

# **Conversion of Input Data between KENO and MCNP File Formats for Computer Criticality Assessments**

Phase 1 Final Report  
Instrument No: DE-FG02-05ER84178  
June 27, 2005 through March 26, 2006

Randolph Schwarz of Visual Editor Consultants  
Leland L. Carter of Carter Monte Carlo Analysis, Inc.  
Alysia Schwarz of Visual Editor Consultants

# Table of Contents

	<b>PAGE</b>
<b>1.0 ABSTRACT</b> .....	<b>3</b>
<b>2.0 SIGNIFICANT ACCOMPLISHMENTS</b> .....	<b>3</b>
2.1 KENO VA TO MCNP GRAPHICAL USER INTERFACE .....	4
2.2 CONVERTING A KENO VA FILE THAT DOES NOT CONTAIN AN ARRAY .....	7
2.3 CONVERTING A KENO VA FILE THAT CONTAINS A SINGLE ARRAY .....	9
2.4 CONVERTING OTHER SPECIAL DATA CARDS IN THE KENO VA INPUT FILE .....	12
<b>3.0 FUTURE WORK</b> .....	<b>13</b>
<b>4.0 REFERENCES</b> .....	<b>14</b>

## 1.0 Abstract

KENO is a Monte Carlo criticality code that is maintained by Oak Ridge National Laboratory (ORNL). KENO is included in the SCALE (Standardized Computer Analysis for Licensing Evaluation) (Reference 1) package. KENO is often used because it was specifically designed for criticality calculations. Because KENO has convenient geometry input, including the treatment of lattice arrays of materials, it is frequently used for production calculations.

MCNP (References 2-4) is a Monte Carlo transport code maintained by Los Alamos National Laboratory (LANL). MCNP has a powerful 3D geometry package and an extensive cross section database. It is a general-purpose code and may be used for calculations involving shielding or medical facilities, for example, but can also be used for criticality calculations. MCNP is becoming increasingly more popular for performing production criticality calculations.

Both codes have their own specific advantages. After a criticality calculation has been performed with one of the codes, it is often desirable (or may be a safety requirement) to repeat the calculation with the other code to compare the important parameters using a different geometry treatment and cross section database. This manual conversion of input files between the two codes is labor intensive.

The industry needs the capability of converting geometry models between MCNP and KENO without a large investment in manpower. The proposed conversion package will aid the user in converting between the codes. It is not intended to be used as a “black box”. The resulting input file will need to be carefully inspected by criticality safety personnel to verify the intent of the calculation is preserved in the conversion. The purpose of this package is to help the criticality specialist in the conversion process by converting the geometry, materials, and pertinent data cards.

## 2.0 Significant Accomplishments

In Phase I it was demonstrated that KENO Va input files could be converted to an MCNP format. The initial effort supported by Phase I only focused on converting KENO Va input files to MCNP input files. Both simple and complex KENO Va files, containing a single array, were converted to an MCNP format. A set of test files was developed using examples from the manual, examples from the class, and a number of simple geometries, to test the conversion of each of the different objects that exist in KENO Va (cubes, cuboids, cylinders, hemi-cylinders, hemi-spheres, spheres).

Along with the geometry, standard materials were converted to an MCNP format and the necessary running parameters were set in the MCNP input file based on the KENO Va input file. With this conversion, a number of KENO Va files could be run in MCNP.

## 2.1 KENO Va to MCNP Graphical User Interface

A program was written that combines a graphical user interface, written in C++, with a number of FORTRAN routines used to convert the KENO Va geometry to MCNP.

The C++ first reads the KENO input file. Because KENO Va builds a geometry from the inside to the outside, the order of the cards in the KENO input file is very important. After reading the KENO input file, the C++ will begin giving the geometry cards to the FORTRAN one card at a time in the proper order.

For each geometry card, the FORTRAN will first convert the KENO surface to MCNP surfaces. For a KENO cylinder, for example, there will be three MCNP surfaces, i.e., the cylinder surface, the upper plane of the KENO cylinder and the lower plane of the KENO cylinder. If this is the first KENO surface, the FORTRAN will then create a cell that is inside this KENO surface (inside the three MCNP surfaces for a cylinder). If it is not the first KENO surface, then the FORTRAN will create a cell with outer perimeter bounded by the KENO surface and the inner perimeter obtained from the previous KENO surface – this is a complicated portion of the conversion (a cylinder inside a cylinder, for example, will involve six MCNP surfaces).

