OAK RIDGE
NATIONAL LABORATORY

MANAGED BY UT-BATTELLE
FOR THE DEPARTMENT OF ENERGY

KENO-VI Primer: A Primer for Criticality Calculations with SCALE/KENO-VI Using GeeWiz

September 2008

Prepared by
S. M. Bowman
SCALE Program Manager

UT-BATTELLE

Nuclear Science and Technology Division

KENO-VI PRIMER: A PRIMER FOR CRITICALITY CALCULATIONS WITH SCALE/KENO-VI USING GEEWIZ

S. M. Bowman

Date Published: September 2008

CONTENTS

iii

CONTENTS (continued)

CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

1-D	one-dimensional
238GROUPNDF5	ENDF/B-V 238-group library
2-D	two-dimensional
27GROUPNDF4	27-group ENDF/B-IV library
3-D	three-dimensional
44GROUPNDF5	ENDF/B-V 44-group library
AEC	U.S. Atomic Energy Commission
CSAS	Criticality Safety Analysis Sequences
CSEWG	Cross-Section Evaluation Working Group
EALF	energy of the average lethargy causing fission
GeeWiz	Graphically Enhanced Editing Wizard
GEN	number of generations
HEU	high-enriched uranium
ICSBEP	International Criticality Safety Benchmark Experiments Project
IEU	intermediate-enriched uranium
INFHOMMEDIUM	infinite homogeneous medium
JRE	JAVA runtime environment
k_{eff}	effective multiplication factor
KMART6	KENO6 Module for Activity-Reaction Rate Tabulation
LAW	Library to Analyze Radioactive Waste
LEU	low-enriched uranium
LWR	light-water reactor
MIPLIB	Material Information Processor Library
MOX	mixed oxide
NGEN	total number of generations
NITAWL	Nordheim Integral Treatment and Workline Library Production
NRC	U.S. Nuclear Regulatory Commission
NPG	neutrons per generation
NSK	number of generations skipped
ORNL	Oak Ridge National Laboratory
PFE	Programmer's File Editor
PMC	Produce Multigroup Cross sections
PWR	pressurized-water reactor
RSICC	Radiation Safety Information Computational Center
SCALE	Standardized Computer Analyses for Licensing Evaluations
SGGP	SCALE Generalized Geometry Package
V6-238GROUP	ENDF/B-VI 238-group library
V7-238GROUP	ENDF/B-VII 238-group library

ABSTRACT

The SCALE (**S**tandardized **C**omputer **A**nalyses for **L**icensing **E**valuation) computer software system developed at Oak Ridge National Laboratory is widely used and accepted around the world for criticality safety analyses. The well-known KENO-VI three-dimensional Monte Carlo criticality computer code is one of the primary criticality safety analysis tools in SCALE.   The KENO-VI primer is designed to help a new user understand and use the SCALE/KENO-VI Monte Carlo code for nuclear criticality safety 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-VI in particular. The primer is designed to teach by example, with each example illustrating two or three features of SCALE/KENO-VI that are useful in criticality analyses.

The primer is based on SCALE 6, which includes the **G**raphically **E**nhanced **E**diting **Wiz**ard (GeeWiz) Windows user interface.   Each example uses GeeWiz to provide the framework for preparing input data and viewing output results. Starting with a *Quickstart* section, the primer gives an overview of the basic requirements for SCALE/KENO-VI input and allows the user to quickly run a simple criticality problem with SCALE/KENO-VI.   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-VI 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 GeeWiz to set up criticality problems in SCALE/KENO-VI.

The primer provides a starting point for the criticality safety analyst who uses SCALE/KENO-VI.   Complete descriptions are provided in the SCALE/KENO-VI manual. Although the primer is self-contained, it is intended as a companion volume to the SCALE/KENO-VI documentation.   (The SCALE manual is provided on the SCALE installation DVD.)   The primer provides specific examples of using SCALE/KENO-VI for criticality analyses; the SCALE/KENO-VI manual provides information on the use of SCALE/KENO-VI and all its modules. The primer also contains an appendix with sample input files.

1. INTRODUCTION

This primer is designed to help you understand and use the KENO-VI three-dimensional (3-D) Monte Carlo code for nuclear criticality safety analyses. KENO-VI is part of the SCALE (**S**tandardized **C**omputer **A**nalyses for **L**icensing **E**valuation) code system.[1] It assumes that you have a college education in a technical field. There is no assumption of familiarity with Monte Carlo codes in general or with SCALE and KENO-VI in particular. The primer is designed to teach by example, with each example illustrating two or three features of KENO-VI that are useful in criticality safety analyses.

The primer is based on SCALE 6, which includes the SCALE **G**raphically **E**nhanced **E**diting **Wiz**ard (**GeeWiz**), a Windows program designed to assist criticality safety analysts in creating and executing SCALE input files for KENO-VI. SCALE 6 pre-release versions of GeeWiz and KENO-VI were used the examples shown in this primer. Each of the examples uses GeeWiz to provide the framework for data input. Starting with a *Quickstart* section, the primer gives an overview of the basic requirements for SCALE/KENO-VI input and allows you to quickly run a simple criticality problem with SCALE/KENO-VI. This section is not designed to explain GeeWiz, the input, or the SCALE/KENO-VI options in detail; rather, it introduces the GeeWiz user interface and some basic concepts that are further explained in following sections. Each following section has a list of basic objectives at the beginning that identifies the goal of the section and the individual SCALE/KENO-VI features, which are then covered in detail in the example problems in that section. It is expected that on completion of the primer, you will be comfortable using GeeWiz to set up criticality problems in SCALE/KENO-VI. You will be able to utilize SCALE/KENO-VI in criticality calculations and will be capable of handling most situations that normally arise in a facility. The primer provides a set of basic input files that you can selectively modify to fit the particular problem at hand.

Although much of the information needed to perform an analysis is provided in the primer, there is no substitute for understanding your problem and the theory of neutron interactions. The SCALE/KENO-VI code is capable only of analyzing the problem as it is specified; it will not necessarily identify inaccurate modeling of the geometry, nor will it know when the wrong material has been specified. However, the two-dimensional (2-D) color geometry plots and the external KENO3D visualization package are quite useful for identifying geometry errors. Remember that a single calculation of k_{eff} and its associated confidence interval with SCALE/KENO-VI or any other code is meaningless without an understanding of the context of the problem, the quality of the solution, and a reasonable idea of what the result should be.

The primer provides a starting point for the criticality safety analyst using SCALE/KENO-VI. Complete descriptions are provided in the SCALE manual.[1] Although the primer is self-contained, it is intended as a companion volume to the SCALE manual. (The SCALE manual is provided on the SCALE installation DVD.) The primer provides specific examples using SCALE/KENO-VI for criticality safety analyses, while the documentation provides information on the use of SCALE and all its modules. The primer also contains an appendix with sample input files.

To make the primer easy to use, there is a standard set of notation that you need to know. The text of this primer is set in Times New Roman font. Information that you type into an input file (or provide to GeeWiz) 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 on the screen by GeeWiz are highlighted in **bold font**. Because the primer often references the SCALE manual, these references will be set in square brackets; e.g., [see SCALE Manual Section x].

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

1.1 OVERVIEW OF SCALE

The SCALE code system dates back to 1969, when the current Nuclear Science and Technology Division at Oak Ridge National Laboratory (ORNL) began providing computational support in the use of the new KENO code to the U.S. Atomic Energy Commission (AEC) staff. From 1969 to 1976 the AEC certification staff relied on the ORNL staff to assist them in the correct use of codes and data for criticality, shielding, and heat transfer analyses of transportation packages. Shortly after creation of the U.S. Nuclear Regulatory Commission (NRC) from the AEC, the NRC staff proposed the development of an easy-to-use analysis system that would provide the technical capabilities of the individual modules with which they were familiar. With this proposal, the concept of the SCALE code system was born.

The NRC staff provided ORNL with some 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 initial version of SCALE was released by the Radiation Safety Information Computational Center (RSICC) at ORNL in 1980. Since that time, the system has been considerably enhanced, and the release used in this primer is Version 6. SCALE runs on Unix, Windows, Linux, and Intel Mac computer platforms.

The concept of SCALE was 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.

At the center of the **C**riticality **S**afety **A**nalysis **S**equences (CSAS) is the library of subroutines referred to as the **M**aterial **I**nformation **P**rocessor **Lib**rary or MIPLIB. The purpose of MIPLIB is to allow 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. MIPLIB also uses other keywords and simple geometry input specifications to prepare input for the modules that perform the problem-dependent multigroup cross-section processing: BONAMI, NITAWL, CENTRM, PMC, and XSDRNPM. 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.

The modular structure of SCALE allows back-to-back execution of the functional modules to perform a system analysis. However, a variety of control modules such as CSAS have been developed that automate and standardize various analytic sequences. The input format of the control module has been designed to help minimize input errors. Upon processing the user-specified input, the SCALE system control modules immediately print an input checklist in which the user (or reviewer) can easily establish that the input describes the system to be analyzed.

The CSAS6 control module is the criticality safety control module used for the calculation of the neutron multiplication factor of a system via KENO-VI.

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. As output, BONAMI produces a problem-dependent master library.

NITAWL uses the Nordheim Integral Treatment to perform neutron cross-section processing in the resolved resonance energy range. This involves a fine-energy-group calculation of the slowing-down flux across each resonance, with subsequent flux weighting of the resonance cross sections. NITAWL also assembles group-to-group transfer arrays from the elastic and inelastic scattering components and performs other tasks in producing the problem-dependent working library. NITAWL is compatible only with ENDF/B-V cross-section libraries in SCALE.

CENTRM computes continuous-energy neutron spectra using one-dimensional (1-D) discrete ordinates or infinite media geometry. CENTRM determines the problem-specific fluxes for processing resonance-shielded multigroup data using a 1-D unit cell. PMC reads the CENTRM continuous-energy flux spectra and cross-section data; calculates problem-dependent, group-averaged cross sections over some specified energy range; and then replaces the corresponding data in a problem-dependent AMPX master library for use by XSDRNPM and/or KENO-VI.

XSDRNPM is a 1-D discrete ordinates transport code for performing neutron or coupled-neutron-gamma calculations. The code has a variety of uses within SCALE. In the CSAS6 sequence, it is optionally used for preparation of cell-averaged cross sections for subsequent system analysis in KENO-VI.

Note that SCALE includes two versions of KENO, known as KENO-V.a and KENO-VI. Both versions are 3-D multigroup Monte Carlo codes employed to determine effective multiplication factors (k_{eff}) for multidimensional systems. KENO V.a has been part of SCALE for more than 20 years. Its simplified geometry system makes it very easy to use and much more computationally efficient than other Monte Carlo criticality safety codes. The geometry models consist of units that are constructed by nesting partial or whole spheres, cylinders, and rectangular cuboids oriented along X-, Y-, and Z-axes. These units can be combined in rectangular arrays or inserted in other units. These restrictions allow for very fast particle tracking, but still permit users to construct complex geometry models.

KENO-VI was introduced in 1995 in SCALE 4.3 to provide SCALE users with a more general geometry system known as the SCALE Generalized Geometry Package (SGGP) for constructing more-complex geometry models. KENO-VI has a much larger assortment of bodies, including the following:

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

In addition to rectangular arrays, KENO-VI provides the following array types: dodecahedral, (a 3-D stack of dodecahedrons) and three hexagonal array types as shown in Fig. 1 through Fig. 3.
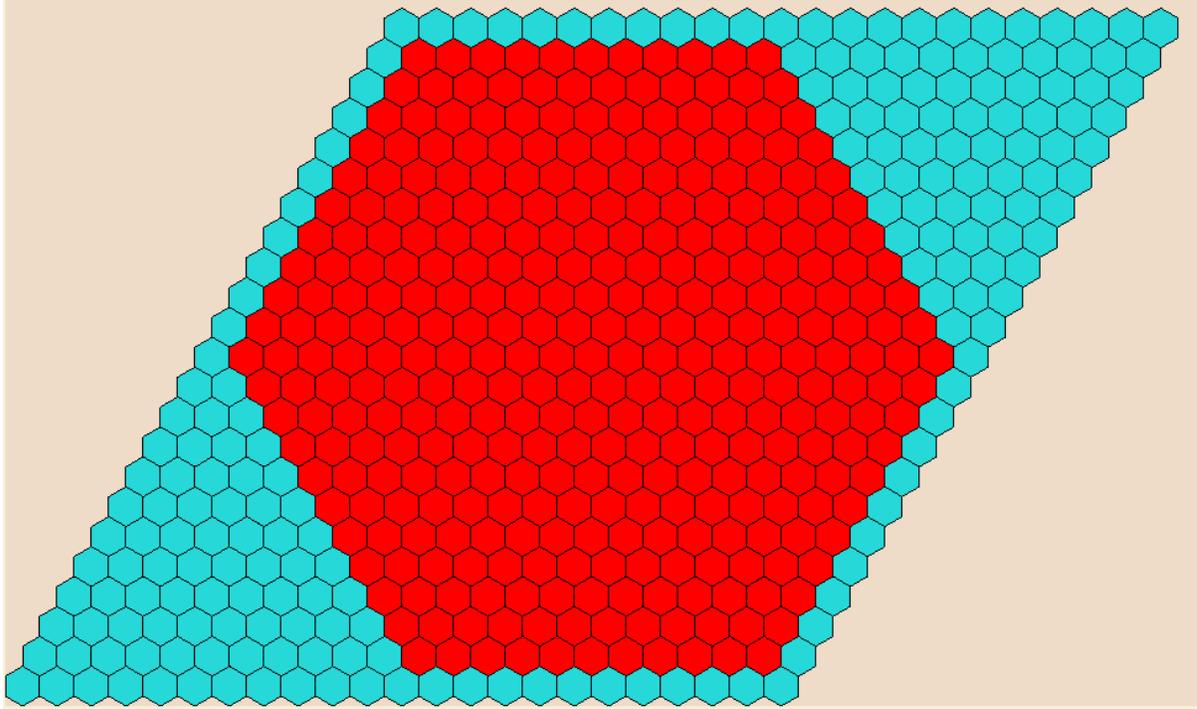
Fig. 1. Hexprism (HEXAGONAL) array.
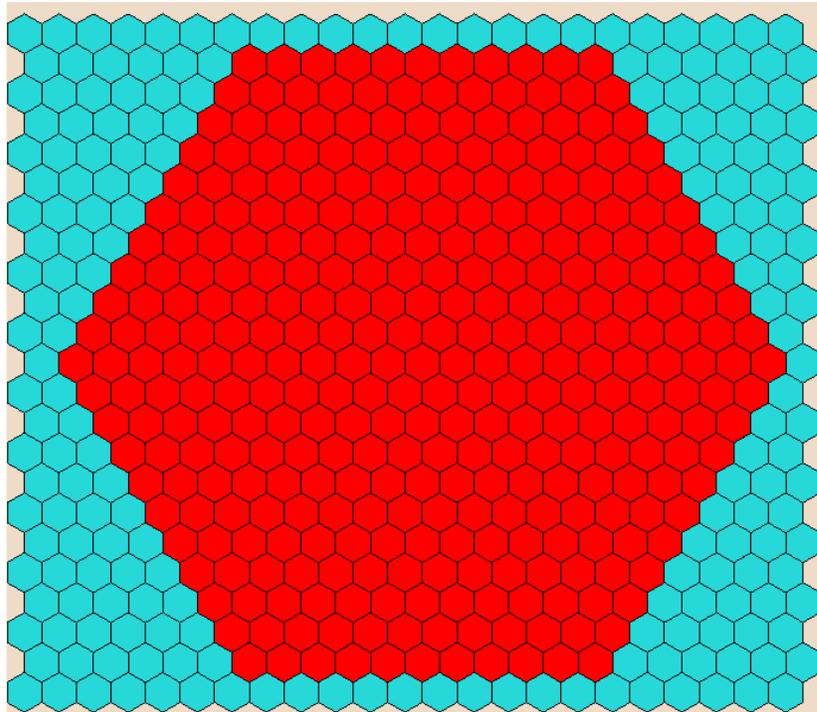
Fig. 2. Standard hexprism (SHEXAGONAL) array.
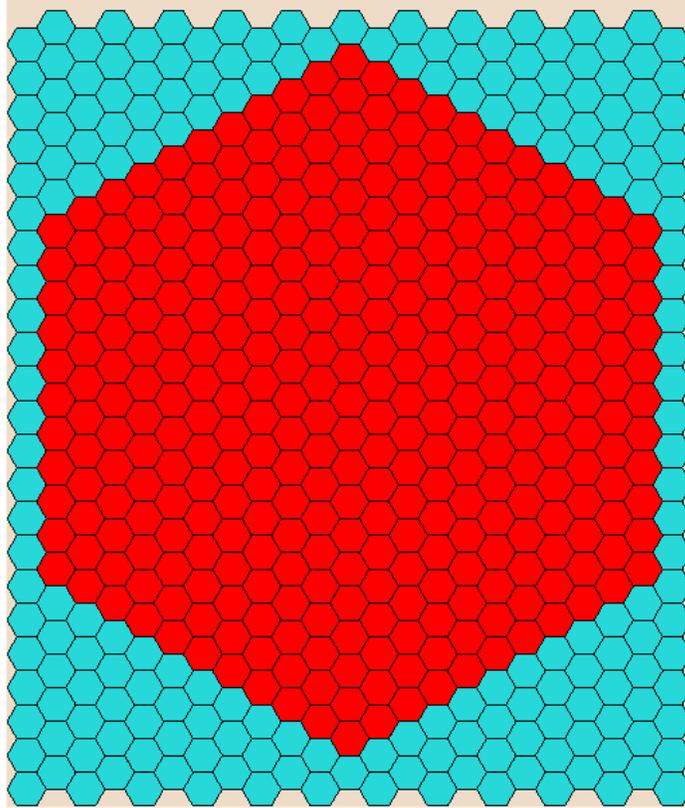
Fig. 3. Rotated hexprism (RHEXAGONAL) array.

The added complexity of the KENO-VI generalized geometry results in longer running times than those for KENO V.a. In general, a KENO V.a model that has been converted to KENO-VI will require approximately 4 times the amount of CPU time in KENO-VI as it did in KENO V.a. The KENO-VI run times are comparable to those for other generalized geometry Monte Carlo codes. Color 2-D plots of geometry models can be generated in both versions of KENO, or the models may be viewed using the 3-D visualization tool KENO3D.

The SCALE code system includes several problem-independent multigroup cross-section libraries for criticality analyses. The most up-to-date library is the 238-group ENDF/B-VII neutron library. The 238-group ENDF/B-VI and ENDF/B-V libraries are also available in SCALE 6.

The SCALE 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 U.S. and international criticality safety communities as the primary computational tool or as a backup/review tool.

2. SCALE/KENO-VI *Quickstart*

2.1 WHAT YOU WILL BE ABLE TO DO

- Describe the structure of SCALE/KENO-VI input files.

- Use the GeeWiz user interface to create a SCALE/KENO-VI input file.

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

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

2.2 SCALE/KENO-VI INPUT FILE

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

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

2.3 SIMPLE SAMPLE PROBLEM

This section should provide you enough information to run a simple sample problem. It is our intent that you immediately gain confidence in using GeeWiz to enter SCALE/KENO-VI input data, so that we can walk through this sample problem step-by-step, explaining each input step. For the present, it is important that you enter this problem exactly as we describe it. As you gain more experience with GeeWiz and SCALE/KENO-VI, you may find other ways to set up input files that are more logical to you. For example, you may find alternate methods for the geometry setup.

2.3.1 Problem Description

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

Delta-phase Pu metal sphere: radius = 6.38493 cm

$$N_{239} = \text{Atom density of }^{239}\text{Pu} \quad = 3.7047\text{E}{-}2 \text{ atoms/b-cm}$$

$$N_{240} = \text{Atom density of }^{240}\text{Pu} \quad = 1.751\text{E}{-}3 \text{ atoms/b-cm}$$

$$N_{241} = \text{Atom density of }^{241}\text{Pu} \quad = 1.17\text{E}{-}4 \text{ atoms/b-cm}$$

$$N_{Ga} = \text{Atom density of Ga} \quad = 1.375\text{E}{-}3 \text{ atoms/b-cm}$$

Spherical nickel coating: thickness = 0.0127 cm

$$N_{Ni} = \text{Atom density of Ni} \quad = 9.1322\text{E}{-}2 \text{ atoms/b-cm}$$

2.3.2 GeeWiz Input—General Information

Now you are ready to begin entering the sample problem. First start the GeeWiz user interface. You should see a screen that looks like Fig. 4.
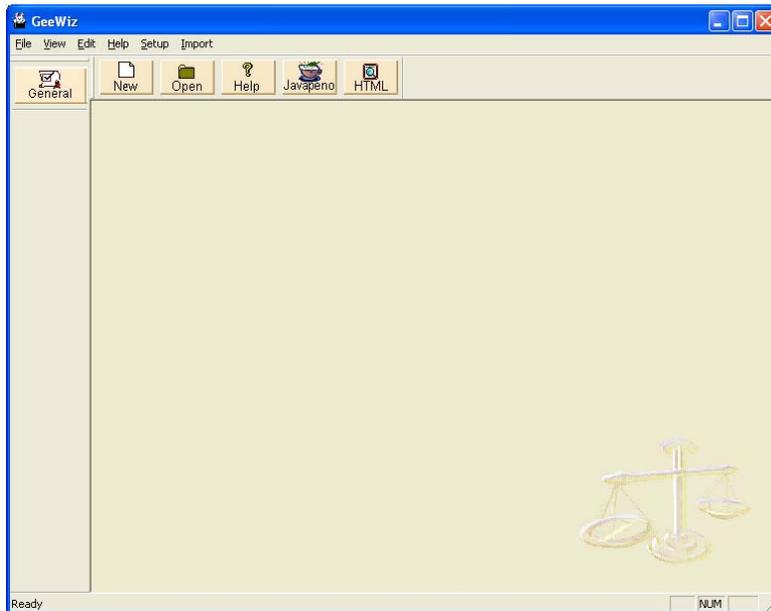
Fig. 4. GeeWiz start screen.

Clicking on the **General** button on the left-hand-side toolbar will bring up a window as shown in Fig. 5. In this window, you enter the **Title**, the SCALE **Application** and **Sequence**, the program to **Use for cross-section processing**, and the **Cross-section Library**. Also included are optional batch arguments. We will not use these optional arguments in this primer.
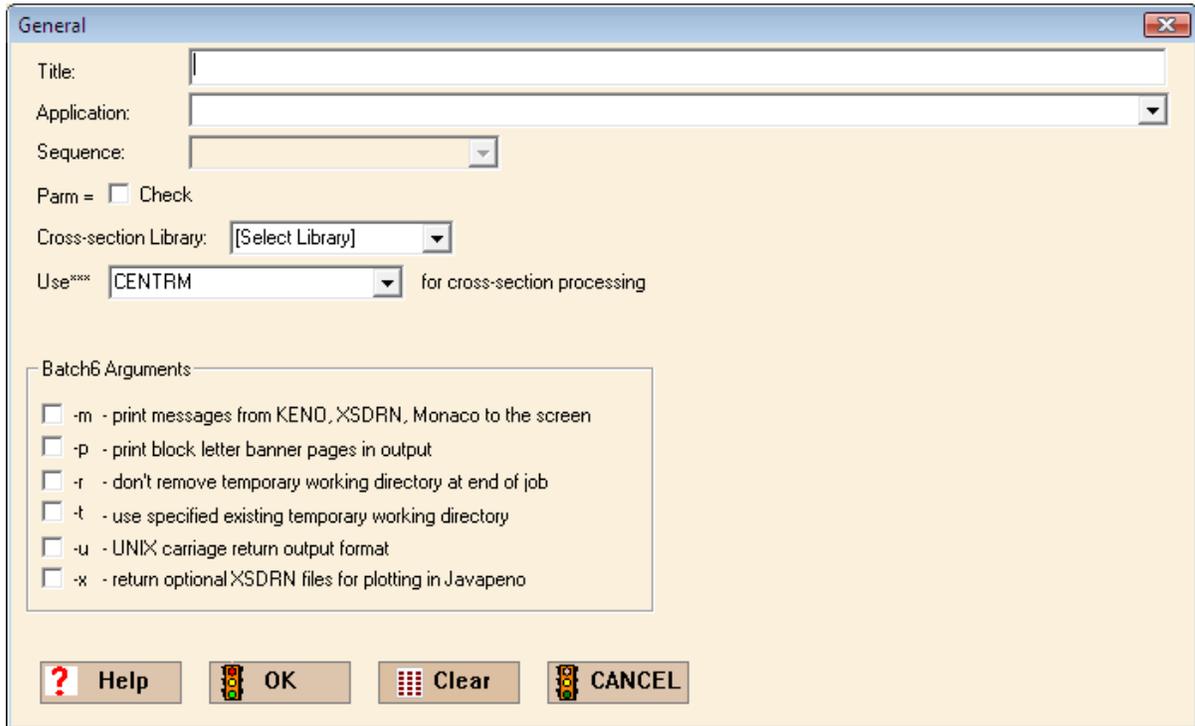
Fig. 5. General form.

Enter the **Title** as follows:

```
jezebel
```

Again note that information to be entered into GeeWiz will appear in the `Courier` `font`. For **Application** click on the down arrow and select `KENOVI - csas6`. Then `csas6` is automatically specified as the **Sequence**, because it is the only KENO-VI sequence in SCALE. Click the `Parm = Check` box to request that CSAS 6 perform an input check only, i.e., no calculations will be performed. This option is recommended when initially setting up a problem and should be used for each problem in this primer. For **Cross-section Library**, select `v7-238`. We will **Use** `CENTRM` **for cross-section processing**. The **General** screen should look like that in Fig. 6.
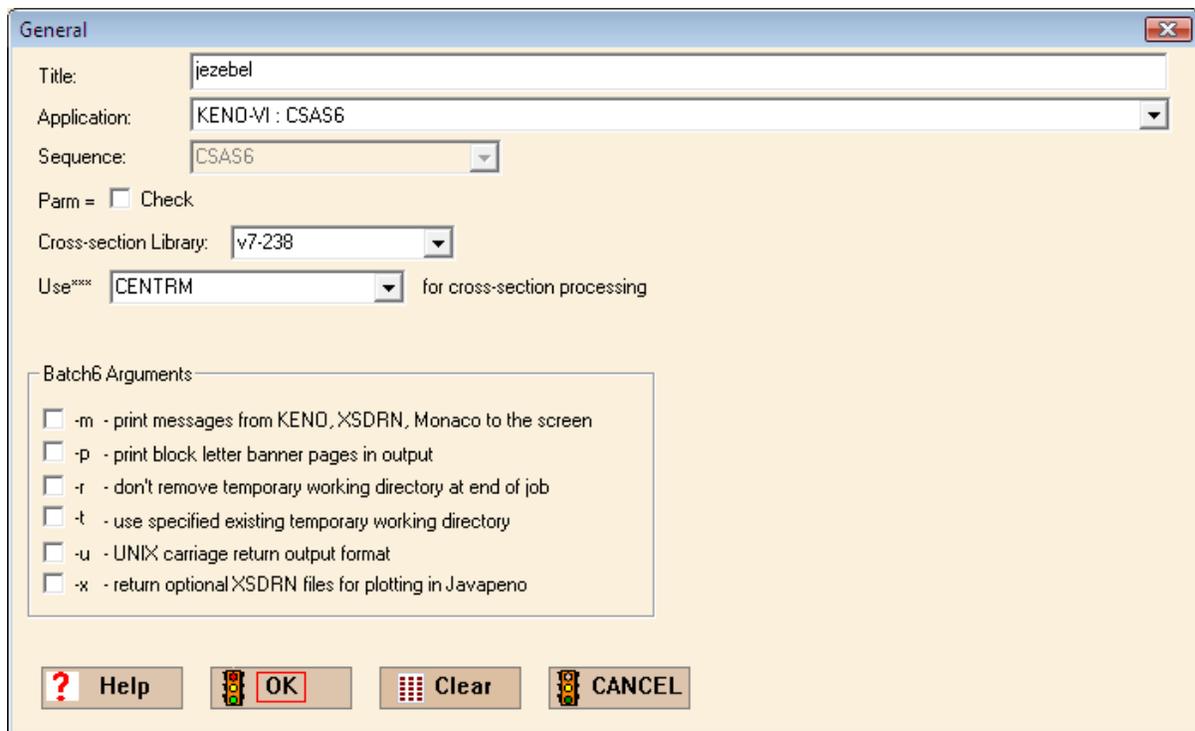
Fig. 6. General information for simple sample problem.

Now select **OK** to save this information in a temporary file. Later you will save all of the input information under a file name.

2.3.3　Materials

The next section of GeeWiz input provides information on the materials in the problem. For the sample problem, there are four materials in the core region and then the nickel shell in the outer region. To enter these, select **Compositions** from the menu on the left-hand side of GeeWiz and then click the **Create** button on toolbar. This button opens the form for specifying new materials in the model, as shown in Fig. 7.
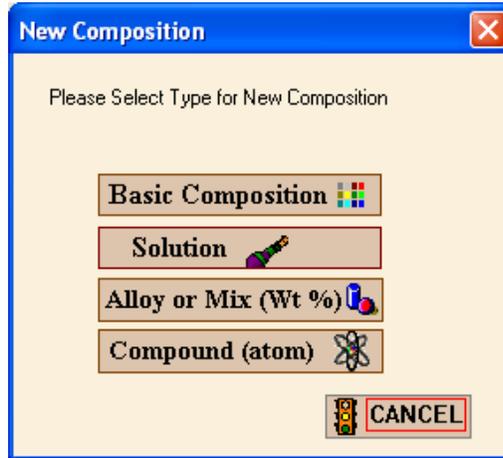
Fig. 7. New composition window.

You will be entering data for ^{239}Pu, ^{240}Pu, ^{241}Pu, gallium, and nickel, all of which are in the Standard Compositions Library and are Basic Compositions. Selecting the **Basic Composition** button will bring up the **Basic Standard Composition** form shown in Fig. 8.
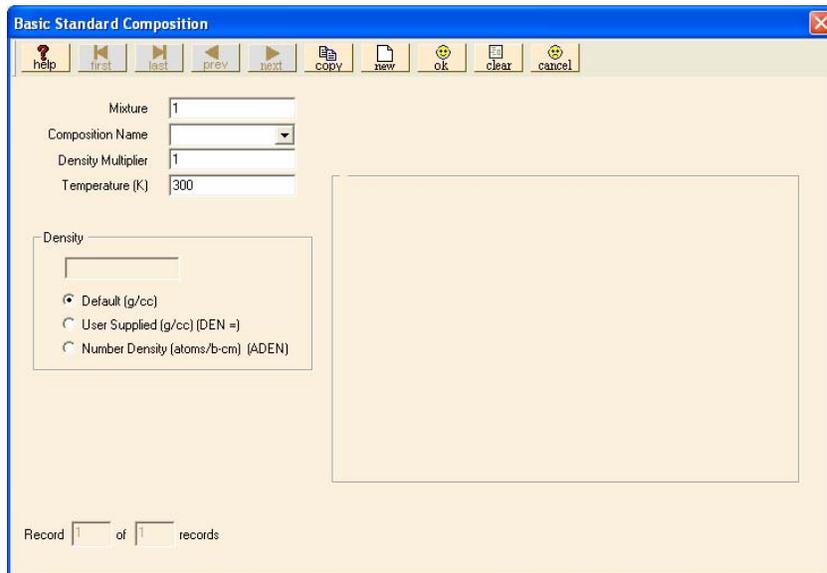
Fig. 8. Basic Standard Composition window.

Because the first four materials are mixed together, they will be part of mixture number 1. You will enter four compositions for mixture 1. The entry for the first material, `pu-239`, should look like that in Fig. 9. The **Mixture** number should already be set to `1`. Select `pu-239` as the **Composition Name** from the list of materials in the library. Set the **Temperature** to 293. To enter the atom densities that are given in the sample problem description, you need to select **Number Density** in the **Density** section and then enter the appropriate value. In this case, enter `3.7047e-2`.

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

To enter the data for the rest of the materials, select **copy** from the toolbar. Then modify the **Composition Name** and **Number 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`) should look like those shown in Fig. 10 through Fig. 12.
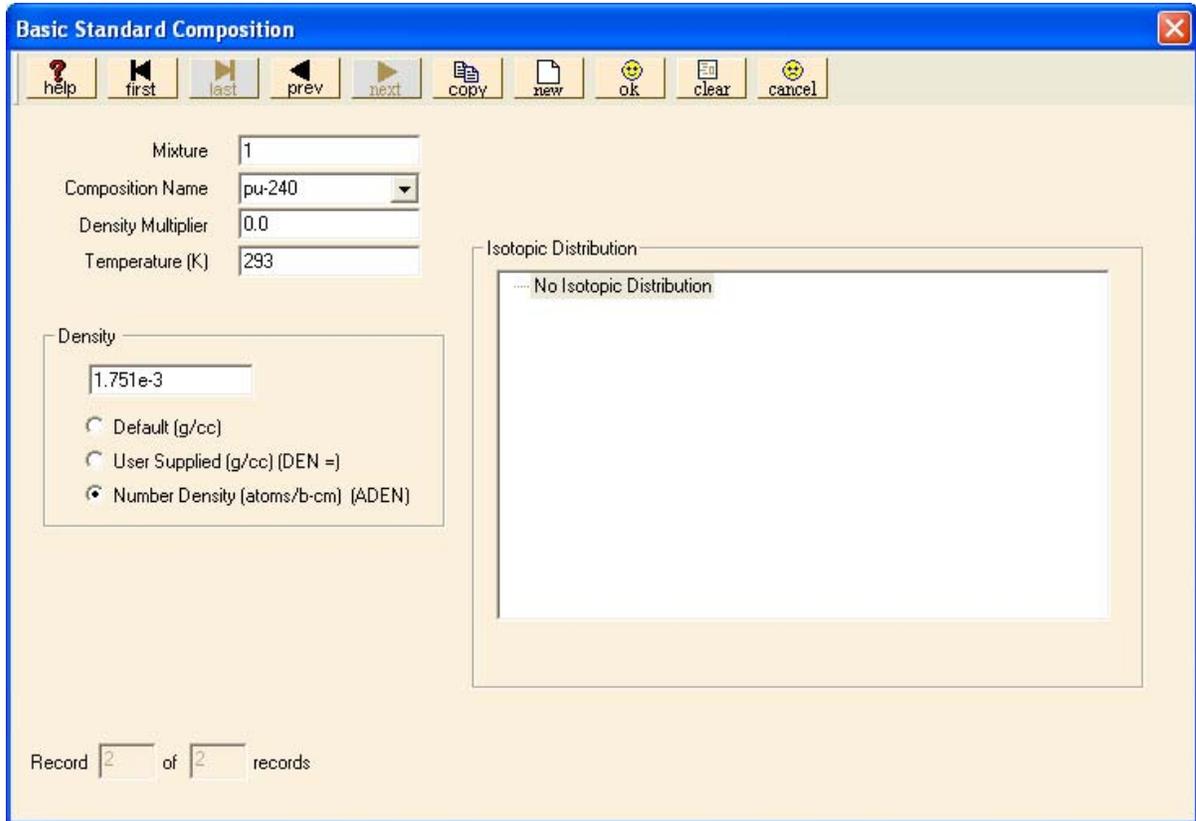
Fig. 10. Data for ^{240}Pu in mixture 1.

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

The last mixture 1 material is gallium; enter its data as shown in Fig. 12. Note that the notation in the lower left-hand corner indicates that it is Record 4 of 4 Records, so there are four materials that have been entered to this point. After entering the data for gallium, select **OK** to save all the material information for mixture 1.

Fig. 12. Data for gallium in mixture 1.

After clicking on **OK** from the toolbar, GeeWiz will bring up the **Standard Basic Compositions** summary screen as shown in Fig. 13. On this screen is the information for each of the four materials, the mixture number, and the atom densities. Now, to enter the information for the second mixture, which has only one material (nickel), select **Create** from the summary screen toolbar. This will bring up the **New Composition** window. Again select **Basic Composition** to bring up the **Basic Standard Composition** form. Note that GeeWiz has already updated the **Mixture** number to 2. Select `ni`; then select **Number Density** to allow you to enter `9.1322e-2.` Set the **Temperature** to 293. The completed screen for nickel should look like that shown in Fig. 14.
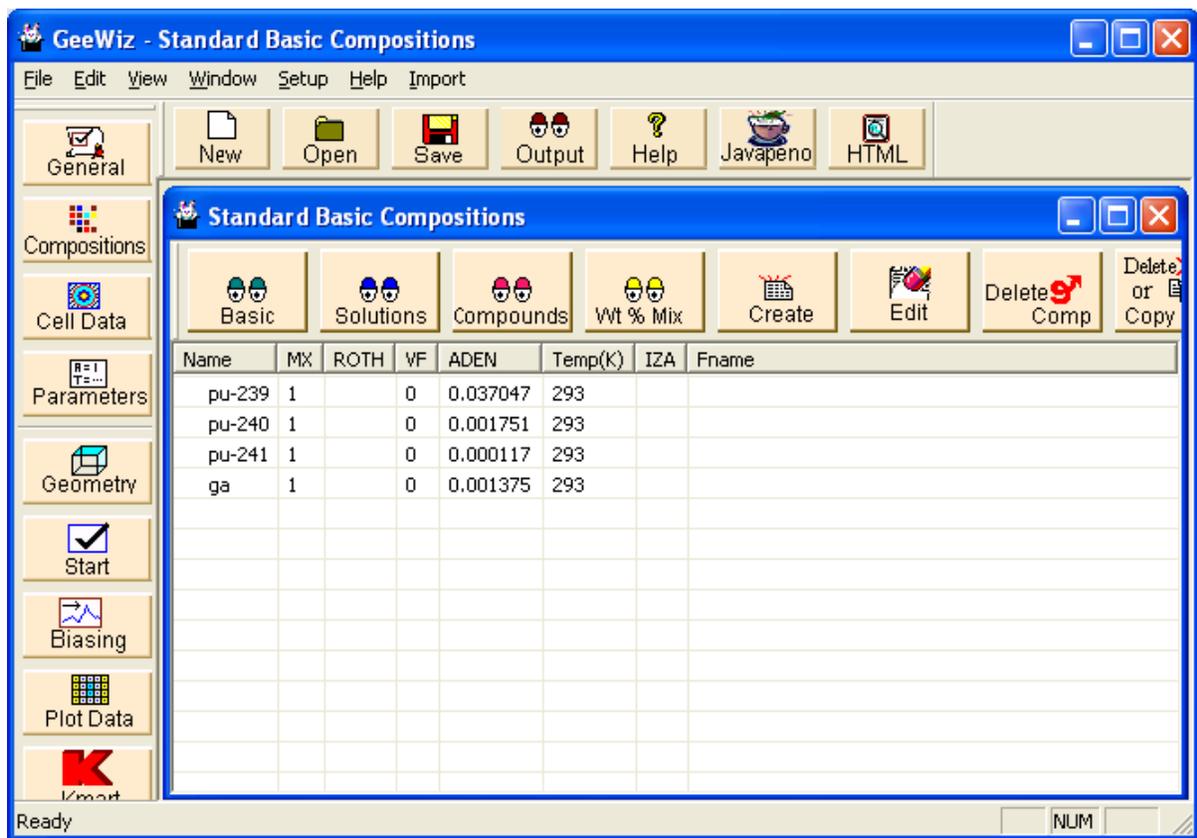
Fig. 13. Standard Basic Compositions summary.

Fig. 14. Data for nickel in mixture 2.

After entering the data for nickel, select **OK** to save all the material information for mixture 2. This brings you back to the composition summary screen with the five materials listed. Because we have no more material data, select **Close** on the summary screen. The next information to be entered is for the unit cell. To enter this information, select **Cell Data** from the toolbar on the left-hand side of GeeWiz. This button opens the form for specifying unit cell information. Now select the **MultiRegion** tab to get the summary form shown in Fig. 15.
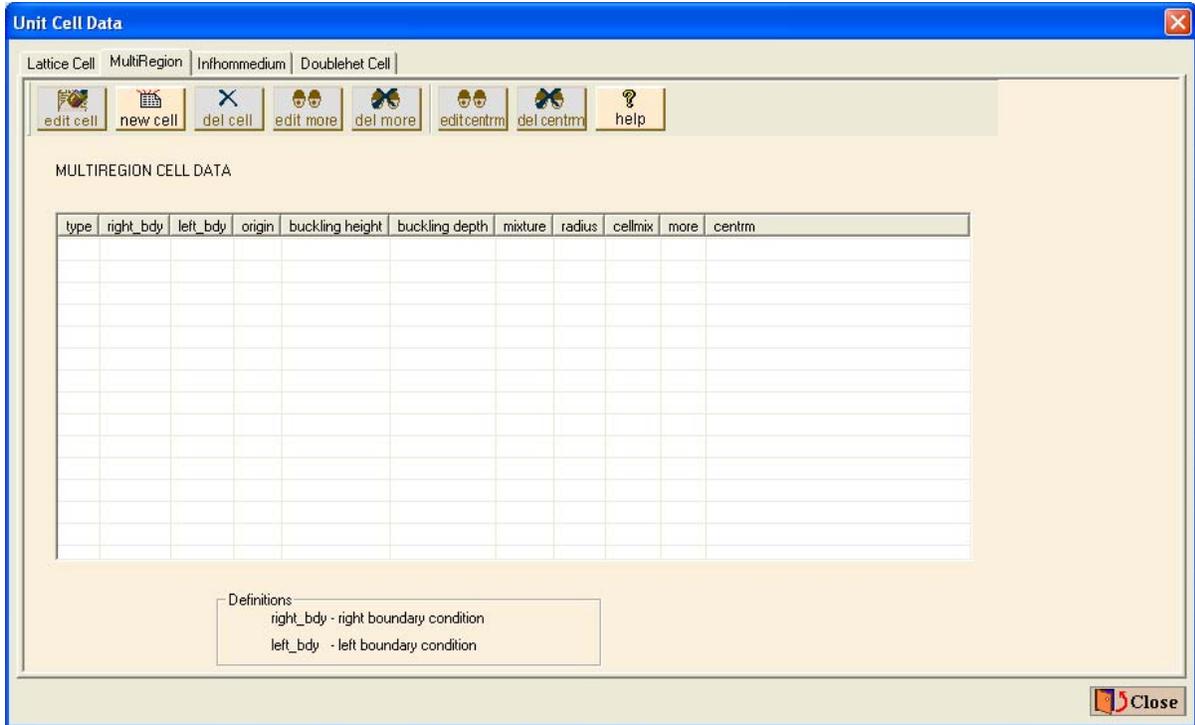
Fig. 15. Multiregion Cell Data summary.

To enter the cell data, select **new cell** from the toolbar. This opens the **Multiregion Cell Data** form.   The **Geometry Type** should be Spherical, the **Inner Boundary** defaults to Reflected, and the **Outer Boundary** defaults to Vacuum. The information should appear as shown in Fig. 16.

Fig. 16. Multiregion Cell Data form.

Now to enter the zone information, select **Insert**. This brings up the **Edit Zone** summary screen, where you can enter information for the first zone. The **Radius (cm)** should be `6.38493` cm, and the **Mixture** should be `1 pu-239,pu-240,pu-241,ga` as shown in Fig. 17. Then click **OK**.
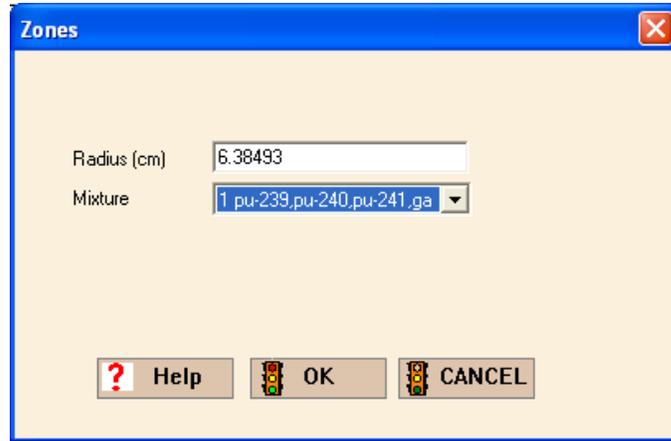
Fig. 17. Zone 1 information.

Now repeat the process for the second zone, with a **Radius** of `6.39763` cm and **Mixture** `2 ni`. Click **OK** to retain this information and return to the **Multiregion Cell Data** form, which should now look like Fig. 18.
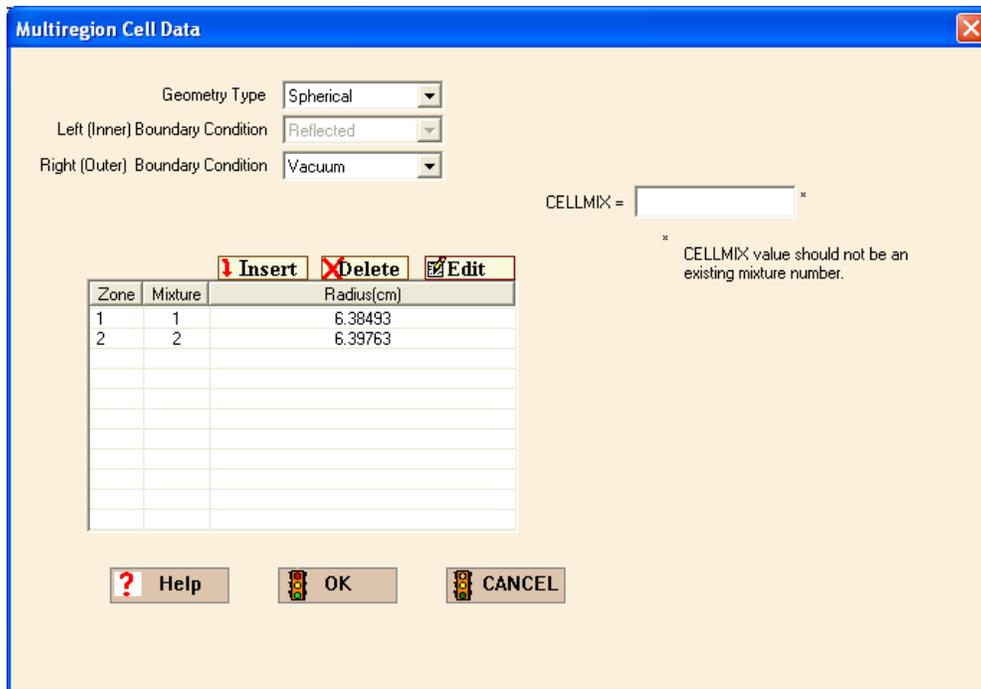
Fig. 18. Unit cell summary data with zone information.

Click **OK,** and GeeWiz will display a summary of the multiregion unit cell information. Click on the "+" next to "spherical" in the first line of the spreadsheet. The form should appear as shown in Fig. 19. Click **Close** to close the **Unit Cell Data** form. To save what we have completed thus far, click on the word **File** at the top of the main GeeWiz screen (shown in Fig. 13) and select **Save Partial File** from the pull-down menu. Then enter the file name `jezebel` and save in the folder of your choice.
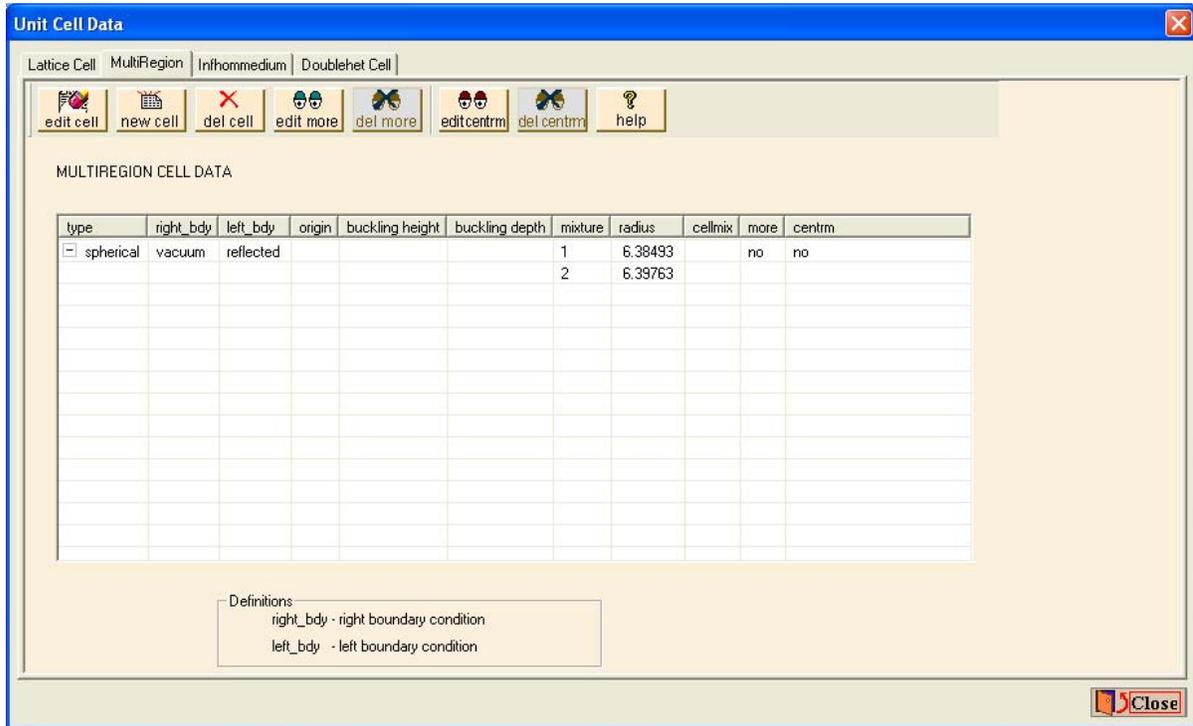
Fig. 19. Multiregion unit cell summary.

2.3.4 KENO-VI

Now we need to enter the information required to run a KENO-VI analysis of the problem. Select **Geometry** from the left toolbar to bring up the geometry summary form. First enter a **Comment** that describes the unit, `jezebel core with nickel plating`.

To enter geometry information, select the appropriate geometry from the right toolbar. For this problem, that would be **sphere**. This opens the **Sphere** form shown in Fig. 20.
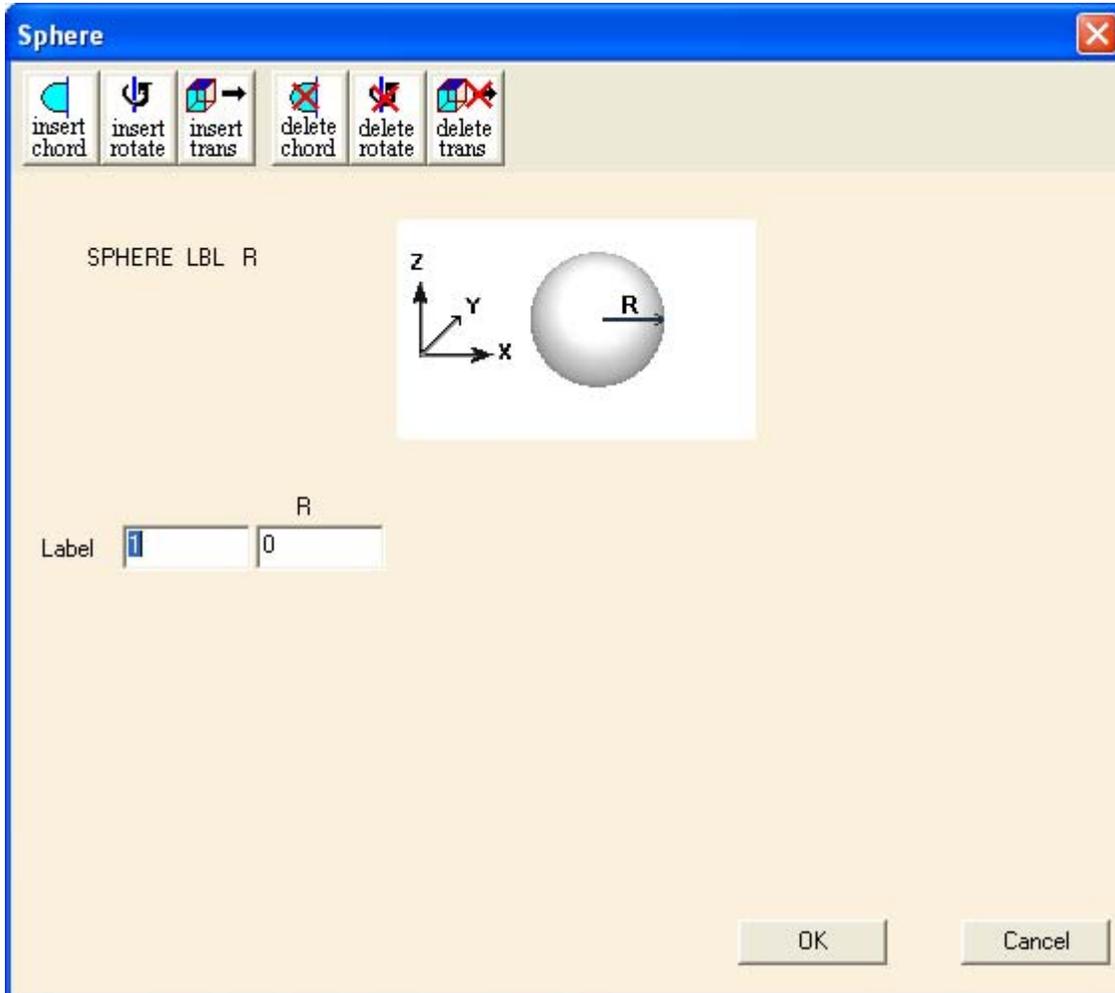
Fig. 20. Sphere form.

Leave the default **Label** value of 1 and enter the radius of `6.38493` for **R**. Click **OK**. To enter the information for the second region, again select the appropriate geometry from the right toolbar, **sphere**. This opens a new **Sphere** form with a **Label** of `2`. Enter the radius of `6.39763` for **R**, and then click **OK.**

Next we must describe the contents of the geometry regions. Click on the **Media** button on the right toolbar to open the Media form. To describe the plutonium core in sphere 1, click on `sphere 1 6.38493` under **Select Geometry**. Then click on the **Inside** button, and the value `1` appears in the box under **Region Definition Vector for Media** at the bottom of the form. Click on the "Select Mixture" pull-down button and select the appropriate mixture, `1 pu-239,pu-240,pu-241,ga`. The form should look like Fig. 21. Click **OK** to close the form.
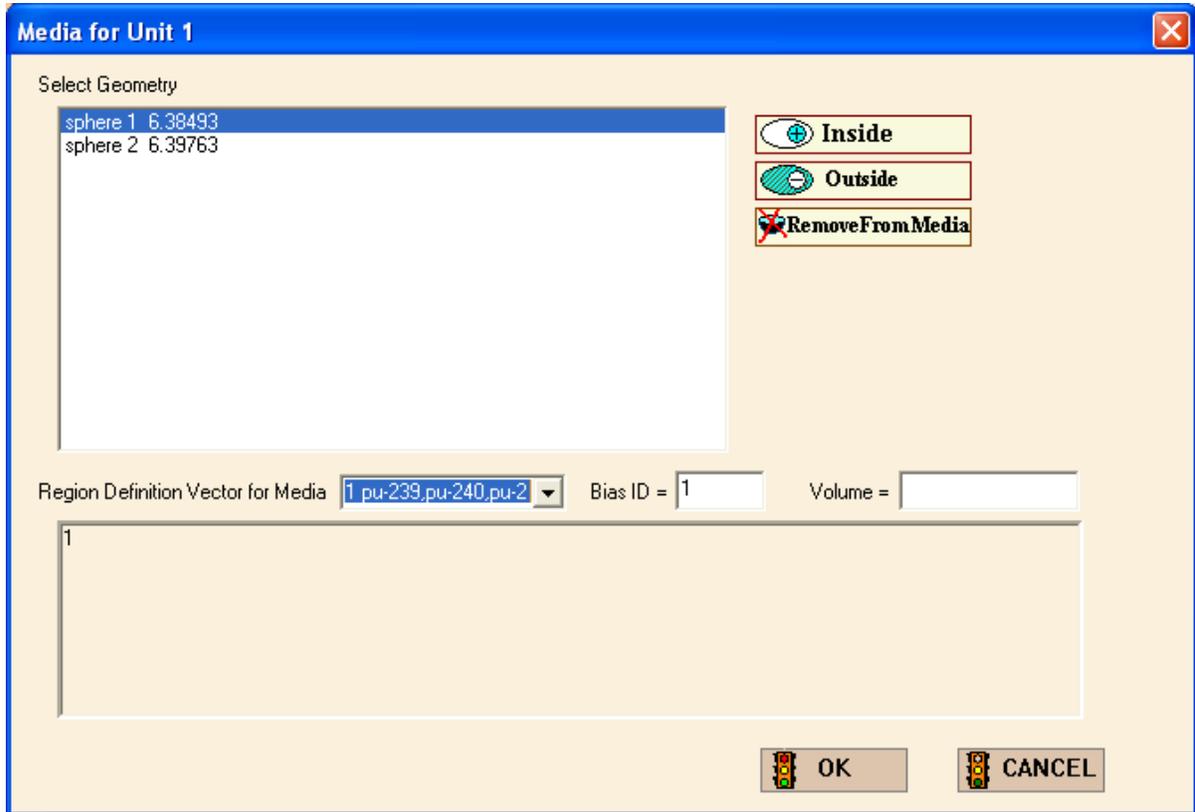
Fig. 21. Media form.

To describe the nickel plating outside the plutonium core, click on the **Media** button again. Click on `sphere 2 6.39763` under **Select Geometry**. Then click on the **Inside** button and the value `2` appears in the box under **Region Definition Vector for Media** at the bottom of the form. Next, click on `sphere 1 6.38493` and click on the **Outside** button. The box at the bottom of the form now contains the values `2 -1`. These values indicate that the media being defined is inside sphere 2 but outside sphere 1. Click on the "Select Mixture" pull-down button, and select `2 ni` as the media. Then click **OK** to close the form.

The outer boundary of the unit must be defined. Click on the **Bound** button to define the **Boundary Region Definition Vector**. This form is similar to the Media form. The outer boundary for the unit is sphere 2. Click on `sphere 2 6.39763` under **Select Geometry**.

Then click on the **Inside** button, and the value 2 appears in the box under **Boundary Region Definition Vector.** Click **OK** to close the form.

Now you need to check the box beside **Global Unit** by clicking on it. The global unit in KENO is the final unit that defines the entire system to be analyzed and must always be specified. The Geometry form should look like Fig. 22. Selecting **Close** on the toolbar will save this information and will return you to the main GeeWiz screen.
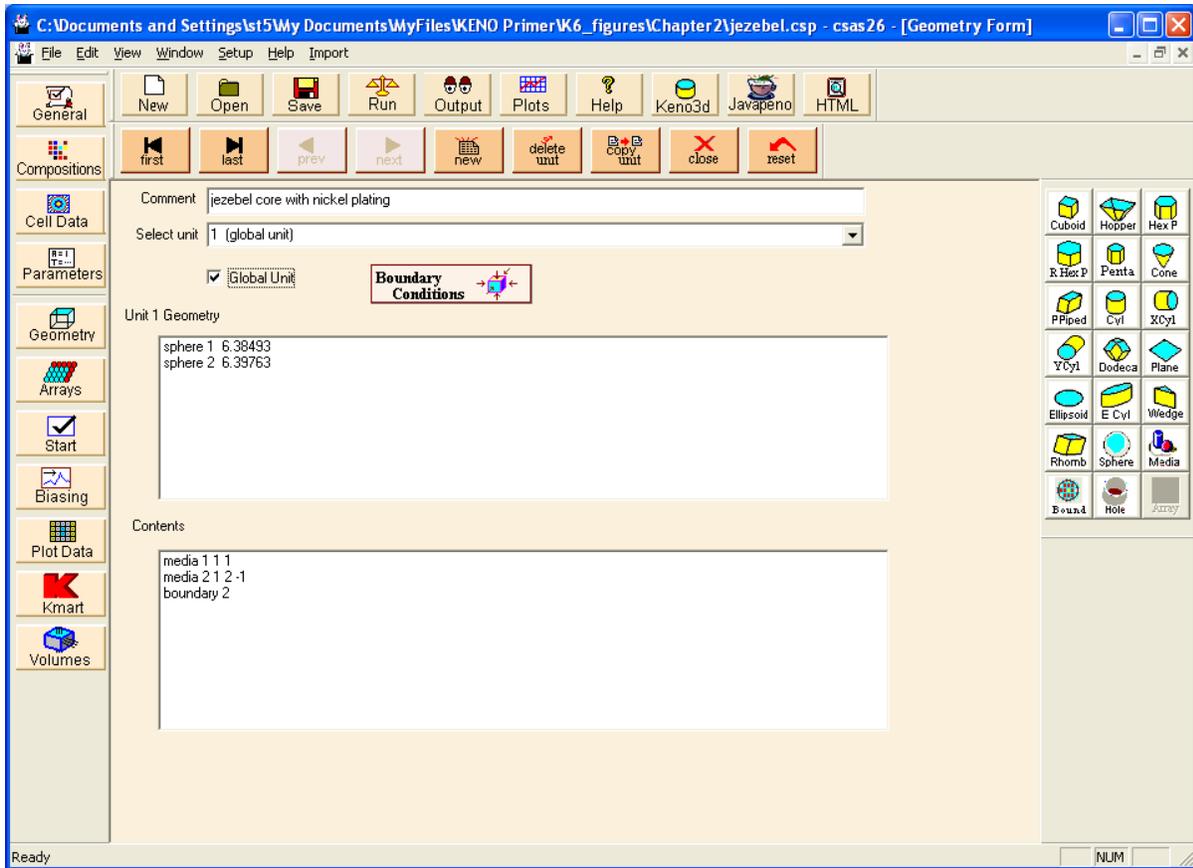
Fig. 22. Geometry form.

This completes the information necessary to run the sample problem. The next step is to save the input file and run SCALE. To do this, click on the **Save** button and enter the file name `jezebel.inp`. You can view the SCALE input file created by GeeWiz by clicking on **Edit** and then **Edit File Ctrl+E** or by using the shortcut keys Ctrl+E. Sample input files for all cases can be found in Appendix A.

2.4   RUNNING SCALE/KENO-VI

To run SCALE/KENO-VI, click the **Run** button on the top toolbar. GeeWiz opens a DOS window to execute SCALE. If you clicked the `Parm=Check` box on the General form, CSAS6 will perform an input check and terminate. Check the message at the end of the output file to see whether any errors were found. If there are no errors, uncheck `Parm=Check` box and click the **Run** button again.

The user can observe the progress of the calculation. The screen will show the input and output file names and then will indicate which programs are being executed. When KENO-VI is started, the title is displayed and then the calculated k_{eff} values from each generation are shown. At the completion of the run, the screen should look like that in Fig. 23. Note that your results may differ within 2 or 3 standard deviations of the results presented in this primer.
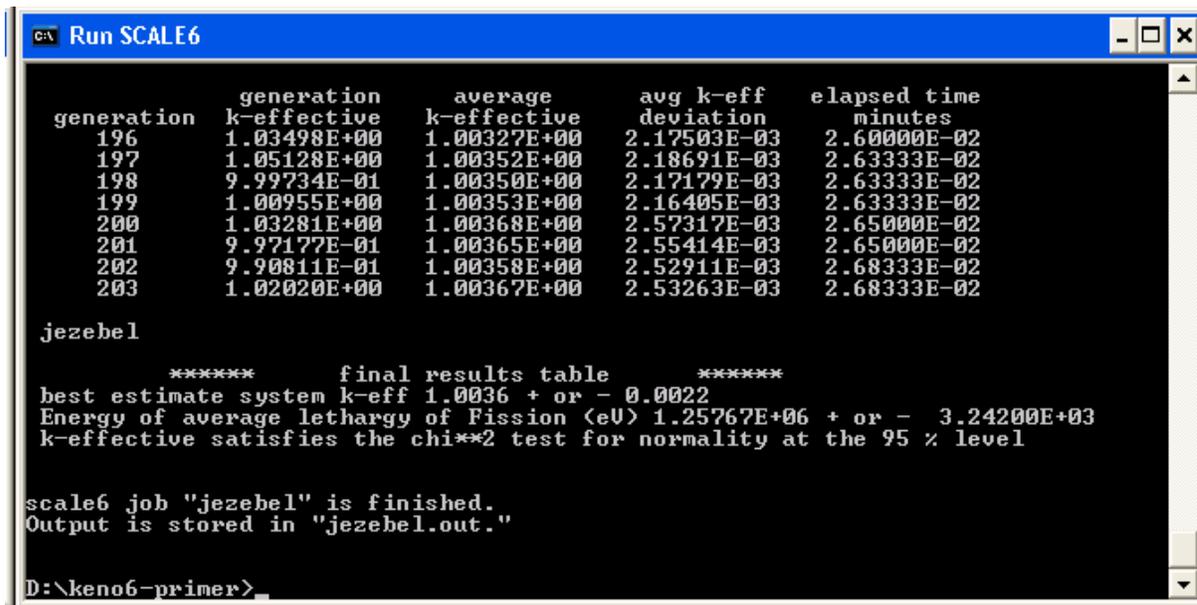
```
Run SCALE6                                                        _ □ ×

              generation      average       avg k-eff    elapsed time
 generation   k-effective    k-effective    deviation      minutes
    196       1.03498E+00    1.00327E+00    2.17503E-03    2.60000E-02
    197       1.05128E+00    1.00352E+00    2.18691E-03    2.63333E-02
    198       9.99734E-01    1.00350E+00    2.17179E-03    2.63333E-02
    199       1.00955E+00    1.00353E+00    2.16405E-03    2.63333E-02
    200       1.03281E+00    1.00368E+00    2.57317E-03    2.65000E-02
    201       9.97177E-01    1.00365E+00    2.55414E-03    2.65000E-02
    202       9.90811E-01    1.00358E+00    2.52911E-03    2.68333E-02
    203       1.02020E+00    1.00367E+00    2.53263E-03    2.68333E-02

jezebel

         ******      final results table       ******
best estimate system k-eff 1.0036 + or - 0.0022
Energy of average lethargy of Fission (eV) 1.25767E+06 + or -  3.24200E+03
k-effective satisfies the chi**2 test for normality at the 95 % level


scale6 job "jezebel" is finished.
Output is stored in "jezebel.out."


D:\keno6-primer>
```

Fig. 23. DOS window showing completed run.

Note that the screen shown in Fig. 23 indicates **SCALE 6 job jezebel is finished.** It also indicates that the **Output is stored in jezebel.out.** If you used a different name for your input file, then the output will be stored in a file of that name with the **.out** extension. The elapsed time in the DOS window may not match that shown in Fig. 23 as each computer has different clock speeds.

2.5   SCALE OUTPUT

To view the output, you can click on the **Output** button to view the ASCII text output file in the text editor specified in Setup (default is Programmer's File Editor [PFE]) or the **HTML** button to view the HTML-formatted output in your web browser. The text output file contains the following sections:

- echo of input;

- CSAS information and brief review of input values;

- cross-section processing and data handling output from CRAWDAD, BONAMI, WORKER, CENTRM, and PMC; and

- KENO-VI output data.

