Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew
Gatlinburg, Tennessee, April 6-11, 2003, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2003)

3D DISPLAY OF VERY LARGE MCNPX AND MCNP LATTICES IN MORITZ

Kenneth A. Van Riper
White Rock Science
P. O. Box 4729, Los Alamos, NM 87544 USA
kvr@rt66.com

ABSTRACT

The Moritz geometry tool uses OpenGL to show an interactive 3D image that can be rotated, panned, and zoomed by the user. The display takes advantage of hardware graphics acceleration in personal computer video cards. The 3D image is based on primitives such as polygons and lines. When a large number of primitives are drawn, the graphics performance suffers and the image is not responsive to user input. MCNPX and MCNP geometry models containing large lattices can lead to poor 3D graphics performance. We discuss strategies implemented in Moritz to reduce the number of polygons and lines drawn when lattices are present. A number of options can be applied to universes that fill a lattice. These options include drawing only the largest cylinder in a universe, omission of inner shells and/or endcaps, reduced tessellation of cylinders, and the representation of the universe by a single line. A scan lattice is special type of lattice where the filling universes consist of single cells that completely enclose the lattice cell. Such lattices are often used to represent data from tomographic scans. Methods for enhancing the 3D display of scan lattices include drawing only elements that border both visible and invisible elements, and displaying points rather than a 3D box. We briefly describe a collapse method for enhancing 3D performance when displaying grid tally data in large meshes.

Key Words: MCNPX, MCNP, Lattice, Moritz, Grid Tally

1. INTRODUCTION

The use of lattices to describe geometry elements repeated in a regular 3-dimensional grid is a powerful feature of MCNPX [1] and MCNP [2]. Lattices are often used in models of reactor cores and nuclear fuel shipping casks (such as in Figs. 1 and 2). Another use is the representation of voxelized data from tomographic scans. The lattice feature enables a compact description of complicated geometries in the input file.

The ability to view the full three-dimensional (3D) model, including lattices, aids in the verification, understanding, and debugging of the model. Ray tracing programs, such as Sabrina [3, 4, 5], produce a static 3D image. Although the model can be viewed from any direction and with any desired set of cells made visible, each new image requires seconds to minutes to compute and display. We are currently developing the Moritz [6, 7] program that offers interactive 3D graphics with which the user can manipulate—rotate, scale, and move—the image with mouse movements and keystrokes.

The Moritz 3D graphics display is built on the OpenGL application interface [8]. OpenGL is a graphics standard that has been widely adopted for 3D computer graphics. In particular, it is

widely used for computer games, which in turn has spurred video card manufacturers to include support for OpenGL, in the form of hardware graphics acceleration. As a result, 3D graphics performance on even moderately priced current personal computers is very good. Given a set of polygons and the viewpoint, the OpenGL library processes the transformations and hidden surface removals required to produce an image on the screen.

