KENO V.a Primer: Performing Calculations using SCALE's Criticality Safety Analysis Sequence (CSAS5) 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

Nuclear Energy and Fuel Cycle Division

KENO V.A PRIMER: PERFORMING CALCULATIONS USING SCALE'S CRITICALITY SAFETY ANALYSIS SEQUENCE (CSAS5) WITH FULCRUM

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

December 2020

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
CE	continuous energy
CSAS	Criticality Safety Analysis Sequences
CSEWG	Cross-Section Evaluation Working Group
EALF	energy of the average lethargy causing fission
GEN	number of generations
GUI	Graphical User Interface
HEU	highly enriched uranium
ICSBEP	International Criticality Safety Benchmark Experiments
IEU	intermediate-enriched uranium
INFHOMMEDIUM	infinite homogeneous medium
JRE	JAVA runtime environment
k_{eff}	effective multiplication factor
KMART	KENO Module for Activity-Reaction Rate Tabulation
LEU	low-enriched uranium
LWR	light-water reactor
MG	multigroup
MOX	mixed oxide
NEA	Nuclear Energy Agency
NPG	neutrons per generation
NRC	US Nuclear Regulatory Commission
NSK	number of generations skipped
OECD	Organization for Economic Cooperation and Development
ORNL	Oak Ridge National Laboratory
PMC	produce multigroup cross sections
PWR	pressurized water reactor
RSICC	Radiation Safety Information Computational Center

ABSTRACT

The SCALE code system developed at Oak Ridge National Laboratory is widely used and accepted around the world for criticality safety analyses. The well-known KENO V.a three-dimensional Monte Carlo criticality computer code is one of the primary criticality safety analysis tools in SCALE. The KENO V.a primer is designed to help a new user understand and use the SCALE/KENO V.a Monte Carlo code for nuclear criticality safety analyses. 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 V.a in particular. The primer is designed to teach by example, with each example illustrating two or three features of SCALE/KENO V.a that are useful in criticality analyses.

The primer is based on SCALE 6.2 and 6.3, which includes the Fulcrum graphical user interface (GUI). 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 V.a input and allows the user to quickly run a simple criticality problem with SCALE/KENO V.a. The sections that follow Quickstart include a list of basic objectives at the beginning that identifies the goal of the section and the individual SCALE/KENO V.a features that are covered in detail in the sample problems in that section. Upon completion of the primer, a new user should be comfortable using Fulcrum to set up criticality problems in SCALE/KENO V.a.

The primer provides a starting point for the criticality safety analyst who uses SCALE/KENO V.a. Complete descriptions are provided in the SCALE/KENO V.a manual. Although the primer is self-contained, it is intended as a companion volume to the SCALE/KENO V.a 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 V.a for criticality analyses; the SCALE/KENO V.a manual provides information on the use of SCALE/KENO V.a 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/kenova.

1.   INTRODUCTION AND PURPOSE

1.1   INTENDED AUDIENCE AND AVAILABLE RESOURCES

This primer is designed to help the user understand and use the KENO V.a three-dimensional (3D) Monte Carlo code for nuclear criticality safety analysis. KENO V.a 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 V.a in particular. The primer is designed to teach by example, with each example illustrating two or three features of KENO V.a that are useful in criticality safety analysis. The theory and implementation details of KENO V.a 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 V.a. 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 V.a input and allows the user to quickly run a simple criticality problem with SCALE/KENO V.a. This section is not designed to explain Fulcrum, the input, or the SCALE/KENO V.a 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 V.a 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 V.a. The user will be able to use SCALE/KENO V.a 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 V.a 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 V.a 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 V.a. 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 V.a 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/kenova.

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 V.a *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 V.a 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 a 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 V.a and for 1D criticality safety analysis.

KENO V.a 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.

Note that SCALE includes another version of KENO known as KENO-VI. As the geometry package for KENO-VI is different from that for KENO V.a, a different control module, CSAS6, provides an

automated criticality safety sequence using KENO-VI for performing 3D Monte Carlo analyses of more complex geometries (e.g., hexagonal lattices or nonorthogonal geometry).

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 V.A QUICKSTART

2.1 WHAT YOU WILL BE ABLE TO DO

- Describe the structure of SCALE/KENO V.a input files.

- Use the Fulcrum user interface to create a SCALE/KENO V.a input file.

- Set up and run a simple criticality problem using SCALE/KENO V.a.

- Find and interpret k_{eff} information from SCALE/KENO V.a output.

2.2 SCALE/KENO V.A INPUT FILE

The SCALE/KENO V.a 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 relationships with other objects in a system. Each object can be filled with a material or a void.

A SCALE/KENO V.a input file consists of some or all of the data described above, 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 allow the user to run a simple example problem. It is our intent that the user will gain confidence in using Fulcrum to enter SCALE/KENO V.a input data right away, so this example problem is presented step by step, explaining each input step. For the present, it is important that the user enter this problem exactly as described. As the user gains more experience with Fulcrum and SCALE/KENO V.a, more logical ways to set up input files may occur to the user, such as 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/cm^3) with a coating of nickel (also known as *Jezebel*). Experimental parameters are as follows:

- Delta-phase Pu metal sphere: radius = 6.38493 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
- Spherical nickel coating: thickness = 0.0127 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.
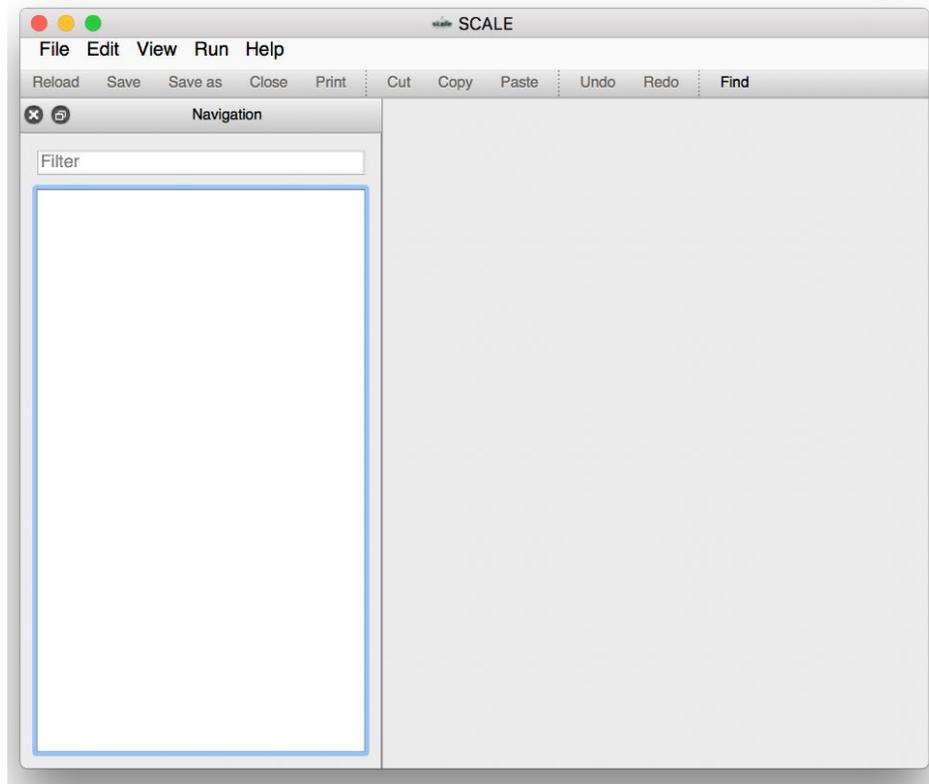
Figure 1. Fulcrum start screen.

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, along with a **.inp** input file extension. Enter the input file name as `jezebel.inp` and click the **Save** button. A blank input document will display as shown in Figure 2.
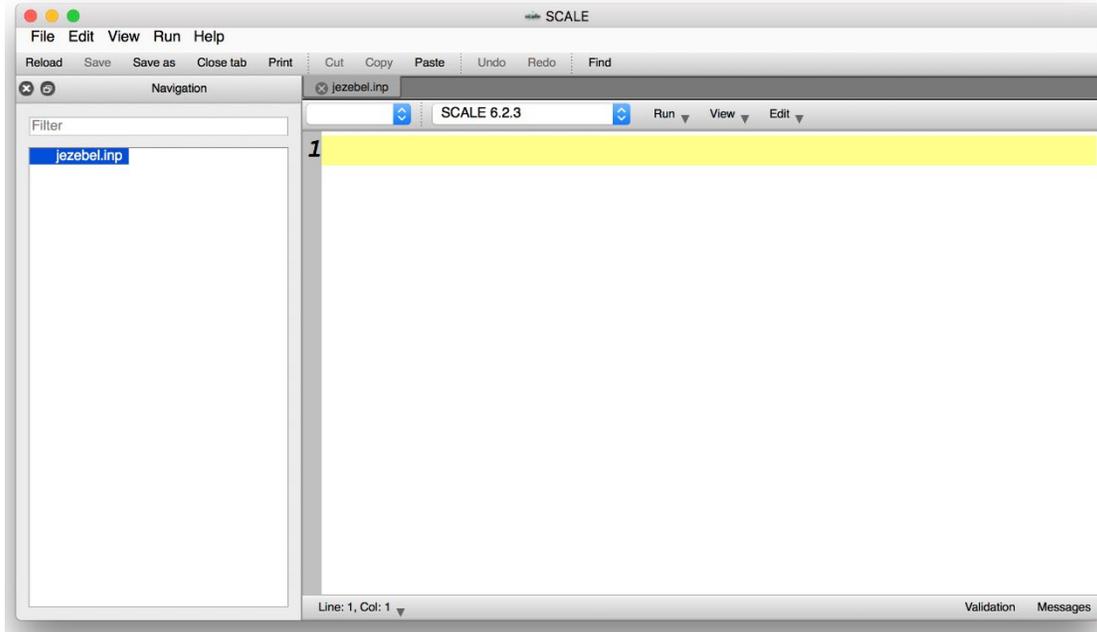
Figure 2. New blank jezebel input document.

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.

Again, note that information 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 **csas5 – Criticality safety analysis using Keno V.a** option as shown in Figure 3.
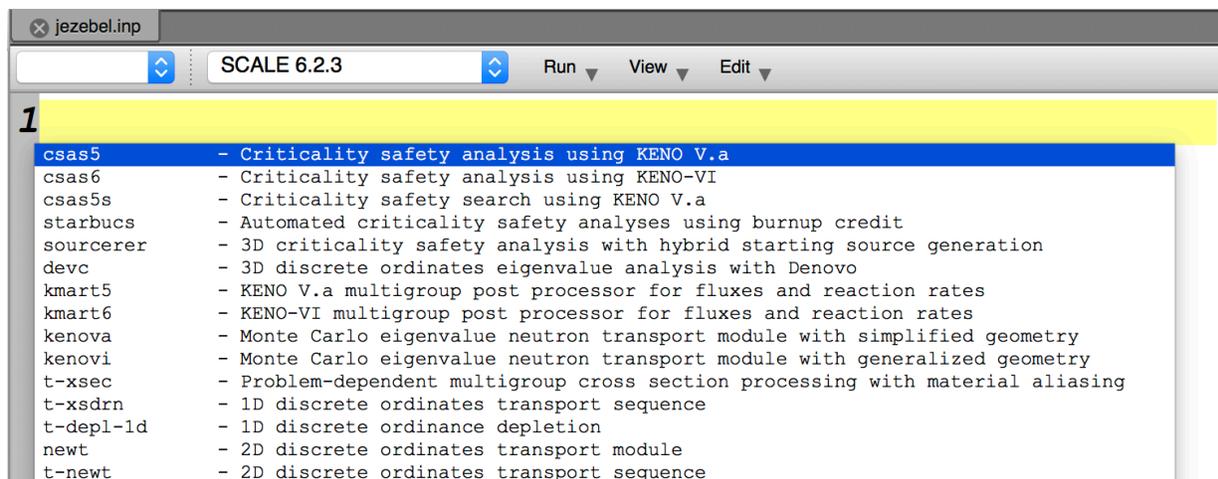
Figure 3. Autocompletion of the example sequence.

A skeleton CSAS5 input is created with placeholders and empty read blocks, as shown in Figure 4. These streamline the additional updates needed for the user to complete the input specification.
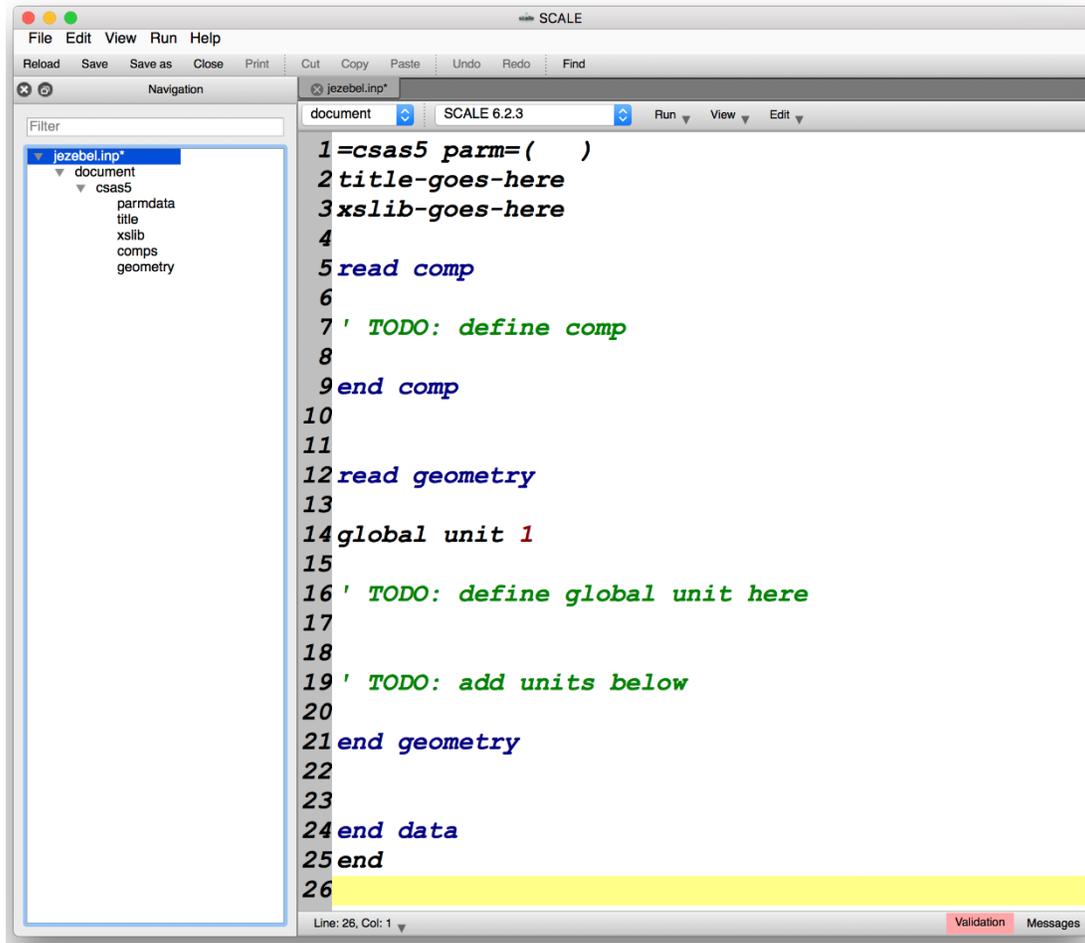
Figure 4. Example problem skeleton input.

With this CSAS5 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 multi-group cross section library, `v7.1-252`.

Figure 5 depicts the CSAS5 input with the title and cross section library entered.
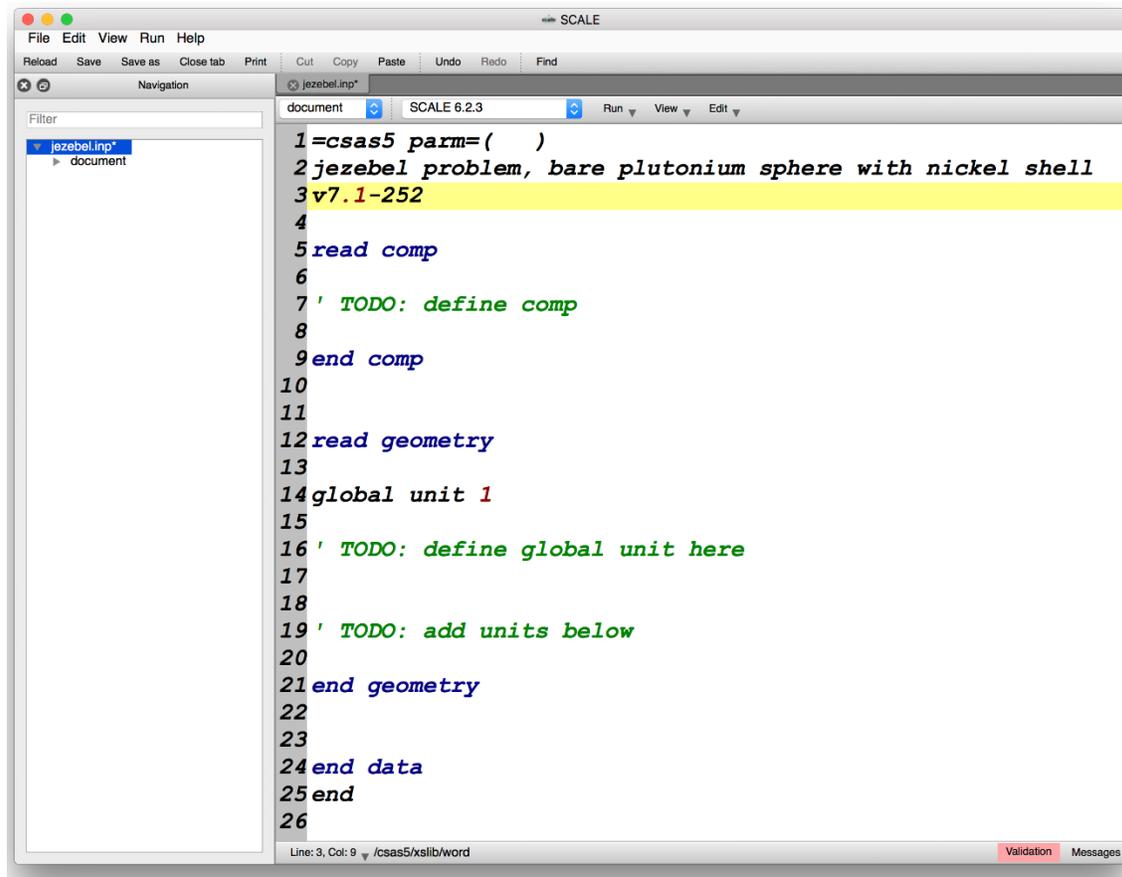
Figure 5. General information for example problem.

The **Validation** panel button is in the lower right of the text editor panel. You will notice it has a red background; this indicates 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.
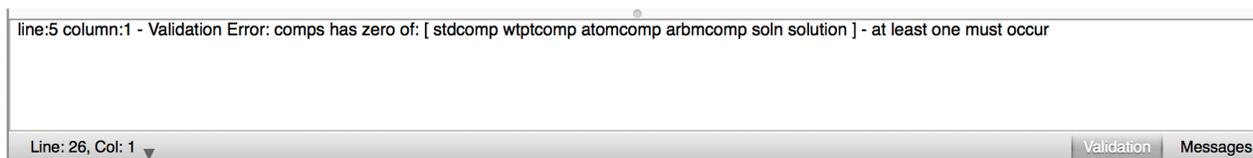
Figure 6. Validation panel indicating required composition are not present.

The validation message is shown below:

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

This 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, or solution. This transitions the user to the **Materials** specification for the example problem.

9

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.
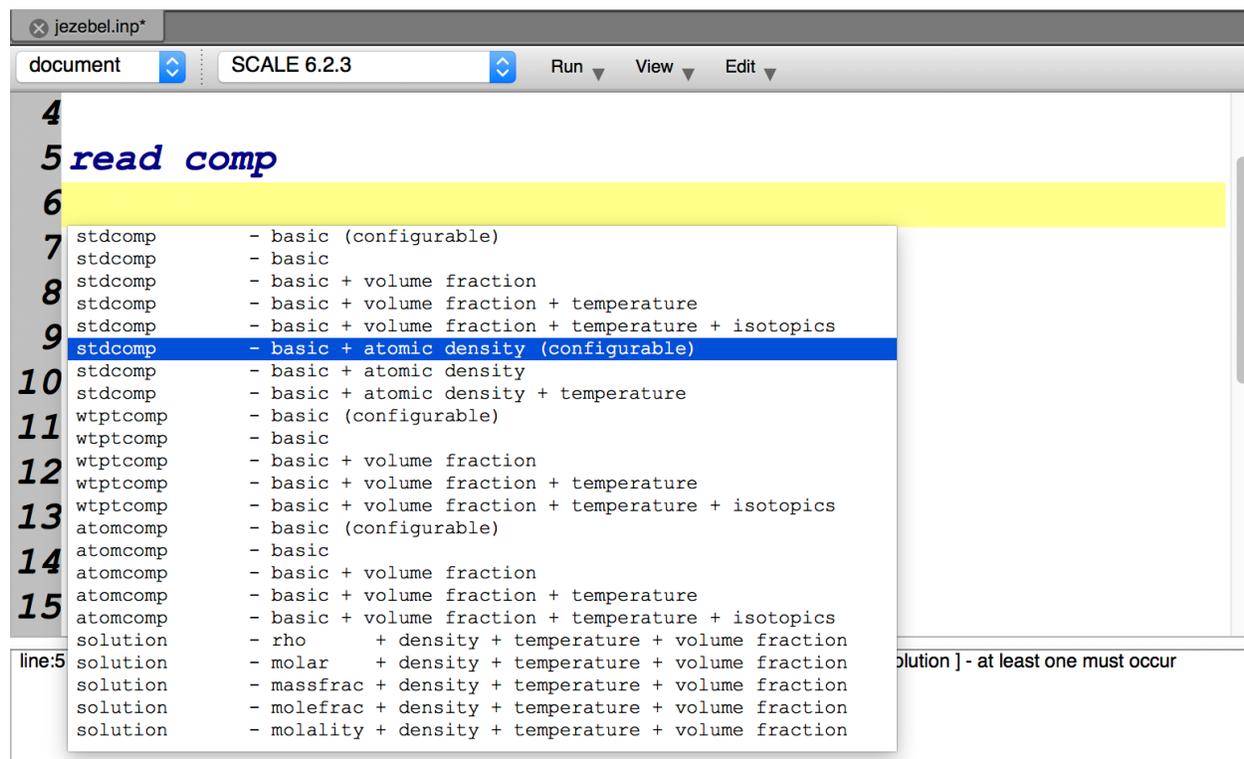
Figure 7. Example problem Basic Composition autocompletion.

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.
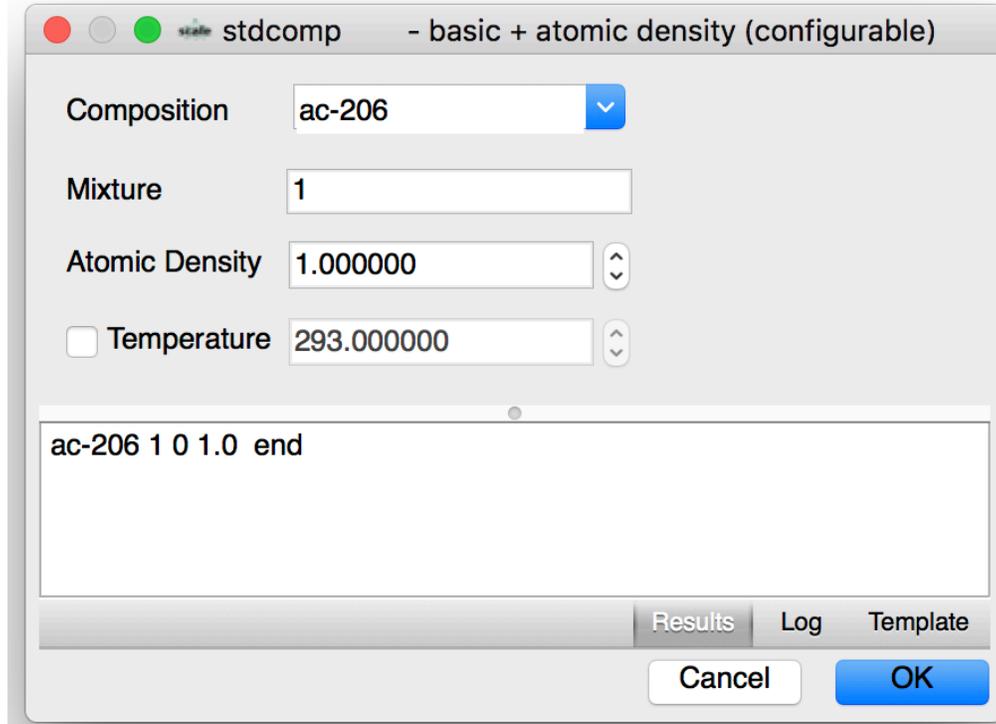
Figure 8. Example default basic stdcomp configurable form.

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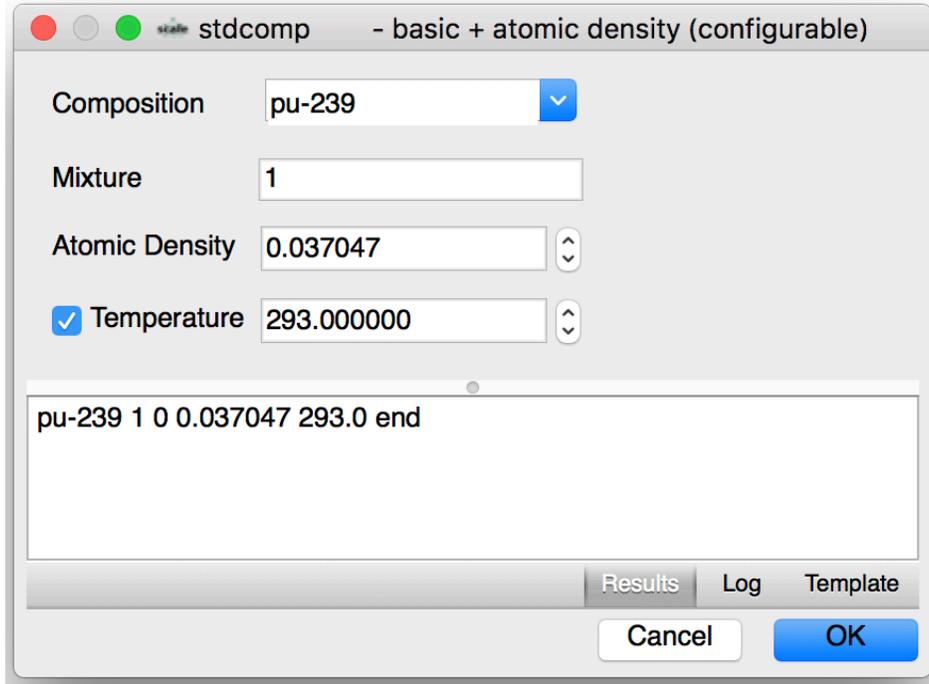
Figure 9. Data for ^{239}Pu in mixture 1.

Select **OK** to insert the **Result** into the document. To enter the data for the remainder of the 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—pu-240, pu-241, and ga—look like those shown in Figure 10 through Figure 12.
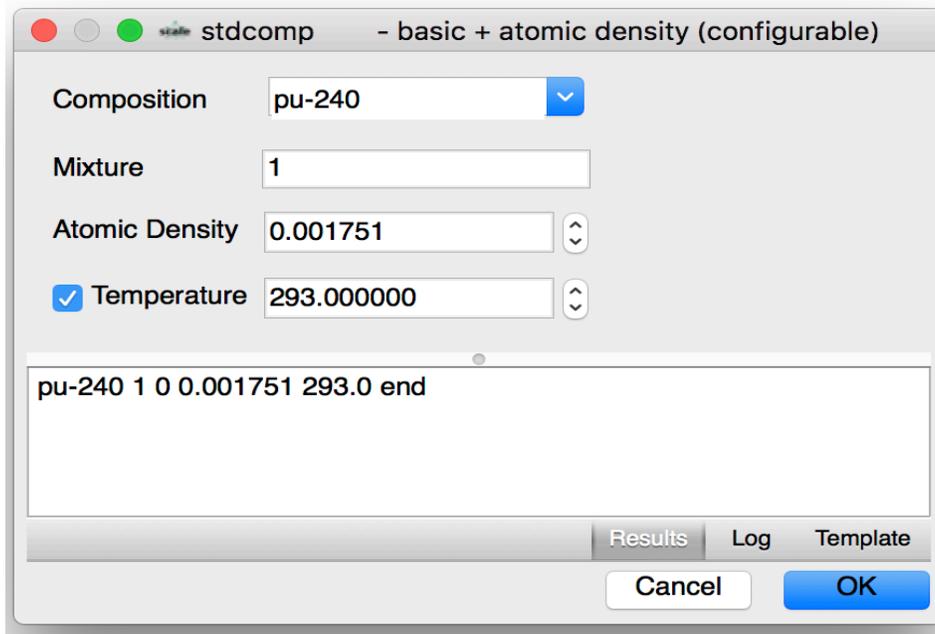
Figure 10. Data for ^{240}Pu in mixture 1.
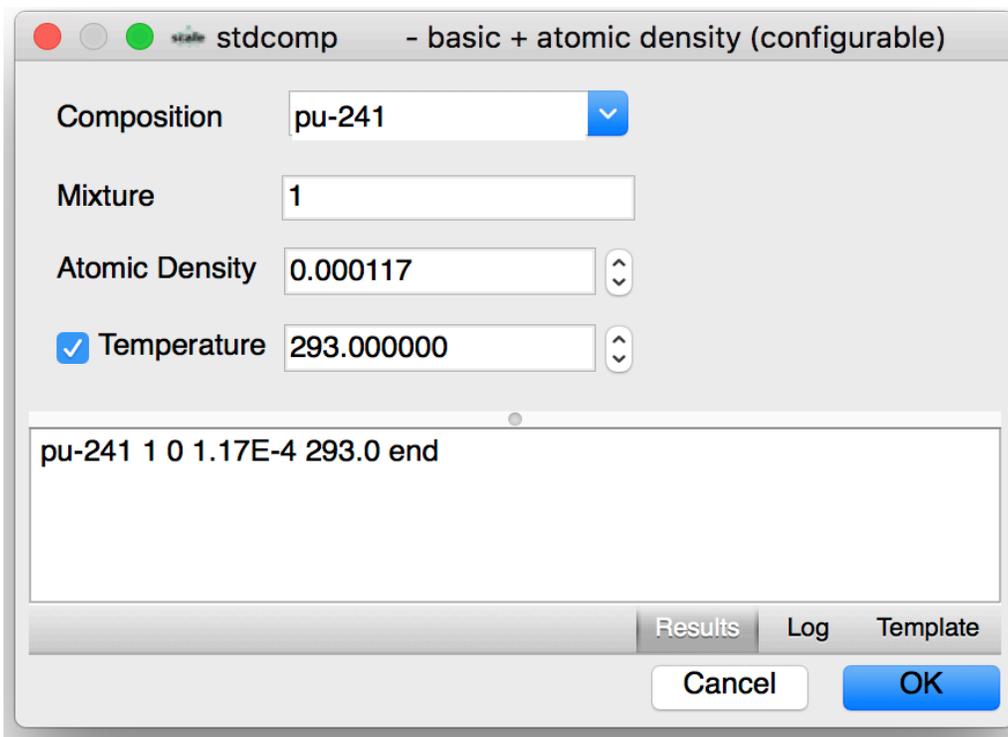
Figure 11. Data for ^{241}Pu in mixture 1.
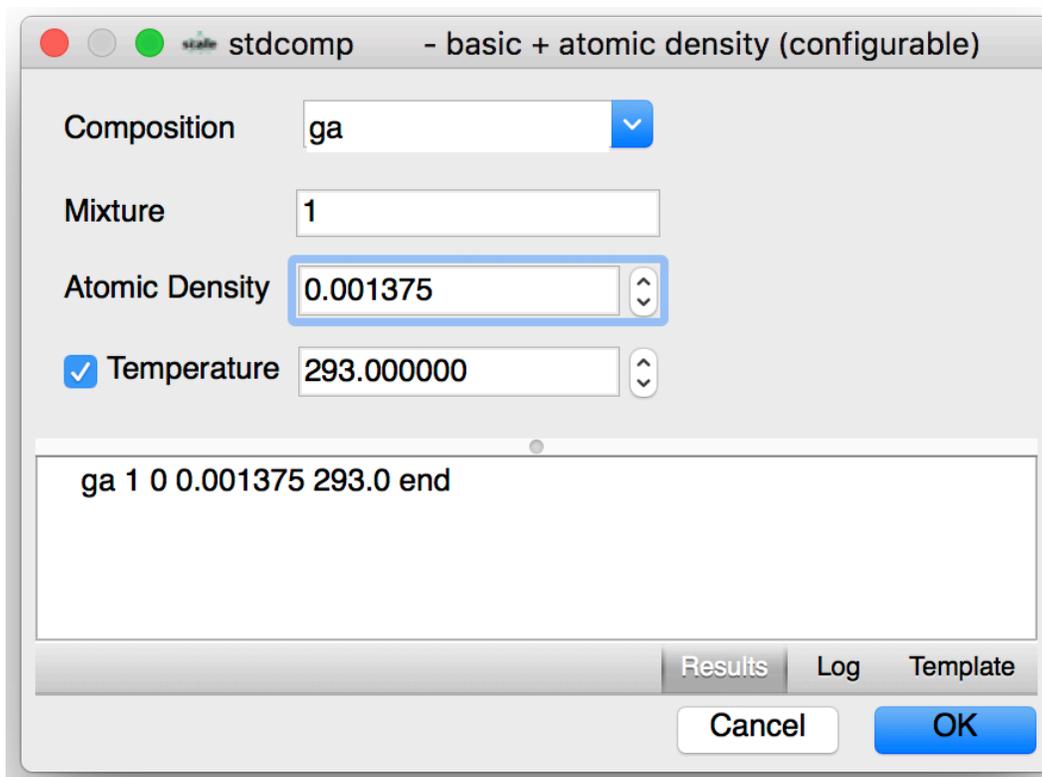
Figure 12. Data for Gallium in mixture 1.

13

Figure 13 illustrates the composition records for mixture 1 in the input file.

Figure 13. Mixture 1 basic composition summary.

Because material input can be position dependent, it is important to remember that Fulcrum can assist you by presenting field labels at any given text cursor position. If you do not remember what the fourth field is in

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

then place your text cursor on the field and observe the cursor context display the parameter's path **/csas5/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.

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 **CSAS5** 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. 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**.**
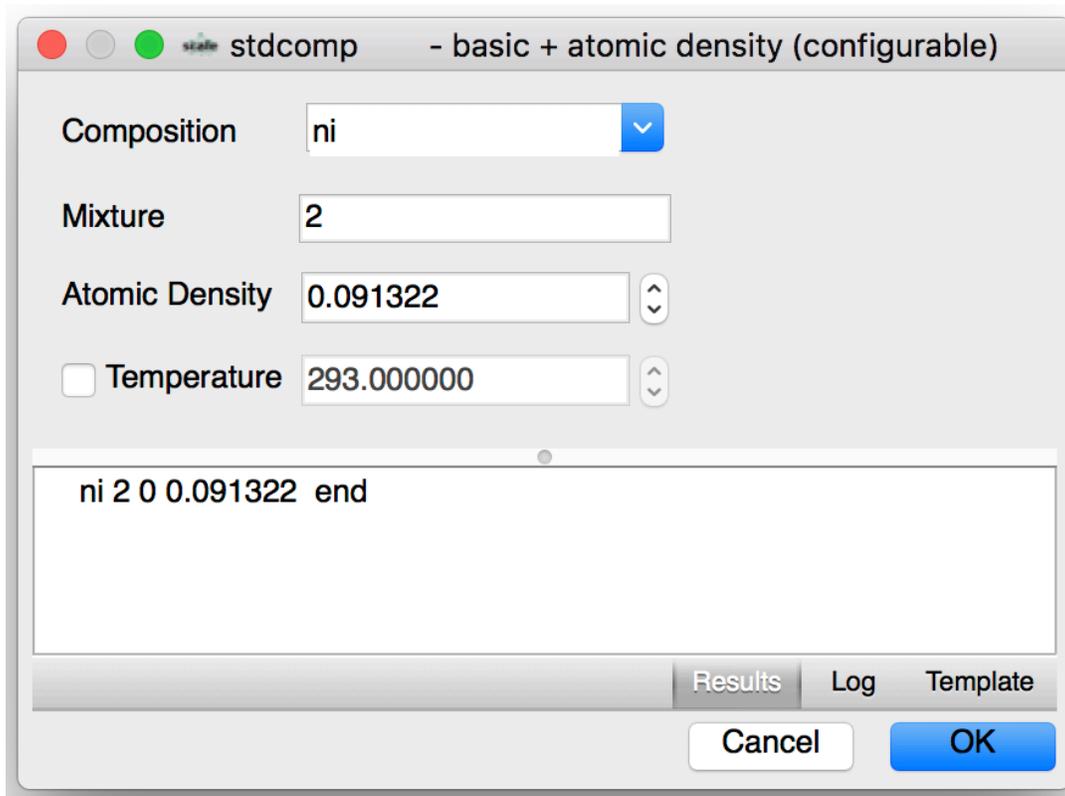
Figure 15. Data for nickel in mixture 2.

Figure 16 illustrates the complete example composition block.

Figure 16. Complete example problem composition block.

Because this example problem uses MG cross sections, you will need to specify a processing method to incorporate spatial affects in the cross section preparation. The unit cell input parameters are specified in the **read celldata** block. With the 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.

Figure 17. Autocompletion of the Cells input block.

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 then 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 default outer boundary of vacuum.
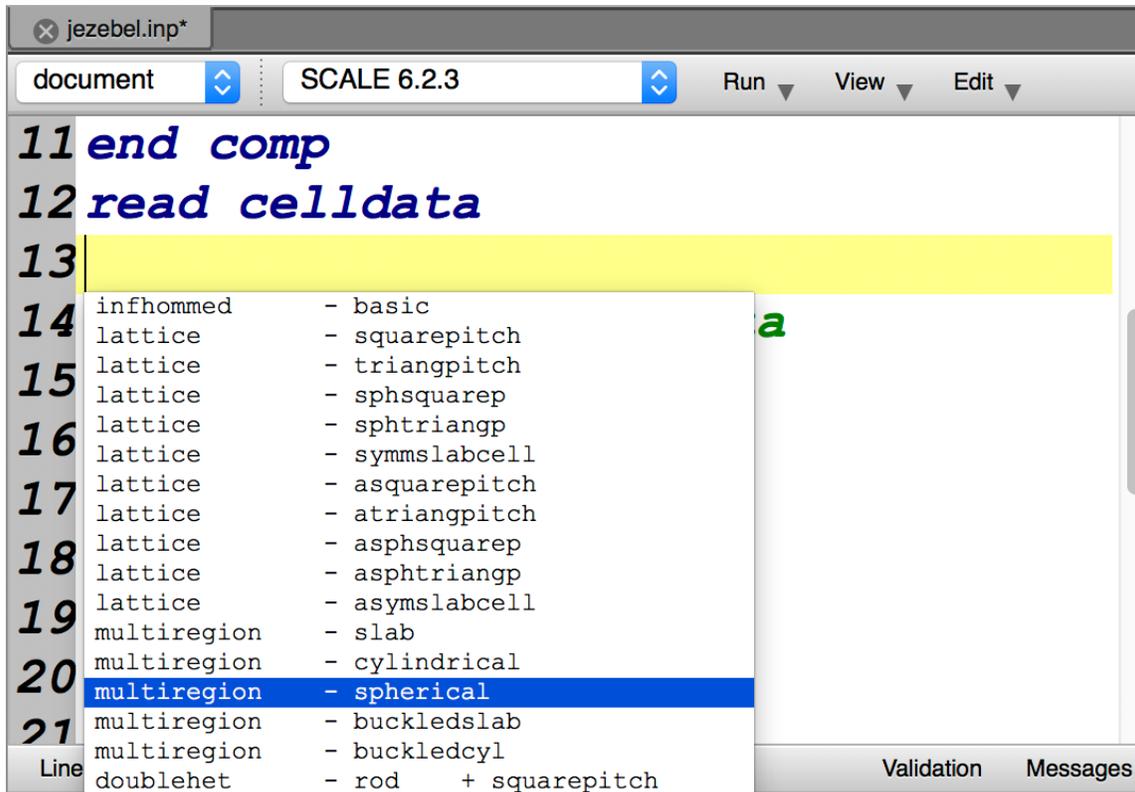
Figure 19. Autocompletion of a spherical multiregion.
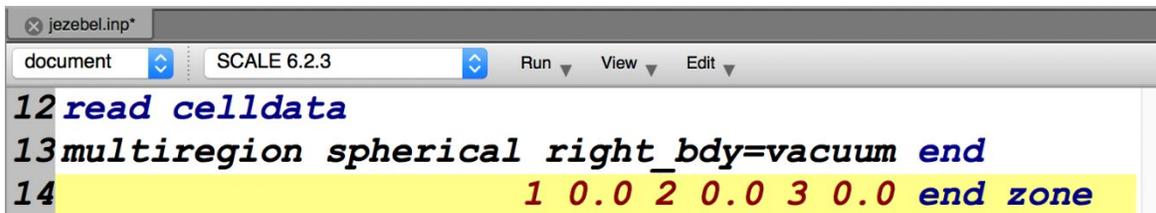
Figure 20. Example autocompleted initial spherical multiregion input.

Remember that 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.**
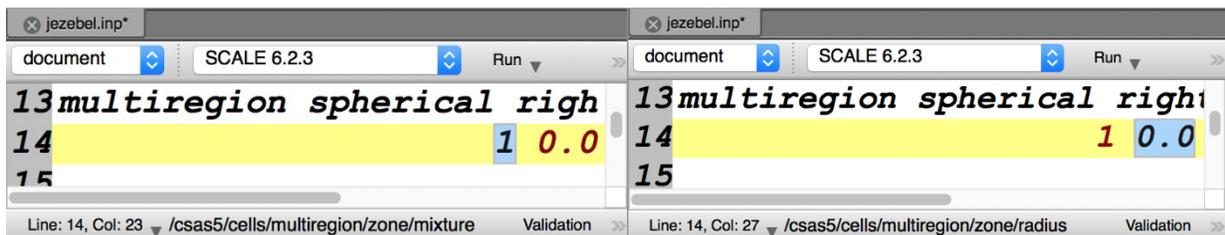
Figure 21. Example spherical multiregion zone mixture and radius placeholders.

The zone mixture-radii pair placeholders must 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 placeholders for mixture 3 can be removed. Figure 22 illustrates the completed example **multiregion** input.
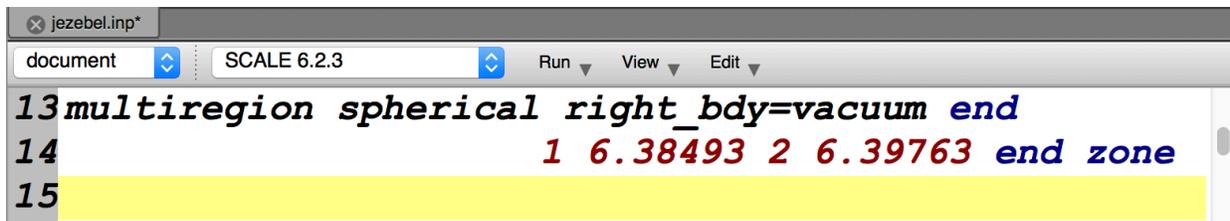
18

Figure 22. Multiregion unit cell.

2.3.4   KENO V.a

Now enter the information required to run a KENO V.a analysis of the problem. Place the cursor inside the **read geometry** block and inside **global unit 1** to prepare for autocompletion of the appropriate geometry for this problem, as shown in Figure 23.
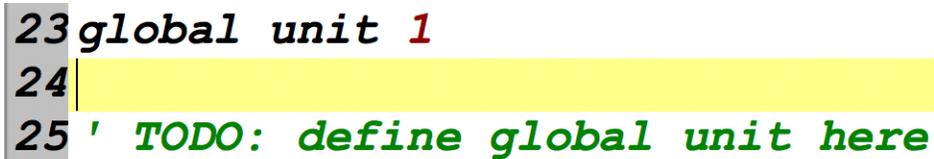
Figure 23. Cursor placement in the global unit.

Press the CTRL-SPACE key combination to display the available geometry, as shown in Figure 24.

```
sphere          - kenova (configurable)
sphere
hemisphere      - kenova (configurable)
hemisphere
hemisphe+x
hemisphe-x
hemisphe+y
hemisphe-y
hemisphe+z
hemisphe-z
xcylinder       - kenova (configurable)
xcylinder
xhemicyl+y
xhemicyl-y
xhemicyl+z
xhemicyl-z
ycylinder       - kenova (configurable)
ycylinder
yhemicyl+x
yhemicyl-x
yhemicyl+z
yhemicyl-z
cylinder
zcylinder       - kenova (configurable)
zcylinder
zhemicyl+x
zhemicyl-x
zhemicyl+y
zhemicyl-y
cube            - kenova (configurable)
cube
cuboid          - kenova (configurable)
cuboid
array           - kenova (configurable)
array
hole            - kenova (configurable)
hole
```

Figure 24. Available autocomplete options.

For this problem the appropriate geometry is a sphere, so select the **sphere – kenova (configurable)** option. This creates a configurable form for a sphere, as shown in Figure 25.
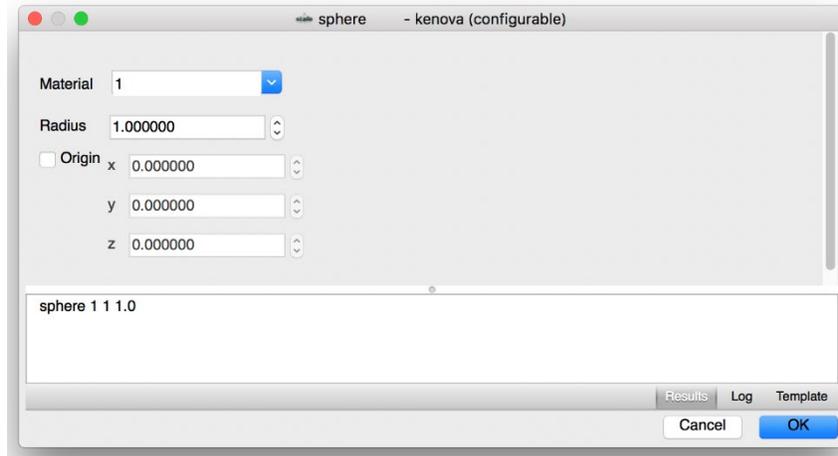
Figure 25. Initial sphere configurable.

Select the appropriate mixture, 1 (pu-239, pu-240, pu-241, and ga), enter the radius of `6.38493`, and then select **OK**. An additional sphere is needed for the nickel outer shell, so copy and paste the sphere and update the mixture to `2` and the radius to `6.39763`, as shown in Figure 26.

Figure 26. Complete problem geometry.

Make sure the file is saved by clicking the **File > Save.**

2.4 RUNNING SCALE/KENO V.A

To run SCALE/KENO V.a, click the **Run** button on the text editor panel's toolbar depicted in Figure 27.

Figure 27. 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 28.
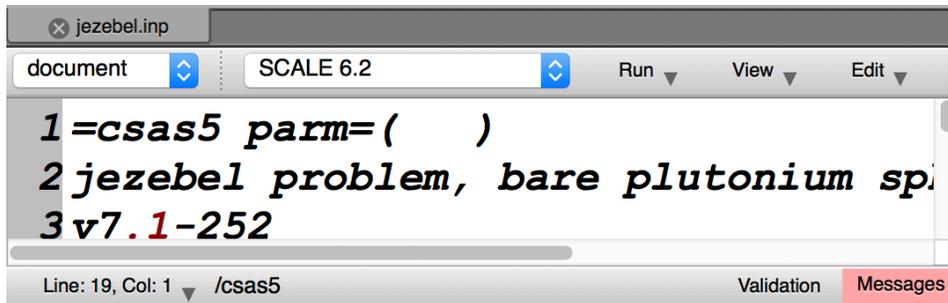
Figure 28. Location of Messages panel and red background color indicating unseen messages.

Click the **Messages** panel to see the jezebel.inp execution messages as depicted in Figure 29.
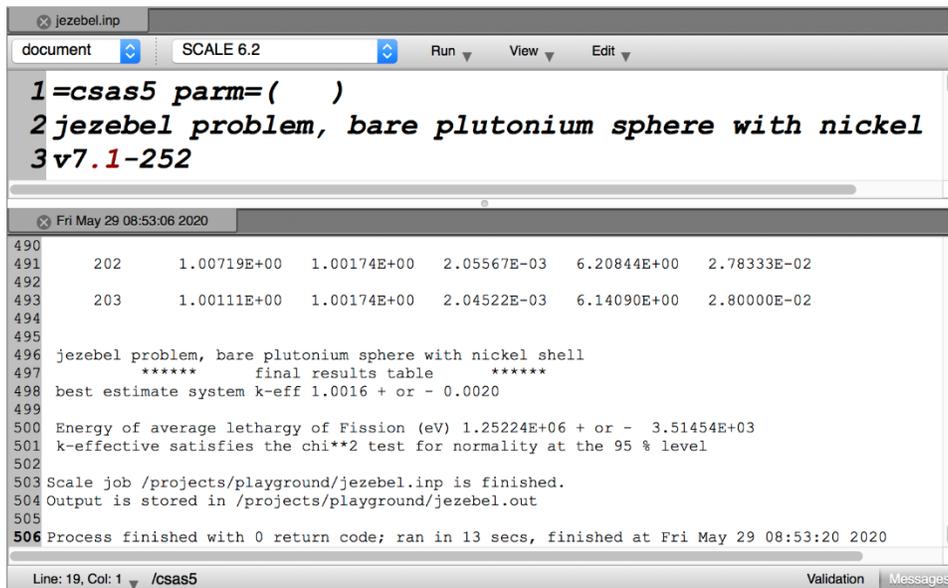
Figure 29. Messages panel showing completed run.