The HTML file contains only the KENO-VI output data. We will first demonstrate how to view the text output file in this section and then view the HTML output in the next section.

When you click on the Output button in GeeWiz, the initial display of the text output file contains input echo as shown in Fig. 24.

Fig. 24. Input echo at top of output file.

There are many pages in the output file. Each module has a section of output, starting with the CSAS6 control module, which is then followed by the functional modules BONAMI, WORKER, CENTRM, PMC, WORKER, and KENO-VI. For the moment, we will focus on portions of the KENO-VI output. After the KENO-VI banner page (search for "program:" to advance to the banner page for each module), there is information on run parameters, materials, and geometry. Next, there is a table of calculated k_{eff} values by generation along with elapsed time and average k_{eff}. These are the same data that appear in the DOS window during execution. At the end of this table, there should be the following message:

```
keno message number k6-123
execution terminated due to completion of the
specified number of generations.
```

This message indicates 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 to match use in table). Below the message are lines with information on the neutron lifetime, generation time, nubar, average fission group, and energy of the average lethargy causing fission (EALF). For our sample problem, this information should appear as shown in Fig. 25.

A table of calculated values for the system such as lifetime, average fission group, and EALF follows. This information allows you to check the spectrum of the modeled system. In this example, the spectrum should be quite hard as there is no moderator material in the system. This is indicated by the short lifetime of 2.47 nanoseconds and the relatively high energy, 1.26 MeV, of neutrons causing fission. The **lifetime** is the average lifespan of a neutron (in seconds) from the time it is born until it is absorbed or leaks from the system. The **generation time** is the average time (in seconds) between successive neutron generations. The average number of neutrons per fission, **nu bar,** and its associated deviation are printed and the **average fission group** (the average energy group at which fission occurs) and its associated deviation are printed. Then, the **energy (ev) of the average lethargy causing fission (EALF)** and its associated deviation are printed. The EALF value is frequently used as a gauge of the average neutron energy of the system (i.e., how fast or thermal the spectrum is).

Following the neutron parameters table is a table providing the calculated k_{eff} of the problem versus number of initial generations skipped. This edit prints the average k_{eff} and its associated deviation and the limits of k_{eff} for the 67, 95, and 99% confidence intervals (Fig. 25). The number of histories used in calculating the average k_{eff} is also printed. These results are based on skipping increasing numbers of initial generations. The user should carefully examine the final k_{eff} edit to determine if the average k_{eff} is relatively stable. If a noticeable drift is apparent as with an increase in the number of initial generations skipped, it may indicate a problem in source convergence. If this appears to be the case, the problem should be rerun with a better initial source distribution (i.e., increase the number of neutrons per generation, NPG) and should be run for a sufficient number of generations (GEN) that the average k_{eff} becomes stable.

```
                           restart data was written for generation   203        random number=BF0E1E69F9F4D3C8

  keno message number  k6-123              execution terminated due to completion of the specified number of generations.
                                           jezebel

lifetime =  2.47441E-09 + or -  9.82397E-12        generation time =  2.93003E-09 + or -  1.14369E-11
nu bar   =  3.15472E+00 + or -  6.27266E-04     average fission group =  2.13121E+01 + or -  2.83260E-02
                  energy(ev) of the average lethargy causing fission =  1.25767E+06 + or -  3.24200E+03
                                       system mean free path (cm) =  1.89158E+00 + or -  2.53117E-03

no. of initial                                                                                          de
 generations     average                      67 per cent          95 per cent          99 per cent     number of
   skipped     k-effective     deviation   confidence interval   confidence interval  confidence interval  histories

      3         1.00367    + or - 0.00253   1.00113 to 1.00620   0.99860 to 1.00873   0.99607 to 1.01126    200000

      4         1.00370    + or - 0.00254   1.00116 to 1.00625   0.99862 to 1.00879   0.99607 to 1.01134    199000

      5         1.00375    + or - 0.00256   1.00119 to 1.00632   0.99862 to 1.00888   0.99606 to 1.01144    198000

      6         1.00372    + or - 0.00258   1.00114 to 1.00630   0.99857 to 1.00888   0.99599 to 1.01145    197000

      7         1.00385    + or - 0.00227   1.00157 to 1.00612   0.99930 to 1.00839   0.99703 to 1.01066    196000

      8         1.00374    + or - 0.00229   1.00144 to 1.00603   0.99915 to 1.00832   0.99686 to 1.01061    195000

      9         1.00376    + or - 0.00231   1.00145 to 1.00607   0.99915 to 1.00838   0.99684 to 1.01068    194000

     10         1.00378    + or - 0.00232   1.00146 to 1.00610   0.99914 to 1.00842   0.99681 to 1.01074    193000

     11         1.00368    + or - 0.00234   1.00134 to 1.00602   0.99899 to 1.00836   0.99665 to 1.01070    192000

     12         1.00340    + or - 0.00237   1.00103 to 1.00576   0.99867 to 1.00813   0.99630 to 1.01049    191000

     17         1.00384    + or - 0.00227   1.00157 to 1.00612   0.99930 to 1.00839   0.99703 to 1.01066    186000

     22         1.00365    + or - 0.00233   1.00132 to 1.00597   0.99900 to 1.00830   0.99667 to 1.01063    181000

     27         1.00336    + or - 0.00237   1.00099 to 1.00573   0.99863 to 1.00810   0.99626 to 1.01047    176000

     32         1.00322    + or - 0.00243   1.00078 to 1.00565   0.99835 to 1.00808   0.99592 to 1.01052    171000

     37         1.00345    + or - 0.00243   1.00102 to 1.00588   0.99860 to 1.00831   0.99617 to 1.01074    166000

     42         1.00376    + or - 0.00261   1.00115 to 1.00636   0.99855 to 1.00897   0.99594 to 1.01157    161000
```

Fig. 25. KENO output for Jezebel problem.

28

Following the final keff edit are two plots of average keff: (1) versus the number of generations run and (2) versus the number of generations skipped. The limits of one standard deviation are plotted on either side of each average k_{eff}. In the first plot (Fig. 26), KENO omits the calculated k_{eff} values of the first NSK generations (number of generations skipped) in the calculation of the average k_{eff}. The dotted line represents the value of the average k_{eff} corresponding to the smallest deviation when the average k_{eff} and its deviation are computed for each generation over the range from NSK + 1 through the total number of generations (NGEN). Figure 26 is an example of this type of plot (showing the first 50 generations). This plot is a graphical representation of the first table of calculated k_{eff} values in the output. The primary use of this plot is to determine if the problem has source convergence difficulties. For this example, the source appears to be well converged, as is indicated by the stability of the average k_{eff} and by the minimum variance in k_{eff} near the end of the plot (not shown).
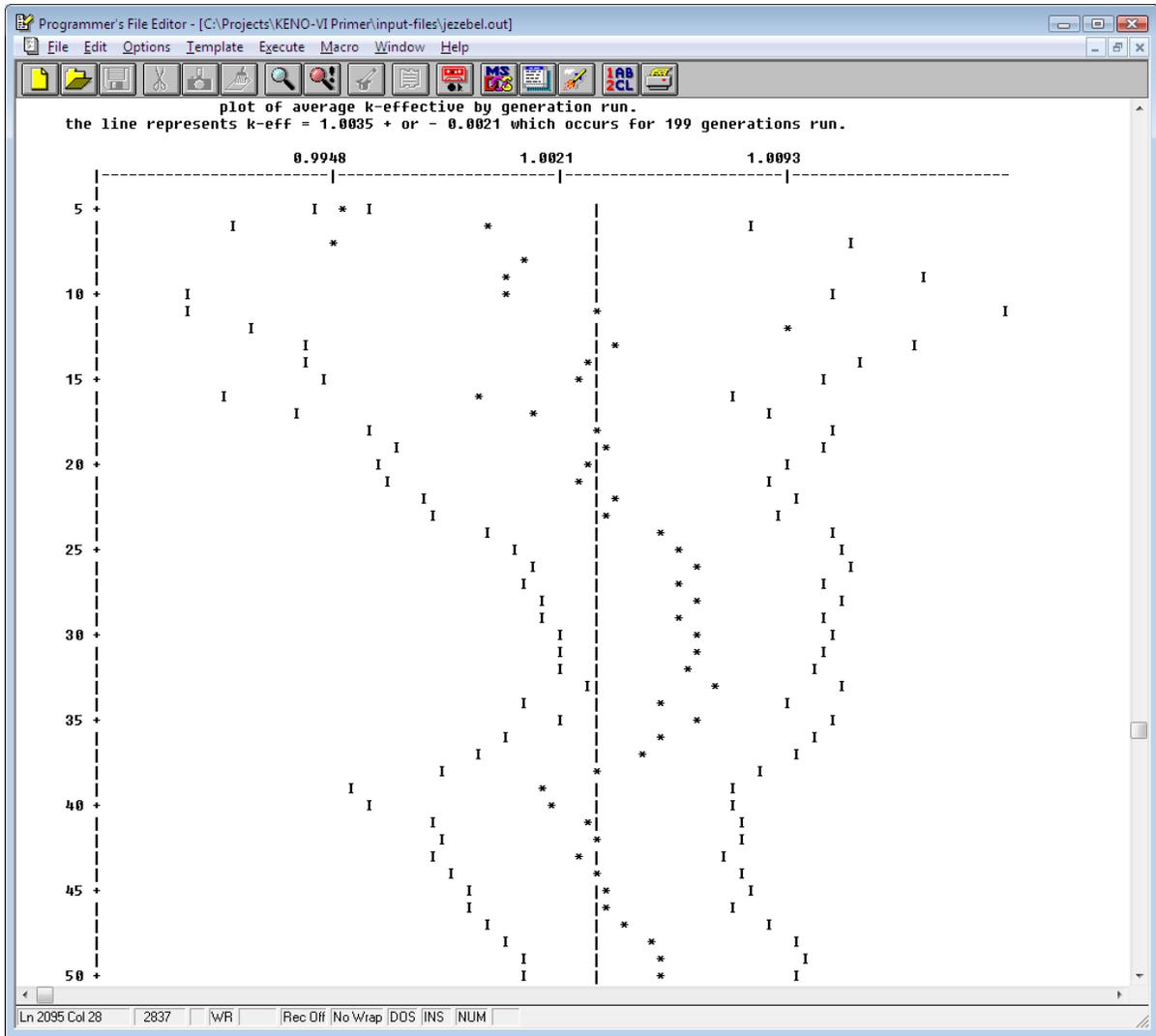
Fig. 26. Plot of average k_{eff} by generation run (beginning).

The second output plot graphically depicts the average calculated k_{eff} by generation skipped from the final k_{eff} edit for all generations from NSK + 1 to NGEN (Fig. 27). The dotted line represents the value of the average k_{eff} corresponding to the smallest deviation in the plot. Note the statement at the top of the plot, "this line represents k-eff = 1.003 + or – 0.002 which occurs for 13 generations skipped." The calculated k_{eff} value printed there corresponds to the number of generations skipped with the minimum standard deviation from the table in Fig. 25. An easy way to conceptualize the data in this plot is to think of each average k_{eff} as the average when the data for all previous generations are discarded. Because it typically takes somewhere between 3 and 100 generations to obtain a converged source distribution, this plot can help the analyst determine how many initial generations should be removed from the average k_{eff} calculation.

Fig. 27. Plot of average k_{eff} by generation skipped (beginning).

After the plots, there is a final edit of fissions, absorptions, and leakage by energy group. The fission fraction for each group and the fission production, absorptions, and leakage with their associated percent deviation are printed as shown in Fig. 28. (Energy groups 22 through 237 have been removed so that the totals at the end of the edit can be shown.) Totals are printed at the end of the table, and the sum of the leakage and absorptions printed for the system total should be approximately 1. For this example, 32.9% of the neutrons were absorbed while 67.2% leaked from the system. Note that no leakage is associated with faces having mirror or periodic reflection. Thus, there is no leakage associated with an infinite problem. The number of fissions printed for the system total should be the same as the first k_{eff} printed in the final k_{eff} edit. The total elapsed time and final random number are printed at the end of this edit.

```
Programmer's File Editor - [C:\Projects\KENO-VI Primer\input-files\jezebel.out *]
 File  Edit  Options  Template  Execute  Macro  Window  Help

group  fission   unit   region    fissions      percent      absorptions     percent        leakage        percent
       fraction                                  deviation                    deviation                     deviation

    1    0.0000            2.06745E-05     70.5328       3.73807E-06     70.5328       0.00000E+00       0.0000
    2    0.0001            1.47091E-04     33.6282       2.79765E-05     33.6282       1.26343E-05      53.3737
    3    0.0000            1.01433E-05    100.0000       1.99806E-06    100.0000       3.33082E-05      37.5197
    4    0.0001            6.31345E-05     48.1722       1.27437E-05     48.1722       3.06233E-05      36.9933
    5    0.0002            1.97009E-04     24.7166       4.07265E-05     24.7166       1.04023E-04      19.5688
    6    0.0029            2.90251E-03      6.6190       6.37268E-04      6.6304       1.10479E-03       6.9132
    7    0.0079            7.91377E-03      3.9701       1.88237E-03      3.9695       3.54683E-03       3.3658
    8    0.0219            2.19457E-02      2.0665       5.57208E-03      2.0668       1.06478E-02       1.8332
    9    0.0455            4.57065E-02      1.2856       1.24417E-02      1.2862       2.90178E-02       1.1229
   10    0.0250            2.50896E-02      1.7131       7.11860E-03      1.7138       1.63718E-02       1.6293
   11    0.1169            1.17344E-01      0.7549       3.47144E-02      0.7550       7.43216E-02       0.7082
   12    0.0790            7.93378E-02      0.9708       2.42997E-02      0.9708       4.83043E-02       0.9898
   13    0.0219            2.20295E-02      1.6701       6.84607E-03      1.6692       1.31193E-02       1.9336
   14    0.1069            1.07244E-01      0.7339       3.38281E-02      0.7338       6.53174E-02       0.7336
   15    0.0886            8.89483E-02      0.8096       2.86859E-02      0.8096       5.61105E-02       0.8625
   16    0.0279            2.79550E-02      1.5350       9.13944E-03      1.5349       1.77339E-02       1.5100
   17    0.0132            1.32876E-02      2.0811       4.36519E-03      2.0811       8.10186E-03       2.0648
   18    0.0113            1.12951E-02      2.2844       3.72116E-03      2.2844       7.31125E-03       2.5007
   19    0.0200            2.01221E-02      1.5584       6.65413E-03      1.5583       1.32625E-02       1.7190
   20    0.0151            1.51998E-02      1.9811       5.04721E-03      1.9810       1.01745E-02       2.0540
   21    0.0299            2.99726E-02      1.3832       1.00061E-02      1.3833       2.14048E-02       1.3913
    -       -        -      -              -             -               -             -                 -

                 (groups 22 - 227 have been removed from the figure for illustrative purposes
    -       -        -      -              -             -               -             -                 -
  238    0.0000            0.00000E+00      0.0000       0.00000E+00      0.0000       0.00000E+00       0.0000

system total =            1.00367E+00      0.1846       3.29023E-01      0.1845       6.71560E-01       0.0911

elapsed time    0.02967 minutes

random number=   C50DA561E057575B

Ln 2517 Col 26      2600  # WR      Rec Off No Wrap DOS INS  NUM
```

Fig. 28. Final edit of fissions, absorptions, and leakage (edited).

The last section of the output contains the frequency distributions that indicate the number of generations in which k_{eff} is within a specified interval. The intervals are determined by the code based on the upper and lower limits of the calculated k_{eff} values for all generations. One asterisk is printed for each generation-calculated k_{eff} value. The distributions should be reasonably symmetrical and bell shaped for a good statistical analysis of k_{eff} (Fig. 29).
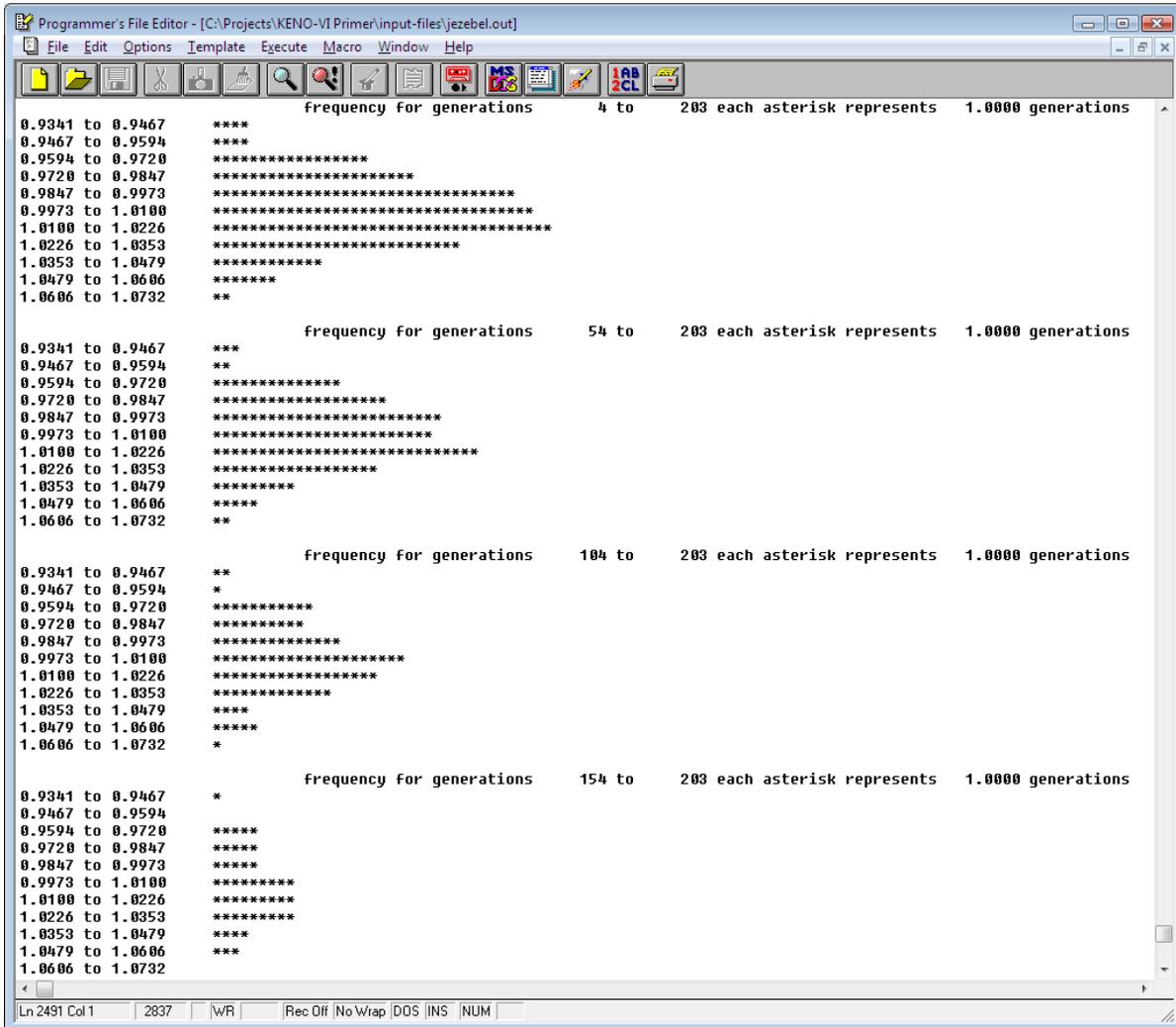
Fig. 29. Calculated k_{eff} by generation frequency plot.

At the end of the output file is the final results table as shown in Fig. 30. This table includes the best estimate k_{eff}, which corresponds to the calculated k_{eff} value with the minimum standard deviation from the table in Fig. 25, and other key calculated results. For this example, the best estimate k_{eff} is calculated as 1.0036 ± 0.0022, which is near critical as expected. Note the message at the bottom of the table that the chi-square statistical test for normality was satisfied at the 95% level. If this test fails, that is typically an indication that

you should increase the number of generations (GEN)—and possibly the number of particles per generation (NPG)—and rerun the case.

If your input did not run successfully, the error messages will be listed on the screen during execution or in the output file. In most cases, the errors are related to input data problems. Check to make sure your input file has the same data as the one listed in Appendix A.

Fig. 30. KENO final results table.

2.6 HTML OUTPUT

When you click on the HTML button in GeeWiz, a screen display similar to Fig. 31 should appear in your default web browser. A notice may appear in the information bar at the top of the page such as "To help protect security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer. Click here for options…" If so, click on the message and give the browser permission to run these. Note that this message will appear every time that you open a SCALE HTML output file unless you modify the options in your browser to allow Java scripts and ActiveX controls to run.

Fig. 31. SCALE KENO-VI HTML output home page.

Click on **1. Keno-VI** to view the KENO-VI output in your browser. This will open the page shown in Fig. 32. The program verification information table contains the same information as the KENO-VI banner page in the text output file. It includes the program version number and date created, plus the date, time, and machine name for the job execution.
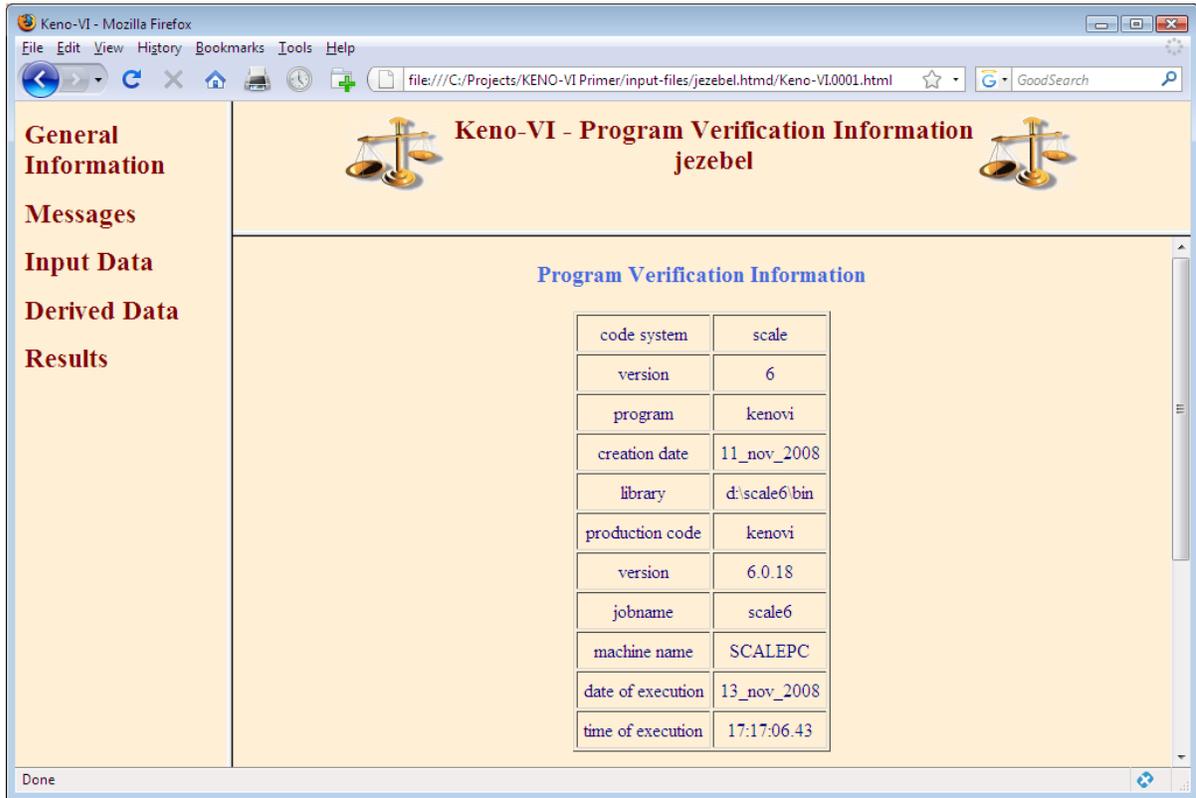
Fig. 32. KENO-VI HTML program verification table.

In the left frame are links to open different portions of the KENO-VI output. Each of these corresponds to a portion of the text output. Under General Information is the program verification information table that is initially displayed. Under **Messages** are **Warning Messages** and **Error Messages** (if any were generated).

The warning messages for this case are shown in Fig. 33. Note that there are no error messages for this case. The k6-222 warning messages for bad moments in cross-section data can usually be ignored, unless there are a large number of them. If you suspect a problem, contact scalehelp@ornl.gov. The k6-239 warning indicates that geometry volume data have not been calculated or input. Volume data are not required to calculate k_{eff}. We will discuss volume data later in the primer. The k6-132 message occasionally occurs if KENO is unable to find unique starting points for each neutron in a generation. (The number per generation in this case is the default value of 1000.) If many of these messages are generated, or the number started is significantly lower than NPG, use of the READ START data block to start neutrons in specific parts of the geometry may be necessary. The last message, k6-123, is an informational message that we discussed earlier in the text output. It indicates that the Monte Carlo calculation ended normally after the specified number of generations.

Fig. 33. KENO-VI HTML warning messages.

The subsections under **Input Data** contain the same information as the text output regarding input data used by KENO. The only subsection under **Derived Data** is the volume data. As in the previous section on text output, we will focus on the **Results** portion of the HTML output.

Click on Results and then on Final Results to display the page shown in Fig. 34. This page contains the same data as shown previously in Fig. 30.
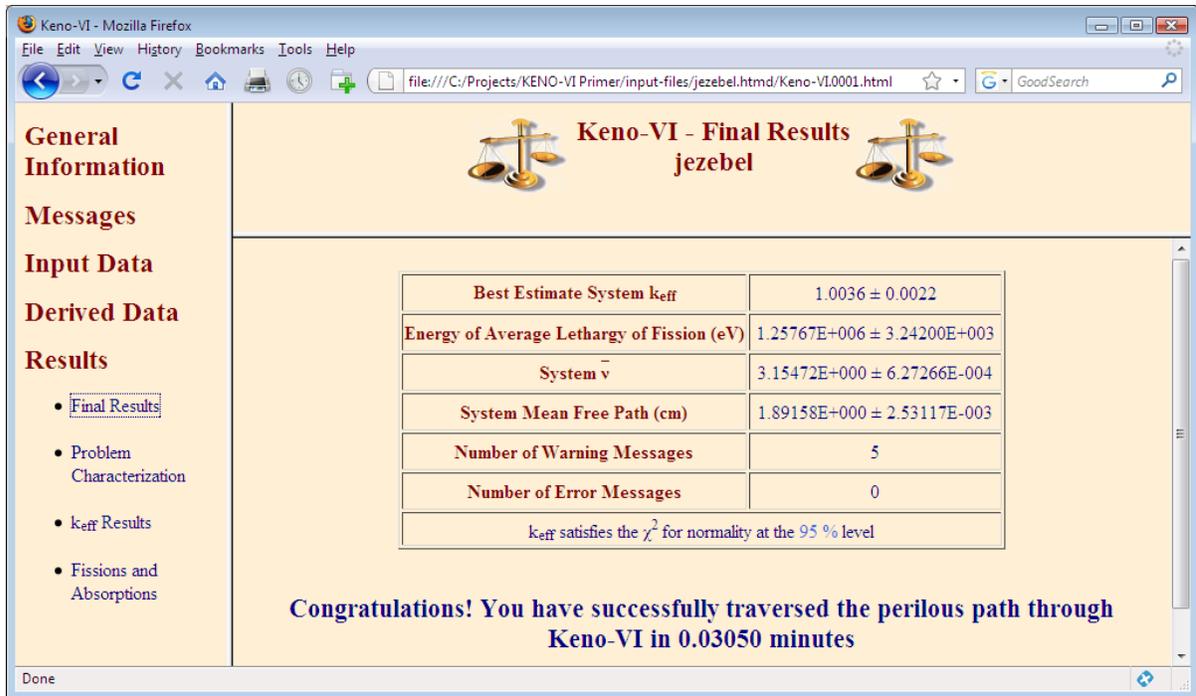
Fig. 34. KENO-VI HTML final results page.

Click on **Problem Characterization** to display the page shown in Fig. 35. This page contains the same data as shown previously in the top portion of Fig. 25.
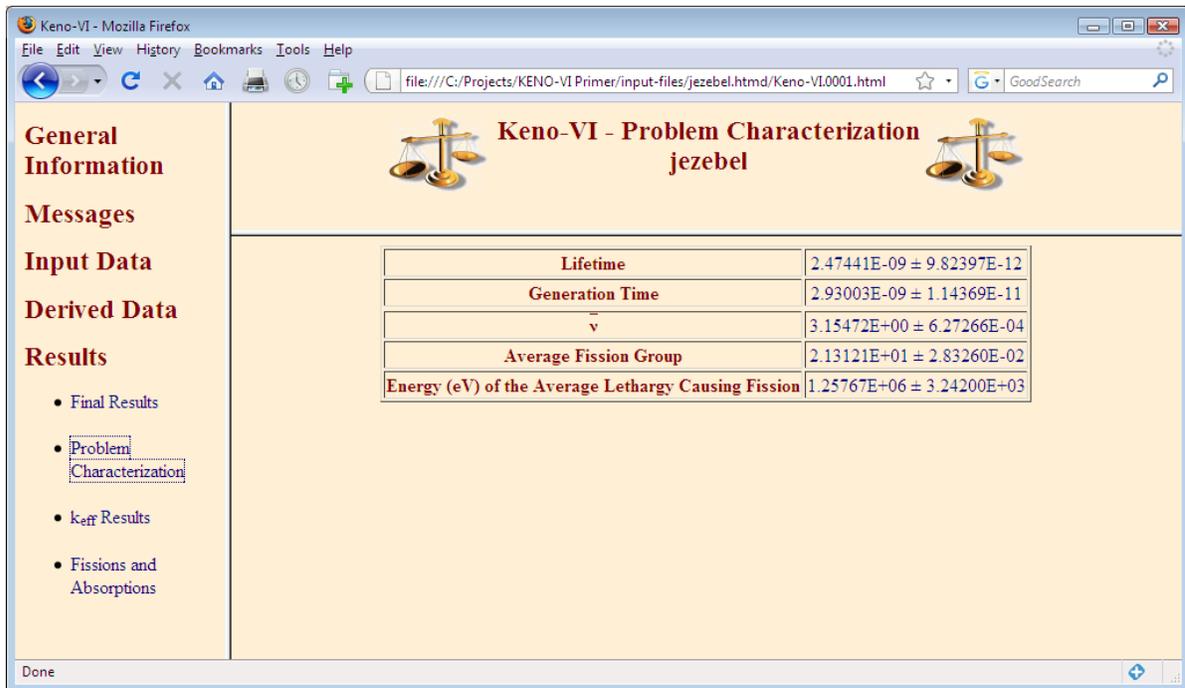
Fig. 35. KENO-VI HTML problem characterization page.

Click on **k$_{eff}$ results** and then on **k$_{eff}$ by Generations Plots**. When you click on this link the first time, you may see the dialog box shown in Fig. 36. These plots use an applet from the SCALE Javapeño plotting program. For these plots to display properly, the Java runtime environment (JRE) and the Java Advanced Imaging components must be installed as documented in the *Getting Started with SCALE 6 for Windows* document, which is installed with SCALE 6. If this dialog appears, check the box "Always trust content from this publisher" and click **Run**.

Fig. 36. Javapeno digital signature dialog window.

If the Java components are installed, the k_{eff} by generations run plot is then displayed in the right frame (Fig. 37). This corresponds to the text output partially shown in Fig. 26. The HTML plots are clearly superior to the text output versions. In addition, these plots are interactive. For example, you can use click and drag the mouse over a portion of the plot to zoom in on that portion of the plot. Right-click and select **Reload Plot** to redraw the original plot. You can click and drag on the legend to move it. You can double-click on the plot to open a dialog to customize its appearance.
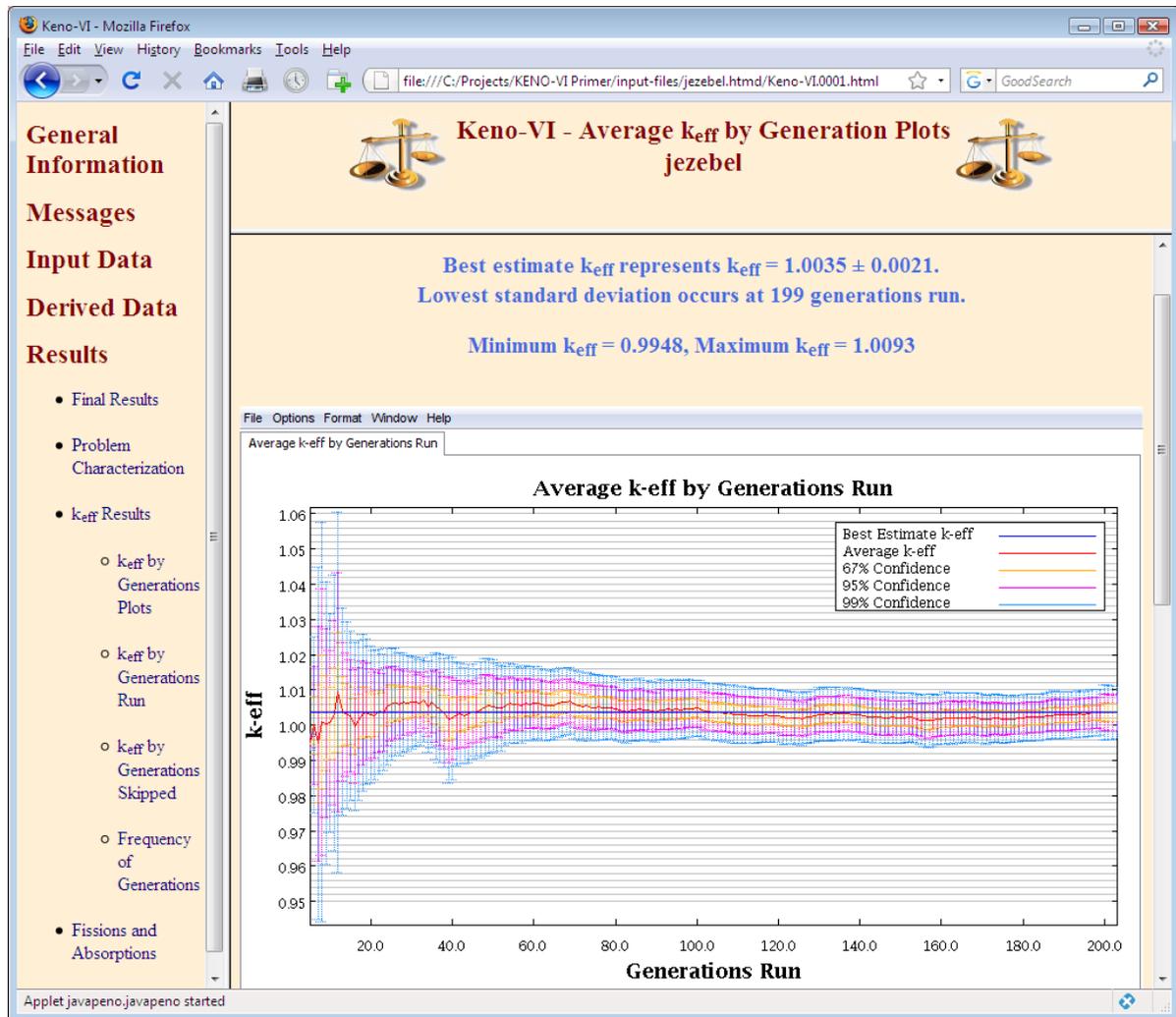
Fig. 37. Average k-eff by generations run plot.

Scroll down the right frame to view the k-eff by generations skipped plot (Fig. 38) that corresponds to the text output partially shown in Fig. 27.

Clicking on k_{eff} by Generations Run or k_{eff} by Generations Skipped displays a table of the values that appear in the corresponding plot.
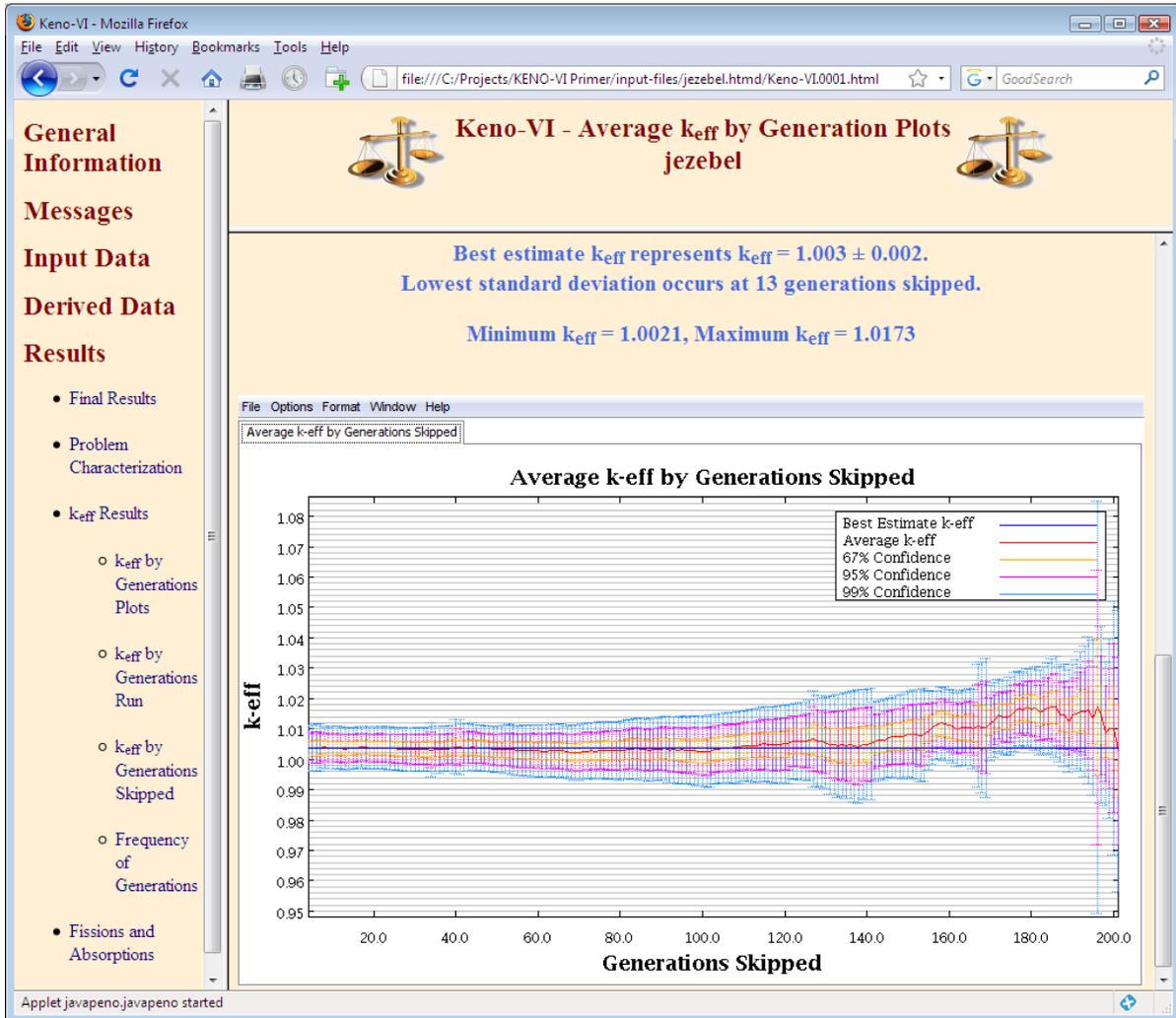
Fig. 38. Average k-eff by generations skipped plot.

Click on **Frequency of Generations** to display the frequency of k-eff by generations plot in Fig. 39. Scrolling down the right frame displays the plots that correspond to the text output in Fig. 29.
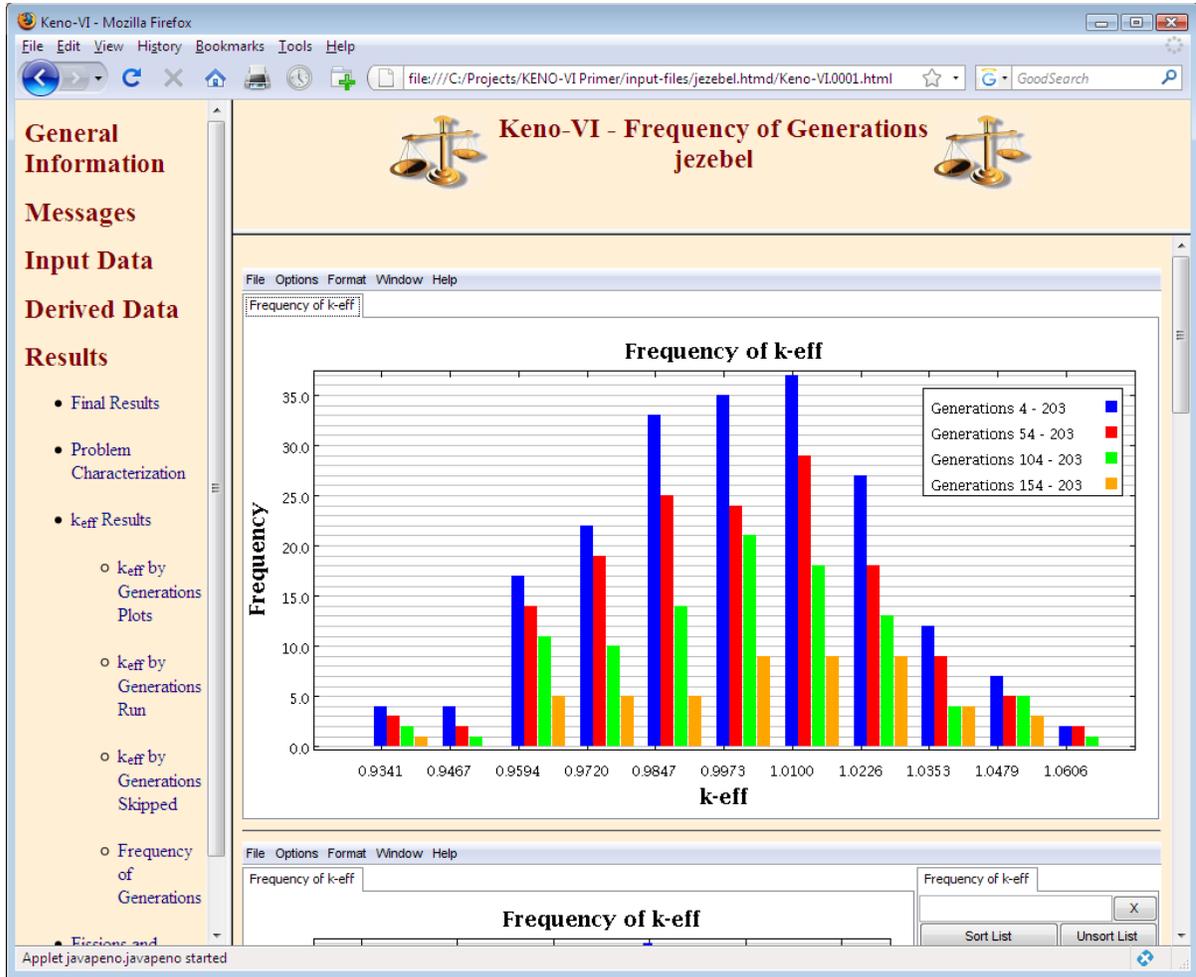
Fig. 39. Frequency of k-eff by generations plot.

Click on **Fissions and Absorptions** and then **Fissions and Absorption Data** to display the table in Fig. 40. This table displays the data corresponding to the text output in Fig. 28. Note that an additional frame appears on the right of the table. The links in this frame display different portions of the fission and absorption data output.
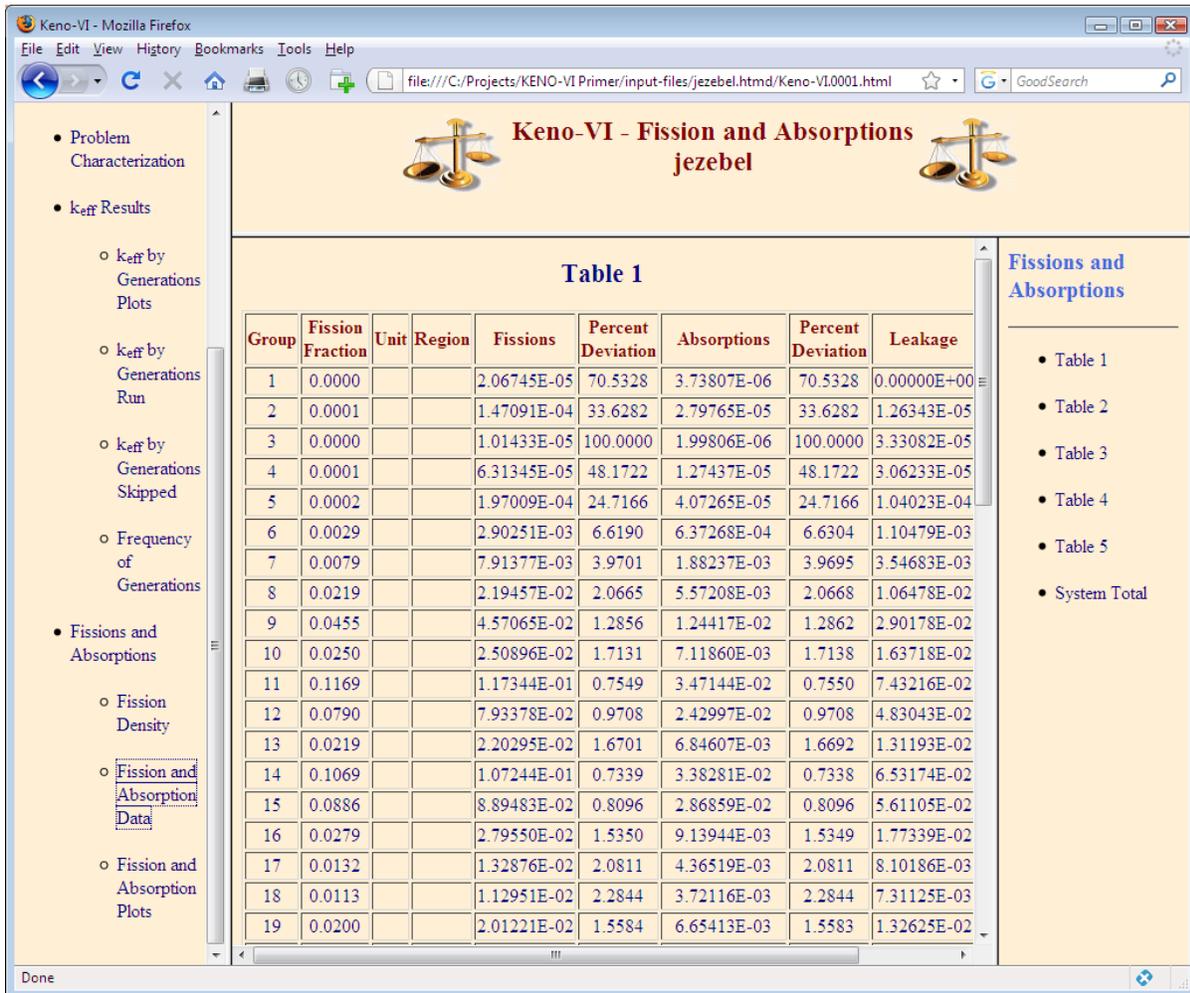
Fig. 40. KENO-VI HTML fissions and absorptions table.

Click on **Fissions and Absorption Plots,** and a blank plot window is displayed in the right frame. Double-click on **System Fission Production** and on **System Absorption** on the right side of the plot window to display the plot in Fig. 41. The data in the plot are from the Fission and Absorption Data table. This plot clearly depicts the fast spectrum of the Jezebel plutonium metal sphere.   The number of fissions is much greater than the number of absorptions because of the high leakage of the system as noted in the previous section.
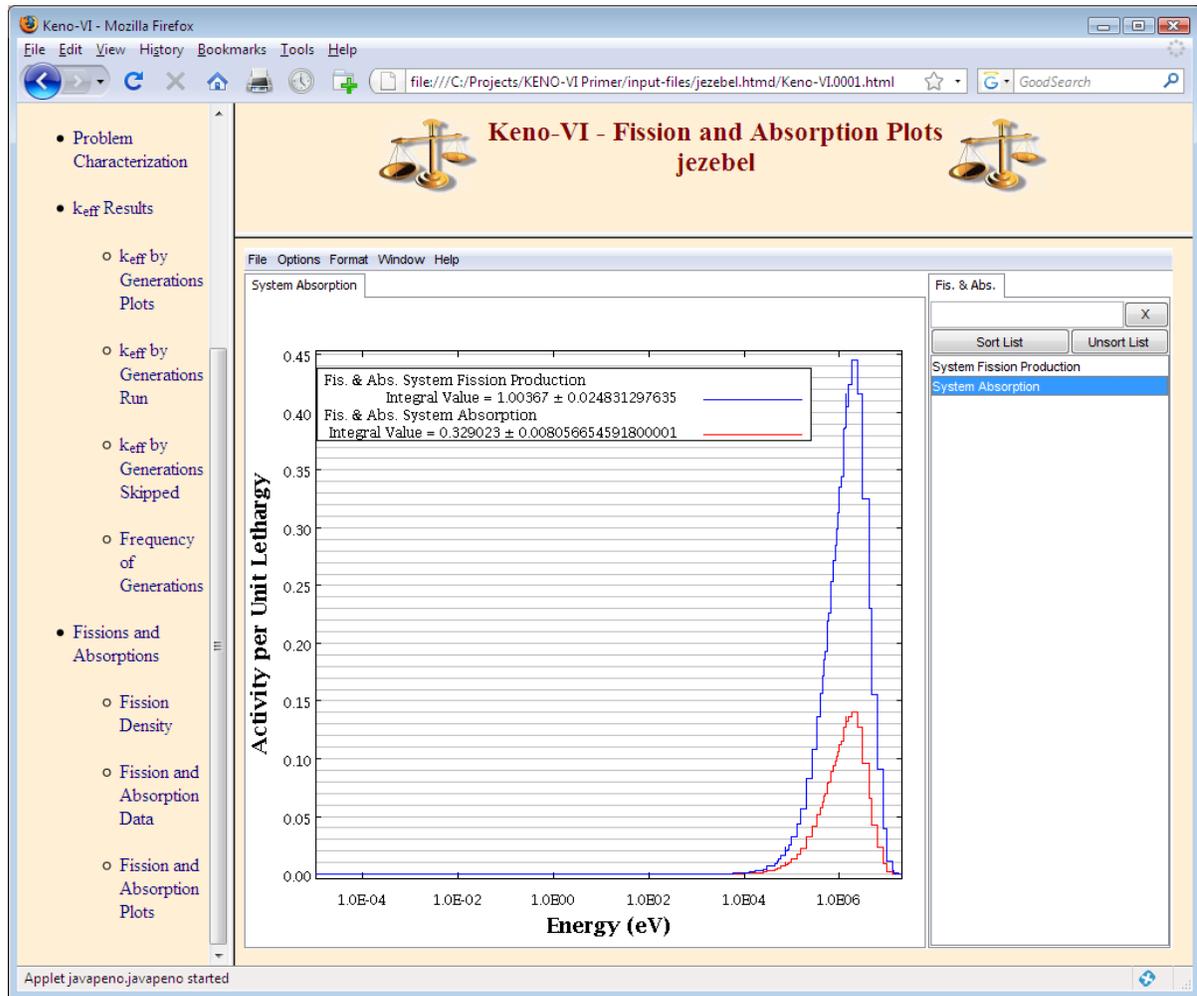
Fig. 41. Plot of system fission production and absorption data.

2.7   SUMMARY

This section has helped you to accomplish the following:

- Understand the structure of SCALE/KENO-VI input files and know that there are parts describing the sequence, the materials, and the KENO geometry information.

- Use the GeeWiz user interface to create a SCALE/KENO-VI input file.

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

- Find and interpret k_{eff} information from the text and HTML output.

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

3. MATERIAL INFORMATION INPUT

In the *Quickstart* section (Sect. 2), you ran a simple problem with SCALE using the GeeWiz User Interface. From this you gained confidence in using the code and some experience with GeeWiz. 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

- Identify the cross-section libraries available for criticality safety analyses.

- Use the GeeWiz user interface to provide input data on elements, isotopes, compounds, and solutions.

- Interpret basic output information from a SCALE/KENO-VI analysis.

3.2   CSAS6 SEQUENCE

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 with KENO-VI, the CSAS6 module uses the CSAS6 sequence (formerly known as CSAS26), as shown in Fig. 42, to provide automated problem-dependent cross-section processing followed by calculation of the neutron multiplication factor for the system being modeled. This control sequence uses the multigroup cross-section processing codes BONAMI (for the unresolved resonance energy range) and NITAWL or CENTRM/PMC (for the resolved resonance range) to provide multigroup resonance-corrected cross sections. CENTRM and PMC provide a more rigorous method than NITAWL for generating resonance-corrected multigroup cross sections using continuous-energy data. Note that CENTRM/PMC is the default for SCALE multigroup criticality safety calculations and must be used with multigroup ENDF/B-VI or B-VII libraries. Note that the cross-section processing codes are not called when continuous energy cross sections are specified.

The selection of a sequence is done through the **General Information** window of GeeWiz. Click on the **General** button to open this window. Information in this window includes a **Title** for the problem, the **Application**, and the **Sequence** that you want to execute. For KENO-VI, the only available sequence is CSAS6 (known as CSAS26 in versions of SCALE prior to Version 6.0). Select the KENO-VI application and the CSAS6 sequence.
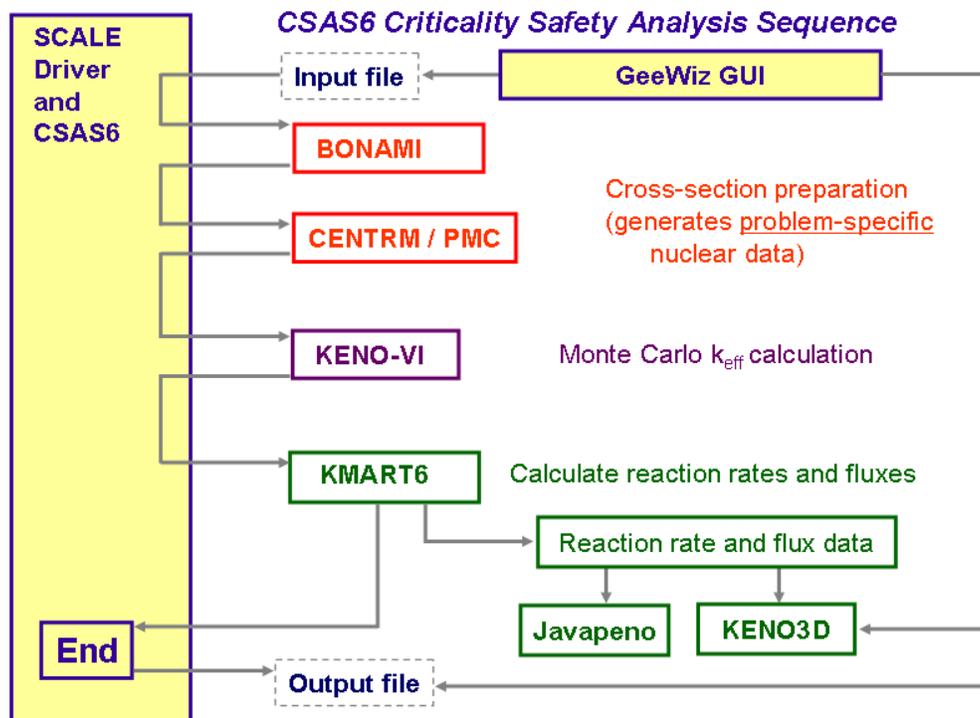
Fig. 42. CSAS6 sequence flowchart.

3.3 CROSS-SECTION LIBRARIES

The CSAS6 sequence (1) uses the Standard Composition Library (described later in this section) for specifying the materials and mixtures used in a calculation and (2) provides automatic problem-dependent multigroup cross-section preparation prior to the criticality calculation (unless continuous-energy cross sections are used). This section describes the cross-section libraries available for criticality safety calculations in SCALE 6.

There are several multigroup cross-section libraries distributed with SCALE for criticality safety analyses. The 238-group ENDF/B-V library (238GROUPNDF5 or V5-238) contains data for all ENDF/B-V nuclides and has 148 fast and 90 thermal groups. Most resonance nuclides in the 238-group library have resonance data in the resolved resonance region and Bondarenko factors in the unresolved resonance region. The 44-group ENDF/B-V library (44GROUPNDF5 or V5-44) is a broad-group version of 238GROUPNDF5 designed for analysis of light-water-reactor (LWR) fresh and spent fuel systems and has been extensively validated against LWR critical experiments. In addition, there are 238-group ENDF/B-VI (V6-238) and ENDF/B-VII libraries (V7-238) that have the same group structure as the ENDF/B-V version.

46

Continuous-energy cross sections based on both ENDF/B-VI and ENDF/B-VII are also available. Note that the ENDF/B-VII libraries (continuous energy and multigroup) are recommended, because they are based on the latest nuclear data.

Table 1 lists the standard SCALE cross-section libraries, along with the alphanumeric name and the main source of data for each library. See (Sect. M4 in the SCALE Manual) for a more detailed discussion of the cross-section libraries.

Table 1. SCALE cross-section libraries for criticality analyses

Name	Description	Primary source of data
v5-44 (44groupndf5)	ENDF/B-V 44-group neutron library	Collapsed from 238-group ENDF/B-V library
v5-238 (44groupndf5)	ENDF/B-V 238-group neutron library	ENDF/B-V data
v6-238	ENDF/B-VI 238-group neutron library	ENDF/B-VI Release 8
v7-238	ENDF/B-VII 238-group neutron library	ENDF/B-VII Release 0
ce_v6_endf	ENDF/B-VI.8 continuous-energy library	ENDF/B-VI Release 8
ce_v7_endf	ENDF/B-VII.0 continuous-energy library	ENDF/B-VII Release 0

A library is selected in GeeWiz from the pull-down menu for **Cross-section Library** (see Fig. 43).
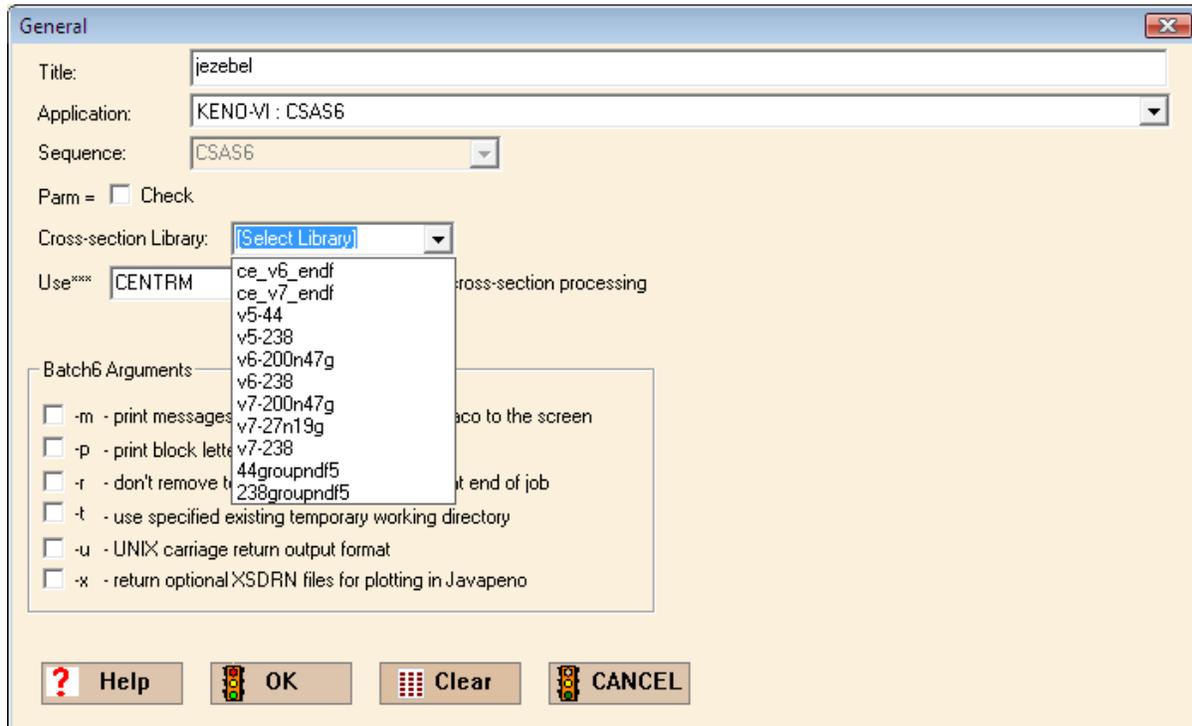
Fig. 43. Cross-section library pull-down menu.

For multigroup libraries, the criticality sequences access BONAMI first to process nuclides with Bondarenko data. The master library output from BONAMI is input to NITAWL, where resonance data are processed via the Nordheim Integral Treatment or to CENTRM, which performs a 1-D continuous-energy calculation that is subsequently collapsed to multigroup cross sections by PMC. In either case, the result is an AMPX working-format library that is used by KENO-VI. **Note that CENTRM must be used to process ENDF/B-VI or later multigroup cross-section libraries.**

3.4   MATERIAL INPUT

Information on materials in a system model is entered into GeeWiz, which puts that information into the format required by MIPLIB in CSAS6. Input data to GeeWiz 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. These input data include (1) the Standard Composition data that are 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) for multigroup calculations, the unit cell description defining the materials, dimensions, and boundary conditions of the geometry that will be used in the Dancoff factor calculations, the resonance self-shielding calculations, and the flux-weighting cell calculations necessary for multigroup cross-section processing.

48

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 (Sect. M8 of the SCALE manual). The library contains over 600 compounds, alloys, elements, and isotopes that one may use in defining the material mixtures for a given problem. Additionally, there are four fissile solutions available for which the user can specify the heavy metal, acid, and water components [UO_2F_2, $UO_2(NO_3)_2$, $Pu(NO_3)_4$, and $Th(NO_3)_4$].

When formulating a mixture, it is often necessary to know the density (g/cc) of the mixture and the weight fractions of the various constituent materials. Note that default densities should not be used for materials containing enriched isotopes, especially light elements with strong absorbers such as boron, B_4C, or lithium. The temperature of a given material can be entered (the default is 300 K, i.e., room temperature). The temperature is used to correctly process resonance data, Bondarenko data, and/or thermal-scattering data.

3.5   MATERIAL INPUT SAMPLE PROBLEMS

Four sample problems will be used to illustrate how to enter material information into GeeWiz. These problems examine uranium in a bare (unreflected) finite cylinder with four different material compositions: pure ^{235}U metal, U(93.7) metal, U(93.7)O_2, and U(30.3)O_2F_2 solution. These examples are derived from data in LA-10860.[2]

3.5.1 Bare ^{235}U Metal Cylinder

The bare metal cylinder problems used in the remainder of Section 3 have an effective radius of 7.82 cm with an estimated critical height of 15.64 cm (H/D = 1) as shown in Fig. 44. A cylinder containing pure ^{235}U is modeled first to demonstrate the use of GeeWiz to enter a single isotope. For Godiva, the uranium density is 18.742 g/cc. The dimensions are derived using buckling conversion from the critical radius of the Godiva sphere (8.741 cm). The radius of the cylinder is 7.82 cm, and the height is ± 7.82 cm.
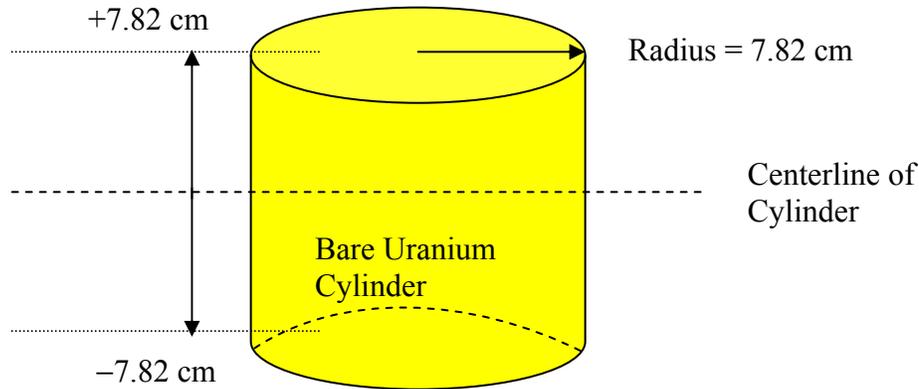
Fig. 44. Bare ^{235}U cylinder.

The initial data entry for this example is the same as that described in the *Quickstart* section. Start the GeeWiz user interface, and then select **File** and **New.** Selecting the **General** button on the left-hand side menu will bring up the screen where you enter the problem title, the SCALE sequence, the cross-section library, and the unit cell type information. Other parameter information can be entered if desired. Now enter the title as follows:

```
Bare u-235 cylinder
```

Again note that information to be entered into GeeWiz will appear in the `Courier` font. Now select `KENO-VI` as the **Application**. For **Cross-section Library**, select `v7-238`. The general information screen should look like that in Fig. 45.
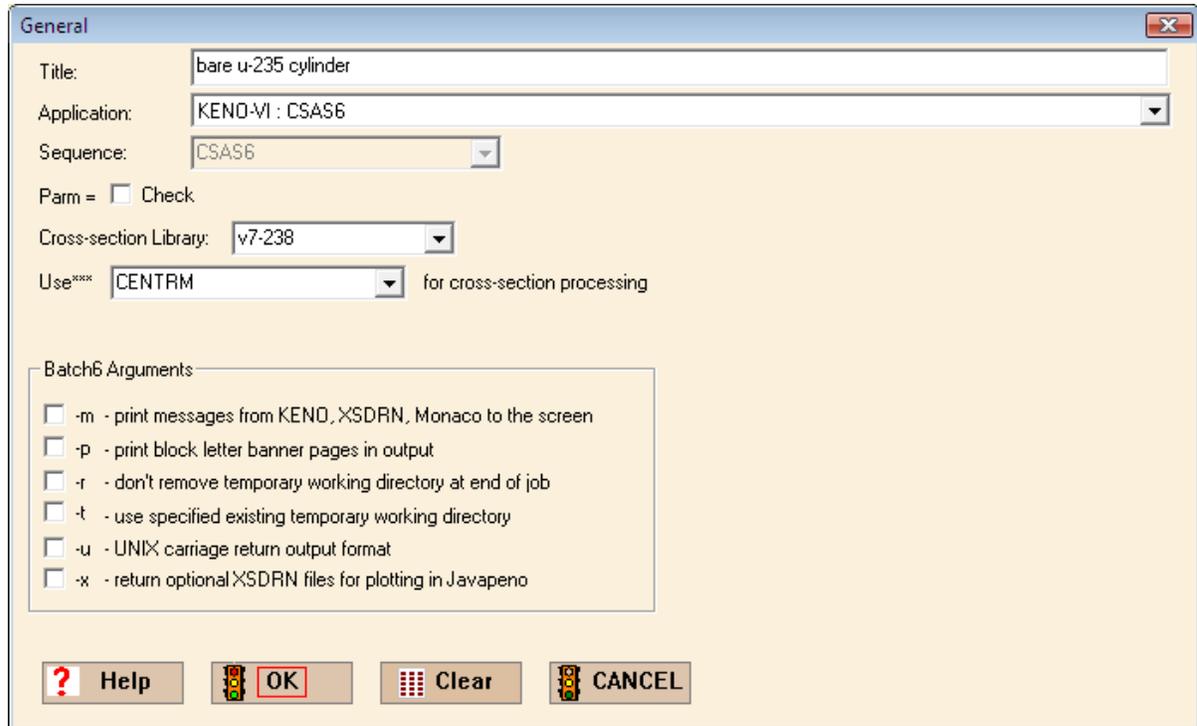
Fig. 45. General information form for bare ²³⁵U cylinder.

