ORNL/TM-2020/1601

KENO-VI Primer: Performing Calculations Using SCALE's Criticality Safety Analysis Sequence (CSAS6) with Fulcrum

Kursat Bekar
Justin Clarity
Mathieu Dupont
Robert Lefebvre
William Marshall
Ellen Saylor

December 2020

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-2020/1601

Nuclear Energy and Fuel Cycle Division

KENO-VI PRIMER: PERFORMING CALCULATIONS USING SCALE'S CRITICALITY SAFETY ANALYSIS SEQUENCE (CSAS6) WITH FULCRUM

Kursat Bekar
Justin Clarity
Mathieu Dupont
Robert Lefebvre
William Marshall
Ellen Saylor

Date Published: December 2020

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
AEC	US Atomic Energy Commission
CE	continuous energy
CSAS	Criticality Safety Analysis Sequence
FCM	fully ceramic microencapsulated
GUI	graphical user interface
HEU	highly enriched uranium
ICSBEP	International Criticality Safety Benchmark Experiments Project
IEU	intermediate-enriched uranium
INFHOMMEDIUM	infinite homogeneous medium
k_{eff}	effective multiplication factor
KMART6	KENO6 Module for Activity-Reaction Rate Tabulation
LEU	low-enriched uranium
MG	multigroup
MoC	method of characteristics
NEA	Nuclear Energy Agency
NRC	US Nuclear Regulatory Commission
NPG	neutrons per generation
NSK	number of generations skipped
OECD	Organization for Economic Co-operation and Development
ORNL	Oak Ridge National Laboratory
OS	operating system
PBMR	pebble bed modular reactor
PMC	produce multigroup cross sections
SGGP	SCALE Generalized Geometry Package
TRISO	tristructural isotropic
TSL	thermal scattering law
V7-238GROUP	ENDF/B-VII 238-group library

ABSTRACT

The SCALE code system developed at Oak Ridge National Laboratory is widely used and accepted around the world for criticality safety analysis. The well-known KENO-VI three-dimensional Monte Carlo criticality computer code is one of the primary criticality safety analysis tools in SCALE. The KENO-VI primer is designed to help a new user understand and use the SCALE/KENO-VI Monte Carlo code for nuclear criticality safety analysis. It assumes that the user has a college education in a technical field. There is no assumption of familiarity with Monte Carlo codes in general or with SCALE/KENO-VI in particular. The primer is designed to teach by example, with each example illustrating two or three features of SCALE/KENO-VI that are useful in criticality analysis.

The primer is based on SCALE 6.2 and 6.3, which includes the Fulcrum graphical user interface. Each example uses Fulcrum to provide the framework for preparing input data and viewing output results. Starting with a Quickstart section, the primer gives an overview of the basic requirements for SCALE/KENO-VI input and allows the user to quickly run a simple criticality problem with SCALE/KENO-VI. Each following section begins with a list of basic objectives identifying the goal of the section and the individual SCALE/KENO-VI features covered in detail in the section's sample problems. Upon completion of the primer, a new user should be comfortable using Fulcrum to set up criticality problems in SCALE/KENO-VI.

The primer provides a starting point for the criticality safety analyst who uses SCALE/KENO-VI. Complete descriptions are provided in the SCALE/KENO-VI manual. Although the primer is self-contained, it is intended as a companion volume to the SCALE/KENO-VI training and documentation. The SCALE manual and training schedule are available at https://scale.ornl.gov. The primer provides specific examples of using SCALE/KENO-VI for criticality analysis; the SCALE/KENO-VI manual provides information on the use of SCALE/KENO-VI and all its modules. The primer also contains an appendix with sample input files. In addition, this primer, its errata, and sample inputs are also available at https://code.ornl.gov/scale/primers/kenovi/.

1. INTRODUCTION AND PURPOSE

1.1 INTENDED AUDIENCE AND AVAILABLE RESOURCES

This primer is designed to help the user understand and use the KENO-VI three-dimensional (3D) Monte Carlo code for nuclear criticality safety analysis. KENO-VI is part of the SCALE code system [1]. It assumes that the user has a college education in a technical field. There is no assumption of familiarity with Monte Carlo codes in general or with SCALE and KENO-VI in particular. The primer is designed to teach by example, with each example illustrating two or three features of KENO-VI that are useful in criticality safety analysis. The theory and implementation details of KENO-VI are discussed in Goluoglu et al. [2].

The primer is based on SCALE 6.2 and 6.3, which includes the Fulcrum graphical user interface (GUI), a cross-platform program (available on Windows, Macintosh, and Linux) to assist criticality safety analysts in creating and executing SCALE input files for KENO-VI. Each example uses Fulcrum to provide the framework for data input. Starting with a Quickstart section, the primer gives an overview of the basic requirements for SCALE/KENO-VI input and allows the user to quickly run a simple criticality problem with SCALE/KENO-VI. This section is not designed to explain Fulcrum, the input, or the SCALE/KENO-VI options in detail; rather, it introduces the Fulcrum user interface and some basic concepts that are further explained in following sections. Each following section begins with a list of basic objectives identifying the goal of the section and the individual SCALE/KENO-VI features covered in detail in the section's example problems. It is expected that on completion of the primer, the user will be comfortable using Fulcrum to set up criticality problems in SCALE/KENO-VI. The user will be able to use SCALE/KENO-VI in criticality calculations and will be capable of handling most situations that normally arise in a facility. The primer provides a set of basic input files that can be selectively modified by the user to fit the particular problem at hand.

Although much of the information to perform an analysis is provided in the primer, there is no substitute for understanding the problem and the theory of neutron interactions. The SCALE/KENO-VI code is only capable of analyzing the problem as it is specified: it does not necessarily identify inaccurate modeling of the geometry; nor does it know when the wrong material has been specified. However, the 2D and 3D visualization capabilities of Fulcrum are quite useful for identifying geometry errors. Remember that a single calculation of k_{eff} and its associated confidence interval with SCALE/KENO-VI or any other code is meaningless without an understanding of the context of the problem, the quality of the solution, and a reasonable idea of what the result should be.

The primer provides a starting point for the criticality safety analyst using SCALE/KENO-VI. Complete descriptions are provided in the SCALE manual [1]. Although the primer is self-contained, it is intended to serve as a companion to the SCALE training courses and manual. The SCALE manual and training schedule is available at https://www.ornl.gov/scale. The primer provides specific examples using SCALE/KENO-VI for criticality safety analyses, whereas the documentation provides information on the use of SCALE and all its modules. The primer also contains appendices that give the user additional information about available cross section libraries in SCALE and example input files. This information is provided in appendices so as not to obscure the basic information illustrated in each example. It is also available at https://code.ornl.gov/scale/primers/kenovi/.

A standard set of notation has been provided to make the primer easy to use. The text is set in Times New Roman font. Information to be directly input is set in `Courier font`. Characters in the `Courier font` represent commands, keywords, or data that would be used as computer input. References to items displayed by Fulcrum are highlighted in **bold font**. Because the primer often references the SCALE manual, these references are set in square brackets; e.g., [see SCALE Manual Section *Title*]. To maximize the applicability of this primer across multiple SCALE releases, the manual section references refer to section titles and not section numbers.

It is hoped that users find the primer useful and easy to read. The user will benefit the most from this tutorial by starting with **Section 2: SCALE/KENO-VI *Quickstart*** and proceeding through the rest of the sections in order. Each section assumes that the user knows and is comfortable with the concepts discussed in the previous sections. Although it may be tempting to pick up the primer and immediately go to an example problem that is similar to an analysis requirement, this approach will not provide the user with the background or confidence in analysis that is necessary for accurate, effective implementation of procedures and limits. There is no substitute for a thorough understanding of the techniques used in a SCALE/KENO-VI analysis. A little extra time spent going through the primer and working through the examples will save many hours of possible confusion and frustration later.

1.2 SCALE OVERVIEW

The SCALE code system provides modern, robust calculations while reducing requirements for user input and knowledge of the intricacies of the underlying methods. SCALE provides standardized sequences to integrate many modern, advanced capabilities into a seamless calculation that the user controls from a single input file. SCALE's heritage dates back to 1969, when Oak Ridge National Laboratory (ORNL) began providing computational support in the use of the new KENO code to staff at the US Atomic Energy Commission, later the US Nuclear Regulatory Commission (NRC). The NRC has continued to instigate code and data improvements for criticality, shielding, reactor physics, and spent fuel characterization for various NRC applications. The NRC staff provided ORNL with the following general development criteria for SCALE:

1. focus on applications related to nuclear fuel facilities and package designs,

2. use well- established computer codes and data libraries,

3. design an input format for the occasional or novice user,

4. prepare "standard" analysis sequences (control modules) that will automate the use of multiple codes (functional modules) and data to perform a system analysis, and

5. provide complete documentation and public availability.

The concept of SCALE is to provide standardized sequences. Input for the control modules has been designed to be free-form, with extensive use of keywords and engineering-type input requirements. The most important feature of the SCALE system is the capability to simplify the user knowledge and effort required to prepare material mixtures and to perform adequate problem-dependent cross section processing.

Today, SCALE is a comprehensive modeling and simulation suite for nuclear safety analysis and design. SCALE is developed and maintained by ORNL under contract with the NRC, the US Department of Energy (DOE), and the National Nuclear Security Administration (NNSA) to perform reactor physics, criticality safety, radiation shielding, and spent fuel characterization for nuclear facilities and transportation/storage package designs.

1.3 CSAS OVERVIEW

The Criticality Safety Analysis Sequence (CSAS) provides a consolidated, streamlined user input for material information and model geometry specification. The material information input allows users to specify problem materials using easily remembered and easily recognizable keywords that are associated with mixtures, elements, and nuclides provided in the Standard Composition Library. The material information also uses other keywords and simple geometry input specifications to prepare input for the cross section processing module, XSProc. The XSProc module performs problem-dependent cross section processing using the BONAMI, CENTRM, PMC, and XSDRNPM functional modules. A keyword supplied by the user selects the cross section library from a standard set provided in SCALE or designates the reference to a user-supplied library.

CSAS has been developed to automate and standardize various analytic sequences. The control module input format has been designed to help minimize input errors. Upon processing the user-specified input, the SCALE code system immediately prints an input checklist in which the user (or reviewer) can easily establish that the input describes the system to be analyzed. The CSAS control module is the primary criticality safety control module for the calculation of the neutron multiplication factor of a system. Multiple sequences within the CSAS module provide capabilities for a number of analyses, such as modeling a one-dimensional (1D) or 3D system and processing cross sections.

The XSProc module provides cross section temperature correction and resonance self-shielding, as well as energy group collapse and spatial homogenization using the BONAMI, CENTRM, PMC, and XSDRNPM functional modules.

BONAMI performs resonance shielding through the application of the Bondarenko shielding factor method. BONAMI is typically used to process data in the unresolved resonance energy range. As input, BONAMI requires the presence of shielding factor data on the AMPX master library interface.

CENTRM computes continuous-energy (CE) neutron spectra using 1D discrete ordinates or infinite media geometry. CENTRM determines the problem-specific fluxes for processing resonance-shielded multigroup (MG) data by using a 1D or 2D unit cell and then using the spectrum as a problem-dependent weight function for MG averaging. The MG data processing is done by the PMC code, which reads the CENTRM CE flux spectra and cross section data and calculates problem-dependent, group-averaged cross sections over some specified energy range.

XSDRNPM is a 1D discrete ordinates transport code for performing neutron or coupled neutron-gamma calculations. The code has a variety of uses within SCALE. In the CSAS-XSProc sequence, it is used for preparation of cell-averaged cross sections for subsequent system analysis in KENO-VI and for 1D criticality safety analysis.

KENO-VI is a 3D MG Monte Carlo code employed to determine effective multiplication factors (k_{eff}) for multidimensional systems. The basic geometrical bodies allowed for defining the model are cuboids, spheres, cylinders, hemispheres, and cuboidal arrays. The model may be viewed using the visualization capabilities in Fulcrum.

Although the control modules are typically designated by their principal analytic sequence, it should be noted that more than one sequence may exist within a control module. The sequences available in CSAS are discussed in Section 3.

SCALE includes another version of KENO, known as KENO V.a. Since the geometry package for KENO-VI is different from that for KENO V.a, a different control module, CSAS5, provides an automated criticality safety sequence using KENO V.a for performing 3D Monte Carlo analyses of simpler geometries, with no support for hexagonal lattices or nonorthogonal geometry. Its simplified geometry system makes it very easy to use and much more computationally efficient than other Monte Carlo criticality safety codes. The geometry models consist of units that are constructed by nesting partial or whole spheres, cylinders, and rectangular cuboids oriented along X-, Y-, and Z-axes. These units can be combined in rectangular arrays or inserted in other units. These restrictions allow for very fast particle tracking, but they still permit users to construct complex geometry models.

KENO-VI was introduced to provide SCALE users with a more general geometry system, known as the *SCALE Generalized Geometry Package (SGGP)*, for constructing geometry models that are more complex. KENO-VI has a much larger assortment of bodies, including the following:

- cone
- cuboid
- cylinder
- dodecahedron
- elliptical cylinder
- ellipsoid
- hexprism
- hopper
- parallelepiped
- pentagon
- plane
- quadratic
- rotated hexprism
- rhomboid
- sphere
- wedge, and
- paired planes (parallel to X-, Y-, or Z-axis).

In addition to rectangular arrays, KENO-VI provides the dodecahedral array type, which is a 3D stack of dodecahedra, as well as three hexagonal array types.

The added complexity of the KENO-VI generalized geometry results in longer running times than those required for KENO V.a. In general, a KENO V.a model that has been converted to KENO-VI will require approximately two times the amount of CPU time in KENO-VI as it did in KENO V.a.

The SCALE code system includes several problem-independent MG cross section libraries for criticality analyses. The recommended general-purpose MG library is the 252-group ENDF/B-VII.1 neutron library [SCALE Manual *SCALE Nuclear Data Libraries*]. A 56-group library is also available for use with thermal systems, but the reduced number of groups does not provide an appreciable speed-up and can lead to large biases, especially for fast-spectrum systems. The CE libraries, based on ENDF/B-VII.0 and ENDF/B-VII.1, are also general-purpose libraries, though KENO execution time is increased in CE mode.

The SCALE XSProc modules used for preparing problem-dependent cross section libraries and performing criticality safety analyses are well established and in routine use by much of the US and international criticality safety communities as the primary computational tool or as a backup/review tool.

2. SCALE/KENO-VI QUICKSTART

2.1 WHAT YOU WILL BE ABLE TO DO

- Describe the structure of SCALE/KENO-VI input files.
- Use the Fulcrum user interface to create a SCALE/KENO-VI input file.
- Set up and run a simple criticality problem using SCALE/KENO-VI.
- Find and interpret k_{eff} information from SCALE/KENO-VI output.

2.2 SCALE/KENO-VI INPUT FILE

The SCALE/KENO-VI input file describes the problem geometry, specifies the materials and the neutron source, and defines the control parameters for analyzing the problem. The geometry is constructed by defining objects and their relationship with other objects in a system. Each object can be filled with a material or a void.

A SCALE/KENO-VI input file consists of some or all of the above data, depending on the type of problem being analyzed and the amount and type of output desired. The most user-friendly method for entering the data is to use the Fulcrum user interface.

SCALE interprets an entire line as a comment if it starts with an apostrophe (`'`) in column 1. The apostrophe must be in column 1, no matter where the text begins on the line. Entirely blank lines are ignored as white space. In-line comments, that is comments starting after other input on a line, are not supported. Comments and blank lines do not impact input processing, though comment lines are printed in the text output file.

2.3 EXAMPLE PROBLEM

This section provides enough information to run a simple example problem so that the user can gain confidence in using Fulcrum to enter SCALE/KENO-VI input data right away. This section walks the user through the example problem step by step, explaining each input step. At this stage, it is important that the user enter the problem exactly as it is described. With more experience using Fulcrum and SCALE/KENO-VI, the user may find other more logical ways to set up input files. For example, the user may find alternate methods for the geometry setup.

2.3.1 Problem Description

This problem is a bare sphere of delta-phase plutonium metal (density of 15.61 g/cc) with a coating of nickel (also known as *Jezebel*). Experimental parameters are as follows:

- Delta-phase Pu metal sphere: radius = 6.38493 cm
- Spherical nickel coating: thickness = 0.0127 cm
- N239 = Atom density of ^{239}Pu = 3.7047E–2 atoms/b-cm
- N240 = Atom density of ^{240}Pu = 1.751E–3 atoms/b-cm
- N241 = Atom density of ^{241}Pu = 1.17E–4 atoms/b-cm
- NGa = Atom density of Ga = 1.375E–3 atoms/b-cm
- NNi = Atom density of Ni = 9.1322E–2 atoms/b-cm

2.3.2 Fulcrum Interface – General Information

Now you are ready to begin entering the example problem. First start the Fulcrum user interface. Background information and an overview of the interface design and function are provided in Rearden et al. [3]. You should see a screen that looks like Figure 1. Note that Fulcrum screen captures throughout this document are made from both Mac and Windows computers; associated minor differences in the operating system (OS) specific appearance of windows is neither unexpected nor made consistent. These differences are inherited from the OS and will vary with user OS as well.

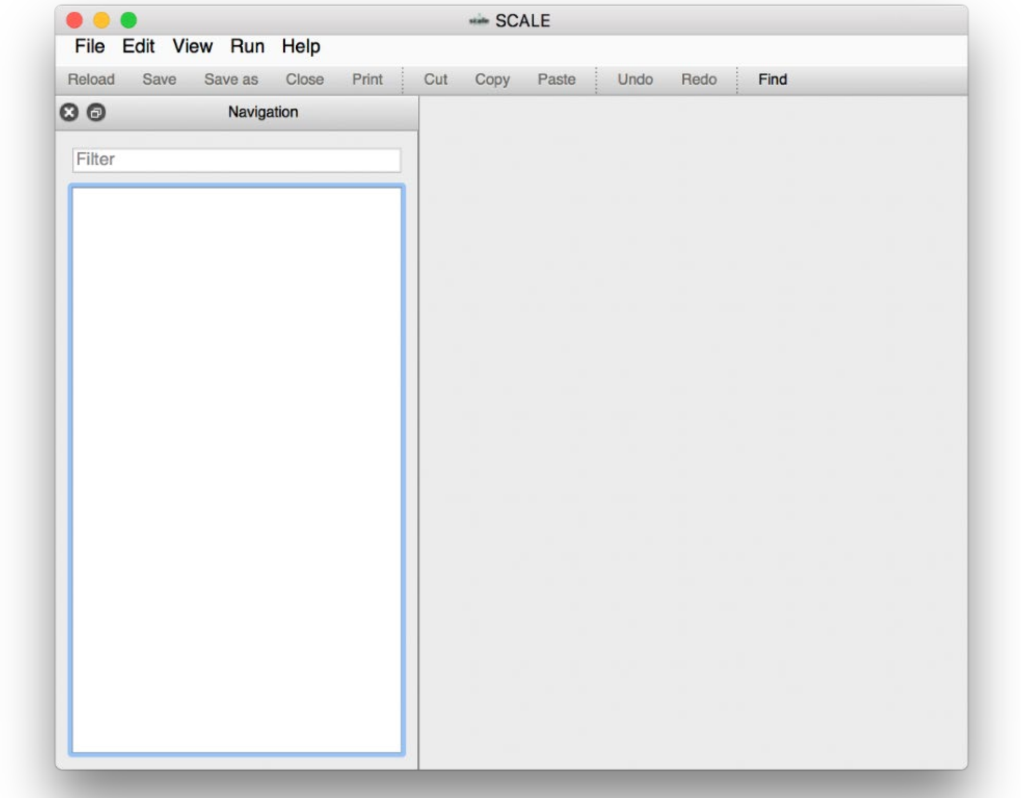

SCALE
File Edit View Run Help
Reload Save Save as Close Print Cut Copy Paste Undo Redo Find
Navigation
Filter

Figure 1. Fulcrum start screen.

Click the **File > New File...** button on the upper left to display the **New file** dialog window. In this window, enter the name of your input file with a **.inp** input file extension. Enter the input file name as `jezebel.inp` and click the **Save** button. A blank input document will be displayed as shown in Figure 2.

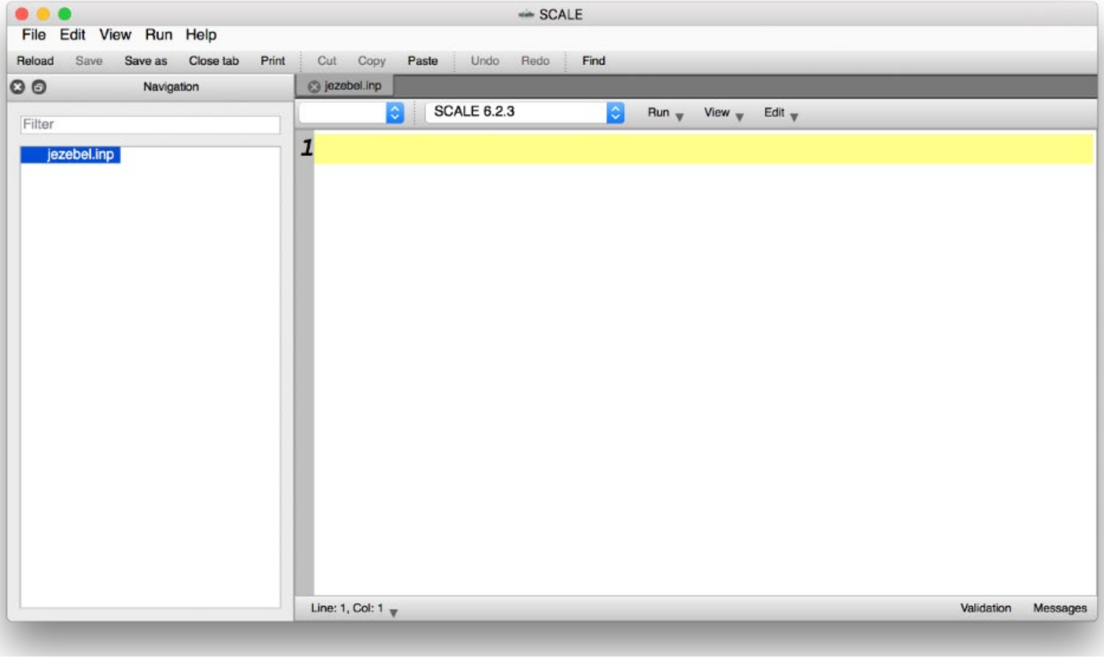
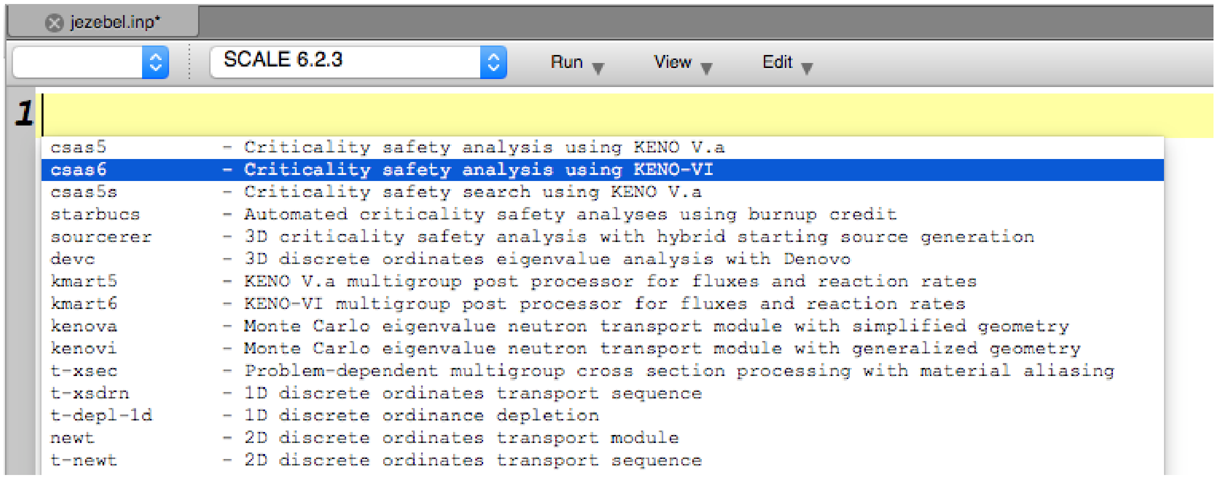


Figure 2. New blank jezebel input document.

The Fulcrum user interface is an advanced text editor with the primary capability of autocompleting input records. This capability is a design feature that allows Fulcrum to scale with the user's experience level, avoiding the inevitable impediments that GUI widgets can present to experienced users. The autocompletion feature is available whenever the cursor is inside the text editor panel, and it is accessible via the **Edit > Autocomplete** button or the `Control-Space` (`CTRL-SPACE`) keyboard combination.

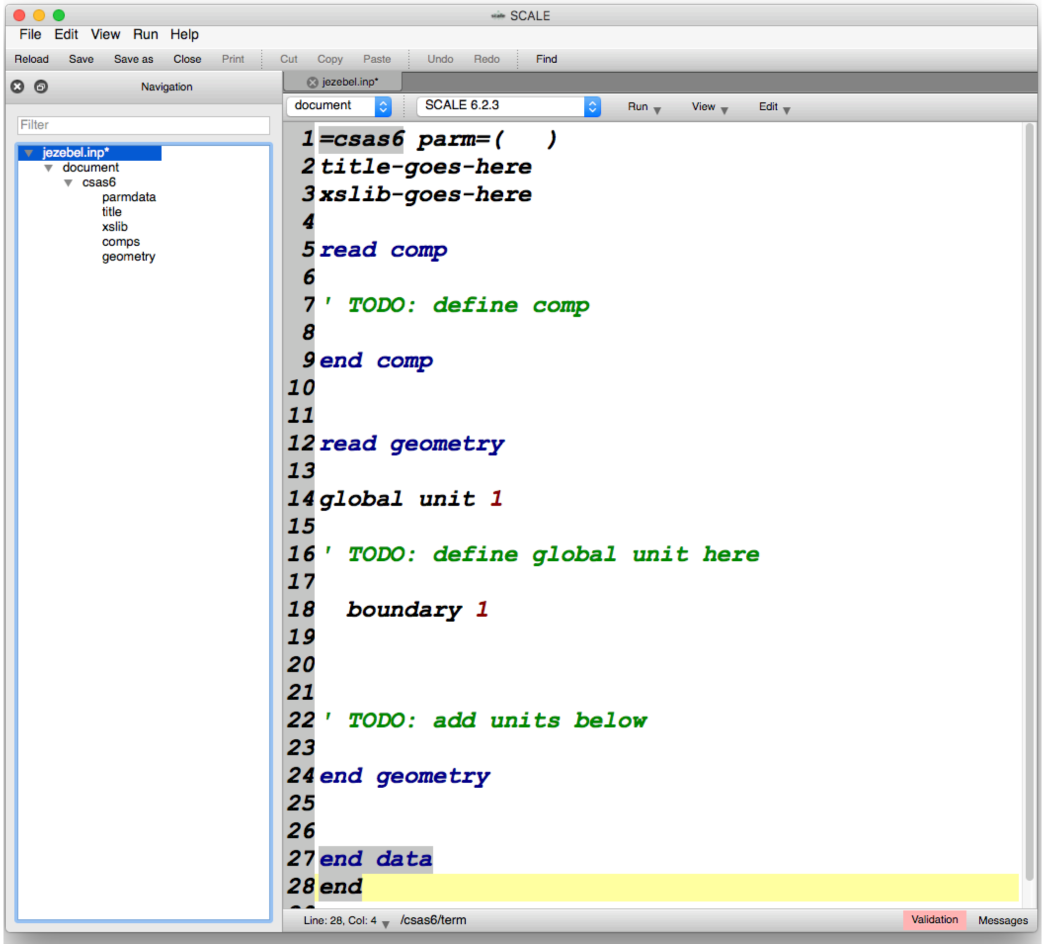
Keep in mind that information presented here that is to be entered into Fulcrum will appear in the `Courier font`. With the cursor on line 1, press the `CTRL-SPACE` autocomplete key combination and select the **csas6 – Criticality safety analysis using KENO-VI** option, as shown in Figure 3.



```
csas5          - Criticality safety analysis using KENO V.a
csas6          - Criticality safety analysis using KENO-VI
csas5s         - Criticality safety search using KENO V.a
starbucs       - Automated criticality safety analyses using burnup credit
sourcerer      - 3D criticality safety analysis with hybrid starting source generation
devc           - 3D discrete ordinates eigenvalue analysis with Denovo
kmart5         - KENO V.a multigroup post processor for fluxes and reaction rates
kmart6         - KENO-VI multigroup post processor for fluxes and reaction rates
kenova         - Monte Carlo eigenvalue neutron transport module with simplified geometry
kenovi         - Monte Carlo eigenvalue neutron transport module with generalized geometry
t-xsec         - Problem-dependent multigroup cross section processing with material aliasing
t-xsdrn        - 1D discrete ordinates transport sequence
t-depl-1d      - 1D discrete ordinance depletion
newt           - 2D discrete ordinates transport module
t-newt         - 2D discrete ordinates transport sequence
```

Figure 3. Autocompletion of the example sequence.

A skeleton CSAS6 input is created with placeholders and empty read blocks, as shown in Figure 4. These blocks streamline the additional updates needed to complete the input specification.



```
 1 =csas6 parm=(   )
 2 title-goes-here
 3 xslib-goes-here
 4
 5 read comp
 6
 7 ' TODO: define comp
 8
 9 end comp
10
11
12 read geometry
13
14 global unit 1
15
16 ' TODO: define global unit here
17
18   boundary 1
19
20
21
22 ' TODO: add units below
23
24 end geometry
25
26
27 end data
28 end
```

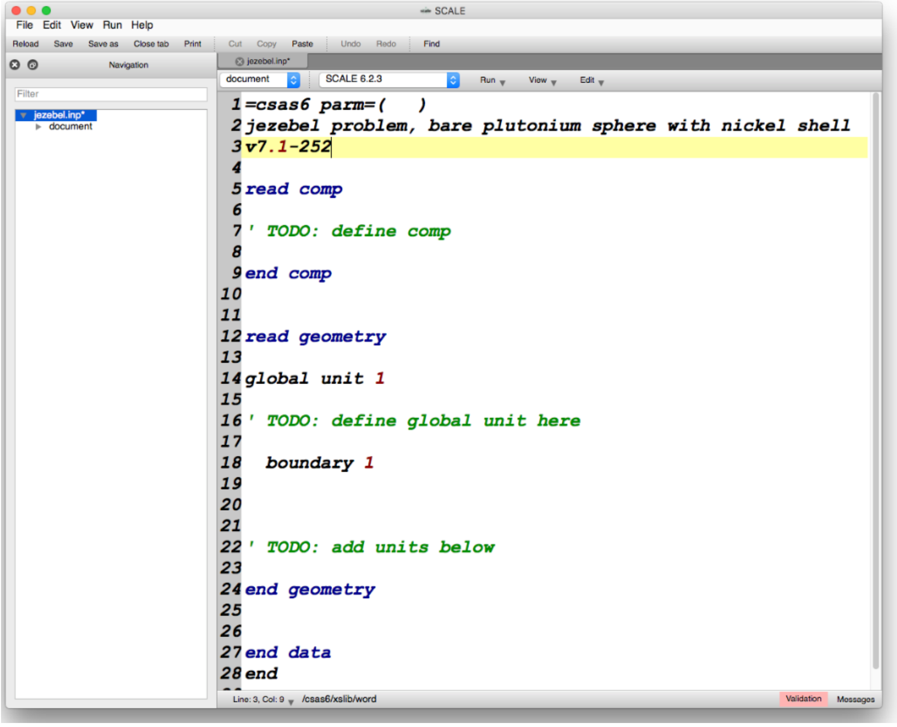
Figure 4. Example problem skeleton input.

With this CSAS6 skeleton available, start entering in the necessary parts of the problem. On line 2, replace the title placeholder, `title-goes-here` with:

```
jezebel problem, bare plutonium sphere with nickel shell
```

On line 3, replace the cross section library placeholder, `xslib-goes-here`, with the AMPX-generated 252 neutron group ENDF/B-VII.1 MG cross section library, `v7.1-252`.

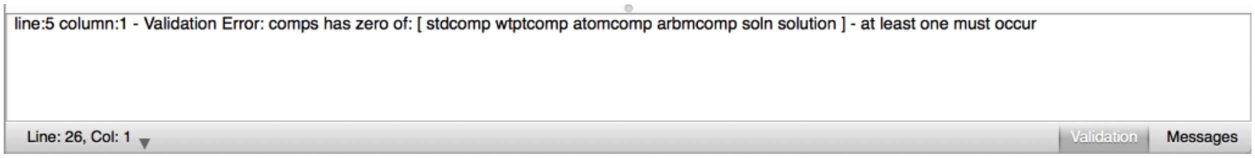
Figure 5 shows the CSAS6 input with the title and cross section library entered.



```
1 =csas6 parm=(    )
2 jezebel problem, bare plutonium sphere with nickel shell
3 v7.1-252
4
5 read comp
6
7 ' TODO: define comp
8
9 end comp
10
11
12 read geometry
13
14 global unit 1
15
16 ' TODO: define global unit here
17
18   boundary 1
19
20
21
22 ' TODO: add units below
23
24 end geometry
25
26
27 end data
28 end
```

Figure 5. General information for the example problem.

The **Validation** panel button is located in the lower right corner of the text editor panel. Notice that it has a red background, indicating that messages related to the correctness of the input are available for viewing. Click the **Validation** button to view the current input messages as depicted in Figure 6.



```
line:5 column:1 - Validation Error: comps has zero of: [ stdcomp wtptcomp atomcomp arbmcomp soln solution ] - at least one must occur
```

Figure 6. Validation panel indicating that required compositions are not present.

The validation message—

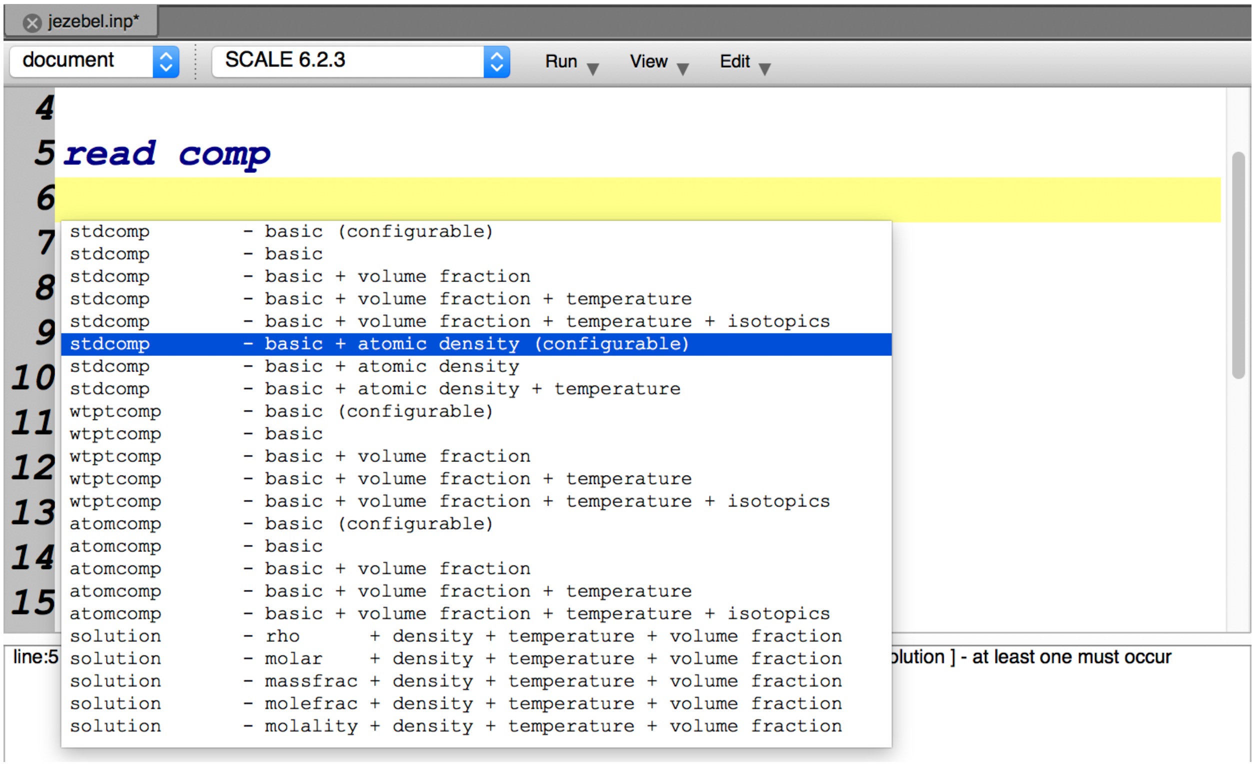
```
line:5 column:1 - Validation Error: comps has zero of: [ stdcomp
wtptcomp atomcomp arbmcomp soln solution ] - at least one must occur
```

—indicates that on line 5, column 1, the `read comp`, referred to as the `comps` input, has no entries specified in the available composition formats of `stdcomp wtptcomp atomcomp arbmcomp soln solution`. This transitions you to the **Materials** specification for the example problem shown in Figure 7.

The autocomplete functionality requires Fulcrum to be able to parse the input and identify key components, such as the **read** and **end** statements for each block. Malformed input can, in some situations, cause parse errors that preclude autocomplete functionality. These errors are provided in the **Validation** pane, and repairing these errors will restore autocomplete functionality.

2.3.3 Materials

The next section of input, the **read comp** block, provides information on the materials in the problem. For the example problem, there are four materials in the core region, and then the nickel shell in the outer region. To enter these, perform autocompletion within the comp block. Placing the text cursor within the read comp block and pressing the CTRL-SPACE autocomplete key combination will display the list of available composition entry formats: the **stdcomp – basic + atomic density (configurable)** shown in Figure 7 will be used for all problem material entries.



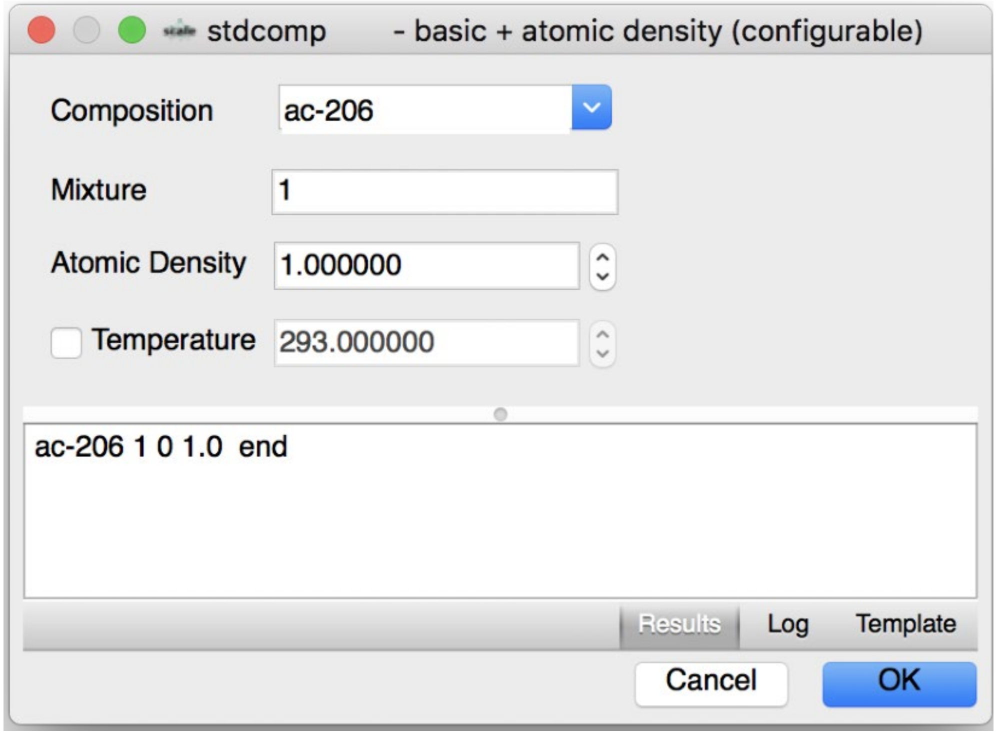
```
read comp
```

```
stdcomp        - basic (configurable)
stdcomp        - basic
stdcomp        - basic + volume fraction
stdcomp        - basic + volume fraction + temperature
stdcomp        - basic + volume fraction + temperature + isotopics
stdcomp        - basic + atomic density (configurable)
stdcomp        - basic + atomic density
stdcomp        - basic + atomic density + temperature
wtptcomp       - basic (configurable)
wtptcomp       - basic
wtptcomp       - basic + volume fraction
wtptcomp       - basic + volume fraction + temperature
wtptcomp       - basic + volume fraction + temperature + isotopics
atomcomp       - basic (configurable)
atomcomp       - basic
atomcomp       - basic + volume fraction
atomcomp       - basic + volume fraction + temperature
atomcomp       - basic + volume fraction + temperature + isotopics
solution       - rho      + density + temperature + volume fraction
solution       - molar    + density + temperature + volume fraction
solution       - massfrac + density + temperature + volume fraction
solution       - molefrac + density + temperature + volume fraction
solution       - molality + density + temperature + volume fraction
```

```
line:5
olution ] - at least one must occur
```

Figure 7. Example problem Basic Composition autocompletion.

For each available composition format, there is a **configurable** option. The **configurable** options display a widget with component labels and input preview. The options that are not listed as configurable will insert text variants of composition formats with default placeholders specified. Selecting the **stdcomp – basic + atomic density (configurable)** will display a widget with the default for the Basic Composition, as shown in Figure 8.



Composition ac-206

Mixture 1

Atomic Density 1.000000

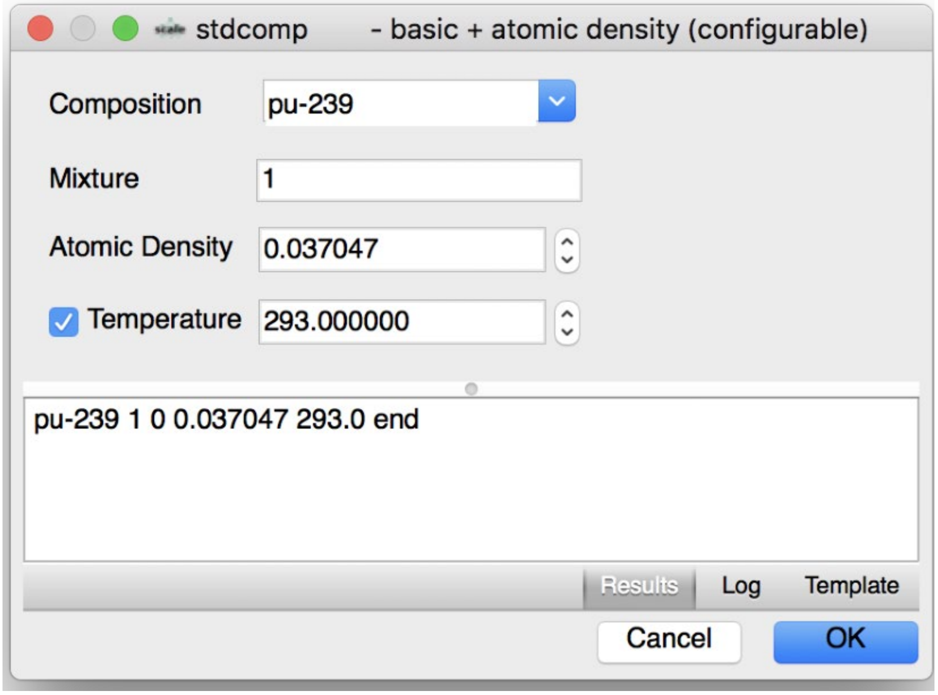
□ Temperature 293.000000

```
ac-206 1 0 1.0  end
```

Figure 8. Example default basic stdcomp configurable form.

You will be entering data for ^{239}Pu, ^{240}Pu, ^{241}Pu, gallium, and nickel, all of which are in the Standard Compositions Library and can be entered as Basic Compositions.

The first four materials are constituents of a single alloy and will therefore be part of mixture number 1. The first four materials are constituents of a single alloy and will therefore be part of mixture number 1. Four compositions will be entered for mixture 1. The entry for the first material, pu-239, should look like that shown in Figure 9. The **Mixture** should already be set to 1. Select pu-239 as the **Composition** from the list of materials in the library. The default **Temperature** is 293, but to incorporate it, click the **Temperature** Checkbox to the left of the **Temperature** label. To enter the atom densities given in the example problem description, you must select **Atomic Density** value field, and then enter the appropriate value. In this case, enter 0.037047.



Composition pu-239

Mixture 1

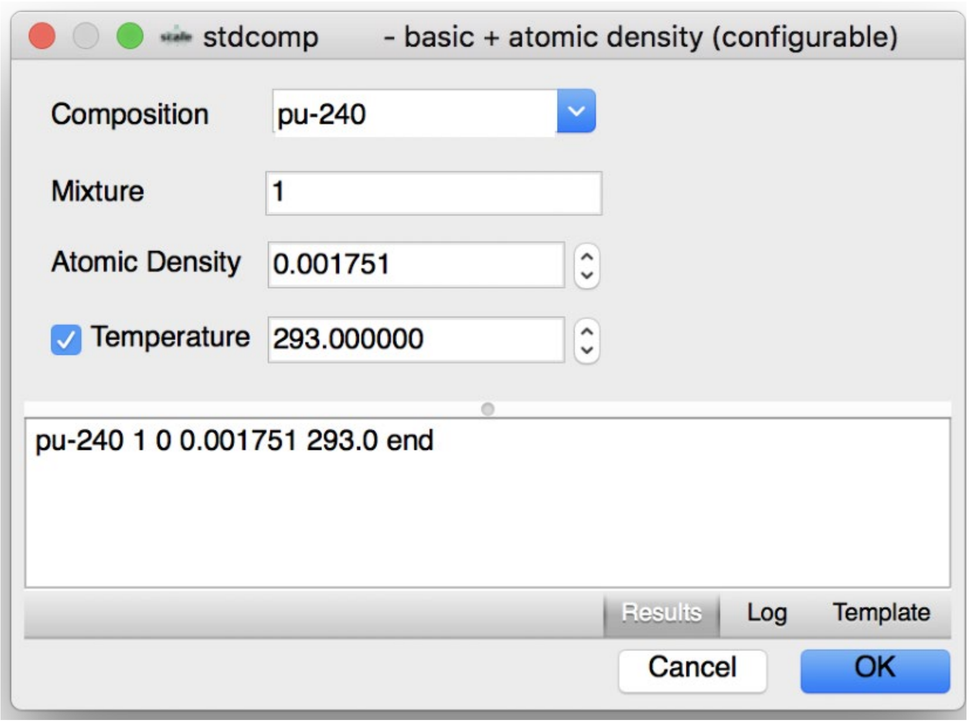
Atomic Density 0.037047

☑ Temperature 293.000000

```
pu-239 1 0 0.037047 293.0 end
```

Figure 9. Data for ^{239}Pu in mixture 1.

Select **OK** to insert the **Result** into the document. To enter the data for the remaining materials, autocomplete the **stdcomp - basic + atom density (configurable)** option. Then modify the **Composition Name** and **Atomic Density**. The mixture number should remain the same for each of the four materials. The entries for the next three materials (`pu240`, `pu-241`, and `ga`) look like those shown in Figure 10 through Figure 12.



stdcomp - basic + atomic density (configurable)

Composition pu-240

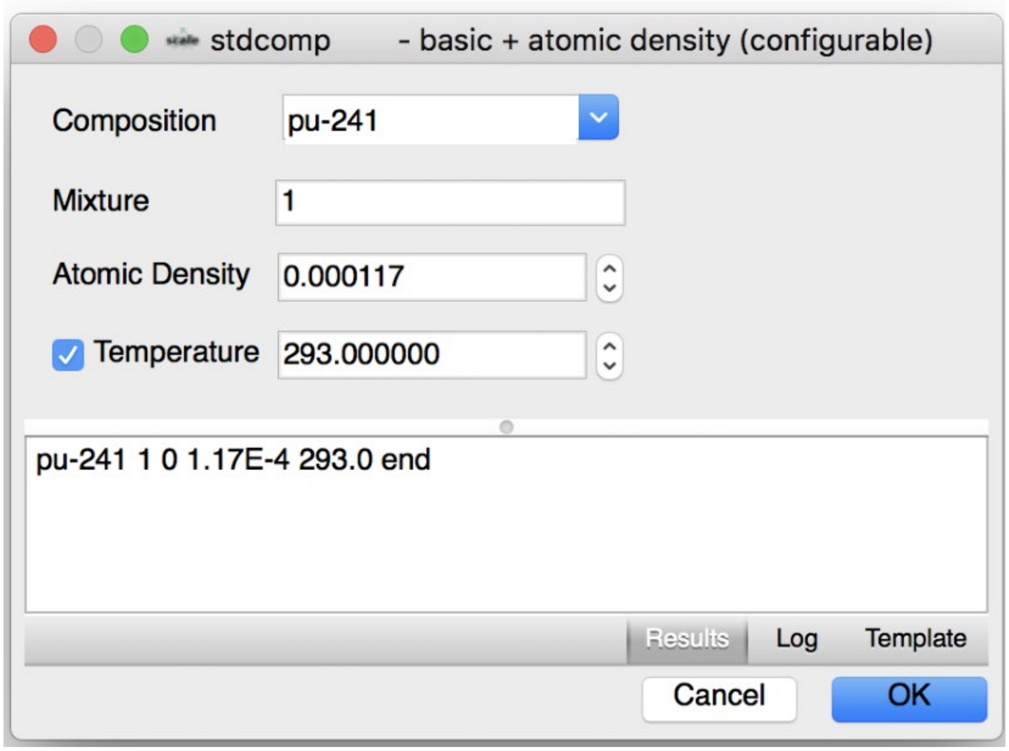
Mixture 1

Atomic Density 0.001751

☑ Temperature 293.000000

```
pu-240 1 0 0.001751 293.0 end
```

Figure 10. Data for ^{240}Pu in mixture 1.



stdcomp - basic + atomic density (configurable)

Composition pu-241

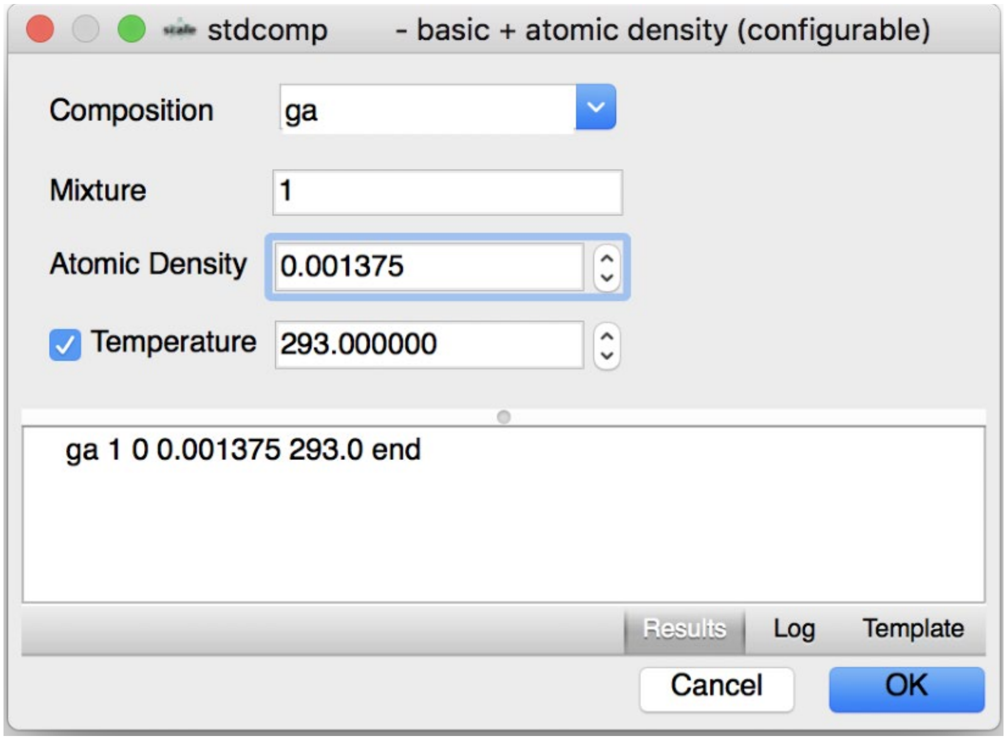
Mixture 1

Atomic Density 0.000117

☑ Temperature 293.000000

```
pu-241 1 0 1.17E-4 293.0 end
```

Figure 11. Data for ^{241}Pu in mixture 1.



Composition: ga

Mixture: 1

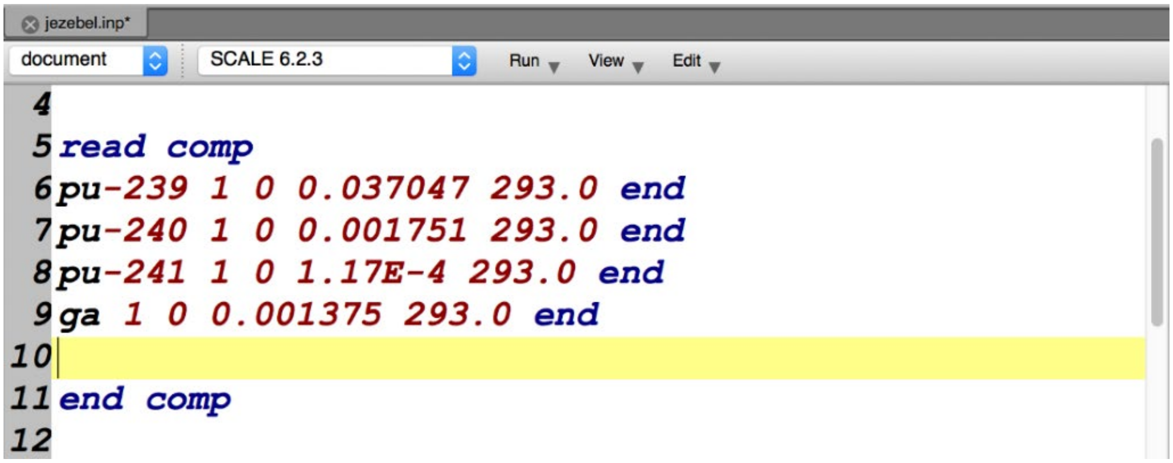
Atomic Density: 0.001375

☑ Temperature: 293.000000

```
ga 1 0 0.001375 293.0 end
```

Figure 12. Data for Gallium in mixture 1.

Figure 13 illustrates the composition records for mixture 1.



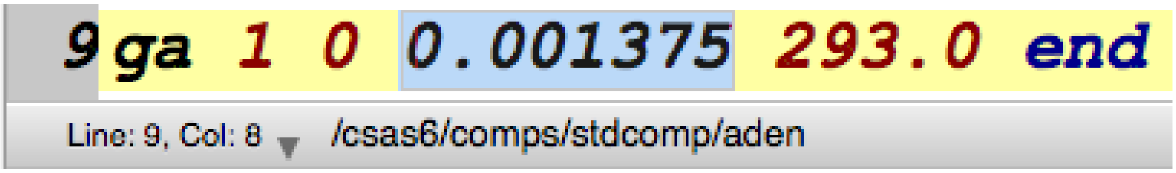
```

read comp
pu-239 1 0 0.037047 293.0 end
pu-240 1 0 0.001751 293.0 end
pu-241 1 0 1.17E-4 293.0 end
ga 1 0 0.001375 293.0 end

end comp

```

Figure 13. Mixture 1 basic composition summary.

Because material input can be position dependent, it is important to remember that Fulcrum can assist the user by presenting field labels at any given text cursor position. To determine what the fourth field is in

```
ga 1 0 0.001375 293.0 end
```

place the text cursor on the field and observe the cursor context display the parameter's path **/csas6/comps/stdcomp/aden**, as shown in Figure 14, at the lower left of the text editor, adjacent to the cursor's line and column numbers.



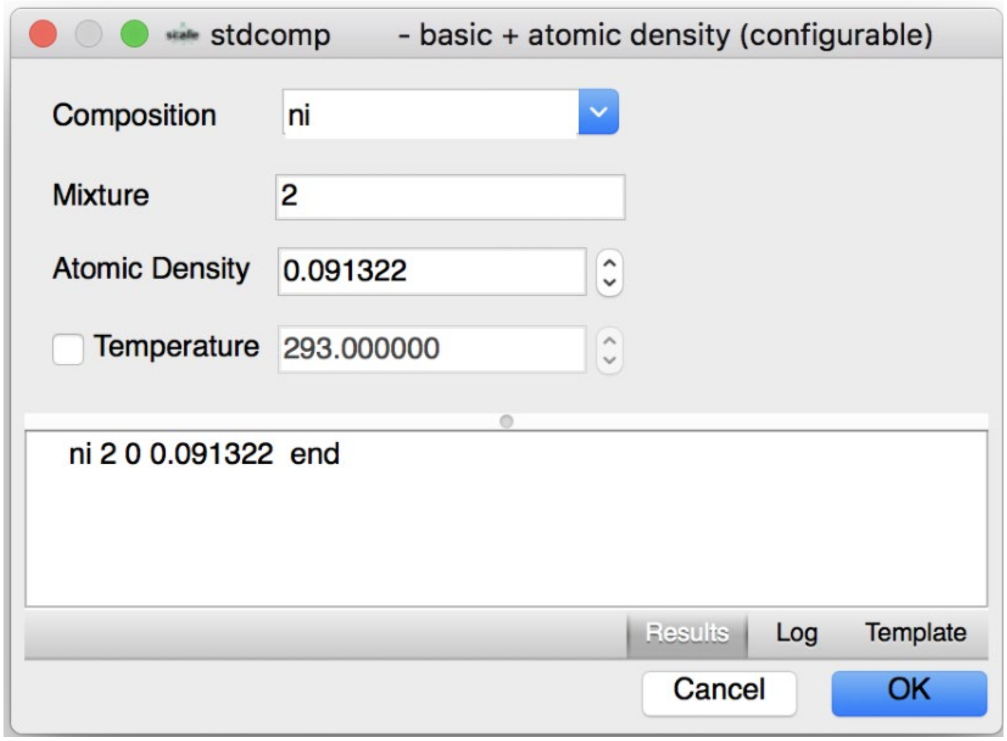
```
ga 1 0 0.001375 293.0 end
```

Line: 9, Col: 8 /csas6/comps/stdcomp/aden

Figure 14. Example input text cursor context.

This cursor context provides the structure and naming of all SCALE input fields. **Aden** is the shorthand name for **Atomic Density**. The context indicates that **aden** is a field in the **stdcomp** record within the **comps** block of the **CSAS6** sequence. These are a pseudo-directory listing of SCALE input that can assist you with identifying unlabeled input fields. Parse errors, like those mentioned in Section 2.3.2 that may disable autocomplete capability, also cause the loss of this cursor context. Moving the cursor through the input until cursor context is restored can help identify the location of problematic input causing the parse errors.

The last material to specify is the nickel shell in the outer region. Again, perform an autocompletion of the **stdcomp – basic + atomic density (configurable)** option to bring up the input entry form. Select `ni`, and then select **Mixture** of `2`. For the **Atomic Density**, enter `0.091322`. Use the default **Temperature** of `293` and click the checkbox to enable its explicit use in the composition input record. The completed screen for nickel should look like that shown in Figure 15. After entering the data for nickel, select **OK** to insert the **Result** into the document.



Composition: ni

Mixture: 2

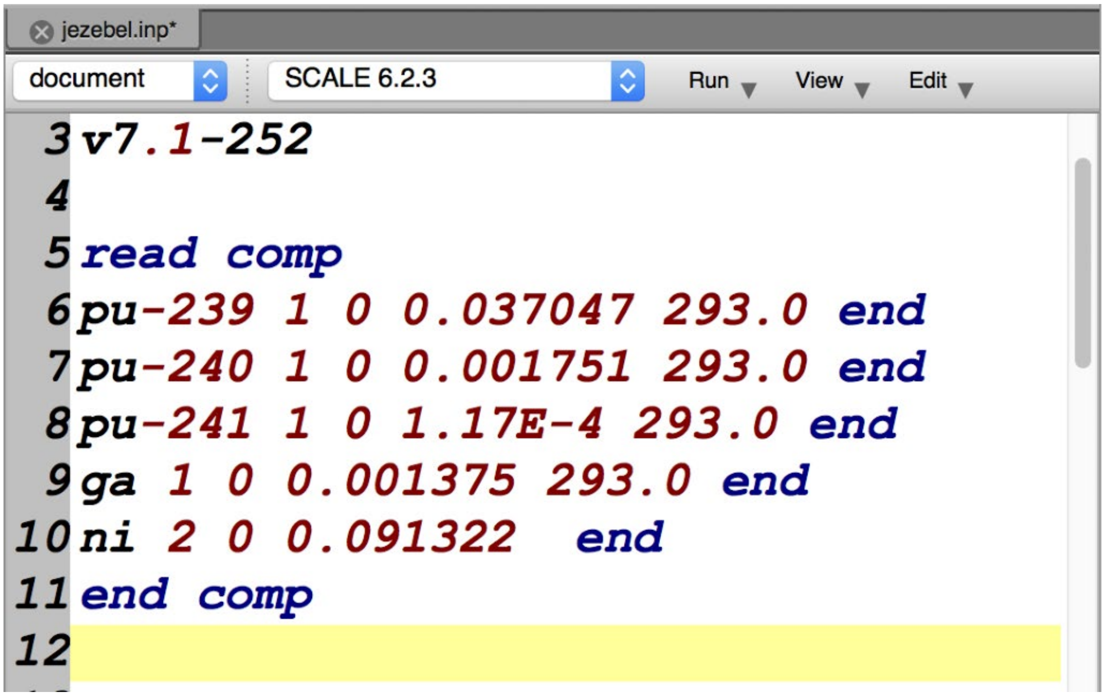
Atomic Density: 0.091322

☐ Temperature: 293.000000

```
ni 2 0 0.091322  end
```

Figure 15. Data for nickel in mixture 2.

Figure 16 illustrates the complete example composition block.

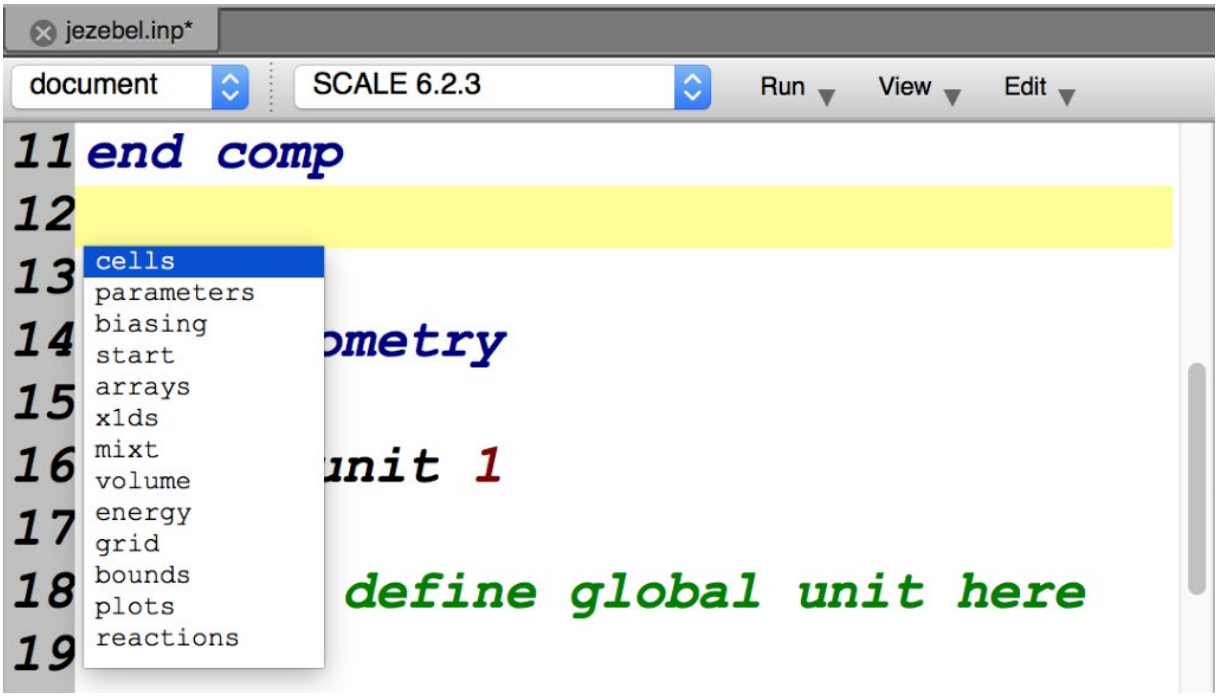


```
v7.1-252

read comp
pu-239 1 0 0.037047 293.0 end
pu-240 1 0 0.001751 293.0 end
pu-241 1 0 1.17E-4 293.0 end
ga 1 0 0.001375 293.0 end
ni 2 0 0.091322  end
end comp

```

Figure 16. Complete example problem composition block.

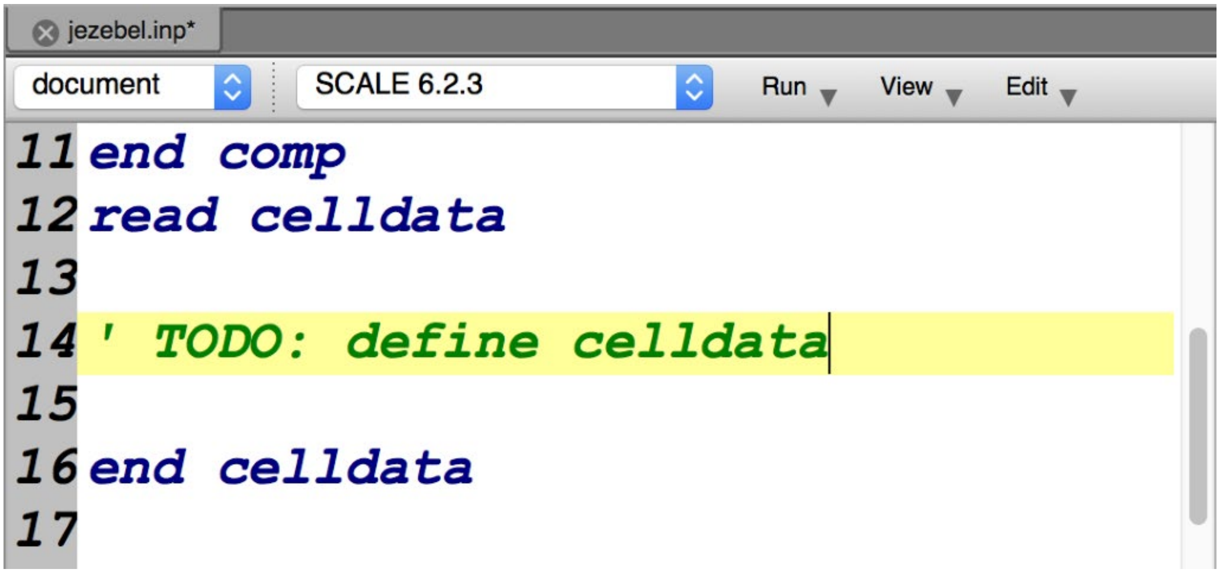
Because this example problem uses MG cross sections, a processing method to incorporate spatial affects in the cross section preparation is also specified. The unit cell input parameters are specified in the **read**

celldata block. With your text cursor between the composition and geometry blocks, press the `CTRL-SPACE` key combination to display the available input blocks, and select the **cells** option as shown in Figure 17. Figure 18 illustrates the empty **read celldata** ready for further autocompletion of the example problem's cell information.



```
end comp

ometry
unit 1
define global unit here
```

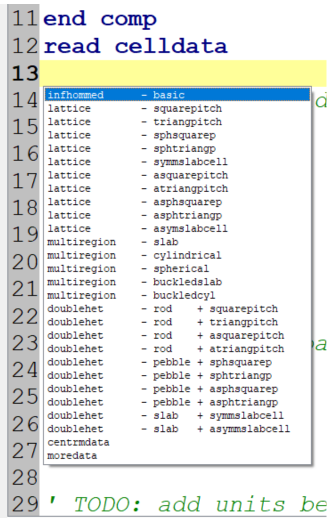
Figure 17. Autocompletion of the cells input block.



```
end comp
read celldata

' TODO: define celldata

end celldata

```

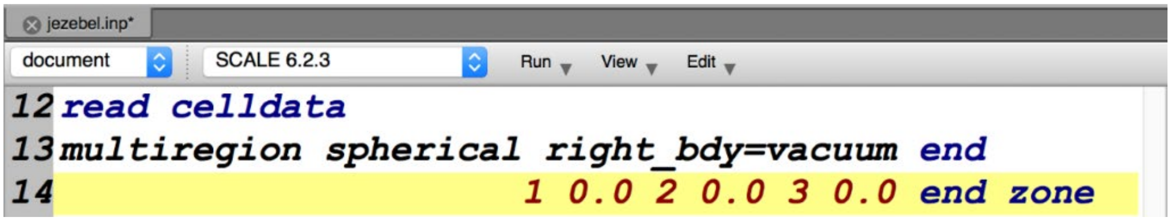
Figure 18. Example autocompleted initial celldata input block.

With the text cursor inside the celldata block, press the `CTRL-SPACE` key combination to display the available celldata types, and select the **MultiRegion - spherical** option as shown in Figure 19. Figure 20 illustrates the default spherical multiregion with placeholders for right or outer boundary, **right_bdy**, and cell zone mixture-radii pairs. Use the default outer boundary of `vacuum`; the inner (left) boundary condition of a spherical unit cell must be reflective and cannot be changed.



```
end comp
read celldata

infhommed       - basic
lattice         - squarepitch
lattice         - triangpitch
lattice         - sphsquarep
lattice         - sphtriangp
lattice         - symmslabcell
lattice         - asquarepitch
lattice         - atriangpitch
lattice         - asphsquarep
lattice         - asphtriangp
lattice         - asymslabcell
multiregion     - slab
multiregion     - cylindrical
multiregion     - spherical
multiregion     - buckledslab
multiregion     - buckledcyl
doublehet       - rod    + squarepitch
doublehet       - rod    + triangpitch
doublehet       - rod    + asquarepitch
doublehet       - rod    + atriangpitch
doublehet       - pebble + sphsquarep
doublehet       - pebble + sphtriangp
doublehet       - pebble + asphsquarep
doublehet       - pebble + asphtriangp
doublehet       - slab   + symmslabcell
doublehet       - slab   + asymmslabcell
centrmdata
moredata

' TODO: add units be
```

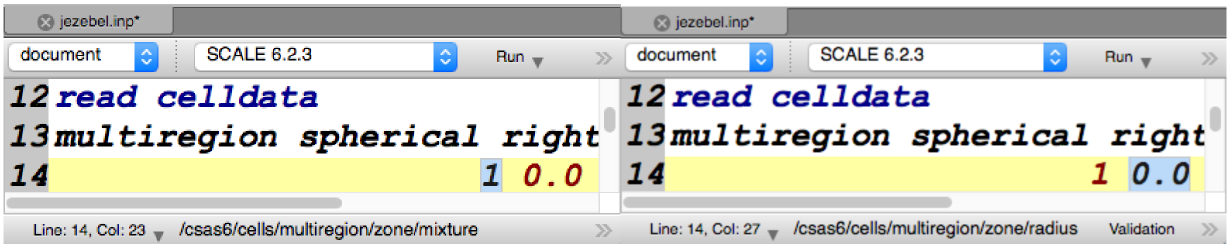
Figure 19. Autocompletion of a spherical multiregion.



```
read celldata
multiregion spherical right_bdy=vacuum end
                    1 0.0 2 0.0 3 0.0 end zone
```

Figure 20. Example autocompleted initial spherical multiregion input.

As a reminder, the text cursor can be used to identify which field is which, as illustrated in Figure 21, which depicts the multiregion zone **mixture** and **radius.**



```
read celldata
multiregion spherical right
                    1 0.0
```

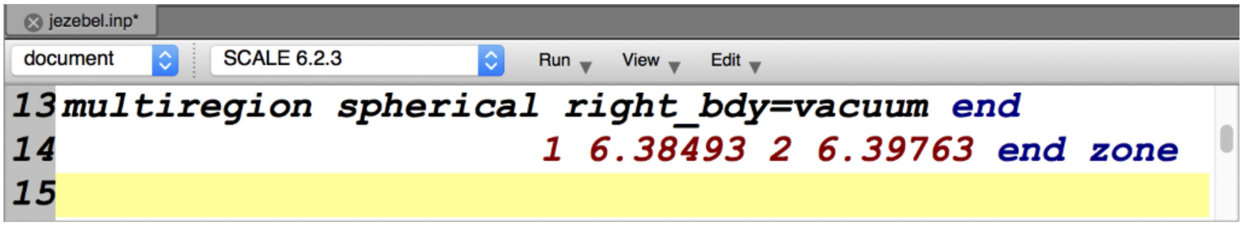
Line: 14, Col: 23 /csas6/cells/multiregion/zone/mixture

```
read celldata
multiregion spherical right
                    1 0.0
```

Line: 14, Col: 27 /csas6/cells/multiregion/zone/radius

Figure 21. Example spherical multiregion zone mixture and radius placeholders.

The zone mixture-radii pair placeholders need to be updated to be **mixture** `1` (^{239}Pu, ^{240}Pu, ^{241}Pu, and gallium) with a **radius** of `6.38493`, and **mixture** `2` (nickel) with a **radius** of `6.39763`. The placeholder for mixture 3 can be removed. Figure 22 illustrates the completed example **multiregion** input.

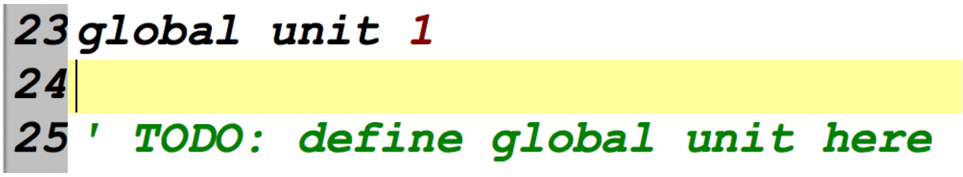


```
multiregion spherical right_bdy=vacuum end
                         1 6.38493 2 6.39763 end zone

```

Figure 22. Multiregion unit cell.

2.3.4 KENO-VI

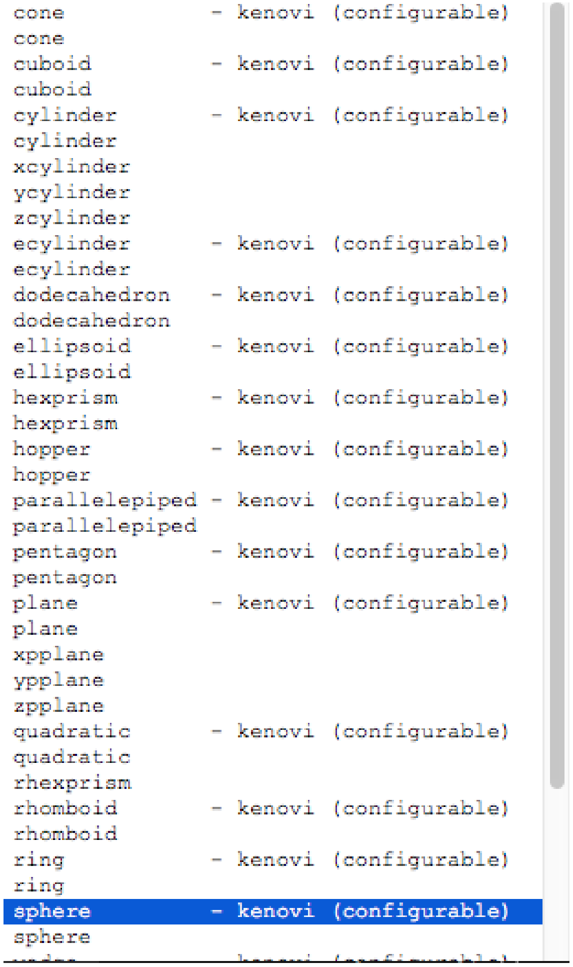
Now enter the information required to run a KENO-VI analysis of the problem. Place the cursor inside the global unit in the **read geometry** block as shown in Figure 23 to prepare for autocompletion of the appropriate geometry for this problem.



```
global unit 1

' TODO: define global unit here
```

Figure 23. Cursor placement in the global unit.

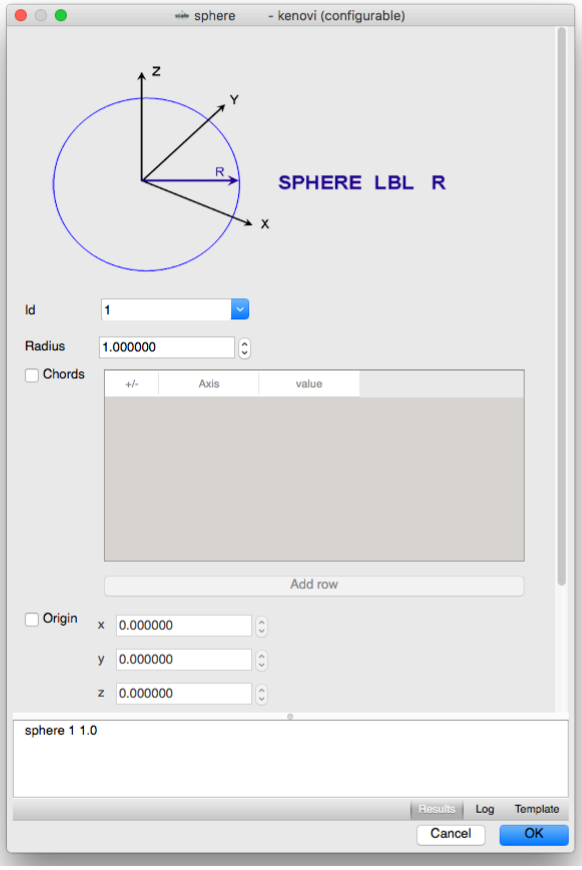
Press the `CTRL-SPACE` key combination to display the available geometry. For this problem, the appropriate geometry is a sphere, so select the **sphere – kenovi (configurable)** option shown in Figure 24.



```
cone           - kenovi (configurable)
cone
cuboid         - kenovi (configurable)
cuboid
cylinder       - kenovi (configurable)
cylinder
xcylinder
ycylinder
zcylinder
ecylinder      - kenovi (configurable)
ecylinder
dodecahedron   - kenovi (configurable)
dodecahedron
ellipsoid      - kenovi (configurable)
ellipsoid
hexprism       - kenovi (configurable)
hexprism
hopper         - kenovi (configurable)
hopper
parallelepiped - kenovi (configurable)
parallelepiped
pentagon       - kenovi (configurable)
pentagon
plane          - kenovi (configurable)
plane
xpplane
ypplane
zpplane
quadratic      - kenovi (configurable)
quadratic
rhexprism
rhomboid       - kenovi (configurable)
rhomboid
ring           - kenovi (configurable)
ring
sphere         - kenovi (configurable)
sphere
```

Figure 24. Available autocomplete options.

As shown in Figure 25, select the appropriate Id, `10`, enter the radius of `6.38493`, and then select **OK**. Now add an additional sphere for the nickel outer shell by copying the sphere and pasting it. Update the Id to `20` and the radius to `6.39763`, as shown in the highlighted portion of Figure 26.



Id 1

Radius 1.000000

□ Chords

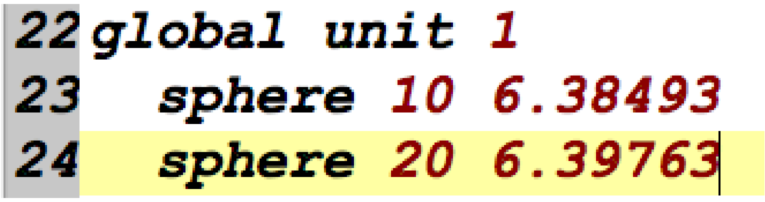
□ Origin x 0.000000

y 0.000000

z 0.000000

```
sphere 1 1.0
```

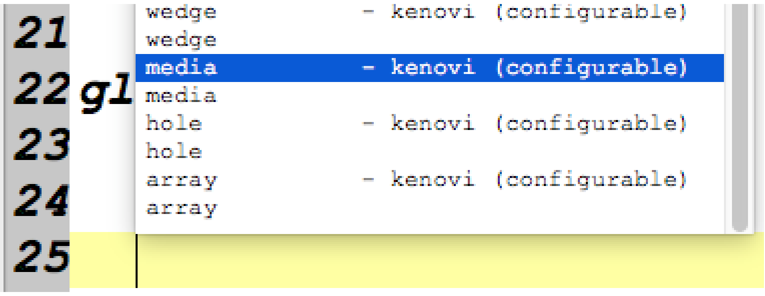
Figure 25. Initial sphere configurable.



```
22 global unit 1
23   sphere 10 6.38493
24   sphere 20 6.39763
```

Figure 26. Geometry regions.

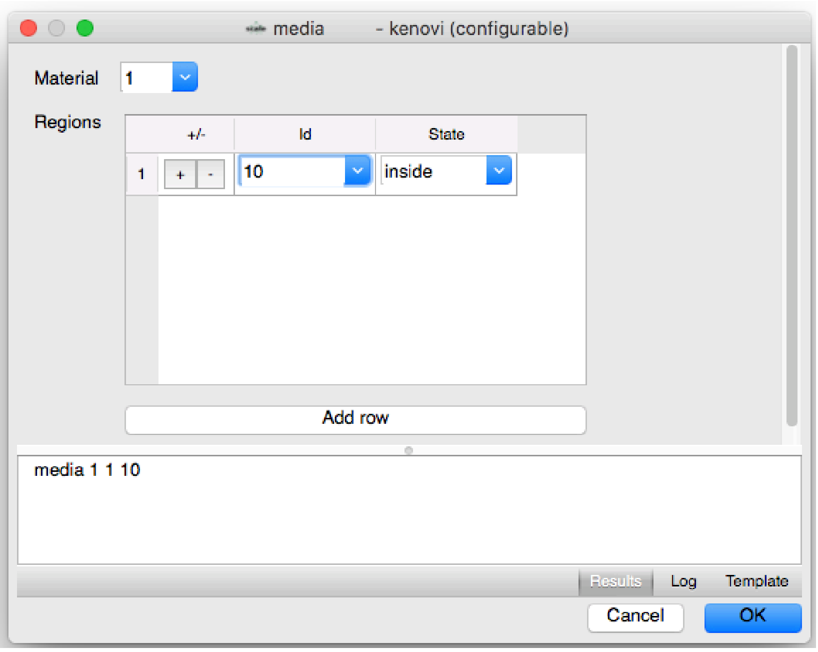
Next, describe the contents of the geometry regions. On line 25, press the `CTRL-SPACE` key combination to display available autocomplete options, and select **media – kenovi (configurable)**, as depicted in Figure 27.



```
21      wedge           - kenovi (configurable)
        wedge
22 gl   media           - kenovi (configurable)
        media
23      hole            - kenovi (configurable)
        hole
24      array           - kenovi (configurable)
        array
25
```

Figure 27. Configurable geometry media autocompletion.

Select the appropriate **Material**, `1`, for ^{239}Pu, ^{240}Pu, ^{241}Pu, Ga. Click the **Add row** button to add an entry to the **Regions** table. Specify the inner sphere id of `10` in the **Id** cell, and ensure the **State** field lists `inside`, as depicted in Figure 28.



media - kenovi (configurable)

Material: 1

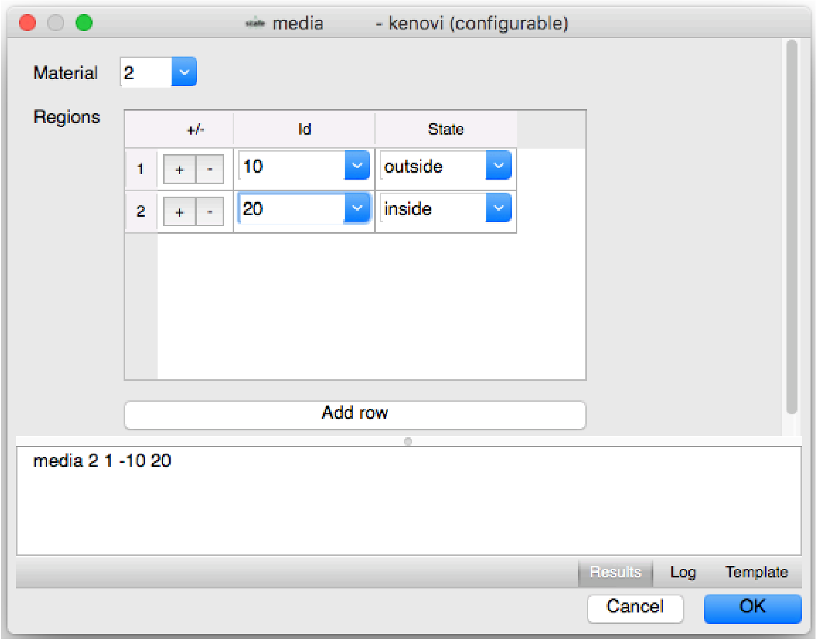
Regions

	+/-	Id	State
1	+ -	10	inside

```
media 1 1 10
```

Figure 28. Plutonium sphere media region definition.

To describe the nickel plating outside the plutonium core, perform an additional autocompletion of **media – kenovi (configurable)** on line 26. Select the appropriate **Material**, `2`, for nickel. Click the **Add row** button to add an entry to the **Regions** table. Specify the inner sphere id of `10` in the **Id** cell and ensure that the **State** field lists `outside`. Click the **Add row** button again to add a second entry to the **Regions** table. In this second row, specify the outer sphere id of `20` in the **Id** cell and ensure that the **State** field lists `inside`, as depicted in Figure 29.



Material: 2

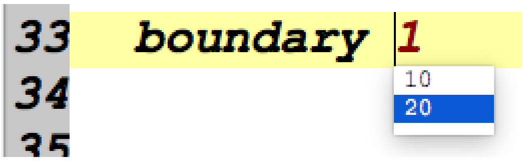
Regions

	+/-	Id	State
1	+ -	10	outside
2	+ -	20	inside

```
media 2 1 -10 20
```

Figure 29. Nickel plating media region definition vector.

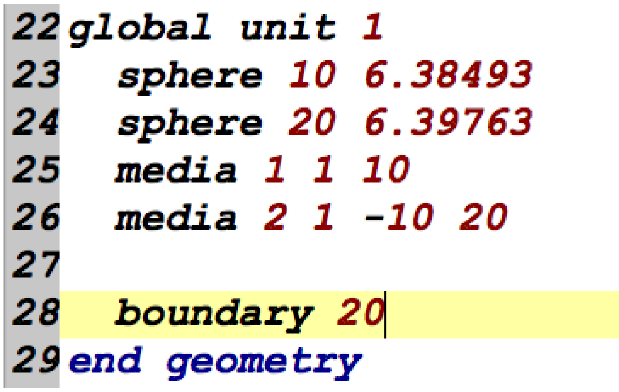
Lastly, the outer region boundary of the unit must be defined. Perform an autocompletion on the default **boundary** region identifier of 1 to display the list of available regions and select the outer sphere, id `20`, as depicted in Figure 30.



```
  boundary 1
```

Figure 30. Global unit outer boundary.

The **global unit** is the unit that defines the entire system to be analyzed and must always be specified. The global unit should look like Figure 31. Note that placeholder comments have been removed in the figure for the purpose of clarity.



```
global unit 1
  sphere 10 6.38493
  sphere 20 6.39763
  media 1 1 10
  media 2 1 -10 20

  boundary 20
end geometry
```

Figure 31. Geometry definition.

This completes the information necessary to run the sample problem. The next step is to save the input file and run SCALE. Make sure the file is saved by clicking the **File > Save.**

2.4 RUNNING SCALE/KENO-VI

To run SCALE/KENO-VI, click the **Run** button on the text editor panel's toolbar depicted in Figure 32.

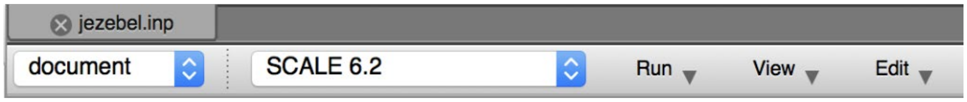
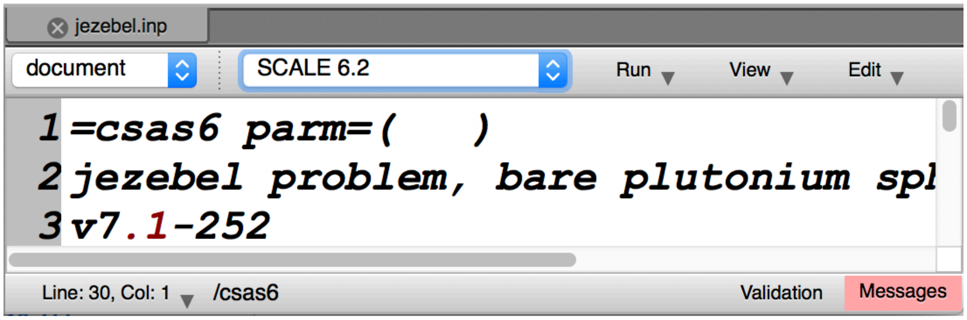


Figure 32. Text editor toolbar.