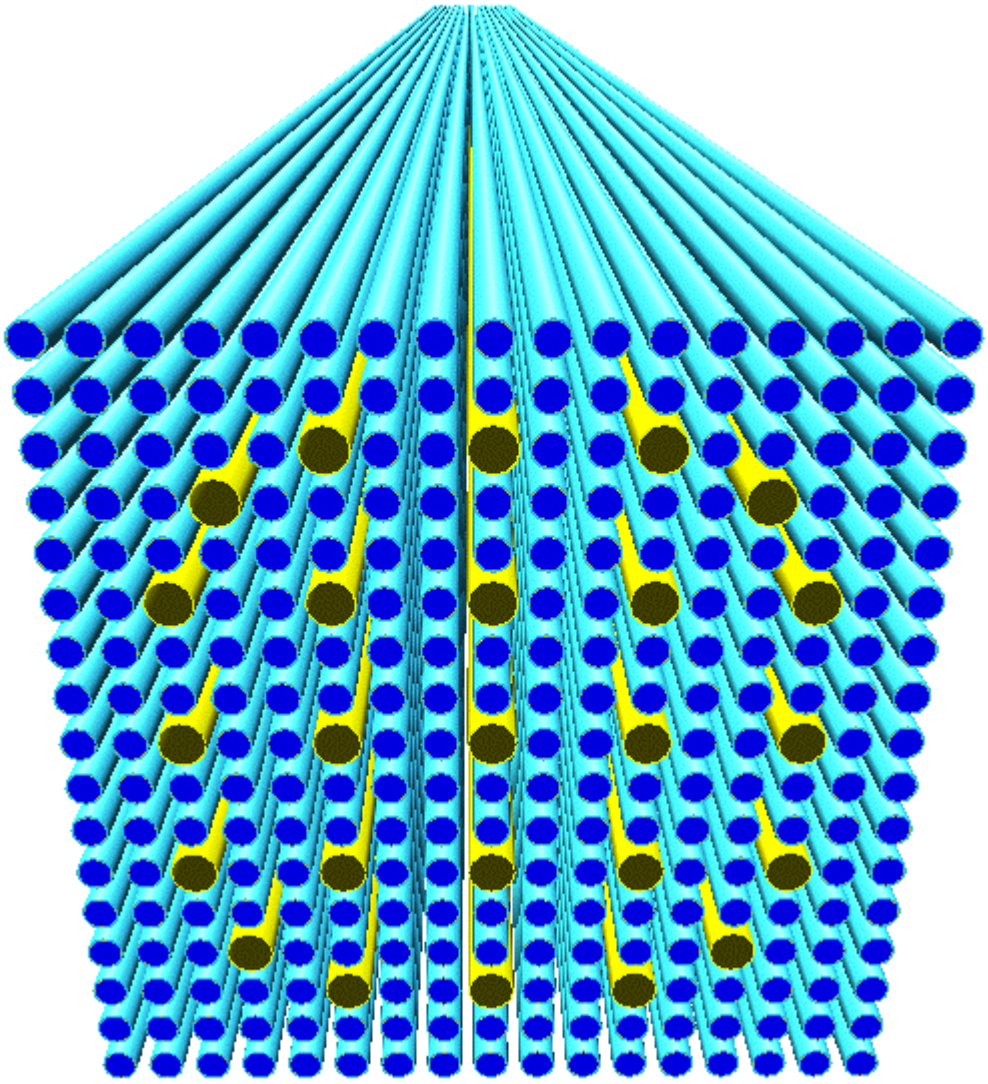

Figure 1. 3D image of a fuel rod bundle.

The performance of a video card is measured, in part, in the number of graphics primitives that it can draw per second. Polygons comprise the majority of these primitives that may also include points and lines. For a small number of polygons, image motion requested by user input appears instantaneous. When the number of polygons in a geometry model exceeds the capacity of the display hardware to render the image in about 1/10 of a second, changes in the image orientation, size, or position cannot keep pace with the user input, resulting in poor 3D graphics performance.

In extreme cases, where the polygon count greatly exceeds the video capability, only a partial image or no image at all is seen.

We have encountered the problem of poor 3D performance most often in models containing large lattices. Any model, not necessarily including a lattice, with a sufficiently large number of primitives will show poor performance. Performance degradation is also seen when displaying grid tally data on very large meshes.

