ORNL/TM-12019

ornl

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

CGVIEW–A Program to Generate Isometric and Perspective Views of Combinatorial Geometries

T. J. Burns

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-12019

Engineering Physics and Mathematics Division

CGVIEW - A Program to Generate Isometric and Perspective Views of Combinatorial Geometries

T. J. Burns

DATE PUBLISHED: JULY 1992

NOTICE This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
managed by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE—AC05—84OR21400

OAK RIDGE NATIONAL LABORATORY
MANAGED BY MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE U.S. DEPARTMENT OF ENERGY

POST OFFICE BOX 2008
OAK RIDGE, TENNESSEE 37831

August 17, 1992

To: Recipients of Technical Memorandum

Report No: ORNL/TM-12019

Classification: Unclassified

Author: T. J. Burns

Due to errors found in the above report we are reissuing the report. Please replace your copy with the new version of Report ORNL/TM-12019.

David Hamrin
Technical Information Officer
Information Services

CONTENTS

LIST OF FIGURES

Figure Page

ABSTRACT

The prototype of a graphical debugger for combinatorial geometry (CG) is described. The prototype debugger consists of two parts: a FORTRAN-based "view" generator and a Microsoft Windows application for displaying the geometry. This document describes the code CGVIEW, which comprises the first part of the system. User-specified options permit the selection of an arbitrary viewpoint in space and the generation of either an isometric or perspective view. Additionally, any combination of zones, materials, or regions can be flagged as invisible to facilitate the inspection of internal details of the geometry. In the same manner, an arbitrary body can be cut away from the geometry to facilitate inspection and debugging. Examples illustrating the various options are described.

1. INTRODUCTION

A recurring difficulty in shielding and radiation transport applications is the verification that the geometric model (and associated material assignments) accurately represent the physical configuration. Exacerbating this difficulty is the fact that for some problems, the number of objects (zones, regions, bodies etc.) used to describe the physical configuration can easily number in the hundreds or thousands. Additionally, the geometric descriptions in the stochastic codes are complicated by the Boolean operators AND, OR, and NOT used to combine the fundamental objects. Both these circumstances largely preclude the successful verification and debugging of the geometric models directly from the input stream of the radiation transport codes. A clear need exists for a means of analyzing and debugging geometric models *before* consuming hours of CPU time on a supercomputer.

A related problem is that the resources expended in "setting up" the problem far outweigh the resources expended in executing the various shielding and transport codes in may cases. Currently, the geometric models required by shielding and transport codes such as MASH-GIFT[1], MORSE[2], and TORT[3] are laboriously translated from blueprints or other descriptions into combinatorial geometries by specifying both the fundamental objects (spheres, cylinders, cones, etc.) as well as the Boolean operations required to combine the objects into meaningful assemblies. For even simplified models, this can be a monumental task.

Two principal improvements would greatly facilitate this model construction process. First, it would be very useful to import the geometry and material parameters for assemblies or components directly from the CAD (Computer-Aided Design) applications. Second, it would be advantageous for the analyst to be able to "see" the geometric model as the algorithms inherent in the transport codes do, during the construction and verification process. Additionally, the potential for graphically positioning of the assemblies relative to each other, as well as the capability of specifying interstitial materials (i.e. void, air, etc.) as appropriate would also be of value.

With a view to such a system as the ultimate goal, a working prototype (with limited capabilities) has been developed at ORNL to serve as a testbed for further development. The system is being developed to ultimately accept geometric data from IGES (Interim Graphics Exchange Specification)[4] files, a format compatible with a number of CAD applications. More pertinent to this discussion however is the current capability of the system to utilize files formatted for the MASH-GIFT, MORSE, and TORT radiation transport codes.

To facilitate initial development and investigation, the prototype has been written as two modules, exchanging data via a metafile. CGVIEW is the first half of a two part graphical display and debugging code for combinatorial geometry. It is designed as a semi-interactive program whose purpose is to generate an intermediate data file containing a particular "view" of the combinatorial geometry. It is designed to operate from a variety of input formats and permit the user a wide selection of the aspects of the combinatorial geometry to be included in the "view".

