GRAVE: An Interactive Geometry Construction and Visualization Software System for the TORT Radiation Transport Code

E. D. Blakeman
Oak Ridge National Laboratory
Oak Ridge, TN 37831
edb@ornl.gov

ABSTRACT

A software system, GRAVE (Geometry Rendering and Visual Editor), has been developed at the Oak Ridge National Laboratory (ORNL) to perform interactive visualization and development of models used as input to the TORT three-dimensional discrete ordinates radiation transport code. Three-dimensional and two-dimensional visualization displays are included. Display capabilities include image rotation, zoom, translation, wire-frame and translucent display, geometry cuts and slices, and display of individual component bodies and material zones. The geometry can be interactively edited and saved in TORT input file format. This system is an advancement over the current, non-interactive, two-dimensional display software. GRAVE is programmed in the Java programming language and can be implemented on a variety of computer platforms. Three-dimensional visualization is enabled through the Visualization Toolkit (VTK), a free-ware C++ software library developed for geometric and data visual display.

Future plans include an extension of the system to read inputs using binary zone maps and combinatorial geometry models containing curved surfaces, such as those used for Monte Carlo code inputs. Also GRAVE will be extended to geometry visualization/editing for the DORT two-dimensional transport code and will be integrated into a single GUI-based system for all of the ORNL discrete ordinates transport codes.

1. INTRODUCTION

A software system, entitled GRAVE (Geometry Rendering and Visual Editor), has been developed at the Oak Ridge National Laboratory (ORNL) to perform interactive visualization and development of models used as input to the TORT[1] three-dimensional discrete ordinates radiation transport code, also developed at ORNL. Visualization displays include both three-dimensional and two-dimensional models. Capabilities within these models include geometry rotation, zoom, and translation. Also, cuts and slices or individual components of the geometry can be displayed, and the geometry can be interactively modified such that component parts are removed, added, or resized. This system represents a significant advancement over current software, which allows only non-interactive two-dimensional cross-sectional geometrical displays with no edit capabilities.

This paper gives an overview of the main features of GRAVE and discusses enhancements for the future. Also, the future integration of GRAVE into a larger Graphical User Interface (GUI) for all the ORNL discrete ordinates codes is discussed.

2. PROGRAMMING SOFTWARE

GRAVE is object-oriented in design and is written in the Java computer programming language. Java was selected because of its cross-platform portability and readily available GUI classes that are integrated into the language. GRAVE has thus far been demonstrated on PCs (running Windows 95/98) and on SUN workstations. It is anticipated that it will also be usable on other Unix-based workstations and PCs running the Linux operating system.

Three-dimensional geometric visualization is enabled in GRAVE through the incorporation of modules from the Visualization Toolkit [2] (VTK), an extensive free-ware library of rendering software developed specifically for geometric and data visual display. VTK is written in the C++ computer language but includes "wrapper" code, so that it can be included in Java-based code using the Java Native Interface (JNI). A disadvantage of this approach is that the VTK library must be compiled for each computer platform. However, VTK has been ported to most common computer platforms; hence, this does not appear to be a limiting issue.

Two-dimensional visualization is performed using a set of Java graphics classes, the 2D Application Programming Interface, also known as Java 2D. Since Java 2D is included within the Java package, no additional software is required for two-dimensional rendering.

3. TORT GEOMETRY

TORT utilizes a simple overlay geometry method in which a geometry model is constructed from simple box-like objects, referred to as right parallelepipeds (RPPs) in Cartesian (XYZ) geometry, or as "bodies" in the more general sense (includes "pie slices" obtained from a cylindrical (R-Θ-Z) geometry). For a given geometry, each body is uniquely assigned to a zone and each zone contains a specified material. The same material can be assigned to more than one zone. Bodies may intersect, wholly or partially; thus one body may completely enclose another. TORT input uses a simple rule of geometric combination in which each body overlays all those that are listed previously. Thus, the material assigned to a coordinate location is that assigned to the zone of the last body which contains that location. Clearly, the order in which the bodies are listed is of utmost importance. In a typical TORT model larger bodies are listed first, then the smaller enclosed bodies. Otherwise, if the smaller enclosed bodies were listed first, they would be "swallowed" by the larger enclosing bodies and as a result, have no effect.