When completed, select **OK** to save this information in a temporary file. Later you will save all of the input information under a file name.

The Material Input is entered using GeeWiz's Materials windows. Clicking on the **Compositions** button on the left opens the **Standard Basic Compositions** form (Fig. 46), and the user must click **Create** to open the **New Composition** form (Fig. 47).
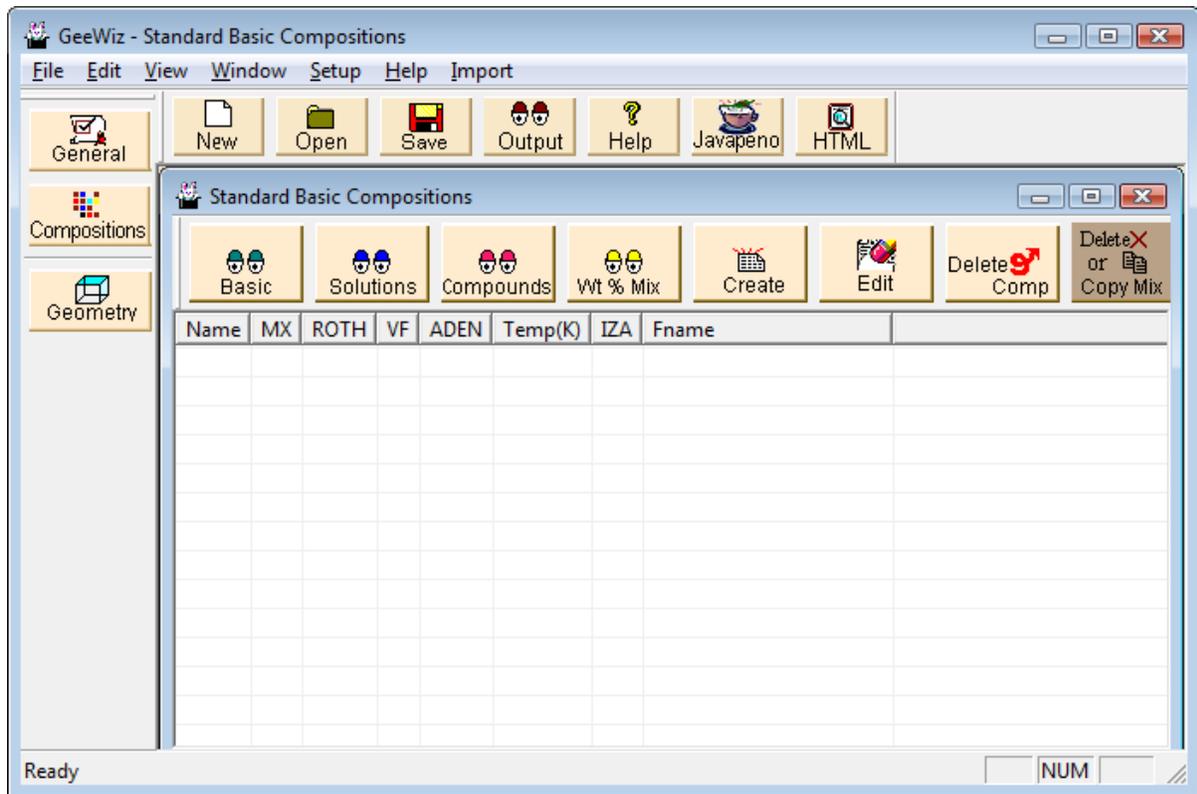
Fig. 46. Standard Basic Compositions input form.
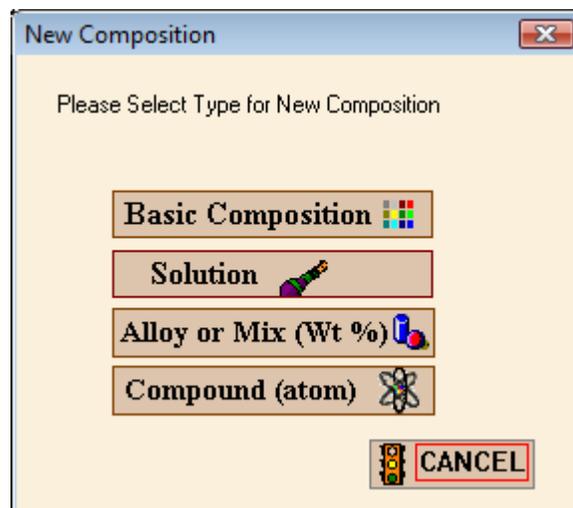
Fig. 47. New Composition form.

You will be entering data for ^{235}U, which is in the Standard Composition Library and is a Basic Composition. For this sample problem, we will select **Basic Composition,** which brings up the form shown in Fig. 48. The **Mixture** number starts with 1. Under **Composition Name** press "u" and scroll down the menu to select u-235. This brings up the default values of **Density Multiplier** (1) and **Temperature (K)** (300). Now we need to enter the density by selecting **User Supplied (g/cc)** and then entering the value 18.742 in the data entry box. The completed screen should look like Fig. 49.

Fig. 48. Basic Standard Composition input form.

Fig. 49. Material input for bare ^{235}U cylinder.

For this example, ^{235}U is the only material, so select **OK** to bring up the summary form for **Standard Basic Compositions** as shown in Fig. 50. Because this is all the material information required for this problem, select **Close** to return to the main GeeWiz screen.

Because we want to use the infinite homogeneous medium unit cell type, unit **Cell Data** are not required.
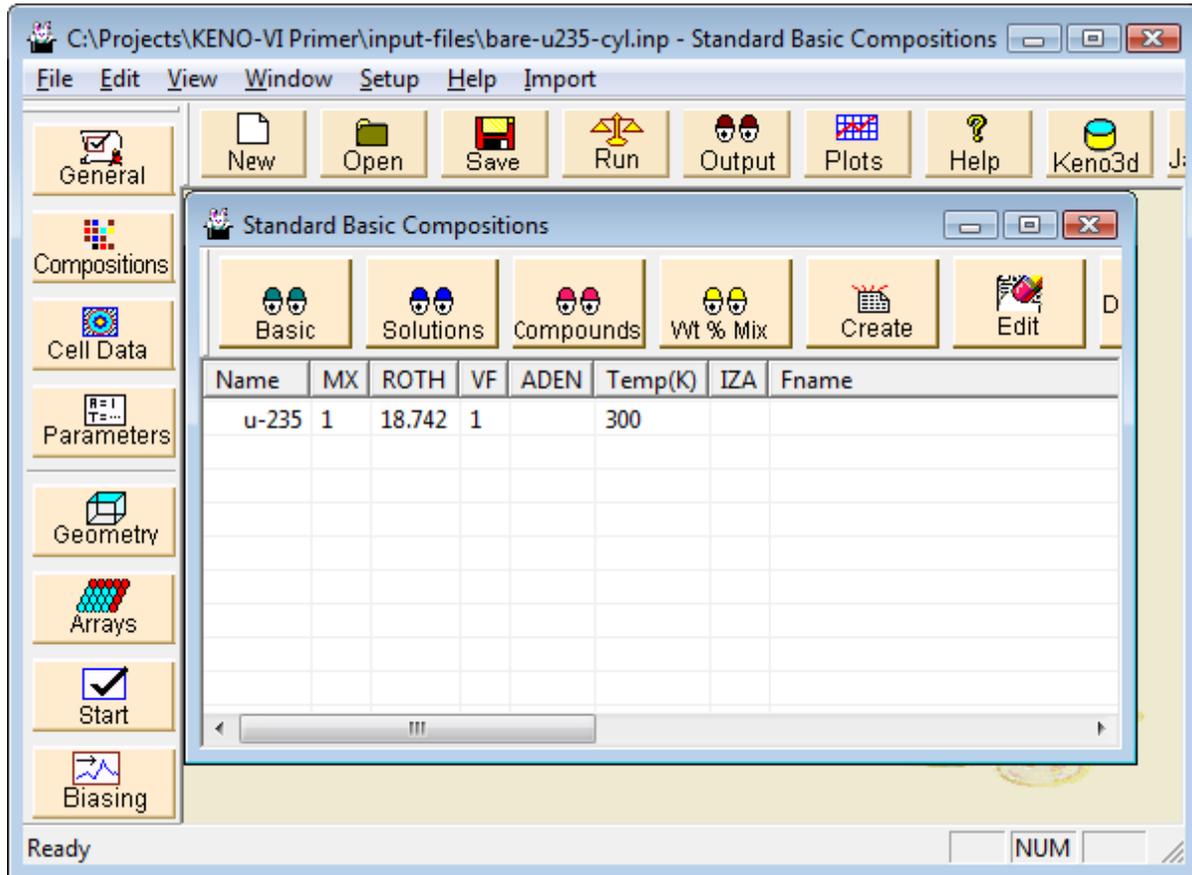
Fig. 50. Summary of material input.

With this simple geometry, only a few entries are needed to complete the KENO input form. The only KENO-VI input required for this problem is the geometry. Select **Geometry** from the left-hand toolbar to bring up the geometry summary form shown in Fig. 51. To enter geometry information, select the appropriate geometry from the right toolbar. For this problem, that would be **Cyl**, which is cylinder. This brings up the **Cylinder** input form shown in Fig. 52. Enter `7.82` for **R** (i.e., radius), `7.82` for **Zt** (Z-top) and `−7.82` for **Zb** (Z-bottom). This completes the Cylinder input form, and it should look like Fig. 53 Click **OK** to close the form.
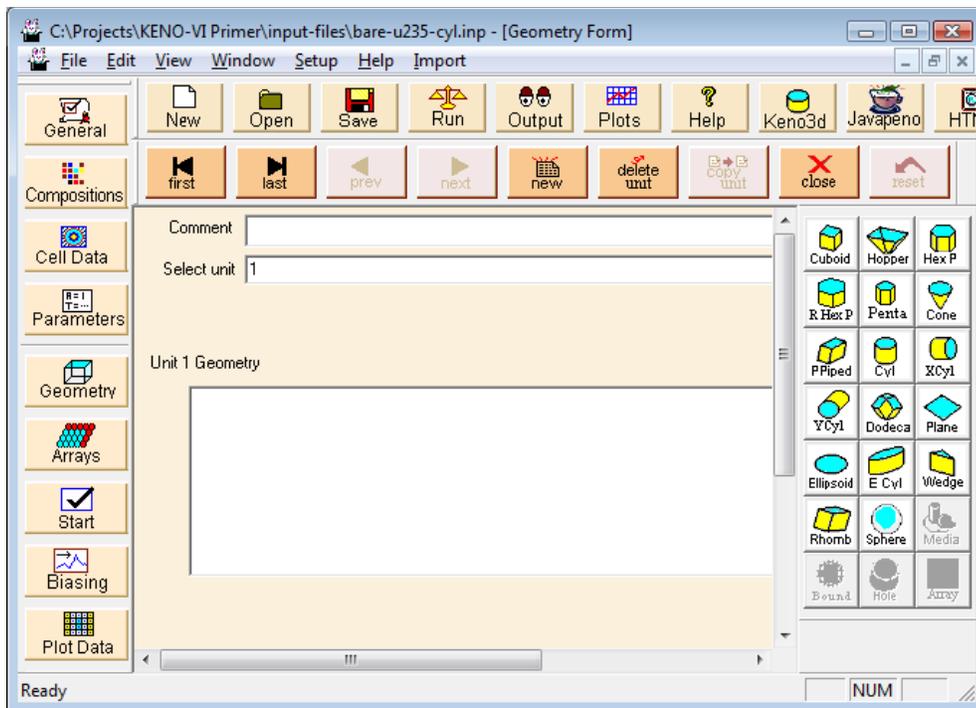
Fig. 51. Geometry summary form.
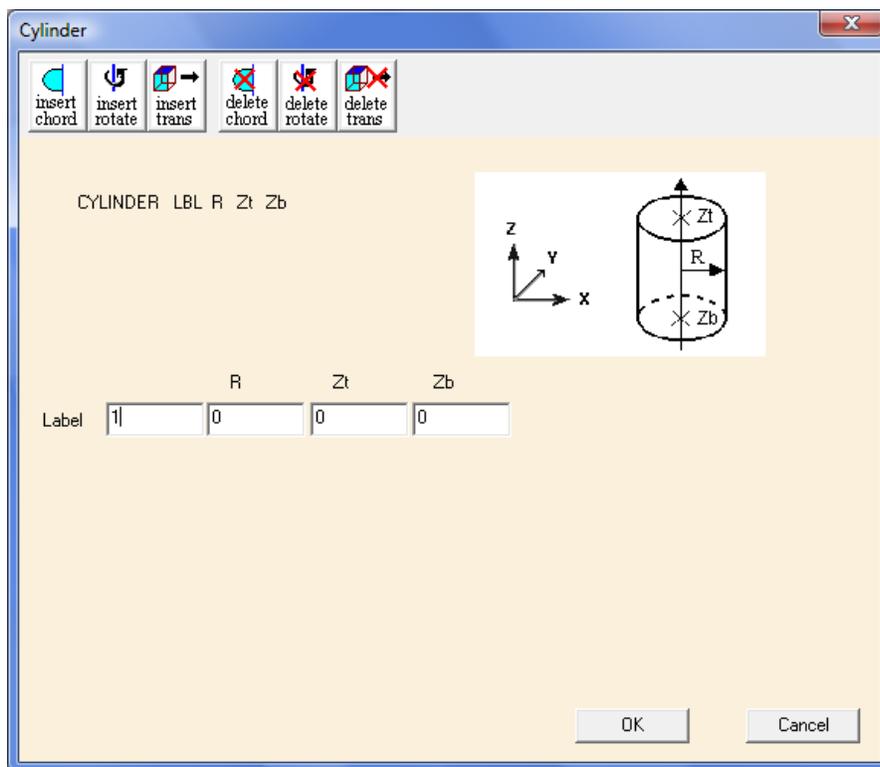
Fig. 52. Cylinder input form.
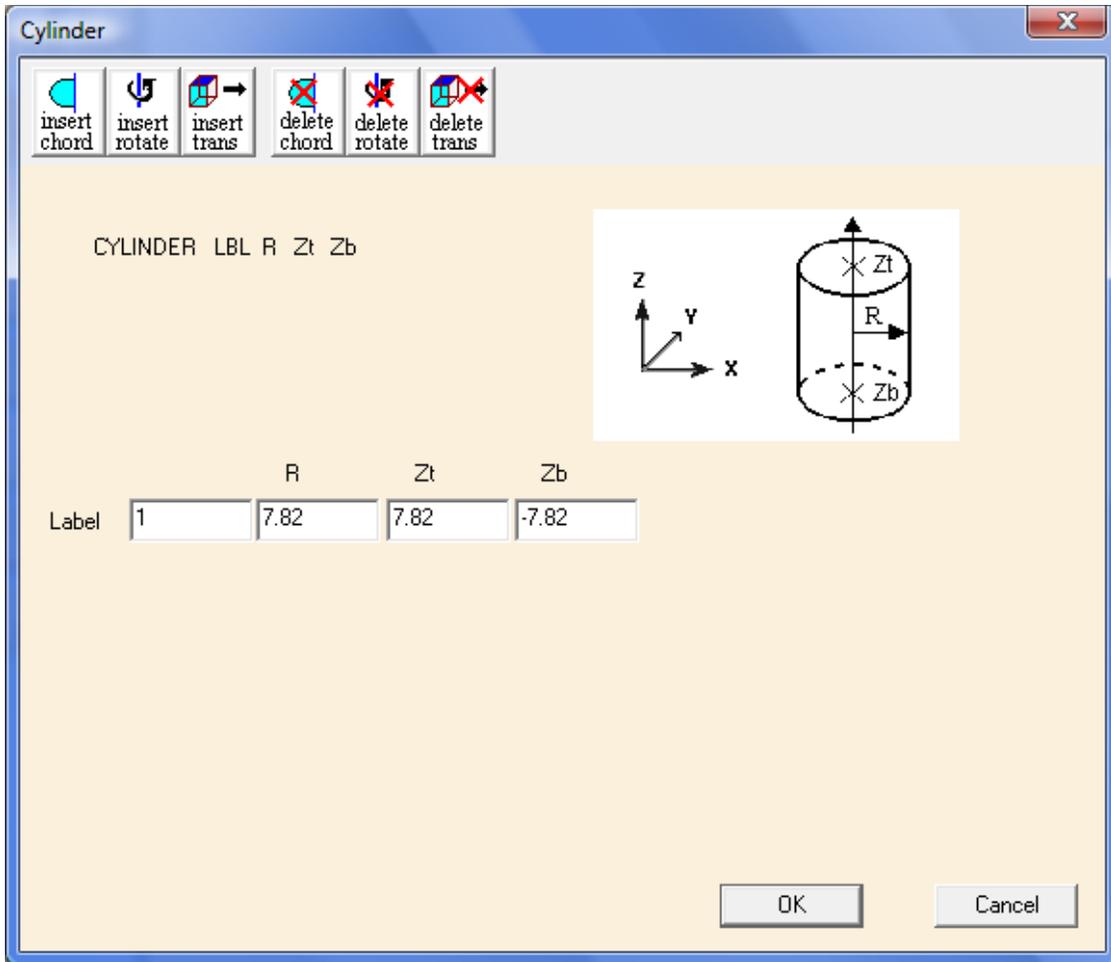
Fig. 53. Cylinder geometry information.

Next we must describe the contents of the geometry regions. Click on the **Media** button on the right toolbar to open the Media form. Click on `cylinder 1 7.82 7.82 -7.82` under **Select Geometry**. Then click on the **Inside** button, and the value `1` appears in the box under **Region Definition Vector for Media** at the bottom of the form. Click on the "Select Mixture" pull-down button and select the appropriate mixture, `1 u-235`. The completed form should look like Fig. 54. Click **OK** to close the form.
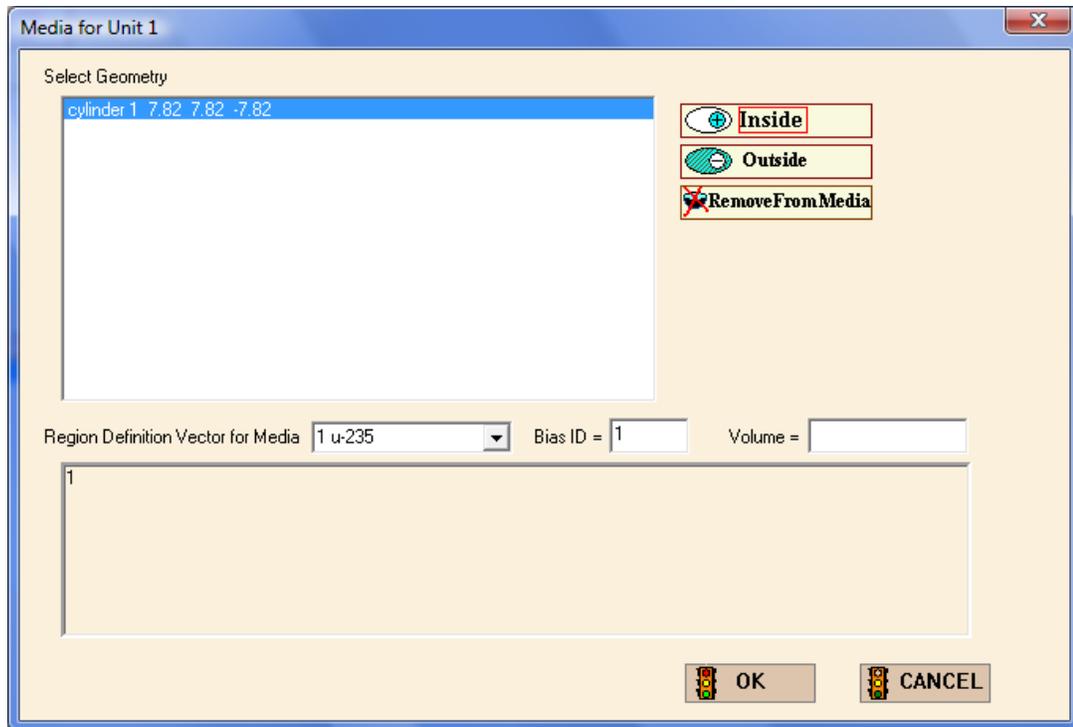
Fig. 54. Media form.

The outer boundary of the unit must be defined. Click on the **Bound** button to define the **Boundary Region Definition Vector**. This form is similar to the Media form. The outer boundary for the unit is cylinder 1. Click on `cylinder 1 7.82 7.82 -7.82` under **Select Geometry**. Then click on the **Inside** button and the value `1` appears in the box under **Boundary Region Definition Vector.**

Click on **OK** to return to the geometry summary form. Enter a comment in the **Comment** box such as `bare u-235 cylinder`, and then check the box beside **Global Unit** by clicking on the box. The global unit in KENO is the final unit that defines the entire system to be analyzed. The geometry summary screen should look like Fig. 55. Selecting **Close** from the menu bar will save this information and will return you to the main GeeWiz screen. This completes the information required for the first example. Click on the **Run** button, and GeeWiz will prompt you to save the input file prior to executing SCALE. For this example, the calculated k_{eff} value is 1.0242 ± 0.0024, which is 2% above critical, as a result of the use of 100% ^{235}U instead of 93.7% as in Godiva.
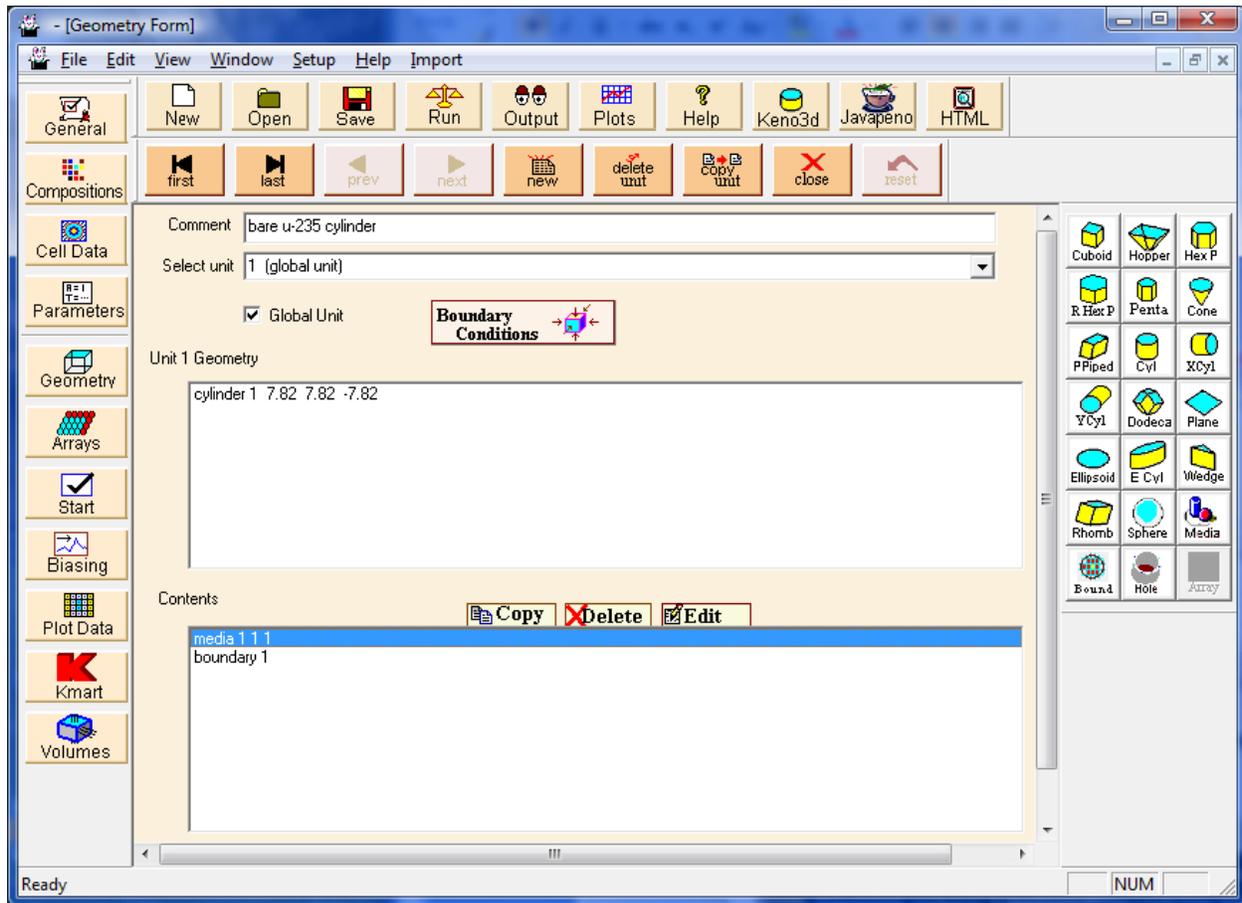
Fig. 55. Geometry summary form for bare ^{235}U cylinder.

3.5.2   Bare U(93.71) Metal Cylinder

This example changes the material from pure ^{235}U to uranium metal enriched to 93.71% ^{235}U. Again the effective radius is 7.82 cm with an estimated critical height of 15.64 cm (H/D = 1). The geometry is unchanged from that shown in Fig. 44. The dimensions are derived using buckling conversion from the critical radius of the Godiva sphere (8.741 cm). For this problem, the uranium density is 18.742 g/cc.

The data entry is the same as for the first problem. The only change is in the materials. You can either create a new file or edit the one from before, saving it under a different name. The following discussion assumes that you are editing the input from the previous case.

For this sample problem, we will again select **Compositions**. Select mixture 1 on the summary form and click Edit. This opens the **Basic Standard Composition** form with the data for mixture 1. Under **Composition Name** scroll down the menu and select u. This brings up a window to the right labeled **Isotopic Distribution**, which displays the default isotopic distribution of uranium (Fig. 56). Clicking on **Edit Selection** opens another window with the isotopes of uranium listed. We need to enter the isotopic distribution of the uranium (in weight percent) for the problem, which is 93.71 for 92235 (^{235}U) and 6.29 for 92238 (^{238}U) (see Fig. 57).

Fig. 56. Standard composition screen for isotopic distribution.

Fig. 57. Isotopic distribution window.

After you select **OK**, the composition box disappears and you can verify the density is **User Supplied (g/cc)** with a value of `18.742` in the data entry box. The information should look like Fig. 58. For this example, `u` is the only material, so select **OK** and then **Close**.
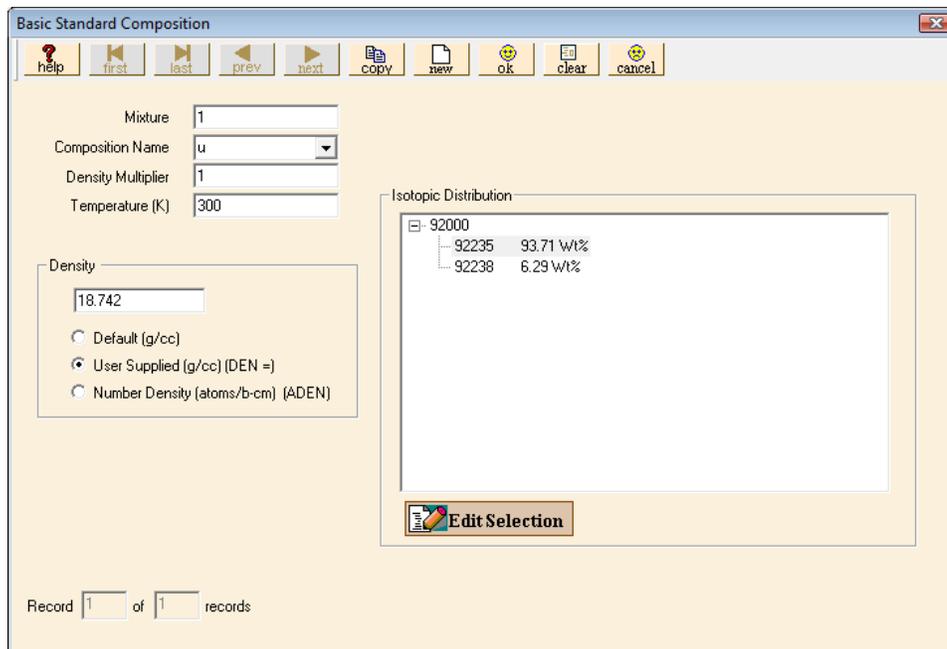
Fig. 58. Material information for U(93.71).

This completes the information required for the second example. Select **File**, then **Save As**. For the filename, you need to use a different name than before, such as `bare_u93_cyl`. After saving, click **Run**. A DOS window will open and show the progress of the calculation. For this example, k_{eff} is calculated as 0.9878 ± 0.0018, which is approximately 1% lower than critical.

3.5.3 Bare U(93.71)O₂ Cylinder

This example changes the material from uranium metal to uranium dioxide while keeping the enrichment at 93.71% ²³⁵U. The material change affects the geometry, so the new estimated critical radius is 12.8 cm with an estimated critical height of 25.6 cm (H/D = 1). The core material has a density of 95% of the theoretical density of UO_2 (10.96 g/cc).

The data entry requirements are the same as for the first problem. Both the Materials and the Geometry will change. It is suggested that you create a new file for this problem. Enter the **General** information as for the previous two examples.

Select **Compositions, Create,** and **Basic Composition**. The **Mixture Number** starts with `1`. Under **Composition Name** scroll down the menu and select `uo2`. This displays a window to the right labeled **Isotopic Distribution**. Click on **Edit Selection** and enter the isotopic composition of the uranium (in weight percent) for the problem, which is `93.71` for 92235 and `6.29` for 92238. Then select **OK** and enter a density multiplier of `0.95`. The theoretical density of UO_2 is listed in the density box as 10.96 g/cc, so the density multiplier will modify that to 0.95*10.96 or 10.412 g/cc. The data for this composition should look like Fig. 59.

For this example, `uo2` is the only material, so select **OK** and then **Close** the **Basic Standard Composition** summary form. This completes the materials input data required for this example.
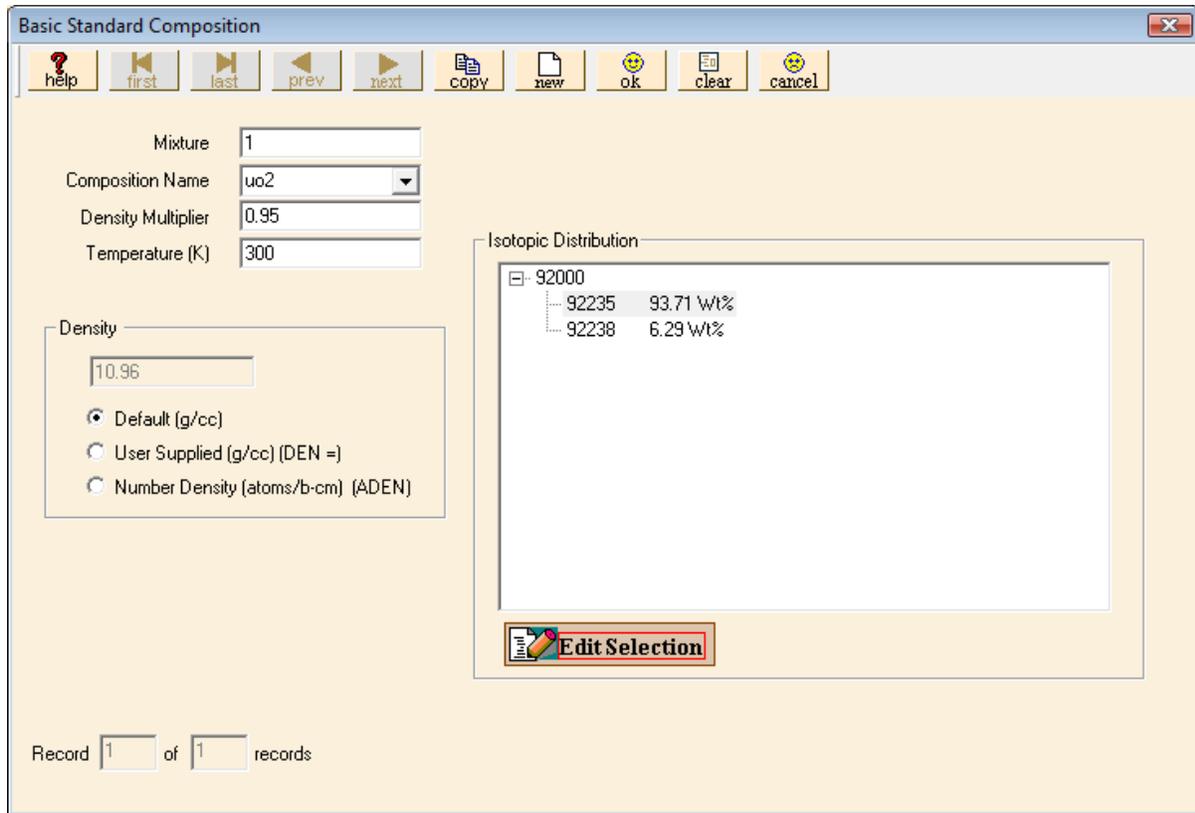
Fig. 59. Standard composition input for U(93.71)O₂.

Open the **Geometry** window. Enter a comment in the **Comment** field. Then click on the **Cyl** button. Enter `12.8` for **R**, `12.8` for **Zt**, and `-12.8` for **Zb**. Click **OK** to close the cylinder form.

Click on the **Media** button on the right toolbar to open the Media form. Click on `cylinder 1 12.8 12.8 -12.8` under **Select Geometry**. Then click on the **Inside** button and the value `1` appears in the box under **Region Definition Vector for Media** at the bottom of the form. Click on the "Select Mixture" pull-down button and select the appropriate mixture, `1 uo2`. Click **OK** to close the form.

Click on the **Bound** button to define the **Boundary Region Definition Vector**. Click on `cylinder 1 12.8 12.8 -12.8` under **Select Geometry**. Then click on the **Inside** button and the value `1` appears in the box under **Boundary Region Definition Vector.** Click **OK** to close the form.

Click the **Global Unit** check box. This completes the information required for this example. **Close** the Geometry window, and click **Run**. A DOS window opens and displays the progress of the calculation. For this example, k_{eff} is calculated as 0.9915 ± 0.0015, which is lower than critical. This is due to the approximate geometry dimensions.

3.5.4 Bare U(30.3)O$_2$F$_2$ Solution Cylinder

This example demonstrates one way of entering information for a fissile solution. The data are from p. 35 in LA-10860 (ref. 2) for U(30.3)O$_2$F$_2$ solution with H/^{235}U ratio of 106 and a fissile density of 0.220 g-^{235}U/cc. The information is given for an unreflected sphere with a critical volume of 20.0 liters. Using buckling conversion, a critical cylinder with H/D = 1 has a critical radius of 15.225 cm with a critical height of 30.45 cm. GeeWiz has a special entry screen for fissile solutions, but you need the fuel density, the excess acid molarity, and the specific gravity of the solution. Because these parameters are available in LA-10860, we will use calculated atom densities for H, O, F, ^{235}U, and ^{238}U for entry in GeeWiz:

$$N_H \quad = 5.97522 \times 10^{-2} \text{ atoms-H / b-cm}$$

$$N_O \quad = 3.35605 \times 10^{-2} \text{ atoms-O / b-cm}$$

$$N_F \quad = 3.6844 \ \times 10^{-3} \text{ atoms-F / b-cm}$$

$$N_{U235} \ = 5.637 \ \ \times 10^{-4} \text{ atoms-}^{235}\text{U / b-cm}$$

$$N_{U238} \ = 1.2802 \ \times 10^{-3} \text{ atoms-}^{238}\text{U / b-cm}$$

The data entry requirements are similar to those for the previous problems. We will concentrate on the Materials entries. Create a new file for this problem and enter the **General** information as previously.

For this problem, again create a **Basic Composition**. As the materials are mixed together and you are entering atom densities, they will all be part of mixture number 1. You will enter five materials (u-235, u-238, h, o, and f) for mixture 1. The **Mixture Number** starts with 1. Under **Composition Name** scroll down the menu and select u-235. In the density boxes below, select **Number Density (atoms/b-cm)**. Enter the ^{235}U atom density of 5.637×10^{-4} as 5.637e-4. (You can also enter the density as 0.0005637.) The form should look like Fig. 60.

Fig. 60. Standard composition input for U-235 in U(30.3)O$_2$F$_2$.

Note that GeeWiz automatically updates the **Mixture Number** when you go to a new composition, so you will need to reset it to `1` for the `u-238`, `h`, `o`, and `f` entries. For each of these materials, enter the atom density as given above. Then select **OK**. This completes the materials input data required for this example. The **Standard Basic Compositions** summary form should look like Fig. 61.

Fig. 61. Standard basic compositions summary form showing atom densities for U(30.3)O$_2$F$_2$.

Open the **Geometry** window. Enter a comment in the **Comment** field. Then click on the **Cyl** button. Then enter `15.225` for the **R**, `15.225` for **Zt**, and `−15.225` for **Zb**. Click **OK** to close the cylinder form.

Enter the **Media** and **Bound** data exactly as in the previous example. Then click the **Global Unit** check box. This completes the information required for the third example. Click **Run**; a DOS window then opens and shows the progress of the analysis. For this example, k_{eff} (skipping three generations) is calculated as 0.9959 ± 0.0026, which is quite close to critical as expected.

3.6   SUMMARY

This section has helped you to accomplish these objectives:

- Define the different criticality sequences used in SCALE.

- Describe the cross-section libraries available for criticality analyses.

- Use the GeeWiz user interface to provide data on elements, isotopes, compounds, and solutions.

- Interpret basic output information from a SCALE/KENO-VI analysis.

Now that you have spent time with the Material Input, we will proceed to more complicated geometries, such as reflected systems and lattices.

4. GEOMETRY INPUT

In the previous section you modeled a simple geometry with various material compositions. You used the GeeWiz user interface to provide data on elements, isotopes, compounds, and solutions. This section presents an explanation of the commands used for basic KENO-VI geometries such as cylinders and spheres and for more complicated geometries such as reflected systems and lattices.

4.1 WHAT YOU WILL BE ABLE TO DO

- Use GeeWiz to describe some basic KENO-VI geometry shapes.

- Understand how units are created (including intersecting of shapes).

- Locate and change the location of the origin for shapes and units.

- Create simple arrays consisting of a single unit.

- Create arrays with multiple units of nominally different sizes.

4.2 BASIC GEOMETRY SHAPES

SCALE/KENO-VI uses a set of basic shapes to build the geometry models. You have used GeeWiz to enter data on cylinders in Sect. 3. This and the other basic shapes are described in Table 2 and illustrated in Fig. 62 and Fig. 63.

Table 2. Basic shapes

Keyword	Description
CONE	Right circular cone
CUBOID	Box or rectangular parallelepiped
CYLINDER	Right circular cylinder
DODECAHEDRON	Surface consists of 12 rhombuses of the same size
ECYLINDER	Right cylinder with an elliptical cross section
ELLIPSOID	Elongated sphere
HEXPRISM	Box whose top and bottom faces are hexagons
HOPPER	Top and bottom faces are rectangles of different sizes centered about the Z-axis and parallel to the X- and Y-axes
PARALLELEPIPED	Six faces composed of parallelograms whose opposing faces are parallel
PENTAGON	Box whose top and bottom faces are pentagons
PLANE	Plane
RHEXPRISM	Hexprism rotated 90 degrees about Z-axis
RHOMBOID	Six identical faces, each one a rhombus
SPHERE	Sphere
WEDGE	Right-triangular prism with five faces
XPPLANE	Two planes parallel to X-axis
YPPLANE	Two planes parallel to Y-axis
ZPPLANE	Two planes parallel to Z-axis
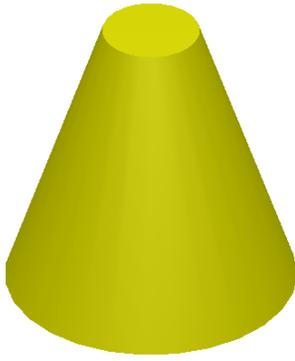
Cone
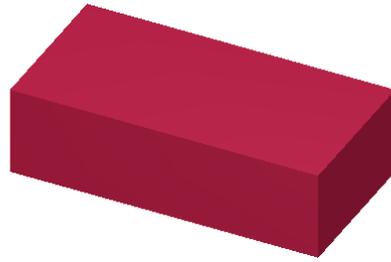
Cuboid
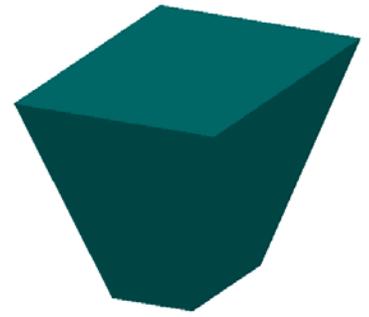
Hopper
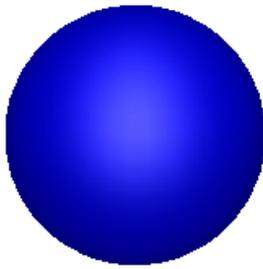
Sphere

Cylinder
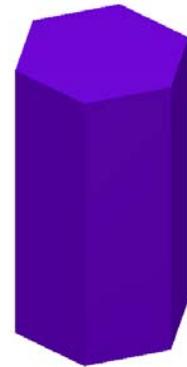
Hexprism

Wedge

Ecylinder
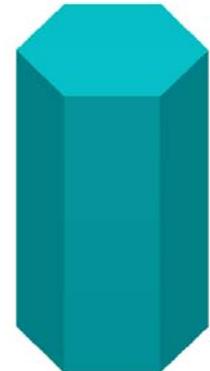
Rhexprism

Fig. 62. KENO-VI geometry shapes.
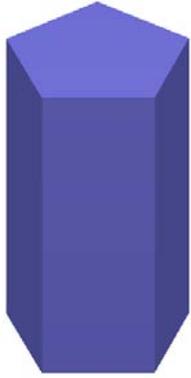
Pentagon
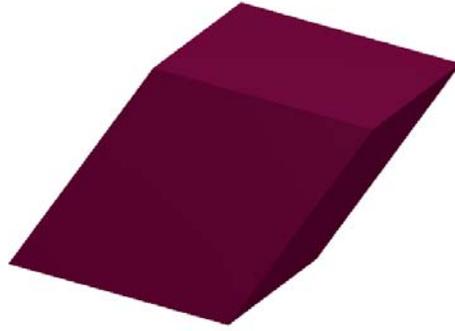
Parallelpiped
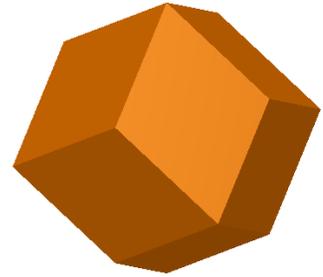
Dodecahedron
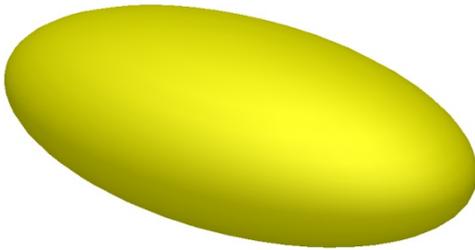
Ellipsoid
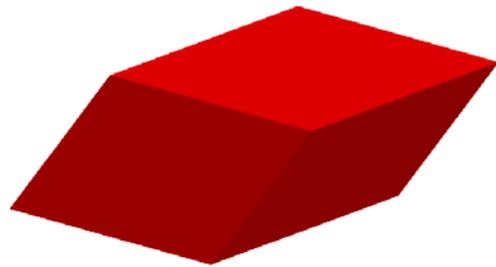
Rhomboid

Fig. 63. More KENO-VI geometry shapes.

These shapes are the basic geometry options in KENO-VI and are shown on the right side of the **Geometry** input form (Fig. 64). Each shape has a set of quantitative information needed to describe its size and location. Selection of a geometry option will bring up the associated shape input form, where the appropriate information is entered. Figure 65 shows the input form for a cuboid; the other input forms ask for similar information. A description of the options for each basic shape can be found in (Sect. F17.4.4 of the SCALE manual). Additional information may be entered to translate (relocate the origin), rotate, or to insert a chord that describes a plane surface that truncates the shape. These options will be discussed in Sect. 5.
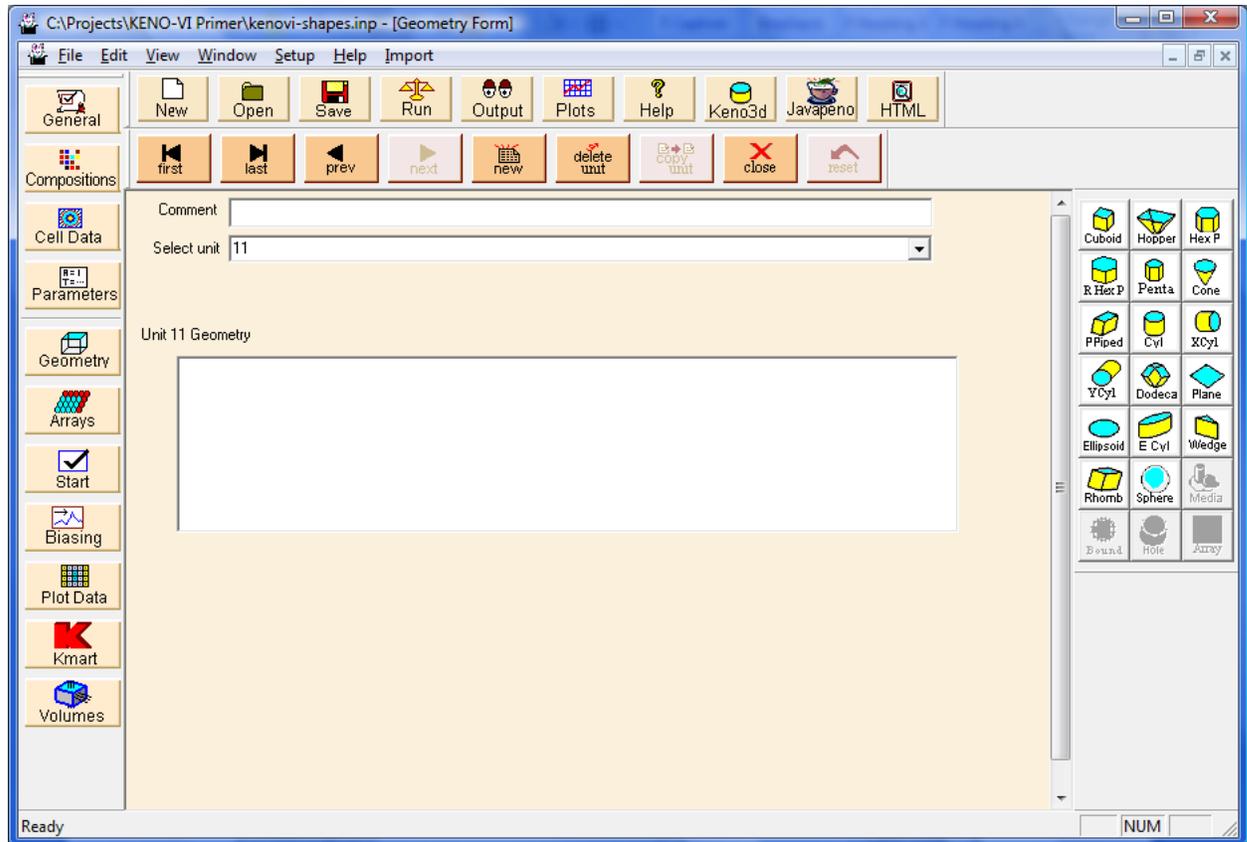
Fig. 64. Geometry form.
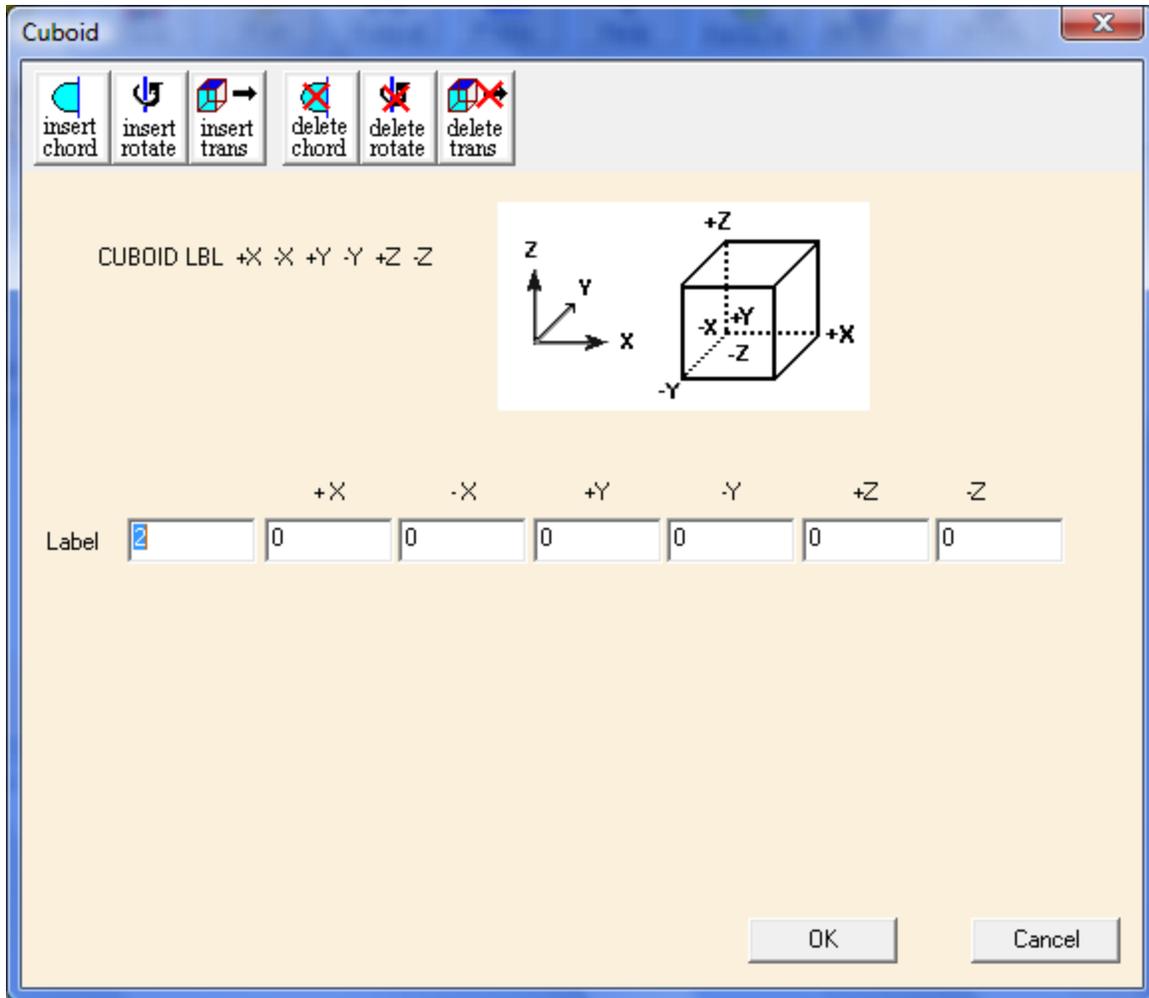
Fig. 65. Cuboid input form.

4.2.1 Simple Cylinder Examples

To understand how shape parameters are entered using GeeWiz, some sample problems will be used. These problems are based on the bare metal cylinder example from Sect. 3. Various cylindrical configurations will be examined.

4.2.2   Bare Metal Cylinder along Z-Axis

Locating the bare metal cylinder centered on the Z-axis (Fig. 66) is identical to the first sample problem in Sect. 3.

Fig. 66. Bare ^{235}U cylinder geometry.

Based on the knowledge that you gained from completing Sect. 3 of the primer, you should be able to open a new file and enter the title, cross section, and material information. Note that all problems in this chapter use the 238-group ENDF/B-VII library (V7-238).

With this simple geometry, only a few entries are needed to complete the KENO geometry input form. Select **Geometry** from the left toolbar to bring up the geometry summary form shown in Fig. 64. To enter geometry information, select the appropriate geometry from the right toolbar. For this problem, that would be the **Cyl** button, which opens the **Cylinder** input form.

Enter `7.82` for **R**, `7.82` for **Zt**, and −`7.82` for **Zb**. This completes the **Cylinder** information; it should look like Fig. 67.
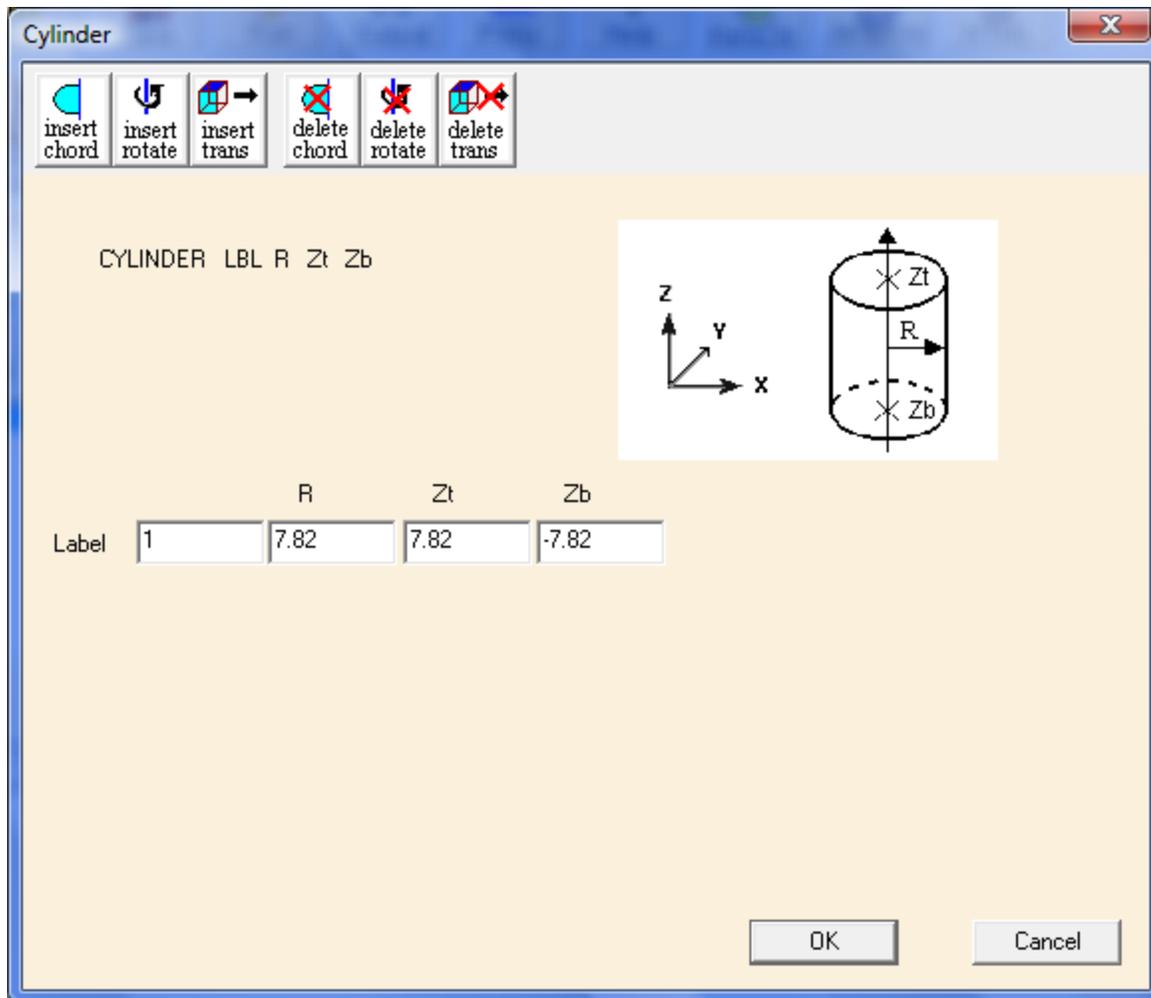
Fig. 67. Cylinder data for bare metal cylinder along Z-axis.

Click on **OK** to return to the Geometry form. Next we must describe the contents of the cylinder. Click on the **Media** button on the right toolbar to open the Media form. Click on `cylinder 1 7.82 7.82 -7.82` under **Select Geometry**. Then click on the **Inside** button, and the value `1` appears in the box under **Region Definition Vector for Media** at the bottom of the form. Click on the "Select Mixture" pull-down button and select the appropriate mixture, `1 u-235`. Click **OK** to close the form.

The outer boundary of the unit must be defined. Click on the **Bound** button to define the **Boundary Region Definition Vector**. This form is similar to the Media form. The outer boundary for the unit is cylinder 1. Click on `cylinder 1 7.82 7.82 -7.82` under **Select Geometry**. Then click on the **Inside** button, and the value `1` appears in the box under **Boundary Region Definition Vector.** Click **OK** to close the form and return to the Geometry form.

Enter a comment in the **Comment** box, such as `single z-axis cylinder`, and check the box beside **Global Unit** by clicking on the box. As stated earlier, the global

74

unit in KENO is the final unit that defines the entire system to be analyzed. The geometry summary form should look like Fig. 68.
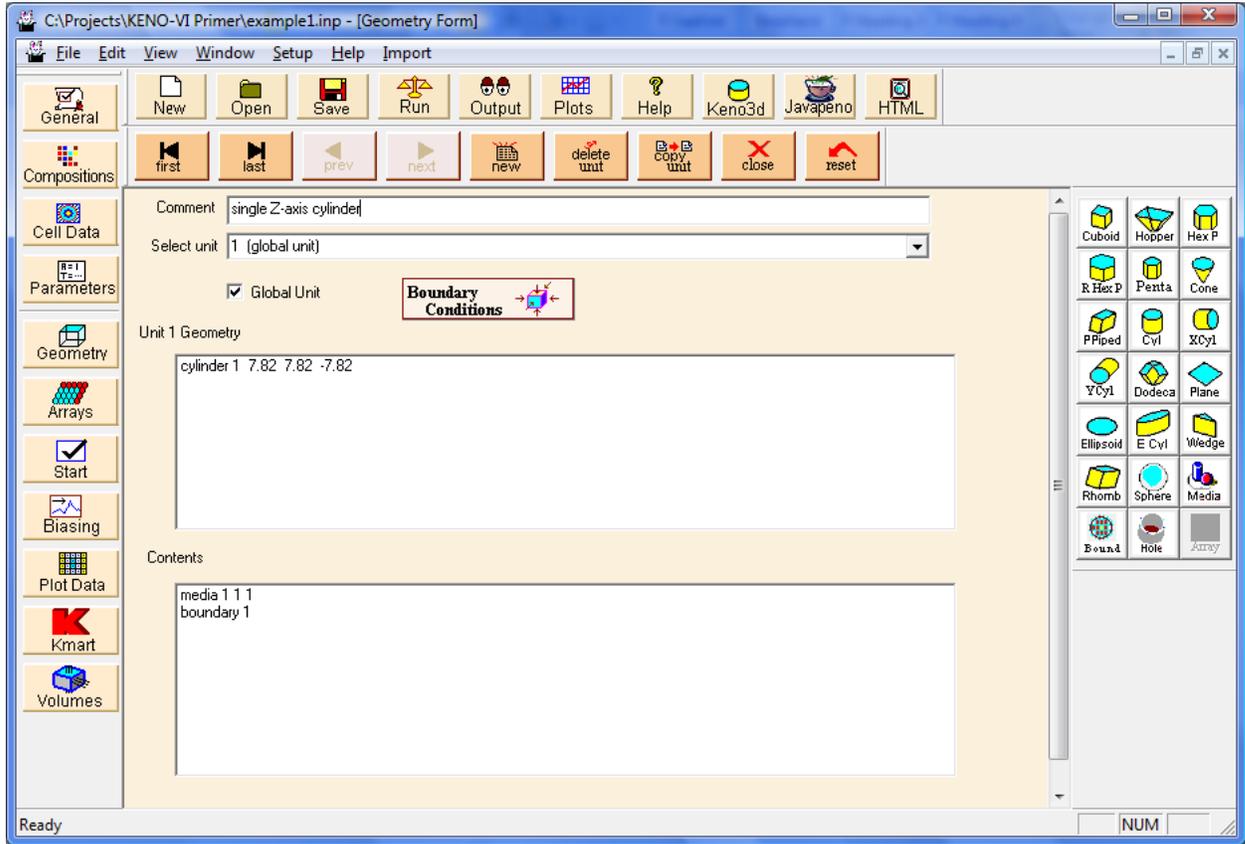
Fig. 68. Geometry for Z-axis cylinder.

Selecting **Close** from the menu bar saves this information and takes you back to the main GeeWiz screen. This completes the information required for this example. Select **Run**. GeeWiz opens a DOS window that shows the progress of the calculation.

To make sure the input is correct, check the calculated value of k_{eff}. As in the first example of the previous section, it should be approximately 1.0242 ± 0.0024. Once again, the k_{eff} is above critical, because we modeled a pure ^{235}U system rather than the critical Godiva, which was a U(93.7) system.

4.2.3 Bare Metal Cylinder along Z-Axis with Origin Located 2 cm above Cylinder Base

There are some fissile systems you will encounter where you want to locate the 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 **Zt** and **Zb**. For example, suppose we want to locate the origin at a point 2 cm above the base of the cylinder (see Fig. 69).
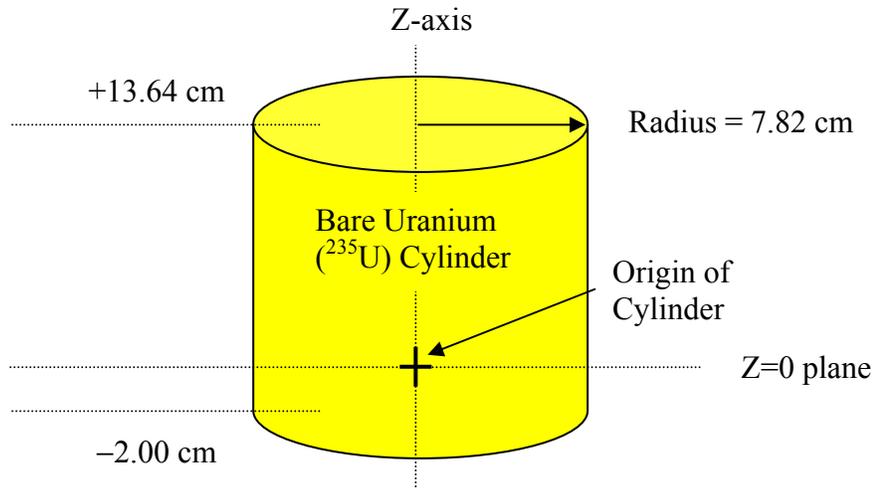
Fig. 69. Base of cylinder located 2 cm below origin.

Using the input file from the previous example, you would only need to change the height parameters on the **Geometry** form. Select **Geometry**, which brings up the geometry summary form. This has one region. Click on the region, and then select **Edit**. This brings up the Z-cylinder input form with the data from the previous example. Change **Zt** to `13.64` and **Zb** to `-2` so that the screen looks like Fig. 70.

76
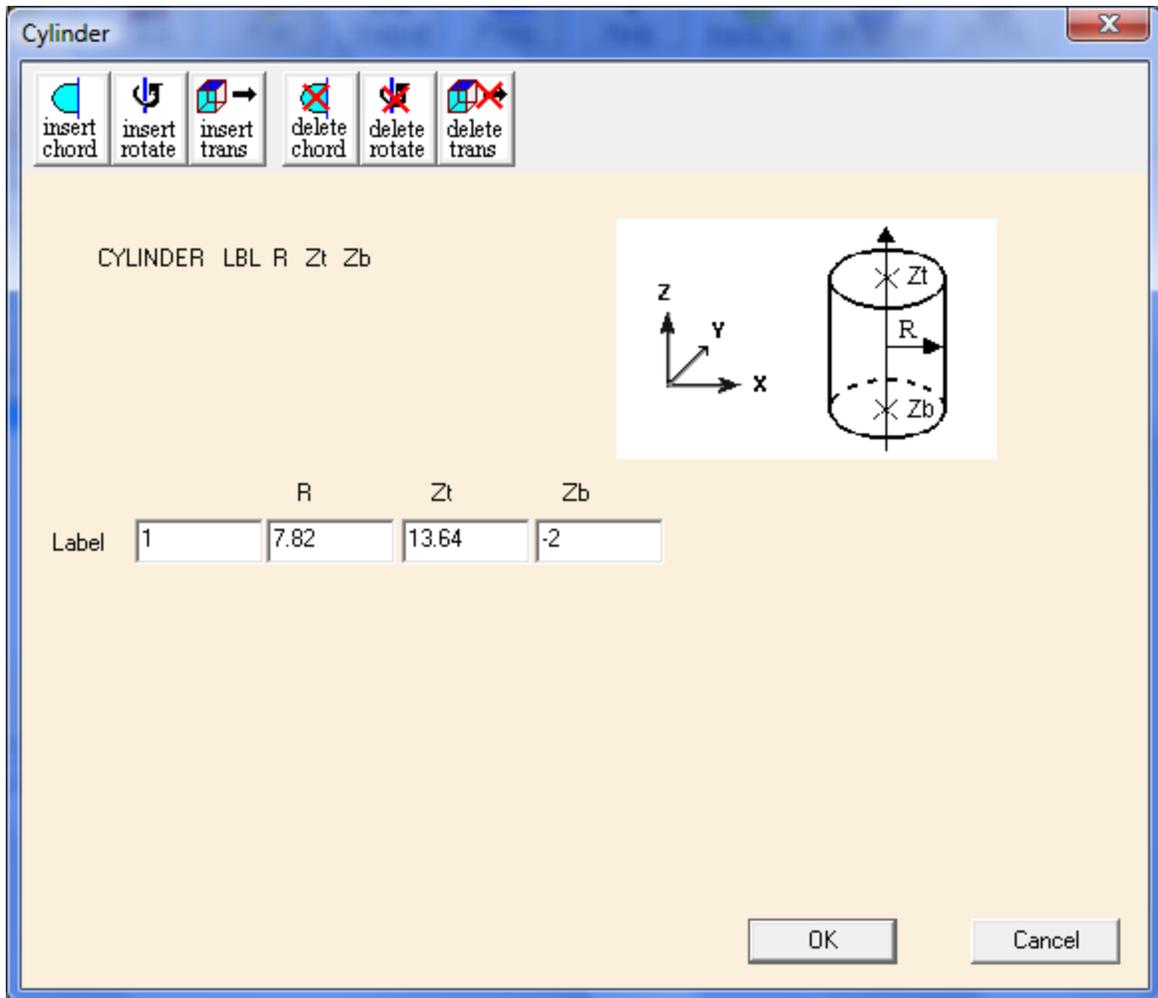
Fig. 70. Geometry input for cylinder base located 2 cm below origin.