The second part of the display and debugging code consists of a Microsoft Windows application, ORGBUG, which accepts the metafiles produced by CGVIEW and which is documented in a separate report[4]. The choice of Microsoft Windows as the graphical user interface (GUI) for the prototype debugger was made primarily as a function of convenience and expediency. The use of Microsoft Windows tends to minimize hardware dependencies since it supports a variety of display and output devices. Additionally, the basic design philosophy of Windows and Xwindows is similar (they are both event-driven), which will simplify the eventual porting of the system to the IBM RISC/6000 workstation.

ORGBUG is structured to read the metafiles generated by CGVIEW, and display the image contained in the file. As a Windows application it permits (via scroll bars) the image to be larger (in terms of resolution) than the window itself. As a consequence of the decision to split the debugger into two modules, however, the image displayed is a fixed one, i.e. no rotation or zoom functions are implemented. Either a monochrome (wireframe) or color image can be selected. If a color representation is selected, "colorization" can be done on a zone, material, or region basis using either a default palette or user-assigned colors.

User-specified options in CGVIEW permit the selection of an arbitrary viewpoint in space and the generation of either an isometric or perspective view. Additionally, any combination of zones, materials, or regions can be flagged as invisible to facilitate the inspection of internal details of the geometry. In the same vein, an arbitrary body can be cut away from the geometry to facilitate inspection and debugging.

While the prototype program documented in this report currently implements only a few of the functions envisioned for the ultimate system, it has already been proven to be extremely useful. As such, it permits a level of verification and debugging for complex CG models which was not previously available. Because of its utility, an interim and admittedly incomplete version of this program is being made available. It should be noted that continued support and/or extensions of the program are subject to funding priorities.

2. CGVIEW USER-SPECIFIED OPTIONS

CGVIEW is designed to accept user input interactively and then execute on an IBM-compatible personal computer. It prompts the user for required or optional information and then constructs a metafile containing the desired "view" of the geometric model. The required (and optional) information generally consists of two types, options related to the combinatorial geometry itself and those describing the position and orientation of the viewpoint.

2.1 GEOMETRY SELECTION OPTIONS

The initial prompt of the CGVIEW code is used to determine the type of geometry to be processed.

The following Input formats are supported:

1 MORSE Geometry Package
2 MORSE Input Deck
3 TORT Input Deck
4 GIFT4 Geometry Package
5 GIFT5 Geometry Package
6 IGES (Version 4) File

Enter the Format Option [1]: ?

Options 1 and 6 are currently disabled since the coding for each is still under development. The data formats required for options 2-5 are documented in references 1-3, and will not be detailed here. However, the notation "geometry package" includes all input from the title card through the material and region specifications for MORSE and GIFT4/GIFT5. By way of contrast, option 3 is designed to read an entire TORT input deck and extract the relevant geometric data. The response to this prompt is also utilized to eliminate certain materials on a global basis from the generated view: materials 0 and 1000 [external and internal voids] for MORSE-based geometry and material 0 [typically air] from GIFT4- and GIFT5-based geometries. No material assumptions are made for TORT-based geometries. A distinction is drawn between GIFT4 and GIFT5 since GIFT5 is not backward compatible with GIFT4. In particular, the parameters for bodies ELL (Ellipsoid of revolution) and TEC (Truncated Elliptical Cone) are interpreted differently by the two versions.

Once a particular input format is selected, the code prompts for the file name of the input file

Geometry File: ?

The appropriate file name (including a drive and path specification if the file is not in the current directory) should be entered in response to this prompt. Once, a valid file specification has been entered, the code requests the degree of editing desired for the input data.

