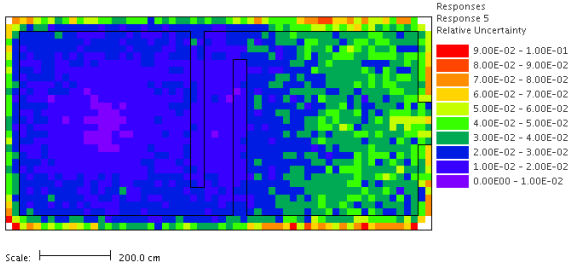
MAVRIC output – Mesh tally StatChecks

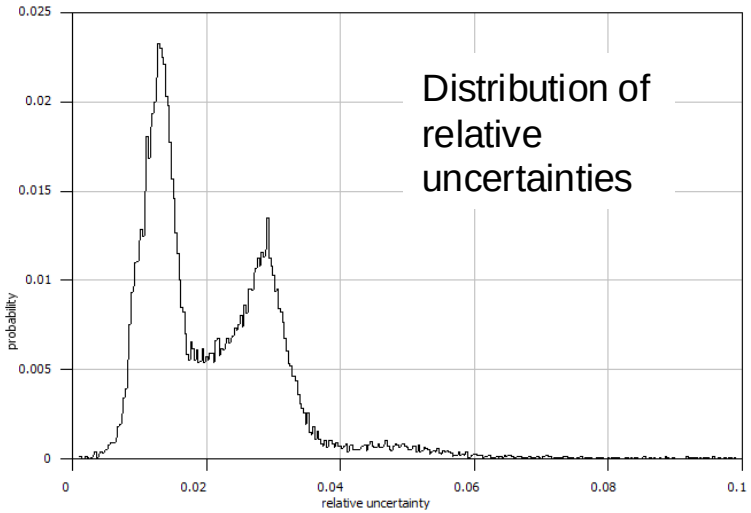
- Tough to make checks, since there are I×J×K tallies

values



rel. unc.





Distribution of relative uncertainties R_i and its mean, $\bar{r}$, of the voxels, V with score

$$\bar{r} = \frac{1}{V}\sum R_i \qquad FOM = \frac{1}{\bar{r}^2 T}$$

If each voxel $R_i \propto 1/\sqrt{N}$, then $\bar{r} \propto 1/\sqrt{N}$

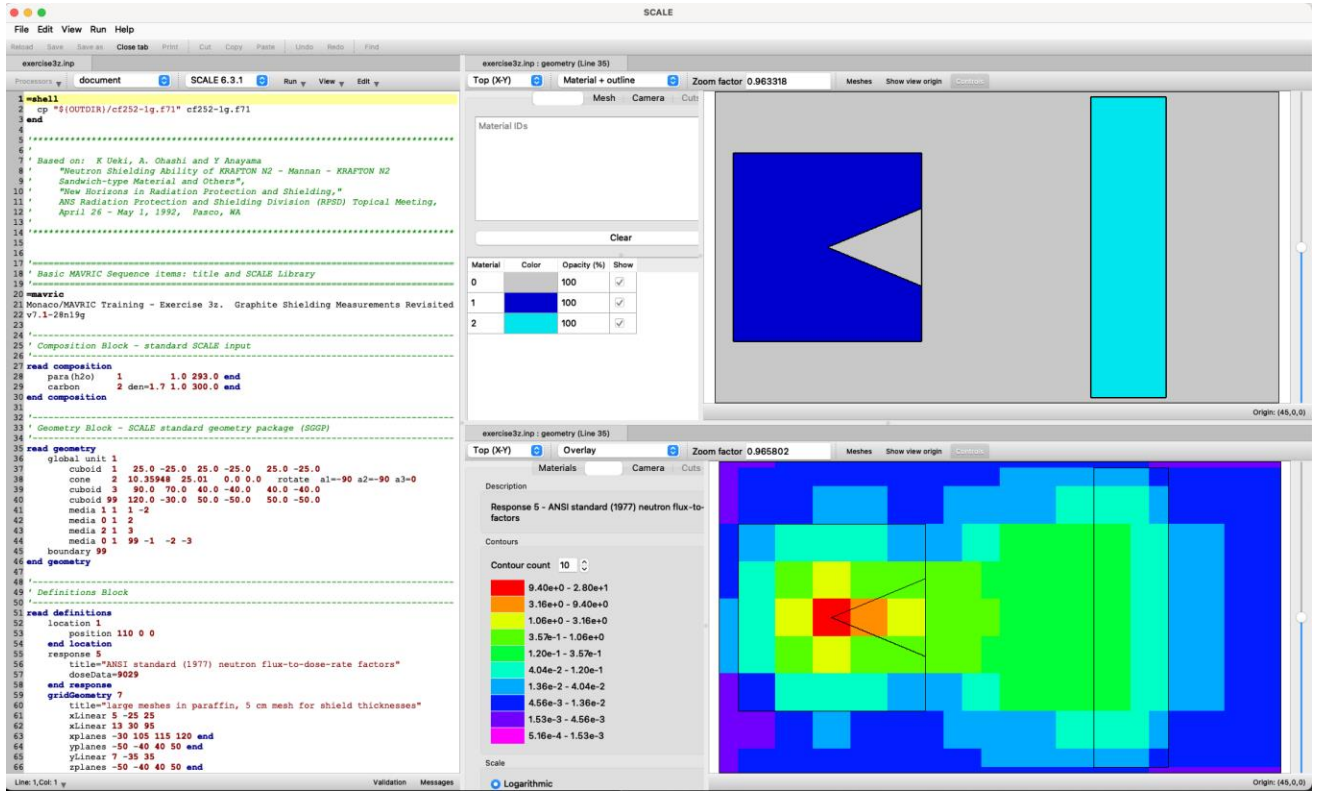
Four Tests:

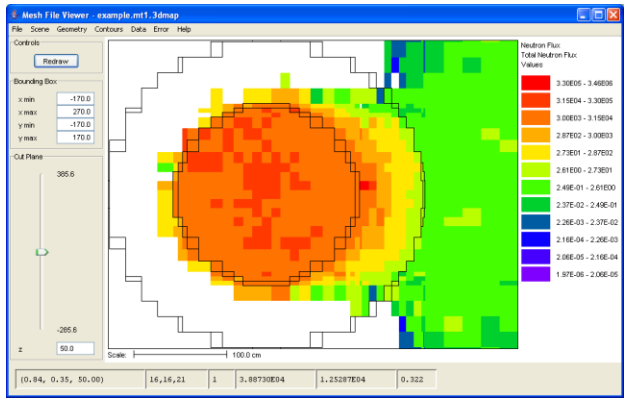
	Quantity	Test	Goal	Within
1.	ζ, fraction with score	relative slope of linear fit	= 0.00	±0.10
2.	$\bar{r}$, mean relative uncertainty	exponent of power fit	= -0.50	$R^2 > 0.99$
3.	variance of $\bar{r}$	exponent of power fit	= -1.00	$R^2 > 0.95$
4.	figure-of-merit	relative slope of linear fit	= 0.00	±0.10

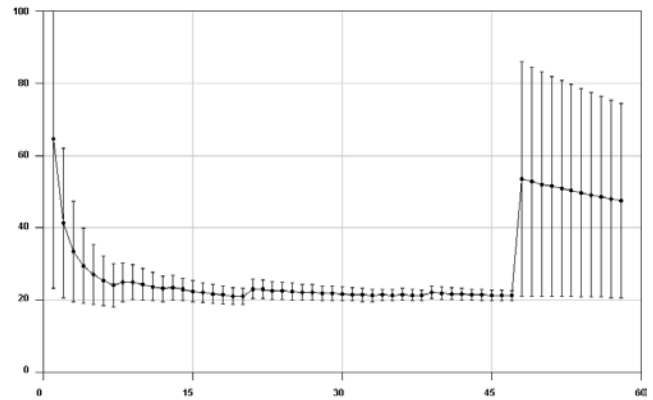
```
Neutron Mesh Tally 30.
                          zeta      mean      var of       FOM      passed
tally/quantity           value    rel unc    rel unc    (/min)    1 2 3 4
------------------      ------   --------   --------   --------   -------
flux, bin 0             0.9991   1.46E-01   1.44E-08   2.38E-01   X - X X
tot flux, bin 0         1.0000   2.14E-02   4.71E-09   1.11E+01   X X X X
resp 5/bin 0            1.0000   3.06E-02   1.07E-08   5.47E+00   X - X X
------------------      ------   --------   --------   --------   -------
```

MAVRIC output

- Fulcrum
  - Mesh tallies
  - Denovo fluxes
  - Importance maps
  - Mesh Sources
- Interactive plotter
  - Distributions
  - Responses
  - Convergence checks
  - Line plots from mesh







OAK RIDGE National Laboratory

In-class example: Iron sphere benchmark problem

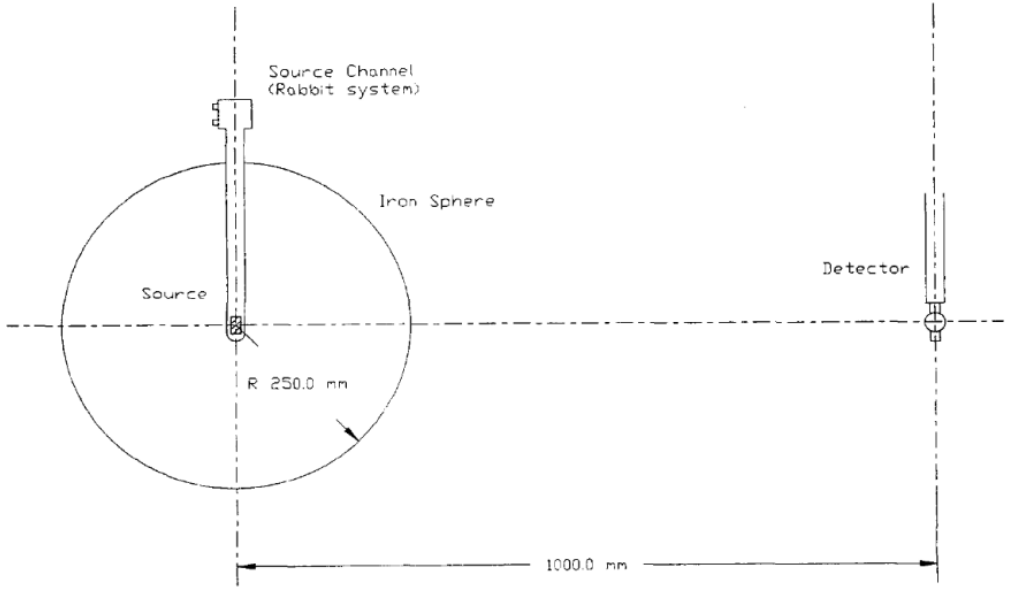


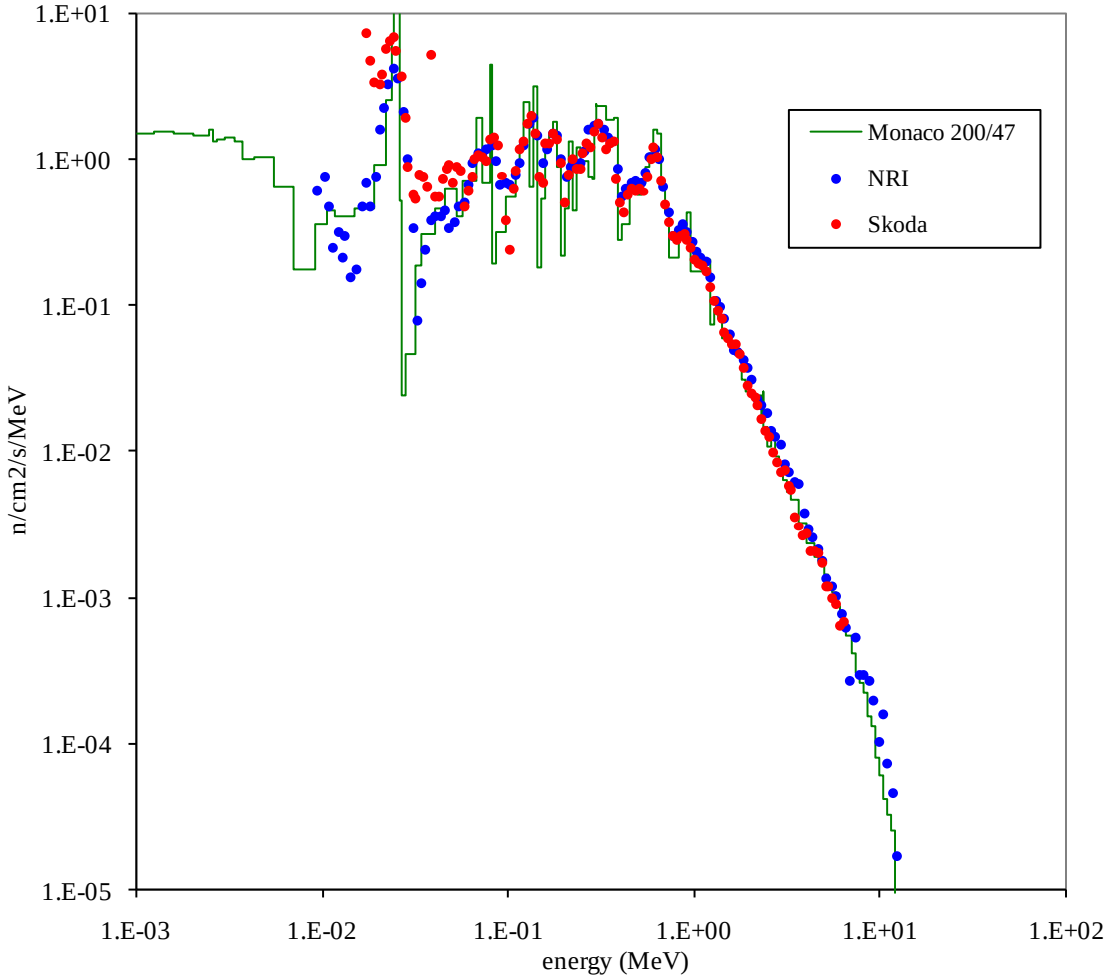
Learning goals

- The trainees and instructor will work an example problem together, which will allow attendees to apply what they have learned about running MAVRIC/Monaco using standard biasing techniques
- Also introduce the trainees to the SCALE GUI, Fulcrum, and identify how the different Fulcrum tabs are used to enter the information required to build a computational model of a shielding problem

Iron sphere benchmark problem

E. Sajo, M. L. Williams, and M. Asgari, "Comparison of Measured and Calculated Neutron Transmission Through Steel for a ^{252}Cf Source," *Annals of Nuclear Energy*, **20**(9), 585–604 (1993).