Save the changes in a new input file and execute SCALE. Because no changes were made to the physical system, the k_{eff} value for this run should be within 2 standard deviations of that from the previous run. The k_{eff} calculated is 1.0242 ± 0.0024, which agrees with the previous result.

4.2.4 Bare Metal Cylinder along X-Axis with Origin Located at Cylinder Base

The next problem is for a horizontal cylinder with the origin located at the base as in Fig. 71. We need to change to an X-cylinder and alter the values of **Xt** and **Xb** to locate the origin at the desired point. (**NOTE:** When entering **Xt** and **Xb** values, the **Xb** value can be positive, negative, or zero, but it must be less than the **Xt** value.)
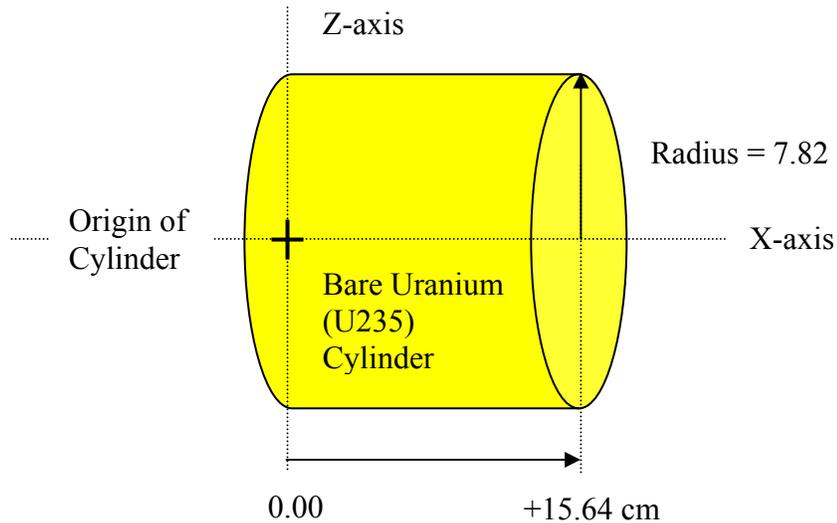
Fig. 71. Cylinder along X-axis.

Using the input file from the previous example, you need to change the cylinder orientation by replacing the **Cylinder** that is oriented on the Z-axis with an **Xcylinder**. From the geometry summary screen, select the first region of Unit 1 (**cylinder**) and click **Delete**. (Note that GeeWiz will also delete the **Media** and **Boundary** records associated with Unit 1.) Click on the **Xcyl** button, and the **Xcylinder** input form appears. Enter `7.82` for **R**, `15.64` for **Xt**, and `0.0` for **Xb** as shown in Fig. 72. Click OK to close the **Xcylinder** form. Reenter the **Media** and **Boundary** records as described earlier.
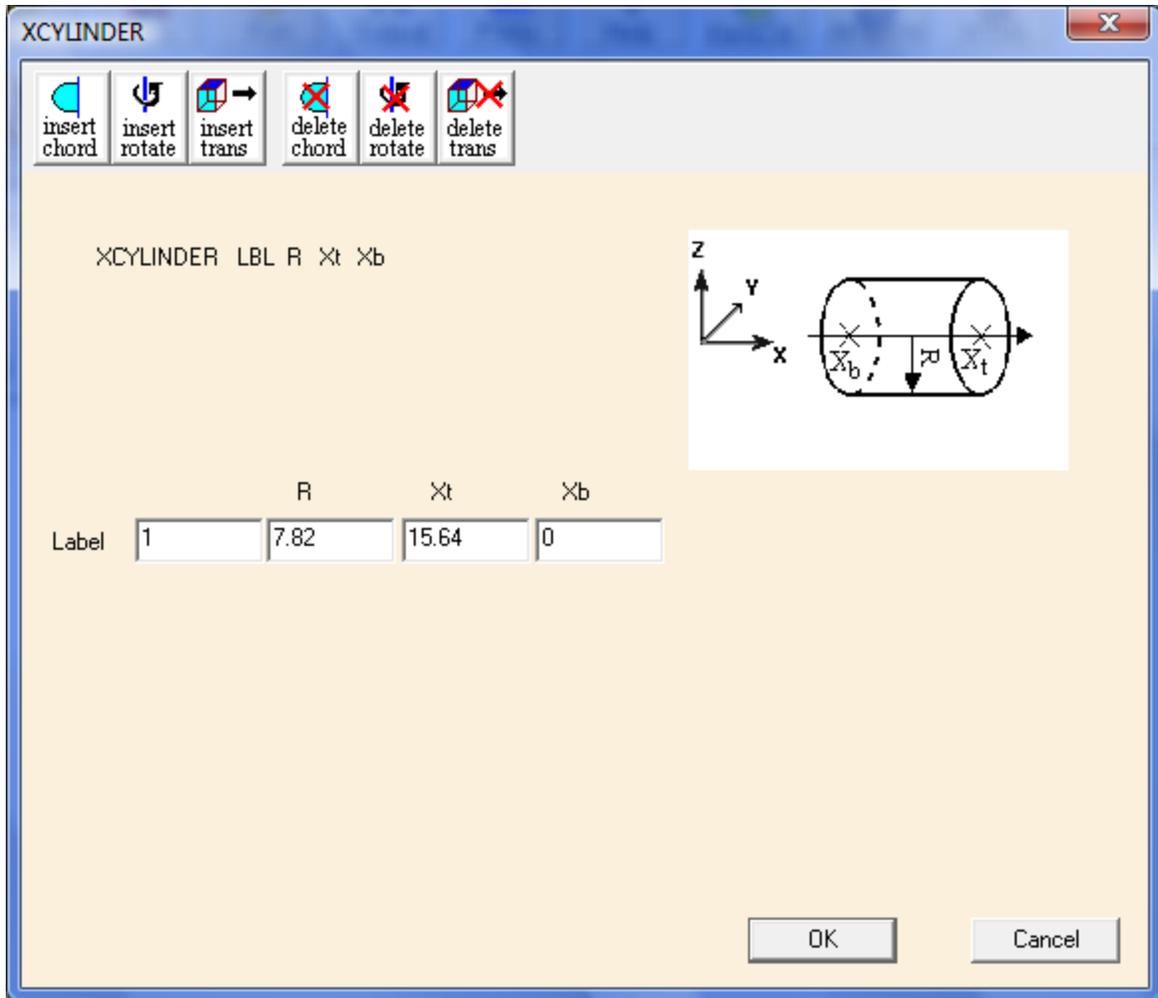
Fig. 72. KENO geometry input for cylinder along X-axis.

Save the changes in a new input file and execute SCALE. Again, because no changes were made to the physical system, the k_{eff} value for this run should be within 2 standard deviations of those from the previous runs. The k_{eff} calculated is 1.0212 ± 0.0020, which agrees within 2 standard deviations of the previous results.

The same approach applies to locating the origin of cuboids (X, Y, and/or Z), hexprisms, pentagons, elliptical cylinders, hoppers, and cones (Z only). You enter values of the maximum and minimum parameters to locate the origin on the axis. For all other shapes and/or axes, you enter the coordinates of the origin using the **insert trans** button to translate the origin.

4.3 KENO-VI BASIC GEOMETRY RULES

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

1. *Volumes are built in sections called units.* Each unit is independent of all other units and has its own coordinate system.

2. *Units are built using regions.* Regions are made using the KENO-VI geometry shapes. The unit boundary must fully enclose all defined regions in the unit. See example in Fig. 73.

3. *Regions may share boundaries and may intersect.* See examples in Fig. 73. (Note that intersecting regions are not allowed in KENO V.a.)

4. *Regions may be rotated.* See example in Fig. 73. (Note that rotations are not allowed in KENO V.a.)

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

6. *An array is an ordered stack of units.* The touching faces of adjacent units in an array must be the same size. Multiple arrays may be placed directly into a unit by placing them in separate regions. Arrays can also be placed within a unit using holes.

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

| Regions in a unit | Shared boundaries | Intersecting regions | Rotated region | Array of units |
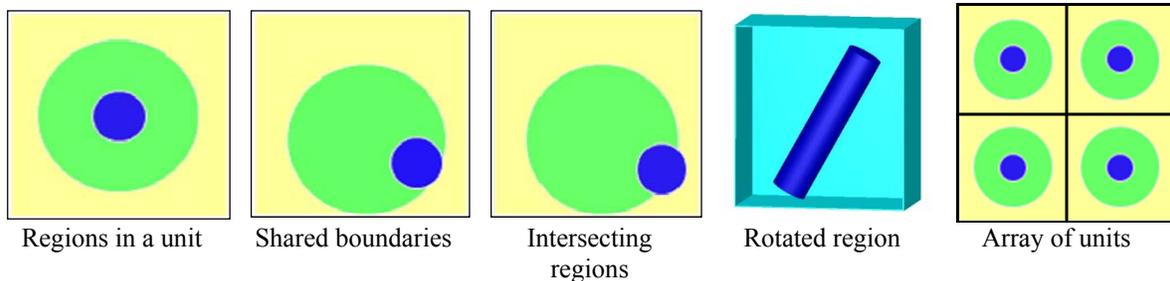
Fig. 73. Examples of KENO basic geometry rules.

4.4 GEOMETRIC ARRANGEMENTS

Geometric arrangements in KENO-VI are achieved in a manner similar to using a child's building blocks. Each building block is called a unit. Units are constructed of combinations of basic shapes. These shapes can be placed anywhere within a unit as long as they do not intersect the outer boundary of the unit.

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. This is the way to model reflected single units or containers with fissile solutions.

4.5   NESTED REGIONS IN A UNIT

To illustrate how to enter KENO geometry information for nested regions in a unit via GeeWiz, two sample problems will be used. 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 Plutonium Metal Cylinder

The reflected plutonium metal cylinder problem comes from LA-10860, p. 102. The fissile material is plutonium metal (density = 15.44 g/cc), with a maximum of 5.0 wt % ^{240}Pu. The remainder is ^{239}Pu. The core has a diameter of 5.72 cm with an H/D = 7.16, or a critical height of 40.96 cm. Surrounding the core is a graphite reflector (density = 1.60 g/cc), with a thickness of 17.78 cm in both the radial and axial directions (see Fig. 74).
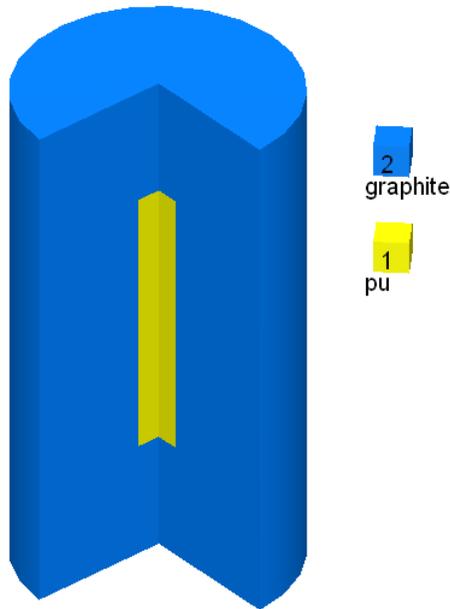
Fig. 74. Reflected Plutonium metal cylinder geometry.
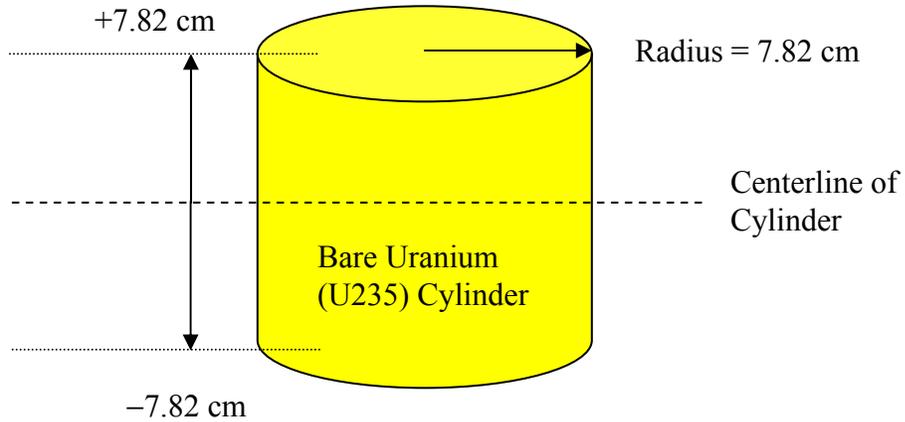
Based on the knowledge gained from completing Sect. 3 of the primer, you should be able to open a new file, enter the title, cross section, and material information. Specify the isotopic composition for mixture 1 as `pu` with 95 wt % ^{239}Pu and 5 wt % ^{240}Pu and a plutonium density of 15.44 g/cc. Mixture 2 should be `c-graphite` with a density of 1.6 g/cc.

For this geometry, there will be two shapes (or regions) in one unit, and that unit will be the global unit for the problem. First we will describe the inner cylinder containing the plutonium and then describe the outer cylinder containing graphite. Open the **Geometry** input form. Enter a comment in the **Comment** box and then select **Cyl** from the geometry buttons on the right. This opens the Cylinder input form with a default orientation along the Z-axis. Enter `2.86` for **R**, `40.96` for **Zt**, and `0.0` for **Zb**. This puts the origin at the bottom center of the core cylinder. This is now the origin for the unit and for all other shapes in the unit. This geometry screen for the core should look like Fig. 75. Then click **OK.** Notice that after entering the first region, the geometry screen now displays a check box for **Global Unit**. Check this box to make this the global unit for this case.
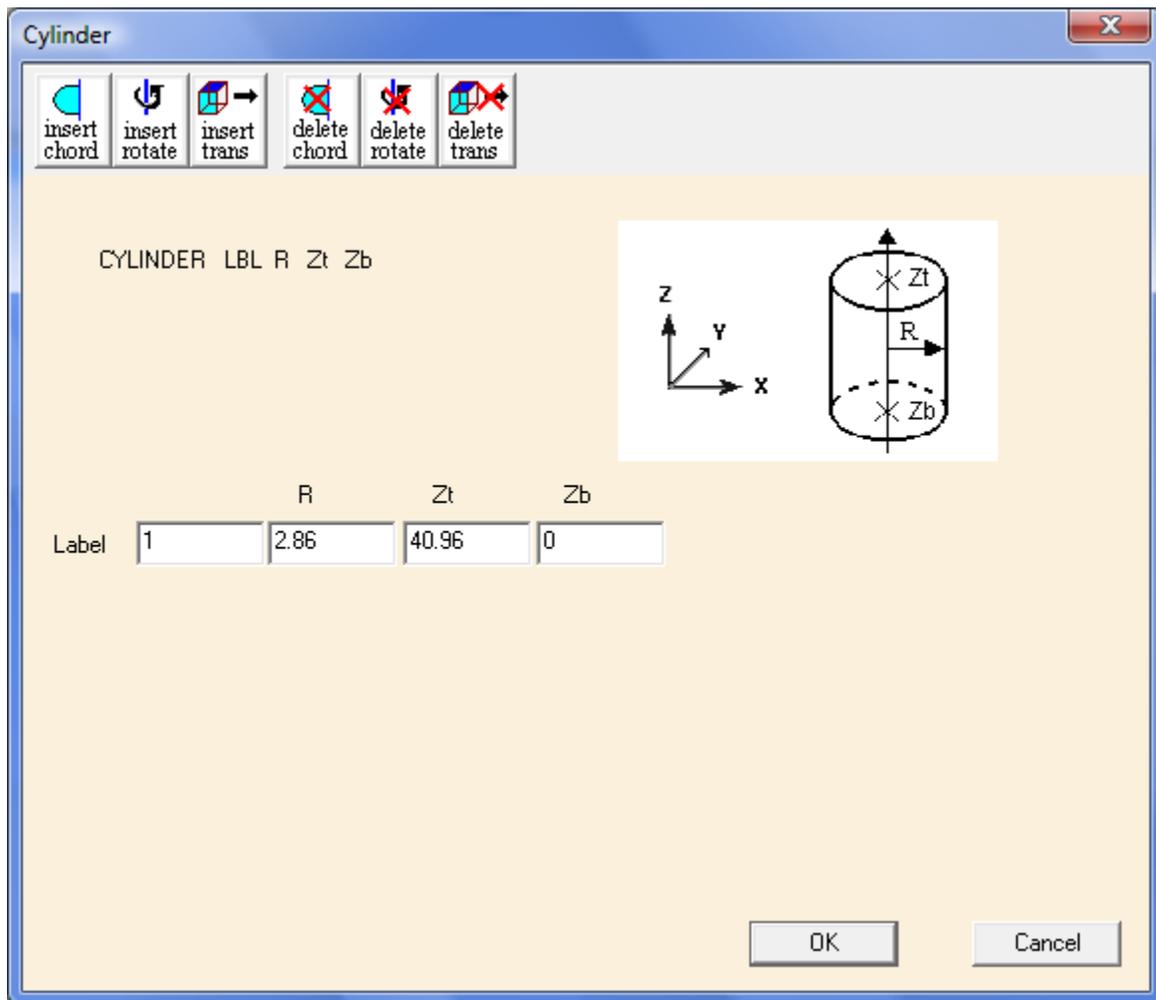
Fig. 75. Geometry description for region 1.

To enter the data for the reflector, select the **Cyl** button on the right. Enter `20.64` (= 2.86 + 17.78) for **R**, `58.74` (= 40.96 + 17.78) for **Zt**, and `−17.78` (= 0.00 − 17.78) for **Zb**. Then click **OK.**

The geometry shapes have been defined. Now the contents of the regions must be defined using media records. Each media record defines one region in the unit by combining one or more shapes using Boolean logic in a region definition vector. The media record also identifies a mixture to fill the region. Following the media records is the boundary record that defines the outer boundary of the unit using a region definition vector.

Click on the **Media** button to open the form to define region 1. The innermost cylinder containing plutonium is the first region. Click on `cylinder 1,` and then click on the **Inside** button. The number `1` appears in the **Region Definition Vector** at the bottom of the form. This information tells KENO-VI that region 1 is inside this cylinder. Next click on `1 pu` from the **Media** pull-down menu to fill the region with the plutonium mixture. The form should look like Fig. 76. Click **OK** to close the form.
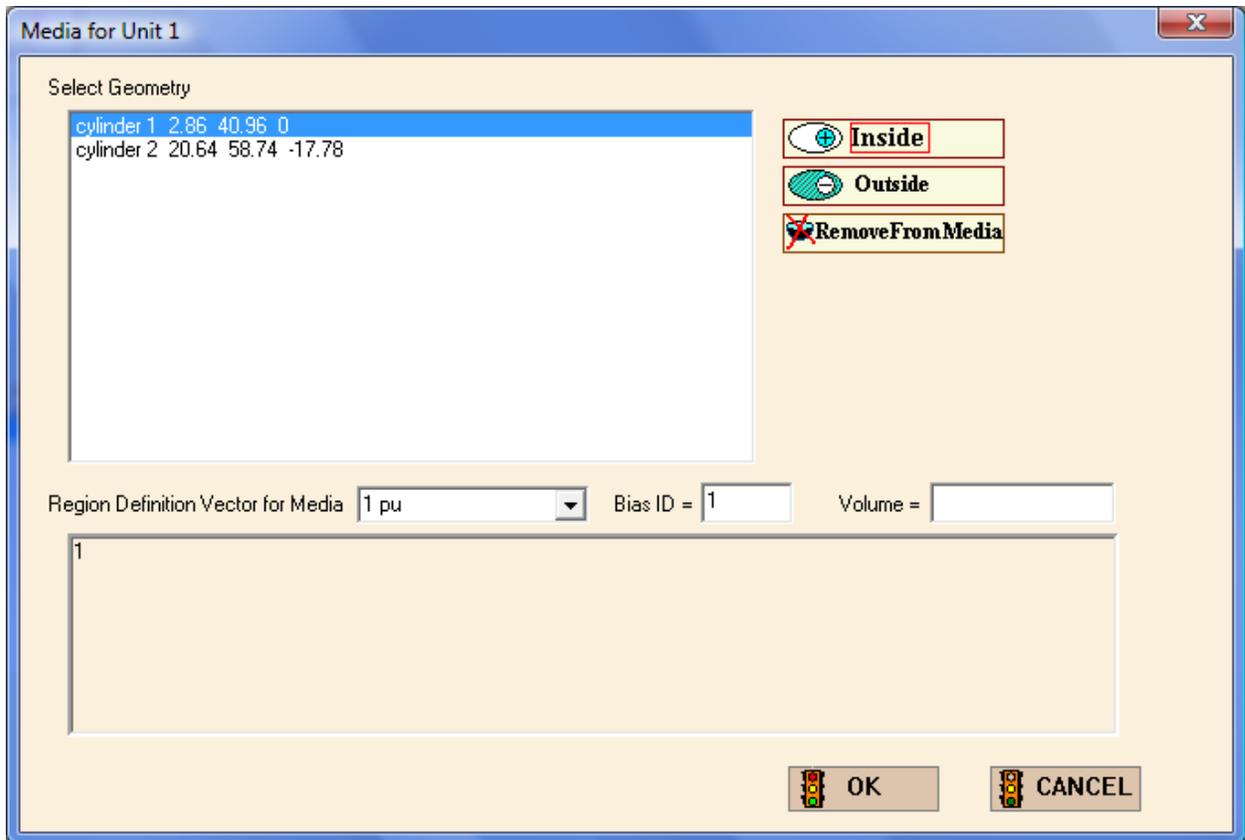
Fig. 76. Media form for region 1 in unit 1.

Repeat these steps to create the second region containing the graphite reflector. For this region the **Region Definition Vector** is `2 -1` which indicates that the graphite is **Inside** `cylinder 2` but **Outside** `cylinder 1`. Select `2 c-graphite` from the **Media** pull-down menu. Click **OK** when finished.

The final step is to define the outer boundary of unit 1. Click on the **Bound** button to open the Boundary form and specify **Inside** `cylinder 2`. This completes the geometry input (see Fig. 77).
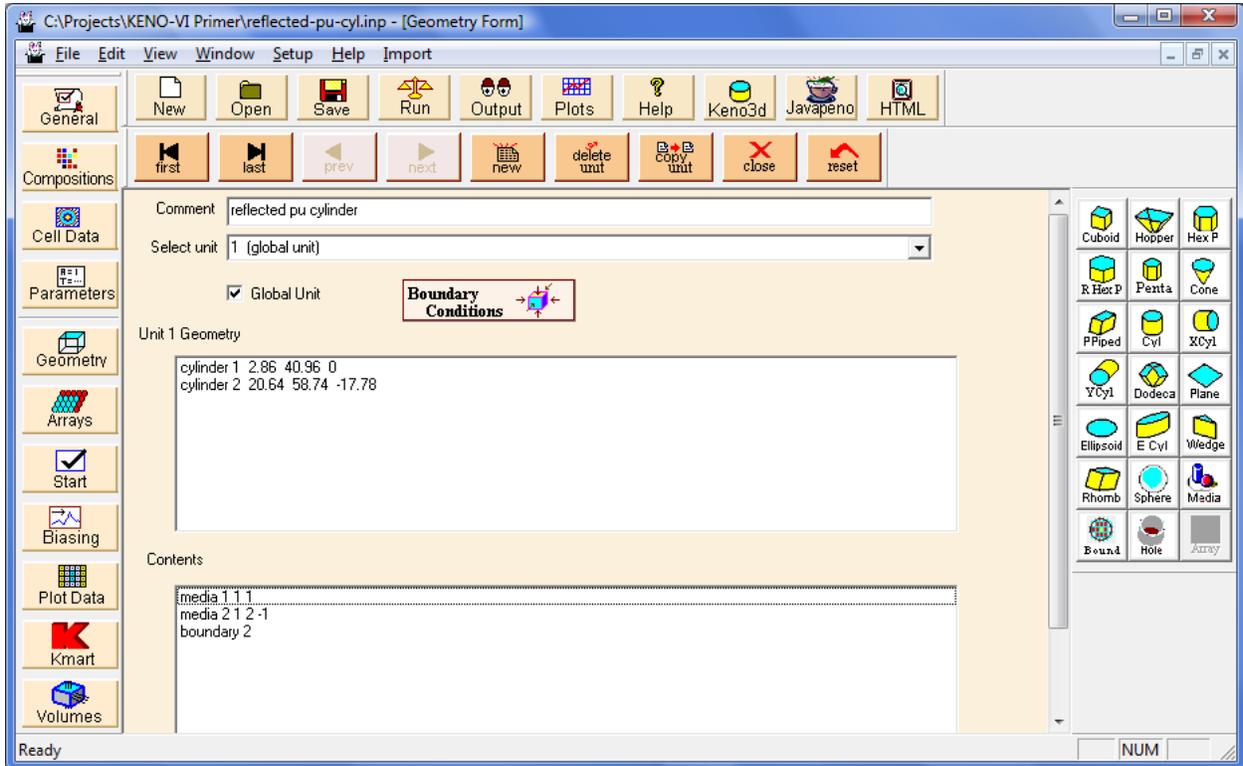
Fig. 77. KENO geometry input for reflected Plutonium cylinder.

Save the changes in a new input file and execute SCALE. As this is a critical experiment, the calculated k_{eff} value for this run would be expected to be near 1.0. However, the calculated k_{eff} value is 0.9849 ± 0.0026, which is 1.5% low. These experiments all seem to calculate low, which may be why they are not included in the International Criticality Safety Benchmarks Experiments Project (ICSBEP) benchmark set.

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 such that the outside edge of the cylinder touches two of the outside edges of the cube (see Fig. 78). The materials are ^{235}U metal (density = 18.74 g/cc), graphite (density = 1.65 g/cc), and water (default density of 0.9982 g/cc). The ^{235}U sphere radius is 7.0 cm; the graphite cylinder has a radius of 10 cm and a height of 20 cm. The water cube surrounding the cylinder is 22 cm on a side. The bottom of the cylinder is coincident with the bottom of the cube.

(12, 12, 12)

Sphere contains fuel and has a radius of 7.0 cm. The origin of the system is at the center of the sphere.

(-10, -10, -10)

Cylinder contains graphite and has a radius of 10.0 cm.
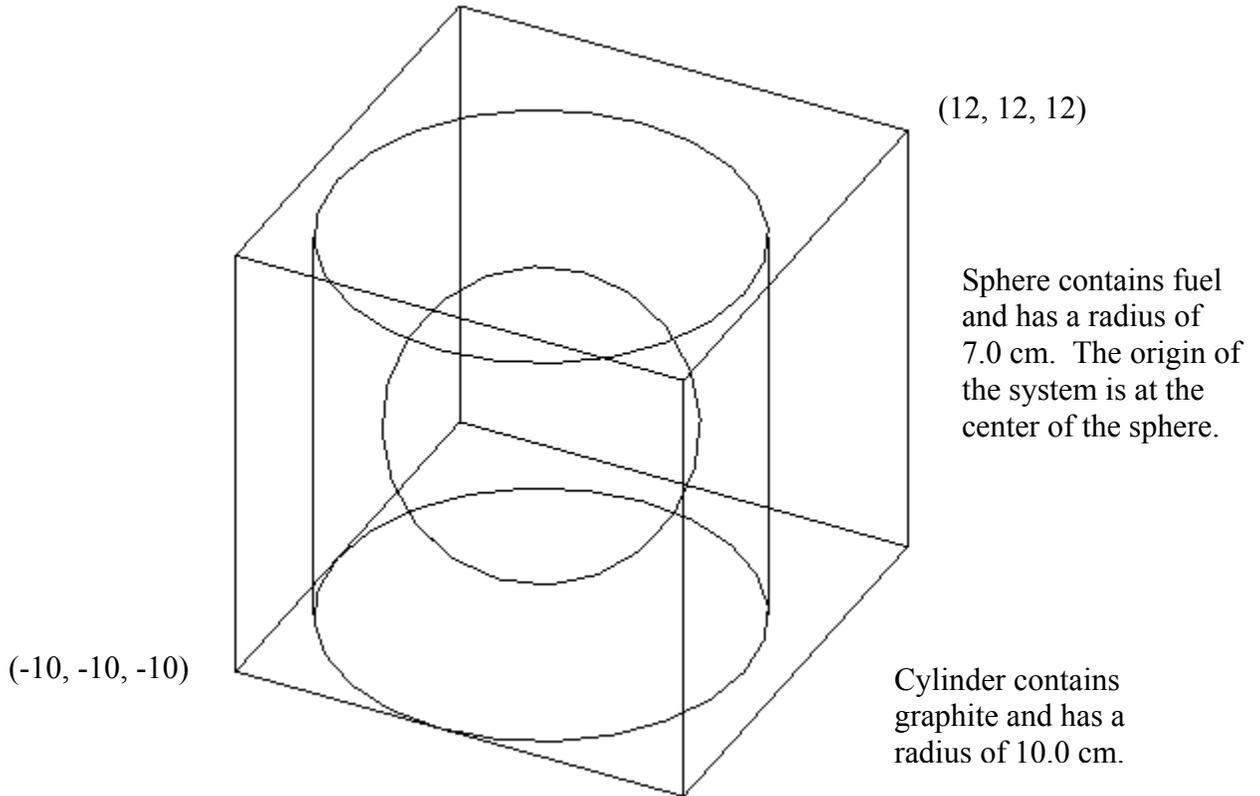
Fig. 78. ^{235}U sphere surrounded by graphite and water.

Two cutaway views of the geometry are shown in Fig. 79. The top 3 cm has been removed from both views to be able 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. Note that the graphite cylinder touches the outer edge of the water cube along the front edge and the left edge. Also note that the bottom of the cylinder is coincident with the bottom of the cube. In the second view, the graphite has been hidden so that the entire ^{235}U sphere can be seen.
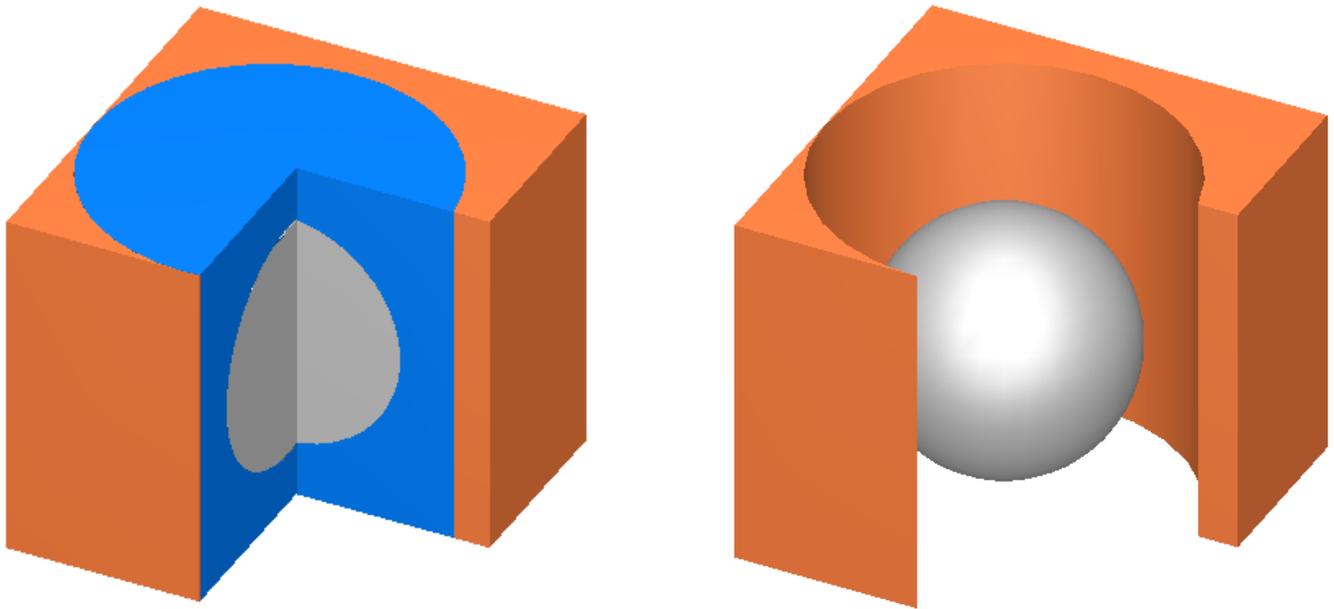
Fig. 79. Cutaway views of ^{235}U sphere inside graphite and water.

Using your knowledge from previous sections, you should be able to open a new file and enter the title, cross section, and material information for the three materials.

For this geometry, there will be three regions in one unit, and that unit will be the global unit for the problem. First, we will describe the sphere containing the uranium, 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. Select the **Geometry** button on the left, and enter a comment in the **Comment** box. Then select the **Sphere** button from the right toolbar. Enter `7.0` for **R**. This puts the origin at the sphere center; this is now the origin for the unit and for all other shapes in the unit. Then select **OK**, which takes you back to the geometry summary screen. Now check the **Global Unit** box. To enter the data for the graphite reflector, select the **Cyl** button from the right. Enter `10` for **R**, `10` for **Zt**, and `-10` for the **Zb**. This creates a cylinder centered around the sphere. Click **OK.** To enter the data for the outside water reflector, select the **Cuboid** button. The cuboid is slightly off center from the cylinder and sphere, so entering `12` for **+X, +Y,** and **+Z**, and `-10` for **–X, –Y,** and **–Z** creates a cube with three sides touching the cylinder and with the other three sides 2 cm beyond the cylinder. This completes the geometry input.

Click on the **Media** button to define the contents of the ^{235}U sphere. Click on `sphere 1` and the **Inside** button. Select `1 u-235` from the **Media** pull-down menu. Click **OK** to close form.

Repeat these steps to create the second region containing the graphite cylinder. For this region the **Region Definition Vector** is `2 -1`, which indicates that the graphite is **Inside** `cylinder 2` but **Outside** `sphere 1`. Select `2 c-graphite` from the **Media** pull-down menu. Click **OK** when finished.

Repeat these steps again to create the third region, which contains the water cuboid. For this region the **Region Definition Vector** is `3 -2,` which indicates that the water is

Inside `cuboid 3` but **Outside** `cylinder 2`. Select 3 `h2o` from the **Media** pull-down menu. Click **OK** when finished.

Click on the **Bound** button to define the outer boundary as **Inside** `cuboid 3`. When you are finished, the Geometry form should look like Fig. 80. Save the changes in a new input file. You can view the geometry by clicking on the **KENO3D** toolbar button. For more information on how to use KENO3D, refer to Sect. 9.3.
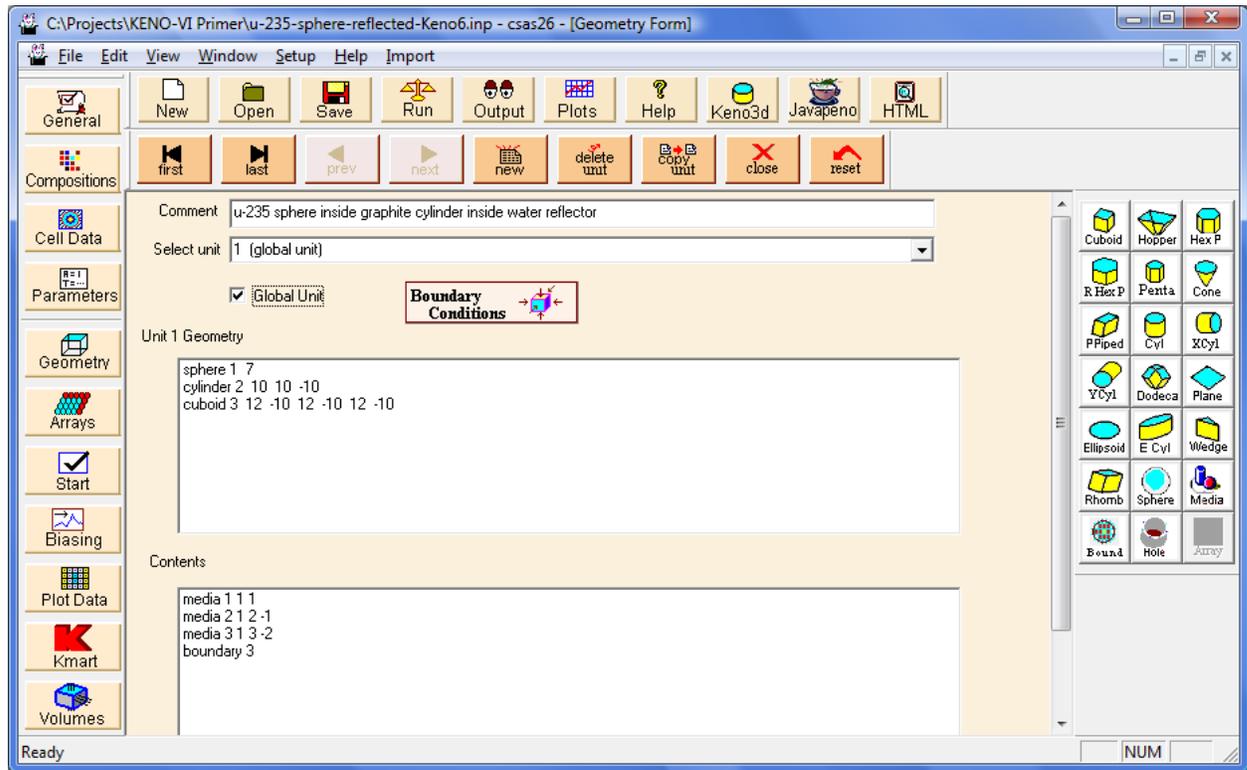
Fig. 80. KENO geometry input for ^{235}U sphere in graphite and water.

Click **Run** to execute SCALE. The calculated k_{eff} value is 1.0021 ± 0.0017.

4.6   ARRAYS

An array or lattice 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.

Lattices or arrays are created by stacking units that have a cuboid outer boundary. The dimensions of adjacent faces of adjacent units stacked in this manner must match exactly. See Fig. 81 for a typical example.

UNIT 1

UNIT 2
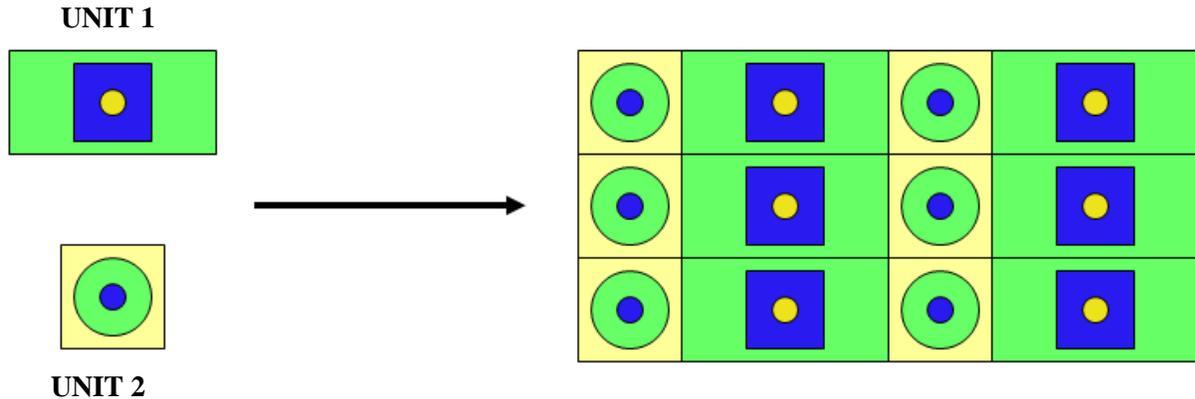
Fig. 81. Example of array construction.

The ARRAY option is provided to allow placing an array or lattice 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 Sect. 5. Arrays of dissimilar arrays can be created by stacking units that contain arrays (i.e., "arrays of arrays").

KENO-VI offers five different 3-D array types as shown in Fig. 82:
- square (or rectangular),
- hexagonal (or triangular),
- standard hexagonal,
- rotated hexagonal, and
- dodecahedron.

The focus of this section will be on simple arrays: that is, arrays containing a single unit or arrays with multiple units of nominally different sizes.
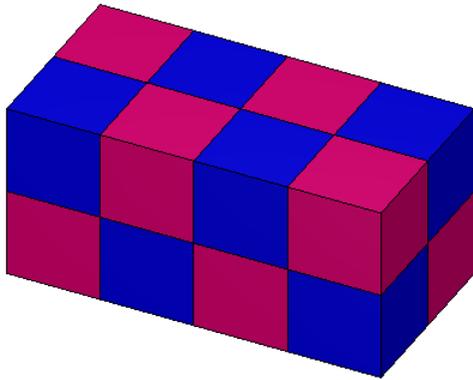
Square (or Rectangular)
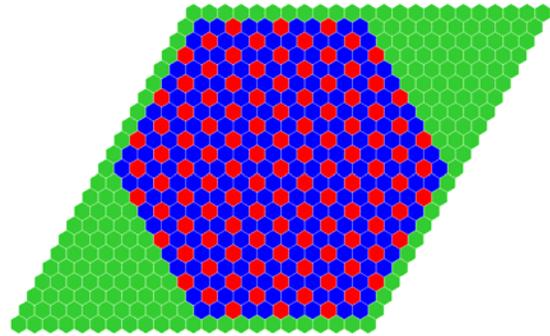
Hexagonal (or Triangular)
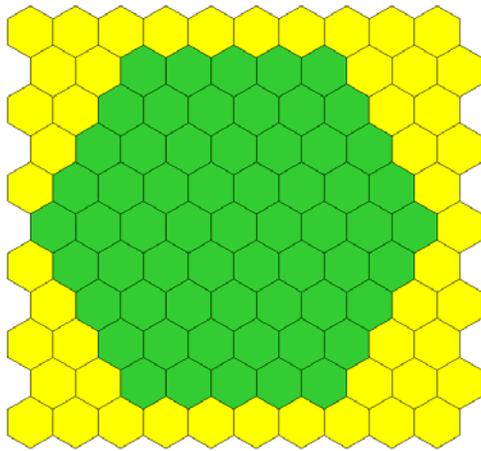
Standard Hexagonal
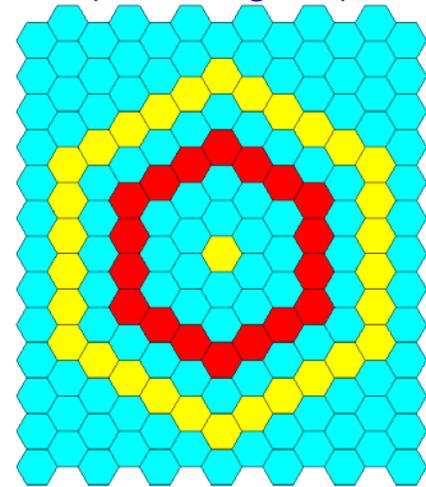
Rotated Hexagonal
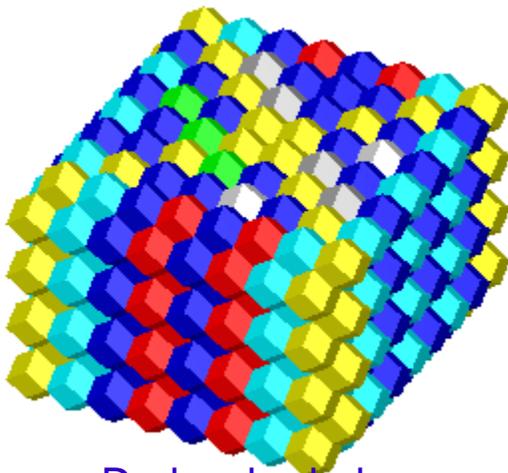
Dodecahedral

Fig. 82. KENO-VI array types.

4.6.1   Arrays with Single Units

This problem consists of a $2 \times 2 \times 2$ array of bare metal U(93.2) cylinders. Each cylinder has a radius of 5.748 cm and an outside height of 10.765 cm. The uranium has a density of 18.76 g/cc with 1 wt % ^{234}U, 93.2 wt % ^{235}U, 0.2 wt % ^{236}U, and 5.6 wt % ^{238}U. The pitch in the X- and Y-directions is 13.74 cm and 13.01 in the Z-direction. There are no external reflectors or other materials in the array. The geometry is shown in Fig. 83.

One of the requirements for a unit to be placed in an array is that the outside shape of the unit must be a cuboid, a hexprism, or a dodecahedron. As the constituents of the array are cylinders, they will need to be enclosed by cuboids to become part of the rectangular array. The dimensions of the cuboid are usually determined by the X-, Y-, and Z-pitch dimensions. For an array of single elements, the cuboid will have an X-dimension equal to the pitch in the X-direction, a Y-dimension equal to the pitch in the Y-direction, and a Z-dimension equal to the pitch in the Z-direction.
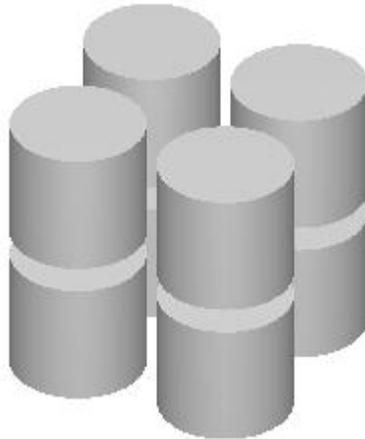
Fig. 83. 3-D view of $2 \times 2 \times 2$ array.

Using your knowledge from previous sections, you should be able to open a new file and enter the title, cross section, 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 contain the array of eight units. First we will describe the cylinder containing the uranium, and then describe the cuboid enclosing the cylinder to create an element of the array. These are nested shapes and are entered in the same manner as for the previous example. However, we must be careful when entering the Z data for both the cylinder and the cuboid. Once we have entered the Z or height information for the cylinder, we will have fixed the coordinate system and its origin. There are three easy choices for the location of the origin: at the bottom center of the

cylinder, at the middle of the cylinder, or at 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, let's choose the origin to be located at the middle of the cylinder, because we want to center the cylinder in the surrounding cuboid. The **Zt** for the cylinder will be `5.3825` and the **Zb** will be `-5.3825` (see Fig. 75). For the cuboid, the origin is located in the middle in the X-, Y-, and Z- directions so the dimensions will be half the pitch: **+X** and **+Y** will be `6.87,` **–X** and **–Y** will be `−6.87,` **+Z** will be `6.505,` and **–Z** will be `-6.505.` These parameters are sufficient for describing the single array element as shown in Fig. 84.

3-D ISOMETRIC VIEW

(Z= 6.505)

(Z= 5.3825)

(6.87, 6.87)

R = 5.748

(Z= -5.3825)

(Z= -6.505)

(-6.87, -6.87)

X-Z Side View
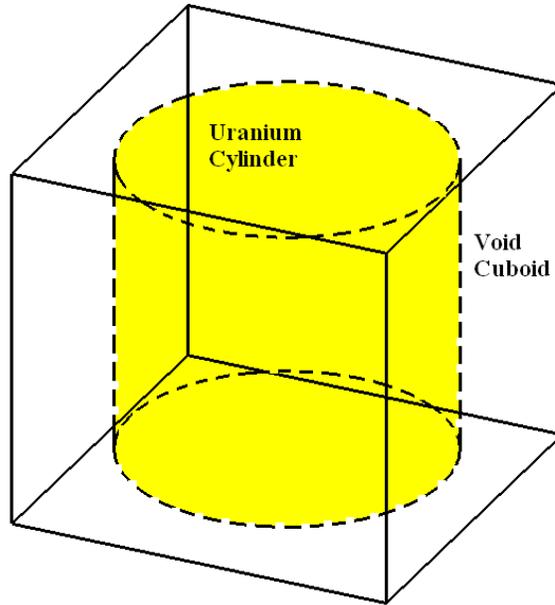
X-Y Top View
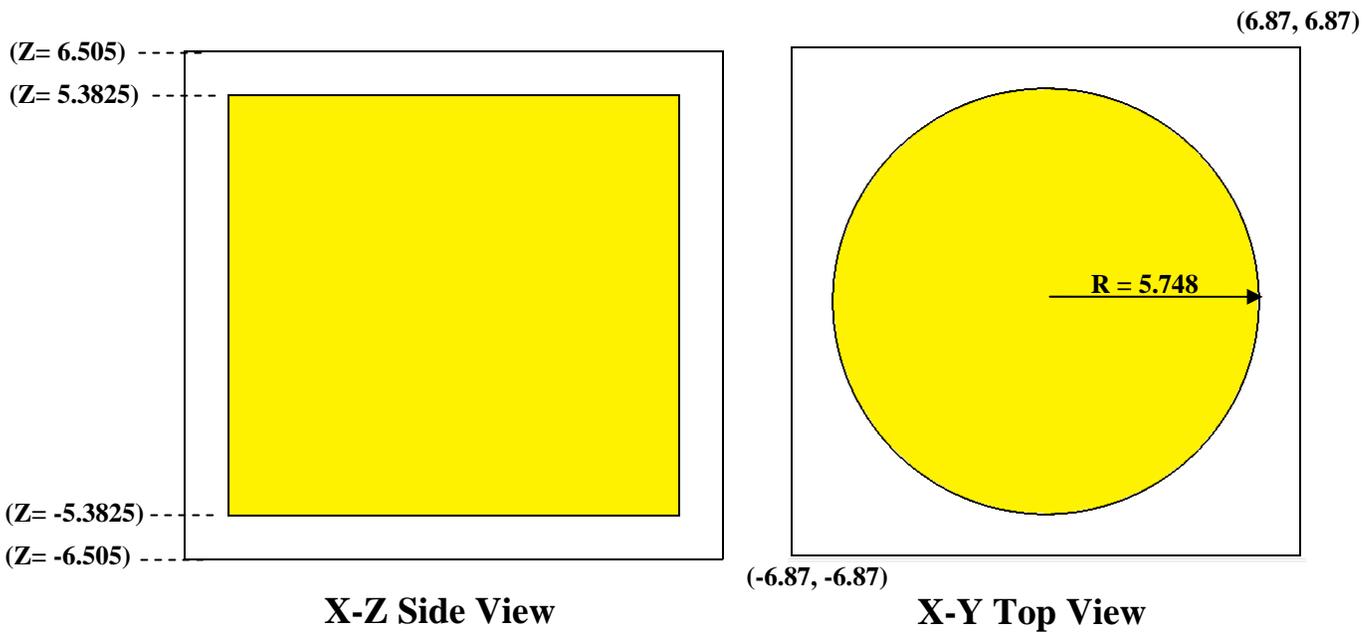
Fig. 84. Single array element (cuboid enclosing a cylinder).

Describe the geometry by opening the **Geometry** window. Enter a comment in the **Comment** box. Then select the **Cyl** button on the right. Enter `5.748` for **R**, `5.3825` for **Zt**, and `-5.3825` for **Zb**. This puts the origin at the center of the cylinder; this is now the origin for the unit and for all other regions in the unit. To enter the data for the enclosing cuboid select the **Cuboid** button. Enter `6.87` for **+X** and **+Y**, `-6.87` for **-X** and **-Y**, `6.505` for **+Z**, and `-6.505` for **-Z**. This creates a cuboid around the cylinder with the proper spacing. Note that GeeWiz uses two shortcut keys to reduce the effort for the user. The "P" key (i.e., "P" = "plus/minus") alternates positive and negative repeating values in consecutive fields, while the "R" key (i.e., "R" = "repeat") repeats the value in the current field. For example, you can enter 6.87 for **+X** and then press the "P" key three times to fill the **–X**, **+Y**, and **–Y** fields for this cuboid. Likewise you can enter `6.505` for **+Z** and then press the "P" key to fill the **–Z** field.

Use the **Media** form twice to define the two regions (uranium cylinder and void inside the cuboid and outside the cylinder). Click on the **Bound** button to designate the cuboid as the outer boundary. When you are finished, the Geometry form should look like Fig. 85. Next we need to enter the **Array** data.
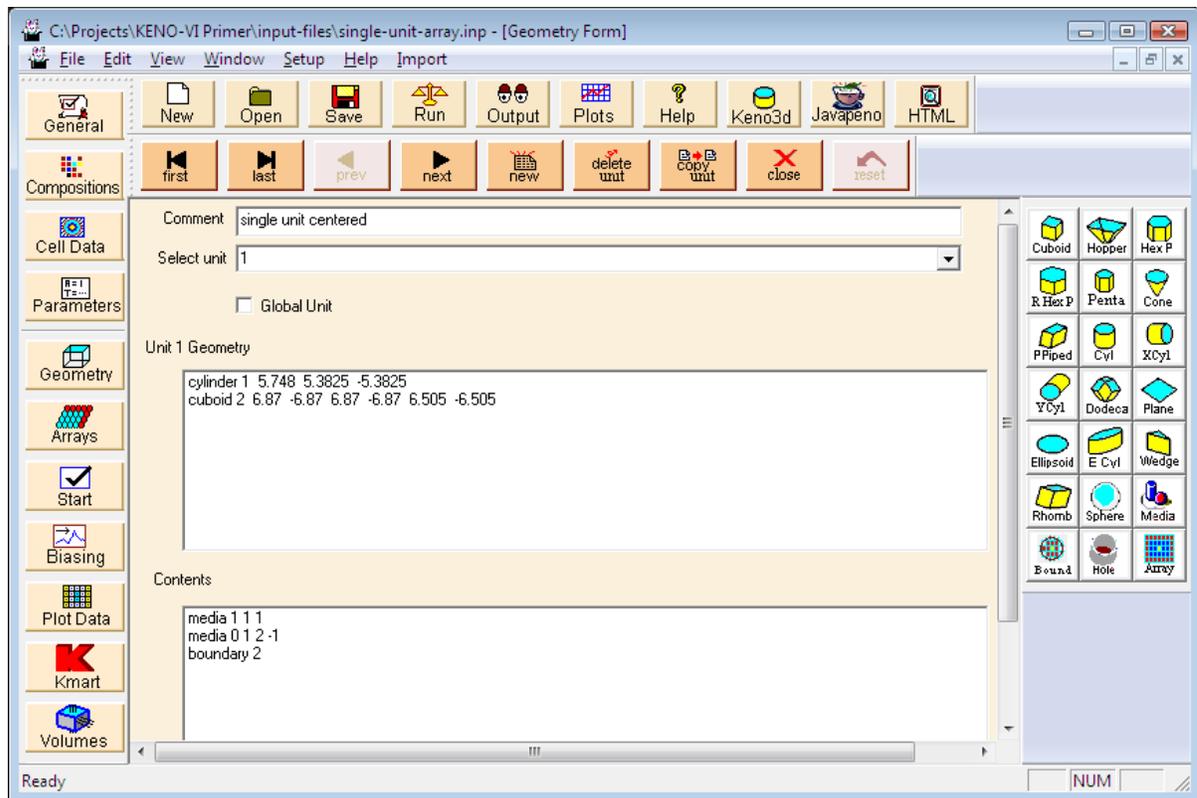
Fig. 85. Single unit geometry for array.

Now describe the array by opening the **Arrays** window (Fig. 86). To enter the array data, first click on **Create** on the array toolbar. This will open a window where you will enter the dimensions of the array. For this problem, the array will be contained in the global

unit, so check the **Global Array** box. Designation of the global array is necessary for some of the advanced output edits in KENO-VI. For array **Type,** select `Square`. The array is 2 units in the X-direction, 2 units in the Y-direction, and 2 units in the Z-direction, so enter `2` for **NUX**, `2` for **NUY** and `2` for **NUZ**. Notice that the window has already identified the array as number 1. At the bottom there is a box that allows you to fill the array with a given unit. When you only have one unit, as is the case here, then that unit number will be in the box. With more than one unit, you should select the unit number for the unit that appears most often in the array. This is the starting arrangement, which you can modify as desired. The screen should look like Fig. 87.
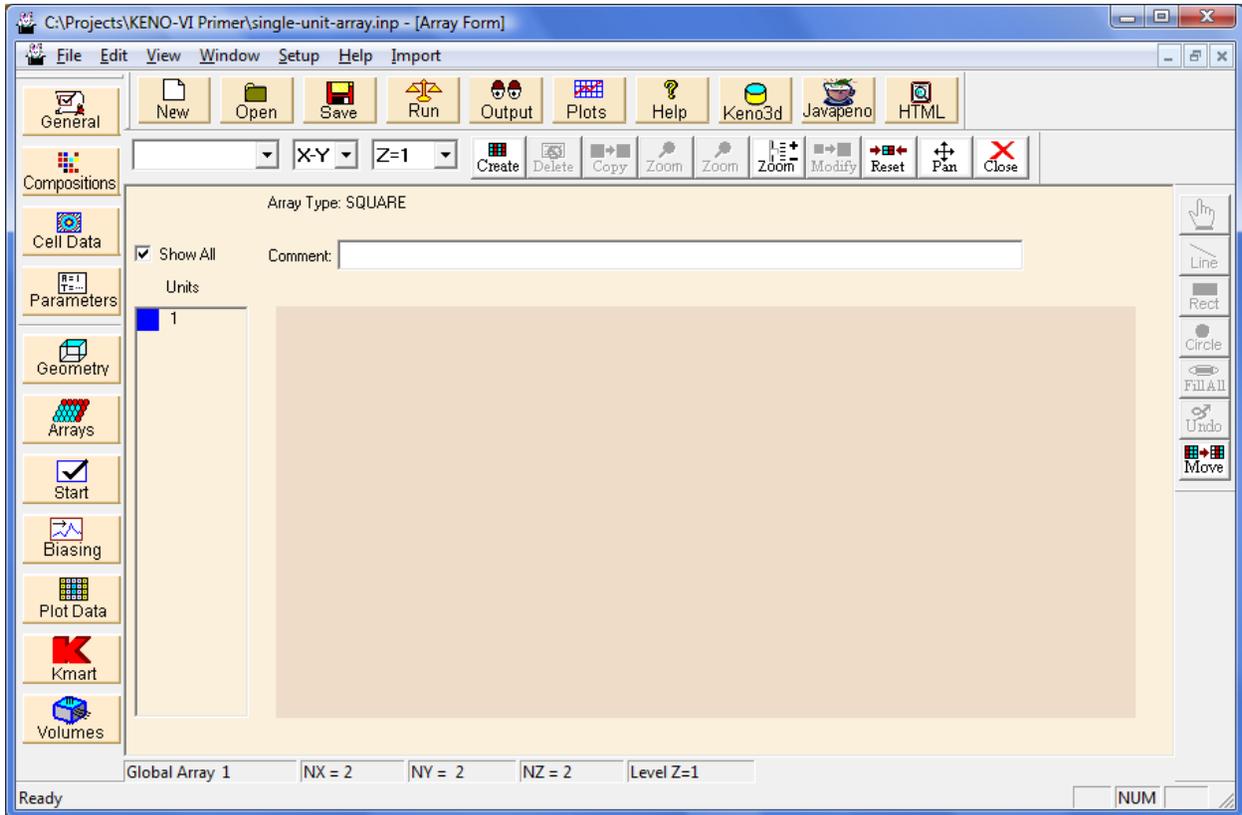
Fig. 86. Array screen in GeeWiz.
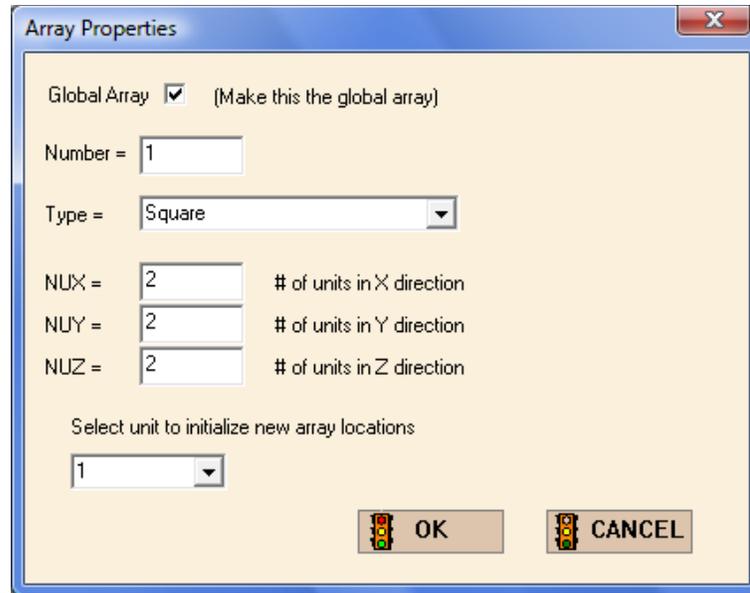
Fig. 87. Array properties for 2 × 2 × 2 array.

Then click **OK**. This will display an X-Y grid of the Z=1 plane on the screen. The grid shows a 2 × 2 arrangement filled with unit 1 for the X-Y plane, as indicated on the screen. As there are two X-Y planes, both levels, Z=1 and Z=2, are filled with unit 1. The screen display should look like Fig. 88. After you have checked your input, click on the **Close** button on the array toolbar. This will save the array information.
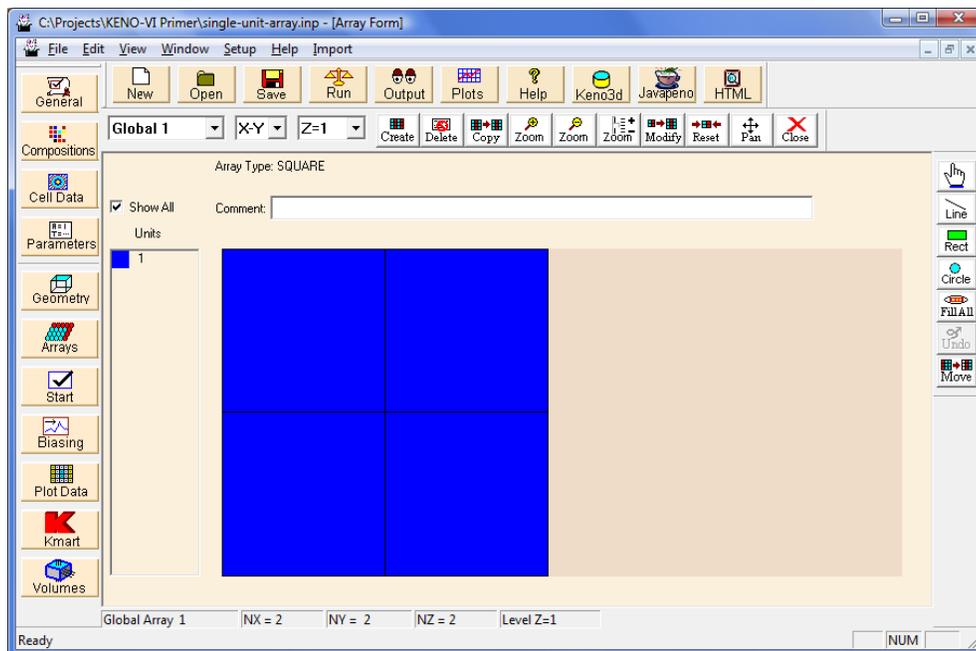
Fig. 88. Array form for 2 × 2 × 2 array.

Return to the Geometry form, and click the **new** button to create unit 2. This unit will be the global unit that contains array 1. Enter a comment such as `2x2x2 2c8 array`.

95

Click on the **Cuboid** button to create a cuboid to contain array 1. The dimensions of the cuboid should be the same as those of the array. Because this is a 2 × 2 × 2 array, we simply double the X-, Y-, and Z-dimensions of unit 1. If we center the cuboid on the origin, the outer dimensions of the cuboid will be ±13.74 for ±**X** and ±**Y** and ±13.01 for ±**Z**. Remember when entering the dimensions on the Cuboid form that you can use the "P" key to repeat alternating positive and negative values. After the dimensions are entered, press the **OK** button to close the cuboid form.

Next click the **Array** button to define the placement of array 1 in the cuboid. At the top of the Array form, select Array 1 from the pull-down menu. Make sure that cuboid 1 is highlighted and click on the **Inside** button to place array 1 inside cuboid 1. The next step is to specify the input data for the **PLACE** keyword to designate where the array is placed within the cuboid. This keyword is followed by six numerical values. The first three values are integers that indicate which element of the array (**nx**, **ny**, and **nz** are the X, Y, and Z array positions). The last three values are real numbers that are the **X**, **Y**, and **Z** coordinates for placing the origin of the unit in the designated array element in the surrounding geometry unit. In this example, we will place the unit in element (1,1,1) of the array. The origin of unit 1 is located at the center of that unit. The origin of the unit 2 cuboid where array 1 is being placed is at the center of the cuboid. Because the array is a 2 × 2 × 2 array of identical units, the origin of element (1,1,1) will be half the pitch from the origin of the unit 2 cuboid: (-6.87, -6.87, -6.505). Enter these values following the **PLACE** keyword on the form: 1, 1, 1, -6.87, -6.87, -6.505. The Array form should look like Fig. 89. Click **OK** to close the Array form.
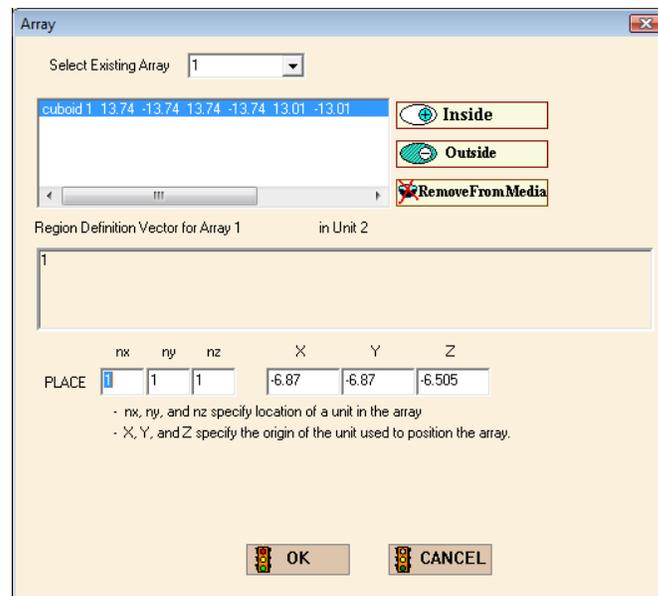
Fig. 89. Array form to place array in geometry.

Click on the **Bound** button to enter the boundary data for unit 2. Make sure that `cuboid 1` is highlighted and click on the **Inside** button to set the boundary inside cuboid 1. Click **OK** to close the Boundary form.