TORT geometry models are also comprised of fine mesh cells. These cells, which should not be confused with bodies, are the volumes between consecutive fine mesh boundaries in the three dimensions. TORT allows both continuous and discontinuous mesh boundary options. In a continuous mesh, each fine mesh boundary defines a plane that runs through the entire geometry at that boundary location. That is, the mesh boundaries in one dimension are the same in that dimension for all locations in the other two dimensions. In a discontinuous mesh, however, the situation is relaxed somewhat, and the fine mesh boundaries in a dimension can vary with the locations in the other two dimensions. Appendix D of Reference 1 provides a discussion of the TORT discontinuous mesh capabilities. Currently, GRAVE is limited to continuous fine mesh problems.

It is not uncommon for a geometrical model representing a complex system to be comprised of hundreds of bodies (and perhaps a million or more cells). This situation often occurs because it is difficult to accurately represent complex or curved volumes in a Cartesian geometry by simple box-like objects that contain only right angles. A TORT geometry representation of a curved surface exhibits a "stair-case" appearance, which is usually an accurate approximation only if a large number of bodies are used. In such a case, it is difficult and tedious to develop or revise a complex three-dimensional geometry without frequent access to visual images of the geometry. An error in the boundary location of a body, the zone/material assignment, or in the body-input sequence, can result in a completely invalid model. Two-dimensional planer slices and three-dimensional views that can be examined while the model is under development and can be quickly updated when changes are made are invaluable to minimizing model errors.

4. USER INTERACTION

4.1 GENERAL REMARKS

Interaction with GRAVE is through a main Graphical User Interface (GUI) window. This window contains tabbed panes for release notes, web links, and a user's manual. Three additional windows are used for the interactive visualization and editing of TORT geometry models. These include the Edit Window for displaying properties of the TORT file being edited or constructed, the 3D Visualization Window for displaying three-dimensional model views, and the 2D Visualization Window for displaying two-dimensional planes along any of the three coordinate axes. The Edit Window can be launched from the main window and the two visualization windows are opened from within the Edit Window.

Currently, neither of the visualization windows can be used to edit the geometry. The overall intent in this approach is to confine changes to the geometry model to the Edit Window and to view updated versions of the model in the visualization windows. Multiple copies of each window can be opened. Thus, it is possible to simultaneously edit multiple geometry models and have multiple views of each model.

Files opened via the main window are in ASCII format with input arrays specified in the TORT input directions[1]. Input of a data file to GRAVE is accomplished by a Java input processor that recognizes and correctly parses Floating Index Data Operation (FIDO)[1] notation that permit interpolation, repetition of values, etc. Therefore, input files may fully utilize FIDO notation.

4.2 EDIT WINDOW

In the Edit Window, geometric properties including mesh interval spacing, the number and boundaries of geometry bodies, zone assignments to bodies, material assignments to zones, and color assignments used for distinctive visualization of materials, are summarized and displayed in editable tables. The tables are conveniently placed in individual tabbed panes. These properties are interrelated so that if one property is changed, other related properties are updated automatically.

The Edit Window can be used to display and edit geometry information from an existing TORT input file or to interactively build a new TORT geometry which can later be saved in ASCII format as a new file. A geometry that is saved is placed into a “partial” TORT file that contains the appropriate geometry arrays. This file may subsequently be incorporated into a complete TORT input file.

Figure 1 shows an example of the Edit Window. In this figure the tabbed pane showing the bodies, including their zone assignments and boundaries, which comprise the model geometry, is displayed.

4.3 3D VISUALIZATION WINDOW

The 3D Visualization Window contains numerous display options including rotation, zoom, and the selective display of component geometry parts. In addition, the geometry may be sliced at any location along a plane perpendicular to one of the three axes. The developer may also view the geometry as a “wire-frame” or in a translucent mode in which some visibility through the component bodies is allowed.