OAK RIDGE National Laboratory

CADIS: Consistent Adjoint Driven Importance Sampling

SCALE 6.3.2

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals

- Introduce the Consistent Adjoint Driven Importance Sampling (CADIS) variance reduction technique
  - Identify the input syntax and where additional input is required in the Monaco/MAVRIC input file to use the CADIS variance reduction technique
  - Identify the additional output files created when using the CADIS variance reduction technique
- Identify the MAVRIC utilities and how they can be used when performing shielding analysis
- In-class example: analyze dose rates near a cask

OAK RIDGE National Laboratory

Variance reduction with CADIS



Variance reduction

- Changing the sampling routines to optimize the simulation to get more particles to do *something*
- Traditional methods:
  - Rely on knowing the expected results
  - Require trial and error
  - Multiple methods may work against each other
  - May require several iterations
  - Typically involve many steps
  - May require specialized Monte Carlo (MC) codes

Variance reduction

- What is desired:
  - Tell MC code what tally/tallies to calculate
  - The MC code figures out the best way to do that

- Importance map (weight windows)
  - If we know how "important" a given particle is as a function of $(\vec{r}, E, \vec{\Omega})$, then weight window target values can be assigned, using roulette and splitting to control particle weight
  - Importance is the solution to the adjoint equation, using the tally location as the spatial component of the adjoint source and the tally response function as the energy component of the adjoint source

A bit of review

- Forward transport equation

$$H\Phi(\vec{r}, E, \hat{\Omega}) = Q(\vec{r}, E, \hat{\Omega})$$

$$H = \hat{\Omega} \cdot \vec{\nabla} + \Sigma(\vec{r}, E) - \int dE' \int d\hat{\Omega}' \, \Sigma_s(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega})$$

- Solve for flux, then compute detector response

$$R = \int_{V_d} \int_E \sigma_d(\vec{r}, E) \, \phi(\vec{r}, E) \, dE \, dV$$

- Adjoint transport equation

$$H^+\Phi^+(\vec{r}, E, \hat{\Omega}) = Q^+(\vec{r}, E, \hat{\Omega})$$

$$H^+ = -\hat{\Omega} \cdot \vec{\nabla} + \Sigma(\vec{r}, E) - \int dE' \int d\hat{\Omega}' \, \Sigma_s(\vec{r}, E \to E', \hat{\Omega} \to \hat{\Omega}')$$

- Solve for adjoint flux using $q^+(\vec{r}, E) = \sigma_d(\vec{r}, E)$, then compute

$$R = \int_{V_s} \int_E q(\vec{r}, E) \, \phi^+(\vec{r}, E) \, dE \, dV$$

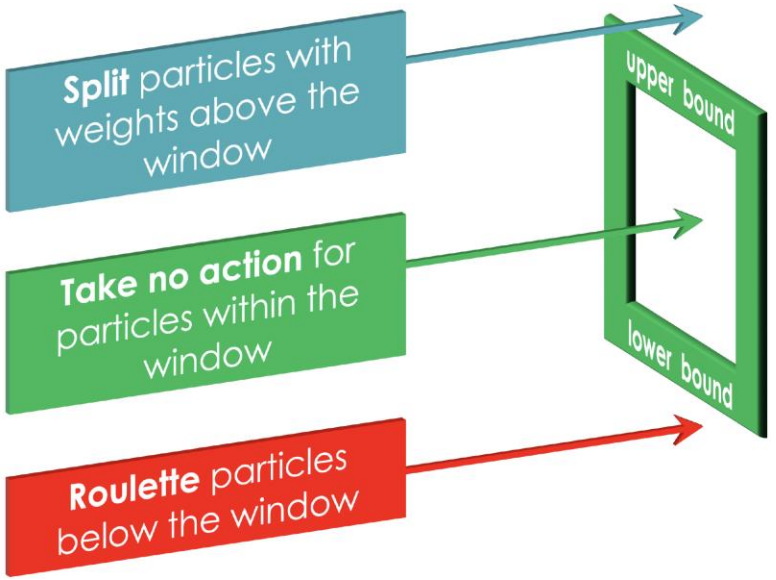
Where;

H : transport operator
Φ : scalar flux
$(\vec{r}, E, \hat{\Omega})$: *spatial, energy, and angular components*
$\sigma_d(\vec{r}, E)$: detector response
$q(\vec{r}, E)$: forward source
$q^+(\vec{r}, E)$: adjoint source
$\phi(\vec{r}, E)$: forward flux
$\phi^+(\vec{r}, E)$: adjoint flux
V_d : detector volume
V_s : source volume

Using importances

- If the true importances are known, then there is no need for the MC calculation
- If the approximate importances are known, then weight windows can be used for a MC game:

$$\bar{w}(\vec{r}, E) = \frac{c}{\phi^{+}(\vec{r}, E)}$$



CADIS methodology

Consistent Adjoint Driven Importance Sampling

- Biased source and importance map work together

Ali Haghighat and John C. Wagner, "Monte Carlo Variance Reduction with Deterministic Importance Functions," Progress in Nuclear Energy, 42(1), 25–53, (2003).

- Solve the adjoint problem using the detector response function as the adjoint source

$$q^{+}(\vec{r},E) = \sigma_d(\vec{r},E)$$

- Weight window targets are inversely proportional to the adjoint flux (i.e., measure of importance of the particles to the response)

$$\overline{w}(\vec{r},E) = \frac{c}{\phi^{+}(\vec{r},E)}$$

CADIS methodology (cont.)

- We want source particles born with a weight matching the weight window targets

$$w_0(\vec{r}, E) = \frac{q(\vec{r}, E)}{\hat{q}(\vec{r}, E)} = \overline{w}(\vec{r}, E)$$

- So, the biased source needs to be

$$\hat{q}(\vec{r}, E) = \frac{q(\vec{r}, E)}{\overline{w}(\vec{r}, E)} = \frac{1}{c} q(\vec{r}, E)\phi^{+}(\vec{r}, E)$$

- Since the biased source is a probability density function (pdf), solve for c

$$c = \int_{V_s} \int_{E} q(\vec{r}, E)\, \phi^{+}(\vec{r}, E)\, dE\, dV$$

Which is the estimate of response: R

CADIS methodology – Summary

- Define the adjoint source

$$q^{+}(\vec{r},E) = \sigma_d(\vec{r},E)$$

- Solve for the adjoint flux $\phi^{+}(\vec{r},E)$
- Estimate the response R

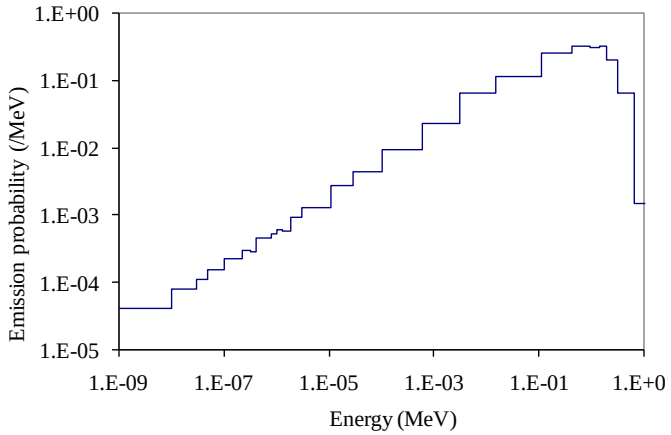
$$R = \int_{V_s}\int_{E} q(\vec{r},E)\,\phi^{+}(\vec{r},E)\,dE\,dV$$

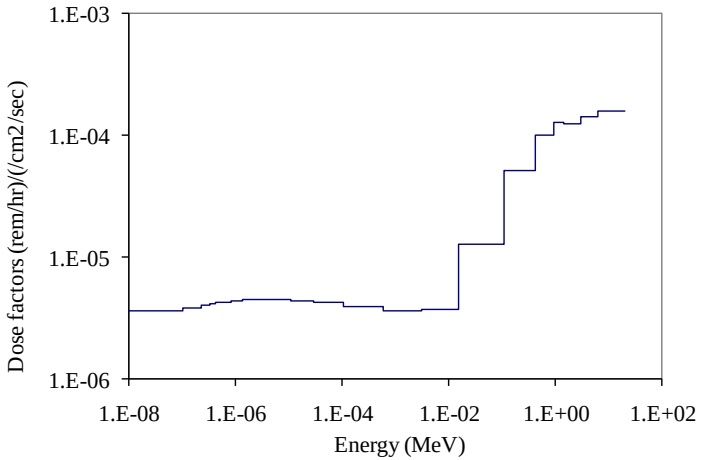
- Construct weight windows and biased source

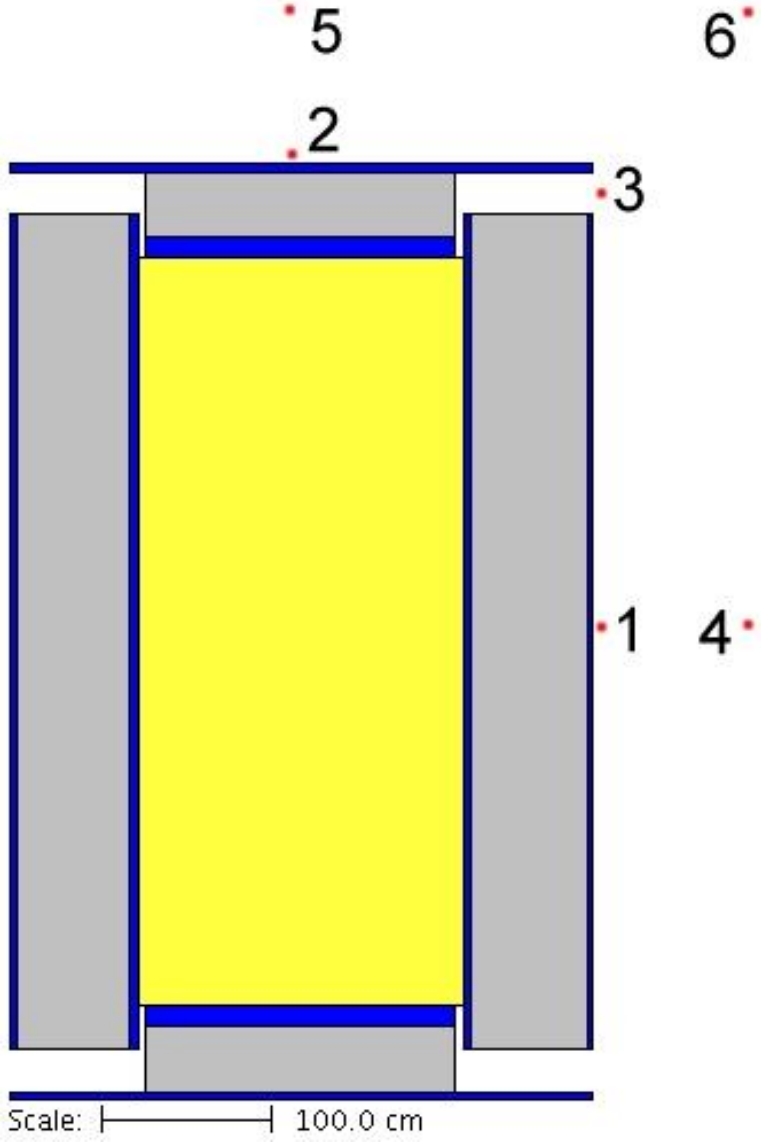
$$\overline{w}(\vec{r},E) = \frac{R}{\phi^{+}(\vec{r},E)} \qquad \hat{q}(\vec{r},E) = \frac{1}{R}\,q(\vec{r},E)\phi^{+}(\vec{r},E)$$