The following Print options are available:

+1 Print Body Data
+2 Print Material/Region Table
+4 Print Zone Boolean Data
+8 Print Code Zone Table
+16 Print Input Zone Table
+32 Print Equiv. RPPs

Enter the Desired Print Option [0]: ?

Note that the options are actually bit flags so that the appropriate response if all the edit options are to be activated is 63 (i.e. 32+16+8+4+2+1). If only the material/region table and the input zone table are desired, the appropriate response is 18 (16 + 2). Output is currently directed to the standard output unit (usually the screen).

The next prompt concerns the degree of aggregation desired for the current view.

The following Objects can be plotted:
[Negative to defeat autoscaling]

1 Plot a Single Body
2 Plot a Single Code Zone
3 Plot a Single Input Zone
4 Plot all zones in a Region
5 Plot all zones containing a medium
6 Exclude AUX File Regions and Materials
7 Plot entire geometry

Enter your choice [7]: ?

The distinction between on input zone and a code zone is identical to that of MORSE. The code is designed to scale the object or objects to be plotted so as to fill the plot frame as far as practicable (autoscaling). However, by responding to this prompt with a negative, the object or objects will be scaled in proportion to the overall geometry. Thus, this represents a way to visualize the position of specific objects in relation to the overall geometry. Selecting options 1-5 will result in one of the following prompts.

Enter the Body number [1]:

Enter the Code Zone number [1]:

Enter the Input Zone number [1]:

Enter the Region number [1]:

Enter the Media number [1]:

A negative entry for any prompt except the body number will activate a NOT operation, i.e., entering -3 as the media number will result in the code plotting all zones which do *NOT* contain media 3. This is particularly useful for certain format options such as TORT in which a medium such as air is to be eliminated on a global basis. Next, the user is prompted for the name of an auxiliary input file.

Auxiliary File: ?

The purpose of this file is to provided detailed information as to the shape of objects to be "cut away" from the geometry as well as explicit zones, regions, and materials to be eliminated from the view. If option 6 (above) is selected, specification of a valid file specification is mandatory; otherwise, a <CR> or <Enter> will bypass this option. If a file specification is entered and a non-zero number of bodies are defined by the auxiliary file, the following prompt appears:

Cutaway Options:

0 None
N Cut away Auxiliary Body N

Enter your choice [0]:

allowing the user to select the particular body to be "cut away" from the geometry. The required format for the auxilary file is described in Appendix A. At this point, the particular features to be included in the view are defined and the code proceeds to request options regarding the particular view to be produced. The code next calculates the size of sphere which will enclose the selected geometry features. The viewpoint is assumed to be a point on a sphere, whose center is at the centroid of the selected portions of the geometry, and whose radius is twice that of the enclosing sphere.

2.2 View Selection Options

The remainder of the prompts govern the particular view to be generated by the code. Two types of projections are available.

The following Projections are available:

1 Isometric
2 Perspective

Enter your choice [1]:

In the isometric option, parallel lines appear to be parallel, while for the perspective option they converge at the respective vanishing points.

The next prompt requests the position of the viewpoint on the enclosing sphere.

View Directions [1.0 -1.0 1.0]: ?

The required numbers are components (x,y,z) of a vector indicating the position of the viewpoint relative to the centroid of the geometry in arbitrary units. For example, the default values define the viewpoint to be equal distances along the +x, -y, and +z axes - which normally will generate a view of the front, left side, and top of the geometry.

The response to the next prompt determines the shape of each pixel in the view.

Aspect Ratio Parameters:

Enter your choices [1, 1]: ?

The order is (horizontal, vertical) and the units are arbitrary. For a standard VGA screen (640 x 480) the pixels are square and the defaults are appropriate. For other video modes however, appropriate adjustments may be required to prevent distortion when viewed.

The next prompt requests the type of adjustment to be done in order to display the geometry.

Adjustment Option:

1 Adjust Vertical
2 Adjust Horizontal
3 Adjust as necessary to fit

Enter your choice [1]: ?