Check the **Global Unit** box. The Geometry form should look like Fig. 90. Close the Geometry form, save the input file, and execute SCALE. The calculated k_{eff} for this critical experiment is 0.9965 ± 0.0016.
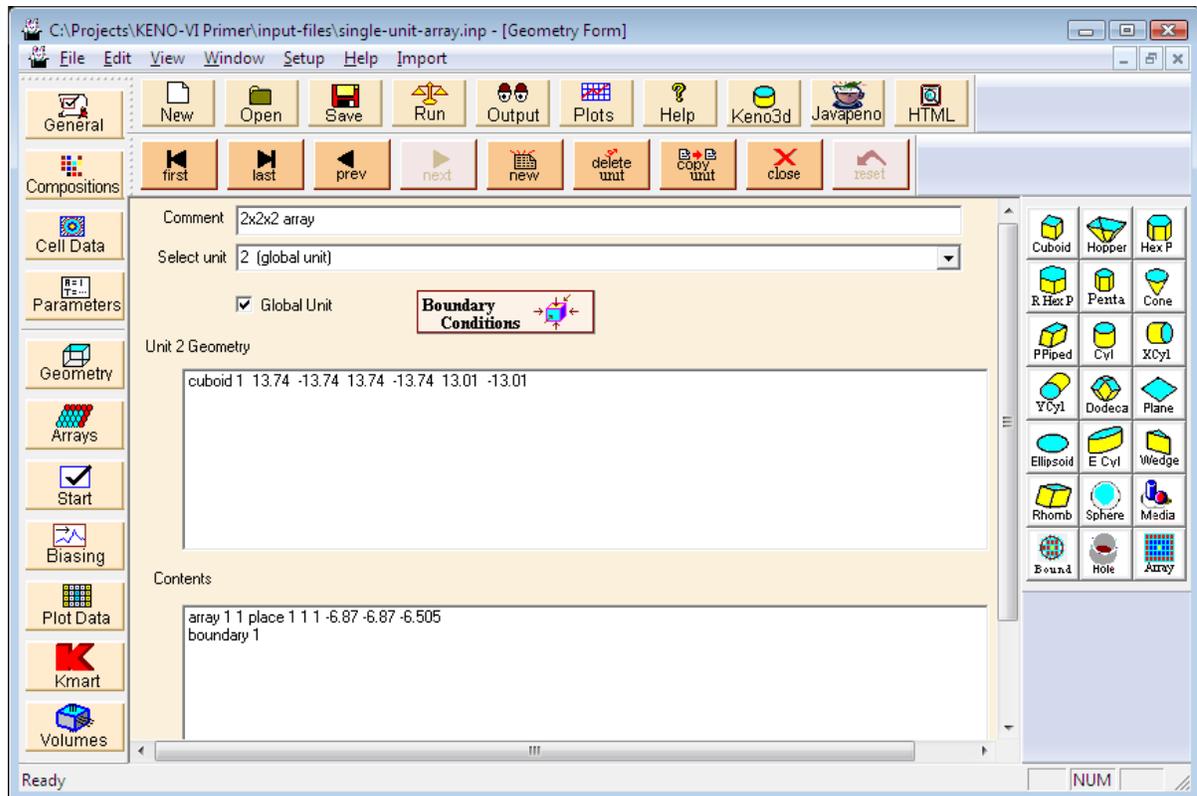
Fig. 90. Geometry form with array record to place array in geometry.

4.6.2   Arrays with Multiple Units of Different Nominal Sizes

This problem consists of a stack of six cylindrical disks covered by a square plate on the top (Fig. 91). There are two types of disks. One contains graphite (density = 1.65 g/cc), and is 10 cm in radius and 2.5 cm thick. The second one contains U (93.2) and is 10 cm in radius and 4 cm thick. The uranium has a density of 18.74 g/cc with 1 wt % ^{234}U, 93.2 wt % ^{235}U, 0.2 wt % ^{236}U, and 5.6 wt % ^{238}U. The bottom disk, which is graphite, is followed by uranium, graphite, uranium, graphite, and uranium. The top plate is aluminum of nominal density, 30×30 cm square, and 3 cm thick. There are no external reflectors or other materials in the array.

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 in the X and Y dimensions as the top plate. Thus, the length of the X and Y sides of the cuboid must be 30 cm. Notice that there is no constraint on the Z-dimension of the cuboid, because there are no adjacent faces in that direction. However, to stack the cylinders so that they touch (i.e., no gaps between them), the height of each cuboid surrounding a cylinder must be the same as the height of the cylinder.
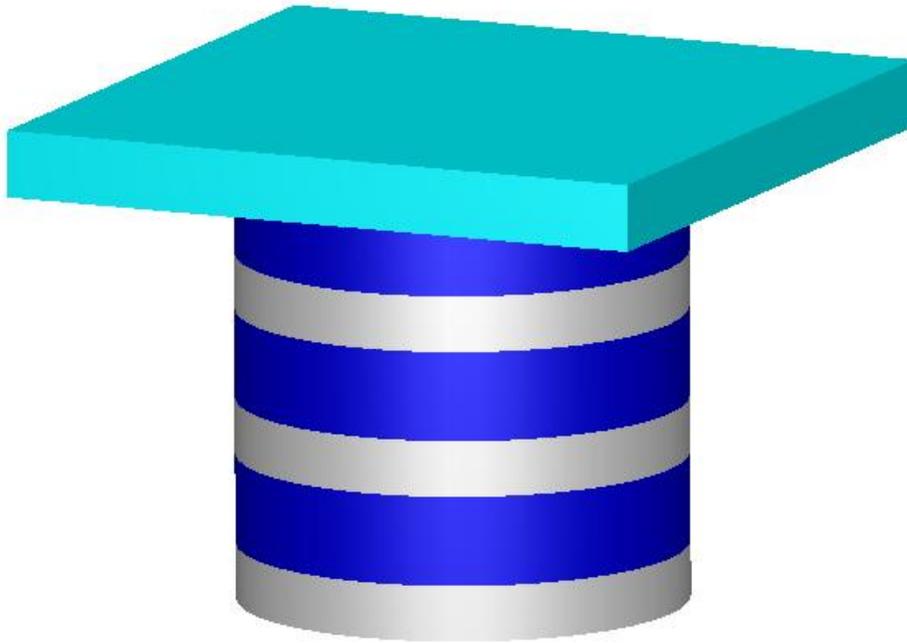
Fig. 91. Geometry of stack.

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

Describe the geometry by opening the **Geometry** screen. The first unit will be the graphite cylinder surrounded by a void cuboid so that the unit can be stacked in an array. Enter a comment in the **Comment** box, and then select the **Cyl** button to define the graphite cylinder. Enter `10` for **R**, `2.5` for **Zt**, and `0.0` for **Zb**. This puts the origin for the unit and for all shapes in the unit at the center of the cylinder. With the origin at the center of the cylinder, the Z-coordinates for the enclosing cuboid become symmetric. To enter the data for the enclosing cuboid, select the **Cuboid** button and enter `15` and `-15` for +**X** and –**X** and for +**Y** and −**Y**. Enter `2.5` and `0.0` for +**Z** and −**Z**. This creates a cuboid around the cylinder with the proper spacing. When the geometry input values alternate sign as they do here, remember that you can use the "P" shortcut key in GeeWiz to repeat the first value entered with alternating plus/minus signs. For example, enter **15** for +**X** and then press "P" three times. GeeWiz will fill the –**X**, +**Y**, and –**Y** fields with values of `-15`, 15, and `-15`,

respectively. Use the **Media** form twice to define the two regions. The first region is **inside** `cylinder 1` and filled with media `1 graphite`. The second region is **inside** `cuboid 2` and **outside** `cylinder 1` and filled with media `0 void`. Click on the **Bound** button to designate the outer boundary as **inside** `cuboid 2`.

Now describe the uranium cylinder surrounded by a void cuboid. Click the **new** button on the geometry toolbar to create unit 2. Again, enter a comment in the **Comment** box and then select the **Cyl** button. Enter `10` for **R**, `4` for **Zt**, and `0` for **Zb**. To enter the data for the enclosing cuboid, select the **Cuboid** button and enter `15` and `-15` for **+X** and **–X** and for **+Y** and **−Y**. Enter `4` and `0` for **+Z** and **−Z**. This creates a cuboid around the cylinder with the proper spacing. Use the **Media** form twice to define the two regions (uranium cylinder and void cuboid). Click on the **Bound** button to designate the cuboid as the outer boundary.

Now describe the aluminum plate by clicking the **new** button on the geometry toolbar to create unit 3. Enter a comment in the **COM =** box; then select the **Cuboid** button and enter `15` and `-15` for **+X** and **–X** and for **+Y** and **−Y**. Enter `3` and `0` for **+Z** and **−Z**. Click on the **Media** button to define the contents of the cuboid as `3 al`. After you close the Media form, click on the **Bound** button to designate the cuboid as the outer boundary.

Next we need to enter the array data. Click **Arrays** on the left toolbar, and then click the **Create** button on the array toolbar. This will open a window where you will enter the dimensions of the array. For this problem, the array will be contained in the global unit, so check the **Global Array** box. Select `Square` as the array **Type**. The array is 1 unit in the X-direction, 1 unit in the Y-direction, and 7 units in the Z-direction, so enter `1` for **NUX**, `1` for **NUY,** and `7` for **NUZ**. Notice that it has already identified the array as number 1. In the bottom box select unit `1` to initialize all array positions. The starting arrangement, which you will later modify, will be a $1 \times 1 \times 7$ array filled with unit 1, as indicated in the bottom box. Then click **OK**. This will display a grid of the first plane on the screen in the left-hand corner. However, it only shows as a 1×1 array because you are looking at the **X-Y** plane.

Click on **X-Z** plane to display the 1 × 7 array as shown in Fig. 92. Make sure that the **Show All** box has been checked to display all the units previously described in the geometry section. Also, depending on the array size, the entire array may not fit on the screen. You can use the **Zoom +** (fewer units on screen) or **Zoom –** (more units on screen) buttons located on the array toolbar to adjust the display as desired.
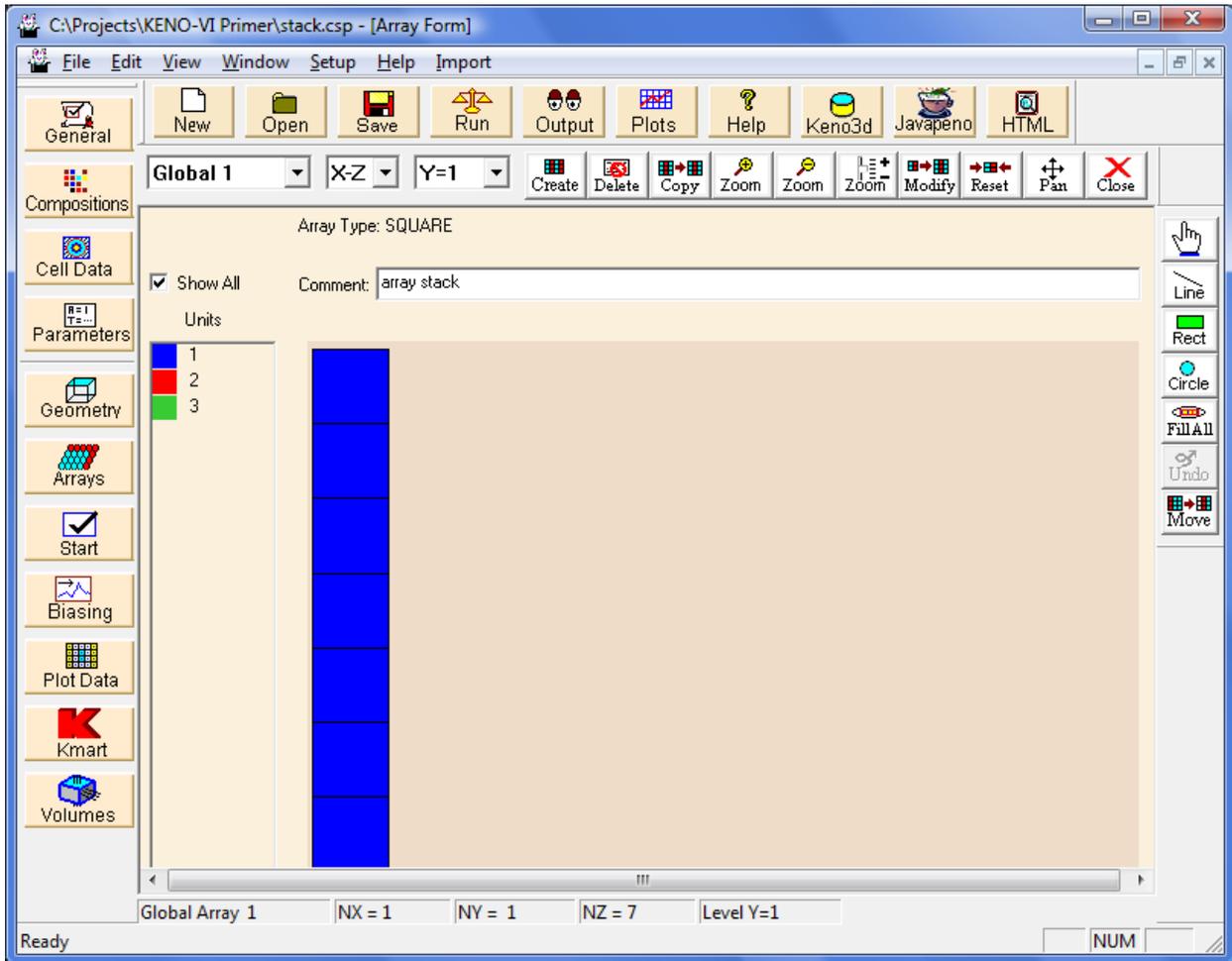
Fig. 92. Initial array screen for stacked disks problem.

Because we have three different units, we must tell GeeWiz which unit to put in each array position. First, click on the unit you want to put in the array. In Fig. 92 the array is initially filled with unit 1, so we will select unit 2 to place it in its proper positions. Then select the pointer button (hand) on the right. Then click on the second location from the bottom; next, the fourth location from the bottom; and lastly, the sixth location from the bottom. This puts unit 2 in these locations in the array as shown in Fig. 93.
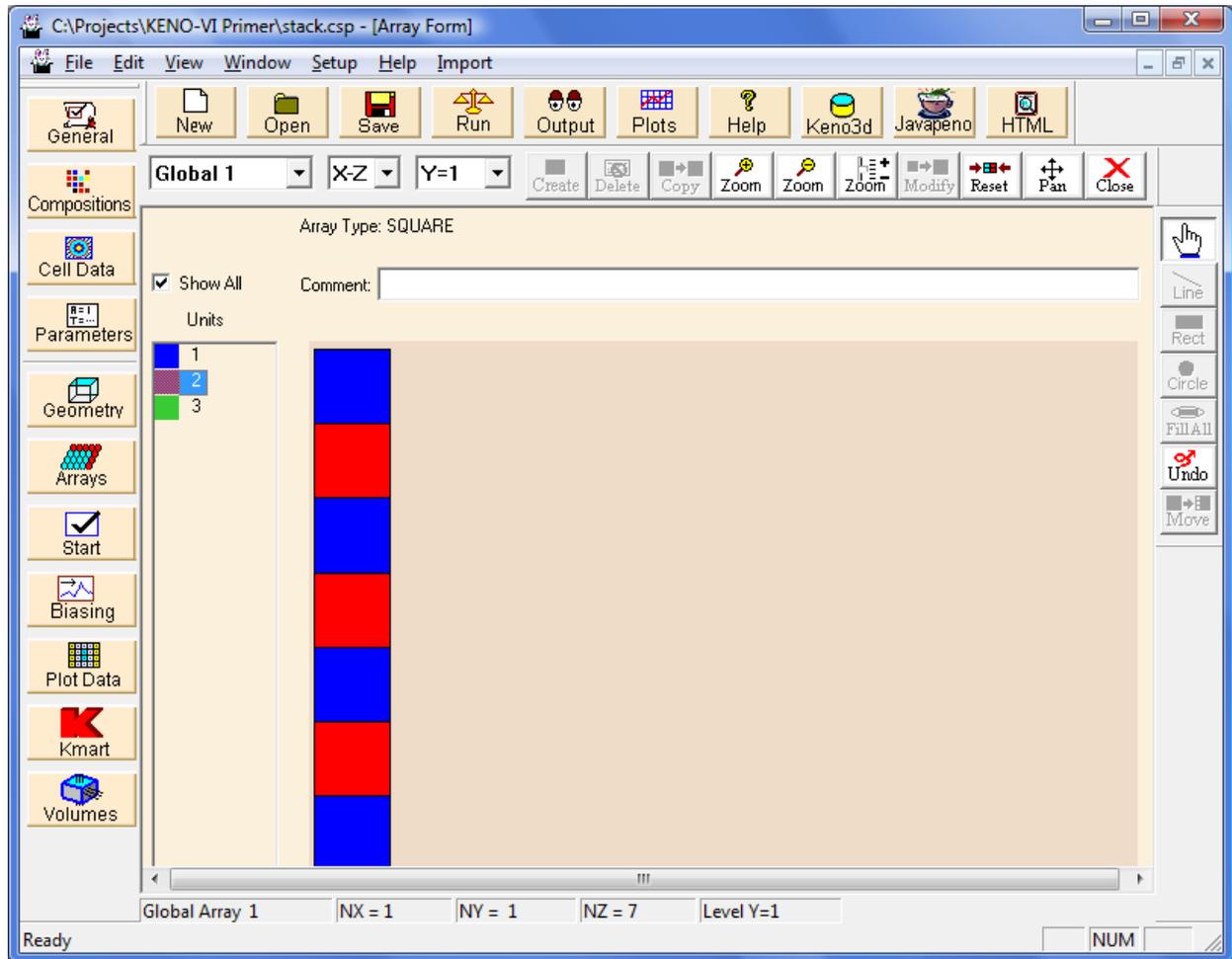
Fig. 93. Placing unit 2 in the array.

Now repeat the process with unit 3, placing it in the top position (or the seventh from the bottom). When finished, click again on the pointer button to toggle off unit placement. When completed, the array screen should look like Fig. 94. After you have checked your input, then click on the **Close** button on the array toolbar. This will save the array information.
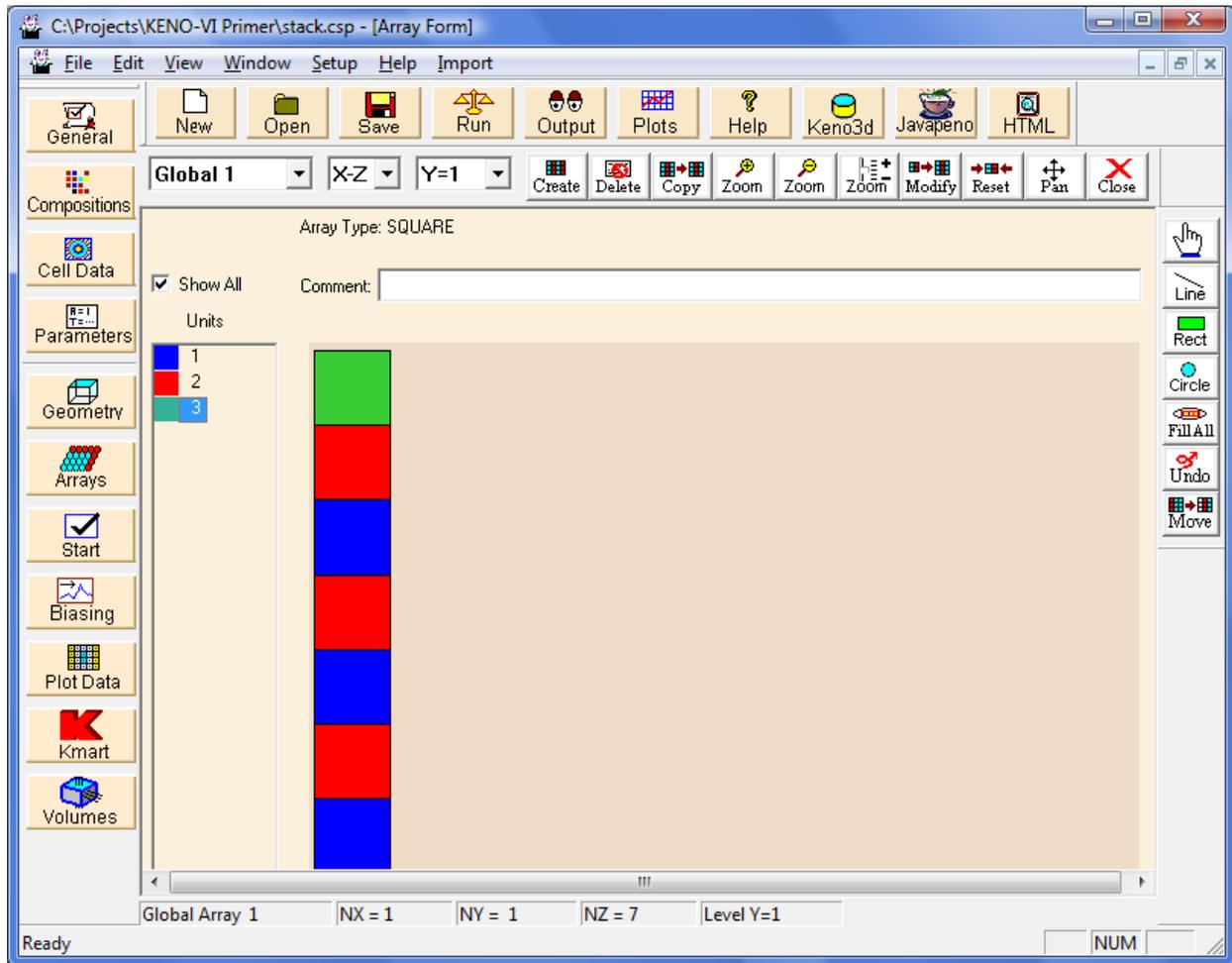
Fig. 94. Completed array screen for stacked disks problem.

Click on the **Geometry** button to return to the geometry screen. Click **new** to create a new unit 4 that will contain the array. Click on the **Cuboid** button on the right side of the geometry screen to define a cuboid that will contain the array. Note that the geometry shape that contains an array must have dimensions that are exactly the same as or smaller than the outer dimensions of the array. Otherwise there will be undefined volumes inside the shape but outside the array. Enter the dimensions for the cuboid: +/-15 for **+/-X** and **+/-Y**, 22.5 for **+Z**, and 0 for **-Z.** Click **OK** to close the cuboid form.

Click on the Array button on the right side of the geometry screen. **Select existing array** 1. Click on cuboid 1, and then click the **Inside** button. For the **PLACE** data, enter 1

1 1 for **nx**, **ny**, and **nz**. The origin of the bottom disk (unit 1) is located at its bottom center, so enter 0 0 0 for **X**, **Y**, and **Z** to place the bottom of the disk at the bottom of the cuboid.

Finally, click on the **Boundary** button and define the boundary as **inside** cuboid 1. **Close** the boundary form. The geometry screen should look like Fig. 95.

Close the geometry screen, **save** the changes in a new input file, and **run** SCALE. The calculated k_{eff} is 0.9539 ± 0.0019.
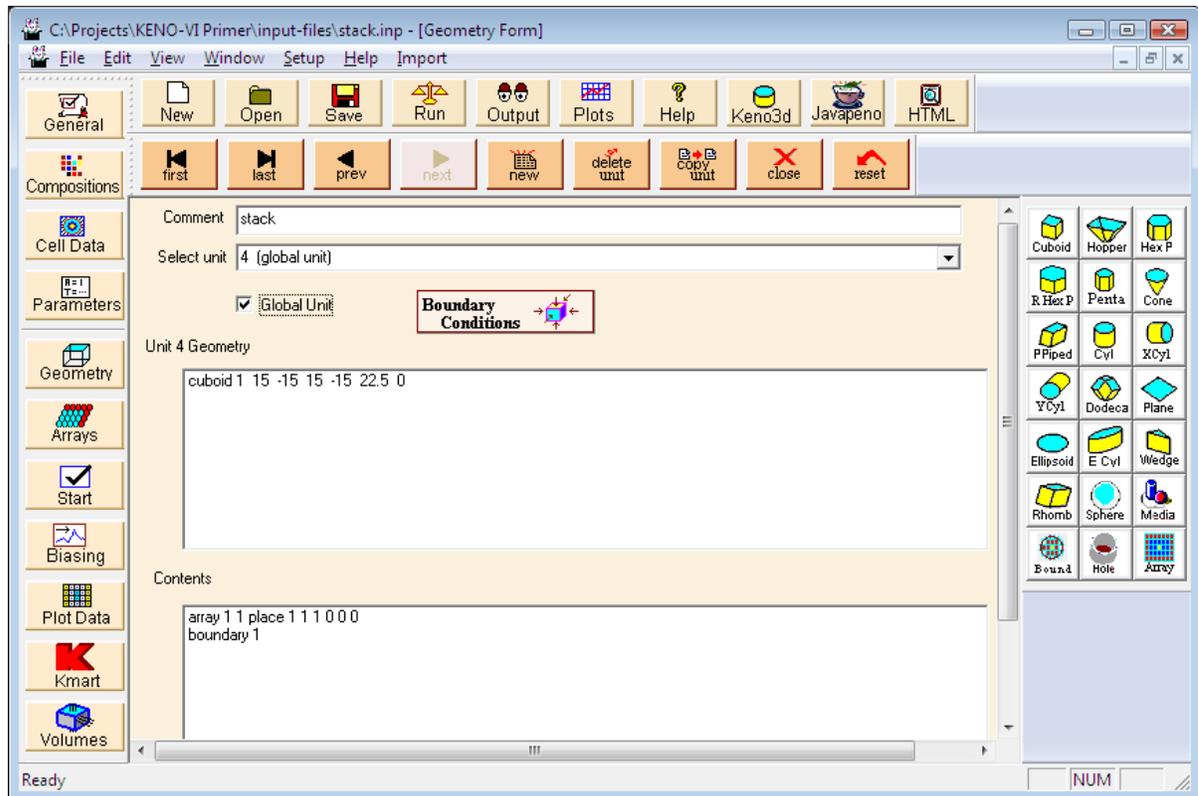
Fig. 95. Completed geometry screen for stacked disks problem.

4.7   SUMMARY

This section has helped you to accomplish these objectives:

- Use GeeWiz to describe some basic shapes (sphere, cylinder, 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 you have spent time with the material and simple geometry and array input, we will proceed to more advanced geometries.

5.   ADVANCED GEOMETRY INPUT

In the last section you set up and ran some simple geometry problems, including simple array problems. This section provides more advanced geometry options using KENO-VI geometry modification data.

5.1   WHAT YOU WILL BE ABLE TO DO

- Use GeeWiz and KENO-VI to describe more advanced geometry models using geometry modification data and holes.

- Use CHORD keyword to truncate a body with a plane perpendicular to a major axis.

- Use ORIGIN keyword to translate the location of a body.

- Use ROTATE keyword to rotate a body to any angle.

- Use HOLE record to place one unit inside another unit.

5.2   USING CHORD KEYWORD TO TRUNCATE A BODY

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

Each subordinate keyword specifies a cutting plane perpendicular to the axis in its name (e.g., "+X=" or "–X=" results in a cutting plane perpendicular to the X-axis). The cutting plane intersects the specified axis at the value that follows the "=" sign (e.g., "+X=5" or "-X=5" specifies a cutting plane perpendicular to the X-axis that crosses the axis at X=5). The positive or negative sign before the axis name indicates on which side of the cutting plane the geometric volume is to be kept. For example, in Fig. 96, "-X=5" specifies a cutting plane perpendicular to the X-axis at X=5 and keeps the volume on the negative side of the plane (i.e., $X \leq 5$). Conversely, "+X=5" specifies a cutting plane perpendicular to the X-axis at X=5 and keeps the volume on the positive side of the plane (i.e., $X \geq 5$).

It is possible to apply multiple chords to a body by specifying more than one subordinate keyword to a body. Figure 97 shows an example where two chords are applied to a cylinder to obtain a quarter cylinder. The two subordinate keywords "+X=0" and "+Y=0" keep the quarter cylinder where $X \geq 0$ and $Y \geq 0$.

```
sphere 1 10.0 chord -X=5.0          sphere 2 10.0 chord +X=5.0
```
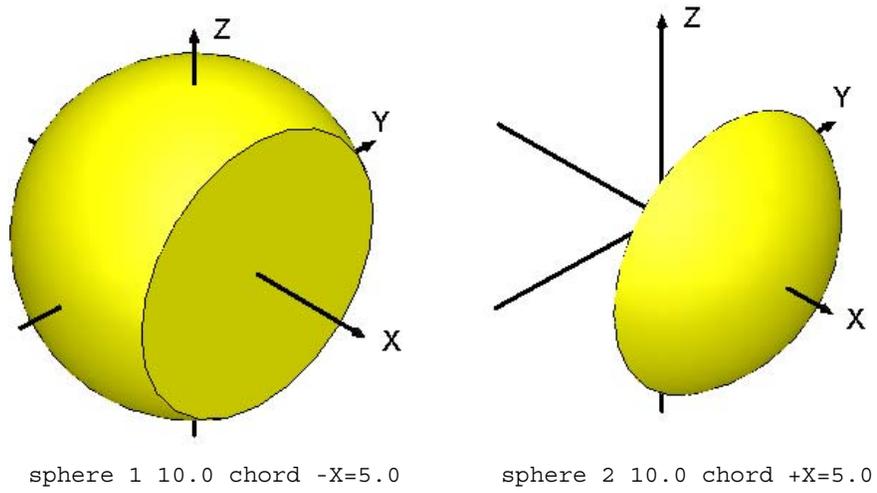
Fig. 96. Examples of body truncation using the CHORD keyword.


```
cylinder  1 10.0 5.0 0.0 chord +x=0 chord +y=0
```
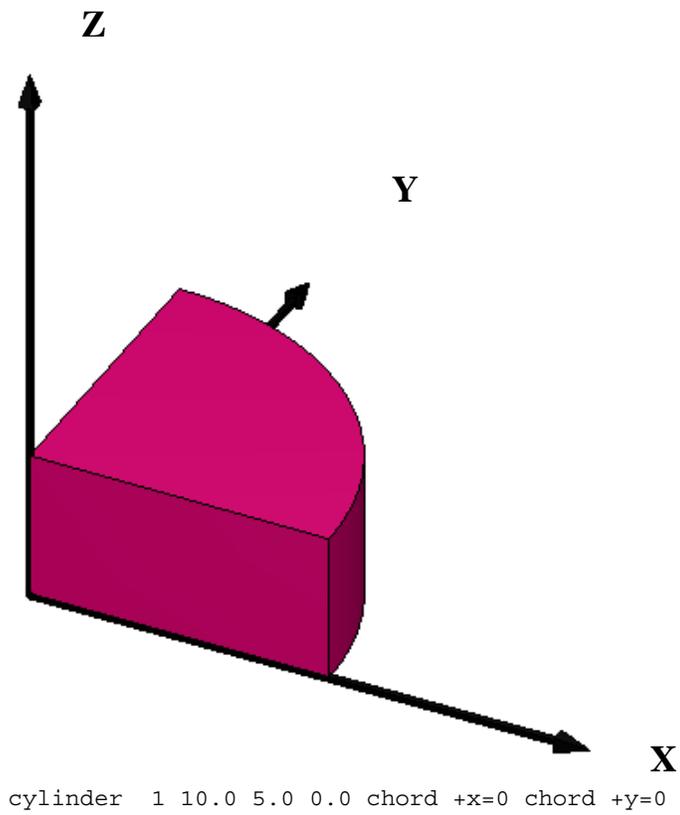
Fig. 97. Examples of body truncation using two chords.

5.3   USING ORIGIN KEYWORD TO TRANSLATE A BODY

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

A simple example is illustrated in Fig. 98, where the origin of a 10 cm radius sphere has been translated 10 cm in X, Y, and Z. The input syntax for this example is as follows:

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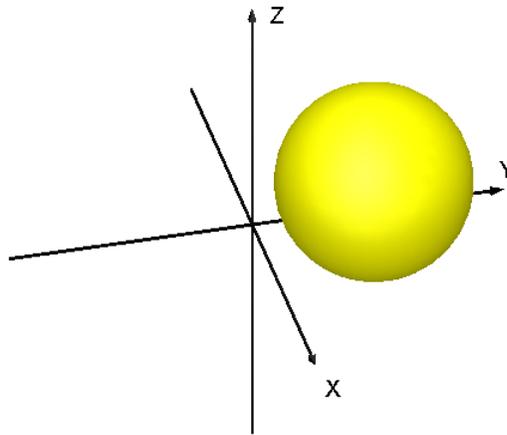

Fig. 98. Example of body translation using the ORIGIN keyword.

5.3.1　Example Using ORIGIN to Create Three Intersecting Cylinders

A more complex example is illustrated in Fig. 99, with three intersecting 10 cm radius cylinders that are 20 cm tall. Cylinder 1 is translated 4.5 cm in the positive Y direction; cylinder 2 is translated 4.5 cm in the negative X and Y directions; cylinder 3 is translated 4.5 cm in the positive X and negative Y directions. The numbers on the top of each colored region are the region definition vectors.
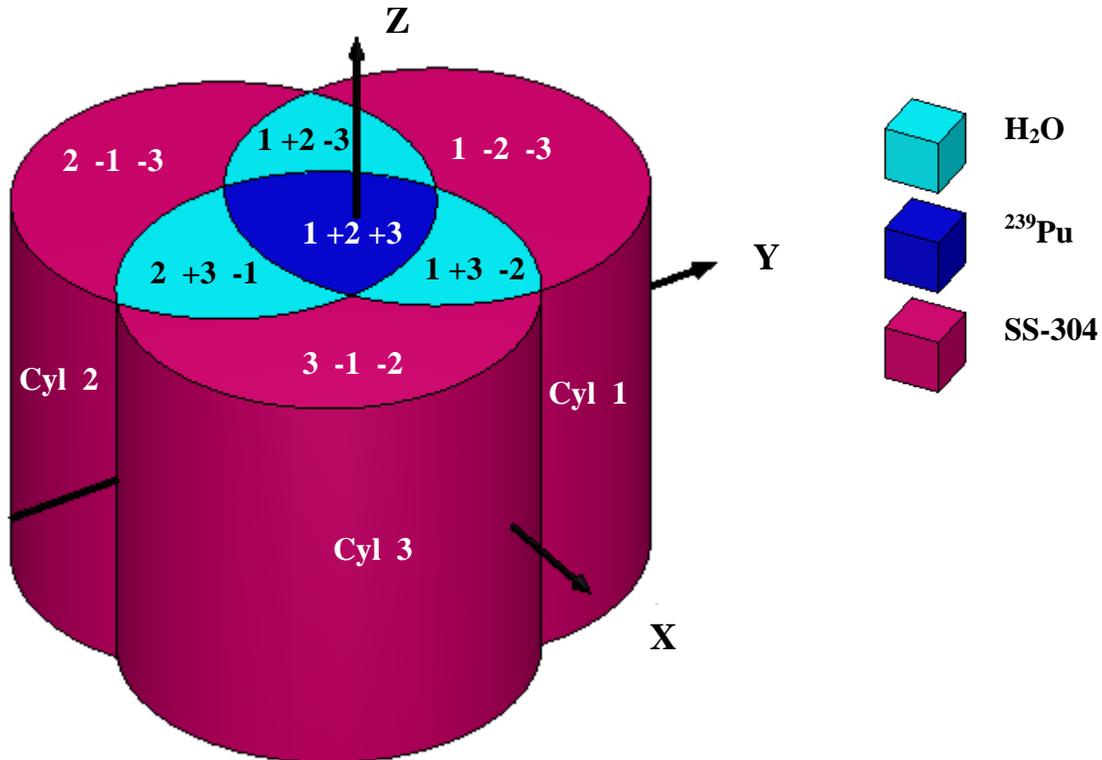
Fig. 99.　Three intersecting cylinders.

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

We will set up this problem using GeeWiz. Based on the knowledge that you have gained from completing the previous sections of the primer, you should be able to open a new file and enter the title, cross section, and material information. Note that all problems in this chapter use the 238-group ENDF/B-VII library (V7-238). Assign ^{239}Pu (pu-239) as mixture 1, water (h2o) as mixture 2, and stainless steel 304 (ss304) as mixture 3.

Click on the **Geometry** button to open the geometry screen. Enter a comment in the **Comment** field, and then click the **Cyl** button to define the first cylinder. Enter 10 for **R**, 10 for **Zt**, and -10 for **Zb**. Next click the **insert trans** button to insert a translation using

the ORIGIN keyword. Enter `4.5` for the **Y** value of **Origin** to translate the origin of the first cylinder to (0, 4.5, 0). Click **OK** to close the cylinder form.

On the main geometry form under **Unit 1 Geometry**, click on "`cylinder 1 10 10 -10 origin y=4.5`." Three buttons (**Copy**, **Edit**, **Delete**) appear. Click the **Copy** button to create `cylinder 2`. Click on `cylinder 2` and **Edit** (or you can double-click on `cylinder 2`). Change the values in **Origin** to `-4.5` for both **X** and **Y**. Click **OK** to save the changes.

Repeat the steps in the previous paragraph to create `cylinder 3`. The **Origin** data are `4.5` for **X** and `-4.5` for **Y**.

Click on **Cuboid** to create an outer boundary for the unit. Enter dimensions of `15` for **+X, +Y,** and **+Z**, and `-15` for **–X, –Y,** and **–Z**.

Next we enter the media data to fill each region. First we will define the [239]Pu region in the center. Click on the **Media** button. Select `cylinder 1`, and then click the **Inside** button. Repeat for `cylinder 2` and `cylinder 3`. The **Region Definition Vector** at the bottom of the form should now be `1 +2 +3`. (Note that the "+" sign is optional.) Select `1 pu-239` for the **Media,** and then click **OK** to close the media form.

Next we will define the three water regions. Click on the **Media** button. Select `cylinder 1`, and click the **Inside** button. Do the same for `cylinder 2`. Select `cylinder 3`, and click the **Outside** button. Select `2 h2o` for the **Media,** and then click **OK** to close the media form. Create two more **Media** records containing `h2o` and using the **Region Definition Vectors** `2 +3 -1` and `1 +3 -2`.

Next we will define the three stainless steel regions. Create three more **Media** records using the **Region Definition Vector** form for each stainless steel region. Select `3 ss304` for the **Media**.

The final **Media** record fills the surrounding void **inside** `cuboid 4` and **outside** `cylinder 1`, `cylinder 2`, and `cylinder 3`. Finally, set the **Boundary** to be **inside** `cuboid 4`.

The geometry form should look like Fig. 100. Click the **Run** button to save the input and run the problem. The calculated k_{eff} value is 1.3690 ± 0.0022.
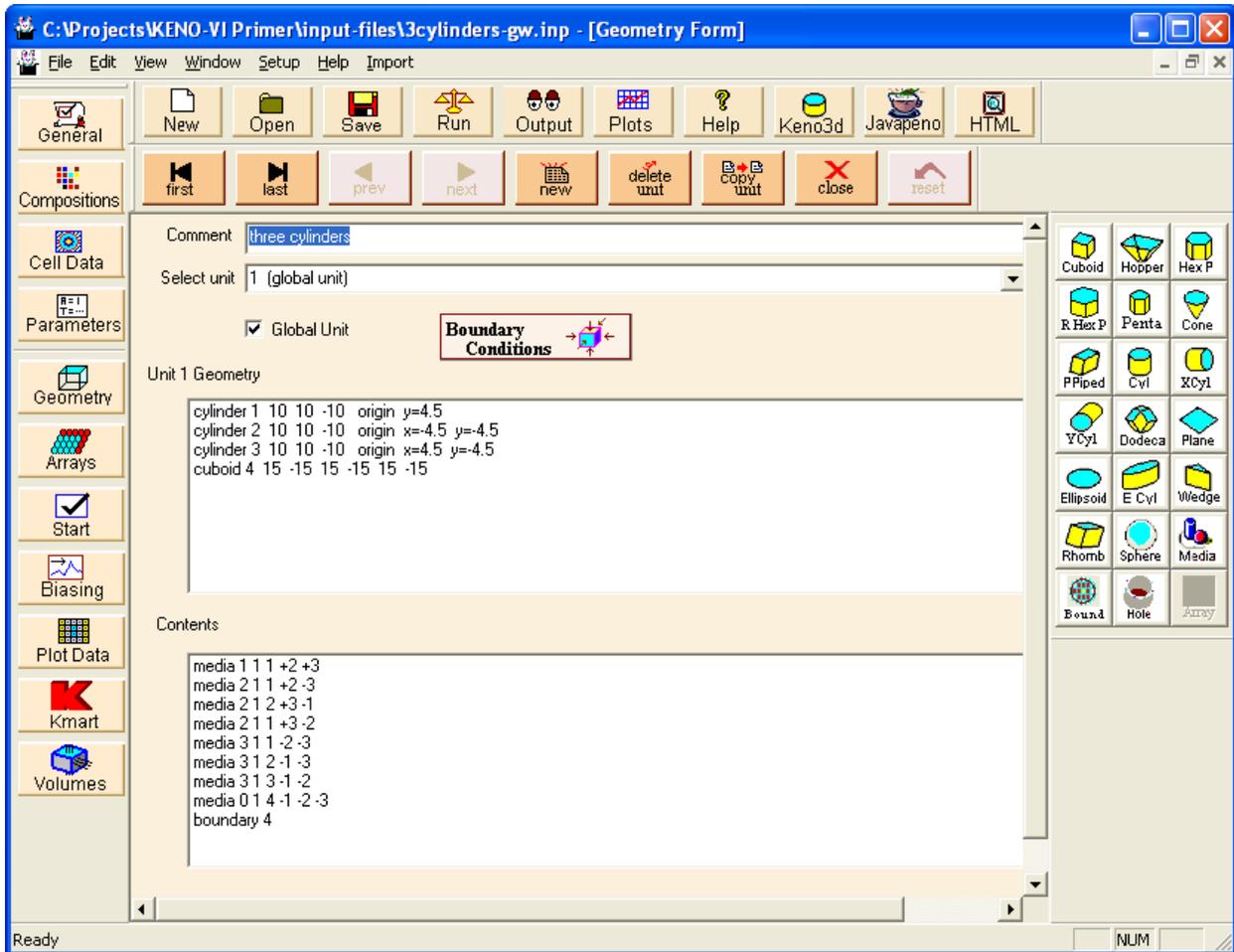
Fig. 100. Geometry form for three intersecting cylinders.

5.4 USING ROTATE KEYWORD TO ROTATE A BODY

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

The functions of the three subordinate keywords are illustrated in Fig. 101 The operations are performed in order from A1 to A3. First, A1 specifies α, the angle (in degrees) to rotate the volume counterclockwise around the Z-axis. The new locations of the

X-, Y-, and Z-axes carry the notations X', Y', and Z'. Next, A2 specifies β, the angle to rotate the volume counterclockwise around the X'-axis. The new locations of the X'-, Y'-, and Z'-axes are denoted as X", Y", and Z". Finally, A3 specifies γ, the angle to rotate the volume counterclockwise around the Z''-axis. We will walk through some examples to clarify how these keywords work.

CYLINDER Label R Zt Zb ROTATE A1=α A2=β A3=γ
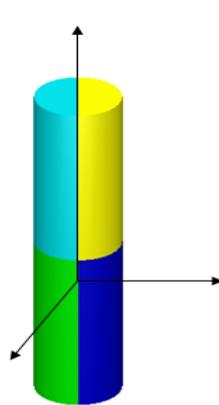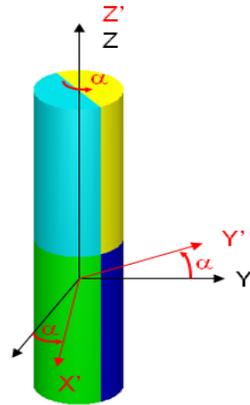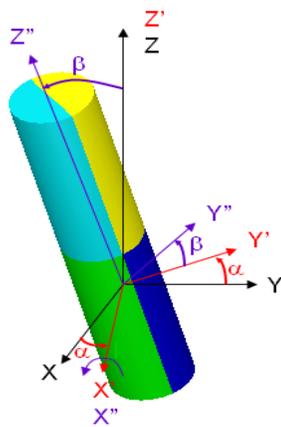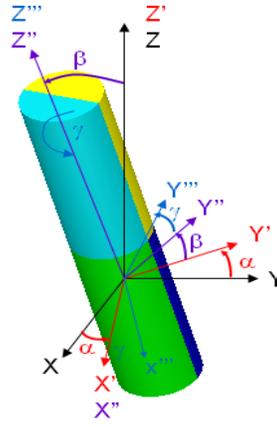
Fig. 101. ROTATE subordinate keywords A1, A2, and A3.

111

5.4.1 Rotation around the Z-Axis

To rotate a volume around the Z-axis, the subordinate keyword A1 should be used. The value specified after the keyword "A1=" is the angle of rotation around the Z-axis. Fig. 102 shows a cuboid in its default orientation with no rotation. The example in Fig. 103 shows the same cuboid with a rotation of 45 degrees counterclockwise around the Z-axis by specifying `ROTATE A1=45`. Note that a negative value for the angle of rotation would result in a clockwise rotation.
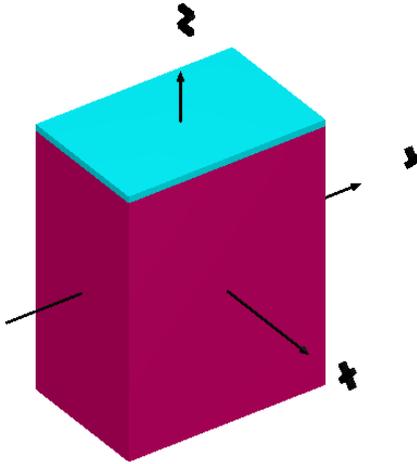
Fig. 102. Cuboid (default orientation).
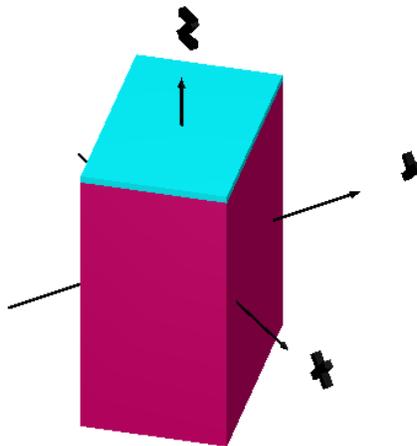
Fig. 103. Cuboid rotated 45 degrees counterclockwise around Z-axis (rotate A1=45).

5.4.2 Rotation around the X-Axis

To rotate a volume around the X-axis, the subordinate keyword A2 should be used. If A1 is not specified, then A1=0 and the X-axis and the X'-axis are the same. The example in Fig. 104 shows the same cuboid with a rotation of 45 degrees counterclockwise around the X-axis using the keywords `ROTATE A2=45`.
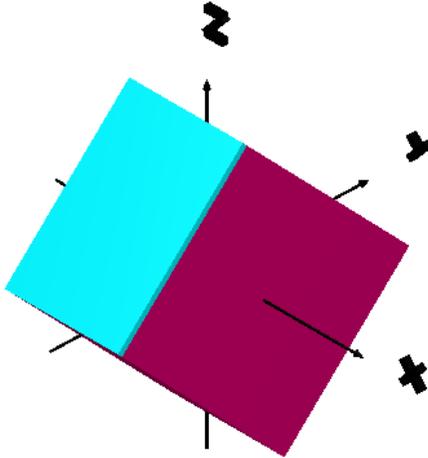
Fig. 104. Cuboid rotated 45 degrees counterclockwise around X-axis (rotate A2=45).

5.4.3 Rotation around the Y-Axis

To rotate a volume around the Y-axis, the subordinate keywords A1 and A2 must be used. First, a 90 degree counterclockwise rotation around the Z-axis must be performed (A1=90) so that the X'-axis is aligned with the Y-axis. Then the rotation around the X'-axis using A2 will be a rotation around the Y-axis. The example in Fig. 105 shows the cuboid from Fig. 104 with a rotation of 45 degrees counterclockwise around the Y-axis using the keywords ROTATE A1=90 A2=45.

.
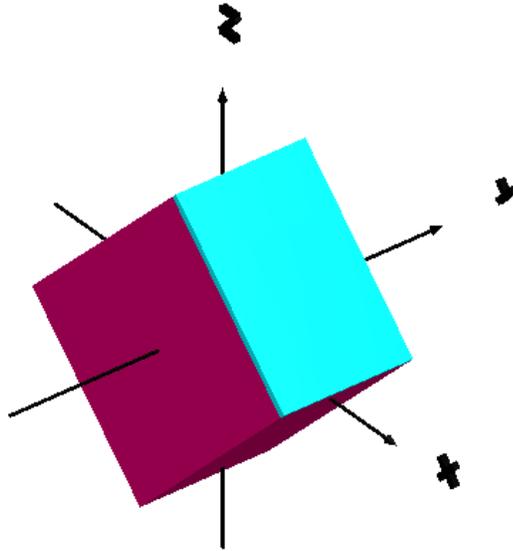
Fig. 105. Cuboid rotated 45 degrees counterclockwise around Y-axis (rotate A1=90 A2=45).

5.4.4 Example Using ROTATE Keyword for 29 Degree Y-Angle Joint

The next example is a critical experiment with intersecting aluminum pipes, in the shape of a Y, filled with a 5% enriched UO_2F_2 solution (Fig. 106). The solution contains 907.0 g/L of uranium, no excess acid, and has a solution density of 2.0289 g/cc. The assembly is composed of a 212.1 cm long vertical pipe and a second pipe that intersects the vertical pipe 76.7 cm from the outside bottom at an angle of 29.26 degrees with the upper vertical pipe. We will define the center of the vertical pipe at this point of intersection to be the origin of the geometry, as shown in Fig. 106. Both pipes have 13.95 cm inner diameters and 14.11 cm outer diameters. The vertical pipe is open on the top and 1.3 cm thick on the bottom. The Y-leg pipe is 126.04 cm in length with the sealed end 0.64 cm thick. The solution height is 128.2 cm.

From the point where the pipes intersect (see the origin in Fig. 106), the assembly is surrounded by water 37.0 cm in the ±X directions, 100 cm in the +Y direction, −37 cm in the −Y direction, to the top of the assembly in the +Z direction, and −99.6 cm in the −Z direction.
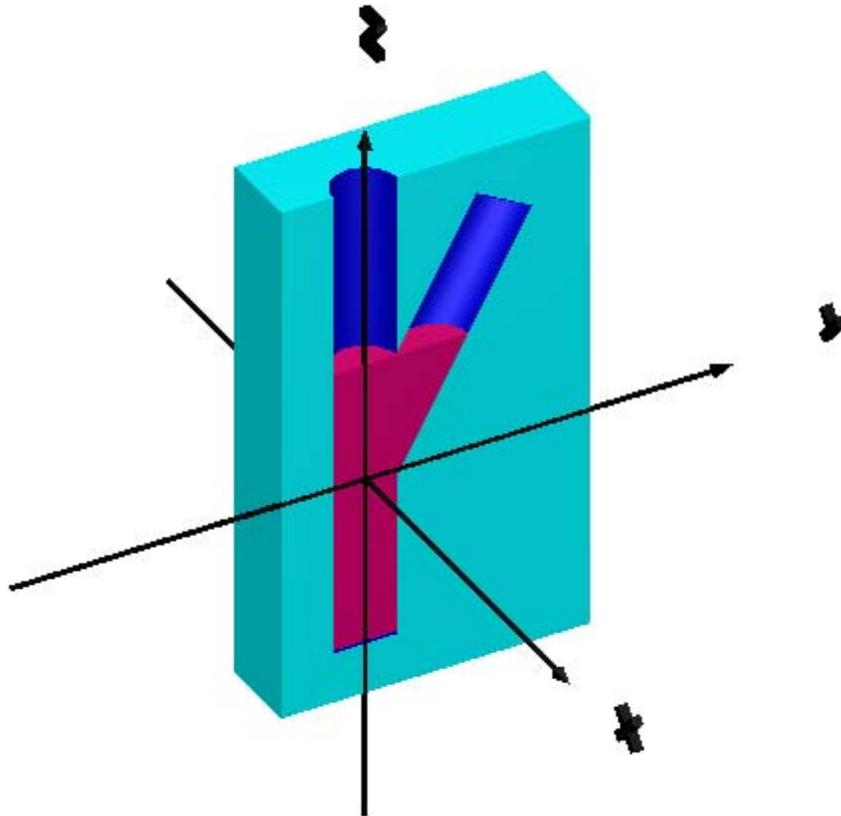
Fig. 106. Cutaway view of critical experiment with 29 degree Y-joint.

Click on the **General** button in GeeWiz to input the general information. Next click on **Compositions**. Click **Create** and then **Solution** to open the fissile solution form. Enter the **Solution Density** of 2.0289. Set **Number of Components** = 1. The **Acid** is hfacid (for hydrofluoric acid) with **Amount** 0.0 moles/liter (MOLAR) because there is no excess acid. Component 1 (**Comp 1**) is uo2f2 with **Amount** 907 grams/liter (RHO). Under **Isotopic Distribution**, click **Edit Selection** and then enter 5.0 for **92235** (^{235}U) and enter 0.0 for **92234** (^{234}U). Then click on **92238** (^{238}U) data entry area and select the **Fill to 100%** box. The completed form should look like Fig. 107.

For composition 2, click on the **new** button on the **Solution** screen and then select **Basic Composition**. From the pull-down menu, select al as the composition name, and accept the default values. Click **new** and select **Basic Composition** to create h2o as composition 3. This completes the material input for this example.

Fig. 107. Fissile solution form for UO$_2$F$_2$ solution.

Click on **Geometry** to open the geometry form. First we will define two cylinders (inner and outer, respectively) for the vertical pipe. Enter the following data on the cylinder form to define cylinders 1 and 2:

- Cylinder 1: `13.95` for **R**, `135.4` for **Zt** and `-75.4` for **Zb**
- Cylinder 2: `14.11` for **R**, `135.4` for **Zt** and `-76.7` for **Zb**

Next we define two more cylinders (inner and outer) for the angled pipe. We will use the `rotate` and `A2` keywords to rotate the cylinders 29.26 degrees clockwise around the X-axis. On the Cylinder form, click on the **insert rotate** button to specify the values for `A2`. The following values should be entered for each cylinder.

- Cylinder 3: `13.95` for **R**, `125.4` for **Zt** and `0.0` for **Zb**; `rotate a2=-29.26`
- Cylinder 4: `14.11` for **R**, `126.04` for **Zt** and `0.0` for **Zb**; `rotate a2=-29.26`

We need a horizontal plane to separate the portion of the cylinders filled with solution from the void portions above them. The top of the solution is 52.8 cm above the origin (128.2 cm solution height – 75.4 cm from origin to bottom of cylinder 1), so we need the plane to be perpendicular to the Z-axis and intersect it at Z = 52.8. Click on the **Plane** button on the geometry shape toolbar on the right side of the geometry screen. Set **ZPL** to `-1` and **CON** to `52.8`. This creates a plane corresponding to the equation –Z + 52.8 = 0. Click **OK** to close the plane form.

The final geometric volume needed is the cuboid for the water reflector. Create a cuboid with the following dimensions: +**X**=37, -**X**=-37, +**Y**=100, -**Y**=-37, +**Z**=135.4, and -**Z**=-99.6. After you close the cuboid form, click on the **Bound** button and define the boundary to be inside cuboid 6. When you close the boundary form, your geometry screen should look like Fig. 108.
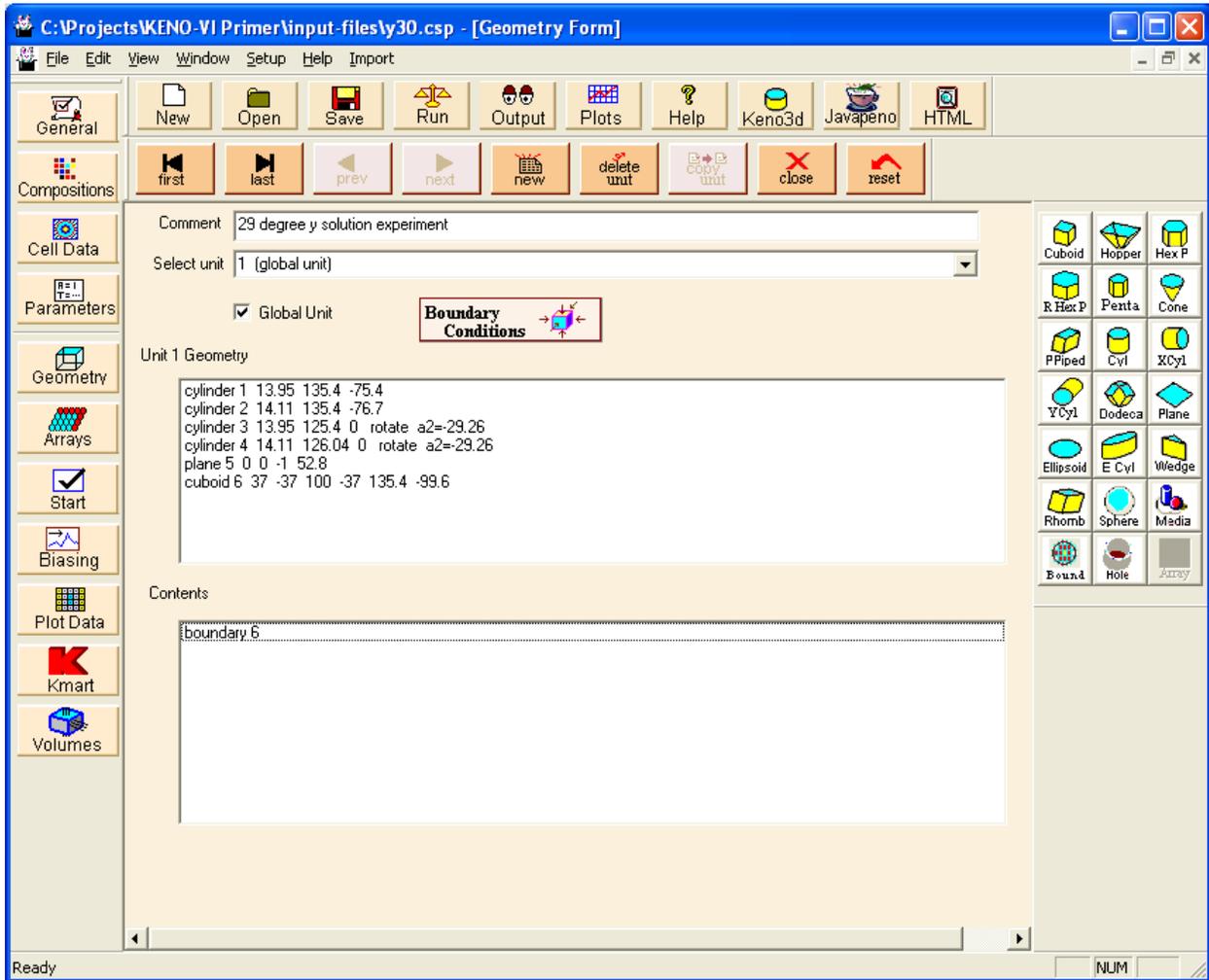
Fig. 108. Geometry data (no media data) for 29 degree Y-joint.

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

118

Click on the **Media** button to input the first media record, which will be the solution inside the vertical cylinder. Select `cylinder 1` and click **Inside**. Likewise, click on `plane 5` and click **Inside**. Note that the positive side of the plane ("inside" the plane) is the side where the equation $aX + bY + cZ + d > 0$ (in this case $-Z + 52.8 > 0$). Select media `1 uo2f2,hfacid` to fill this volume and click **OK** to close the media form.

Next we define the contents of the vertical pipe wall. The region definition vector on the media form will be **inside** `cylinder 2` and **outside** `cylinder 1` and `cylinder 3`. The mixture number is `2 al`.

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

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

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

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

Your completed geometry screen should look like Fig. 109 (be sure the **Global Unit** box is checked). Click the **Run** button to save the input and run the case. The calculated k_{eff} value for this critical experiment is 1.0032 + or -0.0018, which is near 1.0 as expected.
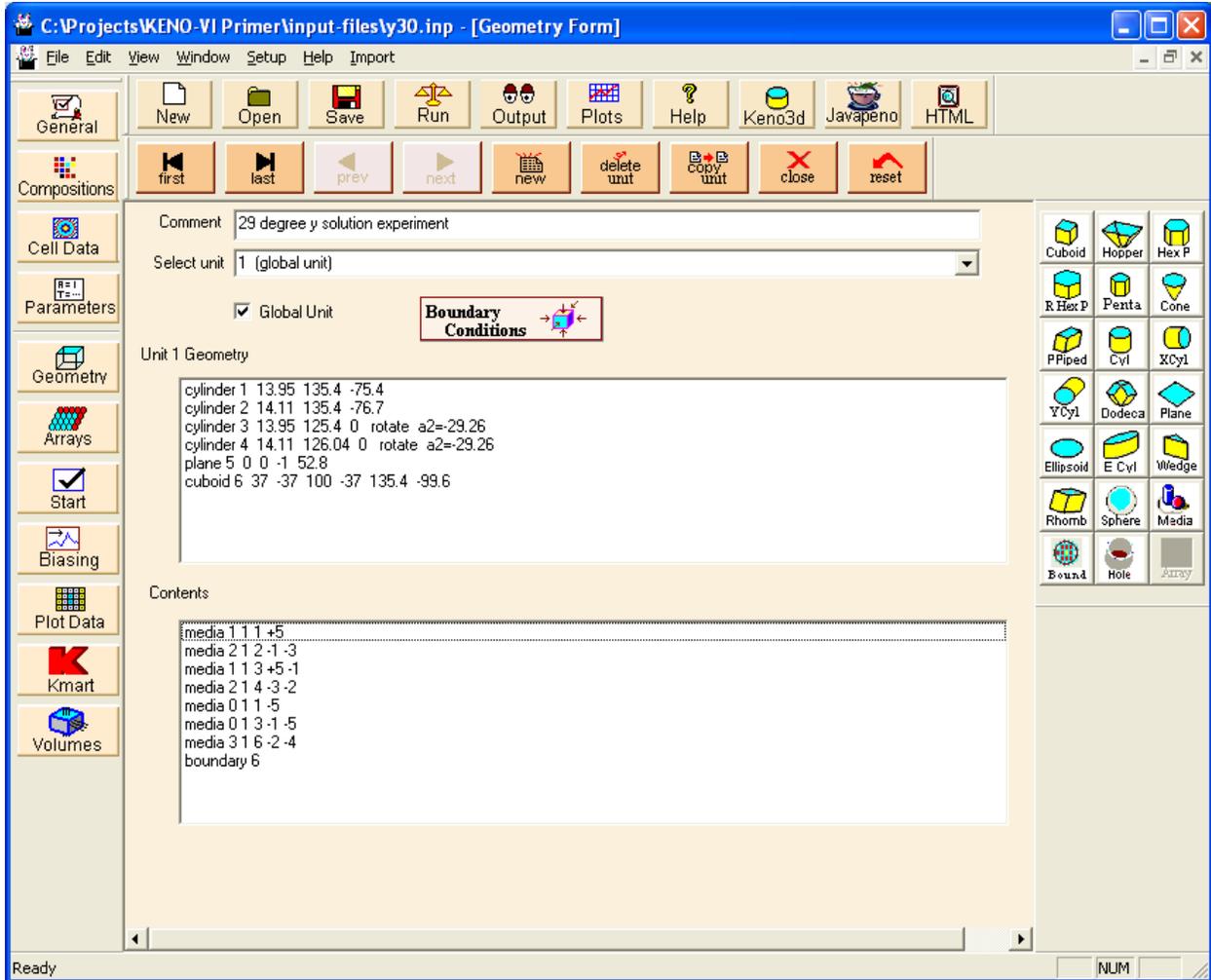
Fig. 109. Complete geometry data for 29 degree Y-joint.

5.5 HOLES

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

- media,
- arrays, and
- holes.

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

However, in some situations, placement of one unit inside another unit may be necessary or helpful. For those situations, a hole is needed. A hole is used to position a unit within a surrounding unit relative to the origin of the surrounding unit. The keyword, HOLE, is followed by the unit number being placed in the hole. The unit being placed in the hole can be rotated and translated using the ROTATE and ORIGIN options that we discussed previously in this chapter. The boundary record of a unit placed in a hole may contain more than one geometry label but all labels must be positive, indicating a location inside the respective geometry bodies.

There are three basic rules for holes.

1. **A hole contains a single unit.**

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

3. **Holes may be nested (i.e., a unit containing a hole may be placed as a hole inside another unit).**

5.5.1   Simple Hole Example with Rotations

In this example we have five stainless steel 304 boxes that contain plutonium metal (1% ^{239}Pu and 99% ^{240}Pu). The boxes are encased in a concrete cylinder. A cutaway view (top half removed) is shown in Fig. 110. The dimensions of the plutonium blocks are $10 \times 8 \times 20$ cm. The stainless steel box is 1 cm thick on three sides and on the top and bottom. The fourth side (i.e., the one oriented toward the center of the cylinder for the four outer boxes) is 3 cm thick. The center of each outer box is spaced 12 cm in X and Y from the center of the cylinder.

Create a new file with appropriate **General** and **Composition** information for the plutonium, stainless steel, and concrete. For the concrete, select `reg-concrete` in the basic compositions list. Once these data are complete, click on the **Geometry** button. The first unit will be the stainless steel box containing the plutonium. We know that the outer dimensions of the box are $12 \times 12 \times 22$ by adding the thickness of the stainless steel to the dimensions of the plutonium block. We want to place the origin of unit 1 at the center of the box. This orientation will make it easy to position each box as a hole in the concrete cylinder, unit 2.
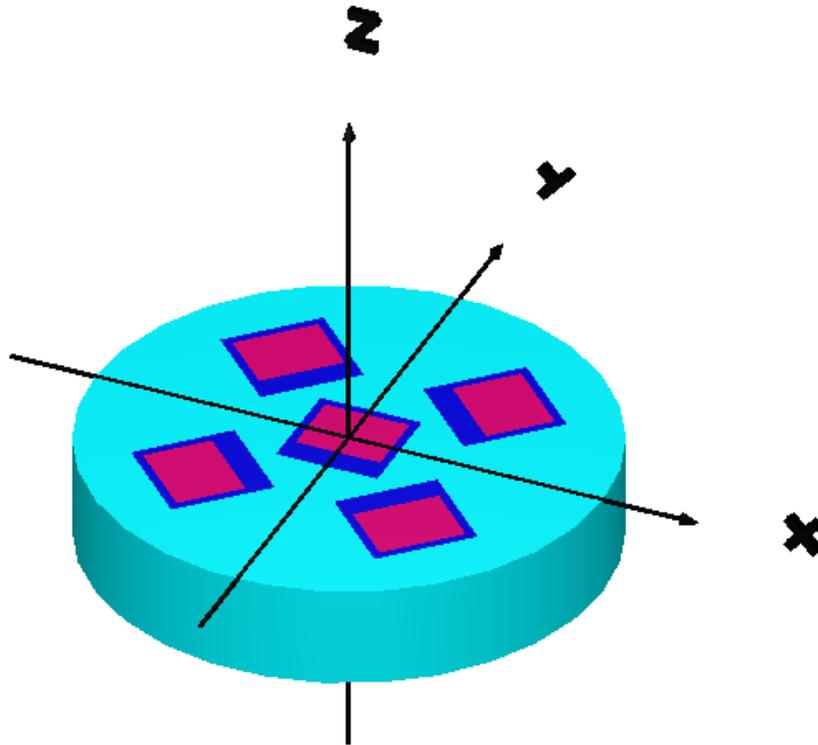
Fig. 110. Cutaway view of rotated holes example.