Example application – spent fuel cask

- Calculate dose rate at six different detector locations

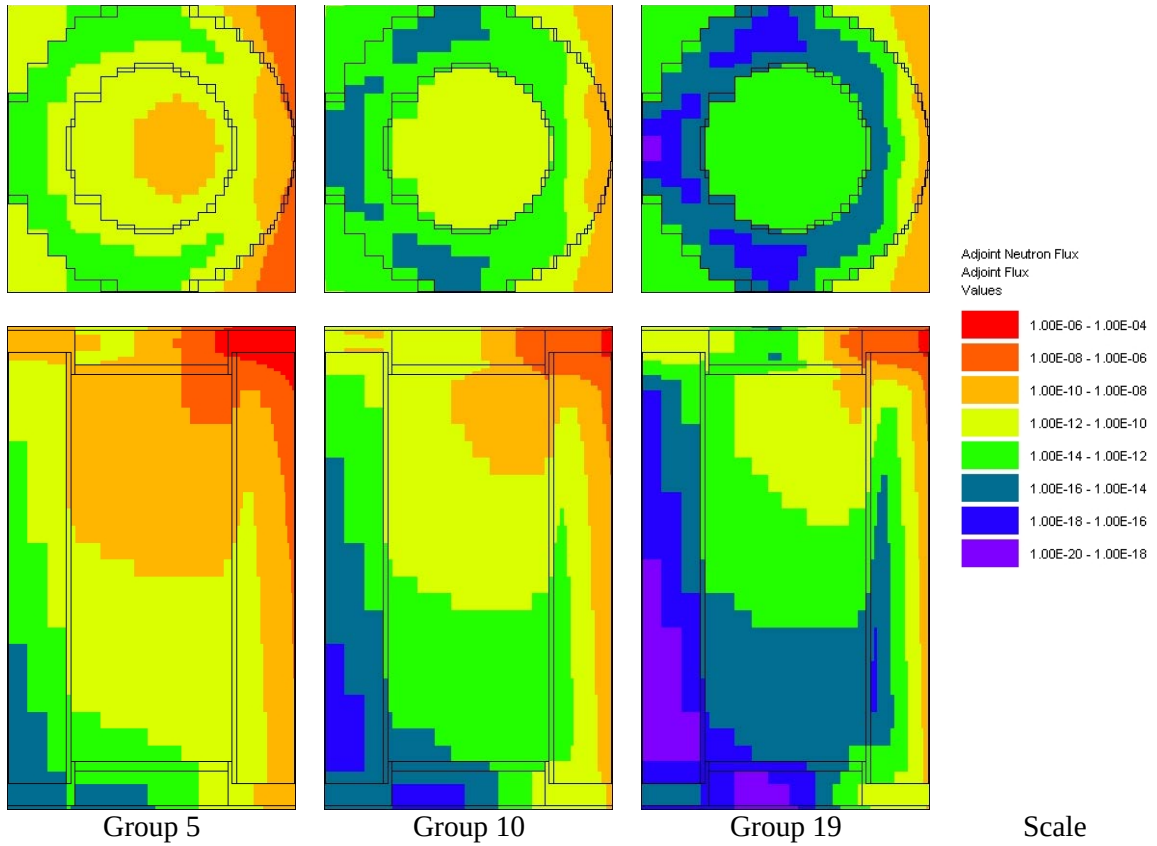






Example application – spent fuel cask (cont.)

- For detector 3: compute adjoint fluxes

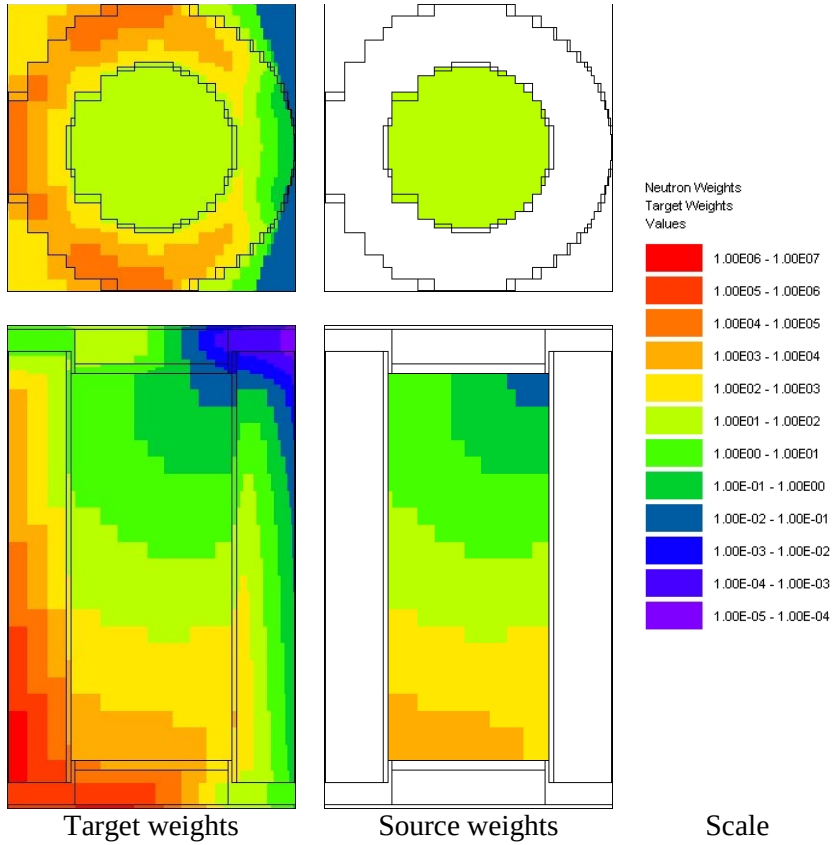


Horizontal cross-sectional views at the cask midplane showing adjoint neutron flux solution

Adjoint neutron fluxes (n/cm^2/s) calculated by Denovo for groups 5 (0.9–1.4 MeV), 10 (0.58–3.0 keV), and 19 (0.8–1 eV)

Example application – spent fuel cask (cont.)

- Compute target weights, biased source weights



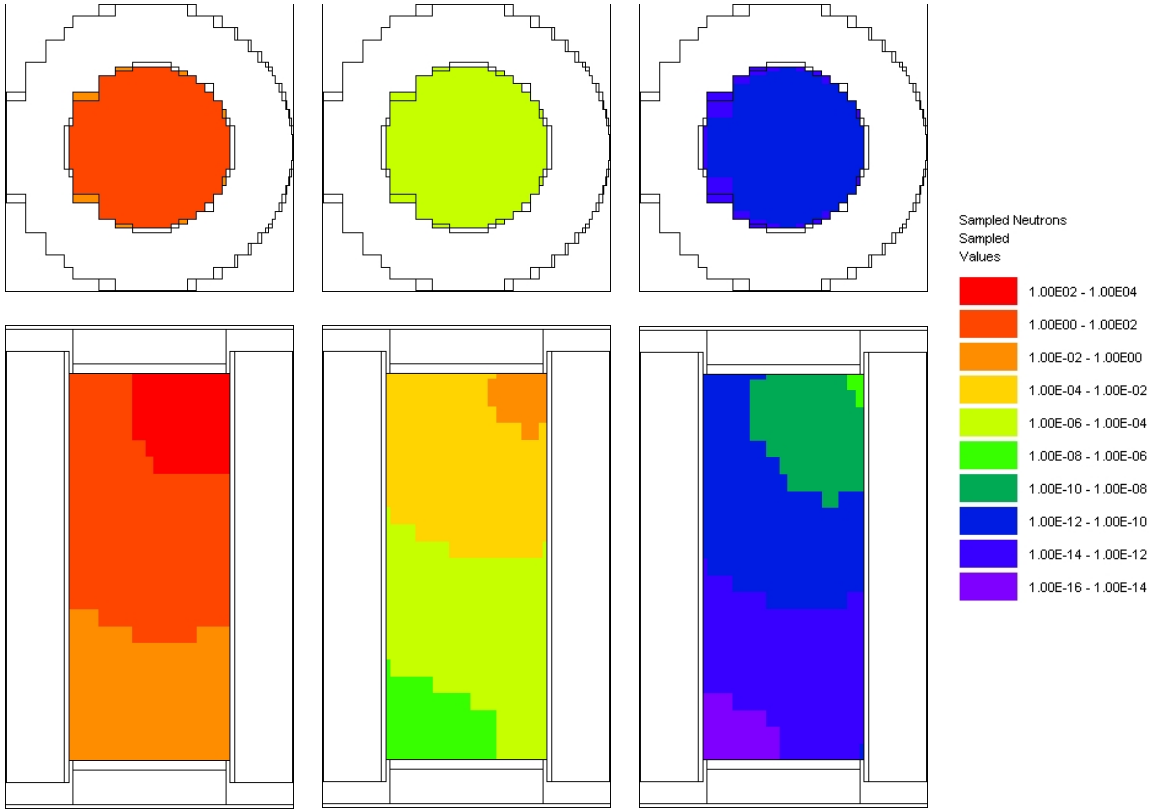
Target weights Source weights Scale

Horizontal cross-sectional views at the cask midplane showing neutron weights

Neutron target weights from the importance map and source weights (at birth) for neutron group 5 (0.9 to 1.4 MeV)

Example application – spent fuel cask (cont.)

- Biased source



Horizontal cross-sectional views at the cask midplane showing biased source

Biased source sampling probability (neutrons/cm^3) for neutron groups 5 (0.9–1.4 MeV), 10 (0.58–3.0 keV), and 19 (0.8–1 eV)

Example application – spent fuel cask (cont.)

- Computational speed up is significant (Six separate calculations)

Results

detector	Analog Monaco 5164 min	MAVRIC 543 min
1	5.54E-04 ± 24.0%	7.07E-04 ± 0.7%
2	6.97E-03 ± 5.9%	7.77E-03 ± 0.5%
3	1.55E-02 ± 2.0%	1.54E-02 ± 0.4%
4	4.57E-04 ± 4.7%	4.27E-04 ± 0.7%
5	1.36E-02 ± 0.9%	1.35E-02 ± 0.2%
6	2.91E-03 ± 1.2%	2.91E-03 ± 0.2%

$$FOM = 1/T\sigma^2$$

Speed up

detector	Monaco	MAVRIC
1	1	11179
2	1	1324
3	1	238
4	1	429
5	1	193
6	1	342

CADIS summary

- Provides consistent relationships for calculating source and transport biasing parameters
- Eliminates the incompatibility between source and transport biasing that has been problematic in other approaches
- Large speedup for source/detector problems
- Described in more detail in:
  - *John Wagner, Alireza Haghighat, "Automated Variance Reduction of Monte Carlo Shielding Calculations Using the Discrete Ordinates Adjoint Function," Nuclear Science & Engineering, vol.* 128, p.p. 186–208, 1998
  - *John Wagner, Alireza Haghighat, "Monte Carlo Variance Reduction with Deterministic Importance Functions," Progress of Nuclear Energy, vol.* 42(1), 2003

OAK RIDGE National Laboratory

MAVRIC input using importance maps



MAVRIC input: block format

```
=mavric    parm=keyword
Some title for this problem
v7.1-28n19g

read compositon
   ...
end composition
read celldata
   ...
end celldata

read geometry
   ...
end geometry
read array
   ...
end array
read volume
   ...
end volume
read plot
   ...
end plot

read definitions
   ...
end definitions
read sources
   ...
end sources
read tallies
   ...
end tallies

read parameters
   ...
end parameters
read biasing
   ...
end biasing
read importanceMap
   ...
end importanceMap

end data
end
```