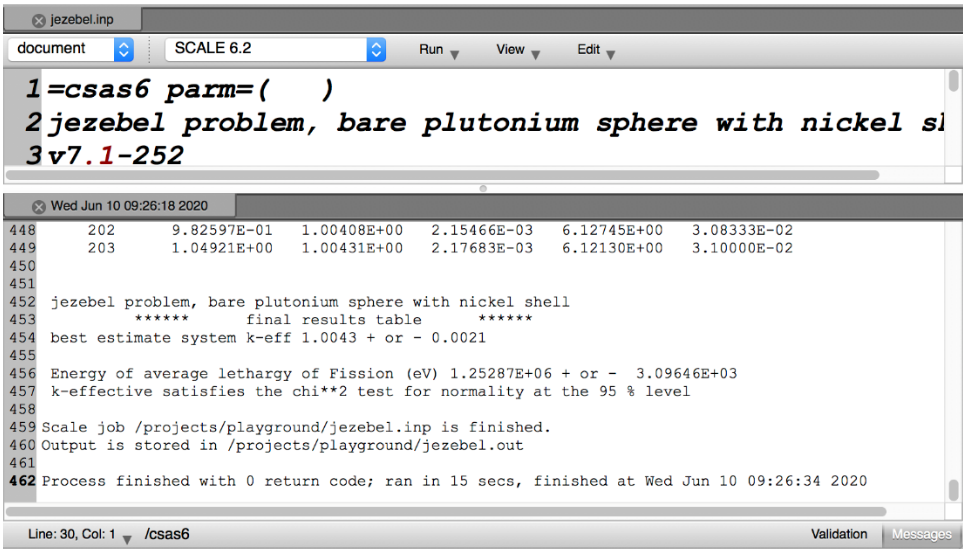
Upon clicking the **Run** button, the **Messages** panel button background color will change to red, indicating that there are unseen messages. The **Messages** panel is located in the bottom right corner of the text editor, as depicted in Figure 33.



```
=csas6 parm=(    )
jezebel problem, bare plutonium spl
v7.1-252
```

Figure 33. Location of Messages panel and red background color indicating unseen messages.

Click the **Messages** panel to see the jezebel.inp execution messages, as depicted in Figure 34.



```
=csas6 parm=(    )
jezebel problem, bare plutonium sphere with nickel sl
v7.1-252
```

```
      202       9.82597E-01   1.00408E+00   2.15466E-03   6.12745E+00   3.08333E-02
      203       1.04921E+00   1.00431E+00   2.17683E-03   6.12130E+00   3.10000E-02


 jezebel problem, bare plutonium sphere with nickel shell
          ******      final results table      ******
 best estimate system k-eff 1.0043 + or - 0.0021

 Energy of average lethargy of Fission (eV) 1.25287E+06 + or -  3.09646E+03
 k-effective satisfies the chi**2 test for normality at the 95 % level

Scale job /projects/playground/jezebel.inp is finished.
Output is stored in /projects/playground/jezebel.out

Process finished with 0 return code; ran in 15 secs, finished at Wed Jun 10 09:26:34 2020
```

Figure 34. Messages panel showing completed run.

Note that the messages provide a running summary of results and should indicate that the **SCALE job is finished.** The messages also indicate that the **Output is stored in jezebel.out**. If you had a different name for your input file, then the output will be stored in a file with the **.out** extension. The elapsed time may not match, as each computer has a different central processing unit (CPU) and disk access speeds.

2.5 SCALE/KENO-VI OUTPUT

To view the output, **right-click** the **jezebel.inp** file in the **Navigation** panel, and select **Open Associated Files > jezebel.out**, as depicted in Figure 35.

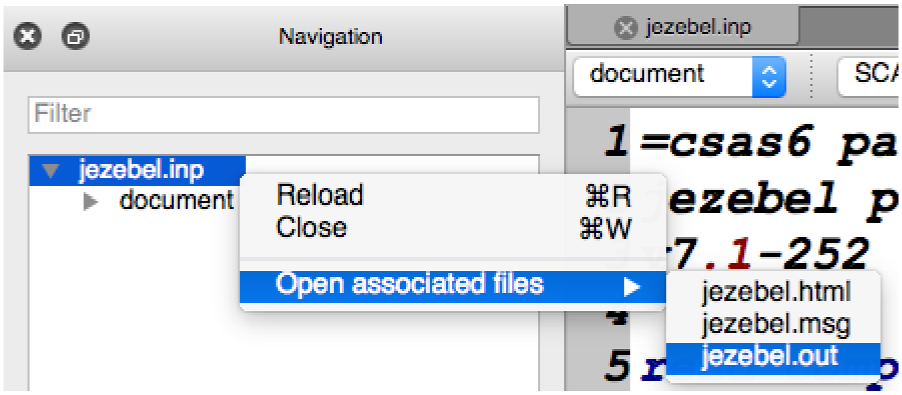
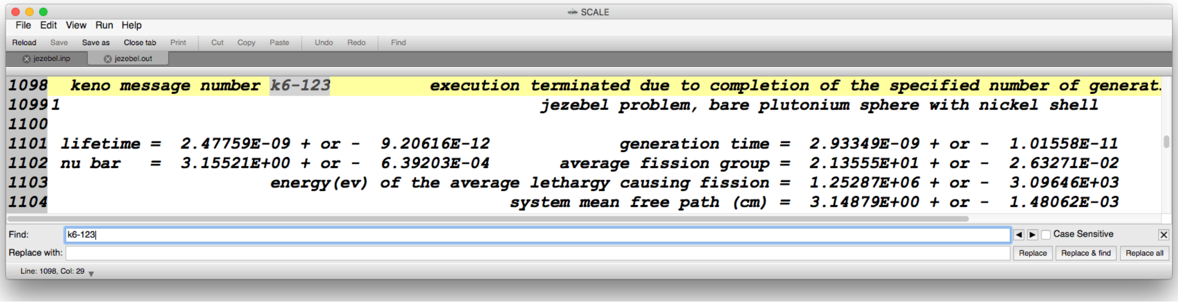


Figure 35. Opening the output file.

The output for this problem consists of the following sections:

- Echo of input
- CSAS information and brief review of input values, cross section processing
- KENO information

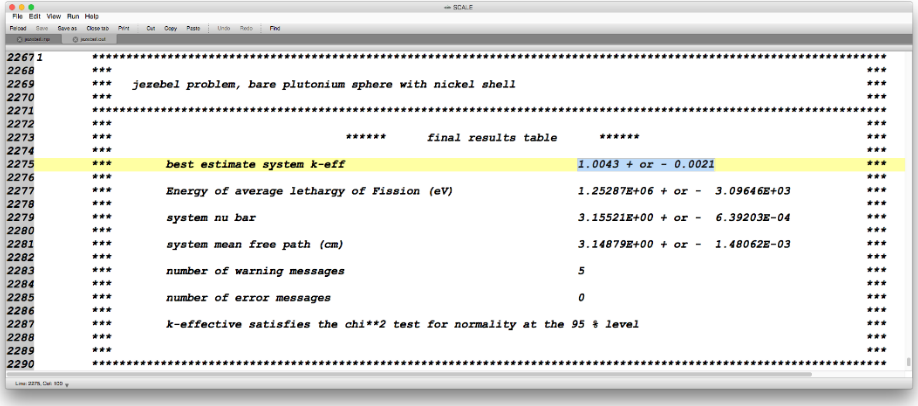
This *Quickstart* section presents a few items in the KENO information section. The first is the information following the line **keno messages number k6-123.** Click the **Find** button on the Fulcrum toolbar and enter `k6-123`. This message states that execution of KENO was terminated due to completion of the specified number of generations. This means that the problem ran to completion based on the generations specified by the user (in this case the default value). The lines below the messages provide information on the neutron lifetime, generation time, nu-bar, average fission group, and energy of the average lethargy causing fission. This information should appear as shown in Figure 36.



```
 keno message number k6-123          execution terminated due to completion of the specified number of generat.
1                                                 jezebel problem, bare plutonium sphere with nickel shell

 lifetime =  2.47759E-09 + or -  9.20616E-12            generation time =  2.93349E-09 + or -  1.01558E-11
 nu bar   =  3.15521E+00 + or -  6.39203E-04         average fission group =  2.13555E+01 + or -  2.63271E-02
                        energy(ev) of the average lethargy causing fission =  1.25287E+06 + or -  3.09646E+03
                                                  system mean free path (cm) =  3.14879E+00 + or -  1.48062E-03
```

Figure 36. KENO output for Jezebel problem.

Following the neutron parameters table is a table providing the calculated k_{eff} of the problem vs. the number of initial generations skipped. The final answer for k_{eff} should be taken from the final results table at the end of the output; here the k_{eff} is labeled as the “best estimate system k-eff.” For the example problem, the calculated k_{eff} is 1.0043 ± 0.0021, as shown in Figure 37.



```
1       *************************************************************************************************************
        ***                                                                                                       ***
        ***   jezebel problem, bare plutonium sphere with nickel shell                                            ***
        ***                                                                                                       ***
        *************************************************************************************************************
        ***                                                                                                       ***
        ***                                   ******      final results table      ******                         ***
        ***                                                                                                       ***
        ***        best estimate system k-eff                           1.0043 + or - 0.0021                      ***
        ***                                                                                                       ***
        ***        Energy of average lethargy of Fission (eV)           1.25287E+06 + or -  3.09646E+03           ***
        ***                                                                                                       ***
        ***        system nu bar                                        3.15521E+00 + or -  6.39203E-04           ***
        ***                                                                                                       ***
        ***        system mean free path (cm)                           3.14879E+00 + or -  1.48062E-03           ***
        ***                                                                                                       ***
        ***        number of warning messages                           5                                         ***
        ***                                                                                                       ***
        ***        number of error messages                             0                                         ***
        ***                                                                                                       ***
        ***        k-effective satisfies the chi**2 test for normality at the 95 % level                          ***
        ***                                                                                                       ***
        ***                                                                                                       ***
        *************************************************************************************************************
```

Figure 37. KENO final results table.

2.6 SUMMARY

This section has helped you to:

- describe the structure of SCALE/KENO-VI input files and know that there are parts describing the sequence, the materials, and the KENO information;
- use the Fulcrum user interface to create a SCALE/KENO-VI input file;
- set up and run a simple criticality problem using SCALE/KENO-VI; and
- find and interpret k_{eff} information from your output.

Now that you have successfully run SCALE/KENO-VI, you are ready to learn in detail the options available in each input segment and how to set up more complex problems. The sections that follow present these details in a format similar to that used in this *Quickstart* section.

3. MATERIAL INFORMATION INPUT

The *Quickstart* section (Sect. 2), focused on a simple problem with SCALE using the Fulcrum User Interface. From this problem, you gained confidence in using the code and some experience with Fulcrum. This section and subsequent sections provide a more detailed explanation of the commands used in the *Quickstart* section.

3.1 WHAT YOU WILL BE ABLE TO DO

- Define the different criticality sequences used in SCALE.
- Describe the cross section libraries available for criticality safety analyses.
- Use the Fulcrum user interface to provide data on elements, isotopes, compounds, and mixtures of these.
- Interpret basic output information from a SCALE/KENO analysis.

To minimize human error, the SCALE data handling and program flow are automated as much as possible through the use of control modules. These control modules are incorporated into sequences that select the modules required for a particular analysis. For criticality safety work, these CSAS sequences provide automated, problem-dependent, cross section processing, if necessary, for MG calculations, followed by calculation of the neutron multiplication factor for the system being modeled. These control sequences use the cross section processing code CENTRM/PMC to provide resonance-corrected cross sections.

KENO uses the either CE or processed MG cross sections and calculates the k_{eff} of 3D system models. The geometric modeling capabilities available in KENO, coupled with the automated cross section processing within the control sequences, allow complex 3D systems to be easily analyzed. The sequences used most often for criticality safety calculations are CSAS5 for KENO V.a calculations and CSAS6 for KENO-VI calculations. This document guides users through the CSAS6 sequence, and [4] is available for CSAS5 users.

3.2 CROSS SECTION LIBRARIES

The CSAS sequences (1) use the Standard Composition Library (described later in this section) for specifying the materials and mixtures used in a calculation and (2) provide automatic, problem-dependent cross section preparation prior to the criticality calculation for MG calculations (not necessary for CE calculations). This section describes the cross section libraries most useful for criticality calculations in the SCALE system.

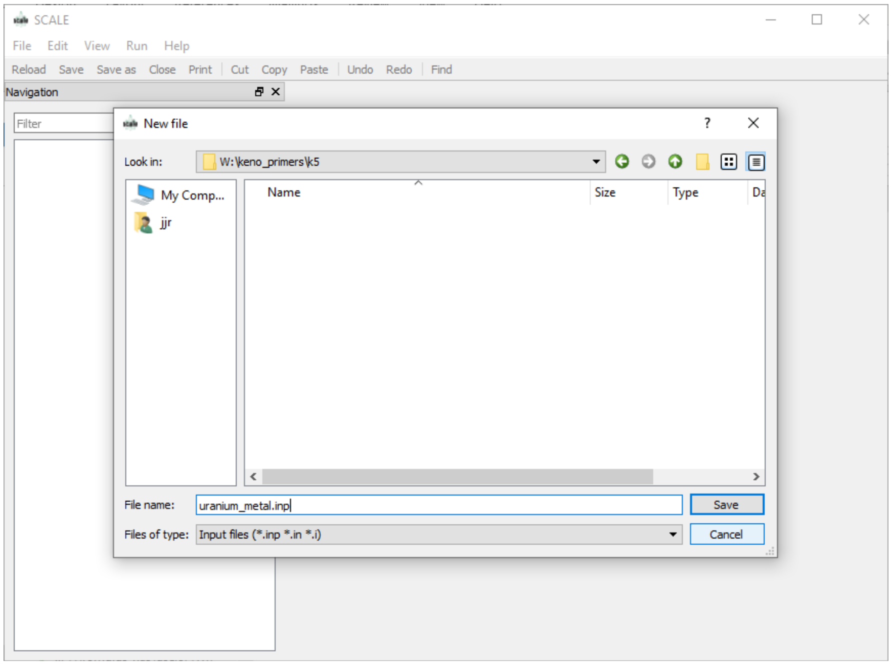
As of the SCALE 6.2.4 release, there are four cross section libraries that are primarily intended for use in criticality safety calculations with SCALE: the (1) CE library based on ENDF/B-VII, (2) the CE library based on ENDF/B-VII.1 (3) the 238-group ENDF/B-VII MG library, and (4) the 252-group ENDF/B-VII.1 MG library. There are other cross section libraries available for use in SCALE, but they are not recommended for criticality safety calculations.

When performing calculations with SCALE, the cross section libraries must be referenced using mnemonic identifiers. Each library has multiple acceptable mnemonic names; the names for the criticality safety cross section libraries are listed in Table 1.

Table 1. SCALE cross section libraries for criticality safety calculations

Mnemonic names	Library
v7-238, v7-238n, v7.0-238n	ENDF/B-VII.0 238-group neutron library
v7-252, v7-252n, v7.1-252n	ENDF/B-VII.1 252-group neutron library
ce_v7, ce_v7_endf, ce_v7.0_endf	ENDF/B-VII.0 CE library
ce_v7.1, ce_v7.1_endf	ENDF/B-VII.1 CE library

To perform a KENO calculation with SCALE using the Fulcrum user interface, the user must first generate an input file with an extension that is either .i, .in, or .inp. This is done by clicking the **File > New File...** button on the upper-left of the screen. Fulcrum will display the **New file** dialog window. To begin the first material exercise, enter the input file name as `uranium_metal.inp`, and click the **Save** button. The entry of the file name is shown in Figure 38 and will result in the blank input file shown in Figure 39.



Look in: W:\keno_primers\k5

File name: uranium_metal.inp

Files of type: Input files (*.inp *.in *.i)

Figure 38. Depiction of the input file name definition.

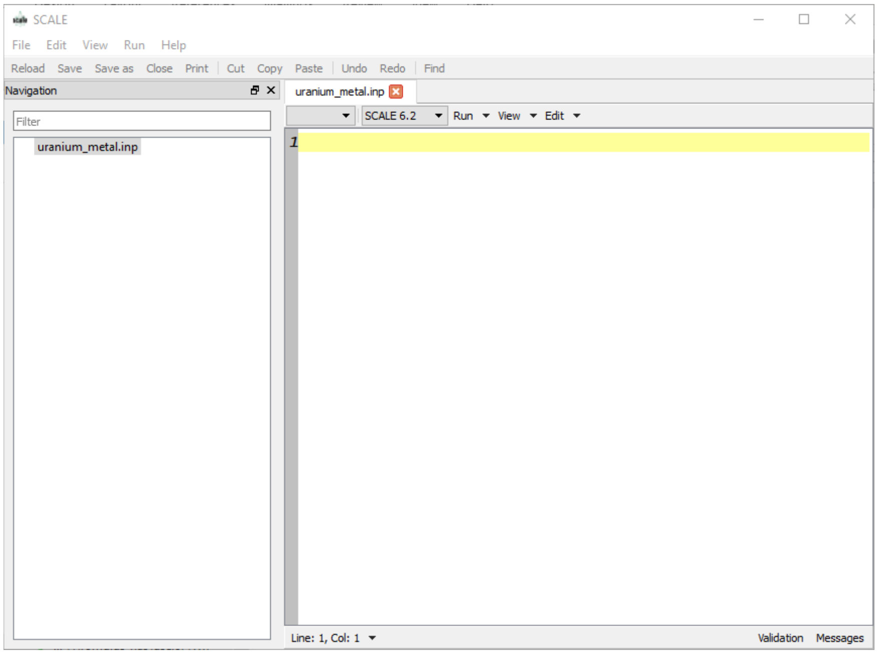
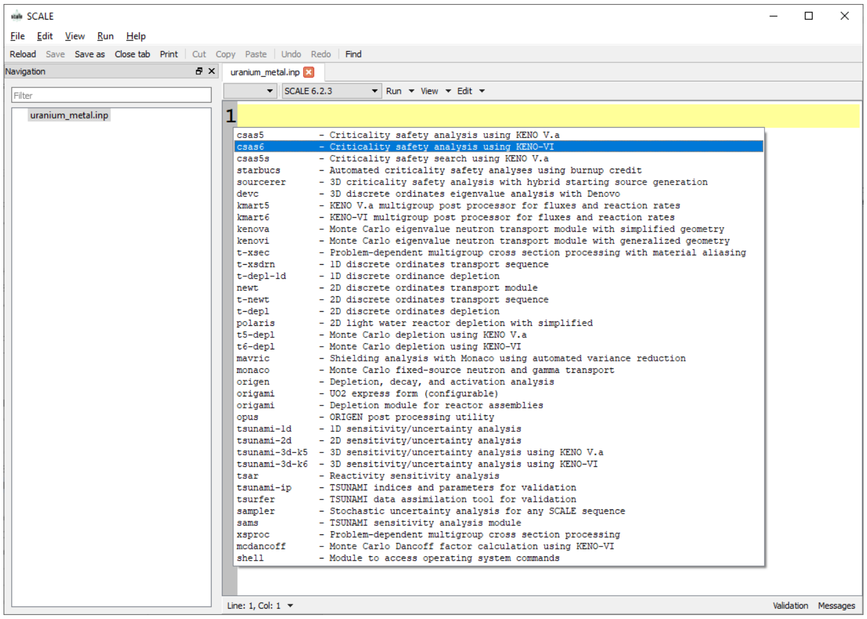


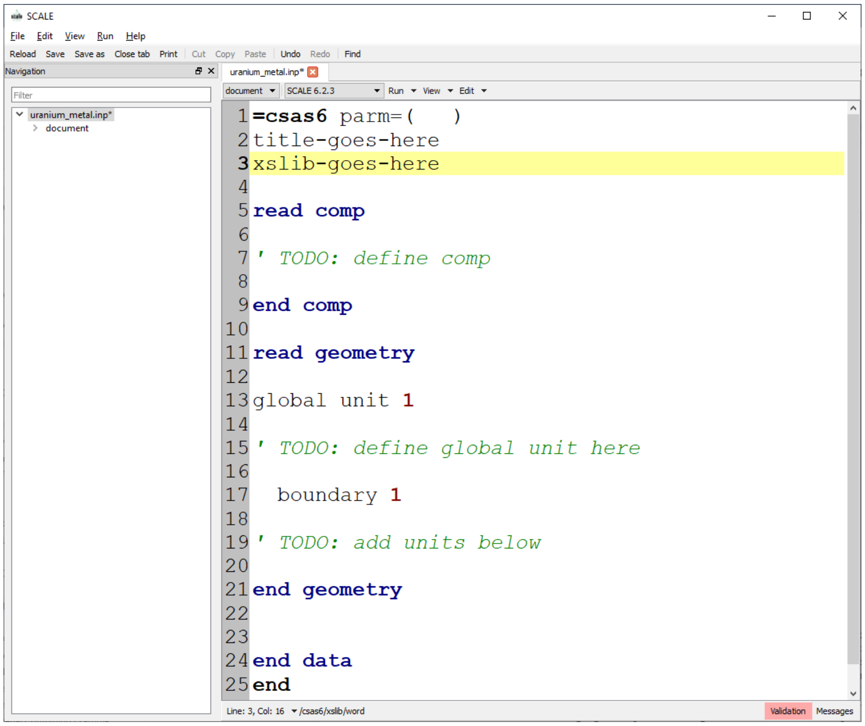
Figure 39. Empty input file after file specification in Fulcrum.

Next, the appropriate sequence must be selected. This is done by pressing `CTRL-SPACE` inside the input file, which will result in a dropdown menu with all SCALE sequences displayed. Then select the CSAS6 sequence from the dropdown menu and press **enter**, as shown in Figure 40. Selecting a sequence will result in a CSAS6 input skeleton, as is shown in Figure 41.



```
csas5          - Criticality safety analysis using KENO V.a
csas6          - Criticality safety analysis using KENO-VI
csas5s         - Criticality safety search using KENO V.a
starbucs       - Automated criticality safety analyses using burnup credit
sourcerer      - 3D criticality safety analysis with hybrid starting source generation
devc           - 3D discrete ordinates eigenvalue analysis with Denovo
kmart5         - KENO V.a multigroup post processor for fluxes and reaction rates
kmart6         - KENO-VI multigroup post processor for fluxes and reaction rates
kenova         - Monte Carlo eigenvalue neutron transport module with simplified geometry
kenovi         - Monte Carlo eigenvalue neutron transport module with generalized geometry
t-xsec         - Problem-dependent multigroup cross section processing with material aliasing
t-xsdrn        - 1D discrete ordinates transport sequence
t-depl-1d      - 1D discrete ordinance depletion
newt           - 2D discrete ordinates transport module
t-newt         - 2D discrete ordinates transport sequence
t-depl         - 2D discrete ordinates depletion
polaris        - 2D light water reactor depletion with simplified
t5-depl        - Monte Carlo depletion using KENO V.a
t6-depl        - Monte Carlo depletion using KENO-VI
mavric         - Shielding analysis with Monaco using automated variance reduction
monaco         - Monte Carlo fixed-source neutron and gamma transport
origen         - Depletion, decay, and activation analysis
origami        - UO2 express form (configurable)
origami        - Depletion module for reactor assemblies
opus           - ORIGEN post processing utility
tsunami-1d     - 1D sensitivity/uncertainty analysis
tsunami-2d     - 2D sensitivity/uncertainty analysis
tsunami-3d-k5  - 3D sensitivity/uncertainty analysis using KENO V.a
tsunami-3d-k6  - 3D sensitivity/uncertainty analysis using KENO-VI
tsar           - Reactivity sensitivity analysis
tsunami-ip     - TSUNAMI indices and parameters for validation
tsurfer        - TSUNAMI data assimilation tool for validation
sampler        - Stochastic uncertainty analysis for any SCALE sequence
sams           - TSUNAMI sensitivity analysis module
xsproc         - Problem-dependent multigroup cross section processing
mcdancoff      - Monte Carlo Dancoff factor calculation using KENO-VI
shell          - Module to access operating system commands
```

Figure 40. Dropdown menu with the SCALE sequence specification.



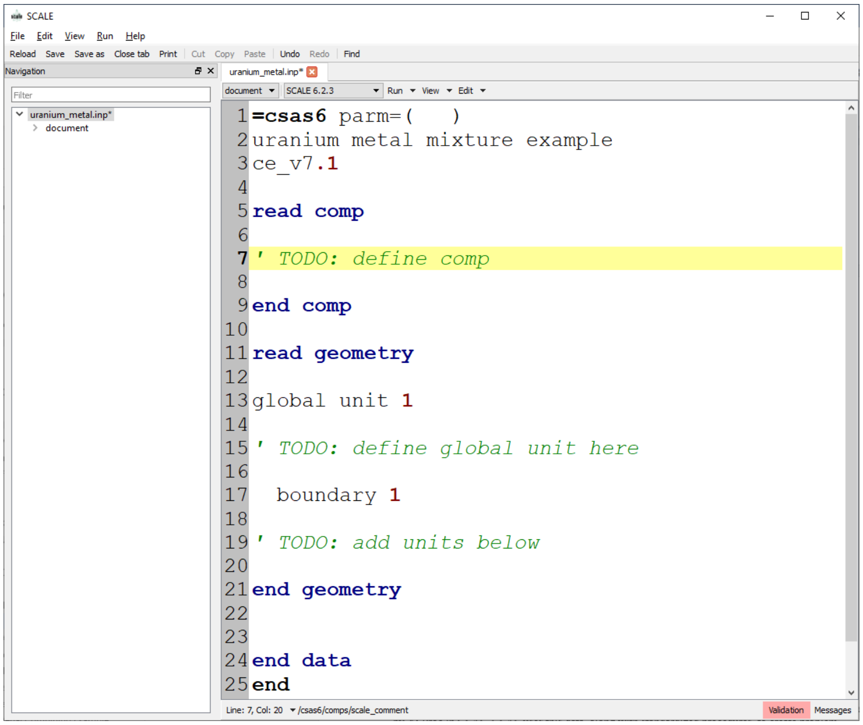
```
 1 =csas6 parm=(    )
 2 title-goes-here
 3 xslib-goes-here
 4
 5 read comp
 6
 7 ' TODO: define comp
 8
 9 end comp
10
11 read geometry
12
13 global unit 1
14
15 ' TODO: define global unit here
16
17   boundary 1
18
19 ' TODO: add units below
20
21 end geometry
22
23
24 end data
25 end
```

Figure 41. Empty input file resulting from the CSAS6 sequence specification.

Once the empty skeleton is produced, two input placeholders are displayed at the top of the input file on lines 2 and 3. The title of the input is always listed on line 2, which is used to label tables throughout the output. For this example, replace **title-goes-here** with `uranium metal mixture example`. The third line is always the cross section library to be used in the calculation. Since cross section processing has not been discussed, select the CE version of the ENDF/B-VII.1 library by replacing **xslib-goes-here** with `ce_v7.1` on the third line of input. These modifications to the input file are shown in Figure 42. It can also be seen that the validation tab in the lower right hand portion of the window is highlighted in pink, indicating that there is an error in the input. Clicking on the validation tab opens a pane at the bottom of Fulcrum with the following error message:

```
line:5 column:1 - Validation Error: comps has zero of: [ stdcomp
wtptcomp atomcomp arbmcomp soln solution ] - at least one must occur
```

This error message indicates that there are no entries in the composition block. The error message is also shown in Figure 42.



```
=csas6 parm=(   )
uranium metal mixture example
ce_v7.1

read comp

' TODO: define comp

end comp

read geometry

global unit 1

' TODO: define global unit here

  boundary 1

' TODO: add units below

end geometry


end data
end
```

Figure 42. Specification of problem title and cross section library.

3.3 MATERIAL INPUT

Information on materials in a system is entered into Fulcrum, which formats the information as required by XSProc in CSAS. CSAS uses this data, along with standardized procedures, to create problem-dependent cross section libraries. Input data to Fulcrum identify the materials from the Standard Composition Library and associated physical densities to calculate the number densities (atoms/b-cm) of each material specified in the problem. The number densities are used to develop data used by the downstream codes. These input data include (1) the Standard Composition data used in the standardized number density calculations (a standardized alphanumeric name, mixture number, and other data to define materials, including volume fraction or percent theoretical density, temperature, and isotopic distribution), and (2) the unit cell description defining the materials, dimensions, and boundary conditions of the geometry to be used in the Dancoff factor calculations, the resonance self-shielding calculations, and the flux-weighting cell calculations necessary for cross section processing.

The Standard Composition Library describes the various predefined isotopes, elements (both symbols and full names), compounds, alloys, and other materials that can be used to define the material mixtures for a given problem. A complete description of the materials in the library is found in the SCALE manual [see *Standard Composition Library*]. The library contains over 600 compounds, alloys, elements, and isotopes that can be used to define the material mixtures for a given problem. Additionally, six fissile solutions are available for which the user can specify the heavy metal, acid, and water components: $UO_2(NO_3)_2$, UO_2F_2, $Pu(NO_3)_4$, PuF_4, $Th(NO_3)_4$ and ThF_4.

When formulating a mixture, it is often necessary to know the density (g/cc) of the mixture and the weight fractions of the various constituent materials. Note that default densities should not be used for materials containing enriched isotopes, especially light elements with strong absorbers such as boron, B_4C, or lithium. The temperature of a given material can be entered (the default is 293 K, which is room temperature). The temperature is used to correctly process resonance data, Bondarenko data, and/or thermal-scattering data. The temperature of a mixture is always specified in Kelvin (K) in SCALE.

3.4 EXAMPLE MATERIAL SPECIFICATIONS

Five example problems are used here to illustrate the method for entering material information into Fulcrum. These problems demonstrate composition entries for pure ^{235}U metal, U(93.7) metal, U(93.7)O_2, U(30.3)O_2F_2 solution, and a mixture of UO_2 and water.

3.4.1 ^{235}U Metal Composition Input

The first material will be used to model a bare metal cylinder in Section 4.2.1. This section also uses the input generated in Section 3.2 as a starting point.

Pure ^{235}U is modeled first to demonstrate the use of Fulcrum to enter a single isotope. The density and material composition of the material is shown below.

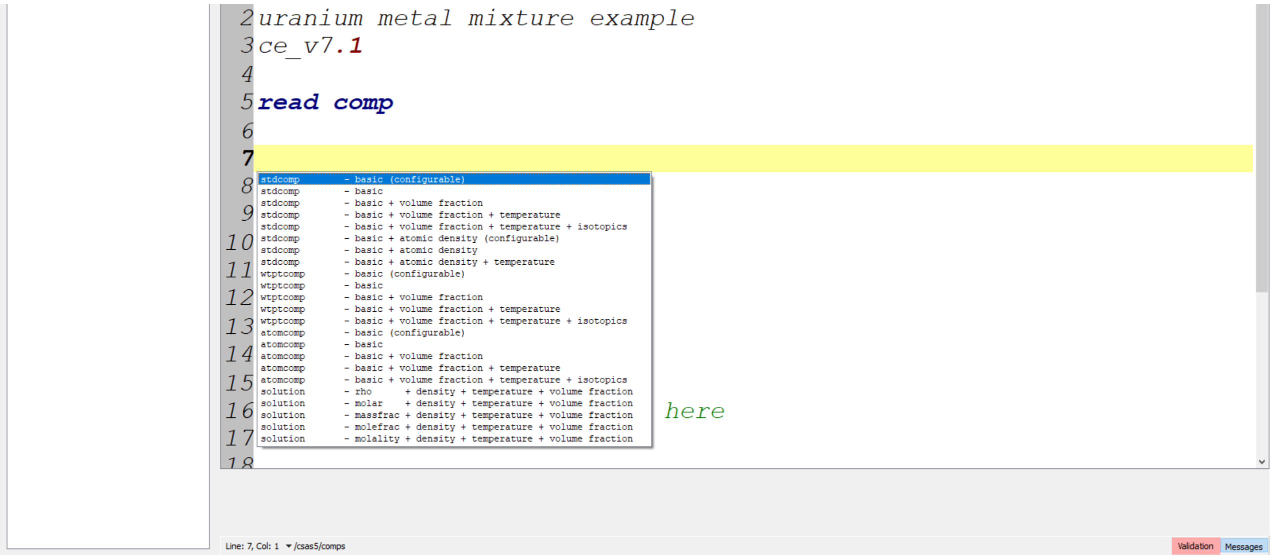
Core Material (ρ_{mix} = 18.742 g/cc-mixture)

U metal (100 percent ^{235}U)

Temperature of 300 K

3.4.1.1 Material input

To define the uranium metal composition. Move the cursor into the **read comp** block and press `CTRL-SPACE` to generate the dropdown menu shown in Figure 43.

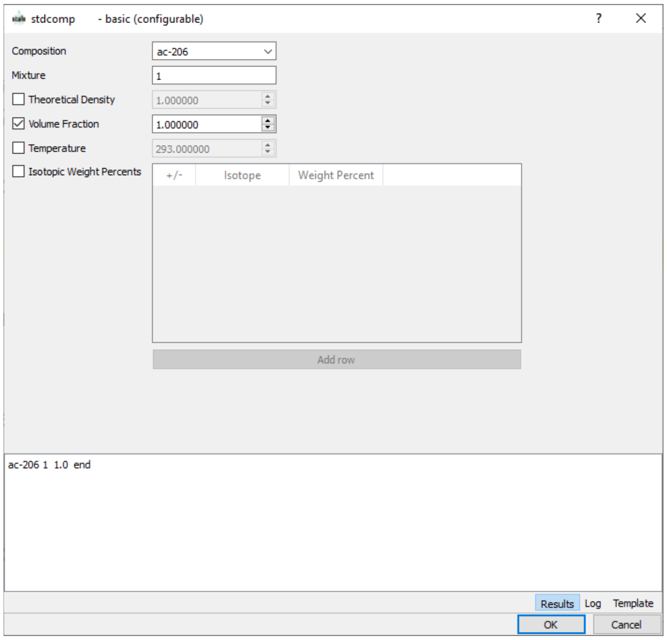


```
uranium metal mixture example
ce_v7.1

read comp
```

stdcomp - basic (configurable)
stdcomp - basic
stdcomp - basic + volume fraction
stdcomp - basic + volume fraction + temperature
stdcomp - basic + volume fraction + temperature + isotopics
stdcomp - basic + atomic density (configurable)
stdcomp - basic + atomic density
stdcomp - basic + atomic density + temperature
wtptcomp - basic (configurable)
wtptcomp - basic
wtptcomp - basic + volume fraction
wtptcomp - basic + volume fraction + temperature
wtptcomp - basic + volume fraction + temperature + isotopics
atomcomp - basic (configurable)
atomcomp - basic
atomcomp - basic + volume fraction
atomcomp - basic + volume fraction + temperature
atomcomp - basic + volume fraction + temperature + isotopics
solution - rho + density + temperature + volume fraction
solution - molar + density + temperature + volume fraction
solution - massfrac + density + temperature + volume fraction
solution - molefrac + density + temperature + volume fraction
solution - molality + density + temperature + volume fraction
here
Line: 7, Col: 1
/csas5/comps
Validation
Messages

Figure 43. Accessing the basic standard composition configurable.

From the dropdown menu, select **stdcomp – basic (configurable)**. This will generate a fillable form, as shown in Figure 44.


stdcomp - basic (configurable)

Composition: ac-206

Mixture: 1

□ Theoretical Density: 1.000000

☑ Volume Fraction: 1.000000

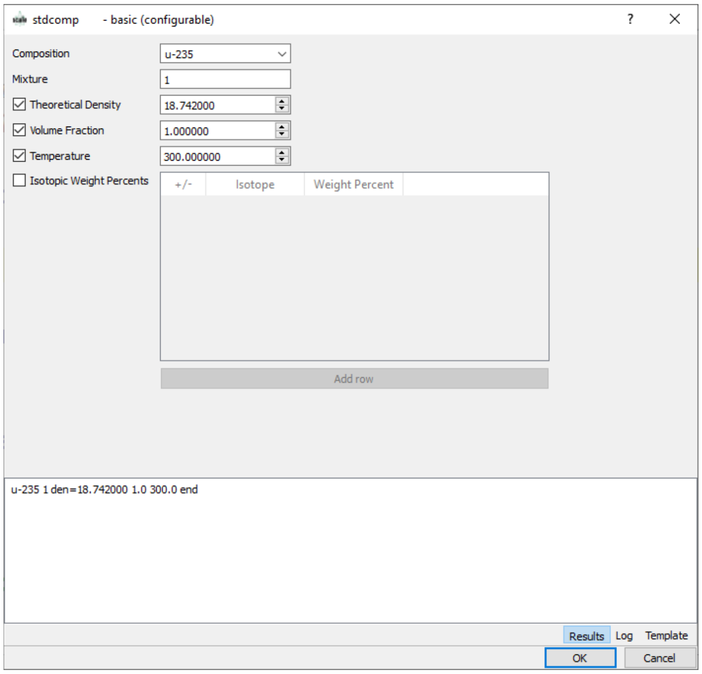
□ Temperature: 293.000000

□ Isotopic Weight Percents

```
ac-206 1 1.0 end
```

Figure 44. Basic standard composition configurable input initial display.

In the **composition** dropdown, clicking on the field that contains **ac-206** and type `u-235` over it. Then click the **Theoretical Density** checkbox and entering the value of `18.742` in place of the default value of `1.0`. Because the material is pure ^{235}U, and because we have already specified the density, the default **Volume Fraction** of `1.0` is appropriate. Modify the temperature by clicking the **Temperature** box and entering a value of `300`. The final configurable window is shown in Figure 45.


stdcomp - basic (configurable)

Composition: u-235

Mixture: 1

☑ Theoretical Density: 18.742000

☑ Volume Fraction: 1.000000

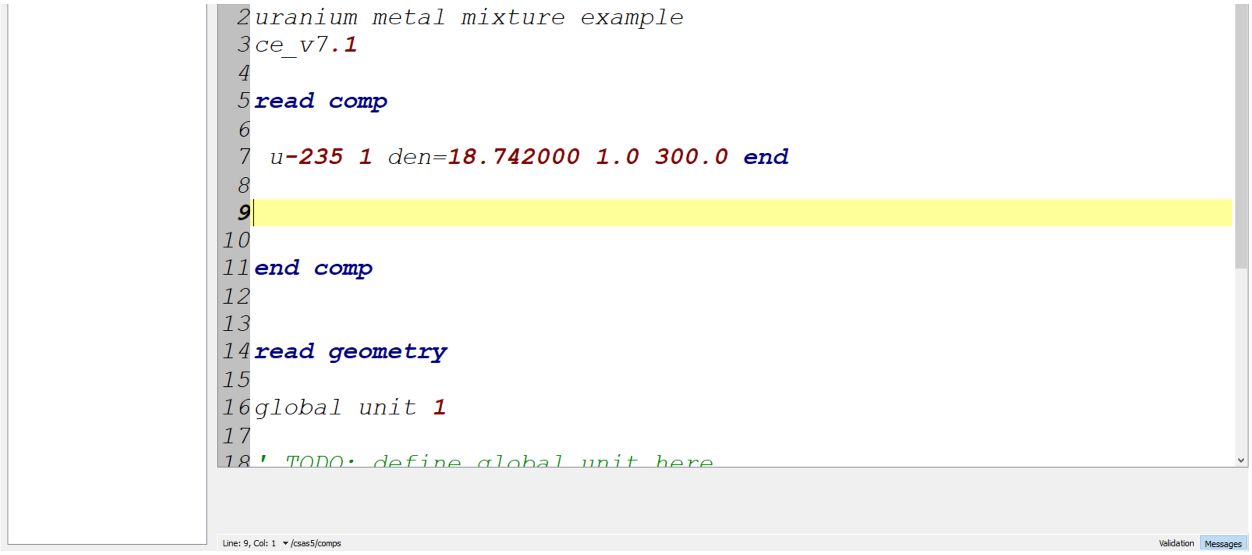
☑ Temperature: 300.000000

□ Isotopic Weight Percents

```
u-235 1 den=18.742000 1.0 300.0 end
```

Figure 45. Basic standard composition configurable input, initial display filled out.

Confirm that the data entry is correct before clicking **OK**, as the data cannot be modified in the configurable form after it has been entered into the input. Once you have clicked **OK** at the bottom of the configurable form, the input file should look like that shown in Figure 46.

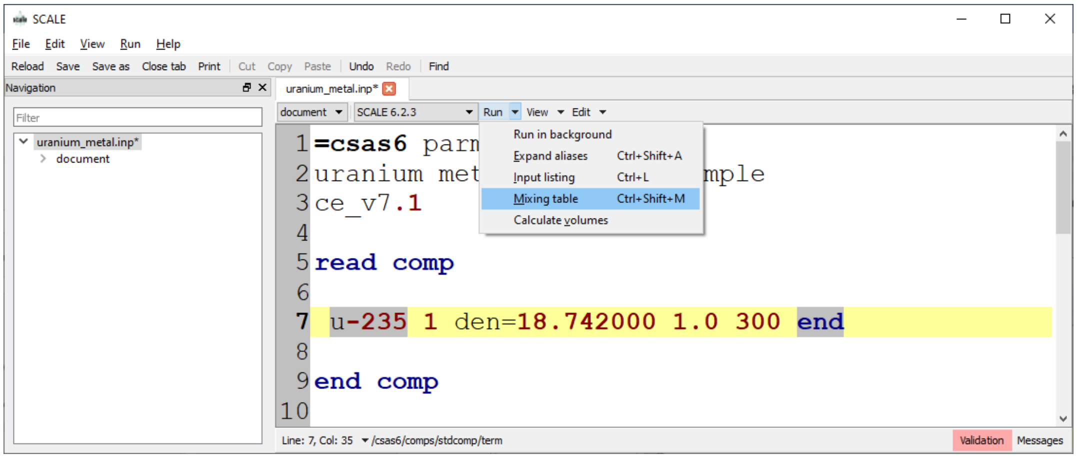


```
uranium metal mixture example
ce_v7.1

read comp

 u-235 1 den=18.742000 1.0 300.0 end



end comp


read geometry

global unit 1

' TODO: define global unit here
```

Figure 46. Input result of the standard basic composition configurable input.

3.4.1.2 Mixing table output

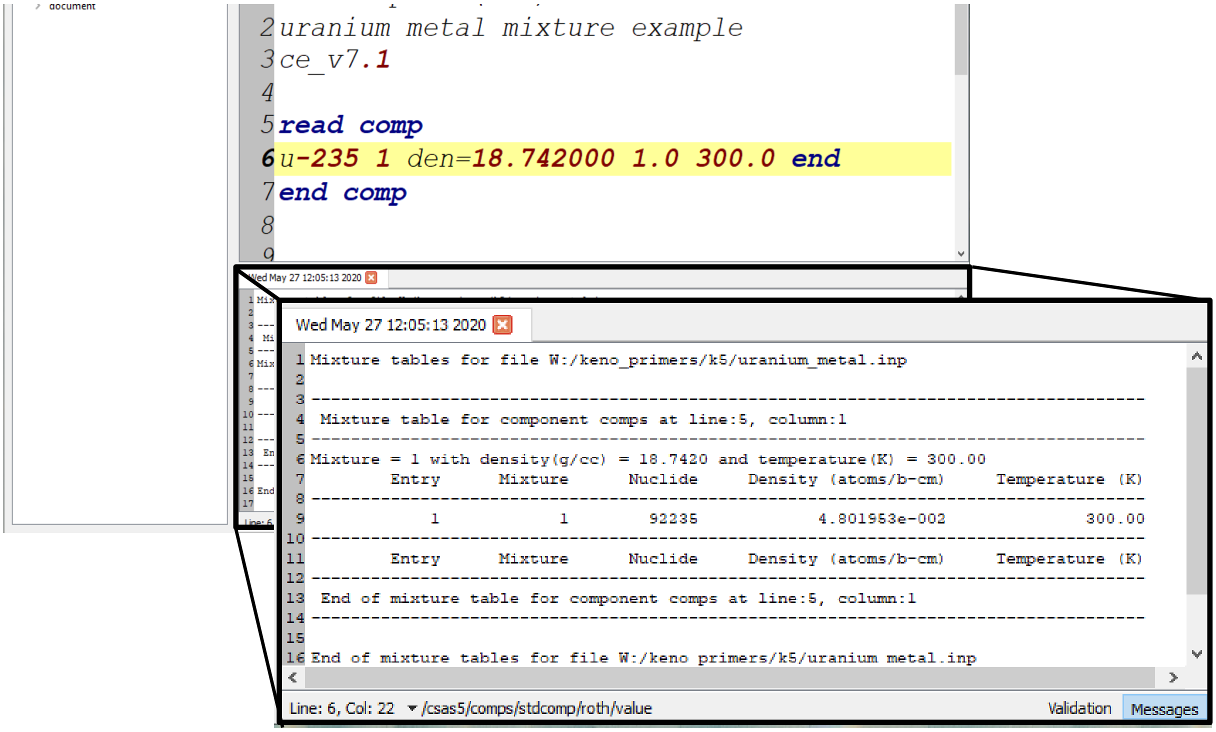
At this point there is not enough input specified to perform a KENO-VI calculation, but the atom densities can be generated based on the standard composition input supplied. To do this, generate a MIXING TABLE with CSAS by clicking the downward pointing triangle next to the **Run** button at the top of the Fulcrum interface and selecting **Mixing table** from the dropdown menu. A picture of the dropdown menu is provided in Figure 47.



```
=csas6 parm
uranium met          mple
ce_v7.1

read comp

 u-235 1 den=18.742000 1.0 300 end

end comp

```

Figure 47. Dropdown menu necessary to create mixing table.

Upon executing the mixing table calculation, the **Messages** tab at the bottom right-hand corner of the Fulcrum panel should illuminate. Clicking on the **Messages** tab will reveal the mixing table results, as shown in Figure 48 (enlarged for readability). The resulting mixing table includes the mixture number, mixture temperature, and mixture density, along with the atom density and the nuclide identifier (92235). Geometric specification of problems is covered in Section 4.



```
uranium metal mixture example
ce_v7.1

read comp
u-235 1 den=18.742000 1.0 300.0 end
end comp
```

```
Mixture tables for file W:/keno_primers/k5/uranium_metal.inp

----------------------------------------------------------------------------------------
 Mixture table for component comps at line:5, column:1
----------------------------------------------------------------------------------------
Mixture = 1 with density(g/cc) = 18.7420 and temperature(K) = 300.00
        Entry      Mixture       Nuclide      Density (atoms/b-cm)      Temperature (K)
----------------------------------------------------------------------------------------
            1            1         92235             4.801953e-002               300.00
----------------------------------------------------------------------------------------
        Entry      Mixture       Nuclide      Density (atoms/b-cm)      Temperature (K)
----------------------------------------------------------------------------------------
 End of mixture table for component comps at line:5, column:1
----------------------------------------------------------------------------------------


End of mixture tables for file W:/keno primers/k5/uranium metal.inp
```

Figure 48. Mixing table results for the uranium metal example.

3.4.2 U(93.71) Metal Material

This example changes from material used from pure ^{235}U metal to U metal enriched to 93.71 wt.% ^{235}U. Whereas the first example was composed of a single isotope, this example is a combination of isotopes from the same element. This example can still be specified with a single standard basic composition input, but the isotopic distribution must be modified. For this example, the density of the core material is again 18.742 g/cm^3. Relevant material specification information is as follows:

Core Material (ρ_{mix} = 18.742 g/cc-mixture)

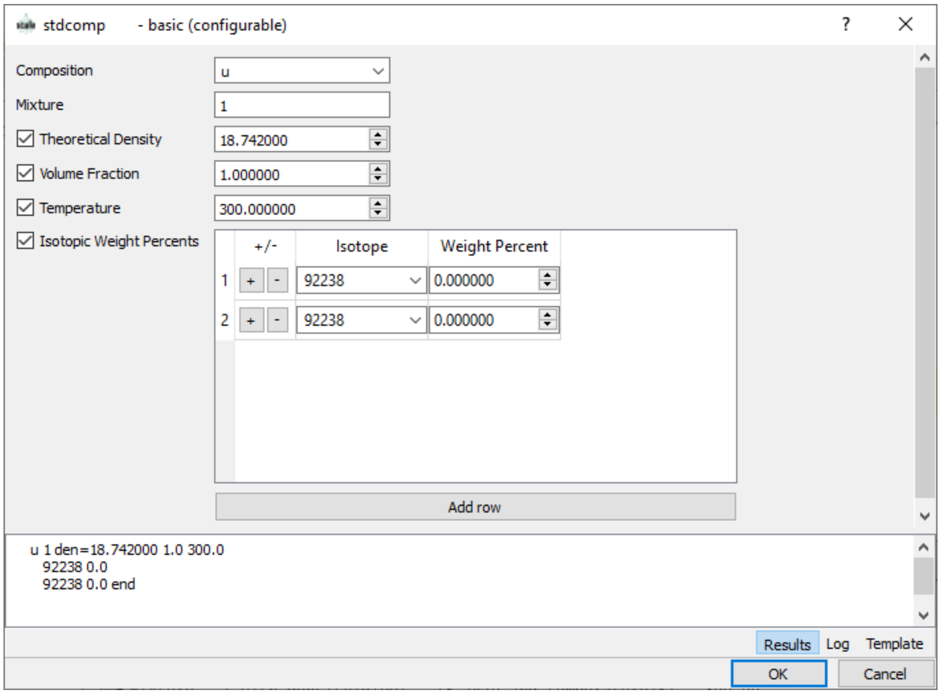
U metal (93.71 wt.% ^{235}U and 6.29 wt.% ^{238}U)

Temperature of 300 K

3.4.2.1 Material input

As in the ^{235}U metal problem, generate a CSAS6 input skeleton from Fulcrum and specify an appropriate title and cross section library. For this example, replace **title-goes-here** with `enriched uranium metal mixture example.`

To define the enriched uranium metal composition, move the cursor into the **read comp** block, and press `CTRL-SPACE` to generate the dropdown menu shown in Figure 43. From the dropdown menu, select again **stdcomp – basic (configurable)**. This will generate a fillable form, as was shown in Figure 44. This time in the **composition** dropdown, select the elemental input `u` by clicking on the blank with `ac-206` in it and typing over it. Next, adjust the density by clicking **Theoretical Density** checkbox and entering the value of `18.742` in place of the default value of `1.0`. Because the material is pure U, and we have already specified the density, the default **Volume Fraction** of `1.0` is appropriate. The temperature of the material is also modified by clicking the **Temperature** box and entering a value of `300`. To specify the isotopic composition of the material, click the **Isotopic Weight Percents** box and then click the **Add row** button twice. At this point, you should have a configurable form that looks like that shown in Figure 49. Note that you may have to expand the window to see the **Add row** button.



Composition: u

Mixture: 1

☑ Theoretical Density: 18.742000

☑ Volume Fraction: 1.000000

☑ Temperature: 300.000000

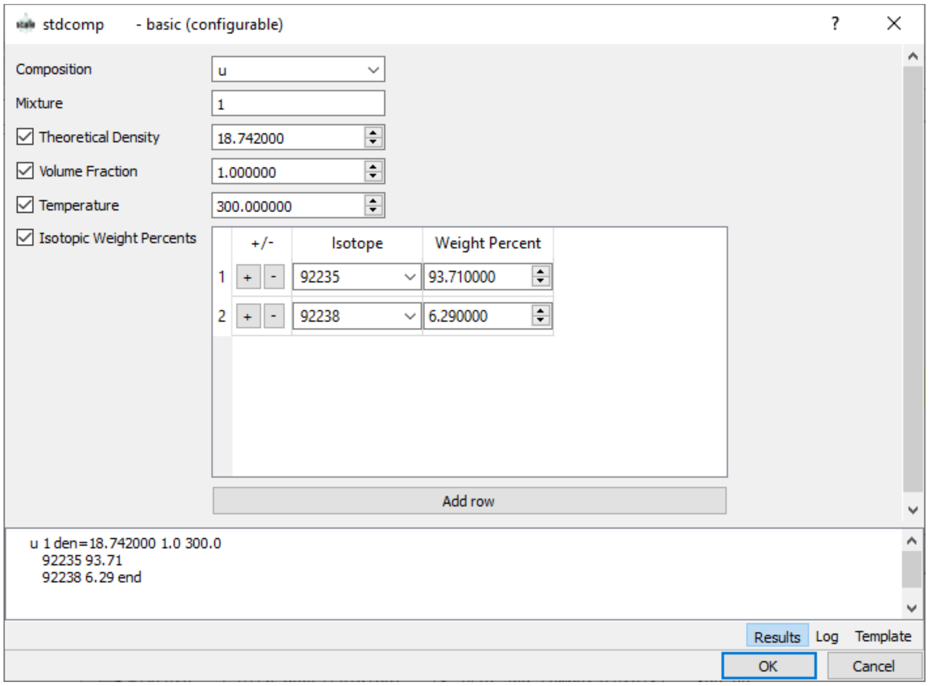
☑ Isotopic Weight Percents

	+/-	Isotope	Weight Percent
1	+ -	92238	0.000000
2	+ -	92238	0.000000

```
u 1 den=18.742000 1.0 300.0
  92238 0.0
  92238 0.0 end
```

Figure 49. Isotopic composition input prior to user modification.

Now you should select one of the **92238** values listed under the **Isotope** column, and change the value to the nuclide identifier for ^{235}U, which is `92235`. Next, enter the isotopic weight percentages in the **Weight Percent** column by clicking on the box with the **0.000000** value inside and over writing it with `93.71` for the ^{235}U entry and `6.29` for ^{238}U. The final configurable window is shown in Figure 50.



Composition: u

Mixture: 1

☑ Theoretical Density: 18.742000

☑ Volume Fraction: 1.000000

☑ Temperature: 300.000000

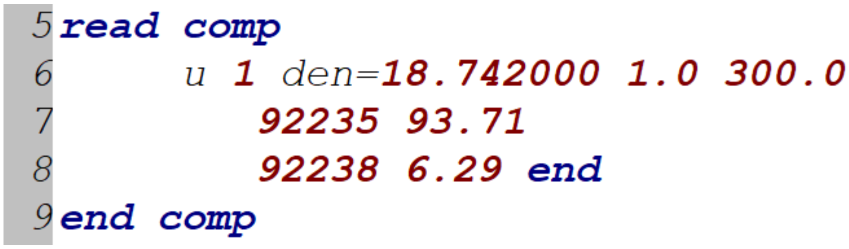
☑ Isotopic Weight Percents

	+/-	Isotope	Weight Percent
1	+ -	92235	93.710000
2	+ -	92238	6.290000

```
u 1 den=18.742000 1.0 300.0
  92235 93.71
  92238 6.29 end
```

Figure 50. Isotopic composition input following user modification for the enriched uranium metal case.

Confirm that the data entry is correct before clicking **OK**, as the data cannot be modified in the configurable form after it has been entered into the input. Once you have clicked **OK** at the bottom of the configurable form, the input file should look like that shown in Figure 51.



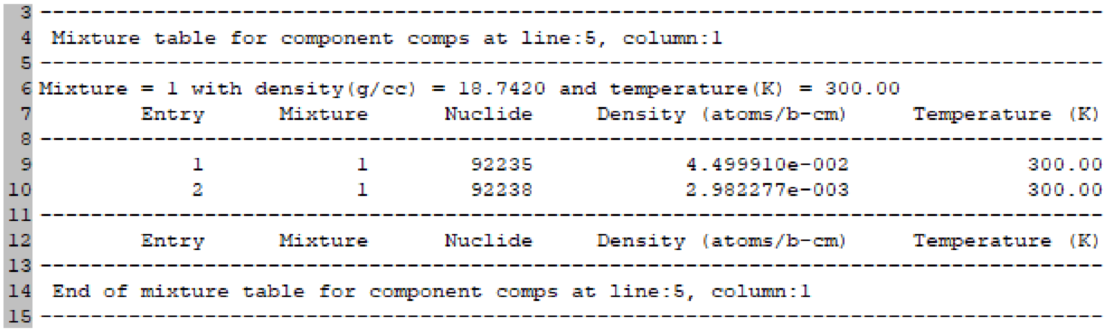
```
read comp
       u 1 den=18.742000 1.0 300.0
         92235 93.71
         92238 6.29 end
end comp
```

Figure 51. Final composition input for the enriched uranium metal case.

3.4.2.2 Mixing table output

At this point there is not enough input specified to perform a KENO-VI calculation, but the atom densities can be generated based on the standard composition input supplied. To do this, generate a MIXING TABLE with CSAS by clicking the downward pointing triangle next to the **Run** button at the top of the Fulcrum interface and selecting **Mixing table** from the dropdown menu. A picture of the dropdown menu for the ^{235}U metal example is provided in Figure 47.

Upon executing the mixing table calculation, the **Messages** tab at the bottom right-hand corner of the Fulcrum panel should illuminate. Clicking on the **Messages** tab will reveal the mixing table results, which are shown in Figure 52 (enlarged for readability). The resulting mixing table includes the mixture number, the mixture temperature, and the mixture density, along with atom density and nuclide identifiers (92235 and 92238). Geometric specification of problems is covered in Section 4.



```
--------------------------------------------------------------------------------
 Mixture table for component comps at line:5, column:1
--------------------------------------------------------------------------------
Mixture = 1 with density(g/cc) = 18.7420 and temperature(K) = 300.00
        Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
--------------------------------------------------------------------------------
            1            1        92235            4.499910e-002              300.00
            2            1        92238            2.982277e-003              300.00
--------------------------------------------------------------------------------
        Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
--------------------------------------------------------------------------------
 End of mixture table for component comps at line:5, column:1
--------------------------------------------------------------------------------
```

Figure 52. Mixing table results for the uranium metal example.

3.4.3 Highly Enriched UO_2

This example changes the material from uranium metal to uranium dioxide while keeping the enrichment at 93.71% ^{235}U. The core material has a density of 95% of the theoretical density of UO_2; the theoretical density is 10.96 g/cc.

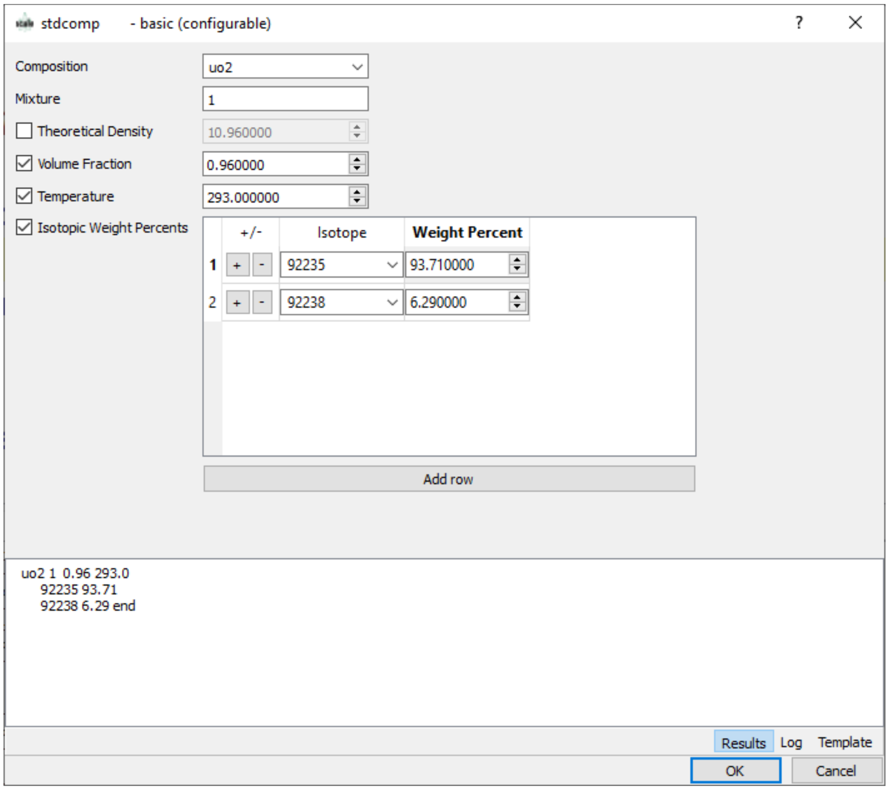
Core Material (ρ_{mix} = 95% of theoretical density – TD = 10.96 g/cc-mixture)

U (93.71 wt.% ^{235}U and 6.29 wt.% ^{238}U)

It is suggested that you create a new file for this problem and change the title of the problem to `highly enriched uranium oxide example`.

3.4.3.1 Material input

Once you have the same input skeleton that was used in the previous examples, the first step is to define the uranium oxide composition. Move the cursor into the **read comp** block and press `CTRL-SPACE` to generate the dropdown menu similar to what is shown in Figure 43 above. From the dropdown menu, select **stdcomp – basic (configurable)**. This will generate a configurable form, as shown in Figure 44. In the **composition** dropdown, select `uo2` by clicking on the blank with **ac-206** and typing over it. Next, adjust the density by either changing the default density or by changing the volume fraction. Since changing the default density has already been demonstrated and the material is specified as a fraction of theoretical density, the volume fraction is modified here. The volume fraction, also known as the *density multiplier*, is modified by clicking the box next to **Volume Fraction** and entering a value of `0.96` in place of **1.00000** in the adjacent box. Modifying the **Volume Fraction** to 0.96 results in a density of 0.96 * 10.96, or 10.412 g/cm^3. For this example, leave the temperature of the material at the default value of 293 K. To specify the isotopic composition of the material, click the **Isotopic Weight Percents** box, and then click the **Add row** button twice. Note that you may have to expand the configurable window to see the **Add row** button. Now select one of the **92238** values listed under the **Isotope** column, and change this value to the nuclide identifier for ^{235}U, which is `92235`. Next, enter the isotopic weight percentages into the **Weight Percent** column by clicking on the box with the **0.000000** value inside and overwriting it with `93.71` for the ^{235}U entry, and enter `6.29` for ^{238}U. The final configurable window is shown in Figure 53. Confirm that the data entry is correct before clicking **OK**, as the data cannot be modified in the configurable form after it has been entered into the input. After clicking **OK** at the bottom of the configurable form, the input file should look like that shown in Figure 54.



stdcomp - basic (configurable)

Composition: uo2

Mixture: 1

☐ Theoretical Density: 10.960000

☑ Volume Fraction: 0.960000

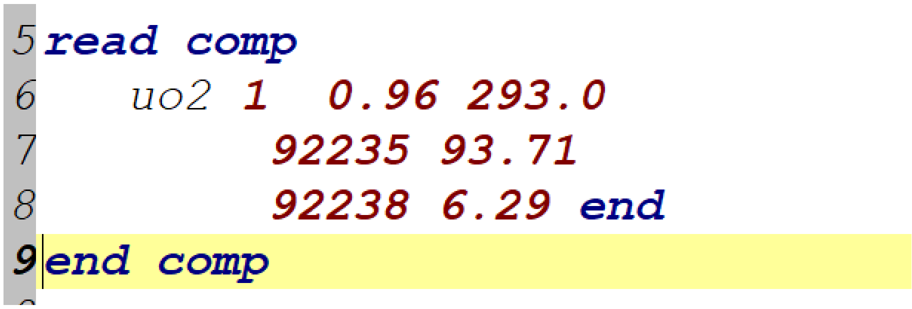
☑ Temperature: 293.000000

☑ Isotopic Weight Percents

	+/-	Isotope	Weight Percent
1	+ -	92235	93.710000
2	+ -	92238	6.290000

```
uo2 1 0.96 293.0
   92235 93.71
   92238 6.29 end
```

Figure 53. Isotopic composition input following user modification for the enriched uranium oxide case.



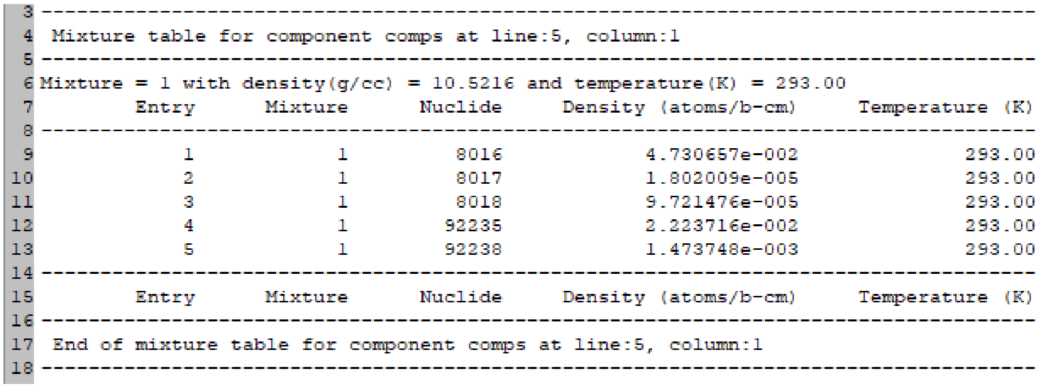
```
read comp
    uo2 1  0.96 293.0
           92235 93.71
           92238 6.29 end
end comp
```

Figure 54. Final composition input for the enriched uranium oxide case.

3.4.3.2 Mixing table output

As encountered in the first two examples, there is not enough input specified to perform a KENO-VI calculation, but the atom densities can be generated based on the standard composition input supplied. Click the downward pointing triangle next to the **Run** button at the top of the Fulcrum interface and select **Mixing table** from the dropdown menu. A picture of the dropdown menu for the uranium oxide example is provided in Figure 47.

Upon executing the mixing table calculation, the **Messages** tab at the bottom right-hand corner of the Fulcrum panel will illuminate. Clicking on the **Messages** tab will reveal the mixing table results. The mixing table results are shown in Figure 55 (enlarged for readability). The resulting mixing table includes the mixture number, the mixture temperature, and the mixture density, along with the atom density and the nuclide identifiers (92235, 92238, 8016, 8017, and 8018).



```
------------------------------------------------------------------------------------
 Mixture table for component comps at line:5, column:1
------------------------------------------------------------------------------------
Mixture = 1 with density(g/cc) = 10.5216 and temperature(K) = 293.00
        Entry      Mixture       Nuclide      Density (atoms/b-cm)      Temperature (K)
------------------------------------------------------------------------------------
            1            1          8016           4.730657e-002                293.00
            2            1          8017           1.802009e-005                293.00
            3            1          8018           9.721476e-005                293.00
            4            1         92235           2.223716e-002                293.00
            5            1         92238           1.473748e-003                293.00
------------------------------------------------------------------------------------
        Entry      Mixture       Nuclide      Density (atoms/b-cm)      Temperature (K)
------------------------------------------------------------------------------------
 End of mixture table for component comps at line:5, column:1
------------------------------------------------------------------------------------
```

Figure 55. Mixing table results for the highly enriched UO_2 case.

3.4.4 Highly Enriched $U(30.3)O_2F_2$ Material

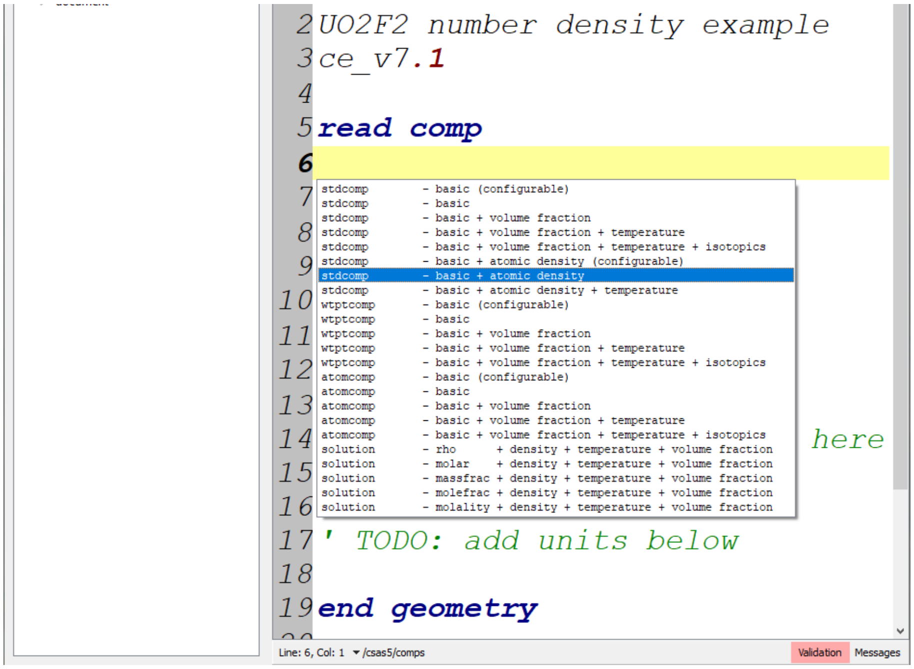
This example demonstrates the method used to enter material composition information when atomic number density information has been calculated outside of SCALE. This example uses a $U(30.3)O_2F_2$ solution from *Critical Dimensions of Systems Containing ^{235}U, ^{239}U, and ^{233}U* (LA-10860) [5] as the example. Later, number densities for a solution will be determined using SCALE's solution input capability. However, the information provided in LA-10860 is formatted in a manner that is not convenient for entry into that method. For now, it is assumed that the number densities of the nuclides in the solution are known to be as follows:

N_H	= 0.0597522 atoms-H/b-cm
N_O	= 0.0335605 atoms-O/b-cm
N_F	= 0.0036844 atoms-F/b-cm
$N_{U\text{-}235}$	= 0.0005637 atoms-^{235}U/b-cm
$N_{U\text{-}238}$	= 0.0012802 atoms-^{238}U/b-cm

It is suggested that a new file be created for this problem with the title of the problem changed to `UO2F2 number density example.`

3.4.4.1 Material input

Once you have the same input skeleton that was used in the previous examples, the first step is to define the UO_2F_2 composition. Move the cursor into the **read comp** block and press `CTRL-SPACE` to generate the dropdown menu shown in Figure 56. This example will be implemented using the place-holder method of material entry rather than with the configurable forms method that was used in the previous examples. From the dropdown menu, select **stdcomp – basic + atomic density**. This will place the text shown in Figure 57 into the input file. This text is correctly formatted to produce an atomic density input, but it is generic and must be updated to fit the specifics of the problem. To do this, the text should be altered to change the **0.983252** atomic density value found in the autocompleted template with `0.0005637`. Next, the line of text that has just been constructed should be copied to four additional lines of input—one for each nuclide in the composition—to accommodate the other constituent materials in the mixture. Those lines should then be modified to have the appropriate elemental or nuclide symbols and atomic densities. It is also necessary for each of the lines to have the same composition numbers, because this links them together in the input processor as a single mixture. The **Volume Fraction** is set to zero in all cases: this signals SCALE that the information that follows is an atomic density rather another type of input. The final material input should look like Figure 58.

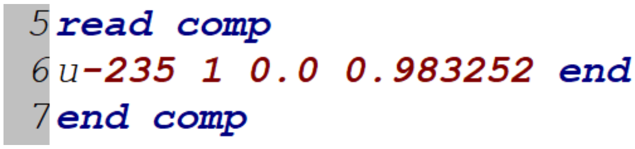


```
UO2F2 number density example
ce_v7.1

read comp

```

```
stdcomp       - basic (configurable)
stdcomp       - basic
stdcomp       - basic + volume fraction
stdcomp       - basic + volume fraction + temperature
stdcomp       - basic + volume fraction + temperature + isotopics
stdcomp       - basic + atomic density (configurable)
stdcomp       - basic + atomic density
stdcomp       - basic + atomic density + temperature
wtptcomp      - basic (configurable)
wtptcomp      - basic
wtptcomp      - basic + volume fraction
wtptcomp      - basic + volume fraction + temperature
wtptcomp      - basic + volume fraction + temperature + isotopics
atomcomp      - basic (configurable)
atomcomp      - basic
atomcomp      - basic + volume fraction
atomcomp      - basic + volume fraction + temperature
atomcomp      - basic + volume fraction + temperature + isotopics
solution      - rho      + density + temperature + volume fraction
solution      - molar    + density + temperature + volume fraction
solution      - massfrac + density + temperature + volume fraction
solution      - molefrac + density + temperature + volume fraction
solution      - molality + density + temperature + volume fraction
```

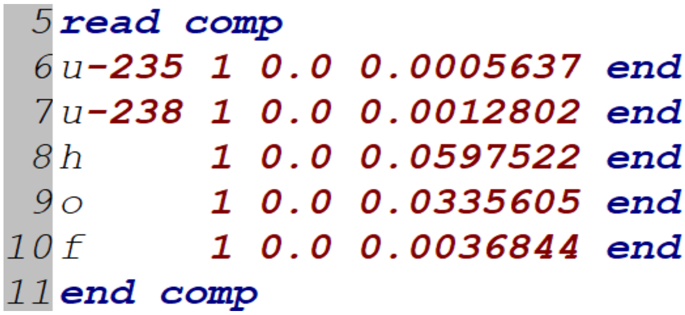
```
                                                       here


' TODO: add units below

end geometry
```

Figure 56. Accessing the basic standard composition configurable.



```
read comp
u-235 1 0.0 0.983252 end
end comp
```

Figure 57. Autocompleted template for atomic density input.



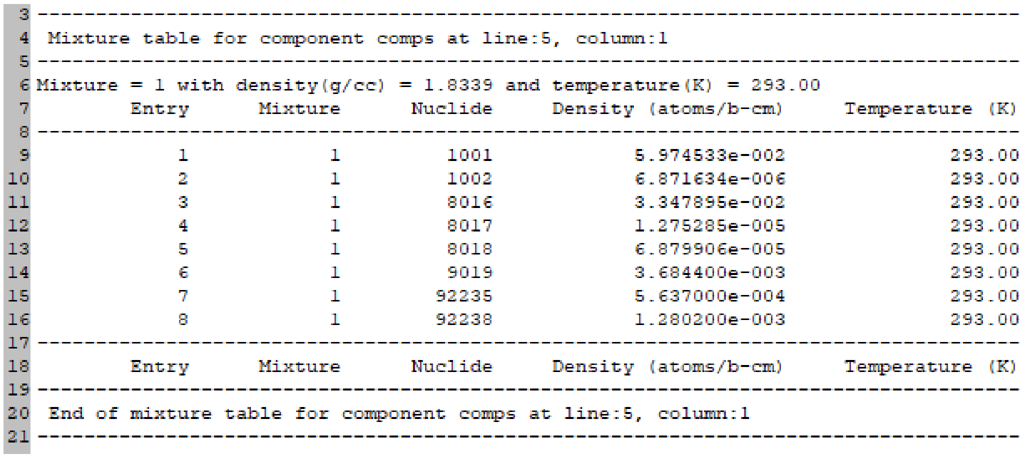
```
read comp
u-235 1 0.0 0.0005637 end
u-238 1 0.0 0.0012802 end
h     1 0.0 0.0597522 end
o     1 0.0 0.0335605 end
f     1 0.0 0.0036844 end
end comp
```

Figure 58. Completed atomic density input.

3.4.4.2 Mixing table output

As in the previous examples, there is not enough input specified to perform a KENO-VI calculation, but the atom densities can be generated based on the standard composition input supplied. Click the downward pointing triangle next to the **Run** button at the top of the Fulcrum interface and select **Mixing table** from the dropdown menu. A picture of the dropdown menu for the uranium metal example is provided in Figure 47.

Upon executing the mixing table calculation, the **Messages** tab at the bottom right-hand corner of the Fulcrum panel will illuminate. Clicking on the **Messages** tab will reveal the mixing table results, as shown in Figure 59 (enlarged for readability). The resulting mixing table includes the mixture number, the mixture temperature, and the mixture density, along with the atom density and nuclide identifiers (1001, 1002, 8016, 8017, 8018, 9019, 92235, and 92238).



```
--------------------------------------------------------------------------------
 Mixture table for component comps at line:5, column:1
--------------------------------------------------------------------------------
Mixture = 1 with density(g/cc) = 1.8339 and temperature(K) = 293.00
        Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
--------------------------------------------------------------------------------
            1            1         1001             5.974533e-002              293.00
            2            1         1002             6.871634e-006              293.00
            3            1         8016             3.347895e-002              293.00
            4            1         8017             1.275285e-005              293.00
            5            1         8018             6.879906e-005              293.00
            6            1         9019             3.684400e-003              293.00
            7            1        92235             5.637000e-004              293.00
            8            1        92238             1.280200e-003              293.00
--------------------------------------------------------------------------------
        Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
--------------------------------------------------------------------------------
 End of mixture table for component comps at line:5, column:1
--------------------------------------------------------------------------------
```

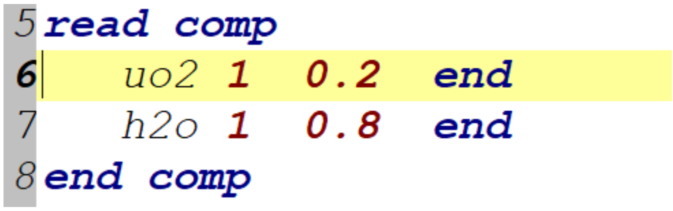
Figure 59. Mixing table results for the highly enriched UO_2F_2 case.

3.4.5 H_2O/UO_2 Material Combining Example

This example demonstrates how a user would combine any two basic standard compositions to define a mixture composed of two materials. This method for combining materials is not limited to standard basic compositions; it may also be used with the material types discussed in Section 7. Typically, the description of a material provided in source documents will provide either the volume fractions of individual components of known density or the weight fractions of individual materials with an overall mixture density. For this example, a mixture of water and uranium dioxide is used to illustrate how individual materials can be added to the same mixture. The mixture is 80% by volume H_2O and 20% by volume natural UO_2. This is equivalent to a mixture that is 26.703 wt.% H_2O and 73.297 wt.% UO_2, with an overall density of 2.9906 g/cm^3. We will model both approaches.

3.4.5.1 Material input

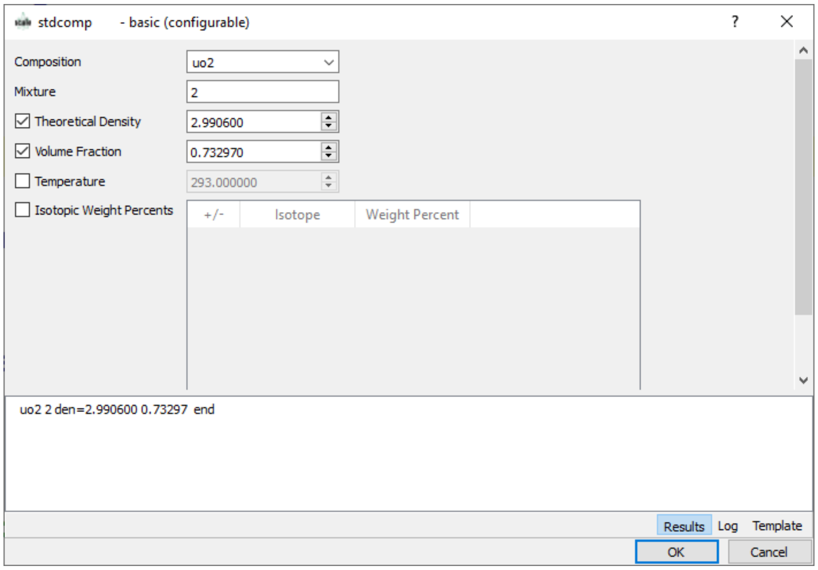
Once you have the same input skeleton that was used in the previous examples, the first step is to define the uranium oxide composition. Move the cursor into the **read comp** block, and press `CTRL-SPACE` to generate the dropdown menu similar to what is shown in Figure 43 above. From the dropdown menu, select **stdcomp – basic (configurable)**. This will generate a configurable form, as shown in Figure 44. In the **composition** dropdown, select `uo2` by clicking on the blank with **ac-206** in it and typing over it. Next, adjust the volume fraction to account for the UO_2 only making up 20% of the mixture by clicking the adjustable window next to **Volume Fraction** and overwriting **1.000000** with `0.20`. At this point, the UO_2 portion of the first entry is complete. Click **OK** at the bottom of the configurable form. Next, add the H_2O portion of the mixture by performing the same process to bring up the composition dropdown menu. Select **stdcomp – basic (configurable)**, and then select `h2o` by clicking on the blank with **ac-206** in it and typing over it. Next adjust the volume fraction to account for the H_2O only making up 80% of the mixture. To do this, click the adjustable window next to **Volume Fraction**, and overwrite **1.000000** with `0.80`. It is important that both the UO_2 and H_2O have the same mixture number. This mixture entry will result in a mixture with $0.2 * \rho_{UO2} + 0.80 * \rho_{H2O} = 0.2*(10.9600) + 0.8*(0.9982) = 2.9906$. Note that this is the value given above for the density. At this point, the input should look like Figure 60.



```
read comp
    uo2 1  0.2  end
    h2o 1  0.8  end
end comp
```

Figure 60. Input with mixing by volume fraction.

Now the same mixture will be specified using the weight fraction method. This is done in a similar manner by using the **stdcomp – basic (configurable)** once for each of the two compositions, UO_2 and H_2O. This first form will be used to enter the UO_2 composition. In the UO_2 form, the **Mixture** entry should be changed to `2`, the **Theoretical Density** value changed to `2.9906`, and the **Volume Fraction** changed to `0.732970`. Once this is done, click the **OK** button at the bottom of the pane. In the H_2O form, the **Mixture** entry should also be changed to `2`, the **Theoretical Density** should be changed to `2.9906`, and the **Volume Fraction** should be changed to `0.26703`. This mixture entry will result in a mixture with 0.2 * ρ_{UO2} + 0.80 * ρ_{H2O} = 0.2*(10.9600) + 0.8*(0.9982) = 2.9906. Each of the configurable forms is shown in Figure 61and Figure 62, and the final input, including both the mixing by volume fraction and mixing by weight fraction, is shown in Figure 63.



stdcomp - basic (configurable)

Composition: uo2

Mixture: 2

☑ Theoretical Density: 2.990600

☑ Volume Fraction: 0.732970

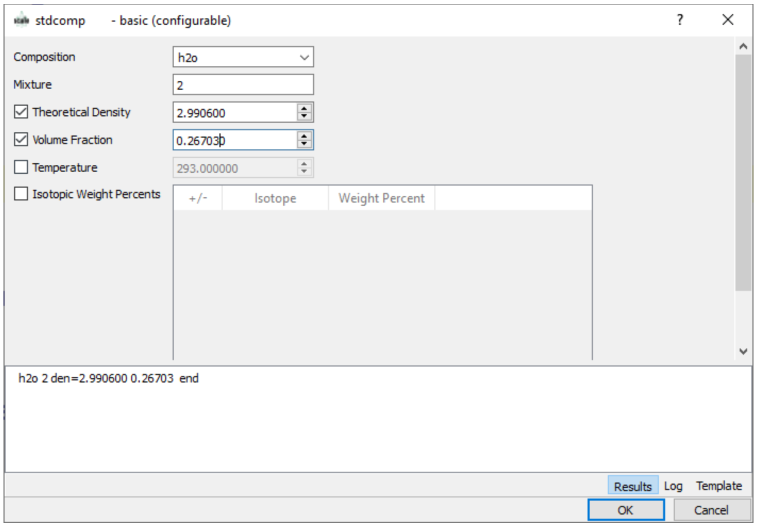
☐ Temperature: 293.000000

☐ Isotopic Weight Percents

+/-	Isotope	Weight Percent

```
uo2 2 den=2.990600 0.73297  end
```

Figure 61. UO_2 configurable form for mixing by weight fraction.



stdcomp - basic (configurable)

Composition: h2o

Mixture: 2

☑ Theoretical Density: 2.990600

☑ Volume Fraction: 0.267030

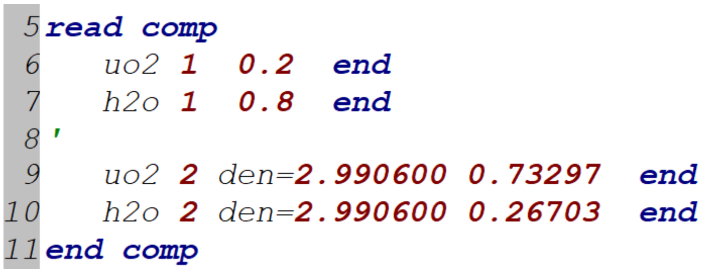
□ Temperature: 293.000000

□ Isotopic Weight Percents

+/-	Isotope	Weight Percent

```
h2o 2 den=2.990600 0.26703  end
```

Figure 62. H_2O configurable form for mixing by weight fraction.



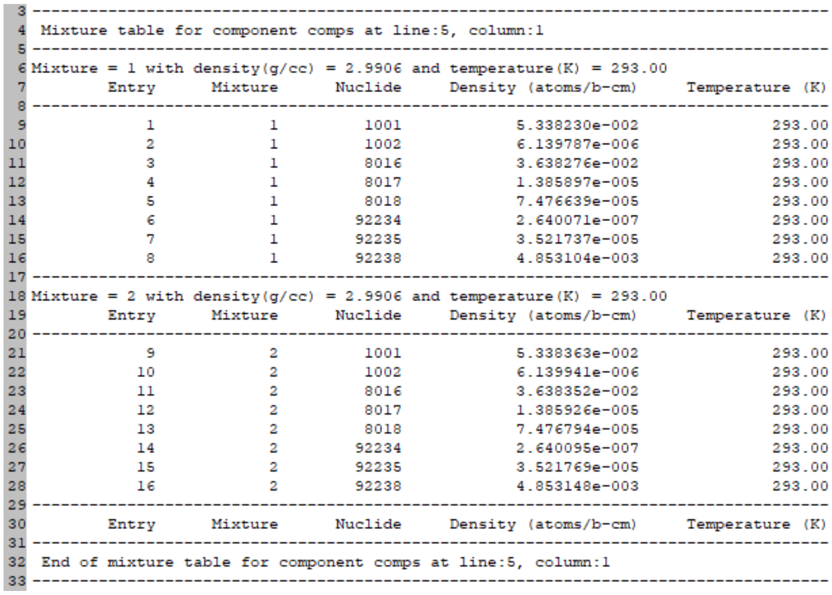
```
read comp
    uo2 1  0.2  end
    h2o 1  0.8  end
'
    uo2 2 den=2.990600 0.73297  end
    h2o 2 den=2.990600 0.26703  end
end comp
```

Figure 63. Final material specification input for mixing by volume fraction and mixing by weight fraction.

3.4.5.2 Mixing table output

As in the previous examples, there is not enough input specified to perform a KENO-VI calculation, but the atom densities can be generated based on the standard composition input supplied. Click the downward pointing triangle next to the **Run** button at the top of the Fulcrum interface and select **Mixing table** from the dropdown menu. A picture of the dropdown menu for the uranium metal example is provided in Figure 47.

Upon executing the mixing table calculation, the **Messages** tab at the bottom right-hand corner of the Fulcrum panel should illuminate. Clicking on the **Messages** tab will reveal the mixing table results. The mixing table results are presented in Figure 64. The resulting mixing table includes the mixture number, the mixture temperature, and the mixture density, along with the atom density and the nuclide identifiers (1001, 1002, 8016, 8017, 8018, 92234, 92235, and 92238).



```
------------------------------------------------------------------------------------
 Mixture table for component comps at line:5, column:1
------------------------------------------------------------------------------------
Mixture = 1 with density(g/cc) = 2.9906 and temperature(K) = 293.00
        Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
------------------------------------------------------------------------------------
            1            1         1001            5.338230e-002              293.00
            2            1         1002            6.139787e-006              293.00
            3            1         8016            3.638276e-002              293.00
            4            1         8017            1.385897e-005              293.00
            5            1         8018            7.476639e-005              293.00
            6            1        92234            2.640071e-007              293.00
            7            1        92235            3.521737e-005              293.00
            8            1        92238            4.853104e-003              293.00
------------------------------------------------------------------------------------
Mixture = 2 with density(g/cc) = 2.9906 and temperature(K) = 293.00
        Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
------------------------------------------------------------------------------------
            9            2         1001            5.338363e-002              293.00
           10            2         1002            6.139941e-006              293.00
           11            2         8016            3.638352e-002              293.00
           12            2         8017            1.385926e-005              293.00
           13            2         8018            7.476794e-005              293.00
           14            2        92234            2.640095e-007              293.00
           15            2        92235            3.521769e-005              293.00
           16            2        92238            4.853148e-003              293.00
------------------------------------------------------------------------------------
        Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
------------------------------------------------------------------------------------
 End of mixture table for component comps at line:5, column:1
------------------------------------------------------------------------------------
```

Figure 64. Mixing table output for the mixing by volume fraction and the mixing by weight fraction examples.

3.5 SUMMARY

This section has helped you to:

- define the different criticality sequences used in SCALE;
- describe the cross section libraries available for criticality analyses;
- use the Fulcrum user interface to provide data on elements, isotopes, compounds, and mixtures of these types; and
- interpret basic mixing table output information from a SCALE analysis.

Now that you have spent time with the material input, you can proceed to geometry definitions.

4. GEOMETRY INPUT

In Section 2, you modeled a simple geometry with a single material composition using the Fulcrum user interface to provide data on elements, isotopes, compounds, and solutions. This section explains the commands used for basic KENO-VI geometries such as cylinders and spheres and for more complicated geometries such as reflected systems and lattices.