Note that the messages provide a running summary of results and should indicate that the **SCALE job is finished.** They also indicate that the output is stored in **jezebel.out**. If a different name was used for the 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 V.A 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 30.
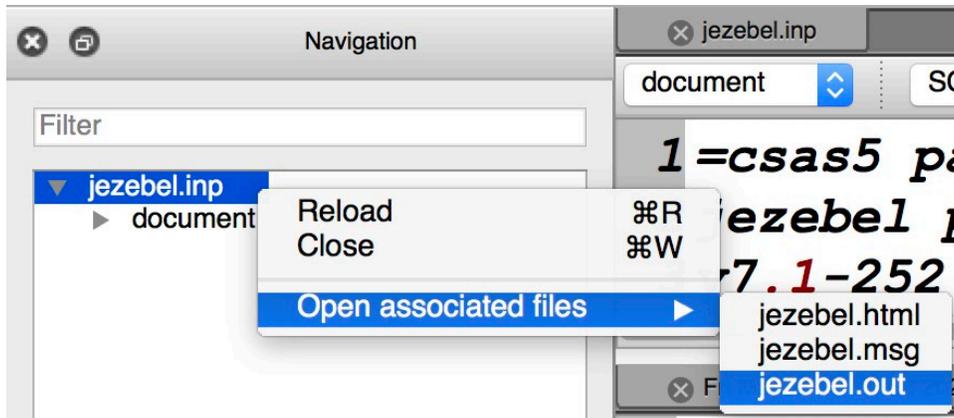
Figure 30. Opening the output file.

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

For this *Quickstart* section, a few items in the KENO information section will be reviewed. The first item is the information following the line **keno messages number k5-123.** Click the **Find** button on the Fulcrum toolbar and enter k5-123. 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). Below the messages are lines with 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 31.
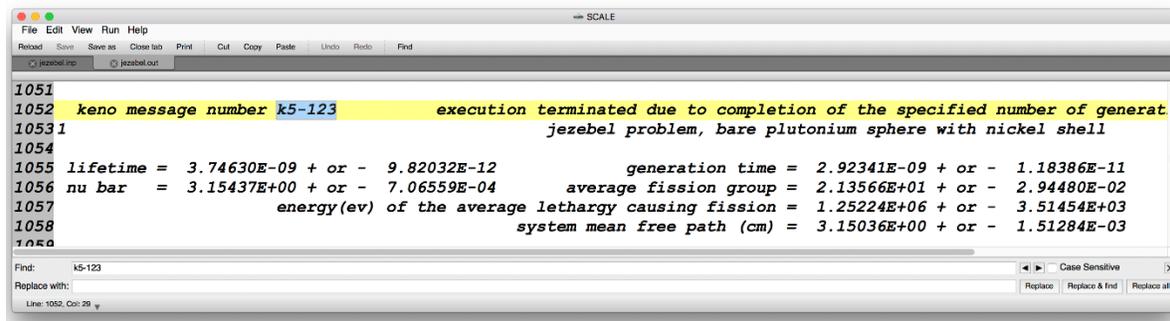
Figure 31. KENO output for Jezebel problem.

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 1.0016 ± 0.0020, as shown in Figure 32.

Figure 32. KENO final results table.

2.6 SUMMARY

This section has helped you to:

- Describe the structure of SCALE/KENO V.a 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 V.a input file.

- Set up and run a simple criticality problem using SCALE/KENO V.a.

- Find and interpret k_{eff} information from your output.

Now that you have successfully run SCALE/KENO V.a, you are ready to learn the options available in each input segment in detail and how to set up more complex problems. The following sections 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 CSAS5 sequence, and [4] is available for CSAS6 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 33 and will result in the blank input file shown in Figure 34.
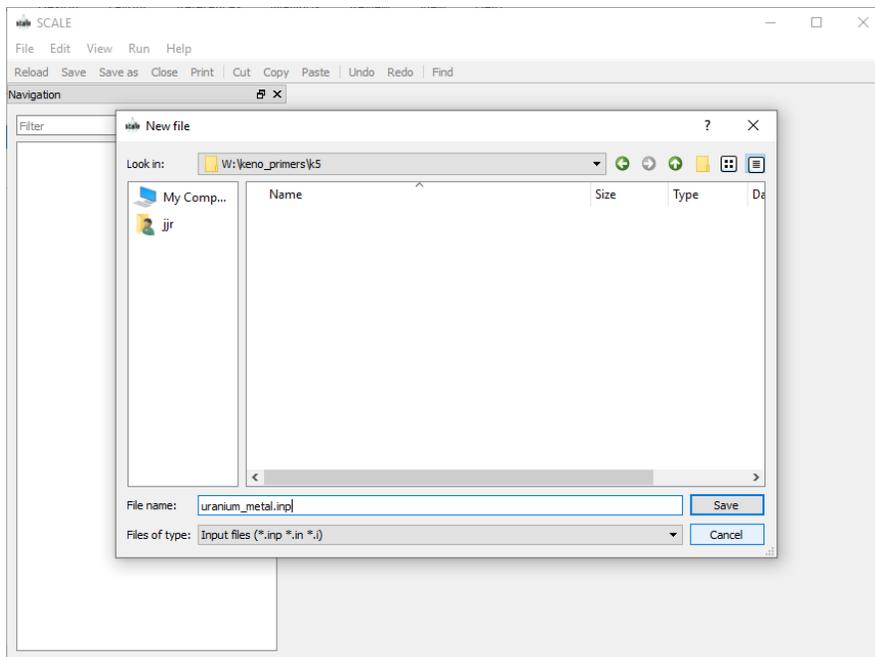
Figure 33. Depiction of the definition of the input file name.
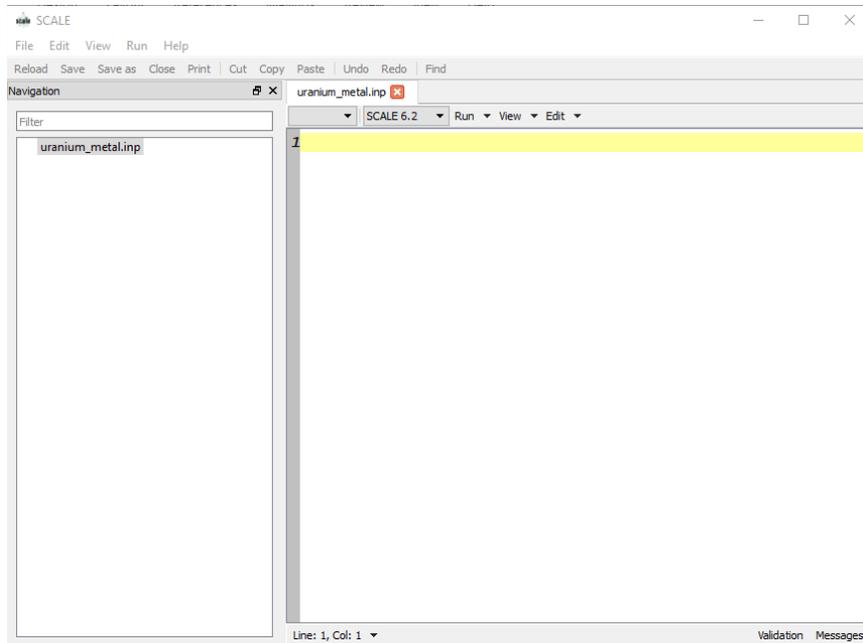
Figure 34. Empty input file after file specification in Fulcrum.

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 CSAS5 sequence from the dropdown menu and press **enter**, as shown in Figure 35. Selecting a sequence should result in a CSAS5 input skeleton, as is shown in Figure 36.
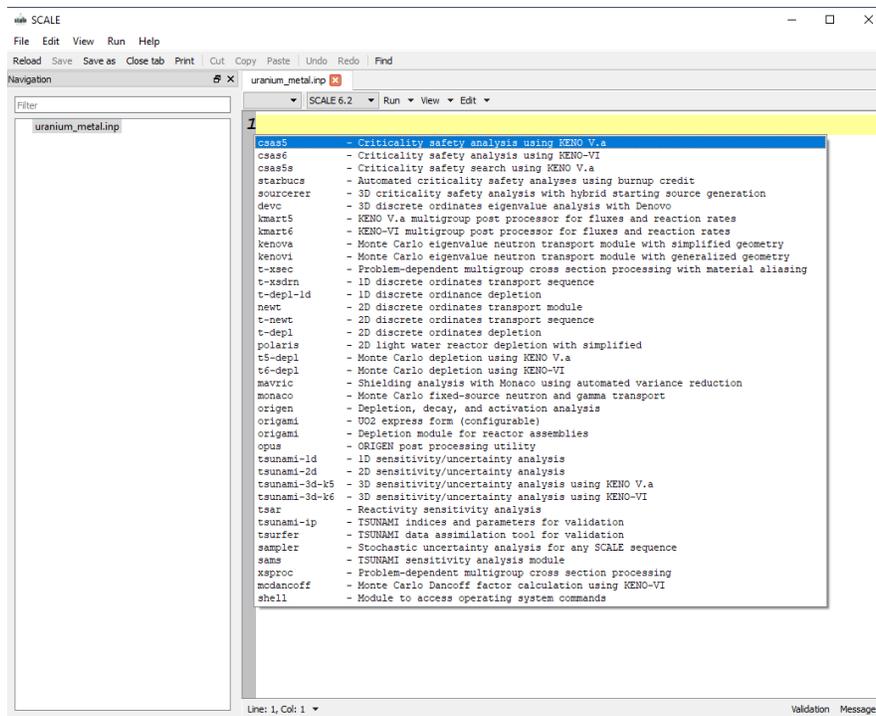
Figure 35. Dropdown menu with SCALE sequence specification.
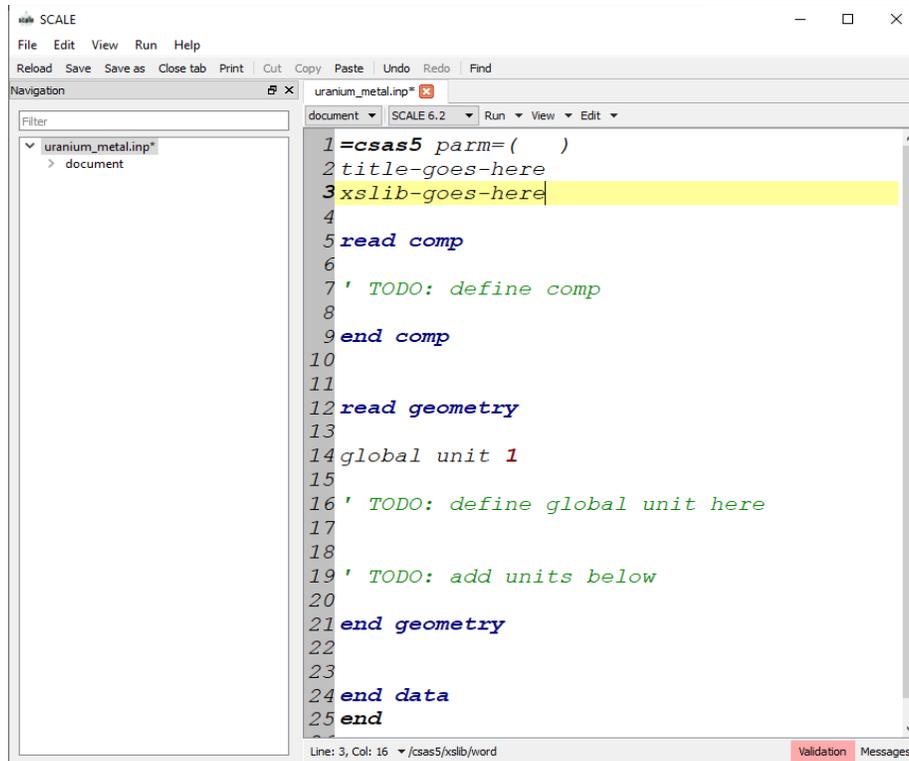
Figure 36. Empty input file resulting from CSAS5 sequence specification.

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 37. 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 37.

Figure 37. 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/cm^3) 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/cm^3)

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 38.
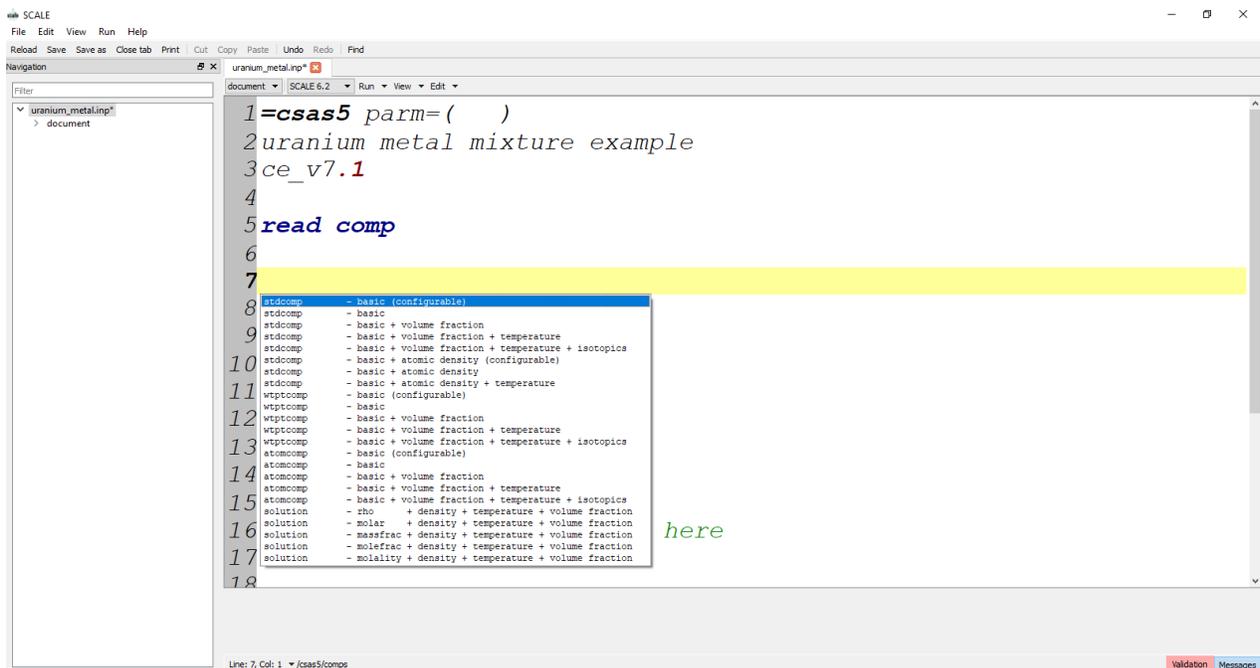
Figure 38. Accessing the basic standard composition configurable.

From the dropdown menu, select **stdcomp – basic (configurable)**. This will generate a fillable form, as shown in Figure 39.

Figure 39. Basic standard composition configurable input initial display.

In the **composition** dropdown, click 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 40.

Figure 40. Basic standard composition configurable input initial 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 41.

Figure 41. 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 V.a 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 42.
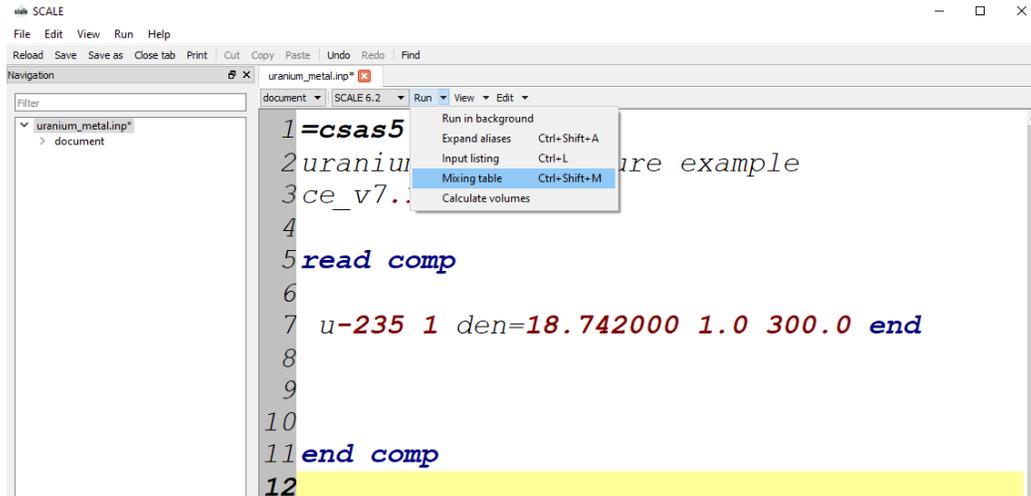
Figure 42. Dropdown menu necessary to create mixing table.

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 43 (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 HEU metal problems is covered in Section 4.2.
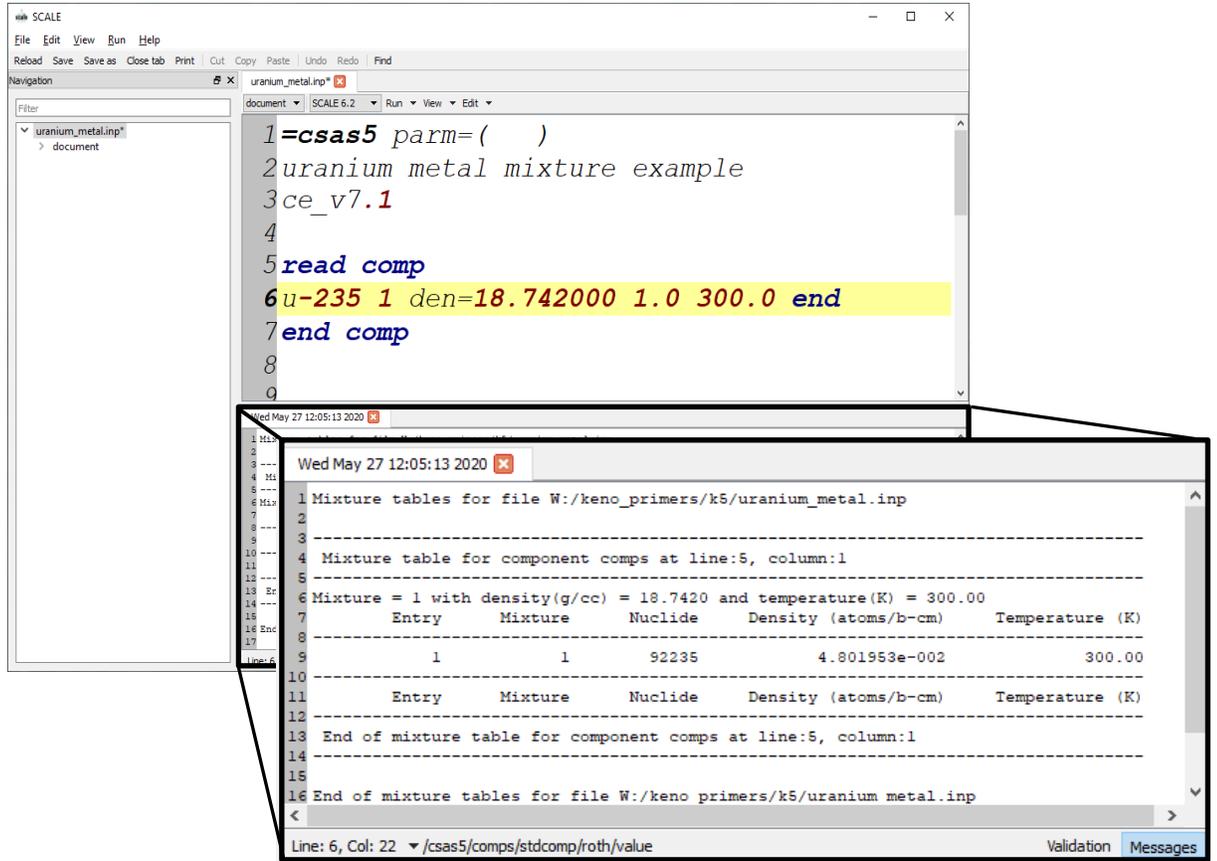
Figure 43. Mixing table results for uranium metal example.

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/cm^3)

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 CSAS5 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 38. From the dropdown menu, select again **stdcomp – basic (configurable)**. This will generate a fillable form, as was shown in Figure 39. 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 44. Note that you may have to expand the window to see the **Add row** button.

Figure 44. 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 45.

Figure 45. 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 46.

Figure 46. 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 V.a 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 42.

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 47. 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 4.2.1.

36

```
 3 ---------------------------------------------------------------------
 4  Mixture table for component comps at line:5, column:1
 5 ---------------------------------------------------------------------
 6 Mixture = 1 with density(g/cc) = 18.7420 and temperature(K) = 300.00
 7         Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
 8 ---------------------------------------------------------------------
 9            1            1        92235            4.499910e-002            300.00
10            2            1        92238            2.982277e-003            300.00
11 ---------------------------------------------------------------------
12         Entry      Mixture      Nuclide     Density (atoms/b-cm)     Temperature (K)
13 ---------------------------------------------------------------------
14  End of mixture table for component comps at line:5, column:1
15 ---------------------------------------------------------------------
```

Figure 47. Mixing table results for uranium metal example.

3.4.3 Highly Enriched UO₂

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/cm^3.

Core Material (ρ_{mix} = 95% of theoretical density – TD = 10.96 g/cm^3)

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 38 above. From the dropdown menu, select **stdcomp – basic (configurable)**. This will generate a configurable form, as shown in Figure 39. 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 48. 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 49.
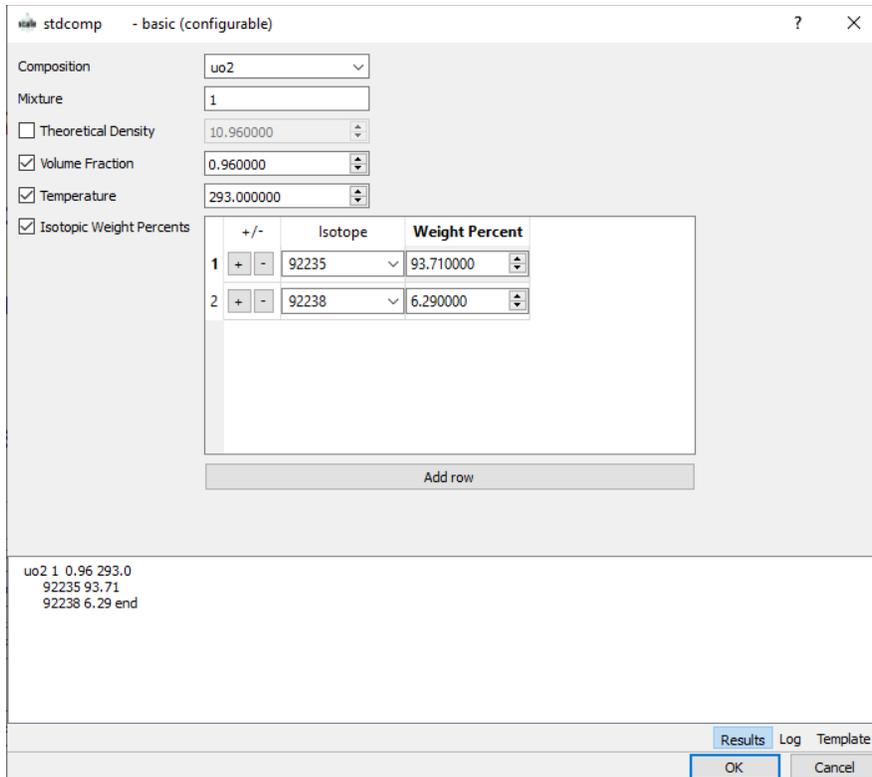
Figure 48. Isotopic composition input following user modification for the enriched uranium oxide case.

Figure 49. 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 V.a 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 42.

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 54 (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).

```
 3 ----------------------------------------------------------------
 4  Mixture table for component comps at line:5, column:1
 5 ----------------------------------------------------------------
 6 Mixture = 1 with density(g/cc) = 10.5216 and temperature(K) = 293.00
 7        Entry      Mixture      Nuclide      Density (atoms/b-cm)      Temperature (K)
 8 ----------------------------------------------------------------
 9          1           1          8016          4.730657e-002              293.00
10          2           1          8017          1.802009e-005              293.00
11          3           1          8018          9.721476e-005              293.00
12          4           1         92235          2.223716e-002              293.00
13          5           1         92238          1.473748e-003              293.00
14 ----------------------------------------------------------------
15        Entry      Mixture      Nuclide      Density (atoms/b-cm)      Temperature (K)
16 ----------------------------------------------------------------
17  End of mixture table for component comps at line:5, column:1
18 ----------------------------------------------------------------
```

Figure 50. Mixing table results for highly enriched UO$_2$ case.

3.4.4   Highly Enriched U(30.3)O$_2$F$_2$ 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$_2$F$_2$ 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-235} = 0.0005637 atoms-^{235}U/b-cm
N_{U-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$_2$F$_2$ composition. Move the cursor into the **read comp** block and press `CTRL-SPACE` to generate the dropdown menu shown in Figure 51. 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 52 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 53.

Figure 51. Mixing table results for highly enriched UO₂ case.

Figure 52. Autocompleted template for atomic density input.

Figure 53. Autocompleted template for 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 V.a 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 42.

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 54. 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).

```
 3  ----------------------------------------------------------------------------
 4  Mixture table for component comps at line:5, column:1
 5  ----------------------------------------------------------------------------
 6 Mixture = 1 with density(g/cc) = 1.8339 and temperature(K) = 293.00
 7       Entry        Mixture       Nuclide      Density (atoms/b-cm)     Temperature (K)
 8  ----------------------------------------------------------------------------
 9          1            1           1001          5.974533e-002            293.00
10          2            1           1002          6.871634e-006            293.00
11          3            1           8016          3.347895e-002            293.00
12          4            1           8017          1.275285e-005            293.00
13          5            1           8018          6.879906e-005            293.00
14          6            1           9019          3.684400e-003            293.00
15          7            1           92235         5.637000e-004            293.00
16          8            1           92238         1.280200e-003            293.00
17  ----------------------------------------------------------------------------
18       Entry        Mixture       Nuclide      Density (atoms/b-cm)     Temperature (K)
19  ----------------------------------------------------------------------------
20  End of mixture table for component comps at line:5, column:1
21  ----------------------------------------------------------------------------
```

Figure 54. Mixing table results for 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 6. 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 38 above. From the dropdown menu, select **stdcomp – basic (configurable)**. This will generate a configurable form, as shown in Figure 39. 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$_2$O 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$_2$O 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$_2$O have the same mixture number. 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. Note that this is the value given above for the density. At this point, the input should look like Figure 55.

```
5 read comp
6     uo2 1   0.2    end
7     h2o 1   0.8    end
8 end comp
```

Figure 55. 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$_2$O. 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$_2$O 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 56 and Figure 57, and the final input, including both the mixing by volume fraction and mixing by weight fraction, is shown in Figure 58.
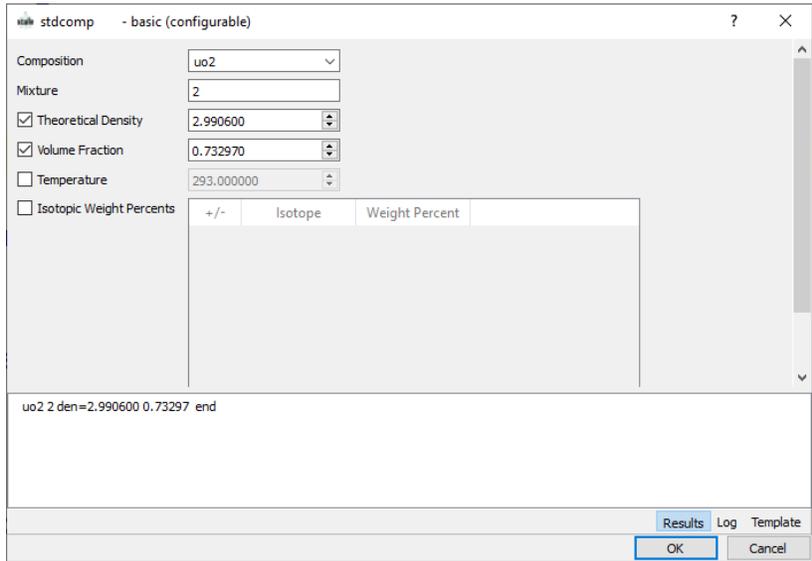
Figure 56. UO$_2$ configurable form for mixing by weight fraction.
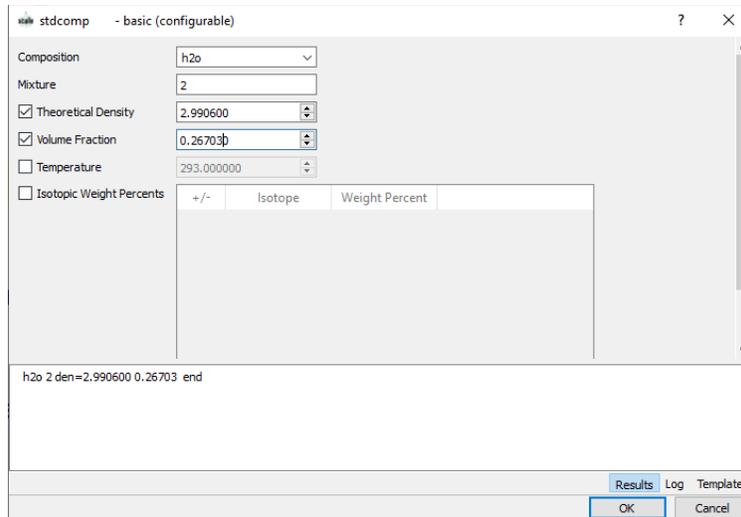
Figure 57. H₂O configurable form for mixing by weight fraction.

```
 5 read comp
 6     uo2 1   0.2   end
 7     h2o 1   0.8   end
 8 '
 9     uo2 2 den=2.990600 0.73297   end
10     h2o 2 den=2.990600 0.26703   end
11 end comp
```

Figure 58. 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 V.a 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 42.

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 59. 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).

```
3 -----------------------------------------------------------------------
4 Mixture table for component comps at line:5, column:1
5 -----------------------------------------------------------------------
6 Mixture = 1 with density(g/cc) = 2.9906 and temperature(K) = 293.00
7      Entry      Mixture     Nuclide    Density (atoms/b-cm)    Temperature (K)
8 -----------------------------------------------------------------------
9         1           1         1001         5.338230e-002           293.00
10        2           1         1002         6.139787e-006           293.00
11        3           1         8016         3.638276e-002           293.00
12        4           1         8017         1.385897e-005           293.00
13        5           1         8018         7.476639e-005           293.00
14        6           1        92234         2.640071e-007           293.00
15        7           1        92235         3.521737e-005           293.00
16        8           1        92238         4.853104e-003           293.00
17 -----------------------------------------------------------------------
18 Mixture = 2 with density(g/cc) = 2.9906 and temperature(K) = 293.00
19      Entry      Mixture     Nuclide    Density (atoms/b-cm)    Temperature (K)
20 -----------------------------------------------------------------------
21        9           2         1001         5.338363e-002           293.00
22       10           2         1002         6.139941e-006           293.00
23       11           2         8016         3.638352e-002           293.00
24       12           2         8017         1.385926e-005           293.00
25       13           2         8018         7.476794e-005           293.00
26       14           2        92234         2.640095e-007           293.00
27       15           2        92235         3.521769e-005           293.00
28       16           2        92238         4.853148e-003           293.00
29 -----------------------------------------------------------------------
30      Entry      Mixture     Nuclide    Density (atoms/b-cm)    Temperature (K)
31 -----------------------------------------------------------------------
32 End of mixture table for component comps at line:5, column:1
33 -----------------------------------------------------------------------
```

Figure 59. Mixing table output for the mixing by volume fraction and 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 the *Quickstart* section, you modeled the simple geometry of a bare sphere with the Fulcrum interface. In the *Material Information Input*, you modeled various material compositions using the Fulcrum user interface to provide data on elements, isotopes, compounds, and solutions. This section presents an explanation of the commands used for simple geometries such as cylinders and spheres and for more complicated geometries such as reflected systems and arrays.

4.1   WHAT YOU WILL BE ABLE TO DO

- Use Fulcrum to describe the basic shapes (sphere, cylinder, cube, cuboid).
- Understand how units are created (including nesting 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 different sizes.

4.2   BASIC GEOMETRY SHAPES

SCALE/KENO V.a uses a set of basic shapes to build geometry models. You used the Fulcrum user interface to enter data on a sphere in Section 2. The sphere and other basic shapes are described in Table 2 and are illustrated in Figure 60. KENO V.a also supports a REPLICATE command that repeats the previous shape with the specified additional thickness on each dimension.

Table 2. Basic shapes

Keyword	Description
CUBE	Box with equal X, Y, and Z dimensions
CUBOID	Box or rectangular parallelepiped; X, Y, and Z dimensions are not necessarily equal
SPHERE	Sphere
CYLINDER	Cylinder that has its length along the X, Y, or Z axis
HEMISPHERE	Partial sphere with a flat surface parallel to the X, Y, or Z axis
HEMICYL	(Hemicylinder) Cylindrical segment whose axis is in the X, Y, or Z direction with a flat surface parallel to the cylinder axis
REPLICATE	Repeat the previous shape with specified additional thicknesses in each dimension

Cube

Cuboid

Sphere

Cylinder

Hemisphere

Hemicylinder
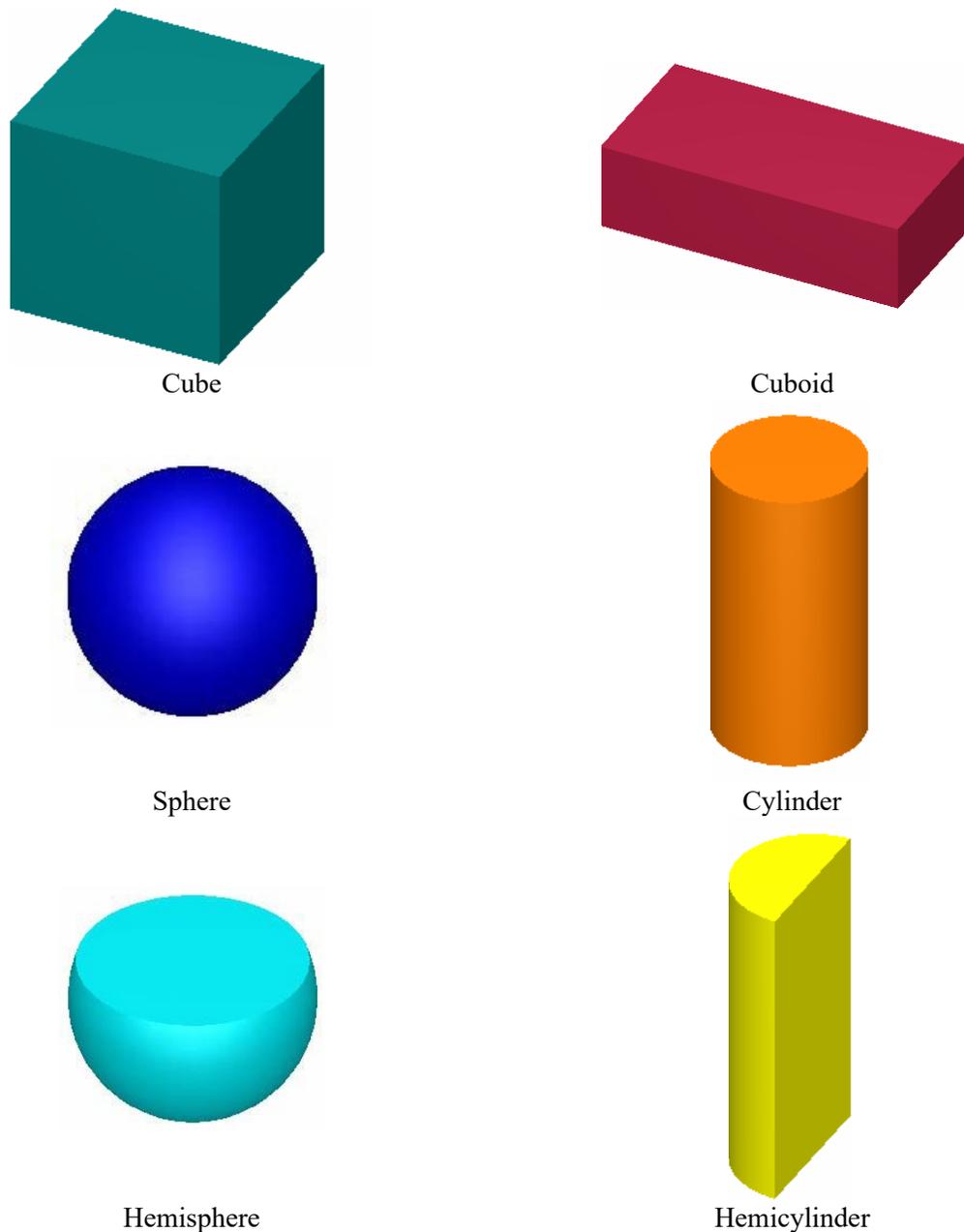
Figure 60. KENO V.a basic geometry shapes.

These six shapes are the basic geometry options in KENO V.a. Each shape has a set of quantitative information needed to describe its size and location. The geometry input type is a fixed length parameter list; each shape entry needs a shape name, a mixture number, a bias identification, the respective dimensions, and, for some, the option for identifying the placement of the shape's origin. For a cuboid, the **+X**, **-X**, **+Y**, **-Y**, **+Z**, and **-Z** data are necessary to specify the shape. Because a cube is a cuboid with equal sides, only **+X** and **-X** need to be specified for this shape. For a sphere, only the radius is required, and a cylinder requires a radius, a top, and a bottom. Figure 61 through Figure 65 show the shapes and respective dimensions necessary for the input. The cuboid/cube are the only shapes that do not have the option of inputting a specific origin. Figure 66 shows how the dimensions can be used to determine the cuboid/cube shape's origin. The axis of a cylinder can be specified using the ORIGIN keyword, as shown in Figure 67, but the top and bottom surfaces must be specified at the appropriate locations.

CUBOID   Mixture BiasID +x -x +y -y +z -z
 or for cubes:
CUBE   Mixture BiasID +x -x
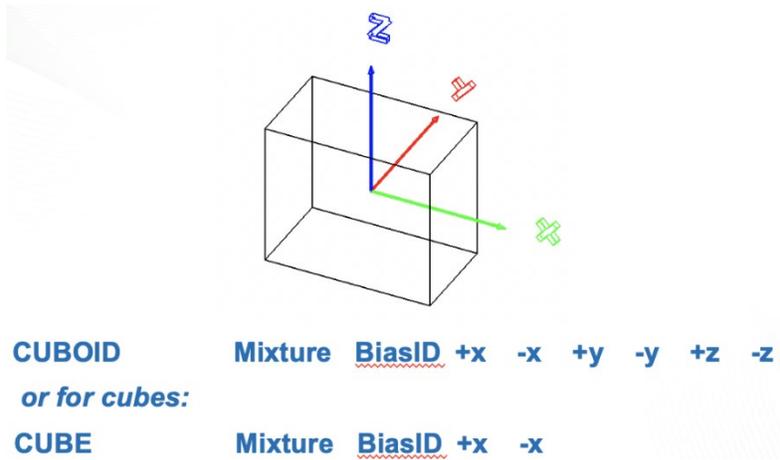
Figure 61. KENO V.a cuboid/cube.
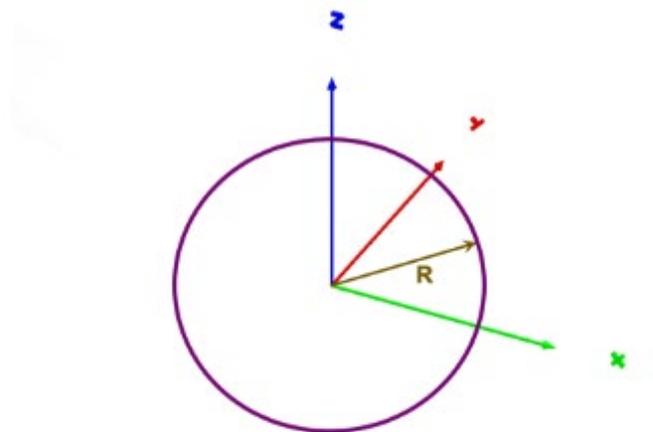
SPHERE Mixture BiasID Radius ORIGIN x-orig y-orig z-orig [0 0 0]

Figure 62. KENO V.a sphere.

CYLINDER
ZCYLINDER

CYLINDER Mixture BiasID Radius +z -z ORIGIN x-orig y-orig [0 0]
ZCYLINDER Mixture BiasID Radius +z -z ORIGIN x-orig y-orig [0 0]

YCYLINDER
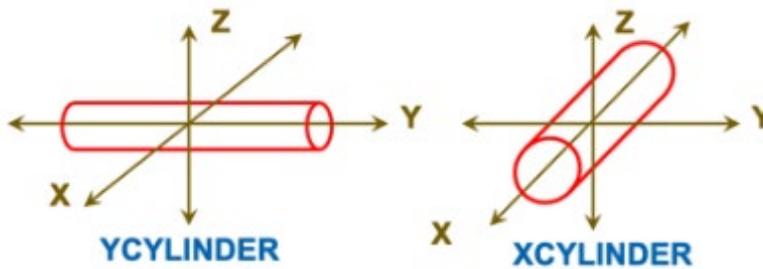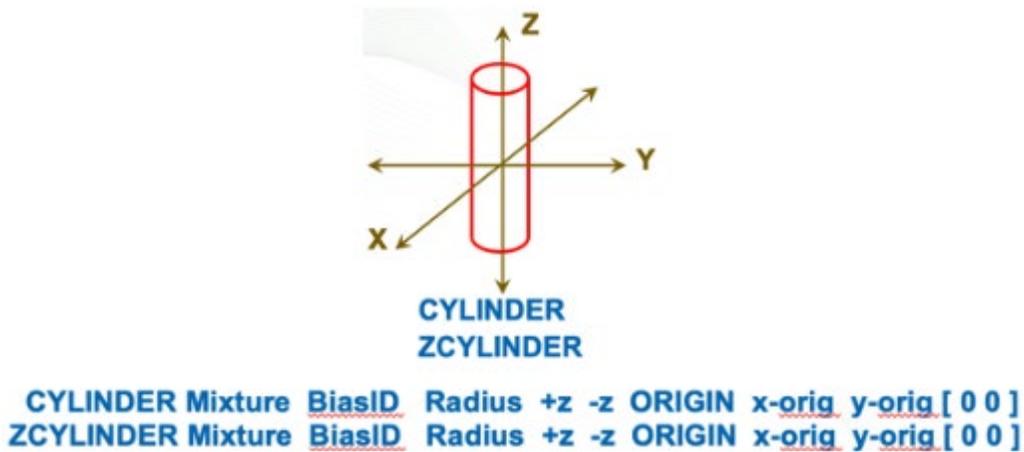
XCYLINDER

Other orientations:
YCYLINDER Mixture BiasID Radius +y -y ORIGIN x-orig z-orig [0 0]
XCYLINDER Mixture BiasID Radius +x -x ORIGIN y-orig z-orig [0 0]

Figure 63. KENO V.a cylinder.

If ρ is:
- Negative: keep *less* than half of the sphere
- Positive: keep *more* than half of the sphere
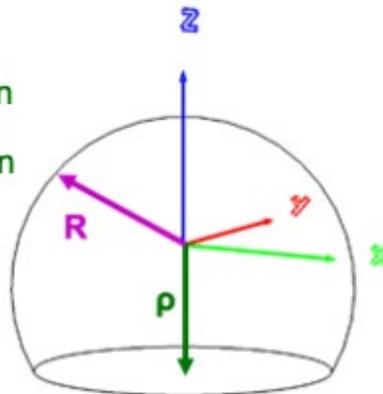
Other orientations:
HEMISPHE+X   HEMISPHE+Y   HEMISPHE+Z*
HEMISPHE-X   HEMISPHE-Y   HEMISPHE-Z
                                   *same as HEMISPHERE

Figure 64. KENO V.a hemisphere.
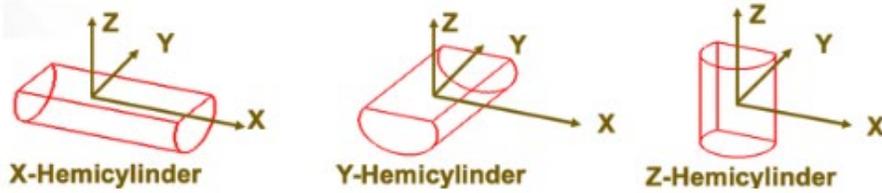
XHEMICYL-Z Mixture BiasID   Radius +x -x CHORD ρ [0] ORIGIN y-orig z-orig [0 0]
XHEMICYL+Z
XHEMICYL-Y
XHEMICYL+Y
YHEMICYL-Z Mixture BiasID   Radius +y -y CHORD ρ [0] ORIGIN x-orig z-orig [0 0]
YHEMICYL+Z
YHEMICYL-X
YHEMICYL+X
ZHEMICYL-Y Mixture BiasID   Radius +z -z CHORD ρ [0] ORIGIN x-orig y-orig [0 0]
ZHEMICYL+Y
ZHEMICYL-X
ZHEMICYL+X

Figure 65. KENO V.a hemicylinder.

cuboid 2 1 7 -7 4 -4 6 -6
or
cuboid 2 1 2p7 2p4 2p6

cuboid 2 1 0 -14 0 -8 0 -12

cuboid 2 1 14 0 8 0 12 0
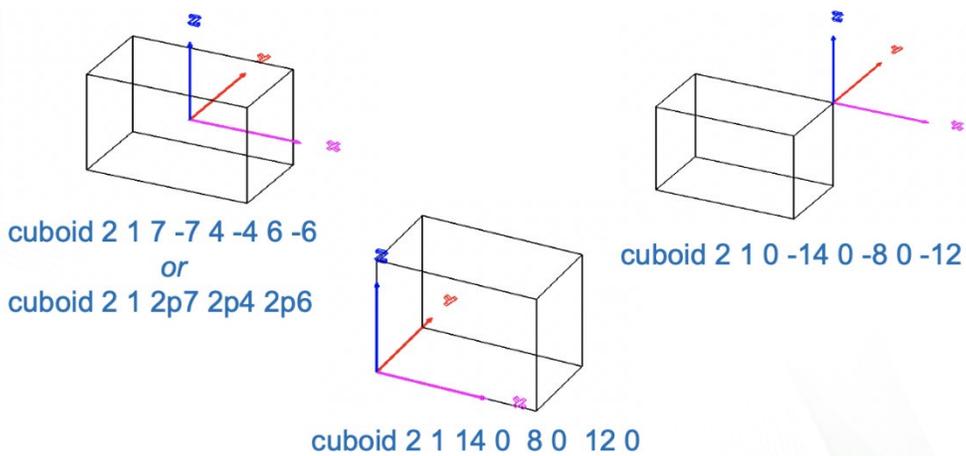
Figure 66. KENO V.a cuboid examples.
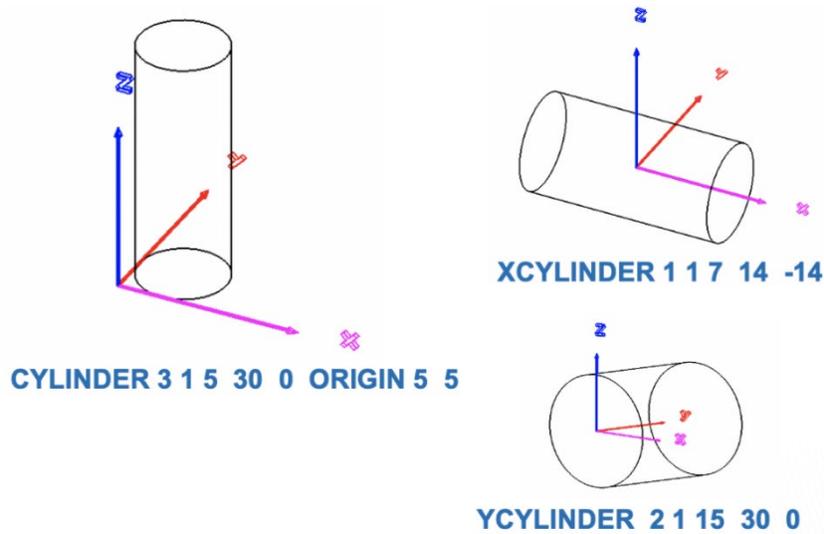
Figure 67. KENO V.a cylinder examples.

As described in the *Quickstart* section, placing your cursor inside the **read geometry** block and pressing CTRL-SPACE displays a dropdown menu of available shapes as shown in Figure 68. Figure 69 shows the input form for a sphere. The box in the lower part of the form shows how the input will appear in the input file. The bias identification (Bias ID) will always be 1 so that KENO V.a performs unbiased neutron transport. Note that for some operating systems in some code versions, the geometry figures with their respective axes may not display in the configurable box.
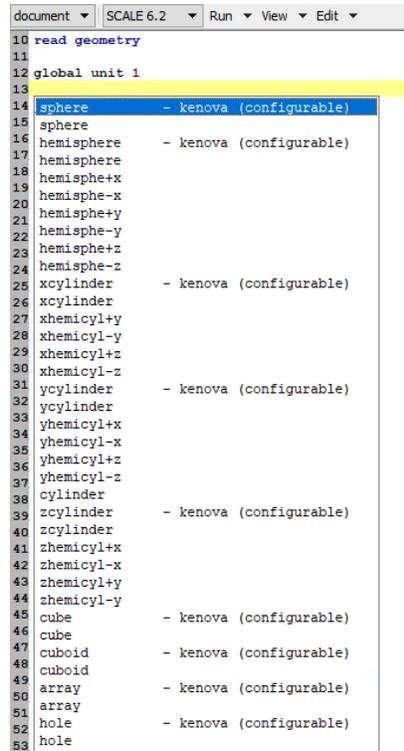
Figure 68. KENO V.a shape options using Fulcrum.
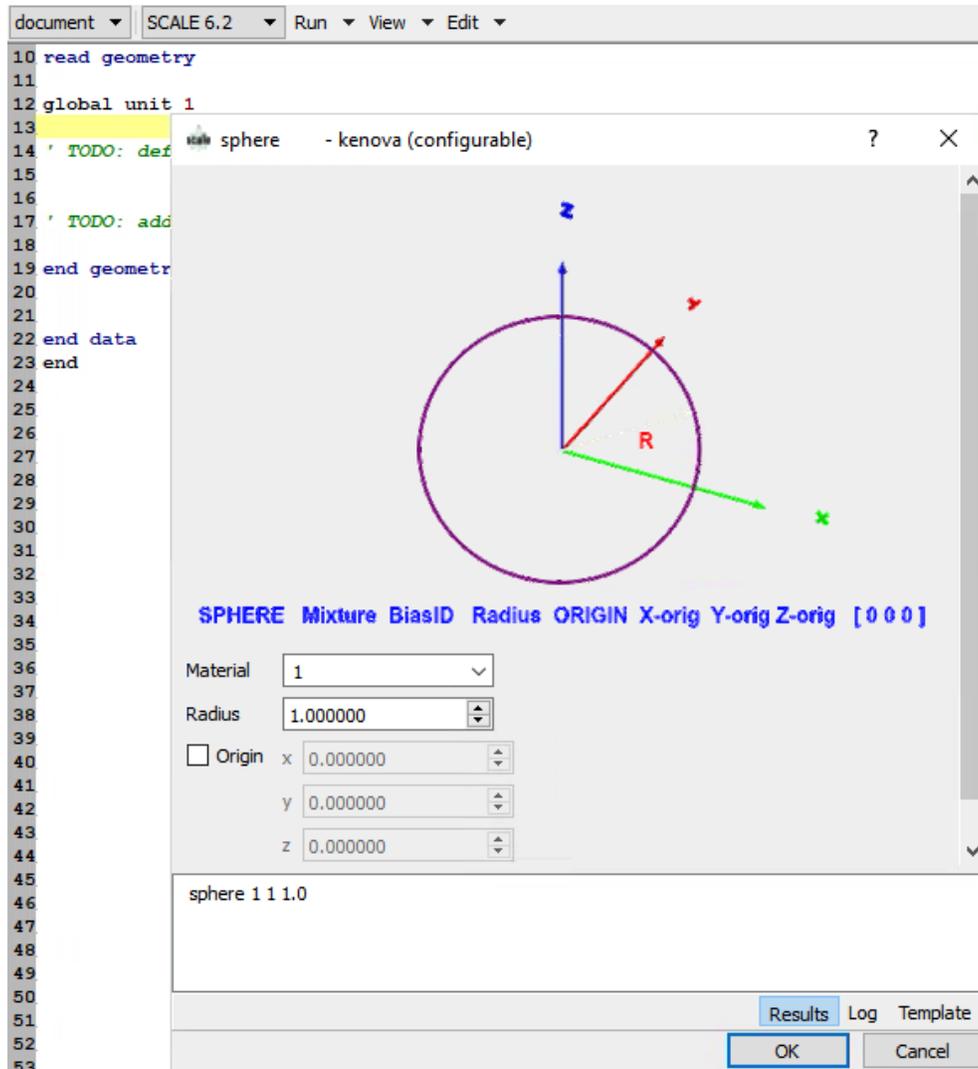
Figure 69. Fulcrum input form for a sphere.