The response to this prompt will govern how the code adjusts the number of pixels in each direction to encompass the desired features of the geometry. Depending on the adjustment option selection, one or both of the following prompts will appear.

Horizontal Resolution [nh]: ?

Vertical Resolution [nv]: ?

The response to this prompt governs the size of the view in pixels. It should be noted that the code is currently limited to 2400 pixels horizontally. The vertical size is essentially unlimited. The next prompt is included to increase the speed of the view generation process. It represents the number of pixels that constitute a "macro" pixel.

Pixel Scanning Parameter [8]: ?

Valid responses are 1,2,4,and 8. The code examines the geometry on each corner of a "macro" pixel. If the zone, body, and surface for each corner are identical, the code assumes that all pixels within the "macro" pixel are the same. Otherwise, it subdivides the "macro" pixel into 4 pixels, and repeats the process recursively. This technique radically reduces the number of pixels which must actually be determined based on the geometry. For example, a full screen view in VGA mode (640 x 480) consists of 307,200 pixels. Using a pixel scanning parameter of 8 typically requires the calculation of approximately 10% of these. It should be noted, however, that using a pixel scanning parameter > 1 has the potential for obscuring details smaller than roughly half the pixel scanning parameter.

The last prompt exists to account for unusual situations.

The following Orientations are Available:

1 Portrait
2 Landscape

Enter your choice [1]:

As noted previously, the routines which generate the view are limited to 2400 pixels in the horizontal direction. No corresponding limit exists for the number of vertical pixels. In the rare instance where the number of horizontal pixels is greater than 2400 *AND* the number of vertical pixels is less than 2400 pixels (i.e. a very wide but short geometry) the landscape option will rotate the view ninety degrees.

At this point, the code will begin execution. It provides a continuous readout of its progress.

Calculating Col,Row #

Upon completion, the following information will be displayed.

STATISTICS npix,nmax,npix/nmax

STARTED AT hh.mm.ss

STOPPED AT hh.mm.ss

where npix is the number of pixels actually calculated, nmax is the total number of pixels required, and npix/nmax is the % of required actually done.

Output from the code consists of a file, CGVIEW.VUE, which is placed in the current directory. It should be noted that the code will overwrite any existing file with the same file specification, and hence previous files to be retained should be renamed.

3. ILLUSTRATIONS

The best way to depict the utility of the geometry debugger is to show the output produced. Views produced using three different combinatorial models are discussed in this section. As noted above, the system can produce color output. This is particularly useful if the various colors are mapped to the model materials. However, due to publication limitations, the various figures used in this section were generated using the wireframe option (black wires on a white background).

To illustrate some of the capabilities of the graphical debugging system, Figures 1a-1d were generated based on a geometric model formatted for the MORSE Monte Carlo radiation transport code. The geometric model itself was originally created as part of a shielding and neutronic analysis for the TFTR (Tokamak Fusion Test Reactor)[6]. Figures 1a-1d were generated by selecting only those objects which were assigned the same material identification, corresponding to the toroidal field coils, the poloidal field coils, the torus itself, and the support structure. Although the default scaling option of the geometry debugger is to scale the selected objects to fill the viewing frame, this particular option was disabled so that the four figures would be scaled identically. This allows the relative positions and sizes of the various objects to be maintained across multiple frames. It should be noted that, at the time this model was created, MORSE did not have a toroidal body. Hence, the torus depicted in Figure 1c is modeled as a set of short cylindrical annuli ORed together. Note that, in this particular model, one of the annuli is missing.

Figure 2 represents an isometric view of the entire geometry of the TFTR, i.e. the aggregation of the pieces depicted in Figures 1a-1d. Additionally, the view was generated by specifying that a 90 degree wedge was to be "cut away" from the geometry. Use of this option is one way of maintaining a perspective relative to the entire geometry, while simultaneously permitting the analyst to examine internal structures of relatively complex groups of objects.