A very useful display option is available in the 3D Visualization Window in which the bodies comprising a geometry model can be sequentially removed (“peeled”) from or added to the model. In the peeling process, the bodies are removed from visibility one at a time from the top of the input list. This has the effect of sequentially stripping away the outer parts of the model to reveal the inner components. In the addition process, the opposite occurs; the geometry model is constructed from the bottom of the list up. This has the effect of visually constructing the geometry from the inner to the outer components. Using a combination of both approaches, each conveniently performed by the click of a computer mouse button, the user can rapidly view the geometry from the perspective of the component parts. In other options, selected bodies or material zones can be displayed in desired combinations. These visualization options, specialized specifically to the TORT modeling methodology, enable better understanding of geometry models, and the rapid isolation of modeling errors.

Figure 2 shows an example of a sample geometry model display in the 3D Visualization Window. The model, which is comprised of 125 RPP bodies, is of radiation beam tube number 4 (HB-4) in the ORNL High Flux Isotope Reactor [3] (HFIR). In this figure, the first body (water) has been peeled away to expose the inner bodies which would otherwise not be visible from the perspective shown. The reader can clearly see the staircase approach that must be used to model curved surfaces.

4.4 2D VISUALIZATION WINDOW

The 2D Visualization Window contains similar but somewhat different display options including plane selection and the ability to highlight or selectively display geometric regions. Using a slider control, the user can quickly scan the planes at any location on any of the three coordinate axes. Thus, although the 3D Visualization Window provides a full three-dimensional view of the model from any perspective, the 2D Visualization Window is more convenient for precise viewing at a particular plane of the geometry.

An example of a sample geometry model display in the 2D Visualization Window is shown in Figure 3. This example is of the same model as depicted in Figure 2, but shows a planar cut of the geometry through the vertical mid-plane of the HFIR core and through the center of the beam tube along its axis.

5. FUTURE EFFORTS

The current version of GRAVE is limited to Cartesian (XYZ) models using a continuous cell mesh. Input/output files are limited to standard ASCII TORT input files in which geometry components are described as combinations of RPPs/bodies. An alternate form of input is a binary file (referred to as a VARMAP) that specifies the material zone at each cell location. Future versions of GRAVE will be able to read, write and display this type of input as well as ASCII input using a cylindrical coordinate system (R-Θ-Z) and/or a discontinuous cell mesh.

In addition, development of the ability to input files with geometry components such as spheres and cylinders that would typically be found in a Monte Carlo input editor, is under consideration.

It is anticipated that GRAVE would first convert an input of this type to a TORT overlay geometry or VARMAP representation. Additional information such as volume fractions of material regions would be calculated which could be used to produce a geometry that would optimize specified parameters, e.g. conservation of the total volume of a region. An automatic mesh generator could also be implemented that would produce the fine mesh boundaries used for the discrete ordinates calculations. Another future consideration is to interface GRAVE with an existing geometry editor or with a CAD/CAM system.

Although GRAVE is a standalone software system that is currently specialized only to the development and visualization of TORT geometry models, it is intended that it will be extended in applicability to the ORNL Two-Dimensional Discrete Ordinates Transport Code[4,5] (DORT), and also be integrated into a broader system. Currently, the development of a GUI for the ORNL One, Two, and Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code System[5] (DOORS) is in progress. In addition to radiation transport codes, the DOORS system includes numerous auxiliary codes. The intent is to integrate all of these codes into a single unified GUI-based system. A GUI for the Group-organized cross-section Input Program[6] (GIP), an auxiliary code which produces material mixtures and nuclear cross sections in a format required by TORT, is also presently under development.

The DOORS GUI will ultimately enable the production of complete code inputs as well as the ability to initiate runs of the codes and monitor their progress and performance while running. Additional modules for visualization of output data, such as the overlay of neutron/photon flux or radiation dose contour plots onto geometry displays, are also under consideration.