As shown in Figure 69, the material number is specified in the first field, and the radius is entered in the second field. All dimensions in KENO V.a are in centimeters. To specify an origin, check the origin box and enter the respective values.

4.2.1 Simple Cylinder Examples

To better understand how shape parameters are entered using Fulcrum, some examples are presented below. These examples use the metal composition 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 will center a bare metal cylinder on the Z-axis, as shown in Figure 70. A summary of the input information follows.

Core Material (ρ_{mix} = 18.742 g/cm^3)
U metal (100 percent ^{235}U)

Configuration 1. Bare Uranium Cylinder along Z-axis

Radius = 7.82 cm
Height = ±7.82 cm

7.82

-7.82
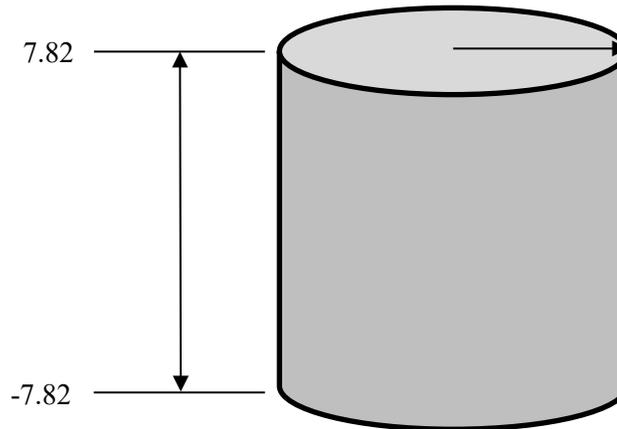
Figure 70. Configuration 1 geometry.

Based on the knowledge gained from the previous sections of the primer, open a new file and enter the title, cross section library, and material information. Note that all examples in this chapter use the continuous energy ENDF/B-V-II.1 library (ce_v7.1).

With this simple geometry, only a few entries are needed to complete the KENO geometry input form. Figure 71 is the uranium metal mixture example from Section 3.4.1.

```
1 =csas5 parm=(   )
2 uranium metal mixture example
3 ce_v7.1
4
5 read comp
6 u-235 1 den=18.742000 1.0 300.0 end
7 end comp
8
9 read geometry
10
11 global unit 1
12
13 ' TODO: define global unit here
14
15
16 ' TODO: add units below
17
18 end geometry
19
20
21 end data
22 end
```

Figure 71. Input without geometry information.

Place the cursor in the global unit (above the **TODO** comment), press `CTRL-SPACE` to display the dropdown menu shown in Figure 68, and select **zcylinder - kenova (configurable)**, this will bring up the **z-cylinder** input form. For **Material**, select 1 from the dropdown menu. Then enter `7.82` for the **Radius**, `7.82` for the **+z** and `-7.82` for the **-z**. This completes the z-cylinder information; the configurable form should look like Figure 72. The text in the preview pane at the bottom of the form shows the input that will be added to the file after clicking ok. The input file, with the extraneous TODO comments removed, is shown in Figure 73.
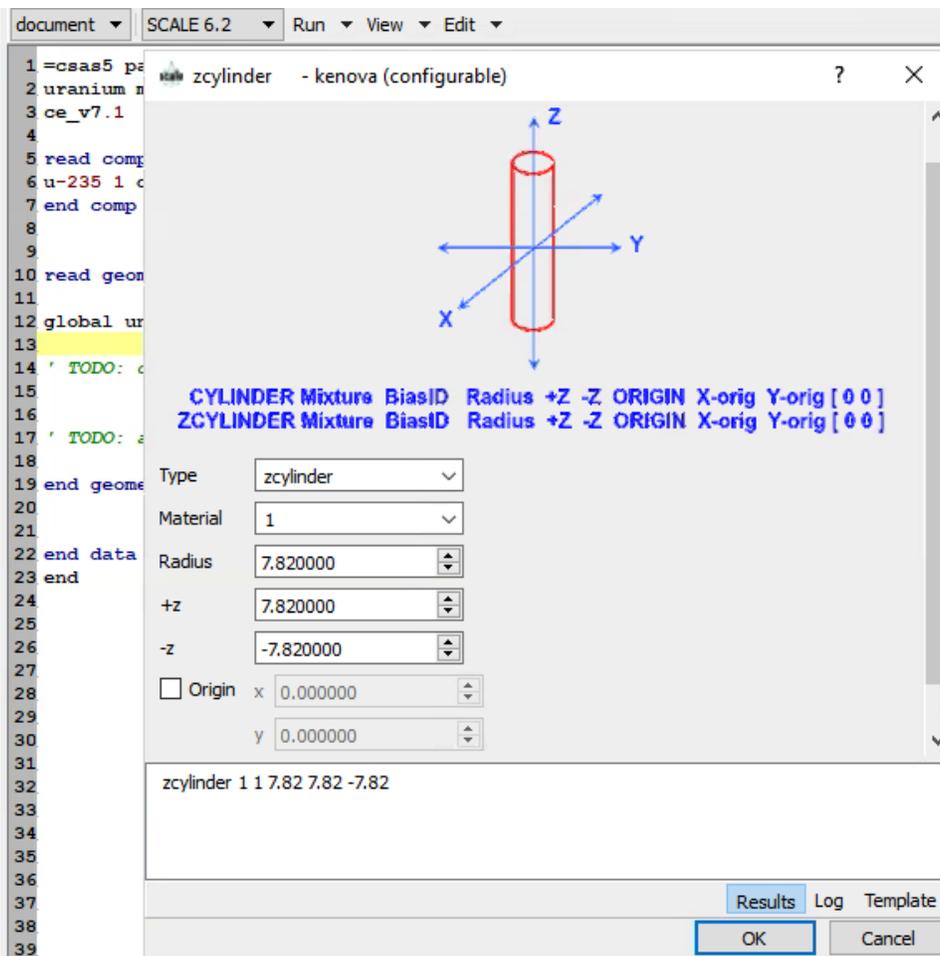
Figure 72. Cylinder data for bare metal cylinder example.

```
 1 =csas5 parm=(   )
 2 uranium metal mixture example
 3 ce_v7.1
 4
 5 read comp
 6 u-235 1 den=18.742000 1.0 300.0 end
 7 end comp
 8
 9 read geometry
10
11 global unit 1
12   zcylinder 1 1 7.82 7.82 -7.82
13
14
15 end geometry
16
17
18 end data
19 end
```

Figure 73. Input with geometry information for bare metal cylinder example.

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

In some fissile systems, the user may want or need to locate a cylinder origin 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 **+z** and **-z**. For example, suppose the origin is located at a point 2 cm above the base of the cylinder (see Figure 74).

13.64

-2.0
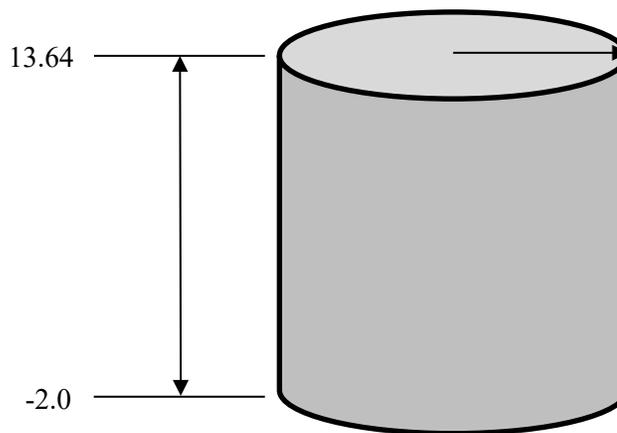
Figure 74. Configuration 2 geometry.

Using the input file from configuration 1, you would only need to change the **z** parameters in the **Geometry** form. Fulcrum does not support importing existing input to a configurable form for modification. However, the previous cylinder input line can be removed and a new cylinder created by pressing the CTRL-SPACE autocomplete key combination and selecting **zcylinder - kenova (configurable)**, to bring up the **z-cylinder** input form. The only differences from configuration 1 are

13.64 for the **+z** and −2.00 for the **-z**. The form and the input are shown in Figure 75 and Figure 76, respectively.
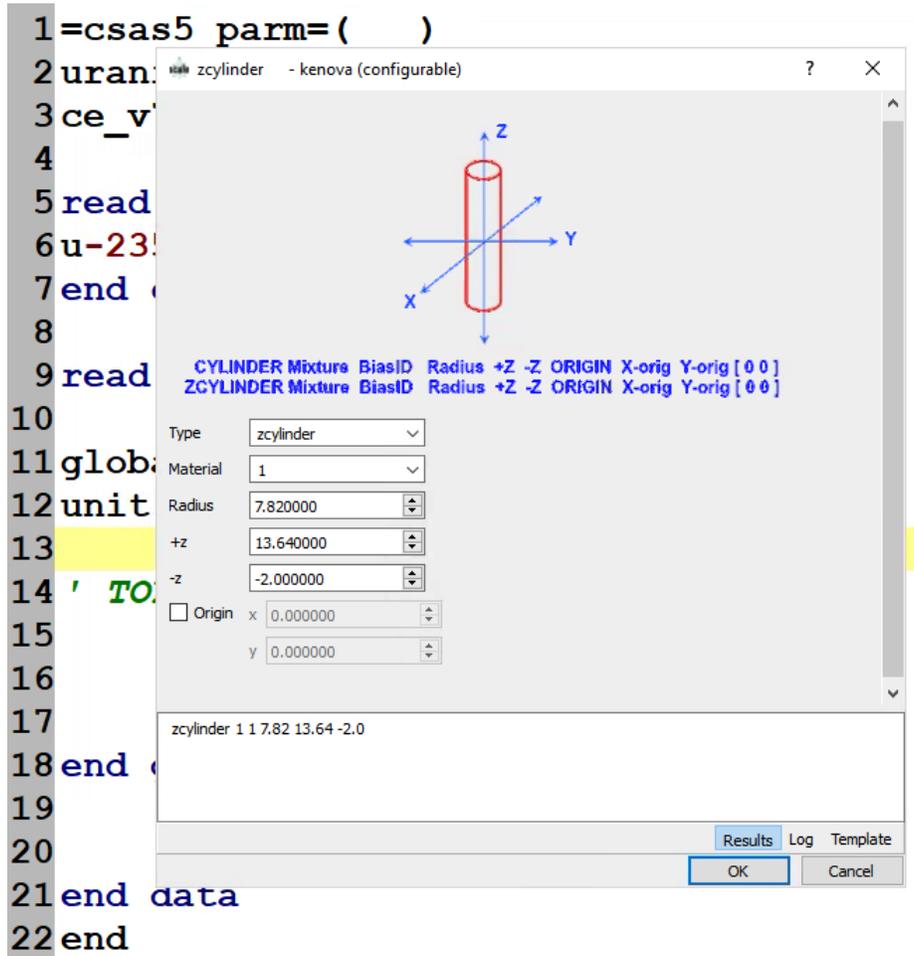
Figure 75. Cylinder data for bare metal cylinder configuration 2.

```
1 =csas5 parm=(    )
2 uranium metal mixture example
3 ce_v7.1
4
5 read comp
6 u-235 1 den=18.742000 1.0 300.0 end
7 end comp
8
9 read geometry
10
11 global unit 1
12 unit 2
13    zcylinder 1 1 7.82 13.64 -2.0
14
15
16 end geometry
17
18
19 end data
20 end
```

Figure 76. Input with geometry information for bare metal cylinder configuration 2.

4.2.1.3   Bare Metal Cylinder along the x-axis with Origin Located at Cylinder Base

The next problem is for a horizontal cylinder with the origin located at the base, as shown in Figure 77. The shape must be changed to an **x-cylinder** and the values of **+x** and **-x** need to be specified to locate the origin at the desired point.
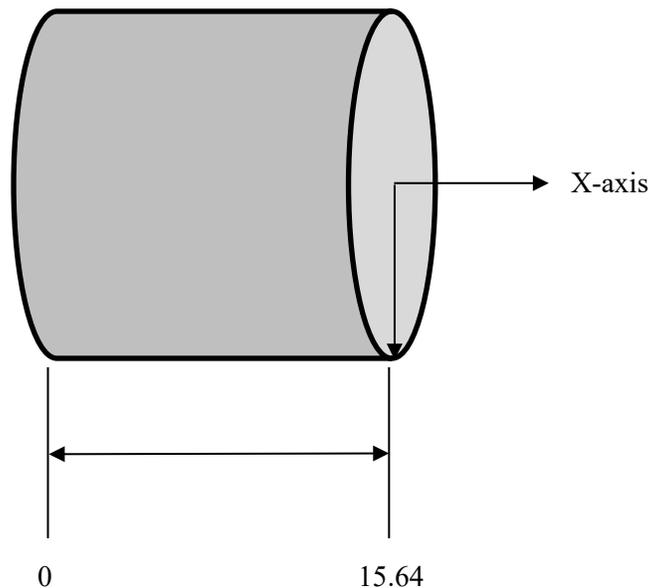
Figure 77. Configuration 3 geometry.

Using the same input file, remove the previous cylinder input line, and create a new cylinder by pressing the CTRL-SPACE autocomplete key combination. This time select the **xcylinder - kenova (configurable)** to bring up the **x-cylinder** input form. Enter 7.82 for the radius, 15.64 for **+x,** and 0 for the **-x**. Figure 78 shows the input form, and Figure 79 shows the input file.
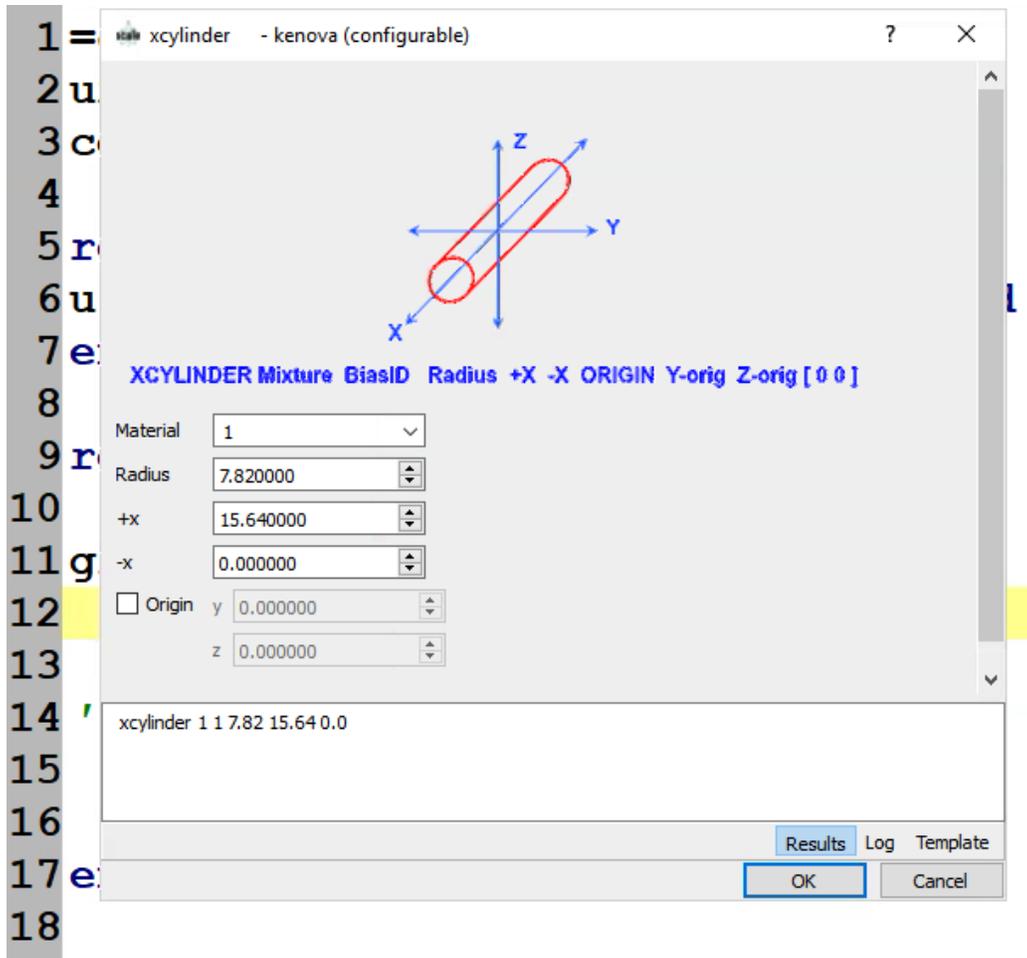
Figure 78. Cylinder data for bare metal cylinder configuration 3.

```
 1 =csas5 parm=(    )
 2 uranium metal mixture example
 3 ce_v7.1
 4
 5 read comp
 6 u-235 1 den=18.742000 1.0 300.0 end
 7 end comp
 8
 9 read geometry
10
11 global unit 1
12   xcylinder 1 1 7.82 15.64 0.0
13
14
15 end geometry
16
17
18 end data
19 end
```

Figure 79. Input with geometry information for bare metal cylinder configuration 3.

The same approach applies to locating the origin of cubes and cuboids. Enter values of the maximum and minimum parameters to locate the origin on each axis. For spheres and hemispheres, the concept is the same, but the coordinates of the origin of the sphere or hemisphere are entered if they differ from the default origin of (0, 0, 0).

4.3   KENO V.A GEOMETRY RULES

There are seven basic rules that guide the geometry input to SCALE/KENO V.a.

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 from the inside out using regions. Regions are made using the basic KENO V.a geometry types. Within a unit, a region must fully enclose all previously defined regions.
3. Regions may share boundaries but may not intersect.
4. All volumes are oriented along a major axis. Volumes may not be rotated.
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. As many holes as required may be placed in a unit as long as they do not intersect other holes or regions.
6. *An **ARRAY** is an ordered stack of units.* The touching faces of adjacent units in an ARRAY must be the same size. Only one ARRAY may be placed directly in a unit. Additional ARRAYS may be placed in units using holes.
7. A GLOBAL unit or array, which defines the outermost boundary of the problem, shall be specified for each problem.

4.4   GEOMETRIC ARRANGEMENTS

Geometric arrangements in KENO V.a 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 (i.e., cubes, cuboids, cylinders, spheres, hemispheres, and hemicylinders). These shapes can be placed

anywhere within a unit as long as they are oriented along the coordinate system of the unit and do not intersect other regions (i.e., shapes). This means, for example, that a cylinder must have its axis parallel to one of the coordinate axes, whereas a rectangular parallelepiped (cuboid) must have its faces perpendicular to the three coordinate axes. The most stringent KENO V.a geometry restriction is that none of the options allow geometry regions to intersect. Figure 80 shows some geometries that are not allowed.
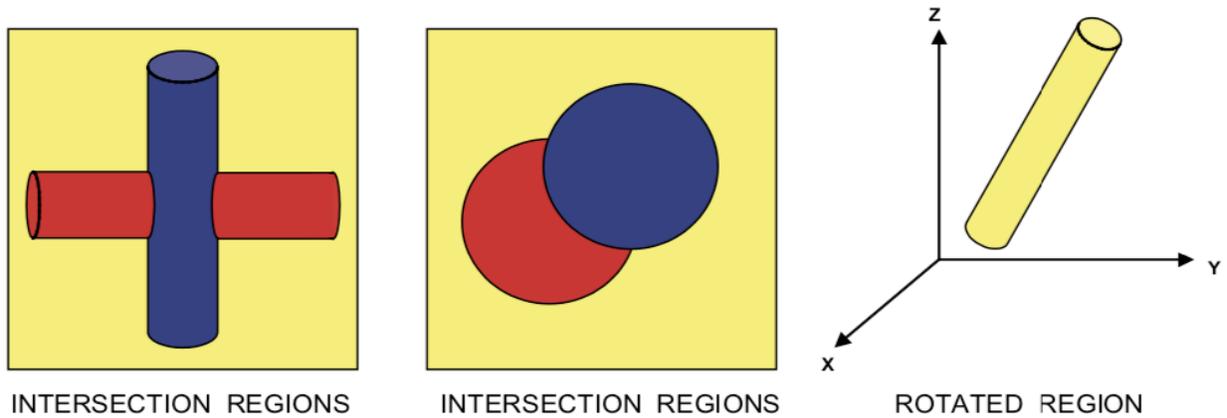
Figure 80. Examples of geometry not allowed in KENO V.a.

Unless special options (e.g., holes — discussed in Section 5.3) are invoked, each geometric region in a unit must completely enclose each interior region. However, regions can touch at points of tangency and share faces. See Figure 81 for examples of allowable geometries.
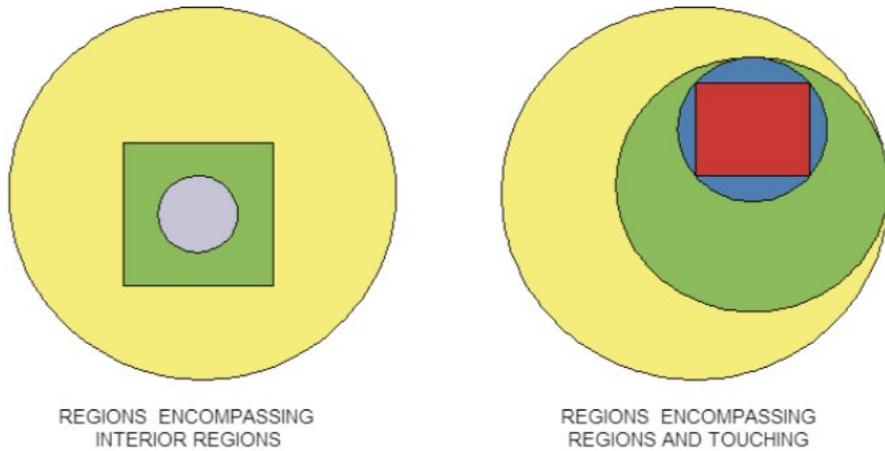
Figure 81. Examples of correct KENO V.a units.

A unit can be thought of as a container that encloses a number of shapes. In its simplest form, a unit encloses one or more basic shapes, with the smallest shape completely enclosed by the next smallest, and so on. It is important to note that the origin of a unit is set by the origin of the first, innermost shape. Also, note that the shapes must be entered in the geometry block from smallest to largest.

4.5   NESTED REGIONS IN A UNIT

Two example problems will be used to illustrate how to enter KENO V.a geometry information for nested regions in a unit via Fulcrum. 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/cm^3) 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/cm^3) with a thickness of 17.78 cm in both the radial and axial directions (see Figure 82).
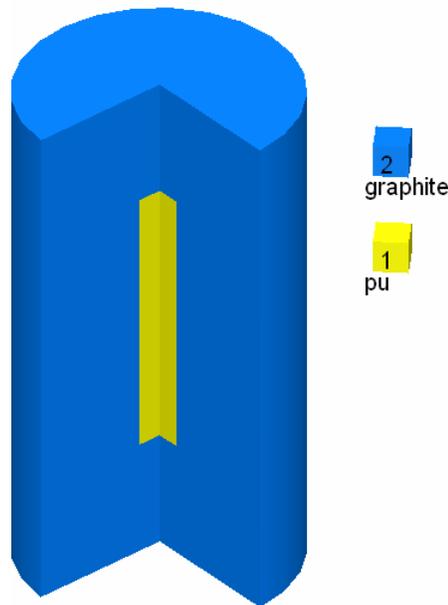
Figure 82. Reflected Pu metal cylinder geometry.

Based on the knowledge gained from the previous sections of the primer, open a new file and enter the title, cross section library, and material information. Remember to specify the isotopic composition for material 1 with 95 wt.% ^{239}Pu and 5 wt.% ^{240}Pu and a plutonium density of 15.44 g/cm^3. Material 2 should be c-graphite with a density of 1.6 g/cm^3.

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 must be described first, and the outer cylinder containing graphite will be described second. These are nested regions, so they obey KENO V.a geometry rules 1 and 2. Place the cursor on the line below global unit 1 and press the CTRL-SPACE autocomplete key combination. Select the **zcylinder - kenova (configurable)** option to open the configurable input form. Enter 2.86 for **Radius**, 40.96 for **+z**, and 0 for **-z.** This puts the origin at the bottom center of the core cylinder. The configurable form for the plutonium cylinder is shown in Figure 83. Click **OK** after confirming that the input is correct.
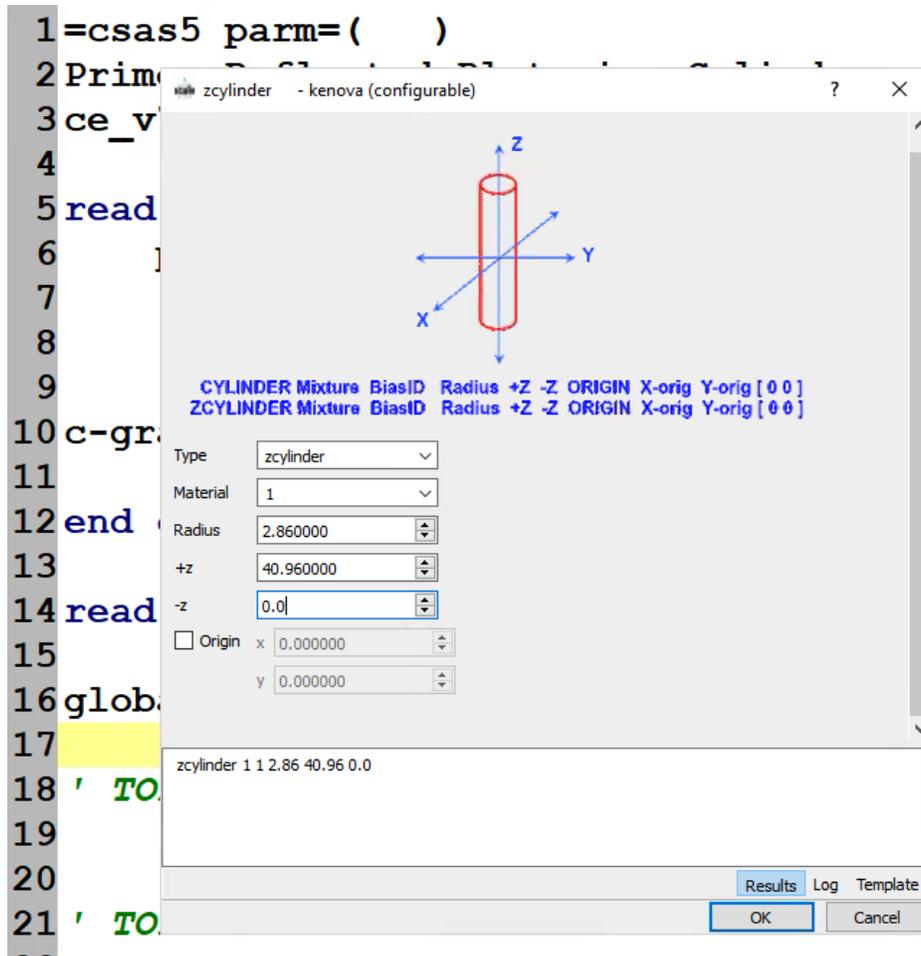
Figure 83. Geometry description for region 1.

To enter data for the reflector, copy the existing cylinder and paste it again on the next line. Change the mixture in the region to graphite by changing the first **1** to a 2, as shown in Figure 84.

```
12 global unit 1
13     zcylinder 1 1 2.86 40.96 0.0
14     zcylinder 2 1 2.86 40.96 0.0
```

Figure 84. Second cylinder specification with only the mixture number changed.

The dimensions of the reflector can be determined using the in-line evaluation capability in Fulcrum. Place the cursor directly after the radius of **2.86**, as shown in Figure 84, and enter +17.78; then highlight the entire expression as shown in Figure 85. With the expression highlighted, click the triangle next to the word **Edit** in the Fulcrum menu bar to display the **Edit** dropdown list, as shown in Figure 86. Click on **Evaluate**, and Fulcrum will perform the arithmetic and substitute the appropriate sum in the input file. Repeat this process for the top of the cylinder (+17.78) and the bottom of the cylinder (−17.78). Note that the bottom of the reflector is below the bottom of the plutonium, so the thickness must be subtracted to generate the appropriate dimension. The expressions for these two surfaces are shown before evaluation in Figure 87 and after evaluation in Figure 88. The order of specifying and evaluating

expressions has no impact on the input, but only one expression can be evaluated at a time, so each dimension must be evaluated separately. The completed input for this model is shown in Figure 89.

```
12 global unit 1
13    zcylinder 1 1 2.86 40.96 0.0
14    zcylinder 2 1 2.86+17.78 40.96 0.0
```

Figure 85. Entering expression for Fulcrum to evaluate to determine the reflector radius.
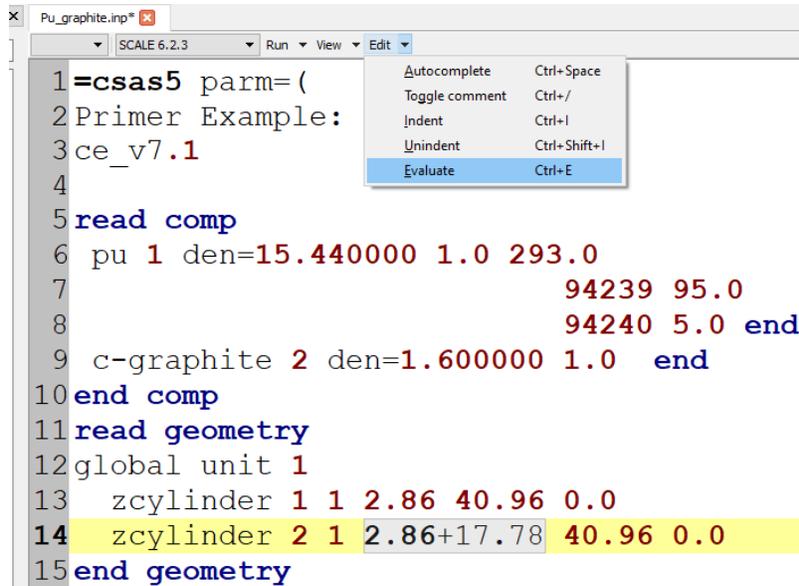
Figure 86. Selecting Evaluate from the Edit dropdown menu.

```
12 global unit 1
13    zcylinder 1 1 2.86 40.96 0.0
14    zcylinder 2 1 20.64 40.96+17.78 0.0-17.78
```

Figure 87. Expressions for the top and bottom dimensions of the graphite reflector cylinder.

```
12 global unit 1
13    zcylinder 1 1 2.86 40.96 0.0
14    zcylinder 2 1 20.64 58.74 -17.78
```

Figure 88. Completed input for the plutonium and graphite cylinders.

```
 1 =csas5 parm=(    )
 2 Primer Reflected Plutonium Cylinder
 3 ce_v7.1
 4
 5 read comp
 6     pu 1 den=15.44 1.0 293.0
 7         94239 95.0
 8         94240 5.0 end
 9
10 c-graphite 2 den=1.600000 1.0  end
11
12 end comp
13
14 read geometry
15
16 global unit 1
17   zcylinder 1 1 2.86 40.96 0.0
18   zcylinder 2 1 20.64 58.74 -17.78
19
20
21 end geometry
22
23
24 end data
25 end
```

Figure 89. Complete input for reflected Pu cylinder.

After confirming that the input is correct, press the **Run** button in the Fulcrum menu bar to execute CSAS5. The results are shown in the **Messages** pane, and the calculated **best estimate system k-eff** should be approximately 0.9780 ± 0.0026. Recall that results will vary based on different computers and operating systems, but the results should agree within the reported uncertainties on the two separate calculations.

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 centers of the sphere and cylinder are offset so that the outside edge of the cylinder touches two of the outside edges of the cube (see Figure 90). This is consistent with KENO V.a geometry rule 3, which indicates regions may share boundaries but may not intersect. The materials are ^{235}U metal (density = 18.74 g/cm^3), graphite (density = 1.65 g/cm^3), and water (default density of 0.9982 g/cm^3). 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.
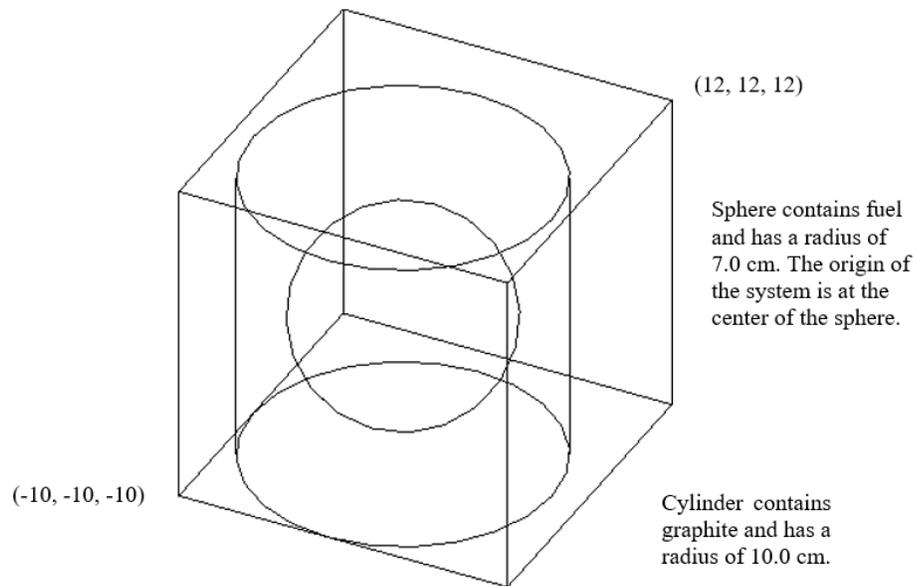
Figure 90. ²³⁵U sphere surrounded by graphite and water.

A cutaway view of the geometry is shown in Figure 91. 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.
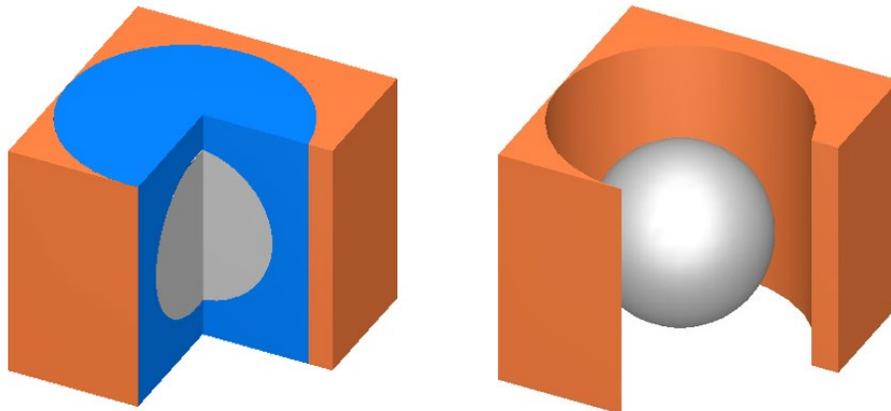
Figure 91. Cutaway view of ²³⁵U sphere inside graphite and water.

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

For this problem, there will be three regions in one unit, and that unit will be the global unit for the problem. The geometry must be described from the inside out, so the sphere containing the uranium will be specified first, then the cylinder containing the graphite, and finally the outer cube containing water. These are nested regions and are entered like those in the previous example.

To start with the sphere, place the cursor after the global unit 1 line and press the CTRL-SPACE autocomplete key combination. Click on the **sphere - kenova (configurable)** option which opens the sphere input form. Enter 7.0 for the radius. The form should look like Figure 92. Click **OK**.
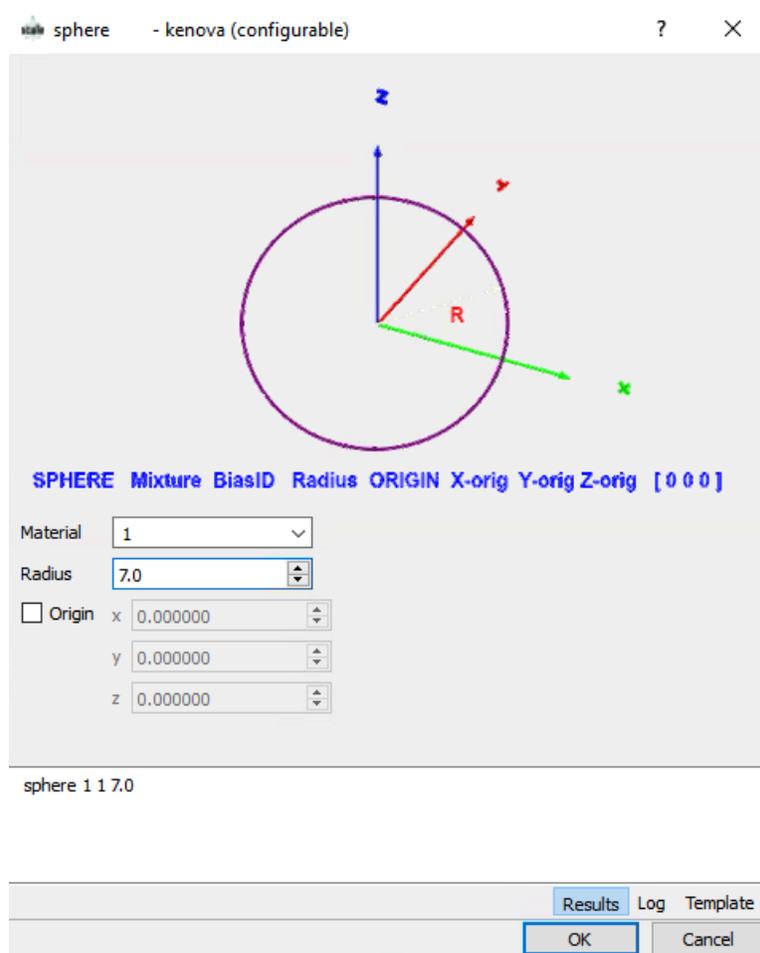
Figure 92. Input for metal sphere.

The next region in the model is the graphite cylinder. Several cylinders have already been input in the primer, so the syntax should be familiar. Direct input is typically faster than using configurable forms once a user is familiar with the syntax. The cylinder specification can be entered as `cylinder 2 1 10 10 -10`. The radius of 10 is specified in the problem statement, and the origin of the sphere is also the origin of the cylinder. Therefore, its axial extent must be from -10 cm to 10 cm. Recall that the first number specified after the geometry keyword is the mixture number describing the material present in this region.

The last region is the water cube, with positive surfaces defined at 12 cm and negative surfaces at -10 cm (see Figure 90). The input specification for the cube is therefore `cube 3 1 12 -10`. The complete geometry block is shown in Figure 93, and the complete input is shown in Figure 94.

```
 9 read geometry
10 global unit 1
11    sphere 1 1 7.0
12    cylinder 2 1 10 10 -10
13    cube 3 1 12 -10
14 end geometry
```

Figure 93. Complete geometry input for water and graphite reflected uranium sphere.

```
 1 =csas5 parm=(    )
 2 Primer example: Reflected U sphere
 3 ce_v7.1
 4 read comp
 5  u-235 1 den=18.740000    end
 6  c-graphite 2 den=1.650000    end
 7  h2o 3      end
 8 end comp
 9 read geometry
10 global unit 1
11    sphere 1 1 7.0
12    cylinder 2 1 10 10 -10
13    cube 3 1 12 -10
14 end geometry
15 end data
16 end
```

Figure 94. Input file for [235]U sphere with graphite and water reflectors.

After confirming that the input is correct, press the **Run** button in the Fulcrum menu bar to execute CSAS5. The results are shown in the **Messages** pane, and the calculated **best estimate system k-eff** should be approximately 1.0011 ± 0.0018. Recall that results will vary based on different computers and operating systems, but the results should agree within the reported uncertainties on the two separate calculations.

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 or a global array.

Arrays are created by stacking units that have a rectangular parallelepiped (CUBE or CUBOID shape) outer region. The dimensions the faces of adjacent units stacked in this manner must match exactly. See Figure 95 for a typical example.
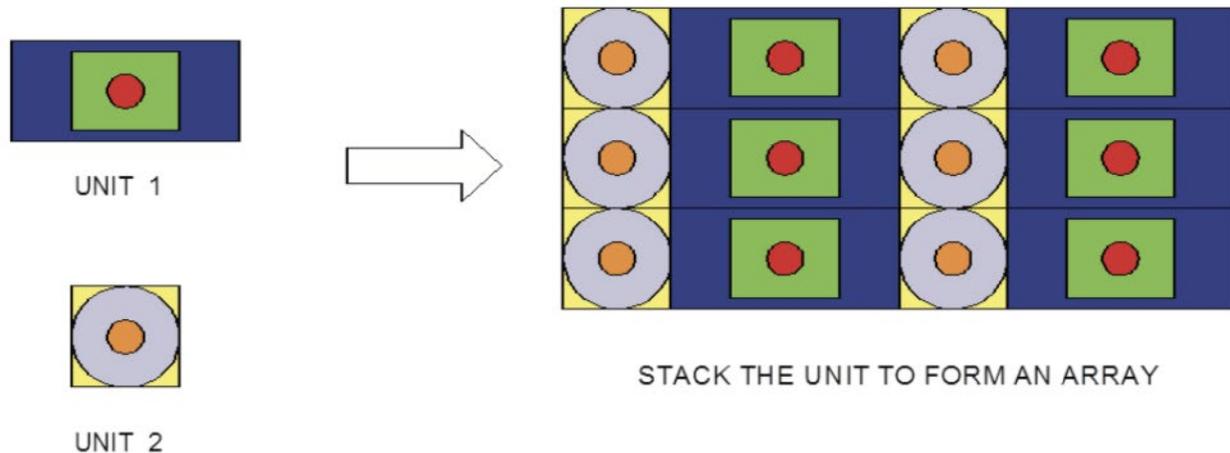
Figure 95. Example of array construction.

The ARRAY option is provided to allow placement of an array within a unit. Only one array can be placed directly in a unit. However, multiple arrays can be placed within a unit by using holes, as described in the Section 5.3. Arrays of dissimilar units or dissimilar arrays can be created by stacking units that contain arrays into an array of arrays. See Figure 96 for an example of an array composed of units containing holes and arrays.
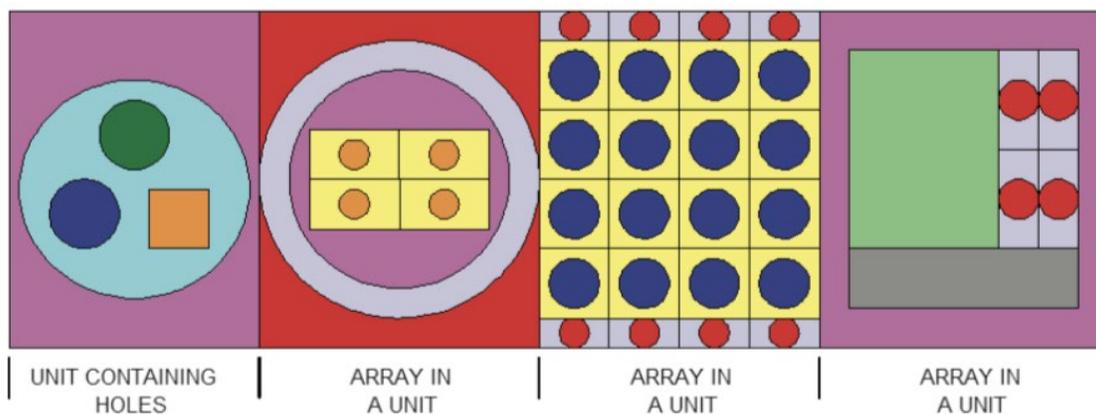
Figure 96. Example of an array composed of units containing arrays and holes.

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 diameter of 11.496 cm and an outside height of 10.765 cm. The uranium has a density of 18.76 g/cm^3 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 97 and Figure 98.

For a unit to be placed in an array, the outside shape of the unit must be either a cube or a cuboid. Because the constituents of the array are cylinders, they must be enclosed by cuboids to become part of the 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.

67

13.74 cm in X and Y

13.01 cm in Z
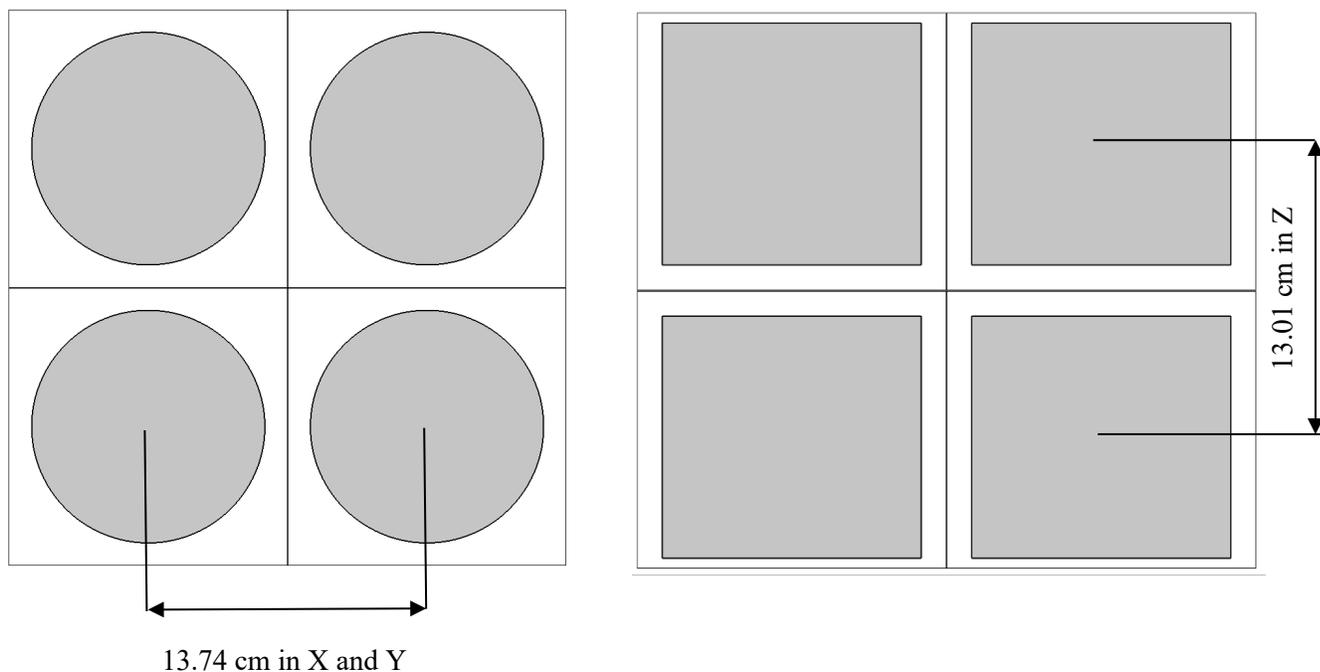
Figure 97. 2 × 2 × 2 array geometry.
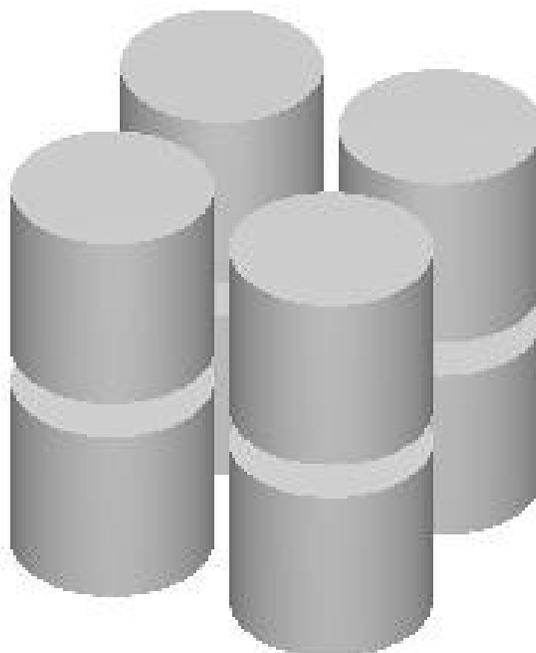
Figure 98. 3D view of array.

Using knowledge gained 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.

For this geometry, there will be two regions in one unit, but that unit is not the global unit for the problem. The global unit for this problem will be the array containing the eight units. To describe a single metal cylinder unit from the inside out, describe the cylinder containing the uranium first, followed by the

cuboid enclosing the cylinder. This unit will be used as an individual element of the array. These are nested shapes and are entered in the same manner as for the previous example. 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; this will 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. Delete the word **global** from unit 1, as this will not be the global unit. Use a configurable form or directly type in the cylinder specification into the input file. The **Radius** for the cylinder will be `5.748`, the **Top** will be `5.3825`, and the **Bottom** will be `-5.3825` (see Figure 99). 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. The enclosing cuboid input is therefore `cuboid 0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505`. The mixture number 0 is used to fill a region with void; mixture 0 is always void. Void is entirely empty space, so it has no cross section. These parameters are sufficient for describing the single array element as shown in Figure 100. The complete input for Unit 1 is shown in Figure 101.