Cross section libraries used with Denovo and Monaco

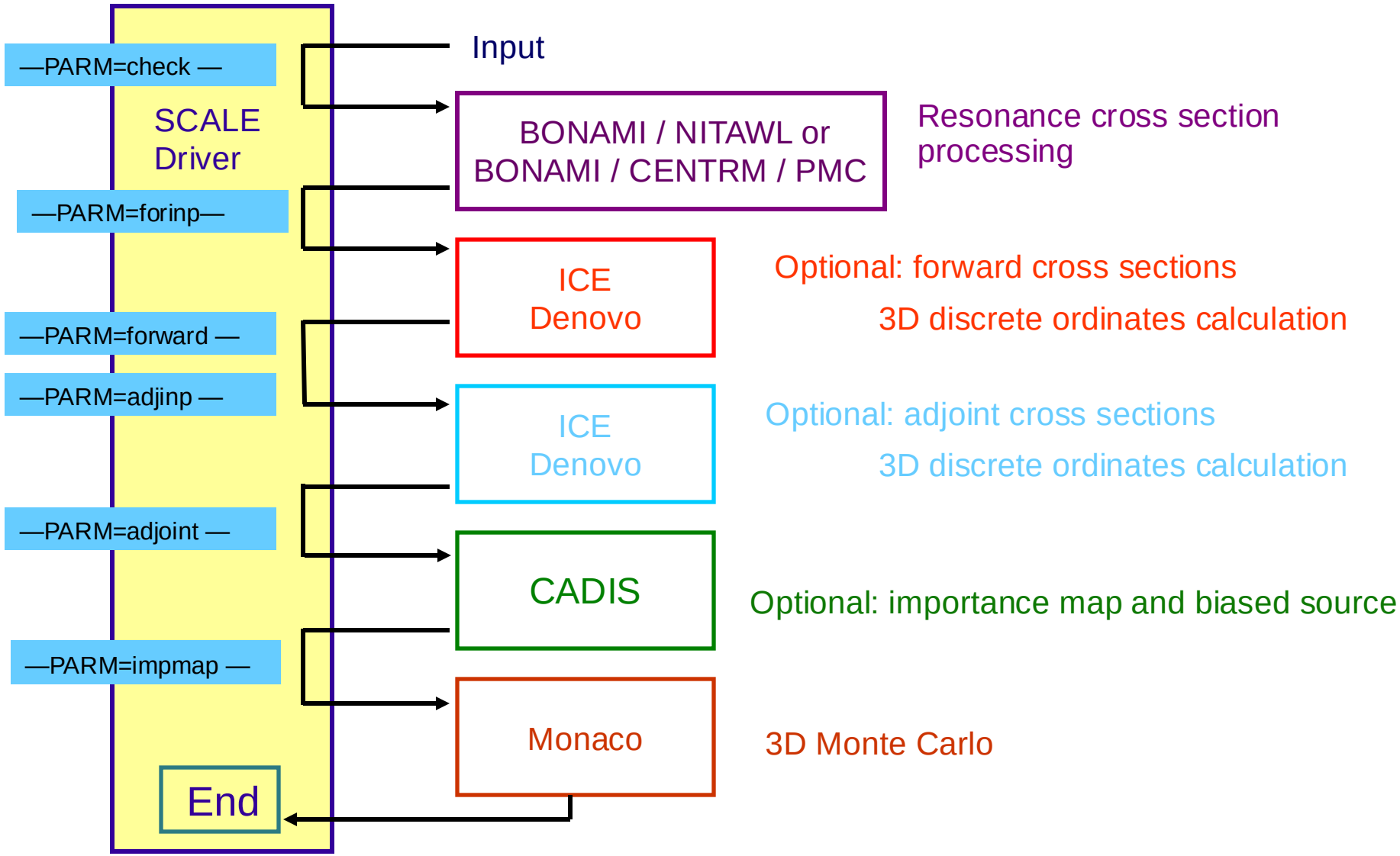
- It is fastest to use different cross section libraries for the Denovo importance map (adjoint) calculation and the Monaco Monte Carlo calculation
  - Use a coarse group library for Denovo, *v7.1-28n19g*
  - Use a fine group library for Monaco, *v7.1-200n47g*
  - Use a continuous-energy library for Monaco, *ce_v7.1_endf.xml*

```
=mavric    parm=keyword
Some title for this problem
v7.1-28n19g                          ←Denovo library

...


read parameters
   ...
   library="v7.1-200n47g"            ←Monaco MG library
           OR
   ceLibrary="ce_v7.1_endf.xml"      ←Monaco CE library
   ...
end parameters
```

MAVRIC sequence workflow



PARM=*keyword*

- PARM=CHECK
  - Checks over input, looks for simple mistakes
- PARM=FORINP
  - Produces forward Denovo input: i_xkba
- PARM=FORWARD
  - Runs forward Denovo, produces forward flux file: *outputName*.forward.dff
- PARM=ADJINP
  - Produces adjoint Denovo input: i_xkba
- PARM=ADJOINT
  - Runs adjoint Denovo, produces adjoint flux file: *outputName*.adjoint.dff
- PARM=IMPMAP
  - Creates biased source and importance map, produces
    - *.mim – Monaco importance map
    - *.msm – Monaco source mesh

MAVRIC simulations using CADIS

- User must provide:
  - All the information required for an analog calculation
    - Materials
    - Geometry
    - Source
    - Tallies
    - Parameters
  - Plus,
    - a grid geometry for the discrete ordinates
    - a description of the adjoint source (based on the tally you wish to optimize)
- Everything else is automatic!

Files generated with CADIS

- Mesh importance map
  - *outputName*.mim
- Mesh source map (biased source)
  - *outputName*.msm
- Or
  - *outputName*.*sid*.msm for each source
- For multiple sources, each source is biased, and the biased strength of each source is computed

MAVRIC utilities

mt2ascii	Convert a mesh tally into an ASCII text file.
mt2msl	Convert a mesh tally into a mesh source lite.
mt2msm	Convert a mesh tally into a mesh source.
mt2silo	Convert a mesh tally file into a Silo file for VisIt.
mt2vtk	Convert one dataset of one family in a mesh tally to VTK format.
mtAdder	Add several Monaco mesh tally files together into one mesh tally.
mtAverager	Average several Monaco mesh tally files into one mesh tally.
mtBinOp	Binary operation of mesh tally files: sum, difference, product and ratio.
mtDisp	Display the basics of a mesh tally file.
mtExpand	Expand a space-only mesh from a mesh tally with an energy function.
mtFilter	Perform various filters on a mesh tally file.
mtInv	Invert all of the values in a mesh tally.
mtMask	Keep only or remove specified voxels of a mesh tally based on geometry.
mtMinMax	Find the location/value of the min or max of each realmesh in a mesh tally.
mtMultiply	Multiply a mesh tally by a constant factor.
mtPull	Pull values from certain voxels out of a mesh tally file.
mtRefine	Subdivide the mesh into smaller meshes for better visualization.
mtResp	Apply a response function to one family of a mesh tally file.
mtSplit	Split off part of a mesh tally file into a separate mesh tally file.

MAVRIC utilities (cont.)

mtBinOp - Binary operation of mesh tally files: sum, difference, product and ratio.

Intended use: Apply simple math to the results stored in mesh tally files.

Input: The first mesh tally, the operator: add (or sum, +), subtract (or difference, -), multiply (or product, x, *) and divide (or ratio, ÷," /"), the second mesh tally name and then name of the resulting mesh tally file.

Output: A new mesh tally file.

Notes: Uncertainties are propagated assuming the two mesh tallies are uncorrelated, which may not always be a good assumption. Mesh tallies need to have the same grid structure and number of families and groups. Dataset names in the results are inherited from the first mesh tally and may not make sense after the operation. When using the / (slash) for division, enclose it in quotes ("/").

Example:

```
=mtBinOp
"neutron.3dmap"     ! first operand
divide              ! operation
"total.3dmap"       ! second operand
"ratio.3dmap"       ! output file name
end
```

MAVRIC utilities (cont.)

dff2dso	Convert a Denovo flux file into a Denovo spatial output file.
dff2mai	Convert a Denovo flux file into a mesh angular information file.
dff2mim	Invert a Denovo flux file and store as a mesh importance map.
dff2msl	Convert a Denovo flux file into a mesh source lite.
dffBinOp	Binary operation of Denovo flux files: sum, difference, product and ratio.
dffDisp	Display the basics of a Denovo flux file.
dffExpand	Expand a space-only Denovo flux file by an energy function.
dffFilter	Perform various filters on a Denovo flux file.
dffFix	Fix the zero and negative values in a Denovo flux file.
dffInt	Integrate a single particle type from a Denovo flux file.
dffInv	Invert the values in a Denovo flux file.
dffMult	Multiply a Denovo flux file by a constant factor.
dffPull	Pull fluxes from certain voxels out of a Denovo flux file.
dffResp	Apply a response function to scalar fluxes in a Denovo flux file.
dffSplit	Split off a single particle type from a Denovo flux file.

MAVRIC utilities (cont.)

dsi2asc	Convert a Denovo simple input (*.dsi) from binary to ascii.
dsiDisp	Display the basics of a Denovo simple input file.
dso2msl	Use a Denovo spatial output to create a mesh source lite.
dsoDisp	Display the basics of a Denovo spatial output file.
mim2wwinp	Convert a mesh importance map into an MCNP weight window file.
mimDisp	Display the basics of a mesh importance map (*.mim) file.
mimNorm	Normalize a mesh importance map to a given location/energy.
msmDisp	Display the basics of a mesh source map (*.msm) file.

MAVRIC utilities (cont.)

```
=mtAverager
3
‘c:\documents\meshFilename_1.3dmap’   800000
‘c:\documents\meshFilename_2.3dmap’   900000
‘c:\documents\meshFilename_3.3dmap’   800000
‘c:\documents\meshFilename.ave.3dmap’
end


=mtSplit
‘c:\documents\meshFilename.mt1.3dmap'    ! the mesh tally
2                                        ! second family - responses
1                                        ! first response – photon dose rate
'photonDoseRate.3dmap'                   ! output file name
end

=mtMultiply
'photonDoseRate.3dmap'                   ! input file name
1000.0                                   ! convert rem/hr to mrem/hr
'c:\documents\mremPhoton.3dmap'          ! output file name
end
```

OAK RIDGE National Laboratory

In-class example: dose rates near a storage cask

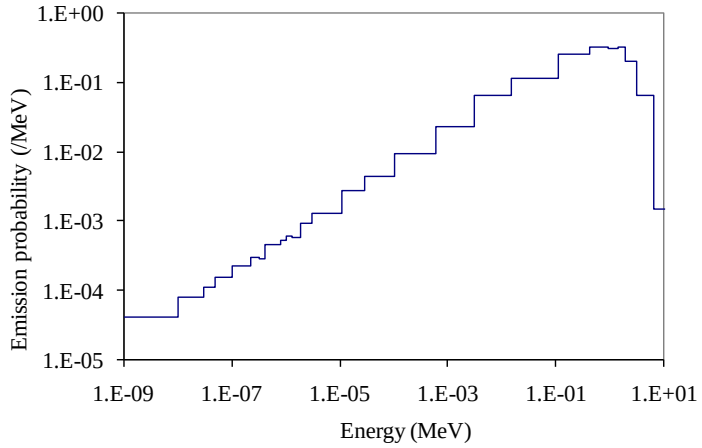


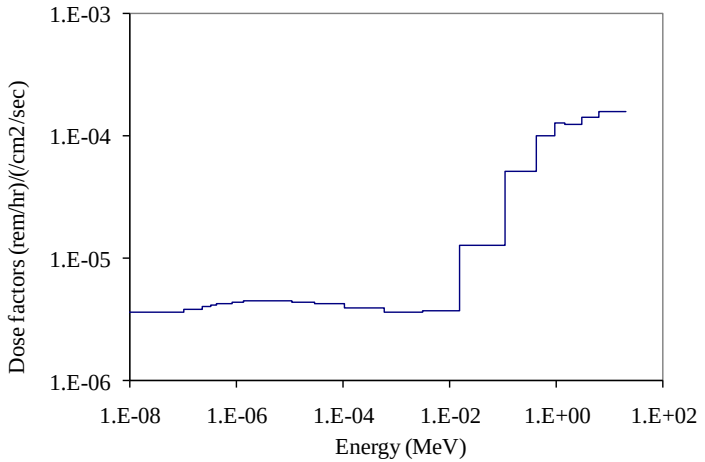
Learning goals

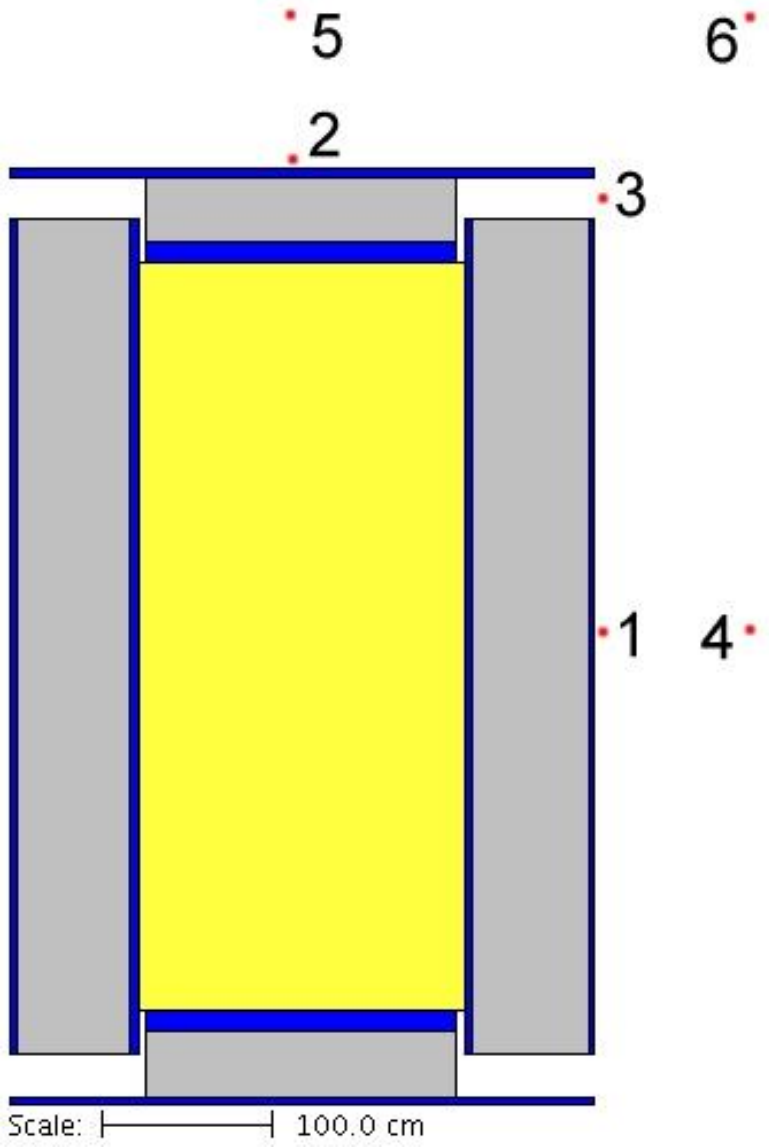
- The students and instructor will work through an example problem together, which will allow them to apply what they have learned about running Monaco/MAVRIC using the CADIS variance reduction technique