CONCLUSIONS

A brief discussion of the principle features of GRAVE has been given. This code is a tool that is designed to interactively aid the analyst in the development of models that use the TORT discrete ordinates transport code. The code enables editing of model parameters and uses two- and three-dimensional visualization to view updated models. GRAVE is coded in the Java programming language, which enables its use on a wide variety of computer platforms.

It is anticipated that many new features will be added to GRAVE in future releases. In particular, inputs specified in the binary zone map (VARMAP) format or inputs containing curved surfaces, e.g. Monte Carlo combinatorial geometry models, will be included. It is also anticipated that the capabilities of GRAVE will be extended to include visualization/editing of geometry models for the DORT two-dimensional transport code and GRAVE will be included as a module in a GUI for the ORNL DOORS transport code system.

ACKNOWLEDGEMENTS

The author would like to acknowledge technical support from Y. Y. Azmy, and technical and managerial support from D. T. Ingersoll. Both colleagues provided encouragement and invaluable suggestions for the GUI features. In addition, appreciation is extended to R. W. Lee for assistance with the Java programming language.

REFERENCES

1. W. A. Rhoades and D. B. Simpson, "The TORT Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code (TORT Version 3)," Oak Ridge National Laboratory, Oak Ridge, TN, ORNL/TM-13221 (October 1997).

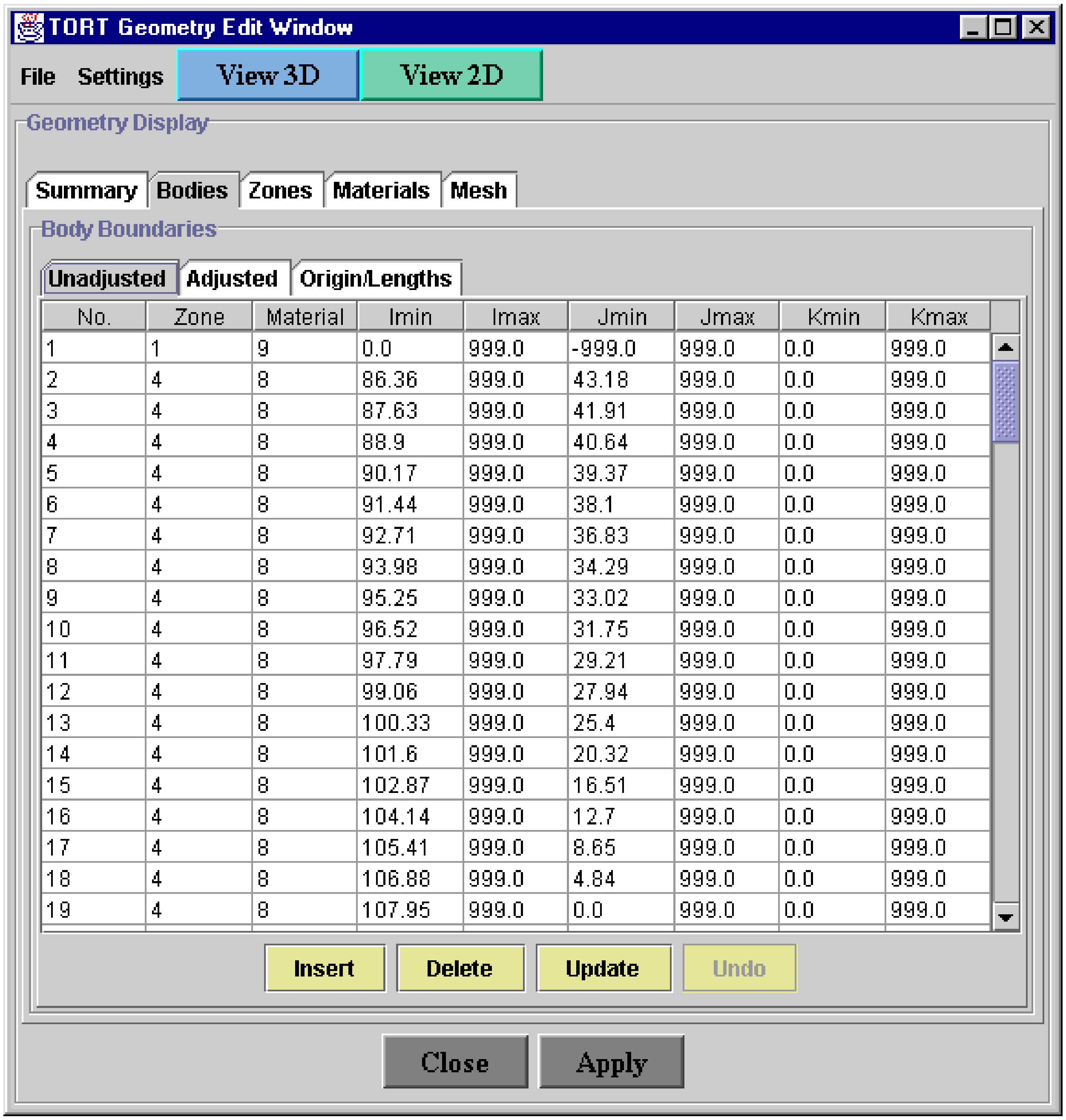
2. Will Schroeder, Ken Martin, and Bill Lorenson, *The Visualization Toolkit, An Object-Oriented Approach to 3D Graphics*, 2nd Edition, Prentice Hall (1997).