Click on **cuboid,** and enter the following data for the plutonium block: 5 for **X**, -5 for **-X**, 5 for **Y**, -3 for **-Y**, 10 for **Z** and -10 for **-Z**. Next create another cuboid for the stainless steel with the following dimensions: 6 for **X** and **Y**, -6 for **-X** and **-Y**, 11 for **Z** and -11 for **-Z**.

Click on the **Media** button to input the first media record, which will be the plutonium. Select cuboid 1 and click **Inside**. Select media 1 pu to fill this volume, and then click **OK** to close the media form. Create the second media record, and specify **inside** cuboid 2 and **outside** cuboid 1. Select media 2 ss304 to fill this volume and click **OK** to close the media form. The final entry for unit 1 is the boundary. Click on **Bound** and specify **inside** cuboid 2.

Click on **new** to create unit 2. Click on **Cyl** to create the concrete cylinder and enter 30 for **R**, 12 for **+Z** and -12 for **-Z**. Next click on **Media,** and select 3 reg-concrete **inside** cylinder 1.

Now we will place the five boxes as holes in unit 2. Select **Hole** from the right-hand toolbar. For **Hole Unit Number,** select 1 from the pull-down menu. The next three entries (**A1**, **A2**, **A3**) specify the angles of rotation. The last three entries specify the location of the hole's origin with respect to the origin of the unit in which it is placed. The first hole will be placed in the center of unit 2 without rotation, so all six values should remain 0. Click **OK** to close the hole form.

Place the next hole with its **Origin** at **X**=12, **Y**=12. The box needs to be rotated 45 degrees clockwise around the Z axis, so enter **A1**=-45.

Continuing clockwise around the cylinder, enter the following data for the remaining holes:

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

Define the boundary of unit 2 using the **Bound** button and specify **inside** cylinder 1.

Check the **Global Unit** box because unit 2 is the global unit for this example. After entering an appropriate comment, the geometry form for unit 2 should look like Fig. 111.

After checking that the input data match, select **Close** on the geometry form, and then select **Run** to execute SCALE. The calculated k_{eff} is 1.0653 ± 0.0018, so this case is obviously not a safe geometry.
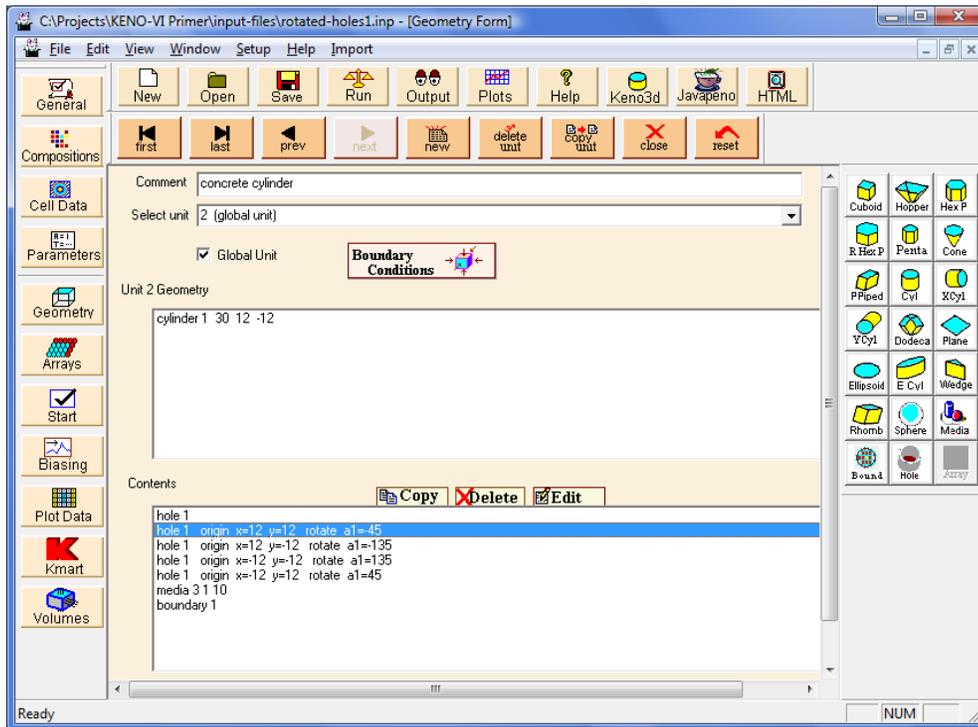
Fig. 111. Unit 2 geometry input for rotated holes example.

5.6 SUMMARY

This section has helped you to accomplish the following:

- Use GeeWiz and KENO-VI to describe more advanced geometry models using geometry modification data and holes.

- Use CHORD keyword to truncate a body with a plane perpendicular to a major axis.

- Use ORIGIN keyword to translate the location of a body (i.e., shape).

- Use ROTATE keyword to rotate a body to any angle.

- Use HOLE record to place one unit inside another unit.

    Next we will proceed to using holes and arrays in more complex geometries.

6. ARRAYS

6.1 WHAT YOU WILL BE ABLE TO DO

- Use GeeWiz and KENO-VI to describe complex geometry models using arrays.

- Remove unwanted portions of arrays using region definition vectors to apply boundaries.

- Build hexagonal arrays.

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

This problem illustrates how to remove unwanted portions of an array using boundaries in the region definition vector on the array record. A cruciform-shaped U(93.2)C ceramic element sits in water inside a SS316 cylinder (see Fig. 112). Five of these cylinders are then placed inside a larger SS316 cylinder (Fig. 113). The cruciform element is 10 cm on a side with a width of 1 cm and a height of 20 cm. Each element is contained in a SS316 cylinder with an inner diameter of 12 cm and an inner height of 20 cm. The SS316 is 0.1 cm thick. These cylinders are packed in a cruciform-shaped array with a pitch of 12.4 cm. The five containers are then placed inside a SS316 cylinder with an inside diameter of 40 cm, an inner height of 20.2 cm, and thickness of 0.2 cm.
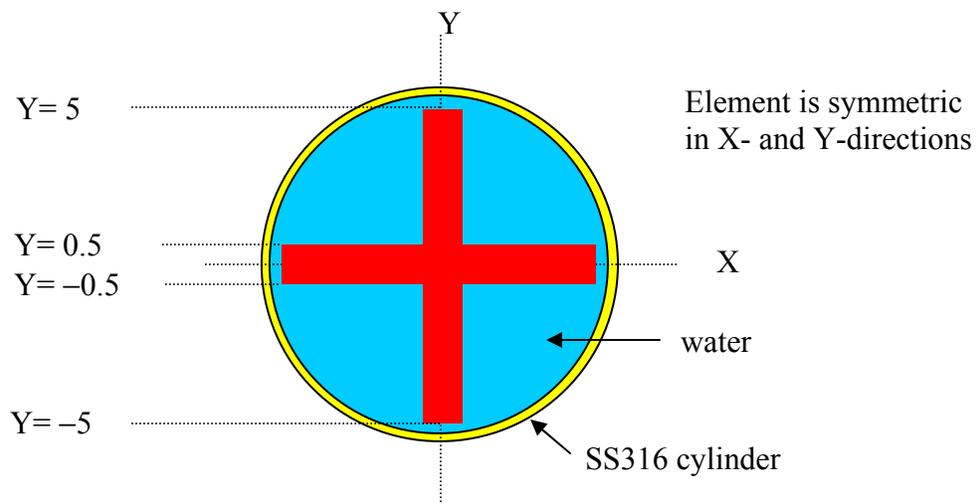
Fig. 112. SS316 cylinder with U(93.2)C cruciform element.
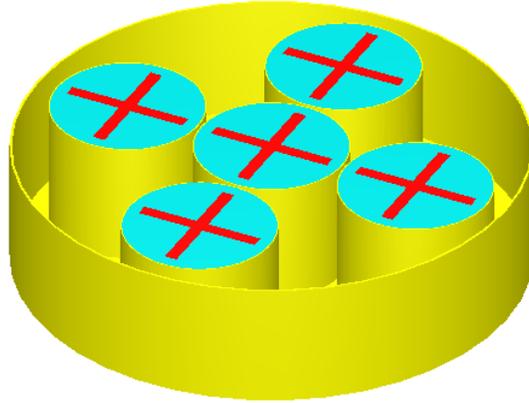
Fig. 113. Five cruciform cylinders in shipping container.

By now, you should be quite comfortable entering the **General** information and **Composition** descriptions. Next proceed to the **Geometry** for this problem. The cruciform elements can be formed by two intersecting cuboids. We will create unit 1, which contains the cruciform element, water, and SS316 cylinder. Because we are putting unit 1 into an array, the outside region of unit 1 must be a cuboid so that the flat surfaces can be touching.

Select **Cuboid** to define the horizontal part of the cruciform. Enter `5, -5, 0.5, -0.5, 10, -10` for +**X** through −**Z**. This puts the origin in the center of the cruciform so it will be centered in the tank. Select **Cuboid** again to define the vertical part of the cruciform. Enter `0.5, -0.5, 5, -5, 10, -10` for +**X** through −**Z**. The next volume is the cylinder that contains the water in which the cruciform will be placed. Select **Cyl** and enter `6` for **R**, `10` for **Zt**, and `−10` for **Zb**. Select **Cyl** again for the SS316 shell. Enter `6.1` for **R**, `10.1` for **Zt**, and `−10.1` for **Zb**. The last volume in unit 1 is the cuboid outside the steel cylinder, which allows us to use this unit in an array. Because the array pitch in the X- and Y-directions is 12.4 cm, the ±**X** and ±**Y** entries are ±`6.2` (i.e., half of 12.4). The ±**Z** entries are ±`10.1` (same as the outer steel cylinder).

Now enter the **Media** for unit 1. First is the UC in the horizontal cruciform element. Select `cuboid 1`, and click on **Inside**. Select `1 uc` for the media from the pull-down menu. Next is the vertical cruciform element. Click on **Media,** and specify **inside** `cuboid 2` and **outside** `cuboid 1`. Select `1 uc` for the media. For the next **Media** record, enter the water. Specify **inside** `cylinder 3` and **outside** `cuboid 1` and `cuboid 2`. Select `2 h2o` from the pull-down menu for the media. The next **Media** entry is the SS316 cylindrical container. Specify **inside** `cylinder 4` and **outside** `cylinder 3`. Select `3 ss316` for the media. The final **Media** record for unit 1 is the void cuboid outside the steel cylinder. Specify **inside** `cuboid 5` and **outside** `cylinder 4`. Select `0 void` for the media. When you have completed the `boundary` record, the geometry form for unit 1 should look like Fig. 114.
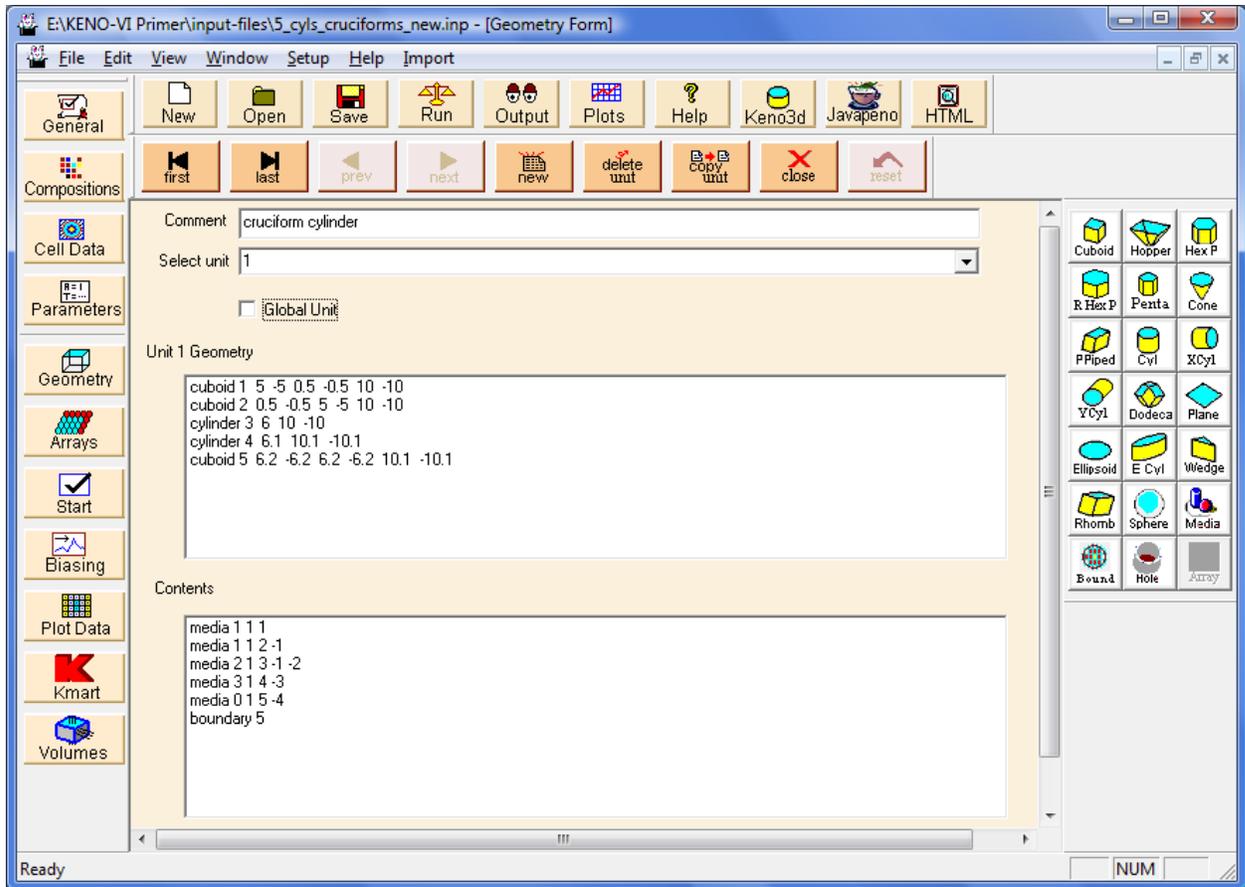
Fig. 114. Geometry input for unit 1.

Before defining the global unit, we need to define the array that will be inserted in that unit. Note that the five individual containers are positioned to fill the five interior positions in a 3 × 3 array with one unit in the top center position, three units along the center row, and one unit in the bottom center position. Because arrays must be rectangular, we must enter empty cuboids to occupy the four corner positions of the 3 × 3 array. First we will define unit 2 to be the empty cuboid. Then we can construct the array. However, because the five individual containers do not tightly fit in the shipping container (i.e., three times the width of unit 1 is less than the inner diameter of the shipping container), there would be gaps between a 3 × 3 array and the inner wall of the container. Therefore, we will construct a 5 × 5 array.

Click the **new** button to create unit 2. Click on **Cuboid,** and enter the same dimensions as the cuboid in unit 1. Then click on **Media,** and specify `0 void` **inside** `cuboid 1`. Define the `boundary` as inside `cuboid 1` to complete unit 2.

Open the **Array** form by selecting the **Arrays** button on the left-hand toolbar. Then select **Create** from the upper toolbar on the array form; this will bring up the array properties form. This array is the **Global Array**, so we check that box. Select `Square` as the array **type**. Specify **NUX**=5, **NUY**=5, and **NUZ**=1. Use unit `1` to **initialize new array locations**. The completed form should look like Fig. 115.

Fig. 115. Array properties for cruciform cylinders example.

After closing the array properties form, the array form should show 5 × 5 squares of the same color. We need to place unit 2 empty cuboids in the four corner positions. Click on the pointing-hand icon on the right toolbar (Step 1 in Fig. 116). Then click on the colored square for unit 2 in the **Units** list on the left side of the array form (Step 2 in Fig. 116). Click on each array position in the top row, bottom row, left column, and right column to load them with unit 2. Then click on positions (2,2), (4,2), (2,4), and (4,4). The modified array form should look like Fig. 117. Then **Close** the array form, and return to the geometry form.
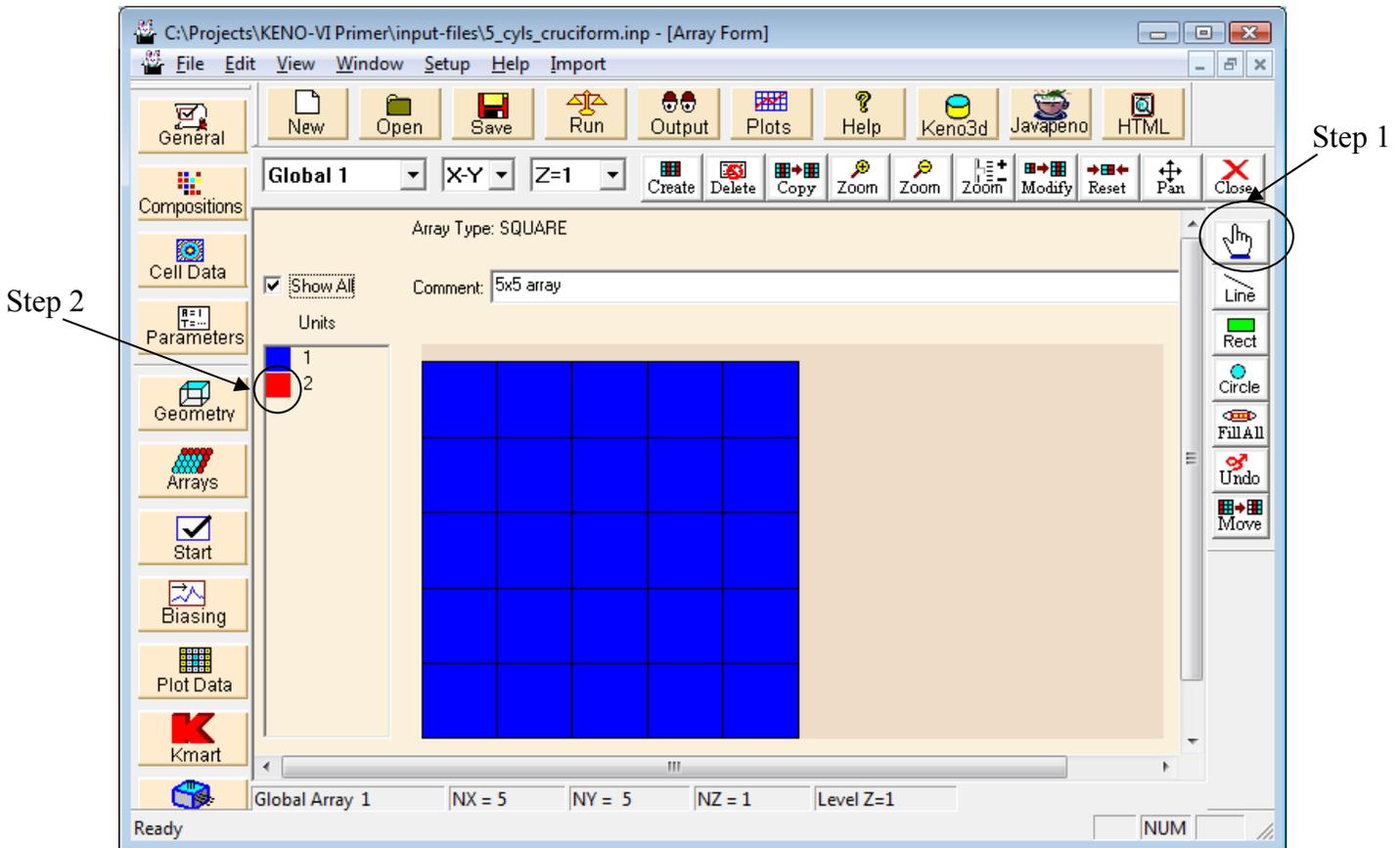
Fig. 116. Array form with steps to change unit numbers in array positions.
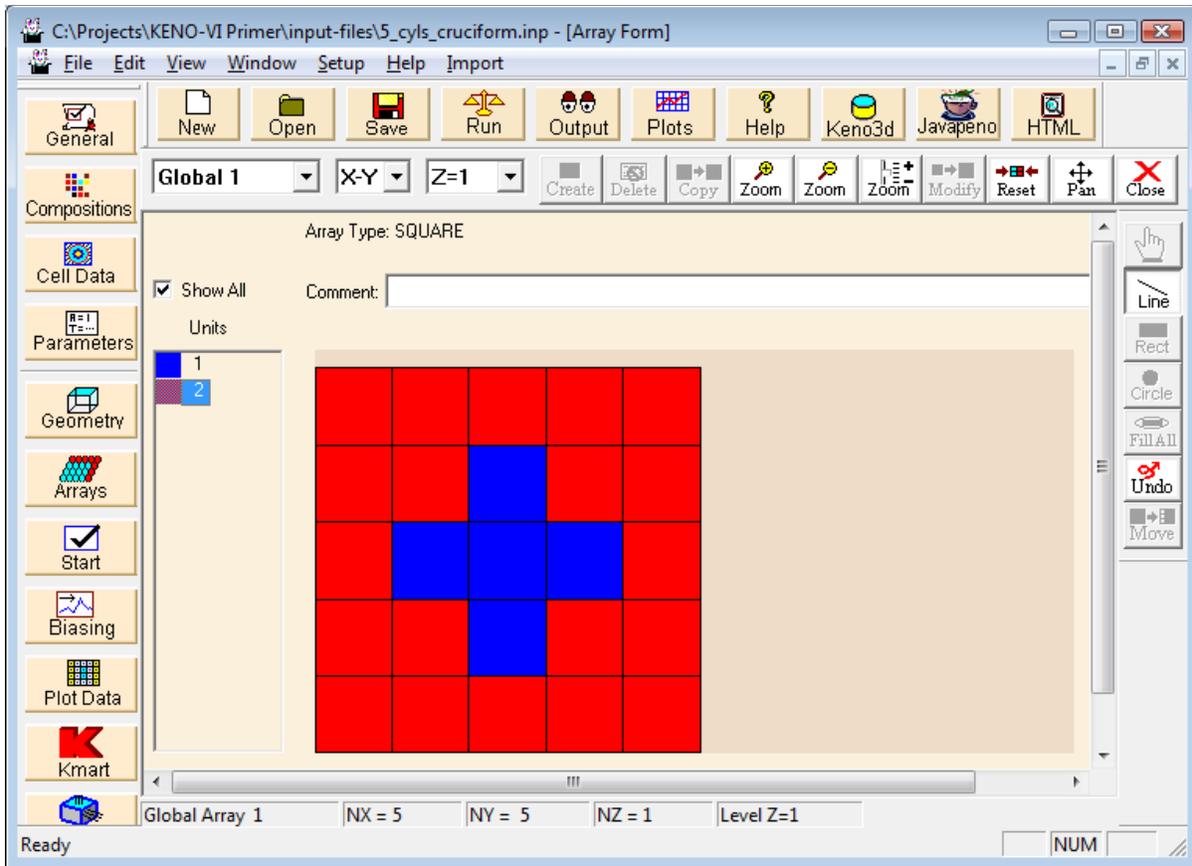
Fig. 117. Array form with modified array layout.

Create a **new** unit 3, which will be the **global unit**. The first geometry record is the inner cylinder of the shipping container. Click on **Cyl,** and enter 20 for **R**, 10.1 for **Zt,** and -10.1 for **Zb**. The next record is the outer cylinder of the container. The shipping container is 0.2 cm thick, so the dimensions are 20.2 for **R**, 10.3 for **Zt,** and -10.3 for **Zb**.

Next we place the array inside cylinder 1. Select **Array** from the right-hand toolbar. **Select existing array** 1, and specify **inside** cylinder 1. **Place** the center array element 3 3 1 at position 0 0 0. Click **OK** to close the Array form. Click on the **Media** button to define the contents of the SS316 outer cylinder. Specify **inside** cylinder 2 and **outside** cylinder 1, and select 3 ss316 for the media. Define the boundary of unit 3 as **inside** cylinder 2. The completed geometry form for unit 3 should look like Fig. 118.
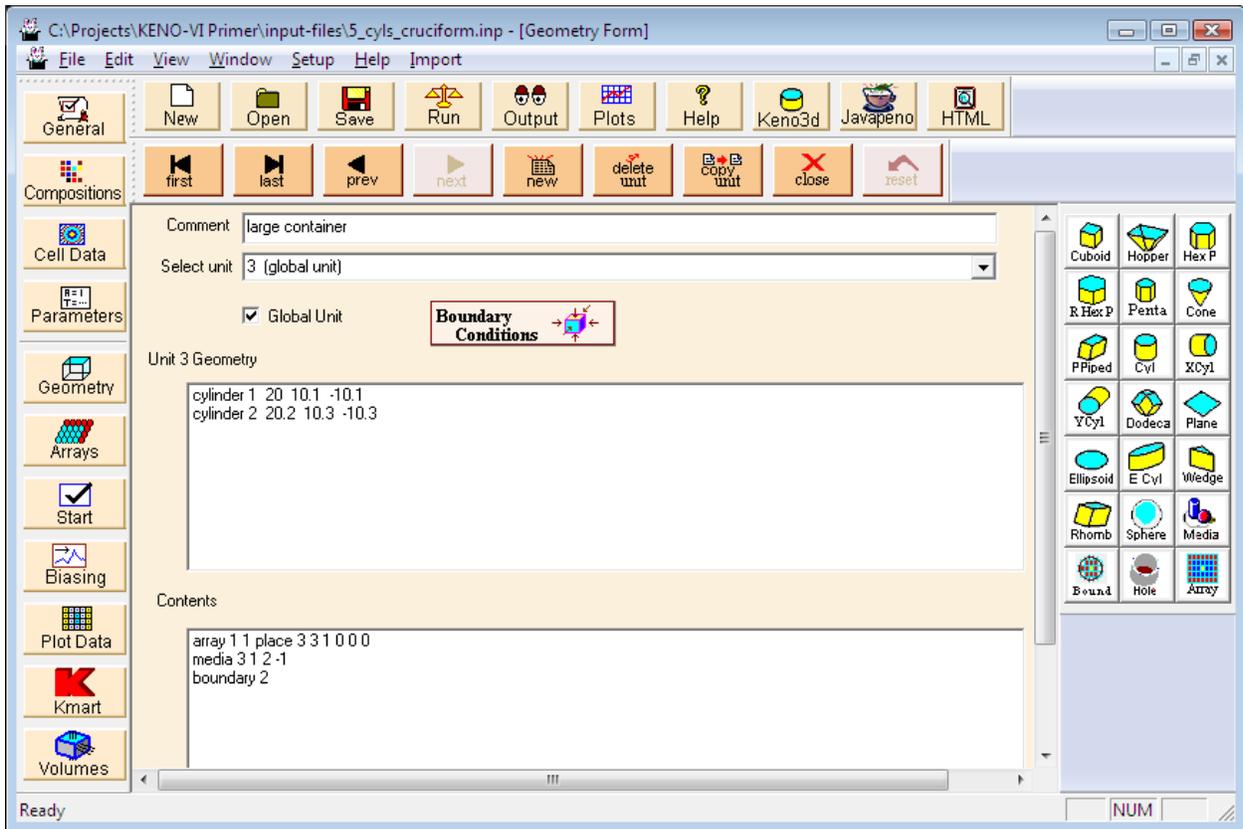
Fig. 118. Geometry data for unit 3.

Save the changes in a new input file and execute SCALE. The calculated k_{eff} value is 0.6239 ± 0.0018.

6.3   HEXAGONAL ARRAY WITH INNER AND OUTER BOUNDARIES

This next problem illustrates how to create a hexagonal array and remove unwanted portions of the array using both inner and outer boundaries in the region definition vector on the array record. An annular carbon steel cylindrical tank contains six aluminum-clad U(10.0) cylinders submerged in water (Fig. 119). The uranium cylinders are 15 cm in diameter and 18.3 cm high. The aluminum clad is 0.5 cm thick on the sides and bottom of the uranium. The cylinders have a center-to-center triangular pitch of 21 cm. The tank has an inner radius of

12 cm and an outer radius of 31.5 cm. The inside height is 18.8 cm. The tank walls and bottom
are 1.5 cm thick. A cutaway view with highlighted edges in KENO3D is shown in Fig. 120.
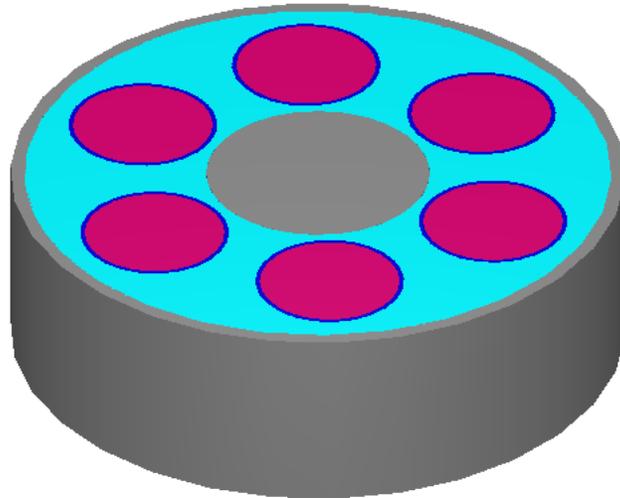
Fig. 119. Annular carbon steel cylindrical tank with aluminum-clad uranium cylinders.
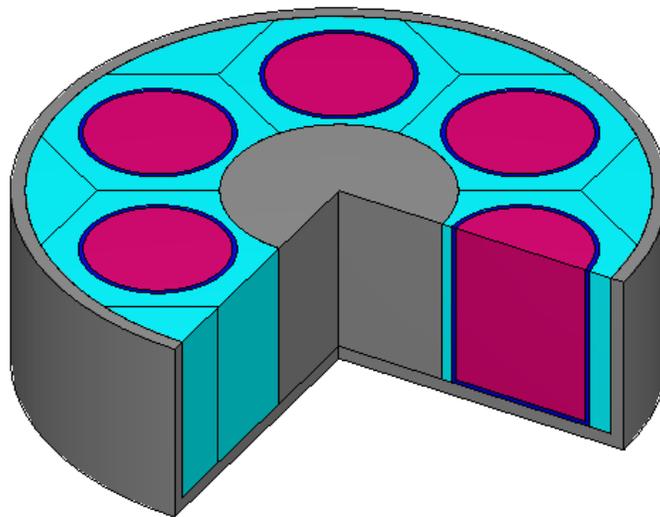
Fig. 120. Cutaway view of tank with highlighted edges.

Enter the **General** information and **Composition** descriptions. Next proceed to the **Geometry** for this problem. The first unit is the uranium cylinder with aluminum clad. Click on **Cyl,** and enter the dimensions of the uranium cylinder: `7.5` for **R**, `18.3` for **Zt**, and `0` for **Zb**. Next enter the data for the aluminum clad cylinder: `8` for **R**, `18.3` for **Zt**, and `-0.5` for **Zb**. We must surround this cylinder with a hexprism of water so that we can place this unit in a hexagonal array. Click on **Hex P,** and enter `10.5` for **R**, `18.3` for **Zt**, and `-0.5` for **Zb**.

The media records for unit 1 are quite simple. Mixture 1 is **inside** `cylinder 1`. Mixture 2 is **inside** `cylinder 2` and **outside** `cylinder 1`. Mixture 3 is **inside** `hexprism 3` and **outside** `cylinder 2`. The boundary of unit 1 is **inside** `hexprism 3`.

Unit 2 will be a hexprism of water that is needed to put in empty hexagonal array positions around the outside of the unit 1 hexprisms. (You can see portions of these in Fig. 120.) We could create the unit with similar steps as those used to create unit 1. However, another option in GeeWiz is to use the **copy unit** button. Click on that button, and unit 2 is automatically created identical to unit 1. Next click on `cylinder 1`, and click the **Delete** button just above it. A warning message is displayed. Click **OK,** and all media record references to the cylinder are automatically deleted along with the geometry record. Repeat the same steps for cylinder 2. The remaining input describes the hexprism of water that we need.

Now we are ready to create the hexagonal array. Click on the **Arrays** button on the left-hand toolbar. Then select **Create** from the upper toolbar on the array form; this will bring up the array properties form. This array is the **Global Array**, so we check that box. Select `Standard Hexagonal` as the array **type**. Specify **NUX**=5, **NUY**=5, and **NUZ**=1. These values allow us to put empty unit 2 hexprisms around the periphery. Use `unit 2` to **initialize new array locations**.

After closing the array properties form, click on the **Line** button on the right toolbar. Then click on the colored square for unit 1 in the **Units** list on the left side of the array form. Hold down the left mouse button and drag a line through the ring of six positions around the center array position (Fig. 121). Each array position that is touched by the line will be assigned unit 1. When you release the mouse button, the modified array form should look like Fig. 122. If you make a mistake, you can click the **Undo** button on the right toolbar. Then **Close** the array form, and return to the geometry form.

Click on **new** to create unit 3, which will be the **global unit**. The first geometry record is the steel annulus of the shipping container. Click on **Cyl,** and enter `12` for **R**, `18.3` for **Zt,** and `-0.5` for **Zb**. The next record is the inner wall of the container, which has a radius of `31.5` cm and the same values for **Zt** and **Zb**. The container is 1.5 cm thick, so the outer dimensions for the third cylinder are `33` for **R**, `18.3` for **Zt,** and `-2` for **Zb**.
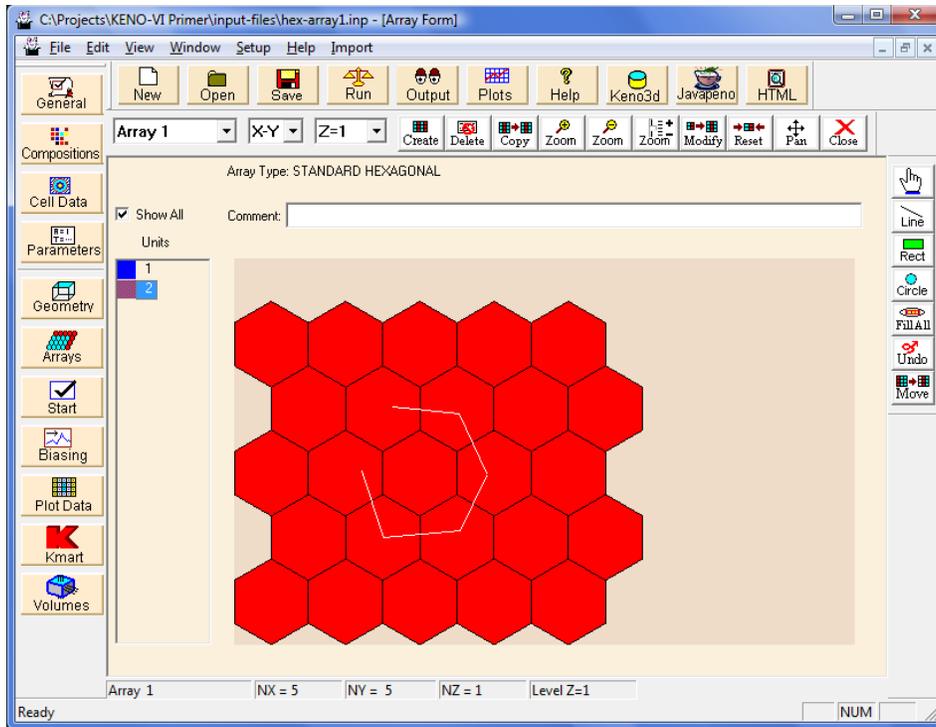
Fig. 121. Array form with line tool.
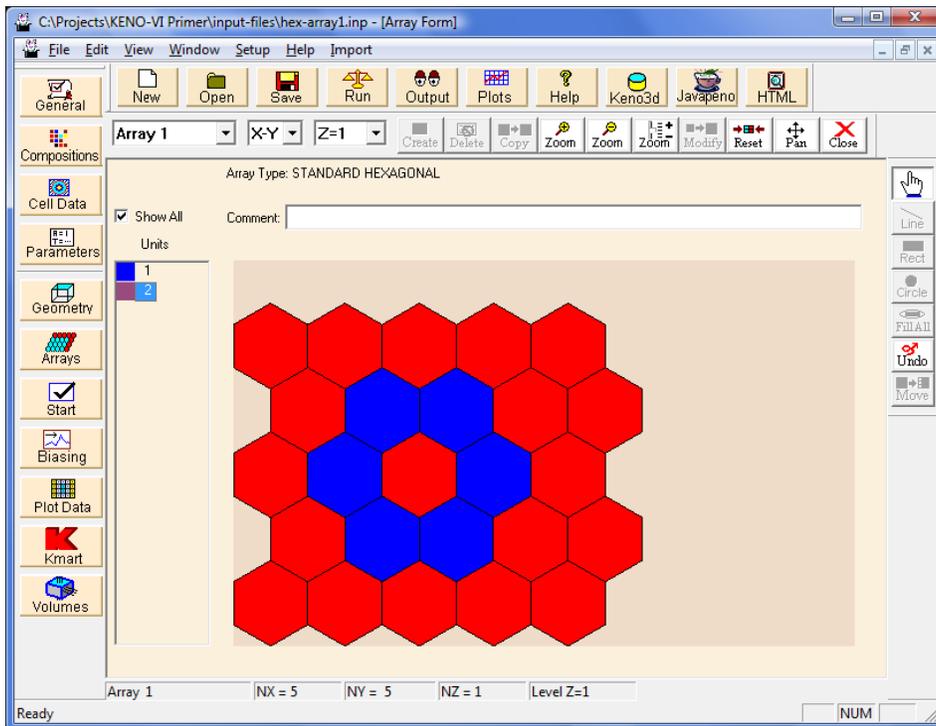
Fig. 122. Modified array form.

134

Click on the **Media** button to define the contents of the steel inner cylinder. Specify **inside** `cylinder 1`, and select `4 carbonsteel` for the media. Next we place the array inside cylinder 2. Select **Array** from the right-hand toolbar. **Select existing array** `1`, and specify **inside** `cylinder 2` and **outside** `cylinder 1`. By combining these two conditions on the region definition vector for the array, we define an inner boundary and an outer boundary on the array. Click on the **Media** button to define the contents of the steel outer cylinder. Specify **inside** `cylinder 3` and **outside** `cylinder 2`, and select `4 carbonsteel` for the media. Define the `boundary` of unit 3 as **inside** `cylinder 3`. The completed geometry form for unit 3 should look like Fig. 123.
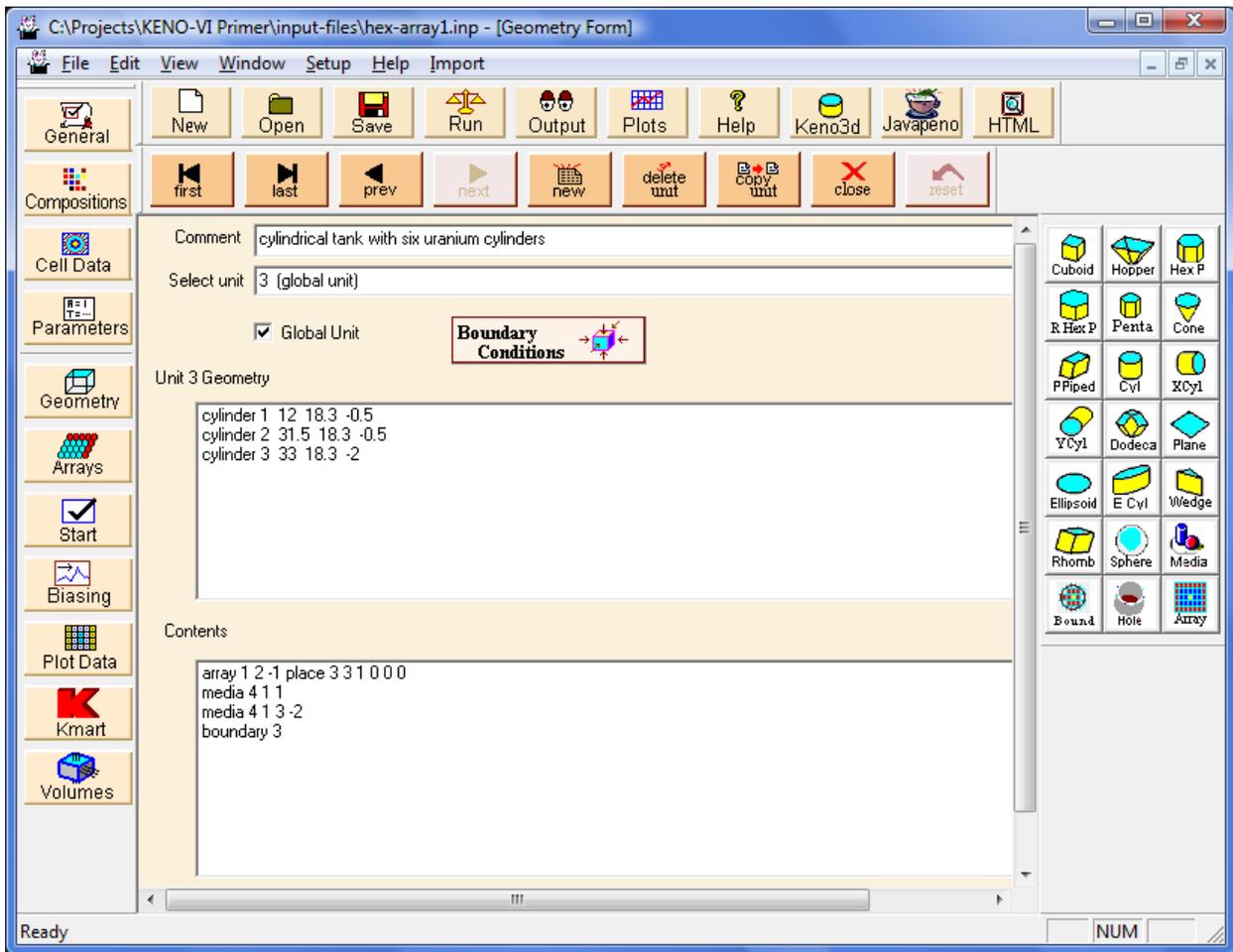
Fig. 123. Geometry form for unit 3.

Save the changes in a new input file, and then execute SCALE. The calculated k_{eff} value is 0.6239 ± 0.0018.

6.4   SUMMARY

This section has helped you to accomplish the following:

- Use GeeWiz and KENO-VI to describe complex geometry models using arrays.

- Remove unwanted portions of arrays using region definition vectors to apply boundaries.

- Build hexagonal arrays.

7. NONSTANDARD MATERIALS

Now that you have an understanding of the geometry and basic standard composition input requirements, this section provides an explanation of some input options that are useful for problems involving compounds or other materials not available in the Standard Composition Library.

7.1   WHAT YOU WILL BE ABLE TO DO

- Define a user-defined (nonstandard) material and use GeeWiz to enter data for it.

- Define a mixture by using a combination of user-defined material and a basic standard composition.

7.2   USER-DEFINED MATERIALS

In Sect. 3, you learned how to enter information for elements, isotopes, and compounds found in the Standard Composition Library. However, there are some materials, particularly compounds, that are not available in the Standard Composition Library in SCALE. For these materials, you need to use either the **Alloy or Mix (Wt %)** or the **Compound (atom)** types from the **New Composition** form as shown in Fig. 124. This approach works best for materials for which the chemical composition or weight fraction is provided. If number densities are available for each of the constituents in a compound, then use the basic standard composition input to enter each nuclide with its appropriate atom density. Materials with weight fractions can be done with either approach but are best entered as basic standard compositions if they appear in that library.
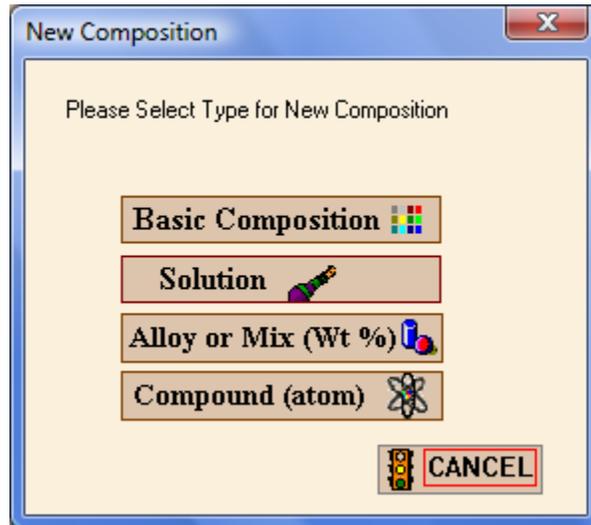
Fig. 124. New compositions type form.

7.2.1 Description of User-Defined Material with Chemical Formula

For this example, we will enter the material information for hydraulic fluid, C_2H_6SiO, with a density of 0.97 g/cc. Click on the **Compound (atom)** type from the **New Composition** form. (**NOTE:** The first four characters of the **Composition name** must be `atom` for a user-designated compound.) For **Composition name,** enter `_oil;` the "atom" keyword will be added at the beginning to create a material name of `atom_oil`. Now enter the **Density** of `0.97` g/cc, and leave the **Density Multiplier** and **Temperature** at the default values. For this material, we know the chemical composition, so we will enter information for each nuclide in terms of atoms per molecule. Select the **Insert Nuclide** box. For each element, select the element name, and then enter the atoms per molecule. For example, select `c` for the **Element Name** and enter `2` for **Atoms Per Molecule**. The form for carbon should look like Fig. 125.
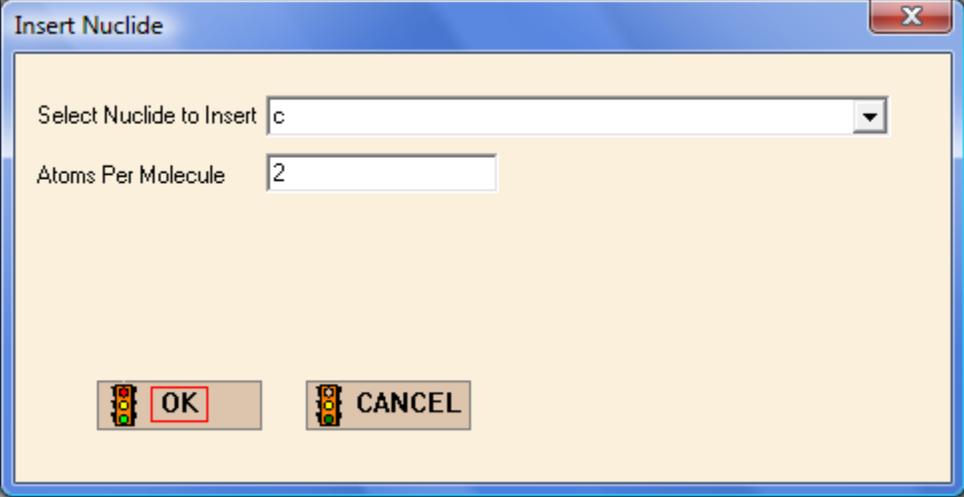
Fig. 125. Insert nuclide form.

Continue with hydrogen, silicon, and oxygen. The completed form should look like Fig. 126.

Fig. 126. Compound composition input for hydraulic fluid.

7.2.2 Description of User-Defined Material with Weight Percents and Isotopic Abundance

For this example, we will enter the material information for borated aluminum that is 2.5 wt % boron and has a density of 2.65 g/cc. The boron is enriched to 90 wt % B-10 and 10 wt % B-11. For this material, we know the weight percents. Therefore, we want to enter information in terms of weight percent, which is the **Alloy or Mix (Wt %)** type. For **Composition name** enter `_boral` after **wtpt** to create the material `wtpt_boral`. Now enter the **Density** of `2.65` g/cc and leave the **Density Multiplier** and **Temperature** at the default values. Select the **Insert Nuclide** box. For each nuclide, select the nuclide name and then the weight percent or select **Fill to 100%** if it is the last nuclide. For example, if the information for boron (wt % = 2.5) has already been entered, then select `al` for the **Nuclide** and then move the cursor to the **Weight Percent** box. Then select the **Fill to 100%** option, and GeeWiz will put `97.5` in the **Weight Percent** box as shown in Fig. 127.
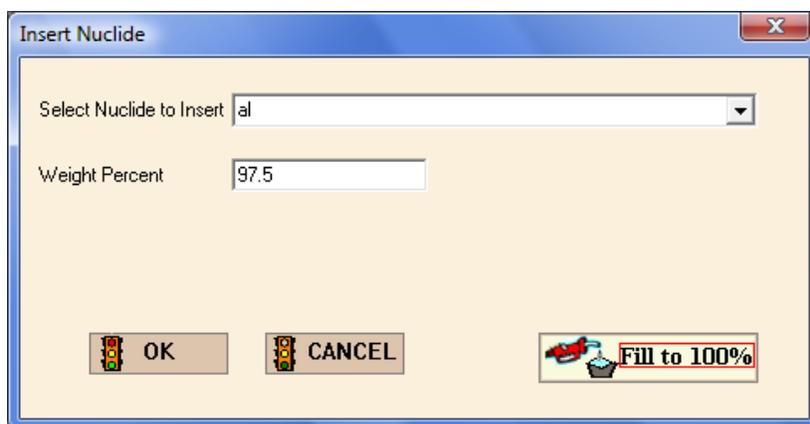
Fig. 127. Weight percent and fill to 100% option.

After entering the data for both boron and aluminum, we need to enter the isotopics for boron. To open the Isotopic Distribution window, click on "b 5000" in the element list. The default isotopic distribution is that for natural boron, 18.4309% B-10 and 81.5691% B-11. For this material we need to change the composition to 90 wt % B-10 and 10 wt % B-11. Click on **Edit Selection**. This opens the **Isotopic Distribution** form, where you can enter `90.0` for `5010` (B-10) and then **Fill to 100%** for `5011` (B-11). After doing this, click **OK** on the **Isotopic Distribution** form. The completed WTPT form should look like Fig. 128. Click **OK** to save this information.
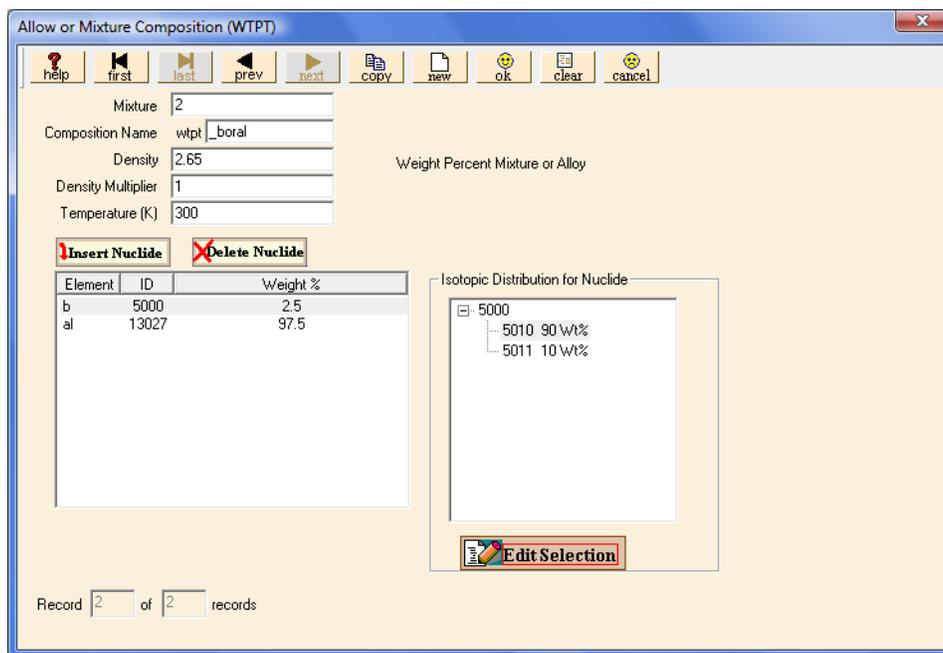
Fig. 128. Weight percent form for borated aluminum.

Note that you cannot mix atom-per-molecule entries with nuclide weight percents. However, if the isotopics of elements in a compound are given in weight percent, then you can enter the atom-per-molecule data for the nuclides in the compound and then use the isotopic distribution box to enter the proper isotopic weight percents.

140

7.2.3 Combinations of Basic Standard Compositions and User-Defined Materials to Define a Mixture

For this example, we will enter the material information for 1000 parts per million (ppm) of boric acid in water at the nominal density of water. Note that water (H_2O) is in the Standard Composition Library but boric acid (H_3BO_3) is not. We will specify the borated water mixture by using a basic standard composition for the water and a user-defined compound type of material for the boric acid.

For water, click on **Create** and **Basic Composition**. For **Composition Name** select `h2o`. We will leave the temperature and density at their default values. We must adjust the density to account for the boric acid. Subtracting the weight fraction of the boric acid (1000 ppm, or 0.001) gives the weight fraction of water as 1.0 − 0.001 = 0.999. Enter `0.999` for the **Density Multiplier**. The completed form should look like Fig. 129. After checking the information, press **OK**.

Fig. 129. Basic composition water to be combined with boric acid.

We know the chemical composition of boric acid, so click on **Create** and **Compound (atom).** Change the **Mixture Number** to be the same as the mixture number (3) for H_2O, so that the boric acid will be part of the same mixture as the water. For **Composition Name** enter `_boric_acid`. Again, the first four characters of the **Composition Name** will be `atom`. Enter the atoms per molecule information for each nuclide. To start, select `h` for the **Nuclide** and enter `3` for **Atoms/Molecule**. Continue with boron and oxygen. Enter the **Density** of the mixture as the default water density of `0.9982 g/cc`. We need to enter the weight fraction, 1000 ppm (i.e., `0.001`), as the **Density Multiplier**. We will leave the **Temperature** at its default value. The completed information should look like Fig. 130.

Fig. 130. Compound composition for boric acid (H_3BO_3) in water.

This completes the description of the borated water mixture using a user-defined material and a basic standard composition. Remember that you can have any number of materials with the same composition number as long as you assign each one the density of the entire mixture and have properly handled the density multipliers to give the correct weight fraction of each material in the mixture.

7.3   SUMMARY

This section has shown you how to use GeeWiz to accomplish the following:

- Define a user-defined material based on a known chemical formula.

- Define a user-defined material for a mixture with known weight percents of constituents.

- Define a mixture by using a combination of a user-defined material and a basic standard composition.

8.   UNIT CELLS AND BOUNDARY CONDITIONS

Note that the use of unit cell data in this section applies only to the use of multigroup cross-section libraries. If you are using continuous-energy cross-section data, the unit cell information in this section is not applicable. However, you can still use these problems (without the unit cell data) as practice for setting up continuous-energy KENO-VI models.

The examples we have used thus far have been limited to systems with relatively large homogeneous components (i.e., the size of homogeneous fissile material is large relative to the neutron mean free path). For such systems, the default INFHOMMEDIUM (infinite homogeneous medium) unit cell type was appropriate. In this section, we will look at more heterogeneous systems and learn how to use the other unit cell types appropriately for resonance self-shielding of multigroup cross sections.

8.1   WHAT YOU WILL BE ABLE TO DO

- Understand the difference in multigroup cross-section processing for Infinite Homogeneous Medium, Lattice Cell, and Multiregion Analyses.

- Identify the appropriate unit cell type to use for various heterogeneous systems.

- Use GeeWiz to define LATTICECELL and MULTIREGION unit cells.

- Define different boundary conditions for a MULTIREGION unit cell.

- Use boundary conditions to create reflective boundaries for a KENO-VI global unit.

8.2   CALCULATION TYPES FOR PROBLEM-DEPENDENT CROSS SECTIONS

The Material Information Processor is used primarily to prepare data for creating the problem-dependent cross-section libraries required by the criticality safety sequences. This is particularly important for heterogeneous systems and large regular lattices of slabs, pins, or spheres. The Material Information Processor utilizes a 1-D 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.

8.3 UNIT CELL TYPES

The Material Information Processor offers four types of unit cell calculations: INFHOMMEDIUM, LATTICECELL, MULTIREGION, and DOUBLEHET. Note that you may define multiple unit cells in a single CSAS6/KENO-VI input file. Each mixture number may be used in only one unit cell.

The DOUBLEHET unit cell type is needed only for doubly heterogeneous fuels such as pebble-bed fuel, where individual grains of fuel are coated with moderator material and then placed in a matrix that is surrounded by another layer of moderator material and placed in a larger matrix. Due to the complexity and the rarity of these applications, the DOUBLEHET unit cell type will not be covered in this primer.

8.3.1 INFHOMMEDIUM (Infinite Homogeneous Medium)

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

8.3.2 LATTICECELL

The LATTICECELL treatment is appropriate for large arrays of fuel in slabs, pins/rods, or spherical pellets. The resonances are treated differently, depending on whether the NITAWL processor or the CENTRM processor is specified. When the LATTICECELL treatment is used with NITAWL, the mixtures specified in the cell have resonance self-shielding with Dancoff corrections applied to the cross sections of the nuclides utilized in the cell. When the LATTICECELL treatment is used with CENTRM/PMC, a set of point fluxes are calculated for a set of mesh intervals that make up the unit cell. The cell is assumed to have white boundary conditions (see Sect. 8.3.3.1 below) to simulate an infinite lattice. The isotopes comprising the mixtures specified in the cell are then collapsed using these point fluxes and point cross sections. Limitations of the LATTICECELL treatment include the following:

1. The cell description is limited to a 1-D cell. Most physical systems consist of 3-D geometry. However, in many instances, a 1-D 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 1-D representation may be inadequate for systems exhibiting a 2-D or 3-D spatial dependence.

2. The LATTICECELL treatment assumes an infinite array of 1-D 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 where multidimensional spatial dependence or leakage can become important.

146

8.3.3   MULTIREGION

The MULTIREGION treatment is appropriate for geometric regions in which the geometry effects may be important but the infinite homogeneous treatment or lattice cell treatment is not general enough. The MULTIREGION unit cell allows more flexibility in the placement of the fuel but 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.

3. A MULTIREGION cell is limited to a 1-D 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.

4. 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 the cuboidal outer region and specifying a white outer boundary condition.

5. 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 are not valid on a curved outer surface.

6. A MULTIREGION cell represents a single cell if the outer boundary has a vacuum boundary condition applied to it. In CENTRM, 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 point cross sections.

7. If the outer boundary of a MULTIREGION cell has a boundary condition other than vacuum, the unit cell approximates an array. In CENTRM, a point flux spectrum is calculated over a set of intervals that make up the unit cell with the incoming flux equal to the outgoing flux on the boundary. This flux spectrum is then used to collapse a set of point 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: If you want to apply boundary conditions to your unit cell model, then you must use the MULTIREGION cell treatment.

8.3.3.1   Boundary Conditions

For the MULTIREGION 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`.

Vacuum Boundary Condition

A vacuum boundary condition means that no neutrons will reenter the boundary.   Thus, any neutron exiting the system through a vacuum boundary is permanently lost to the system. This condition is shown in Fig. 131. (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.)

no return current — vacuum boundary condition on left side
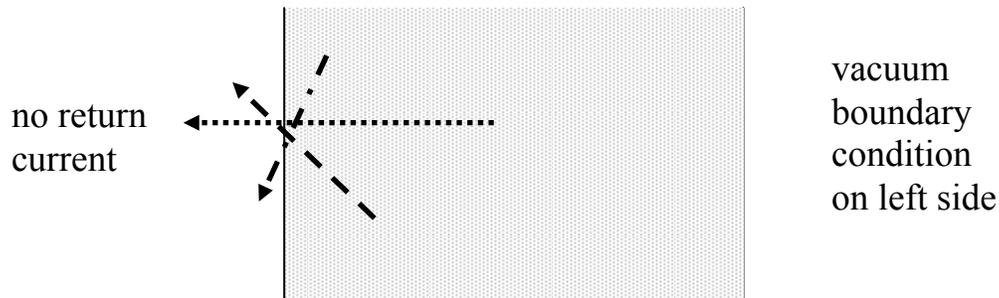
Fig. 131. Vacuum boundary condition.

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 or inner boundary of a cylindrical or spherical system. As shown in Fig. 132, a given quantity of neutrons leaving a boundary (dotted line) in a particular direction will be returned (solid line of same color) to the system with the same quantity but at a mirrored angle to the initial leakage direction.

return as mirror reflector — reflective boundary condition on both sides
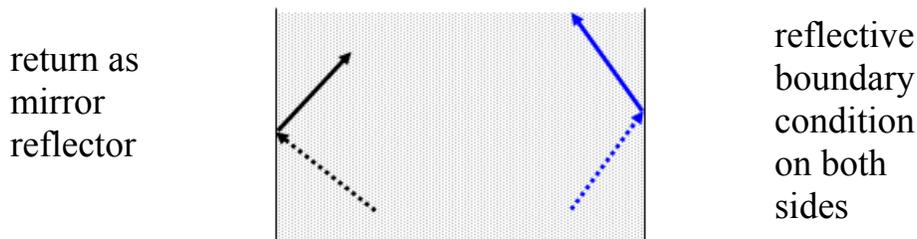
Fig. 132. Reflective boundary condition.

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 133 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, that is, either boundary of a cylinder or a sphere.

return at same angle as leaving other boundary

periodic boundary condition on both sides
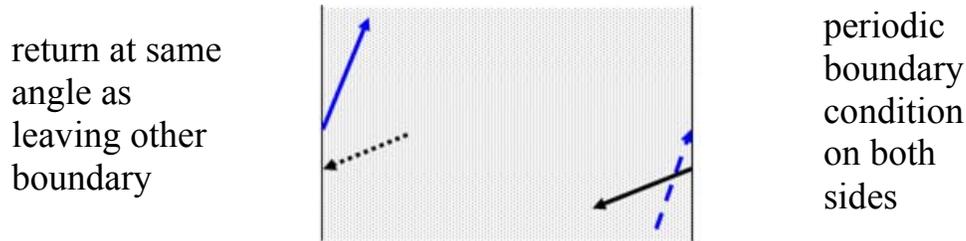
Fig. 133. Periodic boundary condition.

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 Fig. 134) 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 side
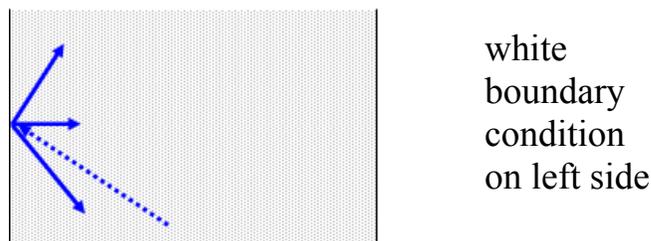
Fig. 134. White boundary condition.

8.4   UNIT CELL SPECIFICATIONS

Unit cell data are not required for IHFHOMMEDIUM. With the LATTICECELL and MULTIREGION treatments, you must specify the unit cell dimensions and materials to be used in the resonance self-shielding process. The unit cell is always 1-D: infinitely long cylinders,

infinite slabs, or spheres. Finite cylinders or slabs can be approximated with geometric buckling in MULTIREGION cells. The configuration of the unit cell depends on the treatment and the type of lattice (if the LATTICECELL treatment is used).

8.4.1   LATTICECELL Unit Cell Data

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 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. A value for CELLMIX= must be specified in the unit cell data if cell-weighted cross sections are used in the problem.

8.4.2   MULTIREGION Unit Cell Data

For the MULTIREGION treatment, the unit cell is more flexible 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., only concentric regions of the same shape are allowed). MULTIREGION does not account for lattice effects, so it is best used for problems where lattice effects are not important.

8.4.3   Cell-Weighted Cross Sections

Cell-weighted cross sections are created by XSDRNPM when CELLMIX= is specified in the unit cell data. 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 to cell weight cross sections, only the cell-weighted mixture number can be used in the KENO-VI geometry data. The original mixtures used in the unit cell description cannot be used. Refer to (Sect. M7.5.12 of the SCALE Manual) for more information on cell-weighting.

8.5   PROBLEM DESCRIPTIONS

Two sample problems will be used to demonstrate how to enter unit cell data for LATTICECELL and MULTIREGION unit cell types using GeeWiz. 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.

8.5.1   Lattice Cell Example: 9×9 Fuel Assembly

This problem is a single fuel assembly at 3.44 wt % enrichment stored in a pool of water at a temperature of 300 K. There is 15.24 cm of water above and below the assembly shroud, as well as 7.62 cm of water on each side of the assembly shroud. The inside dimensions of the shroud are 14.96 cm in the X- and Y-dimensions and 420.68 cm in the Z-dimension. The space between fuel rods is full-density water. The shroud is 304 stainless steel, 0.16 cm thick on all faces. The array of fuel rods and poison rods comprising the assembly is centered within the shroud.

The fuel rods are 0.94 cm in diameter and contain UO_2 at 92.5% of theoretical density. The active fuel length is 365.76 cm. Each fuel rod is contained within Zircaloy-2 cladding with an inner radius of 0.4875 cm and an outer radius of 0.545 cm. The cladding extends 16.96 cm above the top of the active fuel and 23.36 cm below the bottom of the active fuel. The ends are capped with solid SS304 plugs having the same outside diameter as the cladding. The top plug is 10.5 cm thick, and the bottom plug is 4.1 cm thick. The axial configurations of the poison rods and the fuel rods are shown in Fig. 135.