4.1 WHAT YOU WILL BE ABLE TO DO

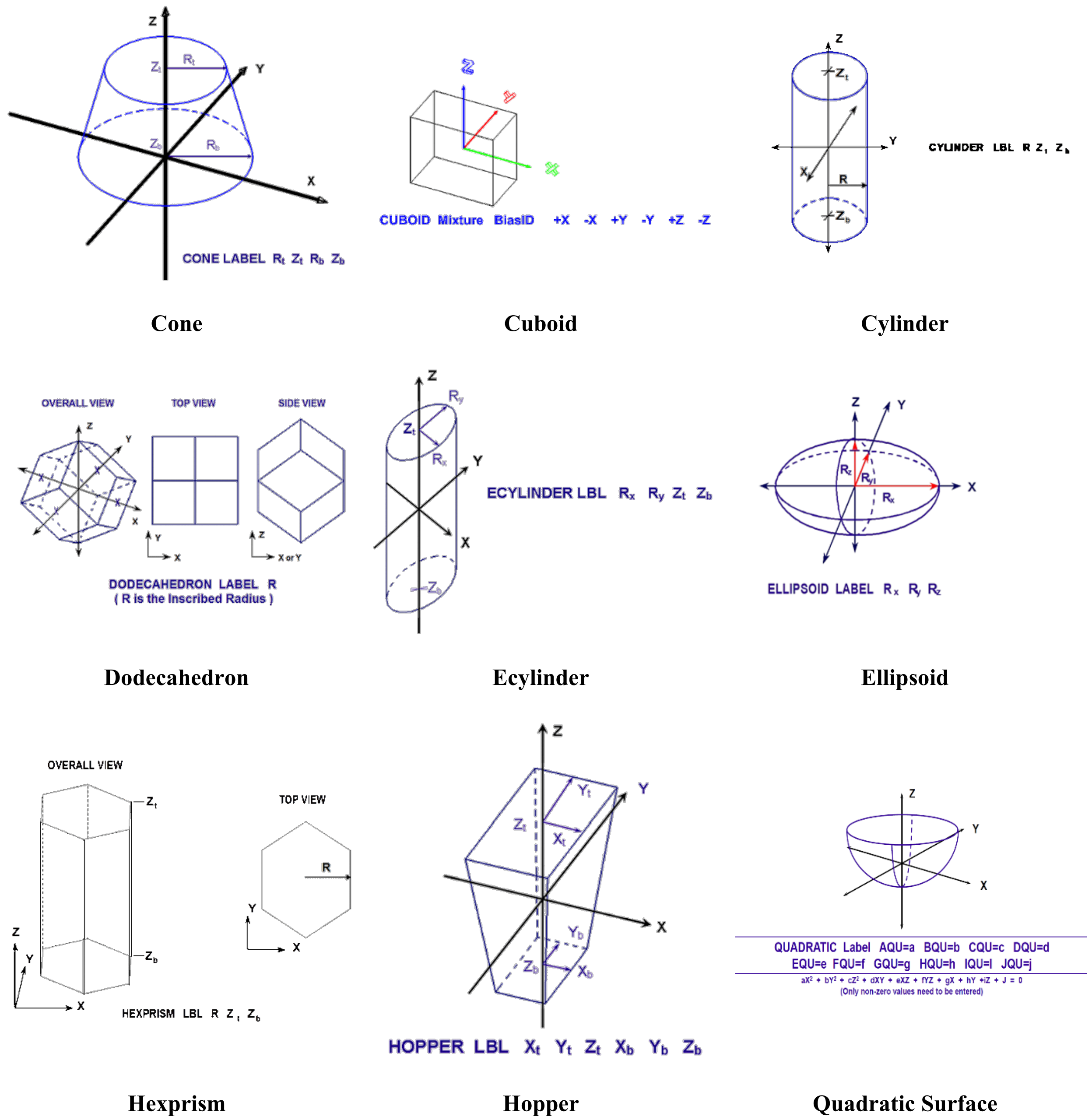
- Use Fulcrum to describe some basic KENO-VI geometry shapes.
- Understand how units are created (including intersecting of shapes).
- Locate and change the location of the origin for shapes and units.
- Create simple arrays consisting of a single unit.
- Create arrays with multiple units of nominally different sizes.
- Use the volume block to calculate volume of regions and/or mixtures.

4.2 BASIC GEOMETRY SHAPES

SCALE/KENO-VI uses a set of basic shapes to build the geometry models. One example has already been presented, using Fulcrum to enter data on spheres in Section 2. The sphere and the other basic shapes are described in Table 2 and are illustrated in Figure 65 and Figure 66.

Table 2. Basic shapes.

Keyword	Description
CONE	Right circular cone, can be truncated
CUBOID	Box or rectangular parallelepiped
CYLINDER	Right circular cylinder
DODECAHEDRON	Surface consisting of 12 rhombi of the same size
ECYLINDER	Right elliptic cylinder
ELLIPSOID	Surface formed by rotating an ellipse
HEXPRISM	Right hexagonal prism (flat up)
HOPPER	Truncated square pyramid
QUADRATIC SURFACE	Quadric shape defined by a quadratic surface equation
PARALLELEPIPED	Six faces composed of parallelograms with opposing parallel faces
PENTAGON	Right pentagonal prism
PLANE	Arbitrary 2D plane
RING	Right annular prism
RHEXPRISM	Hexprism rotated 30 degrees about the long axis (point up)
RHOMBOID	Six identical faces, each one a rhombus
SPHERE	Sphere
WEDGE	Right prism with triangular cross section
XPPLANE	Two planes perpendicular to the X-axis
YPPLANE	Two planes perpendicular to the Y-axis
ZPPLANE	Two planes perpendicular to the Z-axis

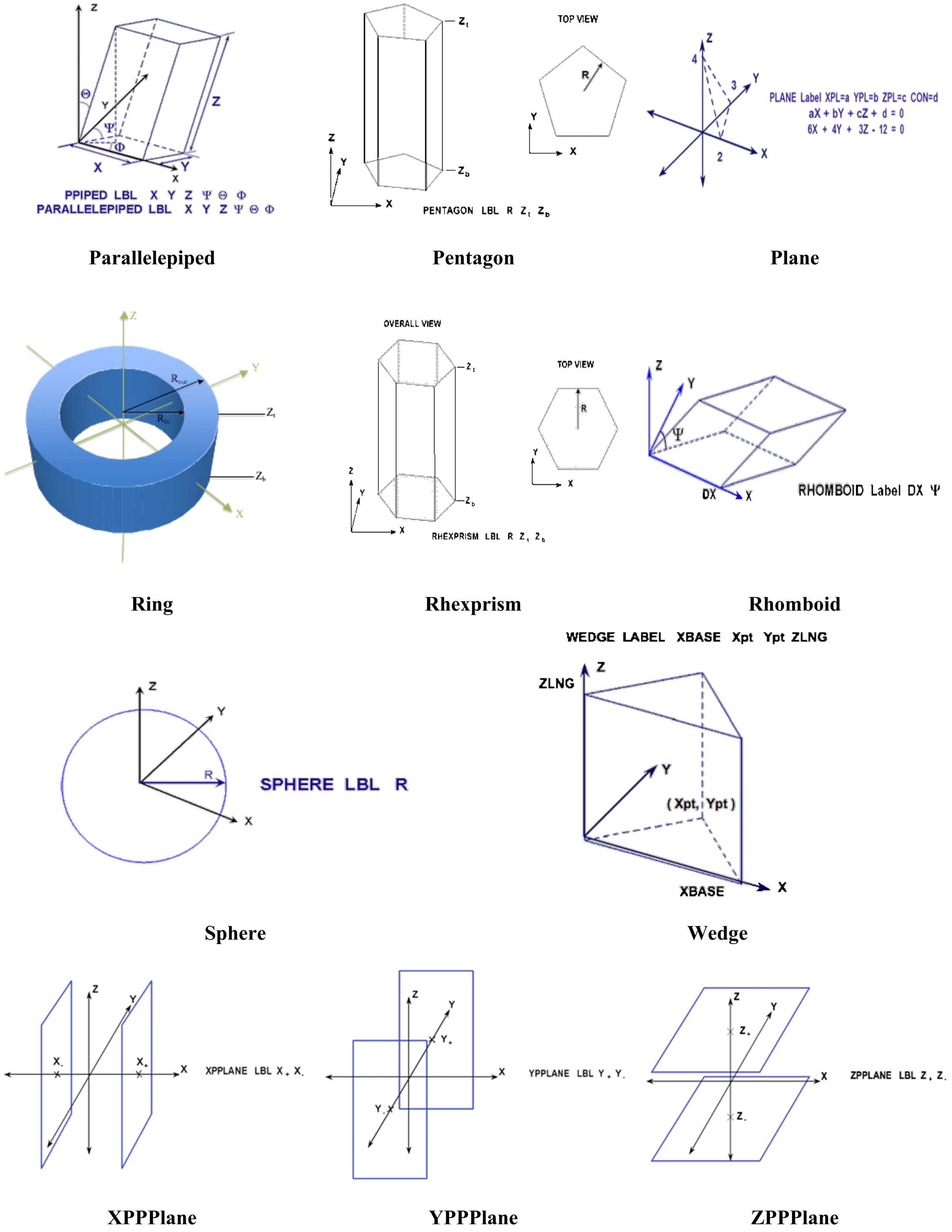

Z
Y
X
Zt
Rt
Zb
Rb
CONE LABEL Rt Zt Rb Zb
Z
Y
X
CUBOID Mixture BiasID +X -X +Y -Y +Z -Z
Z
Zt
Y
X
R
Zb
CYLINDER LBL R Zt Zb
OVERALL VIEW
TOP VIEW
SIDE VIEW
Z
Y
X
Y
X
Z
X or Y
DODECAHEDRON LABEL R
(R is the Inscribed Radius)
Z
Ry
Zt
Rx
Y
X
Zb
ECYLINDER LBL Rx Ry Zt Zb
Z
Y
X
Rz
Ry
Rx
ELLIPSOID LABEL Rx Ry Rz
OVERALL VIEW
Zt
Zb
Z
Y
X
HEXPRISM LBL R Zt Zb
TOP VIEW
R
Y
X
Z
Y
Yt
Zt
Xt
X
Yb
Zb
Xb
HOPPER LBL Xt Yt Zt Xb Yb Zb
Z
Y
X
QUADRATIC Label AQU=a BQU=b CQU=c DQU=d
EQU=e FQU=f GQU=g HQU=h IQU=I JQU=j
aX² + bY² + cZ² + dXY + eXZ + fYZ + gX + hY +iZ + J = 0
(Only non-zero values need to be entered)

Cone Cuboid Cylinder

Dodecahedron Ecylinder Ellipsoid

Hexprism Hopper Quadratic Surface

Figure 65. KENO-VI geometry shapes.



Parallelepiped

Pentagon

Plane

Ring

Rhexprism

Rhomboid

Sphere

Wedge

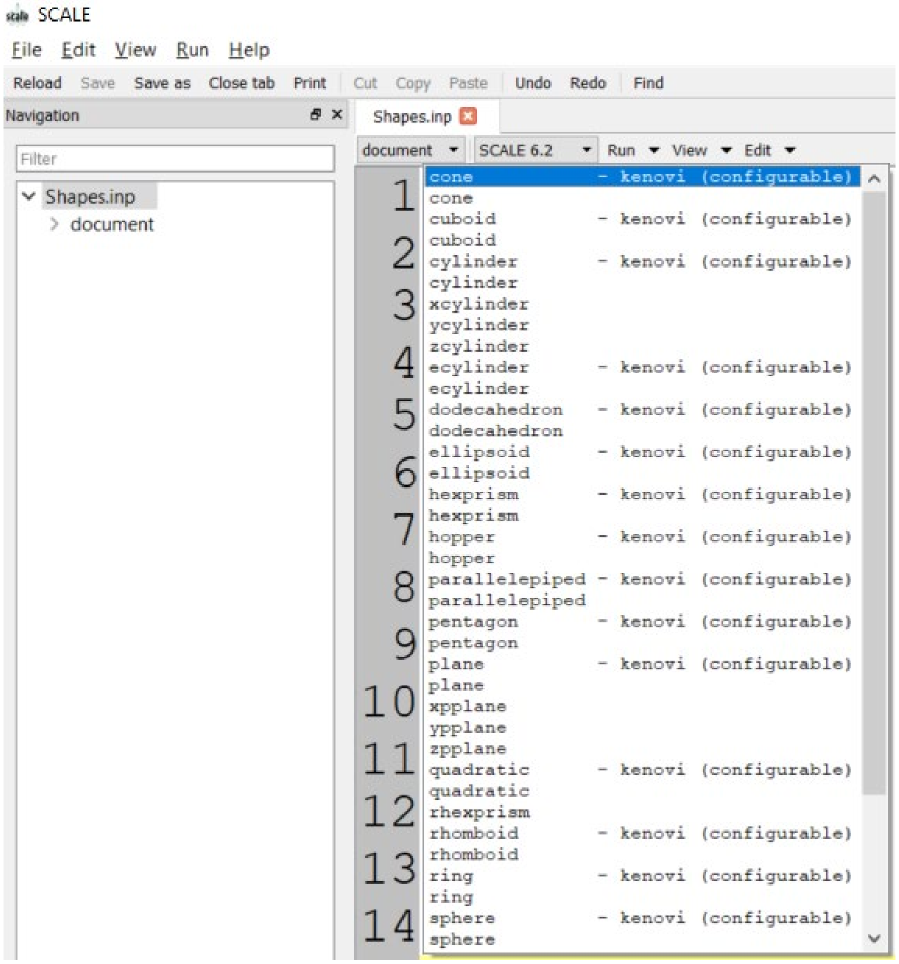
XPPlane

YPPlane

ZPPlane

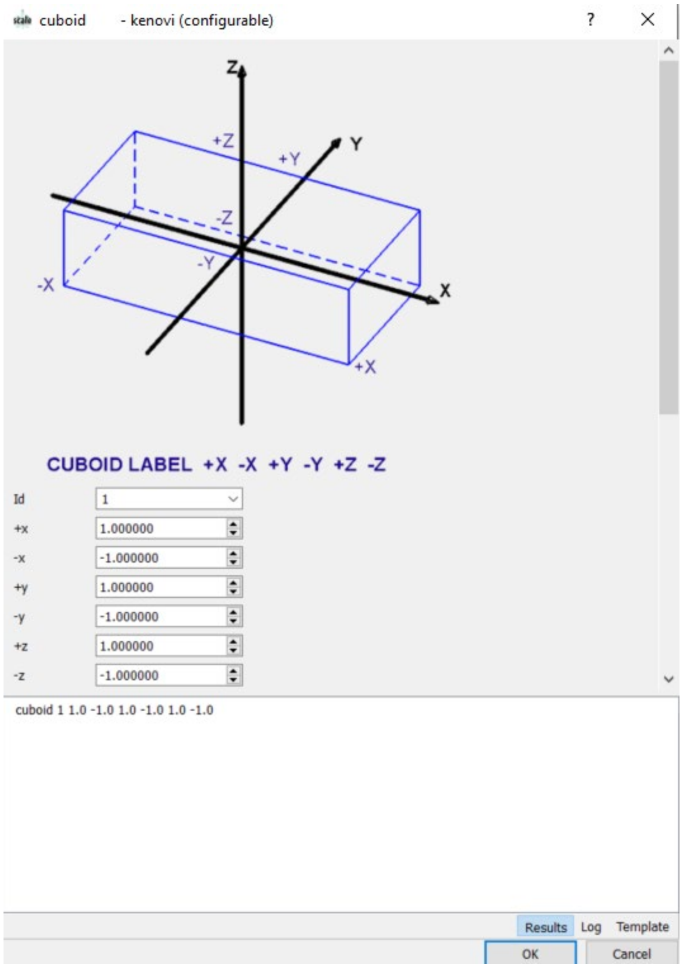
Figure 66. More KENO-VI geometry shapes.

These shapes are the basic geometry options in KENO-VI. Placing the text cursor within the `read geometry` block and pressing the `CTRL-SPACE` autocomplete key combination will display the list of shapes available (Figure 67). Each shape has a set of quantitative information that is needed to describe its size and location. Selecting a geometry option brings up the associated shape configurable input form. Figure 68 shows the input form for a cuboid, which is accessed by clicking on **cuboid – kenovi (configurable)**; the other input forms ask for similar information. A description of the options for each basic shape can be found in [see SCALE Manual section *KENO-VI Shape Descriptions*]. Additional information may be entered to translate (relocate the origin), rotate, or to insert a chord that describes a plane surface that truncates the shape. These options are discussed in Section 5.



```
cone          - kenovi (configurable)
cone
cuboid        - kenovi (configurable)
cuboid
cylinder      - kenovi (configurable)
cylinder
xcylinder
ycylinder
zcylinder
ecylinder     - kenovi (configurable)
ecylinder
dodecahedron  - kenovi (configurable)
dodecahedron
ellipsoid     - kenovi (configurable)
ellipsoid
hexprism      - kenovi (configurable)
hexprism
hopper        - kenovi (configurable)
hopper
parallelepiped - kenovi (configurable)
parallelepiped
pentagon      - kenovi (configurable)
pentagon
plane         - kenovi (configurable)
plane
xpplane
ypplane
zpplane
quadratic     - kenovi (configurable)
quadratic
rhexprism
rhomboid      - kenovi (configurable)
rhomboid
ring          - kenovi (configurable)
ring
sphere        - kenovi (configurable)
sphere
```

Figure 67. Geometry autocomplete form.



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

Id	1
+x	1.000000
-x	-1.000000
+y	1.000000
-y	-1.000000
+z	1.000000
-z	-1.000000

```
cuboid 1 1.0 -1.0 1.0 -1.0 1.0 -1.0
```

Figure 68. Cuboid input form.

4.2.1 Simple cylinder examples

Some sample problems are presented to illustrate how shape parameters are entered using Fulcrum. These problems are based on the pure ^{235}U metal material from Section 3.4.1. Various cylindrical configurations will be examined.

4.2.1.1 Bare Metal Cylinder along the Z-axis

The first example is locating a bare metal cylinder centered on the Z-axis as shown in Figure 69. Based on the knowledge gained in previous sections of this primer, open a new file and enter the title, cross section library, and material information. Note that all problems in this section use the CE ENDF/B-VII.1 library (ce_v7.1).

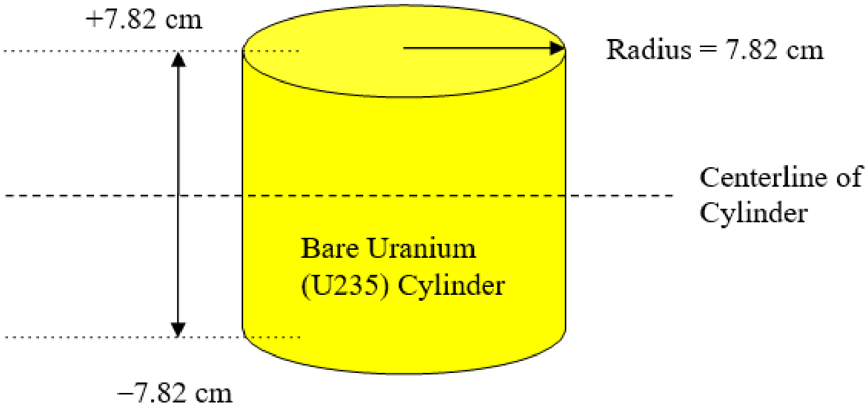
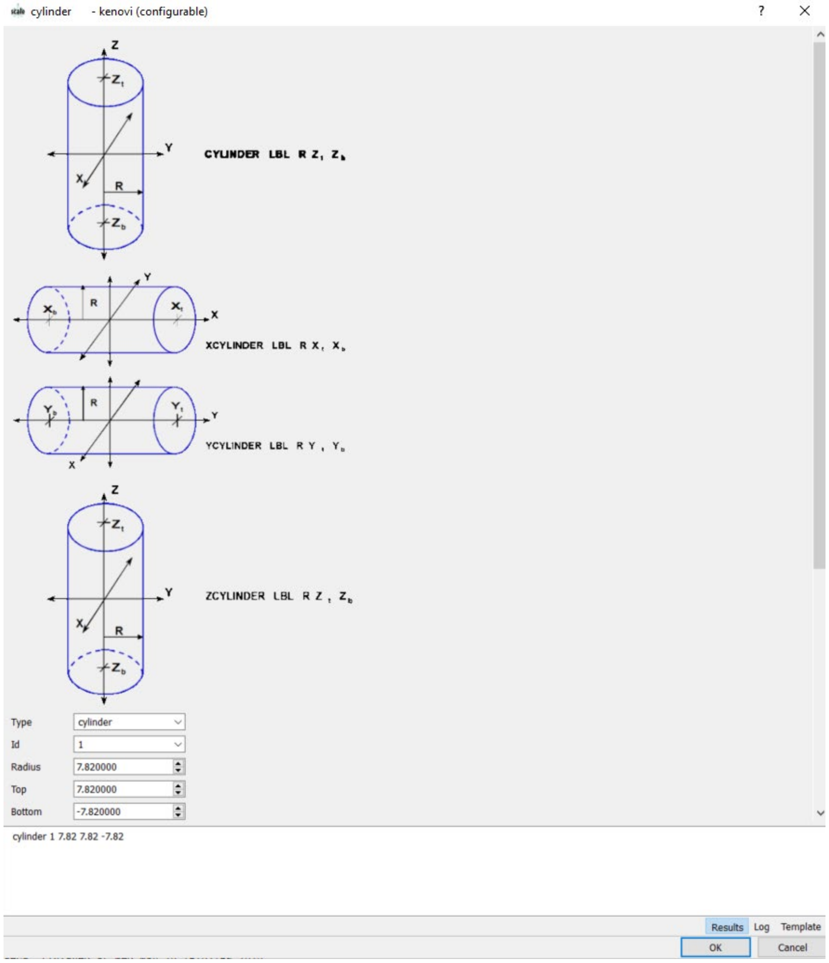


Figure 69. Bare ^{235}U cylinder geometry.

With this simple geometry, only a few entries are needed to complete the KENO geometry input form. Place the text cursor within the `read geometry` block, between `Global Unit 1` and `boundary 1` and press the `CTRL-SPACE` autocomplete key combination to display the list of available shapes shown in Figure 67. To enter geometry information, select the appropriate geometry from the list. For this problem, choose the **cylinder – kenovi (configurable)** button, which opens the **Cylinder** input form as shown in Figure 70. Enter `7.82` for **Radius**, `7.82` for **Top**, and `−7.82` for **Bottom**. This completes the **Cylinder** information, as shown in Figure 70. Click on **OK** to add the input generated to the input file and return to the main Fulcrum screen.



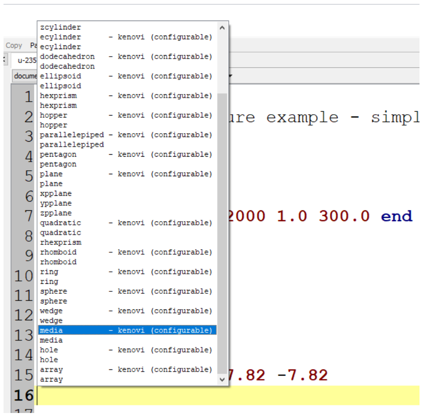
Type: cylinder
Id: 1
Radius: 7.820000
Top: 7.820000
Bottom: -7.820000

```
cylinder 1 7.82 7.82 -7.82
```

Figure 70. Cylinder data for bare metal cylinder along Z-axis.

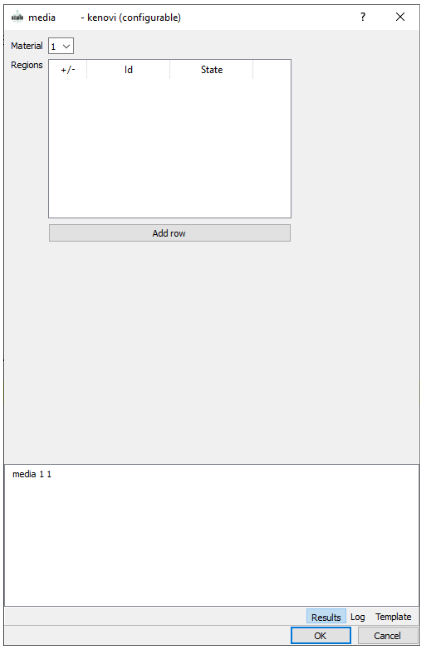
The contents of the cylinder must be described next. Now the contents of the regions must be defined using media records. Each media record defines one region in the unit by combining one or more shapes using Boolean logic in a region definition vector. The media record also identifies a mixture to fill the region. There is only one region in the problem, cylinder region 1, and one mixture, u-235 mixture 1. Below the cylinder input description, use the `CTRL-SPACE` autocomplete key combination to display the list of available shapes and scroll to the bottom and select **media - configurable**, as shown in Figure 71. The media configurable form generated by Fulcrum is shown in Figure 72. Click the **Add row** button below the **Regions** pane to add mixture 1 to region 1. The mixture number is selected in the **Material** dropdown in the upper left corner of the form. The Id column in the **Regions** pane is the Id number for the shape referenced in the media record, and the **State** is either **inside** or **outside**. For this problem, the default values are correct and, once the input has been confirmed by comparison with Figure 73, press the OK button to add the media record to the input file. The first numerical entry on the media record is the mixture number to be used (mixture number 1). The second is a Bias ID, but only unbiased tracking (Bias ID of 1) is recommended in KENO. The third number 1 is the Region Definition Vector: it states that this region is inside cylinder 1.

Alternatively, the media record could be added by typing directly in the Fulcrum editor (enter `media 1 1 1`). Do not enter the media record twice; enter it once by using either the configurable form or typing the text in the input file.


Copy
Pa
u-235
docume
zcylinder
ecylinder   - kenovi (configurable)
ecylinder
dodecahedron   - kenovi (configurable)
dodecahedron
ellipsoid   - kenovi (configurable)
ellipsoid
hexprism   - kenovi (configurable)
hexprism
hopper   - kenovi (configurable)
hopper
parallelepiped - kenovi (configurable)
parallelepiped
pentagon   - kenovi (configurable)
pentagon
plane   - kenovi (configurable)
plane
xpplane
ypplane
zpplane
quadratic   - kenovi (configurable)
quadratic
rhexprism
rhomboid   - kenovi (configurable)
rhomboid
ring   - kenovi (configurable)
ring
sphere   - kenovi (configurable)
sphere
wedge   - kenovi (configurable)
wedge
media   - kenovi (configurable)
media
hole   - kenovi (configurable)
hole
array   - kenovi (configurable)
array

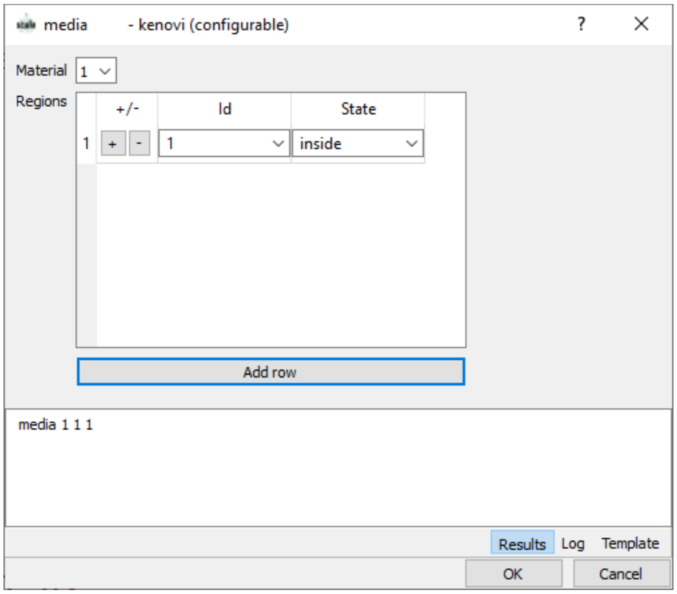
```
ure example - simpl
2000 1.0 300.0 end
7.82 -7.82
```

Figure 71. Selecting the media configurable window.



```
media 1 1
```

Figure 72. Default media configurable window in Fulcrum.



Material 1

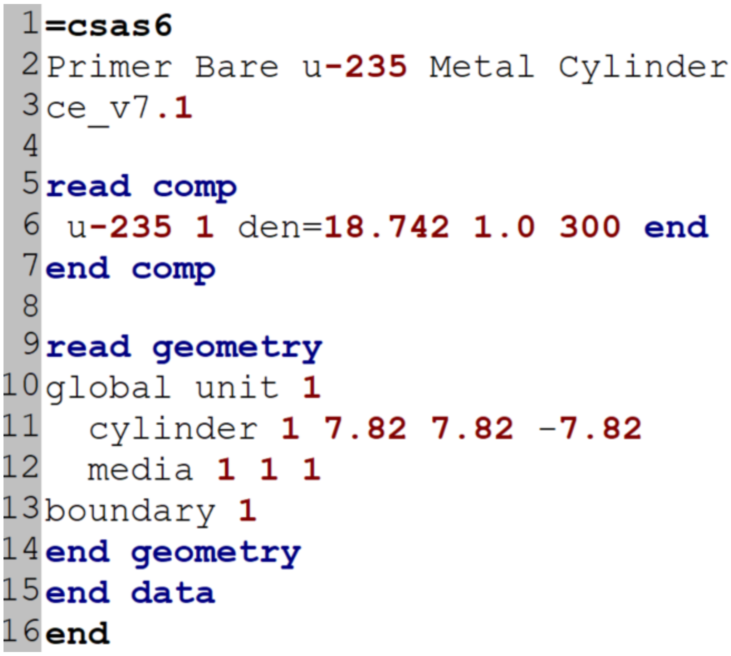
Regions

	+/-	Id	State
1	+ -	1	inside

```
media 1 1 1
```

Figure 73. Completed media configurable form.

The boundary record follows the media records, and it defines the outer boundary of the unit using a region definition vector. The outer boundary of the unit was already defined as `boundary 1` thanks to the Fulcrum autocompletion. This defines region 1 as the outer boundary of the unit. As stated earlier, the global unit in KENO is the final unit that defines the entire system to be analyzed. This completes the information required for this example. Select **Run**. Click on **messages** to allow Fulcrum to show the progress of the calculation. The input file for this example should look like Figure 74.



```
=csas6
Primer Bare u-235 Metal Cylinder
ce_v7.1

read comp
 u-235 1 den=18.742 1.0 300 end
end comp

read geometry
global unit 1
  cylinder 1 7.82 7.82 -7.82
  media 1 1 1
boundary 1
end geometry
end data
end
```

Figure 74. Input file for the bare metal cylinder example.

To make sure the input is correct, check the calculated value of k_{eff}. The value should be approximately 1.0224 ± 0.0022.

4.2.1.2 Bare Metal Cylinder along the Z-axis with Translated Origin

There are some fissile systems for which the cylinder origin is located at a point other than the center of the cylinder. Moving the origin up and down along the Z-axis is done by changing the values of **Top** and **Bottom**. For example, suppose the origin is at a point 2 cm above the base of the cylinder (see Figure 75).

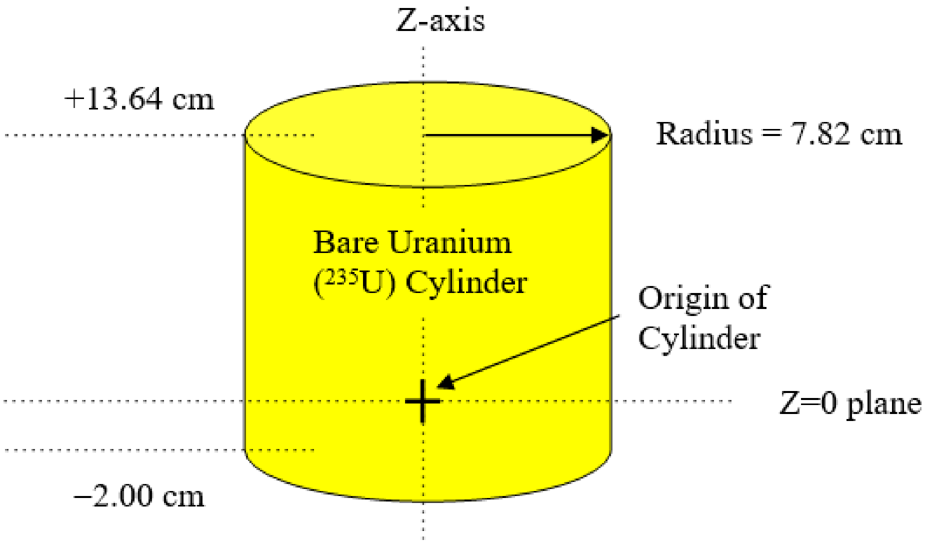
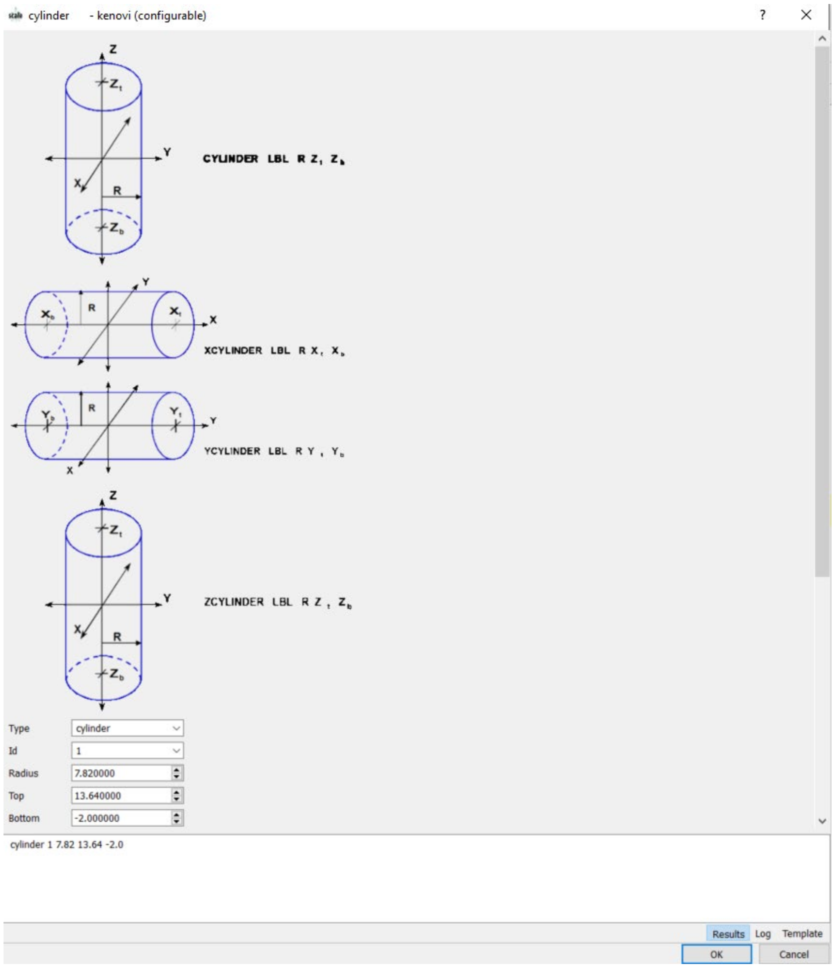


Figure 75. Base of cylinder located 2 cm below origin.

The cylinder region parameters on the **Cylinder** input form can be modified from the previous example. Remove the previous cylinder input line and create a new cylinder by placing the text cursor within the `read geometry` block, between `Global Unit 1` and `boundary 1`, and pressing the `CTRL-SPACE` autocomplete key combination. Select **cylinder – kenovi (configurable)**, which opens the **Cylinder** input form. Enter `7.82` for **Radius**, `13.64` for **Top**, and `−2` for **Bottom** so that the screen looks like Figure 76.


cylinder - kenovi (configurable)
? ×
Z
Z_t
Y
X
R
Z_b
CYLINDER LBL R Z_t Z_b
XCYLINDER LBL R X_t X_b
YCYLINDER LBL R Y_t Y_b
ZCYLINDER LBL R Z_t Z_b
Results
Log
Template
OK
Cancel

Type: cylinder

Id: 1

Radius: 7.820000

Top: 13.640000

Bottom: -2.000000

```
cylinder 1 7.82 13.64 -2.0
```

Figure 76. Cylinder data input with base located 2 cm below origin.

Save the changes in a new input file and execute SCALE. Because no changes were made to the physical system, the k_{eff} value for this run should be identical to that from the previous run.

4.3 KENO-VI GEOMETRY RULES

There are seven basic rules that guide the geometry input to SCALE/KENO-VI.

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 the KENO-VI geometry shapes. The unit boundary must fully enclose all defined regions in the unit. See the example provided in Figure 77.

3. *Regions may share boundaries and may intersect.* See examples in Figure 77. (Note that intersecting regions are not allowed in KENO V.a.)

4. *Regions may be rotated.* See example in Figure 77. (Note that rotations are not allowed in KENO V.a.)

5. *A hole is used to place a unit within a region in a different unit.* The hole must be completely contained within the region 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 size. Multiple arrays may be placed directly into a unit by placing them in separate regions. Arrays can also be placed within a unit using holes.

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.

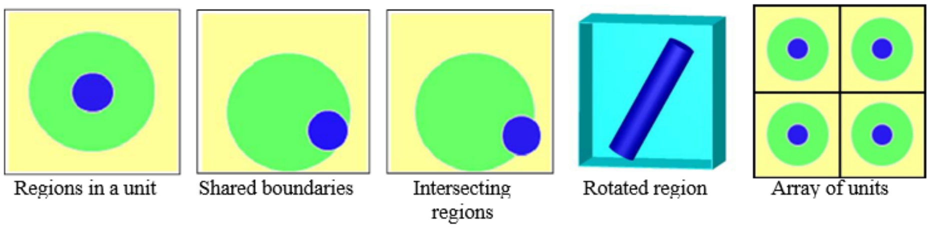


Figure 77. Examples of KENO basic geometry rules.

Note that each point in the geometry must be uniquely defined by one content record (`media`, `hole`, or `array`). If more than one media is assigned to any volume, then that volume is “multiply defined.” If a region is not assigned in any media record, then that volume is “undefined.” Either of these conditions will result in error messages and job termination when a particle is tracked through that volume in KENO-VI.

4.4 GEOMETRIC ARRANGEMENTS

Geometric arrangements in KENO-VI are achieved in a manner similar to using a child’s building blocks. Each building block is called a *unit*. Units are constructed of combinations of basic shapes. These shapes can be placed anywhere within a unit but will be truncated by the outer boundary of the unit if they intersect it or are completely outside of it. A unit can be thought of as a container that encloses a number of shapes. A unit encloses one or more basic shapes.

4.5 NESTED REGIONS IN A UNIT

To illustrate how to enter KENO geometry information for nested regions in a unit via Fulcrum, two sample problems are included. The first is a reflected cylinder of Pu metal, and the second is a sphere inside a cylinder inside a cube.

4.5.1 Reflected Pu Metal Cylinder

The reflected Pu metal cylinder problem comes from page 102 of LA-10860 [5]. The fissile material is Pu metal (density = 15.44 g/cc), with a maximum of 5.0 wt% ^{240}Pu. The remainder is ^{239}Pu. The core has a diameter of 5.72 cm with an H/D = 7.16, or a critical height of 40.96 cm. Surrounding the core is a graphite reflector (density = 1.60 g/cc) with a thickness of 17.78 cm in both the radial and axial directions (see Figure 78).

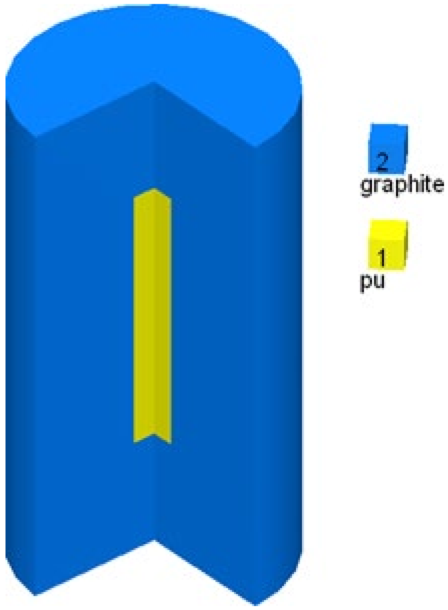
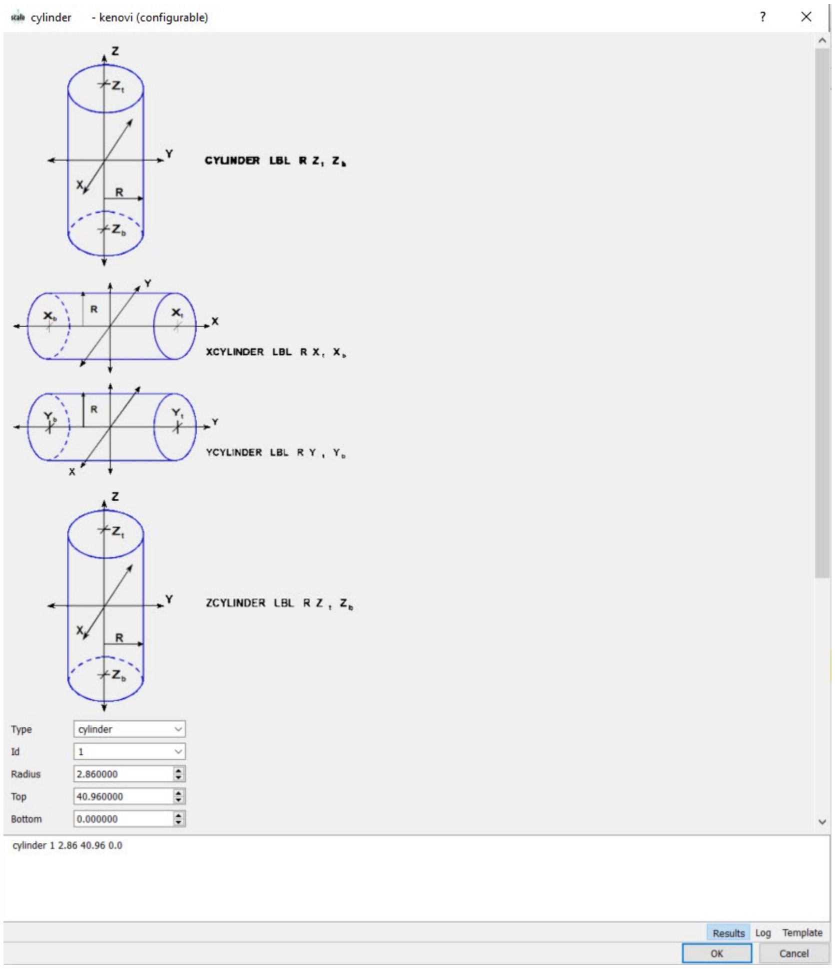


Figure 78. Reflected plutonium metal cylinder geometry.

Based on the knowledge gained in previous sections of this primer, open a new file and enter the title, cross section library, and material information. Specify the isotopic composition for mixture 1 as `pu` with `95` wt.% ^{239}Pu and `5` wt.% ^{240}Pu and a plutonium density of `15.44` g/cc. Mixture 2 should be `c-graphite` with a density of `1.6` g/cc.

For this geometry, there will be two shapes (or regions) in one unit, and that unit will be the global unit for the problem. The inner cylinder containing the plutonium will be described first, and the outer cylinder containing graphite will be described second. The order in which the regions are defined is irrelevant in KENO-VI, so this choice of the inner cylinder first is arbitrary. Place the text cursor within the `read geometry` block, between `Global Unit 1` and `boundary 1`, and press the `CTRL-SPACE` autocomplete key combination. Select **cylinder – kenovi (configurable)**, which opens the **Cylinder** input form. Enter `2.86` for **Radius**, `40.96` for **Top**, and 0 for **Bottom.** This puts the origin at the bottom center of the core cylinder. Check that the configurable form should look like that shown in Figure 79 and then click **OK.**


cylinder - kenovi (configurable)
?
×
Z
Z_t
Y
X
R
Z_b
CYLINDER LBL R Z_t Z_b
Y
R
X_b
X_t
X
XCYLINDER LBL R X_t X_b
R
Y_b
Y_t
Y
X
YCYLINDER LBL R Y_t Y_b
Z
Z_t
Y
X
R
Z_b
ZCYLINDER LBL R Z_t Z_b
Results
Log
Template
OK
Cancel

Type	cylinder
Id	1
Radius	2.860000
Top	40.960000
Bottom	0.000000

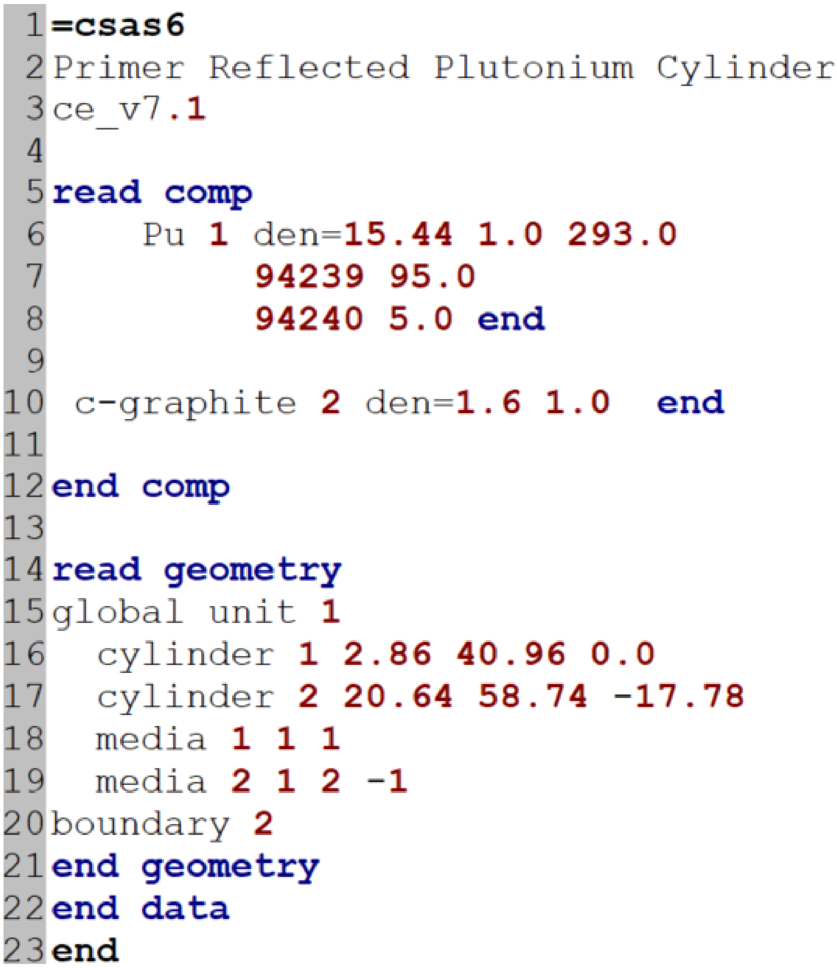
```
cylinder 1 2.86 40.96 0.0
```

Figure 79. Geometry description for region 1.

To enter the data for the reflector, place the text cursor within the `global unit`, and press the `CTRL-SPACE` autocomplete key combination. Select **cylinder – kenovi (configurable)**, which opens another **Cylinder** input form. Change the **Id to** `2`, as the reflector is the second region of the problem. As with the order of entry, the ID numbers for the regions are arbitrary, though in this case must be positive integers. Enter `20.64` (= 2.86 + 17.78) for **Radius**, `58.74` (= 40.96 + 17.78) for **Top**, and `–17.78` (= 0.00 – 17.78) for **Bottom**. Then click **OK.**

The geometry shapes have been defined, so the media records must now be added. Use the `CTRL-SPACE` autocomplete key combination and select **media - kenovi (configurable)**. Use the configurable form to specify that region 1 is filled with plutonium. Repeat this process to specify that the graphite is inside cylinder 2 but outside cylinder 1. Cylinder 1 must be excluded to preclude doubly-defined space with the cylinder containing both plutonium and graphite.

The final step is to define the outer boundary of unit 1, which was already defined thanks to the Fulcrum autocompletion as `boundary 1.` As the outer region is the reflector, region 2, change to `boundary 2` on the line after the media definition. This completes the geometry input (see Figure 80).



```
=csas6
Primer Reflected Plutonium Cylinder
ce_v7.1

read comp
    Pu 1 den=15.44 1.0 293.0
         94239 95.0
         94240 5.0 end

 c-graphite 2 den=1.6 1.0  end

end comp

read geometry
global unit 1
  cylinder 1 2.86 40.96 0.0
  cylinder 2 20.64 58.74 -17.78
  media 1 1 1
  media 2 1 2 -1
boundary 2
end geometry
end data
end
```

Figure 80. Input file for the reflected plutonium metal cylinder example.

Save the changes in a new input file and execute SCALE. The calculated k_{eff} value is 0.9790 ± 0.0021.

4.5.2 ^{235}U sphere with Graphite and Water Reflectors

This problem consists of a metal ^{235}U sphere inside a graphite cylinder inside a cube of water. The center of the cylinder is offset so that the outside edge of the cylinder touches the -X, -Y, and -Z faces of the cube (see Figure 81). The materials are ^{235}U metal (density = 18.74 g/cc), graphite (density = 1.65 g/cc), and water (default density of 0.9982 g/cc). The ^{235}U sphere radius is 7.0 cm; the graphite cylinder has a radius of 10 cm and a height of 20 cm. The water cube surrounding the cylinder is 22 cm on a side. The bottom of the cylinder is coincident with the bottom of the cube.

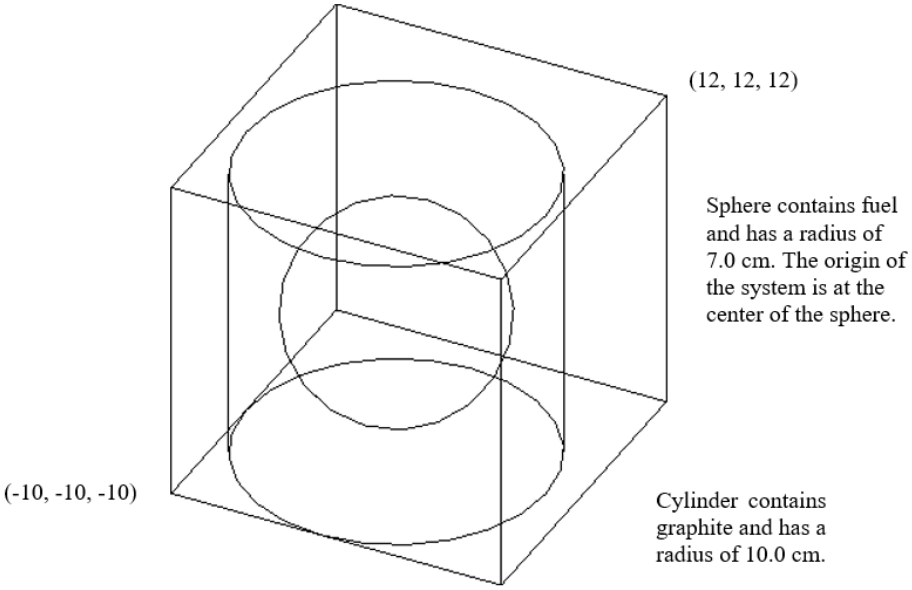

(12, 12, 12)
Sphere contains fuel and has a radius of 7.0 cm. The origin of the system is at the center of the sphere.
(-10, -10, -10)
Cylinder contains graphite and has a radius of 10.0 cm.

Figure 81. ^{235}U sphere surrounded by graphite and water.

Two cutaway views of the geometry are shown in Figure 82. The top 3 cm has been removed from both views to show the graphite cylinder and the sphere inside the water. In the first view, a quarter cut has been removed to reveal the ^{235}U sphere inside the graphite. In the second view, the graphite has been hidden so that the entire ^{235}U sphere can be seen.

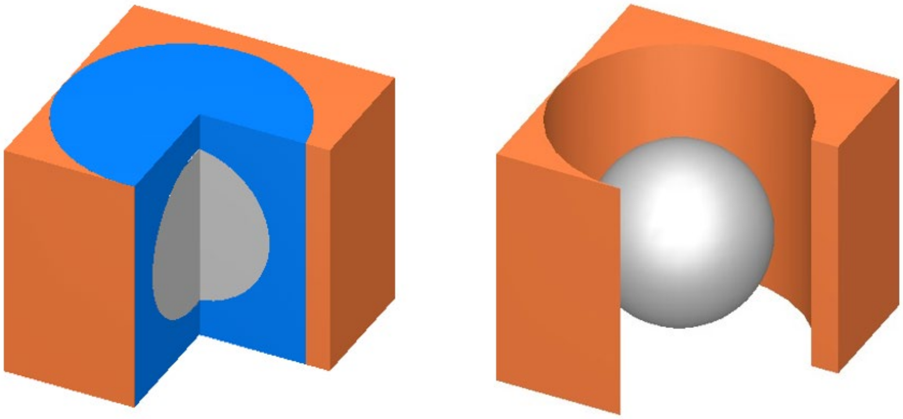


Figure 82. Cutaway views of ^{235}U sphere inside graphite and water.

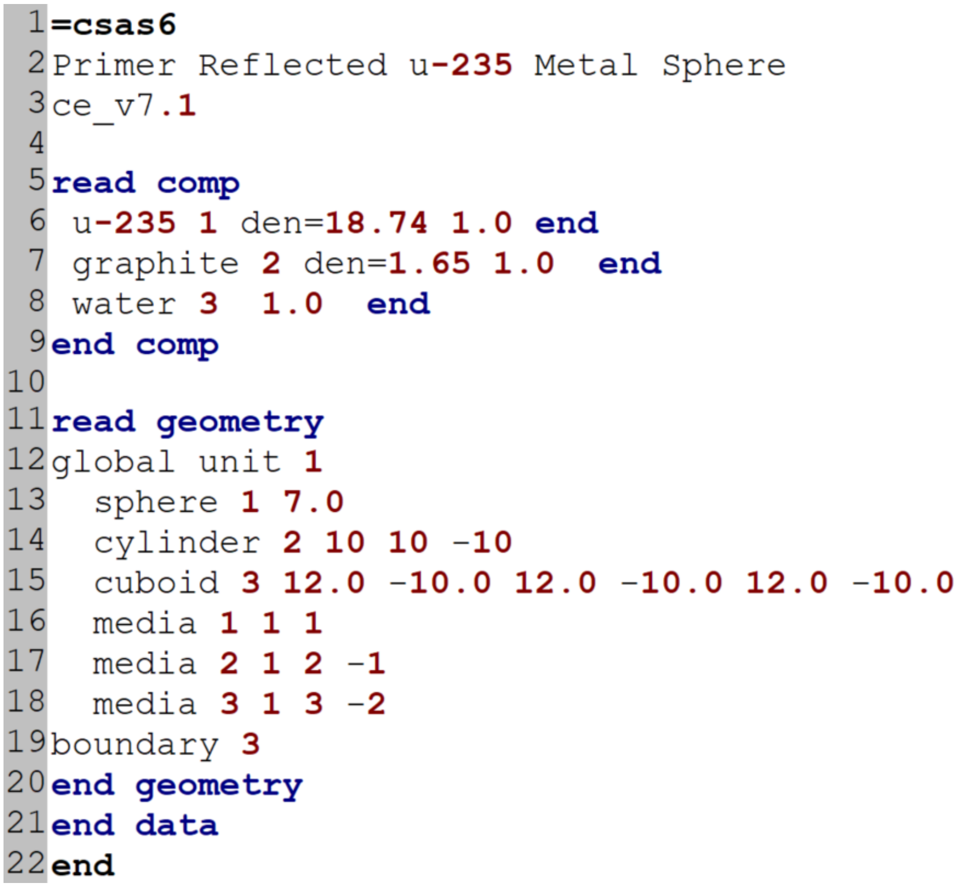
Using knowledge from previous sections, open a new file and enter the title, cross section library, and material information for the three materials.

For this geometry, there will be three regions in one unit, and that unit will be the global unit for the problem. The uranium sphere will be described first, then the graphite cylinder, and finally, the outer cube containing water. These are nested regions and are entered like those in the previous example. Place the text cursor within the `read comp` block and press the `CTRL-SPACE` autocomplete key combination. Select **sphere – kenovi (configurable)**, which opens the **Sphere** input form. Enter `7` for **Radius**, and then press **OK.** To enter the data for the graphite reflector, place your text cursor within the `read geometry` block, and press the `CTRL-SPACE` autocomplete key combination. Select **cylinder– kenovi**

(configurable), which opens the **Cylinder** input form. Change the **Id** to `2,` as the reflector is the 2nd region of the problem. Enter `10` for **Radius**, `10` for **Top**, and `-10` for **Bottom**. This creates a cylinder centered around the sphere. Click **OK.** To enter the data for the outside water reflector, place the text cursor within the `read comp` block and press the `CTRL-SPACE` autocomplete key combination. Select **cuboid– kenovi (configurable)**, which opens the **Cuboid** input form. Change the **Id** to `3,` as the reflector is the 3rd region of the problem. The cuboid is slightly off center from the cylinder and sphere, so entering `12` for **+X, +Y,** and **+Z**, and`-10` for **−X, −Y,** and **−Z** creates a cube with three sides touching the cylinder and the other three sides 2 cm beyond the cylinder.

The geometry shapes have been defined. Now the contents of the regions must be defined using media records. Use the media configurable form to specify that ^{235}U is inside region 1. Repeat these steps to create the second region containing the graphite cylinder. For this region, the Region Definition Vector is `2 -1`, which indicates that the graphite is inside cylinder 2 but outside sphere 1. Repeat these steps again to create the third region containing the water. For this region, the Region Definition Vector is `3 -2`, which indicates that the water is inside cuboid 3 but outside cylinder 2.

The final step is to define the outer boundary of unit 1, which was already defined thanks to the Fulcrum autocompletion as `boundary 1`. As the outer region is the water cuboid, region 3, change to `boundary 3` on the line after the media definition. This completes the geometry input (see Figure 83). Save the changes in a new input file. Click **Run** to execute SCALE. The calculated k_{eff} value is 1.0031 ± 0.0017.



```
=csas6
Primer Reflected u-235 Metal Sphere
ce_v7.1

read comp
 u-235 1 den=18.74 1.0 end
 graphite 2 den=1.65 1.0   end
 water 3  1.0  end
end comp

read geometry
global unit 1
  sphere 1 7.0
  cylinder 2 10 10 -10
  cuboid 3 12.0 -10.0 12.0 -10.0 12.0 -10.0
  media 1 1 1
  media 2 1 2 -1
  media 3 1 3 -2
boundary 3
end geometry
end data
end
```

Figure 83. Input file for the ^{235}U sphere in graphite and water example.

4.6 ARRAYS

An array is constructed by stacking units. Each unit in an array or lattice has its own coordinate system; however, all coordinate systems in all units must have the same orientation. All geometry data used in a problem are correlated to the absolute coordinate system by specifying a global unit.

Arrays are created by stacking units that have an outer boundary that matches the array type in which they will be used. For example, the unit must have a cuboid as an outer boundary to be used in a cuboidal array. The size of adjacent faces of adjacent units stacked in this manner must match exactly. See Figure 84 for a typical example.

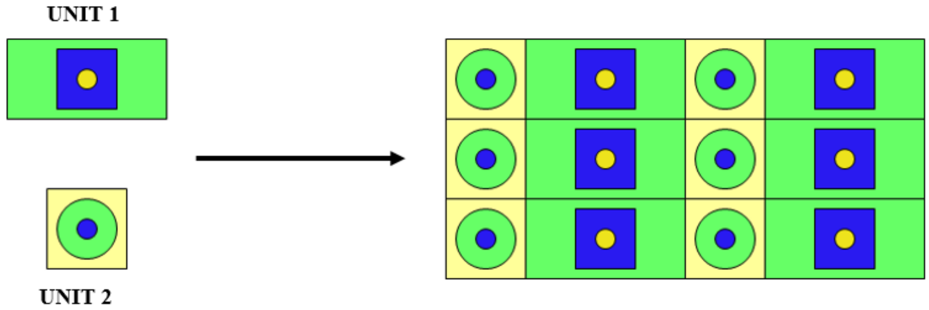
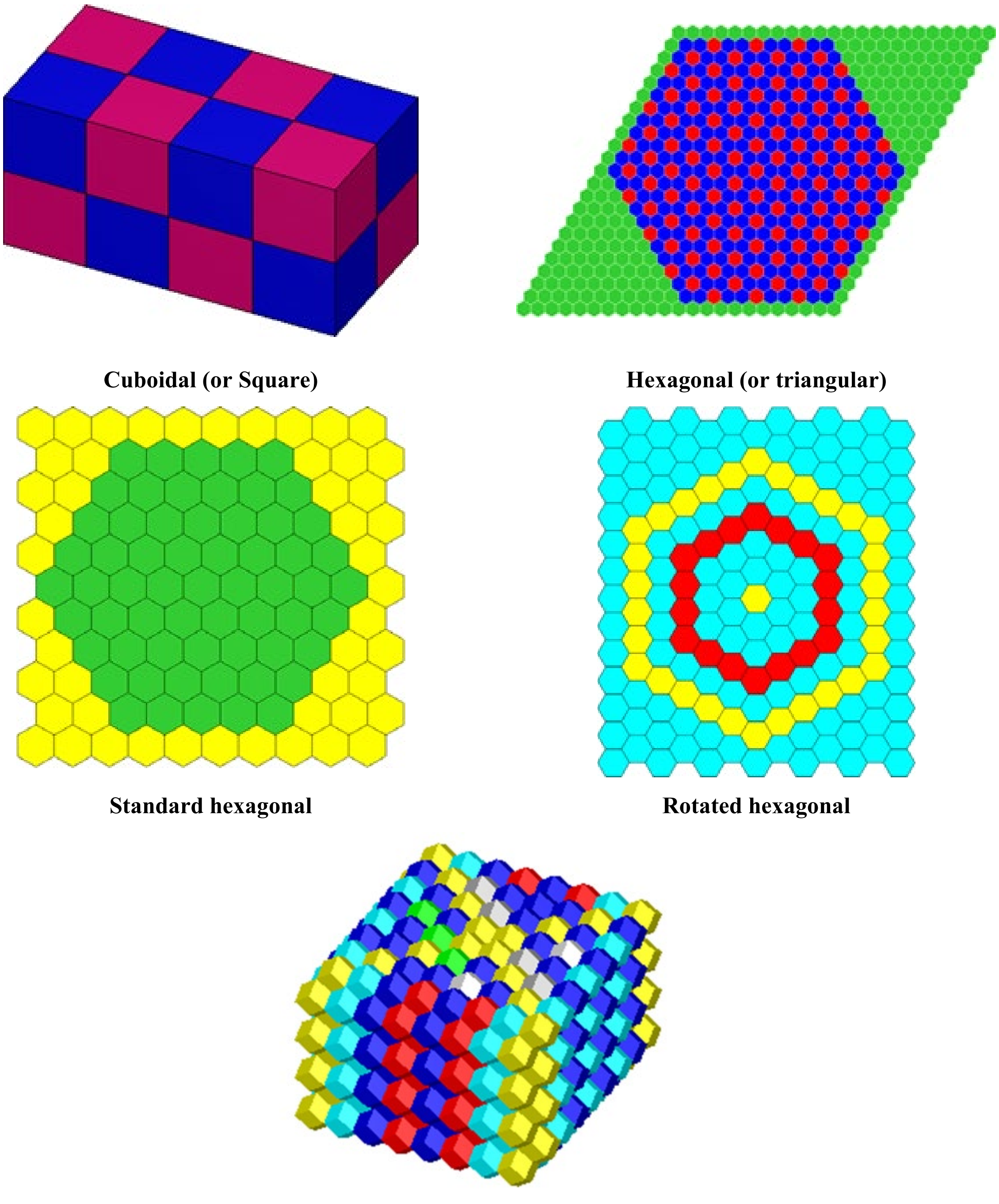


Figure 84. Example of array construction.

The ARRAY option is provided to allow placing an array within a unit. Arrays of dissimilar units can be created in some cases by stacking units that contain arrays (i.e., “arrays of arrays”).

KENO-VI offers five different 3D array types, as shown in Figure 85:

- cuboidal (or square),
- hexagonal (or triangular),
- standard hexagonal,
- rotated hexagonal, and
- dodecahedral.



Cuboidal (or Square)

Hexagonal (or triangular)

Standard hexagonal

Rotated hexagonal

Dodecahedral

Figure 85. KENO-VI array types.

4.6.1 Arrays with Single Units

This problem consists of a 2 × 2 × 2 array of bare metal U(93.2) cylinders. Each cylinder has a radius of 5.748 cm and an outside height of 10.765 cm. The uranium has a density of 18.76 g/cc with 1 wt.% ^{234}U, 93.2 wt.% ^{235}U, 0.2 wt.% ^{236}U, and 5.6 wt.% ^{238}U. The pitch in the X and Y directions is 13.74 cm, and it is 13.01 in the Z direction. There are no external reflectors or other materials in the array. The geometry is shown in Figure 86.

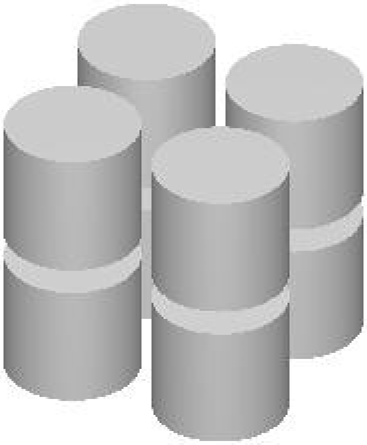


Figure 86. 3D view of 2 × 2 × 2 array.

As mentioned previously, the outside shape of a unit must be a cuboid, a hexprism, or a dodecahedron to be placed in an array. As the constituents of this array are cylinders, they must be enclosed by cuboids to become part of a rectangular array. The dimensions of the cuboid are usually determined by the X-, Y-, and Z-pitch dimensions. For an array of single elements, the cuboid will have an X dimension equal to the pitch in the X direction, a Y dimension equal to the pitch in the Y direction, and a Z dimension equal to the pitch in the Z direction.

Using knowledge from previous sections, open a new file and enter the title, cross section library, and material information for the uranium, including the isotopic composition weight percentages.

This geometry has two regions in one unit, but that unit is not the global unit for the problem. The global unit for this problem will contain the array of eight units. The uranium cylinder will be described first, and then the cuboid enclosing the cylinder to create an element of the array. These are nested shapes and are entered in the same manner as for the previous examples. However, be careful when entering the Z data for both the cylinder and the cuboid. Once the Z or height information has been entered for the cylinder, the coordinate system and its origin will be fixed. There are three obvious choices for the location of the origin: the bottom center of the cylinder, the middle of the cylinder, or the top center of the cylinder. In some problems, one origin location may make data input for other shapes in a unit easier than other origin locations.

For this problem, choose the origin to be located at the middle of the cylinder, to minimize the arithmetic needed to center the cylinder in the cuboid. This is a frequent choice for a unit which will be used multiple times in an array. The **Radius** for the cylinder will be `5.748`, the **Top** will be `5.3825`, and the **Bottom** will be `-5.3825` (see Figure 79). For the cuboid, the origin is located in the middle in the X, Y, and Z directions, so the dimensions will be half the pitch: **+X** and **+Y** will be `6.87,` **–X** and **–Y** will be `–6.87,` **+Z** will be `6.505`, and **–Z** will be `-6.505`. These parameters are sufficient for describing the single array element as shown in Figure 87.

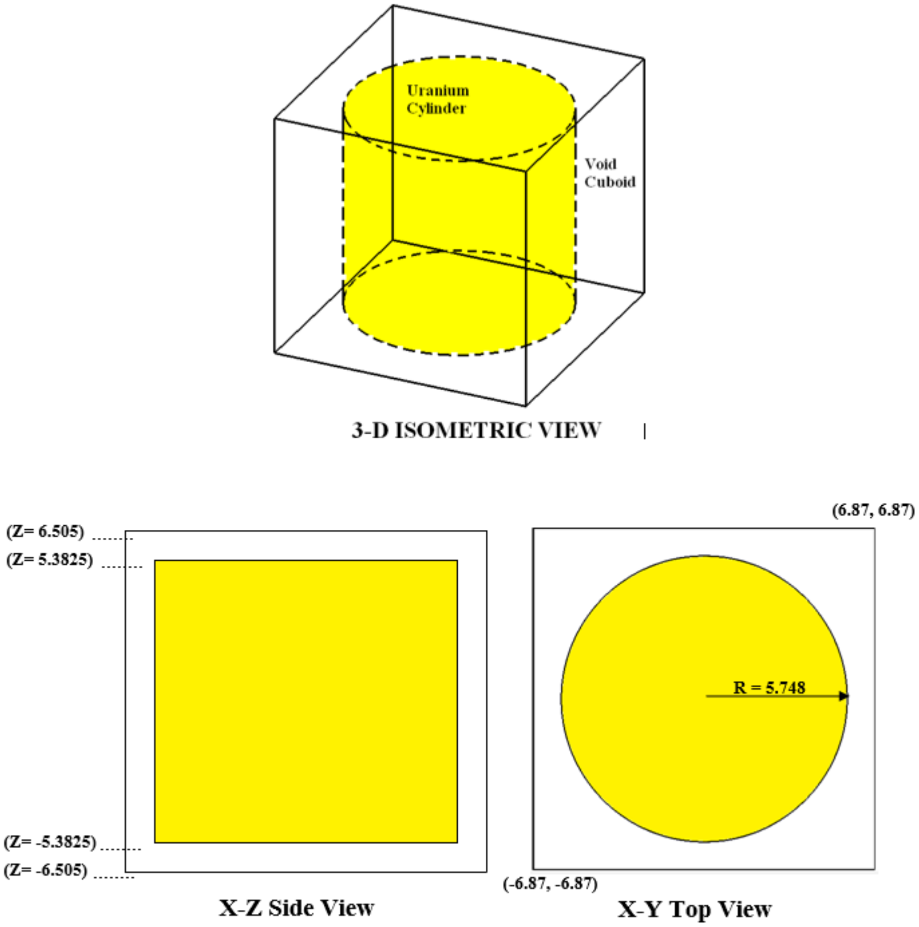
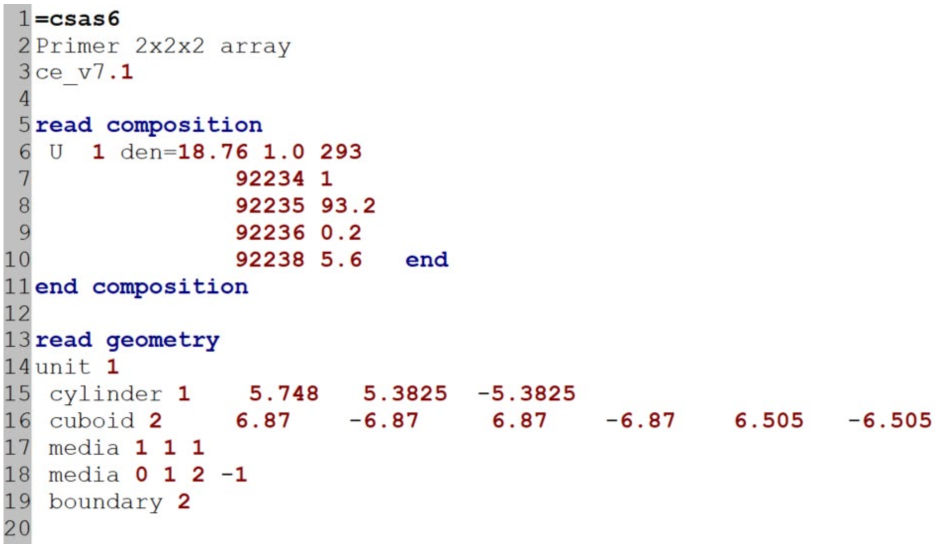


Figure 87. Single array element (cuboid enclosing a cylinder).

Start writing the geometry by placing the text cursor within the `read geometry` block, between `Global Unit 1` and `boundary 1`, and press the `CTRL-SPACE` autocomplete key combination. Select **cylinder– kenovi (configurable)**, which opens the **Cylinder** input form. Enter `5.748` for **Radius**, `5.3825` for **Top**, and `-5.3825` for **Bottom**. This puts the origin at the center of the cylinder; this is now the origin for the unit and for all other regions in the unit. To enter the data for the enclosing cuboid, place the text cursor within the `read comp` block and press the `CTRL-SPACE` autocomplete key combination. Click on the **cuboid– kenovi (configurable)** button, which opens the **Cuboid** input form, to describe the void cuboid. Change the **Id** to `2`. Enter `6.87` for **+X** and **+Y**, `-6.87` for **−X** and **-Y**, `6.505` for **+Z**, and `-6.505` for **−Z**. This creates a cuboid around the cylinder with the proper spacing. Before proceeding with the media definition, note that unit 1 will be placed inside an array, so it is not the global unit of the problem. To tell Fulcrum so, remove `Global` from `Global unit 1`.

Use two media records to define the two regions (uranium inside the cylinder and void (mixture 0) inside the cuboid and outside the cylinder). The final step is to define the outer boundary of unit 1. The outer boundary of the unit was already defined thanks to the Fulcrum autocompletion as `boundary 1`. As the outer region is the void cuboid, region 2, change to `boundary 2` on the line after the media definition. When finished, the composition and geometry blocks should look like Figure 88. The **Array** data will be entered next.

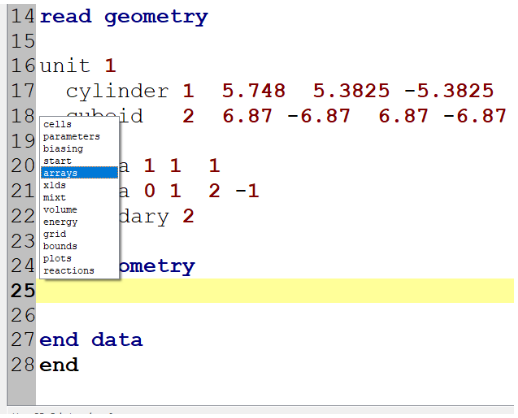


```
=csas6
Primer 2x2x2 array
ce_v7.1

read composition
 U  1 den=18.76 1.0 293
              92234 1
              92235 93.2
              92236 0.2
              92238 5.6   end
end composition

read geometry
unit 1
 cylinder 1    5.748   5.3825  -5.3825
 cuboid 2     6.87    -6.87     6.87    -6.87    6.505   -6.505
 media 1 1 1
 media 0 1 2 -1
 boundary 2

```

Figure 88. Single unit geometry for array.

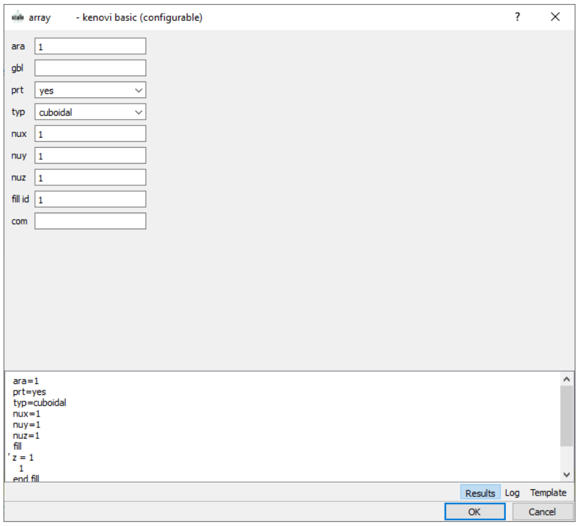
Arrays are defined in a separate block in the input file. Each array is specified with an integer ID, the type of array, the number of elements in the X, Y, and Z directions, and the units that fill the array. The integer ID is used to reference the array when it is placed into a unit in the geometry block. The type of the array, as discussed above, is one of the five array types supported in KENO-VI: cuboidal, hexagonal, standard hexagonal, rotated hexagonal, or dodecahedral. The outer boundary of the units to be used in this array must match the type of the array. The fill data provides the unit numbers to be placed into the array in the order that they will be used. The unit numbers are read in order of increasing X, then increasing Y, then increasing Z. This order will be demonstrated in this example.

To start describing the array, place the cursor on the line after `end geometry` and use the `CTRL-SPACE` autocomplete key combination to generate a list of available blocks to add to the input, as shown in Figure 89. Within the empty array block, use the `CTRL-SPACE` autocomplete key combination to open the configurable array form shown in Figure 90. This array is the first and only array in the problem, so accepting the default entry of 1 in the **ara** field is recommended. The defaults are also acceptable for **gbl** (there is no global array in this problem), **prt** (print a description of this array in the output), and **typ** (this array is cuboidal). The array is 2 units in the X direction, 2 units in the Y direction, and 2 units in the Z direction, so change **nux**, **nuy**, and **nuz** to `2`. Now that the 2 × 2 × 2 array is defined, it must be filled with units. In this problem, there is only unit 1. The **fill id** field specifies the unit with which to initialize the fill data, so the default of **1** is appropriate here as well. The order of the units stored in the array using the FILL parameter 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), as shown in Figure 91. Check that the array configurable form matches the example provided in Figure 92, then press OK to add the array specification to the input file. The completed array block is shown in the input in Figure 93.



```
read geometry

unit 1
  cylinder 1  5.748  5.3825 -5.3825
  cuboid    2  6.87 -6.87  6.87 -6.87

          a 1 1  1
          a 0 1  2 -1
          dary 2

          ometry


end data
end
```

Figure 89. Adding array block to the input with autocomplete.



ara	1
gbl	
prt	yes
typ	cuboidal
nux	1
nuy	1
nuz	1
fill id	1
com	

```
ara=1
prt=yes
typ=cuboidal
nux=1
nuy=1
nuz=1
fill
'z = 1
  1
end fill
```

Figure 90. Default array configurable form in Fulcrum.

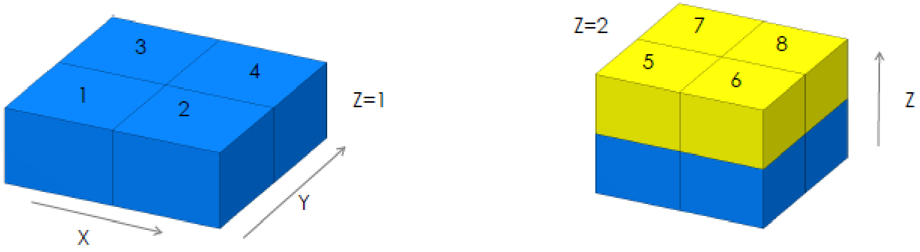
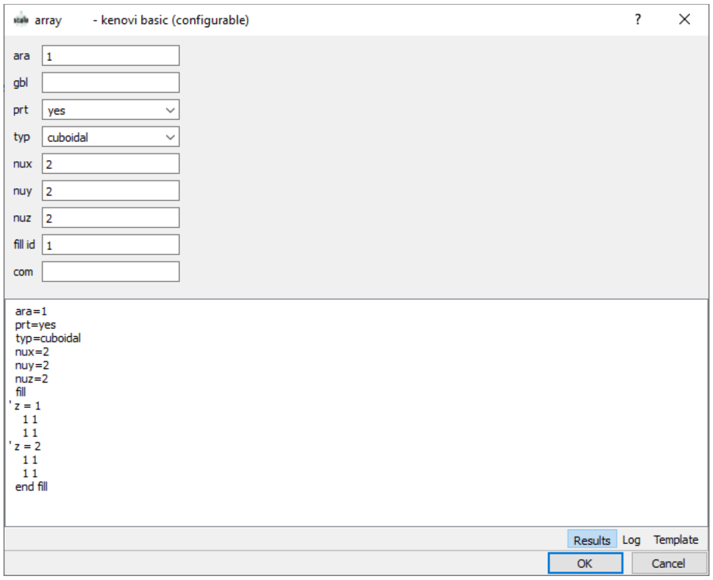


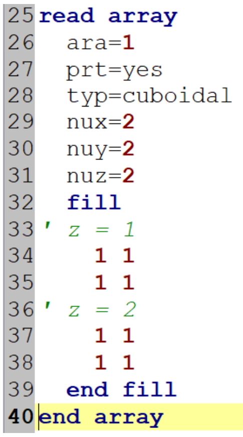
Figure 91. Unit filling order in a basic rectangular array.



ara	1
gbl	
prt	yes
typ	cuboidal
nux	2
nuy	2
nuz	2
fill id	1
com	

```
ara=1
prt=yes
typ=cuboidal
nux=2
nuy=2
nuz=2
fill
' z = 1
  1 1
  1 1
' z = 2
  1 1
  1 1
end fill
```

Figure 92. Completed array configurable form for this example.

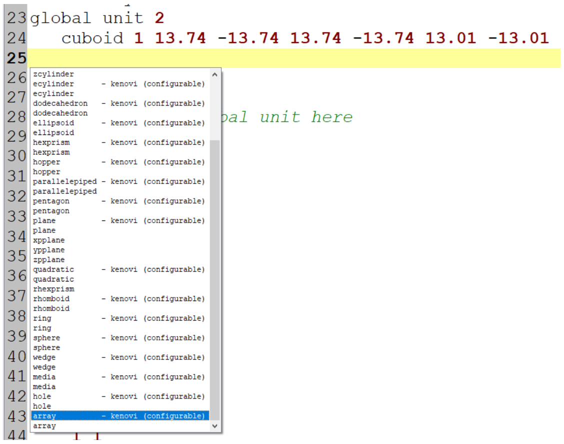


```
read array
  ara=1
  prt=yes
  typ=cuboidal
  nux=2
  nuy=2
  nuz=2
  fill
' z = 1
    1 1
    1 1
' z = 2
    1 1
    1 1
  end fill
end array
```

Figure 93. Completed array block for this example.

Return to the geometry definition block and use the `CTRL-SPACE` autocomplete key combination to add a global unit, then change the unit number to `2`. This unit will be the global unit that contains array 1. Position the text cursor in the global unit and press the `CTRL-SPACE` autocomplete key combination. Select the **cuboid– kenovi (configurable)**, which opens the **Cuboid** input form. Since the cuboid must contain array 1, the dimensions of the cuboid should be the same as those of the array. Because this is a 2 × 2 × 2 array, simply double the X, Y, and Z dimensions of unit 1. If we center the cuboid on the origin, then the outer dimensions of the cuboid will be `±13.74` for **±X** and ±**Y** and `±13.01` for **±Z**. After the dimensions are entered, press the **OK** button to close the cuboid form.

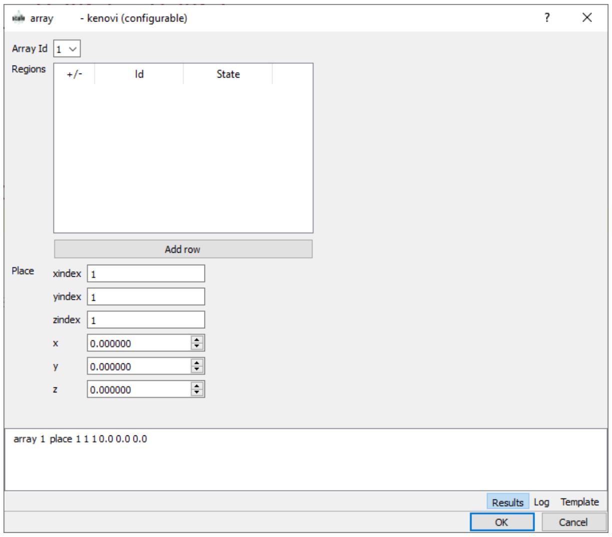
The last step in the array definition is to define the placement of array 1 in the cuboid. Similar to the media record, the ARRAY Contents Record must be used to place an array into a region defined in the current unit. To do so, use the `CTRL-SPACE` autocomplete key combination within the global unit and select **array - kenovi (configurable)** as shown in Figure 94. The default array configurable form is shown in Figure 95. The **Array Id** dropdown in the upper left corner of the form selects the array to be added to the geometry; in this case, the default value of **1** is used. The **Regions** pane allows the specification of the Region Definition Vector defining the region of space the array will fill. The data entry in this pane is the same as in the media record configurable form discussed above. Press the **Add row** button to start entering the Region Definition Vector; the array will be placed inside of region 1 so the default information can be accepted. The last section of the input is used to position the array within the specified region. The first part of the **Place** input selects the element in the array to be used for specifying this position; in this example the **xindex**, **yindex**, and **zindex** of 1 is as easy as any other element in the array. In some cases, there are particular elements of the array, like a central one, that are more useful than others for easily and correctly positioning an array. The **x**, **y**, and **z** coordinates specify the position of the origin of the selected unit in the current unit coordinate system. For this example, only unit 1 is used and the origin is at the center of the unit. The origin is thus 6.87 cm from the -X and -Y faces of the unit and 6.505 cm from the -Z face. The location of these negative surfaces of cuboid 1 have been specified as -13.74 in X and Y and -13.01 in Z, so the position of the origin can be determined. For X and Y, -13.74 + 6.87 = -6.87 and for Z -13.01 + 6.505 = -6.505. Therefore, enter `-6.87` for **x** and **y** in the array configurable and `-6.505` for **z**. This completes the array configurable form and, after confirming that it matches that shown in Figure 96, the array contents record can be added to the input file by pressing the **OK** button.



```
global unit 2
   cuboid 1 13.74 -13.74 13.74 -13.74 13.01 -13.01
```

zcylinder
ecylinder - kenovi (configurable)
ecylinder
dodecahedron - kenovi (configurable)
dodecahedron
ellipsoid - kenovi (configurable)
ellipsoid
hexprism - kenovi (configurable)
hexprism
hopper - kenovi (configurable)
hopper
parallelepiped - kenovi (configurable)
parallelepiped
pentagon - kenovi (configurable)
pentagon
plane - kenovi (configurable)
plane
xpplane
ypplane
zpplane
quadratic - kenovi (configurable)
quadratic
rhexprism
rhomboid - kenovi (configurable)
rhomboid
ring - kenovi (configurable)
ring
sphere - kenovi (configurable)
sphere
wedge - kenovi (configurable)
wedge
media - kenovi (configurable)
media
hole - kenovi (configurable)
hole
array - kenovi (configurable)
array
bal unit here

Figure 94. Autocomplete menu for array contents record.



array - kenovi (configurable)

Array Id: 1

Regions

+/-	Id	State

Place

xindex: 1

yindex: 1

zindex: 1

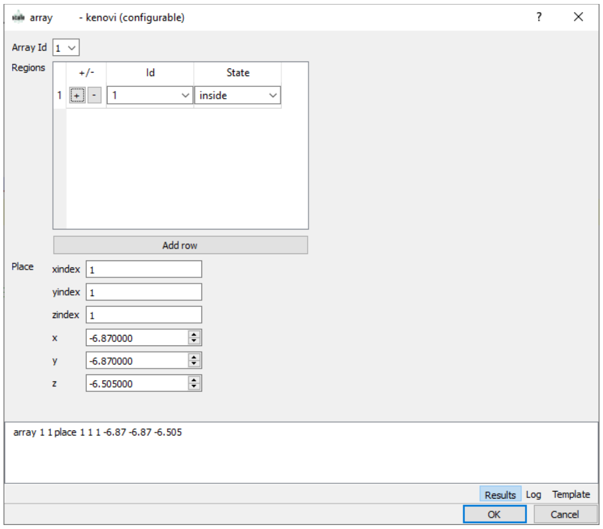
x: 0.000000

y: 0.000000

z: 0.000000

```
array 1 place 1 1 1 0.0 0.0 0.0
```

Figure 95. Default array configurable form.



Array Id 1

Regions

	+/-	Id	State
1	+ -	1	inside

Place

xindex 1

yindex 1

zindex 1

x -6.870000

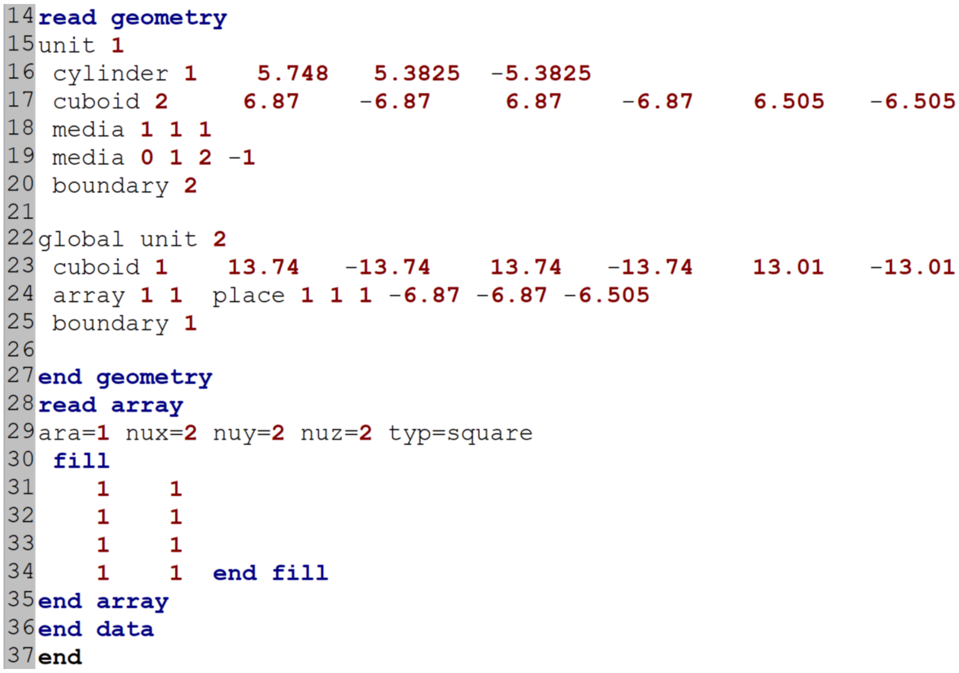
y -6.870000

z -6.505000

```
array 1 1 place 1 1 1 -6.87 -6.87 -6.505
```

Figure 96. Completed array contents record configurable form.

The final step is to define the outer boundary of unit 2. The outer region is the cuboid, region 1. Enter `boundary 1` on the line after the array definition. The geometry and array blocks should look like Figure 97. Save the input file and execute SCALE. The calculated k_{eff} for this critical experiment is 0.9991 ± 0.0017.



```
read geometry
unit 1
 cylinder 1    5.748   5.3825  -5.3825
 cuboid 2     6.87    -6.87     6.87    -6.87     6.505   -6.505
 media 1 1 1
 media 0 1 2 -1
 boundary 2

global unit 2
 cuboid 1    13.74   -13.74    13.74   -13.74    13.01   -13.01
 array 1 1  place 1 1 1 -6.87 -6.87 -6.505
 boundary 1

end geometry
read array
ara=1 nux=2 nuy=2 nuz=2 typ=square
 fill
    1    1
    1    1
    1    1
    1    1  end fill
end array
end data
end
```

Figure 97. Geometry form with array record to place array in geometry.