5.748 cm radius

10.765 cm tall
Half height = 5.3825 cm
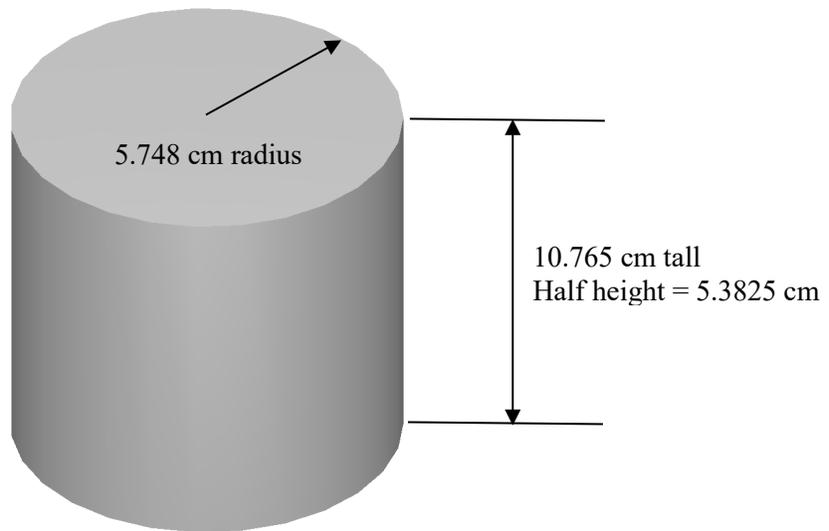
Figure 99. Individual metal cylinder unit.

69
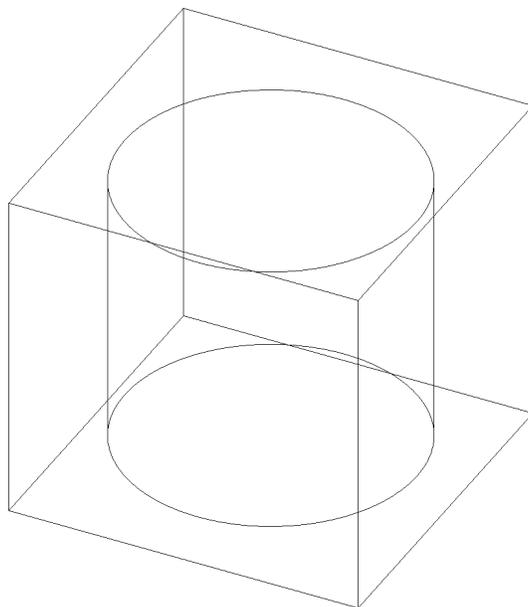
Figure 100. Outline of single metal cylinder inside bounding void cuboid.

```
15 unit 1
16  zcylinder 1 1 5.748 5.3825 -5.3825
17  cuboid 0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
```

Figure 101. Single unit geometry for array.

To start describing the array, place the cursor on the line after **end geometry**, and press the CTRL-SPACE autocomplete key combination. From the dropdown list shown in Figure 102, select **arrays**, which places text in the input file, as shown in Figure 103. Then place the cursor in the **array block** and press the CTRL-SPACE autocomplete key combination. Figure 104 shows the only available option to choose, which opens the array configurable form.
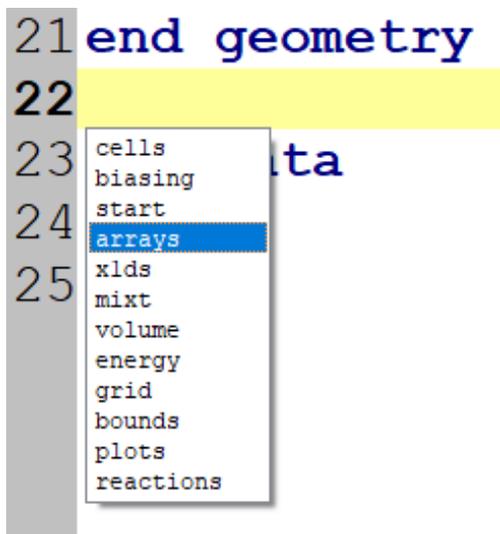
Figure 102. Autocomplete options.

```
19 end geometry
20 read array
21
22 ' TODO: define array
23
24 end array
```

Figure 103. Array autocomplete entry.

```
21
22 array          - kenova basic (configurable)
```

Figure 104. Array autocomplete option.

In the array input window, type 1 in the ara field to designate this as array 1 and 1 in the gbl field to declare array 1 as the global array. No global unit will be used in this model, as the entire system being modeled is represented in this array. The array is 2 units in the X-direction, 2 units in the Y-direction, and 2 units in the Z-direction, so **nux**, **nuy**, and **nuz** are all 2. The **prt** option prints a summary of the contents of this array to the output file. The **fill id** field specifies the unit with which to initialize the fill data, so the default of **1** is appropriate here. 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 105. Figure 106 shows the completed configurable form, and Figure 107 shows the input file.
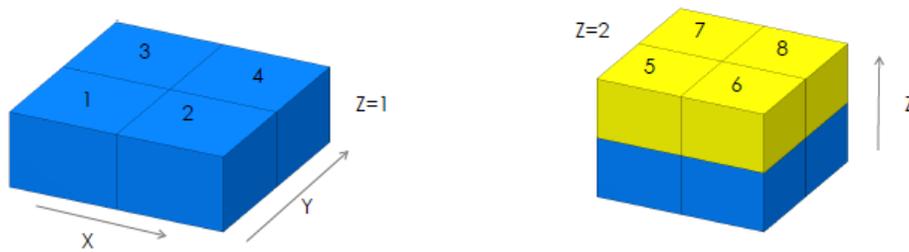
Figure 105. Unit filling order for an array.

71
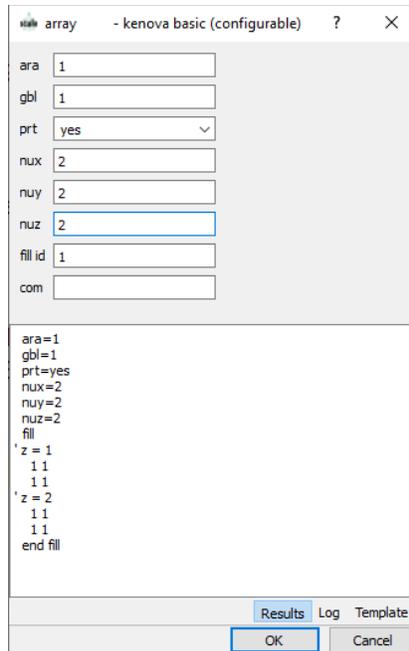
Figure 106. Array autocomplete window.

```
 1 =csas5
 2 Primer example: 2x2x2 array
 3 ce_v7.1
 4 read composition
 5  u 1 den=18.76 1 293
 6        92234 1
 7        92235 93.2
 8        92236 0.2
 9        92238 5.6    end
10 end composition
11 read geometry
12 unit 1
13  zcylinder 1 1 5.748 5.3825 -5.3825
14  cuboid 0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
15 end geometry
16 read array
17   ara=1
18   gbl=1
19   prt=yes
20   nux=2
21   nuy=2
22   nuz=2
23   fill
24 ' z = 1
25     1 1
26     1 1
27 ' z = 2
28     1 1
29     1 1
30   end fill
31 end array
32 end data
33 end
```

Figure 107. Array input file.

72

After confirming that the input is correct, press the **Run** button in the Fulcrum menu bar to execute CSAS5. The results are shown in the **Messages** pane, and the calculated **best estimate system k-eff** should be approximately 0.9992 ± 0.0016. Recall that results will vary based on different computers and operating systems, but the results should agree within the reported uncertainties on the two separate calculations.

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 (See Figure 108). There are two types of disks. One type contains graphite (density = 1.65 g/cm^3) and is 10 cm in radius and 2.5 cm thick. The second type contains U(93.2) and 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. Starting at the bottom, the disks alternate from graphite (gray) to uranium (yellow). 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.

One of the requirements for a unit to be placed in an array is that 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 as the top plate. Thus, the length of the X and Y sides of the cuboid must be 30 cm. Note that there is no constraint on the Z-dimension of the cuboid, as there are no adjacent faces in that direction.
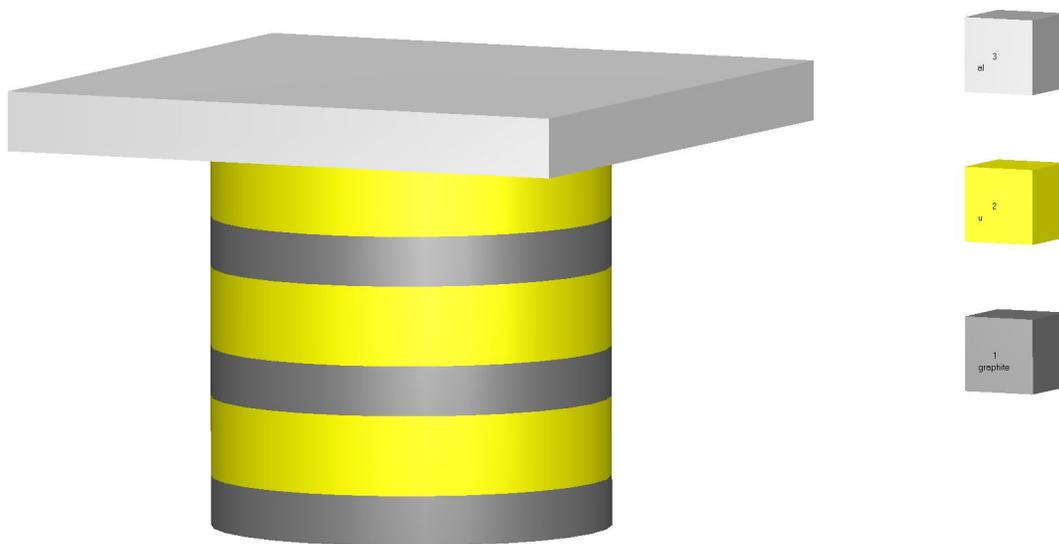
Figure 108. Geometry of stack.

Using knowledge gained 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.

Three units will be needed for this model: one with a graphite disk, one with a uranium disk, and one for the aluminum plate. These units will then be arrayed together to form the final model. Although a global array could be used in this problem, the implementation here will use a global unit to demonstrate positioning an array in a unit.

The first unit will be the graphite disk. It will not be the global unit, so delete the word **global**. The dimensions of the cylinder have been provided, and the axial extent is arbitrary since this unit will be used in an axial array. Directly input the cylinder specification or use a configurable form. The radius is 10 cm, the top is at 2.5 cm, and the bottom is at 0 cm. Use the graphite mixture number assigned in the composition block (1 in the example input). Next, specify a cuboid filled with void around the cylinder with X and Y dimensions ±15 cm and Z dimensions 2.5 cm and 0 cm.

The second unit will be the uranium disk unit. It is very similar to unit 1, so copying unit 1 and pasting it a second time in the geometry block is a good place to start. Change the unit number to 2, the disk mixture to the appropriate number for uranium (2 in the example input), and the height of the cylinder and cuboid to 4 cm.

The third unit will be the aluminum block on top of the stack of disks. This is a very simple unit that will serve as a demonstration for using the unit configurable form in Fulcrum. After unit 2, but still in the geometry block, use the CTRL-SPACE key combination to generate the list of autocomplete options. As shown in Figure 109, the three choices are **global unit**, **unit - kenova (configurable)**, and **unit**. Selecting the **unit - kenova (configurable)** will display the unit configurable form. After appropriate resizing, the form should look like the one shown in Figure 110. This form can be difficult to manage for complex units, but it can be convenient for simple units like this one. Set the Id field in the upper middle portion of the form to 3. This unit will not be the global unit, so leave **Global** set to **no**. In the **Properties** pane on the left side of the form, click **New** in the **Definition** field to generate a dropdown list of regions. Select **Cuboid** to add a new Region; after resizing the form again it should appear as shown in Figure 111. Type the mixture number for aluminum (3 in the example input) into the **Material** field. The right side of the unit form essentially displays the cuboid configurable form. Set **+x** and **+y** to 15, set **-x** and **-y** to −15, set **+z** to 3, and set **-z** to 0. No other regions are needed, so the unit specification should be complete. Confirm the input with the configurable form shown in Figure 112, and then press **OK** to add the input to the file. The complete composition and geometry blocks are shown in Figure 113.

Figure 109. Selecting the unit configurable form.
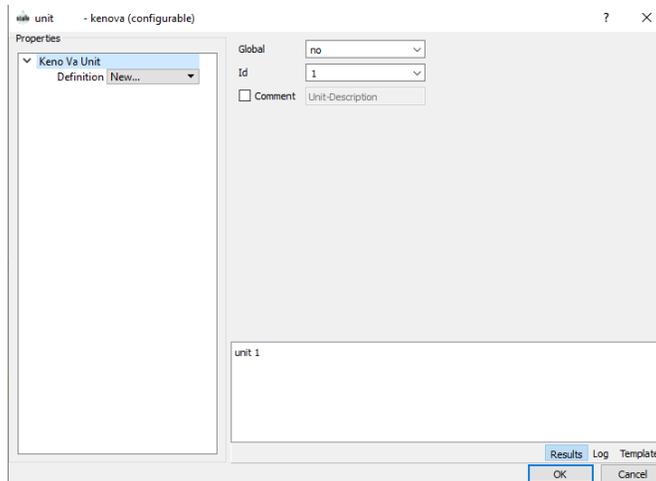
Figure 110. Empty Unit configurable form.
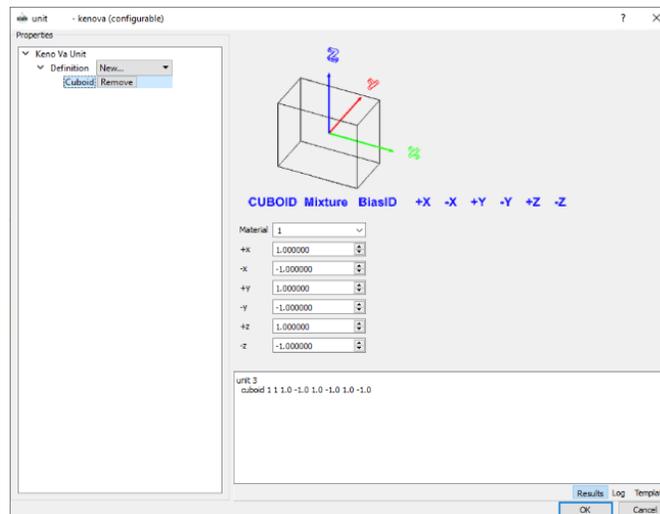
Figure 111. Unit configurable form with cuboid form.
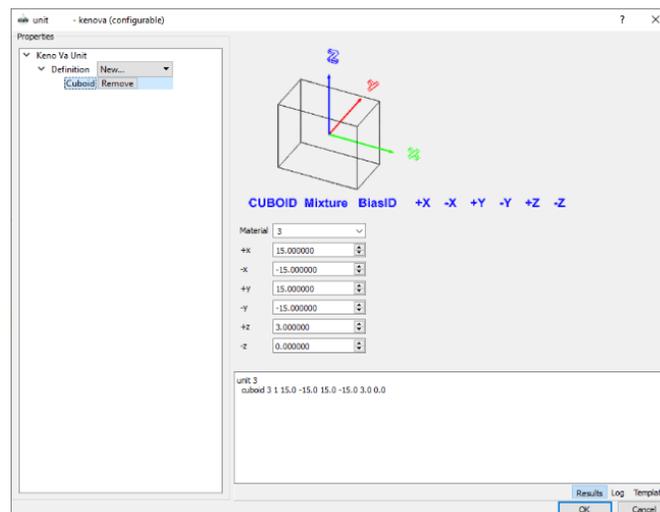
Figure 112. Completed unit configurable form.

```
1 =csas5 parm=(    )
2 primer - disk stack
3 ce_v7.1
4 read comp
5  graphite 1 den=1.65 1.0 end
6  u 2 den=18.74 1.0 293
7              92234 1.0
8              92235 93.2
9              92236 0.2
10             92238 5.6 end
11  al 3 1.0 end
12 end comp
13 read geometry
14 unit 1
15  cylinder 1 1 10   2.5 0
16  cuboid    0 1 15 -15   15 -15   2.5 0
17 unit 2
18  cylinder 2 1 10   4 0
19  cuboid    0 1 15 -15   15 -15   4 0
20 unit 3
21   cuboid 3 1 15.0 -15.0 15.0 -15.0 3.0 0.0
22 end geometry
```

Figure 113. Composition and geometry blocks for the disk stack array model.

To start describing the array, place the cursor on the line after **end geometry** and press the CTRL-SPACE autocomplete key combination. From the dropdown list shown in Figure 102, select **arrays**. Then place the cursor in the **array block** and press the CTRL-SPACE autocomplete key combination and choose **array - kenova basic (configurable)**, which opens the array input window. In the array input window, accept the default array number, **ara**, of 1, and leave the gbl field blank indicating that there is no global array for this problem. This array could be used as the global array, but for demonstration purposes in this exercise, a global unit will be used. The array is a single unit in the x and y directions, so both nux and nuy should remain 1. The number of elements in the axial direction is 7: 3 graphite disks, 3 uranium disks, and the aluminum block. This means that 7 should be entered for **nuz**. The **fill id** field initializes the array **fill**, but no single unit is the majority of elements in this case. The default unit 1 is a simple choice. The completed array configurable form is shown in Figure 114. After confirming that the configurable form is correct, press **OK** to add the array description to the input file. As initialized by the configurable form, all 7 axial elements in the array are unit 1. This does not describe the actual system, so editing the input is necessary at this point. Recall from the last example, as shown in Figure 105, that the order of the fill entries is interpreted in increasing x, then increasing y, and finally in increasing z. This array is 1 unit in the x direction and 1 unit in the y direction, so the entries are increasing in z from entry 1 to 7. This creates a situation in which the input goes down while the array elements are going up in the actual model, which can be confusing. Therefore, Fulcrum adds comments describing the z level associated with each section of the array fill. The uranium disk is unit 2, and it needs to be substituted at z levels 2, 4, and 6. The top element, z level 7, needs to be changed to unit 3 for the aluminum plate. The final array block, including the appropriate **fill**, is shown in Figure 115.

Figure 114. Completed array configurable form.

Figure 115. Complete array block input.

The last step is to introduce the array into the geometry block. The array is positioned in a unit by specifying the position of the most negative point in the array in the coordinate system of the unit. In other words, the array is positioned based on the location of the minimum x, minimum y, and minimum z point in the array. In this case, the position of the array does not matter because it is the only region in the unit. For the purposes of demonstration, however, the array will be positioned so that the center of the bottom surface of the first disk is located at the origin. The minimum x and y faces of unit 1 are the same and are 15 cm more negative than the axis of the disk. Locating the origin of the disk at the origin of the unit thus requires the x and y position to be -15. The bottom of unit 1 is on the bottom of the disk, so the z location is 0.

In the geometry block, press the CTRL-SPACE key combination to generate an autocomplete list, as shown in Figure 109, and select **global_unit**. Change the global unit number to 4. Between the global unit and the **TODO** comment, press the CTRL-SPACE key combination to generate an autocomplete list, as shown in Figure 68, and select **array - kenova (configurable)**. The default geometry block array configurable form is shown in Figure 116. The **Array Id** default value of **1** is correct, as that is the array identifier specified in the array block. The **x Min** and **y Min** values should be changed to -15, and the **z Min** can stay at the default value of **0**. The completed configurable form is shown in Figure 117; after confirming the input, press **OK** to add the array input to the global unit. The completed geometry block is shown in Figure 118.

Figure 116. Default configurable form for adding an array to a unit.

Figure 117. Completed configurable form for adding an array to a unit.

```
13 read geometry
14 unit 1
15   cylinder 1 1 10   2.5 0
16   cuboid    0 1 15 -15   15 -15   2.5 0
17 unit 2
18   cylinder 2 1 10   4 0
19   cuboid    0 1 15 -15   15 -15   4 0
20 unit 3
21     cuboid 3 1 15.0 -15.0 15.0 -15.0 3.0 0.0
22 global unit 4
23     array 1 -15.0 -15.0 0.0
24 end geometry
```

Figure 118. Complete geometry block for the model.

After confirming that the input is correct, press the **Run** button in the Fulcrum menu bar to execute CSAS5. The results are shown in the **Messages** pane, and the calculated **best estimate system k-eff** should be approximately 0.9541 ± 0.0018. Recall that results will vary based on different computers and operating systems, but the results should agree within the reported uncertainties on the two separate calculations.

4.7   SUMMARY

This section had the following objectives for users:

- Use Fulcrum to describe the basic shapes (sphere, cylinders, cube, cuboid).
- Understand how units are created (including nesting 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.

Now that the material and simple geometry input has been introduced, more complicated geometries such as hemispheres, hemicylinders, and holes can be used as well.

5.   ADVANCED GEOMETRY

The previous section covered some nested geometry and simple array problems. This section explains the more complex shapes (HEMISPHERE and HEMICYLINDER) and the keyword "hole" to create non-nested units.

5.1   WHAT YOU WILL BE ABLE TO DO

- Use Fulcrum to describe hemispheres and hemicylinders.
- Define units for partially filled shapes such as tanks or spheres.
- Understand how holes are used to include multiple non-nested shapes in a unit.
- Create arrays containing basic shapes and holes.

5.2   HEMISPHERES AND HEMICYLINDERS

In addition to the basic shapes discussed in Section 4.2, KENO V.a has two additional shapes used to describe parts of a cylinder or sphere. These "hemi" shapes are described in Table 3.

Table 3. "Hemi" shapes

Keyword	Description
HEMISPHERE	Partial sphere with a flat surface perpendicular to the X, Y, or Z axis. It is used to specify a spherical segment with a flat base for which the spherical surface exists in the positive Z direction. The base or flat portion of the spherical segment is centered around a point that may be specified in the optional region origin data. By default, the center of the spherical surface is the origin, and the distance to the base from the center of the spherical surface is zero (i.e., a true hemisphere). See Figure 119.
HEMISPHE*bc*	Used to specify a spherical segment with a flat base for which the spherical surface exists in the *bc* direction (b = + or –, c = X, Y, or Z). The base or flat portion of the spherical segment is located a distance ρ from the center of the spherical surface, and the center may be specified in the optional region origin data. **HEMISPHE+Z** is the same as the previously described **HEMISPHERE**, and **HEMISPHE-Z** is the mirror image of **HEMISPHE+Z**, therefore existing only in the negative Z direction. By default, the center of the spherical surface is the origin, and the distance of the base from the center of the spherical surface is zero (i.e., a true hemisphere).
*b*HEMICYL*cd*	Used to specify a cylindrical segment for which the axis is in the *b* direction (b = X, Y, or Z) and for which the cylindrical surface exists only in the *c* direction (c = + or –) from a plane perpendicular to the *d*-axis (d = X, Y, or Z). The position of the plane (cut surface) can be specified in the optional region chord data. This plane cuts the cylinder parallel to the axis at some distance, ρ, from the axis. By default, the axis passes through the origin and ρ is zero. (Examples: **ZHEMICYL+X, YHEMICYL-Z, XHEMICYL+Y**) See Figure 120.
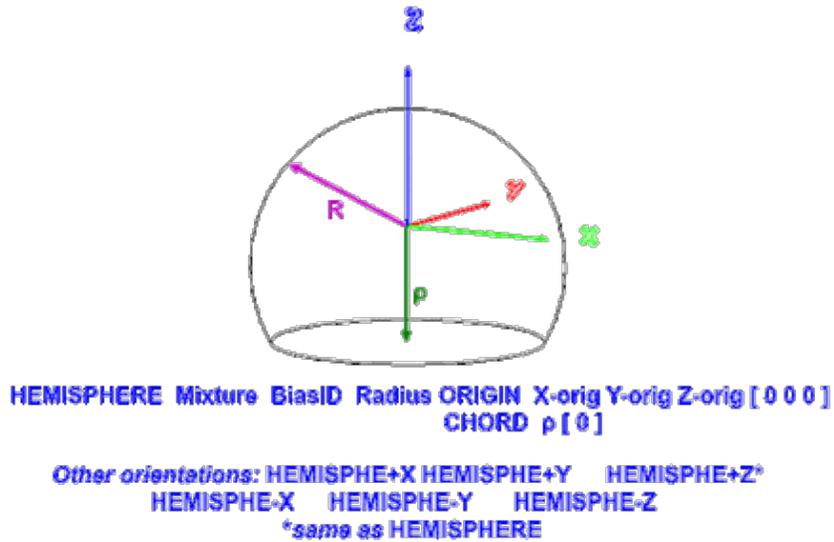
Figure 119. Parameters for a hemisphere.
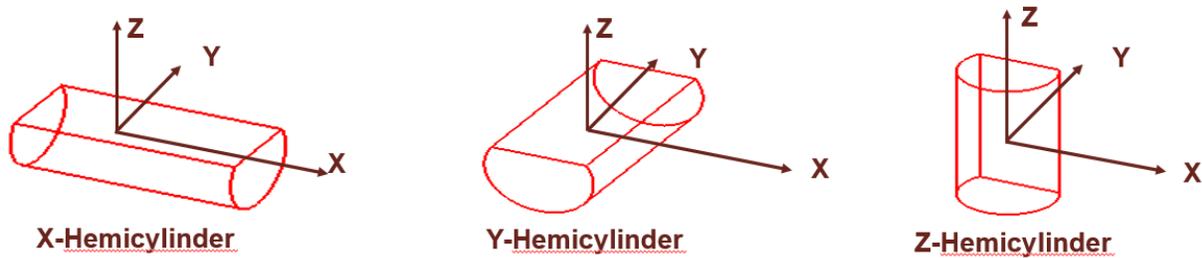
Figure 120. Hemicylinder examples.

5.2.1   Description of a Hemisphere or Hemicylinder

Two parameters are needed to describe the location of the flat surface of the hemisphere or hemicylinder: the axis and the distance from the center of the shape. For example, Figure 121 shows a flat surface perpendicular to the Z- or X-axis. The sign is determined by the curved surface. If the curved surface is in the positive direction, then the sign is +; if it is in the negative direction, then the sign is −.
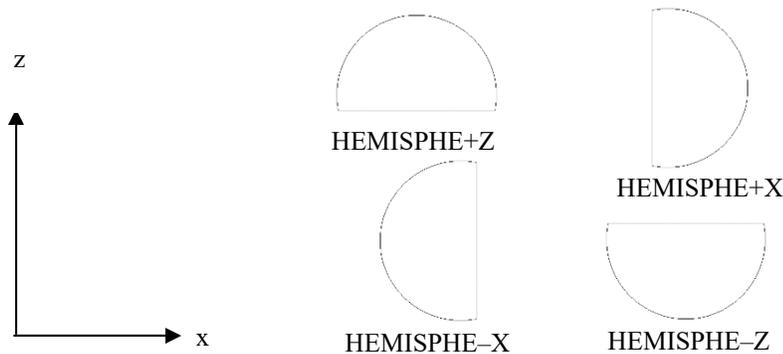
Figure 121. Direction of curvature.

The location of the flat surface is defined by ρ, the chord, as the distance from the origin to the flat surface. The sign of ρ indicates whether the flat surface is between the origin and the curved surface (sign is −) or whether it is on the other side of the origin from the curved surface (sign is +). In other words, a negative chord length leaves less than half of the sphere, and a positive chord length leaves more than half of the sphere. See Figure 122 for examples. The sign of ρ has no impact on where the curved surface is: that is entirely determined by the shape keyword used, as discussed in the previous paragraph.
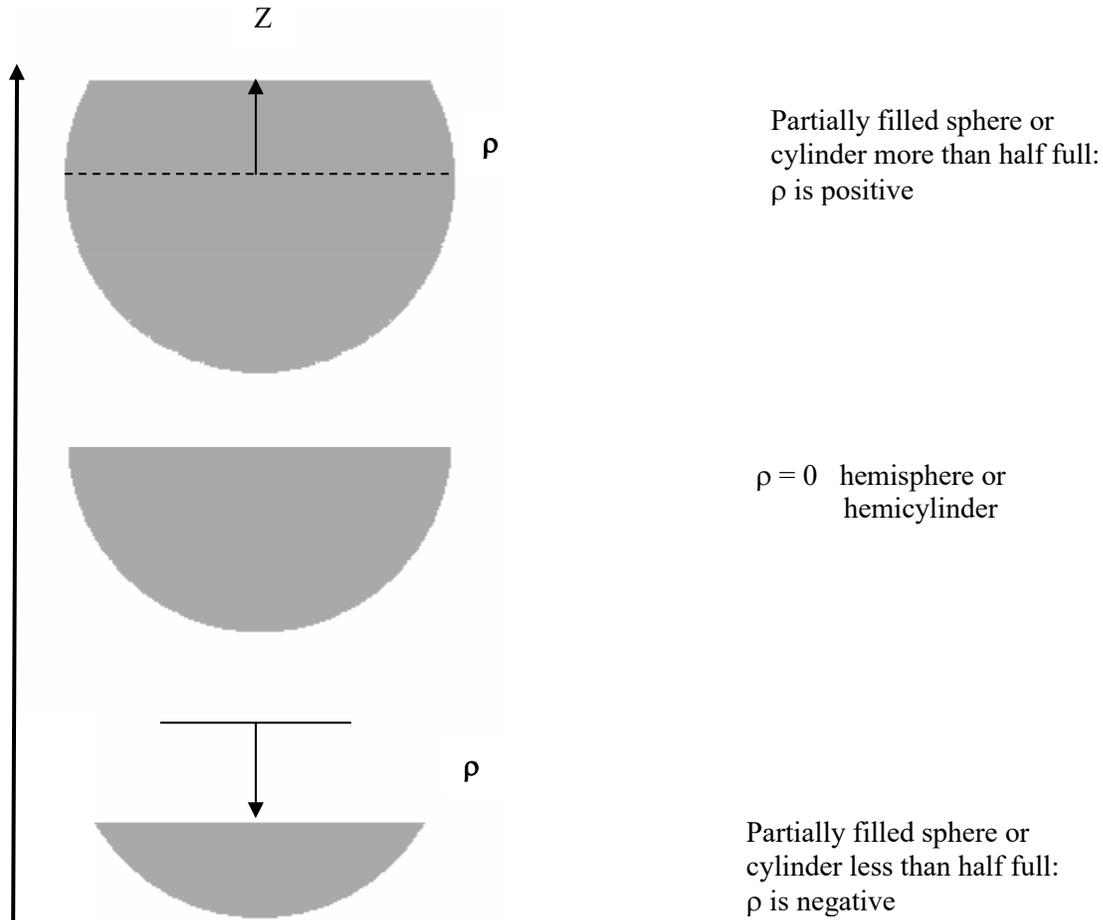
Figure 122. Use of ρ to characterize flat surface.

5.2.2 Partially Filled Flat-Bottomed Spherical Tank

An example problem is provided to illustrate how a hemisphere might be used to describe a partially filled flat-bottomed spherical tank. For this problem, the flat portion of the sphere is located 16 cm from the center, the tank is filled to a point 13.8 cm above the center, and the tank has a radius of 19.5 cm. The solution from Section 3.4.4 will be modeled in the sphere. The spherical tank wall is made of stainless steel with a wall thickness of 0.122 cm. The experimental arrangement is shown in Figure 123.
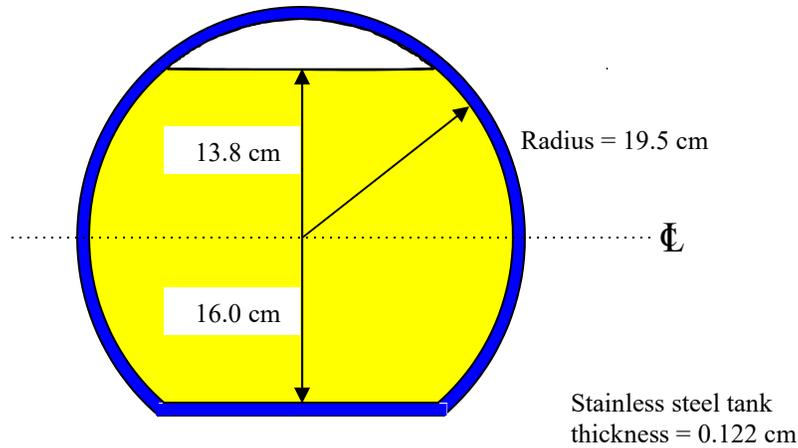
Figure 123. Flat-bottom sphere.

Using the information from earlier sections of this primer, use Fulcrum to open a new file, and enter the title, sequence, and cross section library. In the composition block, enter the solution from Section 3.4.4 and stainless steel 304 as a standard basic composition (**ss304**). The complete composition block is shown in Figure 124.

```
 5 read comp
 6  u-235 1 0.0 0.0005637 end
 7  u-238 1 0.0 0.0012802 end
 8  h      1 0.0 0.0597522 end
 9  o      1 0.0 0.0335605 end
10  f      1 0.0 0.0036844 end
11  ss304 2     end
12 end comp
```

Figure 124. Composition block for partially filled, flat-bottomed tank model.

For this geometry, there will be three hemispheres in one unit, and that unit will be the global unit for the problem. Remember that nested shapes must fully enclose all previous shapes in the unit. There are three shapes needed: the hemisphere containing the void inside the tank above the solution, the hemisphere containing the solution (and the void), and the hemisphere representing the stainless steel container (and the void, the solution). Begin with the smallest shape, which is the hemisphere containing the void. The specifications for a hemisphere are the axis perpendicular to the flat surface, the radius, and the chord.

Start by placing the cursor in the **read geometry** block on the line after global unit 1 and pressing the CTRL-SPACE autocomplete key combination. Select the **hemisphere - kenova (configurable)** option, which opens the hemisphere input form. The default type, **hemisphere**, is applicable to this problem because it has the flat surface perpendicular to the Z-axis and the curved part in the positive Z-direction. For material, enter 0 for void, and 19.5 for the **Radius.** Next, the input for the **Chord** must be determined and entered. The problem specification states that the surface is 13.8 cm above the center of the tank, so the chord *length* is 13.8. Remember that a chord length entered as negative indicates that the hemisphere is less than half full, so the chord input for this model is –13.8. Check the box next to the word **Chord** on the configurable form and enter −13.8. This creates a region like that shown in Figure 125. The complete hemisphere configurable form for this region is shown in Figure 126.
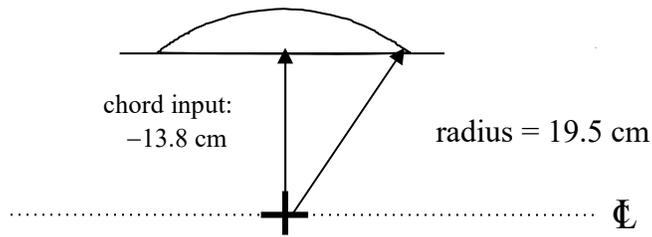
Figure 125. Hemisphere containing void.
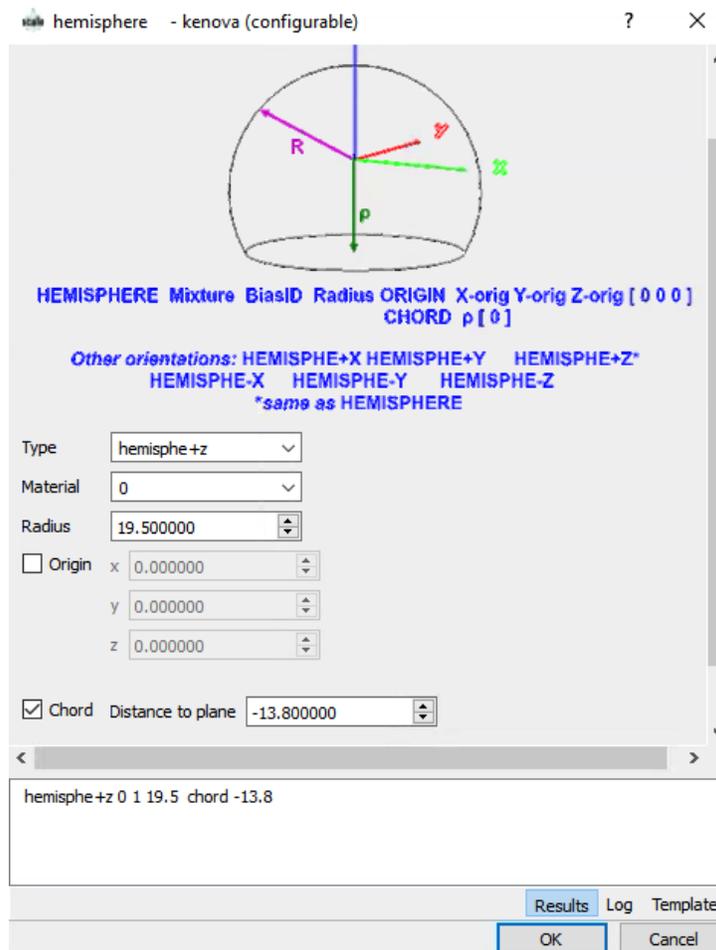
Figure 126. Hemisphere geometry form for the void region.

Keep in mind that a second shape only fills the region that is not already filled by the first shape. Thus, another hemisphere with the same origin can be entered containing the fissile solution. This will not fill the void, but it will fill the hemisphere from its bottom to the bottom of the void. Copy the existing **hemisphere** entry and paste it on the next line. Update the mixture number to 1 for the fissile solution. The radius stays the same, so only the chord length needs to change. The problem specification (see Figure 123) states that the flat bottom surface of the tank is 16 cm below the middle of the tank. Recall that a positive chord length makes the hemisphere more than half full. Therefore, the correct chord entry for this second hemisphere is 16. This entry indicates that the flat surface is 16.0 cm below the origin, on

the other side of the origin from the curved sphere surface, as shown in Figure 127. Remember that the second region contains the first region, but it does not fill that region. Thus, region 1 is a void in region 2.
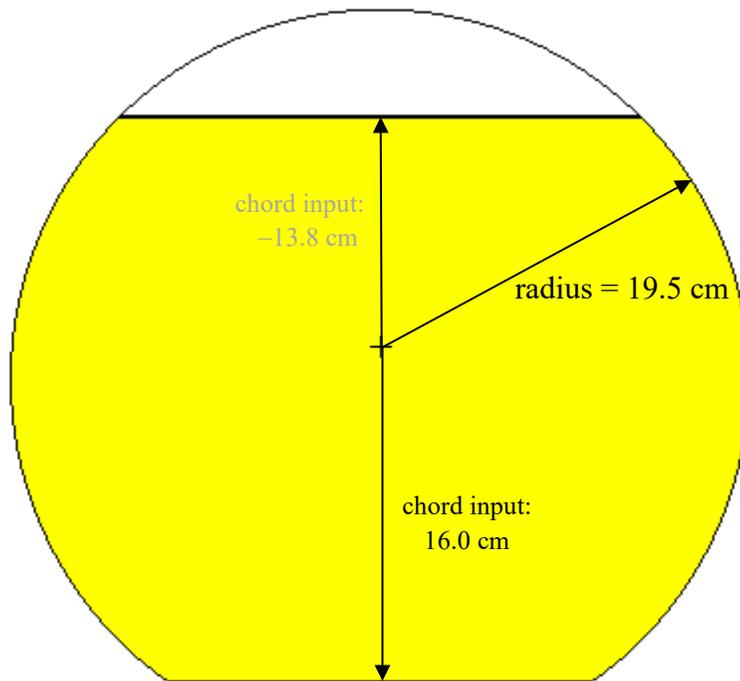
Figure 127. First and second region.

As noted above, the third shape only fills that region not already filled by the previous regions. Therefore, you will enter another hemisphere with the same origin but with a slightly larger radius and chord (rho). This will fill the thin outer edge of the hemisphere with stainless steel. Copy the second hemisphere and paste it on the next line of the input file. Change the material to mixture 2, which is stainless steel 304. Enter +0.122 directly after the radius (19.5), highlight the expression, and evaluate it using Fulcrum, as shown in Figure 128. This results in the outer radius of the tank being calculated to be 19.622. The chord must be larger by the tank thickness of 0.122 cm, as well, so change the chord length to 16.122. With the origin at the center of the sphere, this puts the flat surface 16.122 cm below the origin. A cutaway view of the final geometry is shown Figure 129. The complete input is shown in Figure 130.

Figure 128. Highlighting an expression for Fulcrum evaluation.
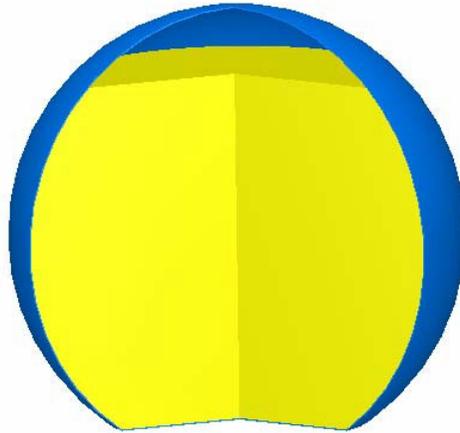
Figure 129. 3D view of nested hemispheres.

```
 1 =csas5 parm=(   )
 2 Primer Example: Hemispherical Tank
 3 ce_v7.1
 4 read comp
 5  u-235 1 0.0 0.0005637 end
 6  u-238 1 0.0 0.0012802 end
 7  h      1 0.0 0.0597522 end
 8  o      1 0.0 0.0335605 end
 9  f      1 0.0 0.0036844 end
10  ss304 2      end
11 end comp
12 read geometry
13 global unit 1
14   hemisphere 0 1 19.5    chord -13.8
15   hemisphere 1 1 19.5    chord 16
16   hemisphere 2 1 19.622   chord 16.122
17 end geometry
18 end data
19 end
```

Figure 130. Input file for nested hemispheres.

After confirming that the input is correct, press the **Run** button in the Fulcrum menu bar to execute CSAS5. The results are shown in the **Messages** pane, and the calculated **best estimate system k-eff** should be approximately 1.0879 ± 0.0021. Recall that results will vary based on different computers and operating systems, but the results should agree within the reported uncertainties on the two separate calculations.

5.3 HOLES

For many problems, the requirement of nested shapes does not interfere with creating an accurate model. However, there will be times when a unit contains more than one shape, and the shapes are not nested. For those situations, KENO V.a. uses a shape called a *hole*. There are five basic rules for holes.

1. A hole contains a single unit.
2. Holes may not intersect.
3. Holes may be nested.
4. As many holes as needed may be placed in a region as long as they do not intersect.
5. Holes are positioned by specifying the location of the origin of the hole in the new unit coordinate system.

5.3.1   Simple Hole Example

A system consisting of a graphite block and a cylinder of uranium in a steel can is a simple example of the use of holes. The geometric arrangement might look like that shown in Figure 131.
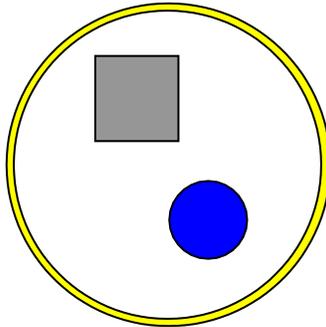
Figure 131. Hole example problem.

Because the graphite block and the uranium cylinder are not nested, they cannot be entered in a single unit. Therefore, two units are required: a uranium cylinder unit and a unit with the graphite block inside the steel cylinder. The graphite block could have been selected as the single unit, and the uranium cylinder could have been included inside the steel cylinder as the second unit. Using both the graphite block and uranium cylinder as holes inside the steel cylinder would also represent a legal geometry, but it would transport neutrons a little more slowly due to the algorithms involved with tracking in and out of holes. The process looks something like that shown in Figure 132.
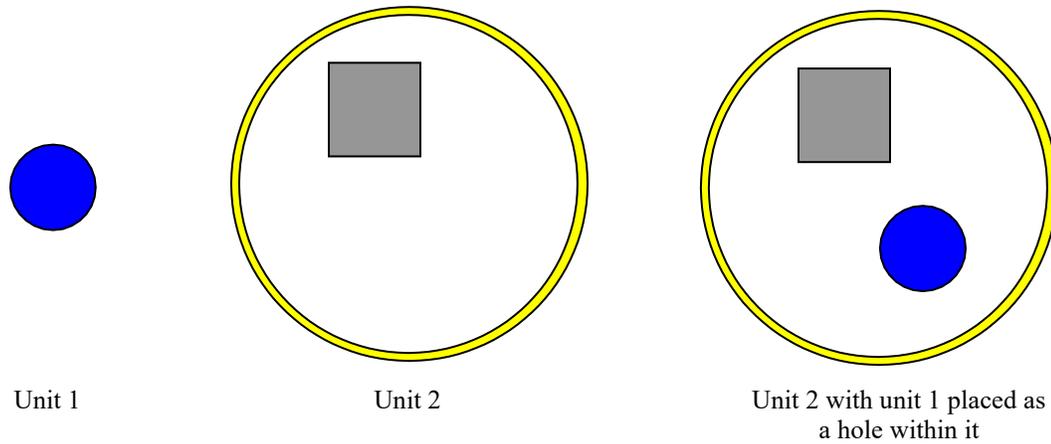
Figure 132. Using a hole to place one unit inside another.

The input for the process would be as follows. First, unit 1 will be the uranium cylinder. Then, unit 2 is created in four steps, as follows:

1. The first shape in unit 2 will be the block with the placement of its origin relative to the center of the steel cylinder.
2. The second shape in unit 2 will be the inside of the steel cylinder and will contain void as its material.
3. The third step is to place unit 1 as a hole inside unit 2. Because the hole is being placed in region 2 (the void), it must appear immediately after that region in the geometry description.
4. The third shape in unit 2 is the outside of the steel cylinder, which will contain steel. This forms the combination of unit 2 and unit 1, as shown in Figure 132.

This problem includes a pure ^{235}U metal cylinder (density = 18.7 g/cm^3) with a radius of 6.85 cm and a height of 20 cm. There is a block of graphite (default density) that is 12 cm long, 12 cm wide, and 20 cm high. The cylinder and block are placed in a stainless steel 316 (SS316) cylinder (default density) with an inside diameter of 40 cm and a wall thickness of 0.2 cm. The inside height of the SS316 cylinder is 20 cm. With the origin located at the center of the SS316 cylinder, the lower left corner of the graphite block is located at (-14, 2, -10). The center of the uranium cylinder is located at (6, -6, 0). The arrangement is shown in Figure 133.
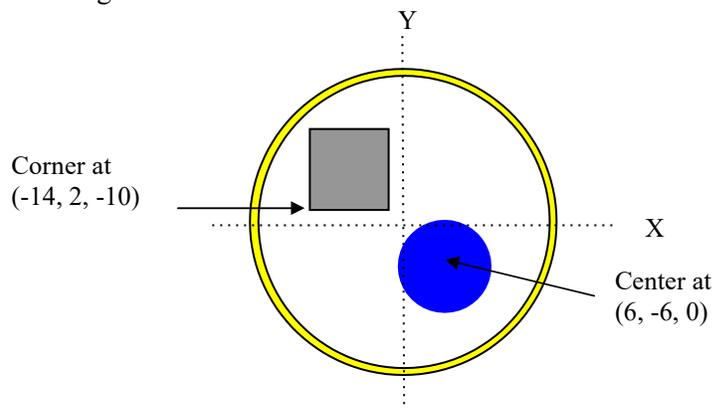
Figure 133. Plan view of holes example problem.

89

Create a new file and provide the sequence, title, and cross section library name. The composition block will contain graphite, uranium, and SS316, all of which are standard basic compositions. The uranium metal is modeled as pure ^{235}U with a density of 18.7 g/cm^3. The input file through the composition block is shown in Figure 134.

```
1 =csas5 parm=(    )
2 Primer example: Simple hole demo
3 ce_v7.1
4 read comp
5  graphite 1 end
6  u 2 den=18.7 1 293 92235 100 end
7  ss316 3 end
8 end comp
```

Figure 134. Composition block for simple hole example model.

For this geometry, follow the steps provided on the previous page. First, delete **global** from unit 1, as it will not be the global unit. The specification of the cylinder will come next. Holes are positioned in the unit into which they are inserted by specifying the location of the origin of the hole unit in the new unit coordinate system. This means that the location of the origin in this unit will impact the input for the hole placement in unit 2. In general, selecting the origin for a unit should be done with care; this is especially true for a unit that will be used as a hole. In this case, the origin will be placed at the bottom center of the uranium cylinder. Now specify the cylinder using either a configurable form or direct input. The radius is 6.85 cm, and since the height is 20 cm, the top of the cylinder is at 20 cm, and the bottom is at 0 cm. The complete unit 1 input specification is shown in Figure 135.

```
13 unit 1
14    zcylinder 2 1  6.85  20.0 0
```

Figure 135. Geometry input for uranium cylinder.

Below the uranium cylinder in unit 1, use the CTRL-SPACE key combination and select **unit - kenova (configurable)** to generate a unit configurable form. At the top middle of the form, select yes for **Global**, and set **Id** to 2.

The first region in unit 2 will be the graphite block, which is a cuboid. In the **Properties** pane on the left side of the form, select **New** > **cuboid** to generate the cuboid configurable form within the unit form. The default mixture number, 1, is correct in this instance, as it fills the cuboid with graphite. The surfaces of the cuboid must be specified to locate the block in the correct position relative to the cylinder. As the coordinates of the lower left-hand corner are (-14, 2, -10), then **-x** will be -14, **-Y** will be 2, and **-Z** will be –10. The maximum values are calculated by adding the appropriate dimension (12×12×20) to the minimum values. Thus **+x** will be −2, **+y** will be 14, and **+z** will be 10. *Do not* click **OK**. This will add the unit to the input file, but the entire unit has not yet been specified.

The second region in unit 2 will be the inside of the SS316 cylinder and will contain void as its material. In the **Properties** pane, select **New** > **zcylinder** to generate a configurable form for the region inside the unit configurable form. Enter **Material** 0 to fill the inner volume with void. Set the **radius** to 20, **+z** to 10, and **-z** of −10. Again, *do not* click **OK**.

The next entry is the hole. A hole must be specified immediately after the region into which it is being inserted, so the uranium cylinder hole must be specified here directly after the void inside the stainless steel cylinder. Placing the hole elsewhere in the unit 2 description will result in an error message indicating that hole 1 intersects with region 3. This type of message usually indicates that the hole does not follow the region in which it is located or that the placement of the origin of the hole is incorrect.

To enter data on the hole, select **New** > **Hole** in the **Properties** pane. The default **Unit Id** of **1** is applicable for this model. The uranium cylinder is centered at (6, -6) and extends from z=-10 to z=10 in the unit 2 coordinate system. Recall that the origin of unit 1 is located at the bottom center of the cylinder. This point must therefore be located at (6, -6, -10) with respect to the SS316 cylinder origin. Enter 6 for **x**, −6 for **y**, and −10 for **z**. One region remains to be added to unit 2, so *do not* click **OK**.