3. R. D. Cheverton and T. L. Dickson, "HFIR Vessel Life Extension with Enlarged HB-2 and HB-4 Beam Tubes," Oak Ridge National Laboratory, Oak Ridge, TN, ORNL/TM-13698 (December 1998).

4. W. A. Rhoades and R. L. Childs, "The DORT Two-Dimensional Discrete Ordinates Transport Code," *Nucl. Sci. & Engr*. **99**,1, pp. 88-89 (1988).

5. "DOORS3.2: One, Two- and Three-Dimensional Discete Ordinates Neutron/Photon Transport Code System," Radiation Safety Information Computational Center Computer Code Collection, Oak Ridge National Laboratory, Oak Ridge, TN, CCC-650, (July 1998).

6. "GIP Code System for Preparation of Group-Organized Cross-Section Libraries", Radiation Safety Information Computational Center Peripheral Shielding Routine Collection, Oak Ridge National Laboratory, Oak Ridge, TN, PSR-229, (March 1990).



No.	Zone	Material	Imin	Imax	Jmin	Jmax	Kmin	Kmax
1	1	9	0.0	999.0	-999.0	999.0	0.0	999.0
2	4	8	86.36	999.0	43.18	999.0	0.0	999.0
3	4	8	87.63	999.0	41.91	999.0	0.0	999.0
4	4	8	88.9	999.0	40.64	999.0	0.0	999.0
5	4	8	90.17	999.0	39.37	999.0	0.0	999.0
6	4	8	91.44	999.0	38.1	999.0	0.0	999.0
7	4	8	92.71	999.0	36.83	999.0	0.0	999.0
8	4	8	93.98	999.0	34.29	999.0	0.0	999.0
9	4	8	95.25	999.0	33.02	999.0	0.0	999.0
10	4	8	96.52	999.0	31.75	999.0	0.0	999.0
11	4	8	97.79	999.0	29.21	999.0	0.0	999.0
12	4	8	99.06	999.0	27.94	999.0	0.0	999.0
13	4	8	100.33	999.0	25.4	999.0	0.0	999.0
14	4	8	101.6	999.0	20.32	999.0	0.0	999.0
15	4	8	102.87	999.0	16.51	999.0	0.0	999.0
16	4	8	104.14	999.0	12.7	999.0	0.0	999.0
17	4	8	105.41	999.0	8.65	999.0	0.0	999.0
18	4	8	106.88	999.0	4.84	999.0	0.0	999.0
19	4	8	107.95	999.0	0.0	999.0	0.0	999.0

Figure 1. Example of Edit Window for a TORT geometry model showing the "Bodies" tabbed pane.