This process continues for all the geometry cards, where other special KENO cards may indicate other features such as an individual “unit” or “hole” of the geometry that must be dealt with by the FORTRAN. After the geometry conversion is completed, the FORTRAN gives the MCNP cell and surface cards back to the C++. The C++ then deals with other types of cards, such as KENO cards specifying cross-sections, and then writes out the final MCNP input file.

The FORTRAN to C++ calls are designed to be the same as those used in generating the MCNP input file for the MCNP Visual Editor. The MCNP Visual Editor is the graphical user interface that is currently distributed with the MCNP code and was developed by Carter and Schwarz. This allows the same C++ files used for the creation of the MCNP cells and surfaces in the Visual Editor to be used to generate the MCNP cells and surfaces for the input generated from the KENO Va file.

Figure 2-1 shows the graphical user interface after a KENO Va file has been read in and converted, showing the top portions of the KENO and MCNP input files. On the left side of Figure 2-1 is the KENO Va input file and on the right side is the MCNP input file. Once the conversion has been made the user can save the MCNP input file and then read it into the Visual Editor for plotting.

Phase 1 Final Report  
 Conversion of Input Data between KENO and MCNP File Formats for Computer Criticality Assessments

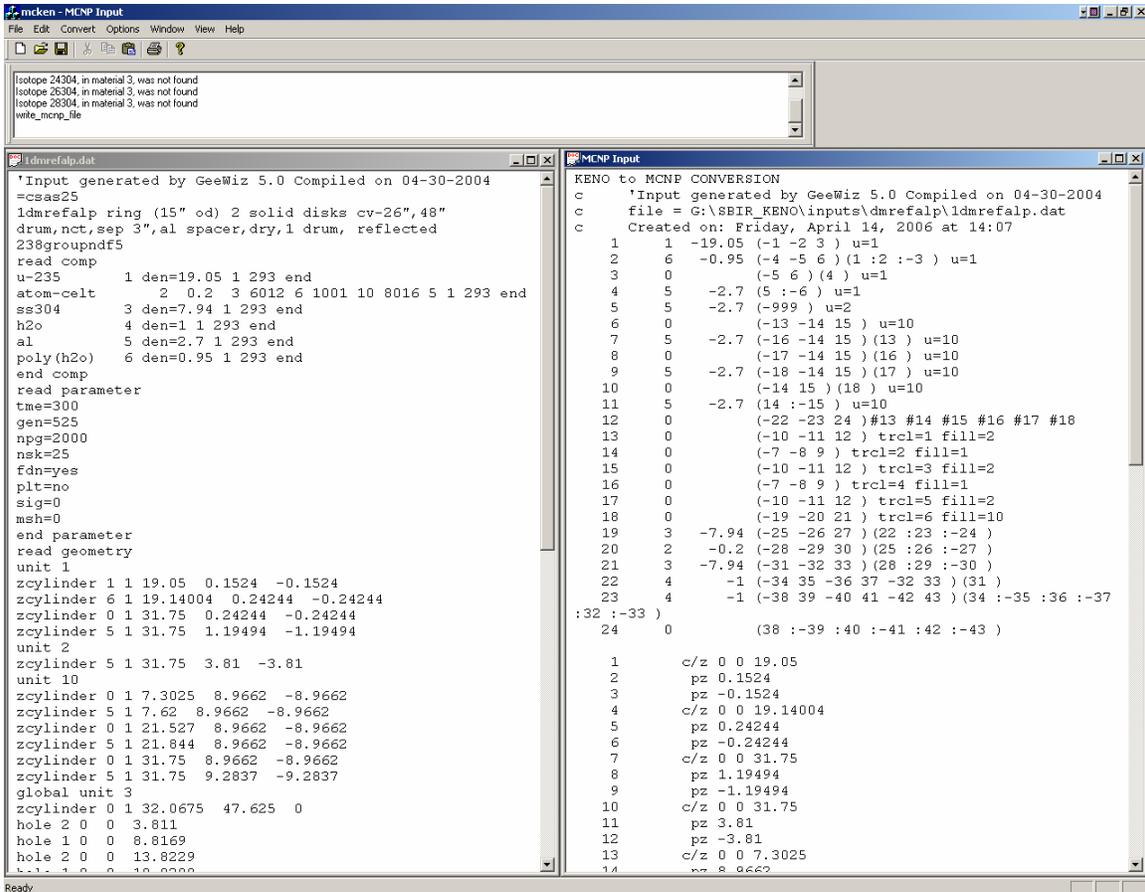


Figure 2-1. KENO Va to MCNP Conversion

Table 2-1 outlines the key words from the KENO Va file that are currently recognized and used in generating the MCNP input file. Included in the table are the corresponding MCNP surface cards that are generated.