The last region in unit 2 is the outside of the SS316 cylinder. In the **Properties** pane, select **New** > **Zcylinder** to generate a cylinder configurable form in the unit form. The **Material** field should be set to 3 for SS316. The wall thickness has been specified as 0.2 cm, so the **Radius** is 20.2, **+z** is 10.2, and **-z** is −10.2. The configurable form for the entire unit is shown in Figure 136. After confirming that the input for all of unit 2 is correct in the **Results** pane, press **OK** to add unit 2 to the input file. The input file is shown in Figure 137, and a 3D rendering of the geometry is shown in Figure 138.
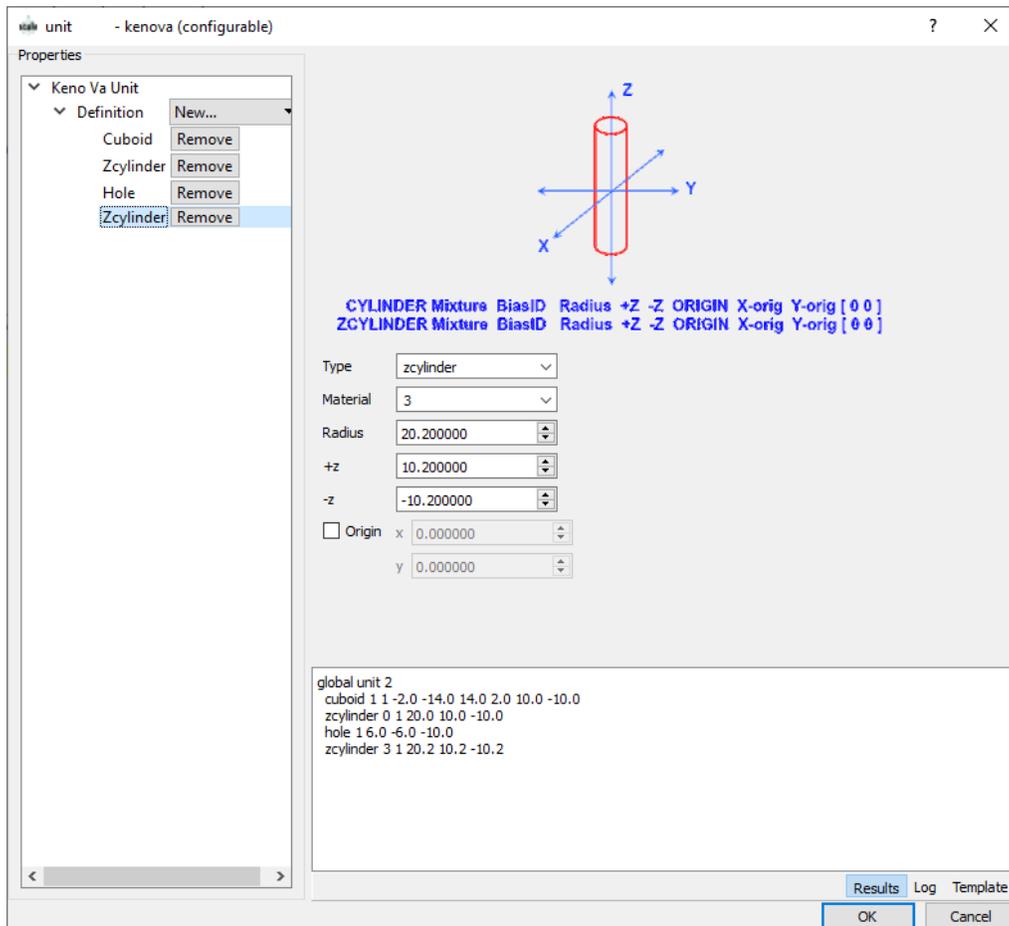
Figure 136. Geometry form for holes example problem.

```
 1 =csas5 parm=(    )
 2 Primer example: Simple hole demo
 3 ce_v7.1
 4 read comp
 5  graphite 1 end
 6  u 2 den=18.7 1 293 92235 100 end
 7  ss316 3 end
 8 end comp
 9 read geometry
10 unit 1
11    zcylinder 2 1  6.85  20.0 0
12 global unit 2
13    cuboid 1 1 -2.0 -14.0 14.0 2.0 10.0 -10.0
14    zcylinder 0 1 20.0 10.0 -10.0
15    hole 1 6.0 -6.0 -10.0
16    zcylinder 3 1 20.2 10.2 -10.2
17 end geometry
18 end data
19 end
```

Figure 137. Input file for holes example problem.
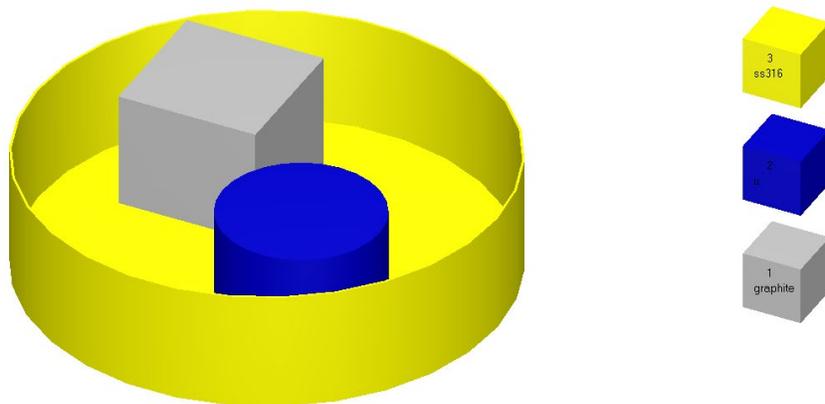
Figure 138. 3D view of geometry for hole example problem with top half removed.

After confirming that the input is correct, press the **Run** button in the Fulcrum menu bar to execute CSAS5. The results are shown in the **Messages** pane, and the calculated **best estimate system k-eff** should be approximately 0.9997 ± 0.0017. Recall that results will vary based on different computers and operating systems, but the results should agree within the reported uncertainties on the two separate calculations.

5.4   USING HOLES AND ARRAYS

Many problems require the use of multiple arrays, such as multiple fuel assemblies. KENO V.a is limited to including only a single array directly into a unit because an array cannot entirely contain another array. Depending on the exact system, one of two options may be used to represent multiple arrays in a single model. The unit with the array may then be used in an array, creating an array of arrays, if the system includes such a regular array. This is typically the case for commercial fuel assembly storage arrays at nuclear power plants or in large spent fuel storage systems. The other more general scenario is to use the unit containing the array as a hole and place it as needed in the geometry. This approach can be used in some fuel assembly transportation cask models and other scenarios. In the following example, a model

akin to many critical experiments involving fuel pins is provided as a demonstration of this latter approach to including multiple arrays in a model.

5.4.1 Pins in a Tank

This example problem includes metal rods in a square-pitched array in a cylindrical tank. A summary of the system follows.

   Rods
         0.75 cm outer diameter (OD) × 30.0 cm length
         Pitch: 1.0 cm
         Fuel: 20 wt% enriched ^{235}U metal with default density
         (No ^{234}U or ^{236}U in fuel)

   Tank
         18 cm inner diameter
         60 cm inside height
         Aluminum wall and bottom 0.5 cm thick
         Tank filled to a height of 45 cm with H_2O
         Tank reflected on the bottom and sides with 30 cm H_2O
         Fuel rods sitting on the tank bottom

Figure 139 shows the lower half of the tank without the external reflecting water, Figure 140 is a plan view of the tank with the reflector, and Figure 141 shows an elevation view of the configuration.
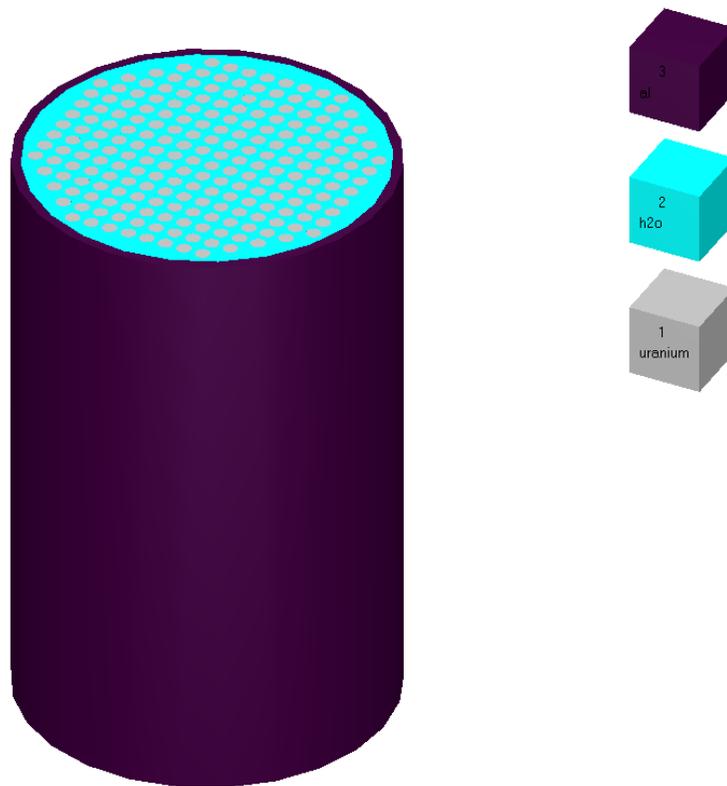
Figure 139. Lower half of the tank without the reflector.
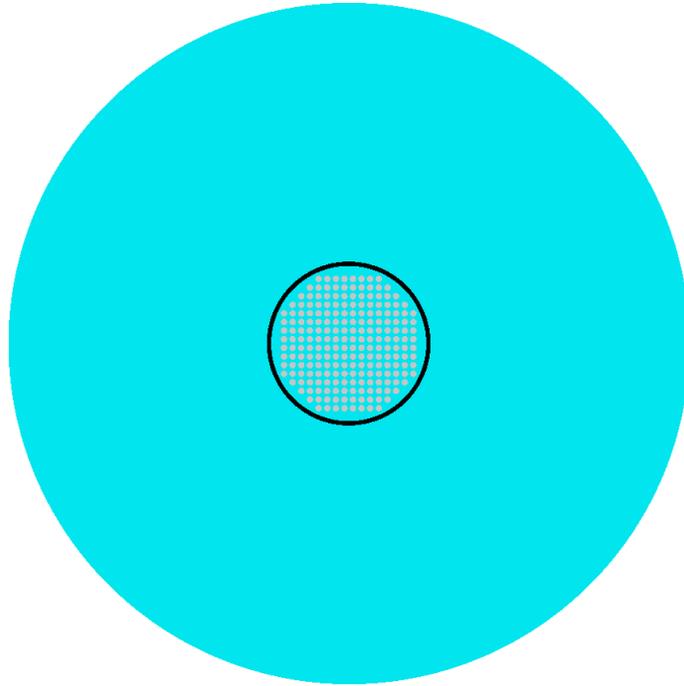
Figure 140. Radial slice through midplane of model.
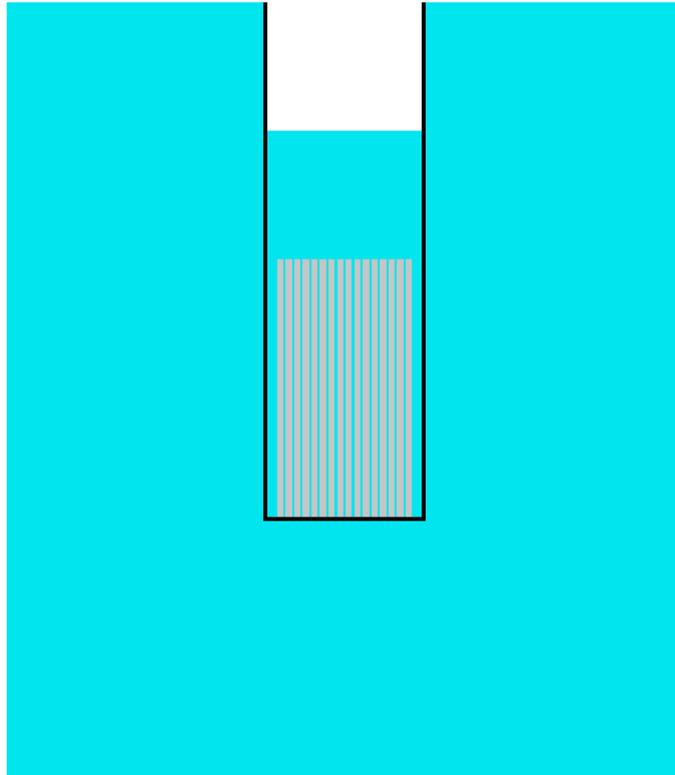
Figure 141. Elevation view of model.

Based on these views, it may seem like an array would be a simple solution, but the full array of pins cannot be described using KENO V.a geometry. Recall that each unit in KENO V.a must be fully encompassed by any unit in which it is placed. The 16×16 array of pins in this model is too large to be contained inside the tank; there is no method in KENO V.a to truncate the corners of an array. Therefore, a single array cannot be used here. The configuration can be described by placing each fuel rod as a hole in the tank unit. While this approach would accurately model the system, the model would run very slowly because of the large number of holes. As mentioned previously, neutron transport is slowed down in KENO V.a by the algorithms associated with tracking particles entering and leaving holes. Generating the input would also be tedious and error prone. A better approach is needed.

The key to solving this example is dividing the array into several smaller arrays. A central core array is centered in the tank, and several smaller arrays of similar size are placed as holes. The following example presents one of several possible approaches to modeling the configuration.

First enter the necessary information in the title card line, cross section library, and material definitions. This part of the input is shown in Figure 142.

```
 1 =csas5 parm=(   )
 2 Primer example: Pins in a tank
 3 ce_v7.1
 4 read comp
 5  u 1   1.0 293 92235 20 92238 80 end
 6  h2o 2 end
 7  al 3 end
 8 end comp
```

Figure 142. Composition block for pins in a tank configuration.

The smallest unit is the individual pin cell: a metal rod centered inside a water cuboid. Remember from Section 4.6.1 that sides of the cuboid should be set equal to the pitch. The water depth (45 cm) is greater than the pin length (30 cm), but the unit here will be set to the pin height. To create the input, first remove **global** from unit 1 as this will not be the global unit. Next, specify a cylinder containing uranium (mixture 1) with a radius of 0.375, a top of 30 and a bottom of 0 using either direct input or a configurable form. Finally, add a cuboid filled with water (mixture 2) around the cylinder. The cuboid will set the spacing between rods, specified as 1 cm, so the x and y faces will be at ±0.5 and the z faces will be at 30 and 0 to completely contain the previous region. The complete input for this unit is shown in Figure 143.

```
10 unit 1
11  cylinder 1 1   0.375   30 0
12  cuboid 2 1   0.5 -0.5   0.5 -0.5   30 0
```

Figure 143. Geometry input for unit 1.

Figure 144 shows how the solution described here defines the respective arrays. The red dots are the respective points used to place the arrays.

- Array 1 is the central array that is 12 × 12 units.
- Array 2 is a short vertical array that is 1 × 8 units and is in unit 2.
- Array 3 is a long vertical array that is 1 × 10 units and is in unit 3.
- Array 4 is a short horizontal array that is 8 × 1 units and is in unit 4.
- Array 5 is a long horizontal array that is 10 × 1 units and is in unit 5.
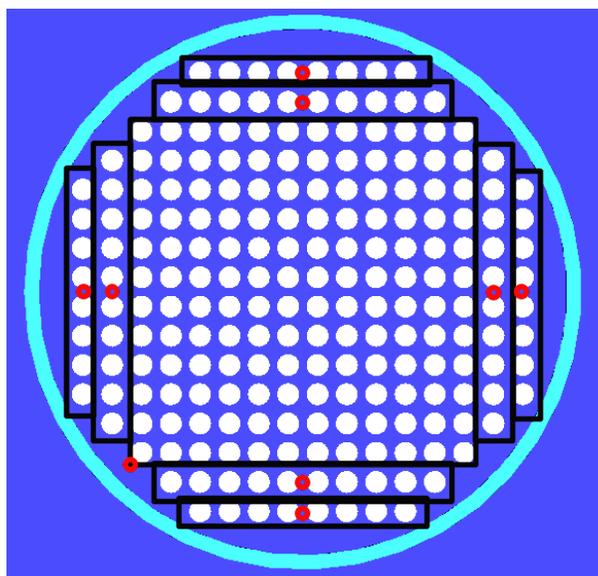
Figure 144. Pins-in-a-tank arrays.

Each of the five arrays will need to be defined in the array block and then subsequently placed into a unit. The four arrays on the side of the tank will be placed in units which will subsequently be placed into the global unit as holes; the central array will be placed directly into the global unit. The central array will be declared first in the input, though the order of the arrays is entirely arbitrary. The two arrays of pins in the y direction will be declared next, and the two in the x direction will be declared last.

Add an array block to the input file using either autocomplete or direct entry. An input shortcut in the array block allows the user to specify a unit that will fill all elements in the array from the current element to the end of the array by prefacing the unit number with the letter *f*. This option will be used in all five arrays in this model because only one unit is used in the arrays. The central array (ara=1) is a 12×12 array, so nux and nuy=12 and nuz=1. If the array is being generated with the array configurable form, then initialize all elements with unit 1. For direct entry and a shorter input, enter `fill f1 end fill` after specifying **nux**, **nuy**, and **nuz**. Figure 145 provides the input from the direct entry approach, incorporating the input shortcut, and Figure 146 provides the input generated by an array configurable form. Array 2 (ara=2) is one unit in the x direction and 8 in the y direction, and array 3 is, similarly, a 1×10 array. Arrays 4 and 5 are the same except that they are an 8×1 array and a 10×1 array, respectively. The entire array block using the fill shortcut is shown in Figure 147.

```
16  ara=1 nux=12 nuy=12 nuz=1
17   fill f1 end fill
```

Figure 145. Simplified array 1 specification using the fill shortcut.

```
19   ara=1
20   prt=yes
21   nux=12
22   nuy=12
23   nuz=1
24   fill
25 ' z = 1
26     1 1 1 1 1 1 1 1 1 1 1 1
27     1 1 1 1 1 1 1 1 1 1 1 1
28     1 1 1 1 1 1 1 1 1 1 1 1
29     1 1 1 1 1 1 1 1 1 1 1 1
30     1 1 1 1 1 1 1 1 1 1 1 1
31     1 1 1 1 1 1 1 1 1 1 1 1
32     1 1 1 1 1 1 1 1 1 1 1 1
33     1 1 1 1 1 1 1 1 1 1 1 1
34     1 1 1 1 1 1 1 1 1 1 1 1
35     1 1 1 1 1 1 1 1 1 1 1 1
36     1 1 1 1 1 1 1 1 1 1 1 1
37     1 1 1 1 1 1 1 1 1 1 1 1
38   end fill
```

Figure 146. Array 1 input from a configurable form.

```
15 read array
16  ara=1 nux=12 nuy=12 nuz=1
17   fill f1 end fill
18  ara=2 nux=1 nuy=8 nuz=1
19   fill f1 end fill
20  ara=3 nux=1 nuy=10 nuz=1
21   fill f1 end fill
22  ara=4 nux=8 nuy=1 nuz=1
23   fill f1 end fill
24  ara=5 nux=10 nuy=1 nuz=1
25   fill f1 end fill
26 end array
```

Figure 147. Array block for pins in a tank model using the fill shortcut.

The arrays must now be entered into units in the geometry block so that they can be positioned as indicated in Figure 144. Recall from Section 4.6 that arrays are positioned in a unit based on the minimum x, minimum y, and minimum z point in the array. Also recall from Section 5.3 that holes are positioned in a unit by specifying the location of the origin of the hole in the new unit. This model is an example of the scenario in which placing the array carefully may make placing the hole easier later in model construction. For instance, array 2, the array of 8 pins in the y direction, will be placed on both the +x and -x sides of the tank. Placing this unit as a hole twice is simpler if the origin is at the center of the unit; the same absolute value for the x coordinate is then used for both arrays. Planning the model and performing some arithmetic at this point will eliminate or simplify the work of assembling the model

97

later. For this reason, arrays 2–5 are all positioned such that the origin of their respective unit is in the center of the array.

Back in the geometry block, just after unit 1, press the CTRL-SPACE key combination to generate the autocomplete dropdown menu and select **unit - kenova (configurable)**. This will not be the global unit; change the **Id** to 2. This unit will contain only array 2 and will be used as a hole in the global unit. In the **Properties** pane, click **New** > **Array** to generate the array configurable geometry form in the unit form. As stated previously, this unit will contain array 2, so set the **Array Id** to 2 at the top middle of the form. The array is a 1×8 array of fuel rods, and the desired result is for the origin of the unit to be at the center of the array. In the x direction, the units are 1 cm wide (specified as spanning ±0.5 cm in unit 1), so the minimum x location (**x Min**) should be specified as −0.5 so that the origin of the unit is in the center of the array. Similarly, the array is 8 cm in length in the y direction, so the minimum y location (**y Min**) should be set to −4. The minimum z value (**z Min**) is not crucial at this point, but it will obviously be relevant to positioning the unit as a hole; a minimum z location of 0 is useful and will be used as the bottom of the fuel rods throughout the model specification. The completed unit configurable form is shown in Figure 148; after confirming that the input is correct, press OK to add the unit specification to the input file.
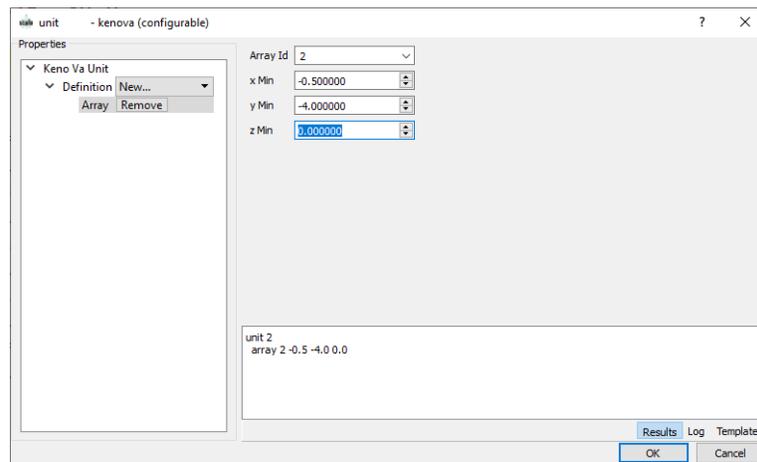
Figure 148. Unit configurable form for unit 2.

Units 3–5 will be specified in the same way, containing arrays 3–5. Array 3 should be positioned with its minimum point at (–0.5, –5, 0) because it is a 1×10 array. Units 4 and 5 are analogous to units 2 and 3, but the arrays are in the x direction. Therefore, array 4 should be positioned at (–4, –0.5, 0) in unit 4, and array 5 should be positioned at (–5, –0.5, 0) in unit 5. The completed input for all 4 of these units is shown in Figure 149.

```
13 unit 2
14    array 2 -0.5 -4.0 0.0
15 unit 3
16    array 3 -0.5 -5.0 0.0
17 unit 4
18    array 4 -4.0 -0.5 0.0
19 unit 5
20    array 5 -5.0 -0.5 0.0
```

Figure 149. Geometry input for units 2 through 5 containing arrays 2 through 5.

The final unit is unit 6, which will be the global unit. This unit will consist of the central 12×12 array of rods, the water inside the tank, the void above the water inside the tank, the tank wall, and the water reflector. Each region completely encloses the regions that came before it with the exception of the holes. The holes will be specified after the water inside the tank because the holes must be included in the input directly after the region into which they are inserted. All of this input can be specified using the unit configurable form in Fulcrum.

Just after unit 5, press the `CTRL-SPACE` key combination to generate the autocomplete dropdown menu and select **unit - kenova (configurable)**. This will be the global unit, so select `yes` in the **Global** dropdown menu. Change the **Id** to `6`. The first region to be specified is the central array, so in the **Properties** pane, select **New** > **Array**. The default **Array Id** 1 is applicable for this case. The subsequent geometry regions are all cylinders which are built along the z axis by default, so it is convenient to locate this array such that the origin is at the center. It is a 12×12 array, and each unit is 1 cm in extent in both the x and y directions, so the minimum point should be specified at (–6, –6, 0). Do not press **OK** until all the regions and holes have been specified. The next region is the water inside the tank; recall that the radius for this region is 9 cm, and the axial extent is from 0 cm to 45 cm. This region is larger than the array that comes before it and thus meets the rule for each region completely containing the prior regions. Add the water region by selecting **New** > **Zcylinder** from the **Definition** field in the **Properties** pane with mixture `2` specified as the **Material**. Once again, *do not* press **OK**.

The next 8 entries in the global unit are the holes to add the fuel rod arrays on the sides of the central array. Remember that holes are positioned by specifying the location of the origin in the new coordinate system. The 8 hole locations, with red marks for the origins of the holes, are shown in Figure 144. The arrays with 10 pins are adjacent to the central array, so the origin will be positioned 6.5 cm from the origin. Add unit 3 first; select **New** > **Hole** in the **Properties** pane and set **Unit Id** to `3`. The origin of the unit needs to be placed 0.5 cm to the left (more negative x) of the central array, so **x** should be set to `-6.5`. The **y** and z coordinates of the hole **Origin** are 0. Select **New** > **Hole** again, and set **Unit Id** to `3` a second time. This time position the unit with **x Origin** of `6.5` and **y** and **z** values of 0. This is an example of using a unit multiple times as a hole in a model; this capability can significantly simplify the input of systems with duplicated instances of a component. Unit 5 can now be added as a hole positioned at (0, 6.5, 0) and again as a hole positioned at (0, –6.5, 0). Similarly, Unit 2 can be added at (7.5, 0, 0) and (-7.5, 0, 0), and unit 4 can be added at (0, 7.5, 0) and (0, –7.5, 0). This completes the specification of the entire set of pins in the tank for this model. *Do not* press **OK** since the remaining regions in the global unit still must be specified.

The next region to specify is the void above the water inside the tank. This region must be next to completely contain the water inside the tank and to be contained by the tank wall region to come next. Select **New** > **Zcylinder** in the **Properties** pane. Set the **Material** to `0` (void) and the **Radius** to `9`. The void region is 15 cm tall, so the **+z** surface is `60`. For this region to completely enclose the prior region, the **-z** surface must be set to `0` for this model. Create another **zcylinder** for the aluminum tank wall. The **Material** is `3`, the **Radius** is `9.5`, **+z** is `60`, and **-z** is `−0.5`. The top of the tank is left open by specifying the same top surface for both the void cylinder and tank cylinder. The final region is the water reflector, so create another **Zcylinder** with **Material** `2` for water. The **Radius** is `39.5`, **+z** is `60`, and **-z** is `−39.5`. This region adds a 30 cm water reflector in the radial direction and on the bottom of the tank, as shown in Figure 140 and Figure 141. The complete global unit 6 configurable form is shown in Figure 150; after confirming that the input is correct, press **OK** to add it to the file. The completed geometry block is shown in Figure 151.
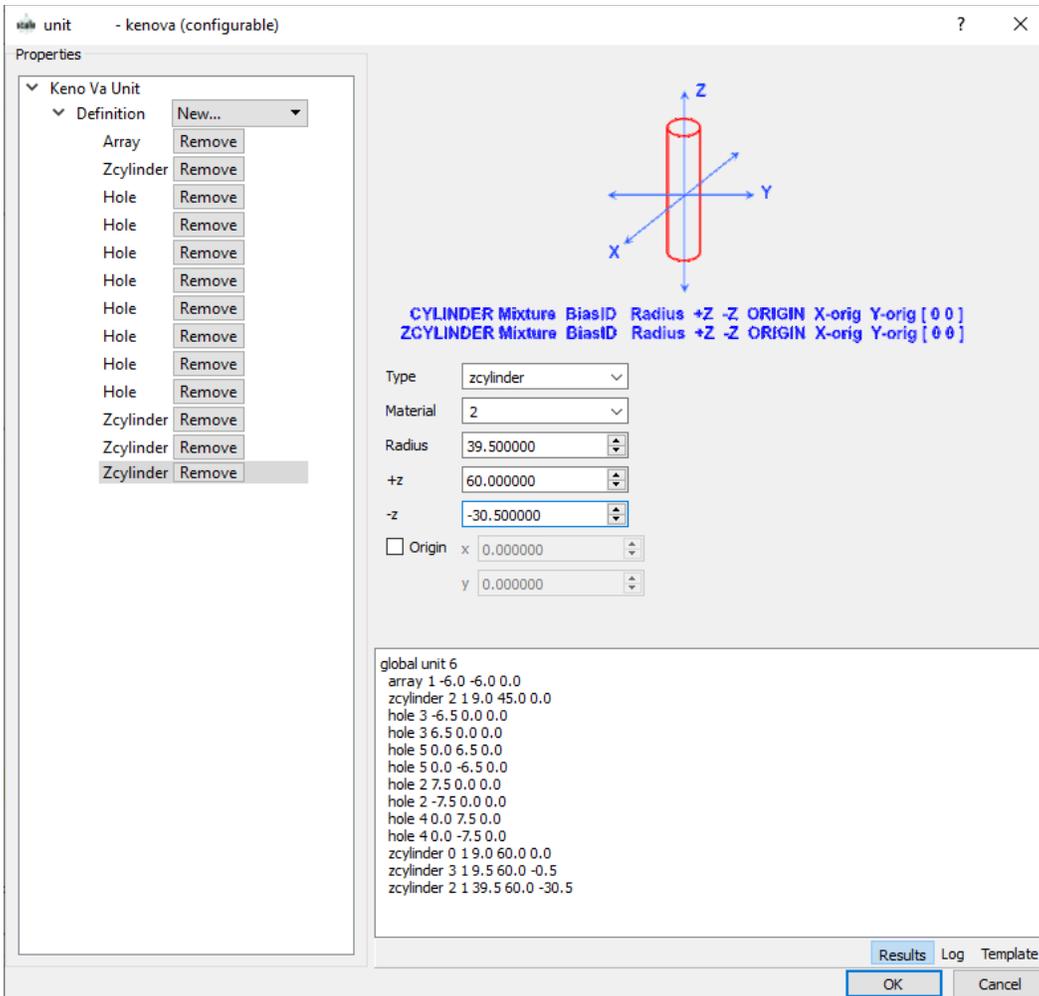
Figure 150. Complete unit configurable form for global unit 6 for the pins in a tank model.

```
 9 read geometry
10 unit 1
11   cylinder 1 1  0.375  30 0
12   cuboid 2 1   0.5 -0.5   0.5 -0.5   30 0
13 unit 2
14    array 2 -0.5 -4.0 0.0
15 unit 3
16    array 3 -0.5 -5.0 0.0
17 unit 4
18    array 4 -4.0 -0.5 0.0
19 unit 5
20    array 5 -5.0 -0.5 0.0
21 global unit 6
22    array 1 -6.0 -6.0 0.0
23    zcylinder 2 1 9.0 45.0 0.0
24    hole 3 -6.5 0.0 0.0
25    hole 3 6.5 0.0 0.0
26    hole 5 0.0 6.5 0.0
27    hole 5 0.0 -6.5 0.0
28    hole 2 7.5 0.0 0.0
29    hole 2 -7.5 0.0 0.0
30    hole 4 0.0 7.5 0.0
31    hole 4 0.0 -7.5 0.0
32    zcylinder 0 1 9.0 60.0 0.0
33    zcylinder 3 1 9.5 60.0 -0.5
34    zcylinder 2 1 39.5 60.0 -30.5
35 end geometry
```

Figure 151. Complete geometry block for the pins in a tank model.

After confirming that the input is correct, press the **Run** button in the Fulcrum menu bar to execute CSAS5. The results are shown in the **Messages** pane, and the calculated **best estimate system k-eff** should be approximately 0.8681 ± 0.0016. Recall that results will vary based on different computers and operating systems, but the results should agree within the reported uncertainties on the two separate calculations.

5.5   SUMMARY

This section covered the following topics:

- Use Fulcrum to describe hemispheres and hemicylinders.
- Define units for partially filled shapes such as tanks or spheres.
- Understand how holes are used to include multiple non-nested shapes in a unit.
- Create arrays containing basic shapes and holes.

6. 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.

6.1 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.

6.2 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 (Figure 136). 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.

6.2.1 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/cm^3. 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 152.

The configurable form should appear and should look like Figure 153, 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 154 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 155 and create input like that shown in Figure 156.

```
 5 read comp
 6 
 7 stdcomp      - basic (configurable)
   stdcomp      - basic
 8 stdcomp      - basic + volume fraction
   stdcomp      - basic + volume fraction + temperature
 9 stdcomp      - basic + volume fraction + temperature + isotopics
   stdcomp      - basic + atomic density (configurable)
10 stdcomp      - basic + atomic density
   stdcomp      - basic + atomic density + temperature
11 wtptcomp     - basic (configurable)
   wtptcomp     - basic
12 wtptcomp     - basic + volume fraction
   wtptcomp     - basic + volume fraction + temperature
   wtptcomp     - basic + volume fraction + temperature + isotopics
   atomcomp     - basic (configurable)
13 atomcomp     - basic
   atomcomp     - basic + volume fraction
14 atomcomp     - basic + volume fraction + temperature
   atomcomp     - basic + volume fraction + temperature + isotopics
   solution     - rho      + density + temperature + volume fraction
15 solution     - molar    + density + temperature + volume fraction
   solution     - massfrac + density + temperature + volume fraction
16 solution     - molefrac + density + temperature + volume fraction
   solution     - molality + density + temperature + volume fraction
17 ' TODO: add units below
```

Figure 152. Dropdown menu selection of atom compound configurable.
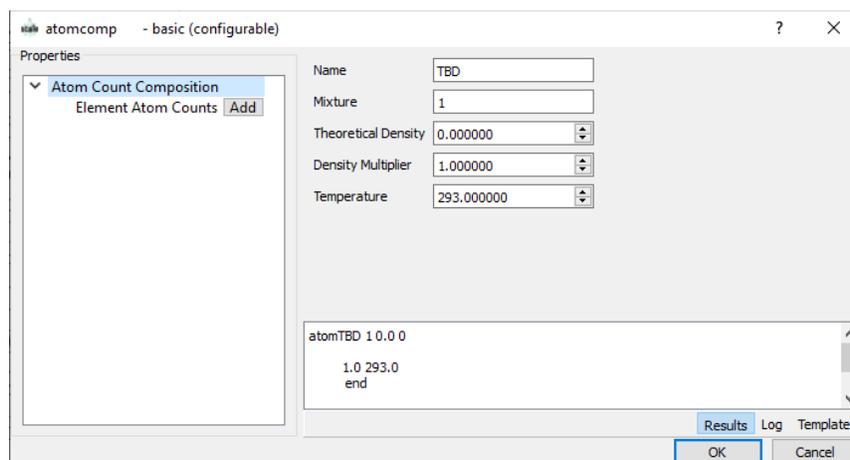
Figure 153. Depiction of initial atom compound configurable.
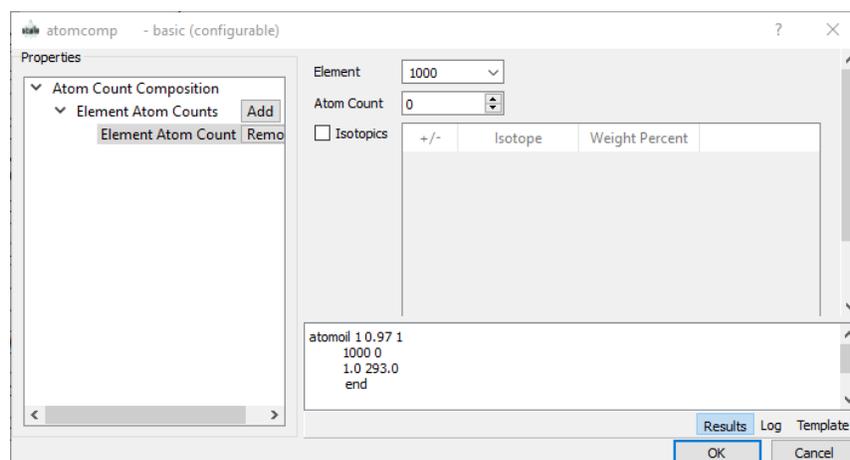
Figure 154. Depiction of element addition portion of the compound configurable.

104
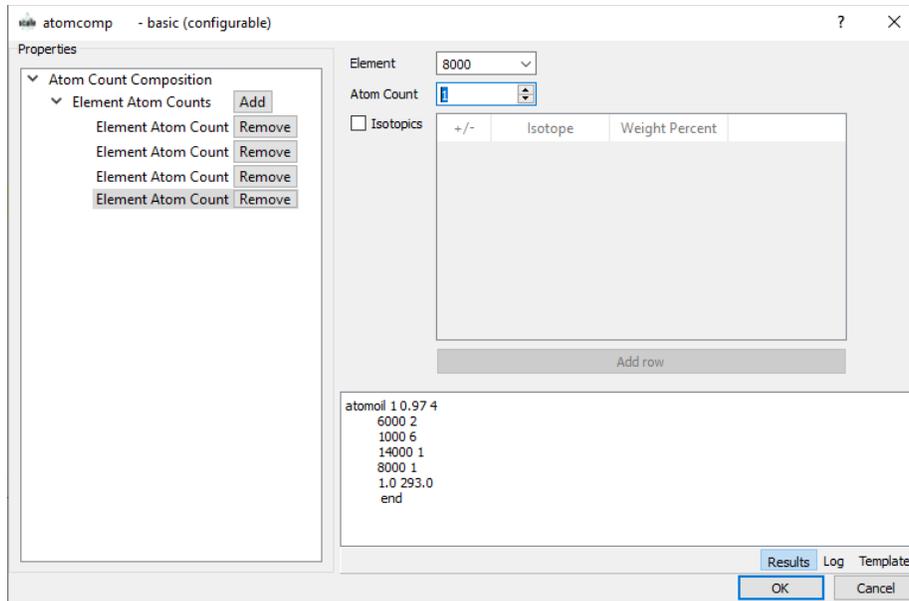
Figure 155. Depiction of element addition portion of the compound configurable.

Figure 156. User defined atom composition final input.

6.2.2   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 157.
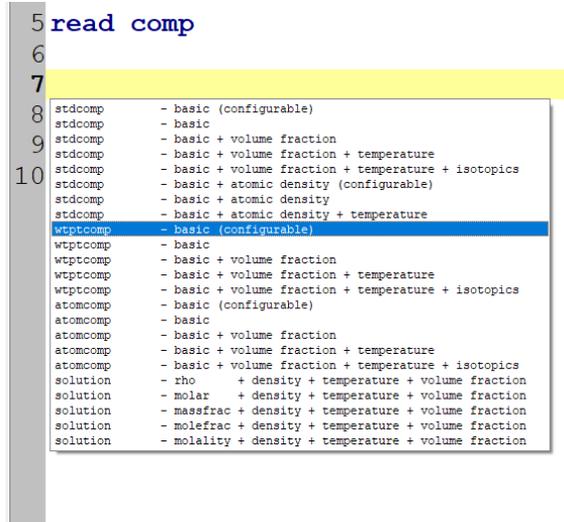
Figure 157. Dropdown menu selection of wtpt compound configurable.

A configurable form similar to that used for the atom compound material (Figure 153) entries will appear. For **Composition name**, replace **TBD** with `bal` to create the material `wtptbal`. Now enter the **Theoretical Density** of 2.65 g/cm^3 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 158. If the configurable looks acceptable, click **OK**, and the final input will look like Figure 159.
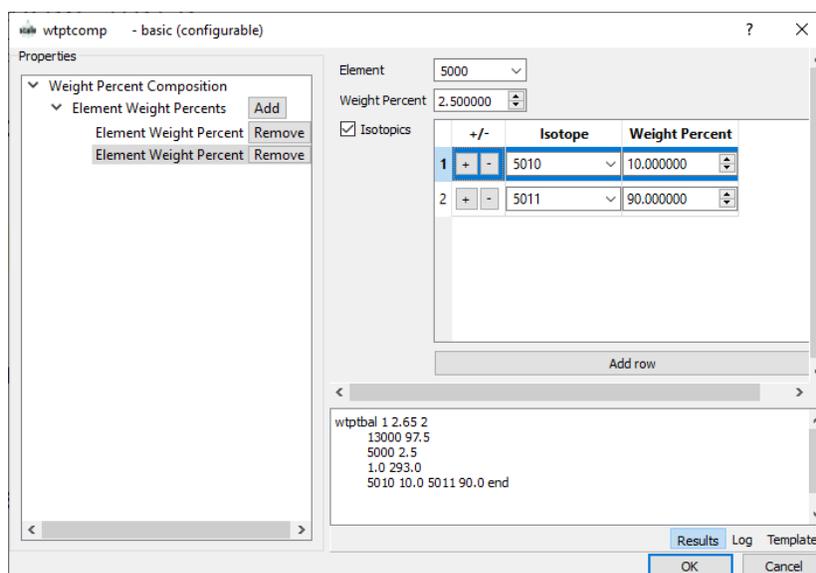
Figure 158. Final configurable for wtpt composition entry.

106

```
 5 read comp
 6
 7
 8 wtptbal 1 2.65 2
 9          13000 97.5
10          5000 2.5
11          1.0 293.0
12          5010 10.0 5011 90.0 end
13 end comp
14
```

Figure 159. User defined wtpt composition final input.

6.2.3   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 160. 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 161.

```
 5 read comp
 6
 7 stdcomp        - basic (configurable)
   stdcomp        - basic
   stdcomp        - basic + volume fraction
 8 stdcomp        - basic + volume fraction + temperature
   stdcomp        - basic + volume fraction + temperature + isotopics
 9 stdcomp        - basic + atomic density (configurable)
   stdcomp        - basic + atomic density
   stdcomp        - basic + atomic density + temperature
10 wtptcomp       - basic (configurable)
   wtptcomp       - basic
11 wtptcomp       - basic + volume fraction
   wtptcomp       - basic + volume fraction + temperature
12 wtptcomp       - basic + volume fraction + temperature + isotopics
   atomcomp       - basic (configurable)
   atomcomp       - basic
13 atomcomp       - basic + volume fraction
   atomcomp       - basic + volume fraction + temperature
14 atomcomp       - basic + volume fraction + temperature + isotopics
   solution       - rho      + density + temperature + volume fraction
15 solution       - molar    + density + temperature + volume fraction
   solution       - massfrac + density + temperature + volume fraction
   solution       - molefrac + density + temperature + volume fraction
16 solution       - molality + density + temperature + volume fraction
```

Figure 160. User defined solution composition final input.

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

Figure 161. Input specified by placeholder insertion. This will normally be placed on one line by Fulcrum but is moved to two lines for readability in this figure.

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 162.

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

Figure 162. Final solution input.

6.3 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.

7. UNIT CELLS AND BOUNDARY CONDITIONS

The examples included thus far have been limited to relatively large homogeneous systems in which the default infinite homogeneous medium unit cell type was appropriate. This section demonstrates more heterogeneous systems and shows the user how to use the other unit cell types appropriately. 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.

7.1 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 the appropriate unit cell type 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 V.a global unit.

7.2 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 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 that can be used to define the neutron spectrum characteristics for part of the problem.

7.3 UNIT CELL TYPES

XSPROC offers four types of unit cell calculations: INFHOMMEDIUM, LATTICECELL, MULTIREGION, and DOUBLEHET. Note that you may define multiple unit cells in a single CSAS/KENO V.a 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.

7.3.1 Infinite Homogeneous Medium

The INFHOMMEDIUM treatment is best suited for large masses of materials if the size of each material is large compared with the average mean-free path of neutrons in the material or if the fraction of the material within one mean-free path from the surface of the material is very small. Every material specified in the problem that is not specified in a unit cell is treated as an infinite homogeneous lump. Systems composed of small fuel lumps should not be treated as an infinite homogeneous medium.

7.3.2 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 spherical pellets. The approximation becomes less rigorous for short fuel pins and/or small arrays in which multidimensional spatial dependence or leakage can become important.

7.3.3   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 boundary condition on the outer boundary. 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 of equal volume to that of the cuboidal outer region and specifying a white outer boundary condition.

3.   The boundary conditions available in a MULTIREGION problem include vacuum (eliminated 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 a real physical situation, 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 collapse 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: To apply boundary conditions to a unit cell model, you must use the MULTIREGION cell treatment. These boundary conditions are completely unrelated to the boundary condition of the outer surface of the KENO V.a model.

7.3.4   Double-Heterogeneous

DOUBLEHET cells use a specialized CENTRM/PMC 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 the 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 can 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, whereas 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 requiring 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, then 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, then all types are homogenized into a single mixture for the compact. As for the one grain type case, 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 slabs, fuel height must also be specified. The slab width must also be specified for slab cells.

7. The CENTRM calculation option must be Sn.

7.3.5 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); the default for the right or outer boundary is Vacuum.

7.3.5.1 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 163. (NOTE: In the following figures, a dashed arrow indicates neutrons leaving the system. A solid arrow represents those 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.)

Vacuum boundary condition on left surface

Figure 163. Vacuum boundary condition.

7.3.5.2 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 164, 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.

Reflective boundary condition on left surface

Figure 164. Reflective boundary condition.

7.3.5.3   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 165 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.

Periodic boundary conditions on left and right surfaces
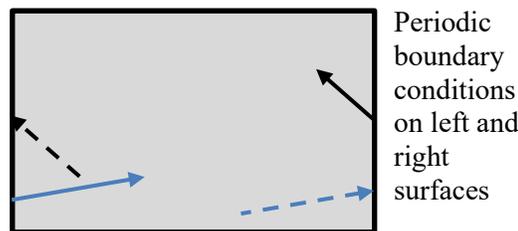
Figure 165. Periodic boundary conditions.

7.3.5.4   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 166) 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.

White boundary condition on left surface
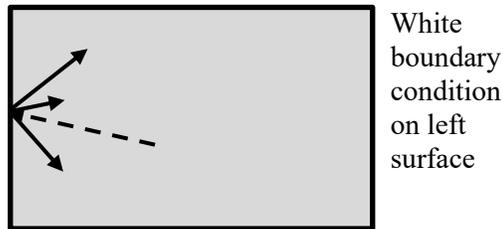
Figure 166. White boundary condition.

7.4 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).

7.4.1 Infinite Homogeneous Medium

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

7.4.2 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.

7.4.3 Multiregion

The unit cell is more flexible for the MULTIREGION treatment than that of the LATTICECELL treatment. In fact, the MULTIREGION treatment allows complete freedom in the placement of materials but is constrained by shape (i.e., concentric regions of the same shape are required). MULTIREGION does not account for lattice effects, so it is best used for problems where 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.

7.4.4 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 that are 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 spherical, cylindrical, and slab geometries.

7.4.5 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 V.a 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.

7.5 PROBLEM DESCRIPTIONS

Three example problems will be 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.

7.5.1 LATTICECELL 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 fuel rod and guide tube pitch is 1.64 cm. The fuel assembly radial layout is provided in Figure 167.

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 having the same OD 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 168.
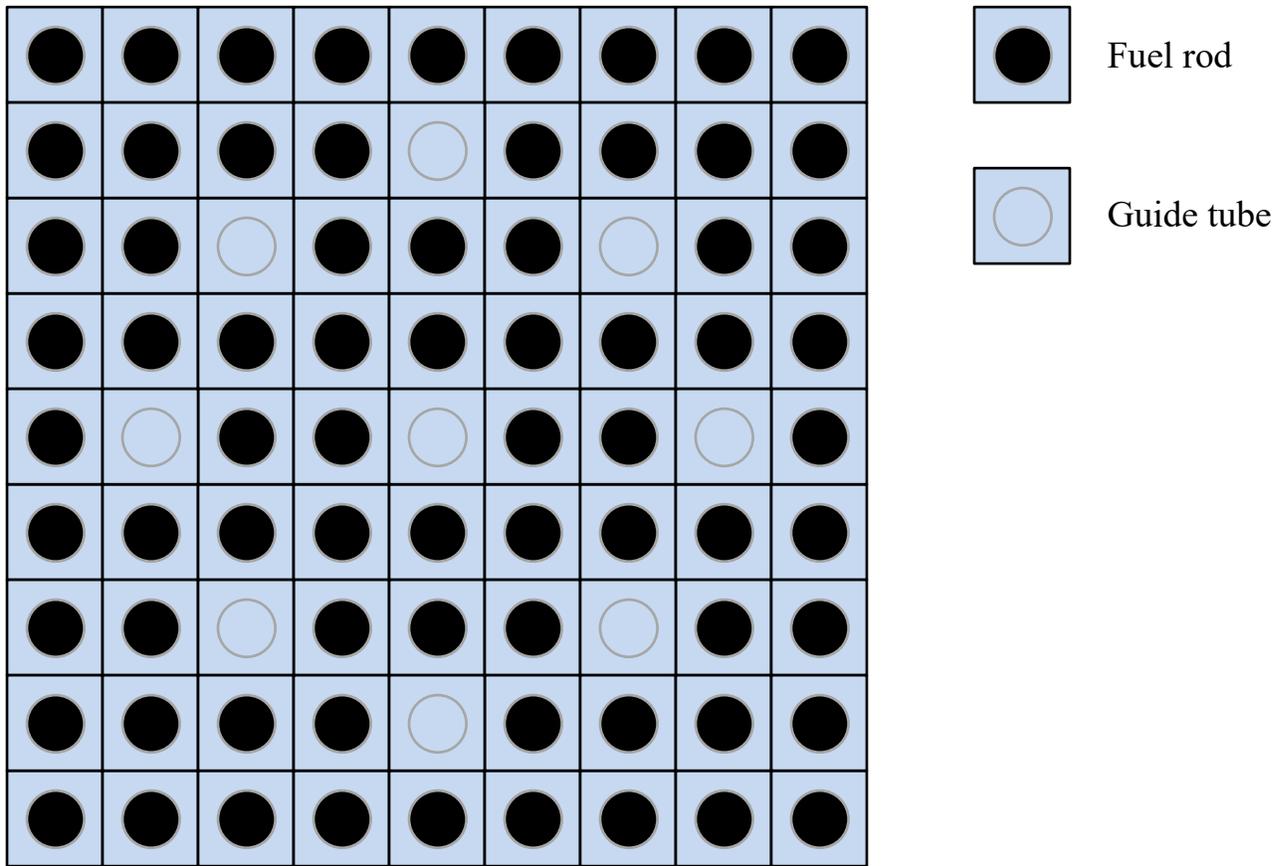
Figure 167. Fuel assembly radial layout.