Spent Fuel Shipping Cask Example in the MAVRIC manual

Calculate dose rate at six different detector locations





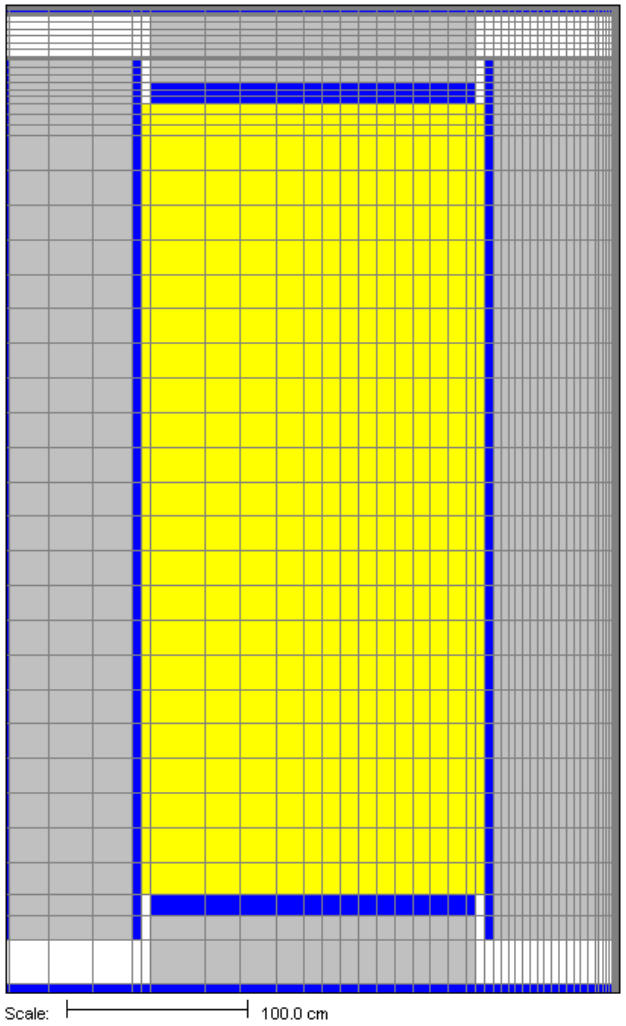


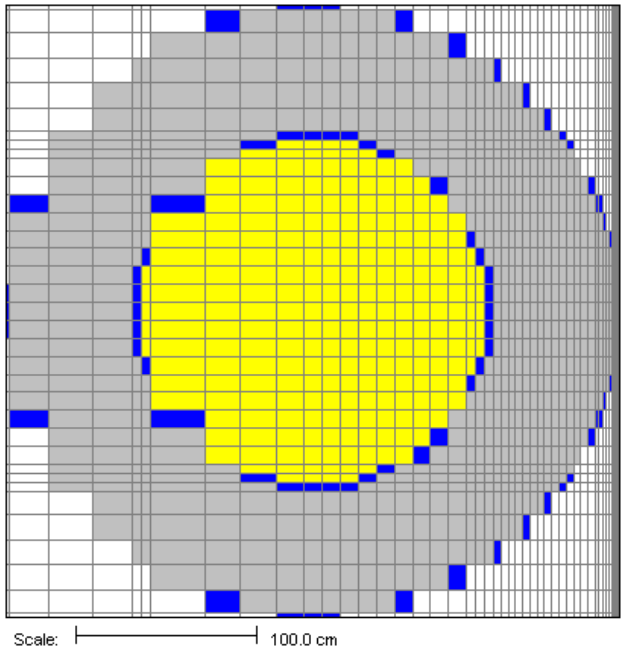
In-Class Example Problem

- Try a cask problem similar to the example shown in the MAVRIC manual
- Start with *starters/caskNanalog.inp*
- Use a coarse mesh (to speed things up)

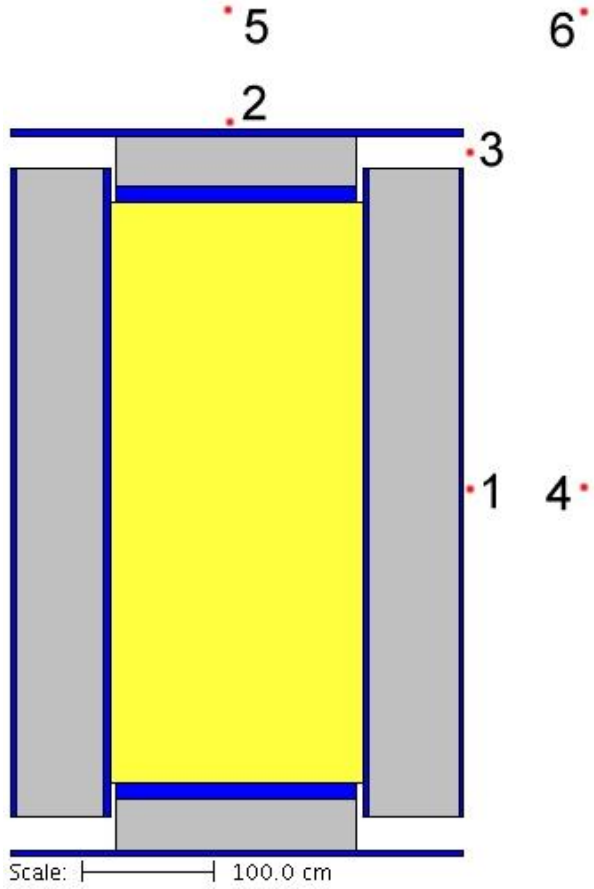
x (47)				*y (36)*			*z (50)*			
–170	15	108	156	–170	–55	65	–285.6	–50	210	264
–168	25	112	158	–168	–45	75	–280.6	–30	216.2	267.9
–146	35	116	160	–155	–35	85	–255.2	–10	222.4	271.8
–122	45	120	162	–141	–25	90	–240.6	10	228.6	275.7
–100	55	124	164	–127	–15	95	–228.6	30	232.6	279.6
–95	65	128	165	–113	–5	100	–210	50	236.6	280.6
–90	75	132	166	–100	5	113	–190	70	240.6	281.6
–60	85	136	167	–95	15	127	–170	90	245.1	282.6
–40	90	140	168	–90	25	141	–150	110	249.7	283.6
–20	95	144	169	–85	35	155	–130	130	254.2	284.6
–5	100	148	170	–75	45	168	–110	150	255.2	285.6
5	104	152		–65	55	170	–90	170	256.2	
							–70	190	260.1	

Coarse mesh for Denovo





Mesh specifically tailored for detector positions 3 and 6



Summary of the CADIS method in MAVRIC

- Minimal changes are needed to convert an analog MAVRIC to use the CADIS method
  - Add "importanceMap" block with parameters for Denovo
  - Use a new/existing "gridGeometry" for Denovo
- Adjoint flux solutions are obtained from Denovo and an importance map is created automatically
- If needed, available MAVRIC utilities can be utilized for further modifications to Denovo solutions

OAK RIDGE National Laboratory

Simultaneous optimization of several tallies with FW-CADIS

SCALE 6.3.2

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Learning goals

- Introduce attendees to the Forward-Weighted CADIS (FW-CADIS) variance reduction technique
- Identify the input and its syntax to use the FW-CADIS technique in MAVRIC

OAK RIDGE National Laboratory

FW-CADIS



Problem

- Analog Monte Carlo tallies tend to have relative uncertainties that are inversely proportional to flux
  - Low flux areas hardest to converge
  - Computation time is controlled by worst uncertainty
- Biasing (typically weight windows) helps move particles to areas of interest
  - Spend more time on "important" particles
  - Sacrifice results in "unimportant" areas
  - Different biasing for different tallies
- CADIS (weight windows and source) is very effective for source–detector type problems

Mesh tallies

- Monte Carlo is used to calculate many tallies
  - Mesh tally: answers everywhere
  - Wide range in relative uncertainties
- Many applications need mesh tallies
  - Dose rate maps
  - Activation of surrounding materials
  - Burnup of fuel pins in reactors

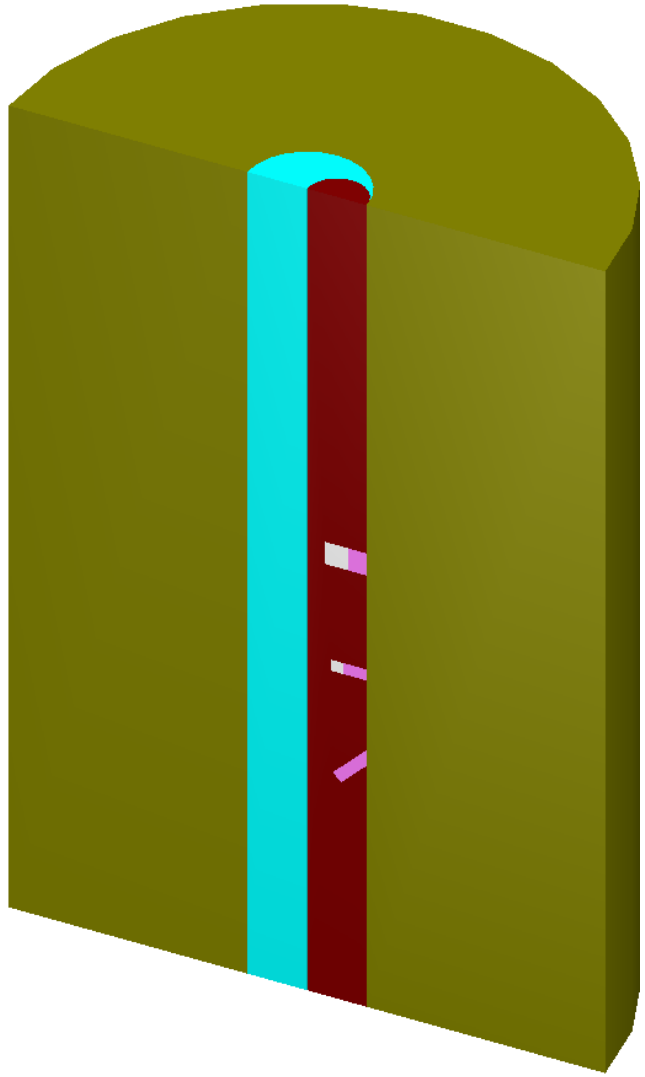


Gamma ray litho-density log

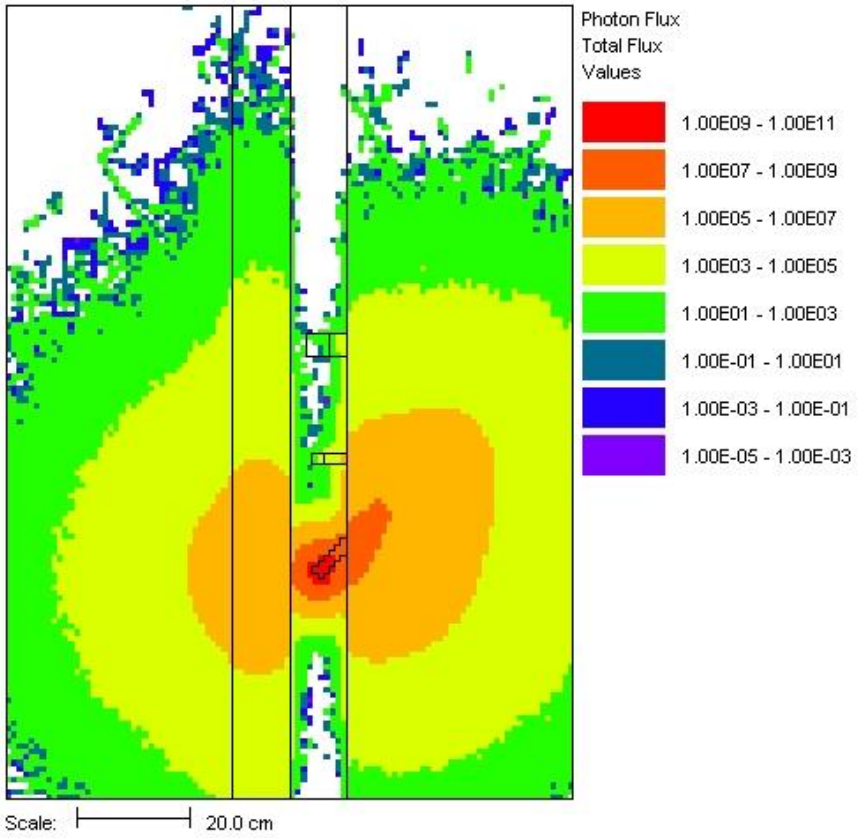
- Problem description from

  Robin P. Gardner and Kuruvilla Verghese, “Monte Carlo Nuclear Well Logging Benchmark Problems with Preliminary Intercomparison Results,” *Nuclear Geophysics* 5(4), 429-438 (1991).