**Table 2-1. KENO Va Geometry Information Supported for Conversion in Phase I**

<b>KENO Va input</b>	<b>Comment</b>
“com=” or ‘	<b>COMMENT CARD. ALSO THE NEXT CARD AFTER THE “END COMP” CARD IS A COMMENT CARD.</b>
read geom.	<b>SIGNALS THE BEGINNING OF THE GEOMETRY.</b>
unit	This signals the beginning of a special portion of the geometry like an MCNP universe.
hole	Geometry unit placed in a global unit. Converts to a fill of an MCNP universe.
global unit	Like “unit” except this defines a master unit.
cube	Cube. It sets $+X = +Y = +Z$ and $-X = -Y = -Z$ . Converts to 6 MCNP surfaces.
cuboid	Rectangular parallelepiped and may be described anywhere relative to the origin. Converts to 6 MCNP surfaces.
sphere	Sphere that is centered about the origin, unless otherwise specified by the optional region origin data. Converts to an MCNP general sphere.
cylinder	Cylinder that has its length described along the Z axis. Its centerline must lie on the Z axis, unless otherwise specified by the optional region origin data. Convert to an MCNP CZ cylinder and two PZ surfaces.
zcylinder	Cylinder that has its length described along the Z axis. Its centerline must lie on the Z axis, unless otherwise specified by the optional region origin data. Convert to an MCNP CZ cylinder and two PZ surfaces.
xcylinder	Cylinder that has its length described along the X axis. Its centerline must lie on the X axis, unless otherwise specified by the optional region origin data. Convert to an MCNP CX cylinder and two PX surfaces.
ycylinder	Cylinder that has its length described along the Y axis. Its centerline must lie on the Y axis, unless otherwise specified by the optional region origin data. Convert to an MCNP CY cylinder and two PY surfaces.
hemisphere	A spherical segment of one base whose spherical surface exists in the positive Z direction.
hemisphebc	A spherical segment of one base whose spherical surface exists in the <i>bc</i> direction ( $b = +$ or $-$ , $c = x, y, \text{ or } z$ ).
bhemicylcd	A cylindrical segment whose axis is in the <i>b</i> direction ( $b = x, y, \text{ or } z$ ) and whose cylindrical surface exists only in the <i>c</i> direction from a plane perpendicular to the <i>d</i> axis ( $c = +$ or $-$ , $d = x, y, \text{ or } z$ ).
origin	Change the origin of a sphere, hemisphere, cylinder, or hemicylinder.
chord	The distance from the "cut surface" to the center of a spherical surface or axis of a hemicylinder or hemisphere.
end geom.	End of the geometry.

## 2.2 Converting a KENO Va File that Does not Contain an Array

Figure 2-2 shows a plot of the KENO Va geometry for the input file shown in Figure 2-1. The geometry consists of cylinders of  $^{235}\text{U}$  on top of cylinders of aluminum and poly. The geometry consists of cylinders, units, global units and holes.