10.5 cm

0.545 cm

7.0 cm

0.4875 cm

■ UO$_2$

■ Zircaloy-2

365.76 cm

0.47 cm

4.0 cm

NOT TO SCALE
All dimensions in cm
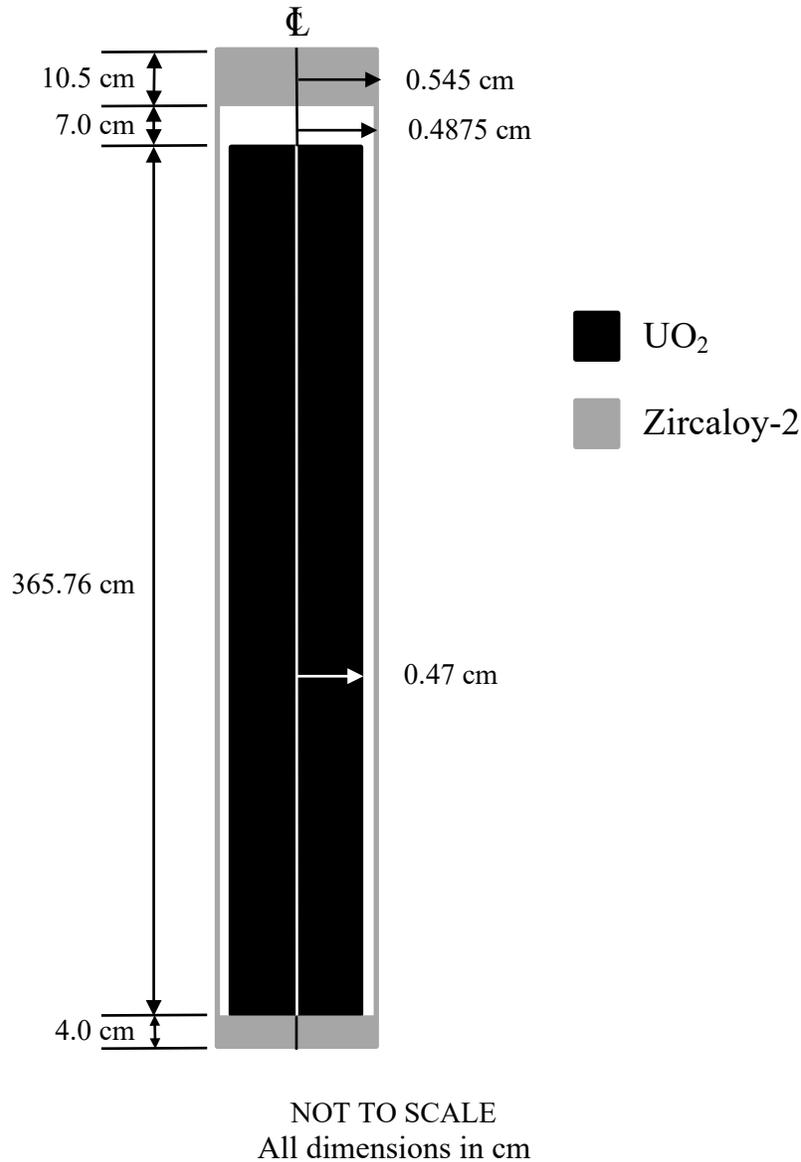
Figure 168. Axial description of the fuel rods.

7.5.1.1 General and Material 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 in which to save the model, and name it `fuel_assembly.inp`. Next, use the autocomplete feature in Fulcrum to create a **csas5 – Criticality safety analysis using KENO V.a** 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 signals to CSAS5 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, discussed in Section 7.5.1.2. The first three lines of the input are shown in Figure 169.

119

Figure 169. General information for fuel assembly problem.

The different mixtures needed for the model must be defined next. These mixtures are UO₂, Zircaloy-2, and water. All three materials are available in the SCALE standard composition library, which will make the definitions much simpler and faster to provide.

First, define the UO₂ composition. Move the cursor into the **read comp** block and autocomplete a **stdcomp – basic (configurable)**. This will generate a fillable form, as shown in Figure 170. In the **composition** dropdown, select uo2 by opening the dropdown 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 ²³⁵U with its nuclide ID of **92235** and set the **Weight Percent** to 3.5. To add ²³⁸U, 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 171. 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.

Figure 170. Initial STDCOMP configurable form.
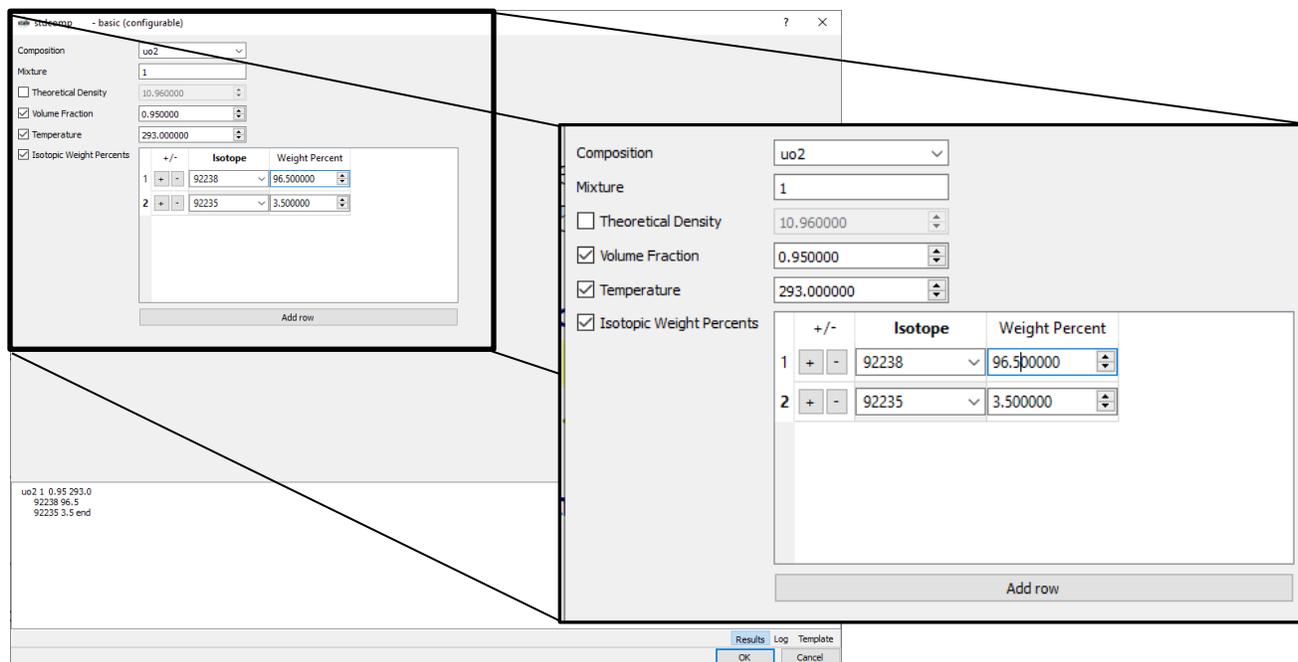
Figure 171. Complete UO₂ configurable form showing a magnified section of the user-specified input.

Next, repeat this process to generate the Zircaloy-2 composition. Autocomplete a **stdcomp – basic (configurable)** and select `zirc2` from the **composition** dropdown. 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 172.
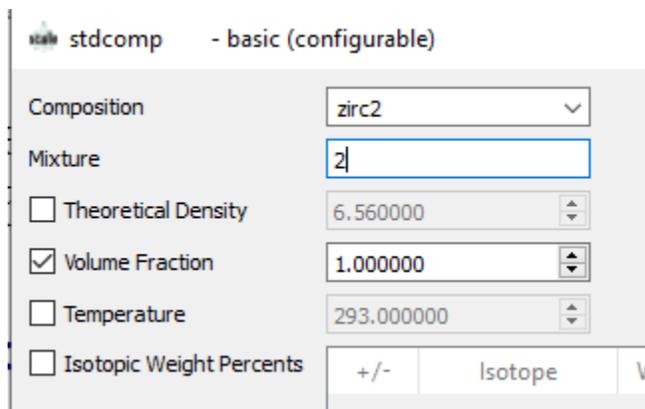
Figure 172. User-specified input portion of the complete zirc2 configurable form.

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, it is likely faster to specify the composition by entering the text directly in Fulcrum than to use 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/cm³) 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 173. 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 will be discussed in Section 7.5.1.2, only a single unit cell is needed for this problem, and correspondingly, no copies of mixtures are needed.

```
 5 read comp
 6    uo2 1   0.95 293.0
 7            92238 96.5
 8            92235 3.5 end
 9
10  zirc2 2   1.0   end
11
12  h2o   3   end
13 end comp
```

Figure 173. Complete COMPOSITION block for the fuel assembly problem.

7.5.1.2   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. 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 174. 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 175. Fulcrum inserts a single line of CELLDATA input, as shown in Figure 176, which must be modified for use in this model.

```
13 end comp
14
15 cells
   parameters
   biasing
16 start       geometry
   arrays
17 xlds
   mixt
18 volume     l unit 1
   energy
   grid
19 bounds
   plots
20 reactions  TODO: define global unit here
21
22
```

Figure 174. Autocomplete list of additional blocks available for inclusion in the model.
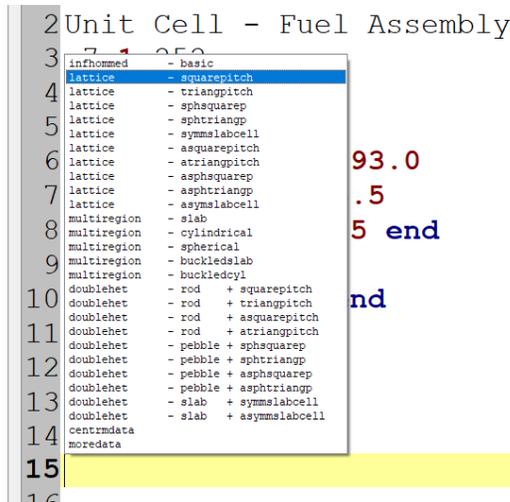
Figure 175. Autocomplete list of cells within the CELLDATA block.

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

Figure 176. 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 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 OD). 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 also completes the CELLDATA block. The complete entry/block is provided in Figure 177; 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 177. Complete LATTICECELL input.

7.5.1.3 KENO V.a Geometry Data

In this problem, 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.

123

The fuel rod unit cell, Unit 1, is the first unit to be defined. Recall that geometry in KENO V.a must be added from the inside out, with each region completely encompassing the previous region. The outer boundary of the unit must also be a cuboid so that Unit 1 can be included in the array. The regions that will be entered are therefore the fuel cylinder, the gap cylinder, the cladding cylinder, and finally, the water cuboid. 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 V.a. To add the cylinder, position the cursor between the unit declaration and the **TODO** comment. Autocomplete in this context provides a list of shapes and their configurable forms, as shown in Figure 178. Select **zcylinder – kenova (configurable)** as shown in Figure 179 to open a configurable form, as shown in Figure 180.
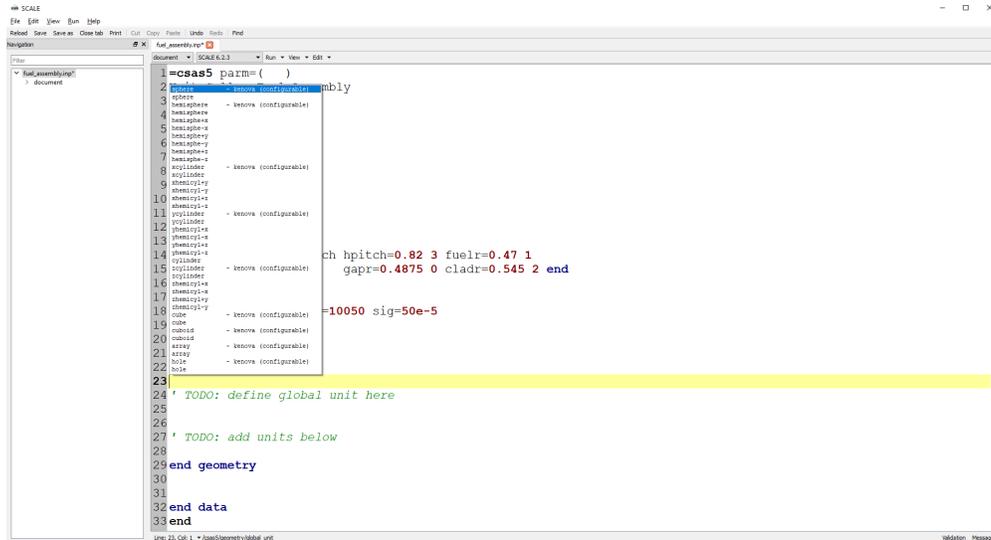
Figure 178. Autocomplete options in a unit.
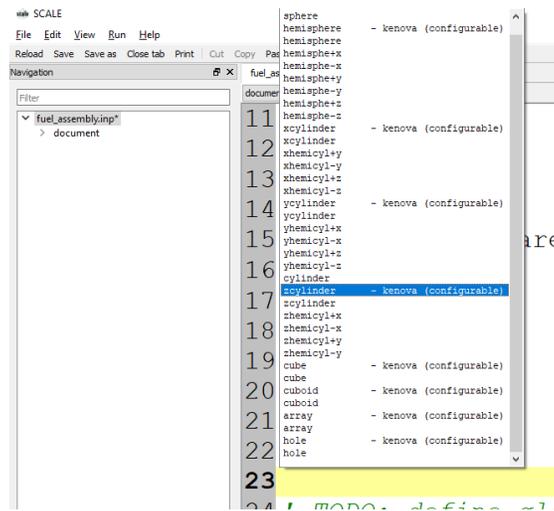
Figure 179. Selecting z-cylinder configurable.
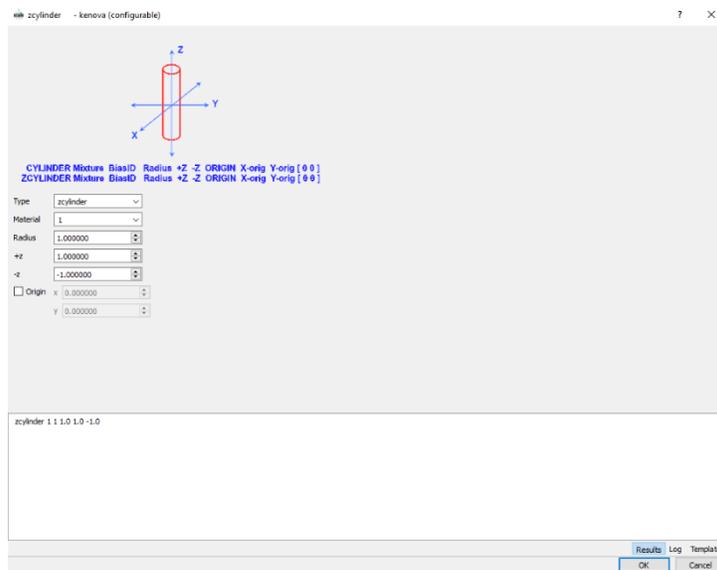
Figure 180. Initial z-cylinder configurable form.

Completing the configurable form will provide the input for a single cylindrical region in Unit 1. The first dropdown is available because a **cylinder** in SCALE is, by default, aligned parallel to the Z axis. The input for the second dropdown is the mixture with which to fill the region; in this case mixture 1 is correct since it is the fuel mixture. 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 181. Press **OK** to add the cylinder region to the input after checking that inputs are correct.
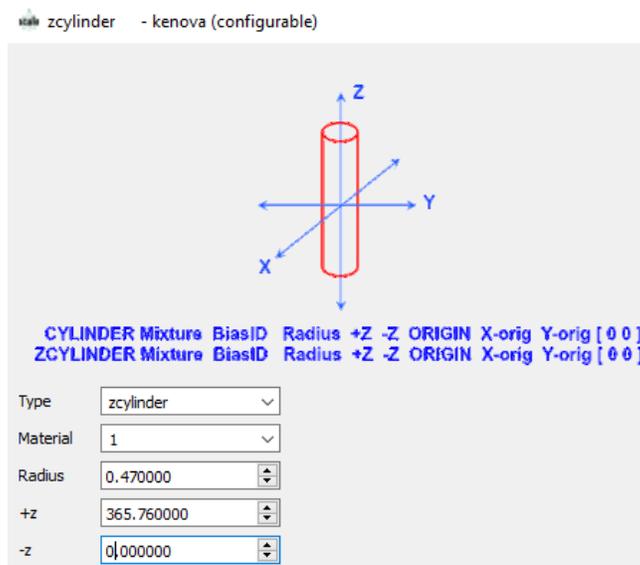
Figure 181. Completed user-specified input portion of the z-cylinder configurable form.

The next region in unit 1 is the void inside the cladding and outside the fissile column. The easiest way to do this is to copy and paste the existing entry and change the mixture and dimensions. This second cylinder is filled with void (mixture 0) and has a radius of 0.4875 cm, a +Z surface of 372.76 cm, and a -Z surface of 0 cm. This second region contains the first region with the larger radius and high +Z surface. The -Z surface is 0 cm in both cases, so no portion of the first region extends through the second region.

The cladding cylinder can also be added by copying the previous zcylinder and pasting it into the unit. The mixture number for the Zircaloy-2 cladding is 2, and the radius is 0.545 cm. The +Z surface for the void zcylinder is 372.76, and the end plug length is 10.5 cm. Enter 372.76+10.5, highlight the expression as shown in Figure 182, and evaluate the expression by selecting the **Edit** dropdown and clicking **Evaluate.** Enter the **-Z** surface as -4.0 cm to represent the lower end plug. The complete three cylinders are shown in Figure 183. The third region completely contains the second region; the +Z is higher, the -Z is lower, and the radius is larger. The KENO V.a modeling requirement is satisfied.

```
22 unit 1
23    zcylinder 1 1 0.47 365.76 0.0
24    zcylinder 0 1 0.4875 372.76 0.0
25    zcylinder 2 1 0.545 372.76+10.5 -4.0
26 ' TODO: define global unit here
```

Figure 182. Highlighting expression for evaluation in Fulcrum.

```
22 unit 1
23    zcylinder 1 1 0.47 365.76 0.0
24    zcylinder 0 1 0.4875 372.76 0.0
25    zcylinder 2 1 0.545 383.26 -4.0
26 ' TODO: define global unit here
```

Figure 183. 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 so 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, that water will be added in the global unit. The cuboid can be entered directly in the Fulcrum editor as cuboid 3 1 0.82 -0.82 0.82 -0.82 383.26 -4.0. Once again, this region contains the previous region in X and Y extents and is the same in the Z dimensions, so the KENO V.a geometry requirement is met. The final input for Unit 1 is shown in Figure 184.

```
22 unit 1
23    zcylinder 1 1 0.47 365.76 0.0
24    zcylinder 0 1 0.4875 372.76 0.0
25    zcylinder 2 1 0.545 383.26 -4.0
26    cuboid 3 1   0.82 -0.82   0.82 -0.82   383.26 -4.0
27 ' TODO: define global unit here
```

Figure 184. Complete input for Unit 1.

126

The second unit needed for this problem is the guide tube unit. Again, the unit must be constructed from the inside out, with each subsequent region containing all previous regions. 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: the water inside the guide tube, and the guide tube itself. The outer cuboid will be the third and final region in Unit 2.

A new unit can be inserted after Unit 1 using autocomplete. A configurable option is available to build an entire unit at once. Selecting the **unit - kenova (configurable)** from the autocomplete list provides a unit configurable form like the one shown in Figure 185. Note that it may be necessary to resize the components within the form. Click on **Keno Va Unit** in the **Properties** pane and change the unit **Id** to 2. This is shown in Figure 186.
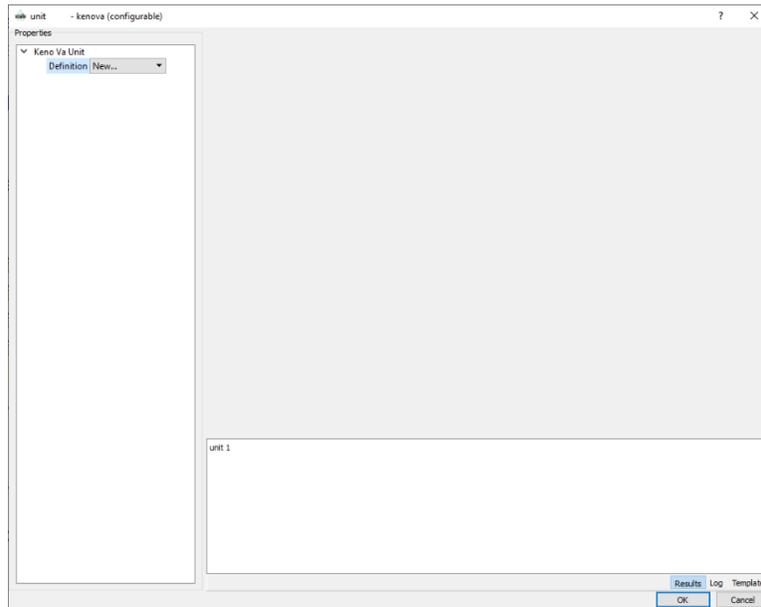
Figure 185. Default unit configurable form.
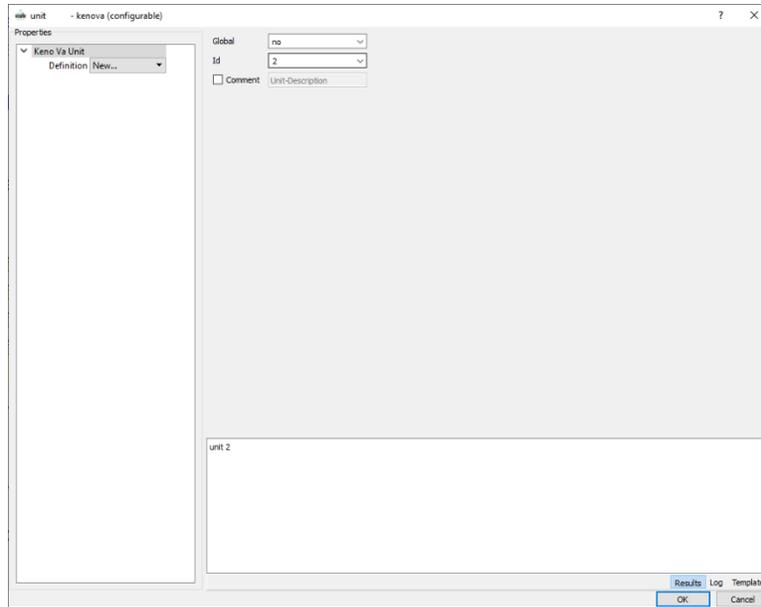
Figure 186. Unit configurable form with Id changed to 2.

The first entry is a cylinder, so select **Zcylinder** from the **Definition** dropdown. The configurable now shows a form much like the zcylinder configurable used in the previous example. In this case, the material is water, which is mixture 3. The mixture number must be entered manually in the **Material** field. The radius for this region is the inner radius of the guide tube, 0.5 cm. The overall length of the guide tube is 387.26 cm, so the +Z face can be set to 387.26 and the -Z face to 0. The configurable form with the complete first cylinder should look like that shown in Figure 187.
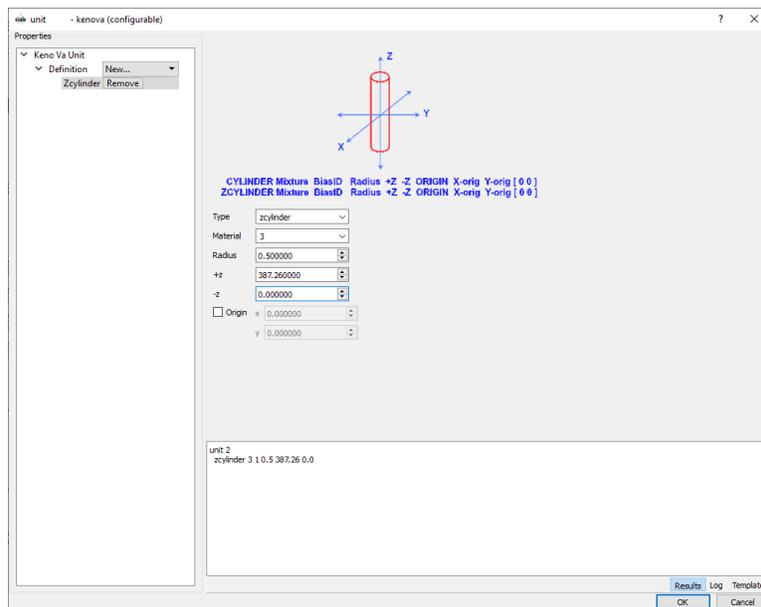
Figure 187. Unit configurable form with the complete first cylinder.

The second cylinder can be added by selecting **Zcylinder** again from the **Definition** dropdown. This is the guide tube itself, so the **Material** is mixture 2: Zircaloy-2. The radius is the outer radius of the guide tube, 0.575 cm, and the Z surfaces are the same as the previous cylinder, 387.26 and 0. The second

region has the same height but a larger radius than the first region, so the subsequent region does contain the first one. The user-specified input for the second cylinder is shown in Figure 188.

Figure 188. User-specified input for guide tube cylinder.

The third and final region in this unit is the cuboid containing the water outside the guide tube. Select **Cuboid** as the new region from the **Definition** dropdown. Water is mixture 3 in this model and should be entered in the **Material** field. As with Unit 1, the X and Y dimensions are set by the assembly pitch, so values of ± 0.82 cm from the X and Y planes are appropriate. The Z dimensions are the same as those in the previous regions. The user-specified input for the cuboid is shown in Figure 189. After confirming that all the inputs are correct in the configurable window, click **OK** to add the unit to the input. The complete unit 2 input is shown in the input file in Figure 190.

Figure 189. User-specified input for the cuboid as the outer region of Unit 2.

```
27 unit 2
28   zcylinder 3 1 0.5 387.26 0.0
29   zcylinder 2 1 0.575 387.26 0.0
30   cuboid 3 1 0.82 -0.82 0.82 -0.82 387.26 0.0
```

Figure 190. Unit 2 input as entered into the input file by the unit configurable form.

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 that are placed in an array in KENO V.a 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 1.62 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. The input with the ARRAYS block skeleton is shown in Figure 192.

Figure 191. Autocomplete options to add the array block to the input.

Figure 192. Empty ARRAY block added by Fulcrum autocomplete.

The only option for autocomplete within the array block is for an **array - kenova basic (configurable)**; select this option to launch an array configurable form like that shown in Figure 193. 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 **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 to populate the ARRAY FILL with 1 which in this case 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 194. After reviewing the input to confirm that it is all correct, press **OK** to add the array to the input. The input in the ARRAY block is shown in Figure 195. 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 including the guide tubes (Unit 2), are shown in Figure 196. Note that the **TODO** comment and extra blank lines have been removed to save space in this document but have no impact on KENO if they remain in the input.
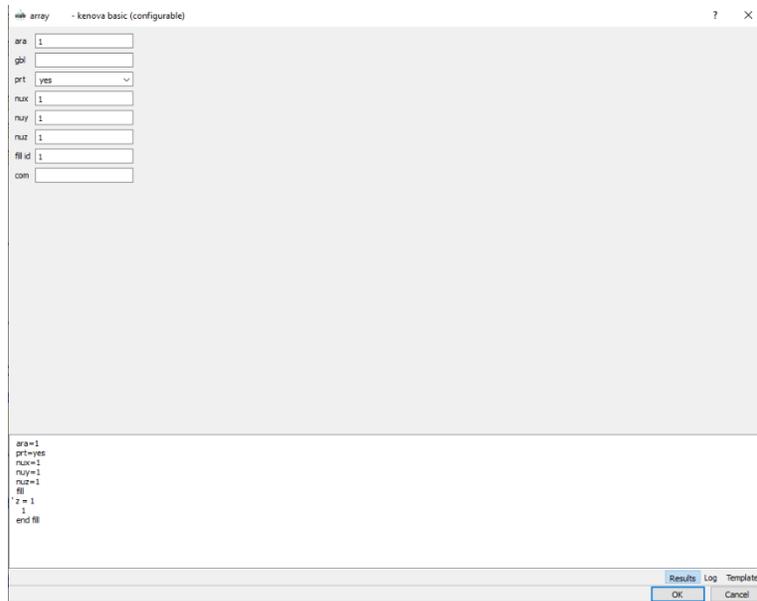
Figure 193. Default ARRAY configurable form.
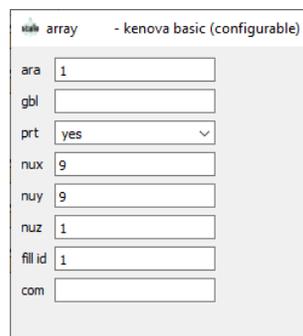
Figure 194. User-specified portion of the ARRAY configurable form.

```
40   ara=1
41   prt=yes
42   nux=9
43   nuy=9
44   nuz=1
45   fill
46 ' z = 1
47      1 1 1 1 1 1 1 1 1 1
48      1 1 1 1 1 1 1 1 1 1
49      1 1 1 1 1 1 1 1 1 1
50      1 1 1 1 1 1 1 1 1 1
51      1 1 1 1 1 1 1 1 1 1
52      1 1 1 1 1 1 1 1 1 1
53      1 1 1 1 1 1 1 1 1 1
54      1 1 1 1 1 1 1 1 1 1
55      1 1 1 1 1 1 1 1 1 1
56   end fill
```

Figure 195. Array input entered in the KENO input by the Fulcrum configurable form.

131

```
39 read array
40   ara=1
41   prt=yes
42   nux=9
43   nuy=9
44   nuz=1
45   fill
46 ' z = 1
47     1 1 1 1 1 1 1 1 1
48     1 1 1 1 2 1 1 1 1
49     1 1 2 1 1 1 2 1 1
50     1 1 1 1 1 1 1 1 1
51     1 2 1 1 2 1 1 2 1
52     1 1 1 1 1 1 1 1 1
53     1 1 2 1 1 1 2 1 1
54     1 1 1 1 2 1 1 1 1
55     1 1 1 1 1 1 1 1 1
56   end fill
57 end array
```

Figure 196. Final ARRAY block for this problem.

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 its constituent units, so a new unit is needed. This will be unit 3 and will be the global unit.

The array must be the first entry in the unit so that the subsequent regions can fully encompass it. The only subsequent region in this unit will be the water above, below, and around the fuel assembly. The first step is to create global unit 3. This can be done by typing `global unit 3` in the GEOMETRY block or by using autocomplete and selecting **global unit**. Autocomplete provides global unit 1, so the unit number must be changed manually to 3. As long as a comment exists between the global unit declaration and the end of the geometry block, the Fulcrum autocomplete can be used to generate an array configurable form as shown in Figure 197. The default array configurable form is shown in Figure 198.

Figure 197. Selecting the array configurable form in Fulcrum.
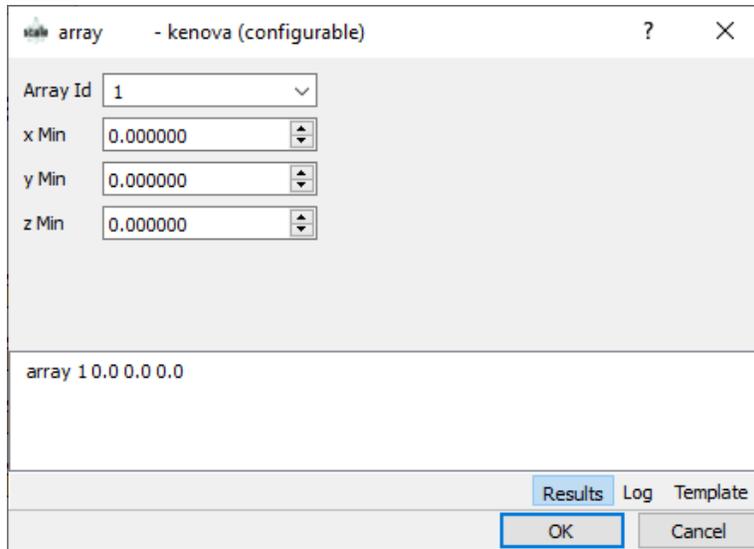
Figure 198. Default array configurable form resized to show detail.

As discussed in Section 4.6, arrays are positioned in KENO V.a by specifying the location of the most negative point, that is the (x-min, y-min, z-min) point in the new unit coordinate system. The array can be positioned anywhere, and any other regions to be added in the unit can be adjusted to that basis. For fuel assembly models, it is often convenient to position the center of the array at the origin. To do that, the most negative point must be positioned at -4.5 times the array pitch in X and Y and at the negative half height of the array. This point is (-7.38, -7.38, -193.63) and should be entered in the array configurable. The **Array Id** is 1, as there is only one array in the model, and it was given the **Id** 1. The completed array configurable form is shown in Figure 199. After checking the inputs, press the **OK** button to enter the array placement in the input file; the resulting input is shown in Figure 200.
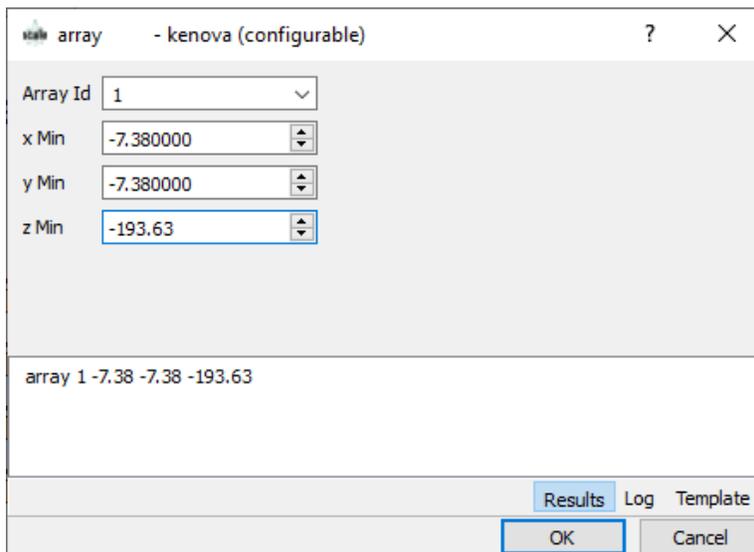
Figure 199. Completed array configurable form, resized to show detail.

```
31 global unit 3
32    array 1 -7.38 -7.38 -193.63
```

Figure 200. Array input in global unit.

The final region to be added is the water around the assembly. The simplest approach is to use the REPLICATE command, as the thickness of the water region is provided; an ARRAY is treated as a cuboid by KENO V.a, so the REPLICATE will create a cuboid. The thicknesses are added to the REPLICATE in the order the dimensions would be provided in specifying a cuboid. The order is therefore the thickness on the +X side, the thickness on -X side, the thickness on the +Y side, and so on. The four X and Y thicknesses are 7.62, and the two Z thickness are 15.24. Recall that water is mixture number 3 in this model. Lastly, remember that a final entry is needed to specify the number of regions to replicate; in this case only one region will be added. To add the replicate, type REPLICATE   3 1 7.62 7.62 7.62 7.62 15.24 15.24 1 after the array entry in the input. This is shown in Figure 201; as before, trailing comments and blank lines have been removed for clarity but have no impact on KENO. Note that the entries are all positive because they are thicknesses and not dimensions. This completes the geometry specification.

```
31 global unit 3
32   array 1 -7.38 -7.38 -193.63
33   replicate 3 1  7.62 7.62 7.62 7.62 15.24 15.24 1
34 end geometry
```

Figure 201. Final global unit, including REPLICATE to add water around fuel assembly.

Fulcrum visualization can now be used to check the model; a more complete description of the Fulcrum's visualization capabilities is provided in Section 8.2.1. 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.
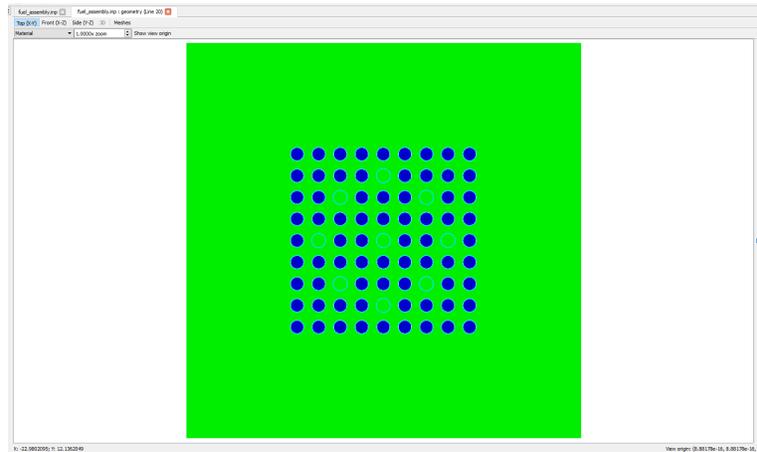
Figure 202. Radial slice visualization of the completed fuel assembly model midplane.

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 PARAMTER block. These parameters yield a k_{eff} estimate of 0.6914 ± 0.0019; 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 325 total generations, skipping the first 50, resulted in a k_{eff} estimate of 0.69061 ± 0.00048. This calculation repeated with CE cross sections yields a k_{eff} of 0.69160 ± 0.00049; the difference from the MG calculation is 0.00099 ± 0.00069. This difference is only ~1.4 standard deviations and thus likely statistically insignificant. The LATTICECELL processing used in this problem provides sufficiently accurate cross sections to reproduce the CE solution.

7.5.2   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.
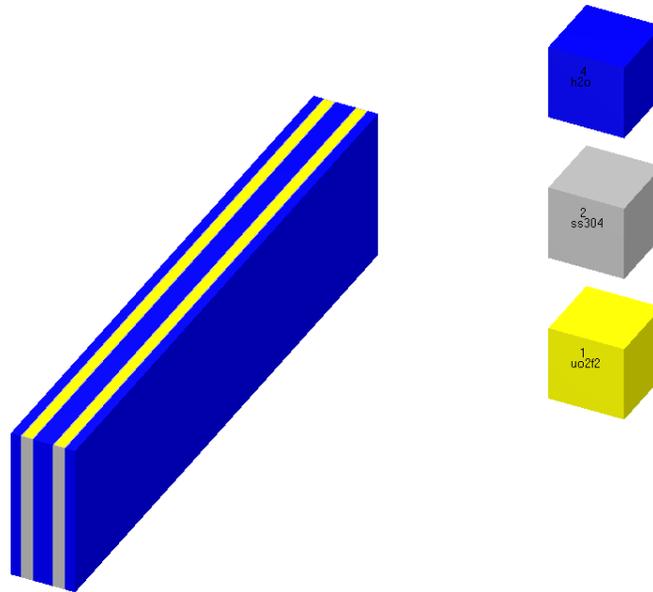
Figure 203. 3D rendering of the two tanks with the top half of the model removed to show the solution.

7.5.2.1   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 in which to save the model, and name it `slab_tanks.inp`. Next, use the autocomplete feature in Fulcrum to create a **csas5 – Criticality safety analysis using KENO V.a** 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 it signals to CSAS5 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 7.5.2.2. The first three lines of the input are shown in Figure 204.

```
1=csas5 parm=(   )
2 Unit Cell - Slab Tanks
3 v7.1-252
```

Figure 204. First three lines of input for slab tank problem.

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

135

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.

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

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

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

Figure 206. Fissile solution input for this problem.

The SS304 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.

Figure 207. Initial STDCOMP configurable form.

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

Figure 208. Final composition block for the model.

136

7.5.2.2 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 nearly 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. This input must be modified for use in this model.

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

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

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

137

(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 explicitly specifies 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 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 in columns as shown in Figure 212. The column arrangement can be more convenient and make it easier to keep track of which mixtures have been used, and it can also help 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 be used in only one region of one cell in the CELLDATA block. These duplicate mixtures will not be used in the KENO V.a 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 would need to be used in the respective regions in the KENO V.a geometry.

```
12 read celldata
13 multiregion slab right_bdy=vacuum left_bdy=reflected end
14     5 5.0
15     3 5.5
16     1 10.5
17     2 11.0
18     4 16.0   end zone
19 end celldata
```

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

7.5.2.3   KENO V.a 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. Like all KENO V.a models, the geometry must be specified in regions from the inside out, with all subsequent regions containing all previous ones. 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 with the cursor in the global unit above the **TODO** comment. Figure 213 shows the menu for selecting the cuboid configurable form, and Figure 214 shows the default form as it is initially generated by 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 215. After checking the input to confirm it is correct, press **OK** to enter the cuboid entry into the input file. Figure 216 shows the cuboid entry in the input file.
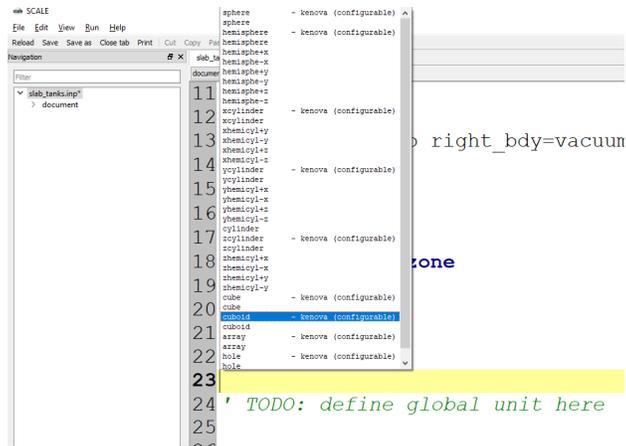
Figure 213. Selecting the cuboid configurable form from the Fulcrum autocomplete menu.
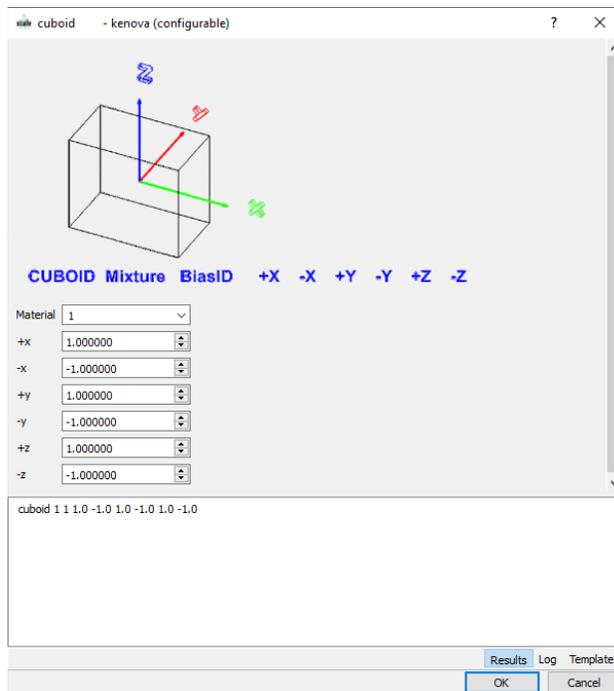
Figure 214. Default cuboid configurable form in Fulcrum.
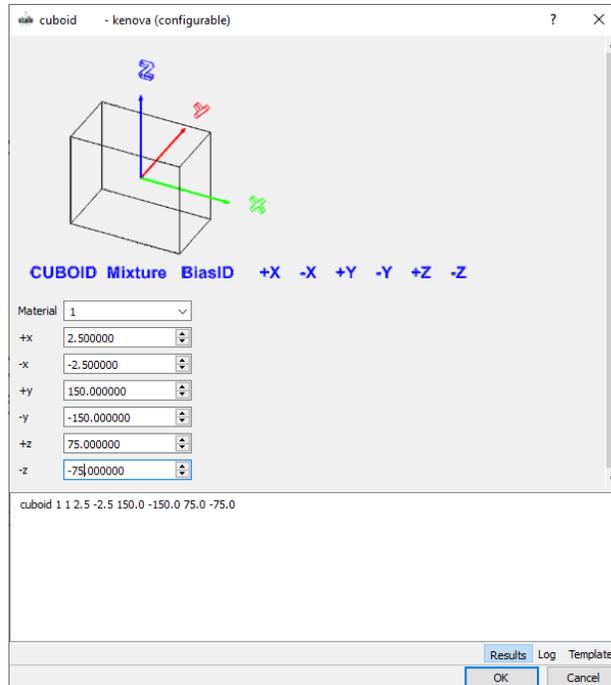
Figure 215. Cuboid configurable form in Fulcrum with the dimensions added for the fissile solution tank.

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

Figure 216. Cuboid input for the fissile solution inside the tank.

The next region is the tank wall, and given that the thickness is specified, the REPLICATE option is the easiest approach. The tank wall is 0.5 cm thick in all 6 directions, and the stainless steel is mixture 2. The input is `replicate 2 1  0.5 0.5 0.5 0.5 0.5 0.5  1`. Recall that the last 1 is the number of times this replicate should be performed. The final region is the water (mixture 4) outside the tank on the +X and -X faces. These water regions are 5 cm on each side of the tank, so the input is `replicate 4 1  5.0 5.0 0 0 0 0  1`. The final GEOMETRY block is shown in Figure 217; comments and blank lines have been removed for clarity in the figure.

```
20 read geometry
21 global unit 1
22    cuboid 1 1 2.5 -2.5 150.0 -150.0 75.0 -75.0
23    replicate 2 1  0.5 0.5 0.5 0.5 0.5 0.5  1
24    replicate 4 1  5.0 5.0 0 0 0 0  1
25 end geometry
```

Figure 217. Final geometry block for the two slab tank model.

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 that 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; only cuboidal outer boundaries should be used in KENO V.a for non-vacuum boundary conditions. The BOUNDS block can be added by autocomplete if the cursor is placed outside of any other blocks, as shown in Figure 218. 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 V.a 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 instance, the face code for the positive X face is +XB= and for the negative X face is -XB=. A mirror boundary condition can be specified with the keywords MIRROR or REFLECT, or a shortened form of either. For this model, the input -XB=REFL should be added in the BOUNDS block as shown in Figure 219.

Figure 218. Fulcrum autocomplete to add the BOUNDS block to the input.

Figure 219. BOUNDS block for this model, making the negative X-face reflective.

The model is complete, and a Fulcrum 2D view of the entire model is shown in Figure 220. A magnified view of one end of the model is shown in Figure 221; the color scheme in these figures is the same as that used in Figure 221: 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.
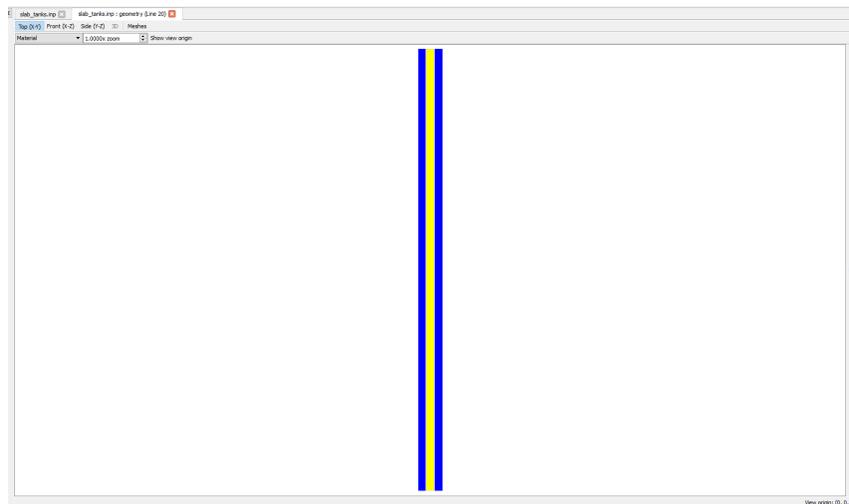
Figure 220. Fulcrum 2D visualization of the axial midplane of the model.

141
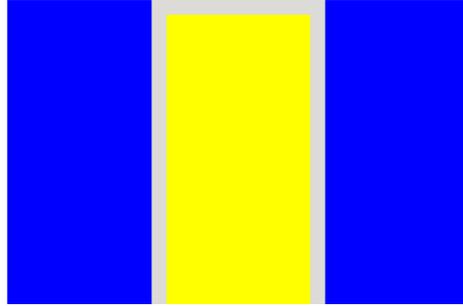
Figure 221. Detailed view of the +Y end of the axial midplane of the model showing the end of the tank.

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 PARAMTER block. These parameters yield a k_{eff} estimate of 0.9460 ± 0.0020; 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 0.9438 ± 0.0020. The difference between these results is 0.0022 ± 0.0028; 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 that simulated 10,000 particles per generation for 410 total generations, skipping the first 50, resulted in a k_{eff} estimate of 0.94920 ± 0.00049. This calculation repeated with CE cross sections yields a k_{eff} of 0.94830 ± 0.00045; the difference from the MG calculation is 0.00090 ± 0.00067. This difference is only ~1.3 standard deviations and thus likely statistically insignificant. The MULTIREGION processing used in this problem provides sufficiently accurate cross sections to reproduce the CE solution.

7.5.3 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 4. 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. A schematic of the grain and pebble are shown in Figure 222. 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 4. 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
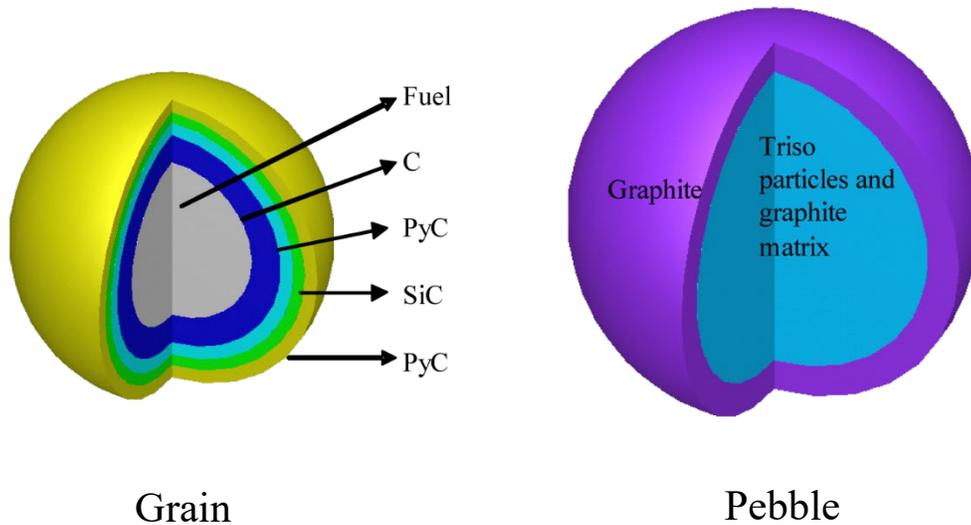
Grain

Pebble

Figure 222. Schematic of individual grain (left) and pebble (right).

7.5.3.1 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 in which to save the model, and name it `pebble.inp`. Next, use the autocomplete feature in Fulcrum to create a **csas5 – Criticality safety analysis using KENO V.a** 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 CSAS5 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, discussed in Section 7.5.3.2. The first three lines of the input are shown in Figure 223.

```
1 =csas5 parm=(    )
2 Unit Cell - Pebble
3 v7.1-252
```

Figure 223. First three lines from pebble problem.

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

143

92235 row or the **Add row** button, and then 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 224.

Figure 224. Configurable form for UO₂ 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 graphite thermal scattering law (TSL) data, as it 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/cm³ 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 225. 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/cm³. The updated composition block is shown in Figure 226.

Figure 225. Configurable form for the porous carbon layer composition.