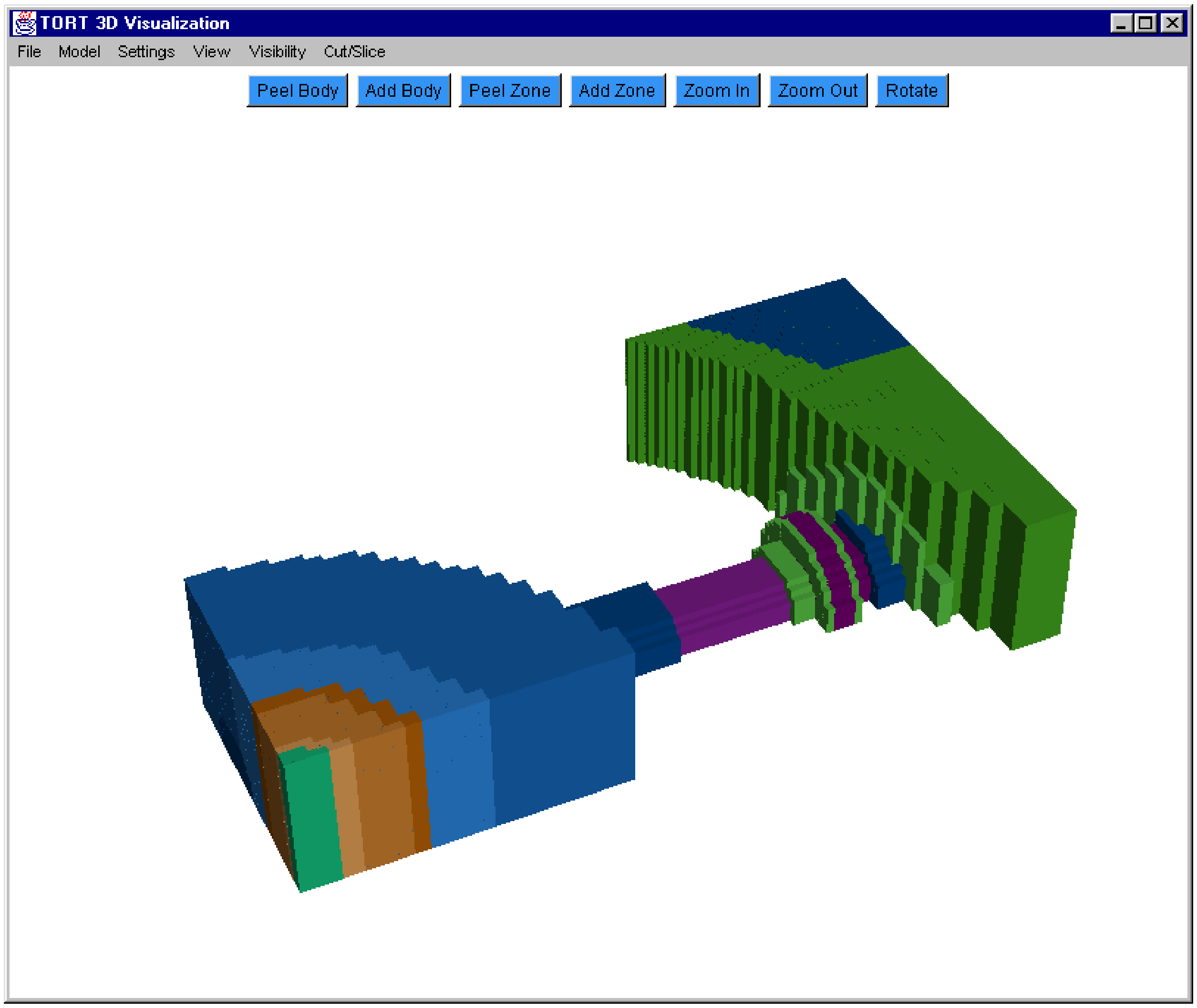
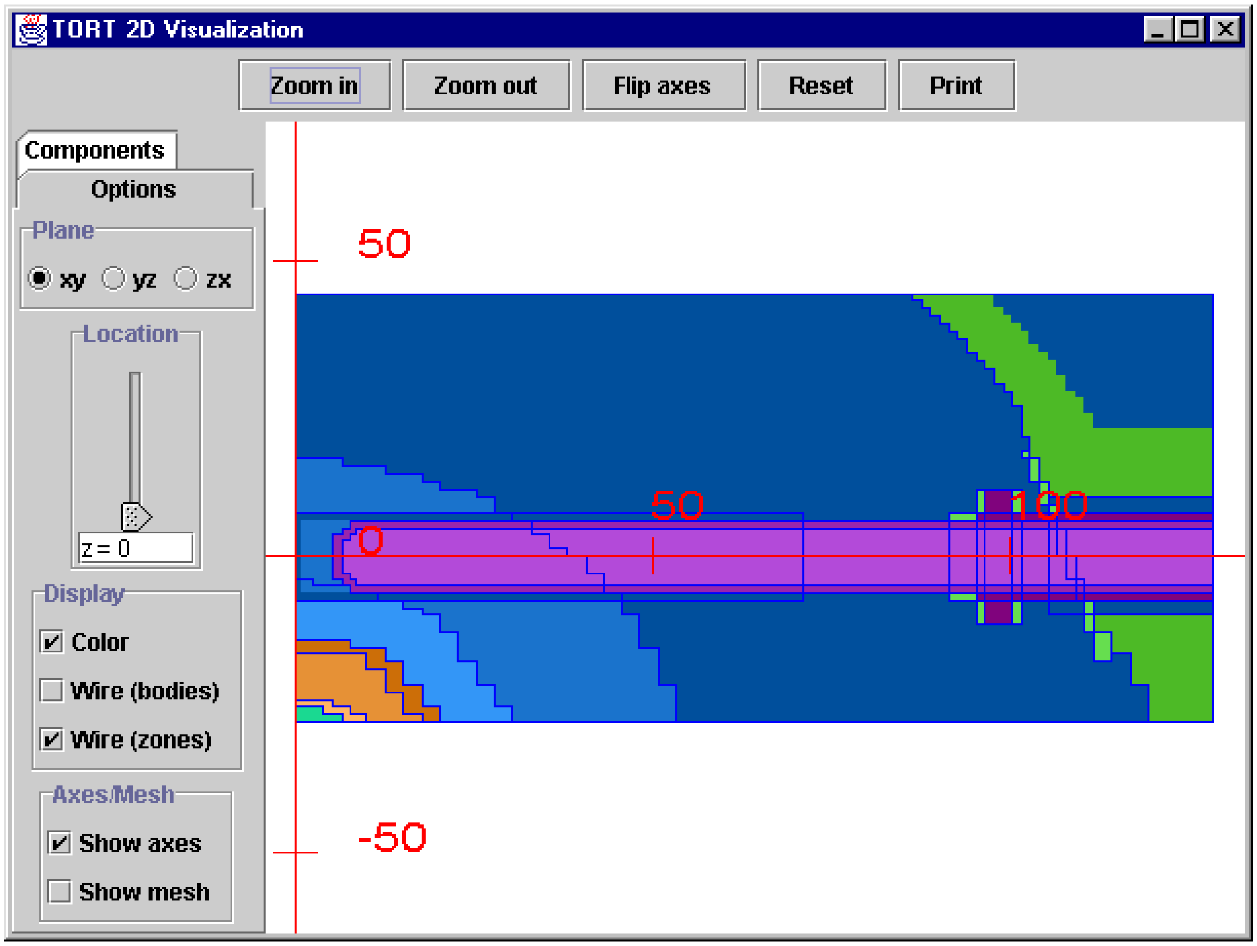


Figure 2. Sample TORT geometry 3D Visualization Window display



● xy ○ yz ○ zx

z = 0

[✓] Color
[ ] Wire (bodies)
[✓] Wire (zones)

[✓] Show axes
[ ] Show mesh

Figure 3. Sample TORT geometry 2D Visualization Window display