Although discrete ordinates codes typically do not employ combinatorial geometry, the fact that TORT uses RPPs (i.e. a MORSE rectangular parallelepiped) to define regions, and then overlays those regions onto a space mesh allows the geometry debugger to be employed. The overlay scheme utilized implies a set of Boolean operations which the geometry debugger constructs. Figure 3 illustrates the use of the geometry debugger based on a TORT input deck. The model is that of the Chinzei school, which was part of a previous radiation transport study[7]. Figure 3 depicts a perspective view of the building with the right front quarter removed in order to expose the interior structures such as floors, walls, and ceilings.

By judicious selection of options, certain extremely useful graphical descriptions of a particular model can be produced. For example, Figure 4 is the result of requesting an isometric view of the Chinzei school model, setting the

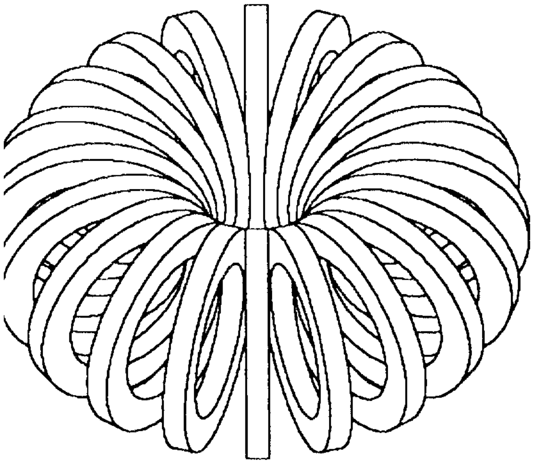

Fig. 1a. TFTR Torodial Field Coils.

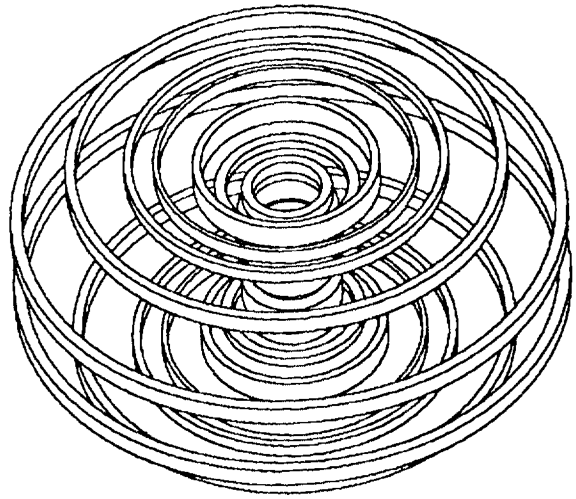

Fig. 1b. TFTR Poloidal Field Coils.

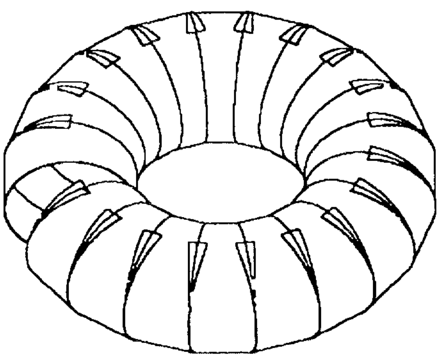

Fig. 1c. TFTR Torus.