144

```
 5 read comp
 6    uo2 1 den=10.400000 1.0 700.0
 7          92238 90.4
 8          92235 9.6 end
 9 graphite 2 den=1.050000 1.0 700.0 end
10 graphite 3 den=1.900000 1.0 700.0 end
```

Figure 226. Input with UO₂, porous carbon layer, and inner pyrolytic carbon layer compositions.

Mixture 4 will be the SiC layer; SiC is not a standard basic composition but 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 227. 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 228. Change the text in the **Name** field to `SiC` and set this as **Mixture** 4. Set the **Theoretical Density** to 3.18 g/cm³ and the **Temperature** to 700 K. Click the **Add** button next to **Element Atom Counts**, select **14000** from the **Element** dropdown 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 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 229; note that the carbon data is showing, as it is impossible to display both the Si and C pages at the same time.

Figure 227. Selecting ATOM mixture configurable form in Fulcrum.

Figure 228. ATOM mixture configurable form showing all necessary buttons and fields.
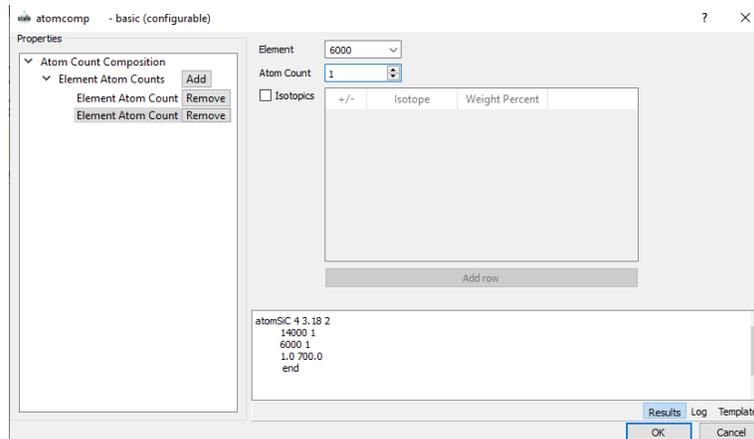
Figure 229. Complete ATOM composition configurable form for the SiC layer composition.

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 230 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 of these mixtures have a density of 2.1 g/cm³. The increased COMPOSITION block is shown in Figure 231. This completes the mixtures needed for the pebble definition.

```
 6     uo2 1 den=10.400000 1.0 700.0
 7           92238 90.4
 8           92235 9.6 end
 9 graphite 2 den=1.050000 1.0 700.0 end
10 graphite 3 den=1.900000 1.0 700.0 end
11 atomSiC 4 3.18 2
12           14000 1
13           6000 1
14           1.0 700.0
15              end
16 graphite 5 den=1.900000 1.0 700.0 end
```

Figure 230. Mixture definitions for all composition in the fuel grains.

```
 6     uo2 1 den=10.400000 1.0 700.0
 7           92238 90.4
 8           92235 9.6 end
 9 graphite 2 den=1.050000 1.0 700.0 end
10 graphite 3 den=1.900000 1.0 700.0 end
11 atomSiC 4 3.18 2
12           14000 1
13           6000 1
14           1.0 700.0
15              end
16 graphite 5 den=1.900000 1.0 700.0 end
17 graphite 6 den=2.100000 1.0 700.0 end
18 graphite 7 den=2.100000 1.0 700.0 end
```

Figure 231. 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/cm³. 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 232.

Figure 232. Complete configurable form for helium coolant mixture.

The addition of comments in the input to clarify which mixtures should be used for which regions 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 1 and pressing enter. Move back to this new blank line, place an apostrophe in column 1, and enter the comment: `Mixture 2 is porous graphite layer`. The updated input is shown in Figure 233. Repeat this process to add comments for Mixture 3 (`' Mixture 3 is inner pyrolytic carbon layer`), Mixture 4 (`' Mixture 4 is silicon carbon 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`).

```
 6    uo2 1 den=10.400000 1.0 700.0
 7          92238 90.4
 8          92235 9.6 end
 9 ' Mixture 2 is porous graphite layer
10 graphite 2 den=1.050000 1.0 700.0 end
```

Figure 233. Comment added to input.

This completes the mixture specifications in the COMPOSITION block for this model. The complete input block is shown in Figure 234. Blank lines and extraneous comments have been removed from the input for clarity in the figure.

```
 5 read comp
 6    uo2 1 den=10.400000 1.0 700.0
 7          92238 90.4
 8          92235 9.6 end
 9 ' Mixture 2 is porous graphite layer
10 graphite 2 den=1.050000 1.0 700.0 end
11 ' Mixture 3 is inner pyrolytic carbon layer
12 graphite 3 den=1.900000 1.0 700.0 end
13 ' Mixture 4 is SiC layer
14 atomSiC 4 3.18 2
15          14000 1
16          6000 1
17          1.0 700.0
18             end
19 ' Mixture 5 is outer pyrolytic carbon layer
20 graphite 5 den=1.900000 1.0 700.0 end
21 ' Mixture 6 is graphite matrix
22 graphite 6 den=2.100000 1.0 700.0 end
23 ' Mixture 7 is graphite pebble cladding
24 graphite 7 den=2.100000 1.0 700.0 end
25    he 8 den=0.006100 1.0 700.0 end
26 end comp
```

Figure 234. Entire COMPOSITION block for pebble model.

7.5.3.2   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 235. 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 236. As shown in Figure 237, Fulcrum inserts generic CELLDATA input, which must be modified for use in this model.

```
26 end comp
27
28   cells
     parameters
29   biasing           geometry
     start
30   arrays
     xids
     mixt
31   volume         1 unit 1
     energy
     grid
32   bounds
     plots
33   reactions       0: define global
```

Figure 235. Autocomplete list of available blocks for inclusion in the model.

148

Figure 236. Autocomplete list of cells within the CELLDATA block.

```
27 read celldata
28 doublehet right_bdy=white fuelmix=10 end
29   gfr=0.0     1
30   coatt=0.0   2
31   coatt=0.0   3
32   coatt=0.0   4
33   coatt=0.0   5
34   matrix=6 numpar=15000 vf=0 end grain
35   pebble sphsquarep right_bdy=white left_bdy=reflected
36   hpitch=0.0 7 fuelr=0.0 cladr=0.0 8 fuelh=0.0 end
37
38 ' TODO: define celldata
39
40 end celldata
```

Figure 237. Templated input provided for the DOUBLET PEBBLE + SPHSQUAREP cell in Fulcrum.

The first line declares the cell type as **DOUBLEHET**, as intended, and specifies the right boundary condition of the grain cell as **white**. This white boundary is the desired boundary condition, as 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 input 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 7.5.3. 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 4, 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 which matrix contains 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 238.

```
29   gfr=0.02397    1
30   coatt=0.0095   2
31   coatt=0.0040   3
32   coatt=0.0035   4
33   coatt=0.0040   5
34   matrix=6 numpar=15000 end grain
```

Figure 238. 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, as it specifies the height of a cylindrical or slab compact. The final DOUBLEHET unit cell specification is provided in Figure 239; as before, extraneous comments and blank lines have been removed for simplicity.

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

Figure 239. Complete DOUBLEHET unit cell specification for this model.

7.5.3.3   KENO V.a Geometry Data

The KENO V.a 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. Like all KENO V.a models, the regions must be specified from the inside out, and each subsequent region must contain the prior ones. 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 above the TODO comment and selecting **sphere - kenova (configurable)** as shown in Figure 240. The **Material** number in the form should be set to 10 because it is the mixture to which the homogenized cross sections have been assigned by XSProc. The radius of this region is 2.5 cm; the completed configurable form is shown in Figure 241. 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. This region could be added

150

with a REPLICATE, but since the outer radius is known, adding a SPHERE directly is simpler. Add this second region in the input by entering `sphere 7 1 3.0`. Mixture 7 is the correct graphite mixture to use because it was processed as the cladding material in the DOUBLEHET unit cell. The final region is a CUBOID filled with helium for the region between the pebbles. The mixture number for helium is 8, and the dimension in ±X, ±Y, and ±Z is 3.05 cm. Following the second SPHERE entry, add `cuboid 8 1 3.05 -3.05 3.05 -3.05 3.05 -3.05`. This completes the geometry input, as shown in Figure 242.

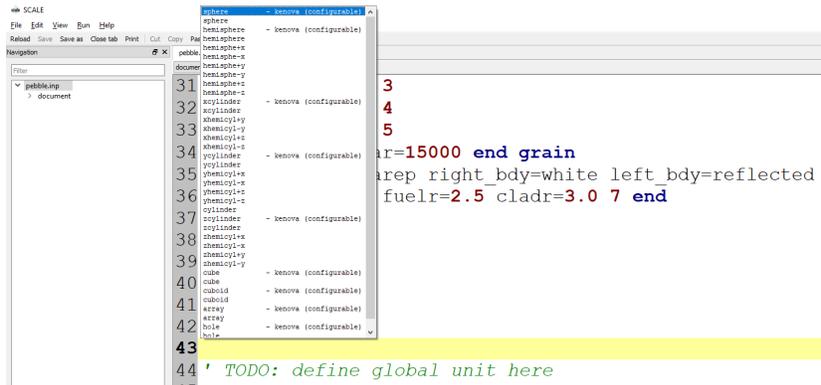**Figure 240. Selecting the sphere configurable form in Fulcrum.**
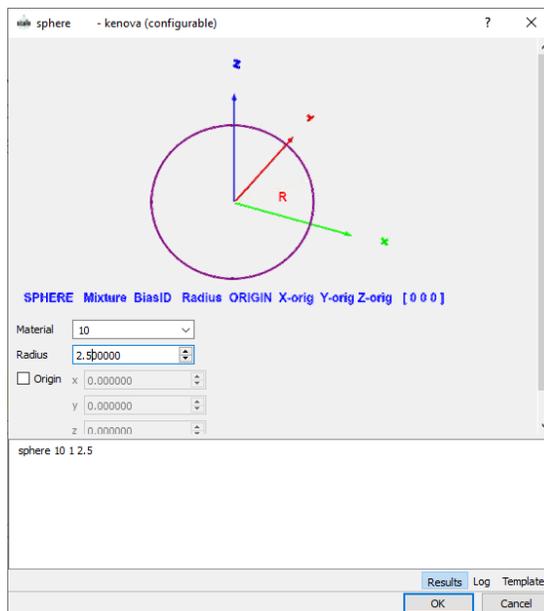
Figure 241. Complete configurable form for the fuel/matrix region of the pebble.

Figure 242. Complete geometry block for this 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

cursor outside the other blocks and use the Fulcrum autocomplete function to add a new block as shown in Figure 243. Autocomplete is not available in the BOUNDS block. To set all the faces to reflective boundaries, enter: `all=refl`. The completed BOUNDARY block is shown in Figure 244.

Figure 243. Adding BOUNDS block with Fulcrum autocomplete.

Figure 244. BOUNDS block making all boundaries reflective.

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 PARAMTER block. These parameters yield a k_{eff} estimate of 1.5396 ± 0.0014; 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 239 total generations, skipping the first 50, resulted in a k_{eff} estimate of 1.53800 ± 0.00049. Homogenized cross sections cannot be generated for CE calculations, so the more difficult explicit model must 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.

7.6 SUMMARY

This section discussed:

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

8. PLOTTING AND VISUALIZATION

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

8.1 WHAT YOU WILL BE ABLE TO DO

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

8.2 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 provides an introduction to these capabilities; a more complete description can be found in Lefebvre's "Advanced User Interface Capabilities." [7].

8.2.1 Geometry

Historically, CSAS and KENO V.a have been able to produce plots of 2D KENO V.a 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 V.a input. This option provides the user with the capability to check the geometry model prior to performing calculations. This capability and the associated input to the PLOT block is provided in the [Plot data] section of the KENO V.a manual.

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 has been used earlier in this primer, as described in Section 7.5.1.3, to display models as they were 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 245 and Figure 246 using the fuel assembly model from Section 7.5.1. This model will be 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 prevent visualization, as well. Check the **Messages** pane in Fulcrum if a global unit exists and the geometry plotter does not launch.

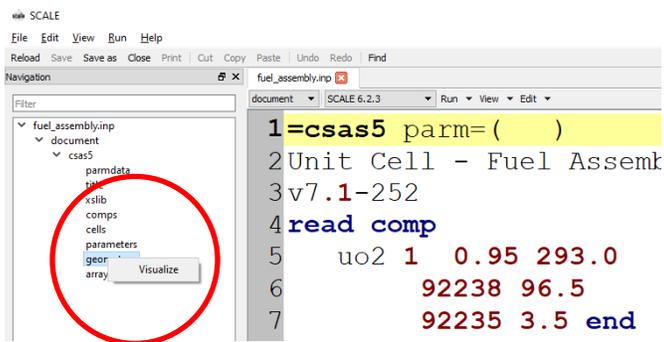**Figure 245. Location of View button in Fulcrum.**

Figure 246. Navigation panel and 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 247; 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 248.
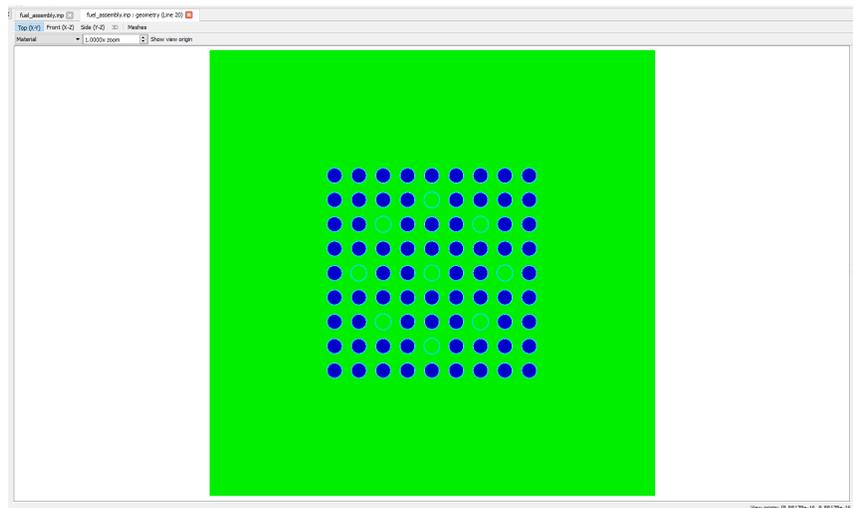

Figure 247. Default plot view of the fuel assembly array model in the Fulcrum plotter.
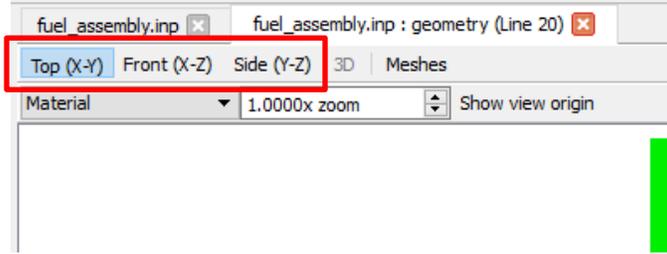
Figure 248. Selection of visualization plane orientation.

The bottom frame of the visualization window displays the current mouse pointer location on the bottom left and the origin of the current view on the bottom right, as shown in Figure 249. 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 250; 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 251), 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 252. 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 253, 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.

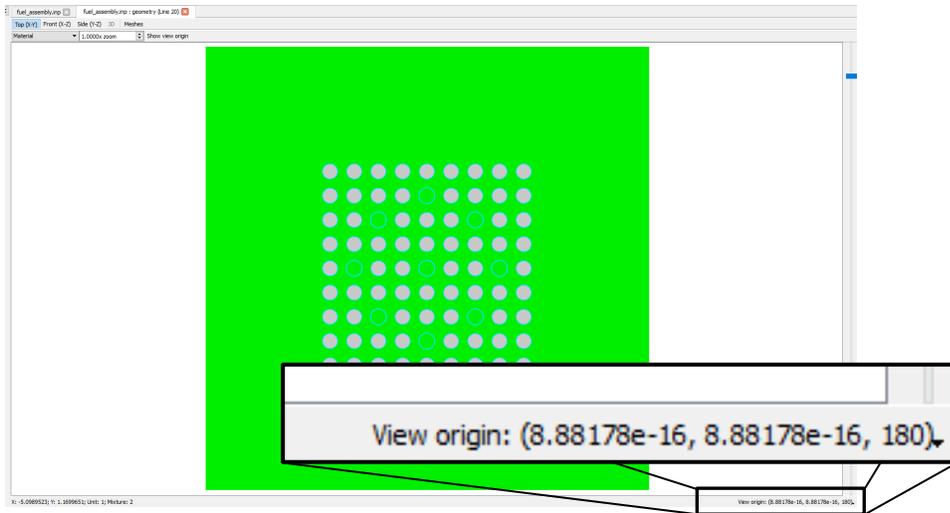**Figure 249. Mouse pointer location and view origin displays.**

155

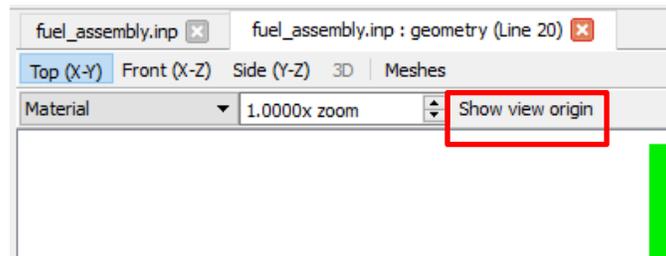**Figure 250. XY visualization at Z=180 cm plane.**

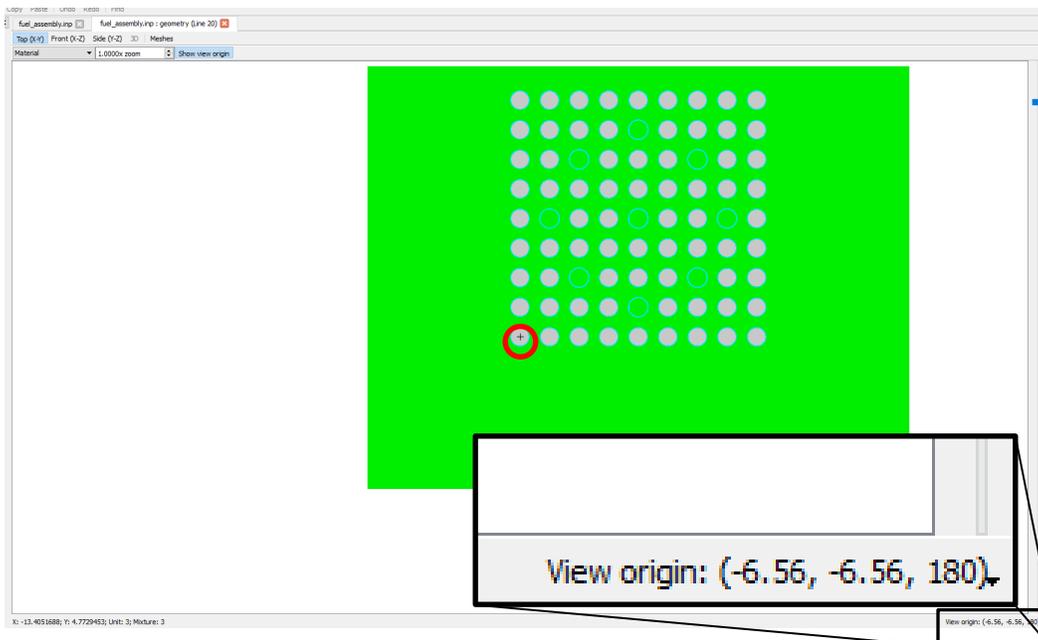**Figure 251. Location of Show view origin button.**

Figure 252. XY visualization centered at (-6.56, -6.56) at Z=180 cm plane.
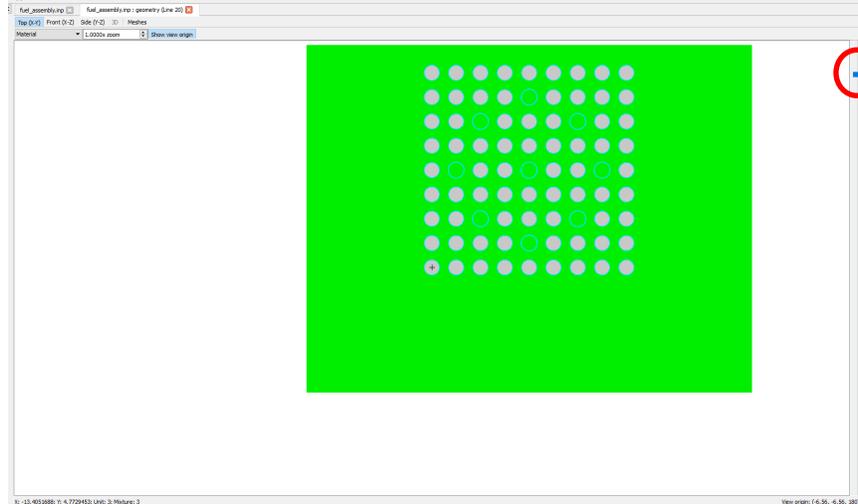
Figure 253. Location of slider to control the position of visualization plane.

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 254. The render mode dropdown is shown in Figure 255; the first three options are discussed here, and **Overlay** and **Overlay + boundaries** are discussed in Section 8.3. 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 256; note that the unit boundaries are now shown in the array, as well as in 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 257.
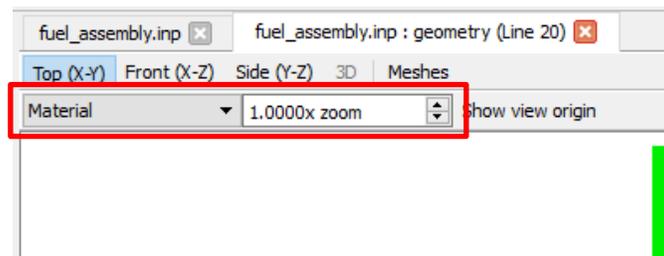

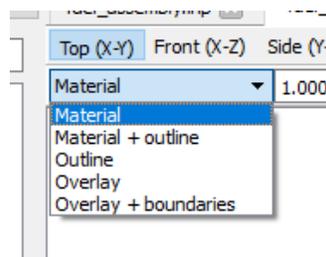**Figure 254. Display controls for render mode and zoom.**
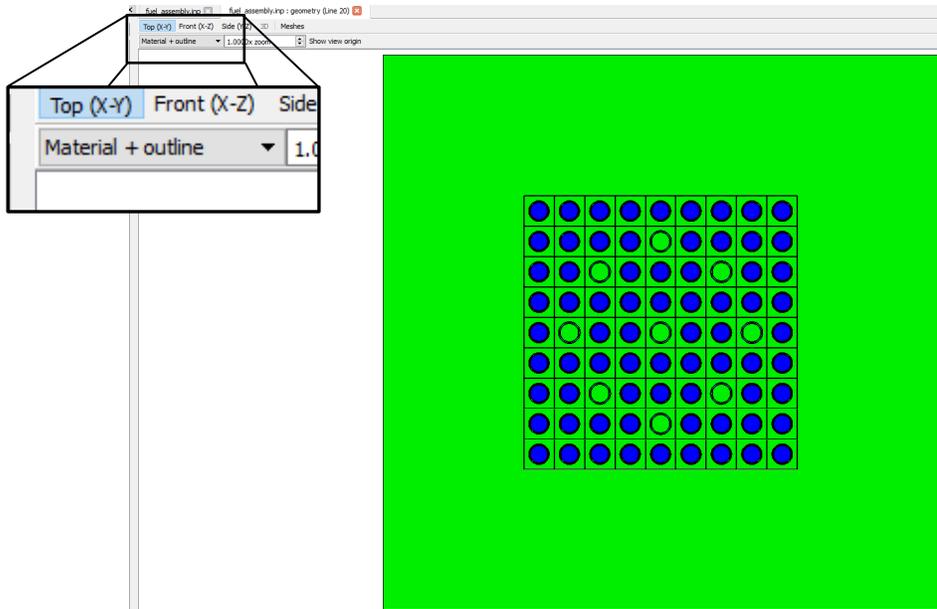

Figure 255. Render mode dropdown.

157

Figure 256. Midplane view of fuel assembly in Material + outline render mode.
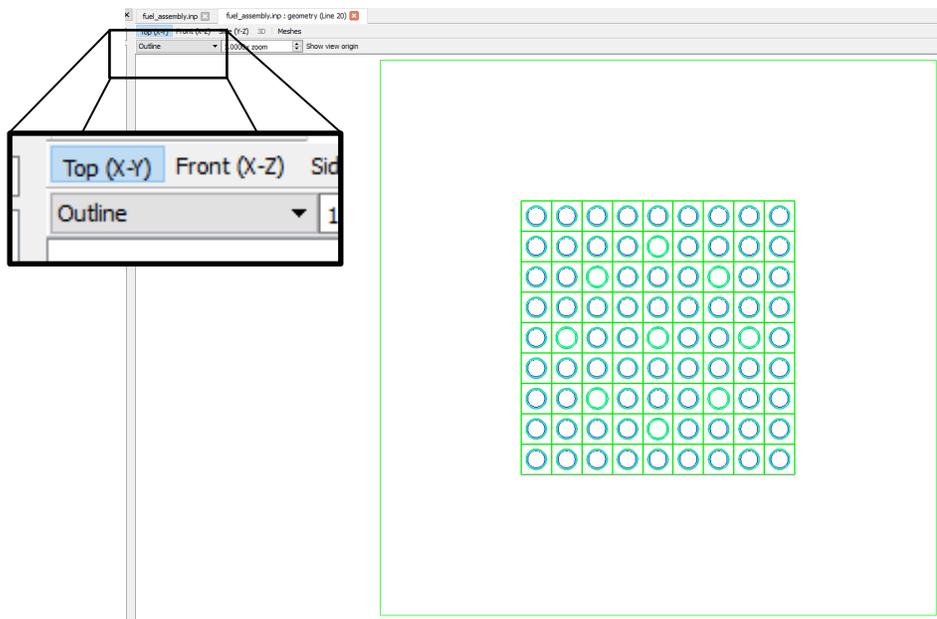
Figure 257. Midplane view of fuel assembly in Outline render mode.

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 254. 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 258, and then attempts to fit this box into the view window, as shown in Figure 259. 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.

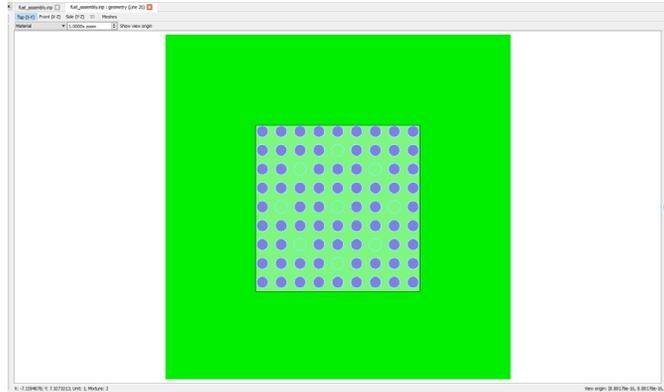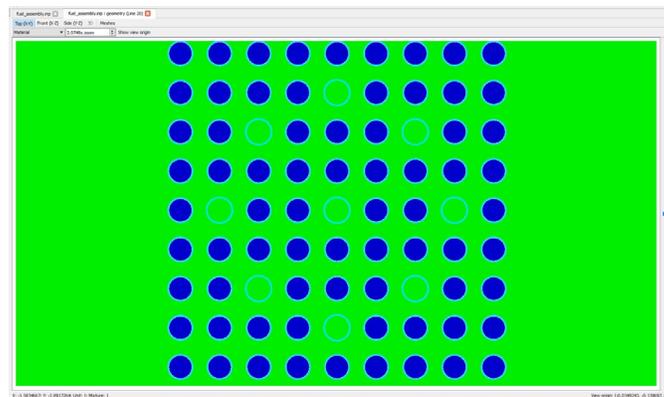**Figure 258. Fulcrum visualization while creating zoom control box.**

Figure 259. Image resulting from the zoom shown in Figure 242.

The last visualization control to discuss here is panning the image. Again, there are two methods of doing this. 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 moves 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.

8.2.2   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 of these plot types 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 260. 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 261. 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 262, 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 263, indicating that the source may converge by approximately generation 75.
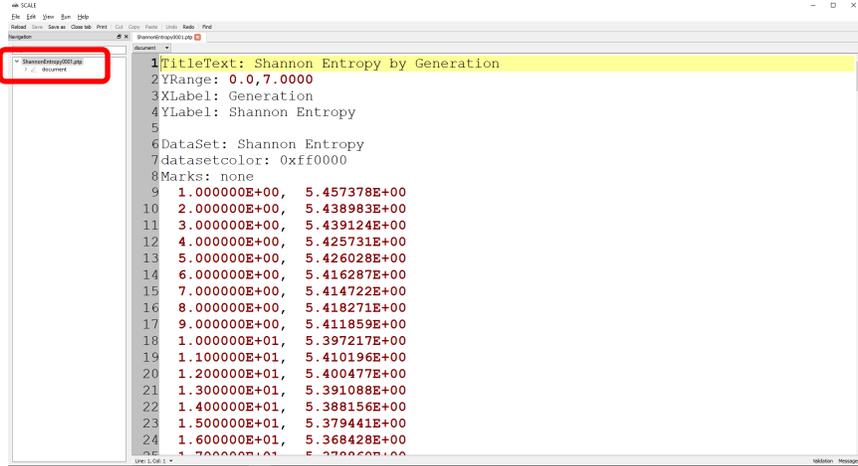

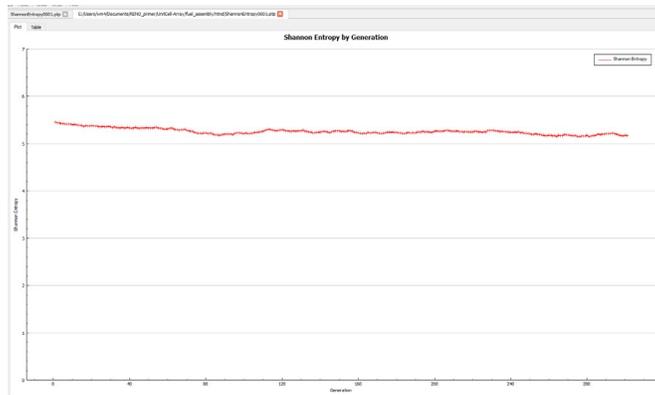**Figure 260. Fulcrum default view of a *ptp file.**

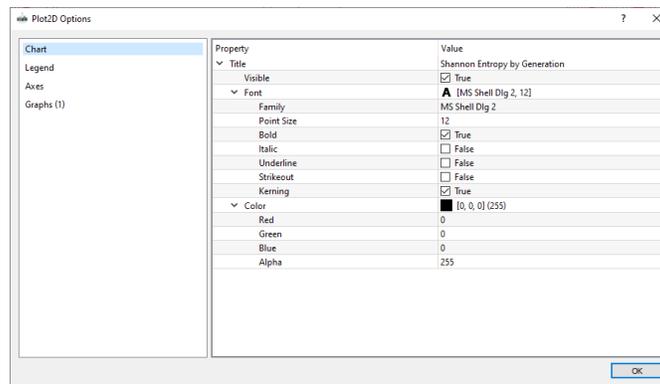**Figure 261. Default view of Shannon entropy by generation plot.**

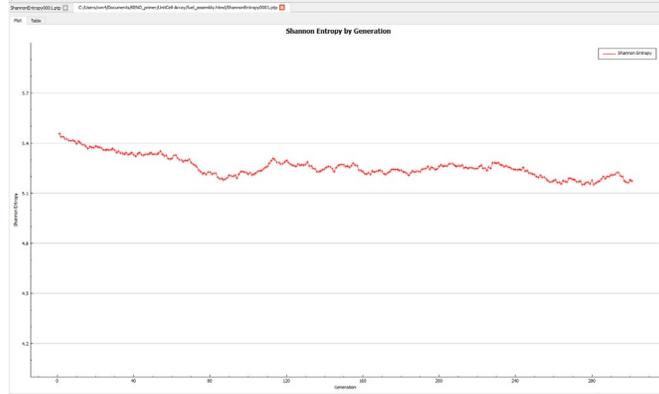**Figure 262. Resized plot options window.**

Figure 263. Rescaled view of Shannon entropy plot.

The second set of plots to discuss are the ones labeled "errorbars0001.ptp" and "errorbars20001.ptp." The plots provide the k_{eff} convergence summary based on number of generations run (0001) and number of generations skipped (20001). To view a plot, click on **File** > **Open file** (or press Ctrl+o) and select errorbars0001.ptp. By default, Fulcrum will 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 264, the average k_{eff} is plotted as a circle, and uncertainty bars are provided corresponding 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 265. This provides the average k_{eff} as a function of generation skipped and also shows excellent k_{eff} convergence behavior.
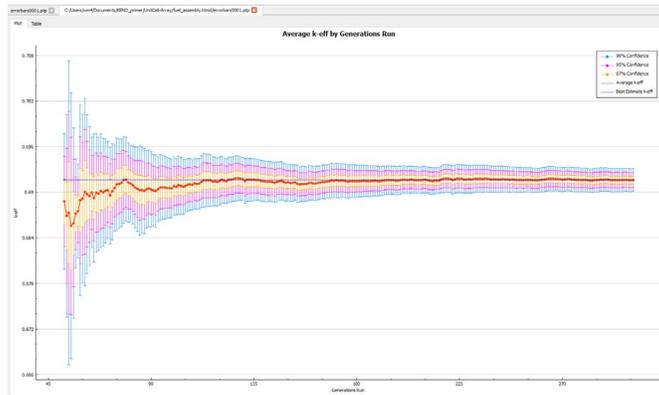

Figure 264. Plot of average k_{eff} by generation run.
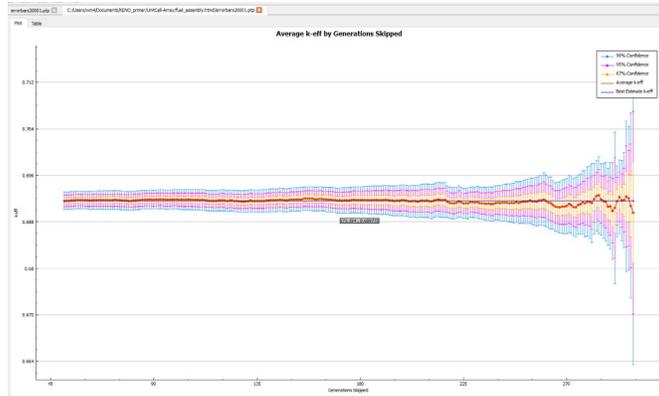
Figure 265. Plot of average k_{eff} by generation skipped.

The final set of plots is the "freqgen" series, which 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 266. The histogram for the last quarter of active generations (40001) is shown in Figure 267. It appears that convergence was achieved during the discarded generations because no large shifts are evident among the histograms in Figure 266. Each histogram can be reviewed separately if desired.

Figure 266. Histograms of generation k_{eff} values.

162

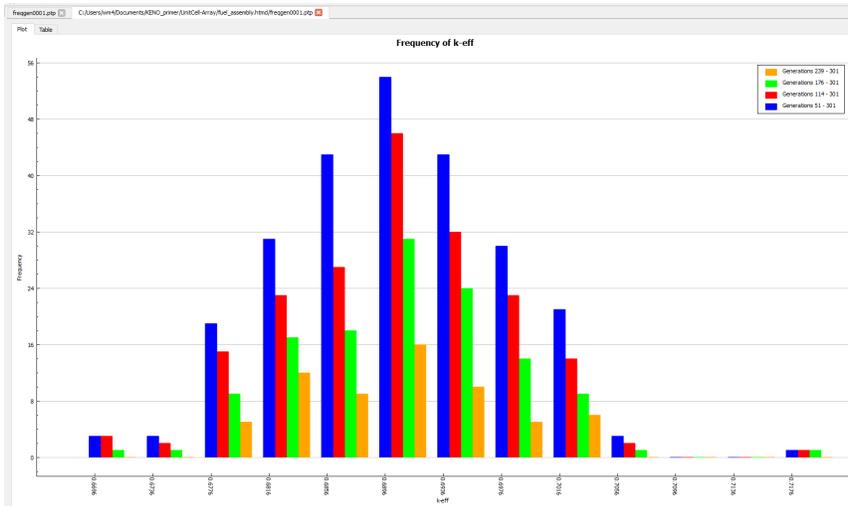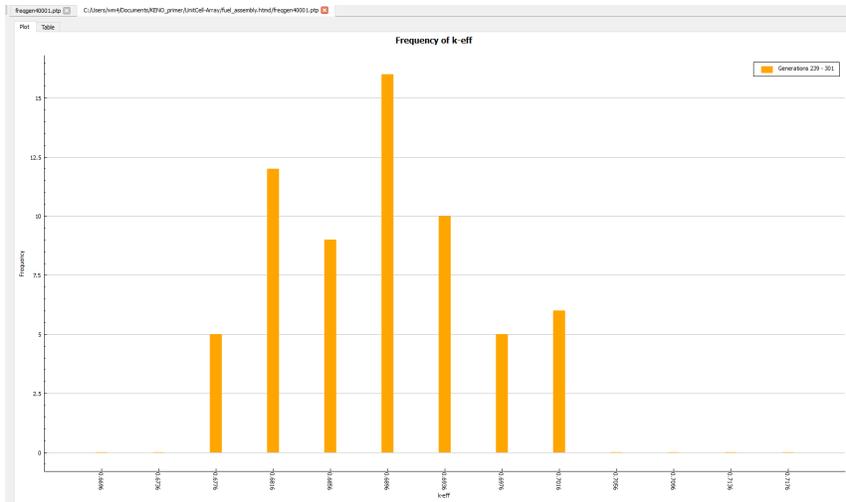**Figure 267. Histogram of generation k_{eff} values for the last quarter of active generations.**

8.2.3 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 using the requested format if a mismatch occurs such as 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 have been installed to another path, then navigate to that directory. Open **ce_v7.1_endf.xml**, as shown Figure 268; this file contains the path information to the actual CE cross section files which are 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 269. Next, expand the **Neutron** data and scroll down to expand u-238. Note that the isotopes available are listed alphabetically and that available temperatures are displayed in Kelvin after expansion of the **u-238** entry, as shown in Figure 270. Next double-click on the **293** entry to generate the list of reaction cross sections available at that temperature, as shown in Figure 271. Scroll down and expand the **u-238 mt=102 n,gamma** entry, 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 272; note that, as is typical for cross section plots, the data are displayed on a log-log scale.
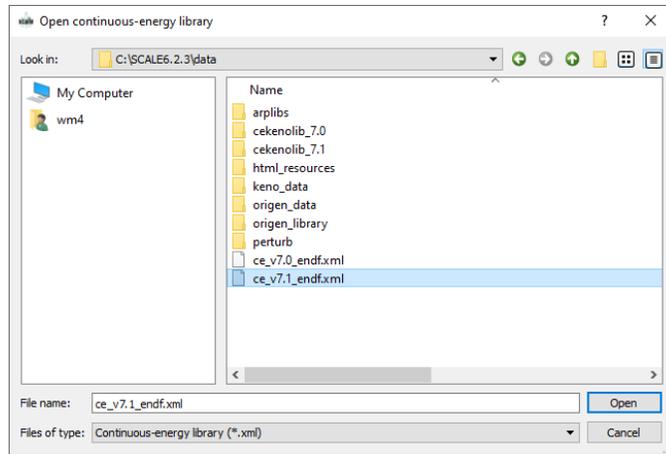
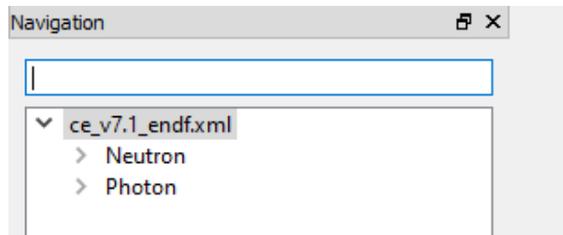**Figure 268. Opening the CE xml file.**

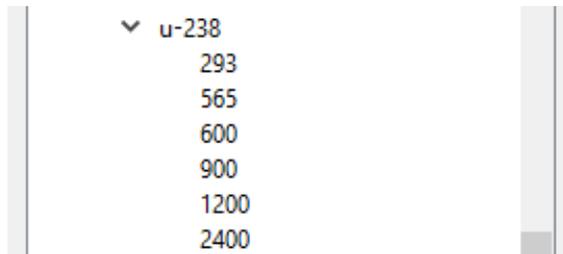**Figure 269. Default Navigation panel after opening a CE library.**

Figure 270. Expanded ^{238}U entry showing temperatures at which data are available.
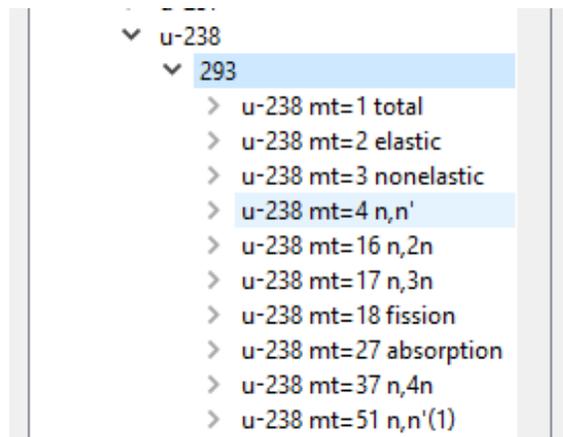
Figure 271. Top portion of the list of reactions at 293K available for plotting.
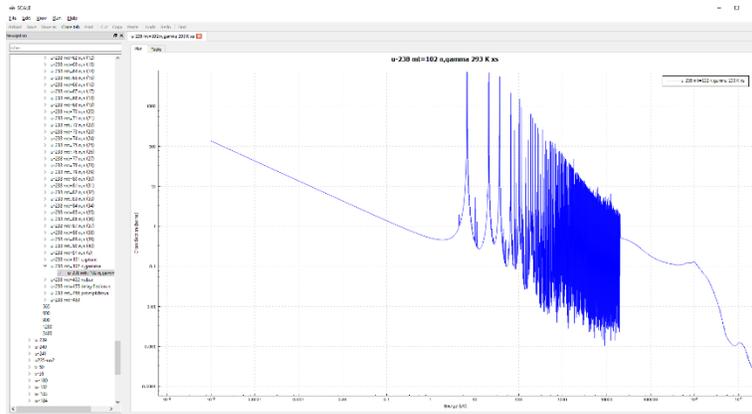
Figure 272. Default view of ^{238}U (n,γ) CE cross section in Fulcrum.

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 273. 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 274. 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 275. The resulting plot comparing the MG and CE cross sections is shown in Figure 276.

Figure 273. MG library loaded into the Navigation panel below the CE library.

Figure 274. MG library in the Navigation panel with ^{238}U expanded.

Figure 275. Adding MG ^{238}U (n,γ) cross section to existing CE plot.

Figure 276. 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 r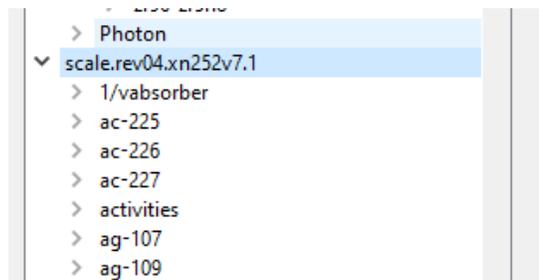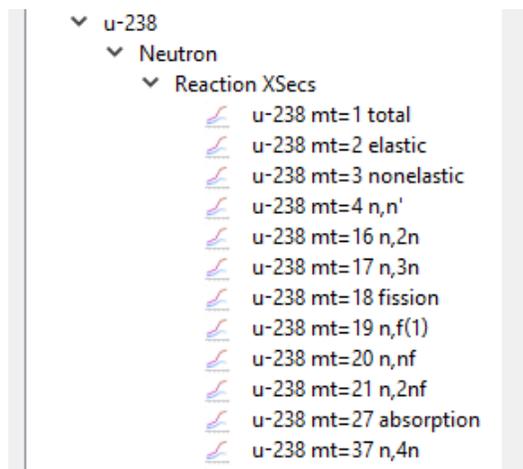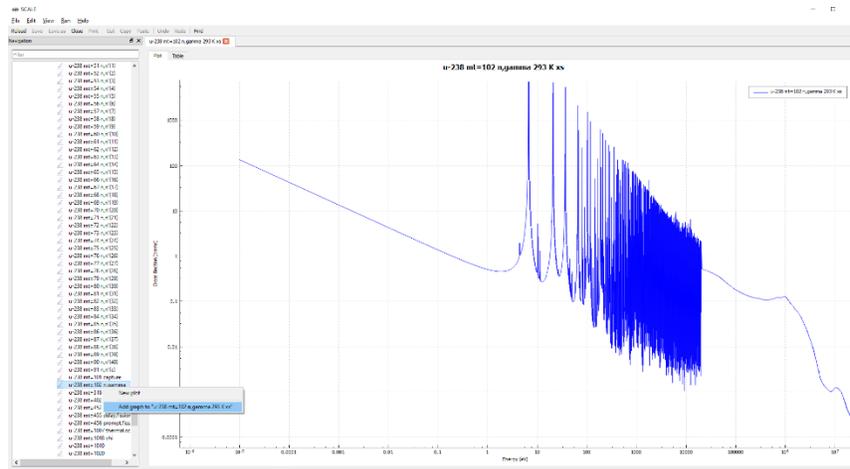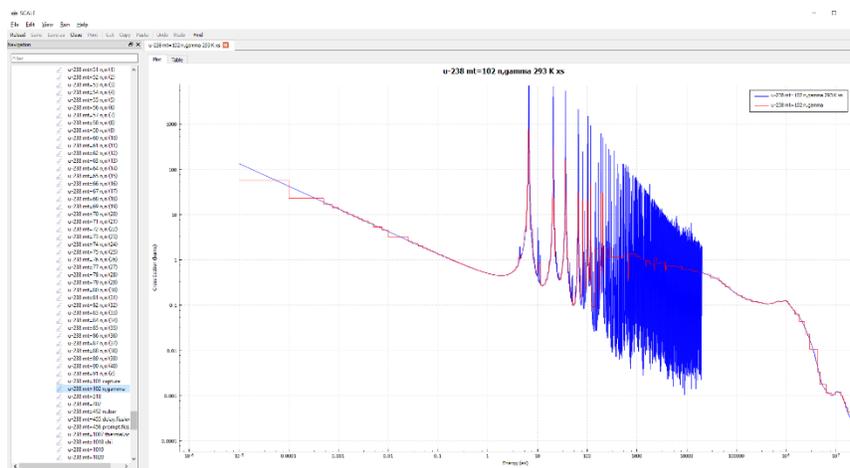oot of the diagonal elements of the covariance matrix) will be discussed here. This data cannot be plotted on the same figure as cross section data because the unit used for the covariance data, relative error (%), is not the same as that for the cross section data, barns. To load the covariance data, 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 SCALE 6.2.3. The covariance library loads in the Navigation panel below the MG library because of its higher revision number; see Figure 277. 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 278. 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 will provide a table of the points and data used in generating the plot, as shown in Figure 279. 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 280. 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

166

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 281 shows the screen while the plot is being repositioned, and Figure 282 shows the resulting view of both plots. In this configuration, the cross section and its covariance can be compared directly.
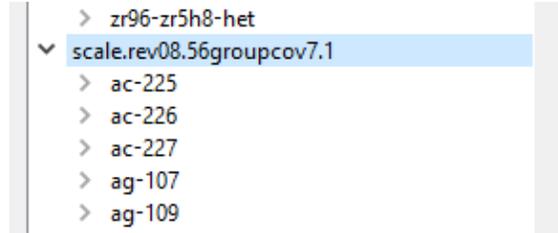
Figure 277. Covariance library loaded into the Navigation panel below the MG library.
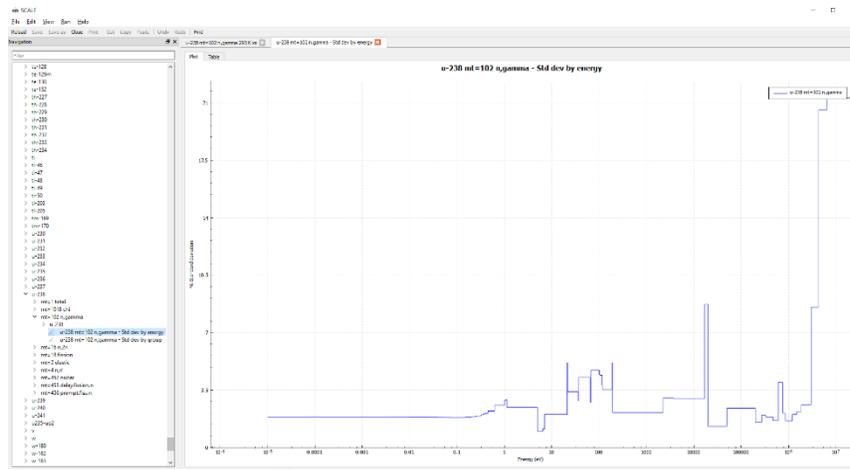
Figure 278. Plot of standard deviation of ^{238}U (n,γ) cross section.