4.6.2 Arrays with Multiple Units of Different Sizes

This problem consists of a stack of six cylindrical disks covered by a square plate on the top (Figure 98). There are two types of disks; one containing graphite (density = 1.65 g/cc) that is 10 cm in radius and 2.5 cm thick, and the other containing U (93.2) that is 10 cm in radius and 4 cm thick. The uranium has a density of 18.74 g/cm^3, with 1 wt.% ^{234}U, 93.2 wt.% ^{235}U, 0.2 wt.% ^{236}U, and 5.6 wt.% ^{238}U. Progressing upward from the bottom, the disks are graphite, uranium, graphite, uranium, graphite, and uranium. The top plate is aluminum of nominal density, 30 × 30 cm square and 3 cm thick. There are no external reflectors or other materials in the array.

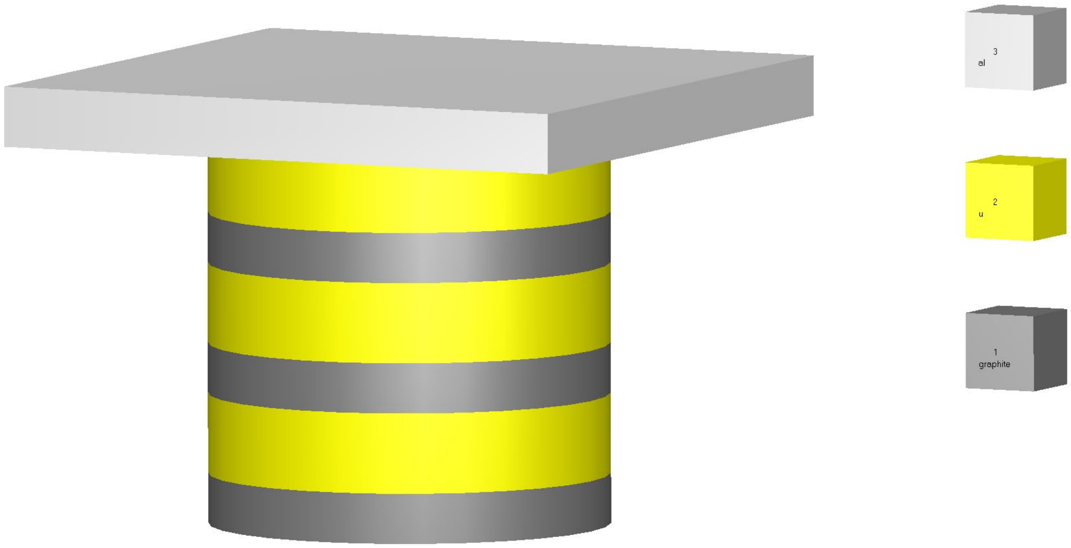


Figure 98. Geometry of stack.

For a unit to be placed in an array, the touching faces of adjacent units must be the same size. This means that the enclosing cuboid for both cylinder types must be the same size in the X and Y dimensions as the top plate. Thus, the length of the X and Y sides of the cuboid must be 30 cm. Notice that there is no constraint on the Z dimension of the cuboid because there are no adjacent faces in that direction. However, to stack the cylinders so that they touch (i.e., no gaps between them), the height of each cuboid surrounding a cylinder must be the same as the height of the cylinder.

Using your knowledge from previous sections, open a new file and enter the title, cross section library, and material information for the graphite, uranium (including the isotopic composition weight percentages), and aluminum.

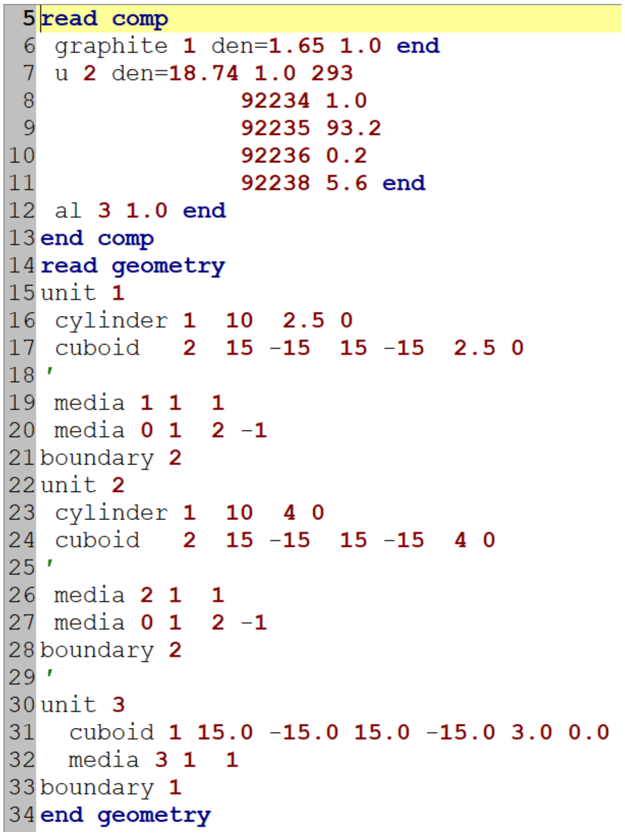
The first unit will be the graphite cylinder surrounded by a void cuboid so that the unit can be stacked in an array. Start writing the geometry by placing the text cursor within the `read geometry` block, between `Global Unit 1` and `boundary 1,` and press the `CTRL-SPACE` autocomplete key combination. Select the **cylinder– kenovi (configurable)**, which opens the **Cylinder** input form. To describe the graphite cylinder, enter `10` for **Radius**, `2.5` for **Top**, and `0.0` for **Bottom**. This puts the origin for the unit and for all shapes in the unit at the center of the cylinder on its bottom face. To enter the data for the enclosing cuboid, place the text cursor within the `read comp` block, and press the `CTRL-SPACE` autocomplete key combination. Select the **cuboid– kenovi (configurable)**, which opens the **Cuboid** input form, to describe the void cuboid. Change the **Id** to 2. Enter `15` and `-15` for **+X** and **–X** and for **+Y** and **–Y**. Enter `2.5` and `0.0` for **+Z** and **–Z**. This creates a cuboid around the cylinder with the proper spacing. Remove `Global` from `Global unit 1`, as unit 1 will be placed in an array.

Use the media records twice to define the two regions. The first region is inside cylinder 1 and is filled with graphite. The second region is inside cuboid 2 and outside cylinder 1 and is filled with void. Change

boundary 1 to `boundary 2` on the line after the media definition to designate the outer boundary as inside cuboid 2.

Now describe the uranium cylinder surrounded by a void cuboid. After the **TODO** comment `boundary 2`, create unit 2 by pressing `CTRL-SPACE` autocomplete key combination and select **unit**. Change the unit number to `2.` Then press the `CTRL-SPACE` autocomplete key combination inside the new unit. Again, select the **cylinder– kenovi (configurable)**, which opens the **Cylinder** input form, to describe the uranium cylinder. Enter `10` for **Radius**, `4` for **Top**, and `0.0` for **Bottom**. To enter the data for the enclosing cuboid, place the text cursor within unit 2 and press the `CTRL-SPACE` autocomplete key combination. Select **cuboid– kenovi (configurable)**, change the **Id** to `2` and enter `15` and `-15` for **+X** and **–X** and for **+Y** and **–Y**. Enter `4` and `0.0` for **+Z** and **–Z**. This creates a cuboid around the cylinder with the proper spacing. Use the media records twice to define the two regions: uranium in the cylinder and void in the cuboid. Change the boundary record to `boundary 2` to designate the outer boundary as inside cuboid 2.

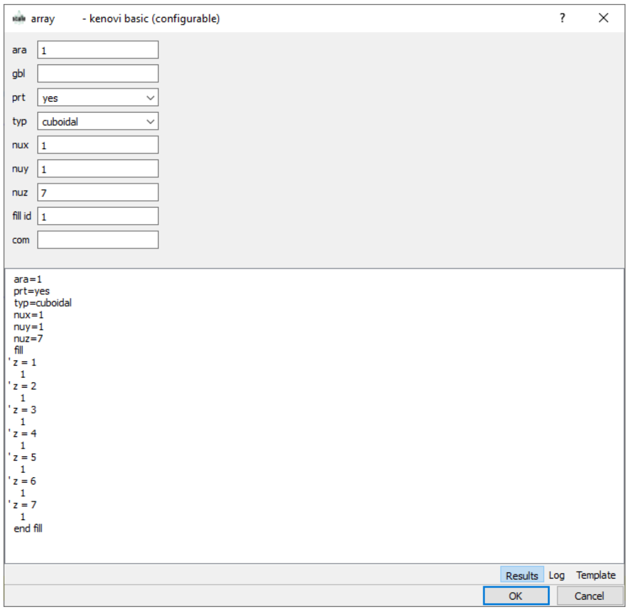
Now describe the aluminum plate by creating unit 3. On the line after `boundary 2`, create unit 3 by typing `unit 3` and on the next line type `boundary 1`. Then placing the text cursor between these two lines and press the `CTRL-SPACE` autocomplete key combination. Select the **cuboid– kenovi (configurable)**, which opens the **Cuboid** input form, to describe the aluminum cuboid. Enter `15` and `-15` for **+X** and **–X** and for **+Y** and **–Y**. Enter `3` and `0` for **+Z** and **–Z**. Use a media record to specify that this cuboid contains aluminum. When you are finished, the composition and geometry blocks should look like Figure 99.



```
read comp
 graphite 1 den=1.65 1.0 end
 u 2 den=18.74 1.0 293
             92234 1.0
             92235 93.2
             92236 0.2
             92238 5.6 end
 al 3 1.0 end
end comp
read geometry
unit 1
 cylinder 1  10  2.5 0
 cuboid   2  15 -15  15 -15  2.5 0
'
 media 1 1  1
 media 0 1  2 -1
boundary 2
unit 2
 cylinder 1  10  4 0
 cuboid   2  15 -15  15 -15  4 0
'
 media 2 1  1
 media 0 1  2 -1
boundary 2
'
unit 3
  cuboid 1 15.0 -15.0 15.0 -15.0 3.0 0.0
  media 3 1  1
boundary 1
end geometry
```

Figure 99. Composition and geometry blocks prior to array definition.

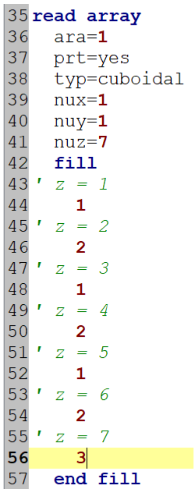
To start describing the array, place the cursor on the line after `end geometry` and use the `CTRL-SPACE` autocomplete key combination to generate a list of available blocks to add to the input, as shown in Figure 89. Within the empty array block, use the `CTRL-SPACE` autocomplete key combination to open the configurable array form shown in Figure 90. This array is the first and only array in the problem, so accepting the default entry of 1 in the **ara** field is recommended. The defaults are also acceptable for **gbl** (there is no global array in this problem), **prt** (print a description of this array in the output, and **typ** (this array is cuboidal). This array is a 1×1×7, so **nux** and **nuy** should be left as **1** and **nuz** should be set to `7`. Any initial **fill id** can be used as several will have to be changed, so accepting the default of 1 is the simplest answer. The completed configurable form is shown in Figure 100; after checking the input press OK to add the array definition to the input file. Axial levels 2, 4, and 6 must be changed to unit `2` to add the uranium unit and axial level 7 must be changed to unit `3` to put the aluminum slab on the top of the array. The final complete array specification is provided in Figure 101.



ara	1
gbl	
prt	yes
typ	cuboidal
nux	1
nuy	1
nuz	7
fill id	1
com	

```
ara=1
prt=yes
typ=cuboidal
nux=1
nuy=1
nuz=7
fill
' z = 1
   1
' z = 2
   1
' z = 3
   1
' z = 4
   1
' z = 5
   1
' z = 6
   1
' z = 7
   1
end fill
```

Figure 100. Fulcrum configurable form for the array in this example.

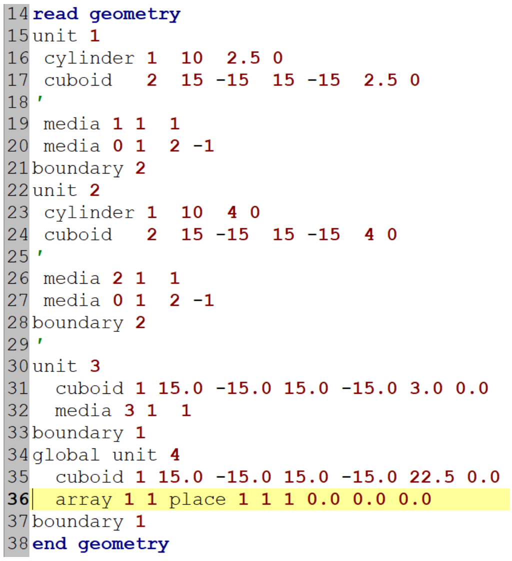


```
read array
  ara=1
  prt=yes
  typ=cuboidal
  nux=1
  nuy=1
  nuz=7
  fill
' z = 1
    1
' z = 2
    2
' z = 3
    1
' z = 4
    2
' z = 5
    1
' z = 6
    2
' z = 7
    3
  end fill
```

Figure 101. Completed array block for this example.

Return to the Geometry definition block and enter `global unit 4` to create unit 4 and enter `boundary 1` on the next line. This unit will be the global unit that contains array 1. With the text cursor in the unit, press the `CTRL-SPACE` autocomplete key combination and select the **cuboid– kenovi (configurable)** button to open the **Cuboid** input form. Note that the geometry shape that contains an array must have dimensions that are exactly the same as or smaller than the outer dimensions of the array. Otherwise there will be undefined volumes inside the shape but outside the array. Enter the dimensions for the cuboid: `±15` for **±X** and **±Y,** `22.5` for +**Z**, and `0` for **-Z**. After the dimensions are entered, press the **OK** button to close the cuboid form.

The last step in the array definition is to define the placement of array 1 in the cuboid using the ARRAY Contents Record. To do so, use the `CTRL-SPACE` autocomplete key combination and select **array - kenovi (configurable)**. The **Array Id** of 1 is correct as the array defined in the array block is declared as array 1. Press the **Add row** button to start defining the region definition vector for the array contents record. The default values are useful here as well as the array is positioned inside cuboid 1. The simplest choice for positing this array is the first element in the array which is unit 1. This is a simple choice because the origin of unit 1 is in the center on the bottom surface of the cylinder; this can be positioned at (0, 0, 0) to align the entire array correctly in cuboid 1. To accomplish this, ensure that the **xindex**, **yindex**, and **zindex** fields are all `1` and the **x**, **y**, and **z** are `0`. The geometry block should look like Figure 102. Save the input file and execute SCALE. The calculated k_{eff} for this critical experiment is 0.9550 ± 0.0019.



```
read geometry
unit 1
 cylinder 1  10  2.5 0
 cuboid   2  15 -15  15 -15  2.5 0
'
 media 1 1  1
 media 0 1  2 -1
boundary 2
unit 2
 cylinder 1  10  4 0
 cuboid   2  15 -15  15 -15  4 0
'
 media 2 1  1
 media 0 1  2 -1
boundary 2
'
unit 3
  cuboid 1 15.0 -15.0 15.0 -15.0 3.0 0.0
  media 3 1  1
boundary 1
global unit 4
  cuboid 1 15.0 -15.0 15.0 -15.0 22.5 0.0
  array 1 1 place 1 1 1 0.0 0.0 0.0
boundary 1
end geometry
```

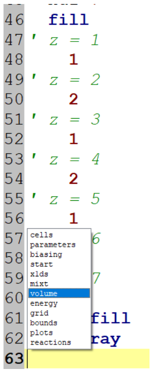
Figure 102. Complete geometry description for this example.

4.7 VOLUME CALCULATIONS

In some applications, the knowledge of region and/or mixture volumes is necessary. For example, calculating fission densities or fluxes in a particular cell region requires the code to know the volume of that region. It is possible to calculate the volume by using the volume block. Different ways of calculating volumes are available in the volume block; only one example is introduced in this primer.

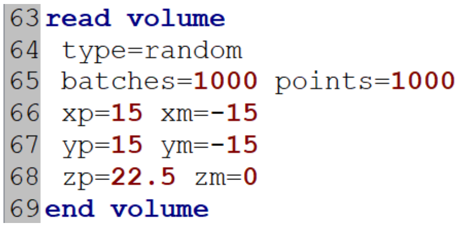
In this example, the volumes of the regions from the previous example, the array with multiple units of the same size, will be calculated. Open the input file that was previously saved, start defining the volume block by placing the text cursor after `end array`, and use the `CTRL-SPACE` autocomplete key combination to generate a list of available blocks to add to the input. Select volume, as shown in Figure 103. First, use the `CTRL-SPACE` autocomplete key combination in the volume block to select the

random volume calculation type. Next, add **batches** with the `CTRL-SPACE` autocomplete key combination and set it to `1000` and add **points** and set it to `1000`. The uncertainty of the calculated volumes is inversely proportional to the total number of samples, which is the product of the points and batches. Finally, a new region must be defined to designate the region of the model in which the volume calculation will be performed. The volume of all regions in the model are desired in this example, so the boundaries of the global unit are used. The syntax in the volume block is different from the geometry block: the positive and negative dimensions are provided in the x,y, and z dimensions. Use the `CTRL-SPACE` autocomplete key combination to add **xp**, **xm**, **yp**, **ym**, **zp**, and **zm** keywords to the input. The **xp** and **yp** values should be set to `15`, the **xm** and **ym** values should be set to `-15`, **zp** should be set to `22.5` and **zm** should be set to `0`. Note that these dimensions match exactly with the cuboid defining the boundary of the global unit. The volume block should look like Figure 104. Save the updated input file and launch SCALE. In the output file, the region volumes, mixture volumes, and associated uncertainties are now available, labeled **total mixture volume (cm**3)**. The result of the volume calculation of region 1 of unit 3 should be 2.35420E+03 +/- 6.34943E+00 cm^3, and the volume calculation should only take a fraction of a second. Higher precision volume estimates are not needed for this demonstration.



```
  fill
' z = 1
    1
' z = 2
    2
' z = 3
    1
' z = 4
    2
' z = 5
    1
        6

        7

        fill
        ray

```

Figure 103. Adding volume block to the input in Fulcrum.



```
read volume
 type=random
 batches=1000 points=1000
 xp=15 xm=-15
 yp=15 ym=-15
 zp=22.5 zm=0
end volume
```

Figure 104. Volume calculation block.

4.8 SUMMARY

This section had the following objectives for users:

- to use Fulcrum to describe some basic shapes (sphere, cylinder, cuboid),
- to understand how units are created (including nesting of shapes),
- to locate and change the location of the origin for shapes and units,
- to create simple arrays consisting of a single unit,
- to create arrays with multiple units of nominally different sizes, and
- to use the volume block to calculate volume of regions and/or mixtures.

The next step is to proceed to more advanced geometries.

5. ADVANCED GEOMETRY

The last section focused on some simple geometry problems, including simple array problems. This section provides more advanced geometry options using KENO-VI geometry modification data.

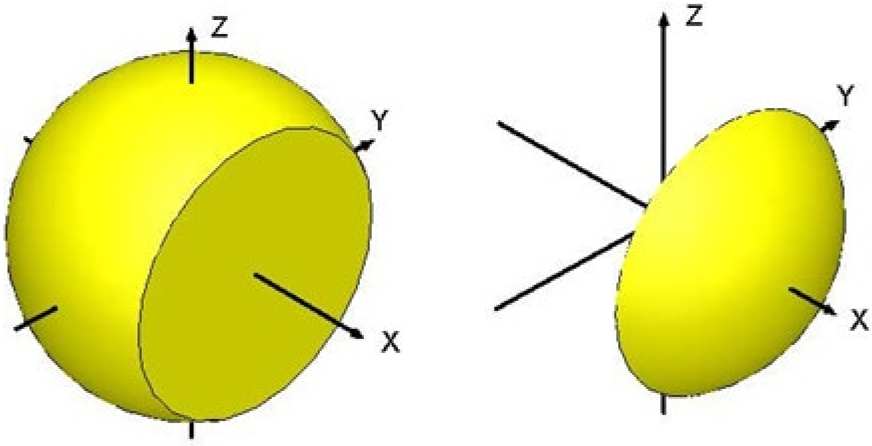
5.1 WHAT YOU WILL BE ABLE TO DO

- Use Fulcrum and KENO-VI to describe more advanced geometry models using geometry modification data and holes.
- Use CHORD keyword to truncate a body with a plane perpendicular to a major axis.
- Use ORIGIN keyword to translate the location of a body.
- Use ROTATE keyword to rotate a body to any angle.
- Use HOLE record to place one unit inside another unit.

5.2 USING THE CHORD KEYWORD TO TRUNCATE A BODY

The CHORD keyword truncates a geometric volume with a plane perpendicular to a major axis. The location of the cutting plane is specified by the subordinate keywords +X=, -X=, +Y=, -Y=, +Z=, and -Z=. Only nonzero data need to be entered. The CHORD keyword applies to the geometry or hole record that immediately precedes it.

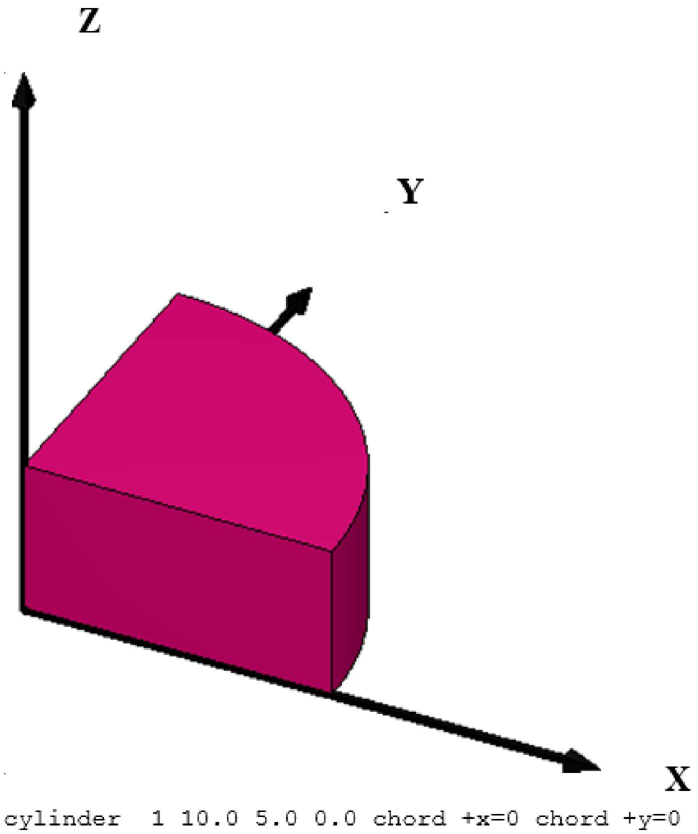
Each subordinate keyword specifies a cutting plane perpendicular to the axis in its name (e.g., “+X=” or “–X=” results in a cutting plane perpendicular to the X-axis). The cutting plane intersects the specified axis at the value that follows the “=” sign. For example, “+X=5” or “-X=5” specifies a cutting plane perpendicular to the X-axis that crosses the axis at X=5. The positive or negative sign before the axis name indicates on which side of the cutting plane the geometric volume is to be kept. For example, in Figure 105, “-X=5” specifies a cutting plane perpendicular to the X-axis at X=5 and keeps the volume on the negative side of the plane (i.e., X ≤ 5). Conversely, “+X=5” specifies a cutting plane perpendicular to the X-axis at X=5 and keeps the volume on the positive side of the plane (i.e., X ≥ 5).



sphere 1 10.0 chord -X=5.0   sphere 2 10.0 chord +X=5.0

Figure 105. Examples of body truncation using the CHORD keyword.

It is possible to apply multiple chords to a body by specifying more than one chord and associated subordinate keyword to a body. Figure 106 shows an example in which two chords are applied to a cylinder to obtain a quarter cylinder. The two subordinate keywords “+X=0” and “+Y=0” keep the quarter cylinder where X ≥ 0 and Y ≥ 0. Note that the keyword CHORD must be supplied for both cuts.



```
cylinder  1 10.0 5.0 0.0 chord +x=0 chord +y=0
```

Figure 106. Examples of body truncation using two chords.

5.3 USING THE ORIGIN KEYWORD TO TRANSLATE A BODY

The ORIGIN keyword allows translation of the origin of a KENO-VI body to another position within a unit by specifying the X, Y, and Z coordinates of the body's origin. The command uses the subordinate keywords X=, Y=, and Z=. Only nonzero data need to be entered. The ORIGIN keyword applies to the geometry or hole record that immediately precedes it.

A simple example is illustrated in Figure 107, in which the origin of a 10 cm radius sphere has been translated 10 cm in X, Y, and Z. The input syntax for this example is as follows:

```
sphere 1 10.0   origin x=10.0 y=10.0 z=10.0
```

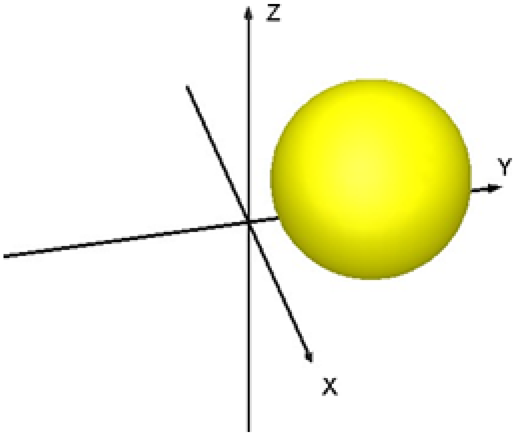


Figure 107. Example of body translation using the ORIGIN keyword.

5.3.1 Example Using the ORIGIN Keyword to Create Three Intersecting Cylinders

A more complex example is illustrated in Figure 108, with three intersecting 10 cm radius cylinders that are 20 cm tall. Cylinder 1 is translated 4.5 cm in the positive Y direction; cylinder 2 is translated 4.5 cm in the negative X and Y directions; cylinder 3 is translated 4.5 cm in the positive X and negative Y directions. The numbers on the top of each colored region are the region definition vectors.

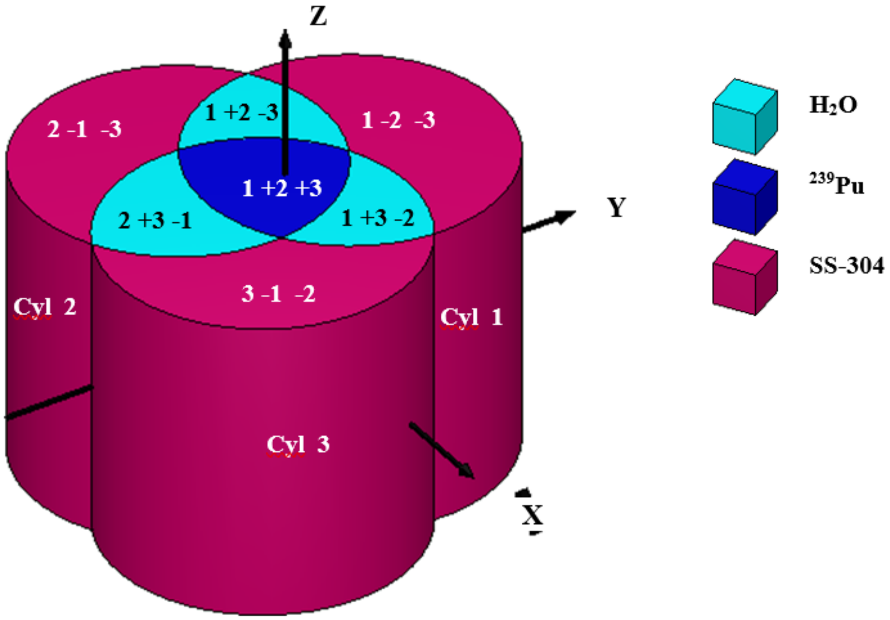
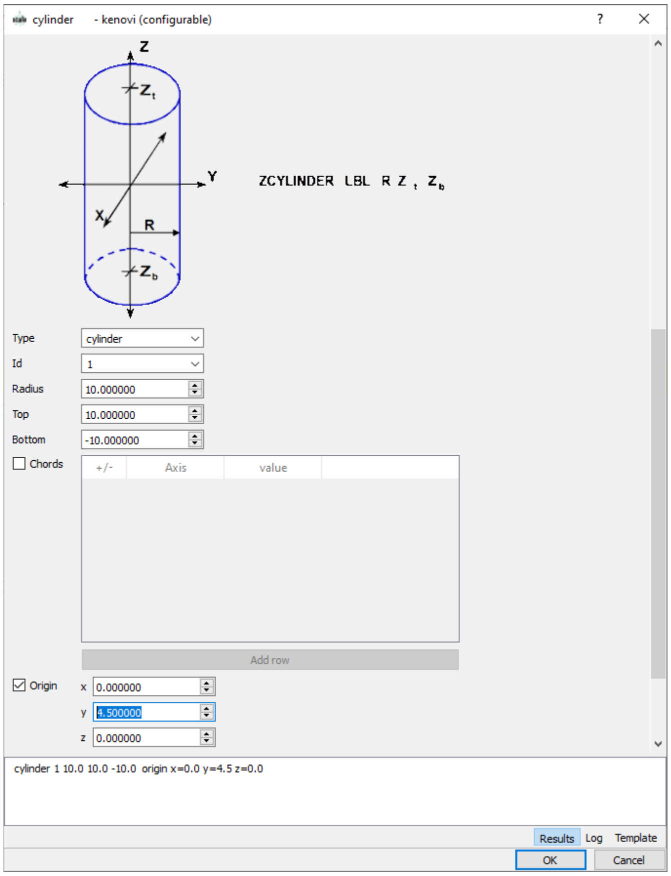


Figure 108. Three intersecting cylinders.

The inner region (dark blue) is ^{239}Pu, with a density of 18.5 g/cc. The surrounding light blue regions contain water. The outer regions (magenta) are stainless steel 304.

Based on the knowledge that you have gained from completing the previous sections of the primer, open a new file in Fulcrum and enter the title, cross section library, and material information. Note that this problem uses the CE ENDF/B-VII.1 library (ce_v7.1). Assign ^{239}Pu (`pu-239`) as mixture `1`, water (`h2o`) as mixture `2`, and stainless steel 304 (`ss304`) as mixture `3`.

Start the geometry input by placing the text cursor within unit 1, between `Global Unit 1` and `boundary 1`. Press the `CTRL-SPACE` autocomplete key combination and select **cylinder– kenovi (configurable)**, to open the **Cylinder** input form. Enter `10` for **Radius** and **Top**, and `-10` for **Bottom**. Next, click on the **Origin** checkbox to specify the origin of the cylinder. Enter `4.5` for the **Y** value of **Origin** to translate the origin of the first cylinder to (0, 4.5, 0). Click **OK** to close the cylinder form, after confirming it matches that shown in Figure 109. Recall that only non-zero data need be entered in the ORIGIN specification; declaring the other coordinates equal to zero is acceptable.

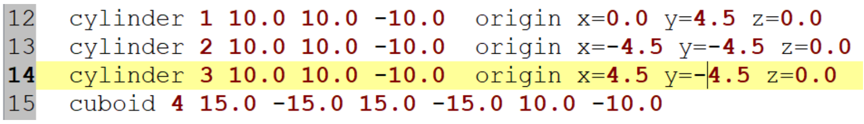


Type: cylinder
Id: 1
Radius: 10.000000
Top: 10.000000
Bottom: -10.000000
☐ Chords
☑ Origin
x 0.000000
y 4.500000
z 0.000000

```
cylinder 1 10.0 10.0 -10.0 origin x=0.0 y=4.5 z=0.0
```

Figure 109. Cylinder configurable form with ORIGIN specification data.

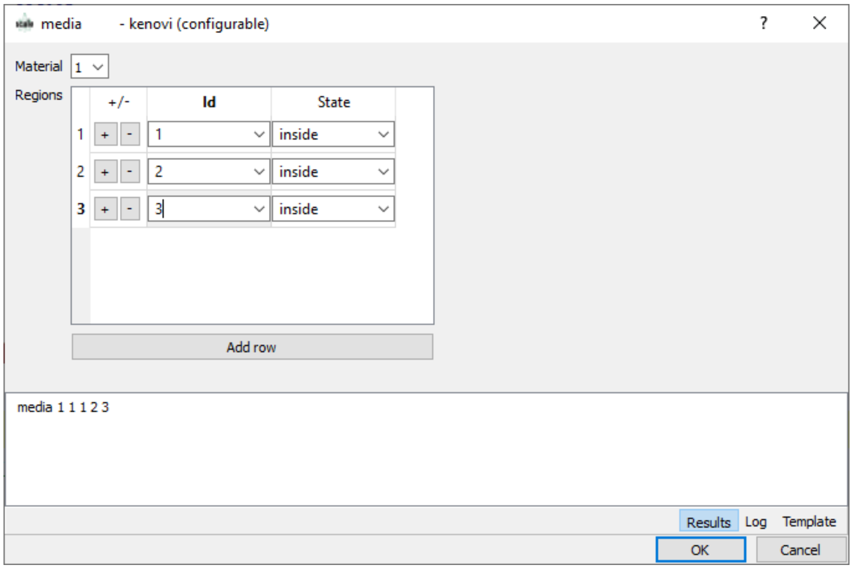
The simplest way to create the other type cylinders is to copy and paste cylinder 1 and update the surface numbers and ORIGIN data. Copy the input for cylinder 1 and paste it on the next line. Change the surface ID to 2 and the ORIGIN data to `x=-4.5` and `y=-4.5`. Similarly, copy cylinder 2 and paste it. Change the surface ID to 3 and the ORIGIN data to `x=4.5`. On the line after cylinder 3, press the `CTRL-SPACE` autocomplete key combination. Select **cuboid - kenovi (configurable)** to open the **Cuboid** input form. Change the **Id** to `4`, enter dimensions of `15` for **+X** and **+Y** and `-15` for **−X** and **−Y.** Set **+Z** and **−Z** to `10` and `-10`, respectively. This region will serve as the outer boundary for the unit, and is sized so that no portion of the three intersecting cylinders will be truncated. The complete set of geometry records is shown in Figure 110.



```
cylinder 1 10.0 10.0 -10.0  origin x=0.0 y=4.5 z=0.0
cylinder 2 10.0 10.0 -10.0  origin x=-4.5 y=-4.5 z=0.0
cylinder 3 10.0 10.0 -10.0  origin x=4.5 y=-4.5 z=0.0
cuboid 4 15.0 -15.0 15.0 -15.0 10.0 -10.0
```

Figure 110. Input for three cylinders with ORIGIN data specified.

The media records are supplied next to fill each region. First, define the ^{239}Pu region in the center. The ^{239}Pu region, mixture 1, is defined as being inside cylinders 1, 2 and 3 at the same time. To create it, press the `CTRL-SPACE` autocomplete key combination and select **media - kenovi (configurable)**. Press the **Add row** button below the **Regions** pane. Add additional rows and declare **Material 1** to be inside of **Id** numbers 2 and 3 as well by typing the numbers into the **Id** fields. The completed media configurable form is shown in Figure 111; after reviewing the form press the OK button to add the media record to the input. This record fills the region of space that is inside surface 1 and surface 2 and surface 3 with mixture 1. The point must be simultaneously inside all 3 surfaces to satisfy the media record and be included in this sector.



Material 1

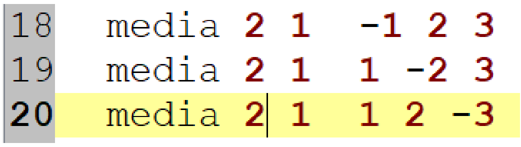
Regions

	+/-	Id	State
1	+ -	1	inside
2	+ -	2	inside
3	+ -	3	inside

```
media 1 1 1 2 3
```

Figure 111. Complete media record configurable form for the region inside all three cylinders.

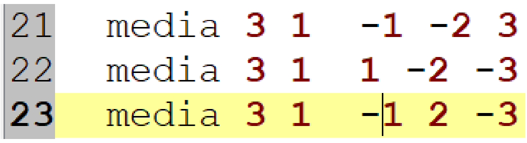
Next the regions containing water, mixture 2, will be defined. The quickest approach is to copy the existing media record and copy it onto the next line. Change the mixture number from 1 to 2, and change the sign of surface 1 in the region definition vector from positive to negative. This region will be filled with water and is defined as inside cylinders 2 and 3 but outside cylinder 1. Copy and paste this record two more times, moving the negative sign in the region definition vector to cylinder 2 and then subsequently to cylinder 3, as shown in Figure 112.



```
media 2 1  -1  2  3
media 2 1   1 -2  3
media 2 1   1  2 -3
```

Figure 112. Media records for water in overlapping cylinder example.

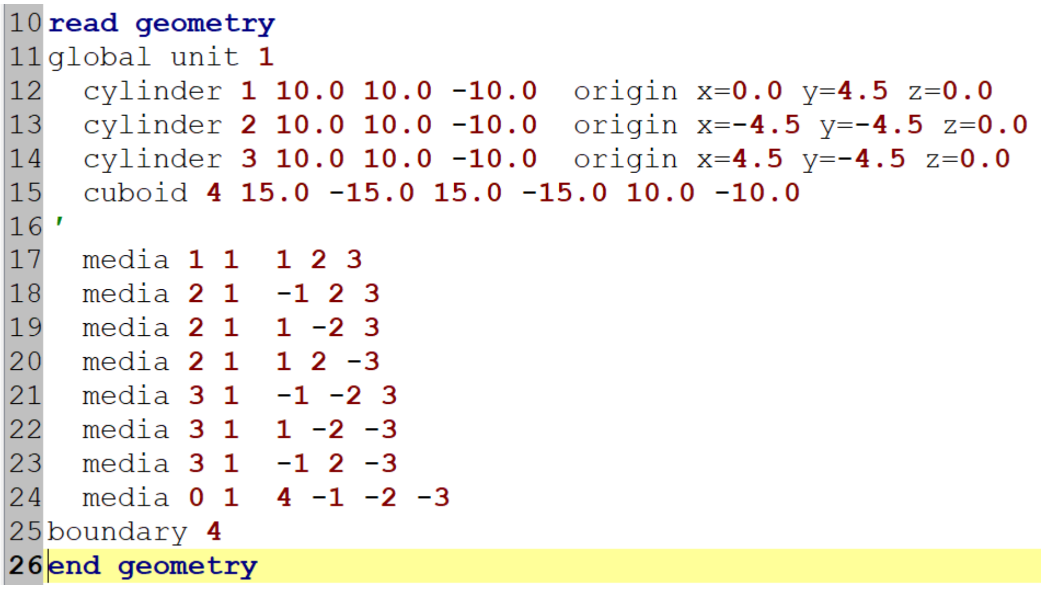
Next the three stainless steel regions will be defined. Again, copy and paste is the fastest way to generate the input. Copy the three water media records defined above and paste them on the following three lines. Change the mixture number for all three records from **2** to `3`. This can be accomplished with column highlighting: hold the ALT key and drag the mouse from top to bottom of the three numbers to change. Once all three are highlighted, press 3 and all three mixture numbers are changed at once. Each media record also must have a second negative value added to the region definition vector. For example, change cylinder 2 to negative in the first record, cylinder 3 to negative in the second record, and cylinder 1 to negative in the third record. The three stainless steel media records are shown in Figure 113.



```
media 3 1  -1 -2 3
media 3 1  1 -2 -3
media 3 1  -1 2 -3
```

Figure 113. Media records for stainless steel in overlapping cylinder example.

The final media record fills the surrounding void inside cuboid 4 and outside cylinders 1, 2, and `3`. This is done by typing `media 0 1  4 -1 -2 -3.` The cuboid is the boundary of the unit, so change `boundary 1` to `boundary 4`. The geometry form should look like Figure 114. Click the **Run** button to save the input and run the problem. The calculated k_{eff} value is 1.3674 ± 0.0023.



```
read geometry
global unit 1
  cylinder 1 10.0 10.0 -10.0  origin x=0.0 y=4.5 z=0.0
  cylinder 2 10.0 10.0 -10.0  origin x=-4.5 y=-4.5 z=0.0
  cylinder 3 10.0 10.0 -10.0  origin x=4.5 y=-4.5 z=0.0
  cuboid 4 15.0 -15.0 15.0 -15.0 10.0 -10.0
'
  media 1 1  1 2 3
  media 2 1  -1 2 3
  media 2 1  1 -2 3
  media 2 1  1 2 -3
  media 3 1  -1 -2 3
  media 3 1  1 -2 -3
  media 3 1  -1 2 -3
  media 0 1  4 -1 -2 -3
boundary 4
end geometry
```

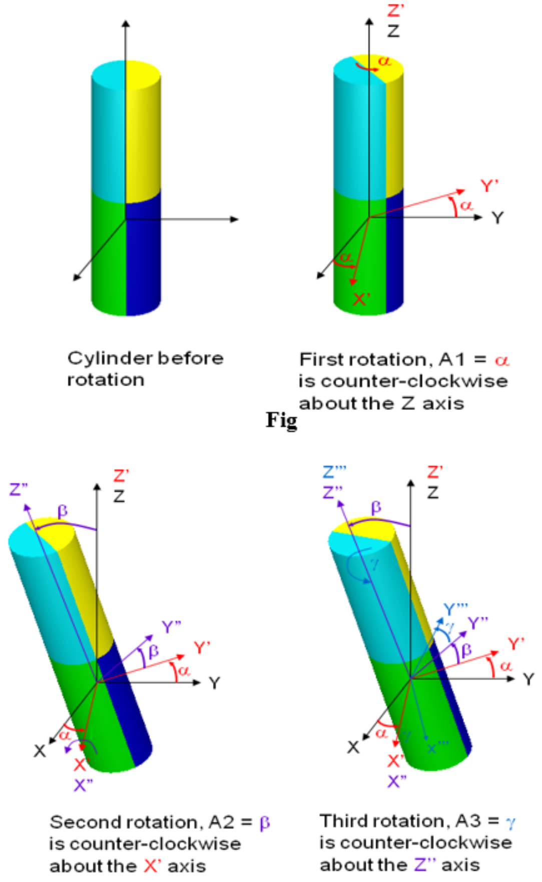
Figure 114. Geometry form for three intersecting cylinders.

5.4 USING THE ROTATE KEYWORD TO ROTATE A BODY

The ROTATE keyword allows the rotation of a KENO-VI shape to any angle. It rotates a volume about the origin using the Euler angle-x convention. Like CHORD and ORIGIN, it applies to the geometry or hole record that immediately precedes it. Note that for any geometry record that contains both ORIGIN and ROTATE input, the rotation is performed by KENO-VI first, prior to translating the origin of the body. Any truncation performed by the CHORD keyword will be performed *prior* to the rotation. The ROTATE keyword has three subordinate keywords that specify the angles of rotation: A1=, A2=, and A3=.

The functions of the three subordinate keywords are illustrated in Figure 115 The operations are performed in order from A1 to A3. First, A1 specifies α, the angle (in degrees) to rotate the volume counterclockwise around the Z-axis. The new locations of the X-, Y-, and Z-axes carry the notations X′, Y′, and Z′. Next, A2 specifies β, the angle to rotate the volume counterclockwise around the X′-axis. The new locations of the X′-, Y′-, and Z′-axes are denoted as X″, Y″, and Z″. Finally, A3 specifies γ, the angle to rotate the volume counterclockwise around the Z″-axis. Some examples will help to clarify the use of these keywords.

```
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

Fig

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

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

Figure 115. ROTATE subordinate keywords A1, A2, and A3.

5.4.1 Rotation about the Z-axis

Rotation around the Z-axis is performed using the keyword A1. The value specified after the keyword “A1=” is the angle of rotation around the Z-axis. Figure 116 shows a cuboid in its default orientation with no rotation. The example in Figure 117 shows the same cuboid with a rotation of 45 degrees counterclockwise around the Z-axis by specifying ROTATE A1=45. Note that a negative value for the angle of rotation would result in a clockwise rotation.

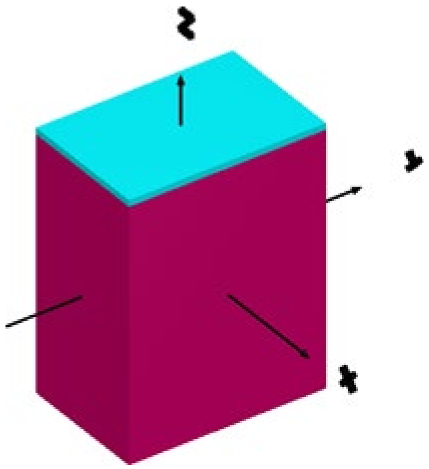


Figure 116. Cuboid (default orientation).

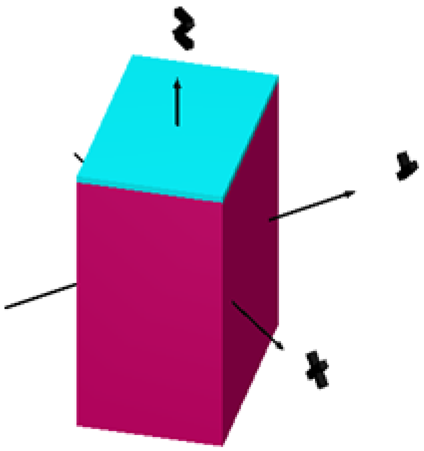


Figure 117. Cuboid rotated 45 degrees counterclockwise around Z-axis (rotate A1=45).

5.4.2 Rotation about the X-axis

The keyword A2 is used for rotation around the X-axis. If A1 is not specified there is no rotation about the Z-axis and the X-axis and the X′-axis are the same. The example in Figure 118 shows the cuboid from Figure 116 with a rotation of 45 degrees counterclockwise around the X-axis using the keyword ROTATE A2=45.

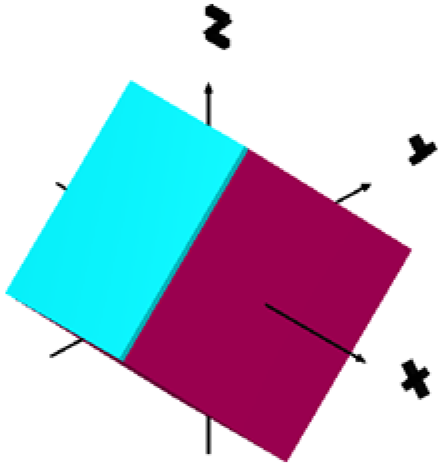


Figure 118. Cuboid rotated 45 degrees counterclockwise around X-axis (rotate A2=45).

5.4.3 Rotation about the Y-axis

The subordinate keywords A1 and A2 must be used to perform rotation about the Y-axis. First, a 90 degree counterclockwise rotation around the Z-axis is performed (A1=90) so that the X′-axis is aligned with the Y-axis. Then the rotation around the X′-axis using A2 will be a rotation around the Y-axis. The example in Figure 119 shows the cuboid from Figure 116 with a 45 degree counterclockwise rotation around the Y-axis using the keywords ROTATE A1=90 A2=45.

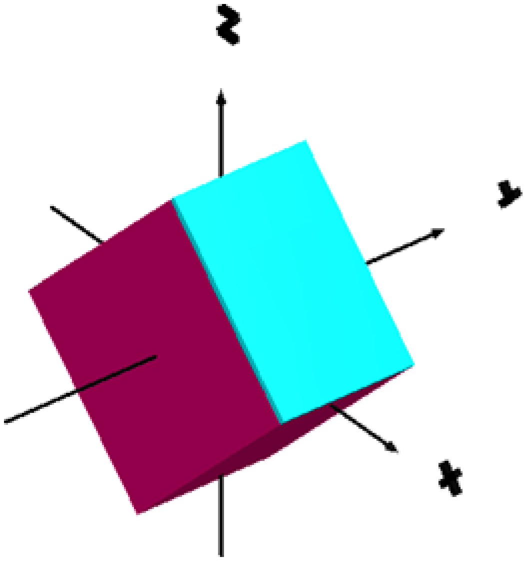


Figure 119. Cuboid rotated 45 degrees counterclockwise around Y-axis (rotate A1=90 A2=45).

5.4.4 Example using ROTATE keyword for Y-angle Joint

The next example is a critical experiment with intersecting aluminum pipes in the shape of a Y, filled with a 5% enriched UO_2F_2 solution (Figure 120). The solution contains 907.01 g/L of uranium, no excess acid, and has a solution density of 2.0289 g/cc. The temperature of the materials is of 293 K. The assembly is composed of a 212.1 cm long vertical pipe and a second pipe that intersects the vertical pipe 76.7 cm from the outside bottom at an angle of 29.26 degrees with the upper vertical pipe. The center of the vertical pipe at this point of intersection is used as the origin of the geometry, as shown in Figure 120.

Both pipes have a 13.95 cm inner radius and a 14.11 cm outer radius. The vertical pipe is open on the top and 1.3 cm thick on the bottom. The Y-leg pipe is 126.04 cm in length, with the sealed end 0.64 cm thick. Both pipes are fabricated from default density pure aluminum. The solution height is 129.5 cm. From the point where the pipes intersect (see the origin in Figure 120), the assembly is surrounded by water 37.0 cm in the ±X directions, 100 cm in the +Y direction, –37 cm in the –Y direction, to the top of the assembly in the +Z direction, and –99.6 cm in the –Z direction.

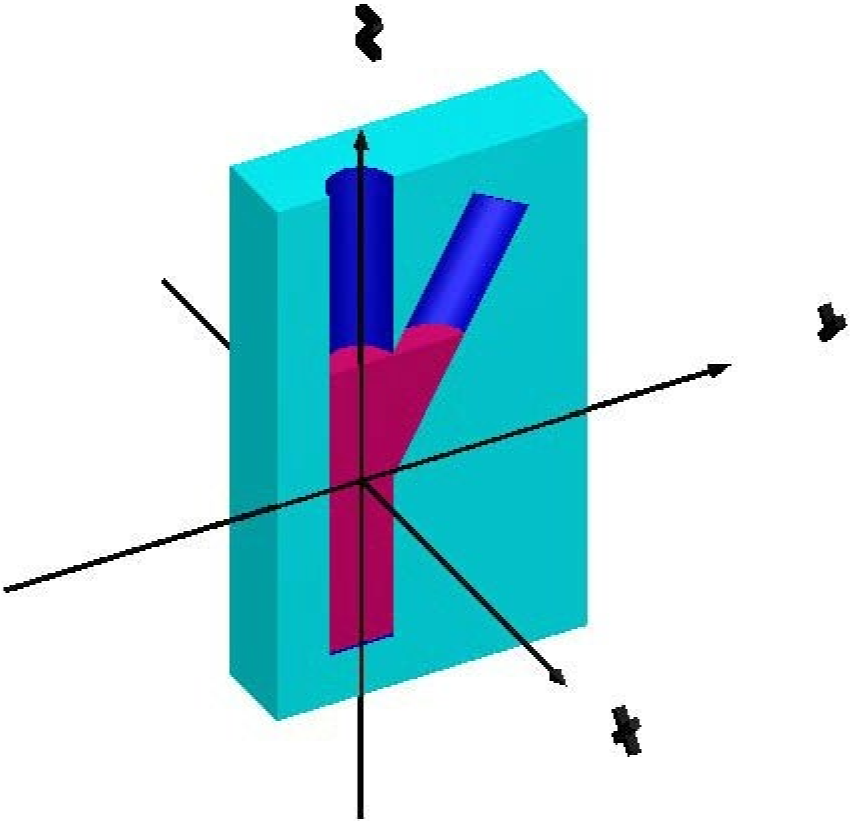
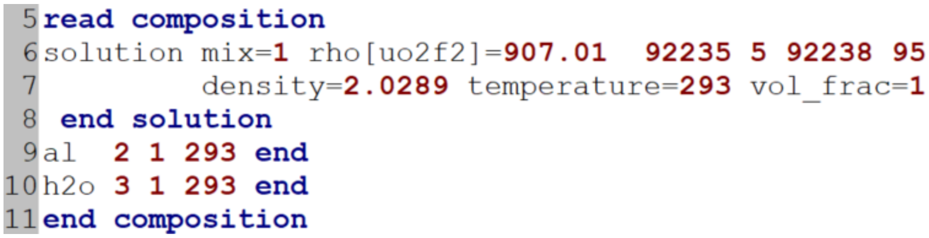


Figure 120. Cutaway view of critical experiment with 29-degree Y-joint.

Based on the knowledge gained from completing the previous sections of the primer, open a new file in Fulcrum and enter the title, cross section library, and material information. The uranium solution is defined with the **solution – rho + density + temperature + volume fraction** Fulcrum autocomplete option. The completed composition block should look like Figure 121.



```
read composition
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 composition
```

Figure 121. Composition block for the 29-degree Y-joint.

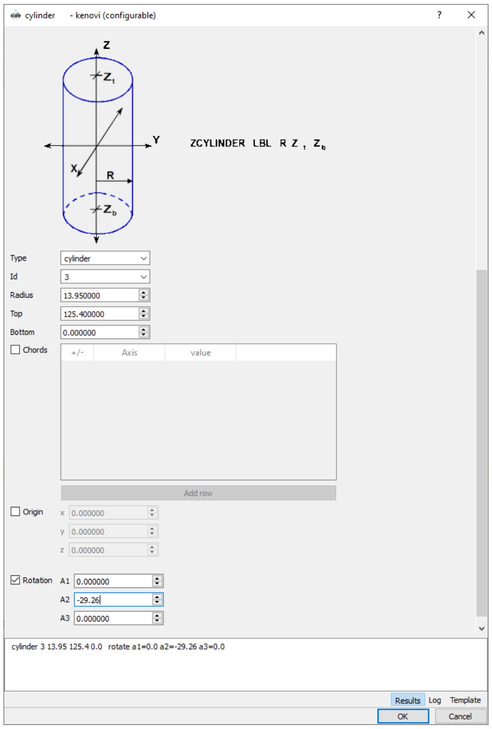
The geometry definition begins with defining two cylinders (inner and outer, respectively) for the vertical pipe. Enter the following data in cylinder configurable forms, typing the input directly in the input, or one configurable form and a copy/paste to define cylinders 1 and 2:

- Cylinder 1: `13.95` for **Radius**, `135.4` for **Top** and `-75.4` for **Bottom**
- Cylinder 2: `2` for **Id,** `14.11` for **Radius**, `135.4` for **Top** and `-76.7` for **Bottom**

Next, define two more cylinders (inner and outer) for the angled pipe. Use the `rotate` and `A2` keywords to rotate the cylinders 29.26 degrees clockwise around the X-axis. On the cylinder form, click on the **Rotation** checkbox to allow for rotation configuration A2 of the cylinder. The configurable form for the inner cylinder is shown in Figure 122. The following values should be entered for each cylinder.

- Cylinder 3: `3` for **Id,** `13.95` for **Radius**, `125.4` for **Top** and `0.0` for **Bottom**; `rotate a2=-29.26` for the rotation.

- Cylinder 4: 4 for **Id,** `14.11` for **Radius**, `126.04` for **Top** and `0.0` for **Bottom**; `rotate a2=-29.26` for the rotation.

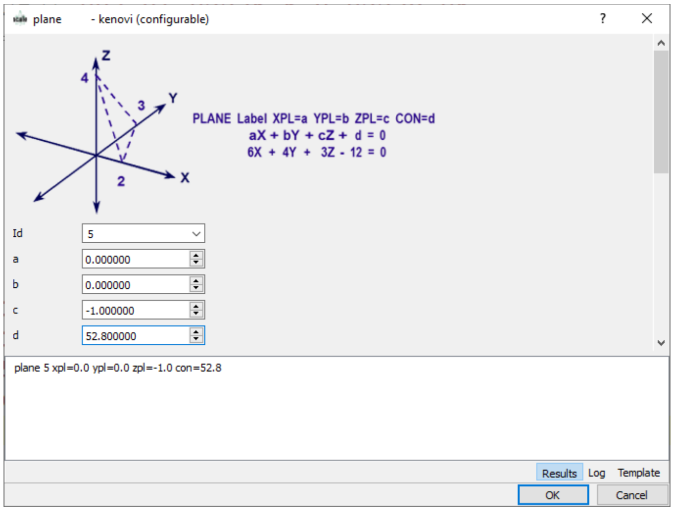


Type	cylinder
Id	3
Radius	13.950000
Top	125.400000
Bottom	0.000000
□ Chords	
□ Origin	x 0.000000 y 0.000000 z 0.000000
☑ Rotation	A1 0.000000 A2 -29.26 A3 0.000000

```
cylinder 3 13.95 125.4 0.0  rotate a1=0.0 a2=-29.26 a3=0.0
```

Figure 122. Cylinder configurable form for inner angled pipe.

A horizontal plane is needed to separate the portion of the cylinders filled with solution from the void portions above them. The top of the solution is 52.8 cm above the origin (129.5 cm solution height, 76.7 cm from the origin to the bottom of cylinder 2), so the plane is perpendicular to the Z-axis and at Z = 52.8. Press the `CTRL-SPACE` autocomplete key combination. Click on the **plane– kenovi (configurable)** button, which opens the **Plane** input form. Set **Id** to `5`, **c** to `-1` and **d** to `52.8`. This creates a plane corresponding to the equation –Z + 52.8 = 0, which is equivalent to the desired plane at Z=52.8. The same plane can also be defined with c (ZPL=) set to 1 and d (CON=) set to -52.8. This alternative definition generates the surface equation Z – 52.8 = 0, which is also equivalent to the desired surface of Z=52.8. The difference between the equivalent forms is which half space, above the plane or below the plane, is assigned as positive (inside) relative to the surface. The complete plane configurable form is shown in Figure 123. Click **OK** to close the plane form and add the geometry record to the input file.



Id: 5
a: 0.000000
b: 0.000000
c: -1.000000
d: 52.800000

```
plane 5 xpl=0.0 ypl=0.0 zpl=-1.0 con=52.8
```

Figure 123. Completed plan configurable form.

The final geometric volume needed is the cuboid for the water reflector. Create a cuboid with the following dimensions: **+X**=`37`, **-X**=`-37`, **+Y**=`100`, **-Y**=`-37`, **+Z**=`135.4`, and **-Z**=`-99.6`. After you close the cuboid form, define the boundary to be inside cuboid 6 by setting `boundary 6`.

Next, define the contents of the geometric volumes using media records. First, define the solution, mixture 1, inside the vertical cylinder and below the plane. Note that the positive side of the plane ("inside" the plane) is the side where the equation aX + bY + cZ + d > 0: in this case –Z + 52.8 > 0. This allows the determination of the positive side of the plane by selecting a known point and evaluating the inequality. For example, in this model the origin can be used (-1(0) + 52.8 = 52.8) to determine that it is on the positive side of the plane. This is the portion of the inner cylinder containing the solution, so the media record needs to define mixture 1 (solution) as inside both cylinder 1 and plane 5. Enter the media record either with the configurable form or by typing the input directly.

Next, define the contents of the vertical pipe wall. The region definition vector on the media form will be inside cylinder 2, and it will be outside cylinder 1 and cylinder 3. Cylinder 3 is excluded for the solution in the side leg. The mixture number for this media record is `2` (al).

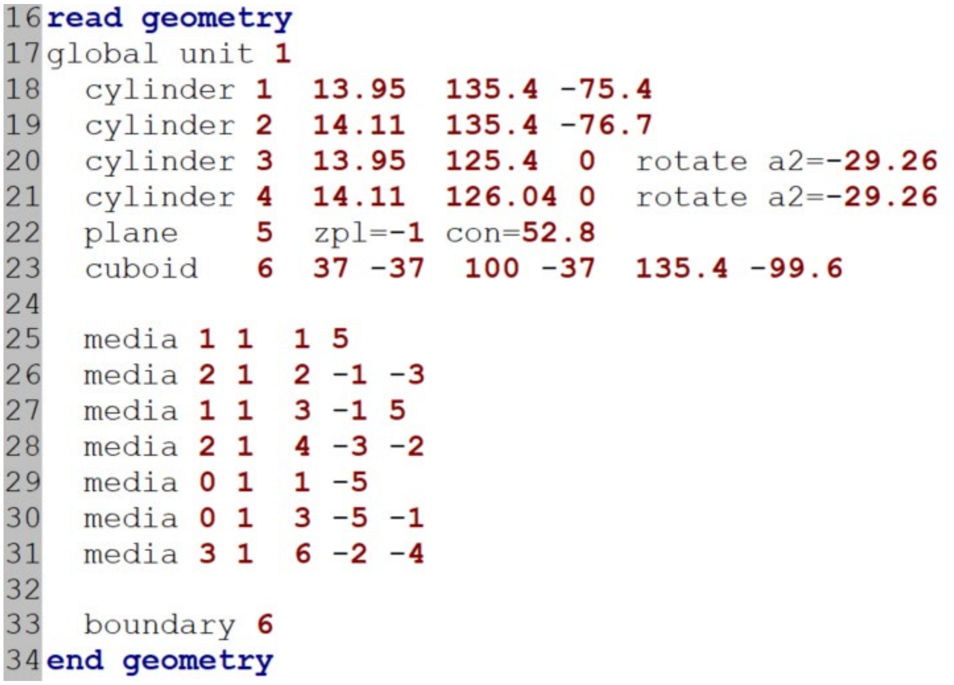
The solution in the angled pipe is defined next. The region definition vector on the media form will be inside cylinder 3 and plane 5, and it will be outside cylinder 1. The mixture number is `1` (uo2f2).

Next, define the contents of the angled pipe wall. The region definition vector on the media form will be inside cylinder 4, and it will be outside cylinder 3 and cylinder 2. The mixture number is `2` (al).

The next two media records define the voids in the top of the two pipes. The region definition vector for the void region in the vertical pipe is inside cylinder 1 and outside plane 5. The region definition vector for the void region in the angled pipe is inside cylinder 3, and outside cylinder 1 and plane 5. The mixture for both media records is `0` (void).

The final media record is the water reflector. The region definition vector is inside cuboid 6, and it is outside cylinder 2 and cylinder 4. The mixture number is `3` (h2o).

Your completed geometry block should look like Figure 124. Click the **Run** button to save the input and run the case. The calculated k_{eff} value for this critical experiment is 1.0012 ± 0.0018, which is near 1.0, as expected.



```
read geometry
global unit 1
  cylinder 1  13.95  135.4 -75.4
  cylinder 2  14.11  135.4 -76.7
  cylinder 3  13.95  125.4  0  rotate a2=-29.26
  cylinder 4  14.11  126.04 0  rotate a2=-29.26
  plane    5  zpl=-1 con=52.8
  cuboid   6  37 -37  100 -37  135.4 -99.6

  media 1 1  1 5
  media 2 1  2 -1 -3
  media 1 1  3 -1 5
  media 2 1  4 -3 -2
  media 0 1  1 -5
  media 0 1  3 -5 -1
  media 3 1  6 -2 -4

  boundary 6
end geometry
```

Figure 124. Geometry block for the 29-degree Y-joint.

5.5 HOLES

5.5.1 Introduction

There are three types of contents records for defining the contents of the geometry regions:

- media,
- arrays, and
- holes.

As shown in Section 4, a media record is used to place a mixture in a specified volume, and an array record is used to place an array in a specified volume. For many problems, the use of media and arrays is adequate to create an accurate model.

However, in some situations, placement of one unit inside another may be necessary or helpful. For those situations, a hole is needed. A hole is used to position a unit within a surrounding unit relative to the origin of the surrounding unit. The keyword, HOLE, is followed by the unit number being placed in the hole. The unit being placed in the hole can be positioned and rotated using the ORIGIN and ROTATE keywords discussed earlier in this section. The origin of the unit being placed as a HOLE is positioned in the current unit coordinate system using the ORIGIN keyword; as with geometry records, only nonzero entries must be provided. The unit being placed as a HOLE is rotated about its origin, not that of the current unit, using the ROTATE keyword. As with the geometry record and ORIGIN keyword, only nonzero entries must be provided. The boundary record of a unit placed in a hole may contain more than one geometry label, but all labels must be positive, indicating a location inside the respective geometry bodies.

There are three basic rules for holes:

1. A hole contains a single unit.

2. Holes may share surfaces with but may not intersect other holes, the boundary of the surrounding unit that contains the hole, or an array boundary.

3. Holes may be nested; that is, a unit containing a hole may be placed as a hole inside another unit.

5.5.2 Simple Hole Example with Rotations

This example includes five stainless steel 304 boxes that contain plutonium metal (1% ^{239}Pu and 99% ^{240}Pu and default density). The boxes are encased in a concrete cylinder. A cutaway view (top half removed) is shown in Figure 125. The dimensions of the plutonium blocks are 10 × 8 × 20 cm. The stainless steel box is 1 cm thick on three sides and on the top and bottom. The fourth side (i.e., the one oriented toward the center of the cylinder for the four outer boxes) is 3 cm thick. The center of each outer box is spaced 12 cm in X and Y from the center of the cylinder.

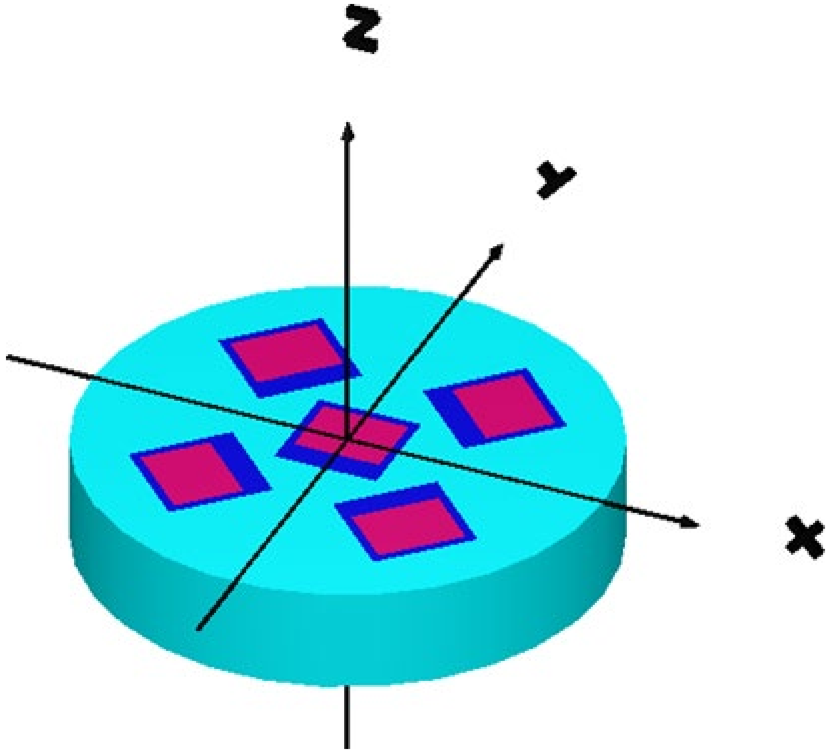


Figure 125. Cutaway view of rotated holes example.

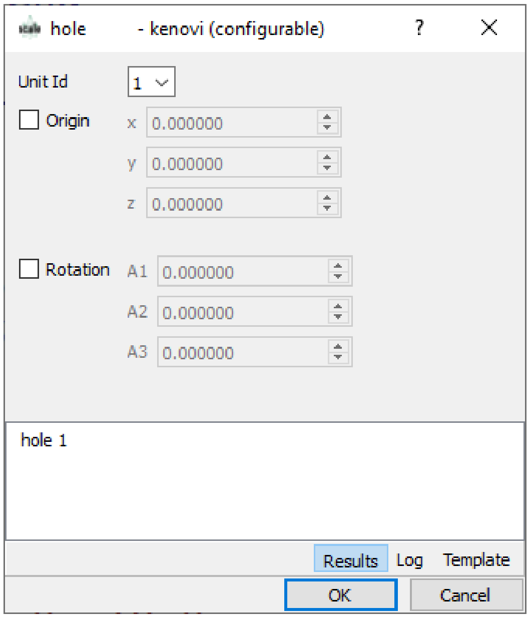
Create a new file with Fulcrum including title, cross section library, and composition information for the plutonium, stainless steel, and concrete. For the concrete, select `reg-concrete` in the basic compositions list.

Once these data are complete, start the geometry definition. The first unit will be the stainless steel box containing the plutonium. The outer dimensions of the box are 12 × 12 × 22 based on adding the stainless steel dimensions to the plutonium block dimensions. Placing the origin of unit 1 at the center of the box will make it easy to position each box as a hole in the concrete cylinder, unit 2.

Start the geometry by deleting the Global label from unit 1. Then place the text cursor within `Unit 1` and press the `CTRL-SPACE` autocomplete key combination. Select the **cuboid – kenovi (configurable)**, to open the **Cuboid** configurable form. Enter the following data for the plutonium block: `5` for **X**, `-5` for **-X**, `5` for **Y**, `-3` for **–Y**, `10` for **Z** and `-10` for **–Z**. Next, create another cuboid for the stainless steel with the following dimensions: `6` for **+X** and **+Y**, `-6` for **–X** and **–Y**, `11` for **+Z** and `-11` for **–Z**. Set the media record for the plutonium. The plutonium, mixture 1, is inside cube 1. Create the second media record inside cuboid 2 and outside cuboid 1. The mixture number is 2 (ss304). The last step for unit 1 is to change the boundary record region definition vector to `2`.

Create `global unit 2`. Open the **Cylinder** input form to define the concrete cylinder, and enter `30` for **Radius**, `12` for **Top** and `-12` for **Bottom**. For the media record, set mixture `3` (reg-concrete) inside cylinder 1. Set the boundary of the problem as the concrete cylinder by writing `boundary 1`.

Now place the five boxes as holes in unit 2. As mentioned previously, the location of the hole is controlled by placing its origin with respect to the origin of the unit in which it is placed. The first hole will be placed in the center of unit 2 without rotation. To set the first hole, press the `CTRL-SPACE` autocomplete key combination somewhere in unit 2 and select **hole - kenovi (configurable)**. The default hole configurable form is shown in Figure 126. The **Unit Id** in the upper left corner of the form should remain 1 since unit 1 is the unit to be inserted as a hole. No further data is needed as the origin of unit 1 should be positioned at the origin of unit 2. Press the **OK** button to add the hole record to the input file.



Unit Id 1

☐ Origin x 0.000000 y 0.000000 z 0.000000

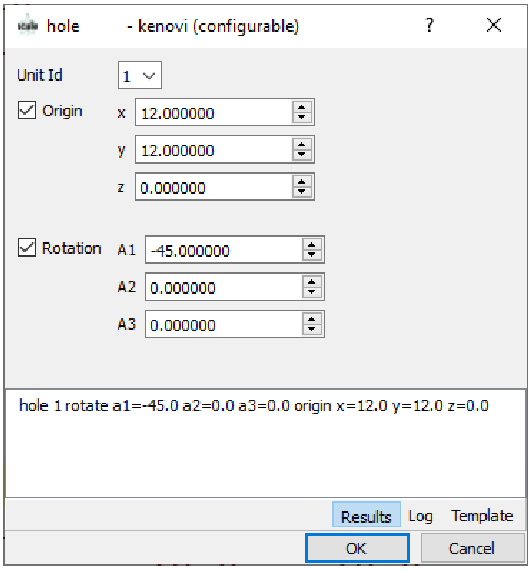
☐ Rotation A1 0.000000 A2 0.000000 A3 0.000000

```
hole 1
```

Figure 126. Default hole configurable form.

The next hole will be placed with at (12, 12) and it must also be rotated 45 degrees clockwise around the Z axis for the thick side of the box to face the center of the cylinder as designed. Use the `CTRL-SPACE` autocomplete key combination somewhere in unit 2 and select **hole - kenovi (configurable)**. Unit 1 is still the applicable **Unit Id**, but in this case the **Origin** and **Rotation** checkboxes must be checked to allow specification of translation and rotation data. **X** and **Y** should be set to `12`, and **A1** should be set to

−45. After confirming the input with the completed hole configurable form shown in Figure 127 press the **OK** button to add the hole record to the input file.



Unit Id 1

☑ Origin x 12.000000

y 12.000000

z 0.000000

☑ Rotation A1 -45.000000

A2 0.000000

A3 0.000000

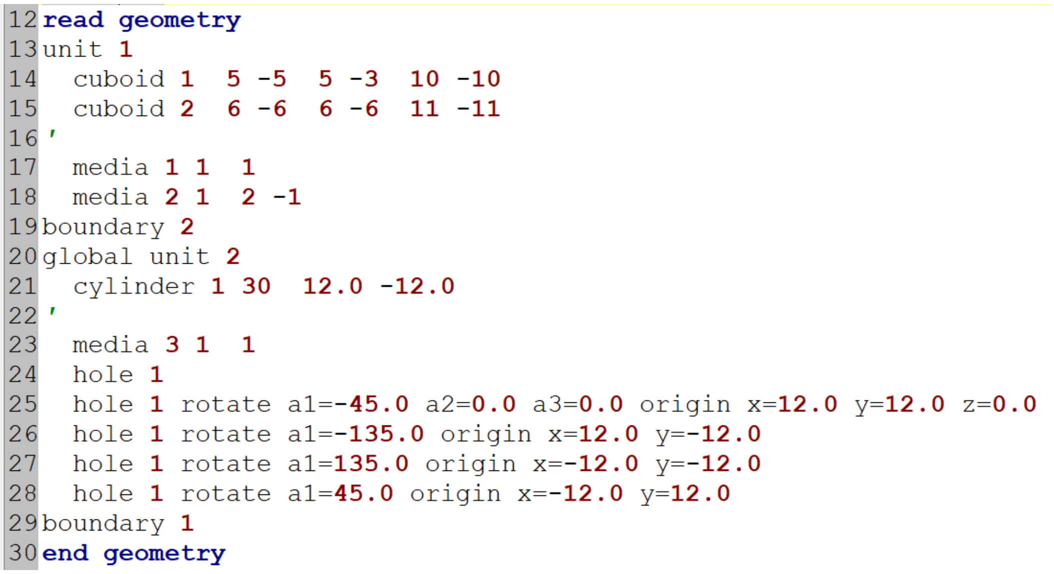
```
hole 1 rotate a1=-45.0 a2=0.0 a3=0.0 origin x=12.0 y=12.0 z=0.0
```

Figure 127. Completed hole record for second hole record.

Continuing clockwise around the cylinder, enter the following data for the remaining holes using the configurable form, directly typing the input, or copy/paste and updating the records:

- **Origin** is located at **X**=12, **Y**=−12 and **Rotate A1**=−135.
- **Origin** is located at **X=**−12, **Y**=−12 and **Rotate A1=**135.
- **Origin** is located at **X**=−12, **Y**=12 and **Rotate A1**=45.

The geometry block may look like Figure 128, depending on the methods used to specify the hole records. After checking that the input data match, click **Run** to execute SCALE. The calculated k_{eff} is 1.0406 ± 0.0018.



```
read geometry
unit 1
  cuboid 1  5 -5  5 -3  10 -10
  cuboid 2  6 -6  6 -6  11 -11
'
  media 1 1  1
  media 2 1  2 -1
boundary 2
global unit 2
  cylinder 1 30  12.0 -12.0
'
  media 3 1  1
  hole 1
  hole 1 rotate a1=-45.0 a2=0.0 a3=0.0 origin x=12.0 y=12.0 z=0.0
  hole 1 rotate a1=-135.0 origin x=12.0 y=-12.0
  hole 1 rotate a1=135.0 origin x=-12.0 y=-12.0
  hole 1 rotate a1=45.0 origin x=-12.0 y=12.0
boundary 1
end geometry
```

Figure 128. Geometry input for rotated holes example.

5.6 SUMMARY

This section has discussed the following:

- to use Fulcrum and KENO-VI to describe more advanced geometry models using geometry modification data and holes,
- to use the CHORD keyword to truncate a body with a plane perpendicular to a major axis,
- to use the ORIGIN keyword to translate the location of a body (i.e., shape),
- to use the ROTATE keyword to rotate a body to any angle, and
- to use the HOLE record to place one unit inside another unit.

The next step is to learn about even more complex geometries.

6. ADVANCED ARRAY TOPICS

6.1 WHAT YOU WILL BE ABLE TO DO

- Use Fulcrum and KENO-VI to describe complex geometry models using arrays.
- Remove unwanted portions of arrays using region definition vectors to apply boundaries.
- Build hexagonal arrays.

6.2 SHIPPING CONTAINER WITH FIVE CYLINDERS OF CRUCIFORM U(93.2)C CERAMIC ELEMENTS

This problem illustrates how to remove unwanted portions of an array using the region definition vector on the array record. A cruciform-shaped U(93.2)C ceramic element sits in water inside a SS316 cylinder (see Figure 129). Five of these cylinders are then placed inside a larger SS316 cylinder (Figure 130). The cruciform element is 10 cm on a side, with a width of 1 cm and a height of 20 cm. Each element is contained in a SS316 cylinder with an inner diameter of 12 cm and an inner height of 20 cm. The SS316 is 0.1 cm thick. These cylinders are packed in a cruciform-shaped array with a pitch of 12.4 cm. The five containers are then placed inside a SS316 cylinder with an inside diameter of 40 cm, an inner height of 20.2 cm, and thickness of 0.2 cm.

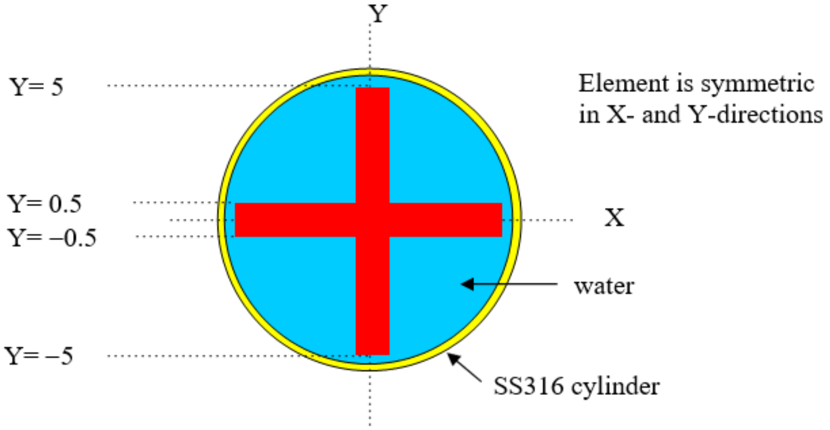


Figure 129. SS316 cylinder with U(93.2)C cruciform element.

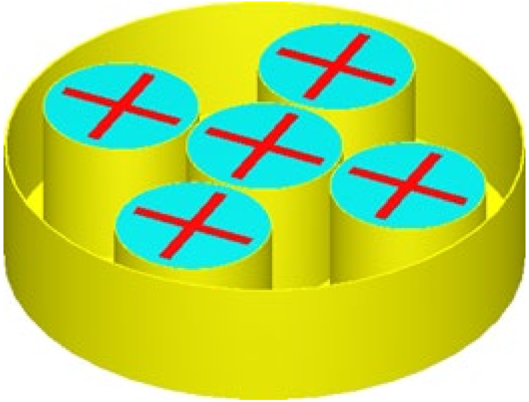
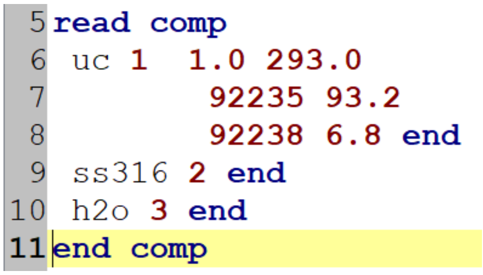


Figure 130. Five cruciform cylinders in shipping container.

Specify the sequence, title, cross section library and composition descriptions as done in previous sections of this primer. Note that there are basic standard compositions for water, UC, and SS316, which make the composition specifications straightforward. The finished composition block is shown in Figure 131.

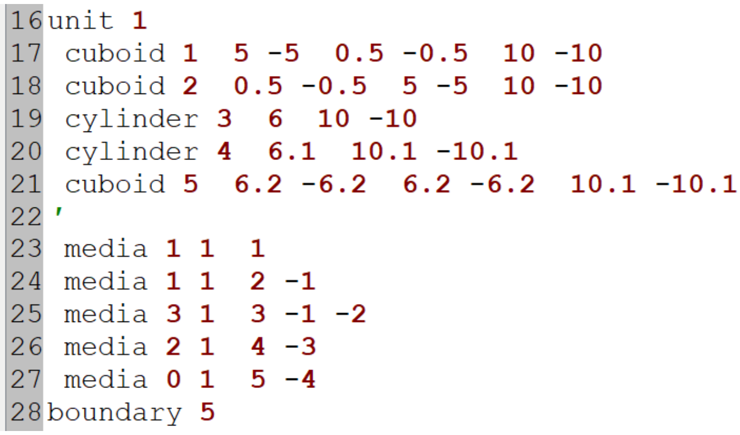


```
read comp
 uc 1  1.0 293.0
         92235 93.2
         92238 6.8 end
 ss316 2 end
 h2o 3 end
end comp
```

Figure 131. Composition block for cruciform array problem.

The cruciform elements can be formed by two intersecting cuboids. First create unit 1, which contains the cruciform element, water, and the SS316 cylinder. The outside region of unit 1 must ultimately be a cuboid because it will be used in a cuboidal array. First, remove **Global** from **Global unit 1.** Then create a cuboid to define the horizontal part of the UC cruciform with ±X dimensions of ±5, ±Y dimensions of ±0.5, and ±Z dimensions of ±10. This puts the origin in the center of the cruciform so it will be centered in the cylindrical can. Create another cuboid to define the vertical part of the cruciform. Use ±0.5 for the ±X dimensions, ±5 for the ±Y dimensions, and ±10 for the ±Z dimensions. The next volume is the cylinder containing the water, in which the cruciform will be placed. Create a **Cylinder**, with **Id** 3**,** and enter 6 for **R**, 10 for **Top**, and -10 for **Bottom**. Create another **Cylinder** for the SS316 shell. Use the **Id** of 4, and a radius of 6.1 cm, 10.1 for **Top**, and −10.1 for **Bottom**. The last volume in unit 1 is the cuboid outside the steel cylinder, which allows the use of this unit in an array. Use the **Id** 5 for this cuboid. Because the array pitch in the X and Y directions is 12.4 cm, the **±X** and **±Y** entries are ±6.2 (i.e., half of 12.4). The **±Z** entries are ±10.1 (same as the height of the steel cylinder).

Now enter the media records for unit 1; this can be done using either the configurable form or by entering the records directly. First is the uranium, mixture 1, in the horizontal cruciform element. Next is the vertical cruciform element, mixture 1, inside cuboid 2 but outside cuboid 1. Recall that the region of space that is within both cuboids must be included in one but excluded from the other to prevent the space being doubly defined. For the third media record, enter the water by specifying it inside cylinder 3 and outside cuboids 1 and 2. The next media entry is the SS316 cylindrical container: inside cylinder 4 and outside cylinder 3. The final media record for unit 1 is the void cuboid outside the steel cylinder. Specify it as inside cuboid 5 and outside cylinder 4. Finally, the boundary record region definition vector must be updated to inside 5. The geometry and contents for unit 1 should look like Figure 132.



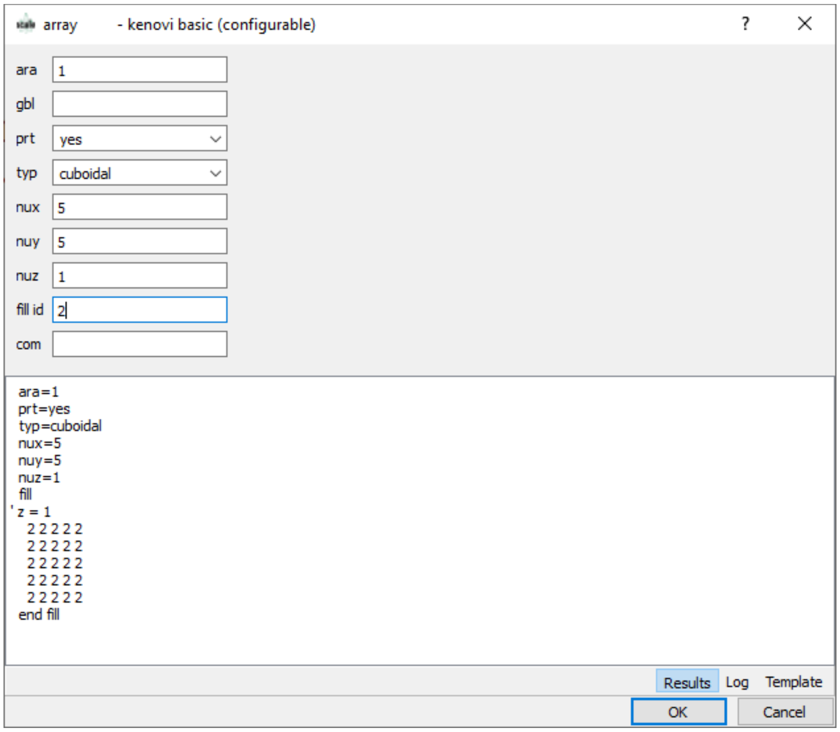
```
unit 1
 cuboid 1  5 -5  0.5 -0.5  10 -10
 cuboid 2  0.5 -0.5  5 -5  10 -10
 cylinder 3  6  10 -10
 cylinder 4  6.1  10.1 -10.1
 cuboid 5  6.2 -6.2  6.2 -6.2  10.1 -10.1
 '
 media 1 1  1
 media 1 1  2 -1
 media 3 1  3 -1 -2
 media 2 1  4 -3
 media 0 1  5 -4
boundary 5
```

Figure 132. Geometry input for unit 1.

Create a void cuboid unit as `unit 2`. This unit is needed to complete the array that will be used to position the five cans inside the larger cylinder. Create a **Cuboid** with the same outer dimensions that were used for the outer cuboid in unit 1. Then define the **Media** to fill the cuboid with void and define the `boundary` as inside cuboid `1` to complete unit 2.

Before defining the global unit, the array to be inserted in that unit must be defined. Note that the five individual containers are positioned to fill the five interior positions in a 3×3 array, with one unit in the top center position, three units along the center row, and one unit in the bottom center position. Because arrays cannot have empty elements, use unit 2 in the four corner positions of the 3×3 array. The 3×3 array does not entirely fill the inner volume of the shipping container. One option is to add a geometry record in the global unit with void filling it to define the space between the array and the shipping container wall. It is also possible to use the shipping container inner surface to truncate elements in the array, so a larger array with more void units can be used as the single contents record in the global unit. Therefore, a larger, 5×5 array will be constructed.

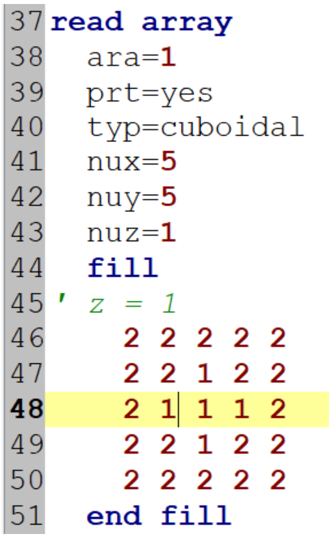
To start describing the array, place the cursor on the line after **end geometry** and enter add an array block, either by using the Fulcrum autocomplete capability or by typing the input directly into the input file. A single array will be defined, and is likely most simply started with the `CTRL-SPACE` autocomplete key combination. The default **ara** number of 1 can be used, and the defaults for **gbl**, **prt**, and **typ** can also be retained. The values for **nux** and **nuy** should be updated to `5`, but **nuz** will still be **1**. There are 25 elements in a 5×5 array, and only 5 cans so 20 of the 25 locations will be unit 2. The simplest choice for **fill id** is therefore `2`, so that only 5 **fill** entries need to be changed. The completed configurable form for this array is shown in Figure 133. After reviewing the input, press the **OK** button to add the array definition to the input file. The fill data must still be modified to add unit 1 in the appropriate locations. The central element (3,3) should be changed from **2** to `1`, as should the 4 face adjacent elements so that a total of 5 cans are included in the array. The final array input is shown in Figure 134.



ara	1
gbl	
prt	yes
typ	cuboidal
nux	5
nuy	5
nuz	1
fill id	2
com	

```
ara=1
prt=yes
typ=cuboidal
nux=5
nuy=5
nuz=1
fill
'z = 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
end fill
```

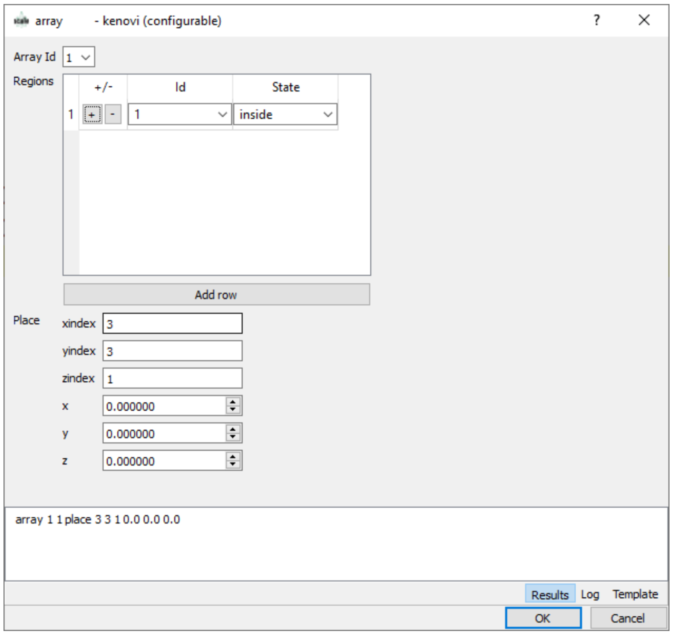
Figure 133. Completed array configurable for the 5×5 array.