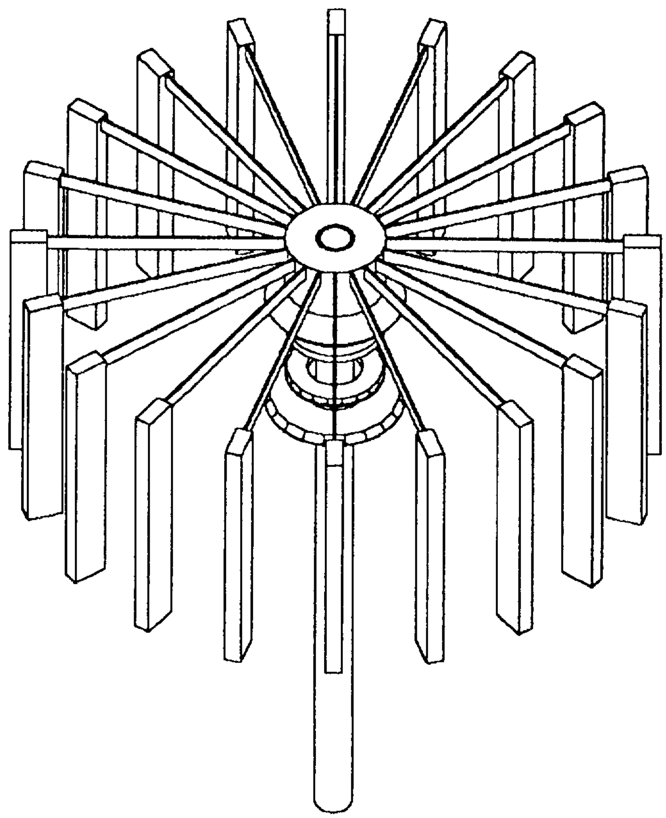

Fig. 1d. TFTR Support Structure.