Axial Rod Layout
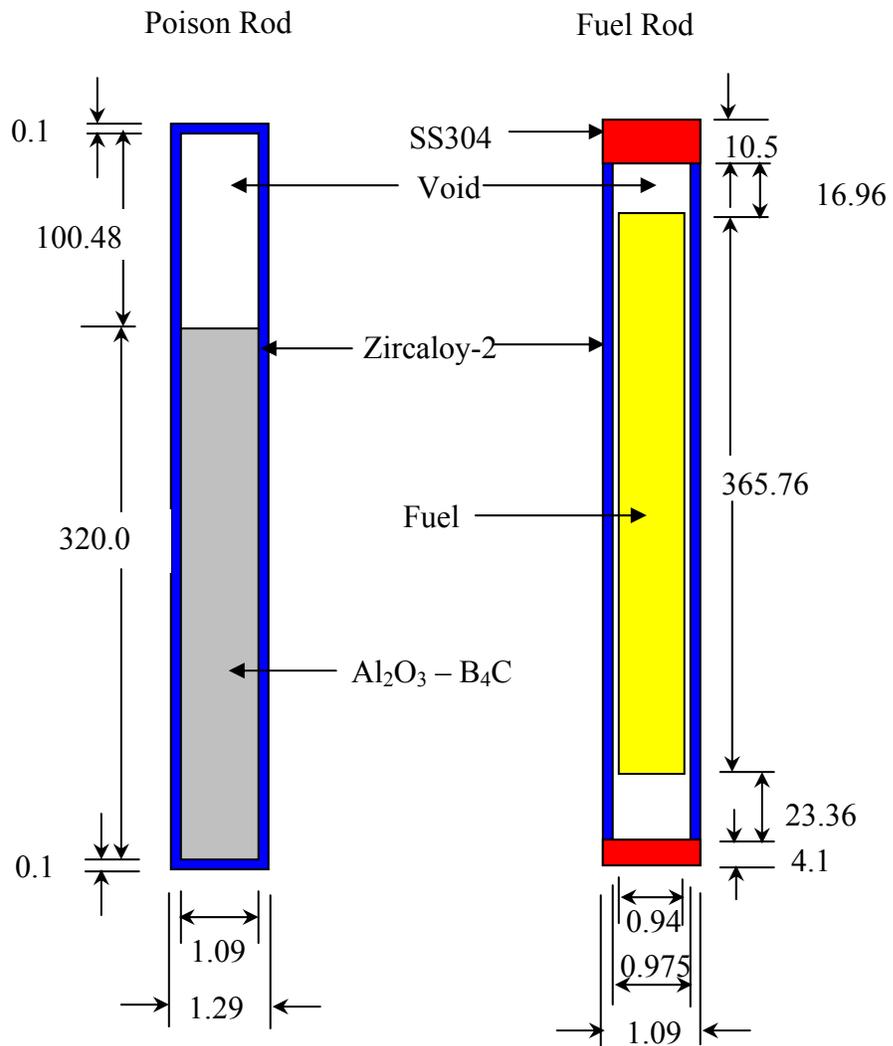
Fig. 135. Axial layout of assembly rods.

The fuel assembly has a water-filled central instrumentation hole with eight burnable poison rods in various lattice positions, as shown in Fig. 136. The burnable poison rods are composed of Al_2O_3-B_4C, having a density of 3.7 g/cc and are 1.260 wt % B_4C. The boron in the B_4C is natural boron. The burnable poison rod has a diameter of 1.09 cm and an active poison length of 320.0 cm. A 100.48 cm void space extends above the active poison rod. The poison rod and void space have a tight-fitting 0.1 cm thick Zircaloy-2 cladding. The rod pitch within the assembly is 1.64 cm.
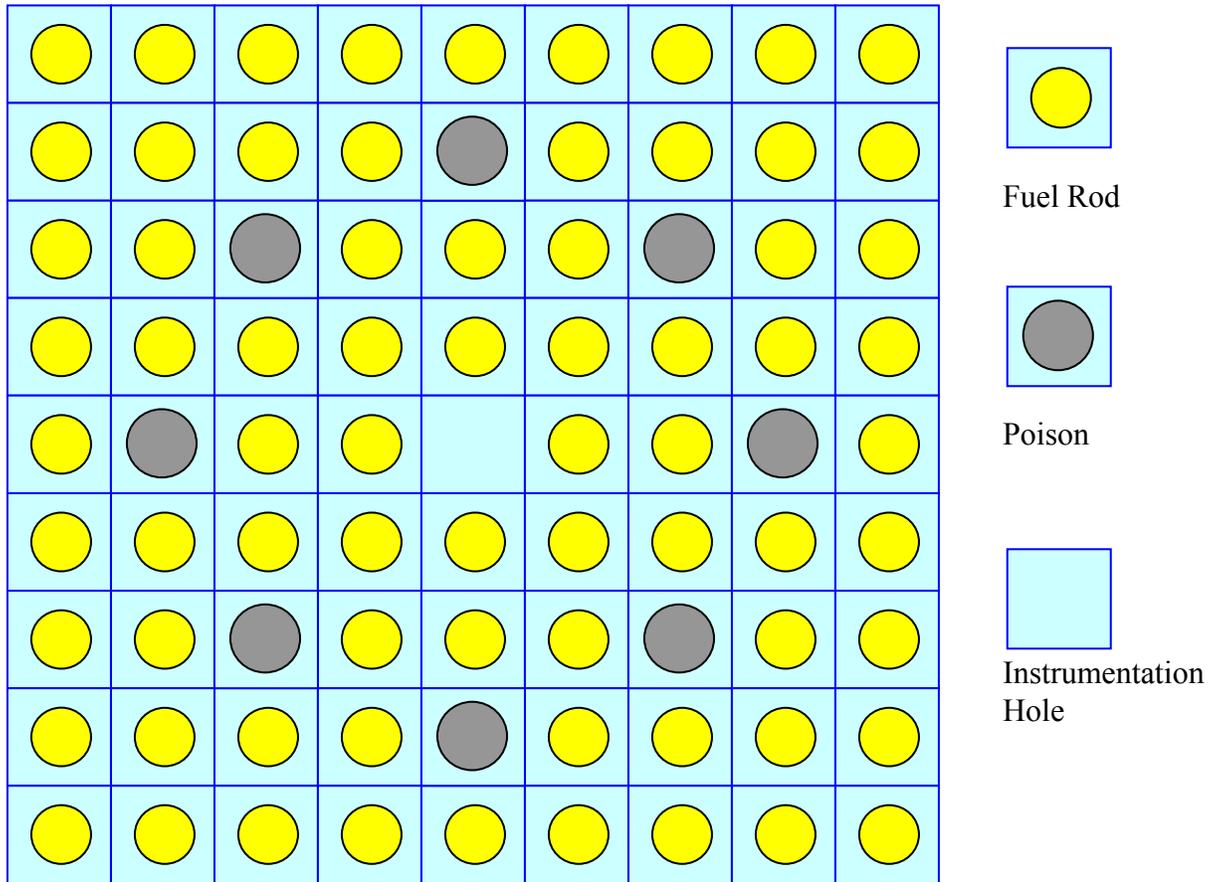
Fig. 136. Plan view of rod positions in 9 × 9 assembly.

8.5.1.1 General and Material Data

The initial data entry for this example is the same as in previous problems. Start the GeeWiz user interface, and then select the **New** button from the top toolbar. Select the **General** button on the left-hand side, and enter the problem **Title**, **Application**, and the SCALE **Sequence**. Then select `v7-238` as the **Cross-section Library.** The **General Information** window should look like that in Fig. 137.

Now we need to enter the **Compositions** information. For four of the materials, Basic Standard Compositions can be used: `h2o` (Mixture No. 2), `uo2` (Mixture No. 3), `zirc2` (Mixture No. 4), and `ss304` (Mixture No. 5). You should be able to enter the information for these from the problem statement. The default theoretical densities are used for all. All materials have a **Density Multiplier** of `1.0` except `uo2`, which has a **Density Multiplier** of `0.925`. Also remember that the `uo2` is 3.44 wt % ^{235}U, so the isotopics need to be entered for this material.
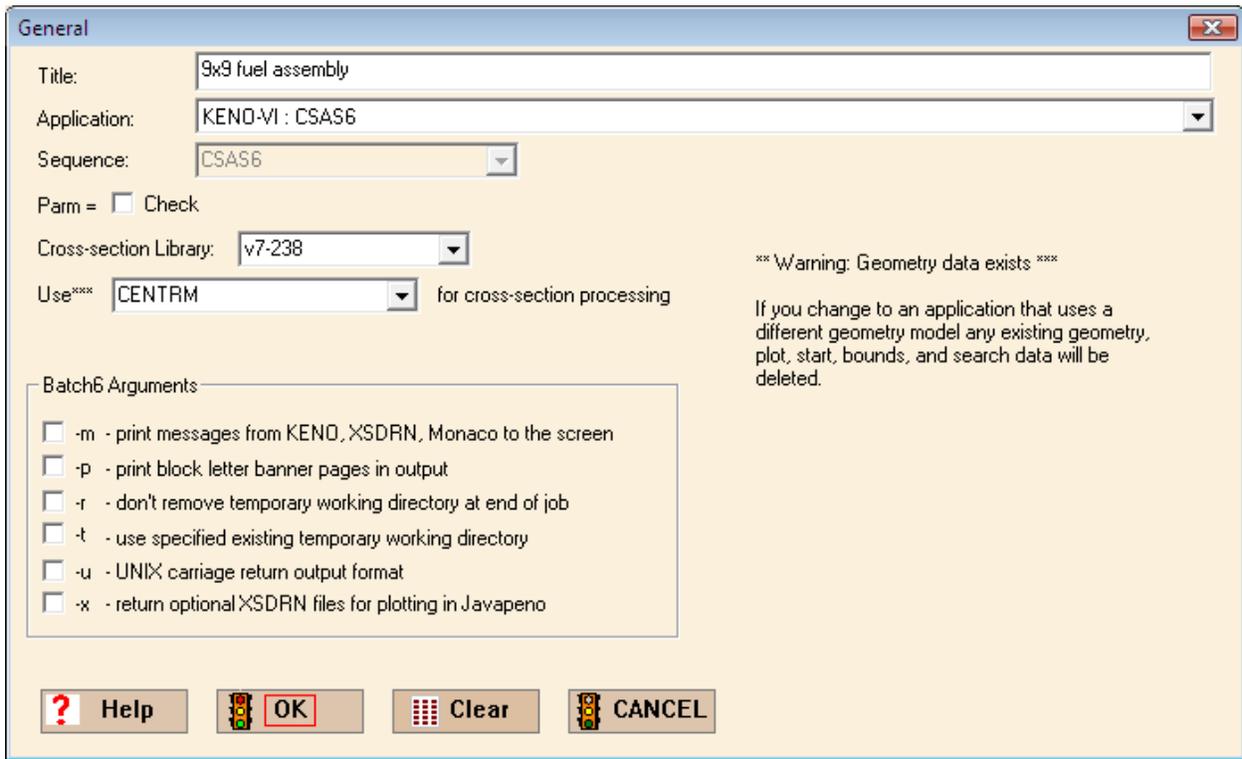
Fig. 137. General information for lattice problem.

For the burnable poison, we will enter the data as two compositions with the same mixture number. The burnable poison has a density of 3.7 g/cc and is a combination of B_4C and Al_2O_3. The material is 1.260 wt % B_4C with natural boron. We will specify this mixture using a compound material for Al_2O_3 and a basic composition for B_4C. From the composition summary screen, select **Create** and then select **Compound (atom)**. For **Mixture,** enter `1.`   (We will call this mixture 1.) For **Composition Name,** enter `_al2o3`. Now for the **Density**, enter the mixture density of `3.7` g/cc with a **Density Multiplier** of $(1 - 0.0126) = 0.9874.$ Then select **Insert Nuclide** to enter the information for `al` (2 atoms per molecule) and for `o` (3 atoms per molecule). When completed, the input form should look like Fig. 138. Click **OK**.

Fig. 138. Data for Al_2O_3 portion of burnable poison mixture.

For the B$_4$C, click the **Create** button, then select **Basic Composition** as the type. Change the **Mixture Number** to `1` so that it will be part of the burnable poison mixture. For **Composition Name** select `b4c`, and enter a **Density Multiplier** of `0.0126.` Because the boron is natural, the **Isotopic Distribution** does not need to be altered. Select **User Supplied (g/cc),** and enter the mixture **Density** of `3.7` g/cc. The completed information should look like Fig. 139. Click **OK**.

Fig. 139. B$_4$C portion of burnable poison mixture.

8.5.1.2 Unit Cell Data

Now that we have entered the material information, we are ready to enter the unit cell information for the cross-section treatment. Selecting the **Cell Data** button on the left-hand toolbar will open the **Unit Cell Data** window, where you enter the unit cell data. By default, the **Lattice Cell** tab is active. To create a new lattice cell, click on **New Cell** to bring up the **Lattice Cell Data** form shown in Fig. 140. The first entry is **Type of Lattice**. The menu has a number of choices; those starting with the letter `A` are for annular lattices. In this problem, we have a regular square pitch lattice, so select `SquarePitch` from the menu. Now we need to give the dimensions of the unit cell and the materials for the fuel, clad, and moderator. First, enter `0.82` (=1.64/2) for the **Half Pitch (cm)** and `0.47` (=0.94/2) for the **Fuel Radius (cm).** Select the `3 uo2` material from the menu for **Fuel Mixture**. The moderator is water, so select the `2 h2o` material from the menu for **Moderator Mixture.** The fuel rods have both a gap and a clad; therefore, we need to check the **Clad/Gap Exists** box. Checking this box activates data entry for the **Clad** and the **Gap.**

Fig. 140. Lattice cell data.

For the clad, enter `0.545` (=1.09/2) for the **Clad Radius (cm)** and select material `4 zirc2` for the **Clad Mixture**. For the gap, enter `0.4875` (=0.975/2) for the **Gap Radius (cm)** and select material `0 void` for the **Gap Mixture**. Note that the **Gap Radius** is the inner radius of the clad. The completed Lattice Cell form should look like Fig. 140.

Now click **OK** to see the information summarized on the **Unit Cell Data** screen shown in Fig. 141. This completes the Unit Cell Data, so save the information by clicking on **Close**.
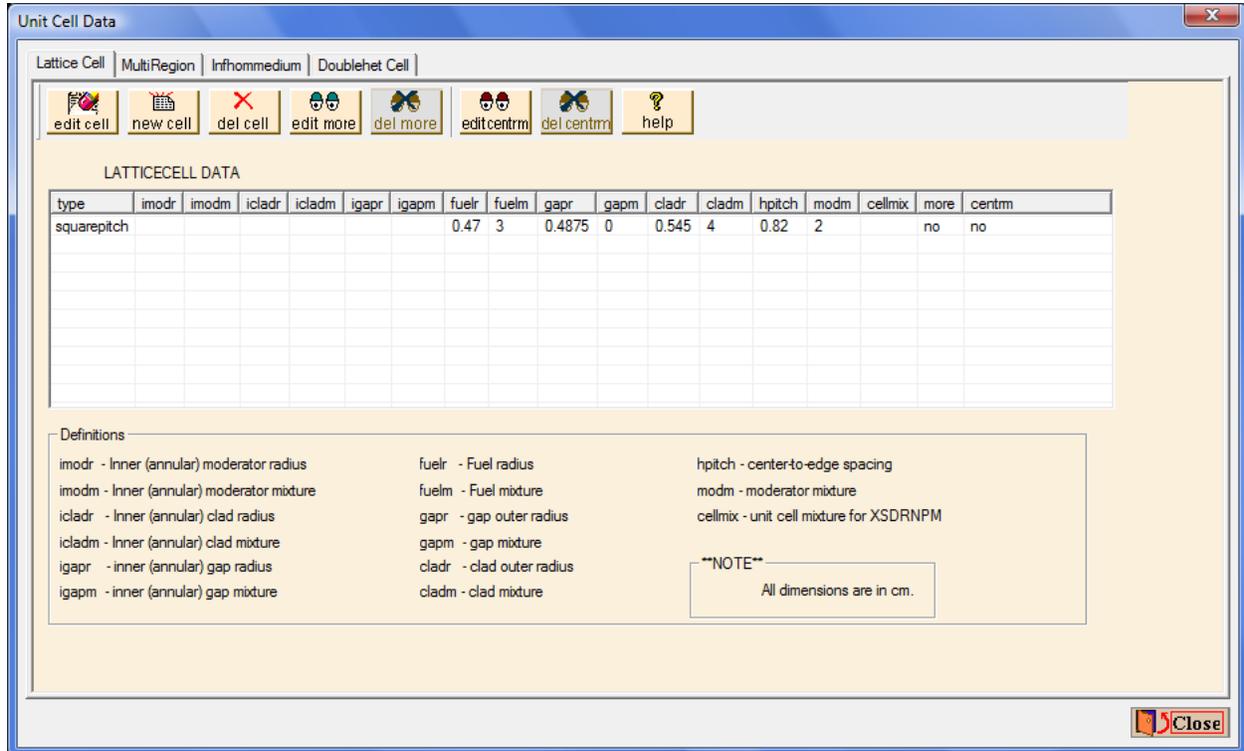
Fig. 141. Unit cell data.

8.5.1.3 KENO-VI Data

For the KENO input, select the **Geometry** button. In this problem, we will define four units as follows. Unit 1 will be a fuel rod cell; Unit 2 will be a burnable poison rod cell, Unit 3 will be the instrumentation tube cell, and Unit 4 will be the global unit containing the array of rods and the surrounding shroud and water.

We will start with unit 1. The first shape is the fuel cylinder, which is then enclosed by the gap cylinder, and then the clad cylinder. Enter the data for these three shapes using the dimensions from Fig. 135. The fuel rod has end caps made from SS304; they require a fourth cylinder enclosing the previous three cylinders. The radial dimension of this cylinder is the same as that of the clad cylinder, and the additional height is to account for the cap thickness at the top and bottom of the rod. Remember that any unit used in an array must have flat

surfaces where it touches other array elements; thus, we need a fifth shape, a cuboid that encloses the fuel rod and contains the moderator (water). The X- and Y-dimensions are ± 0.5 * pitch, while the Z-dimensions are the same as those for the SS304 end caps. After the media and boundary records have been entered, the Geometry form should look like Fig. 142.
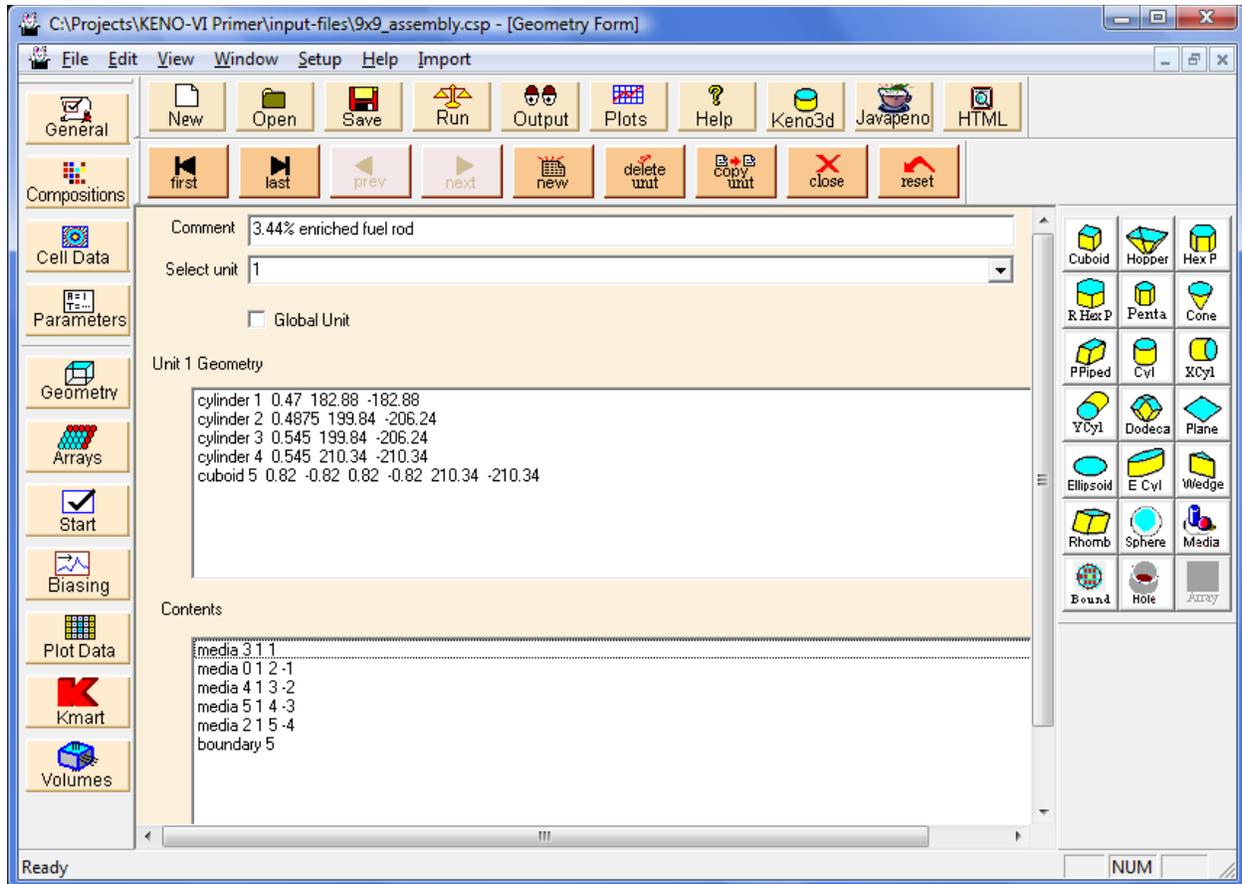
Fig. 142. Geometry information for a fuel rod.

Unit 2 is the burnable poison rod. This unit is handled like unit 1 but with three cylindrical shapes instead of four. (The end caps are made of the same material as the cladding.) Enter the data using the dimensions from Fig. 135. Now enclose the cylinders in a cuboid containing the moderator. The X- and Y-dimensions are again ± 0.5 * pitch, while the Z-dimensions are the same as those for the `zirc2` cylinder. After the media and boundary records have been entered, the Geometry form should look like Fig. 143.
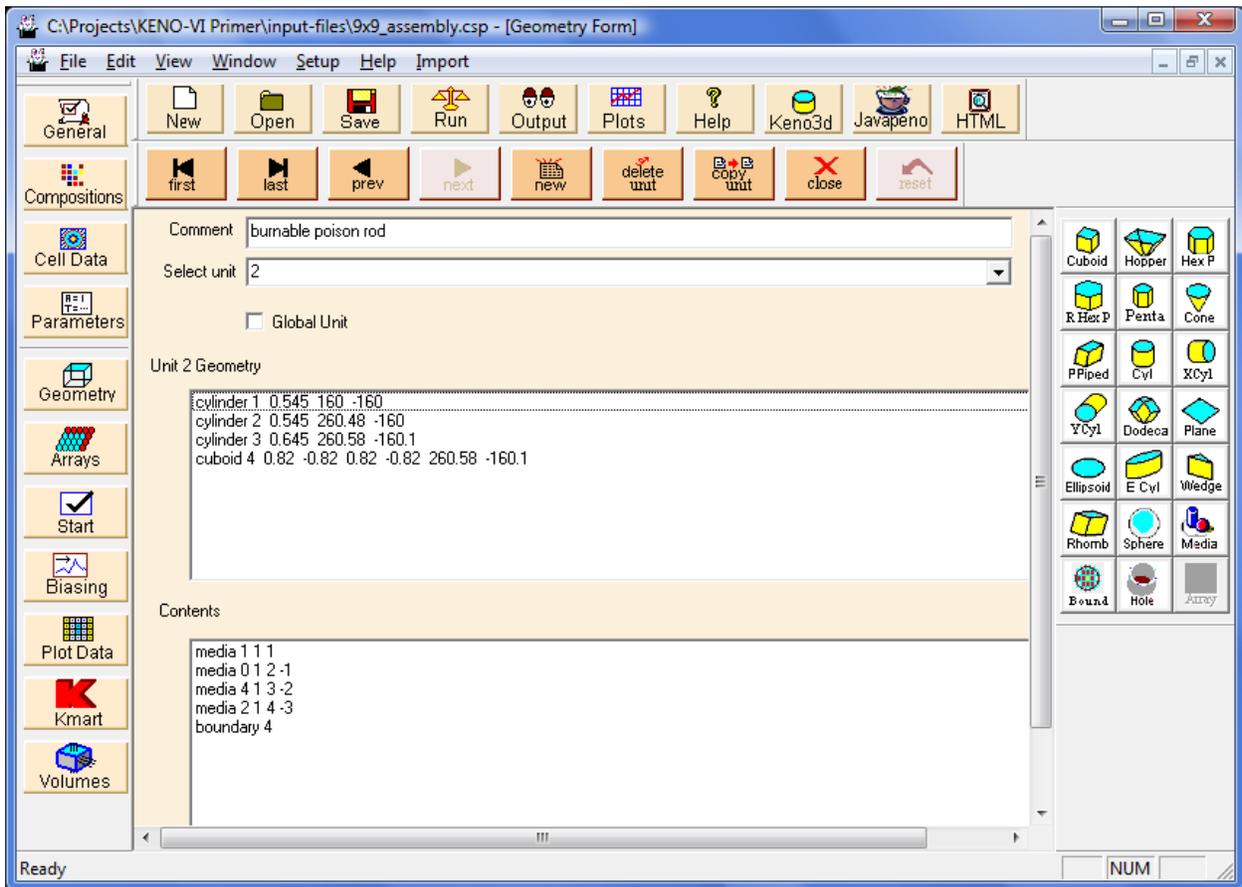
Fig. 143. Geometry information for burnable poison rod.

For unit 3, the geometry is quite simple. We have a cuboid containing water with the same dimensions as the cuboid from unit 1. Before defining unit 4, which is the global unit, we need to define the array. In this case, the array is nine units in the X-, nine units in the Y-, and one unit in the Z-directions. Open the **Arrays** window, and **Create** a `Square` array with **NUX** = 9, **NUY** = 9, and **NUZ** = 1 and check the **Global Array** box. Select Unit 1 as the unit to initialize new locations. Then, using Fig. 136, place the poison rods (unit 2) and the instrumentation tube (unit 3) into their appropriate positions. Provide a **Comment** for the array and then save the information. The screen should look like Fig. 144.
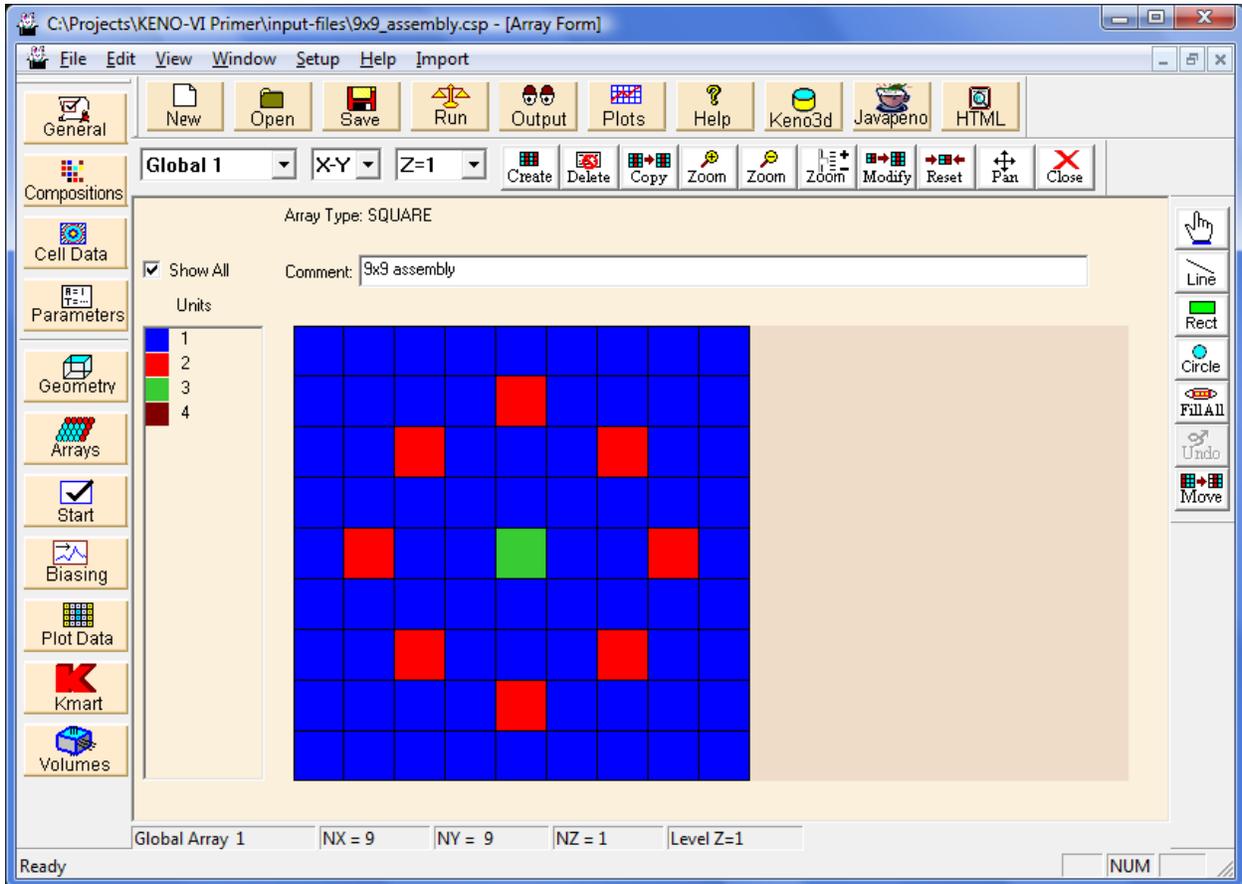
Fig. 144. Array layout for 9 × 9 fuel assembly problem.

Now we will enter the data for the global unit, unit 4. The first region in unit 4 will be a **cuboid** to contain the array. The dimensions of the cuboid must match the size of the array. We will define the dimensions as ± 7.38 (9 * the half-pitch) for X and Y and ± 210.34 (half of the fuel assembly height) for Z. These values place the origin at the center of the fuel assembly.

The next region is the cuboid containing the water inside the stainless steel shroud. The shroud is 14.96 cm in width (inside dimension), so the X- and Y-dimensions will be ±7.48 while the Z-dimensions are ±210.34 (half of the fuel assembly height). The SS304 shroud is a cuboid whose thickness is 0.16 cm greater than that for the cuboid just entered; thus, the dimensions are ±7.64 for X and Y and ±210.34 for Z.

The last geometry region contains the water outside the shroud. There is 15.24 cm of water above and below the assembly shroud, as well as 7.62 cm of water on each side of the assembly shroud. The cuboid dimensions are therefore ±15.26 in X and Y and ±225.58 in Z.

Next enter the media and boundary records data. The first record is the **Array** record. Select array 1 and **inside** cuboid 1. We will place the center array location (5, 5, 1) at the center of the cuboid (0, 0, 0). The array form should look like Fig. 145.

Fig. 145. Array record form.

Enter the media records for the surrounding regions and the boundary record. Check the **Global Unit** box and provide a **Comment** for this unit. The completed geometry form should look like Fig. 146.
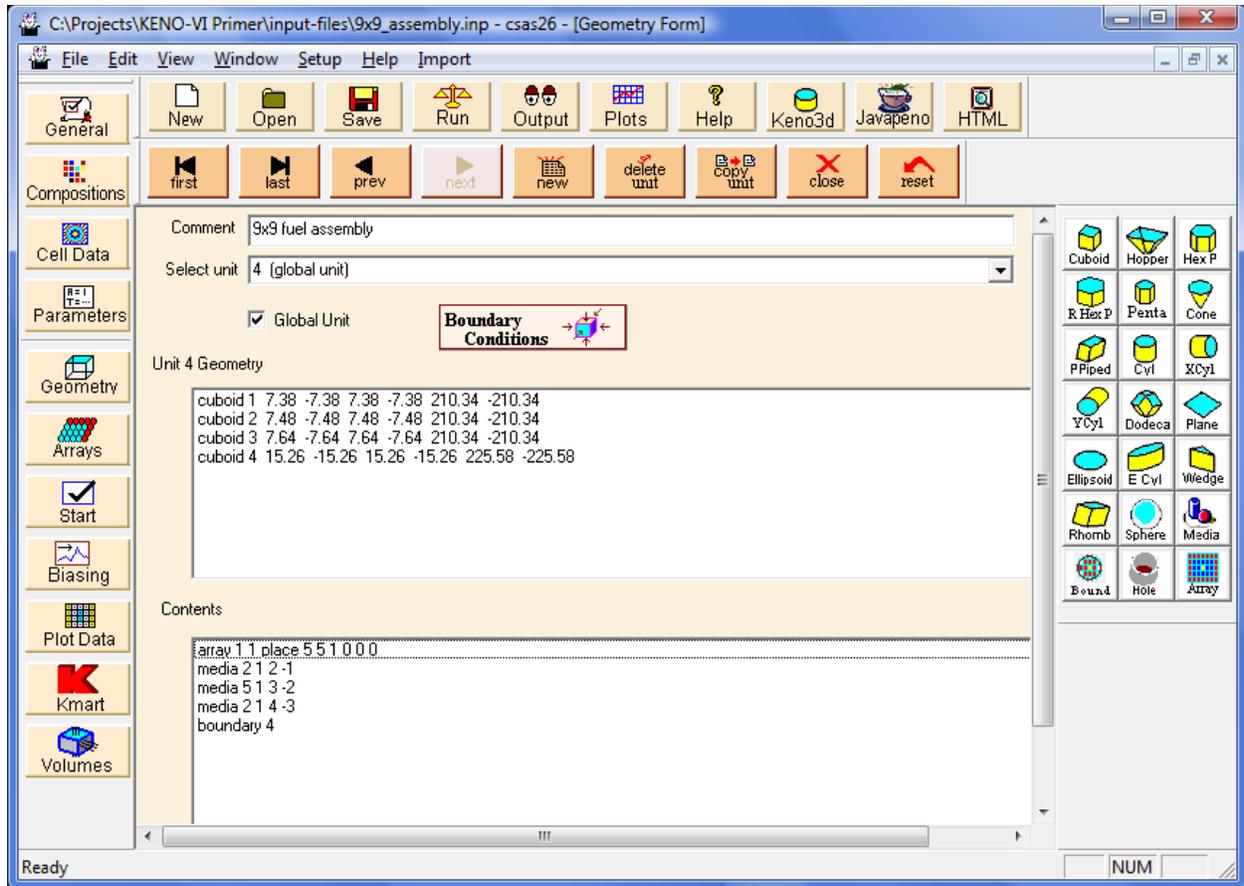
Fig. 146. Global unit for 9 × 9 fuel assembly.

Because this geometry model is more complex than previous examples, we will run 505 generations, skipping the first 5 generations. To change the run parameters, select **Parameters** from the left-hand side menu. This brings up the screen shown in Fig. 147. We will change the number of generations (**GEN=**) to `505` and the number of generations skipped (**NSK=**) to `5`. Then click **OK**.
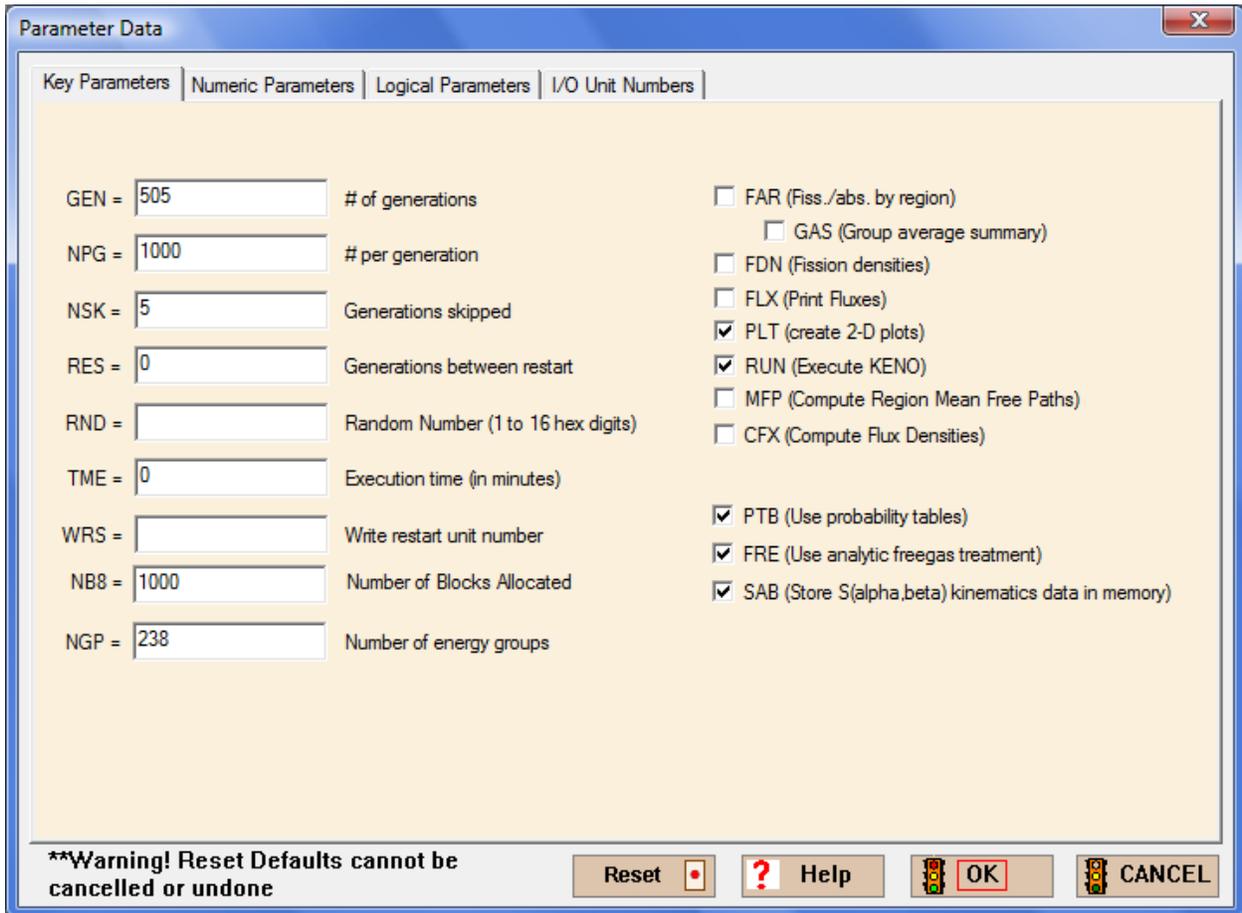
Fig. 147. Parameter data.

A cutaway view from KENO3D of the bottom portion of the completed assembly is shown in Fig. 148. This view was obtained by using the **Remove section** button in KENO3D twice and then zooming in. First the top half was removed (with the cutting edge for Z offset to -150), then the front right quarter.
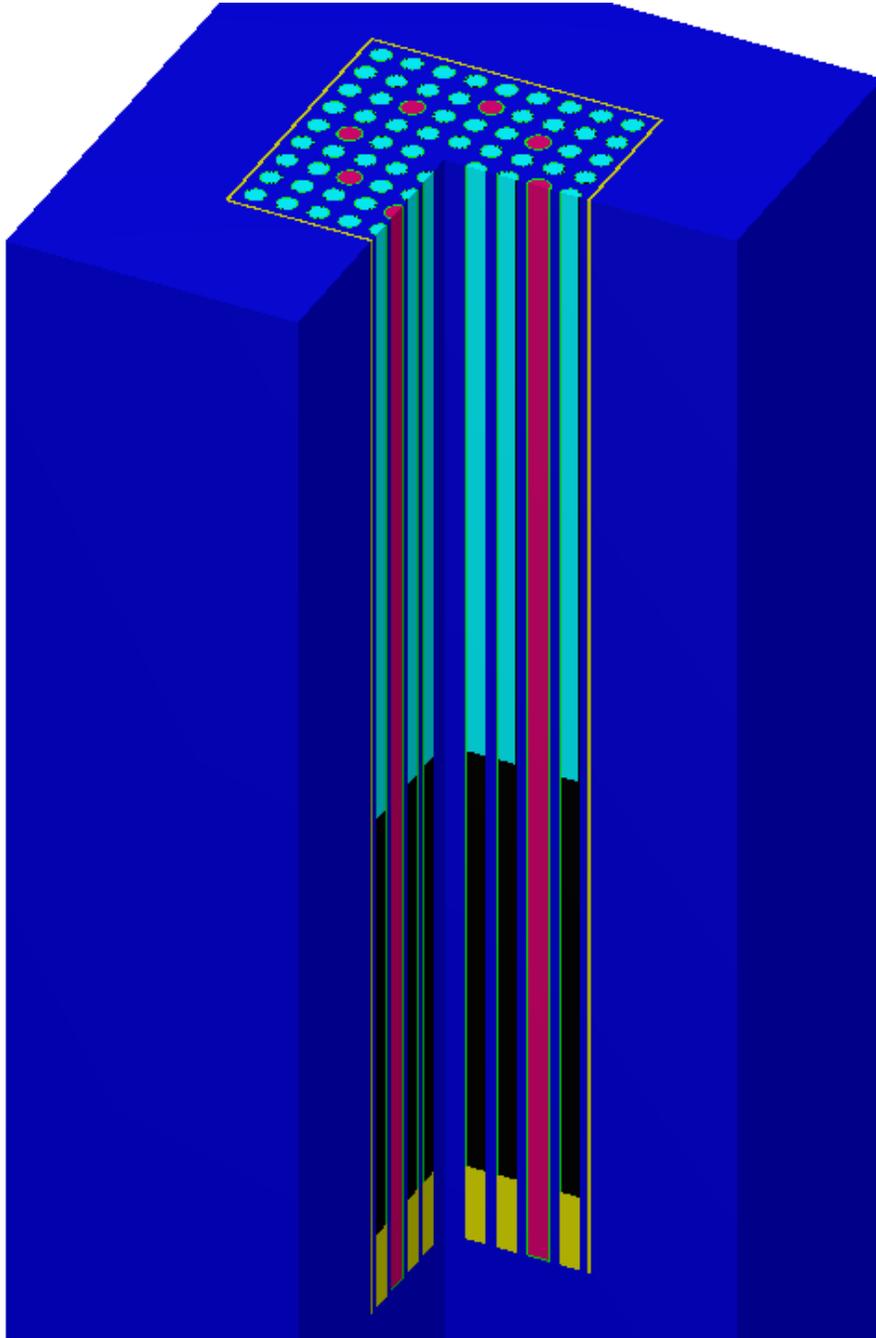
Fig. 148. 3-D view of bottom part of 9 × 9 array.

Save the changes in a new input file, and execute SCALE. For this problem, 505 generations were run to stabilize the k_{eff}. The calculated k_{eff} value is 0.6158 ± 0.0011.

8.5.2   Multiregion Example: Two Parallel Slab Tanks

This problem consists of two parallel slab tanks (Fig. 149) constructed of 0.5 cm thick SS304. The material in the tanks is U(93)O$_2$F$_2$, with a fuel density of 0.459 g/cc and a solution density of 1.566 g/cc. 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.
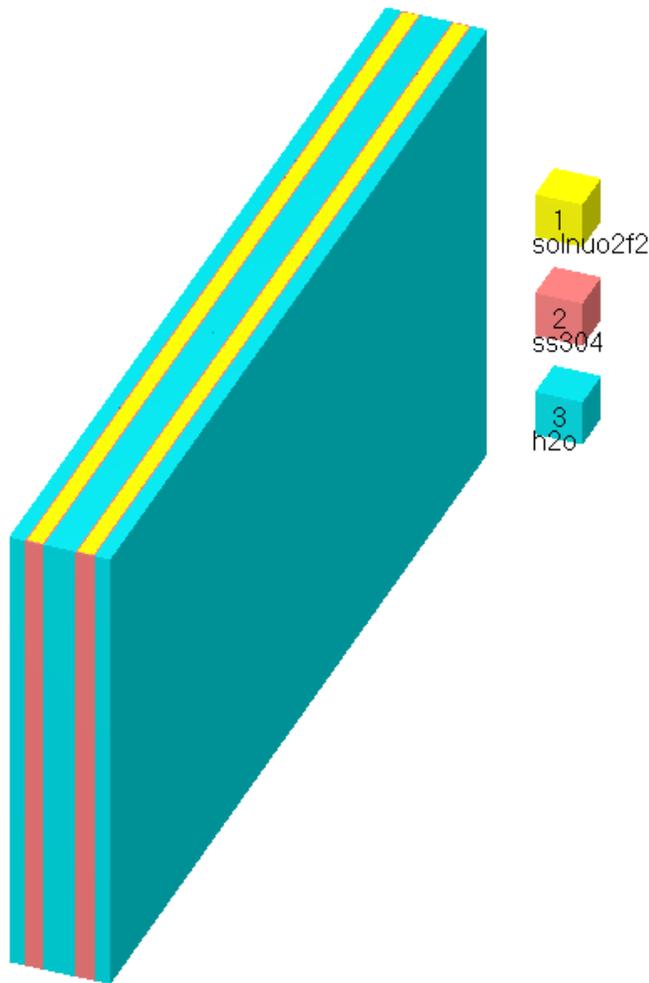
Fig. 149. Two parallel slab tanks.

8.5.2.1　General and Material Data

The initial data entry for this example is the same as that in the previous problems. Start GeeWiz, and select the **General** button. Enter the problem **Title**, the **Application,** and the SCALE **Sequence**. Then select `v7-238` as the **Cross-section Library.** The initial data entry for this example is the same as that in the previous problems.

Next enter the **Compositions** information. For the three materials, **Solution** is used for `uo2f2` (Mixture No. 1); **Basic Composition** is used for `ss304` (Mixtures No. 2 and 3) and for `h2o` (Mixtures No. 4 and 5). You must create two `ss304` and two `h2o` mixtures because a unique mixture number is required for each region in the unit cell.

You should be able to enter the information for these from the problem statement. Remember that the UO_2F_2 is 93 wt % ^{235}U, so the isotopics need to be entered for this material. The **solution density** is `1.566`. The molarity of `hfacid` is `0 moles/liter`, and the `uo2f2` density is `459 grams/liter`. The default theoretical densities are used for the other compositions.

8.5.2.2　Unit Cell Data

Now that we have entered the material information, we are ready to enter the unit cell information for the cross-section treatment. Selecting the **Cell Data** button on the left-hand toolbar will bring up the Unit Cell Data form. Click on the **MultiRegion** tab. The form should look like Fig. 150.

Fig. 150.　Unit cell summary form for multiregion cell.

Selecting **new cell** will open the **Multiregion Cell Data** form shown in Fig. 151. The first entry is **Geometry Type**. The menu has the three standard 1-D geometry choices and choices for a cylinder or a slab with leakage (`buckledcyl`, `buckledslab`). Because the slab tank height and width are very large relative to its width, the tank is similar to an infinite slab in leakage. We will select `Slab` for the geometry. For the **Left (Inner) Boundary Condition** there are four choices: `Periodic`, `Reflected`, `Vacuum`, or `White`. For this problem, we will use the `Reflected` boundary condition on the inside to utilize the symmetry of the slab tank. Thus, we create a model that starts at the center of the tank and goes outward to the right. Descriptions of the various boundary conditions can be found in the previous section and in (Sect. M7.5.6.3 of the SCALE manual). For the **Right (Outer) Boundary Condition,** select `Vacuum`, which is the default.

For the **Zone** information, you need to enter the mixture and position starting from the left or innermost and moving toward the right or outermost material. Remember that you are entering the information for a 1-D unit cell. Because the two tanks are symmetric, we will start with the centerline through the water between the two tanks and will work out to the external water reflector keeping the default boundary conditions of `reflected` and `vacuum`. For **Zone 1**, select `4 h2o` as the **Mixture No.**; then enter `5.0` for the **Distance from center-line (cm).** For **Zone 2,** select `2 ss304` as the **Mixture No.**, and enter `5.5` (i.e., = 5.0 cm water + 0.5 cm SS304) for the **Distance from center-line (cm).** For **Zone 3,** select `1 uo2f2,hfacid` as the **Mixture No.**; then enter `10.5` for the **Distance from center-line (cm)**. For **Zone 4,** select `3 ss304` as the **Mixture No.**, and enter `11.0` for the **Distance from center-line (cm)**. Finally for **Zone 5,** select `5 h2o` as the **Mixture No.**, and then enter `16.0` (i.e., = 11.0 + 5.0) for the **Distance from center-line (cm)**. The completed **Multiregion Cell Data** form should look like Fig. 151. Click **OK** and **Close** to close all unit cell forms.

Fig. 151. Multiregion unit cell description.

8.5.2.3 KENO-VI Data

Select the **Geometry** button on the left toolbar. In this problem, we will define one unit. Unit 1 will be a slab tank with fuel, stainless steel container, and water reflector. Because the two tanks are arranged symmetrically, we will use a mirror boundary condition to model the entire system.

The first region is the cuboid of fuel ($5 \times 300 \times 150$ cm), which is then enclosed by the SS304 tank (0.5 cm thick) on five sides (not including the top), and by the 5 cm water reflector in the X-direction. We will use the center of the slab tank as the origin for the dimensions. For the SS304 tank, we add 0.5 cm on all sides except +**Z**. Create three cuboids for the solution, the tank, and the reflector. Then create three media records for these. Finally, create the boundary record.

Enter a **Comment,** and check **Global Unit.** The Geometry form for the completed unit should look like Fig. 152.
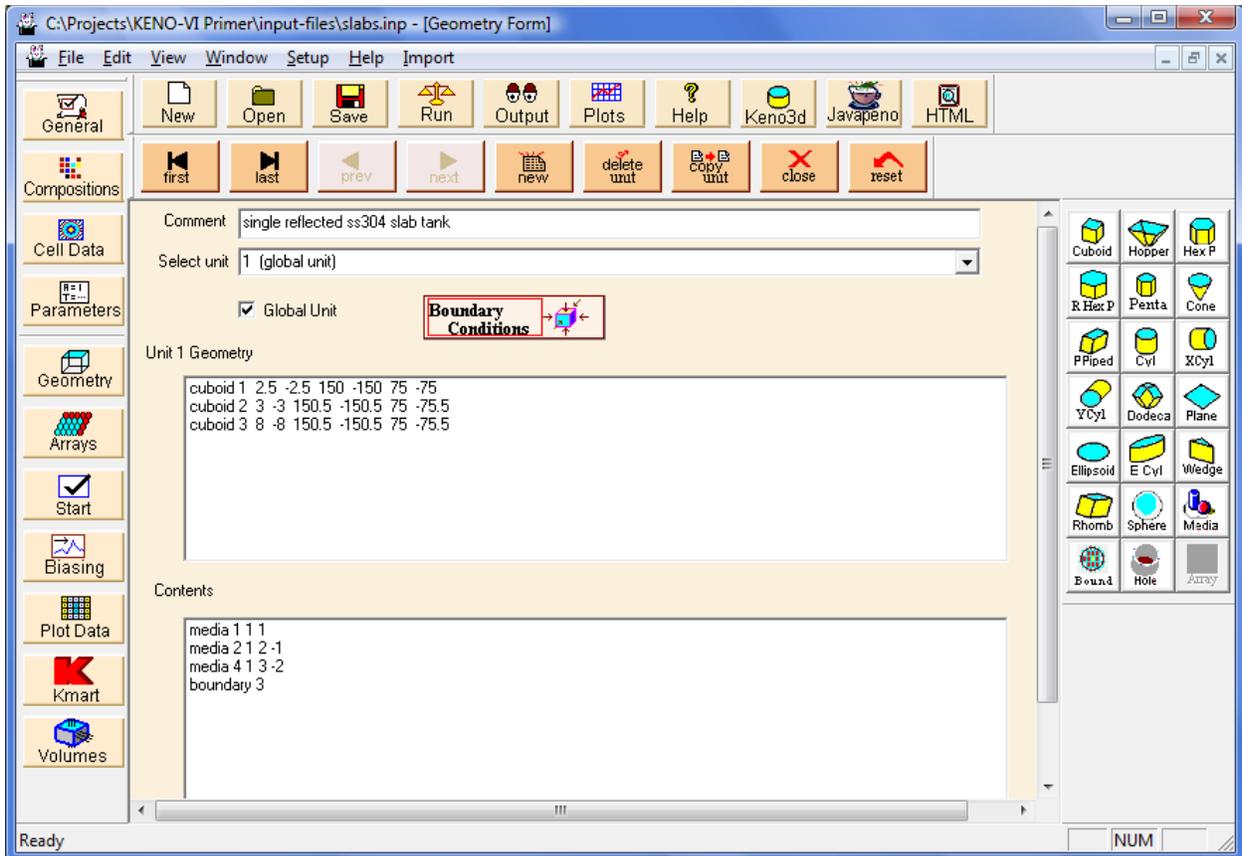
Fig. 152. Geometry information for slab tank.

When you check **Global Unit**, the **Boundary Conditions** button appears on the Geometry form. Click on the **Boundary Conditions** button to open the boundary conditions

form for the global unit. The default boundary conditions are `vacuum`. Because the number of outer faces varies with different KENO-VI geometry shapes (e.g., a cuboid has 6 faces, a hexprism has 8 faces, and a dodecahedron has 12 faces), the boundary conditions are input by face number in KENO-VI. Table F17.4.3 of the SCALE Manual, included here as Table 3, lists the face numbers and corresponding orientations for each KENO-VI shape. Generally, the face numbers are consistent with the input order of the shape dimensions in KENO-VI. For this case, we need to specify the boundary condition for the –**X** face as `mirror`. This setup will simulate an array of two tanks side by side. As you can see from the table, face 2 is the –X face for a cuboid. The completed form should look like that in Fig. 153.

Close all forms, save the changes in a new input file, and execute SCALE. The calculated k_{eff} value is 0.9598 ± 0.0024.

Table 3. Face codes for entering boundary (albedo) conditions

Geometry body	Albedo surface numbers related to KENO-VI geometry bodies[a]											
	1	2	3	4	5	6	7	8	9	10	11	12
Cone	Radial	+Z	–Z									
Cuboid	+X	–X	+Y	–Y	+Z	–Z						
Cylinder	Radial	+Z	–Z									
Dodecahedron	+X	–X	+Y	–Y	+X +Y +Z	–X –Y –Z	–X +Y +Z	+X –Y –Z	–X –Y +Z	+X +Y –Z	+X –Y +Z	–X +Y –Z
Ecylinder	Radial	+Z	–Z									
Ellipsoid	Radial											
Hexprism	+X	–X	+X +Y	–X –Y	–X +Y	+X –Y	+Z	–Z				
Hopper	+X	–X	+Y	–Y	+Z	–Z						
Pentagon	–Y	+X –Y	+X +Y	–X +Y	–X –Y	+Z	–Z					
Paralellepiped Ppiped	+X	–X	+Y	–Y	+Z	–Z						
Plane	Surface											

Table 3. Face codes for entering boundary (albedo) conditions (continued)

Geometry body	Albedo surface numbers related to KENO-VI geometry bodies[a]												
Quadratic	Surface												
Rhexprism	+Y	−Y	−X +Y	+X −Y	+X +Y	−X −Y	+Z	−Z					
Rhomboid	+X	−X	+Y	−Y	+Z	−Z							
Sphere	Radial												
Wedge	−Y	−X +Y	+X +Y	+Z	−Z								
Xcylinder	Radial	+X	−X										
Xpplane	+X	−X											
Ycylinder	Radial	+Y	−Y										
Ypplane	+Y	−Y											
Zcylinder	Radial	+Z	−Z										
Zpplane	+Z	−Z											

[a] Surfaces refer to the surface of the body that occurs in the indicated quadrant.
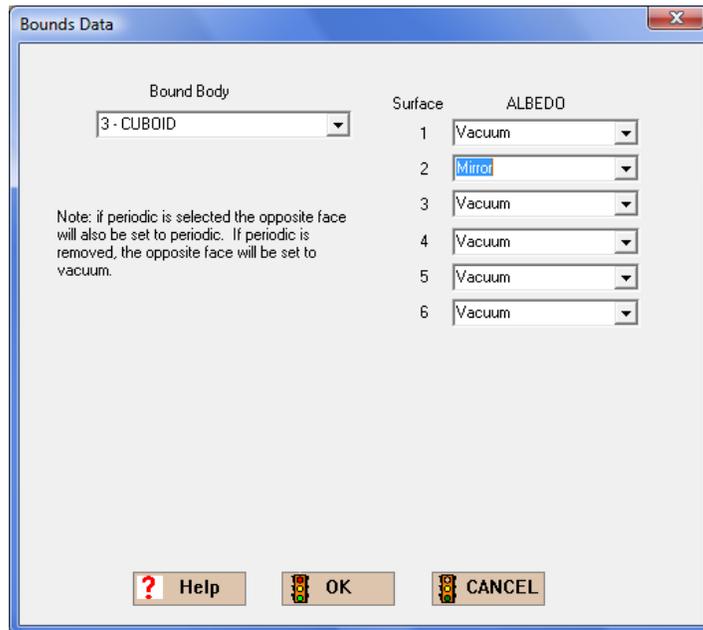
Fig. 153. Boundary conditions form.

8.6   SUMMARY

This section helped you to accomplish these objectives:

- Understand the difference in multigroup cross-section processing for Infinite Homogeneous Medium, Lattice Cell, and Multiregion Analyses.

- Identify the appropriate unit cell type to use for various heterogeneous systems.

- Use GeeWiz to define LATTICECELL and MULTIREGION unit cells.

- Define different boundary conditions for a MULTIREGION unit cell.

- Use boundary conditions to create reflective boundaries for the KENO-VI global unit.

9. PLOTTING AND VISUALIZATION

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

9.1   WHAT YOU WILL BE ABLE TO DO

- Generate 2-D color plots of KENO-VI geometry models.

- Interactively view 3-D wireframe and rendered images of KENO-VI geometry models using KENO3D.

- Plot calculated results overlaid on the 3-D geometry model using KMART6 and KENO3D.

9.2   CREATING AND VIEWING 2-D COLOR PLOTS

CSAS6 and KENO-VI have the built-in capability to produce plots of 2-D KENO-VI geometry slices. These plots can be generated without performing a calculation by using the PARM=CHECK option in the CSAS6/KENO-VI general input. This option provides the user with the capability to check the geometry model prior to performing calculations. The 2-D plots use the geometry routines in KENO-VI to produce precise model representations.

To demonstrate the 2-D plotting capability, use GeeWiz to open the 9×9 fuel assembly model that was generated in the previous section. Click on the **Plot Data** button on the left-hand toolbar to open the form in Fig. 154.

Fig. 154. Screen plotting data form.

Note that you can choose from **X−Y**, **X−Z**, and **Y−Z** plots, as well as plot by **M**ixture or **Unit Number**. Mixture plots are typically performed, because they show the geometry using a unique color for each material in the model. The coordinates of the **Upper Left Corner** and the **Lower Right Corner** of the plot must be specified by the user. The direction that does not vary in the plot (e.g., Z in an X−Y plot) must have the same coordinate values for both corners. GeeWiz automatically calculates the coordinate values for the corners based on the coordinates of the global unit. It will choose the minimum and maximum values of the two directions that vary and the midpoint of the third direction that does not vary (i.e., the location of the cutting plane).

In this case, by default GeeWiz has set the coordinates for an X−Y plot through the center of the model (Z=0). You will also need to enter a **Title** for the plot and one of the **Scaling Parameters** (**NAX or NDN**). You must define the size in pixels of the color PNG file in one dimension. KENO-VI automatically calculates the size of the other dimension for a plot that is proportionally scaled. Enter a value of 800 pixels for **NAX**. This value is typically a good size for most cases. Because resolution of the plot is proportional to the number of pixels, increasing the value of **NAX** or **NDN** will improve the resolution of the plot.

Note that you can select the color for each mixture in the plot. For a new plot, GeeWiz displays its default color palette. You can change any color by clicking on the **User Specified Colors** button and then clicking on the color you want to change. A color palette is then displayed. The completed form should look like Fig. 155. Click **OK** to close the form.

Fig. 155. Screen plot form for X-Y plot.

Click on the **General** button on the left-hand toolbar to open the general information form. Click on the **Parm=Check** box so that CSAS6 will perform only input checking and plotting without calling KENO-VI. Click **OK** to close the form. Then click on the **Run** button. When the job has finished, close the DOS window and click on the **Plots** button on the main toolbar. If a program for viewing plots has not been specified previously, the form in Fig. 156 will be displayed.
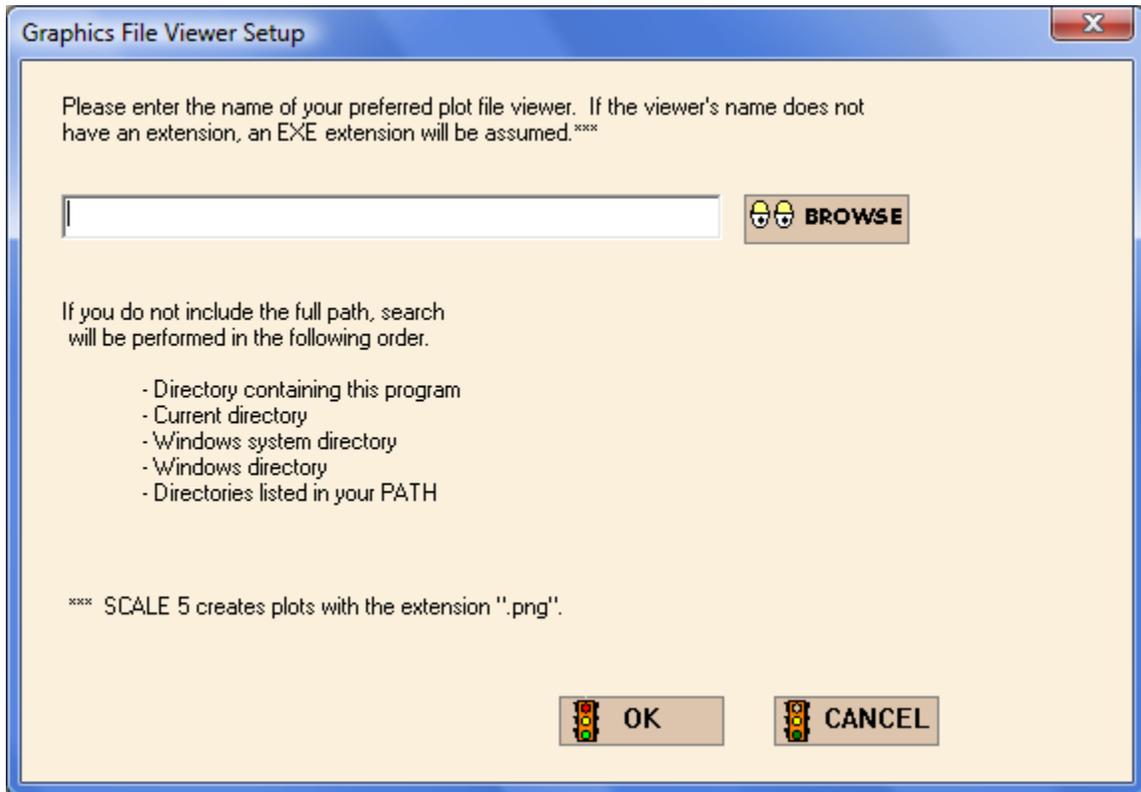
Fig. 156. Graphics file viewer setup form for 2-D plots.

Click on the **Browse** button to open an Explorer window to find the graphics viewing program that you want to use. The PNG files can be viewed via many different graphics viewers, such as IrfanView (http://www.irfanview.com), and via web browsers, such as Internet Explorer.

Once the viewing program has been identified, the plot will be displayed. It should look like Fig. 157.
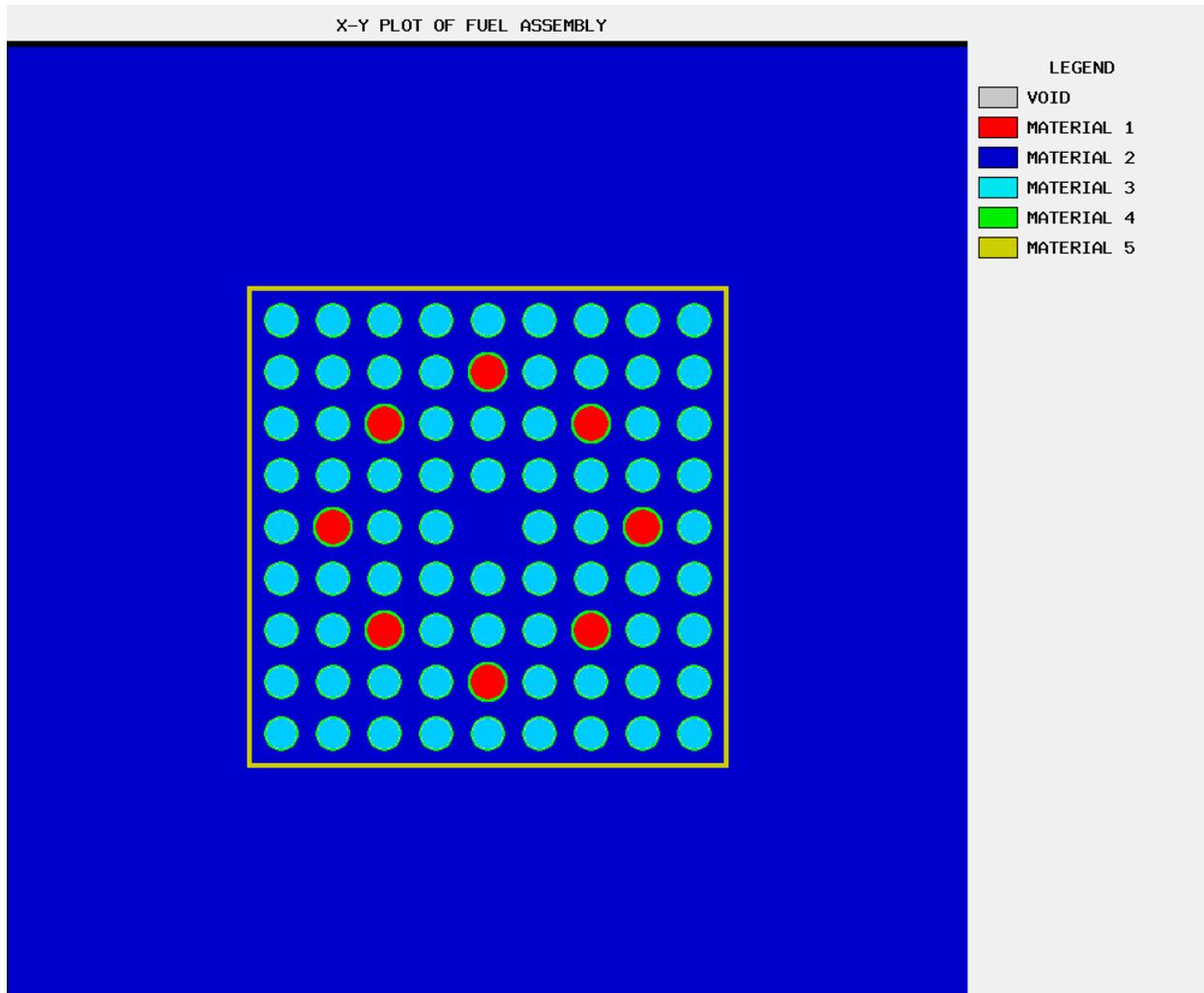
Fig. 157. X-Y plot of 9 × 9 fuel assembly.

9.3   INTERACTIVE 3-D VISUALIZATION WITH KENO3D

KENO3D enables KENO-VI and KENO-VI users to interactively display their 3-D geometry models. The interactive options include

- shaded or wireframe images;

- standard views, such as top view, side view, front view, and isometric (3-D) view;

- rotating the model;

- zooming in on selected locations;

- selecting parts of the model to display;

- editing colors and displaying legends;

- displaying properties of any unit in the model;

- creating cut-away views;

- removing units from the model; and

- copying images to clipboard or saving images to common graphics formats.

KENO3D reads CSAS5/KENO V.a or CSAS6/KENO-VI input files and can be opened directly from GeeWiz. It attempts to verify that the KENO geometry input is "legal" (i.e., that it conforms to the code input guidelines). KENO3D prints a warning message for illegal geometry input and, if possible, displays the illegal KENO geometry to facilitate debugging of the input. Note that KENO3D does not use the geometry routines in KENO-VI, so supplementing your use of KENO3D with 2-D plots as discussed in the previous section is recommended.