```
read array
  ara=1
  prt=yes
  typ=cuboidal
  nux=5
  nuy=5
  nuz=1
  fill
' z = 1
    2 2 2 2 2
    2 2 1 2 2
    2 1 1 1 2
    2 2 1 2 2
    2 2 2 2 2
  end fill
```

Figure 134. Array definition.

Now return to the geometry block and create a new unit 3 as the global unit. The first geometry record is the inner cylinder of the shipping container. Create a **Cylinder** and enter `20` for **Radius**, `10.1` for **Top,** and `-10.1` for **Bottom**. The next record is the outer cylinder of the container. The shipping container is 0.2 cm thick, so the dimensions are `20.2` for **Radius**, `10.3` for **Top,** and `-10.3` for **Bottom**. With the curser between the unit declaration and the boundary record, the `CTRL-SPACE` autocomplete key combination to open an array contents record configurable form. The default **Array Id** in the upper left corner of the form is correct as only array 1 has been defined. Press the Add row button to add a region in the Regions pane; the default values of **Inside** and **Id 1** are correct. The simplest way to position the array is to use the central element and position it at the origin. So **xindex** and **yindex** should be set to 3 and **zindex** should be left as **1**. The coordinates for **x**, **y**, and **z** can all be left as **0**. The completed array contents record configurable form is shown in Figure 135; after confirming the input press the OK button to add the record to the input file. Not that the array is larger than the inner volume of the shipping container, but that it is truncated by cylinder 1. This feature can dramatically simplify the input description for some models.



Array Id: 1

Regions:

	Id	State
1	1	inside

Place

xindex: 3

yindex: 3

zindex: 1

x: 0.000000

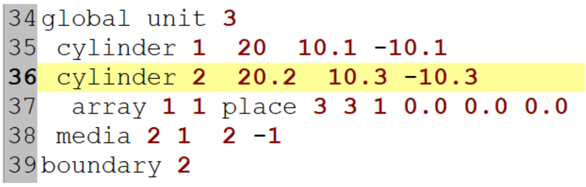
y: 0.000000

z: 0.000000

```
array 1 1 place 3 3 1 0.0 0.0 0.0
```

Figure 135. Completed array contents record configurable form.

Now, define the media record for the SS316 outer cylinder, mixture 2 as inside cylinder 2 and outside cylinder 1. Define the boundary of unit 3 as inside cylinder 2. The completed input for unit 3is provided in Figure 136. Save the changes and execute SCALE. The calculated k_{eff} value is 0.6277 ± 0.0018.



```
global unit 3
 cylinder 1  20  10.1 -10.1
 cylinder 2  20.2  10.3 -10.3
  array 1 1 place 3 3 1 0.0 0.0 0.0
 media 2 1  2 -1
boundary 2
```

Figure 136. Geometry input for unit 3.

6.3 HEXAGONAL ARRAY WITH INNER AND OUTER BOUNDARIES

This next problem illustrates how to create a hexagonal array and remove unwanted portions of the array using both inner and outer boundaries in the region definition vector on the array record. An annular carbon steel cylindrical tank contains six aluminum-clad U(10.0) cylinders submerged in water (Figure 137). The uranium cylinders are 15 cm in diameter and 18.3 cm high. The aluminum clad is 0.5 cm thick on the sides and bottom of the uranium. The cylinders have a center-to-center triangular pitch of 21 cm. The tank has an inner radius of 12 cm and an outer radius of 31.5 cm. The inside height is 18.8 cm. The tank walls and bottom are 1.5 cm thick. A cutaway view with highlighted edges in KENO3D (see Section 9.4) is shown in Figure 138.

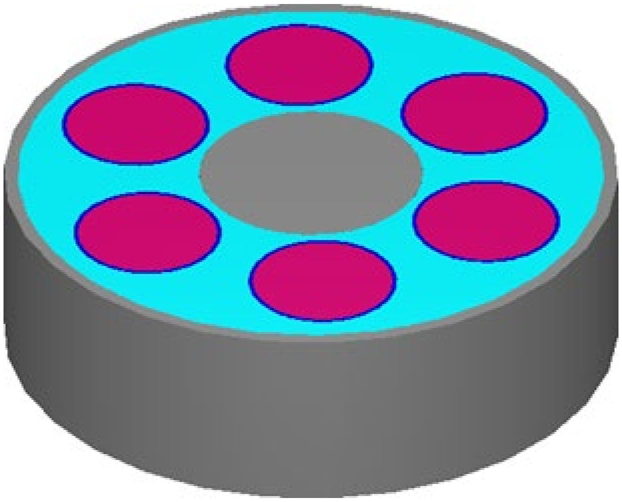


Figure 137. Annular carbon steel cylindrical tank with aluminum-clad uranium cylinders.

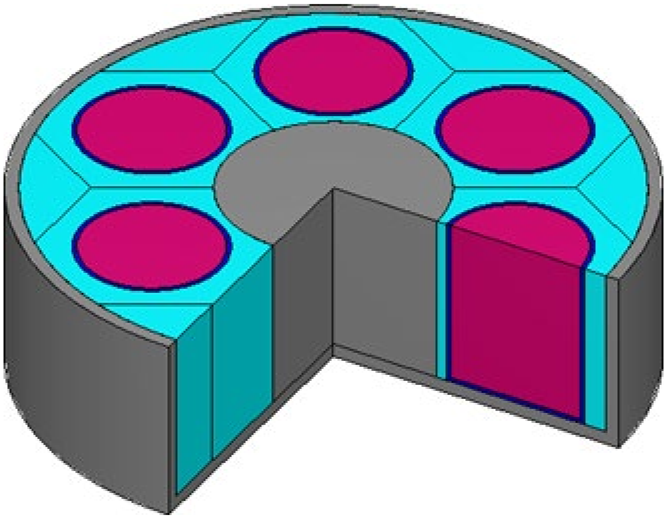
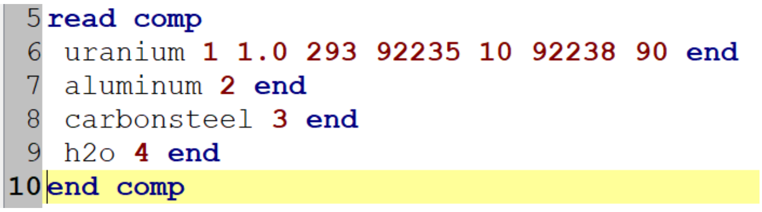


Figure 138. Cutaway view of tank with highlighted edges.

Specify the sequence, title, cross section library and composition descriptions as done in previous sections of this primer. Note that there are basic standard compositions for uranium, aluminum, carbon steel (`carbonsteel`), and water, which make the composition specifications straightforward. The finished composition block is shown in Figure 139.



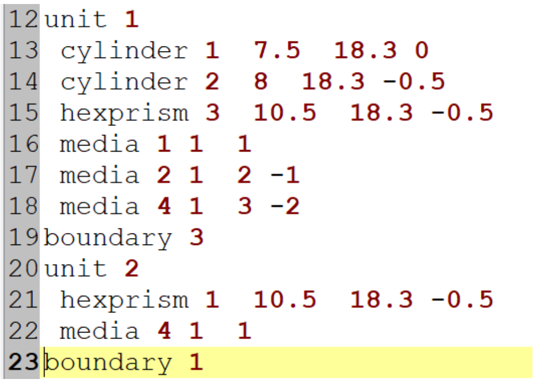
```
read comp
 uranium 1 1.0 293 92235 10 92238 90 end
 aluminum 2 end
 carbonsteel 3 end
 h2o 4 end
end comp
```

Figure 139. Composition block for hexagonal array example.

The first unit is not the global unit but a uranium cylinder with aluminum clad. Create a first **Cylinder** by entering the dimensions of the uranium cylinder: `7.5` for **Radius**, `18.3` for **Top**, and `0` for **Bottom**.

Next enter the data for the aluminum clad cylinder: 8 for **Radius**, 18.3 for **Top**, and -0.5 for **Bottom**. This cylinder must be surrounded with a hexprism of water so that this unit can be placed in a hexagonal array. Create a **hexprism** and enter 10.5 for **Radius**, 18.3 for **+Z**, and -0.5 for **-Z**. Note that the radius used to specify a hexprism or rhexprism is the radius of the inscribed circle; this dimension is also referred to as the apothem of the hexagon. The media records for unit 1 place uranium (mixture 1) inside cylinder 1, aluminum (mixture 2) inside cylinder 2 and outside cylinder 1, and water (mixture 4) inside hexprism 3 and outside cylinder 2. The boundary of unit 1 is inside hexprism 3.

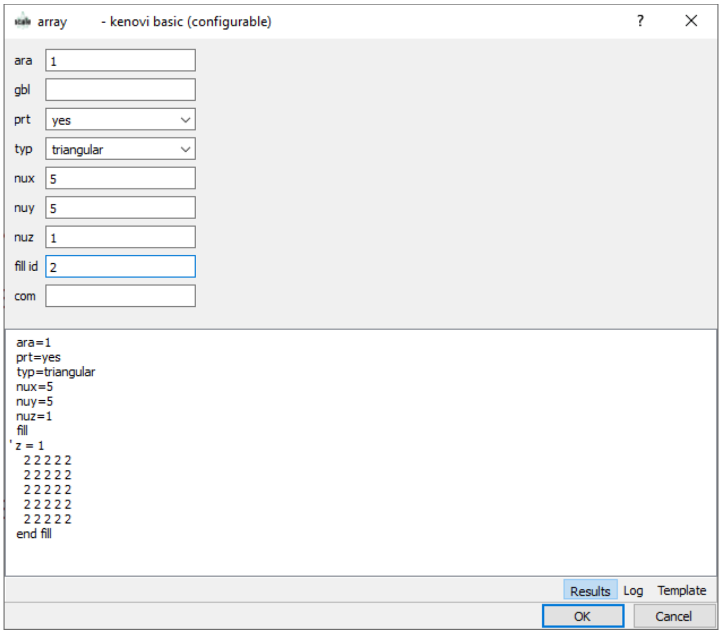
Unit 2 will be a hexprism of water that is so that empty hexagonal array positions can be included around the outside of the unit 1 hexprisms. (Portions of these units are visible in Figure 138.) Create a **hexprism** and enter 10.5 for **Radius**, 18.3 for **+Z**, and -0.5 for **-Z**. The only media record fills the hexprims with water (mixture 4). The boundary is defined by boundary 1. The geometry and contents records for units 1 and 2 are shown in Figure 140.



```
unit 1
 cylinder 1  7.5  18.3 0
 cylinder 2  8  18.3 -0.5
 hexprism 3  10.5  18.3 -0.5
 media 1 1  1
 media 2 1  2 -1
 media 4 1  3 -2
boundary 3
unit 2
 hexprism 1  10.5  18.3 -0.5
 media 4 1  1
boundary 1
```

Figure 140. Input for units 1 and 2 for hexagonal array example.

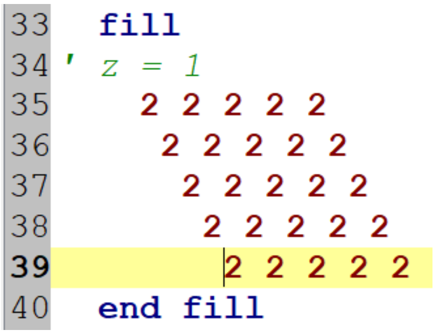
Now the hexagonal array can be defined. Add an array block after the geometry block, and use the CTRL-SPACE autocomplete key combination to begin the specification for array 1. Select triangular from the dropdown for the array type. Sizing the array to be 5 units in the X and Y directions, and 1 unit in the Z direction ensures that it will be large enough to fill the cylinder in the global unit and simplifies selecting the central unit. Specify **nux** and **nuy** as 5 and **nuz** as 1. The simplest unit to initialize the fill data with is unit 2 since there are 25 elements in the array and only 6 need to be changed to unit 1. The completed array configurable form is shown in Figure 141. After confirming the input, press the OK button to add the array definition to the input file. Since the triangular (hexagonal) array type shifts to the right with each layer in the Y direction, adding spaces to the input file can help identify the elements in the array to modify. An example of this spacing, prior to adding unit 1 to the array, is shown in Figure 142. Recall that the order of the Y entries is inverted in the input compared to the model as the entries are read in *increasing* order (see Figure 91). Either 6 or 7 elements can be changed from unit 2 to unit 1 because the central element will be removed from the array in the model. Changing only 6 entries is slightly simpler; change the third and fourth elements in the Y=2 rank of the array, the second and fourth elements in the Y=3 rank, and the second and third elements in the Y=3 rank. The final fill statement is shown in Figure 143. The complete array definition is shown in Figure 144.



ara	1
gbl	
prt	yes
typ	triangular
nux	5
nuy	5
nuz	1
fill id	2
com	

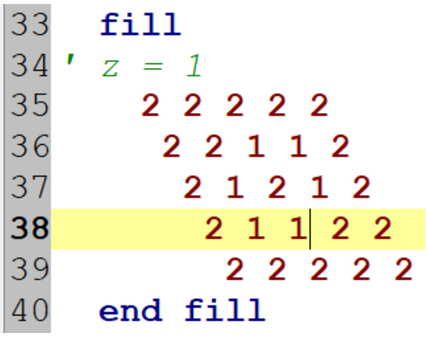
```
ara=1
prt=yes
typ=triangular
nux=5
nuy=5
nuz=1
fill
' z = 1
 2 2 2 2 2
 2 2 2 2 2
 2 2 2 2 2
 2 2 2 2 2
 2 2 2 2 2
end fill
```

Figure 141. Completed array configurable form for the hexagonal array example.



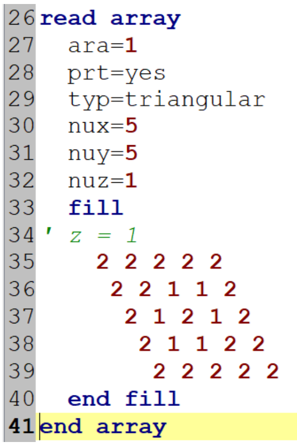
```
  fill
' z = 1
    2  2  2  2  2
     2  2  2  2  2
      2  2  2  2  2
       2  2  2  2  2
        2  2  2  2  2
  end fill
```

Figure 142. Fill statement with additional spacing to help visualize final array geometry.



```
  fill
' z = 1
    2  2  2  2  2
     2  2  1  1  2
      2  1  2  1  2
       2  1  1  2  2
        2  2  2  2  2
  end fill
```

Figure 143. Final fill statement for hexagonal array example.

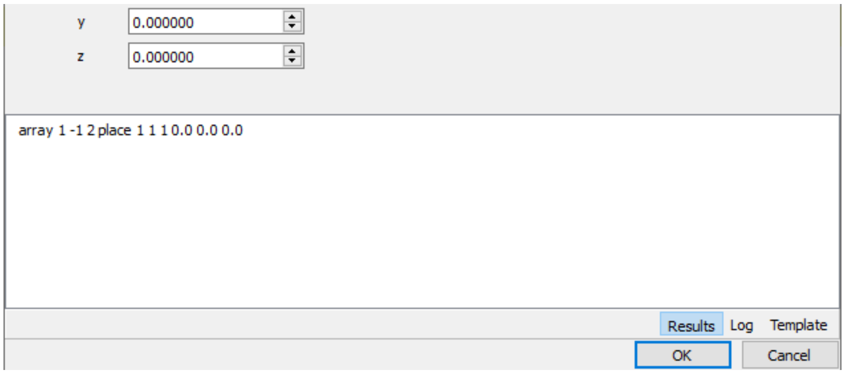


```
read array
  ara=1
  prt=yes
  typ=triangular
  nux=5
  nuy=5
  nuz=1
  fill
' z = 1
    2  2  2  2  2
      2  2  1  1  2
        2  1  2  1  2
          2  1  1  2  2
            2  2  2  2  2
  end fill
end array
```

Figure 144. Array definition for the hexagonal array.

Create global unit 3. The first geometry record is the central steel cylinder within the tank. Create a **Cylinder,** and enter `12` for **Radius**, `18.3` for **Top**, and `-0.5` for **Bottom**. The next record is the inner wall of the container, which has a **Radius** of `31.5` cm and the **Top** and **Bottom** of `18.3` and `-0.5`, respectively. The tank is 1.5 cm thick, so the outer dimensions for the third cylinder are `33` for **Radius**, `18.3` for **Top**, and `-2` for **Bottom**.

Next, create the array contents record. The simplest way to do this is by using the array contents configurable form. Click the **Add row** button to add a region to the **Regions** pane. Set the first row to be `outside` **Id 1** and add a second row to be `inside` **Id 2**. The geometry **Id** must be typed in the field. Note the region definition vector in the array contents record in the preview pane (as shown Figure 145) now shows that the array will be inside region 2 but outside region 1. The array must still be positioned and can be positioned with any unit in the array, regardless of whether that unit is used in the model or not. Placing the central element at the origin is the simplest approach for this model, so set **xindex** and **yindex** to `3` and leave **zindex** as **1**. The **x**, **y**, and **z** coordinates of the origin of element (3,3,1) are all **0**. The complete array contents record is shown in Figure 146; after confirming the input press the **OK** button to add the contents record to the input file. Now, define the media record to fill the central cylinder with carbon steel cylinder (mixture 3). The outer shell of the tank is also carbon steel (mixture 4) and is inside cylinder 3 and outside cylinder 2. Define the boundary of unit 3 as inside cylinder 3. Note that the region inside cylinder 2 but outside cylinder 1 is already filled with array 1 so all of the defined geometry in unit 3 has contents assigned. The completed unit 3 input is provided in Figure 147. Save the changes and execute SCALE. The calculated k_{eff} value is 0.6182 ± 0.0016.

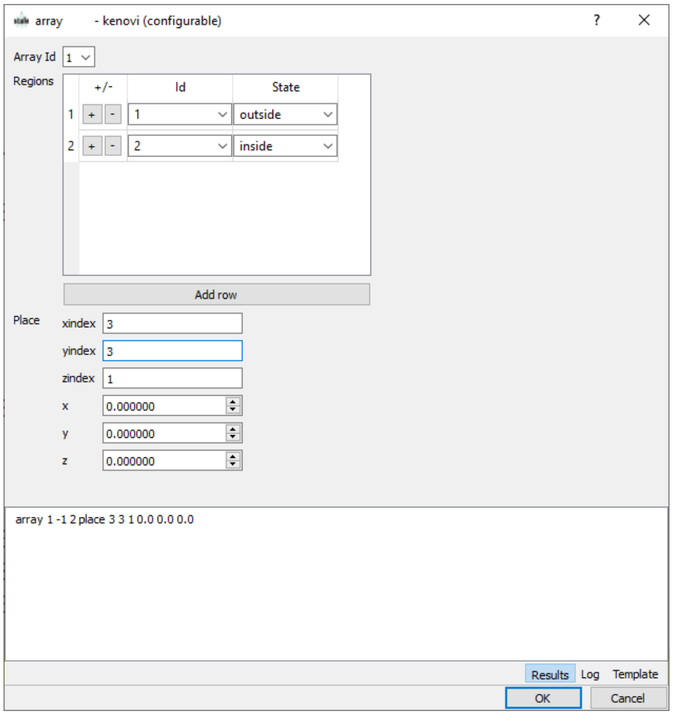


y 0.000000

z 0.000000

array 1 -1 2 place 1 1 1 0.0 0.0 0.0

Figure 145. Preview pane of array contents record.



Array Id 1

Regions

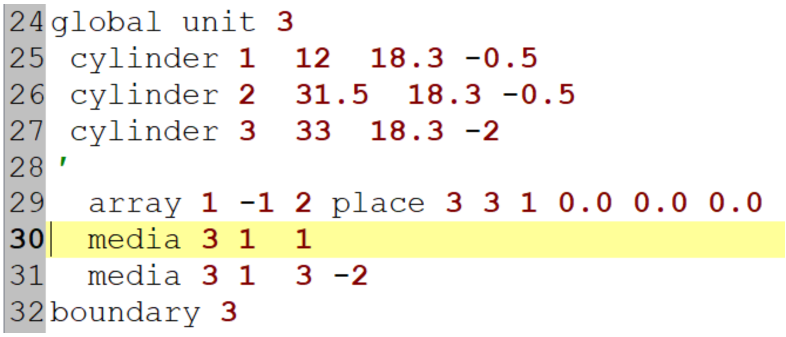
	+/-	Id	State
1	+ -	1	outside
2	+ -	2	inside

Place

xindex	3
yindex	3
zindex	1
x	0.000000
y	0.000000
z	0.000000

```
array 1 -1 2 place 3 3 1 0.0 0.0 0.0
```

Figure 146. Complete array contents configurable form for the hexagonal array example.



```
global unit 3
 cylinder 1  12  18.3 -0.5
 cylinder 2  31.5  18.3 -0.5
 cylinder 3  33  18.3 -2
'
  array 1 -1 2 place 3 3 1 0.0 0.0 0.0
  media 3 1  1
  media 3 1  3 -2
boundary 3
```

Figure 147. Unit 3 for the hexagonal array example.

6.4 SUMMARY

This section has discussed the following:

- to use Fulcrum and KENO-VI to describe complex geometry models using arrays,
- to remove unwanted portions of arrays using region definition vectors to apply boundaries, and
- to build hexagonal arrays.

7. NONSTANDARD MATERIALS

Now that you understand the geometry and basic standard composition input requirements, this section provides an explanation of a few inputs you might find useful for problems involving compounds or other materials not available in the Standard Composition Library.

7.1 WHAT YOU WILL BE ABLE TO DO

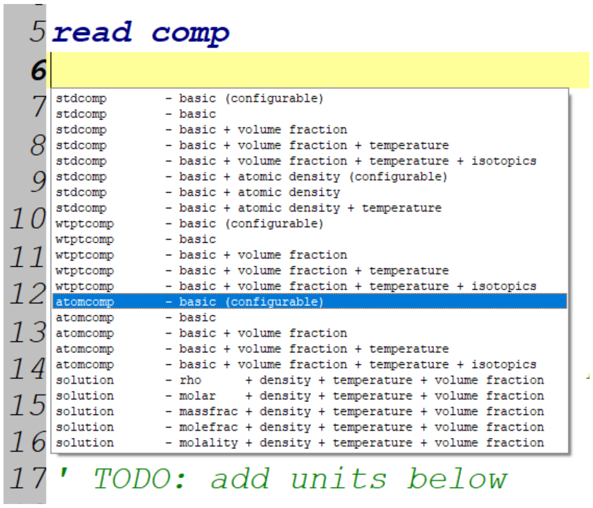
- Define user-defined (non-standard) materials using Fulcrum.
- Define a mixture by using a combination of user-defined material and a basic standard composition.

7.2 USER-DEFINED MATERIALS

Section 3 covered entry of information for elements, isotopes, and compounds found in the Standard Composition Library. However, there are some materials that are not available in the Standard Composition Library in SCALE. These materials can be defined using the **Alloy or Mixture (wtpt)** or the **Compound (atom)** configurable forms, or placeholder templates from the material specification menu form used in Section 3 (see, for example, Figure 148). This approach works best for materials in which the chemical composition or weight fraction is provided. If a compound or alloy is in the Standard Composition Library, or if number densities are available for each of the constituents of the desired material, then using the basic standard composition input is likely more efficient and less error prone.

7.2.1 Description Based on Chemical Formula

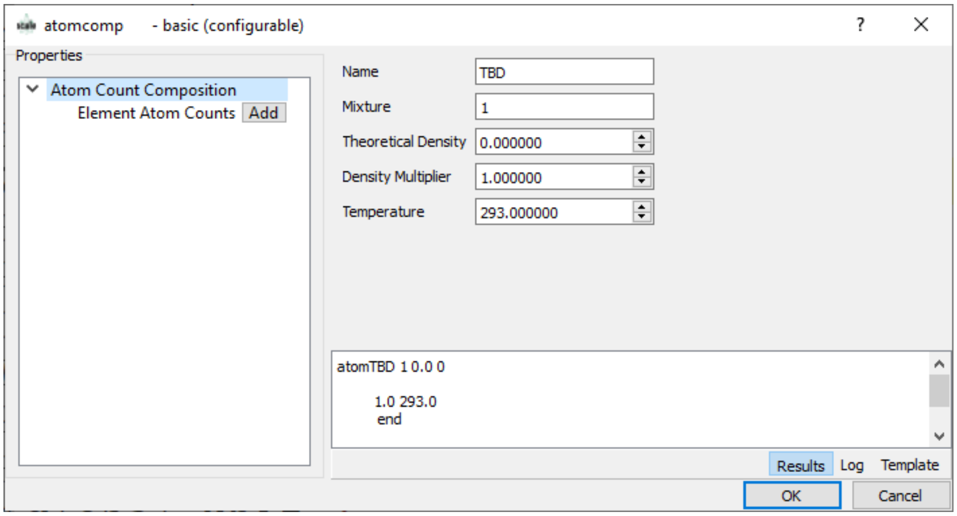
In this example, enter the material information for hydraulic fluid, C_2H_6SiO, with a density of 0.97 g/cc. Move the cursor into the composition block between **Read Comp** and **End Comp** and press `CTRL-SPACE` to trigger the dropdown menu containing the list of available composition configurable and placeholder options. From this list, click select **atomcomp – basic (configurable)** as shown in Figure 148.



```
read comp

stdcomp       - basic (configurable)
stdcomp       - basic
stdcomp       - basic + volume fraction
stdcomp       - basic + volume fraction + temperature
stdcomp       - basic + volume fraction + temperature + isotopics
stdcomp       - basic + atomic density (configurable)
stdcomp       - basic + atomic density
stdcomp       - basic + atomic density + temperature
wtptcomp      - basic (configurable)
wtptcomp      - basic
wtptcomp      - basic + volume fraction
wtptcomp      - basic + volume fraction + temperature
wtptcomp      - basic + volume fraction + temperature + isotopics
atomcomp      - basic (configurable)
atomcomp      - basic
atomcomp      - basic + volume fraction
atomcomp      - basic + volume fraction + temperature
atomcomp      - basic + volume fraction + temperature + isotopics
solution      - rho      + density + temperature + volume fraction
solution      - molar    + density + temperature + volume fraction
solution      - massfrac + density + temperature + volume fraction
solution      - molefrac + density + temperature + volume fraction
solution      - molality + density + temperature + volume fraction
' TODO: add units below
```

Figure 148. Dropdown menu selection of atom compound configurable.

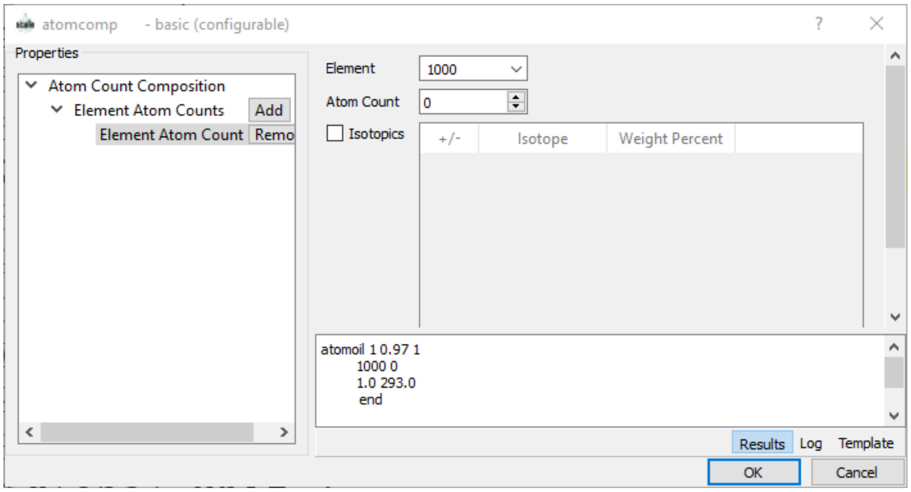
The configurable from should appear and should look like Figure 149, although you may have to expand the left pane to see the **Add** button. In the **Name** portion of the configurable, replace **TBD** with `oil`, and enter `0.97` in the **Theoretical Density** box. Next, click on the **Add** button in the left pane to begin adding the elements to the compound. The configurable form should change to look like Figure 150 below. Using the new right pane, click on the dropdown menu next to **Element**, select `6000` for carbon, and enter `2` in the **Atom Count** box below it. This process should be repeated using **Element** IDs of `1000`, `14000`, and `8000`, and atom counts of `6`, `1`, and `1` for hydrogen, silicon, and oxygen, respectively, while pressing the **Add** button between each elemental entry. The final configurable form should look like the one in Figure 151 and create input like that shown in Figure 152.



Name	TBD
Mixture	1
Theoretical Density	0.000000
Density Multiplier	1.000000
Temperature	293.000000

```
atomTBD 1 0.0 0

     1.0 293.0
     end
```

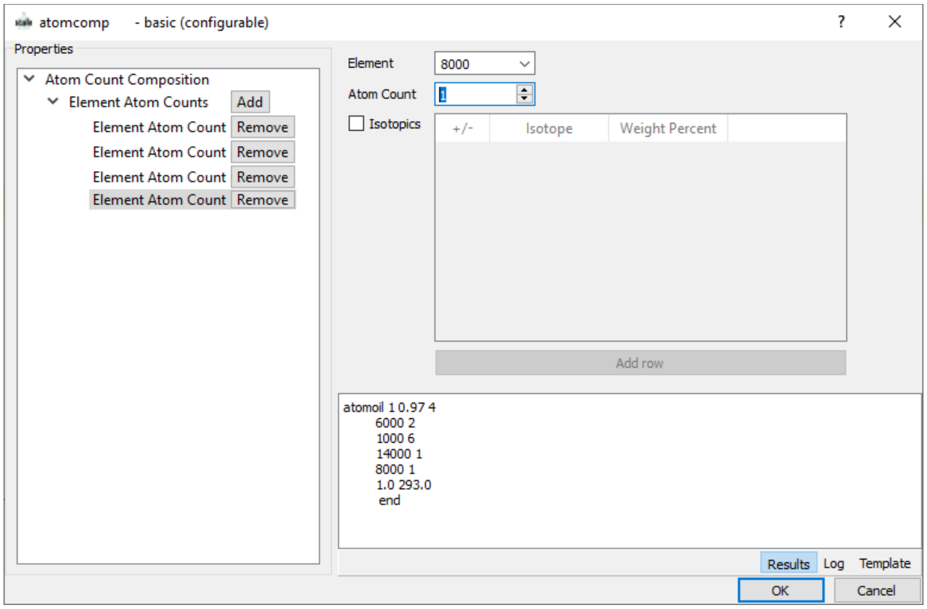
Figure 149. Depiction of initial atom compound configurable.



Element	1000
Atom Count	0
☐ Isotopics	

```
atomoil 1 0.97 1
     1000 0
     1.0 293.0
     end
```

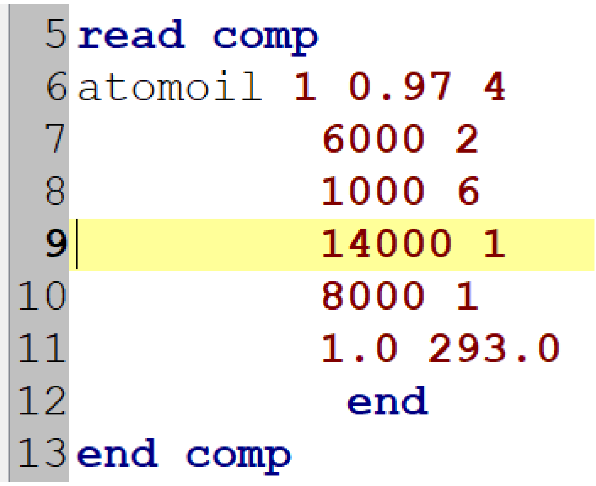
Figure 150. Depiction of element addition portion of the compound configurable.



Element: 8000
Atom Count: 1
☐ Isotopics

```
atomoil 1 0.97 4
     6000 2
     1000 6
     14000 1
     8000 1
     1.0 293.0
      end
```

Figure 151. Second depiction of element addition portion of the compound configurable.

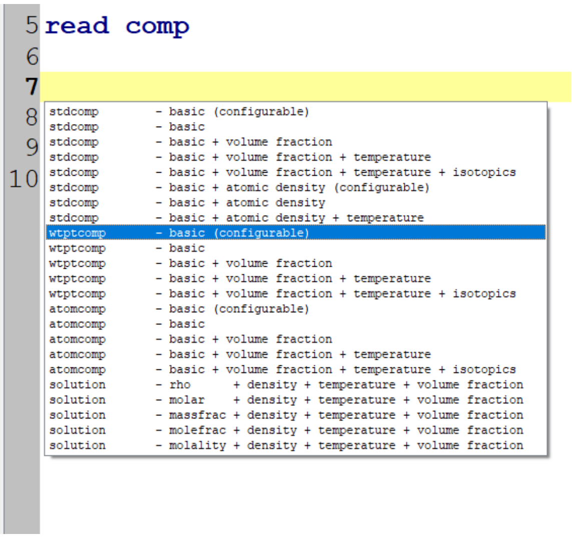


```
read comp
atomoil 1 0.97 4
                6000 2
                1000 6
                14000 1
                8000 1
                1.0 293.0
                  end
end comp
```

Figure 152. User-defined atom composition final input.

7.2.2 Description Based on Weight Percent

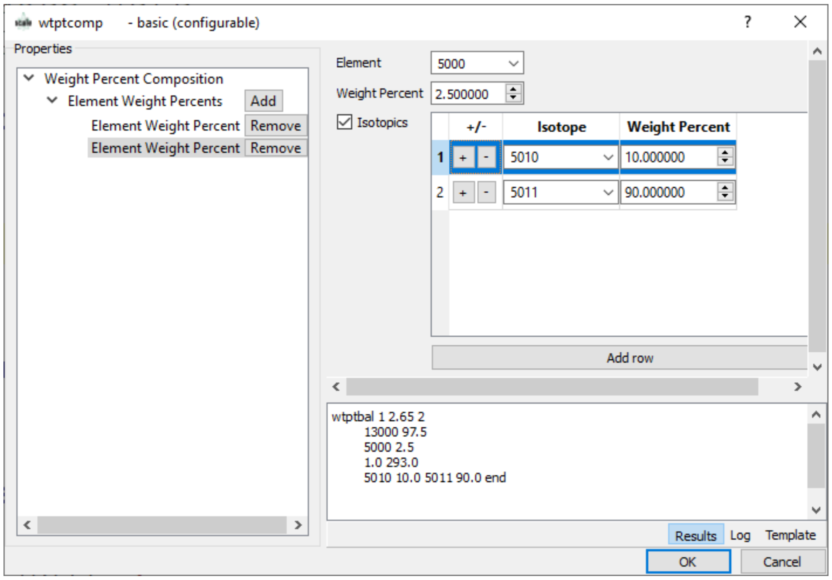
In this example, enter the material information for borated aluminum that is 2.5 wt.% boron and has a density of 2.65 g/cm^3. The boron is 10 wt.% ^{10}B and 90 wt.% ^{11}B. For this material, the weight percents are known, so the information should be entered in terms of weight percent, which is the **Alloy or Mix (Wt %)** type. Move the cursor into the composition block between **Read Comp** and **End Comp** and press `CTRL-SPACE` to trigger the dropdown menu containing the list of available composition configurable and placeholder options. From this list, select **wtptcomp – basic (configurable)**, as is shown in Figure 153.



```
read comp

stdcomp        - basic (configurable)
stdcomp        - basic
stdcomp        - basic + volume fraction
stdcomp        - basic + volume fraction + temperature
stdcomp        - basic + volume fraction + temperature + isotopics
stdcomp        - basic + atomic density (configurable)
stdcomp        - basic + atomic density
stdcomp        - basic + atomic density + temperature
wtptcomp       - basic (configurable)
wtptcomp       - basic
wtptcomp       - basic + volume fraction
wtptcomp       - basic + volume fraction + temperature
wtptcomp       - basic + volume fraction + temperature + isotopics
atomcomp       - basic (configurable)
atomcomp       - basic
atomcomp       - basic + volume fraction
atomcomp       - basic + volume fraction + temperature
atomcomp       - basic + volume fraction + temperature + isotopics
solution       - rho      + density + temperature + volume fraction
solution       - molar    + density + temperature + volume fraction
solution       - massfrac + density + temperature + volume fraction
solution       - molefrac + density + temperature + volume fraction
solution       - molality + density + temperature + volume fraction
```

Figure 153. Dropdown menu selection of wtpt compound configurable.

A configurable form similar to that used for the atom compound material (Figure 149) entries will appear. For **Composition name**, replace **TBD** with `bal` to create the material `wtptbal`. Now enter the **Theoretical Density** of `2.65` g/cc and leave the **Density Multiplier** and **Temperature** at the default values. In the left pane of the configurable, click the **Add** button (the pane may need to be expanded to display the button). This will allow entry of individual nuclides. In the **Element** dropdown menu, select `13000` for aluminum, and enter `97.5` in the **Weight Percent** box. Back in the left pane, click the **Add** button again, select `5000` for boron from the **Element** dropdown menu, and enter `2.5` for **Weight Percent**. Because there is an isotopic specification for this example, check the **Isotopics** box, and click **Add row** twice. Adjust one of the **5010 Isotope** values to be `5011`, enter `90.0` for the **Weight Percent** value corresponding to **5011**, and enter `10.0` in the **Weight Percent** box corresponding to **5010**. When finished, the configurable from should look like Figure 154. If the configurable looks acceptable, click **OK**, and the final input will look like Figure 155.



wtptcomp - basic (configurable)

Properties

- Weight Percent Composition
  - Element Weight Percents Add
    - Element Weight Percent Remove
    - Element Weight Percent Remove

Element: 5000

Weight Percent: 2.500000

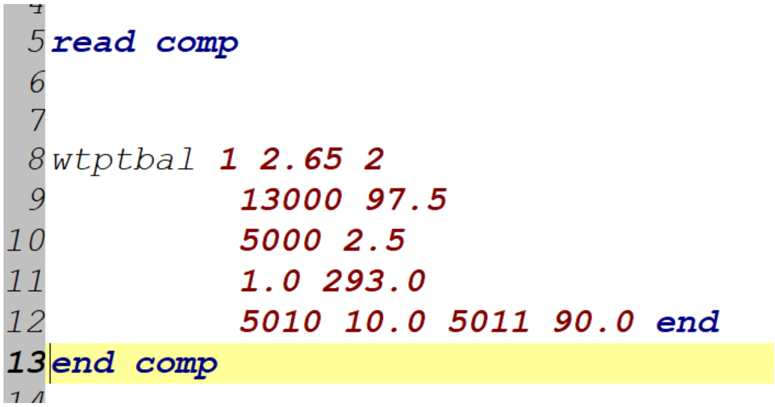
☑ Isotopics

	+/-	Isotope	Weight Percent
1	+ -	5010	10.000000
2	+ -	5011	90.000000

Add row

```
wtptbal 1 2.65 2
      13000 97.5
      5000 2.5
      1.0 293.0
      5010 10.0 5011 90.0 end
```

Figure 154. Final configurable for wtpt composition entry.



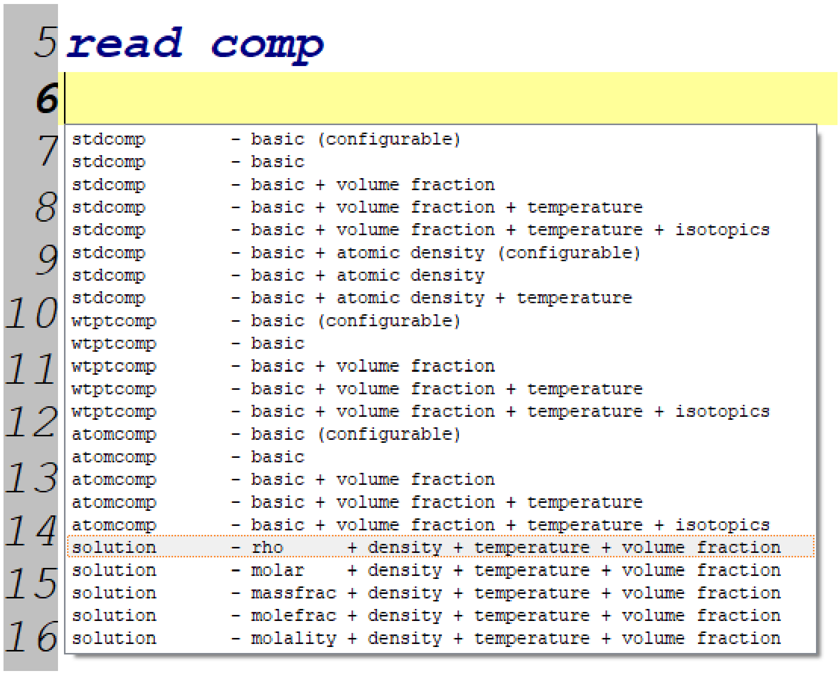
```
read comp


wtptbal 1 2.65 2
         13000 97.5
         5000 2.5
         1.0 293.0
         5010 10.0 5011 90.0 end
end comp
```

Figure 155. User-defined wtpt composition final input.

7.2.3 Description Based on Actinide Solution

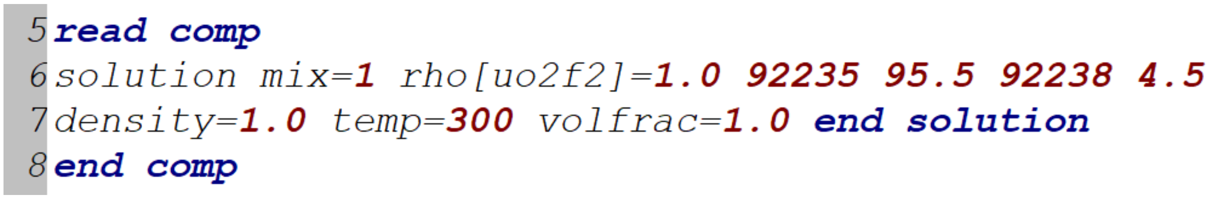
The next method of entering material composition information uses SCALE’s actinide solution composition specification method. In this example, specify a UO_2F_2 solution containing 907.0 gU/L, with an overall solution density of 2.0289 g/cm^{3}. The uranium is enriched to 5 wt.% ^{235}U; neither ^{234}U nor ^{236}U will be included in this example. At this time, a configurable form for actinide solutions is not present within the Fulcrum interface, so the placeholder method is used to enter this information.

Move the cursor into the composition block between **Read Comp** and **End Comp**, and press `CTRL-SPACE` to trigger the dropdown menu containing list of available composition configurable and placeholder options. The dropdown menu is displayed with the five solution specification options shown in Figure 156. The variation placeholder options for solution specification are designed to account for the various methods of expressing the actinide concentration in the solution. In this example, the concentration is expressed in terms of the number of grams of uranium in the solution per liter. The appropriate selection for this set of units is **solution – rho + density + temperature + volume fraction**. Selecting that option yields the initial input found in Figure 157.



```
read comp

stdcomp        - basic (configurable)
stdcomp        - basic
stdcomp        - basic + volume fraction
stdcomp        - basic + volume fraction + temperature
stdcomp        - basic + volume fraction + temperature + isotopics
stdcomp        - basic + atomic density (configurable)
stdcomp        - basic + atomic density
stdcomp        - basic + atomic density + temperature
wtptcomp       - basic (configurable)
wtptcomp       - basic
wtptcomp       - basic + volume fraction
wtptcomp       - basic + volume fraction + temperature
wtptcomp       - basic + volume fraction + temperature + isotopics
atomcomp       - basic (configurable)
atomcomp       - basic
atomcomp       - basic + volume fraction
atomcomp       - basic + volume fraction + temperature
atomcomp       - basic + volume fraction + temperature + isotopics
solution       - rho      + density + temperature + volume fraction
solution       - molar    + density + temperature + volume fraction
solution       - massfrac + density + temperature + volume fraction
solution       - molefrac + density + temperature + volume fraction
solution       - molality + density + temperature + volume fraction
```

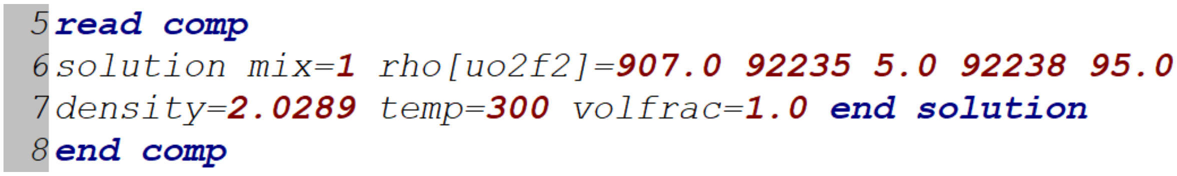
Figure 156. User defined solution composition final input.



```
read comp
solution mix=1 rho[uo2f2]=1.0 92235 95.5 92238 4.5
density=1.0 temp=300 volfrac=1.0 end solution
end comp
```

Figure 157. Input specified by placeholder insertion.
(normally placed on one line by Fulcrum but shown on two lines for readability)

Once the placeholder input has been inserted in the input file, it must be modified to correctly model the composition. In this case, the placeholder has the correct chemical form that accompanies the fissile species, **uo2f2**, but the number of grams per liter (**rho)** should be changed from **1.0** to `907.0`. The isotopic specification should be modified from **92235 95.5 92238 4.5** to `92235 5.0 92238 95.0`, and the density of the solution should be modified from **1.0** to `2.0289`. For this case, there is no excess acid specified; however, if there were, it would be specified as `molar[hfacid]=X`, where X is the molarity of the acid in the solution. The final solution input is shown Figure 158.



```
read comp
solution mix=1 rho[uo2f2]=907.0 92235 5.0 92238 95.0
density=2.0289 temp=300 volfrac=1.0 end solution
end comp
```

Figure 158. Complete solution input.

7.3 SUMMARY

This section has demonstrated how to use Fulcrum to:

- define user-defined material based on a known chemical formula,
- define user-defined material for a mixture with known weight percents of constituents, and
- define actinide solutions.

8. UNIT CELLS AND BOUNDARY CONDITIONS

The examples thus far have been limited to relatively large homogeneous systems in which the default infinite homogeneous medium unit cell type was appropriate. CE cross sections were also used to avoid any errors associated with improper MG cross section treatment. Cross section processing for resonance self-shielding is only necessary for MG libraries. The resonance self-shielding is a result of using group-average cross sections in the transport solution in KENO. In this section, more heterogeneous systems will be presented, along with demonstrations of how to use the other unit cell types appropriately. Transport results determined with CE cross sections can provide a reference solution to ensure that the MG cross section processing has been performed correctly in most cases.

8.1 WHAT YOU WILL BE ABLE TO DO

- Understand the difference in cross section processing for infinite homogeneous medium, lattice cell, multiregion, and doubly heterogeneous analyses.
- Identify which unit cell type is appropriate to use for various heterogeneous systems.
- Use Fulcrum to define LATTICECELL, MULTIREGION, and DOUBLEHET unit cells.
- Define different boundary conditions for MULTIREGION and DOUBLEHET unit cells.
- Use boundary conditions to create reflective boundaries for a KENO-VI global unit.

8.2 CALCULATION TYPES FOR PROBLEM-DEPENDENT CROSS SECTIONS

XSProc is primarily used to prepare data for creating the problem-dependent cross section libraries required by CSAS. This is particularly important for heterogeneous systems and large regular lattices of slabs, pins, or spheres. XSProc uses a unit cell description to provide information for the resonance self-shielding corrections and the Dancoff corrections (where appropriate) that are applied to the cross sections to create a problem-dependent cross section library. A unit cell is typically a portion of the system containing fuel and, if applicable, moderator that can be used to define the neutron spectrum characteristics for part of the problem.

8.3 UNIT CELL TYPES

XSProc offers four types of unit cell calculations: INFHOMMEDIUM, LATTICECELL, MULTIREGION, and DOUBLEHET. Note that multiple unit cells may be defined in a single CSAS/KENO-VI input file. Each mixture number may be used only once in any unit cell, except for void (mixture 0), which can be used as frequently as needed.

8.3.1 Infinite Homogeneous Medium

The INFHOMMEDIUM treatment is best suited for large masses of materials in which (1) the size of each material is large compared with the average mean-free path of neutrons in the material, (2) the size of each material is very small compared with the average mean-free path, or (3) the fraction of the material that is within one mean-free path from the surface of the material is very small. An example of the first situation is a large solution tank; the neutron mean-free path may be on the order of a centimeter so a tank would only need to be a few centimeters in diameter to count as “large”. This example also satisfies the third condition that a small fraction of the solution is within a mean-free path of the boundary. The second situation is more rare, and a good example is likely to be a fast reactor. In these systems, the mean free path may be several centimeters (or more) and the neutrons do not interact on the same distance scale as the heterogeneity present for heat transfer. Every material specified in the problem that is not specified in a unit cell is treated as an infinite homogeneous lump. A system composed of small fuel lumps, such as rods, slabs, or pebbles, should not be treated as an infinite homogeneous medium.

8.3.2 Latticecell

The LATTICECELL treatment is appropriate for large arrays of fuel in slabs, pins/rods, or spheres. When the LATTICECELL treatment is used in XSProc, a set of point fluxes is calculated for a set of mesh intervals that make up the unit cell. The Wigner-Seitz equivalent cylindrical cell is used, except for the SQUAREPITCH option, and the cell is assumed to have white boundary conditions. The SQUAREPTICH option for the LATTICECELL treatment uses a 2D method of characteristics (MoC) solver in place of the 1D discrete ordinates solver that is used for other unit cell types. The cross sections for the mixtures specified in the cell are then weighted using these point fluxes and the CE cross sections. Limitations of the LATTICECELL treatment include the following:

1. The cell description is limited to a 1D cell. Most physical systems consist of 3D geometry, but in many instances, a 1D representation is adequate, as in the case of a large array of spheres in a regular lattice, a large array of slabs, or a large array of long uniform pins in a regular lattice. A 1D representation may be inadequate for systems exhibiting a 2D or 3D spatial dependence.

2. The LATTICECELL treatment assumes an infinite array of 1D cells. This assumption is an excellent approximation for large arrays of long fuel pins or large arrays of spheres. The approximation becomes less rigorous for short fuel pins and/or small arrays in which multidimensional spatial dependence or leakage can become important.

8.3.3 Multiregion

The MULTIREGION treatment is appropriate for geometric regions in which the geometry effects may be important, but the infinite homogeneous treatment or lattice cell treatment is not general enough. The MULTIREGION unit cell allows more flexibility in the placement of the fuel, but it requires all regions of the cell to have the same geometric shape (i.e., slab, cylinder, sphere, buckled slab, or buckled cylinder). Lattice arrangements can be approximated by specifying a white or reflective boundary condition on the outer boundary. The white boundary should be used for curved outer boundaries (cylindrical of spherical) and the mirror boundary condition should be used for planar boundaries (slab). Limitations of the MULTIREGION cell treatment are listed below.

1. A MULTIREGION cell is limited to a 1D approximation of the system being represented. This constraint is appropriate for a sphere, an infinitely long cylinder, a slab, or an infinite array of slabs.

2. The shape of the outer boundary of the MULTIREGION cell is the same as the shape of the inner regions. Cells with curved outer surfaces cannot be stacked to represent arrays. However, arrays can be represented by making a curved outer region that is of equal volume to the cuboidal outer region and then specifying a white outer boundary condition.

3. The boundary conditions available in a MULTIREGION problem include vacuum (escape at the boundary), reflected (reflected about the normal to the surface at the point of impact), periodic (a particle exiting the surface effectively enters an identical cell having the same orientation and continues traveling in the same direction), and white (isotropic return about the point of impact). Reflected and periodic boundary conditions on a slab can represent real physical situations, but they are not valid on a curved outer surface.

4. A MULTIREGION cell represents a single cell if the outer boundary has a vacuum boundary condition applied to it. A point flux spectrum is calculated over a set of intervals that make up the unit cell with the incoming flux on the boundary set to zero. This flux spectrum is then used to collapse a set of cross sections.

5. If the outer boundary of a MULTIREGION cell has a boundary condition other than vacuum, then the unit cell approximates an array. A point flux spectrum is calculated over a set of intervals that make up the unit cell with the incoming flux to the outgoing flux on the boundary. This flux spectrum is then used to weight a set of cross sections. For a slab having a periodic or reflected boundary condition, the incoming and outgoing fluxes are exact, but for a curved surface, the incoming flux is approximated by an isotropic return flux.

NOTE: The MULTIREGION cell treatment must be used to apply boundary conditions to your unit cell model. These boundary conditions are completely unrelated to the boundary condition of the outer surface of the KENO-VI model.

8.3.4 Double-heterogeneous

DOUBLEHET cells use a specialized XSProc calculational approach to treat resonance self-shielding in “doubly heterogeneous” systems. The fuel for these systems typically consists of small, heterogeneous, spherical fuel particles (grains) embedded in a moderator matrix to form a fuel compact. The fuel-grain/matrix compact constitutes the first level of heterogeneity. Cylindrical (rod), spherical (pebble), or slab fuel elements composed of the compact material are arranged in a moderating medium to form a regular or irregular lattice, producing the second level of heterogeneity. The fuel elements are also referred to as “macro cells.” Advanced reactor fuel designs that use tri-structural isotropic (TRISO) or fully ceramic microencapsulated (FCM) fuel require the DOUBLEHET treatment to account for both levels of heterogeneities in the self-shielding calculations. Simply ignoring the double-heterogeneity by volume-weighting the fuel grains and matrix material into a homogenized compact mixture is likely to result in a large reactivity bias.

In the DOUBLEHET cell input, the keywords and the geometry description for grains are similar to those of the MULTIREGION treatment, while the keywords and the geometry for the fuel element (macro-cell) are similar to those of the LATTICECELL treatment. The following rules apply to the DOUBLEHET cell treatment and must be followed. Violation of any rules may cause a fatal error.

1. As many grain types as needed may be specified for each unique fuel element. Note that grain type is different from the number of grains of a certain type. For example, a fuel element that contains both UO_2 and PuO_2 grains has two grain types. The same fuel element may contain 10,000 UO_2 grains and 5,000 PuO_2 grains. In this case, the number of grains of type UO_2 is 10,000, and the number of grains of type PuO_2 is 5,000.

2. As many fuel elements as needed may be specified, each of which requires its own DOUBLEHET cell. This may be the case for systems with many fuel elements at different fuel enrichments, burnable poisons, etc. Each fuel element may have one or more grain types.

3. Since the grains are homogenized into a new mixture to be used in the fuel element (macro-cell) cell calculation, a unique fuel mixture number must be entered. XSProc creates a new material with the new mixture number designated by the keyword fuelmix=, containing all the nuclides that are homogenized. The user must assign the new mixture number in the transport solver geometry (e.g., KENO) input unless a cell-weighted mixture is created.

4. The type of lattice or array configuration for the fuel-element may be spheres on a triangular pitch (SPHTRIANGP), spheres on a square pitch (SPHSQUAREP), annular spheres on a triangular pitch (ASPHTRIANGP), annular spheres on a square pitch (ASPHSQUAREP), cylindrical rods on a triangular pitch (TRIANGPITCH), cylindrical rods on a square pitch (SQUAREPITCH), annular cylindrical rods on a triangular pitch (ATRIANGPITCH), annular cylindrical rods on a square pitch (ASQUAREPITCH), a symmetric slab (SYMMSLABCELL), or an asymmetric slab (ASYMSLABCELL).

5. If there is only one grain type for a fuel element, the user must enter either the pitch, the aggregate number of particles in the element, or the volume fraction for the grains. The code needs the pitch and will directly use it if entered. If pitch is not given, then the volume fraction (if given) is used to calculate the pitch. If neither the pitch nor the volume fraction is given, then the number of particles is used to calculate the pitch and the volume fraction. The user should only enter one of these items.

   If the fuel matrix contains more than one grain type, all types are homogenized into a single mixture for the compact. As for the case of one grain type, the pitch is needed for the spherical cell calculations. However, the pitch by itself is not sufficient to perform the homogenization. Since each grain’s volume is known (grain dimensions must always be entered), entering the number of particles for each grain type essentially provides the total volume of each grain type and therefore enables the calculation of the volume fraction and the pitch. Likewise, entering the volume fraction for each grain type essentially provides the total volume of each grain type and therefore enables the calculation of the number of particles and the pitch. Therefore, one of these two quantities must be entered for multiple grain types. In these cases, since pitch is not given, the available matrix material is distributed around the grains of each grain type proportional to the grain volume and is used to calculate the corresponding pitch. Over-specification is allowed as long as the values are not inconsistent to greater than 0.01%.

6. For cylindrical rods and for slabs, fuel height must also be specified. For slabs, the slab width must also be specified.

7. The CENTRM calculation option must be Sn.

8.3.5 Unit Cell Boundary Conditions

For the MULTIREGION unit cell and the element level of the DOUBLEHET unit cell, there are four possible boundary conditions. The default for the left or inner boundary is reflected (required for a cylinder or sphere), and the default for the right or outer boundary is vacuum.

8.3.5.1 Vacuum Boundary Condition

A vacuum boundary condition means that no neutrons will re-enter the boundary. Thus, any neutron exiting the system through a vacuum boundary is permanently lost to the system. This condition is shown in Figure 159. (NOTE: In the following figures, a dashed arrow indicates neutrons leaving the system, whereas a solid arrow represents neutrons returning to the system. The length of the arrow is proportional to the number of neutrons, so longer arrows represent more neutrons than shorter arrows.)

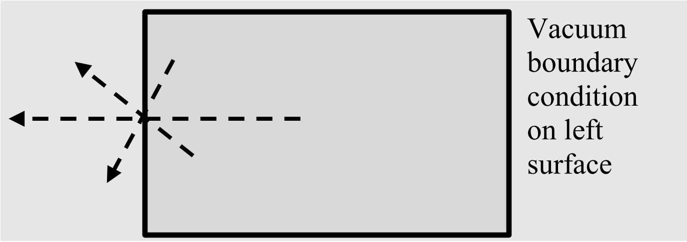


Figure 159. Vacuum boundary condition.

8.3.5.2 Reflective Boundary Condition

For the reflective boundary condition, the incoming angular flux is set equal to the outgoing angular flux in the direction corresponding to mirror or specular reflection. The reflective boundary should not be used on curved surfaces, that is, the outer boundary of a cylinder or sphere. It is the default (and required) for the left (inner) boundary of a cylindrical or spherical system. As shown in Figure 160, a given quantity of neutrons leaving a boundary (dotted line) in a particular direction will be returned (solid line) to the system with the same quantity but at a mirrored angle to the initial leakage direction.

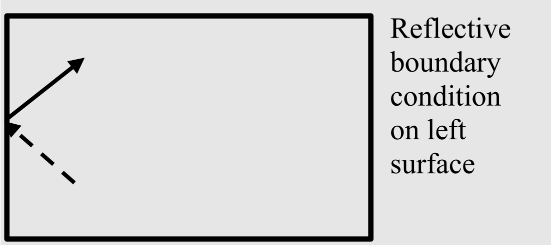


Figure 160. Reflective boundary condition.

8.3.5.3 Periodic Boundary Condition

For the periodic boundary condition, the incoming angular flux on a boundary is set equal to the outgoing angular flux on the opposite boundary. Figure 161 shows the leakage leaving each boundary (dotted lines) being returned at the same quantity and angle on the opposite boundary (solid line of same color). When the periodic boundary condition is used, it must be applied to both opposing boundaries. It should not be used on curved surfaces such as the curved boundary of a cylinder or a sphere. It should also be noted that SCALE only supports periodic boundaries on parallel planes; rotationally periodic boundary conditions are not supported.

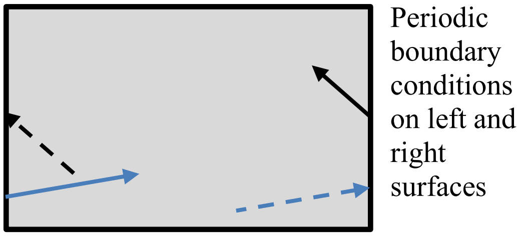


Figure 161. Periodic boundary condition.

8.3.5.4 White Boundary Condition

For the white boundary condition, the incoming angular fluxes are each set equal to a single value chosen such that the net flow across the boundary is zero. The white boundary provides isotropic return (solid lines) at a boundary (see Figure 162) and is suitable for all geometries. Although the white boundary condition is in a sense nonphysical on a curved surface, it is the best possible approximation of a reflective or repeating boundary condition, such as a lattice cell.

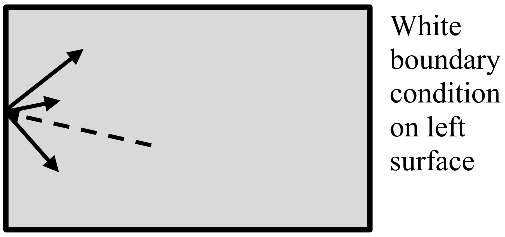


Figure 162. White boundary condition.

8.4 UNIT CELL SPECIFICATIONS

Unit cell data are not required for INFHOMMEDIUM. With the LATTICECELL, MULTIREGION, and DOUBLEHET treatments, you must specify the unit cell dimensions and materials to be used in the resonance self-shielding process. The unit cell is always 1D: infinitely long cylinders, infinite slabs, or spheres. Finite cylinders or slabs can be approximated with geometric buckling in MULTIREGION and DOUBLEHET cells. The configuration of the unit cell depends on the treatment and the type of lattice (if the LATTICELL treatment is used).

8.4.1 Infinite Homogeneous Medium

No input is required for a mixture to be treated as an infinite homogenous medium. These “cells” can be specified in the input with a single mixture number as the infinite medium to process.

8.4.2 Latticecell

For the LATTICECELL treatment, both “regular” and “annular” cells are allowed. These cells are rigorously constrained as to the placement of fuel, gap, clad, and moderators. Materials not used in the cell or other defined cells are treated as infinite homogeneous media. The “regular” cells allow spherical, cylindrical, or symmetric slab fuel regions that are constrained to a central fuel region surrounded by an optional gap, an optional clad, and an external moderator material. The “annular” cells for spherical, cylindrical, or asymmetric slab configurations are constrained to a central (second) moderator material surrounded by a fuel region having an optional gap and optional clad on both sides of the fuel with an external (first) moderator material.

The unit cell data in a LATTICECELL problem are used (1) to provide the dimensions and shape of the lump and the moderator material for resonance cross section processing, (2) to provide lattice corrections for the cross section processing, and (3) to provide information used in creating cell-weighted cross sections when XSDRNPM is executed. CELLMIX= must be specified in the unit cell data if cell-weighted cross sections are used in the problem, although the use of cell-weighted cross sections is not recommended.

8.4.3 Multiregion

The unit cell is more flexible for the MULTIREGION treatment than for the LATTICECELL treatment. In fact, the MULTIREGION treatment allows complete freedom in the placement of materials, but it is constrained by shape, as concentric regions of the same shape are required. The MULTIREGION treatment does not account for lattice effects, so it is best used for problems in which lattice effects are not important. To some degree, lattice effects can be approximated with a reflective, periodic, or white boundary condition; these boundary conditions effectively create an infinite array of the defined MULTIREGION cell.

8.4.4 Double-heterogeneous

The DOUBLEHET cell type allows for the description of a range of different fuel types that contain grains of fissile material with coatings, subsequently pressed into pebbles or compacts. The grains are assumed to be spherical, but there is no restriction on the number of coating layers surrounding the grain kernel. The input for the compact level is very similar to the MULTIREGION cell, including the spherical, cylindrical, and slab geometries.

8.4.5 Cell-Weighted Cross Sections

Cell-weighted cross sections are created by XSDRNPM when CELLMIX= is specified in the unit cell data. This option is not encouraged, but it is maintained for backwards compatibility; the user must ensure that the homogenized unit cell is identical to the real geometry. The unit cell description is then used by XSDRNPM to calculate the eigenvalue of the cell. For LATTICECELL problems, a white boundary condition is assumed. For MULTIREGION problems, the boundary conditions specified in the unit cell are used. The resultant fluxes are used to weight the cross sections of the materials in the cell and create homogenized cell-weighted cross sections that have the characteristics of the heterogeneous cell configuration. Whenever XSDRNPM is executed for cross section processing, only the cell-weighted mixture number can be used in the KENO-VI geometry data. The original mixtures used in the unit cell description cannot be used. For more information on cell-weighting refer to [Cell weighting of MG cross sections] in the Material Specification and Cross Section Processing section of the SCALE manual.

8.5 PROBLEM DESCRIPTIONS

Three example problems are used to demonstrate how to enter unit cell data for LATTICECELL, MULTIREGION, and DOUBLEHET unit cell types using Fulcrum. The first problem is a 9 × 9 fuel assembly array of UO_2 fuel rods with Zircaloy-2 cladding. The second problem involves two parallel, fully reflected SS304 slab tanks containing $U(93)O_2F_2$ solution. The third problem is based on an Organization for Economic Co-operation and Development/Nuclear Energy Agency (OECD/NEA) pebble bed reactor benchmark problem.

8.5.1 Lattice Cell Example: Fuel Assembly

This problem is a single fuel assembly at 3.5 wt.% enrichment stored in a pool of water at a temperature of 293 K. There is 15.24 cm of water above and below the assembly and 7.62 cm of water on each side of the assembly. The space between fuel rods is full-density water. The guide tubes are made of Zircaloy-2 and have an inner radius of 0.5 cm, a thickness of 0.075 cm, and a length of 387.26 cm. The pitch of the fuel rods and guide tubes is 1.64 cm. The fuel assembly radial layout is provided in Figure 163.

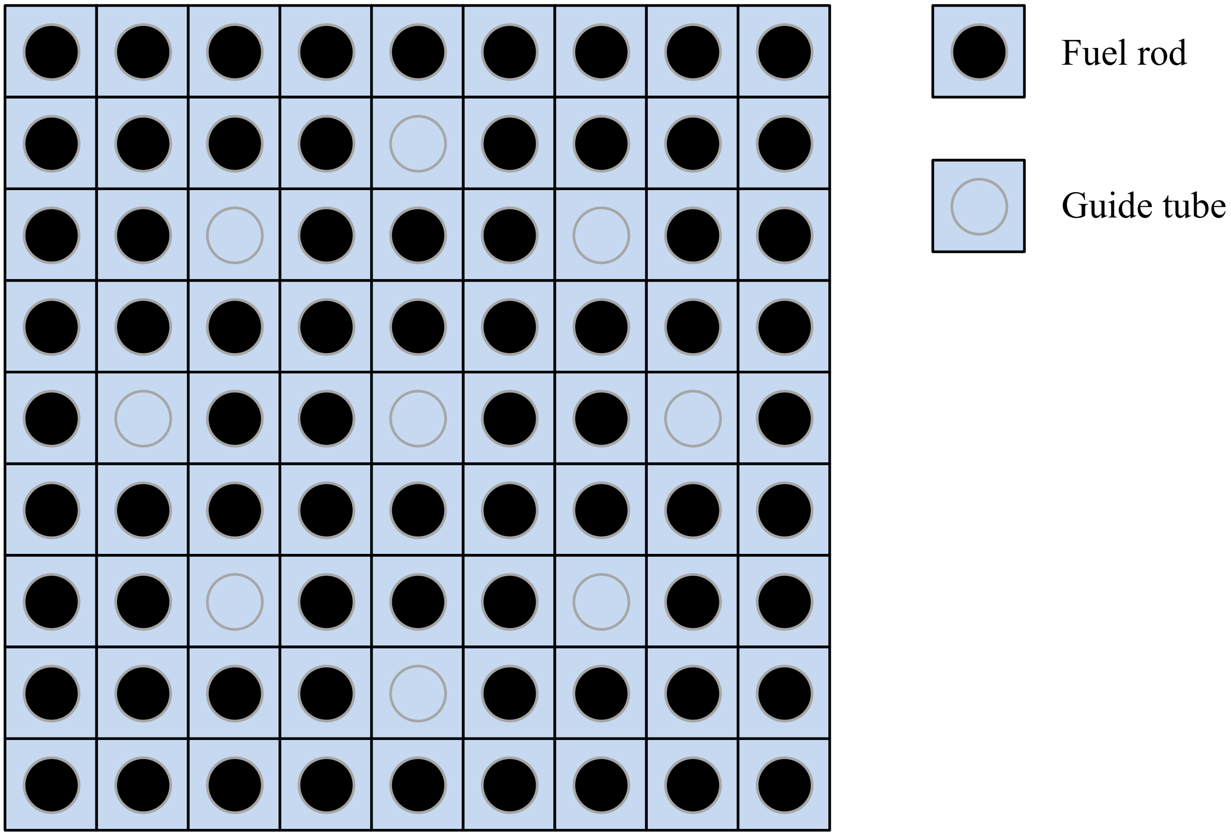


Figure 163. Fuel assembly radial layout.

The fuel pellets have a 0.47 cm radius and contain UO_2 at 95% of theoretical density. The active fuel length is 365.76 cm. Each fuel rod is made of Zircaloy-2 with an inner radius of 0.4875 cm and an outer radius of 0.545 cm. The cladding extends 7.0 cm above the top of the active fuel, and the ends are capped with solid Zircaloy-2 plugs that have the same outer diameter as the cladding. The top plug is 10.5 cm long, and the bottom plug is 4.0 cm long. The inner volume of the cladding tube is modeled containing void. The axial configuration of the fuel rods is shown in Figure 164.

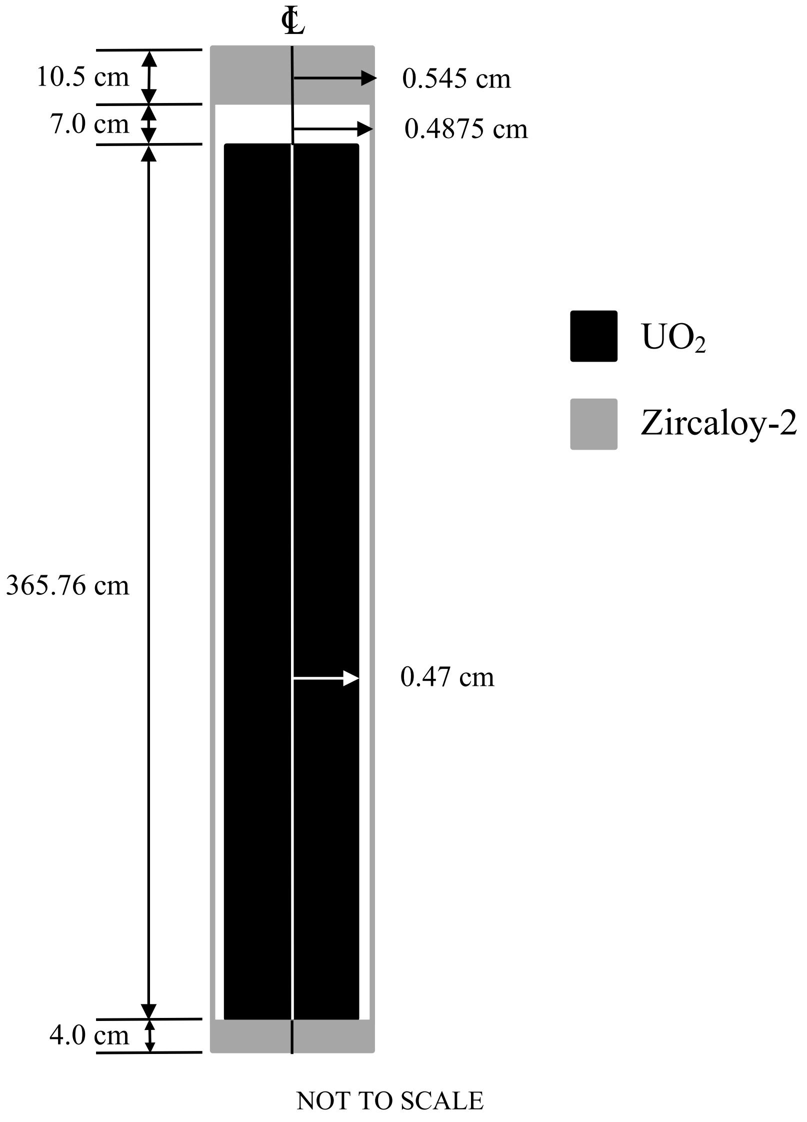
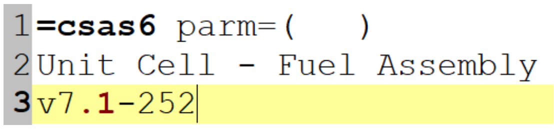


Figure 164. Axial depiction of the fuel rods.

8.5.1.1 General and Material Data

The initial data entry for this example is the same as previous problems. Start Fulcrum and then select the **New file…** option from the **File** menu. Select a directory in which to save the model, and name it `fuel_assembly.inp`. Next, use the autocomplete feature in Fulcrum to create a **csas6 – Criticality safety analysis using KENO-VI** input. Replace **title-goes-here** with `Unit Cell - Fuel Assembly`. Now change **xs-lib-goes-here** to `v7.1-252` (`v7.1-252n` if using SCALE 6.2, 6.2.1, or 6.2.2). This selects the 252-group neutron cross section library based on ENDF/B-VII.1, and it signals to CSAS6 that this will be a MG KENO calculation. The appropriate cross section processing modules will be called using the input provided in the CELLDATA block, as discussed below in Section 8.5.1.2. The first three lines of the input are shown in Figure 165.

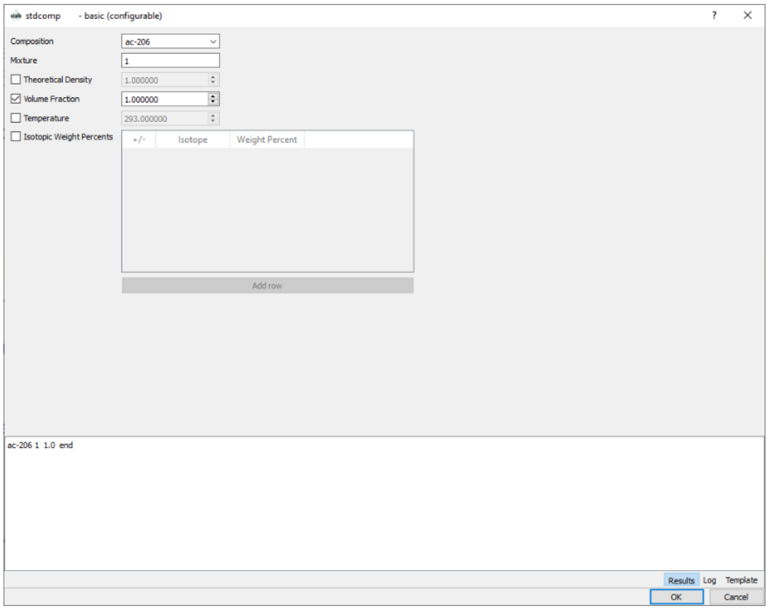


```
=csas6 parm=(   )
Unit Cell - Fuel Assembly
v7.1-252
```

Figure 165. General information for fuel assembly problem.

Next, define the different mixtures needed for the model: UO_2, Zircaloy-2, and water. All three materials are available in the SCALE standard composition library, making the definitions much simpler and faster to provide.

First, define the UO_2 composition. Move the cursor into the **read comp** block, and then autocomplete a **stdcomp – basic (configurable)**. This will generate a fillable form as shown in Figure 166. In the **composition** dropdown menu, select `uo2` by opening the dropdown menu and typing `uo2`. Change the **Volume Fraction** to `0.95` to set the density multiplier to 95% of theoretical density. Next, check the box for **Isotopic Weight Percents** and click the **Add row** button under the isotopic weight percent pane. Select ^{235}U with its nuclide ID of **92235** and set the **Weight Percent** to `3.5`. To add ^{238}U, click either the **Add row** button again, or click the + button next to **92235**. Set the **Weight Percent** to `96.5`. The final configurable window is shown in Figure 167. Confirm that the data entry is correct before clicking **OK**, as the data cannot be modified in the configurable form after it has been entered into the input.



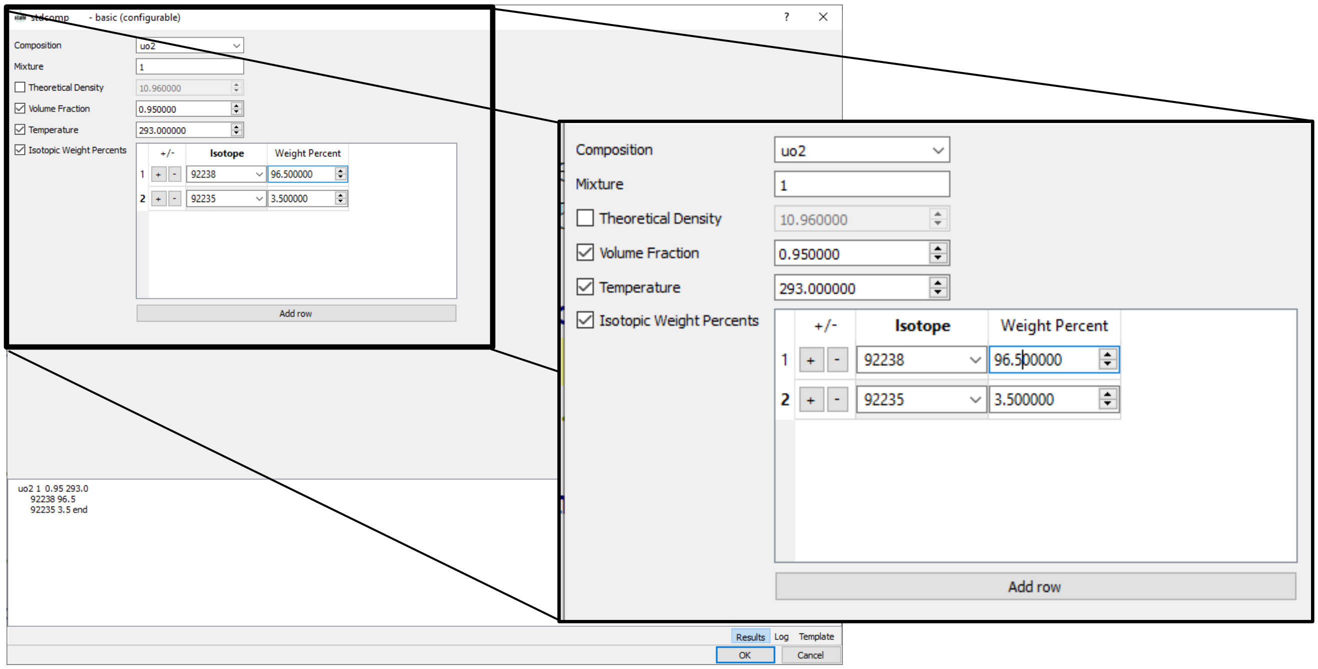
stdcomp - basic (configurable)

Composition	ac-206
Mixture	1
□ Theoretical Density	1.000000
☑ Volume Fraction	1.000000
□ Temperature	293.000000
□ Isotopic Weight Percents	

+/-	Isotope	Weight Percent

```
ac-206 1 1.0 end
```

Figure 166. Initial STDCOMP configurable form.



stdcomp - basic (configurable)

Composition: uo2

Mixture: 1

☐ Theoretical Density: 10.960000

☑ Volume Fraction: 0.950000

☑ Temperature: 293.000000

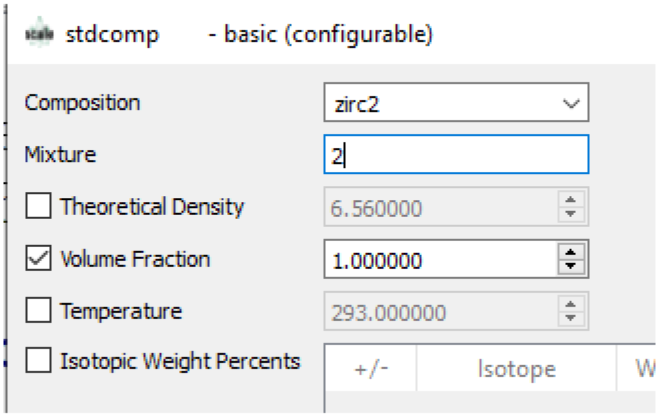
☑ Isotopic Weight Percents

	+/-	Isotope	Weight Percent
1	+ -	92238	96.500000
2	+ -	92235	3.500000

```
uo2 1 0.95 293.0
   92238 96.5
   92235 3.5 end
```

Figure 167. Complete UO₂ configurable form, showing an enlarged section of the user-specified input.

Next, repeat this process to generate the Zircaloy-2 composition. Autocomplete a **stdcomp – basic (configurable)** and select `zirc2` from the **composition** dropdown menu. Change the mixture number to 2; remember that if multiple entries are assigned to a single mixture number, then SCALE will create a mixture containing all of these materials. This can be detected by a lack of available mixtures later in the **geom** block or by checking the mixing table for total mixture density or unexpected isotopes in a specific mixture. No further modifications are needed to the default Zircaloy-2 composition, so the mixture can be added by clicking the **OK** button on the configurable form. The user-specified input section of the form is shown in Figure 168.



stdcomp - basic (configurable)

Composition: zirc2

Mixture: 2

☐ Theoretical Density: 6.560000

☑ Volume Fraction: 1.000000

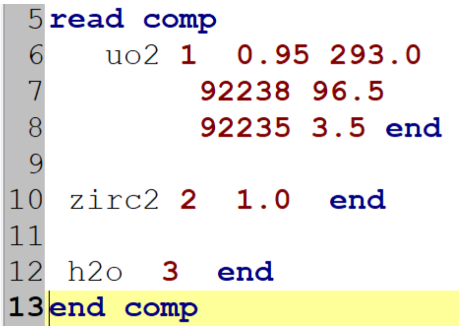
☐ Temperature: 293.000000

☐ Isotopic Weight Percents

Figure 168. User-specified input portion of the complete zirc2 configurable form.

The water composition is the final mixture to create, and it is also not modified from the defaults provided in the SCALE standard composition library. In many cases such as this, specifying the composition by entering the text directly into Fulcrum is likely faster than using a configurable. In this case, simply type `h2o 3 end` inside the composition block. This specifies mixture 3 as containing water at the default density (0.9982 g/cm^3) and the default temperature (293 K).

The comment **TODO: define comp** can be removed, as the compositions have been defined. The comment can also be left in the input, as it has no impact on the execution of SCALE. The completed composition block is shown in Figure 169. Note that additional copies of some mixtures should be created in the composition block if needed for multiple unit cell definitions in the CELLDATA block. As discussed in Section 8.5.1.2 below, only a single unit cell is needed for this problem, and correspondingly, no copies of mixtures are needed.



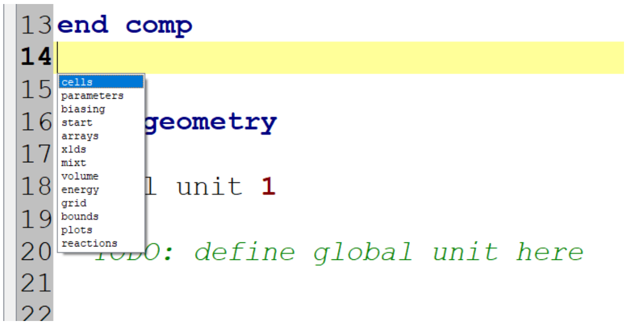
```
read comp
   uo2 1  0.95 293.0
        92238 96.5
        92235 3.5 end

 zirc2 2  1.0  end

 h2o  3  end
end comp
```

Figure 169. Complete COMPOSITION block for the fuel assembly problem.

8.5.1.2 Unit Cell Data

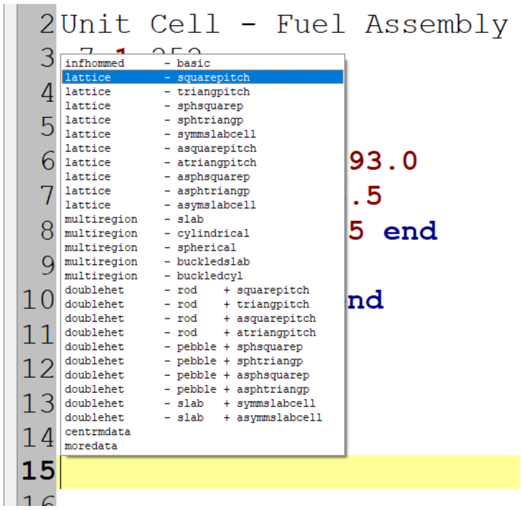
Now that the material information has been entered, the unit cell information can be specified for cross section treatment. One of the four unit cell types (INFHOMMEDIUM, LATTICECELL, MULTIREGION, DOUBLEHET) will be specified to process the MG cross sections for this problem. An ordered array of moderated fuel pins is well described by a LATTICECELL unit cell, and this will be the option used in this model. The specific array option that best describes the problem is SQUAREPITCH based on the cylindrical rods in a square pitched array. The 3D effects of the plenum space above the fissile material and of the end plugs will be neglected because the unit cell specifications are all 1D cells.

Move the cursor to a line between the composition and geometry blocks and autocomplete to generate a list of other blocks to be entered, as shown in Figure 170. Select **cells** to instruct Fulcrum to enter a CELLDATA block. There are no configurable forms currently available in the CELLDATA block. Autocompleting in the CELLDATA block provides a list of available unit cell specifications; select the **lattice - squarepitch** option, as shown in Figure 171. Fulcrum inserts a single line of CELLDATA input, as shown in Figure 172, which must be modified for use in this model.



```
end comp


    geometry

   l unit 1

    O: define global unit here
```

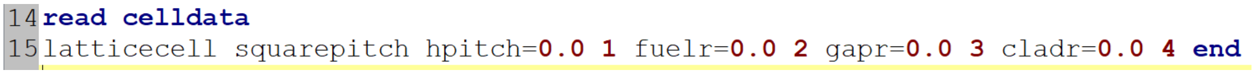
Figure 170. Autocomplete list of additional blocks available for inclusion in the model.



```
Unit Cell - Fuel Assembly
```

93.0
.5
5 end
nd
infhommed - basic
lattice - squarepitch
lattice - triangpitch
lattice - sphsquarep
lattice - sphtriangp
lattice - symmslabcell
lattice - asquarepitch
lattice - atriangpitch
lattice - asphsquarep
lattice - asphtriangp
lattice - asymslabcell
multiregion - slab
multiregion - cylindrical
multiregion - spherical
multiregion - buckledslab
multiregion - buckledcyl
doublehet - rod + squarepitch
doublehet - rod + triangpitch
doublehet - rod + asquarepitch
doublehet - rod + atriangpitch
doublehet - pebble + sphsquarep
doublehet - pebble + sphtriangp
doublehet - pebble + asphsquarep
doublehet - pebble + asphtriangp
doublehet - slab + symmslabcell
doublehet - slab + asymmslabcell
centrmdata
moredata

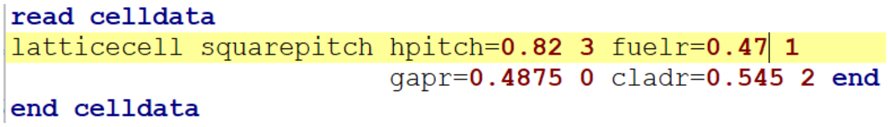
Figure 171. Autocomplete list of cells available within the CELLDATA block.



```
read celldata
latticecell squarepitch hpitch=0.0 1 fuelr=0.0 2 gapr=0.0 3 cladr=0.0 4 end
```

Figure 172. Templated input for LATTICECELL SQUAREPITCH provided by Fulcrum.

The first entry in the LATTICECELL SQUAREPITCH input is **hpitch=0.0**. This input specifies the half pitch for the unit cell; the full pitch is the spacing between fuel rods, and it can be provided in SCALE by changing the keyword from **hpitch** to **pitch**. In this case, the pitch is 1.64 cm, so the half pitch is 0.82. The single integer provided after the dimension of the half pitch, 1 in the template input, is the mixture number for the material filling the space between the fuel rods. In this model, that is water as specified in the COMPOSITION block as mixture 3. Similarly, **fuelr** is the radius of the fuel material (**fueld** is the keyword for the diameter), and the mixture number for the fuel is provided. Replace the fuel radius input with `0.47` cm and the fuel mixture with `1`. The gap between the fuel and the cladding is modeled as a void, which is always mixture `0` in SCALE. The **gapr** is equal to the inner radius of the cladding (**gapd** for the inner diameter), `0.4875` cm. Finally, the **cladr** is the outer radius of the cladding (**cladd** for the outer diameter). The relevant dimension is `0.545` cm, and the mixture number for Zircaloy-2 in this model is `2`. This completes the unit cell entry, and since it is the only unit cell specified in this model, it completes the CELLDATA block. The complete entry/block is provided in Figure 173; note that the entry is allowed to span multiple lines and is not complete until **end** is specified for the cell specification. White space also has no impact in the input, so the input shown is split on to two lines to narrow the figure, and white space is added to align the entries for user convenience.



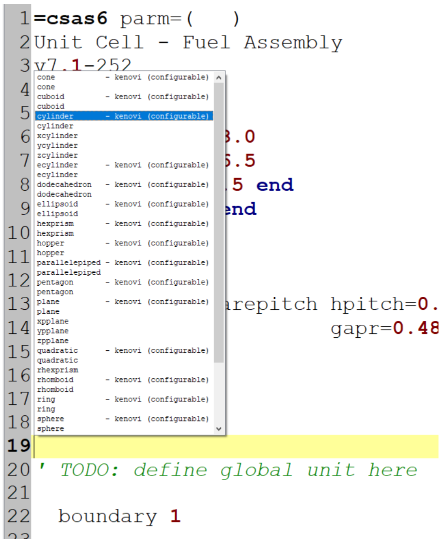
```
read celldata
latticecell squarepitch hpitch=0.82 3 fuelr=0.47 1
                        gapr=0.4875 0 cladr=0.545 2 end
end celldata
```

Figure 173. Complete LATTICECELL input.

8.5.1.3 KENO-VI Geometry Data

In this problem, the following three units are needed: Unit 1 will be a fuel rod cell; Unit 2 will be a guide tube cell, and Unit 3 will be the global unit containing the array of rods and the surrounding water.

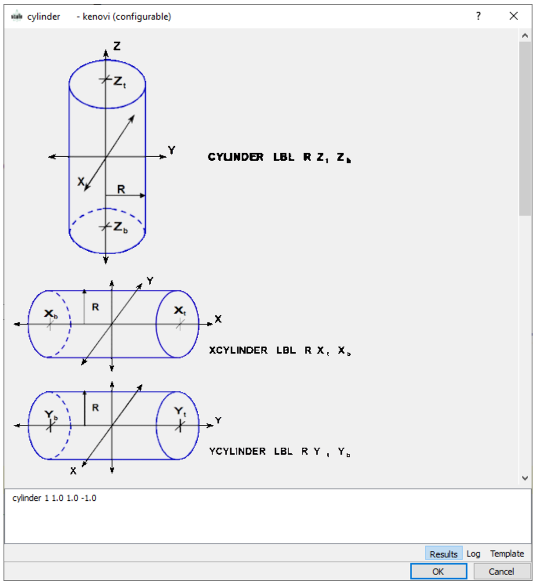
The fuel rod unit cell, Unit 1, is the first unit to be defined. The outer boundary of the unit must be a cuboid so that Unit 1 can be included in the array. The regions to be entered are the fuel cylinder, the gap cylinder, the cladding cylinder, and finally the water cuboid. In KENO-VI, there is no requirement for a particular order for the regions in a unit; this is merely a logical progression. This is a significant difference from KENO V.a. The first step in defining Unit 1 is to delete **global** from the unit declaration. Configurable forms are included for all the shapes available in KENO-VI. To add the cylinder, position the cursor between the unit declaration and the **boundary** record. Autocomplete in this context provides a list of shapes and their configurable forms and select **cylinder – kenovi (configurable)**, as shown in Figure 174. The default configurable form is shown in Figure 175.