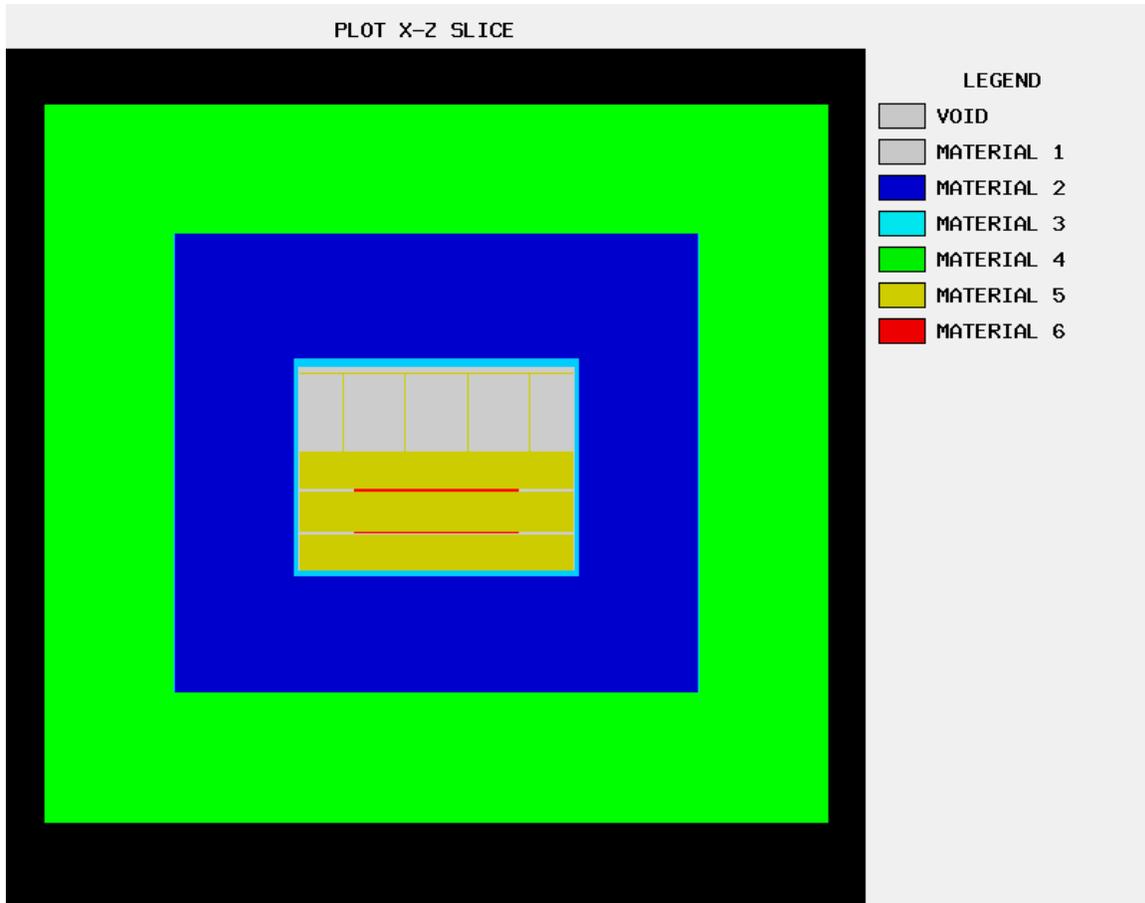


Figure 2-2. KENO Va Geometry Plot

Figure 2-3 shows the corresponding MCNP geometry created from the KENO Va input file. Each unit has been turned into a universe. These universes are then placed in the location where the holes occur in the KENO Va file, with a fill parameter used to place the universe in the hole.