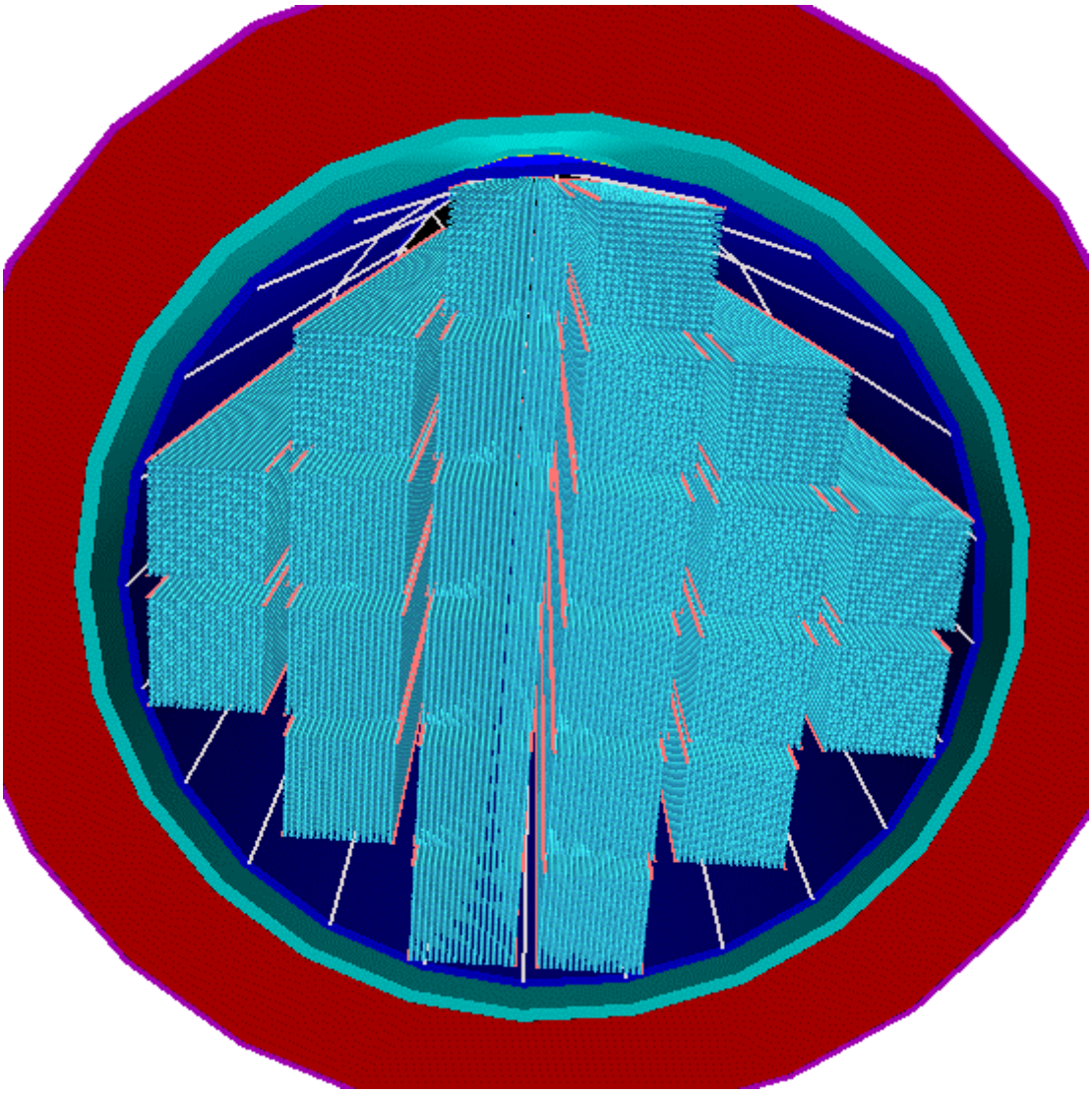

Figure 2. 3D image of a shipping cask model.

Lattices require some additional processing in Moritz before generation of the polygons. To determine if these extra computations affect the rendering time, we constructed a 61 × 61 × 1 lattice filled with identical cylinders and an equivalent model with each cylinder explicitly

defined. (This model size just exceeds the size where performance begins to be noticeably impaired on our *NVIDIA TNT2 M128 4xAGP Ultra* graphics card.) The rendering times were the same.

To retain adequate 3D performance, the number of polygons drawn must be reduced. The reduction can be accomplished by omitting some cells or parts of cells and by decreasing the resolution with which cells are drawn. (A cell *part* is separately drawn portion of a cell used when union operators combine distinct volumes into a single cell.) We describe below the methods available in Moritz for accomplishing these reductions. The methods can be implemented in the user interface or by inserting special comments in the MCNP(X) input file.

2. UNIVERSE RESOLUTION

A lattice consists of 1-, 2-, or 3-dimensional grid of elements. Each element is filled with a *universe* that is a group of one or more cells. The origins of the universe's cells are translated by the offset of the lattice element that it fills. There can be multiple universes filling different elements of the lattice. A filling universe may contain another lattice.

Moritz contains a number of settings that can be used to reduce the number polygons used when drawing a universe. The settings can be applied to all lattice filling universes or to specific universes so that different strategies may be used for different universes. Abbreviated representations of a universe can be used while scaling and positioning the 3D image and applying clip planes. A more realistic representation can then be used once the desired view has been reached.

Large lattices are often encountered in reactor and reactor fuel shipping cask models where many cells are cylindrical, such as in Figs. 1 and 2. A number of reduction settings affect how cylinders are drawn.

2.1. Largest Cylinder Only

One can choose to draw only the largest cylinder in a universe. The cylindrical cell (or cell part) with the largest radius is drawn when the box is checked. All other cells are omitted. Only right circular (RCC) and elliptical (ERCC) cylinder shapes are considered. For an ERCC, the largest semi-major axis radius is used in the largest radius test. If no RCC or ERCC is found, all the universe's cells are drawn.

For the models shown in Figs. 1 and 2, the largest cylinder extends the full length, or nearly the full length, of the universe. In other models, however, the largest cylindrical element may be divided into multiple cells. At present, only a single cell is shown, not all cells that share the same largest cylindrical surface. This restriction also affects the single line mode described in §2.4.

2.2. Omission of Inner Shells and Endcaps

Many fuel rod models consist of a number of nested cylinders. The default representation of a cell between two cylinders includes both the outer and inner cylinders and the endcaps formed by the planes perpendicular to the axis. The user may omit the inner shells and/or the endcaps.