```
=csas6 parm=(   )
Unit Cell - Fuel Assembly
v7.1-252

                                        3.0
                                        5.5
                                        .5 end
                                        end




                                        arepitch hpitch=0.
                                                  gapr=0.48





' TODO: define global unit here

  boundary 1
```

cone - kenovi (configurable)
cone
cuboid - kenovi (configurable)
cuboid
cylinder - kenovi (configurable)
cylinder
xcylinder
ycylinder
zcylinder
ecylinder - kenovi (configurable)
ecylinder
dodecahedron - kenovi (configurable)
dodecahedron
ellipsoid - kenovi (configurable)
ellipsoid
hexprism - kenovi (configurable)
hexprism
hopper - kenovi (configurable)
hopper
parallelepiped - kenovi (configurable)
parallelepiped
pentagon - kenovi (configurable)
pentagon
plane - kenovi (configurable)
plane
xpplane
ypplane
zpplane
quadratic - kenovi (configurable)
quadratic
rhexprism
rhomboid - kenovi (configurable)
rhomboid
ring - kenovi (configurable)
ring
sphere - kenovi (configurable)
sphere

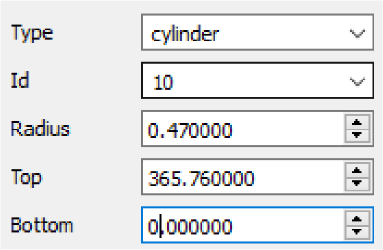
Figure 174. Selecting a cylinder configurable form.



```
cylinder 1 1.0 1.0 -1.0
```

Figure 175. Default cylinder configurable form.

Completing the configurable form will provide the input for a single cylindrical region in Unit 1. The first dropdown menu (**Type**) is available because by default, a **cylinder** in SCALE is aligned parallel to the Z-axis; cylinders parallel to the X- or Y-axis can also be generated using this form. The input for the second dropdown menu is the ID for this region; set this to `10`. The next three fields specify the dimensions of the cylinder; the radius is `0.47` cm. The height of the fissile column is 365.76 cm, but the actual +Z and -Z entries depend on the selection of the origin in the unit. For this model, the origin will be placed at the bottom of the fissile column, so the +Z entry is `365.76` cm and the -Z entry is `0` cm. No origin input is needed, as the unit will be built with the centerline of the fuel rod passing through the origin. The complete user-specified input for the cylinder is provided in Figure 176. Press **OK** to add the cylinder region to the input after checking that the inputs are correct.

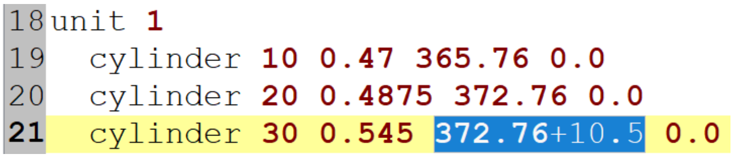


Type	cylinder
Id	10
Radius	0.470000
Top	365.760000
Bottom	0.000000

Figure 176. User-specified portion of the completed cylinder configurable form.

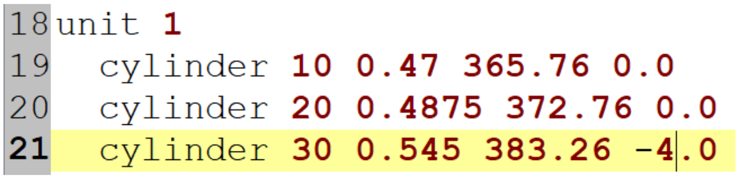
The next region in unit 1 is the void inside the cladding and outside the fissile column. The easiest way to complete this portion is to copy and paste the existing entry and change the dimensions. Assign this second cylinder an ID of `20`; it has a radius of `0.4875` cm, a +Z surface of `372.76` cm, and it has a -Z surface of `0` cm.

The cladding cylinder can also be added by copying the previous cylinder and pasting it into the unit. The ID number for the Zircaloy-2 cladding should be `30`, and the radius is `0.545` cm. The +Z surface for the void cylinder is 372.76, and the end plug length is 10.5 cm. Enter `372.76+10.5`, highlight the expression as shown in Figure 177, and evaluate the expression by selecting the **Edit** dropdown menu and clicking **Evaluate.** Enter the **-**Z surface as `-4.0` cm to represent the lower end plug. The completed input for the three cylinders is shown in Figure 178.



```
unit 1
  cylinder 10 0.47 365.76 0.0
  cylinder 20 0.4875 372.76 0.0
  cylinder 30 0.545 372.76+10.5 0.0
```

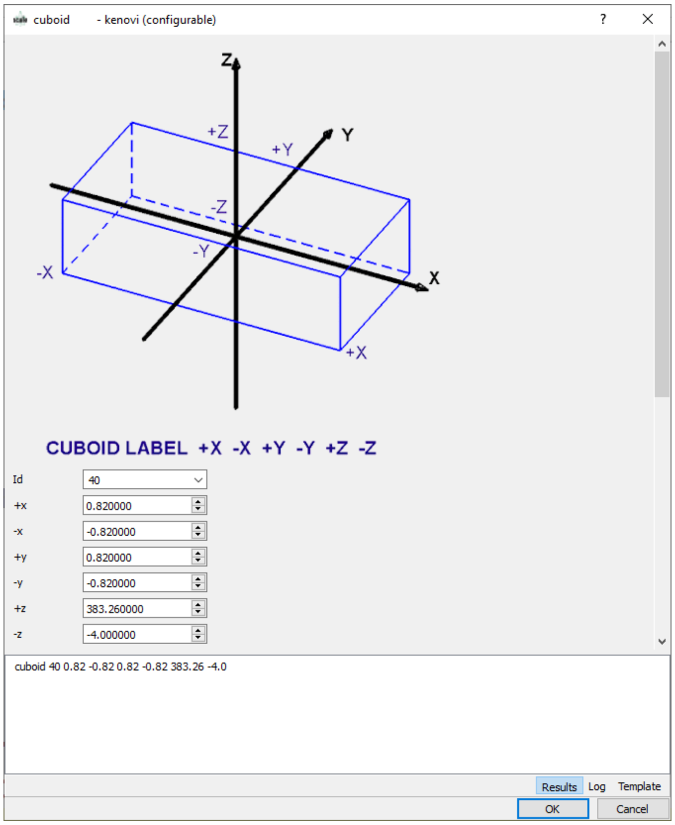
Figure 177. Highlighting expression for evaluation in Fulcrum.



```
unit 1
  cylinder 10 0.47 365.76 0.0
  cylinder 20 0.4875 372.76 0.0
  cylinder 30 0.545 383.26 -4.0
```

Figure 178. Final input for the three cylinders in Unit 1.

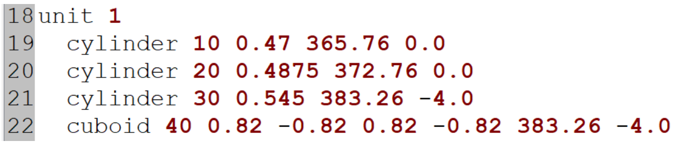
The last region required in Unit 1 is the cuboid to define the unit cell for use in the array. The X and Y dimensions will be equal to plus and minus the half pitch of the array. This will ensure that the appropriate spacing is generated when two unit cells are arrayed together. The Z extent must be at least that of the fuel rod, but it could include the water above and below the assembly. In this case, the water will be added in the global unit. The cuboid can be added with a configurable form; place the cursor anywhere between the **unit 1** declaration and the **boundary** record for the unit, and then autocomplete and select **cuboid - kenovi (configurable)**. Change the **ID** to `40`, and then set the **+x** and **+y** values to `0.82` and the **-x** and **-y** values to `-0.82`. The **+z** face is `383.26` and the **-z** face is `-4.0`. The completed configurable form is shown in Figure 179; after confirming that the input is correct, press **OK** to add the cuboid definition to the input. The complete set of 4 region inputs is shown in Figure 180.


cuboid - kenovi (configurable)
?
×
Z
Y
X
+Z
+Y
-Z
-Y
-X
+X
CUBOID LABEL +X -X +Y -Y +Z -Z
Results
Log
Template
OK
Cancel

Id	40
+x	0.820000
-x	-0.820000
+y	0.820000
-y	-0.820000
+z	383.260000
-z	-4.000000

```
cuboid 40 0.82 -0.82 0.82 -0.82 383.26 -4.0
```

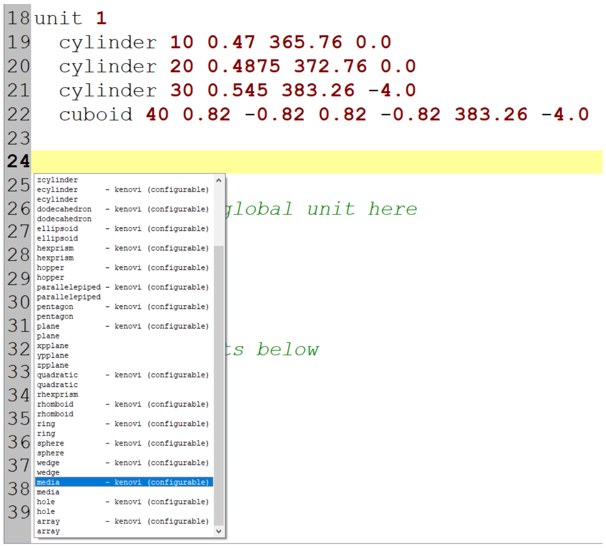
Figure 179. Completed configurable form for the cuboid in Unit 1.



```
unit 1
  cylinder 10 0.47 365.76 0.0
  cylinder 20 0.4875 372.76 0.0
  cylinder 30 0.545 383.26 -4.0
  cuboid 40 0.82 -0.82 0.82 -0.82 383.26 -4.0
```

Figure 180. Input for all four regions in Unit 1.

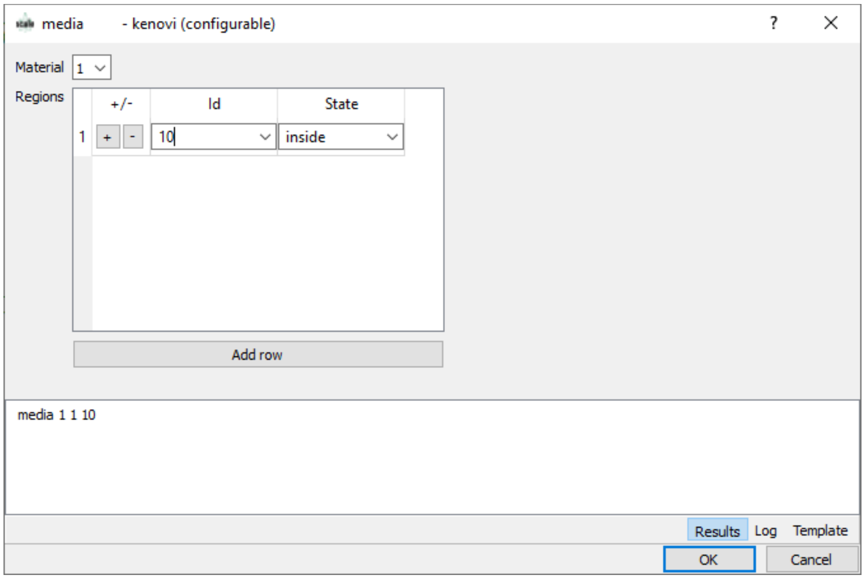
All of the regions needed in Unit 1 have been entered; at this point, media records can be specified to fill the regions with the appropriate mixtures. Autocomplete can be used to select **media - kenovi (configurable)**, as shown in Figure 181, to generate a configurable form for media entry. The first entry will be the fissile material, which is Mixture 1 and needs to be placed inside cylinder 10. By default, mixture 1 is filled in the **Material** field; click the **Add row** button at the bottom of the **Regions** pane to add a line to the pane for use. Change the **Id** to `10`, and leave the State dropdown menu set to **inside**. As shown in the **Results** pane, this will generate a MEDIA record placing Mixture 1 inside region 10. After confirming that the configurable form is correct, as shown in Figure 182, press the **OK** button to add the MEDIA record to the input. The next MEDIA record will be used to fill the gap space inside cylinder 20 but outside cylinder 10 with void. Generate another MEDIA configurable form and change the **Material** to `0`. Click the **Add row** button and set the **Id** to `20` with a state of **inside**; add another row and set the **Id** to `10` and the state to **outside**. The completed MEDIA configurable form is shown in Figure 183; after confirming that the input is correct, press the **OK** button to add the MEDIA record to the input. The third MEDIA record will place Zircaloy-2 in the cladding region. The fastest way to add this record is to copy the previous MEDIA record and paste it into Unit 1. Change the mixture number from `0` to `2`, and then set the region definition vector to `30 -20` (inside region 30 and outside region 20). The last MEDIA record fills cuboid 40 with water. Again, copy and paste a previous MEDIA record, and then change the mixture number to `3` and the region definition vector to `40 -30` (inside region 40 and outside region 30). Recall that there is no requirement to enter the MEDIA records in any order relative to each other or relative to the region definitions. All four MEDIA records are shown in Figure 184.



```
unit 1
  cylinder 10 0.47 365.76 0.0
  cylinder 20 0.4875 372.76 0.0
  cylinder 30 0.545 383.26 -4.0
  cuboid 40 0.82 -0.82 0.82 -0.82 383.26 -4.0


                  global unit here





                  ts below
```

zcylinder
ecylinder - kenovi (configurable)
ecylinder
dodecahedron - kenovi (configurable)
dodecahedron
ellipsoid - kenovi (configurable)
ellipsoid
hexprism - kenovi (configurable)
hexprism
hopper - kenovi (configurable)
hopper
parallelepiped - kenovi (configurable)
parallelepiped
pentagon - kenovi (configurable)
pentagon
plane - kenovi (configurable)
plane
xpplane
ypplane
zpplane
quadratic - kenovi (configurable)
quadratic
rhexprism
rhomboid - kenovi (configurable)
rhomboid
ring - kenovi (configurable)
ring
sphere - kenovi (configurable)
sphere
wedge - kenovi (configurable)
wedge
media - kenovi (configurable)
media
hole - kenovi (configurable)
hole
array - kenovi (configurable)
array

Figure 181. Selecting the MEDIA configurable form in Fulcrum.



media - kenovi (configurable)

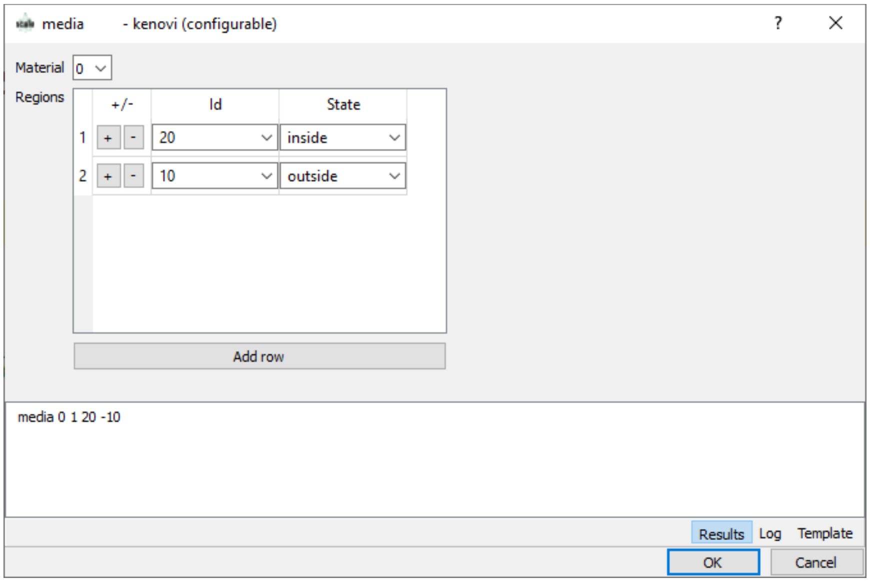
Material 1

Regions

	+/-	Id	State
1	+ -	10	inside

media 1 1 10

Figure 182. Completed MEDIA configurable form placing UO_2 in cylinder 10 in Unit 1.



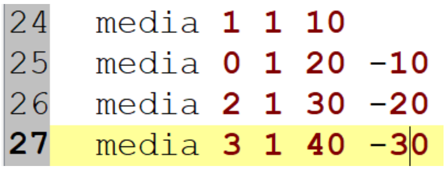
Material 0

Regions

	+/-	Id	State
1	+ -	20	inside
2	+ -	10	outside

```
media 0 1 20 -10
```

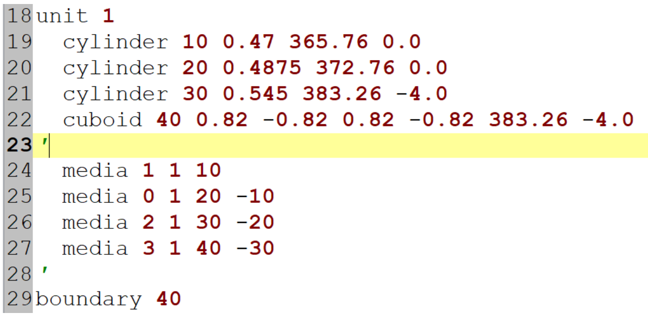
Figure 183. Completed MEDIA configurable form placing void in cylinder 20 and outside cylinder 10 in Unit 1.



```
  media 1 1 10
  media 0 1 20 -10
  media 2 1 30 -20
  media 3 1 40 -30
```

Figure 184. Input for all four media records un Unit 1.

The last step to complete the definition of Unit 1 is to update the **boundary** record. The default record generated by Fulcrum is for the boundary to contain all of space inside surface 1, but there is no surface 1 in this unit. The outermost region in Unit 1 is **cuboid 40**, which contains the water outside the fuel rod. Change the region definition vector for the boundary record to `40` (all of space inside region 40 is part of Unit 1). The complete input for Unit 1 is shown in Figure 185; most comments and blank lines have been removed to clarify the figure. Neither comments nor blank lines have any effect on SCALE input processing.

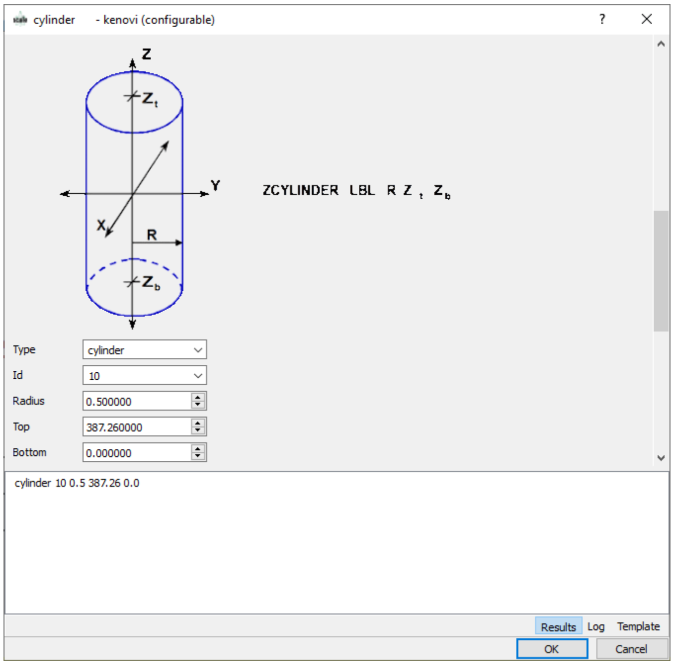


```
unit 1
  cylinder 10 0.47 365.76 0.0
  cylinder 20 0.4875 372.76 0.0
  cylinder 30 0.545 383.26 -4.0
  cuboid 40 0.82 -0.82 0.82 -0.82 383.26 -4.0
'
  media 1 1 10
  media 0 1 20 -10
  media 2 1 30 -20
  media 3 1 40 -30
'
boundary 40
```

Figure 185. Complete input for Unit 1.

The second unit needed for this problem is the guide tube unit. The outer boundary again must be a cuboid so that this unit can be used in the array, and the cuboid must be the same size as the Unit 1 cuboidal outer surface so that the matching faces are the same size in all cases. This unit only requires two cylinders for the water inside the guide tube and for the guide tube itself. The outer cuboid will be the third and final region in Unit 2.

Place the cursor after the boundary record for Unit 1 and enter `unit 2`. Use autocomplete to generate a configurable form for the first cylinder; set the **Id** to `10` and the **Radius** to `0.5`, the **top** to `387.26`, and the **bottom** to `0`. The completed input form is shown in Figure 186; after confirming that the input is correct, press **OK** to add the cylinder specification to the input. The second cylinder can also be added with a configurable form; set the **Id** to `20`, the **Radius** to `0.575`, and the **top** and **bottom** are again `387.26` and `0`, respectively. The user-specified input for this region is shown in Figure 187; after confirming that the input is correct, press **OK** to add the cylinder specification to the input. The final surface needed in Unit 2 is the cuboid for water outside of the guide tube. This surface (cuboid 40) can be copied from Unit 1 and pasted in Unit 2; the ±z dimensions must be changed to `387.26` and `0`, respectively, after being pasted into Unit 2. The complete set of three region inputs is shown in Figure 188.

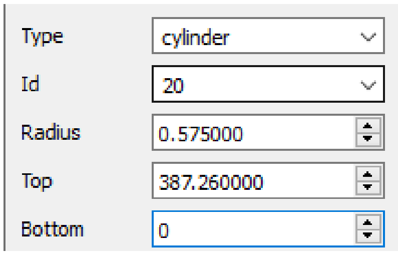


ZCYLINDER LBL R Z $_t$ Z$_b$

Type	cylinder
Id	10
Radius	0.500000
Top	387.260000
Bottom	0.000000

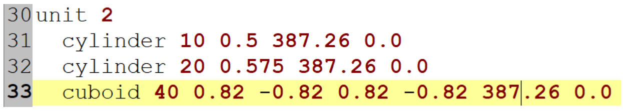
```
cylinder 10 0.5 387.26 0.0
```

Figure 186. Completed configurable form in Fulcrum for first cylinder in Unit 2.



Type	cylinder
Id	20
Radius	0.575000
Top	387.260000
Bottom	0

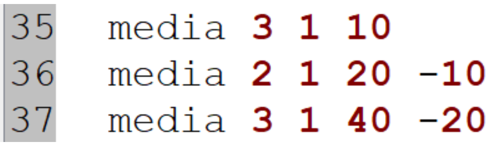
Figure 187. User-specified input in configurable form for second cylinder un Unit 2.



```
unit 2
  cylinder 10 0.5 387.26 0.0
  cylinder 20 0.575 387.26 0.0
  cuboid 40 0.82 -0.82 0.82 -0.82 387.26 0.0
```

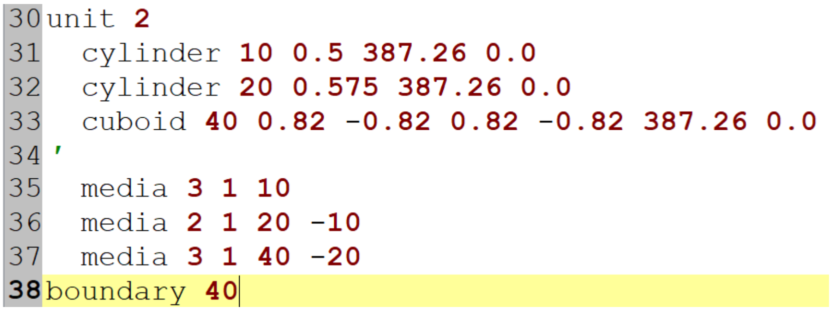
Figure 188. Input for all three regions in Unit 2.

All of the regions needed in Unit 2 have been entered; at this point, media records can be specified to fill the regions with the appropriate mixtures. The MEDIA entries for this unit are straightforward: the first is the water inside the guide tube, the second is the Zircaloy-2 forming the guide tube, and the third is the water outside the guide tube. The first MEDIA record can be entered directly in Unit 2 as `media 3 1 10`, which fills surface 10 (the first cylinder in the unit) with water. Similarly, the second MEDIA record, placing Zircaloy-2 in the guide tube wall, can be entered as `media 2 1 20 -10`. Finally, the water outside the guide tube can be entered as `media 3 1 40 -20`. All three MEDIA records are shown in Figure 189. Finally, the boundary record can be added; as with Unit 1, the outer surface is **cuboid 40**. Enter `boundary 40` after the MEDIA records to complete Unit 2. The complete Unit 2 input is shown in Figure 190.



```
media 3 1 10
media 2 1 20 -10
media 3 1 40 -20
```

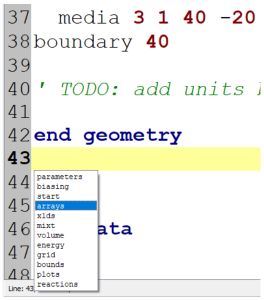
Figure 189. Input for all three MEDIA records in Unit 2.



```
unit 2
  cylinder 10 0.5 387.26 0.0
  cylinder 20 0.575 387.26 0.0
  cuboid 40 0.82 -0.82 0.82 -0.82 387.26 0.0
'
  media 3 1 10
  media 2 1 20 -10
  media 3 1 40 -20
boundary 40
```

Figure 190. Complete input for Unit 2.

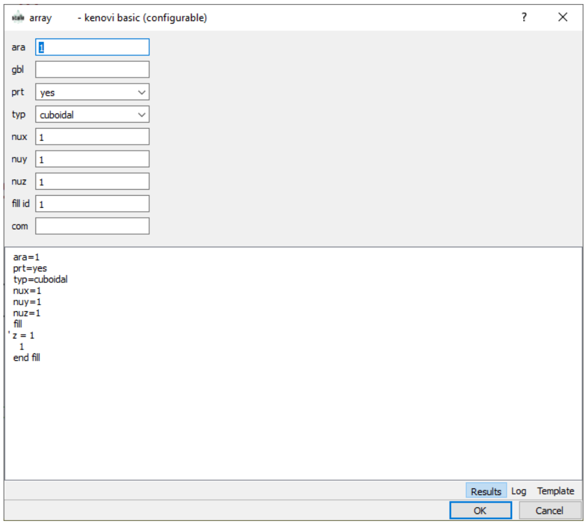
The next step is to create an array of units 1 and 2 for the fuel assembly. As discussed in Section 4.6, the faces that touch between units placed in an array in KENO-VI must be the same size. The X and Y faces of both units 1 and 2 extend from -0.82 cm to 0.82 cm, for a total length of 1.64 cm. In the axial dimension, Unit 1 extends from -4 cm to 383.26 cm, a total length of 387.26 cm. Unit 2 extends from 0 cm to 387.26 cm, which matches the Unit 1 axial extent exactly. The dimensions are the same, so the units can be used together in an array. Note that the X and Y faces must match because both X and Y faces touch in various places within the array. Also note that the Z dimensions are the same length despite having different coordinates. The array need not be specified in the ARRAY block before it is included in the geometry block, but it is often simpler to do this. The ARRAY block can be added with autocomplete once the cursor is positioned outside the geometry and all other blocks. An example is shown in Figure 191. Select **arrays** from the list of available blocks to add.



```
  media 3 1 40 -20
boundary 40

' TODO: add units i

end geometry

```

Figure 191. Autocomplete list of available blocks to add to the input file.

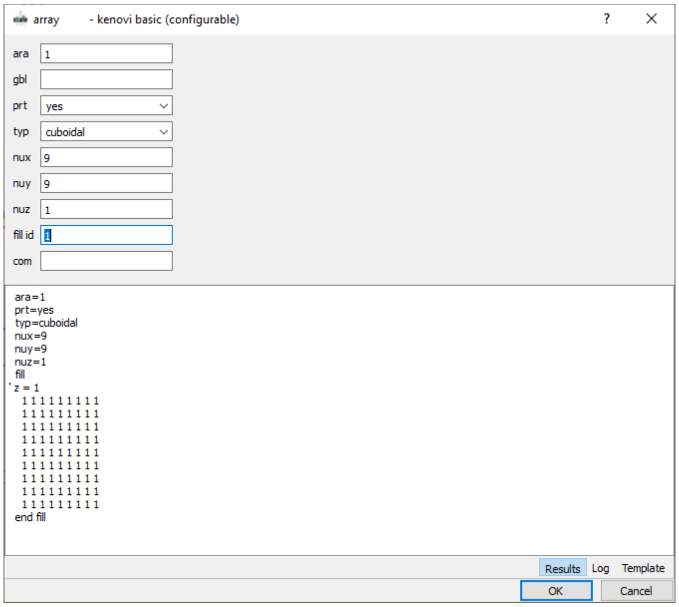
The only option for autocomplete within the array block is for an **array - kenovi basic (configurable)**; select this option to launch an array configurable form like that shown in Figure 192. The array number (**ara**) can remain as 1, and the **gbl** field can remain blank, as this will not be a global array. The **prt** option controls printing the contents of the array in the output file; **yes** is the default option and is generally retained. **PRT** is only set to **no** for particularly large, uniform arrays which generate large edits that are not helpful in understanding the model. The **Typ** remains cuboidal, as this is an array of units with cuboids as outer boundaries. The **nux**, **nuy**, and **nuz** entries specify the number of units in the array in the X, Y, and Z directions, respectively. The array for the fuel assembly being modeled here is a 9 × 9 element array with only one axial level. **nux** and **nuy** are therefore `9`, and **nuz** is `1`. The **fill id** option allows the user to select a unit with which to populate the ARRAY FILL; in this case, 1 is the better choice, as 72 of the elements in the array are Unit 1, while 9 are Unit 2. This completes the user-specified input for this array, as shown in Figure 193. After reviewing the input to confirm that it is correct, press **OK** to add the array to the input. The final step is to add Unit 2 in the appropriate places within the ARRAY FILL to place guide tubes within the fuel assembly. Recall that the processing of entries is first in X, then in Y, and finally in Z. The Fulcrum configurable form inserts the text in NUY rows that have NUX columns as a convenience for the user in creating the correct FILL entries, but this is not required in KENO input. The first row to be read will be the Y=1 row, and the last row will be the Y=9 row, so the input is essentially upside down relative to the model. The fuel assembly layout is symmetric, so this is not a concern for this model but is worth noting. The final ARRAY data for this problem, including the FILL entries with the guide tubes (Unit 2), are shown in Figure 194. Note that the **TODO** comment and extra blank lines have been removed to save space in this document, but they have no impact on KENO if they remain in the input.



ara	1
gbl	
prt	yes
typ	cuboidal
nux	1
nuy	1
nuz	1
fill id	1
com	

```
ara=1
prt=yes
typ=cuboidal
nux=1
nuy=1
nuz=1
fill
'z = 1
  1
end fill
```

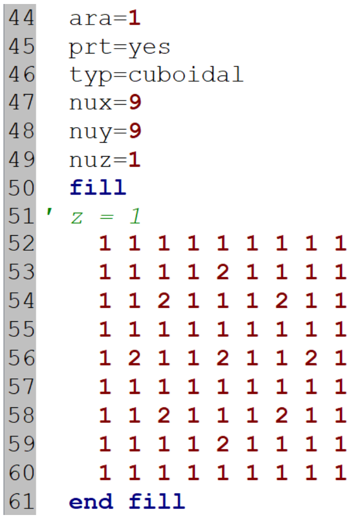
Figure 192. Default configurable form for a KENO-VI array.



ara	1
gbl	
prt	yes
typ	cuboidal
nux	9
nuy	9
nuz	1
fill id	1
com	

```
ara=1
prt=yes
typ=cuboidal
nux=9
nuy=9
nuz=1
fill
'z = 1
  1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1
end fill
```

Figure 193. Complete configurable form for the array in the model.



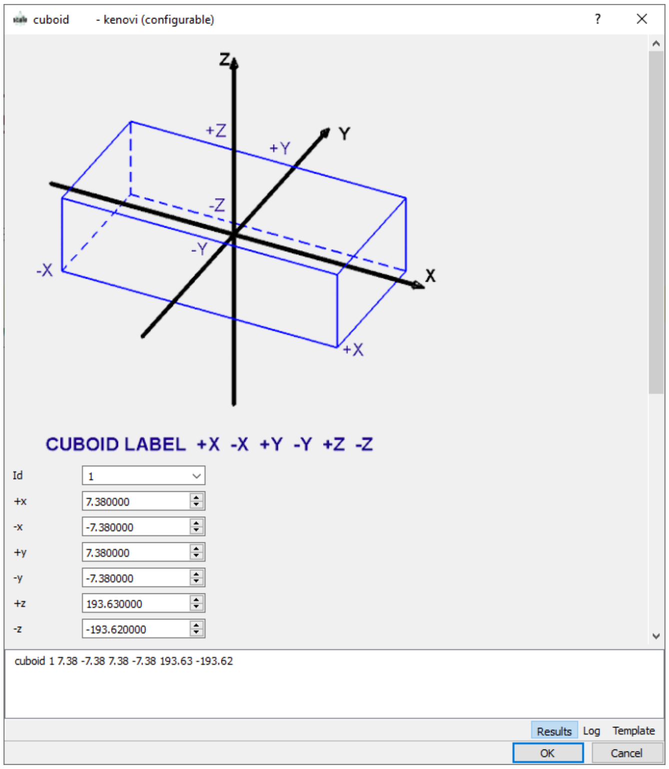
```
ara=1
prt=yes
typ=cuboidal
nux=9
nuy=9
nuz=1
fill
' z = 1
   1 1 1 1 1 1 1 1 1
   1 1 1 1 2 1 1 1 1
   1 1 2 1 1 1 2 1 1
   1 1 1 1 1 1 1 1 1
   1 2 1 1 2 1 1 2 1
   1 1 1 1 1 1 1 1 1
   1 1 2 1 1 1 2 1 1
   1 1 1 1 2 1 1 1 1
   1 1 1 1 1 1 1 1 1
end fill
```

Figure 194. Complete array specification with guide tube units in FILL block.

The array for the fuel assembly can be inserted into a unit in the GEOMETRY block now that it has been created. An array cannot be placed into one of the units that has been used in it, so a new unit is needed. This will be Unit 3 and will be the global unit. The array must be placed inside a region that it fills completely, as discussed in Section 4.6. Then a subsequent region in this unit will be added for the water above, below, and around the fuel assembly.

The first step is to create global unit 3. This can be done by placing the cursor directly above **end geometry** and using autocomplete to add **global_unit**. Autocomplete provides global unit 1, so the unit number must be changed manually to 3. The array is 9 units wide in X and Y, with each unit having a width of 1.64 cm, so the total width of the array is 14.76 cm. The height of the array is 387.26 cm. Placing the origin of the unit at the middle of the array will make adding the water region simpler, so a cuboid with X and Y dimensions of ±7.38 cm and Z dimensions of ±193.63 can be created to contain the array. Autocomplete can be used to generate a configurable form of a cuboid by selecting **cuboid -**

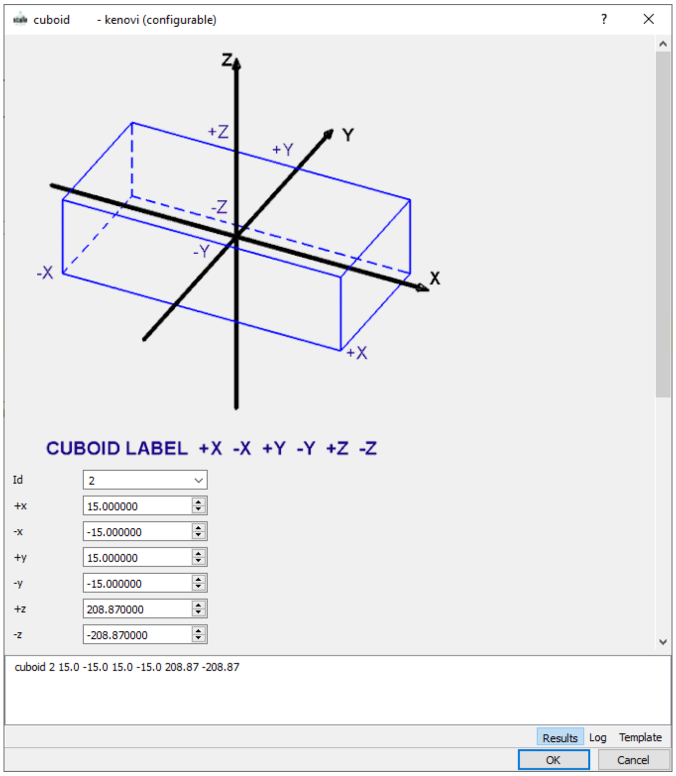
kenovi (configurable) from the list. Leave the **Id** as `1` and enter the dimensions provided for the dimensions of the cuboid. The completed configurable form is shown in Figure 195. After confirming that the input is correct, press the **OK** button to add the cuboid specification to the input file. A second cuboid is needed to contain the water around the assembly in the problem specification. This region was stated to be 7.62 cm in X and Y and 15.24 cm in Z, adding reflection on all 6 faces of the array. The dimensions for this new cuboid are therefore ±15 cm in both X and Y and ±208.87 in Z. This cuboid can also be added using a configurable form, this time with Id `2`. The completed configurable form is shown in Figure 196. After confirming that the input is correct, press the **OK** button to add the cuboid specification to the input file. The complete input for these two regions is shown in Figure 197.


cuboid - kenovi (configurable)
?
Z
Y
X
+Z
-Z
+Y
-Y
+X
-X
CUBOID LABEL +X -X +Y -Y +Z -Z
Results
Log
Template
OK
Cancel

Id	1
+x	7.380000
-x	-7.380000
+y	7.380000
-y	-7.380000
+z	193.630000
-z	-193.620000

```
cuboid 1 7.38 -7.38 7.38 -7.38 193.63 -193.62
```

Figure 195. Completed configurable form for first cuboid in Unit 3.

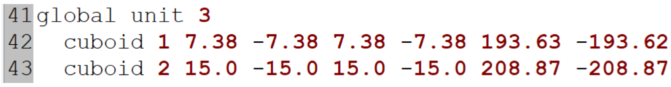


CUBOID LABEL +X -X +Y -Y +Z -Z

Id	2
+x	15.000000
-x	-15.000000
+y	15.000000
-y	-15.000000
+z	208.870000
-z	-208.870000

```
cuboid 2 15.0 -15.0 15.0 -15.0 208.87 -208.87
```

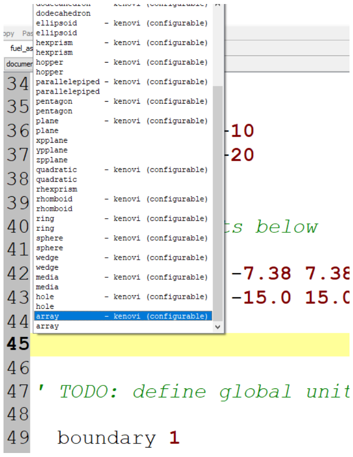
Figure 196. Completed configurable form for second cuboid in Unit 3.



```
global unit 3
  cuboid 1 7.38 -7.38 7.38 -7.38 193.63 -193.62
  cuboid 2 15.0 -15.0 15.0 -15.0 208.87 -208.87
```

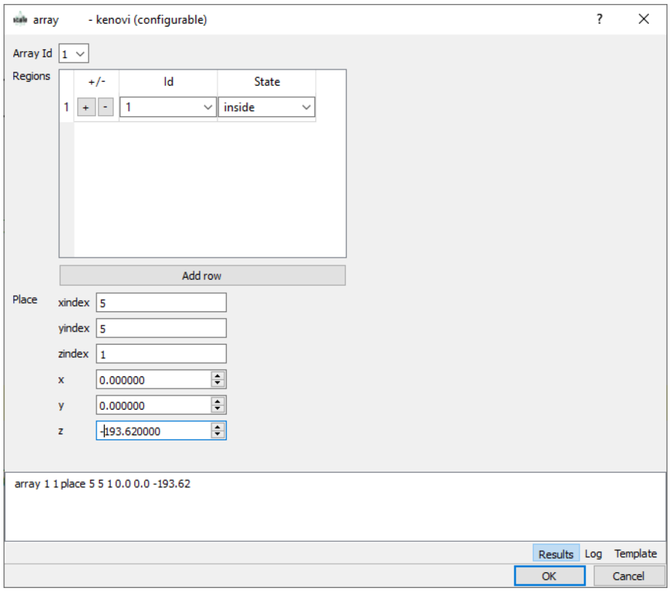
Figure 197. Input for both cuboids in Unit 3.

Now the contents of the regions defined in Unit 3 must be specified. The first cuboid (**cuboid 1**) will contain the array and the second cuboid (**cuboid 2**) will contain the water reflector. Cuboid 1 has been sized to be completely filled by array 1; recall from Section 4.6 that an ARRAY must be placed inside a region and must fill it completely. Array 1 will be positioned inside cuboid 1 using the PLACE entries on the ARRAY CONTENTS record; cuboid 1 is specified such that the center of the array will be at the origin of the unit. The central element of array 1 is element (5, 5, 1), which is a guide tube and thus Unit 2. The origin of Unit 2 is specified to be at the radial center of the guide tube and the bottom of the unit. The X and Y coordinates of the origin in the Unit 3 coordinate system are therefore (0, 0) so that the origin stays at the center of the UNIT. The origin of Unit 2 must be placed at -193.62 cm in the Z direction such that the axial middle of the array is at Z=0 in the Unit 3 coordinate system. This information can be provided in an array configurable form (Figure 198); select **array - kenovi (configurable)** from the autocomplete list. The **Array Id** is **1**, so no change is needed to the first field in the form. Press the **Add row** button at the bottom of the **Regions** pane to add a row to the pane; this array is being placed inside of region 1, so no changes are needed to the **Id** or **State** fields. As discussed above, the **xindex** and **yindex** should be set to `5`, and the **zindex** is `1`. Finally, the **x** and **y** fields remain `0`, and the **z** field is set to `-193.62`. The completed configurable form for the array 1 CONTENTS record (geometry block) is shown in Figure 199. Press the **OK** button to add the specification to the input file after confirming that all inputs are correct. The next step is to add a MEDIA record to place the water reflector around the assembly. This can be accomplished by entering `media 3 1 2 -1` in the input file. Recall that region 1 is filled with the array, so the MEDIA record must only specify that the water (mixture 3) be placed inside of region 2 and outside of region 1. The two CONTENTS records for Unit 3 are shown in Figure 200. Finally, the boundary record must be corrected; the outer boundary of Unit 3 is surface 2, not surface **1**. The complete input for **global unit 3** is shown in Figure 201; as before, additional comments and blank lines have been removed for the sake of clarity in this document.


opy Pas
fuel_as
documer
dodecahedron - kenovi (configurable)
dodecahedron
ellipsoid - kenovi (configurable)
ellipsoid
hexprism - kenovi (configurable)
hexprism
hopper - kenovi (configurable)
hopper
parallelepiped - kenovi (configurable)
parallelepiped
pentagon - kenovi (configurable)
pentagon
plane - kenovi (configurable)
plane
xpplane
ypplane
zpplane
quadratic - kenovi (configurable)
quadratic
rhexprism
rhomboid - kenovi (configurable)
rhomboid
ring - kenovi (configurable)
ring
sphere - kenovi (configurable)
sphere
wedge - kenovi (configurable)
wedge
media - kenovi (configurable)
media
hole - kenovi (configurable)
hole
array - kenovi (configurable)
array

```
-10
-20

ts below

 -7.38 7.38
 -15.0 15.0



' TODO: define global unit

  boundary 1
```

Figure 198. Selecting an ARRAY configurable form in the autocomplete menu in Fulcrum.



Array Id 1

Regions

	+/-	Id	State
1	+ -	1	inside

Place

xindex 5

yindex 5

zindex 1

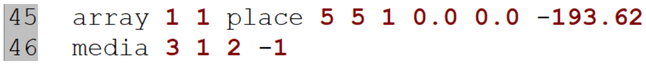
x 0.000000

y 0.000000

z -193.620000

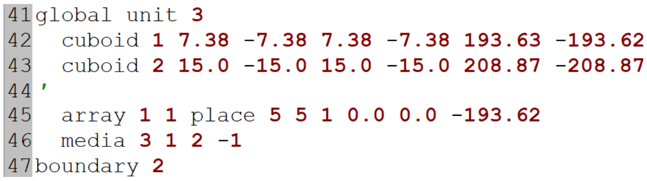
```
array 1 1 place 5 5 1 0.0 0.0 -193.62
```

Figure 199. Completed configurable form for the array 1 CONTENTS record.



```
  array 1 1 place 5 5 1 0.0 0.0 -193.62
  media 3 1 2 -1
```

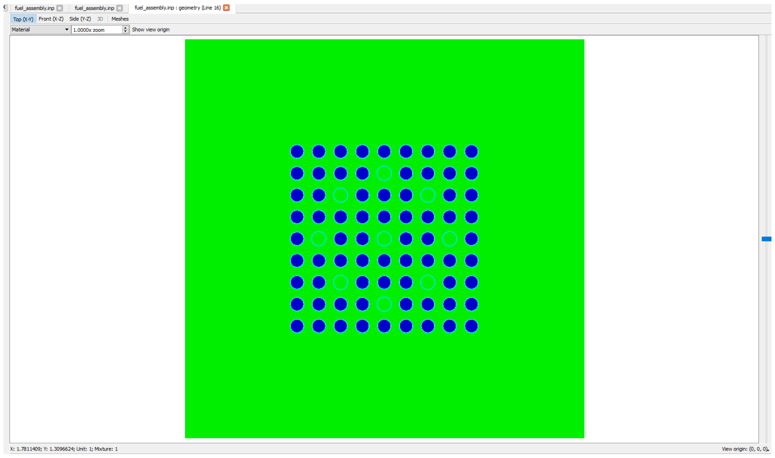
Figure 200. Both contents records needed in Global Unit 3.



```
global unit 3
  cuboid 1 7.38 -7.38 7.38 -7.38 193.63 -193.62
  cuboid 2 15.0 -15.0 15.0 -15.0 208.87 -208.87
'
  array 1 1 place 5 5 1 0.0 0.0 -193.62
  media 3 1 2 -1
boundary 2
```

Figure 201. Complete input for Global Unit 3.

Fulcrum visualization can now be used to check the model; a more complete description of Fulcrum's geometry visualization capabilities is provided in Section 9.2.1. To launch the visualization, click the **View** button between the **Run** and **Edit** buttons just above the text input screen. The default view is an XY slice through the axial center of the model, as shown in Figure 202.



X: 1.7811409; Y: 1.3096624; Unit: 1; Mixture: 1

View origin: (0, 0, 0)

Figure 202. Radial slice of the completed fuel assembly at the model midplane.

Now save all changes and run SCALE by clicking the **Run** button at the top of the Fulcrum window. The default run parameters of 1,000 particles per generation, 203 total generations, and 3 discarded generations will be used if no specific values are chosen by adding a PARAMETER block. These parameters yield a k_{eff} estimate of 0.6915 ± 0.0018; exact values will vary, as different computers and different operating systems generate different random numbers. A more rigorous calculation that simulates 10,000 particles per generation for 282 total generations and that skips the first 50 results in a k_{eff} estimate of 0.69087 ± 0.00049. This calculation repeated with CE cross sections yields a k_{eff} of 0.69168 ± 0.00059; the difference from the MG calculation is 0.00082 ± 0.00077. This difference is only ~1.1 standard deviations and thus is statistically insignificant. The LATTICECELL processing used in this problem provides sufficiently accurate cross sections to reproduce the CE solution.

8.5.2 Multiregion example: Parallel slab tanks

This problem consists of two parallel slab tanks constructed of SS304 that is 0.5 cm thick. The material in the tanks is a $U(93)O_2F_2$ solution with a fuel density of 459 g/L, a solution density of 1.566 g/cm^3, and no excess acid. Each slab tank has internal dimensions of 5 cm thick by 300 cm long by 150 cm high. Each tank is reflected by 5 cm of water on each side in the X direction (i.e., 10 cm of water between the tanks). There is no reflector material above the tanks or on the ends of the tanks. A 3D rendering of the system is shown in Figure 203.

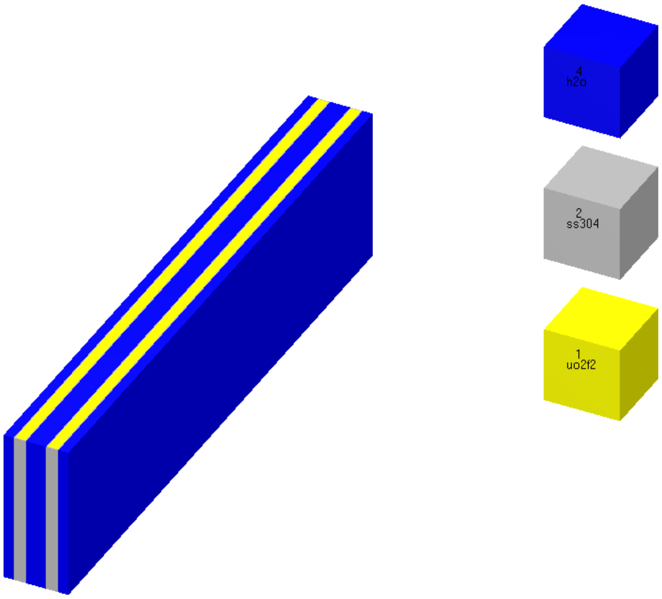
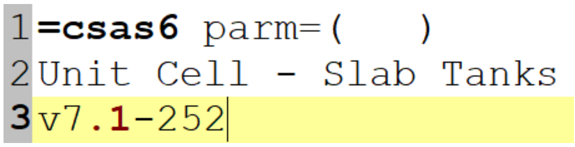


Figure 203. 3D rendering of the two tanks, with the top half of the model removed to show the solution.

8.5.2.1 General and Materials Data

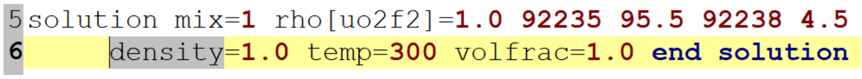
The initial data entry for this example is the same as in previous problems. Start Fulcrum and then select the **New file…** option from the **File** menu. Select a directory where the model will be saved, and name it `slab_tanks.inp`. Next, use the autocomplete feature in Fulcrum to create a **csas6 – Criticality safety analysis using KENO-VI** input. Replace **title-goes-here** with `Unit Cell - Slab Tanks`. Now change **xs-lib-goes-here** to `v7.1-252` (`v7.1-252n` if using SCALE 6.2, 6.2.1, or 6.2.2). This selects the 252-group neutron cross section library based on ENDF/B-VII.1 and signals to CSAS6 that this will be a MG KENO calculation. The appropriate cross section processing modules will be called using the input provided in the CELLDATA block as discussed in Section 8.5.2.2. The first three lines of the input are shown in Figure 204.



```
=csas6 parm=(    )
Unit Cell - Slab Tanks
v7.1-252
```

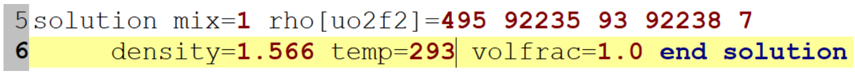
Figure 204. First three lines of the slab tanks model.

The materials information will be entered for the fissile solution and standard basic compositions for stainless steel and water. Position the cursor in the COMPOSITION block and select the **solution - rho + density + temperature + volume fraction** option. No configurable forms are available for fissile solution input, so template text is provided in the input, as shown in Figure 205. In the figure, the input has been divided onto two lines to allow better presentation in this document, but this has no impact on how CSAS processes the input. Change the concentration of the solution (**rho[uo2f2]**) from **1.0** to `459`, the ^{235}U (**92235**) wt.% to `93`, and the ^{238}U (**92238**) wt.% to `7`. Set the **density** of the solution to `1.566` and the temperature (**temp**) to `293` K. The updated, applicable solution input is shown in Figure 206.



```
solution mix=1 rho[uo2f2]=1.0 92235 95.5 92238 4.5
     density=1.0 temp=300 volfrac=1.0 end solution
```

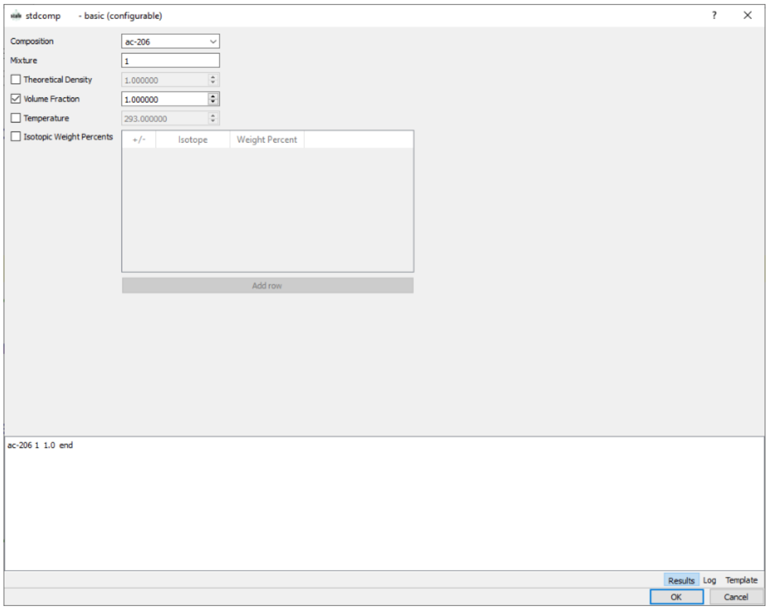
Figure 205. Initial template text provided by Fulcrum for a fissile solution.



```
solution mix=1 rho[uo2f2]=495 92235 93 92238 7
      density=1.566 temp=293 volfrac=1.0 end solution
```

Figure 206. Fissile solution input for this problem.

The stainless steel 304 and water compositions can both be added via configurable forms for basic standard compositions. The initial standard basic composition configurable is shown in Figure 207. Type `ss304` into the **Composition** box in the configurable form, set the **Mixture** number to `2`, and press **OK** to add the specification to the SCALE input. Two `ss304` and two `h2o` mixtures must be created because a unique mixture number is required for each region in the unit cell. The easiest way to do this is to copy and paste the existing **ss304** entry and change the mixture number to `3`. Another standard basic composition configurable form can be used to add the first water composition. Type `h2o` into the **Composition** box in the configurable form, set the **Mixture** number to `4`, and press **OK** to add the specification to the SCALE input. As with the stainless steel, copy and paste the mixture and change the mixture number from **4** to `5`. The complete composition block is shown in Figure 208.



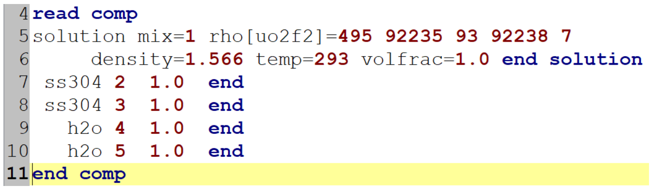
stdcomp - basic (configurable)

Composition: ac-206
Mixture: 1
☐ Theoretical Density: 1.000000
☑ Volume Fraction: 1.000000
☐ Temperature: 293.000000
☐ Isotopic Weight Percents

+/-	Isotope	Weight Percent

```
ac-206 1 1.0 end
```

Figure 207. Initial STDCOMP configurable form.



```
read comp
solution mix=1 rho[uo2f2]=495 92235 93 92238 7
     density=1.566 temp=293 volfrac=1.0 end solution
 ss304 2  1.0  end
 ss304 3  1.0  end
   h2o 4  1.0  end
   h2o 5  1.0  end
end comp
```

Figure 208. Final composition block for this model.

8.5.2.2 Unit Cell Data

The unit cell information for the cross section treatment can be specified now that the material information has been entered. One of the four unit cell types (INFHOMMEDIUM, LATTICECELL, MULTIREGION, DOUBLEHET) will be specified to process the MG cross sections for this problem. A MULTIREGION SLAB cell will allow cross section processing to be performed on almost the exact geometry for this problem. While INFHOMMEDIUM may result in acceptable answers, the fact that the tanks are thin relative to their length and height makes MULTIREGION a better option. On the other hand, the tanks are wide enough and tall enough that the infinite slab approach should yield good results, and the BUCKLEDSLAB option is not needed. In this case, the left boundary can be set as reflective to allow specification of one side of the problem to represent both tanks.

To add the CELLDATA block, place the cursor between the COMPOSITION and GEOMETRY blocks and autocomplete for a list of available blocks to add. Select **cells**, as show in Figure 209. There are no configurable forms currently available in the CELLDATA block. Autocompleting in the CELLDATA block provides a list of available unit cell specifications; select the **multiregion - slab** option as shown in Figure 210. Fulcrum inserts generic CELLDATA input, as shown in Figure 211, which must be modified for use in this model.

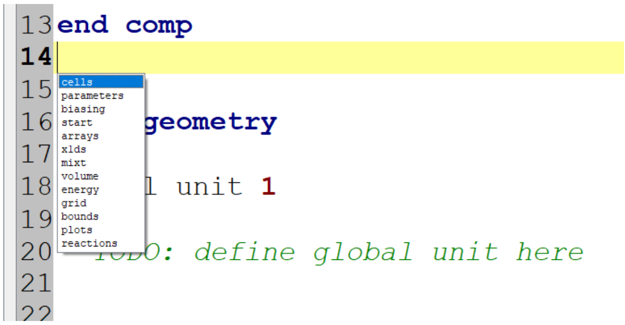


Figure 209. Autocomplete list of available blocks for inclusion in the model.

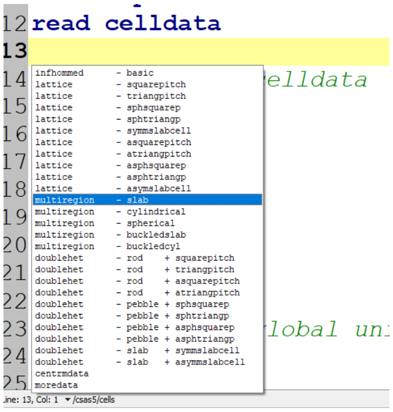
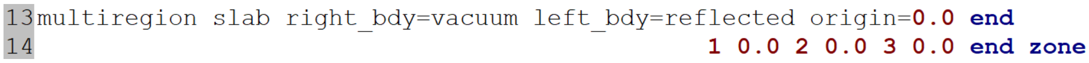

read celldata
infhommed - basic
lattice - squarepitch
lattice - triangpitch
lattice - sphsquarep
lattice - sphtriangp
lattice - symmslabcell
lattice - asquarepitch
lattice - atriangpitch
lattice - asphsquarep
lattice - asphtriangp
lattice - asymslabcell
multiregion - slab
multiregion - cylindrical
multiregion - spherical
multiregion - buckledslab
multiregion - buckledcyl
doublehet - rod + squarepitch
doublehet - rod + triangpitch
doublehet - rod + asquarepitch
doublehet - rod + atriangpitch
doublehet - pebble + sphsquarep
doublehet - pebble + sphtriangp
doublehet - pebble + asphsquarep
doublehet - pebble + asphtriangp
doublehet - slab + symmslabcell
doublehet - slab + asymmslabcell
centrmdata
moredata
elldata
lobal un
ine: 13, Col: 1 /csas5/cells

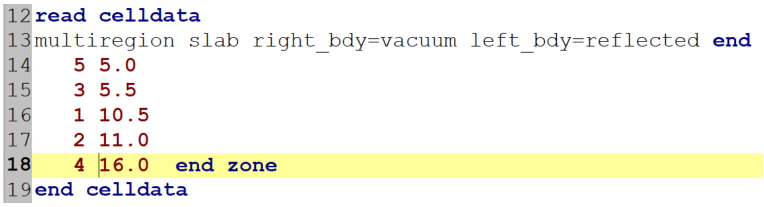
Figure 210. Autocomplete list of cells within the CELLDATA block.



```
multiregion slab right_bdy=vacuum left_bdy=reflected origin=0.0 end
                                                  1 0.0 2 0.0 3 0.0 end zone
```

Figure 211. Templated input provided for MULTIREGION SLAB cell by Fulcrum.

The templated input must now be replaced with the appropriate input for this problem. The geometry to be constructed in the CELLDATA block will be one tank, with the associated water around the tank. The right boundary (**right_bdy**) will therefore remain a **vacuum** boundary condition, and the left boundary (left_bdy) will remain a **reflected** boundary condition. The specification of the origin at 0.0 can remain or be removed; it has no effect, as it is explicitly specifying the default value. The remaining inputs are the zones present, from left to right, represented as infinite slabs. These regions are water, tank wall, solution, tank wall, and water; each will be specified with the appropriate mixture number. The distances to be specified are those from the left boundary of the cell to the right boundary of the region; these inputs must be monotonically increasing, as they represent the accumulating distance and not the thickness of the relevant regions. The zone inputs to be provided are as follows: `5 5.0  3 5.5  1 10.5  2 11.0 4 16.0`. The input is not white-space dependent and can be entered on a single line as provided, or it can be entered in columns as shown in Figure 212. This arrangement is more convenient, makes it easier to track which mixtures have been used, and helps to ensure that the correct increasing dimensions have been provided. Note that different water mixtures and different stainless steel mixtures are used to meet the requirement that each mixture is used in only one region of one cell in the CELLDATA block. These duplicate mixtures will not be used in the KENO-VI geometry, but they are necessary for cross section processing. If the neutron flux, and therefore the MG cross sections, were significantly different between the two sides of the tank, then the different mixtures should be used in the respective regions in the KENO-VI geometry.



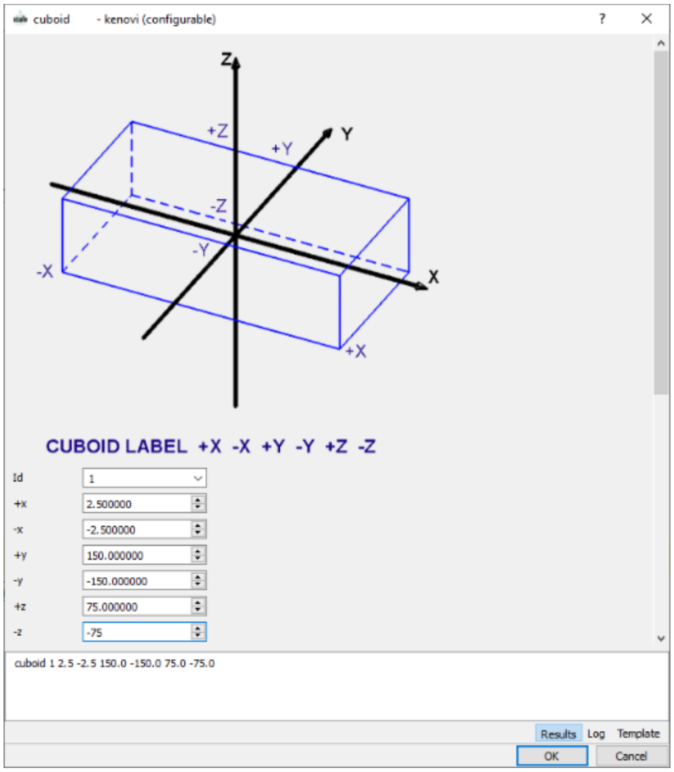
```
read celldata
multiregion slab right_bdy=vacuum left_bdy=reflected end
   5 5.0
   3 5.5
   1 10.5
   2 11.0
   4 16.0  end zone
end celldata
```

Figure 212. Final CELLDATA block showing the correct input for this model.

8.5.2.3 KENO-VI Geometry Data

This model requires only a single unit and will use a reflecting boundary condition to reduce the amount of the problem that must be specified. The entire geometry could be specified using an array, but this model provides a demonstration of using boundary conditions to simplify the geometry specification for the model. In this case, three regions are required: the solution, the tank, and the water outside the tank.

The first region is the inner volume of the tank, containing the solution. This is a cuboid that is 5 cm thick in the X direction, 300 cm long in the Y direction, and 150 cm tall in the Z direction. This region can be added with a cuboid configurable form in Fulcrum. The appropriate dimensions can be added to the form; the origin of the model is arbitrary and is positioned in the center of the fissile solution in the form shown in Figure 213. After checking the input to confirm that it is correct, press **OK** to enter the cuboid entry into the input file. Figure 214 shows the cuboid entry in the input file.



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

Id	1
+x	2.500000
-x	-2.500000
+y	150.000000
-y	-150.000000
+z	75.000000
-z	-75

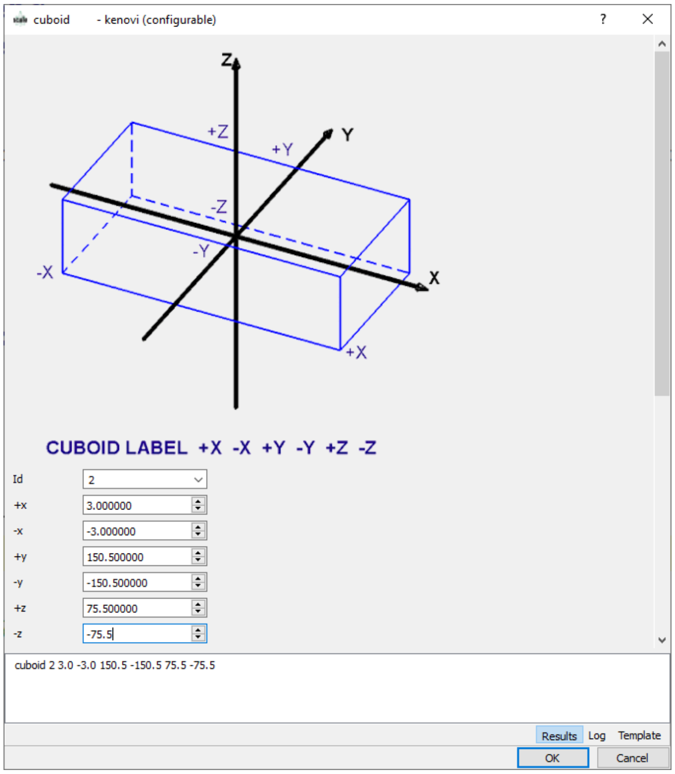
```
cuboid 1 2.5 -2.5 150.0 -150.0 75.0 -75.0
```

Figure 213. Cuboid configurable form for region inside slab tank.

```
global unit 1
  cuboid 1 2.5 -2.5 150.0 -150.0 75.0 -75.0
```

Figure 214. Cuboid input for region inside slab tank.

The next region is the tank wall, which is 0.5 cm thick in all 6 directions. The cuboid can be added with a configurable form with an **Id** of 2. The dimensions of cuboid 2 are ±3.0 in the X direction, ±150.5 in the Y direction, and ±75.5 in the Z direction. The final configurable form for this region is shown in Figure 215; after confirming that all input is correct, press the **OK** button to add **cuboid 2** to the input. The final region is the water outside the tank on the +X and -X faces. These water regions are 5 cm on each side of the tank, so the dimensions are ±8. This cuboid can be added via a configurable form as cuboid 3, as shown in Figure 216, after the inputs have been confirmed to be correct. All the geometry records are shown in Figure 217.

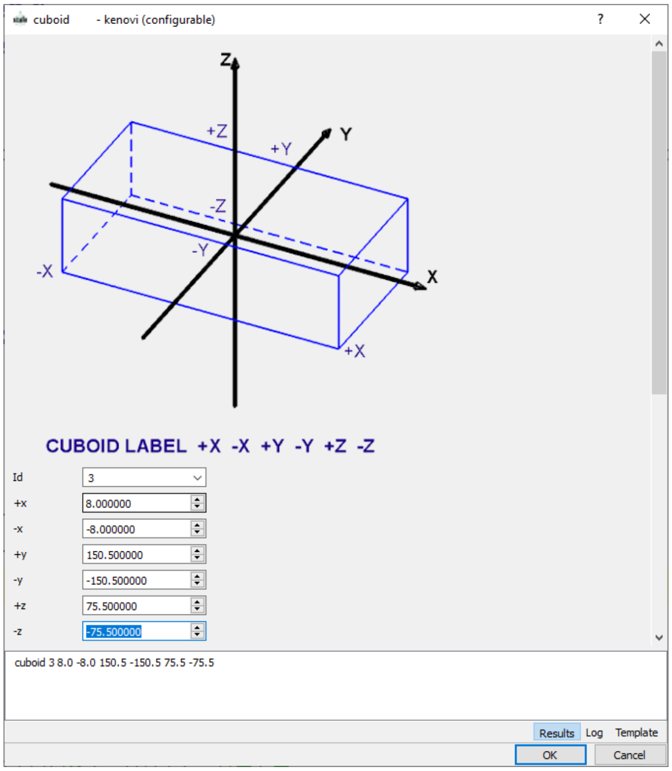


CUBOID LABEL +X -X +Y -Y +Z -Z

Id	2
+x	3.000000
-x	-3.000000
+y	150.500000
-y	-150.500000
+z	75.500000
-z	-75.5

```
cuboid 2 3.0 -3.0 150.5 -150.5 75.5 -75.5
```

Figure 215. Complete configurable form for cuboid 2, the tank wall.

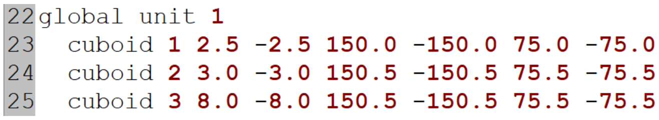


CUBOID LABEL +X -X +Y -Y +Z -Z

Id	3
+x	8.000000
-x	-8.000000
+y	150.500000
-y	-150.500000
+z	75.500000
-z	-75.500000

```
cuboid 3 8.0 -8.0 150.5 -150.5 75.5 -75.5
```

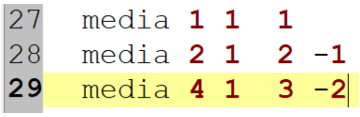
Figure 216. Complete configurable form for cuboid 3, the water reflector.



```
global unit 1
  cuboid 1 2.5 -2.5 150.0 -150.0 75.0 -75.0
  cuboid 2 3.0 -3.0 150.5 -150.5 75.5 -75.5
  cuboid 3 8.0 -8.0 150.5 -150.5 75.5 -75.5
```

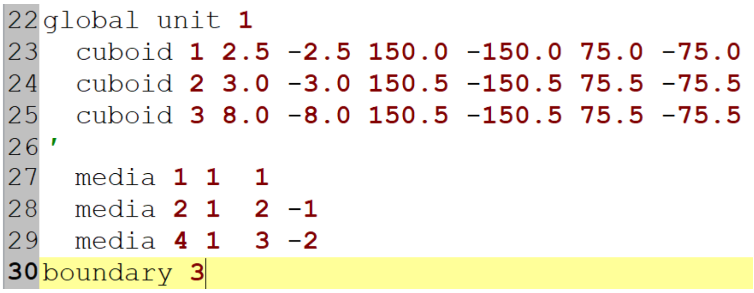
Figure 217. All three cuboids, the only three geometry records needed for this model.

Next, the contents records must be added. In this case, three MEDIA records are needed to add the solution, the stainless steel, and the water into the appropriate cuboids. The first record can be added as `media 1 1  1`, to add the solution (mixture 1) into cuboid 1. The input can be added directly or by using a media configurable form. The second record, which can also be added either directly or via configurable form, places the stainless steel (mixture 2) inside the tank wall but outside the inside volume of the tank. The input for this record is `media 2 1  2 -1`. Finally, filling cuboid 3, outside of cuboid 2, with water (mixture 4) can be accomplished with `media 4 1  3 -2`. All three contents records are shown in Figure 218. The last entry to complete **global unit 1** is to update the **boundary** record. Cuboid 3 forms the outer boundary of the unit, so the boundary record should be updated to `boundary 3`. The entire specification for global unit 1 is provided in Figure 219.



```
  media 1 1  1
  media 2 1  2 -1
  media 4 1  3 -2
```

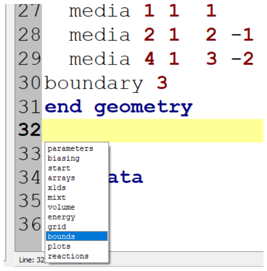
Figure 218. All three media records needed for this model.



```
global unit 1
  cuboid 1 2.5 -2.5 150.0 -150.0 75.0 -75.0
  cuboid 2 3.0 -3.0 150.5 -150.5 75.5 -75.5
  cuboid 3 8.0 -8.0 150.5 -150.5 75.5 -75.5
'
  media 1 1  1
  media 2 1  2 -1
  media 4 1  3 -2
boundary 3
```

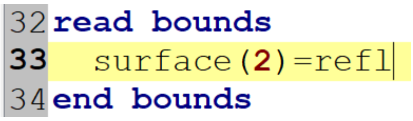
Figure 219. Entire specification for global unit 1.

The model with a single tank explicitly represented in it can be made to represent a system with two tanks by adding a reflective boundary condition to one of the X surfaces. This returns a neutron through the 5 cm water region it just traversed, so it must travel through a total of 10 cm of water to return to the tank. This exactly describes the problem as stated above. The boundary conditions on the outer surface of the **global unit** are supplied in the BOUNDS block. The BOUNDS block can be added by autocomplete if the cursor is placed outside of any other blocks, as shown in Figure 220. No autocomplete is available in Fulcrum in the BOUNDS block, but the options are described in [Albedo data] in the KENO section of the manual. The default boundary condition in KENO-VI is vacuum, so only one X face needs to be changed. The boundary condition is changed by specifying a face code and the boundary condition to apply to that face. For cuboid outer boundaries, the KENO V.a face codes may be used. For instance, the face code for the positive X face is +XB=, and for the negative X face it is -XB=. For cuboids and all other shapes, each face of the surface can be referenced via a number; the list of these face numbers for each surface is provided in [Face codes and surface numbers for entering boundary (albedo) conditions] in the [Albedo data] section of the KENO manual. A mirror boundary condition can be specified with the keywords MIRROR or REFLECT, or a shortened form of either. For this model, the KENO V.a style input would be `-XB=REFL`. In the more general approach, the number of the -X face of a cuboid is 2, and the input is then `surface(2)=REFL`. This approach is shown in Figure 221.



```
  media 1 1  1
  media 2 1  2 -1
  media 4 1  3 -2
boundary 3
end geometry

      ata
```

Figure 220. Adding the BOUNDS block from the Fulcrum autocomplete list.



```
read bounds
   surface(2)=refl
end bounds
```

Figure 221. Boundary condition specification to apply a reflective boundary condition to the -X face of the global unit.

The model is complete, and a Fulcrum 2D view of the entire model is shown in Figure 222. A magnified view of one end of the model is shown in Figure 223; the color scheme in these figures is the same as that used in Figure 203: yellow for UO_2F_2 solution, gray for stainless steel, and blue for water. There is no indication in the geometry rendering that a reflective boundary condition has been applied to the -X face of the model.

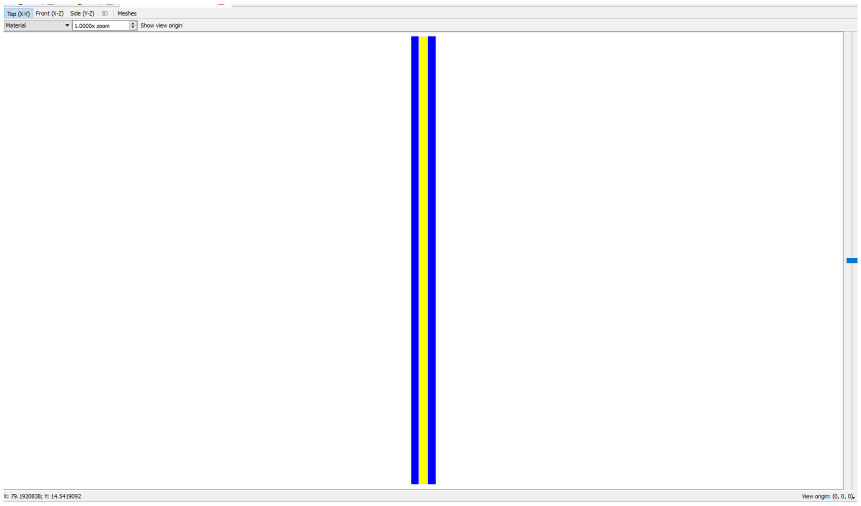


Figure 222. Fulcrum 2D visualization of the axial midplane of the model.

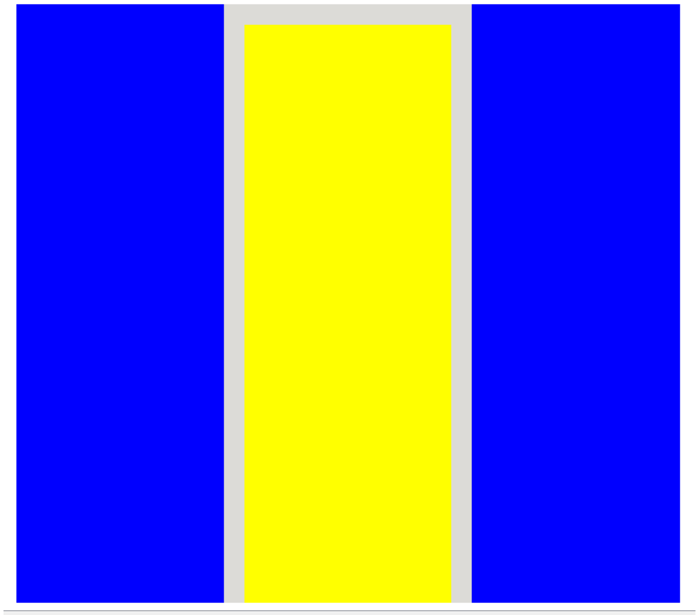


Figure 223. Detailed view of the +Y end of the axial midplane of the model, showing the end of the tank.

Now save all changes and run SCALE by clicking the **Run** button at the top of the Fulcrum window. The default run parameters of 1,000 particles per generation, 203 total generations, and 3 discarded generations will be used if no specific values are chosen by adding a PARAMETER block. These parameters yield a k_{eff} estimate of 0.9519 ± 0.0022; exact values will vary, as different computers and different operating systems generate different random numbers. A model using an array to explicitly represent both tanks resulted in a k_{eff} estimate of 0.9497 ± 0.0021. The difference between these results is 0.0022 ± 0.0030; since this difference is less than one standard deviation, the estimates are statistically equivalent. This demonstrates that the model with the single explicit tank and the reflective boundary condition can be used in place of the more complicated geometry with both tanks explicitly represented.

A more rigorous calculation, simulating 10,000 particles per generation for 377 total generations and skipping the first 50, resulted in a k_{eff} estimate of 0.94895 ± 0.00049. This calculation repeated with CE cross sections yields a k_{eff} of 0.94841 ± 0.00056; the difference from the MG calculation is 0.00054 ± 0.00074. This difference is less than one standard deviation, so the estimates are therefore statistically equivalent. The MULTIREGION processing used in this problem provides sufficiently accurate cross sections to reproduce the CE solution.

8.5.3 DOUBLEHET Example: Infinite, Square-pitched Array of Pebbles

This sample problem is based on a pebble bed modular reactor (PBMR) computational benchmark organized by the OECD/NEA [6]. Each fuel pebble contains 15,000 TRISO grains in a graphite matrix, and each grain is UO_2 with an enrichment of 9.6 wt.% ^{235}U, and the balance is ^{238}U. The UO_2 density is 10.4 g/cm^3, and the radius of the fuel grain is 0.02397 cm. Each kernel has four coatings: a porous carbon layer, a pyrolytic carbon layer, a silicon carbide layer, and another pyrolytic carbon layer. The dimensions and densities of these layers is provided in Table 3. The radius of the fuel region is 2.5 cm, the total pebble radius is 3.0 cm, and the pitch of the pebbles is 6.1 cm. The outer layer of the pebble is composed of graphite; the matrix and pebble outer layer graphite has a density of 2.1 g/cm^3. Schematics of the grain and pebble are shown in Figure 224. Helium gas is the coolant flowing through the pebble bed at a density of 0.0061 g/cm^3; for simplicity in this problem, all materials are assumed to be at 700 K.

Table 3. Thickness and density of grain coatings

Coating material	Material density	Coating thickness (μm)
Porous carbon	1.05	95
Pyrolytic carbon	1.90	40
Silicon carbide	3.18	35
Pyrolytic carbon	1.90	40

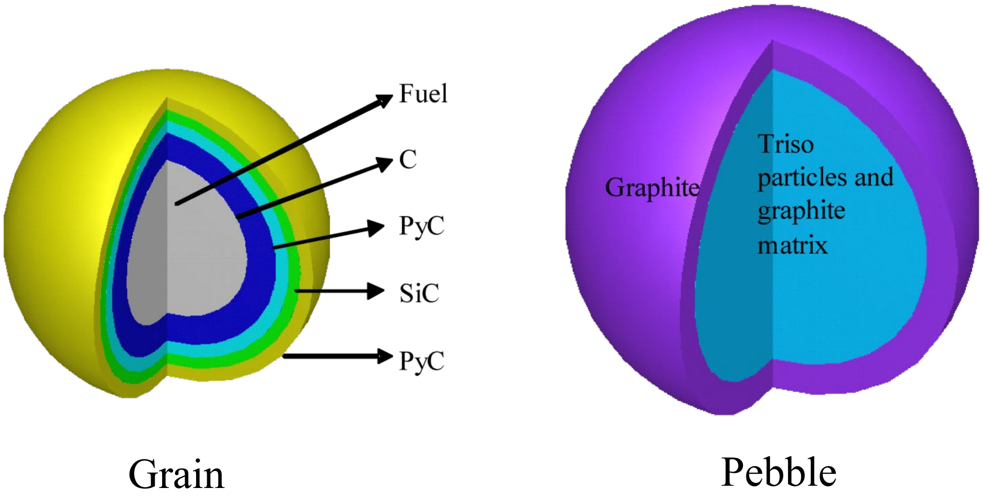
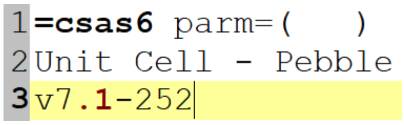


Figure 224. Schematic of individual grain (right) and pebble (left).

8.5.3.1 General and materials data

The initial data entry for this example is the same as previous problems. Start Fulcrum and then select the **New file…** option from the **File** menu. Select a directory where the model will be saved, and name it `pebble.inp`. Next, use the autocomplete feature in Fulcrum to create a **csas6 – Criticality safety analysis using KENO-VI** input. Replace **title-goes-here** with `Unit Cell - Pebble`. Now change **xs-lib-goes-here** to `v7.1-252` (`v7.1-252n` if using SCALE 6.2, 6.2.1, or 6.2.2). This selects the 252-group neutron cross section library based on ENDF/B-VII.1 and signals to CSAS6 that this will be a MG KENO calculation. The appropriate cross section processing modules will be called using the input provided in the CELLDATA block as discussed in Section 8.5.3.2 below. The first three lines of the input are shown in Figure 225.

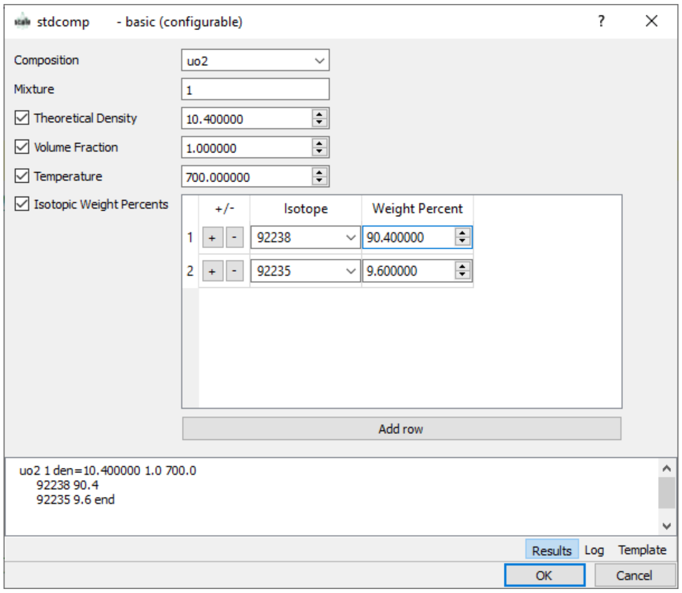


```
=csas6 parm=(    )
Unit Cell - Pebble
v7.1-252
```

Figure 225. First three lines from pebble problem input.

A number of different mixtures must be specified in the COMPOSITION block because each mixture can only be used in one region in the CELLDATA block. A single UO_2 mixture is needed, as well as the porous carbon mixture, two pyrolytic carbon mixtures, a SiC layer, a graphite matrix, a graphite pebble layer, and a helium coolant mixture.

The first mixture to define is the UO_2 fuel mixture. UO_2 is a basic standard composition, so the configurable form can be used to create the composition input. With the cursor in the COMPOSITION block, choose **stdcomp - basic (configurable)** to launch the configurable form. Type `uo2` into the **Composition** field. Check the box for **Theoretical Density** and enter `10.4` g/cm^3 in the associated field; leave the **Volume Fraction** as 1.0. Check the box for **Temperature** and enter `700` K in the associated field. Check the box for **Isotopic Weight Percents** and click the **Add row** button at the bottom of the pane. Select **92235** for the **Isotope** and enter `9.6` wt.%. Add another row by clicking the + button on the **92235** row or the **Add row** button and select **92238** and `90.4` wt.%. After confirming that the input is correct, press the **OK** button to add the UO_2 composition definition to the input. A picture of the final configurable form is provided in Figure 226.



stdcomp - basic (configurable)

Composition: uo2

Mixture: 1

☑ Theoretical Density: 10.400000

☑ Volume Fraction: 1.000000

☑ Temperature: 700.000000

☑ Isotopic Weight Percents

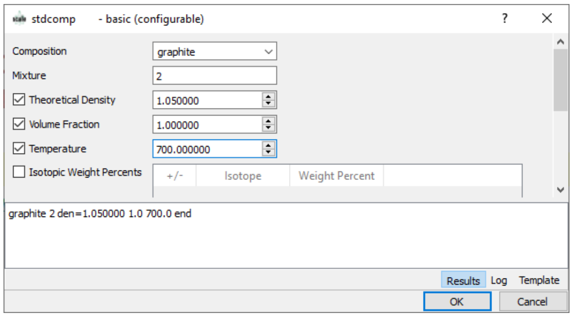
	+/-	Isotope	Weight Percent
1	+ -	92238	90.400000
2	+ -	92235	9.600000

```
uo2 1 den=10.400000 1.0 700.0
    92238 90.4
    92235 9.6 end
```

Figure 226. Configurable form for UO_2 composition.

The next four mixtures will be for the coatings on the grain. The porous and pyrolytic carbon layers will be specified as graphite to use the graphite thermal scattering law (TSL) data, which is the most appropriate option available in ENDF/B-VII.1. The SiC layer will be created as an ATOM mixture.

Create a new standard basic composition configurable form and enter `graphite` in the **Composition** field. Make this **Mixture** `2`, check the **Theoretical Density** check box, and enter `1.05` g/cm^3 in the associated field. Check the **Temperature** box and enter `700` K in the field. After checking the input, press the **OK** button to add the composition input to the CSAS input. A picture of the final configurable form is provided in Figure 227. Once the mixture has been defined in the COMPOSITION block, copy the line of input and paste it on the next line. Change the mixture number to `3` and the theoretical density to `1.90` g/cm^3. The updated composition block is shown in Figure 228.

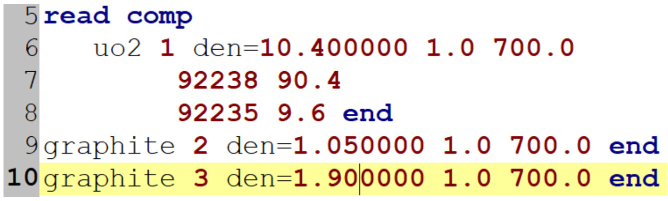


Composition: graphite
Mixture: 2
☑ Theoretical Density: 1.050000
☑ Volume Fraction: 1.000000
☑ Temperature: 700.000000
☐ Isotopic Weight Percents

+/-	Isotope	Weight Percent

```
graphite 2 den=1.050000 1.0 700.0 end
```

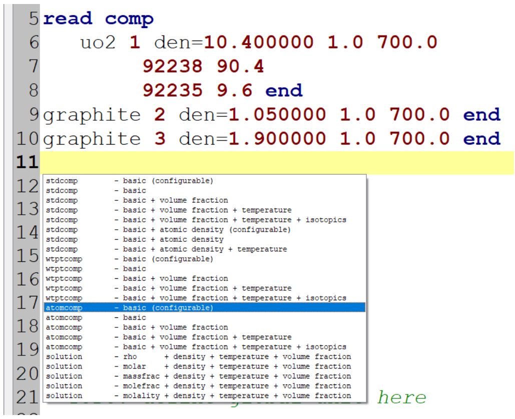
Figure 227. Configurable form for the porous carbon layer composition.