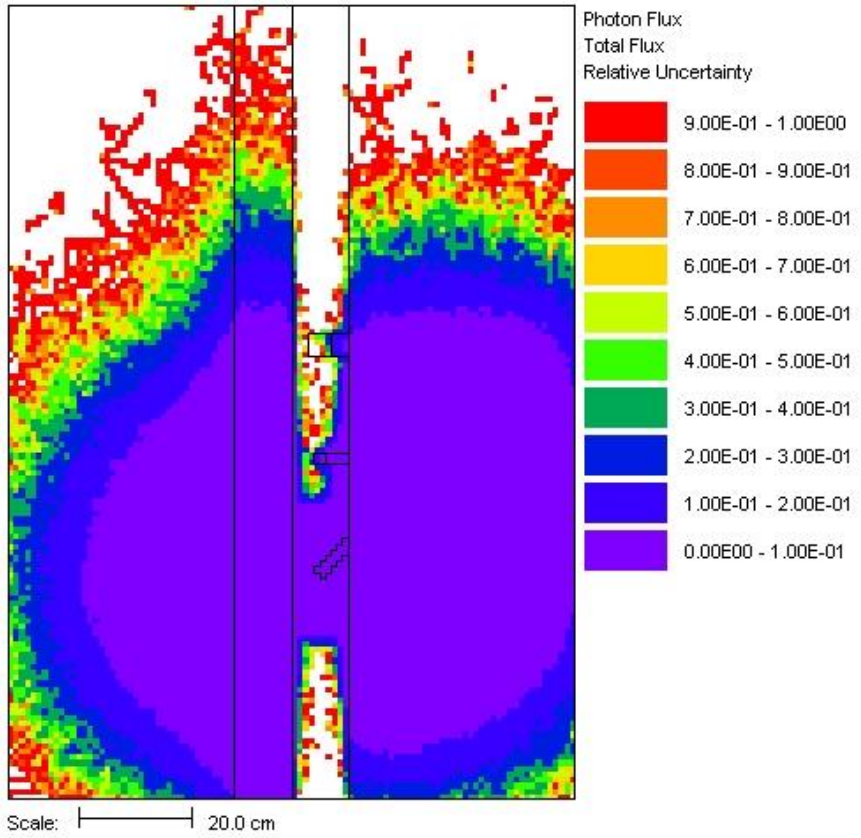
- Source: Cs-137, 2.7 Ci
- Detectors: NaI
  - Near: 2x2 at 20 cm
  - Far: 4x4 at 40 cm
- Borehole: 20 cm diameter
- Tool: 10 cm diameter



Analog



Total Photon Flux



Relative Uncertainty

Calculation	Minutes
Monte Carlo	2762

Results

Near	1.49×10^{3} (±8.2%)
Far	6.13×10^{1} (±19%)

How to improve uncertainties

- Weight windows/biased source with CADIS
  - Calculate adjoint fluxes from an adjoint source at one of the detector locations

$$q^{+}(\vec{r},E) = \sigma_d(\vec{r},E)$$

  - Estimate response

$$R = \int_{V_s}\int_{E} q(\vec{r},E)\,\phi^{+}(\vec{r},E)\,dE\,dV$$

  - Create importance map

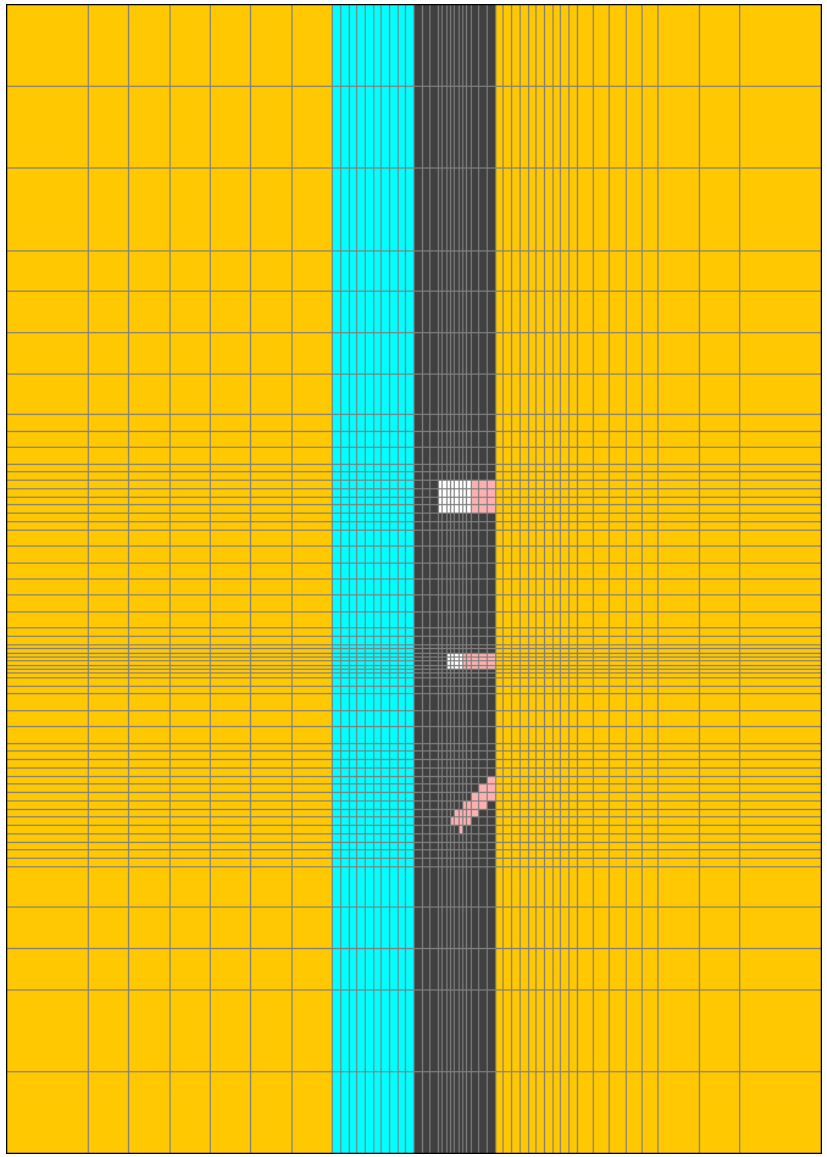
$$\overline{w}(\vec{r},E) = \frac{R}{\phi^{+}(\vec{r},E)}$$