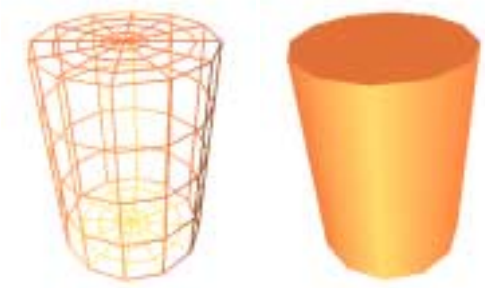

Figure 3. Default cylinder tessellation.

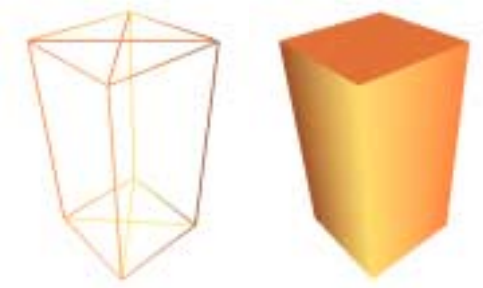

Figure 4. Reduced cylinder tessellation.

2.3. Alternate Cylinder Tessellation

The division of a cylinder into planar polygons is known as tessellation. Fig. 3 shows wireframe and solid representations of an RCC shape with the default tessellation that uses 12 angular divisions, 4 stacks along the axis, and 4 rings in the endcap. Each of the 144 regions on the surface bounded by the wireframe lines is a polygon in the solid representation. Alternate tessellations can be used for all cylinders in a universe. Fig. 4 shows a tessellation using 4 angular divisions and no stacks nor rings, resulting in 12 polygons. The tessellation of cylinders in other universes (or in no universe) is not affected.

2.4. Single Line

The universe can be represented by a single line drawn at the center of the largest cylinder that is found as described above for the largest cylinder only setting. If no such cylinder is found, all the universe's cells are drawn. The user can specify the thickness of the line.

2.5. Omit Universe

If necessary, the entire universe can be omitted when used to fill a lattice element.

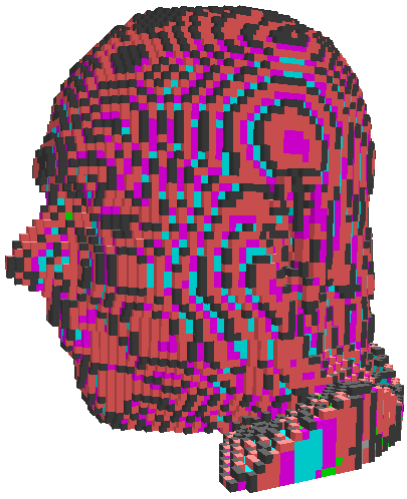

Fig 5. Scan lattice drawn in default style. **Fig. 6. Scan lattice drawn as points.**

3. SCAN LATTICES

A particular type of large lattice is a scan lattice. They are large hexahedral (6-sided) lattices with each element filled with a single cell that is larger than the element. They are often used for calculations based on a tomographic scan. Figure 5 shows an example. Moritz assigns a lattice to the scan type if the cells filling each element are sufficiently larger than the lattice cell so that only the unit cell surfaces are used for the polygonalization of the cell. This method of filling is most easily accomplished with a single filling cell inside a sphere centered on the lattice unit cell with a radius much larger than the unit cell dimensions.

Because scan lattices are often very large, the 3D performance can be poor if the usual methods are used for the 3D display. On some computers, the lattice may not appear at all. In addition to the default drawing algorithm used for other lattices (Fig. 5), three special drawing modes exist.

In the *visible elements* mode, only those lattice elements that are bounded by both visible and invisible lattice elements are drawn. This mode is useful for compact objects, such as the head in Figs 5 and 6. It will produce the same image as in the default style shown in Fig. 5 when viewed from the outside. Clip planes, however, will reveal a hollow interior.

In the *points* mode, shown in the Fig. 6, the lattice elements are represented by points at the element position. The points are square (rather than rounded). The point size is set by the user. If the size is too small, points in the interior will be visible. If too large, elements will overlap and/or be obscured. The *visible points* mode is a combination of the *points* and *visible elements* modes.

The cells on the periphery of the lattice may contain an uninteresting material, such as air surrounding a body part. When the *edge material invisible* mode is in effect, after a scan lattice is read, the material that occupies the greatest number of cells in the exterior layers of the lattice is made invisible. Changing the mode does not affect the display of existing lattices.