```
read comp
    uo2 1 den=10.400000 1.0 700.0
          92238 90.4
          92235 9.6 end
graphite 2 den=1.050000 1.0 700.0 end
graphite 3 den=1.900000 1.0 700.0 end
```

Figure 228. Input with UO_2, porous carbon layer, and inner pyrolytic carbon layer compositions.

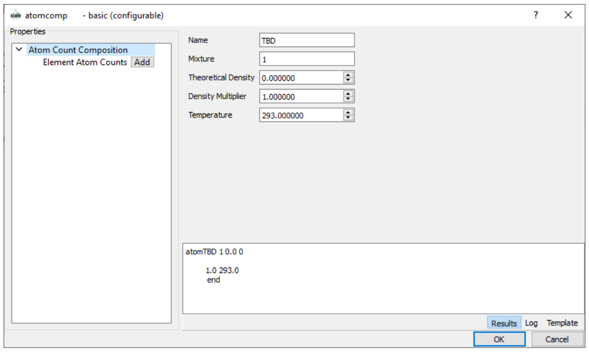
Mixture 4 will be the SiC layer; SiC is not a standard basic composition, but it can be entered as an ATOM composition because it has a chemical formula. With the cursor below the three mixtures already defined but still in the COMPOSITION block, autocomplete and select **atomcomp - basic (configurable)**, as shown in Figure 229. It may be necessary to resize the configurable window, and especially the **Properties** pane, to show the **Add** button for **Element Atom Counts**. An empty but appropriately sized ATOM composition configurable form is shown in Figure 230. Change the text in the **Name** field to `SiC` and set this as **Mixture** `4`. Set the **Theoretical Density** to `3.18` g/cm^3 and the **Temperature** to `700` K. Click the **Add** button next to **Element Atom Counts**, select **14000** from the **Element** dropdown menu to select Si, and set the **Atom Count** to `1`. Click the **Add** button next to **Element Atom Counts** again, select **6000** from the **Element** dropdown menu to select C, and again set the **Atom Count** to `1`. After checking the input, press the **OK** button to add the SiC composition to the input. The complete configurable form is shown in Figure 231; note that the carbon data are showing since it is impossible to display both the Si and C pages at the same time.



```
read comp
    uo2 1 den=10.400000 1.0 700.0
          92238 90.4
          92235 9.6 end
graphite 2 den=1.050000 1.0 700.0 end
graphite 3 den=1.900000 1.0 700.0 end

```

stdcomp - basic (configurable)
stdcomp - basic
stdcomp - basic + volume fraction
stdcomp - basic + volume fraction + temperature
stdcomp - basic + volume fraction + temperature + isotopics
stdcomp - basic + atomic density (configurable)
stdcomp - basic + atomic density
stdcomp - basic + atomic density + temperature
wtptcomp - basic (configurable)
wtptcomp - basic
wtptcomp - basic + volume fraction
wtptcomp - basic + volume fraction + temperature
wtptcomp - basic + volume fraction + temperature + isotopics
atomcomp - basic (configurable)
atomcomp - basic
atomcomp - basic + volume fraction
atomcomp - basic + volume fraction + temperature
atomcomp - basic + volume fraction + temperature + isotopics
solution - rho + density + temperature + volume fraction
solution - molar + density + temperature + volume fraction
solution - massfrac + density + temperature + volume fraction
solution - molefrac + density + temperature + volume fraction
solution - molality + density + temperature + volume fraction
here

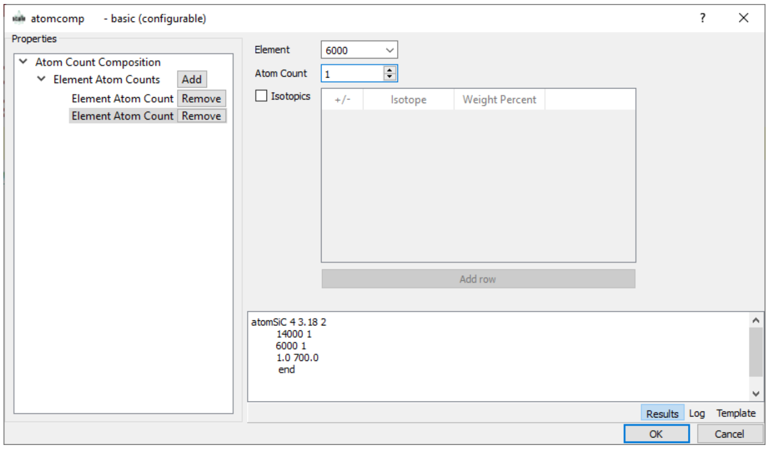
Figure 229. Selecting ATOM mixture configurable form in Fulcrum.



Name: TBD
Mixture: 1
Theoretical Density: 0.000000
Density Multiplier: 1.000000
Temperature: 293.000000

```
atomTBD 1 0.0 0

    1.0 293.0
    end
```

Figure 230. ATOM mixture configurable form showing all necessary buttons and fields.

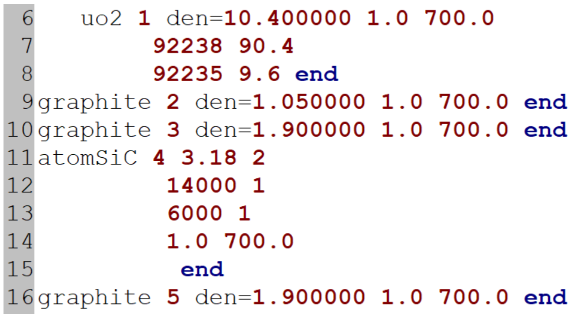


Element: 6000
Atom Count: 1
☐ Isotopics

```
atomSiC 4 3.18 2
     14000 1
     6000 1
     1.0 700.0
      end
```

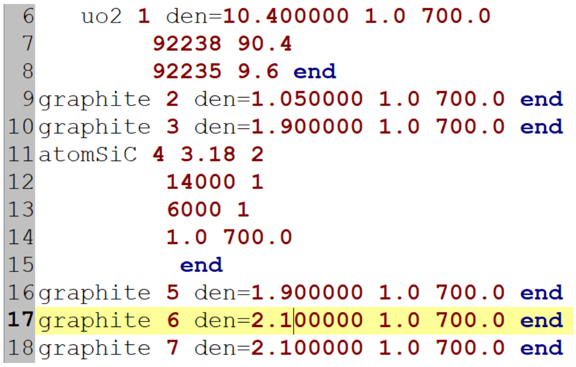
Figure 231. Complete ATOM composition configurable form for the SiC layer composition.

Copy the input for the inner pyrolytic carbon layer (mixture 3) and paste it below the **atomSiC** input. The only necessary modification to the input is to change the mixture number to 5. Figure 232 shows the first five mixture inputs in the COMPOSITION block. The next two compositions needed are the graphite for the matrix and the outer coating of the pebble. These can also be created by copying the mixture 5 graphite and pasting it into the input. These mixtures should be 6 for the matrix and 7 for the outer layer of the pebble. Both these mixtures have a density of 2.1 g/cm^3. The increased COMPOSITION block is shown in Figure 233. This completes the mixtures needed for the pebble definition.



```
   uo2 1 den=10.400000 1.0 700.0
        92238 90.4
        92235 9.6 end
graphite 2 den=1.050000 1.0 700.0 end
graphite 3 den=1.900000 1.0 700.0 end
atomSiC 4 3.18 2
         14000 1
         6000 1
         1.0 700.0
          end
graphite 5 den=1.900000 1.0 700.0 end
```

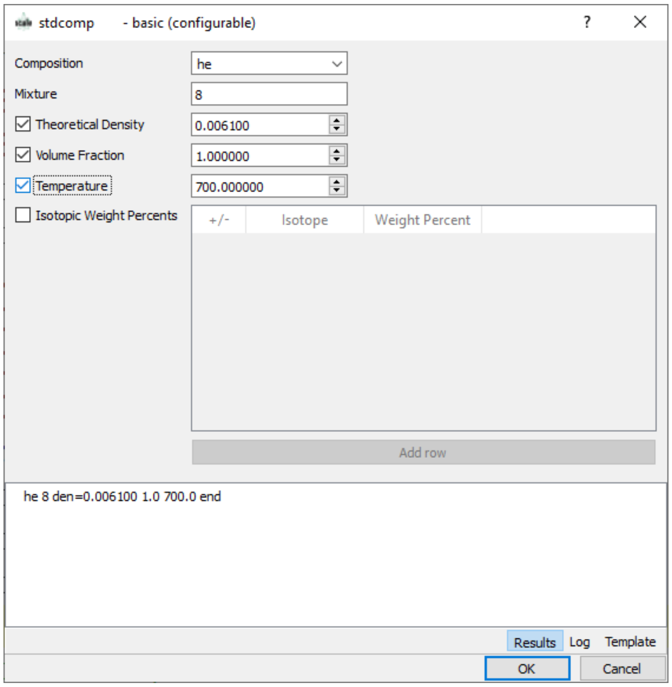
Figure 232. Mixture definitions for all composition in the fuel grains.



```
   uo2 1 den=10.400000 1.0 700.0
        92238 90.4
        92235 9.6 end
graphite 2 den=1.050000 1.0 700.0 end
graphite 3 den=1.900000 1.0 700.0 end
atomSiC 4 3.18 2
         14000 1
         6000 1
         1.0 700.0
          end
graphite 5 den=1.900000 1.0 700.0 end
graphite 6 den=2.100000 1.0 700.0 end
graphite 7 den=2.100000 1.0 700.0 end
```

Figure 233. Composition definitions for all mixtures in the pebble.

The last mixture definition for this problem is the helium coolant. This can be added as a standard basic composition using a configurable form. Use the **Composition** name `he` (be sure to select the element and not one of the isotopes), set the **Mixture** number to `8`, check the box for **Theoretical Density**, and set it to `0.0061` g/cm^3. Finally, check the box for **Temperature** and set it for `700` K. After checking the input, press the **OK** button to add the helium definition to the input; the complete configurable form is shown in Figure 234.



stdcomp - basic (configurable)

Composition: he

Mixture: 8

☑ Theoretical Density: 0.006100

☑ Volume Fraction: 1.000000

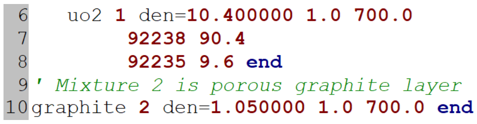
☑ Temperature: 700.000000

☐ Isotopic Weight Percents

```
he 8 den=0.006100 1.0 700.0 end
```

Figure 234. Complete configurable form for helium coolant mixture.

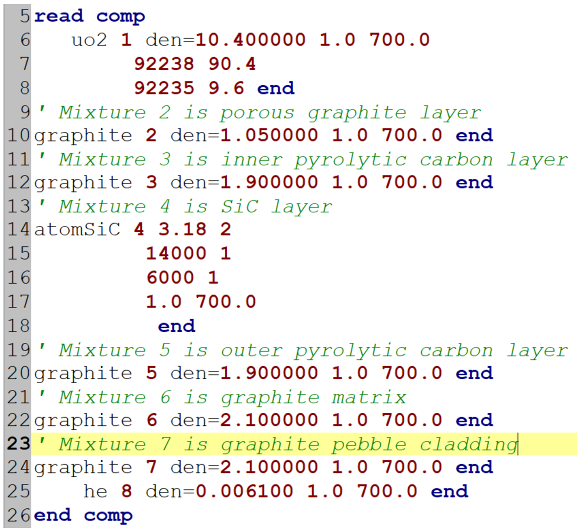
Comments should be added in the input to clarify which mixtures are to be used for which regions. This will make future updates and reviews much easier. Recall that SCALE interprets an entire line as a comment if it starts with an apostrophe (`'`) in column 1. Insert a blank line above the graphite mixture that is Mixture 2 by placing the cursor in column one and pressing enter. Move back to this new blank line, place an apostrophe in column 1, and enter the following comment: `Mixture 2 is porous graphite layer`. The updated input is shown in Figure 235. Repeat this process to add comments for Mixture 3 (`' Mixture 3 is inner pyrolytic carbon layer`), Mixture 4 (`' Mixture 4 is silicon carbide layer`), Mixture 5 (`' Mixture 5 is outer pyrolytic carbon `layer`), Mixture 6 (`' Mixture 6 is graphite matrix`), and Mixture 7 (`' Mixture 7 is graphite pebble cladding`).



```
   uo2 1 den=10.400000 1.0 700.0
        92238 90.4
        92235 9.6 end
' Mixture 2 is porous graphite layer
graphite 2 den=1.050000 1.0 700.0 end
```

Figure 235. Comment added to input.

This completes the mixture specifications in the COMPOSITION block for this model. The complete input block is shown in Figure 236. Blank lines and extraneous comments have been removed from the input for clarity in the figure.



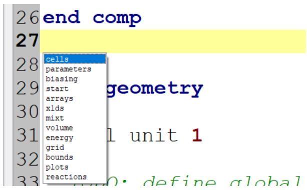
```
read comp
   uo2 1 den=10.400000 1.0 700.0
        92238 90.4
        92235 9.6 end
' Mixture 2 is porous graphite layer
graphite 2 den=1.050000 1.0 700.0 end
' Mixture 3 is inner pyrolytic carbon layer
graphite 3 den=1.900000 1.0 700.0 end
' Mixture 4 is SiC layer
atomSiC 4 3.18 2
         14000 1
         6000 1
         1.0 700.0
          end
' Mixture 5 is outer pyrolytic carbon layer
graphite 5 den=1.900000 1.0 700.0 end
' Mixture 6 is graphite matrix
graphite 6 den=2.100000 1.0 700.0 end
' Mixture 7 is graphite pebble cladding
graphite 7 den=2.100000 1.0 700.0 end
    he 8 den=0.006100 1.0 700.0 end
end comp
```

Figure 236. Entire COMPOSITION block for the pebble model.

8.5.3.2 Unit cell data

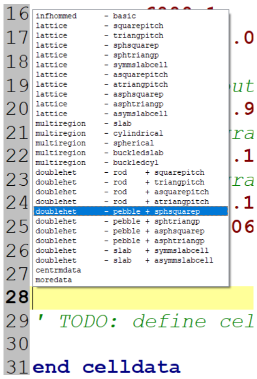
The unit cell information for the cross section treatment can be specified now that the material information has been entered. One of the four unit cell types (INFHOMMEDIUM, LATTICECELL, MULTIREGION, DOUBLEHET) will be specified to process the MG cross sections for this problem. A DOUBLEHET cell will allow cross section processing to be performed accurately for this problem. The individual grain and coatings will be specified, and the PEBBLE option will allow the description of the pebble and unit cell boundary conditions to approximate the infinite array of pebbles.

To add the CELLDATA block, place the cursor between the COMPOSITION and GEOMETRY blocks and autocomplete for a list of available blocks to add. Select **cells** as show in Figure 237. There are no configurable forms currently available in the CELLDATA block. Autocompleting in the CELLDATA block provides a list of available unit cell specifications; select the **doublehet - pebble + sphsquarep** option as shown in Figure 238. Fulcrum inserts generic CELLDATA input, as shown in Figure 239, which must be modified for use in this model.



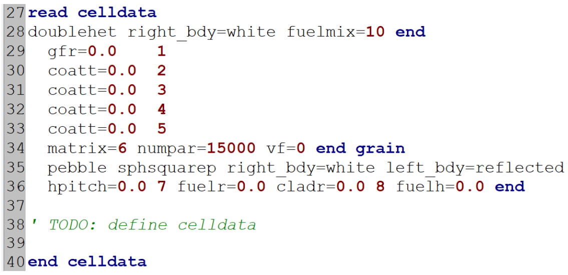
```
end comp

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

Figure 237. Autocomplete list of available blocks for inclusion in the model



```
infhommed      - basic
lattice        - squarepitch
lattice        - triangpitch
lattice        - sphsquarep
lattice        - sphtriangp
lattice        - symmslabcell
lattice        - asquarepitch
lattice        - atriangpitch
lattice        - asphsquarep
lattice        - asphtriangp
lattice        - asymslabcell
multiregion    - slab
multiregion    - cylindrical
multiregion    - spherical
multiregion    - buckledslab
multiregion    - buckledcyl
doublehet      - rod    + squarepitch
doublehet      - rod    + triangpitch
doublehet      - rod    + asquarepitch
doublehet      - rod    + atriangpitch
doublehet      - pebble + sphsquarep
doublehet      - pebble + sphtriangp
doublehet      - pebble + asphsquarep
doublehet      - pebble + asphtriangp
doublehet      - slab   + symmslabcell
doublehet      - slab   + asymmslabcell
centrmdata
moredata

' TODO: define cel

end celldata
```

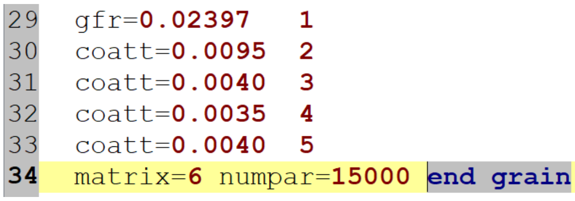
Figure 238.Autocomplete list of cells within the CELLDATA block.



```
read celldata
doublehet right_bdy=white fuelmix=10 end
  gfr=0.0    1
  coatt=0.0  2
  coatt=0.0  3
  coatt=0.0  4
  coatt=0.0  5
  matrix=6 numpar=15000 vf=0 end grain
  pebble sphsquarep right_bdy=white left_bdy=reflected
  hpitch=0.0 7 fuelr=0.0 cladr=0.0 8 fuelh=0.0 end

' TODO: define celldata

end celldata
```

Figure 239. Templated input provided for the DOUBLET PEBBLE + SPHSQUAREP cell in Fulcrum.

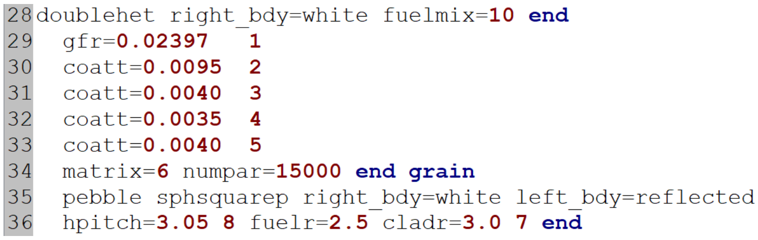
The first line declares the cell type as **DOUBLEHET**, as intended, and it specifies the right boundary condition of the grain cell as **white**. This white boundary is the desired boundary condition because it simulates a single grain in an infinite array of grains, which is a good approximation for the grains in the matrix. The **fuelmix** specified is the mixture number to be used for the homogenized cross sections for the fuel/matrix region; it must be a number that is not already specified in the COMPOSITION block, so **10** is an acceptable number to use for this model. The next section of input provides the fuel grain and coating dimensions and mixture numbers. The template input includes the radius of the fuel grain and the thicknesses of the coatings. Alternatively, the diameter can be entered for the fuel grain with the keyword **gfd**. The coatings can be specified with **coatr** or **coatd** to provide the monotonically increasing radii or diameters of the coatings; there is no practical limit on the number of coatings that can be used in a DOUBLEHET cell specification. The template keywords are convenient for the problem details provided in Section 8.5.3. The fuel grain radius (**gfr**) should be set to `0.02397`; the UO_2 fuel is mixture 1, so no change is necessary. Four coatings are included in the template input, and this model also includes 4 coatings. The relevant mixture numbers are also 2 through 5 for this model. Only the coating thicknesses provided in Table 3 need to be entered. The input is in centimeters, but the thicknesses are provided in µm, so the entries are `0.0095`, `0.0040`, `0.0035`, and `0.0040`, respectively. The next line specifies the matrix containing the coated grains, and mixture 6 (graphite) is the desired mixture for this model. The templated number of particles in each pebble (15000) is also applicable to this model. The **vf=0** input should be removed; the volume fraction can be specified in lieu of the number of particles. The end of the grain specification is declared with **end grain**; multiple different types of grains could be used in a single DOUBLEHET cell in CSAS. The completed grain section of the input is provided in Figure 240.



```
  gfr=0.02397   1
  coatt=0.0095  2
  coatt=0.0040  3
  coatt=0.0035  4
  coatt=0.0040  5
  matrix=6 numpar=15000 end grain
```

Figure 240. Grain and coating details for the DOUBLEHET unit cell for this model.

The last section of the input deals with the pebble level of the problem. The keyword **pebble** specifies that this DOUBLEHET cell is based on pebbles and not cylindrical compacts or slabs. The type of **pebble** cell is **sphsquarep** for spherical bodies in a square pitched array; the options for these geometries are essentially the same as for the MULTIREGION cell type. The **white** boundary condition is applied to approximate an infinite array of pebbles, and the left boundary is specified as **reflected** because it is the inner "surface" of the sphere. The half-pitch (**hpitch**) for this model is 3.05 cm, and mixture 8 (He) is the mixture for the region between the pebbles. The fueled region has a radius (**fuelr**) of 2.5 cm; no mixture number is specified here because the homogenized cross sections for the fuel/matrix region are calculated by the code and assigned to the mixture number provided earlier in the input for the **fuelmix** parameter. The **cladr** input provides the outer radius of the overall pebble and the mixture number for the material around the fuel/matrix fuel region; in this model, the radius is 3.0 cm, and the "cladding" is mixture 7. The **fuelh=0** input should be removed since it specifies the height of a cylindrical or slab compact. The final DOUBLEHET unit cell specification is provided in Figure 241; as before, extraneous comments and blank lines have been removed for simplicity.



```
doublehet right_bdy=white fuelmix=10 end
  gfr=0.02397   1
  coatt=0.0095  2
  coatt=0.0040  3
  coatt=0.0035  4
  coatt=0.0040  5
  matrix=6 numpar=15000 end grain
  pebble sphsquarep right_bdy=white left_bdy=reflected
  hpitch=3.05 8 fuelr=2.5 cladr=3.0 7 end
```

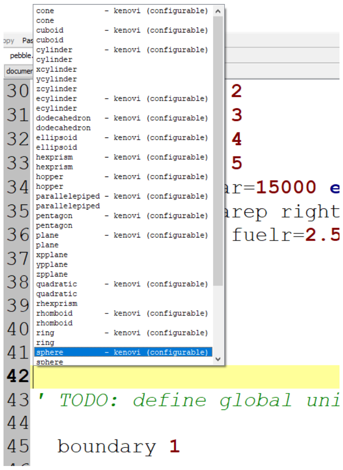
Figure 241. Complete DOUBLEHET unit cell specification for this model.

8.5.3.3 KENO-VI Geometry data

The KENO-VI data for this problem consists of a single unit containing a single pebble and the helium coolant around it. Reflective boundary conditions will be applied to all 6 faces of the cuboid to approximate an infinite array of pebbles. This problem differs from the previous two example problems because it contains a homogenized region for the fuel/matrix region inside the pebble. The homogenization in this case significantly accelerates tracking because it eliminates tracking through the 15,000 explicit grains.

The innermost region of the problem is the homogenized fuel/matrix region. This is a sphere with a radius of 2.5 cm, followed by graphite "cladding" and the cuboid with the helium coolant. The initial sphere can be added via a configurable form by placing the cursor in the global unit and selecting **sphere - kenovi (configurable)** as shown in Figure 242. Set the **Id** to 10 and the radius of this region is 2.5 cm; the completed configurable form is shown in Figure 243. After confirming that the input is correct, press the

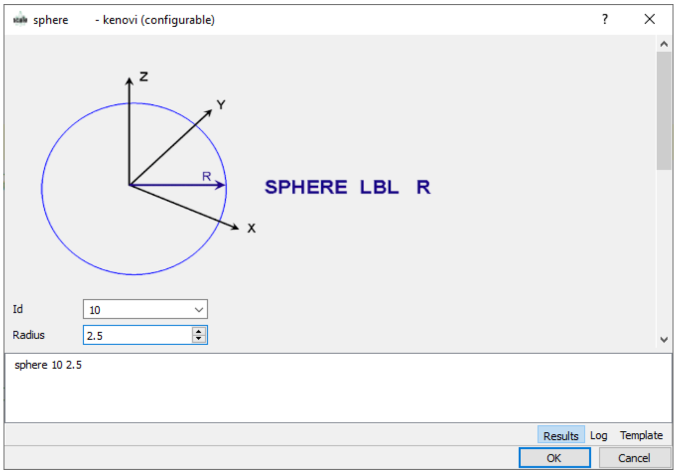
OK button to enter the sphere definition in the CSAS input. The next region is the graphite "cladding," or outer layer, on the pebble. Add this second region in the input by entering `sphere 20 3.0` or by using a sphere configurable form. The final region is a CUBOID filled with helium for the region between the pebbles; the dimension in ±X, ±Y, and ±Z is 3.05 cm. Add the cuboid by entering `cuboid 30 3.05 -3.05 3.05 -3.05 3.05 -3.05` or by using a cuboid configurable form. This completes the geometry records as shown in Figure 244.



opy
Pas
pebble.
documen
cone - kenovi (configurable)
cone
cuboid - kenovi (configurable)
cuboid
cylinder - kenovi (configurable)
cylinder
xcylinder
ycylinder
zcylinder
ecylinder - kenovi (configurable)
ecylinder
dodecahedron - kenovi (configurable)
dodecahedron
ellipsoid - kenovi (configurable)
ellipsoid
hexprism - kenovi (configurable)
hexprism
hopper - kenovi (configurable)
hopper
parallelepiped - kenovi (configurable)
parallelepiped
pentagon - kenovi (configurable)
pentagon
plane - kenovi (configurable)
plane
xpplane
ypplane
zpplane
quadratic - kenovi (configurable)
quadratic
rhexprism
rhomboid - kenovi (configurable)
rhomboid
ring - kenovi (configurable)
ring
sphere - kenovi (configurable)
sphere

```
2
3
4
5
ar=15000 e
arep right
fuelr=2.5


' TODO: define global uni

  boundary 1
```

Figure 242. Selecting the sphere configurable form in Fulcrum.

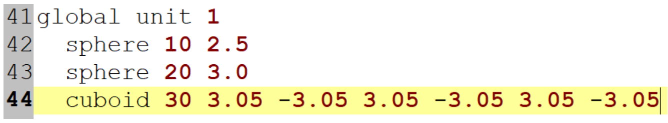


Id 10

Radius 2.5

```
sphere 10 2.5
```

Figure 243. Complete configurable form for the fuel/matrix region of the pebble.

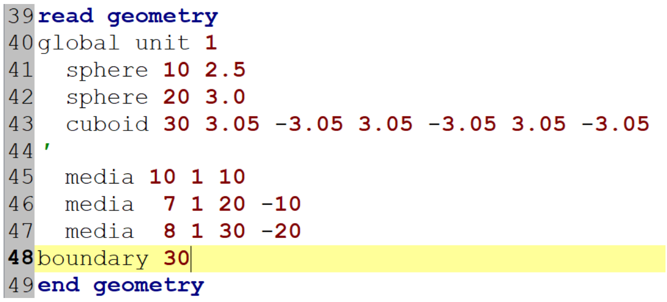


```
global unit 1
  sphere 10 2.5
  sphere 20 3.0
  cuboid 30 3.05 -3.05 3.05 -3.05 3.05 -3.05
```

Figure 244. All three geometry records for the pebble model.

The next step is to define the contents records, which in this case are all MEDIA records. The first media record fills the innermost sphere with the homogenized mixture of grains and matrix generated in the CELLDATA block and assigned to Mixture 10. The media entry can be directly entered as `media 10 1 10` or through the use of a configurable form. The next region is the graphite "cladding" on the pebble, which is mixture 7. Again, the media record can be entered directly as text in Fulcrum as `media 7 1`

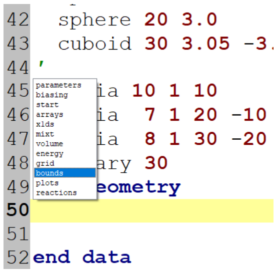
`20 -10` or with the configurable form. The media record specifies that graphite is to be placed inside the larger sphere but outside the smaller inner sphere. The final region is the helium coolant outside the pebble, which is mixture 8. The text input is media `8 1 30 -20`, or this can be entered using the **media - kenovi (configurable)** form. This input fills the volume inside the cuboid but outside the larger sphere with helium. The last entry in the GEOMETRY block is the **boundary** record. This must be updated to `boundary 30`, as everything within cuboid 30 is part of the model. The complete GEOMETRY block is shown in Figure 245.



```
read geometry
global unit 1
  sphere 10 2.5
  sphere 20 3.0
  cuboid 30 3.05 -3.05 3.05 -3.05 3.05 -3.05
'
  media 10 1 10
  media  7 1 20 -10
  media  8 1 30 -20
boundary 30
end geometry
```

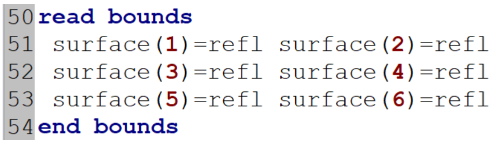
Figure 245. Complete geometry block for the pebble model.

The last step is to add reflective boundary conditions to all 6 faces in the problem; recall that the global unit boundary conditions are specified in the BOUNDS block. To add the BOUNDS block, position the cursory outside the other blocks and use the Fulcrum autocomplete function to add a new block, as shown in Figure 246. Autocomplete is not available in the BOUNDS block. The most efficient way to set all the faces to reflective boundaries is to use the cuboid face code from the first portion of the Face Codes and Surface Numbers table [see SCALE Manual Section *Albedo data*] and enter `all=refl`. The more explicit method is to specify the boundary condition on each face of the cuboid. This approach is shown in Figure 247. Both methods will set all 6 faces of the cuboid to reflective boundary conditions and generate equivalent models.



```
  sphere 20 3.0
  cuboid 30 3.05 -3.
'
          ia 10 1 10
          ia  7 1 20 -10
          ia  8 1 30 -20
          ary 30
          eometry


end data
```

Figure 246. Adding the BOUNDS block to the model in Fulcrum with autocomplete.



```
read bounds
 surface(1)=refl surface(2)=refl
 surface(3)=refl surface(4)=refl
 surface(5)=refl surface(6)=refl
end bounds
```

Figure 247. Complete BOUNDS block for the pebble model.

Now save all changes and run SCALE by clicking the **Run** button at the top of the Fulcrum window. The default run parameters of 1,000 particles per generation, 203 total generations, and 3 discarded generations will be used if no specific values are chosen by adding a PARAMETER block. These parameters yield a k_{eff} estimate of 1.5358 ± 0.0020; exact values will vary, as different computers and different operating systems generate different random numbers. A more rigorous calculation simulating

10,000 particles per generation for 197 total generations, skipping the first 50, resulted in a k_{eff} estimate of 1.53833 ± 0.00049. Homogenized cross sections cannot be generated for CE calculations, so the more difficult explicit model would need to be built to check this model. The explicit representation of the 15,000 grains brings additional challenges in terms of distribution and truncation; these issues are not discussed here, as they are beyond the scope of this primer.

8.6 SUMMARY

This section has demonstrated:

- the difference in cross section processing for Infinite Homogeneous Medium, Lattice Cell, Multiregion, and Double Heterogeneous analyses;
- identification of the appropriate unit cell type to use for various heterogeneous systems;
- using Fulcrum to define LATTICECELL, MULTIREGION, and DOUBLEHET unit cells;
- defining different boundary conditions for a MULTIREGION unit cell; and
- using boundary conditions to create reflective boundaries for the KENO-VI global unit.

9. PLOTTING AND VISUALIZATION

A strongly recommended practice for all KENO users is to visually check the geometry models prior to performing calculations. This section discusses the use of the 2D and 3D plotting and visualization tools available with KENO-VI and Fulcrum.

9.1 WHAT YOU WILL BE ABLE TO DO

- Generate 2D color plots of KENO-VI geometry models.
- View 2D source convergence diagnostics generated by KENO-VI.
- Generate 2D plots of nuclear data used in KENO-VI calculations.
- Generate and display mesh-based data in Fulcrum.
- Interactively view 3D wireframe and rendered images of KENO-VI geometry models using KENO3D.
- Plot calculated results overlain on the 3D geometry model using KMART6 and KENO3D.

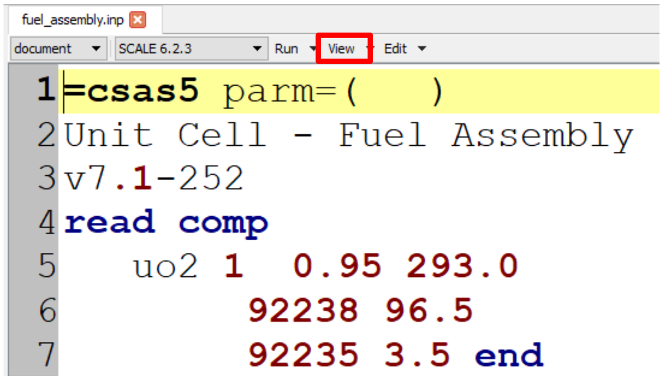
9.2 CREATING AND VIEWING 2D PLOTS WITH FULCRUM

The Fulcrum interface contains a robust interface for visualizing 2D data, including geometry plots, cross section data, scattering and covariance matrices, and generic plot data. This section introduces the plotting capabilities of Fulcrum; a more complete description can be found in Lefebvre's "Advanced User Interface Capabilities" [7].

9.2.1 Geometry

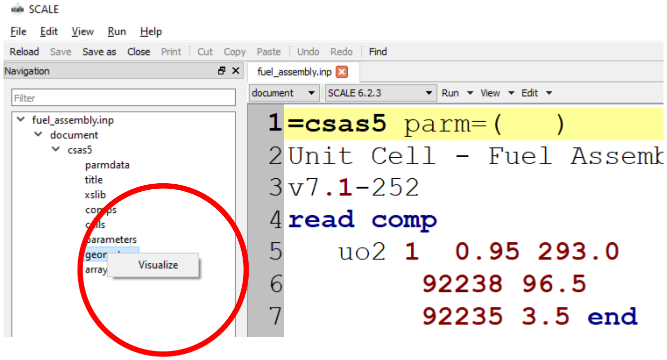
CSAS and KENO-VI have historically had a capability to produce plots of 2D KENO-VI geometry slices. These plots can be generated without performing a calculation by using the **PARM=CHECK** option on the first line of the CSAS/KENO-VI input. This option provides the user with the capability to check their geometry model prior to performing calculations [see SCALE Manual Section *Plot data*].

An easier approach to generating 2D plots normal to the X-, Y-, and Z-axes is to use the 2D plotting capability in Fulcrum. This capability is used in Sections 8.5.1.3 and 8.5.2.3 to display models as they are being built. The geometry viewer can be launched by clicking the **View** button near the top of the editor window, by pressing `Ctrl+Shift+V`, or by right clicking the **geometry** block name in the navigation panel and selecting **Visualization**. These options are shown in Figure 248 and Figure 249 using the fuel assembly model from Section 8.5.1. This model is used to demonstrate the features of the Fulcrum geometry plotting capabilities, but the features are applicable to all models. Note that Fulcrum will not launch visualization without a global unit in the model, and some parse and validation errors may also prevent visualization. Check the **Messages** pane in Fulcrum if a global unit exists and the geometry plotter does not launch.



```
=csas5 parm=(     )
Unit Cell - Fuel Assembly
v7.1-252
read comp
    uo2 1  0.95 293.0
          92238 96.5
          92235 3.5 end
```

Figure 248. Location of View button in Fulcrum.



```
=csas5 parm=(     )
Unit Cell - Fuel Assemb
v7.1-252
read comp
    uo2 1  0.95 293.0
          92238 96.5
          92235 3.5 end
```

Figure 249. Navigation panel for visualizing a model.

The default initial slice presented in the plotter is an XY plot through the midplane of the model. The default view is shown in Figure 250; note that the plotter is launched in a separate tab within Fulcrum. The orientation of the plot view can be selected from the list of buttons just above the view window to the left edge of the tab. The three choices are **Top (X-Y)**, **Front (X-Z)**, and **Side (Y-Z)**, as shown in Figure 251.

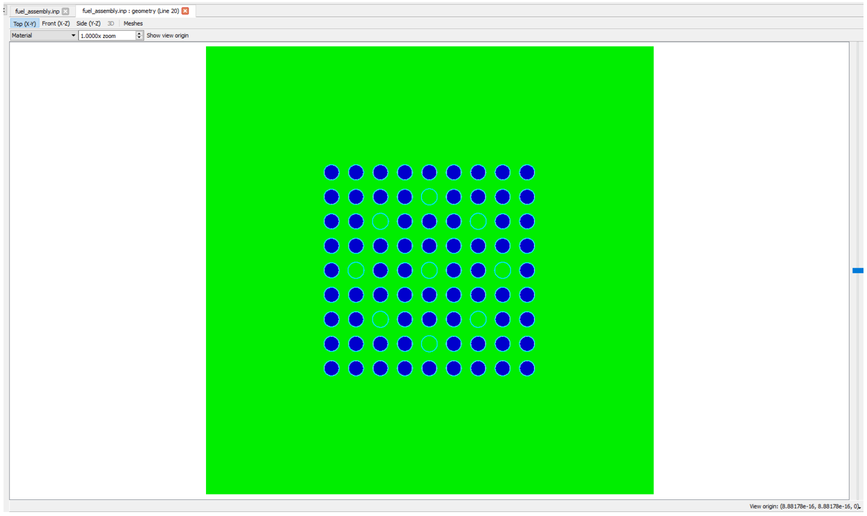


Figure 250. Default plot view of the fuel assembly model in the Fulcrum plotter.

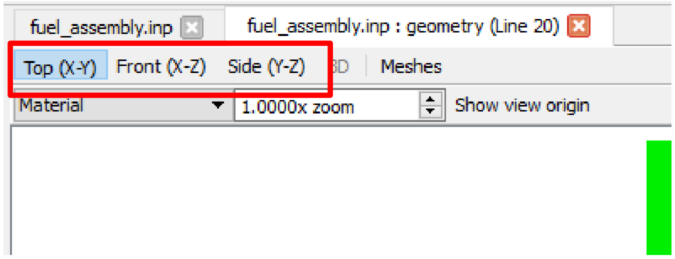
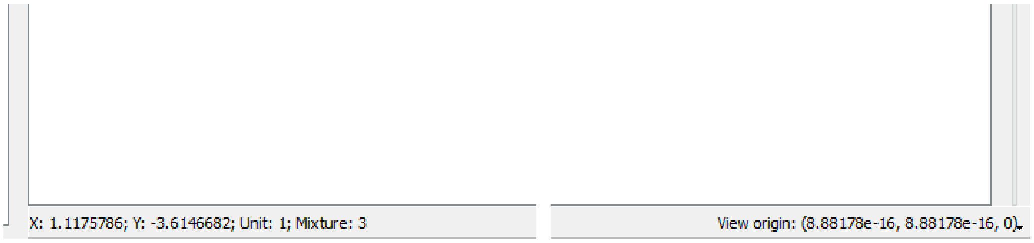


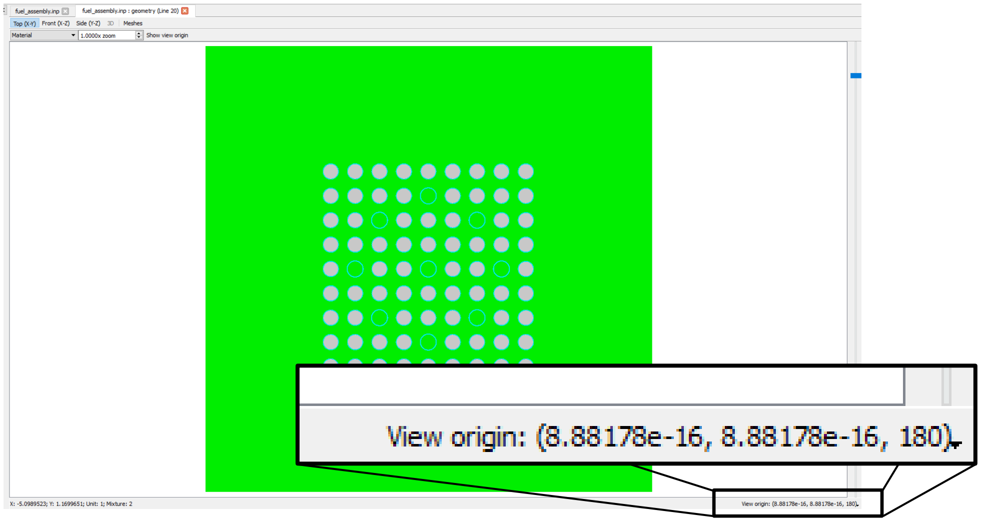
Figure 251. Selection of visualization plane orientation.

The bottom frame of the visualization window displays the current mouse pointer location on the bottom left, and the origin of the current view is shown on the bottom right, as presented in Figure 252. The view origin can be set by clicking on the **View origin** field in the lower right corner of the visualization frame. Entering one number will set the location of the visualization plane along the axis perpendicular to it. For example, click on the **View origin** field and enter `180`. For the fuel assembly model, this creates an XY plane through the upper plenum of the fuel rods, as shown in Figure 253; note that the **View origin** has a Z coordinate of **180**. Entering two numbers in the View origin field re-centers the view in the current plane at the specified coordinates. For example, click on the **View origin** field and enter `-6.56 -6.56`. For the fuel assembly model, this centers the view on the center of the fuel rod in the lower left corner of the assembly; click the **Show view origin** button on the top of the visualization frame (Figure 254), and a crosshair will appear displaying the exact center of the view. Note that the **View origin** display shows **-6.56 -6.56 180**, as shown in Figure 255. Entering three numbers in the **View origin** field centers the visualization plane at the coordinates entered in X, Y, and Z as expected. Finally, the slider on the right side of the frame (blue rectangle on Windows and white circle on Mac), shown in Figure 256, allows control of the visualization plane along the axis to which it is perpendicular. For example, sliding up and down the frame controls the Z elevation of an XY visualization plane.


X: 1.1175786; Y: -3.6146682; Unit: 1; Mixture: 3

View origin: (8.88178e-16, 8.88178e-16, 0)

Figure 252. Mouse pointer location and view origin displays.



View origin: (8.88178e-16, 8.88178e-16, 180)

X: -5.0989523; Y: 1.1699651; Unit: 1; Mixture: 2

Figure 253. XY visualization at Z=180 cm plane.

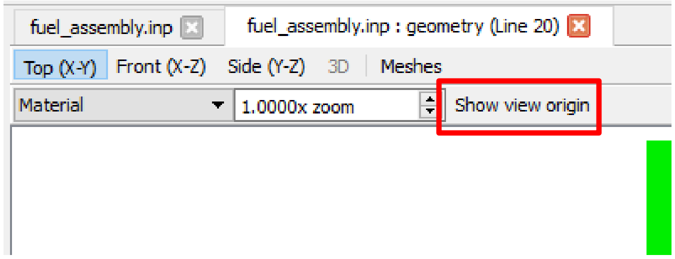
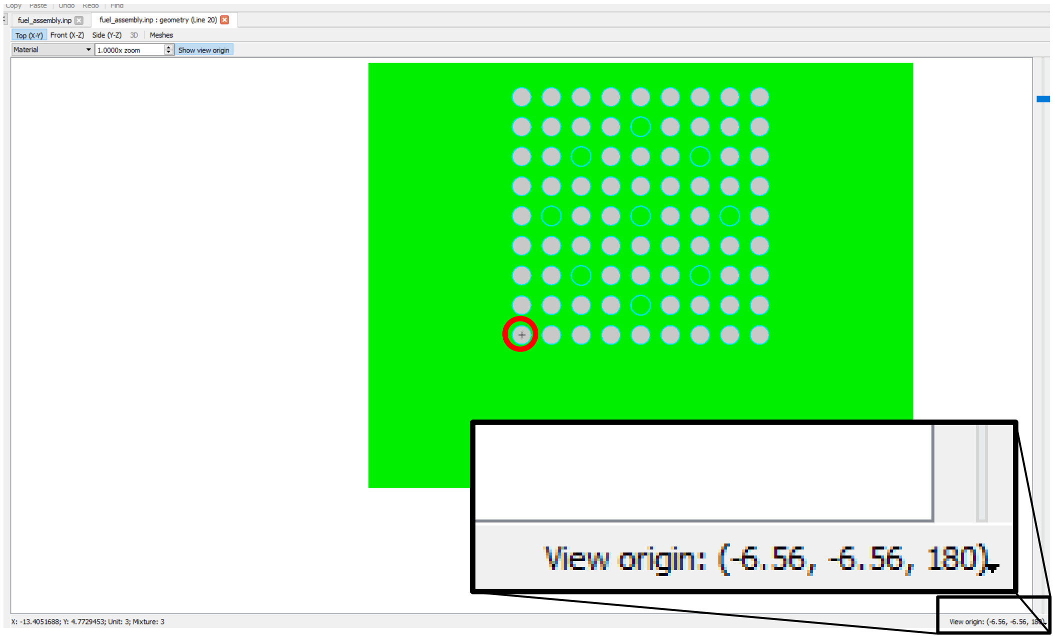


Figure 254. Location of Show view origin button.



View origin: (-6.56, -6.56, 180)

X: -13.4051688; Y: 4.7729453; Unit: 3; Mixture: 3

Figure 255. XY visualization centered at (-6.56, -6.56) at Z=180 cm plane.

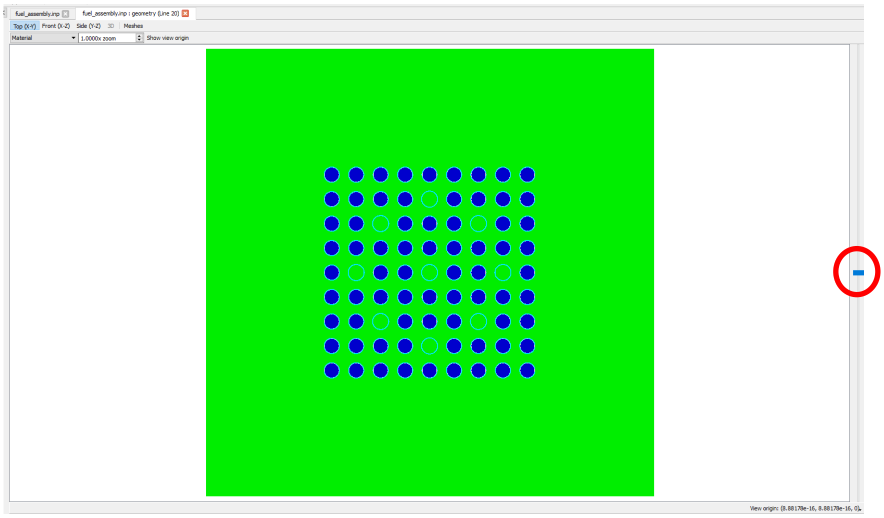


Figure 256. Location of slider to control position of the visualization plane.

Two key visualization controls are on the top of the visualization frame and control the render mode and zoom level of the current view. These controls are shown in Figure 257. The render mode dropdown menu is shown in Figure 258; the first three options are discussed here, and **Overlay** and **Overlay + boundaries** are discussed in Section 9.3. The default render mode is **Material**, and it colors the view according to the mixture number (material) in each region. No region boundaries are shown between regions of the same material unless the **Material + outline** render mode is selected. An example of this is shown in Figure 259; note that the unit boundaries are now shown in the array, as well as the outer boundary of the array. Region boundaries for the fuel pellet, gap, and cladding are also shown. The third option is for **Outline** only and displays the region boundaries in the color of the mixture in that region. An example is provided in Figure 260.

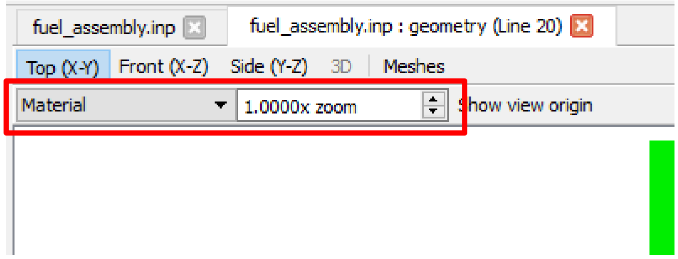


Figure 257. Display controls for render mode and zoom.

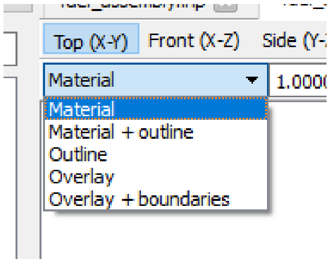


Figure 258. Render mode dropdown menu.

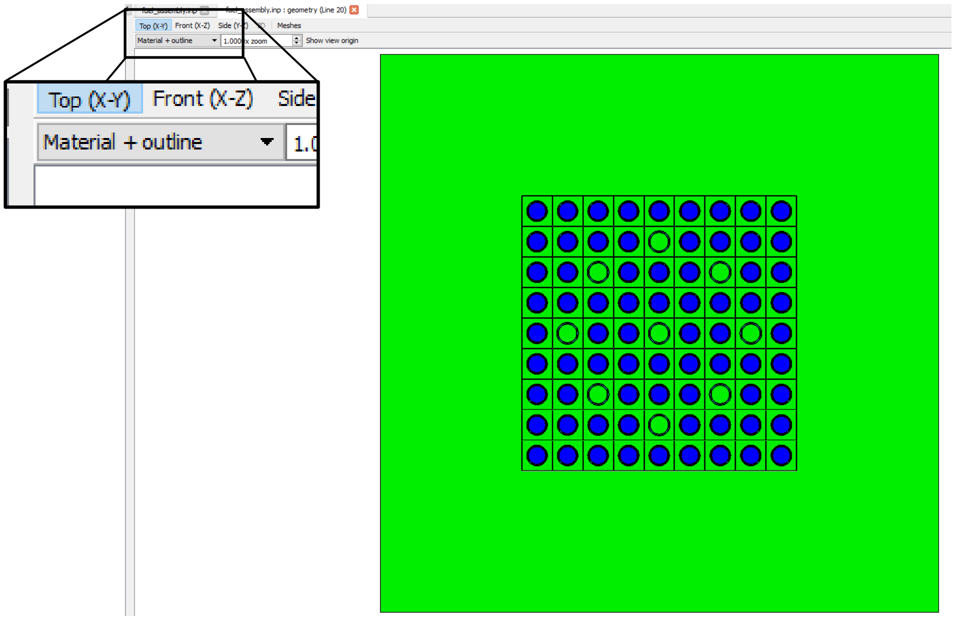


Figure 259. Midplane view of fuel assembly in Material + outline render mode.

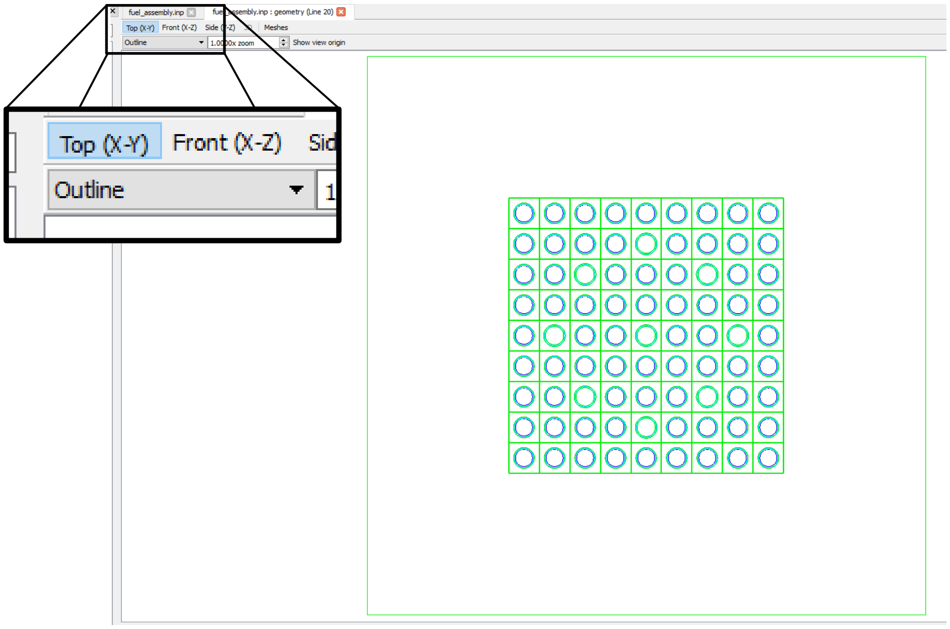
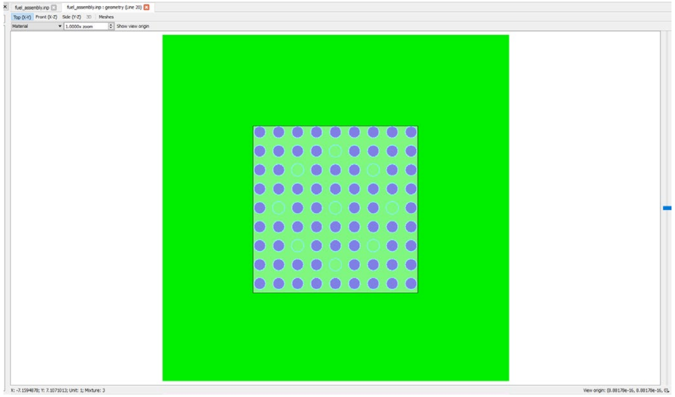


Figure 260. Midplane view of fuel assembly in Outline render mode.

There are two primary ways to control the zoom level of the image. The first is the zoom field at the top of the frame, as shown in Figure 257 above. A zoom factor can be typed directly into the field, or the arrows can be used to increase or decrease the zoom factor. The second primary zoom control is to left click and drag from the upper left corner to the lower right corner of a box. Fulcrum displays the box as it is created, as shown in Figure 261, and then attempts to fit this box into the view window as shown in Figure 262. Typically, the vertical distance is fit to the view window, so the zoomed image will likely be wider than the zoom box. To reset to the default zoom level, click and drag from the lower right to the upper left corner of a box. Fulcrum does not display the box because the image is being returned to the default view of the current visualization plane.



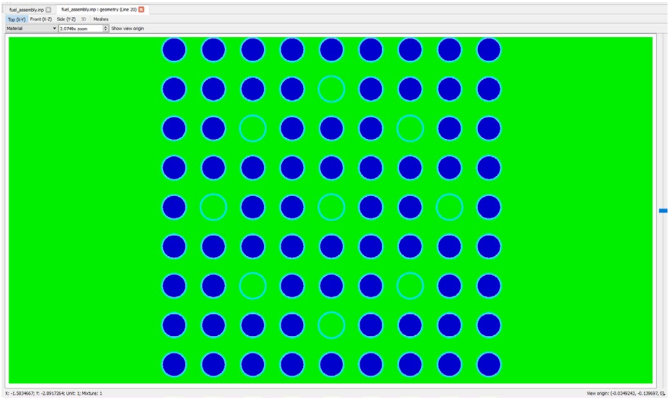
Material

1.0000x zoom

X: -7.1594878; Y: 7.1071013; Unit: 1; Mixture: 3

View origin: (8.88178e-16, 8.88178e-16, 0)

Figure 261. Fulcrum visualization while creating zoom control box.



Material

2.0749x zoom

X: -1.5834667; Y: -2.8917264; Unit: 1; Mixture: 1

View origin: (-0.0349243, -0.139697, 0)

Figure 262. Image results from the zoom shown in Figure 260.

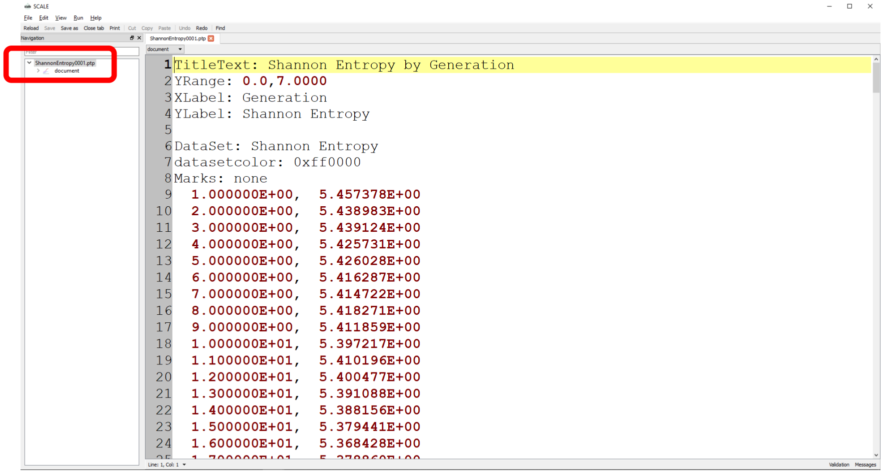
The last visualization control to discuss here is panning; there are two methods that can be used to pan the image. One is to hold the Shift key, left click on the image, and drag it to a new location. The image is translated as far as the mouse moved and is then redrawn; the image does not move as it is being translated. The second option is to double left click at a point on the image that will become the new view origin. The image is then redrawn with the selected point as the view origin.

9.2.2 KENO Convergence Summary Plots

KENO convergence summary plots are generated with the HTML output and can be opened directly in Fulcrum. A series of *ptp files is created in the htmd directory when execution is complete. Three different categories of *ptp files exist: “errorbars,” “freqgen,” and “ShannonEntropy.” Each of these types of plots is discussed briefly below.

The most obvious plot type is ShannonEntropy. This file contains the Shannon entropy calculated for each generation in a format that can be plotted in Fulcrum. After the calculation has been executed, click on **File > Open file** and navigate into the htmd directory associated with the input, in this case “**fuel_assembly.htmd**.” Double click on **ShannonEntropy00001.ptp**, and Fulcrum opens the file in the text editor, as shown in Figure 263. As highlighted in the figure, a small plot icon appears next to the word **document** in the Navigation panel. This icon indicates an entry in the Navigation panel that can be double clicked to generate a plot; double click on the word **document** to plot the Shannon entropy data for this problem. The resulting figure is shown in Figure 264. It may be difficult to assess convergence on this scale. To reset the Y-axis scale, right click on the figure and select **Plot options**. Adjust the window so that all the options are visible, as shown in Figure 265, and select **Axes** from the pane on the left.

Scroll down to **Y-axis** and **Range Min** and **Range Max**; set the **Range Min** to 4 and the **Range Max** to 6. The updated plot is shown in Figure 266, indicating that the source may converge by approximately generation 50–75.



```
TitleText: Shannon Entropy by Generation
YRange: 0.0,7.0000
XLabel: Generation
YLabel: Shannon Entropy

DataSet: Shannon Entropy
datasetcolor: 0xff0000
Marks: none
  1.000000E+00,  5.457378E+00
  2.000000E+00,  5.438983E+00
  3.000000E+00,  5.439124E+00
  4.000000E+00,  5.425731E+00
  5.000000E+00,  5.426028E+00
  6.000000E+00,  5.416287E+00
  7.000000E+00,  5.414722E+00
  8.000000E+00,  5.418271E+00
  9.000000E+00,  5.411859E+00
  1.000000E+01,  5.397217E+00
  1.100000E+01,  5.410196E+00
  1.200000E+01,  5.400477E+00
  1.300000E+01,  5.391088E+00
  1.400000E+01,  5.388156E+00
  1.500000E+01,  5.379441E+00
  1.600000E+01,  5.368428E+00
```

Figure 263. Fulcrum default view of a *ptp file.

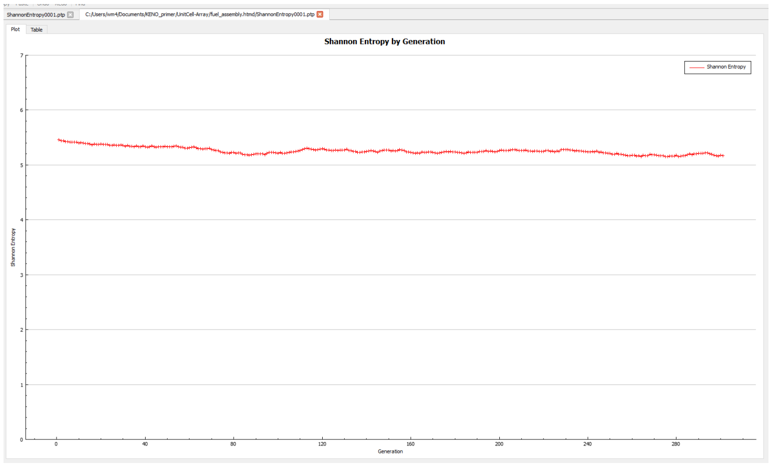
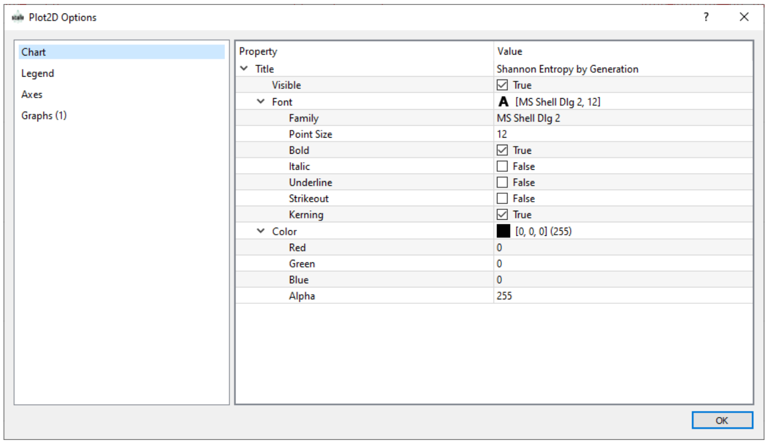


Figure 264. Default view of Shannon entropy by generation plot.



Property	Value
Title	Shannon Entropy by Generation
Visible	☑ True
Font	A [MS Shell Dlg 2, 12]
Family	MS Shell Dlg 2
Point Size	12
Bold	☑ True
Italic	□ False
Underline	□ False
Strikeout	□ False
Kerning	☑ True
Color	■ [0, 0, 0] (255)
Red	0
Green	0
Blue	0
Alpha	255

Figure 265. Resized plot options window.

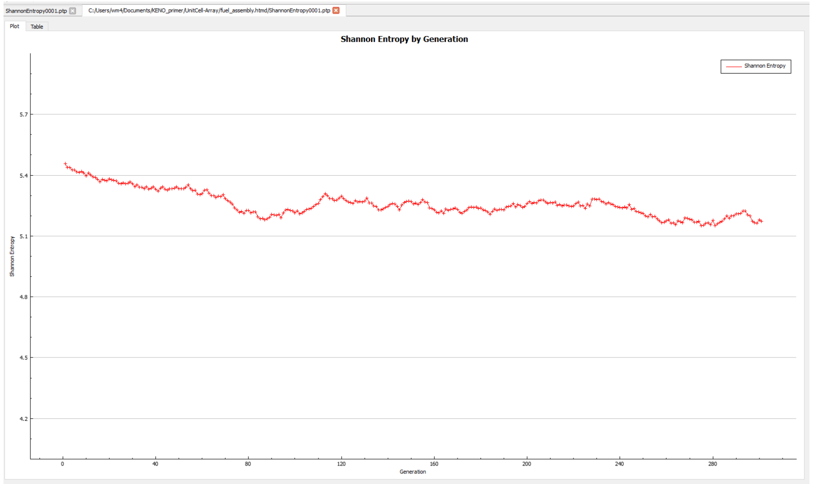


Figure 266. Rescaled view of the Shannon entropy plot.

The next plots to discuss are those labeled "errorbars0001.ptp" and "errorbars20001.ptp." These plots provide the k_{eff} convergence summary based on the number of generations run (0001) and the number of generations skipped (20001). To view a plot, click on **File** > **Open file** (or press `Ctrl+o`) and select errorbars0001.ptp. Fulcrum will by default attempt to open a file from the same directory used previously. As before, the document is opened in the editor. To plot the data, double-click on **document**. As shown in Figure 267, the average k_{eff} is plotted as a circle, and the uncertainty bars correspond to a number of different confidence intervals. Note that the plot is showing the average k_{eff} value for each number of generations run, not the individual generation k_{eff} values. The k_{eff} convergence summary for this problem indicates acceptable convergence for this simulation. Similarly, the second plot (errorbars20001.ptp) is shown in Figure 268. This provides the average k_{eff} as a function of generation skipped, and it also shows excellent k_{eff} convergence behavior.

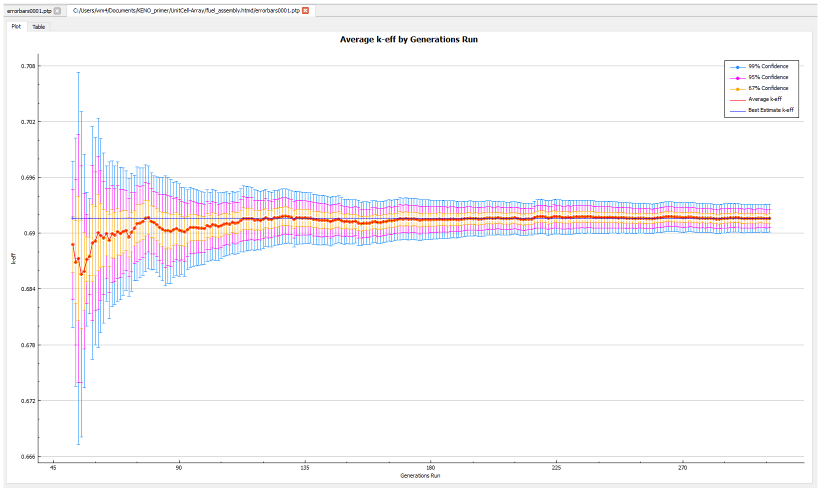


Figure 267. Plot of average k_{eff} by generation run.

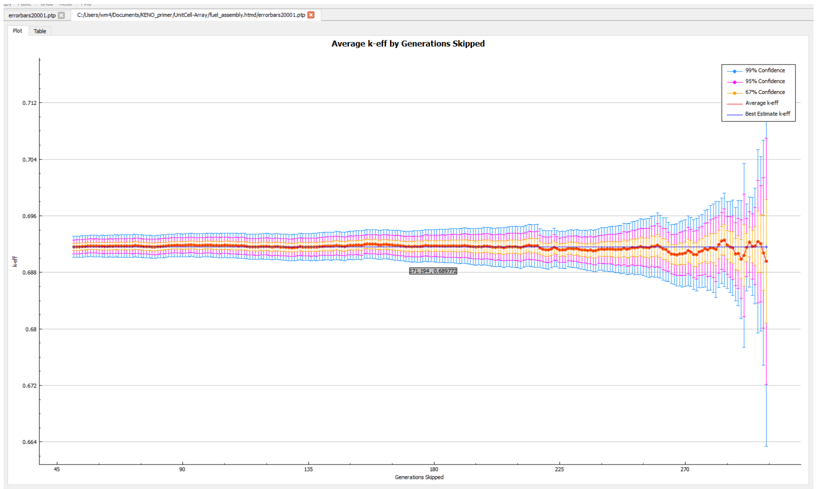


Figure 268. Plot of average k_{eff} by generation skipped.

The final set of plots is the "freqgen" series; these plots are histograms binning generation k_{eff} values. If source convergence is achieved prior to the start of active generations, then the distribution of generation k_{eff} values should follow a normal distribution, as the only variations are random. Separate histograms are generated over all of the active generations, over the last three quarters of active generations, the last half of active generations, and the last quarter of active generations. The 5 plots available are labeled 0001, 10001, 20001, 30001, and 40001. The summary plot with all 4 histograms combined is on "freqgen0001.ptp" and is shown in Figure 269. The histogram for the last quarter of active generations (40001) is shown in Figure 270. It appears that convergence was achieved during the discarded generations because no large shifts are evident among the histograms in Figure 269. Each histogram can be reviewed separately if desired.

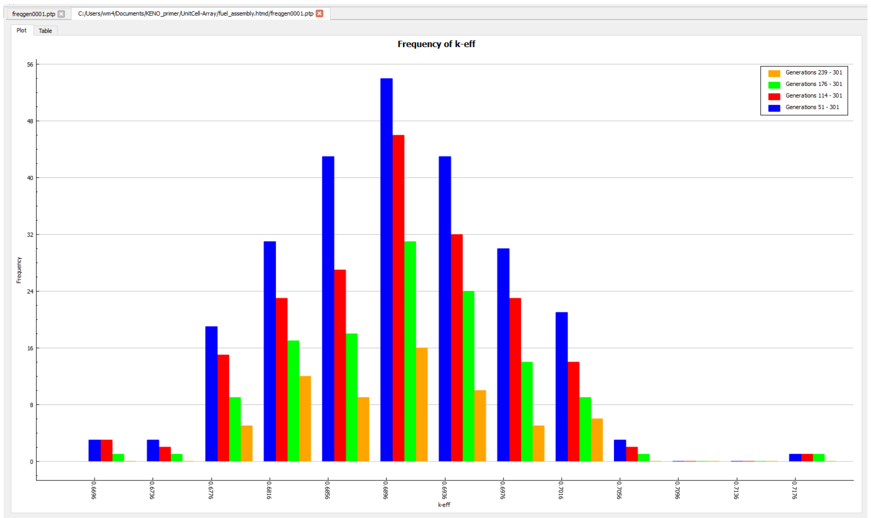


Figure 269. Histograms of generation k_{eff} values.

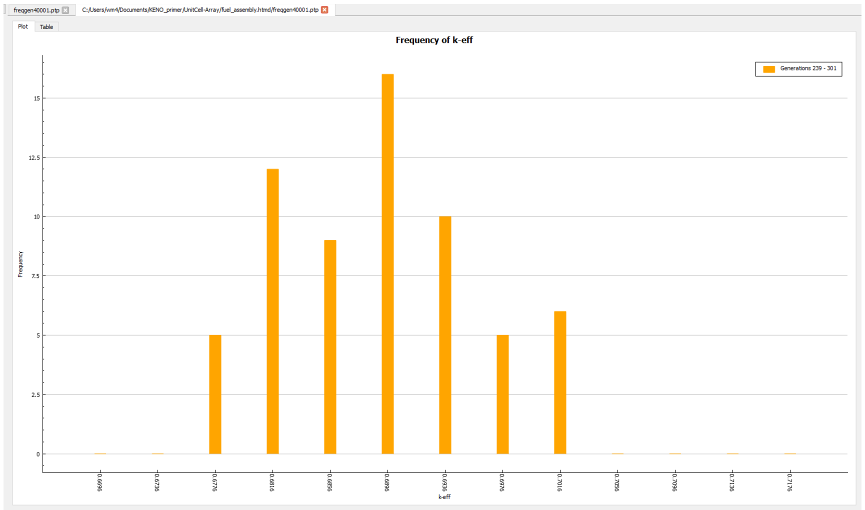


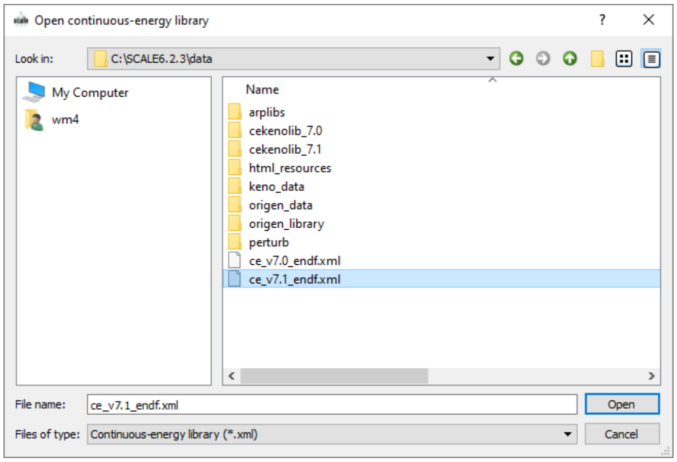
Figure 270. Histogram of generation k_{eff} values for the last quarter of active generations.

9.2.3 Cross Sections and Other Nuclear Data in Fulcrum

Three types of data plotting will be discussed in this section for various 1D data plots within Fulcrum: CE cross section data, MG cross section data, and MG uncertainty data. All of these data are 1D in that they are plotted against energy. Advanced data visualizations for 2D data, scattering, covariance, and correlation matrices are available within Fulcrum but are not covered in this primer. See Lefebvre's "Advanced User Interface Capabilities" [7] for more information on these advanced options.

There are no enforced naming conventions for nuclear data libraries in SCALE, so there are no defined extensions for the different types of libraries. Therefore, each type of library must be opened using the appropriate option from the Fulcrum **File** menu. Fulcrum issues an error that it is unable to open a library with the requested format if a mismatch occurs, such as when attempting to open a MG library after selecting **File** > **Open continuous-energy library**.

Select **File** > **Open continuous-energy library** to open a CE library. Fulcrum will open a files navigation window in the default data installation location; if the data were installed to another path, then navigate to that directory. Open **ce_v7.1_endf.xml**, as shown Figure 271; this file contains the path information to the actual CE cross section files contained in the **cekenolib** directory. Fulcrum will follow these data paths to plot the data as directed, but first the available isotopes and temperatures will be displayed in the Navigation panel, as shown in Figure 272. Next, expand the **Neutron** data and scroll down to expand u-238. Note that the isotopes available are listed alphabetically and that available temperatures in Kelvin are displayed after expansion of the **u-238** entry, as shown in Figure 273. Next, double-click on the **293** entry to generate the list of reaction cross sections available at that temperature, as shown in Figure 274. Scroll down and expand the **u-238 mt=102 n,gamma** entry, and then double-click on **u-238 mt=102 n,gamma 293 K xs** to plot the (n,γ) cross section for ^{238}U. An example of the plot is shown in Figure 275; note that, as is typical for cross section plots, the data are displayed on a log-log scale.

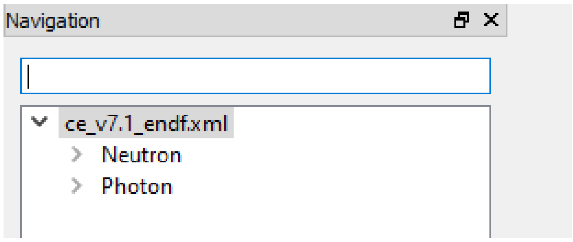


Look in: C:\SCALE6.2.3\data

File name: ce_v7.1_endf.xml

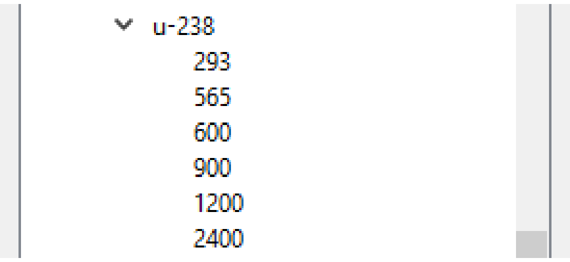
Files of type: Continuous-energy library (*.xml)

Figure 271. Opening the CE xml file.



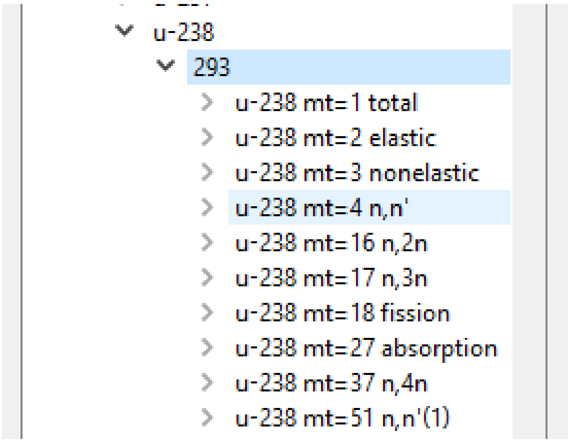
ce_v7.1_endf.xml
- Neutron
- Photon

Figure 272. Default Navigation panel after opening a CE library.



u-238
- 293
- 565
- 600
- 900
- 1200
- 2400

Figure 273. Expanded ^{238}U entry showing temperatures at which data are available.



u-238
- 293
  - u-238 mt=1 total
  - u-238 mt=2 elastic
  - u-238 mt=3 nonelastic
  - u-238 mt=4 n,n'
  - u-238 mt=16 n,2n
  - u-238 mt=17 n,3n
  - u-238 mt=18 fission
  - u-238 mt=27 absorption
  - u-238 mt=37 n,4n
  - u-238 mt=51 n,n'(1)

Figure 274. Top portion of the list of reactions at 293 K available for plotting.

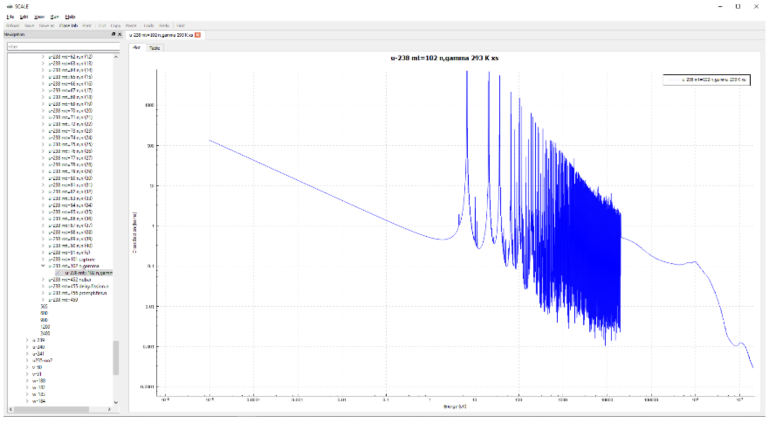
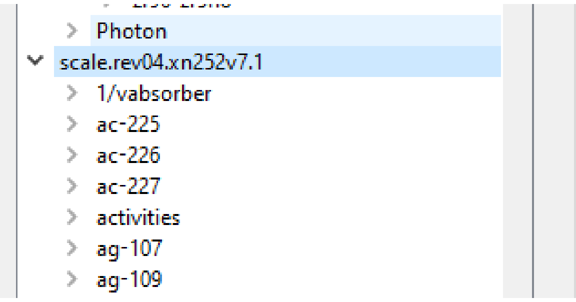


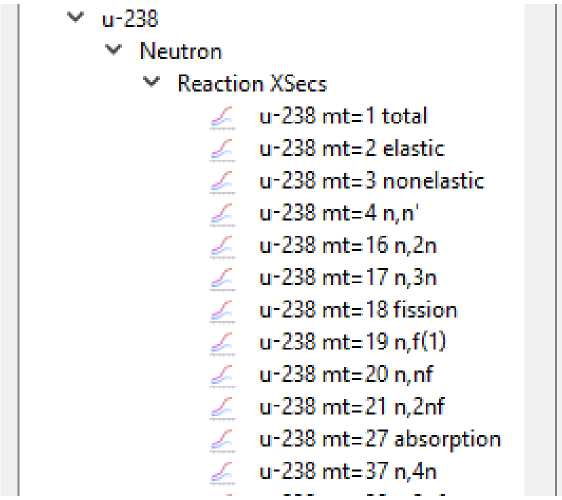
Figure 275. Default view of ^{238}U (n,γ) CE cross section in Fulcrum.

Fulcrum can display MG and CE cross sections on the same plot because the unit (barns) is the same. Select **File** > **Open multigroup library** and select **scale.rev04.xn252v7.1** for the 252-group library based on ENDF/B-VII.1. This library opens expanded to the isotopes with neutron cross sections in the library, as shown in Figure 276. The libraries are listed alphabetically in the Navigation panel, so in this case, the MG library is listed below the CE library. Scroll down to expand **u-238**, **Neutron**, and **Reaction XSecs**, as shown in Figure 277. Scroll down to the **u-238 mt=102 n,gamma** entry, right click on it, and select **Add graph to "u-238 mt=102 n,gamma 293 K xs"**, as shown in Figure 278. The resulting plot comparing the MG and CE cross sections is shown in Figure 279.



- Photon
- scale.rev04.xn252v7.1
  - 1/vabsorber
  - ac-225
  - ac-226
  - ac-227
  - activities
  - ag-107
  - ag-109

Figure 276. MG library loaded into the Navigation panel below the CE library.



- u-238
  - Neutron
    - Reaction XSecs
      - u-238 mt=1 total
      - u-238 mt=2 elastic
      - u-238 mt=3 nonelastic
      - u-238 mt=4 n,n'
      - u-238 mt=16 n,2n
      - u-238 mt=17 n,3n
      - u-238 mt=18 fission
      - u-238 mt=19 n,f(1)
      - u-238 mt=20 n,nf
      - u-238 mt=21 n,2nf
      - u-238 mt=27 absorption
      - u-238 mt=37 n,4n

Figure 277. MG library in the Navigation panel with ^{238}U expanded.

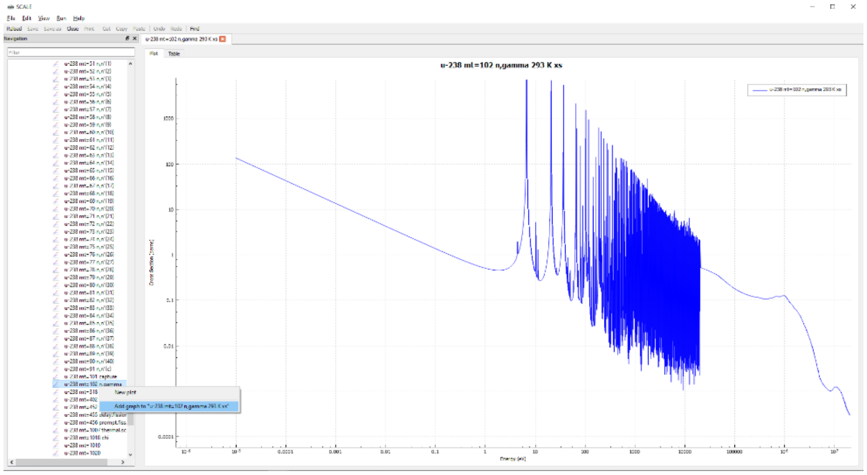

SCALE
File Edit View Run Help
Navigation
u-238 mt=102 n,gamma 293 K xs
Plot
Table
u-238 mt=102 n,gamma 293 K xs
New plot
Add graph to "u-238 mt=102 n,gamma 293 K xs"
Cross Section (barns)
Energy (eV)

Figure 278. Adding MG ^{238}U (n,γ) cross section to the existing CE plot.

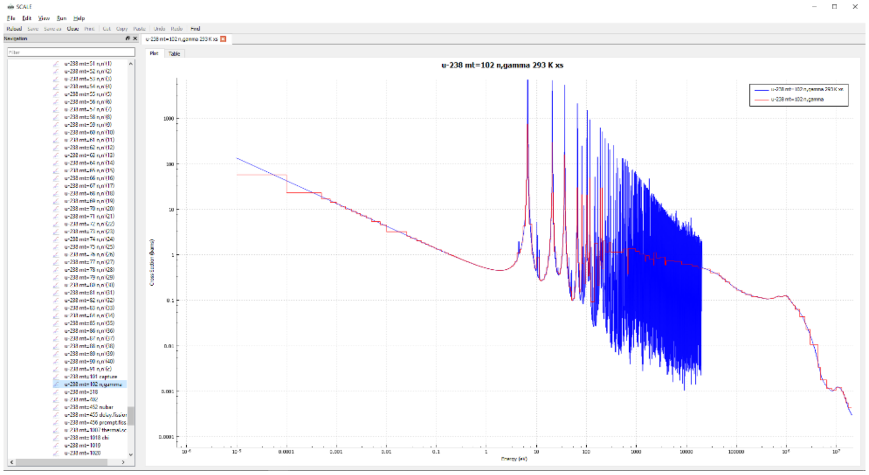


Figure 279. MG and CE cross sections for the ^{238}U (n,γ) reaction.

The last type of data visualization covered here is covariance data. As mentioned previously, only plotting of the cross section uncertainty (the square root of the diagonal elements of the covariance matrix) will be discussed here. The uncertainty data cannot be plotted on the same figure as cross section data because the unit used for the uncertainty data (relative uncertainty [%]) is not the same as that for the cross section data (barns). To load the covariance data, select **File** > **Open covariance library** and select **scale.rev08.56groupcov7.1** for the 56-group covariance library based on ENDF/B-VII.1 and other sources. This is the default covariance library for use in TSUNAMI in the SCALE 6.2 releases. The covariance library loads in the Navigation panel below the MG library because of its higher revision number; see Figure 280. Scroll down to **u-238**, expand the entry, and expand **mt=102 n,gamma**. Double click **u-238 mt=102 n,gamma – Std dev by energy** to create a new plot with the relative standard deviation for this reaction as a function of energy, as shown in Figure 281. Just above the plot there are tabs labeled **Plot** and **Table**; on windows the tabs are on the left side above the plot and on Mac they are in the center of the top ribbon. Clicking the **Table** tab provides a table of the points and data used in generating the plot, as shown in Figure 282. This plot of the standard deviation of the cross section is the square root of the diagonal of the covariance matrix, but it is still quite useful for showing the evaluated uncertainty in the cross section. Left click on the **Plot** tab to return to the plot. Left click on the legend in the upper right-hand portion of the plot and drag it to the left so that it does not overlap with the plot, as shown in Figure 283. Note that the legend in data plots in Fulcrum can be positioned in one of 9 locations: along the left, center, or right and at the top, middle, or bottom of the view window. Finally, left click on the tab for the covariance plot labeled **u-238 mt = 102 n,gamma – Std dev by energy**, and drag it down to create a top/bottom split view showing both the cross section and covariance plots. Figure 284 shows the screen while the plot is being repositioned, and Figure 285 shows the resulting view of both plots. In this configuration, the cross section and its uncertainty can be compared directly.

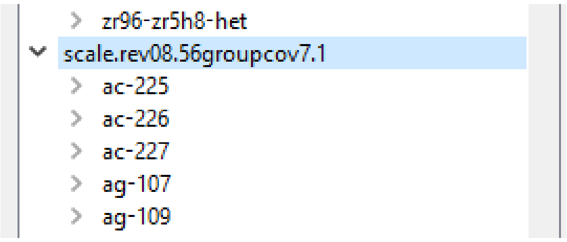


Figure 280. Covariance library loaded into the Navigation panel below the MG library.

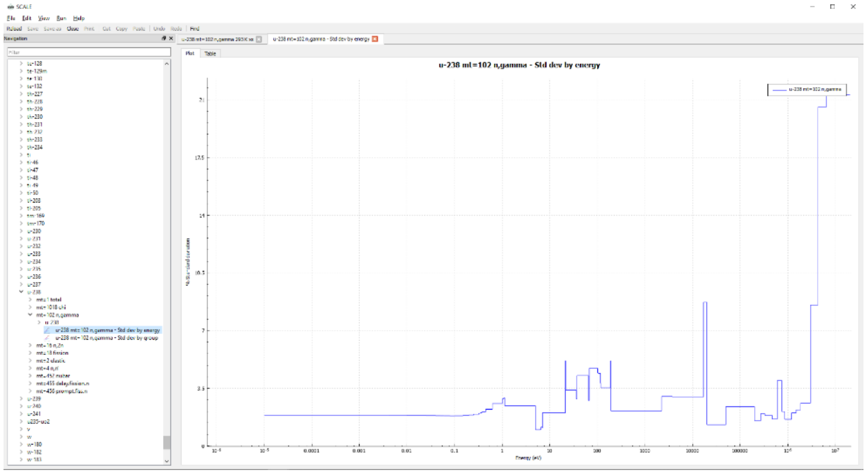
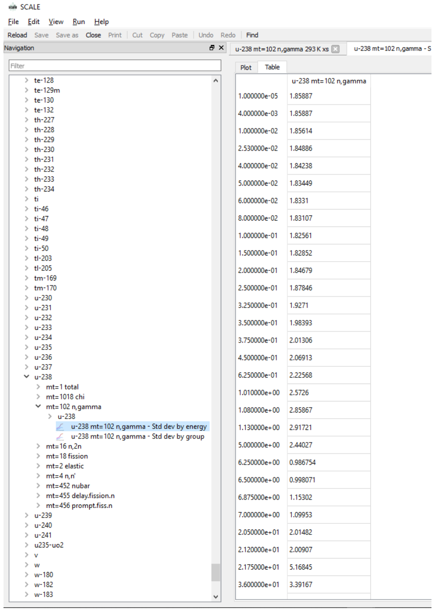

SCALE
File Edit View Run Help
Navigation
u-238 mt=102 n,gamma - Std dev by energy
Plot
Table
u-238 mt=102 n,gamma - Std dev by energy
u-238 mt=102 n,gamma
u-238 mt=102 n,gamma - Std dev by group
Energy (eV)

Figure 281. Plot of standard deviation of ^{238}U (n,γ) cross section.



	u-238 mt=102 n,gamma
1.000000e-05	1.85887
4.000000e-03	1.85887
1.000000e-02	1.85614
2.530000e-02	1.84886
4.000000e-02	1.84238
5.000000e-02	1.83449
6.000000e-02	1.8331
8.000000e-02	1.83107
1.000000e-01	1.82561
1.500000e-01	1.82852
2.000000e-01	1.84679
2.500000e-01	1.87846
3.250000e-01	1.9271
3.500000e-01	1.98393
3.750000e-01	2.01306
4.500000e-01	2.06913
6.250000e-01	2.22568
1.010000e+00	2.5726
1.080000e+00	2.85867
1.130000e+00	2.91721
5.000000e+00	2.44027
6.250000e+00	0.986754
6.500000e+00	0.998071
6.875000e+00	1.15302
7.000000e+00	1.09953
2.050000e+01	2.01482
2.120000e+01	2.00907
2.175000e+01	5.16845
3.600000e+01	3.39167

Figure 282. Table of values for standard deviation of ^{238}U (n,γ) cross section.