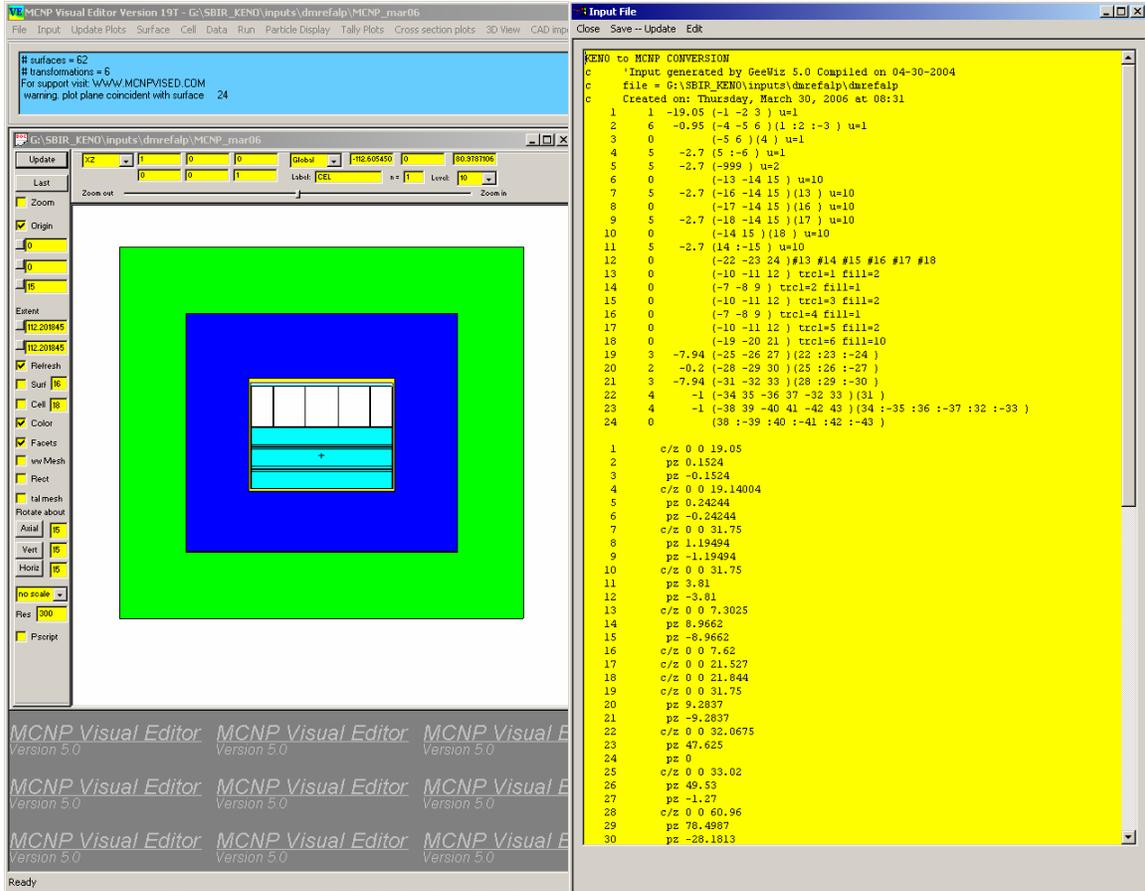


Figure 2-3. MCNP Geometry Plot

### 2.3 Converting a KENO Va File that Contains a Single Array

Table 2-2 shows the extended geometry keywords that are supported with the conversion program. For Phase I, only the conversion of a single array is supported.

**Table 2-2. Extended KENO Va Geometry Information Supported for Conversion in Phase I**

KENO Va Input	Comment
CORE	Create a rectangular parallelepiped that snugly fits the exterior of the array. The purpose of this region was to define the location of the array relative to the geometry regions external to the CORE. Converts to an MCNP lattice.
ARRAY	The geometry word ARRAY was added to allow specifying the array number and encasing the specified array in a snug fitting rectangular parallelepiped without having to specify the bias ID. Converts to an MCNP lattice.
REFLECTOR/ REPLICATE	Generates regions having the shape of the geometry region preceding the REFLECTOR description.
LOOP	LOOP enters unit orientation data in a manner resembling FORTRAN DO loops. The first field contains the unit number, followed by three fields that are treated like FORTRAN DO loops.
FILL	Enters data into an array by stringing in unit numbers starting at X = 1, Y = 1, Z = 1, and varying X, then Y, and then Z to fill the array. This converts to an MCNP fill matrix.

Figure 2-4 shows a KENO Va input file that contains an array and the MCNP input file that is generated after conversion. The top portion of the KENO input file is shown on the left and the top portion of the MCNP input file is shown on the right. Notice the array has been converted into a lattice (cell 3) for MCNP.