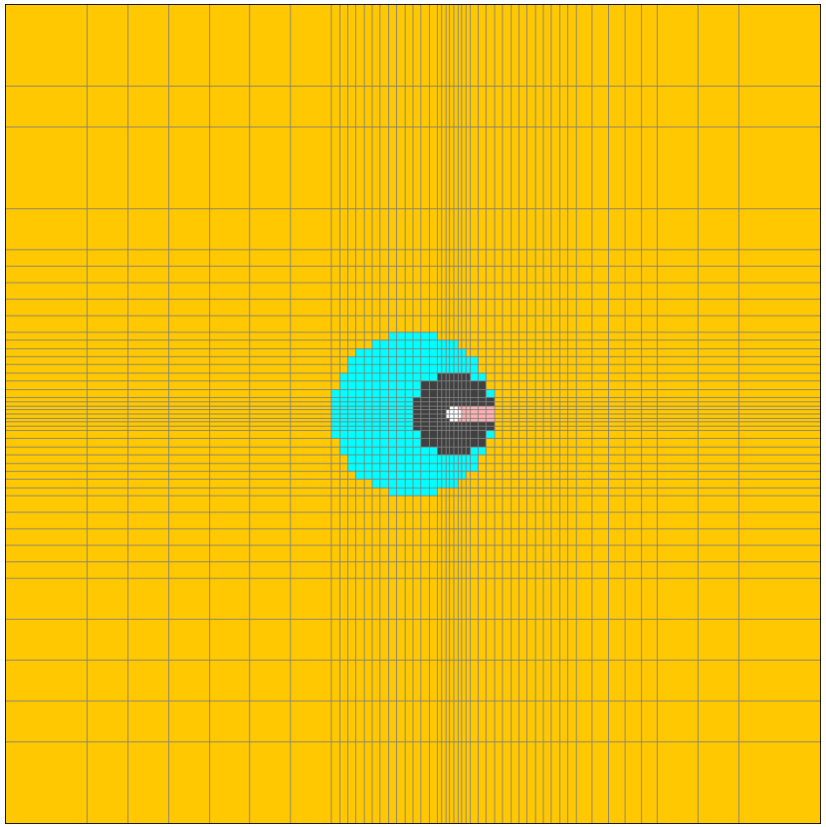
  - and consistent biased source

$$\hat{q}(\vec{r},E) = \frac{1}{R}\,q(\vec{r},E)\phi^{+}(\vec{r},E)$$

  - Run the forward Monte Carlo

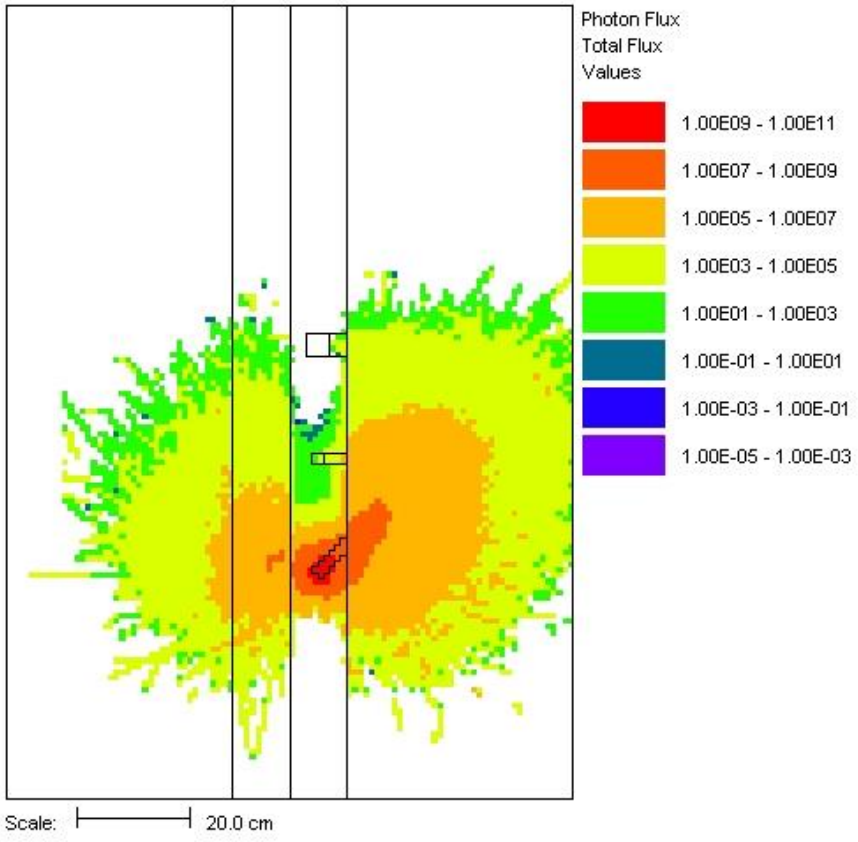
Discrete ordinates mesh



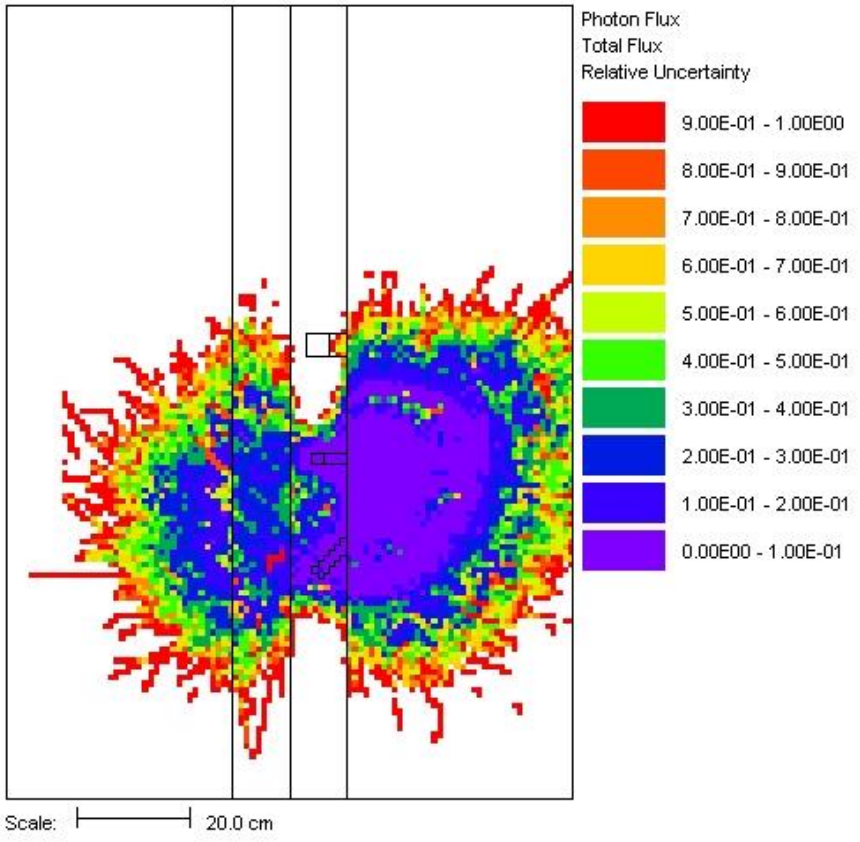


49×43×59 = 124,313

CADIS – for near detector



Total Photon Flux



Relative Uncertainty

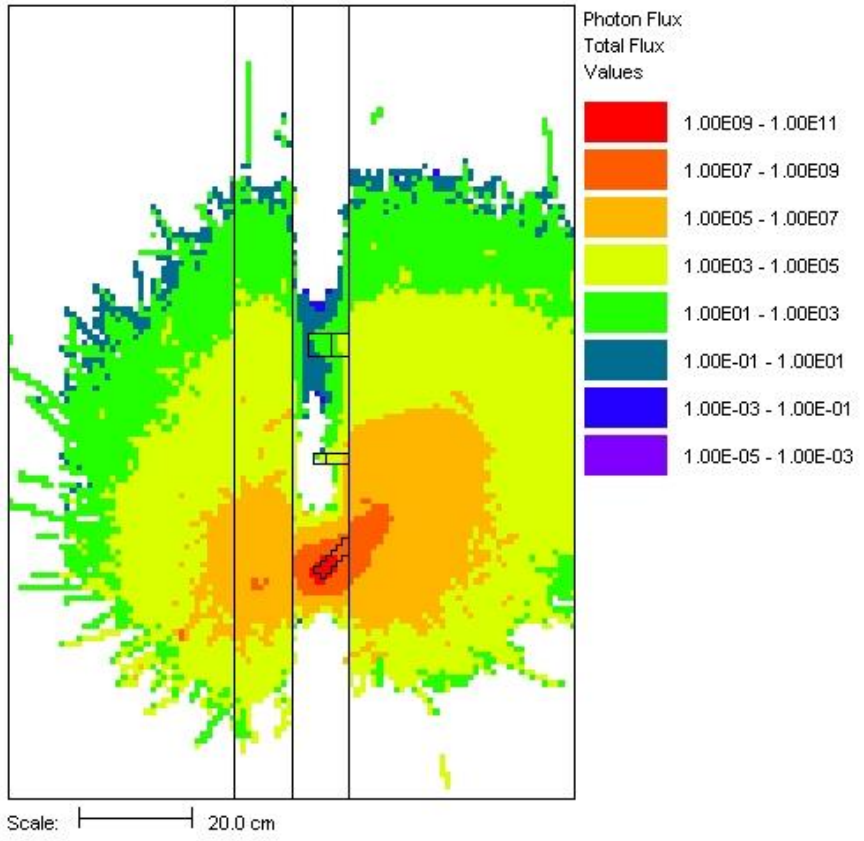
Calculation	Minutes
Adjoint DO	7
Monte Carlo	126

Results

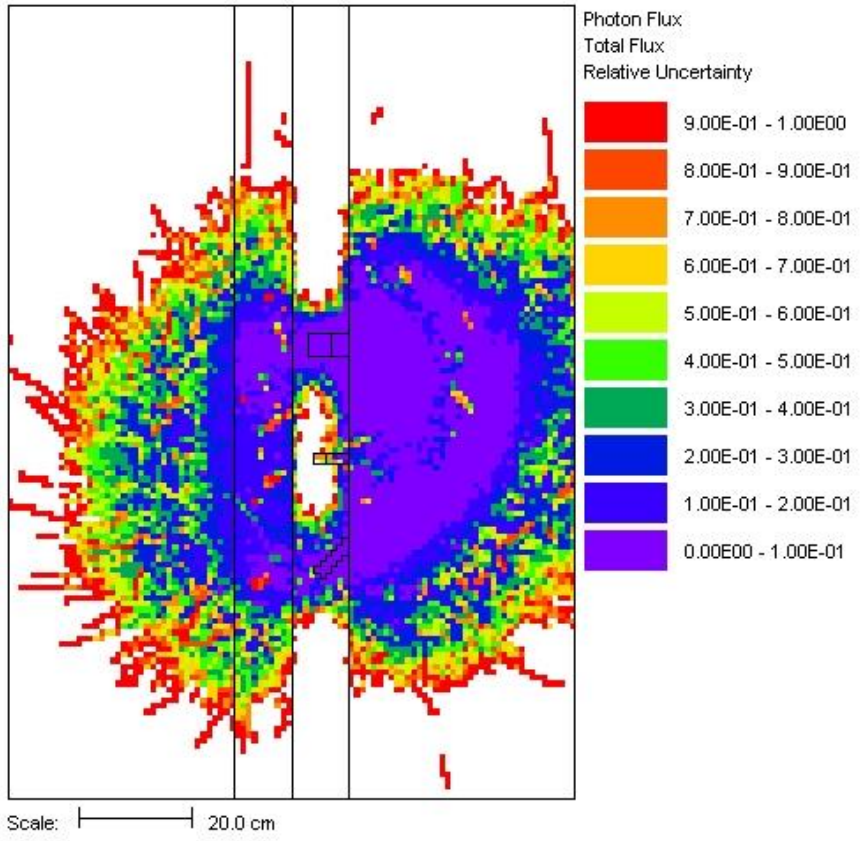
Near 1.54×10^3 (±0.5%)

Far 0.00

CADIS – for far detector



Total Photon Flux



Relative Uncertainty

Calculation	Minutes
Adjoint DO	7
Monte Carlo	124

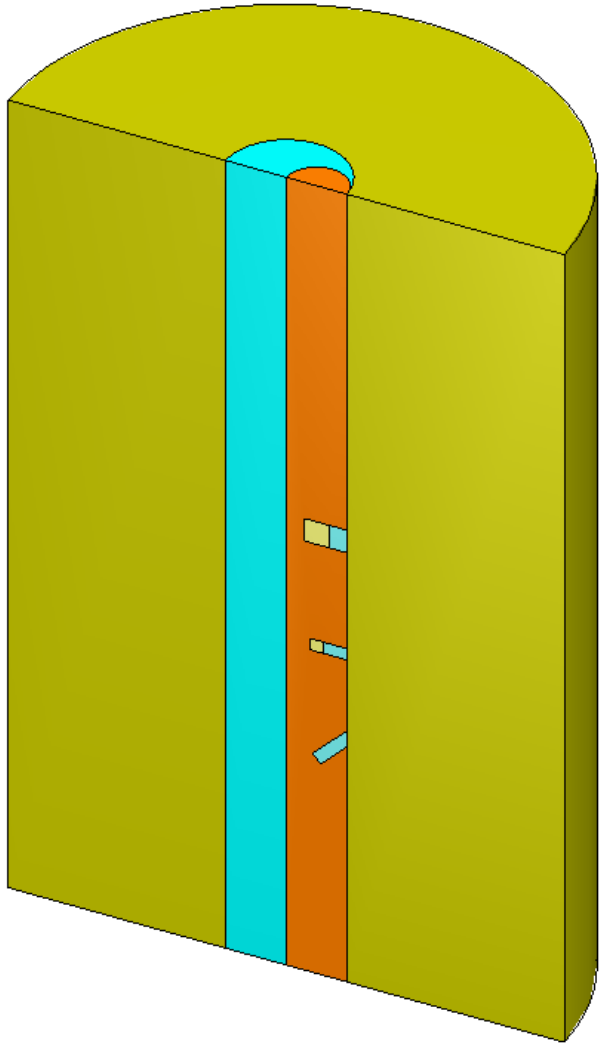
Results

Near 3.10×10^{3} (±65%)

Far 5.59×10^{1} (±0.3%)

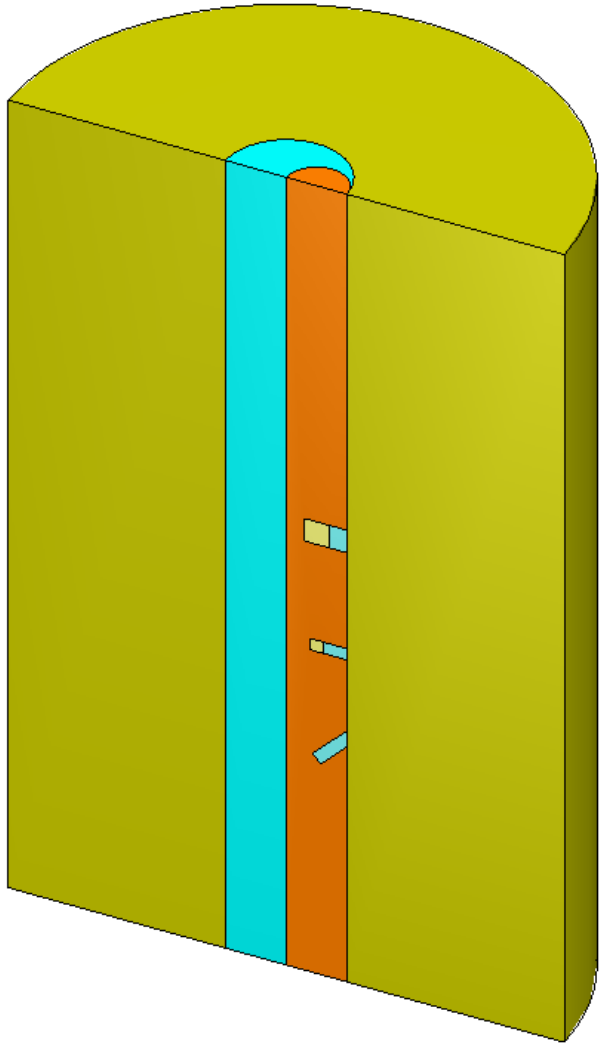
How to improve uncertainties

- CADIS works
  - Improves the figure of merit (FOM) for one detector
  - At the expense of tracking particles deep into the formation
  - At the expense of the other detector
- Can we get both detectors simultaneously?
- Try adjoint source in both detectors



How to improve uncertainties (cont.)

- Adjoint source in both detectors:
  - Recall that the adjoint source locations act like particle attractors in the MC simulations
  - MC particles will tend to go to the "easiest" adjoint source location
  - Need to put more adjoint source in the far detector so that the same number of particles get to each
- How much adjoint source strength to put into the far detector relative to the near?
- Ratio needs to be the same as ratio of the responses!