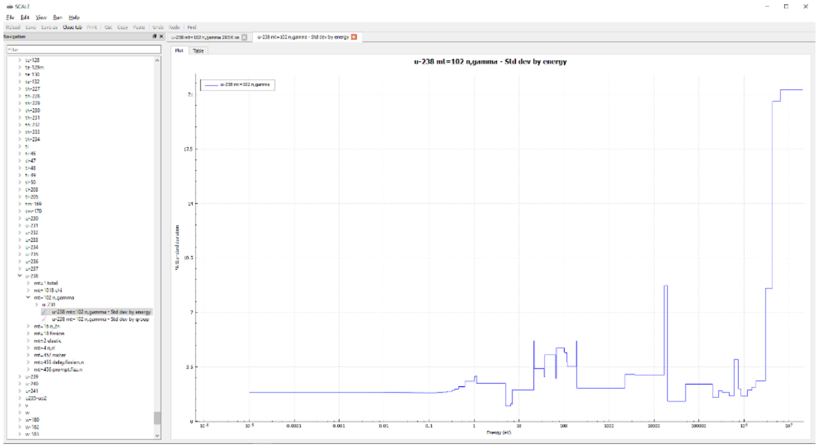

SCALE
File Edit View Run Help
Navigation
u-238 mt=102 n,gamma 293 K xs
u-238 mt=102 n,gamma - Std dev by energy
Plot Table
u-238 mt=102 n,gamma - Std dev by energy
u-238 mt=102 n,gamma
Energy (eV)

Figure 283. Uncertainty plot with repositioned legend.

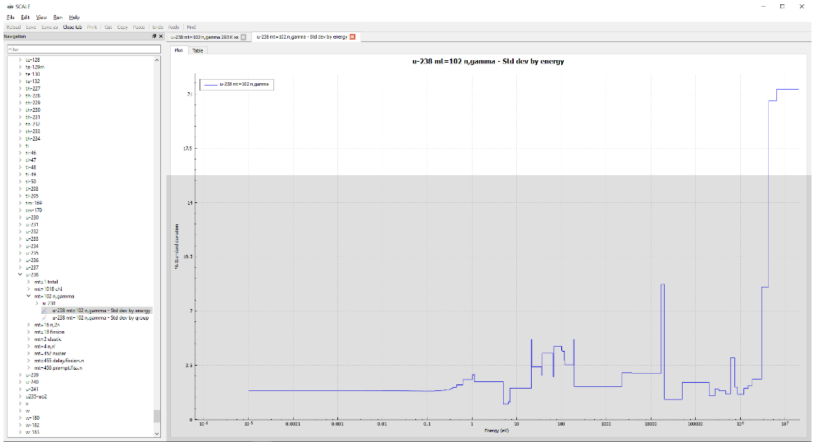

SCALE
File Edit View Run Help
Navigation
u-238 mt=102 n,gamma 293 K xs
u-238 mt=102 n,gamma - Std dev by energy
Plot Table
u-238 mt=102 n,gamma - Std dev by energy
u-238 mt=102 n,gamma
Energy (eV)

Figure 284. Positioning the uncertainty plot in the upper half of the view window.

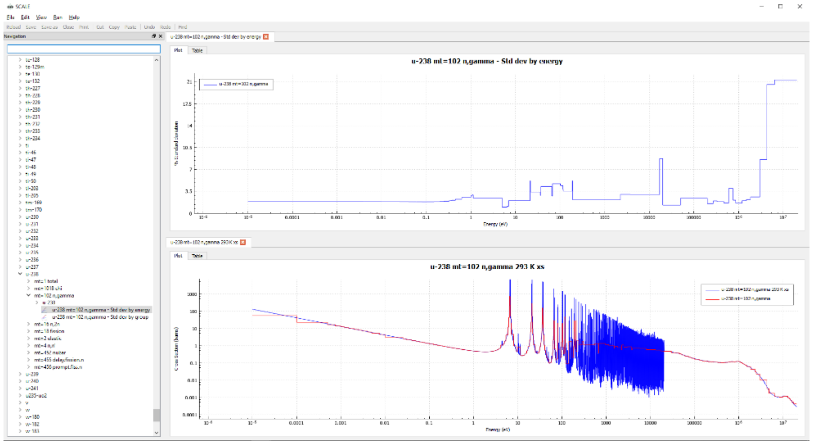

SCALE
File Edit View Run Help
Navigation
u-238 mt=102 n,gamma - Std dev by energy
Plot Table
u-238 mt=102 n,gamma - Std dev by energy
u-238 mt=102 n,gamma
Energy (eV)
u-238 mt=102 n,gamma 293 K xs
u-238 mt=102 n,gamma 293 K xs
u-238 mt=102 n,gamma 293 K xs
u-238 mt=102 n,gamma
Energy (eV)

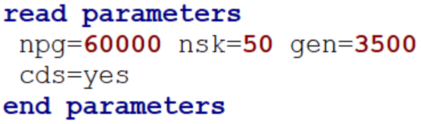
Figure 285. ^{238}U (n,γ) uncertainty and cross sections in a single view in Fulcrum.

The zoom and pan controls for nuclear data plots are different from those used in geometry visualization. Zoom is controlled by the scroll wheel: scrolling up zooms in and scrolling down zooms out. The image can be panned by simply left clicking on the plot and moving the mouse. The default image can be restored by right clicking on the plot and selecting **Fit graphs**.

9.3 PLOTTING MESH DATA IN FULCRUM

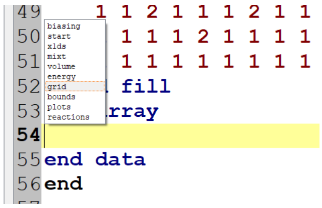
The primary mesh-based data generated and used in KENO is the neutron production edit enabled by the *cds=yes* entry in the PARAMETER block. The mesh is specified in the GRIDGEOMETRY block, and the neutron production in each voxel is tallied by KENO and written to a *3dmap file. This file can be opened in Fulcrum and the mesh data can be plotted on top of the Fulcrum 2D geometry visualization. This capability will be demonstrated here by specifying a mesh on the fuel assembly problem from Section 8.5.1; the mesh will be equal to the fuel rod pitch in X and Y and 4 inches (10.16 cm) in Z.

To create the input, add the parameter `cds=yes` to the PARAMETER block. The PARAMETER block can be added to the KENO input outside of the other blocks by typing it directly or by entering it using autocomplete. The updated parameter block is shown in Figure 286, including **cds=yes** and an increased number of particles per generation (**NPG**), skipped generations (**NSK**), and total generations (**GEN**). Next, the GRIDGEOMETRY block must be added. Insert a blank line after **end array**, and use autocomplete to add the grid block to the input as shown in Figure 287. Mesh planes can be added explicitly or by specifying a range for linear interpolation; only Cartesian mesh planes are supported in KENO at this time. Unfortunately, autocomplete is not supported in Fulcrum in the GRIDGEOMETRY block. The mesh over the fuel assembly will be specified with a series of linear interpolation specifications. In each case, the input is the keyword, the number of intervals, and the minimum and maximum values within which the interpolation will be performed. Individual mesh planes can also be added explicitly in the GRIDGEOMETRY block. The radial mesh to locate each rod in a separate interval is therefore `xlinear 9 -7.38 7.38` and `ylinear 9 -7.38 7.38`. Similarly, the axial mesh can be specified as `zlinear 36 -182.88 182.88`; the complete input specification for the GRIDGEOMETRY block is shown in Figure 288. Finally, run the KENO simulation. Larger numbers of particles and the associated longer runtimes will reduce the uncertainties in the local neutron production tallies and result in smoother distributions. The results shown below are based on a calculation which simulated 50,000 neutrons per generation for 2,450 generations, skipping the first 50.



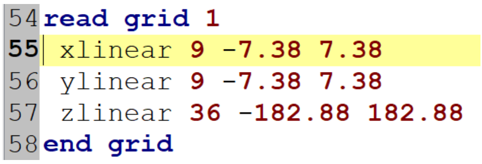
```
read parameters
 npg=60000 nsk=50 gen=3500
 cds=yes
end parameters
```

Figure 286. Updated parameter block with cds=yes included to enable the neutron production tally.



```
      1 1 2 1 1 1 2 1 1
        1 1 1 2 1 1 1 1
        1 1 1 1 1 1 1 1
        fill
       rray

end data
end
```

Figure 287. Adding GRID block to specify mesh for the neutron production tally.



```
read grid 1
 xlinear 9 -7.38 7.38
 ylinear 9 -7.38 7.38
 zlinear 36 -182.88 182.88
end grid
```

Figure 288. GRIDGEOMETRY block for fuel assembly problem.

After execution, the results can be visualized in Fulcrum. Open the input and visualize it, as discussed in Section 9.2.1. Click the **Meshes** button, which is located next to the view orientation buttons at the top middle of the visualization window, as shown in Figure 289. This opens the meshes pane and allows management of different mesh-based SCALE data formats. In this case, the neutron production rate is written to a *3dmap file; to load it into Fulcrum, right click in the meshes pane and select **Load mesh file**, as shown in Figure 290. The *3dmap file will be located in the same directory as the input, and in this case it will be named “fuel_assembly.fissionSource.3dmap.” Once the mesh file is loaded, the available data sets are shown. Expand **fission rate** as shown in Figure 291; note that the neutron generation rate has been tallied by group, along with absolute and relative uncertainties. Scroll to the bottom of the list for total reaction rates and uncertainties and click the check box for **fission rate – total**. Next, change the render mode from **Material** to **Overlay** to display the total neutron production rate on the default midplane view, as shown in Figure 292. Scroll down in the mesh view control panel on the left hand side of the viewing window and change the Scale from Logarithmic to Linear. This shifts the color scale onto a linear scale, and the rates displayed on the geometry appear more realistic, as shown in Figure 293.

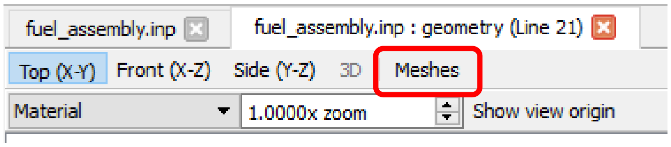


Figure 289. Location of the Meshes button in the Fulcrum visualization window.

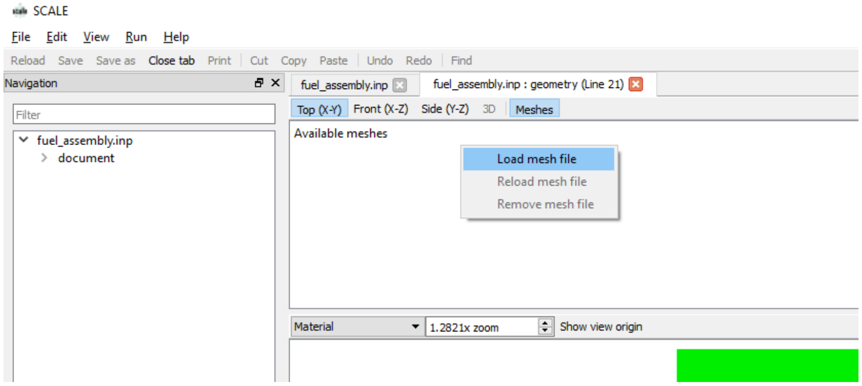


Figure 290. Loading a mesh file in Fulcrum.

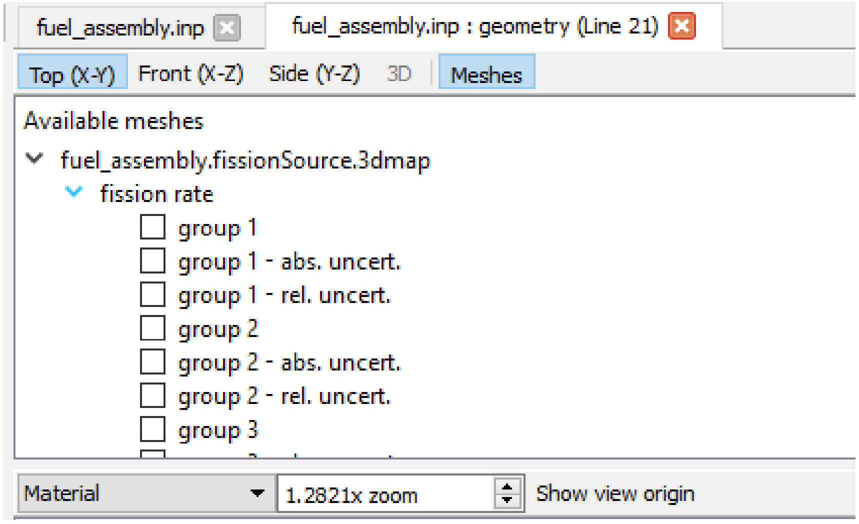
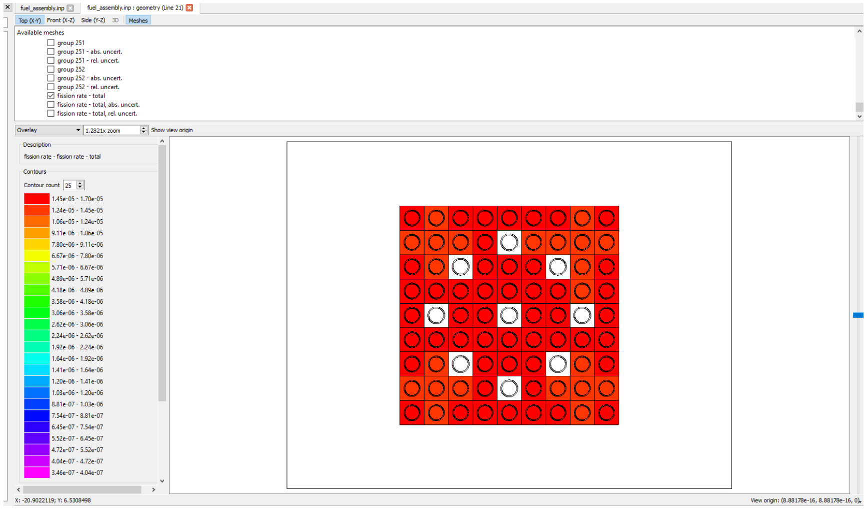


Figure 291. Loaded and expanded 3dmap file for the fuel assembly model.



Available meshes

- ☐ group 251
- ☐ group 251 - abs. uncert.
- ☐ group 251 - rel. uncert.
- ☐ group 252
- ☐ group 252 - abs. uncert.
- ☐ group 252 - rel. uncert.
- ☑ fission rate - total
- ☐ fission rate - total, abs. uncert.
- ☐ fission rate - total, rel. uncert.

Overlay | 1.2821x zoom

Description: fission rate - fission rate - total

Contours

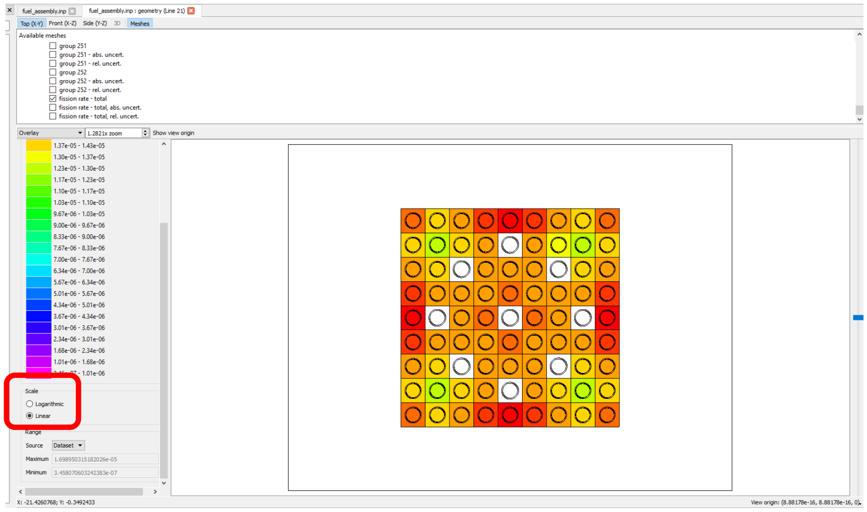
Contour count: 25

1.45e-05 - 1.70e-05
1.24e-05 - 1.45e-05
1.06e-05 - 1.24e-05
9.11e-06 - 1.06e-05
7.80e-06 - 9.11e-06
6.67e-06 - 7.80e-06
5.71e-06 - 6.67e-06
4.89e-06 - 5.71e-06
4.18e-06 - 4.89e-06
3.58e-06 - 4.18e-06
3.06e-06 - 3.58e-06
2.62e-06 - 3.06e-06
2.24e-06 - 2.62e-06
1.92e-06 - 2.24e-06
1.64e-06 - 1.92e-06
1.41e-06 - 1.64e-06
1.20e-06 - 1.41e-06
1.03e-06 - 1.20e-06
8.81e-07 - 1.03e-06
7.54e-07 - 8.81e-07
6.45e-07 - 7.54e-07
5.52e-07 - 6.45e-07
4.72e-07 - 5.52e-07
4.04e-07 - 4.72e-07
3.46e-07 - 4.04e-07

X: -20.9022119; Y: 6.5308498

View origin: (8.88178e-16, 8.88178e-16, 0)

Figure 292. Default view of neutron production rate at the fuel assembly midplane.



Available meshes

- ☐ group 251
- ☐ group 251 - abs. uncert.
- ☐ group 251 - rel. uncert.
- ☐ group 252
- ☐ group 252 - abs. uncert.
- ☐ group 252 - rel. uncert.
- ☑ fission rate - total
- ☐ fission rate - total, abs. uncert.
- ☐ fission rate - total, rel. uncert.

Overlay | 1.2821x zoom

1.37e-05 - 1.43e-05
1.30e-05 - 1.37e-05
1.23e-05 - 1.30e-05
1.17e-05 - 1.23e-05
1.10e-05 - 1.17e-05
1.03e-05 - 1.10e-05
9.67e-06 - 1.03e-05
9.00e-06 - 9.67e-06
8.33e-06 - 9.00e-06
7.67e-06 - 8.33e-06
7.00e-06 - 7.67e-06
6.34e-06 - 7.00e-06
5.67e-06 - 6.34e-06
5.01e-06 - 5.67e-06
4.34e-06 - 5.01e-06
3.67e-06 - 4.34e-06
3.01e-06 - 3.67e-06
2.34e-06 - 3.01e-06
1.68e-06 - 2.34e-06
1.01e-06 - 1.68e-06
3.46e-07 - 1.01e-06

Scale

- ○ Logarithmic
- ● Linear

Range

Source: Dataset

Maximum: 1.698950315182026e-05

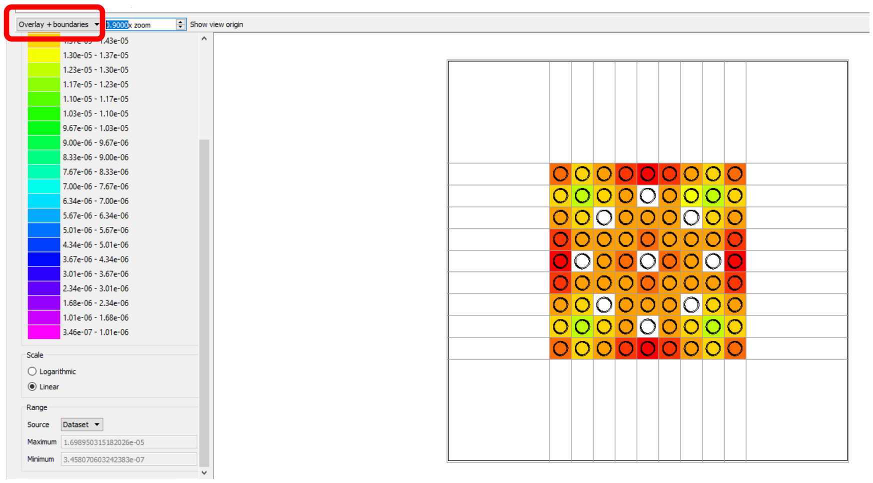
Minimum: 3.458070603242383e-07

X: -21.4260768; Y: -0.3492433

View origin: (8.88178e-16, 8.88178e-16, 0)

Figure 293. Fuel assembly midplane with colors rescaled to linear bins.

The mesh itself can also be displayed in Fulcrum. Change the Render mode dropdown menu from **Overlay** to **Overlay + boundaries** to show the mesh planes. It may be necessary to change the zoom factor slightly to display all mesh planes, as shown in Figure 294. The uncertainties in the mesh tallies can also be displayed; check the box for **fission rate – total, rel. uncert.** in the meshes pane as shown in Figure 295. The resulting updated figure is shown in Figure 296. The color scale can also be specified to allow better contrast for the data being displayed. In the lower portion of the mesh view control panel, change the **Source** dropdown in the **Range** box from **Dataset** to **Custom**, as shown in Figure 297. It is clear that all the uncertainties are in the lowest bin, so set the **Minimum** to `7.5e-4` and the **Maximum** to `1e-3`, as shown in Figure 297. The updated plot shown in Figure 298 provides a better view of how the uncertainty varies in this specific plane.

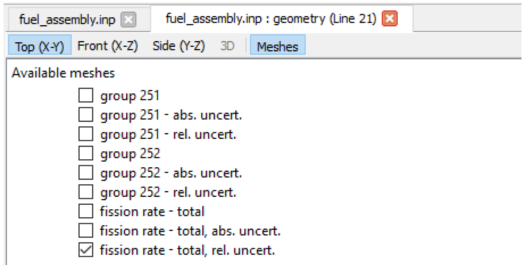


Overlay + boundaries | 0.9000x zoom | Show view origin

Scale
○ Logarithmic
● Linear

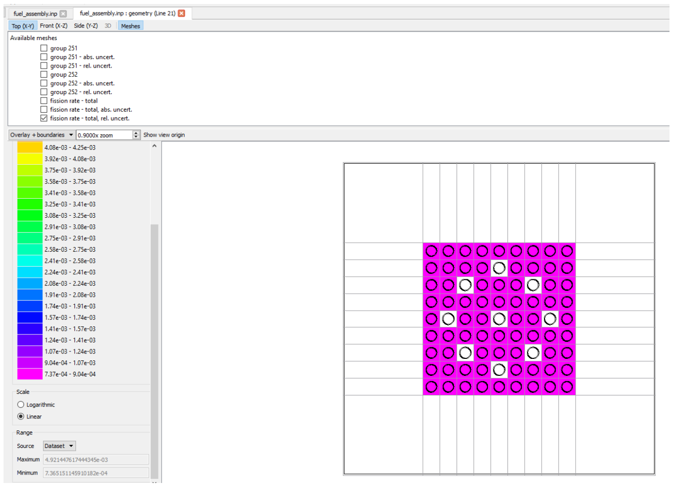
Range
Source: Dataset
Maximum: 1.698950315182026e-05
Minimum: 3.458070603242383e-07

Figure 294. Midplane neutron production rate with mesh boundaries displayed.



Available meshes
☐ group 251
☐ group 251 - abs. uncert.
☐ group 251 - rel. uncert.
☐ group 252
☐ group 252 - abs. uncert.
☐ group 252 - rel. uncert.
☐ fission rate - total
☐ fission rate - total, abs. uncert.
☑ fission rate - total, rel. uncert.

Figure 295. Selecting relative uncertainties as the overlay quantity to view.



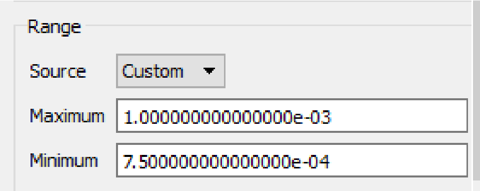
Available meshes
☐ group 251
☐ group 251 - abs. uncert.
☐ group 251 - rel. uncert.
☐ group 252
☐ group 252 - abs. uncert.
☐ group 252 - rel. uncert.
☐ fission rate - total
☐ fission rate - total, abs. uncert.
☑ fission rate - total, rel. uncert.

Overlay + boundaries | 0.9000x zoom | Show view origin

Scale
○ Logarithmic
● Linear

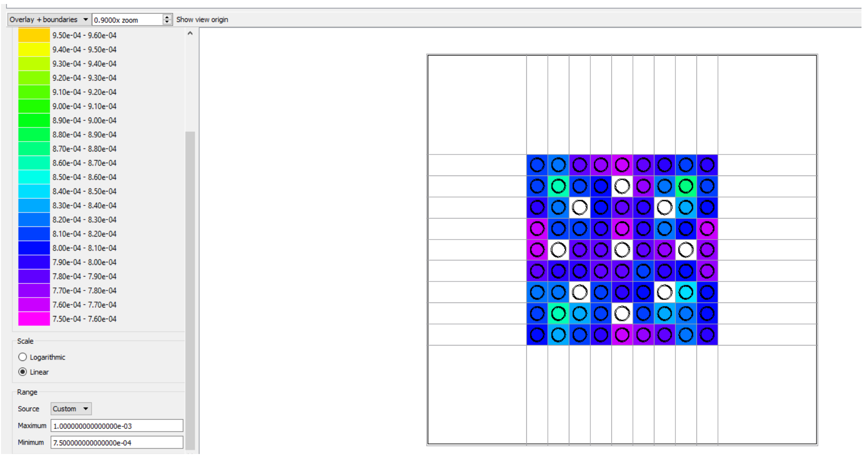
Range
Source: Dataset
Maximum: 4.921447617444345e-03
Minimum: 7.365151145910182e-04

Figure 296. Relative uncertainties for the neutron production rate at the fuel assembly midplane.



Range
Source: Custom
Maximum: 1.000000000000000e-03
Minimum: 7.500000000000000e-04

Figure 297. Updated uncertainty range.



Overlay + boundaries ▾ 0.9000x zoom

Scale
○ Logarithmic
● Linear

Range
Source: Custom ▾
Maximum: 1.000000000000000e-03
Minimum: 7.500000000000000e-04

Figure 298. Overlay plot of uncertainties with custom value range.

The visualization controls discussed in Section 9.2.1 allow for detailed review of the neutron production rates throughout the model. The XY visualization plane can be moved axially with the slider or by setting the plane in the **View origin** field. As shown in Figure 299, axial slices can be viewed by selecting the XZ or YZ orientation buttons. Check the box to overlay **fission rate – total** again in the meshes pane and reset the **Source** dropdown in the **Range** control box to **Dataset**.

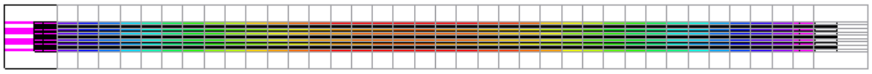


Figure 299. Axial view of the fuel assembly neutron production rate distribution rotated such that +Z is to the right.

The last capability that will be highlighted here is the ability to generate 1D plots from the mesh data displayed in Fulcrum. The example here will be for an axial trace along a single rod, but plots can also be generated along a row (X axis), column (Y axis), or as a function of energy. Right click on the middle rod of the left-most column of the assembly, as indicated in Figure 300, and select **Create plot** (Figure 301). Select the **z axis** as the **Independent axis** in the **Options** box of the subsequent window as shown in Figure 302. Fulcrum creates a new file and opens the text version of that file in a separate tab in much the same way that the convergence summary plot files open, as discussed in Section 9.2.2 and shown in Figure 263. Double click the word **document** in the **Navigation** panel as shown in Figure 303, and Fulcrum will generate a new plot as shown in Figure 304.

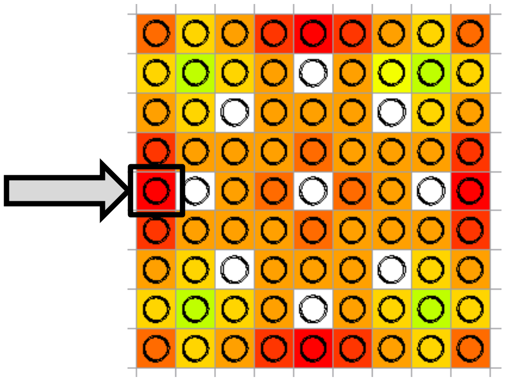


Figure 300. Selection of rod for axial plot.

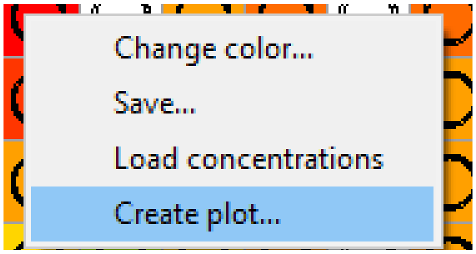
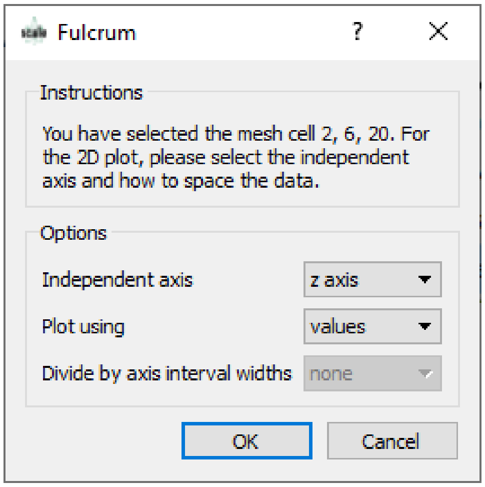


Figure 301. Selecting Create plot… in Fulcrum.


Fulcrum

Instructions

You have selected the mesh cell 2, 6, 20. For the 2D plot, please select the independent axis and how to space the data.

Options

Independent axis: z axis

Plot using: values

Divide by axis interval widths: none

Figure 302. Selecting Independent axis for axial trace.

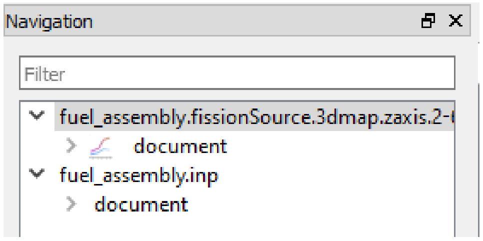


Figure 303. Plot figure generated by Fulcrum and displayed in the Navigation panel.

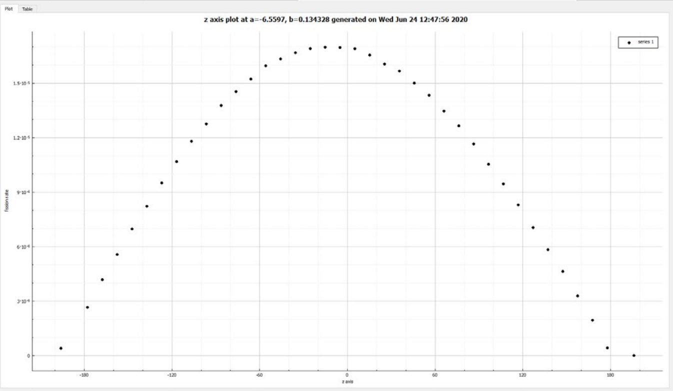


Figure 304. Axial trace of neutron production rate.

9.4 INTERACTIVE 3D VISUALIZATION WITH KENO3D

KENO3D enables KENO-VI users to interactively display their 3D geometry models. KENO3D only runs on the Windows operating system. The interactive options include the following:

- Shaded or wireframe images
- Standard views, such as top view, side view, front view, and isometric (3D) view
- Rotating the model
- Zooming in on selected locations
- Selecting parts of the model to display
- Editing colors and displaying legends
- Displaying properties of any unit in the model
- Creating cut-away views
- Removing units from the model
- Copying images to clipboard or saving images to common graphics formats

KENO3D reads CSAS/KENO-VI input files. It attempts to verify that the KENO geometry input is "legal," that is, KENO3D checks that the input conforms to the code guidelines. KENO3D prints a warning message for illegal geometry input, and if possible, it displays the illegal KENO geometry to facilitate debugging of the input. KENO3D has a fully integrated help system to aid users in learning how to use it. Only a brief demonstration showing how to use KENO3D will be presented here. Users are strongly encouraged to follow the tutorials in the KENO3D help files to quickly learn how to use the many powerful options. KENO3D has several options that provide users with increased flexibility in visualizing portions of a model in greater detail or for visualizing parts of larger models that may be too big to provide a useful image when viewed in their entirety.

To open KENO3D, double-click the **keno3d.exe** file located in the KENO3D directory within the SCALE install directory. Next, to display the 9 × 9 fuel assembly descried in Section 8.5.1, select **File** > **Open…** (or `Ctrl+o`), navigate to the directory containing the input, and select **fuel_assembly.inp**. The default isometric projection is shown in Figure 305.

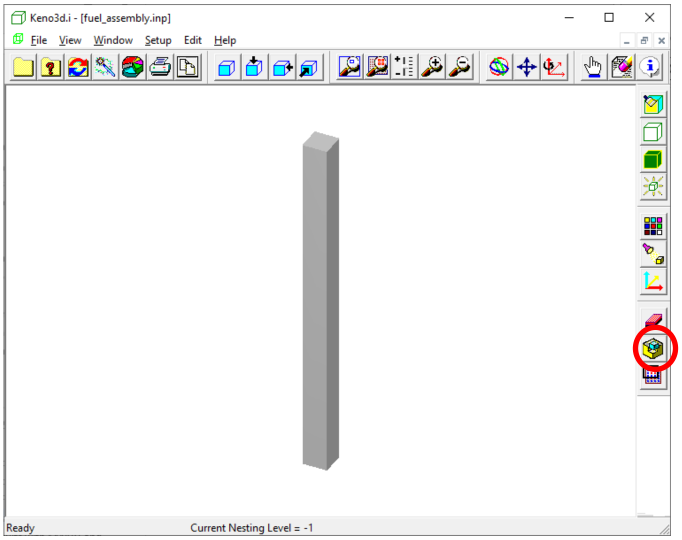
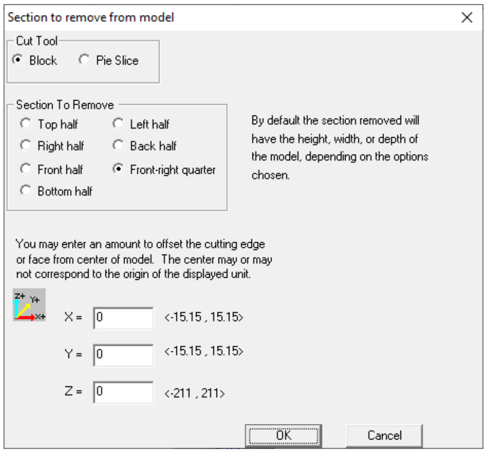


Figure 305. Default KENO3D view of the fuel assembly model.

This isometric view shows the water reflector that surrounds the assembly. Next, click on the **Remove a Section** button (see circle in Figure 305 above). This opens the **Section to remove from model** dialog form shown in Figure 306. Click on the **Top half** radio button under **Section to Remove** and click **OK**. This action removes the top half of the model. Click the **Remove a Section** button again and click **OK** to remove the **Front-right quarter**. Click on the blue magnifying glass button (**Zoom all**) on the top toolbar. The image should now look similar to that shown in Figure 307.



Section to remove from model

Cut Tool

● Block ○ Pie Slice

Section To Remove

○ Top half ○ Left half
○ Right half ○ Back half
○ Front half ● Front-right quarter
○ Bottom half

By default the section removed will have the height, width, or depth of the model, depending on the options chosen.

You may enter an amount to offset the cutting edge or face from center of model. The center may or may not correspond to the origin of the displayed unit.

X = 0 <-15.15 , 15.15>

Y = 0 <-15.15 , 15.15>

Z = 0 <-211 , 211>

OK Cancel

Figure 306. Remove section dialog.

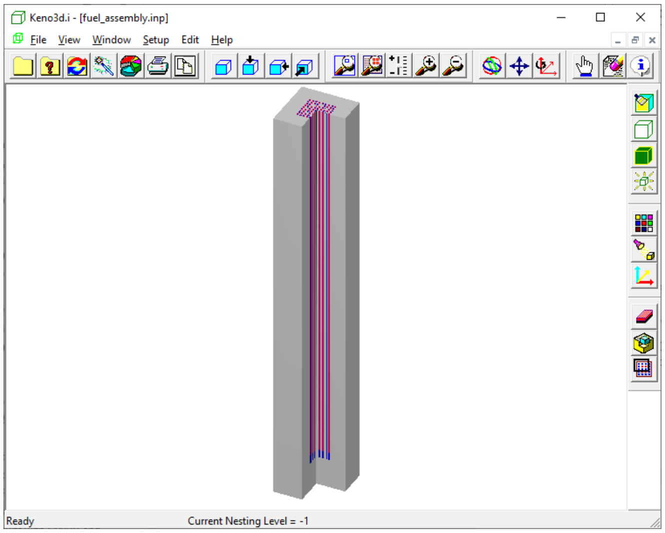

Keno3d.i - [fuel_assembly.inp]
File View Window Setup Edit Help
Ready
Current Nesting Level = -1

Figure 307. Cutaway view of the bottom half of the assembly.

Next, hide the water so that the fuel rods are more easily visible. Click on the **Hide mixture (eraser)** button on the top toolbar and select `0 void` and `3 h2o` under **Select mixture(s) to hide** and click **OK**. The image now looks like that presented in Figure 308.

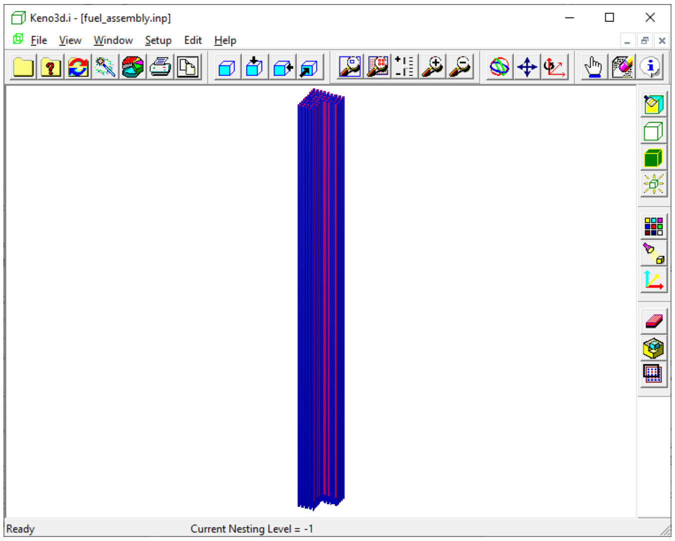

Keno3d.i - [fuel_assembly.inp]
File View Window Setup Edit Help
Ready
Current Nesting Level = -1

Figure 308. Cutaway view of the bottom half of the assembly with water removed.

To zoom in on the bottom part of the fuel assembly to see the details more clearly, click on the **Zoom Window** button (red magnifying glass on the top toolbar). Then left-click and drag to make a rectangle around the bottom portion of the assembly (Figure 309). Once the window is satisfactory, left click again to zoom in. The resulting image should look like that shown in Figure 310.

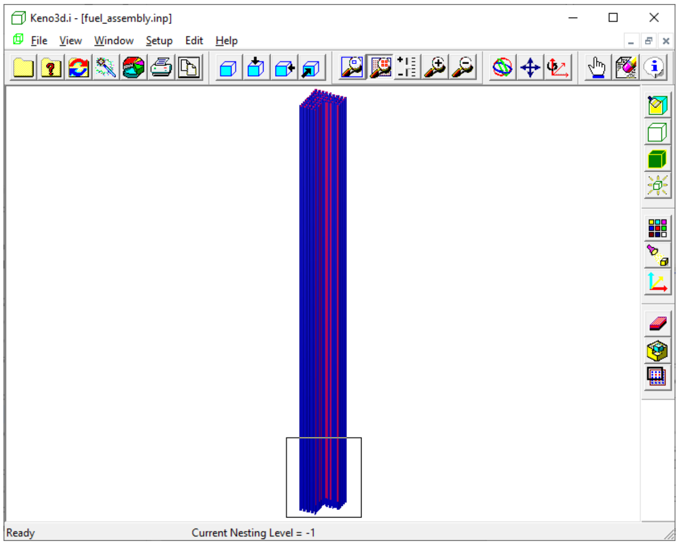

Keno3d.i - [fuel_assembly.inp]
File View Window Setup Edit Help
Ready
Current Nesting Level = -1

Figure 309. Zooming on lower portion of the fuel assembly.

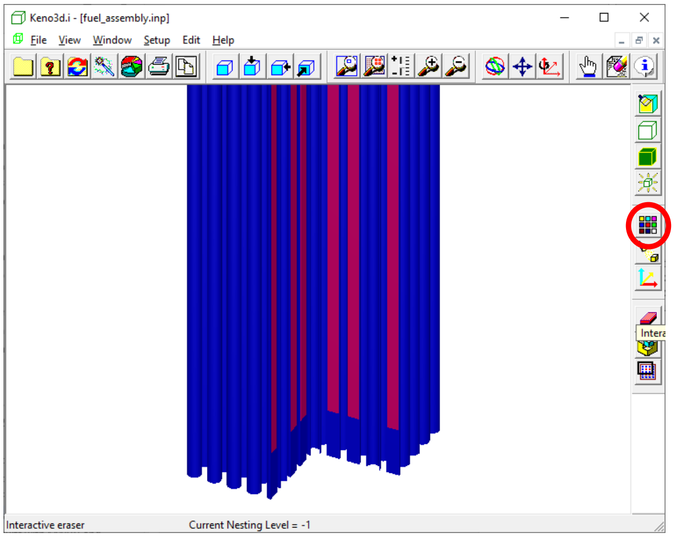

Keno3d.i - [fuel_assembly.inp]
File View Window Setup Edit Help
Intera
Interactive eraser
Current Nesting Level = -1

Figure 310. Zoomed view of cutaway view of the bottom of the fuel assembly with water removed.

Next, display the legend by clicking the **Display legend** button, circled in Figure 310 above. Click **OK** in the Legend options dialog for an image like that shown in Figure 311.

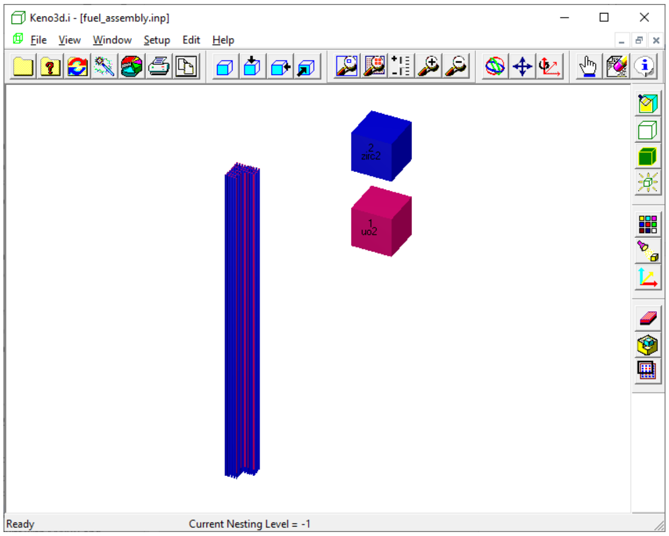
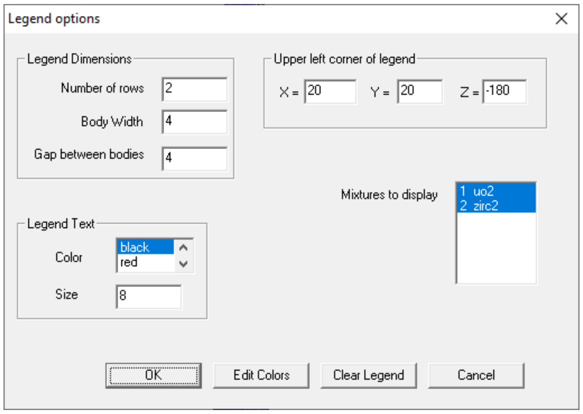

Keno3d.i - [fuel_assembly.inp]
File View Window Setup Edit Help
2
zirc2
1
uo2
Ready
Current Nesting Level = -1

Figure 311. Default view with legend.

Adding the legend has removed the focus on the zoom window and has returned the view of the bottom half of the fuel assembly. In addition, the legend is too large and is not positioned near the bottom of the fuel assembly. This can be fixed by changing the display legend options. Click on the **Display legend** button again. Change the **Body Width** and **Gap between bodies** to `4`. Set the **Upper left corner of legend** to **X=** `20`, **Y=** `20`, and **Z=** `-180`. Set the **Size** of the **Legend Text** to `8` (Figure 312). Click **OK**, and the screen will look like Figure 313. Finally, zoom in on the lower portion of the fuel assembly again to generate an image like that shown in Figure 314.



Legend options

Legend Dimensions

Number of rows: 2

Body Width: 4

Gap between bodies: 4

Upper left corner of legend

X = 20 Y = 20 Z = -180

Mixtures to display: 1 uo2, 2 zirc2

Legend Text

Color: black, red

Size: 8

Figure 312. Legend options dialog.

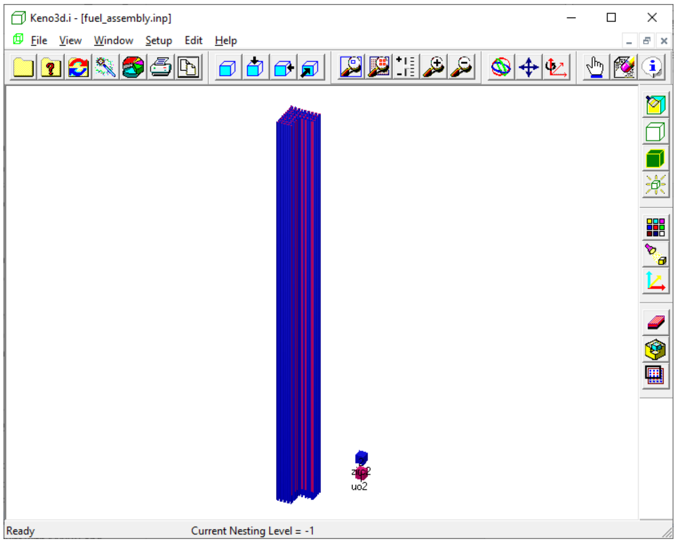


Figure 313. Figure with modified legend.

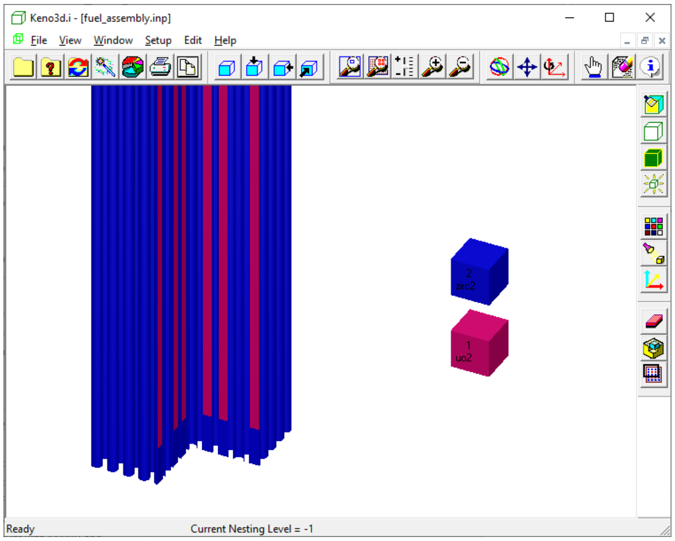


Figure 314. Zoomed image with modified legend.

9.5 PLOTTING CALCULATED RESULTS WITH KMART AND KENO3D

KMART6 (KENO Module for Activity-Reaction Rate Tabulation Version 6) is a tool that post-processes information from a KENO-VI run to provide reaction rates by nuclide and the collapsing and printing of fluxes. See the [SCALE manual section titled *KMART5 and KMART6*] for details on KMART. KENO3D has the capability to plot the calculated results produced by KENO-VI/KMART6 overlaid on the 3D geometry model. The resulting *kmt file also contains the data in text tables, which are often easier to read than the output KMART adds to the standard output file.

The KENO-VI user must be aware of a few items when using KMART.

- KMART uses a restart file for its evaluations, so the user must make sure that a restart unit number (**WRS=**) is identified, and the number of generations between restart (**RES=**) must be greater than zero. Both of these values are set in the PARAMETER block.

- If detailed reaction rates or fluxes are desired, then the user must modify the geometry input to create finer regions to provide more detail for the plot, because KENO-VI collects calculated data on a unit and region basis. The level of detail in the results is directly dependent upon the level of detail in the individual geometry regions.

- Region volumes are highly desired to normalize activities so that low reaction rates in large volumes do not appear to be more important than high reaction rates in small volumes. Volume calculations in CSAS6 are discussed in Section 4.7.

The Sheba solution reactor will be used as an example in this section. A simple KENO-VI input is provided in Figure 315. A view of the model in KENO3D is shown in Figure 316. In this example, the input file name is sheba2.inp.

```
=csas6
sheba 2
v7.1-252
read comp
  solution mix=1 rho[uo2f2]=978
    92234 0.027 92235 4.998 92236 0.049 92238 94.926
    molar[hfacid]=0.25 density=2.1092 temp=293 end solution
  ss304s 2 den=7.8419 1 293 end
end comp
read parameter
  npg=2000 nsk=50 gen=250
end parameter
read geometry
unit 1
com='fuel in tank - three radial subdivisions'
cylinder 1 2.54 44.8 0
cylinder 2 3.175 44.8 0
cylinder 3 8.14917 44.8 0
cylinder 4 16.29833 44.8 0
cylinder 5 24.4475 44.8 0
cylinder 6 25.4 44.8 0
cuboid 7 25.4 -25.4 25.4 -25.4 44.8 0
media 0 1 1
media 2 1 2 -1
media 1 1 3 -2
media 1 1 4 -3
media 1 1 5 -4
media 2 1 6 -5
media 0 1 7 -6
boundary 7
unit 10
com='bottom of tank'
 cylinder 1  2.54  2.8575 0
 cylinder 2  25.4  2.8575 0
 cuboid 3 25.4 -25.4  25.4 -25.4  2.8575 0
 media 0 1  1
 media 2 1  2 -1
 media 0 1  3 -2
boundary 3
unit 11
com='void section of tank'
  cylinder 1 2.54     76.5175 44.8
 cylinder 2 3.175    76.5175 44.8
 cylinder 3 24.4475  76.5175 44.8
 cylinder 4 25.4     76.5175 44.8
 cuboid 5 25.4 -25.4  25.4 -25.4  76.5175 44.8
 media 0 1  1
 media 2 1  2 -1
 media 0 1  3 -2
 media 2 1  4 -3
 media 0 1  5 -4
boundary 5
unit 12
com='top of tank'
 cylinder 1  2.54  1.905 0
 cylinder 2  25.4  1.905 0
 cuboid 3 25.4 -25.4  25.4 -25.4  1.905 0
 media 0 1  1
 media 2 1  2 -1
 media 0 1  3 -2
boundary 3
global unit 13
 cuboid 1  50.8 0  50.8 0  81.28 0
 array 1 1 place 1 1 1  25.4 25.4 0
boundary 1
end geometry
read array
ara=1 nux=1 nuy=1 nuz=4 typ=square
 fill
  10 1 11 12
end fill
end array
end data
end
```

Figure 315. Sample input for simplified Sheba solution reactor.

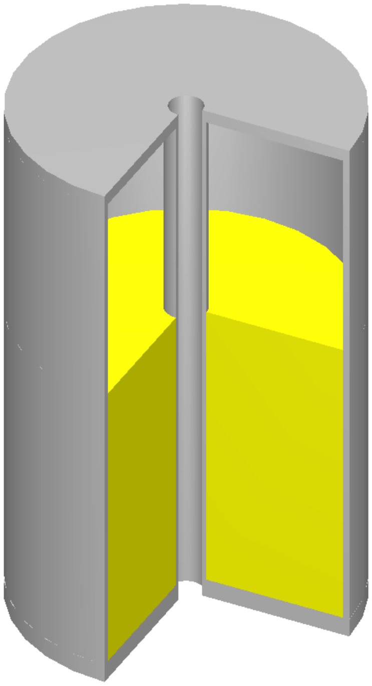
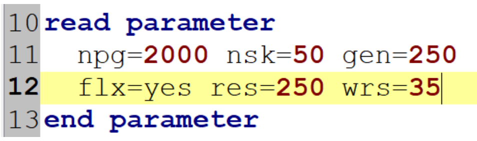


Figure 316. KENO3D rendering of simplified Sheba model.

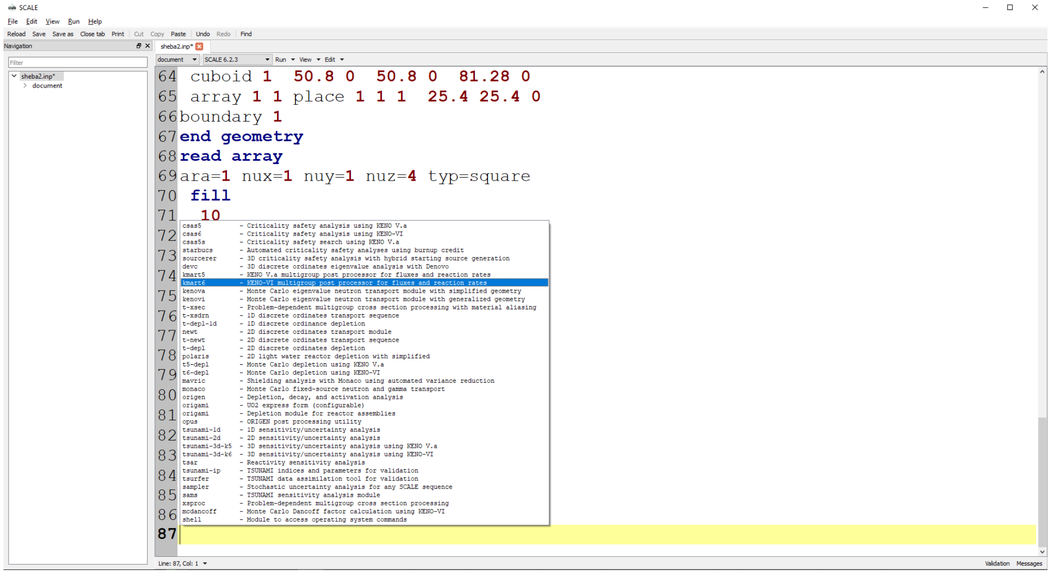
The next step is to add parameters to write the restart file and add the flux edits to be used by KMART. The flux edits are enabled by specifying `flx=yes` in the PARAMETER block, either by typing it in directly or using Fulcrum autocomplete. The restart file must be written to a specific unit after a specific number of generations. The restart file can be written at any interval of generations, or it can be written at the end of the calculation. In this case, it only needs to be written at the end of the calculation, so set `RES=250` in the PARAMETER block. Finally, set `WRS=35` in the PARAMETER block as well; unit 35 is not used for any other purpose within CSAS by default, so it is available for the restart file. The final PARAMETER block is shown in Figure 317.



```
read parameter
  npg=2000 nsk=50 gen=250
  flx=yes res=250 wrs=35
end parameter
```

Figure 317. Updated PARAMETER block for KMART calculations.

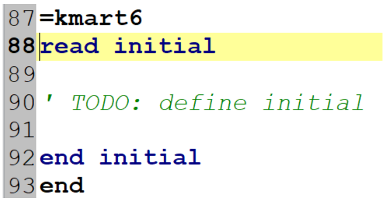
The next step is to provide the input for KMART itself; KMART is a separate sequence from CSAS, so the input will be provided after the final **end** terminating the CSAS input. The stacked cases, CSAS and KMART, will be run in a single SCALE execution, so the temporary directory containing the restart file and cross section library will remain intact. Add a blank line to the end of the input and use autocomplete (`CTRL-SPACE`) to add a KMART6 sequence, as shown in Figure 318. The skeleton KMART input is shown in Figure 319.



```
 cuboid 1  50.8 0  50.8 0  81.28 0
 array 1 1 place 1 1 1  25.4 25.4 0
boundary 1
end geometry
read array
ara=1 nux=1 nuy=1 nuz=4 typ=square
 fill
  10
```

```
csas5          - Criticality safety analysis using KENO V.a
csas6          - Criticality safety analysis using KENO-VI
csas5s         - Criticality safety search using KENO V.a
starbucs       - Automated criticality safety analyses using burnup credit
sourcerer      - 3D criticality safety analysis with hybrid starting source generation
devc           - 3D discrete ordinates eigenvalue analysis with Denovo
kmart5         - KENO V.a multigroup post processor for fluxes and reaction rates
kmart6         - KENO-VI multigroup post processor for fluxes and reaction rates
kenova         - Monte Carlo eigenvalue neutron transport module with simplified geometry
kenovi         - Monte Carlo eigenvalue neutron transport module with generalized geometry
t-xsec         - Problem-dependent multigroup cross section processing with material aliasing
t-xsdrn        - 1D discrete ordinates transport sequence
t-depl-1d      - 1D discrete ordinance depletion
newt           - 2D discrete ordinates transport module
t-newt         - 2D discrete ordinates transport sequence
t-depl         - 2D discrete ordinates depletion
polaris        - 2D light water reactor depletion with simplified
t5-depl        - Monte Carlo depletion using KENO V.a
t6-depl        - Monte Carlo depletion using KENO-VI
mavric         - Shielding analysis with Monaco using automated variance reduction
monaco         - Monte Carlo fixed-source neutron and gamma transport
origen         - Depletion, decay, and activation analysis
origami        - UO2 express form (configurable)
origami        - Depletion module for reactor assemblies
opus           - ORIGEN post processing utility
tsunami-1d     - 1D sensitivity/uncertainty analysis
tsunami-2d     - 2D sensitivity/uncertainty analysis
tsunami-3d-k5  - 3D sensitivity/uncertainty analysis using KENO V.a
tsunami-3d-k6  - 3D sensitivity/uncertainty analysis using KENO-VI
tsar           - Reactivity sensitivity analysis
tsunami-ip     - TSUNAMI indices and parameters for validation
tsurfer        - TSUNAMI data assimilation tool for validation
sampler        - Stochastic uncertainty analysis for any SCALE sequence
sams           - TSUNAMI sensitivity analysis module
xsproc         - Problem-dependent multigroup cross section processing
mcdancoff      - Monte Carlo Dancoff factor calculation using KENO-VI
shell          - Module to access operating system commands
```

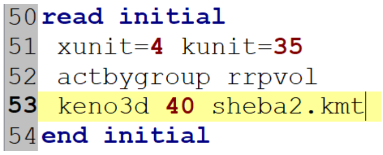
Figure 318. Adding the KMART sequence input after the CSAS sequence input.



```
=kmart6
read initial

' TODO: define initial

end initial
end
```

Figure 319. Skeleton KMART input provided by Fulcrum autocomplete.

The first block, and the only one provided by Fulcrum, is the INITIAL block. This block is analogous to the PARAMETER block in KENO. Two inputs are needed: the unit numbers for the restart file and the cross section library used in the KENO calculation. The cross section library is written on unit 4 in all CSAS calculations, so the specification to provide is `xunit=4`. In this example, the restart file has been written to unit 35, so the input specification for that is `kunit=35`. KMART generates activities by group by declaring `actbygrp` in the INITIAL block. Normalizing the activities by region volumes is also frequently helpful to prevent small reaction rates in large regions from appearing more important than large reaction rates in small volumes of high importance; this option is triggered with the input `rrpvol`. The last input in the INITIAL block in this example is the unit number and filename to which KENO3D data will be written. Both of these inputs are preceded by the keyword `keno3d`. Any unused unit number is acceptable; in this example, use `40`. The KMART filename must be the same as the input filename, but a kmt extension is necessary for KENO3D to recognize it. In this example, use the filename sheba2.kmt. The complete KENO3D input specification is therefore `keno3d 40 sheba2.kmt`. The complete INITIAL block input is provided in Figure 320, with extra blank lines and extraneous comments removed for clarity and brevity.



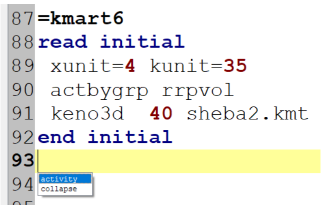
```
read initial
 xunit=4 kunit=35
 actbygroup rrpvol
 keno3d 40 sheba2.kmt
end initial
```

Figure 320. Complete INITIAL block for Sheba model.

The next input block to supply for the KMART sequence is the ACTIVITY block. This block provides the specific nuclides and reactions for which activities will be calculated. The nuclides are specified by nuclide ID number; these numbers can be found in the Standard Composition Library section of the SCALE manual. The MT numbers for specific reactions of interest can be found in [MT Reaction Types on SCALE Cross section Libraries], an appendix to the SCALE Cross Section Libraries section of the manual. Some common MT numbers are provided in Table 4. In this example, activities will be generated only for absorption in ^{1}H and fission in ^{235}U. The ACTIVITY block can be added to the KMART input via autocomplete, as shown in Figure 321. The nuclide ID for ^{1}H is 1001, and the MT number for absorption is 27, so the first input in the ACTIVITY block is `1001 27`. Similarly, the nuclide ID for ^{235}U is 92235, and the MT number for fission is 18; so the input is `92235 18`. The complete ACTIVITY block is shown in Figure 322.

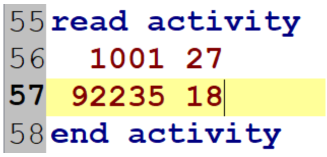
Table 4. Frequently used MT numbers.

MT Number	Reaction Description
1	Total
2	Elastic scattering
18	Total fission
27	Total absorption
1452	Neutron generation from fission (ν × fission)



```
=kmart6
read initial
 xunit=4 kunit=35
 actbygrp rrpvol
 keno3d  40 sheba2.kmt
end initial

```

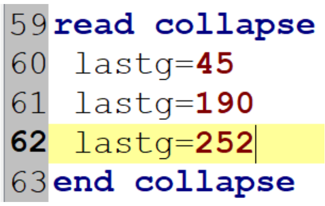
Figure 321. Adding ACTIVITY block to the KMART input.



```
read activity
  1001 27
 92235 18
end activity
```

Figure 322. Complete ATIVITY block for the Sheba model.

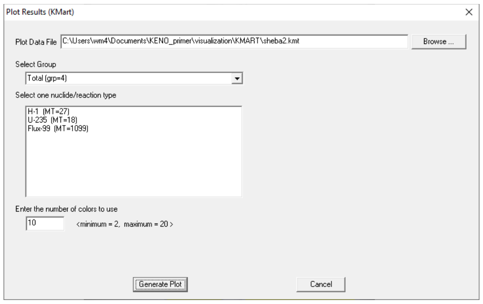
The last input block is an optional block to collapse the results into fewer groups, primarily to make display and analysis of the data more manageable. This is accomplished in the COLLAPSE block by providing the last fine group number to be included in each broad group. The fine group structures are provided in the SCALE Cross Section Libraries section of the SCALE Manual. In this example, a three-group structure will be used for the collapsed, broad-group edits: a fast group cutoff at 100 keV, an intermediate group cut off at 1 eV, and a thermal group extending to the bottom of the library at 10^{-5} eV. First, add the COLLAPSE block using autocomplete in Fulcrum. Then add the three bottom energy groups for the broad groups: `lastg=45`, `lastg=190`, and `lastg=252`. The final COLLAPSE block is shown in Figure 323.



```
read collapse
 lastg=45
 lastg=190
 lastg=252
end collapse
```

Figure 323. Complete COLLAPSE block for the Sheba model.

The input is now complete and can be executed in SCALE. A *kmt file is also generated and returned alongside the output file. With KENO3D open and displaying the model as shown in Figure 316 above, for example, the KMART data can be visualized by selecting **View** > **Plot KMart Data**. This generates a **Plot Results** dialog, as shown in Figure 324, providing available data for plotting. Select **U-235 (MT=18)** and click **Generate Plot** to generate an image such as shown in Figure 325. Finally, click the **Show legend** button and click **OK**, as described in Section 9.4, to display the color scale for the fission rates as shown in Figure 326.



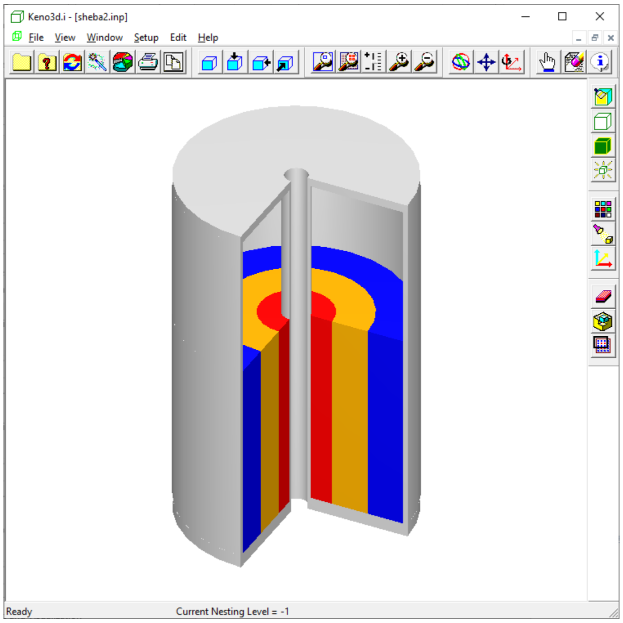
Plot Data File C:\Users\wm4\Documents\KENO_primer\visualization\KMART\sheba2.kmt

Select Group
Total (grp=4)

Select one nuclide/reaction type
H-1 (MT=27)
U-235 (MT=18)
Flux-99 (MT=1099)

Enter the number of colors to use
10 <minimum = 2, maximum = 20 >

Figure 324. Plot Results dialog in KENO3D for selecting KMART results to display.



Ready
Current Nesting Level = -1

Figure 325. KENO3D rendering of the Sheba model with fission reaction rates.

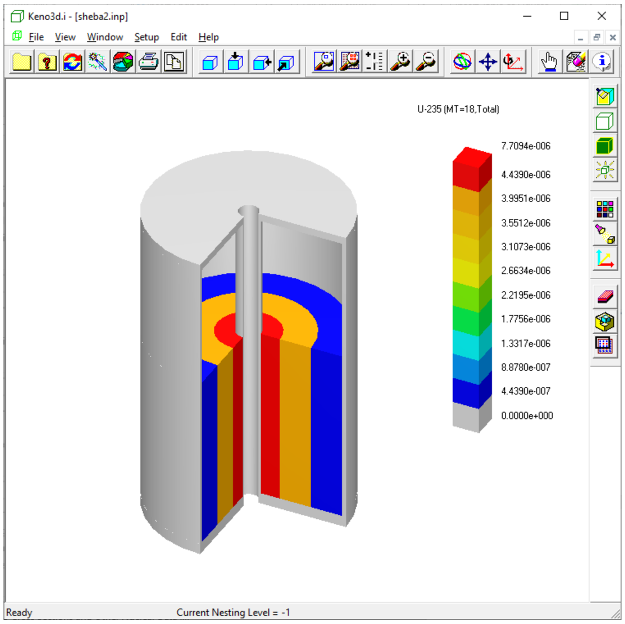


Figure 326. KENO3D rendering of the Sheba model with fission reaction rates and color scale.

The addition of more refinements axially and radially can improve the resolution of the reported fluxes and reaction rates. An example with 45 axial slices, each containing 21 radial divisions, is shown in Figure 327. This is significantly more detailed than the single axial slice with 3 radial divisions used here: the detailed model can be developed using the geometry and array constructs discussed in Sections 4.5 and 4.6.

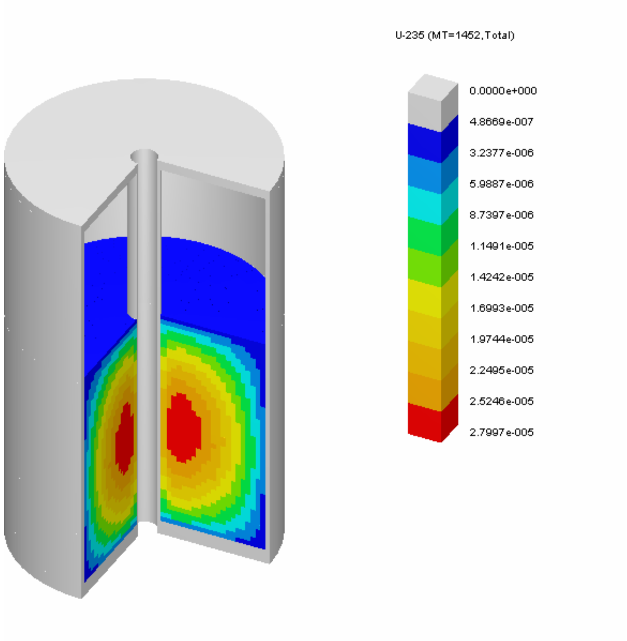


Figure 327. Detailed fission neutron generation rate (MT=1452) in Sheba model.

9.6 SUMMARY

This section has provided an overview of the following capabilities within SCALE and Fulcrum:

- generating 2D color plots of KENO-VI geometry models,
- viewing 2D source convergence diagnostics generated by KENO-VI,
- generating plots of 1D nuclear data used in KENO-VI calculations,
- generating and display mesh-based data in Fulcrum,
- interactively viewing 3D wireframe and rendered images of KENO-VI geometry models using KENO3D, and
- plotting calculated results overlain on the 3D geometry model using KMART and KENO3D.

10. CONCLUSION

This primer provides an introduction to the KENO-VI Monte Carlo transport code within the CSAS6 sequence for use in calculating the neutron multiplication factor, also known as the k_{eff} eigenvalue, primarily for nuclear criticality safety analyses. Aside from serving as a quick start guide, the primer specifies compositions in SCALE, describes model geometry, explains problem-specific cross section processing for MG transport problems, and presents instructions for plotting and visualizing models and data. A number of other resources exist to expand on the information presented here, including:

- The SCALE manual
- SCALE training – check for information and updates at: https://www.ornl.gov/scale/scale-training
- The SCALE web page: https://www.ornl.gov/scale
- A Google discussion group (email scalehelp@ornl.gov for information on joining the group)
- Direct user assistance via scalehelp@ornl.gov

11. REFERENCES

1. W. A. Wieselquist, R. A. Lefebvre, and M. A. Jessee, Eds., SCALE Code System, ORNL/TM-2005/39, Version 6.2.4, Oak Ridge National Laboratory, Oak Ridge, TN (2020).

2. S. Goluoglu, L. M. Petrie, M. E. Dunn, D. F. Hollenbach, and B. T. Rearden, "Monte Carlo Criticality Methods and Analysis Capabilities in SCALE," *Nucl. Technol.* **174(2)**, pp. 214–235 (2011).

3. B. T. Rearden, B. R. Langley, A. B. Thompson, and R. A. Lefebvre, "Fulcrum User Interface for SCALE 6.2," *Proceedings of NCSD 2017: Criticality Safety – pushing boundaries by modernizing and integrating data, methods, and regulations*, Carlsbad, NM (2017).

4. K. Bekar, J. B. Clarity, M. Dupont, R. A. Lefebvre, W. J. Marshall, and E. M. Saylor, KENO V.a Primer: Performing Calculations Using SCALE's Criticality Safety Analysis Sequence (CSAS5) with Fulcrum, ORNL/TM-2020/1664, Oak Ridge, TN (2020).

5. H. C. Paxton and N. L. Pruvost, *Critical Dimensions of Systems Containing* ^{235}U, ^{239}Pu, *and* ^{233}U, LA-10860-MS, Los Alamos, NM (1987).

6. "PBMR Coupled Neutronics/Thermal-hydraulics Transient Benchmark: The PBMR-400 Core Design, Volume I – The Benchmark Definition," NEA/NSC/DOC(2013)10, Paris, France (2013).

7. R. A. Lefebvre, "Advanced User Interface Capabilities," presented at the 2020 SCALE Users' Group Workshop, Oak Ridge, TN (2020). Available at https://www.ornl.gov/file/advanced-user-interface-capabilities-tutorial/display, as of September 4, 2020.

APPENDIX A. INPUT FILES FOR EXAMPLE PROBLEMS

APPENDIX A. INPUT FILES FOR EXAMPLE PROBLEMS

INPUT FOR SECTION 2.3

```
=csas6 parm=(   )
jezebel problem, bare plutonium sphere with nickel shell
v7.1-252

read comp
 pu-239 1 0 0.037047 293.0 end
 pu-240 1 0 1.751e-3 293.0 end
 pu-241 1 0 1.17e-4  293.0 end
 ga     1 0 1.375e-3 293.0 end
 ni     2 0 9.1322e-2 293.0 end
end comp
read celldata
multiregion spherical right_bdy=vacuum end
                     1 6.38493 2 6.39763 end zone
end celldata
read geometry
global unit 1
  sphere 10 6.38493
  sphere 20 6.39763
  media 1 1  10
  media 2 1  20 -10

  boundary 20
end geometry
end data
end
```

INPUT FOR SECTION 3.4.1

```
=csas6 parm=(   )
uranium metal example
ce_v7.1
read comp
u-235 1 den=18.742000 1.0 300 end
end comp
```

INPUT FOR SECTION 3.4.2

```
=csas6 parm=(   )
uranium metal example
ce_v7.1
read comp
    u 1 den=18.742000 1.0 300
     92235 93.71
     92238 6.29    end
end comp
```

INPUT FOR SECTION 3.4.3

```
=csas6 parm=(   )
highly enriched uranium oxide example
ce_v7.1
read comp
   uo2 1  0.96 293.0
        92235 93.71
        92238 6.29 end
end comp
```

INPUT FOR SECTION 3.4.4

```
=csas6 parm=(   )
UO2F2 number density example
ce_v7.1
read comp
u-235 1 0.0 0.0005637 end
u-238 1 0.0 0.0012802 end
h     1 0.0 0.0597522 end
o     1 0.0 0.0335605 end
f     1 0.0 0.0036844 end
end comp
```

INPUT FOR SECTION 3.4.5

```
=csas6 parm=(   )
uo2/h2o mixing example
ce_v7.1
read comp
    uo2 1  0.2 end
    h2o 1  0.8 end
'
    uo2 2 den=2.990600 0.73297 end
    h2o 2 den=2.990600 0.26703 end
end comp
```

INPUT FOR SECTION 4.2.1.1

```
=csas6
Primer Bare u-235 Metal Cylinder
ce_v7.1

read comp
u-235 1 den=18.742 1.0 300 end
end comp

read geometry
global unit 1
cylinder 1 7.82 7.82 -7.82
media 1 1 1
boundary 1
end geometry
end data
end
```

INPUT FOR SECTION 4.2.1.2

```
=csas6 parm=( )
Primer Bare u-235 Metal Cylinder
ce_v7.1

read comp
u-235 1 den=18.742 1.0 300 end
end comp

read geometry
global unit 1
cylinder 1 7.82 13.64 -2.0
media 1 1 1
boundary 1
end geometry
end data
end
```

INPUT FOR SECTION 4.5.1

```
=csas6
Primer Reflected Plutonium Cylinder
ce_v7.1

read comp
Pu 1 den=15.44 1.0 293.0
94239 95.0
94240 5.0 end
c-graphite 2 den=1.6 1.0 end
end comp

read geometry
global unit 1
cylinder 1 2.86 40.96 0.0
cylinder 2 20.64 58.74 -17.78
media 1 1 1
media 2 1 2 -1
boundary 2
end geometry
end data
end
```

INPUT FOR SECTION 4.5.2

```
=csas6
Primer Reflected u-235 Metal Sphere
ce_v7.1

read comp
u-235 1 den=18.74 1.0 end
graphite 2 den=1.65 1.0 end
water 3 1.0 end
end comp

read geometry
```

```
global unit 1
sphere 1 7.0
cylinder 2 10 10 -10
cuboid 3 12.0 -10.0 12.0 -10.0 12.0 -10.0
media 1 1 1
media 2 1 2 -1
media 3 1 3 -2
boundary 3
end geometry
end data
end
```

INPUT FOR SECTION 4.6.1

```
=csas6
Primer 2x2x2 array
ce_v7.1

read composition
U 1 den=18.76 1.0 293
92234 1
92235 93.2
92236 0.2
92238 5.6 end
end composition

read geometry
unit 1
cylinder 1 5.748 5.3825 -5.3825
cuboid 2 6.87 -6.87 6.87 -6.87 6.505 -6.505
media 1 1 1
media 0 1 2 -1
boundary 2

global unit 2
cuboid 1 13.74 -13.74 13.74 -13.74 13.01 -13.01
array 1 1 place 1 1 1 -6.87 -6.87 -6.505
boundary 1
end geometry

read array
ara=1 nux=2 nuy=2 nuz=2 typ=square
fill
1 1
1 1
1 1
1 1 end fill
end array
end data
end
```

INPUT FOR SECTION 4.6.2

```
=csas6
Primer Array with Multiple Units of Different Nominal Sizes
ce_v7.1
```

```
read composition
graphite 1 den=1.65 1.0 end
U 2 den=18.76 1.0 293
92234 1
92235 93.2
92236 0.2
92238 5.6 end
aluminum 3 1.0 end
end composition

read geometry
unit 1
cylinder 1 10 2.5 0
cuboid 2 15 -15 15 -15 2.5 0
media 1 1 1
media 0 1 2 -1
boundary 2

unit 2
cylinder 1 10 4 0
cuboid 2 15 -15 15 -15 4 0
media 2 1 1
media 0 1 2 -1
boundary 2

unit 3
cuboid 1 15 -15 15 -15 3 0
media 3 1 1
boundary 1

Global unit 4
cuboid 1 15 -15 15 -15 22.5 0
array 1 1 place 1 1 1 0 0 0
boundary 1
end geometry

read array
ara=1 nux=1 nuy=1 nuz=7 typ=square
fill
1 2 1 2 1 2 3 end fill
end array

read volume
type=random points=100000 batches=10000
xp=15 xm=-15 yp=15 ym=-15 zp=22.5 zm=0
end volume
end data
end
```

INPUT FOR SECTION 5.3.1

```
=csas6
Primer Three intersecting cylinders
ce_v7.1
```

```
read composition
pu-239 1 den=18.5 1.0 end
h2o 2 1.0 end
ss304 3 1.0 end
end composition

read geometry
Global unit 1
cylinder 1 10 10 -10 origin y=4.5
cylinder 2 10 10 -10 origin x=-4.5 y=-4.5
cylinder 3 10 10 -10 origin x=4.5 y=-4.5
cuboid 4 6p15
media 1 1 1 2 3
media 2 1 1 3 -2
media 2 1 2 3 -1
media 2 1 1 2 -3
media 3 1 1 -2 -3
media 3 1 2 -1 -3
media 3 1 3 -1 -2
media 0 1 4 -1 -2 -3
boundary 4
end geometry
end data
end
```

INPUT FOR SECTION 5.4.4

```
=csas6
Primer 29 Degrees Y-Angle Joint
ce_v7.1

read composition
solution mix=1 rho[uo2f2]=907.01 92235 5 92238 95
density=2.0289 temperature=293 vol_frac=1
end solution
solution mix=1 rho[uo2f2]=1.0 92235 95.5 92238 4.5 density=1.0 temp=300
volfrac=1.0 end solution
al 2 1 293 end
h2o 3 1 293 end
end composition
read geometry
global unit 1
cylinder 1 13.95 135.4 -75.4
cylinder 2 14.11 135.4 -76.7
cylinder 3 13.95 125.4 0 rotate a2=-29.26
cylinder 4 14.11 126.04 0 rotate a2=-29.26
plane 5 zpl=-1 con=52.8
cuboid 6 37 -37 100 -37 135.4 -99.6
media 1 1 1 5
media 2 1 2 -1 -3
media 1 1 3 -1 5
media 2 1 4 -3 -2
media 0 1 1 -5
media 0 1 3 -5 -1
media 3 1 6 -2 -4
boundary 6
```

```
end geometry
end data
end
```

INPUT FOR SECTION 5.5.2

```
=csas6
Primer Simple hole example
ce_v7.1

read composition
Pu 1 den=18.5 1.0 293.0
94240 99.0 94239 1.0 end
ss304 2 1.0 end
reg-concrete 3 1.0 end
end composition

read geometry
unit 1
cuboid 1 5 -5 5 -3 10 -10
cuboid 2 6 -6 6 -6 11 -11
media 1 1 1
media 2 1 2 -1
boundary 2

Global unit 2
cylinder 1 30 12 -12
media 3 1 1
hole 1
hole 1 origin x=12 y=12 rotate A1=-45
hole 1 origin x=12 y=-12 rotate A1=-135
hole 1 origin x=-12 y=-12 rotate A1=135
hole 1 origin x=-12 y=12 rotate A1=45
boundary 1
end geometry
end data
end
```

INPUT FOR SECTION 6.2

```
=csas6
Primer 5 cruciform cylinders in a shipping container
ce_v7.1

read composition
U 1 den=18.76 1.0 293
92234 1
92235 93.2
92236 0.2
92238 5.6 end
h2o 2 1.0 end
ss316 3 1.0 end
end composition

read geometry
unit 1
```

```
cuboid 1 5 -5 0.5 -0.5 10 -10
cuboid 2 0.5 -0.5 5 -5 10 -10
cylinder 3 6 10 -10
cylinder 4 6.1 10.1 -10.1
cuboid 5 6.2 -6.2 6.2 -6.2 10.1 -10.1
media 1 1 1
media 1 1 2 -1
media 2 1 3 -1 -2
media 3 1 4 -3
media 0 1 5 -4
boundary 5

unit 2
cuboid 1 -6.2 6.2 -6.2 6.2 10.1 -10.1
media 0 1 1
boundary 1

global unit 3
cylinder 1 20 10.1 -10.1
cylinder 2 20.2 10.3 -10.3
array 1 1 place 3 3 1 0 0 0
media 3 1 2 -1
boundary 2

end geometry
read array
ara=1 nux=5 nuy=5 nuz=1 typ=square
fill
2 2 2 2 2
2 2 1 2 2
2 1 1 1 2
2 2 1 2 2
2 2 2 2 2
end fill
end array
end data
end
```

INPUT FOR SECTION 6.3

```
=csas6
Primer 5 cruciform cylinders in a shipping container
ce_v7.1

read composition
U 1 den=18.76 1.0 293
92234 1
92235 10
92236 0.2
92238 88.8 end
aluminum 2 1.0 end
h2o 3 1.0 end
carbonsteel 4 1.0 end
end composition

read geometry
```

```
unit 1
cylinder 1 7.5 18.3 0
cylinder 2 8 18.3 -0.5
hexprism 3 10.5 18.3 -0.5
media 1 1 1
media 2 1 2 -1
media 3 1 3 -2
boundary 3

unit 2
hexprism 1 10.5 18.3 -0.5
media 3 1 1
boundary 1

global unit 3
cylinder 1 12 18.3 -0.5
cylinder 2 31.5 18.3 -0.5
cylinder 3 33 18.3 -2
array 1 2 -1 place 3 3 1 0 0 0
media 4 1 1
media 4 1 3 -2
boundary 3

end geometry
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 2 1 2
2 1 1 2 2
2 2 2 2 2
end fill
end array
end data
end
```

INPUT FOR SECTION 7.2.1

```
=csas6 parm=(   )
hydraulic fluid atom composition example
ce_v7.1

read comp
atomoil 1 0.97 4
         6000 2
         1000 6
         14000 1
         16000 1
         1.0 293.0
    end
end comp
```

INPUT FOR SECTION 7.2.2

```
=csas6 parm=(   )
```

```
borated aluminum weight percent mixture example
ce_v7.1

read comp
wtptbal 1 2.65 2
         13000 97.5
         5000   2.5
         1.0 293.0
         5010 10.0 5011 90.0 end
end comp
```

INPUT FOR SECTION 7.2.3

```
=csas6 parm=(   )
actinide solution example
ce_v7.1

read comp
solution mix=1 rho[uo2f2]=907.0 92235 5.0 92238 95.0
density=2.0289 temp=300 volfrac=1.0 end solution
end comp
```

INPUT FOR SECTION 8.5.1

```
=csas6 parm=(   )
Unit Cell - Fuel Assembly
v7.1-252

read comp
 uo2 1  0.95 293.0
        92238 96.5
        92235 3.5 end
 zirc2 2  1.0  end
 h2o   3  end
end comp
read celldata
latticecell squarepitch hpitch=0.82 3 fuelr=0.41 1
                        gapr=0.4875 0 cladr=0.545 2 end
end celldata
read parameters
  npg=10000 nsk=50 gen=10050 sig=50e-5
end parameters

read geometry
unit 1
  cylinder 10 0.47 365.76 0.0
  cylinder 20 0.4875 372.76 0.0
  cylinder 30 0.545 383.26 -4.0
  cuboid 40 0.82 -0.82 0.82 -0.82 383.26 -4.0
'
  media 1 1 10
  media 0 1 20 -10
  media 2 1 30 -20
  media 3 1 40 -30
'
```

```
boundary 40
unit 2
  cylinder 10 0.5 387.26 0.0
  cylinder 20 0.575 387.26 0.0
  cuboid 40 0.82 -0.82 0.82 -0.82 387.26 0.0
'
  media 3 1 10
  media 2 1 20 -10
  media 3 1 40 -20
boundary 40

' TODO: add units below
global unit 3
  cuboid 1 7.38 -7.38 7.38 -7.38 193.63 -193.62
  cuboid 2 15.0 -15.0 15.0 -15.0 208.87 -208.87
'
  array 1 1 place 5 5 1 0.0 0.0 -193.62
  media 3 1 2 -1
boundary 2
end geometry
read array
  ara=1
  prt=yes
  typ=cuboidal
  nux=9
  nuy=9
  nuz=1
  fill
' z = 1
    1 1 1 1 1 1 1 1 1
    1 1 1 1 2 1 1 1 1
    1 1 2 1 1 1 2 1 1
    1 1 1 1 1 1 1 1 1
    1 2 1 1 2 1 1 2 1
    1 1 1 1 1 1 1 1 1
    1 1 2 1 1 1 2 1 1
    1 1 1 1 2 1 1 1 1
    1 1 1 1 1 1 1 1 1
  end fill
end array
end data
end
```

INPUT FOR SECTION 8.5.2

```
=csas6 parm=(   )
Unit Cell - Slab Tanks
v7.1-252
read comp
solution mix=1 rho[uo2f2]=495 92235 93 92238 7
     density=1.566 temp=293 volfrac=1.0 end solution
 ss304 2  1.0  end
 ss304 3  1.0  end
   h2o 4  1.0  end
   h2o 5  1.0  end
end comp
```

```
read celldata
multiregion slab right_bdy=vacuum left_bdy=reflected end
   5 5.0
   3 5.5
   1 10.5
   2 11.0
   4 16.0  end zone
end celldata
read parameters
 npg=10000 nsk=50 gen=1000 sig=50e-5
end parameters
read geometry

global unit 1
  cuboid 1 2.5 -2.5 150.0 -150.0 75.0 -75.0
  cuboid 2 3.0 -3.0 150.5 -150.5 75.5 -75.5
  cuboid 3 8.0 -8.0 150.5 -150.5 75.5 -75.5
'
  media 1 1  1
  media 2 1  2 -1
  media 4 1  3 -2
boundary 3
end geometry
read bounds
  surface(2)=refl
end bounds
end data
end
```

INPUT FOR SECTION 8.5.3

```
=csas6 parm=(   )
Unit Cell - Pebble
v7.1-252

read comp
   uo2 1 den=10.400000 1.0 700.0
        92238 90.4
        92235 9.6 end
' Mixture 2 is porous graphite layer
graphite 2 den=1.050000 1.0 700.0 end
' Mixture 3 is inner pyrolytic carbon layer
graphite 3 den=1.900000 1.0 700.0 end
' Mixture 4 is SiC layer
atomSiC 4 3.18 2
         14000 1
         6000 1
         1.0 700.0
          end
' Mixture 5 is outer pyrolytic carbon layer
graphite 5 den=1.900000 1.0 700.0 end
' Mixture 6 is graphite matrix
graphite 6 den=2.100000 1.0 700.0 end
' Mixture 7 is graphite pebble cladding
graphite 7 den=2.100000 1.0 700.0 end
    he 8 den=0.006100 1.0 700.0 end
```

```
end comp
read celldata
doublehet right_bdy=white fuelmix=10 end
  gfr=0.02397   1
  coatt=0.0095  2
  coatt=0.0040  3
  coatt=0.0035  4
  coatt=0.0040  5
  matrix=6 numpar=15000 end grain
  pebble sphsquarep right_bdy=white left_bdy=reflected
  hpitch=3.05 8 fuelr=2.5 cladr=3.0 7 end
end celldata
read parameters
 npg=10000 nsk=50 gen=10050 sig=50e-5
end parameters
read geometry
global unit 1
  sphere 10 2.5
  sphere 20 3.0
  cuboid 30 3.05 -3.05 3.05 -3.05 3.05 -3.05
'
  media 10 1 10
  media  7 1 20 -10
  media  8 1 30 -20
boundary 30
end geometry
read bounds
 surface(1)=refl surface(2)=refl
 surface(3)=refl surface(4)=refl
 surface(5)=refl surface(6)=refl
end bounds
end data
end
```