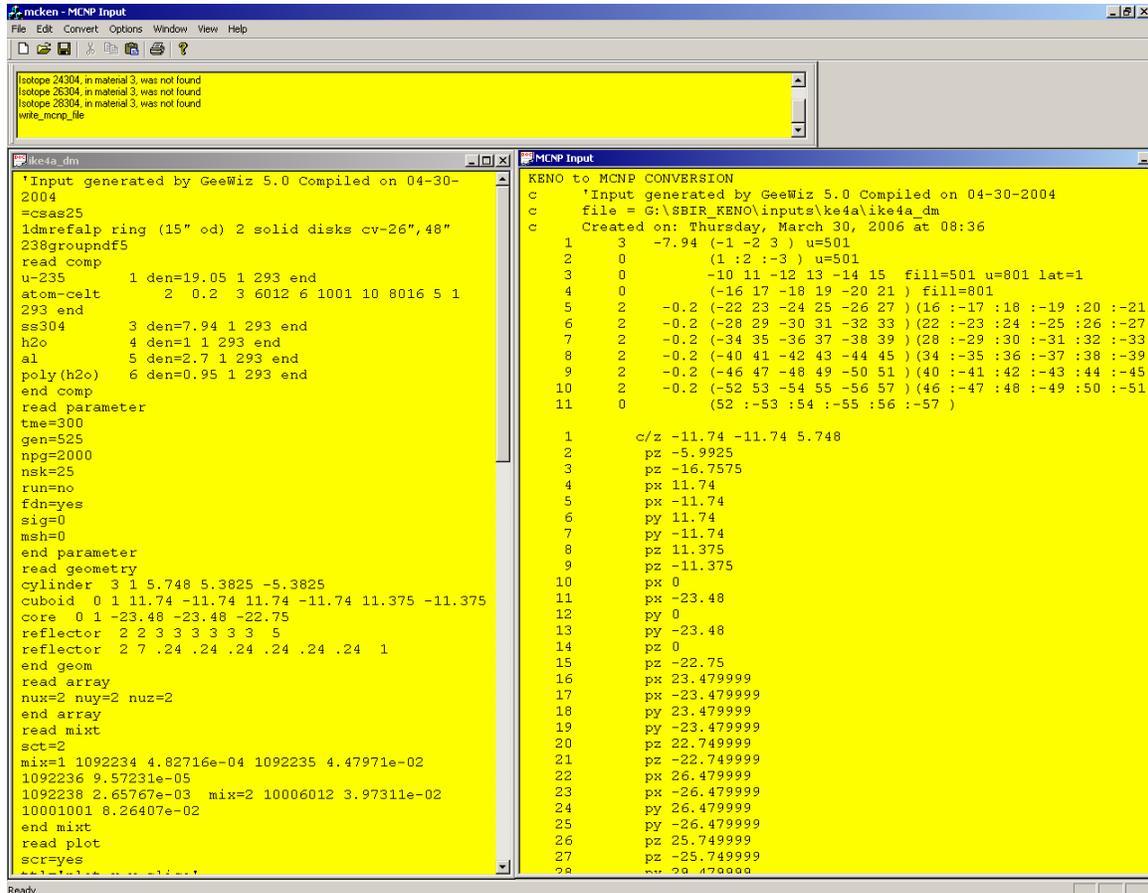


Figure 2-4. KENO Va to MCNP Conversion of an Array

Figure 2-5 shows a plot of a KENO Va geometry for the input file shown in Figure 2-4. The geometry consists of eight cylinders containing uranium in an array. The geometry consists of cylinders, cores/arrays, cubes, and reflectors.