4. PERFORMANCE ISSUES WITH GRID TALLY DATA

The grid tally feature of MCNPX permits the calculation of tally quantities such as flux and dose rate on a Cartesian, cylindrical, or spherical mesh. In addition to capabilities for defining and displaying the mesh, Moritz can display the tally data in both 2D and 3D for Cartesian and cylindrical meshes. The 3D representation of the data, as for cells, is based on polygons. Each data cell of a Cartesian mesh is a box with 6 sides. A modified cylindrical shape is used for data cells in the cylindrical mesh. Very large meshes can result in too many polygons for adequate 3D performance. To reduce the number of polygons, the user can collapse the data so that it is displayed on a plane lying at the center of the mesh elements in the chosen direction. For a Cartesian mesh, the number of polygons is reduced by 1/6. The user can specify in which direction the collapse takes place, or let the program choose the direction with the least number of mesh lines. Fig. 7 shows a collapsed Cartesian grid tally. There is a single mesh element in the direction perpendicular to the plane extending to the limits of the box shown in wire frame. Fig. 8 shows a cylindrically symmetric grid tally collapsed in the Θ (around the axis) direction. The circular surface is an optional data plane.

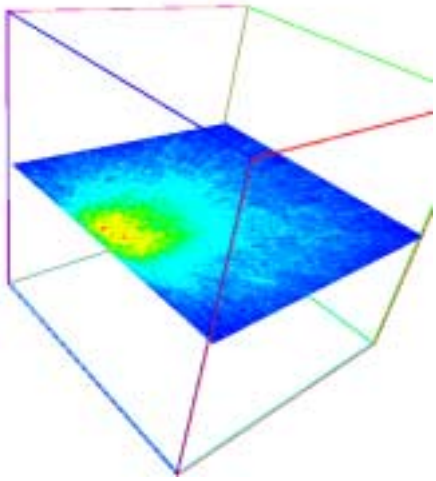

Fig. 7. A collapsed Cartesian grid tally.

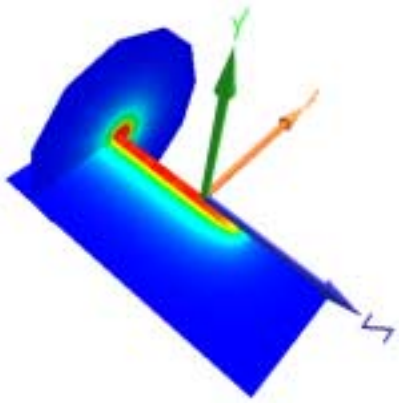

Fig. 8. A collapsed cylindrical grid tally.

5. CONCLUSIONS

Interactive 3D graphics is powerful tool for understanding and presentation of a geometry model, but interactivity can suffer when the model is rendered using a very large number of polygons. Problems with 3D performance are most often seen in models with large lattices or with grid tally data on a large mesh. Moritz allows the user to access a number of techniques to reduce the number of polygons drawn in these cases, thereby restoring acceptable 3D performance.

ACKNOWLEDGMENTS

We are grateful to Laurie Waters and the MCNPX team for supplying the example models used for Figs. 5, 6, and 7 and for more models that have aided the development of Moritz, to Dom Napolitano for the models on which Figs. 1 and 2 are based, and to Firas Mourtada for the grid tally shown in Fig. 8.

REFERENCES

1. L. S. Waters, Editor, *MCNPX User's Manual*, Los Alamos National Laboratory Report TPO-E83-G-UG-X-0001, (1999).
2. J. F. Briesmeister, Editor, *MCNP – A General Monte Carlo N-Particle Transport Code*, Los Alamos National Laboratory Report LA-13709-M, (2000).
3. James T. West III, *SABRINA: An Interactive Three-Dimensional Geometry-Modeling Program for MCNP*, Los Alamos National Laboratory Report LA–10688–M (1986).
4. Kenneth. A. Van Riper, *Sabrina User's Guide*, Los Alamos National Laboratory Report LAUR–93–3696 (1993).
5. Kenneth. A. Van Riper, "New Features in Sabrina," *Proceedings of the Topical Meeting on Radiation Protection for our National Priorities*, Spokane, WA, Sept. 17–21, 2000, pp. 316–323 (2000).
6. Kenneth. A. Van Riper, "Interactive 3D Display of MCNP Geometry Models," *Proceedings of the ANS International Meeting on Mathematical Methods for Nuclear Applications*, Salt Lake, UT, Sept. 10–13, (2001).
7. Kenneth. A. Van Riper, "Creation and Viewing of MCNP/MCNPX Meshes and Grid Tally Data in Moritz," *Proceedings of the Topical Meeting on Radiation Serving Society*, Santa Fe, NM, April 14–18, (2002).
8. Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner, *OpenGL Programming Guide*, Reading MA: Addison–Wesley, (1999).