Figure 279. Table of values for standard deviation of ^{238}U (n,γ) cross section.
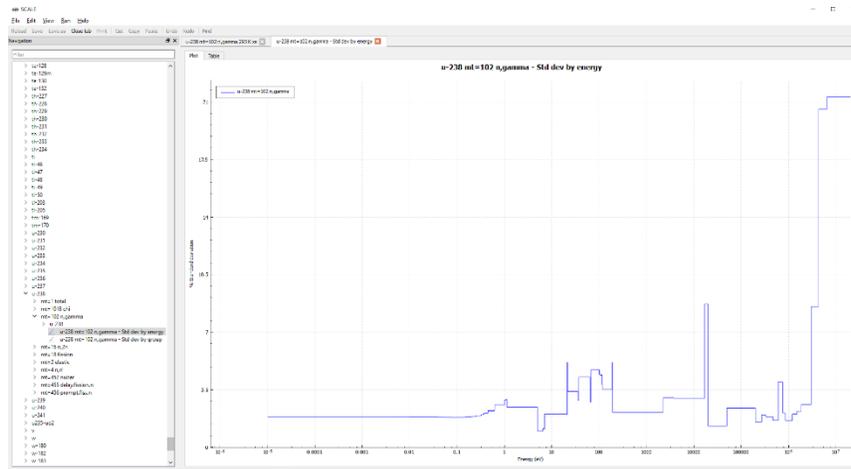
Figure 280. Covariance plot with repositioned legend.
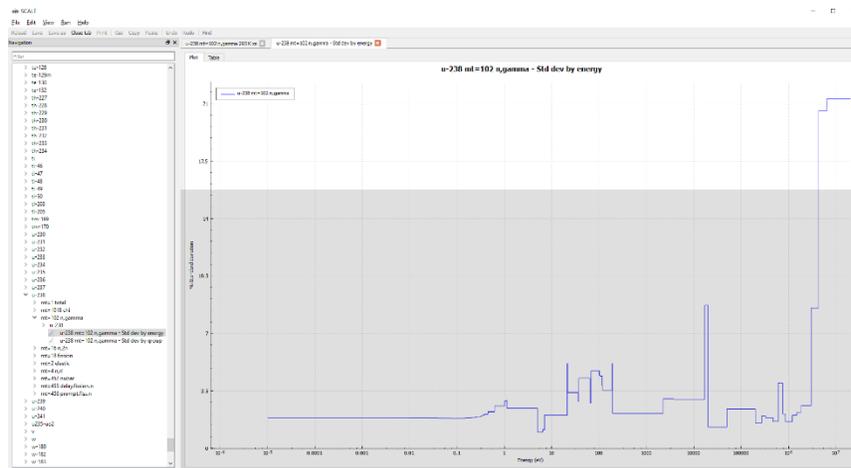
Figure 281. Positioning the covariance plot in the upper half of the view window.
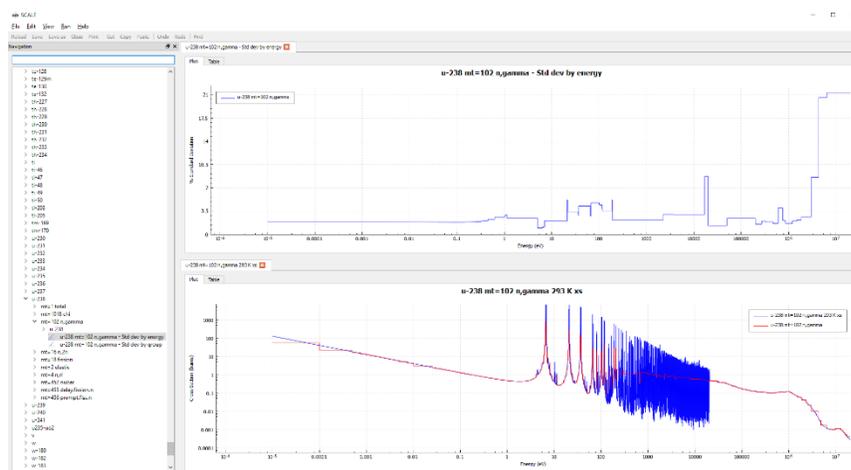
Figure 282. ^{238}U (n,γ) covariance and cross sections in a single view in Fulcrum.

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. Panning the

image is accomplished 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**.

8.3 PLOTTING MESH DATA IN FULCRUM

The primary mesh-based data generated and used in KENO is the neutron production edit enabled by a *cds=yes* entry in the PARAMETER block. The mesh or grid 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 plotted on top of the Fulcrum 2D geometry visualization. This capability will be generated here by specifying a mesh on the fuel assembly problem from Section 7.5.1; the mesh will be equal to the fuel rod pitch in X and Y and 4 in. (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. An updated PARAMETER block is shown in Figure 283, 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 284. Mesh planes can be added explicitly or by specifying a range for linear interpolation; only Cartesian mesh are supported in KENO. Unfortunately, autocomplete is not supported in Fulcrum in the GRIDGEOMETRY block. First, specify explicit planes in the X, Y, and Z directions with `xplanes 15.0 -15.0 end`, `yplanes 15.0 - 15.0 end`, and `zplanes 208 -208 end`. These planes are near or just beyond the outer boundary of the model, and the input is shown in Figure 285. Next, 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. The radial mesh to locate each rod in a separate interval is thus `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 286. 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 60,000 neutrons per generation for 3,500 generations, skipping the first 50.

```
read parameters
 npg=60000 nsk=50 gen=3500
 cds=yes
end parameters
```

Figure 283. Updated PARAMETER block with cds=yes included to enable the neutron production tally.

Figure 284. Adding GRID block to specify mesh for the neutron production tally.

```
54 read grid 1
55  xplanes 15.0 -15.0 end
56  yplanes 15.0 -15.0 end
57  zplanes 208 -208 end
58 ' TODO: define grid
59
60 end grid
```

Figure 285. GRIDGEOMETRY block with explicit plane declarations included.

```
54 read grid 1
55  xplanes 15.0 -15.0 end
56  yplanes 15.0 -15.0 end
57  zplanes 208 -208 end
58  xlinear 9 -7.38 7.38
59  ylinear 9 -7.38 7.38
60  zlinear 36 -182.88 182.88
61 end grid
```

Figure 286. Complete GRIDGEOMETRY block for fuel assembly problem.

After execution, the results can be visualized in Fulcrum. Open the input and visualize the results, as discussed in Section 8.2.1. 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 287. 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 288. 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 datasets are shown. Expand **fission rate** as shown in Figure 289; 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 290. 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 291.
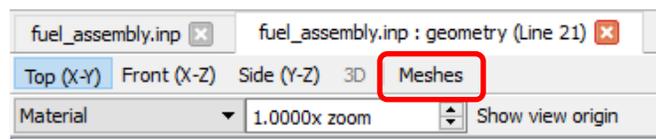
Figure 287. Location of the Meshes button in the Fulcrum visualization window.
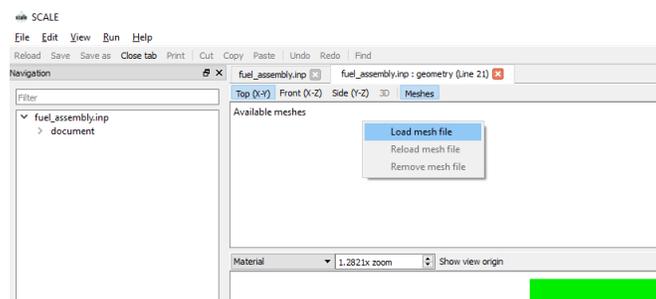
Figure 288. Loading a mesh file in Fulcrum.
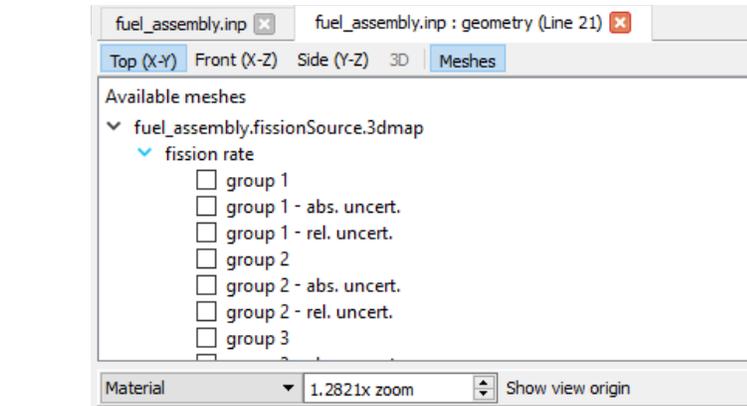
Figure 289. Loaded and expanded 3dmap file for the fuel assembly model.
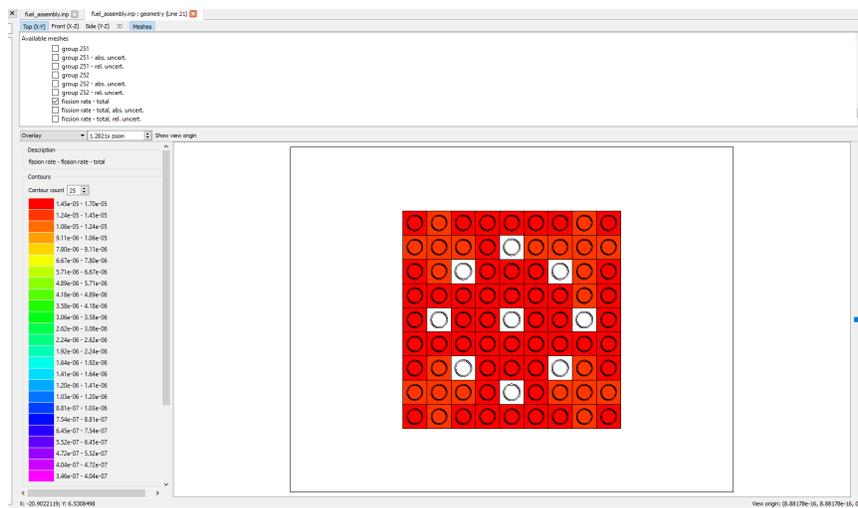
Figure 290. Default view of neutron production rate at the fuel assembly midplane.
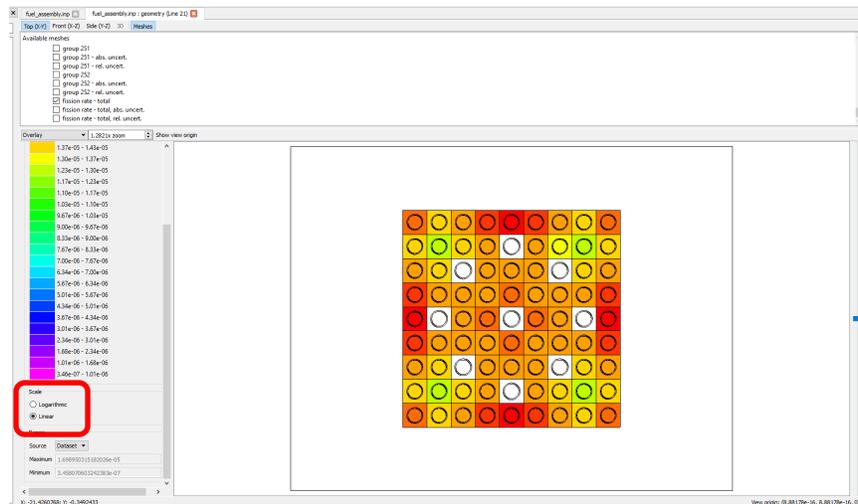
Figure 291. Fuel assembly midplane with colors rescaled to linear bins.

The mesh itself can also be displayed in Fulcrum. Change the Render mode dropdown from **Overlay** to **Overlay + boundaries** to show the mesh planes. It may be necessary to change the zoom factor slightly

171

to display all mesh planes, as shown in Figure 292. 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 293. The resulting updated figure is shown in Figure 294. 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 295. 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`, also shown in Figure 295. The updated plot, shown in Figure 296, provides a better view of how the uncertainty varies in this specific plane.
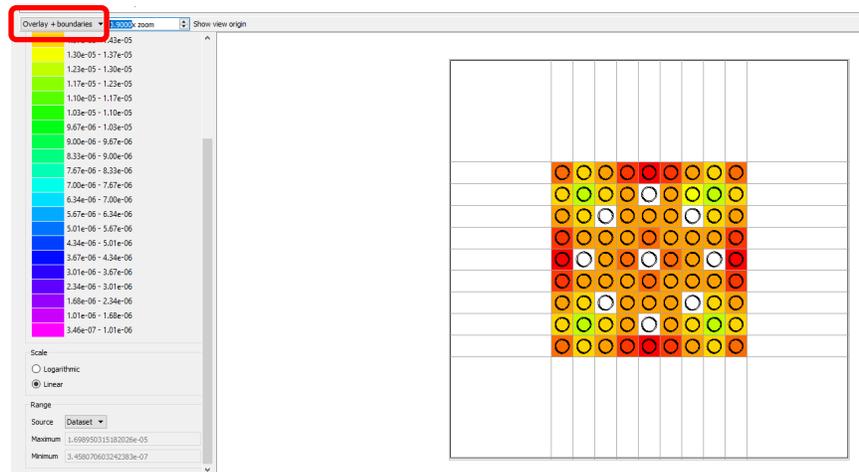
Figure 292. Midplane neutron production rate with mesh boundaries displayed.
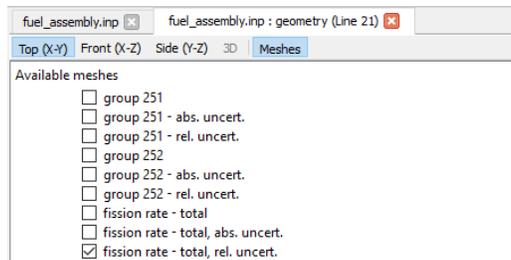
Figure 293. Selecting relative uncertainties as the overlay quantity to view.
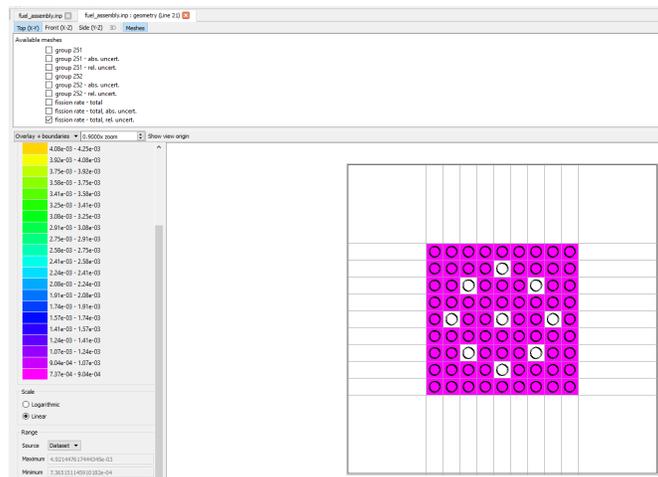
Figure 294. Relative uncertainties for the neutron production rate at the fuel assembly midplane.

172
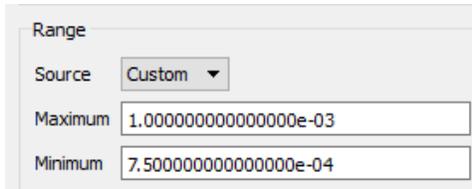
Figure 295. Updated uncertainty range.
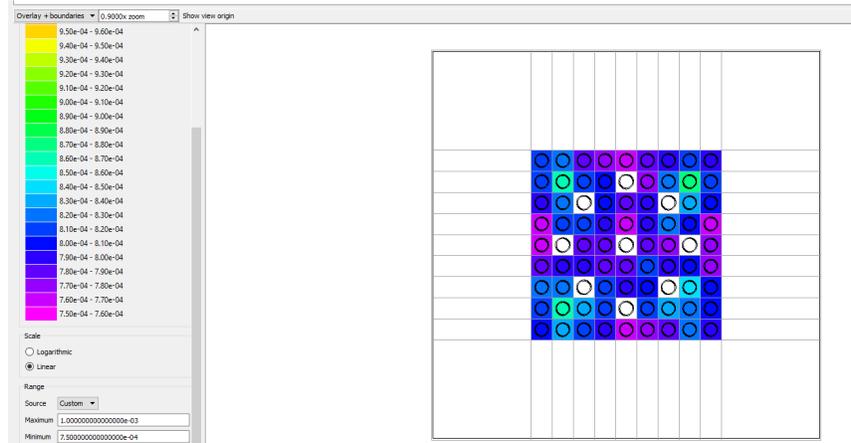
Figure 296. Overlay plot of uncertainties with custom value range.

The visualization controls discussed in Section 8.2.1 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. Axial slices can also be viewed, as shown in Figure 297, 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 297. Axial view of the fuel assembly neutron production rate distribution rotated so 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, although 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 298, and select **Create plot** (Figure 299). Select **z axis** as the **Independent axis** in the **Options** box of the subsequent window as shown in Figure 300. 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 8.2.2 and as shown in Figure 260. Double click the word **document** in the **Navigation** panel, as shown in Figure 301, and Fulcrum will generate a new plot, as shown in Figure 302.

173
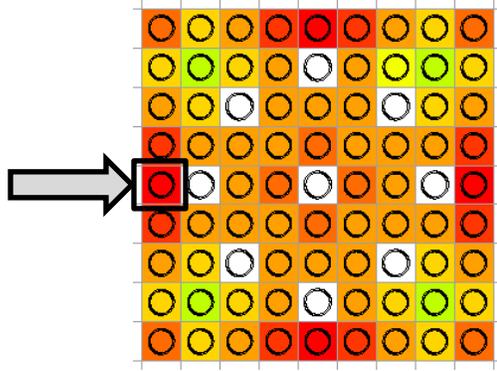
Figure 298. Selection of rod for axial plot.
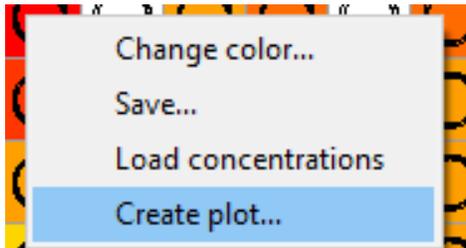
Figure 299. Selecting Create plot… in Fulcrum.
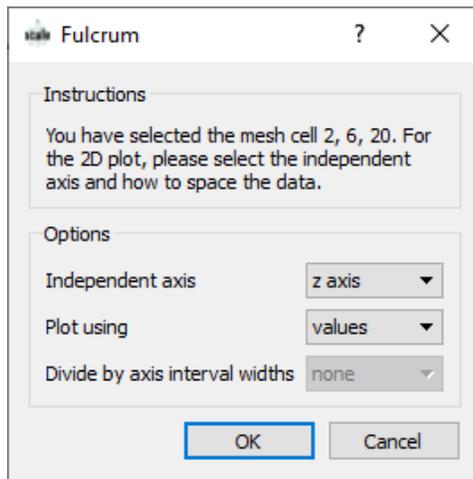
Figure 300. Selecting Independent axis for axial trace.
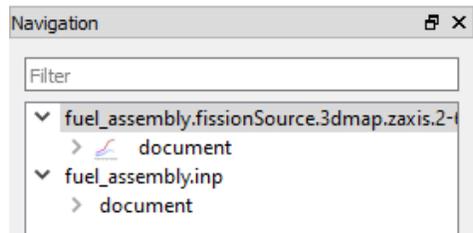
Figure 301. Plot file generated by Fulcrum and displayed in the Navigation panel.
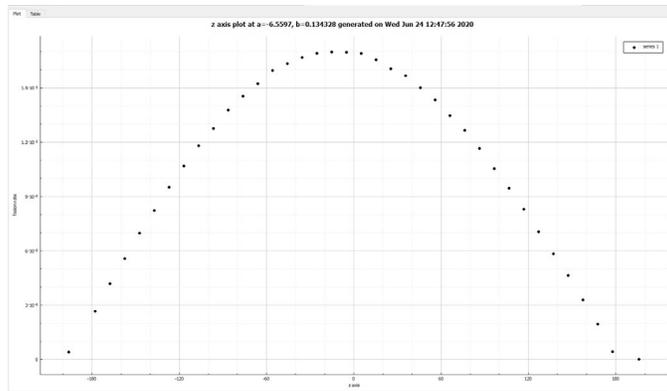
Figure 302. Axial trace of neutron production rate.

8.4 INTERACTIVE 3D VISUALIZATION WITH KENO3D

KENO3D enables KENO V.a users to interactively display their 3D geometry models. KENO3D only runs on the Windows operating system. The interactive options include the following:

- Choice of shaded or wireframe images
- Standard views such as top view, side view, front view, and isometric (3D) view
- Model rotation
- 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 V.a input files. It attempts to verify that the KENO geometry input is "legal," i.e., it conforms to the code input 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 of how to use KENO3D is 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 described in Section 7.5.1, 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 303.
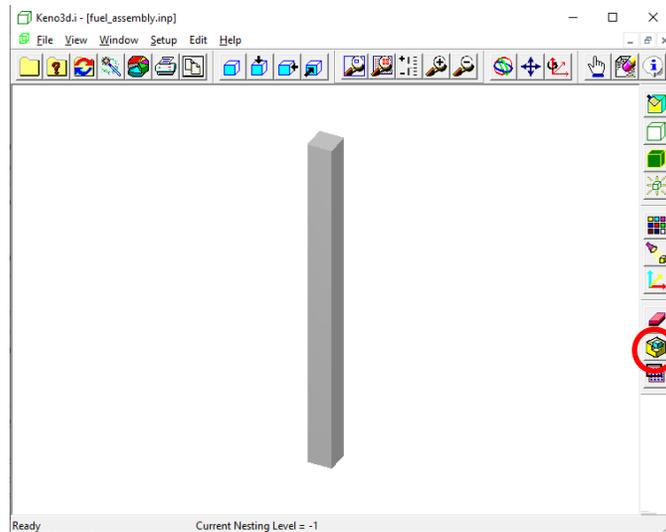
Figure 303. Default KENO3D view of the fuel assembly model.

This isometric view shows the water reflector that surrounds the assembly. Next, click on the **Remove a Section** button (circled in red in Figure 303). This opens the **Section to remove from model** dialog form shown in Figure 304. 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 305.
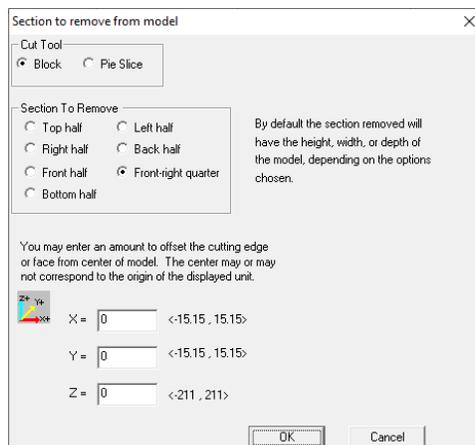
Figure 304. Remove section dialog.
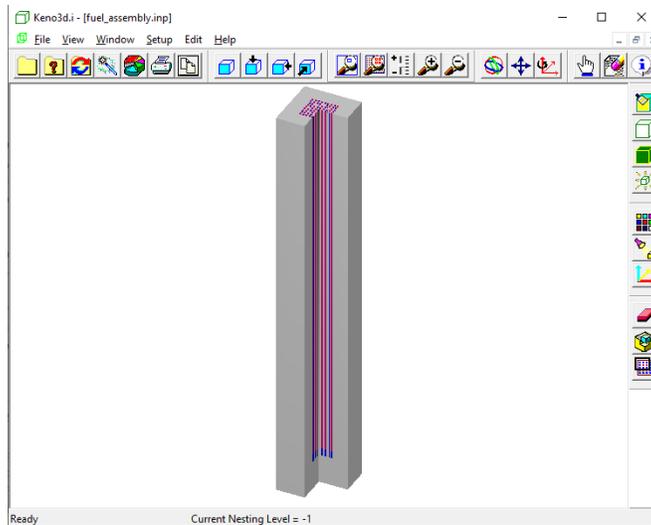
Figure 305. Cutaway view of the bottom half of the assembly.

Next, hide the water so that the fuel rods are more 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 Figure 306.
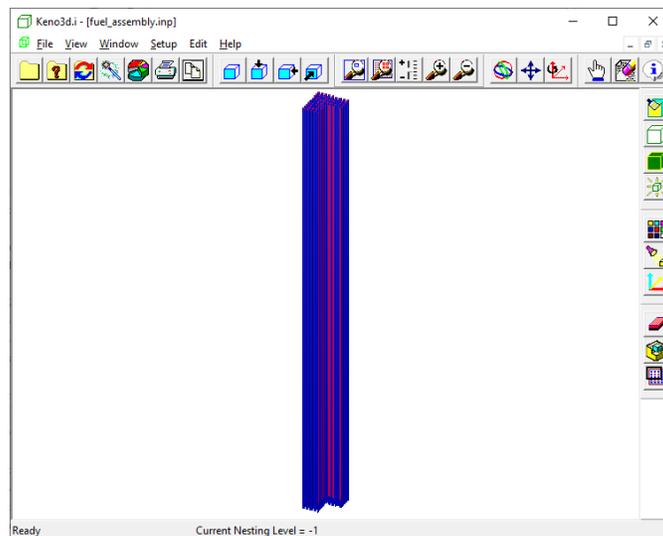
Figure 306. Cutaway view of the bottom half of the assembly with water removed.

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 window around the bottom portion of the assembly (Figure 307). Once the window is satisfactory, left click again to zoom in. The resulting image should look like Figure 308.

177
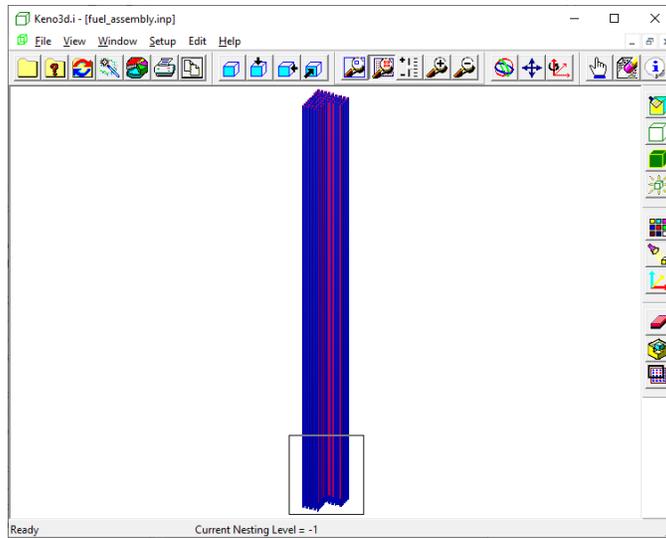
Figure 307. Zooming on lower portion of the fuel assembly.
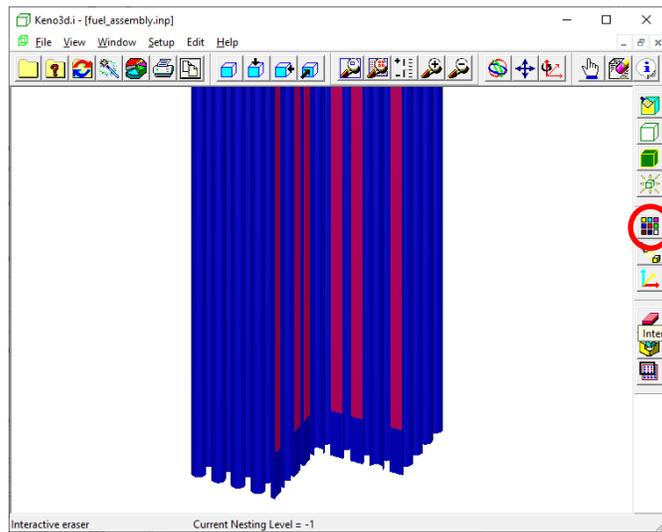
Figure 308. Zoomed view of cutaway view of the bottom of the fuel assembly with water removed.

Next, display the legend by clicking the **Display legend** button, circled in red in Figure 308. Click **OK** in the Legend options dialog to display an image like that shown in Figure 309.
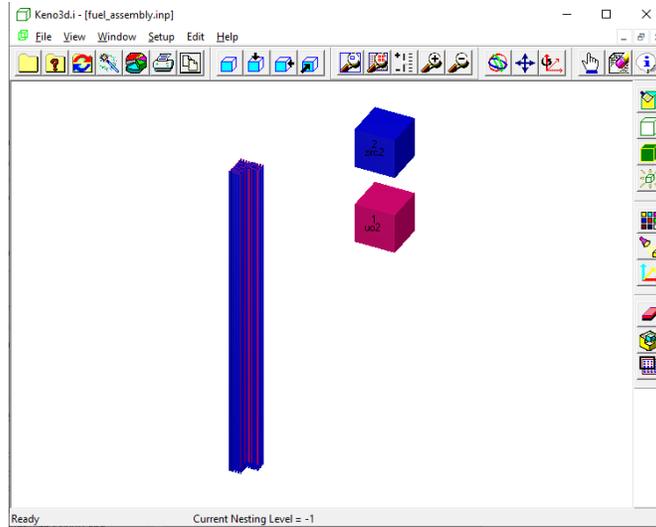
Figure 309. Default view with legend.

Adding the legend removes the focus on the zoom window and returns the view of the bottom half of the fuel assembly. In addition, the legend is too large and 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 310). Click **OK** and the screen looks like Figure 311. Finally, zoom in on the lower portion of the fuel assembly again to generate an image like Figure 312.
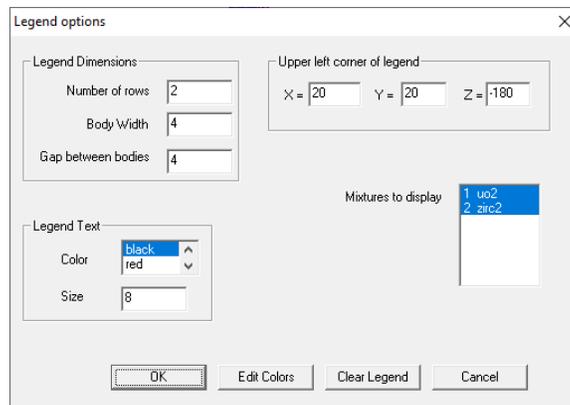
Figure 310. Legend options dialog.
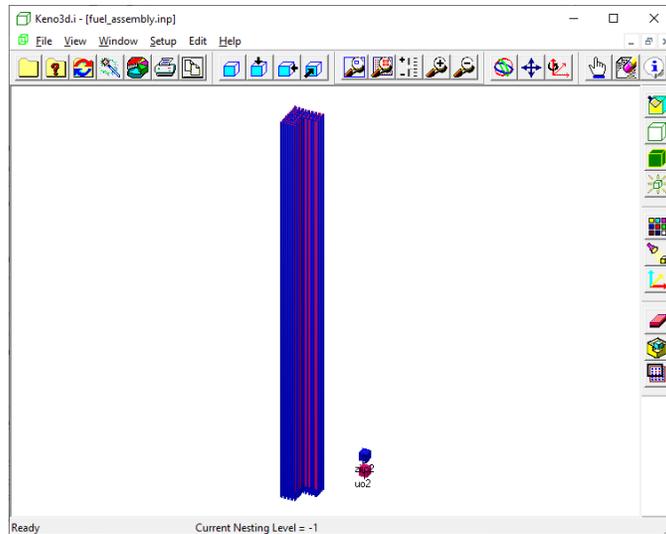
Figure 311. Image with modified legend.
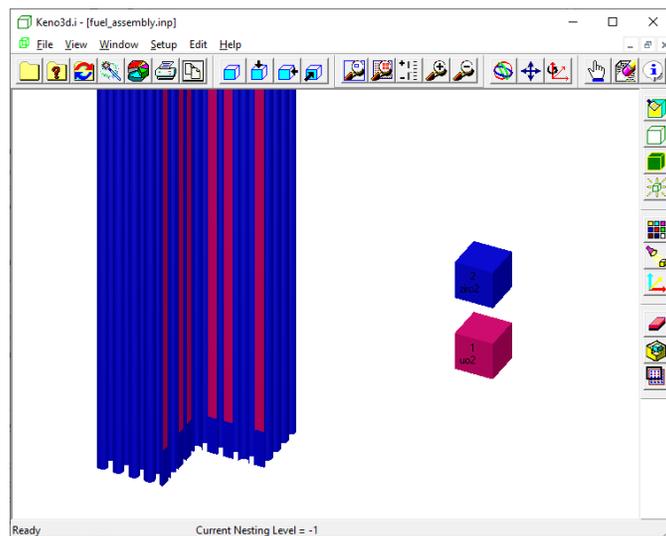
Figure 312. Zoomed image with modified legend.

8.5   PLOTTING CALCULATED RESULTS WITH KMART AND KENO3D

KMART (KENO Module for Activity-Reaction Rate Tabulation) is a tool that post-processes information from a KENO V.a run to provide reaction rates by nuclide and the collapsing and printing of fluxes. See the [KMART section] for details on KMART. KENO3D has the capability to plot the calculated results produced by KENO V.a/KMART overlaid on the 3D geometry model. The resulting *kmt file also contains the data in text tables that are often easier to read than the output KMART adds to the standard output file.

There are a few items which the KENO V.a user must be aware of when using KMART. First, 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 KENO PARAMETER block. Secondly, 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 V.a collects calculated data on a unit and region basis. The level of detail in the

180

results is directly dependent upon the level of detail in the individual geometry regions. The SHEBA solution reactor is used as an example in this section. A simple KENO V.a input is provided in Figure 313. A view of the model in KENO3D is shown in Figure 314. In this example, the input file name is sheba2.inp.

```
=csas5
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'
 zcylinder 0 1 2.54 44.8 0
 zcylinder 2 1 3.175     44.8  0
 zcylinder 1 1  8.14917 44.8    0
 zcylinder 1 1 16.29833 44.8    0
 zcylinder 1 1 24.4475  44.8  0
 zcylinder 2 1 25.4     44.8  0
 cuboid 0 1 25.4  -25.4 25.4  -25.4 44.8  0
unit 10
com='bottom of  tank'
 zcylinder 0 1 2.54     2.8575      0
 zcylinder 2 1 25.4     2.8575      0
 cuboid 0 1 25.4  -25.4 25.4  -25.4 2.8575      0
unit 11
com='void section of tank'
 zcylinder 0 1 2.54     76.5175     44.8
 zcylinder 2 1 3.175    76.5175     44.8
 zcylinder 0 1 24.4475  76.5175     44.8
 zcylinder 2 1 25.4     76.5175     44.8
 cuboid 0 1 25.4  -25.4 25.4  -25.4 76.5175     44.8
unit 12
com='top of tank'
 zcylinder 0 1 2.54     1.905 0
 zcylinder 2 1 25.4     1.905 0
 cuboid 0 1 25.4  -25.4 25.4  -25.4 1.905 0
global unit 13
 array 1  0 0 0
end geometry
read array
ara=1 nux=1 nuy=1 nuz=4
 fill 10 1 11 12
end fill
end array
end data
end
```

Figure 313. Sample input for simplified Sheba solution reactor.
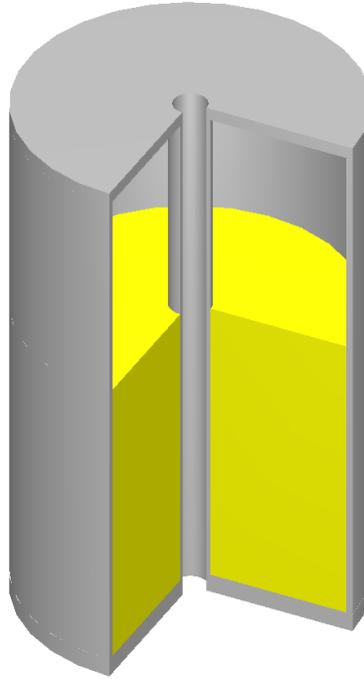
Figure 314. KENO3D rendering of simplified Sheba model.

The next step is to add parameters to write the restart file and add the flux edits to it for use 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 just at the end of the calculation. In this case, it need only 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 315.

```
10 read parameter
11   npg=2000 nsk=50 gen=250
12   flx=yes res=250 wrs=35
13 end parameter
```

Figure 315. 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 KMART5 sequence, as shown in Figure 316. The skeleton KMART input is shown in Figure 317.
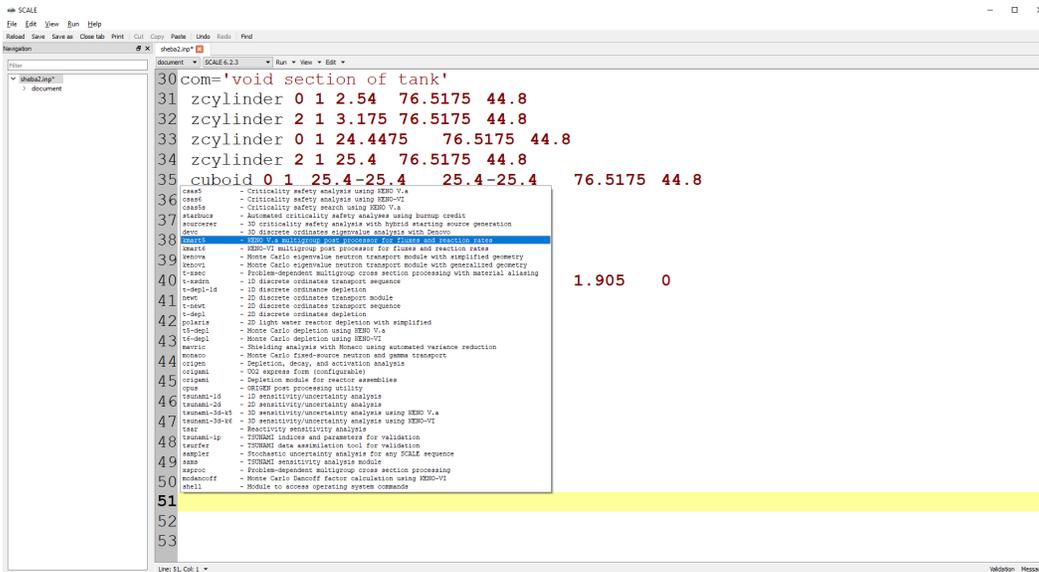
Figure 316. Adding the KMART sequence input after the CSAS sequence input.

```
49=kmart5
50
51read initial
52
53' TODO: define initial
54
55end initial
56
57
58end
```

Figure 317. 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 that are needed are 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 so that small reaction rates in large regions do not appear 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 with a kmt extension to allow KENO3D to recognize it; in this example, that means that the filename is sheba2.kmt. The complete KENO3D input specification is therefore keno3d 40 sheba2.kmt. The complete INITIAL block input is provided in Figure 318, with extra blank lines and extraneous comments removed for clarity and brevity.

```
50read initial
51  xunit=4 kunit=35
52  actbygroup rrpvol
53  keno3d 40 sheba2.kmt|
54end initial
```

Figure 318. 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 5. 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 319. 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; input 92235 18. The complete ACTIVITY block is shown in Figure 320.

```
49 =kmart5
50 read initial
51  xunit=4 kunit=35
52  actbygrp rrpvol
53  keno3d 40 sheba2.kmt
54 end initial
55
56 activity
   collapse
57 end
58
```

Figure 319. Adding ACTIVITY block to the KMART input.

```
55 read activity
56   1001 27
57  92235 18
58 end activity
```

Figure 320. Complete ACTIVITY block for the Sheba model.

Table 5. Frequently used MT numbers

MT Number	Reaction Description
1	Total
2	Elastic scattering
18	Total fission
27	Total absorption
1452	Neutron generation from fission ($v \times$ fission)

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

184

groups for the broad groups: `lastg=45`, `lastg=190`, and `lastg=252`. The final COLLAPSE block is shown in Figure 321.

```
59 read collapse
60  lastg=45
61  lastg=190
62  lastg=252
63 end collapse
```

Figure 321. 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 314, the KMART data can be visualized by selecting **View** > **Plot KMart Data**. This generates a **Plot Results** dialog, as shown in Figure 322, providing available data for plotting. Select **U-235 (MT=18)** and click **Generate Plot** to generate an image such as the one shown in Figure 323. Finally, click the **Show legend** button and click **OK**, as described in Section 8.4, to display the color scale for the fission rates as shown in Figure 324.
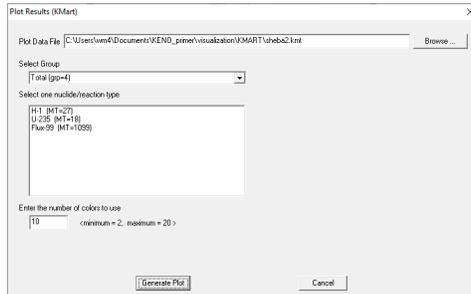
Figure 322. Plot Results dialog in KENO3D for selecting KMART results to display.
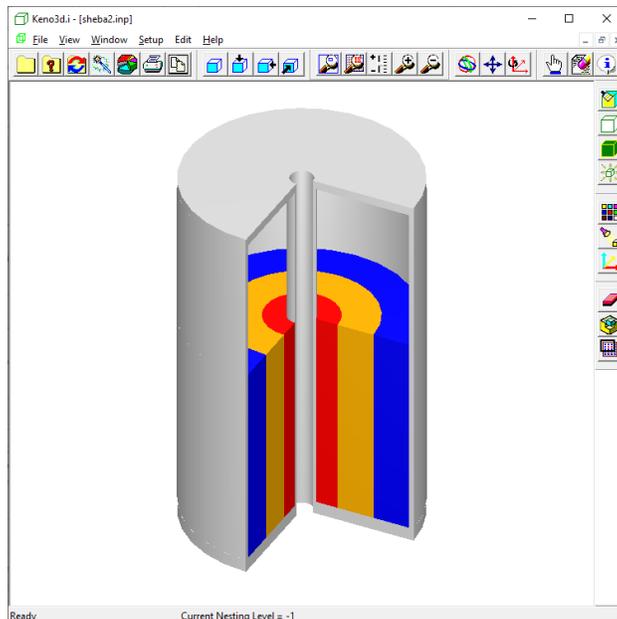
Figure 323. KENO3D rendering of the Sheba model with fission reaction rates.

185
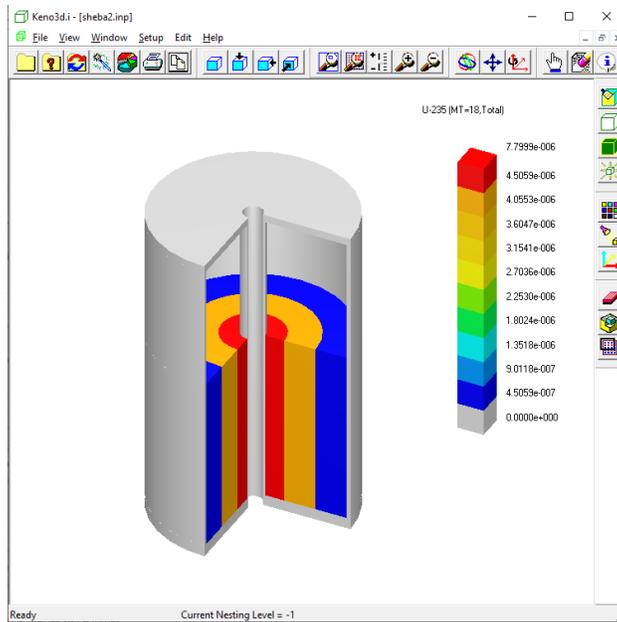
Figure 324. KENO3D rendering of the Sheba model with fission reaction rates and color scale.

The addition of more refinements both 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 325. Obviously this is significantly more detailed than the single axial slice with 3 radial divisions used here, but the detailed model can be developed using the geometry and array constructs discussed in Sections 4.5 and 4.6.
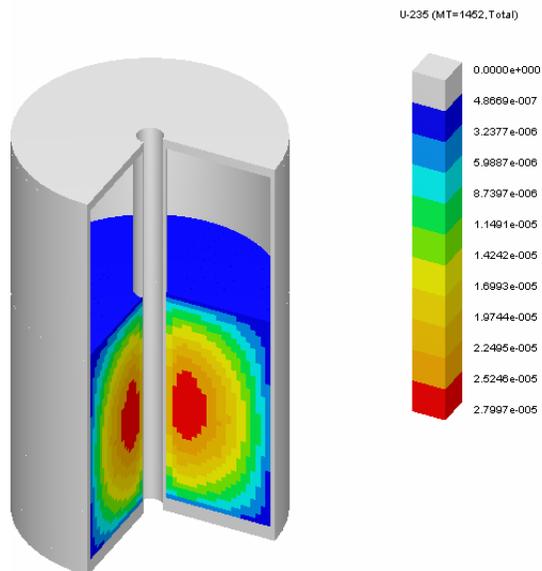
Figure 325. Detailed fission neutron generation rate (MT=1452) in Sheba model.

8.6   SUMMARY

This section provides an overview of the following capabilities within SCALE and Fulcrum:

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

9. CONCLUSION

This primer provides an introduction to the KENO V.a Monte Carlo transport code within the CSAS5 sequence for use in calculating the neutron multiplication factor, also known as the k_{eff} eigenvalue, primarily for nuclear criticality safety analyses. Aside from a quick start guide, the document provides introductions to specifying compositions in SCALE, describing model geometry, problem-specific cross section processing for MG transport problems, and plotting and visualization of 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

10. 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-VI Primer: Performing Calculations Using SCALE's Criticality Safety Analysis Sequence (CSAS6) with Fulcrum*, ORNL/TM-2020/1601, 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

```
=csas5 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 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
read celldata
multiregion spherical right_bdy=vacuum end
1 6.38493 2 6.39763 end zone
end celldata
read geometry
global unit 1
sphere 1 1 6.38493
sphere 2 1 6.39763
end geometry
end data
end
```

INPUT FOR SECTION 3.4.1

```
=csas5 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

```
=csas5 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

```
=csas5 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

```
=csas5 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

```
=csas5 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

```
=csas5 parm=( )
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
 zcylinder 1 1 7.82 7.82 -7.82
end geometry
end data
end
```

INPUT FOR SECTION 4.2.1.2

```
=csas5 parm=( )
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
 zcylinder 1 1 7.82 13.64 -2.0
end geometry
end data
end
```

INPUT FOR SECTION 4.2.1.3

```
=csas5 parm=( )
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
 xcylinder 1 1 7.82 15.64 0.0
end geometry
end data
end
```

INPUT FOR SECTION 4.5.1

```
=csas5 parm=(   )
Primer Example: Reflected Pu
ce_v7.1
read comp
 pu 1 den=15.440000 1.0 293.0
                              94239 95.0
                              94240 5.0 end
 c-graphite 2 den=1.600000 1.0  end
end comp
read geometry
global unit 1
  zcylinder 1 1 2.86 40.96 0.0
  zcylinder 2 1 20.64 58.74 -17.78
end geometry
end data
end
```

INPUT FOR SECTION 4.5.2

```
=csas5 parm=(   )
Primer example: Reflected U sphere
ce_v7.1
read comp
 u-235 1 den=18.740000   end
```

```
  c-graphite 2 den=1.650000   end
 h2o 3    end
end comp
read geometry
global unit 1
  sphere 1 1 7.0
  cylinder 2 1 10 10 -10
  cube 3 1 12 -10
end geometry
end data
end
```

INPUT FOR SECTION 4.6.1

```
=csas5
Primer example: 2x2x2 array
ce_v7.1
read composition
 u 1 den=18.76 1 293
      92234 1
      92235 93.2
      92236 0.2
      92238 5.6   end
end composition
read geometry
unit 1
 zcylinder 1 1 5.748 5.3825 -5.3825
 cuboid 0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
end geometry
read array
  ara=1
  gbl=1
  prt=yes
  nux=2
  nuy=2
  nuz=2
  fill
' z = 1
    1 1
    1 1
' z = 2
    1 1
    1 1
  end fill
end array
end data
end
```

INPUT FOR SECTION 4.6.2

```
=csas5 parm=(   )
primer - disk stack
ce_v7.1
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 1 10  2.5 0
 cuboid   0 1 15 -15  15 -15  2.5 0
unit 2
 cylinder 2 1 10  4 0
 cuboid   0 1 15 -15  15 -15  4 0
unit 3
   cuboid 3 1 15.0 -15.0 15.0 -15.0 3.0 0.0
global unit 4
   array 1 -15.0 -15.0 0.0
end geometry
read array
  ara=1
  prt=yes
  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
end array
end data
end
```

INPUT FOR SECTION 5.2.2

```
=csas5 parm=(   )
Primer Example: Hemispherical Tank
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
 ss304 2     end
end comp
```

```
read geometry
global unit 1
  hemisphere 0 1 19.5  chord -13.8
  hemisphere 1 1 19.5  chord 16
  hemisphere 2 1 19.622  chord 16.122
end geometry
end data
end
```

INPUT FOR SECTION 5.3.1

```
=csas5 parm=(   )
Primer example: Simple hole demo
ce_v7.1
read comp
 graphite 1 end
 u 2 den=18.7 1 293 92235 100 end
 ss316 3 end
end comp
read geometry
unit 1
  zcylinder 2 1  6.85  20.0 0
global unit 2
  cuboid 1 1 -2.0 -14.0 14.0 2.0 10.0 -10.0
  zcylinder 0 1 20.0 10.0 -10.0
  hole 1 6.0 -6.0 -10.0
  zcylinder 3 1 20.2 10.2 -10.2
end geometry
end data
end
```

INPUT FOR SECTION 5.4.1

```
=csas5 parm=(   )
Primer example: Pins in a tank
ce_v7.1
read comp
 u 1  1.0 293 92235 20 92238 80 end
 h2o 2 end
 al 3 end
end comp
read geometry
unit 1
 cylinder 1 1  0.375  30 0
 cuboid 2 1  0.5 -0.5  0.5 -0.5  30 0
unit 2
  array 2 -0.5 -4.0 0.0
unit 3
  array 3 -0.5 -5.0 0.0
unit 4
  array 4 -4.0 -0.5 0.0
unit 5
  array 5 -5.0 -0.5 0.0
global unit 6
  array 1 -6.0 -6.0 0.0
  zcylinder 2 1 9.0 45.0 0.0
```

```
  hole 3 -6.5 0.0 0.0
  hole 3 6.5 0.0 0.0
  hole 5 0.0 6.5 0.0
  hole 5 0.0 -6.5 0.0
  hole 2 7.5 0.0 0.0
  hole 2 -7.5 0.0 0.0
  hole 4 0.0 7.5 0.0
  hole 4 0.0 -7.5 0.0
  zcylinder 0 1 9.0 60.0 0.0
  zcylinder 3 1 9.5 60.0 -0.5
  zcylinder 2 1 39.5 60.0 -30.5
end geometry
read array
 ara=1 nux=12 nuy=12 nuz=1
  fill f1 end fill
 ara=2 nux=1 nuy=8 nuz=1
  fill f1 end fill
 ara=3 nux=1 nuy=10 nuz=1
  fill f1 end fill
 ara=4 nux=8 nuy=1 nuz=1
  fill f1 end fill
 ara=5 nux=10 nuy=1 nuz=1
  fill f1 end fill
end array
end data
end
```

INPUT FOR SECTION 6.2.1

```
=csas5 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 6.2.2

```
=csas5 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 6.2.3

```
=csas5 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 7.5.1

```
=csas5 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 geometry
unit 1
  zcylinder 1 1 0.47 365.76 0.0
  zcylinder 0 1 0.4875 372.76 0.0
  zcylinder 2 1 0.545 383.26 -4.0
  cuboid 3 1  0.82 -0.82  0.82 -0.82  383.26 -4.0
unit 2
  zcylinder 3 1 0.5 387.26 0.0
  zcylinder 2 1 0.575 387.26 0.0
  cuboid 3 1 0.82 -0.82 0.82 -0.82 387.26 0.0
global unit 3
  array 1 -7.38 -7.38 -193.63
  replicate 3 1  7.62 7.62 7.62 7.62 15.24 15.24 1
end geometry
read array
  ara=1
  prt=yes
  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 7.5.2

```
=csas5 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 geometry
global unit 1
  cuboid 1 1 2.5 -2.5 150.0 -150.0 75.0 -75.0
  replicate 2 1  0.5 0.5 0.5 0.5 0.5 0.5  1
  replicate 4 1  5.0 5.0 0 0 0 0  1
end geometry
read bounds
 -XB=REFL
end bounds
end data
end
```

INPUT FOR SECTION 7.5.3

```
=csas5 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 geometry
global unit 1
  sphere 10 1 2.5
  sphere  7 1 3.0
  cuboid  8 1 3.05 -3.05 3.05 -3.05 3.05 -3.05
end geometry
read bounds
  all=refl
end bounds
end data
end
```