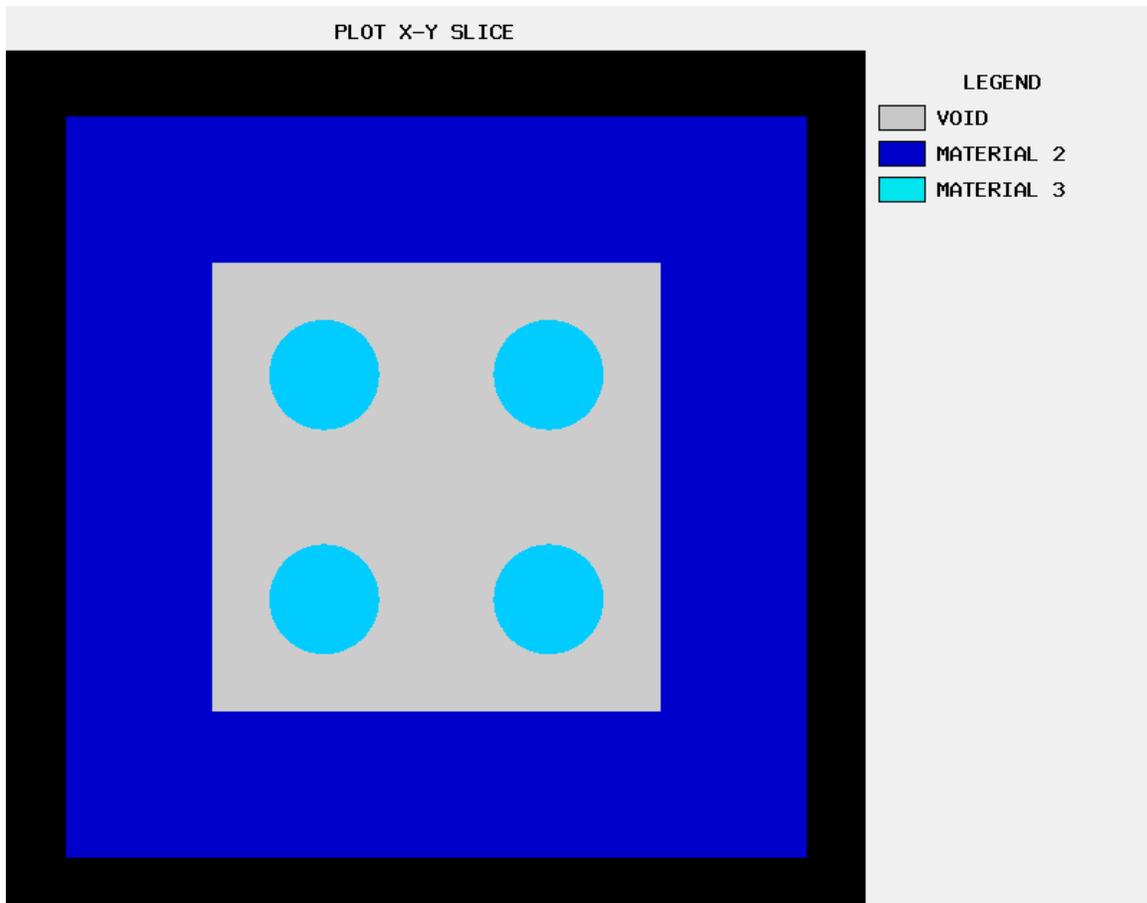


Figure 2-5. KENO Va Geometry Plot for an Array

Figure 2-6 shows the corresponding MCNP geometry created from the KENO Va input file. These elevation and plan view plots were created in the MCNP Visual Editor. The array has been turned into a lattice. The MCNP input file is shown at the bottom of the figure.