Litho-density log input file

```
read importanceMap
    gridGeometryID=1

'   near detector
    adjointSource 1
        boundingBox 6 4  1 -1  21 19
        unit=1 region=2
        responseID=1
        weight=1.00
    end adjointSource

'   far detector
    adjointSource 2
        boundingBox 7 3  2 -2  42 38
        unit=1 region=4
        responseID=1
        weight=27.55
    end adjointSource
end importanceMap
```

```
response 1
   title="total photon flux"
   photon
   values  19r1.0 end
end response
```

Sure would be nice if this could be done automatically…

FW-CADIS in MAVRIC

- Perform a forward discrete ordinates calculation
- Estimate the responses $R(\vec{r}, E)$everywhere
- Construct the CADIS adjoint source but weight the source strength with $1/R(\vec{r}, E)$

- Perform the adjoint discrete ordinates calculation
- Create the weight windows and biased source
- Perform the Monte Carlo calculation

FW-CADIS

- To achieve the same relative uncertainties in both detectors, the amount of adjoint source in each detector location must be inversely proportional to the expected response

For the optimization of:	Use adjoint source:
$\phi(\vec{r},E)$	$\boldsymbol{q^{+}(\vec{r},E)=\frac{1}{\phi(\vec{r},E)}}$
$\int \phi(\vec{r},E)dE$	$\boldsymbol{q^{+}(\vec{r},E)=\frac{1}{\int \phi(\vec{r},E)dE}}$
$\int \phi(\vec{r},E)\sigma_d(\vec{r},E)dE$	$\boldsymbol{q^{+}(\vec{r},E)=\frac{\sigma_d(\vec{r},E)}{\int \phi(\vec{r},E)\sigma_d(\vec{r},E)dE}}$

Litho-density log input file

```
read importanceMap
    gridGeometryID=1

'   near detector
    adjointSource 1
        boundingBox 6 4  1 -1  21 19
        unit=1 region=2
        responseID=1
    end adjointSource

'   far detector
    adjointSource 2
        boundingBox 7 3  2 -2  42 38
        unit=1 region=4
        responseID=1
    end adjointSource

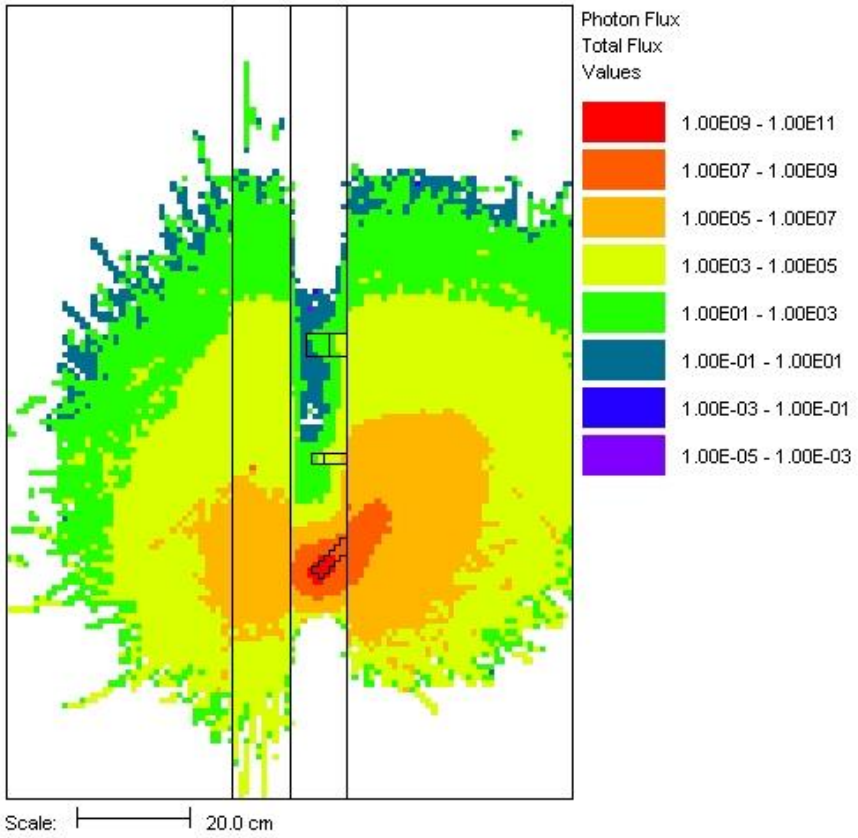
 respWeighting
end importanceMap
```

```
response 1
   title="total photon flux"
   photon
   values  19r1.0 end
end response
```

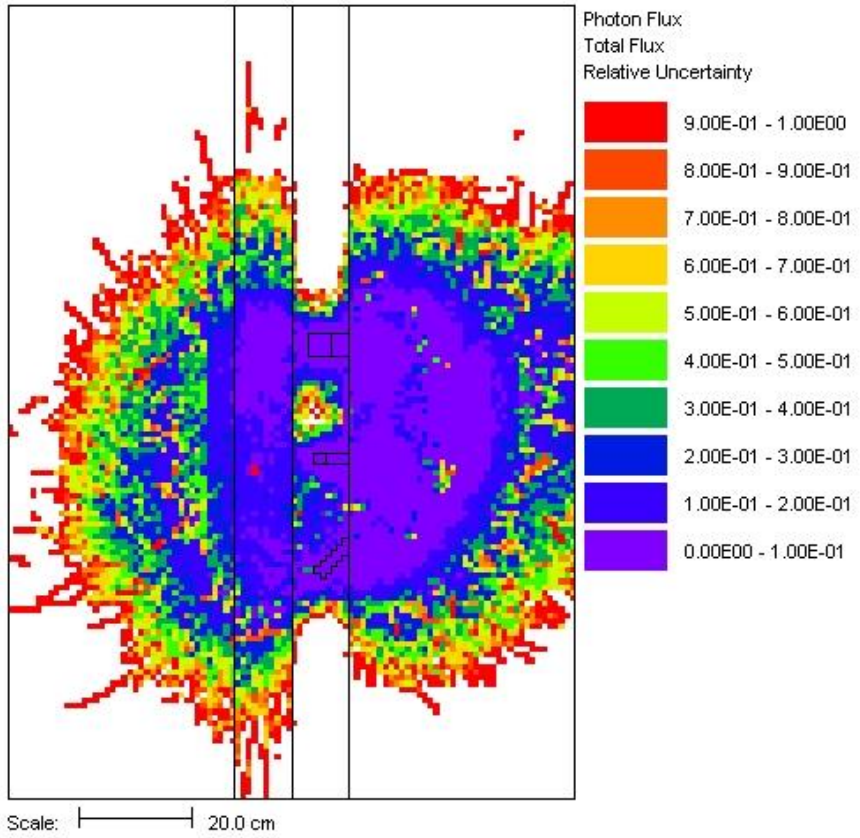
One keyword is almost automatic!

Use "***fluxWeighting***" or "***respWeighting***".

FW-CADIS – for both detectors



Total Neutron Flux



Relative Uncertainty

Calculation	Minutes
Forward DO	6
Adjoint DO	7
Monte Carlo	126

Results

Near 1.54×10^{3} (±0.6%)

Far 5.56×10^{1} (±0.4%)

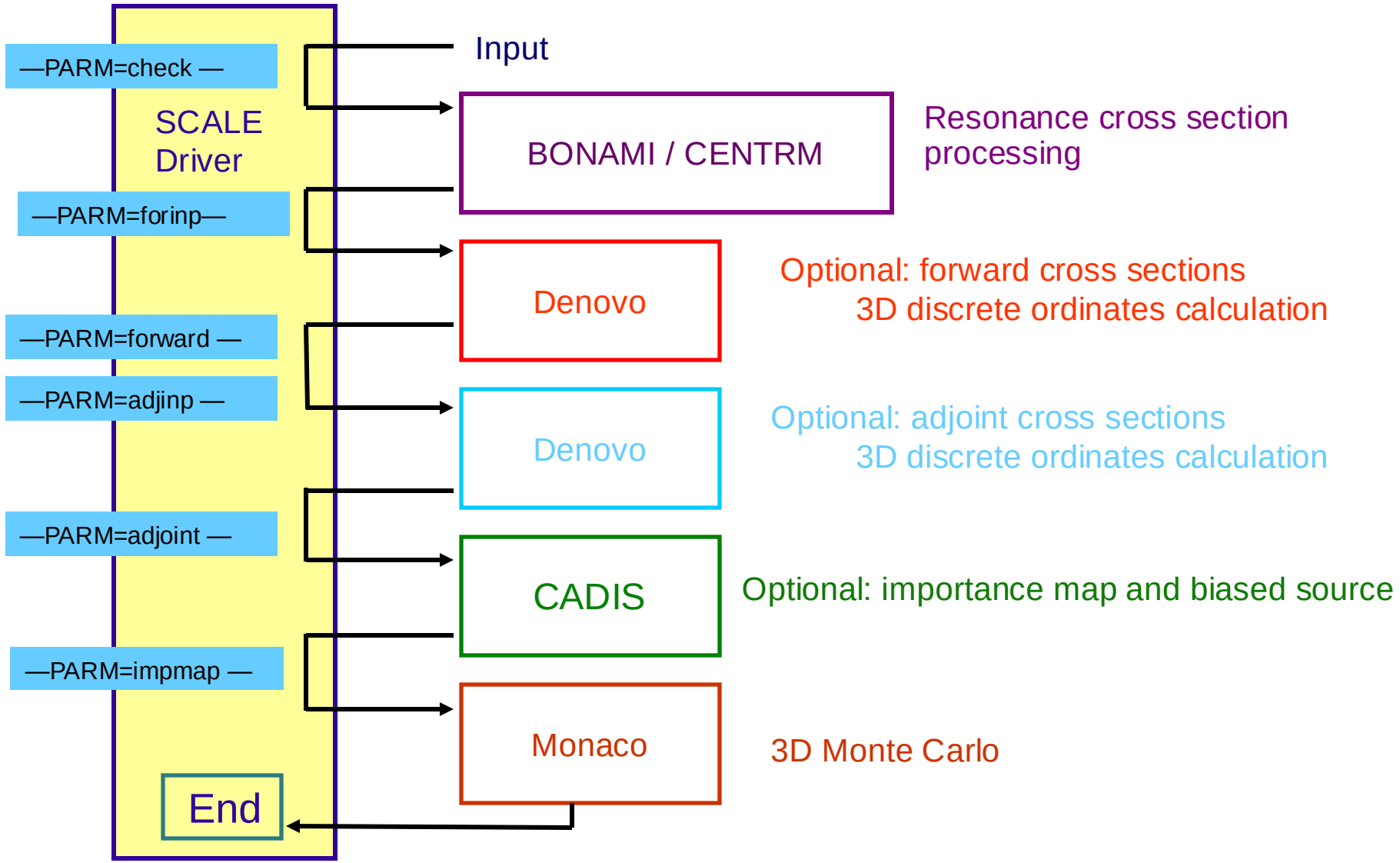
Photon methods comparison

	Analog	CADIS		FW-CADIS
		Near	Far	
Simulation times (min)				
Forward dis. ord.				6
Adjoint dis. ord.		7	7	7
Monte Carlo	2762	126	123	126
Near Detector				
Charged pairs/sec	1.494E+03	1.545E+03		1.543E+03
Relative uncertainty	8.15%	0.46%		0.61%
Far Detector				
Charged pairs/sec	6.130E+01		5.587E+01	5.564E+01
Relative uncertainty	19.14%		0.34%	0.43%
Monte Carlo FOM[a] ratio				
Near detector	1	6830		3867
Far detector	1		71821	43278
Time (min) required for 1%				
Near detector	183547	34		60
Far detector	1012136		21	36
Time (min) required				
For both detectors	1012136	55		60
Time improvement				
Speed-up over analog	1	18321		16748

FW-CADIS for mesh tally

- Mesh tally is just a large set of tallies for which we want roughly uniform relative uncertainties
- Instead of two tallies, there are $I \times J \times K$ tallies
- Adjoint source: the response function divided by the expected forward response in each cell of the mesh tally volume
  - Expected flux
  - Expected total flux
  - Expected dose rate

Sequence MAVRIC calls Monaco



Importance map block format

For weighting each adjoint source by space- and energy-dependent flux

```
read importanceMap
    adjointSource id  …   end adjointSource
    adjointSource id  …   end adjointSource
    …
    adjointSource id  …   end adjointSource

    gridGeometryID=g

    fluxWeighting
end importanceMap
```

For weighting each adjoint source by its energy-dependent response

```
read importanceMap
    …
    respWeighting
end importanceMap
```

Importance map block example

```
read importanceMap
   adjointSource 33
       boundingBox 10 -20  30 -30   25 0
       unit=2  region=4
       responseID=2
   end adjointSource
   gridGeometryID=1
   respWeighting
end importanceMap
```

For the optimization of:	Use adjoint source:
$\phi(\vec{r},E)$	$q^{+}(\vec{r},E)=\frac{1}{\phi(\vec{r},E)}$
$\int \phi(\vec{r},E)dE$	$q^{+}(\vec{r},E)=\frac{1}{\int \phi(\vec{r},E)dE}$
$\int \phi(\vec{r},E)\sigma_d(\vec{r},E)dE$	$q^{+}(\vec{r},E)=\frac{\sigma_d(\vec{r},E)}{\int \phi(\vec{r},E)\sigma_d(\vec{r},E)dE}$

FW-CADIS summary

- Used for optimizing multiple tallies at the same time
  - Convenient for optimizing mesh tallies
  - Each adjoint source is weighted by the forward flux solution
  - Ideal for global solutions in a large region
- A single keyword in the importance map block changes optimization
  - Optimize flux (`fluxWeighting`) or response (`respWeighting`)