KENO3D has a fully integrated help system to aid users in learning how to use it. Only a brief demonstration of the use of 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 unique 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 large to provide a useful image when viewed in their entirety.

To display the 9×9 fuel assembly in KENO3D, click on the KENO3D button on the main toolbar in GeeWiz. KENO3D opens the file and displays the geometry as shown in Fig. 158.
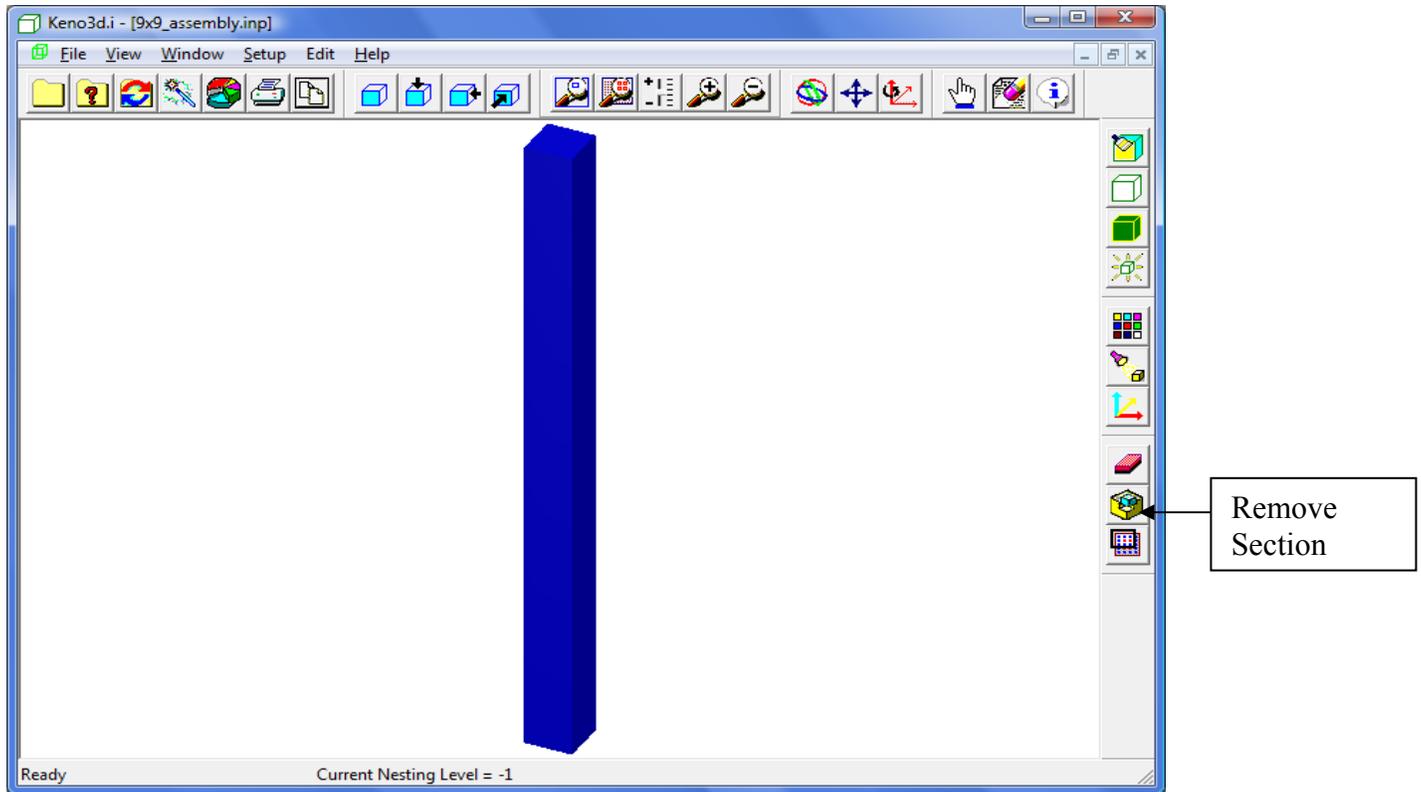
Fig. 158. Initial isometric view of 9×9 fuel assembly.

This isometric view shows the water reflector that surrounds the outer stainless steel shroud. Next click on the **Remove Section** button (see arrow in Fig. 158). This opens the **Remove Section** dialog form shown in Fig. 159. 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 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 Fig. 160.
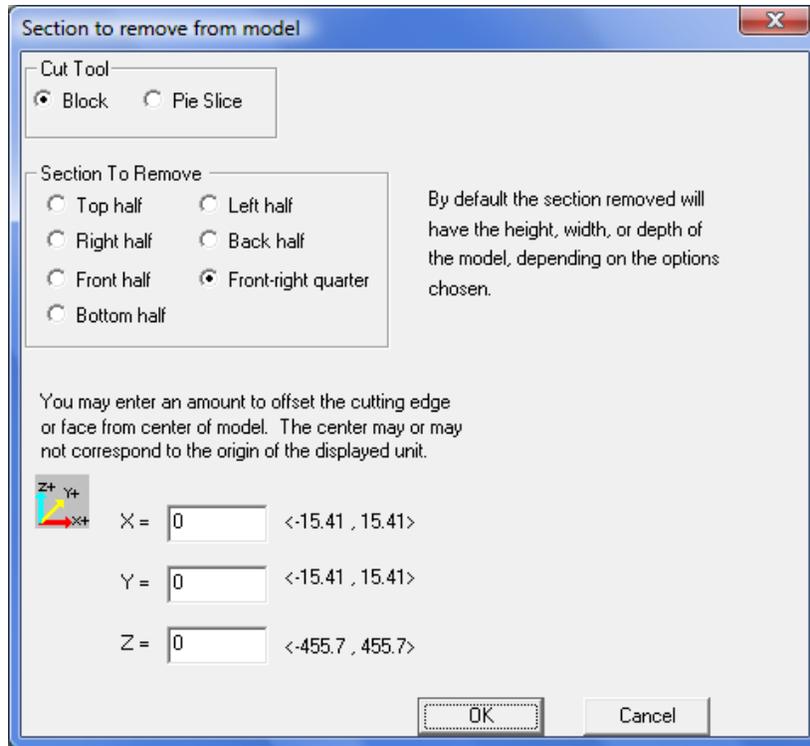
Fig. 159. Remove section dialog form.
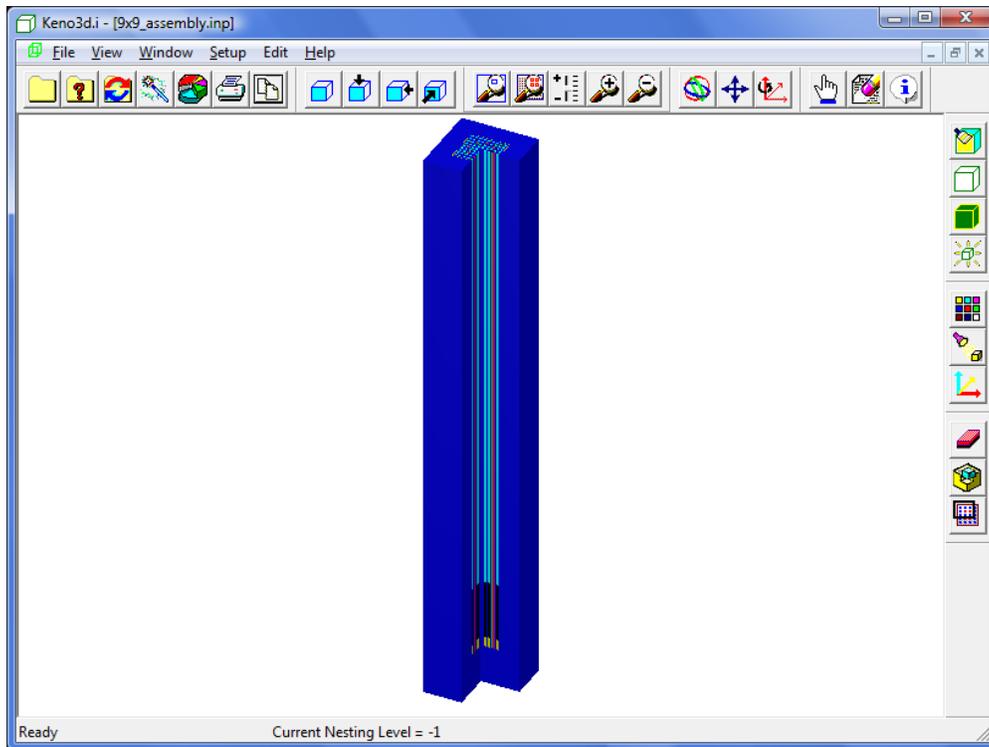
Fig. 160. Cutaway view of bottom half of assembly.

Next, hide the void and the water so that the stainless steel shroud and the fuel assembly are more easily visible. Click on the **Hide Mixture** (pencil eraser) button on the top toolbar, and select `0 void` and `2 h2o` under **Select mixture(s) to hide** and click **OK**. The image now looks like Fig. 161.

We want to zoom in on the bottom part of the fuel assembly so that we can the details more clearly. To do this, click on the **Zoom Window** button (red magnifying glass on the top toolbar). Then hold down the left mouse button and drag to make a window around the bottom portion of the assembly (Fig. 162). Once the window is satisfactory, release the left mouse button and then click it to zoom in on that part of the model.

Now we will hide the stainless steel shroud. Click on the **Display Properties** button (pointing finger on top toolbar). Then click on the shroud. It will be highlighted in red with an information window about it. Click on the **Hide Selected Body** button. The view now looks like Fig. 163. Click **Close** to close the Display Properties dialog box.
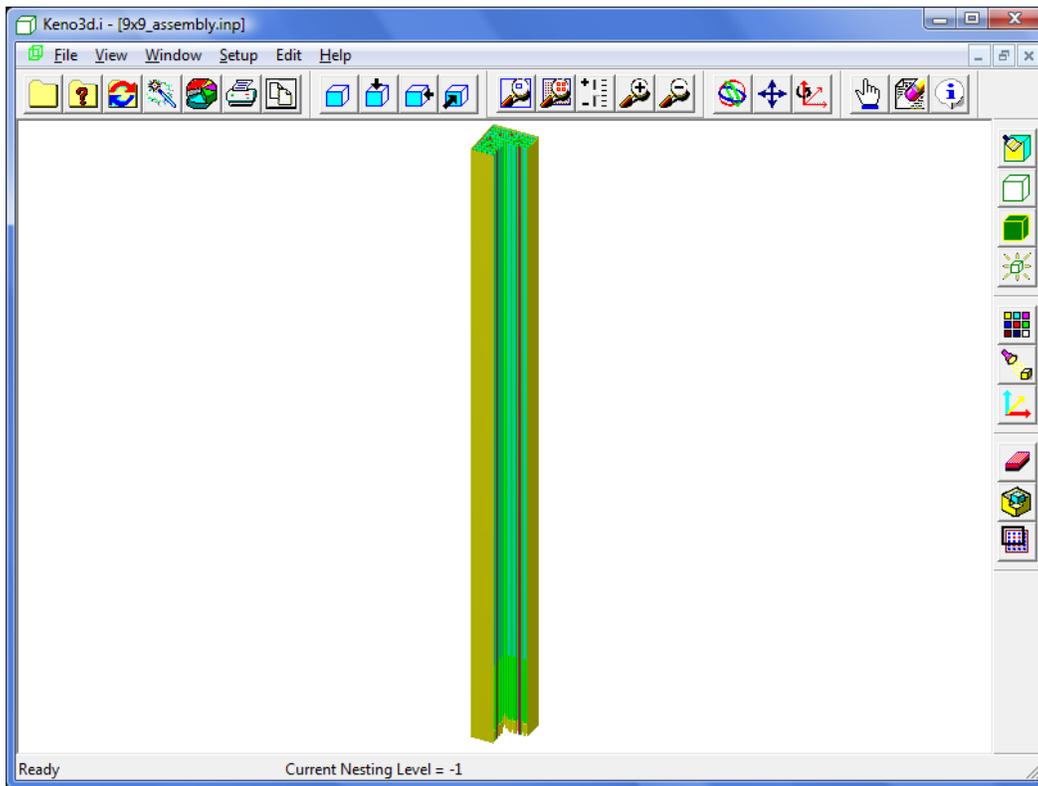
Fig. 161. Cutaway view with void and water hidden.
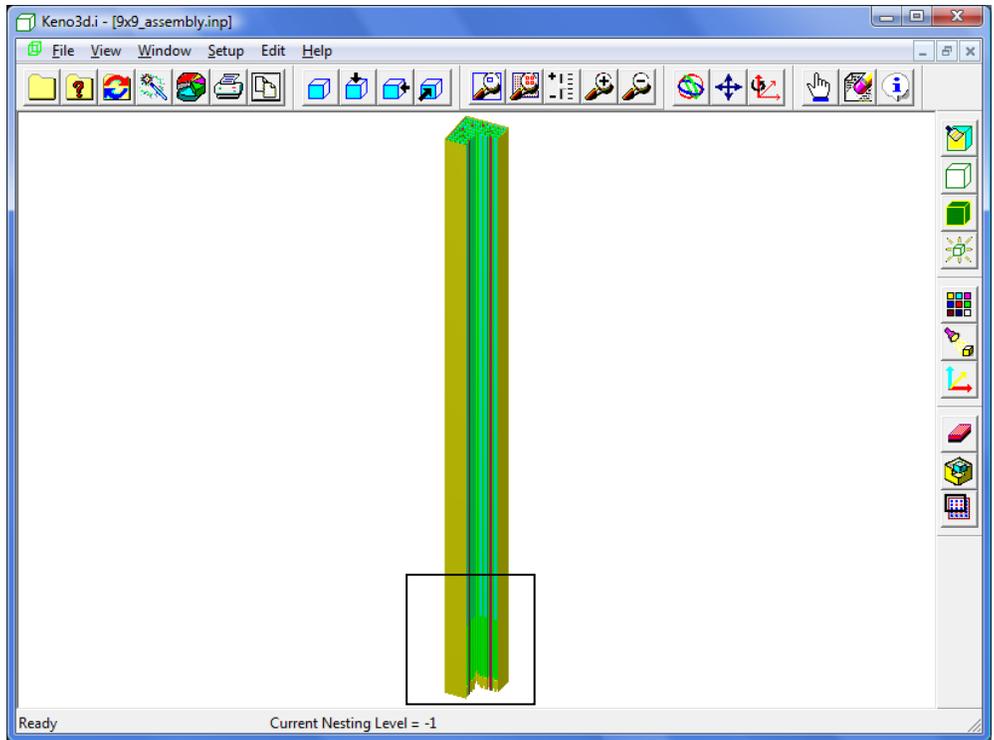
Fig. 162. Zoom window on bottom portion of fuel assembly in shroud.
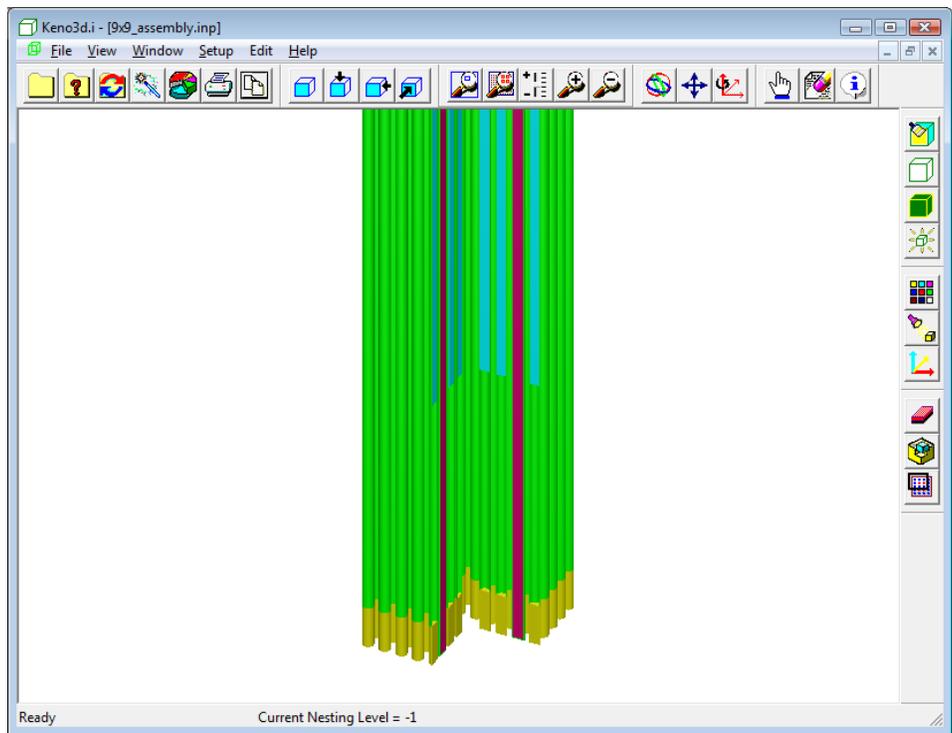
Fig. 163. Zoomed view of fuel assembly bottom with shroud hidden.

Now we can clearly see the structure of the fuel and burnable poison rods. It would be helpful to have a legend that labels the different components in the view. Click on the **Display Legend** button on the right toolbar, and click **OK** on the **Legend options** form to display the legend with the default settings. The view in Fig. 164 now appears.
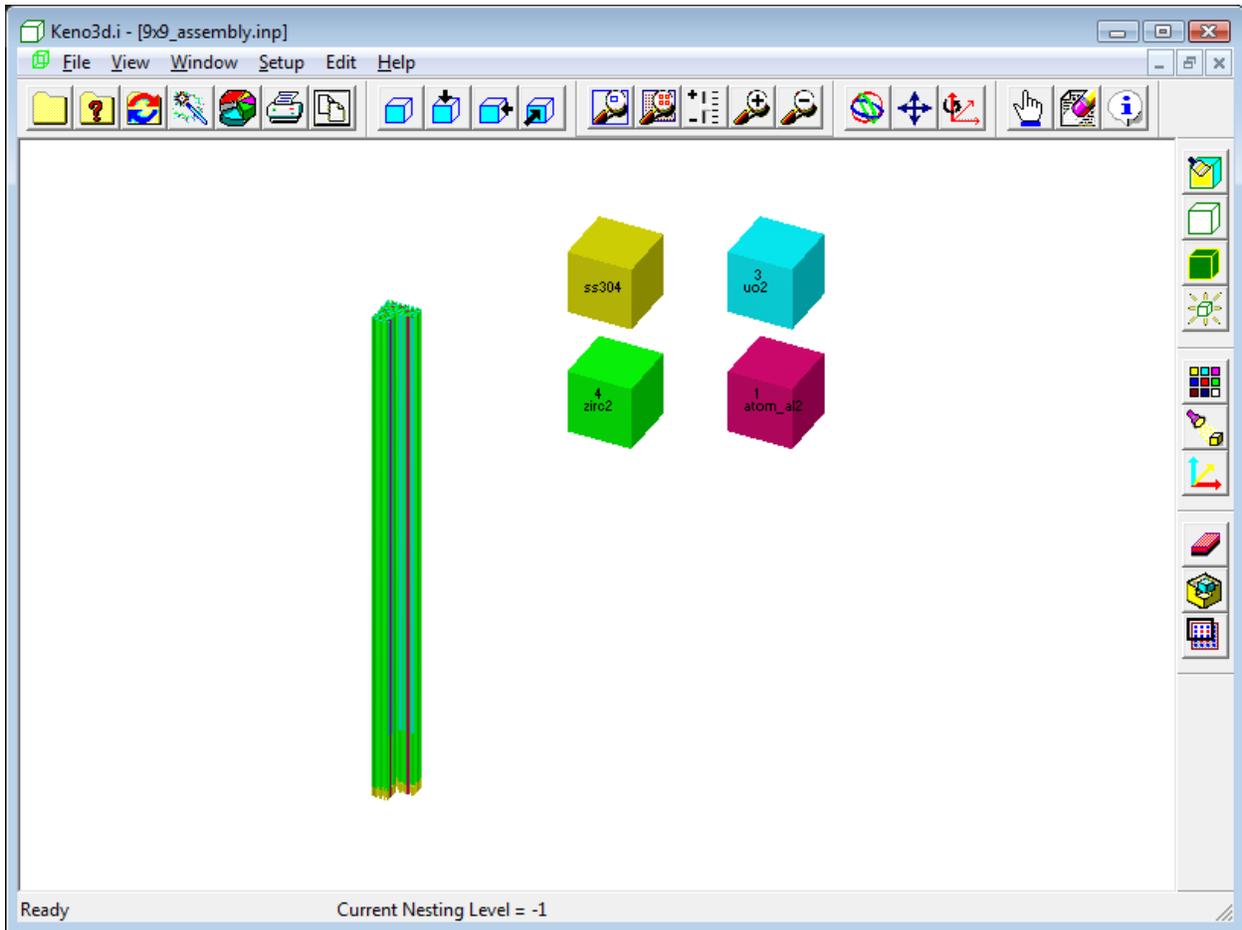
Fig. 164. View with default legend (white text).

Adding the legend has removed the focus on the zoom window and has returned the view of the bottom half of the fuel assembly. In addition, the legend is too large and not positioned near the bottom of the fuel assembly. We can correct this 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. (See Fig. 165.) Click **OK**; the screen then looks like Fig. 166.

Finally, to obtain the zoomed image of the fuel assembly bottom with the modified legend, click on the **Zoom window** button and draw the zoom window around the bottom of the fuel assembly and the legend. Click, and the final image in Fig. 167 appears.
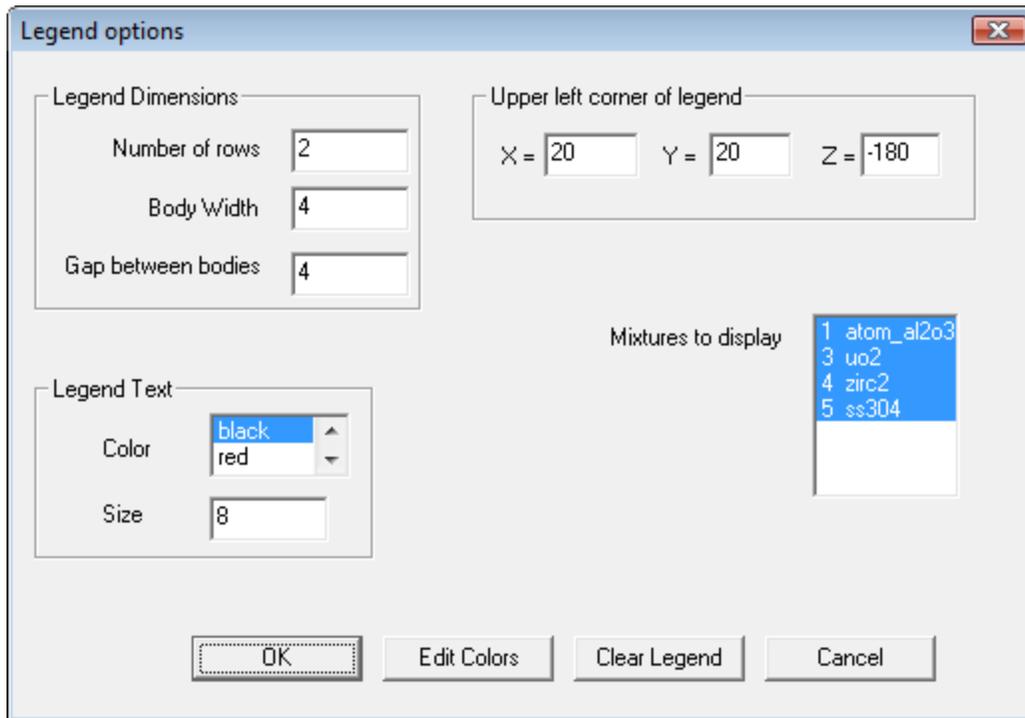
Fig. 165. Modified legend options.
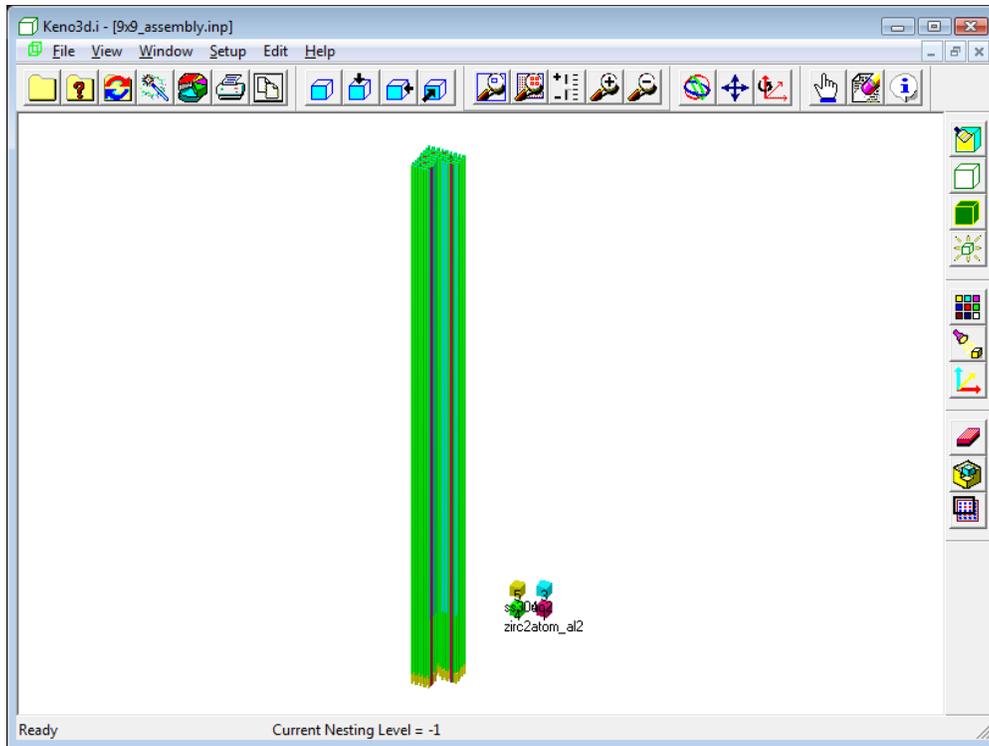
Fig. 166. Image with modified legend.
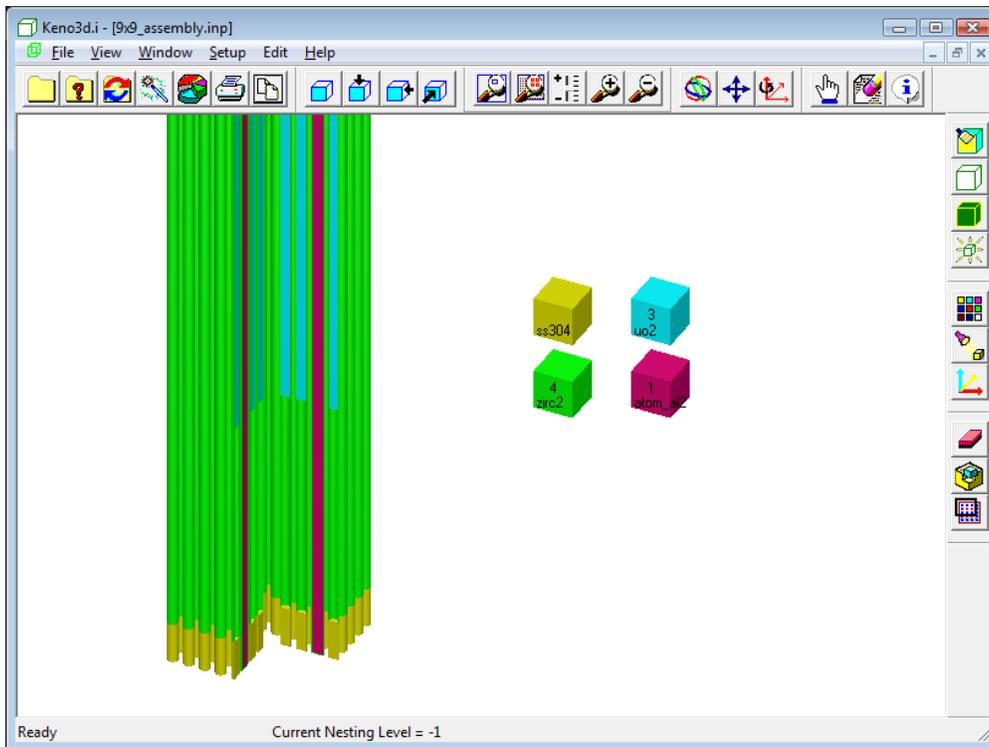
Fig. 167. Final image with legend.

9.4 PLOTTING CALCULATED RESULTS WITH KMART6 AND KENO3D

KMART6 (**K**ENO **M**odule for **A**ctivity-Reaction **R**ate **T**abulation Version **6**) is a tool that post-processes information from a KENO-VI run to provide reaction rates by nuclide and to collapse and print fluxes. Details on KMART6 are provided in (Sect. M17.9 of the SCALE Manual). KENO3D has the capability to plot the calculated results produced by KENO-VI/KMART6 overlaid on the 3-D geometry model.

There are a few items of which the KENO-VI user must be aware when utilizing KMART6. First, KMART6 uses a restart file for its evaluations. Thus the user must make sure that a restart unit number (**WRS=**) is identified, and the generations between restart (**RES=**) must be greater than zero in the KENO parameter input. The KENO parameters for flux and fission density edits (**FLX=** and **FDN=**) should be set to **YES**.

9.4.1 KENO-VI Volume Calculations

KMART6 needs volumes from KENO-VI to correctly calculate fluxes and reaction rates. By default, KENO-VI does not calculate volumes. The reason is that the calculation of volumes in KENO-VI is more difficult than in KENO V.a due to the generalized geometry in KENO-VI that allows body intersections. Volumes are not needed for the calculation of k_{eff}, but they are necessary for the calculation of other parameters such as fluxes and fission densities.

Note that in older versions of SCALE, KENO-VI would print fluxes and fission densities based on a volume of 1.0 for each region if volumes were not input or calculated. Because these values were nonphysical, the code was modified in SCALE 5.1 to use -1.0 so that users would not mistake these for valid values. The options for calculation of volumes in KENO-VI include two statistical methods: TRACE (ray tracing) or RANDOM (Monte Carlo sampling). In addition, the KENO3D visualization tool has been enhanced to calculate KENO-VI volumes analytically (i.e., exactly) and write those volumes directly to a KENO-VI input file. The KENO3D volume-calculation capability is available on the GeeWiz Volume Data form. The volume calculation in KENO3D has been optimized so that it takes only a few seconds in most cases, even for very complex models. This is the recommended method for most cases.

The following example demonstrates how to perform the volume calculation using GeeWiz and KENO3D. Use GeeWiz to open the KENO-VI input file annular_hex_array.inp in the \scale6\Keno3D\Examples folder. Click on the **Volumes** button in the left toolbar to open the Volumes form (Fig. 168). Under **Selected Option** click on `Interactive Options - calculate (KENO3D)`. The **KENO3D Volumes** button will appear on the top toolbar of the form. Click on this button. Click **Yes** on dialog to open KENO3D. You will be prompted to save the input file with a new name. When KENO3D finishes, click **OK** on dialog. Return to GeeWiz and click **Yes** on dialog to attach volume data to media records. Click **OK** to close the Volume Data form. If you click on the **Geometry** button, you will see the volume data on the media records.
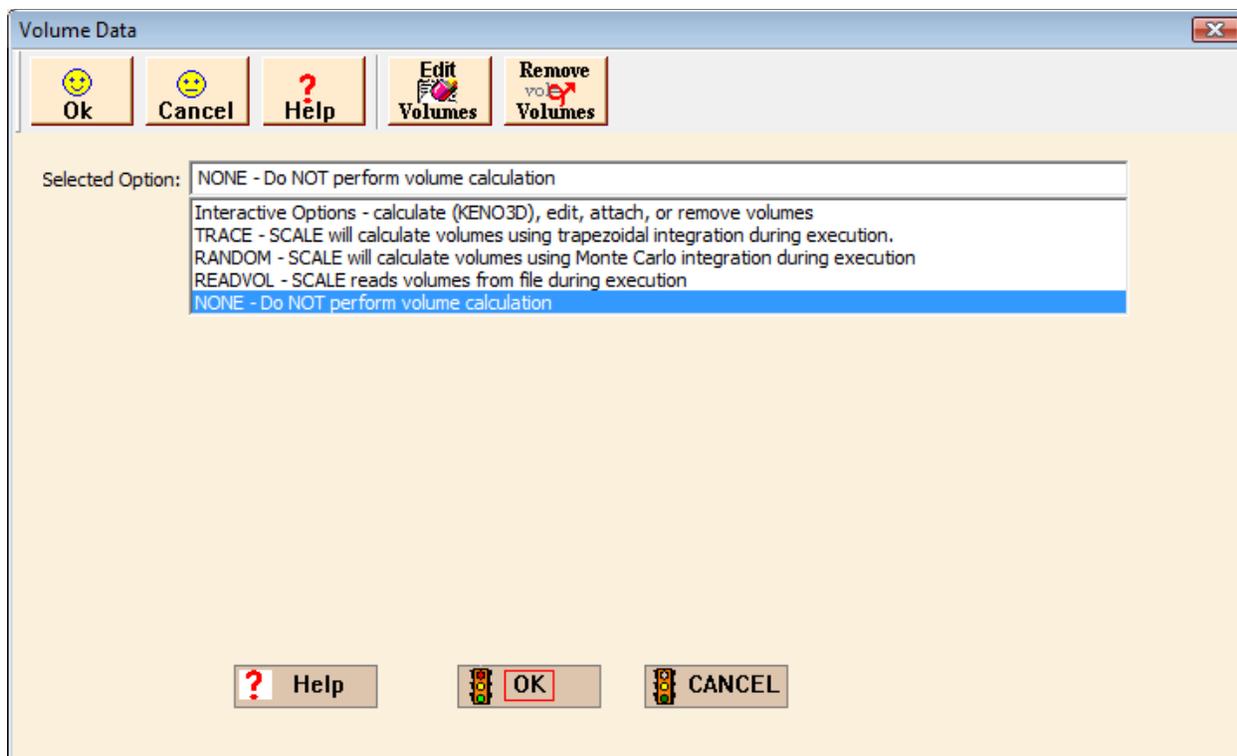
Fig. 168. GeeWiz Volume Data form.

9.4.2 Plotting KENO-VI Results in KENO3D

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. KENO-VI collects calculated data on a unit and region basis. Each unit in an array needs a unique unit number. This may require creating many duplicate units. The level of detail in the results is directly dependent upon the level of detail in the individual geometry regions. As an example, if the user desires to model the SHEBA solution experiment, a simple KENO-VI file might look like the one in Fig. 169. A view of the model in KENO3D is shown in Fig. 170.

```
'Input generated by GeeWiz SCALE 6.0 Compiled on February 11, 2008
=csas6
sheba 2
v7-238
read composition
 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 temperature=293 vol_frac=1
  end solution
 ss304s       2 den=7.8419 1 293    end
end composition
read parameter
 npg=2000
 res=100
 flx=yes
 fdn=yes
 wrs=35
 mkp=yes
 fmp=yes
 mku=yes
 fmu=yes
end parameter
read geometry
unit 1
com='section 0 - 5.0 of solution in the tank'
 cylinder 1    2.54     44.8         0
 cylinder 2    3.175    44.8         0
 cylinder 3  24.4475    44.8         0
 cylinder 4    25.4     44.8         0
 cuboid 5     25.4    -25.4    25.4    -25.4     44.8         0
 media 0 1 1
 media 2 1 2 -1
 media 1 1 3 -2 -1
 media 2 1 4 -3 -2 -1
 media 0 1 5 -4 -3 -2 -1
 boundary 5
unit 10
com='bottom of tank'
 cylinder 1    2.54   2.8575        0
 cylinder 2    25.4   2.8575        0
 cuboid 3     25.4   -25.4    25.4    -25.4   2.8575        0
 media 0 1 1
 media 2 1 2 -1
 media 0 1 3 -2 -1
 boundary 3
unit 11
com='void section of tank'
 cylinder 1    2.54  76.5175     44.8
 cylinder 2    3.175 76.5175     44.8
 cylinder 3  24.4475 76.5175     44.8
 cylinder 4    25.4  76.5175     44.8
 cuboid 5     25.4    -25.4    25.4    -25.4  76.5175     44.8
```

Fig. 169. Simple SHEBA-II input file.

```
 media 0 1 1
 media 2 1 2 -1
 media 0 1 3 -2 -1
 media 2 1 4 -3 -2 -1
 media 0 1 5 -4 -3 -2 -1
 boundary 5
unit 12
com='top of tank'
 cylinder 1     2.54     1.905          0
 cylinder 2     25.4     1.905          0
 cuboid 3     25.4    -25.4     25.4    -25.4     1.905          0
 media 0 1 1
 media 2 1 2 -1
 media 0 1 3 -2 -1
 boundary 3
global unit 13
 cuboid 1     50.8         0     50.8         0     81.28         0
 array 1 1  place 1 1 1 25.4 25.4 0
 boundary 1
end geometry
read array
ara=1 nux=1 nuy=1 nuz=4 typ=square gbl=1
 fill
   10
    1
   11
   12
 end fill
end array
end data
end
```

Fig. 169. Simple SHEBA-II input file. (continued)
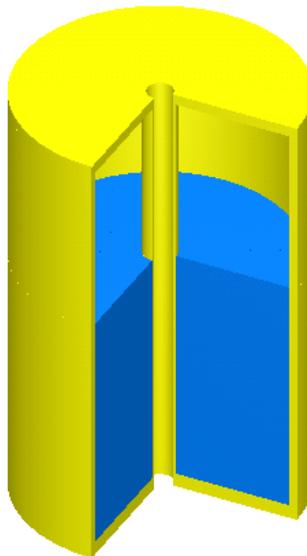
Fig. 170. Simple model of SHEBA-II.

189

However, if you want to look at reaction rates, fluxes, or neutron production rates, then you need to divide the solution into finer segments. So for this example, you might want to create sections that are 5 cm in height and 3 cm in radius. The input file with the associated KMART6 input can be found in the \scale6\Keno3d\Examples folder in SCALE 6. Open the input file sheba-detail-k6.inp in GeeWiz. Click on the **KMart** button to view the **KMART parameters** input data (Fig. 171). The **Activity by group** and **Reaction rates per unit volume** flags are checked. The compositions specified include ^{1}H (1001), ^{16}O (8016), ^{235}U (92235), and ^{238}U (92238). The reaction types include absorption (27), fission (18), and neutron production (nu * fission = 1452). Click on the **Insert** button to view the **KMart Activity** form (Fig. 172). On this form you can select any nuclide in the model and the associated reaction type that you want. The definition for each reaction type is shown in the list. These reaction type numbers are also documented in (Sect. F3.B of the SCALE Manual). Click **Cancel** to remove each form from the screen.

Fig. 171. KMART parameters input data.

Fig. 172. KMART activity form.

We will use KENO3D to plot the neutron production. Click on the **KENO3D** button in GeeWiz to display the model. Click on the **Remove Section** button, and click **OK** to remove the right front quarter. The view should look like Fig. 170. Click on the **Highlight Edges** button on the right toolbar in KENO3D to view the additional regions that have been created to plot KMART data. The output and KMART files are included in the KENO3D\Examples folder, so we can plot the KMART data. Click **View** and then **Plot KMart data** to open the **Plot Results (KMART)** form (Fig. 173).
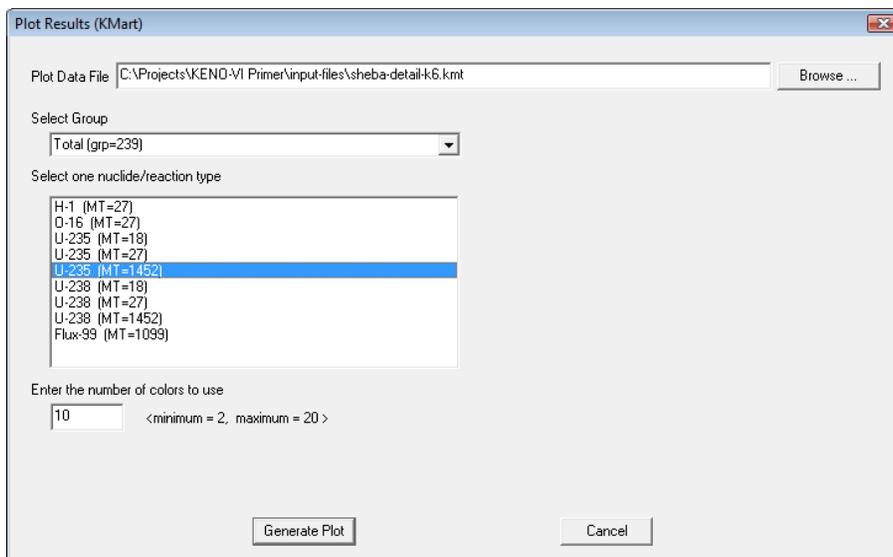
Fig. 173. Plot results (KMART) form.

Under **Select Group**, click on `Total (grp=239)` to select the total reaction rate over all energy groups. Click on `U-235 (MT=1452)` to select the ^{235}U neutron production. Click on **Generate Plot**. Then click on the **Display Legend** button. The neutron production plot (Fig. 174) appears on the screen.

Note that the default KENO3D view of the model appears the same as the simple model in Fig. 170, but dividing it into smaller segments allows KMART to produce neutron production plots like Fig. 174. Dividing the model into even smaller segments can result in a much higher resolution plot, as shown in Fig. 175.

U-235 (MT=1452,Total)

	2.6521e-005
	2.4633e-005
	2.2170e-005
	1.9706e-005
	1.7243e-005
	1.4780e-005
	1.2316e-005
	9.8532e-006
	7.3899e-006
	4.9266e-006
	2.4633e-006
	0.0000e+000
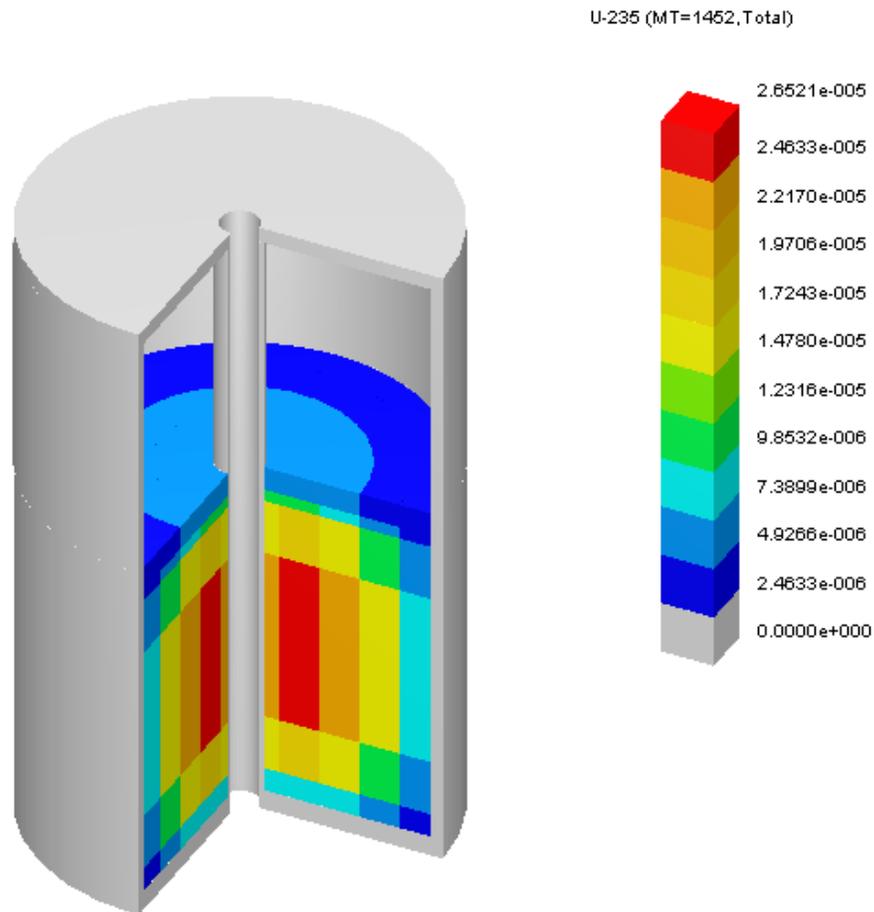
Fig. 174. SHEBA-II ^{235}U neutron production.

U-235 (MT=1452,Total)

	0.0000e+000
	4.8669e-007
	3.2377e-006
	5.9887e-006
	8.7397e-006
	1.1491e-005
	1.4242e-005
	1.6993e-005
	1.9744e-005
	2.2495e-005
	2.5246e-005
	2.7997e-005
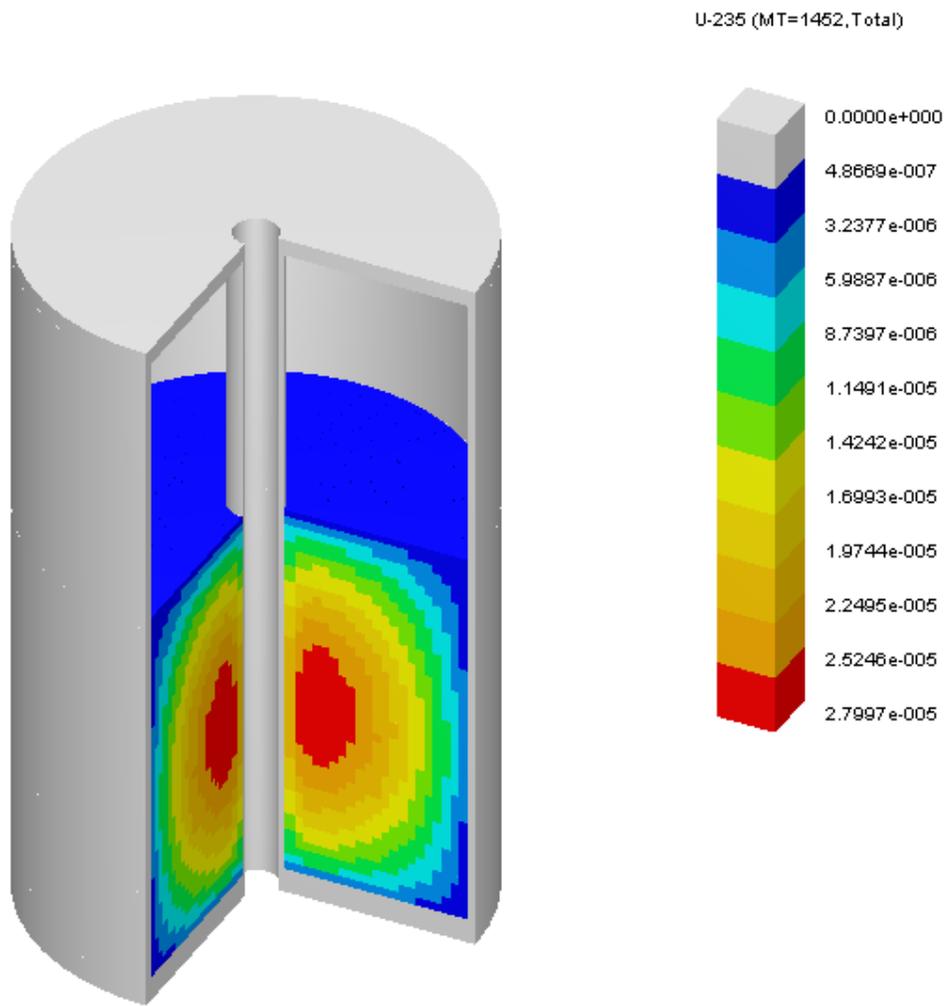
Fig. 175. Higher resolution plot of SHEBA-II neutron production.

9.5 SUMMARY

This section has helped you to accomplish the following:

- Generate 2-D color plots of KENO-VI geometry models.

- Interactively view 3-D rendered images of KENO-VI geometry models using KENO3D.

- Plot calculated results overlaid on the 3-D geometry model using KMART6 and KENO3D.

10. CONCLUSION

Congratulations! If you have completed all previous sections, you now should be able to perform the following objectives:

- Describe the structure of SCALE/KENO-VI input files, such as the sequence, the materials, and the KENO information.

- Use the GeeWiz user interface to create and run SCALE/KENO-VI input files to model criticality problems and view results.

- Find and interpret k_{eff} information from your output and perform simple checks for reasonableness on your output.

Additional information and help can be obtained at the SCALE website www.ornl.gov/sci/scale. There you can also find

- current and past issues of the *SCALE Newsletter,*

- SCALE training course information, and

- SCALE Notebook with answers to many user questions.

The 3-D illustrations in this primer were generated with the KENO3D interactive visualization tool. For more information about KENO3D, including how to order, visit the KENO3D website at www.ornl.gov/sci/scale/keno3d.

User assistance is also available by e-mail at scalehelp@ornl.gov.

11. REFERENCES

1. *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluations*, ORNL/TM-2005/39, Version 5.1, Vols. I–III, November 2006. Available from Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-732.

2. H. C. Paxton and N. L. Pruvost, *Critical Dimensions of Systems Containing U-235, Pu-239, and U-233*, 1986 Revision, LA-10860-MS, July 1987, Los Alamos National Laboratory.

APPENDIX A

INPUT FILES FOR EXAMPLE PROBLEMS

APPENDIX A
INPUT FILES FOR EXAMPLE PROBLEMS

SECTION 2 – SCALE/KENO-VI *QUICKSTART*

2.3.1.   Jezebel (jezebel.inp)

```
=csas26 parm=(centrm)
jezebel
v7-238
read composition
 pu-239       1 0 0.037047 293    end
 pu-240       1 0 0.001751 293    end
 pu-241       1 0 0.000117 293    end
 ga           1 0 0.001375 293    end
 ni           2 0 0.091322 293    end
end composition
read celldata
  multiregion spherical left_bdy=reflected right_bdy=vacuum end
          1       6.38493
          2       6.39763
      end zone
end celldata
read parm htm=yes end parm
read geometry
global unit 1
com='jezebel core with nickel plating'
 sphere 1  6.38493
 sphere 2  6.39763
 media 1 1 1
 media 2 1 2 -1
 boundary 2
end geometry
end data
end
```

SECTION 3 – MATERIAL INFORMATION INPUT

3.5.1. Bare ^{235}U Metal Cylinder (bare–u235–cyl.inp)

```
=csas26 parm=(centrm)
Bare u-235 cylinder
v7-238
read composition
 u-235        1 den=18.742 1 300    end
end composition
read geometry
global unit 1
com='bare u-235 cylinder'
 cylinder 1     7.82     7.82    -7.82
 media 1 1 1
 boundary 1
end geometry
end data
end
```

3.5.2. Bare U(93.7) Metal Cylinder (bare–u93–cyl.inp)

```
=csas26 parm=(centrm)
bare U(93.71) cylinder
v7-238
read composition
 u            1 den=18.742 1 300
                                92235 93.71
                                92238 6.29   end
end composition
read geometry
global unit 1
com='Bare U(93.71) cylinder'
 cylinder 1     7.82     7.82    -7.82
 media 1 1 1
 boundary 1
end geometry
end data
end
```

3.5.3.   Bare U(93.7)O$_2$ Cylinder (bare–uo2–u93–cyl.inp)

```
=csas26 parm=(centrm)
bare uo2(93.71) cylinder
v7-238
read composition
 uo2        1 0.95 300
                                   92235 93.71
                                   92238 6.29   end
end composition
read geometry
global unit 1
com='bare uo2(93.71) cylinder'
 cylinder 1     12.8     12.8    -12.8
 media 1 1 1
 boundary 1
end geometry
end data
end
```

3.5.4.   Bare U(30.3)O$_2$F$_2$ Solution Cylinder (bare–uo2f2–cyl.inp)

```
=csas26 parm=(centrm)
Bare uo2f2 cylinder
v7-238
read composition
 u-235      1 0 0.0005637 300   end
 u-238      1 0 0.0012802 300   end
 f          1 0 0.0036844 300   end
 o          1 0 0.0335605 300   end
 h          1 0 0.0597522 300   end
end composition
read geometry
global unit 1
com='uo2f2 cylinder'
 cylinder 1   15.225   15.225  -15.225
 media 1 1 1
 boundary 1
end geometry
end data
end
```

SECTION 4 – GEOMETRY INPUT

4.2.2. Z-Cylinder (bare–Zcyl.inp)

```
=csas26 parm=(centrm)
bare u-235 cylinder
v7-238
read composition
 u-235        1 den=18.742 1 300    end
end composition
read geometry
global unit 1
com='bare u-235 cylinder'
 cylinder 1     7.82      7.82     -7.82
 media 1 1 1
 boundary 1
end geometry
end data
end
```

4.2.3. Z-Cylinder Offset 2 cm from Origin (bare–Zcyl2.inp)

```
=csas26 parm=(centrm)
bare u-235 cylinder with bottom at Z=-2
v7-238
read composition
 u-235        1 den=18.742 1 300    end
end composition
read geometry
global unit 1
com='bare u-235 cylinder with bottom at Z=-2'
 cylinder 1     7.82    13.64        -2
 media 1 1 1
 boundary 1
end geometry
end data
end
```

4.2.4. X-Cylinder (bare–Xcyl.inp)

```
=csas26 parm=(centrm)
bare u-235 cylinder on X axis
v7-238
read composition
 u-235        1 den=18.742 1 300    end
end composition
read geometry
global unit 1
com='bare u-235 cylinder on X axis'
 xcylinder 1     7.82    15.64        0
 media 1 1 1
 boundary 1
end geometry
end data
end
```

4.5.1. Reflected Pu Metal Cylinder (reflected–pu–cyl.inp)

```
=csas26 parm=(centrm)
pu cylinder reflected with graphite
v7-238
read composition
 pu           1 den=15.44 1 300
                                  94239 95
                                  94240 5   end
 c-graphite  2 den=1.6 1 300    end
end composition
read geometry
global unit 1
com='reflected pu cylinder'
 cylinder 1     2.86    40.96        0
 cylinder 2    20.64    58.74   -17.78
 media 1 1 1
 media 2 1 2 -1
 boundary 2
end geometry
end data
end
```

4.5.2. ^{235}U Sphere with Graphite and Water Reflectors (u-235–sphere–reflected.inp)

```
=csas26
u-235 sphere with graphite and water reflectors
v7-238
read composition
 u-235       1 den=18.74 1 300   end
 c-graphite  2 den=1.65 1 300    end
 h2o         3 1 300    end
end composition
read geometry
global unit 1
com='u-235 sphere inside graphite cylinder inside water reflector'
 sphere 1        7
 cylinder 2      10        10        -10
 cuboid 3        12       -10        12       -10        12       -10
 media 1 1 1
 media 2 1 2 -1
 media 3 1 3 -2
 boundary 3
end geometry
end data
end
```

4.6.1. $2 \times 2 \times 2$ Array of U(93.2) Metal Cylinders (single–unit–array.inp)

```
=csas26 parm=(centrm)
critical experiment 2c8 2x2x2 array
v7-238
read composition
 uranium     1 den=18.76 1 293
                            92234 1
                            92235 93.2
                            92236 0.2
                            92238 5.6    end
end composition
read geometry
unit 1
com='single unit centered'
 cylinder 1    5.748    5.3825   -5.3825
 cuboid 2      6.87     -6.87     6.87     -6.87     6.505    -6.505
 media 1 1 1
 media 0 1 2 -1
 boundary 2
global unit 2
com='2x2x2 array'
 cuboid 1    13.74    -13.74    13.74    -13.74    13.01    -13.01
 array 1 1   place 1 1 1 -6.87 -6.87 -6.505
```

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

4.6.2.   Stack of Six Disks and Flat Plate (stack.inp)

```
=csas26 parm=(centrm)
stacked array of disks and plate
v7-238
read composition
 c-graphite  1 den=1.65 1 300    end
 uranium     2 den=18.74 1 300
                                   92234 1
                                   92235 93.2
                                   92236 0.2
                                   92238 5.6   end
 al          3 1 300    end
end composition
read geometry
unit 1
com='graphite disk'
 cylinder 1       10      2.5         0
 cuboid 2       15     -15      15     -15      2.5        0
 media 1 1 1
 media 0 1 2 -1
 boundary 2
unit 2
com='uranium disk'
 cylinder 1       10       4         0
 cuboid 2       15     -15      15     -15       4         0
 media 2 1 1
 media 0 1 2 -1
 boundary 2
unit 3
com='aluminum square'
 cuboid 1       15     -15      15     -15       3         0
 media 3 1 1
 boundary 1
global unit 4
com='stack'
 cuboid 1       15     -15      15     -15     22.5        0
```

```
 array 1 1  place 1 1 1 0 0 0
 boundary 1
end geometry
read array
ara=1 nux=1 nuy=1 nuz=7 typ=square gbl=1
 com=''
 fill
     1
     2
     1
     2
     1
     2
     3
 end fill
end array
end data
end
```

SECTION 5 – ADVANCED GEOMETRY INPUT

5.3.1.   Three Intersecting Cylinders (3cylinders.inp)

```
=csas26
3 intersecting cylinders
v7-238
read composition
 pu-239        1 den=18.5 1 300    end
 h2o           2 1 300    end
 ss304         3 1 300    end
end composition
read geometry
global unit 1
com='three cylinders'
 cylinder 1        10        10      -10    origin  x=0 y=4.5 z=0
 cylinder 2        10        10      -10    origin  x=-4.5 y=-4.5 z=0
 cylinder 3        10        10      -10    origin  x=4.5 y=-4.5 z=0
 cuboid 4       15      -15      15     -15       15       -15
 media 1 1 1 2 3
 media 2 1 1 2 -3
 media 2 1 2 3 -1
 media 2 1 1 3 -2
 media 3 1 1 -2 -3
 media 3 1 2 -1 -3
 media 3 1 3 -1 -2
 media 0 1 4 -1 -2 -3
 boundary 4
end geometry
end data
end
```

5.4.4   29 Degree Y Angle Joint (y29.inp)

```
=csas26 parm=(centrm)
29 degree y
v7-238
read composition
 solution   mix=1   rho[uo2f2]=907 92235 5
                                   92238 95
              molar[hfacid]=0
              density=2.0289 temperature=300 vol_frac=1
  end solution
 al          2 1 300    end
 h2o         3 1 300    end
end composition
read geometry
global unit 1
```

```
com='29 degree y solution experiment'
 cylinder 1    13.95    135.4    -75.4
 cylinder 2    14.11    135.4    -76.7
 cylinder 3    13.95    125.4       0    rotate   a1=0 a2=-29.26 a3=0
 cylinder 4    14.11   126.04       0    rotate   a1=0 a2=-29.26 a3=0
 plane 5  zpl=-1  con=52.8
 cuboid 6        37      -37      100      -37    135.4     -99.6
 media 1 1 1 5
 media 2 1 2 -1 -3
 media 1 1 3 5 -1
 media 2 1 4 -3 -2
 media 0 1 1 -5
 media 0 1 3 -1 -5
 media 3 1 6 -2 -4
 boundary 6
end geometry
end data
end
```

5.5.1. Rotated Holes (rotated–holes.inp)

```
=csas26 parm=(centrm)
rotated holes example
v7-238
read composition
 pu          1 1 300
                              94239 1
                              94240 99   end
 ss304       2 1 300   end
 reg-concrete 3 1 300   end
end composition
read geometry
unit 1
com='pu in ss304 box'
 cuboid 1       5       -5       5       -3       10      -10
 cuboid 2       6       -6       6       -6       11      -11
 media 1 1 1
 media 2 1 2 -1
 boundary 2
global unit 2
com='concrete cylinder'
 cylinder 1      30       12      -12
 hole 1
 hole 1   origin  x=12 y=12 z=0    rotate   a1=-45 a2=0 a3=0
 hole 1   origin  x=12 y=-12 z=0   rotate   a1=-135 a2=0 a3=0
 hole 1   origin  x=-12 y=-12 z=0   rotate   a1=135 a2=0 a3=0
 hole 1   origin  x=-12 y=12 z=0   rotate   a1=45 a2=0 a3=0
 media 3 1 1
 boundary 1
end geometry
end data
end
```

SECTION 6 – ARRAYS

6.2.   Five Cylinders with Cruciforms (5_cyls_cruciform.inp)

```
=csas26
5 cruciform cylinders in shipping container
v7-238
read composition
 uc          1 1 300
                              92235 93.2
                              92238 6.8   end
 h2o         2 1 300   end
 ss316       3 1 300   end
end composition
read geometry
unit 1
com='cruciform cylinder'
 cuboid 1        5        -5       0.5      -0.5       10       -10
 cuboid 2      0.5      -0.5         5        -5       10       -10
 cylinder 3       6        10       -10
 cylinder 4     6.1      10.1     -10.1
 cuboid 5      6.2      -6.2       6.2      -6.2     10.1     -10.1
 media 1 1 1
 media 1 1 2 -1
 media 2 1 3 -1 -2
 media 3 1 4 -3
 media 0 1 5 -4
 boundary 5
unit 2
com='empty array location'
 cuboid 1      6.2      -6.2       6.2      -6.2     10.1     -10.1
 media 0 1 1
 boundary 1
global unit 3
com='large container'
 cylinder 1       20      10.1     -10.1
 cylinder 2     20.2      10.3     -10.3
 array 1 1  place 3 3 1 0 0 0
 media 3 1 2 -1
 boundary 2
end geometry
read array
ara=1 nux=5 nuy=5 nuz=1 typ=square gbl=1
 com=''
 fill
    2    2    2    2    2
    2    2    1    2    2
    2    1    1    1    2
    2    2    1    2    2
    2    2    2    2    2
 end fill
end array
end data
end
```

6.3. Hexagonal Array (hex-array1.inp)

```
=csas26 parm=(centrm)
hexagonal array
v7-238
read composition
 uranium     1 1 300
                                    92235 10
                                    92238 90    end
 al          2 1 300    end
 h2o         3 1 300    end
 carbonsteel 4 1 300    end
end composition
read geometry
unit 1
com='unit 1'
 cylinder 1      7.5      18.3         0
 cylinder 2        8      18.3      -0.5
 hexprism 3     10.5      18.3      -0.5
 media 1 1 1
 media 2 1 2 -1
 media 3 1 3 -2
 boundary 3
unit 2
com='unit 2'
 hexprism 1     10.5      18.3      -0.5
 media 3 1 1
 boundary 1
global unit 3
com='global unit 3 references array 1'
 cylinder 1       12      18.3      -0.5
 cylinder 2     31.5      18.3      -0.5
 cylinder 3       33      18.3        -2
 array 1 2 -1  place 3 3 1 0 0 0
 media 4 1 1
 media 4 1 3 -2
 boundary 3
end geometry
read array
ara=1 nux=5 nuy=5 nuz=1 typ=shexagonal
 com=''
 fill
    2    2    2    2    2
    2    1    1    2    2
    2    1    2    1    2
    2    1    1    2    2
    2    2    2    2    2
 end fill
end array
end data
end
```

SECTION 7 – NONSTANDARD MATERIALS

No Example Problems.

SECTION 8 – UNIT CELLS AND BOUNDARY CONDITIONS

8.5.1. 9 × 9 Fuel Assembly (9x9_assembly.inp)

```
=csas26 parm=(centrm)
9x9 fuel assembly
v7-238
read composition
 atom_al2o3      1   3.7   2
                                      13027 2
                                      8016 3
                        0.9874 300    end
 b4c           1 den=3.7 0.0126 300   end
 h2o           2 1 300    end
 uo2           3 0.925 300
                                      92235 3.44
                                      92238 96.56    end
 zirc2         4 1 300     end
 ss304         5 1 300     end
end composition
read celldata
   latticecell squarepitch fuelr=0.47 3 gapr=0.4875 0 cladr=0.545 4
hpitch=0.82 2 end
end celldata
read parameter
 gen=505
 nsk=5
end parameter
read geometry
unit 1
com='3.44% enriched fuel rod'
 cylinder 1      0.47    182.88   -182.88
 cylinder 2    0.4875    199.84   -206.24
 cylinder 3     0.545    199.84   -206.24
 cylinder 4     0.545    210.34   -210.34
 cuboid 5       0.82     -0.82     0.82    -0.82    210.34   -210.34
 media 3 1 1
 media 0 1 2 -1
 media 4 1 3 -2
 media 5 1 4 -3
 media 2 1 5 -4
 boundary 5
unit 2
com='burnable poison rod'
 cylinder 1     0.545       160      -160
 cylinder 2     0.545    260.48      -160
 cylinder 3     0.645    260.58    -160.1
 cuboid 4       0.82     -0.82      0.82    -0.82    260.58    -160.1
 media 1 1 1
 media 0 1 2 -1
 media 4 1 3 -2
 media 2 1 4 -3
```

```
 boundary 4
unit 3
com='instrumentation hole'
 cuboid 1      0.82    -0.82      0.82    -0.82    210.34  -210.34
 media 2 1 1
 boundary 1
global unit 4
com='9x9 fuel assembly'
 cuboid 1      7.38    -7.38      7.38    -7.38    210.34  -210.34
 cuboid 2      7.48    -7.48      7.48    -7.48    210.34  -210.34
 cuboid 3      7.64    -7.64      7.64    -7.64    210.34  -210.34
 cuboid 4     15.26   -15.26     15.26   -15.26    225.58  -225.58
 array 1 1  place 5 5 1 0 0 0
 media 2 1 2 -1
 media 5 1 3 -2
 media 2 1 4 -3
 boundary 4
end geometry
read array
ara=1 nux=9 nuy=9 nuz=1 typ=square gbl=1
 com='9x9 assembly'
 fill
    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    3    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
```

8.5.2.   Two Parallel Slab Tanks (slabs.inp)

```
=csas26 parm=(centrm)
2 parallel slab tanks
v7-238
read composition
 solution   mix=1  rho[uo2f2]=459 92235 93.2
                                  92238 6.8
               molar[hfacid]=0
               density=1.566 temperature=300 vol_frac=1
  end solution
 ss304       2 1 300    end
 ss304       3 1 300    end
 h2o         4 1 300    end
 h2o         5 1 300    end
end composition
```

```
read celldata
  multiregion slab left_bdy=reflected right_bdy=vacuum end
          4              5
          2             5.5
          1            10.5
          3             11
          5             16
      end zone
end celldata
read geometry
global unit 1
com='single reflected ss304 slab tank'
 cuboid 1       2.5      -2.5       150      -150       75      -75
 cuboid 2         3        -3     150.5    -150.5       75    -75.5
 cuboid 3         8        -8     150.5    -150.5       75    -75.5
 media 1 1 1
 media 2 1 2 -1
 media 4 1 3 -2
 boundary 3
end geometry
read bnds
  body=3
    surface(1)=vacuum
    surface(2)=mirror
    surface(3)=vacuum
    surface(4)=vacuum
    surface(5)=vacuum
    surface(6)=vacuum
  end bnds
end data
end
```