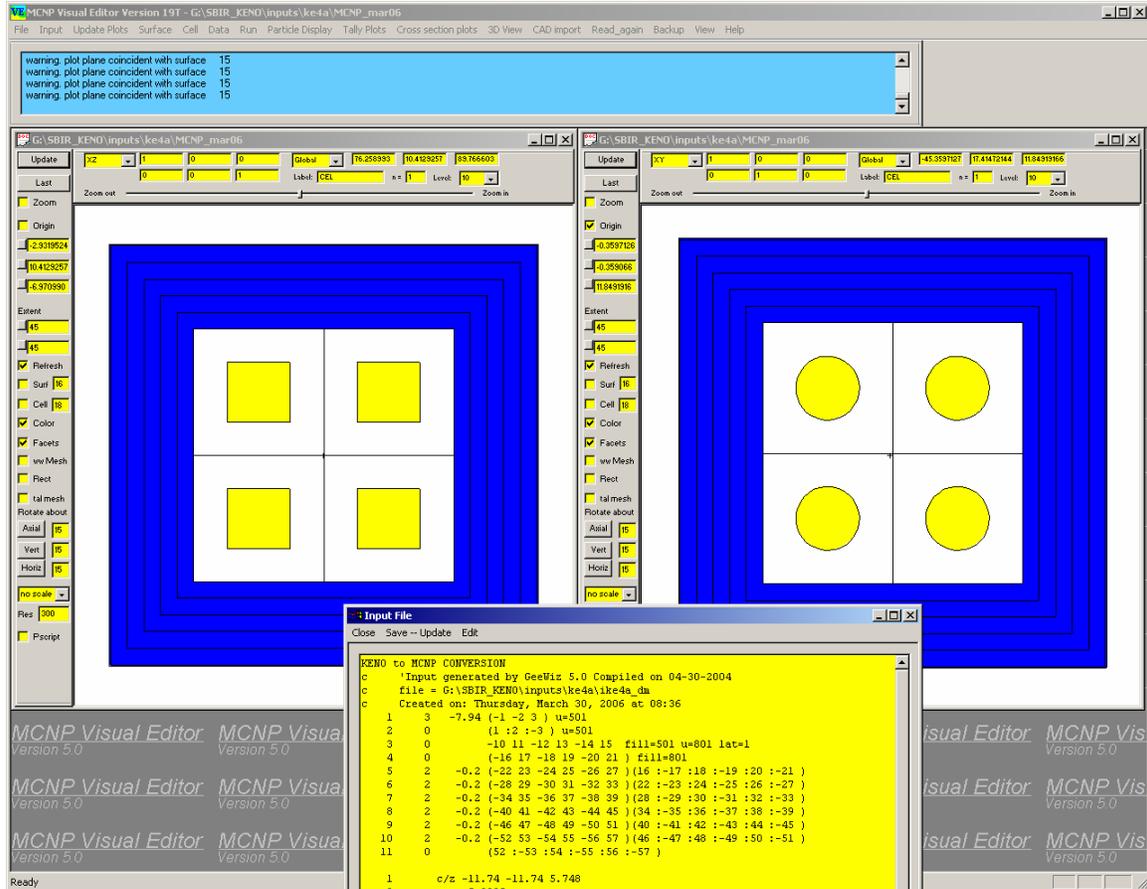


Figure 2-6. MCNP Geometry Plot of an Array

## 2.4 Converting Other Special Data Cards in the KENO Va Input File

The conversion of the KENO files includes the conversion of the standard materials. Not all materials could be converted. Materials that could be easily converted to MCNP were converted. To accomplish this task, the conversion program was linked to the KENO Va code, so the KENO Va code could be used to read in the material information from the KENO Va library files. Once this information was read in, it was sent to the conversion code to generate the MCNP materials.

In addition, parameters that had an MCNP counterpart were also converted as part of the Phase I work. For many problems, this enables the creation of an MCNP input file that will execute without further fine tuning. Table 2-3 shows the Keff values calculated by KENO Va and by the MCNP code generated from the conversion process.

**Table 2-3. Comparison of KENO Va and MCNP Calculations**

File	Keno	MCNP
2-2c8.inp (1)	<b>0.9981</b> +/- 0.0019	<b>0.99843</b> +/- 0.00135
2_cylinder_in_annulus.inp(2)	<b>0.9925</b> +/- 0.0017	<b>0.99180</b> +/- 0.00153
2_godiva_keno.inp(4)	<b>0.9963</b> +/- 0.0017	<b>1.00026</b> +/- 0.00130
godiva.inp(8)	<b>0.9963</b> +/- 0.0017	<b>1.00026</b> +/- 0.00130
godiva_gvs.inp(9)	<b>0.9978</b> +/- 0.0020	<b>0.99974</b> +/- 0.00128

The Phase I work has demonstrated the feasibility of converting simple KENO Va geometries and geometries containing a single array to MCNP format. The objectives set out in Phase I were accomplished as demonstrated by the above discussion.

### 3.0 Future Work

Future work should enhance the conversion program developed in phase 1 to work with more complex KENO Va files, in particular those containing multiple arrays. In addition, the conversion should be adapted to work with the latest KENO VI file formats. Work should also be performed to allow for the conversion of an MCNP file back to a KENO file.

The specific technical objectives that could be pursued in future work are listed in Table 3-1.

**Table 3-1. Technical Objectives**

Item	Objective
1	Enhance the Phase I conversion code to convert complex KENO Va files.
2	Test and debug the KENOVa to MCNP conversion.
3	Modify the converter to convert MCNP files to KENO format.
4	Test and debug the MCNP to KENO conversion.
5	Enhance the converter to convert KENO VI files to MCNP.
6	Test and debug the KENOVl to MCNP conversion.
7	Investigate and implement where possible the conversion of complex MCNP geometries to and from KENO format.
8	Investigate and implement graphical displays as necessary to aid in the conversion
9	Document the use of the conversion code.

## 4.0 References

1. "SCALE: A Modular Code System for Performing Standardized Computer Analysis for Licensing Evaluation," NUREG/CR-0200, Rev. 7, ORNLNUREG/CSD-2/R7, Vols. I, II, and III (2004).
2. X-5 Monte Carlo Team, "MCNP-A General Monte Carlo N-Particle Transport Code, Version 5 - Volume I: Overview and Theory," LA-UR-03-1987, April 2003 revised September 2003, distributed with RSICC package C00710.
3. X-5 Monte Carlo Team, "MCNP-A General Monte Carlo N-Particle Transport Code, Version 5 - Volume II: Users Guide," LA-CP-03-0245, April 2003, distributed with RSICC package C00710.
4. X-5 Monte Carlo Team, "MCNP-A General Monte Carlo N-Particle Transport Code, Version 5 - Volume III: Developers Guide," LA-CP-03-0284, April 2003, distributed with RSICC package C00710.