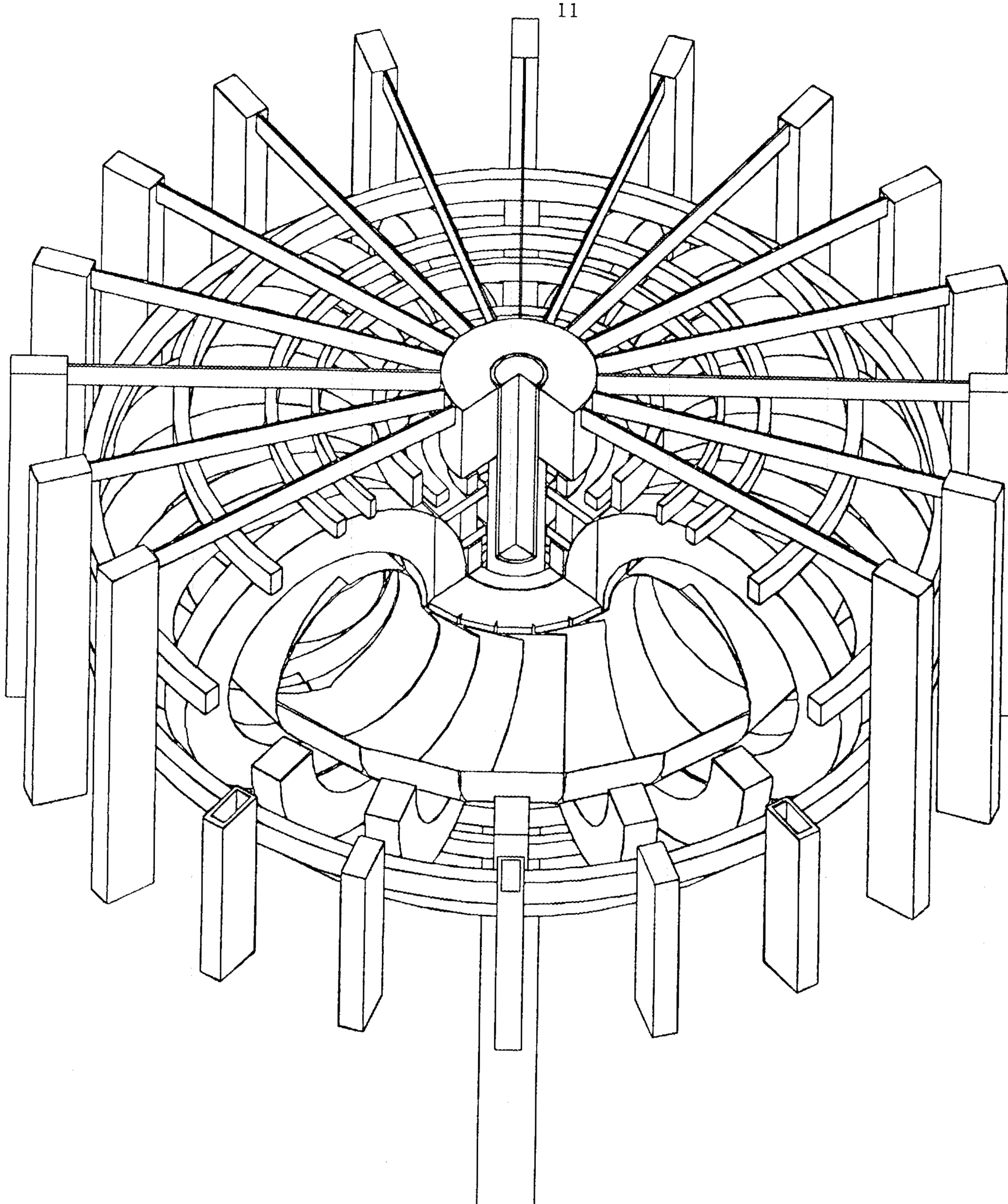

Fig. 2. TFTR Model with Cutaway.

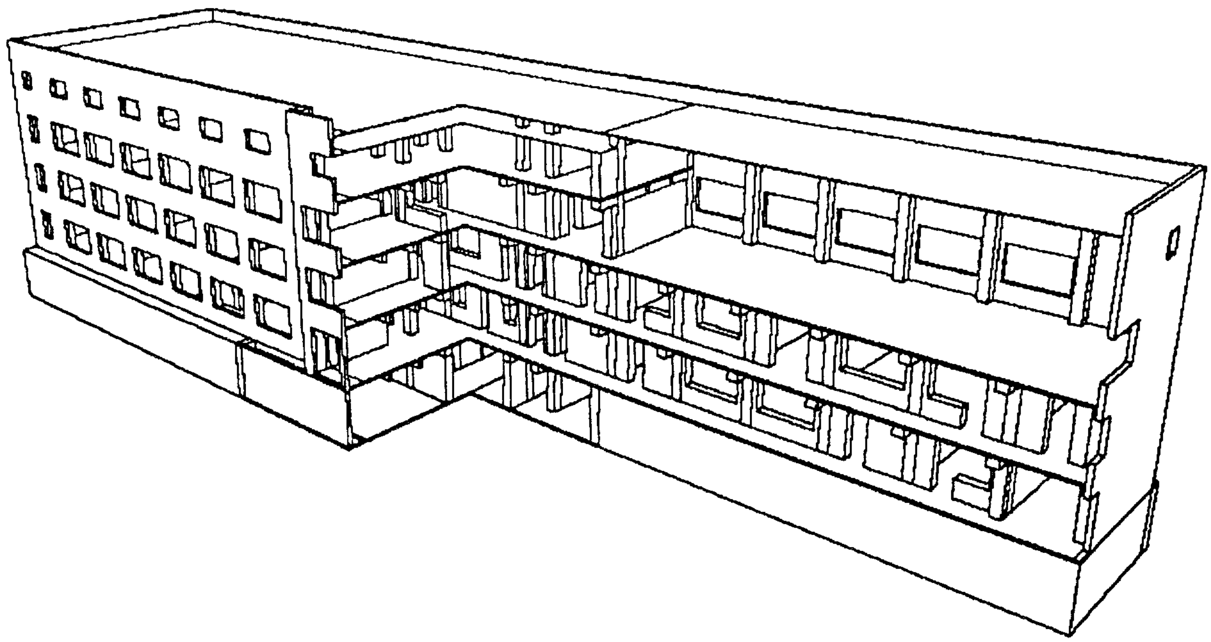

Fig. 3. Chinzei School Building with Cutaway.

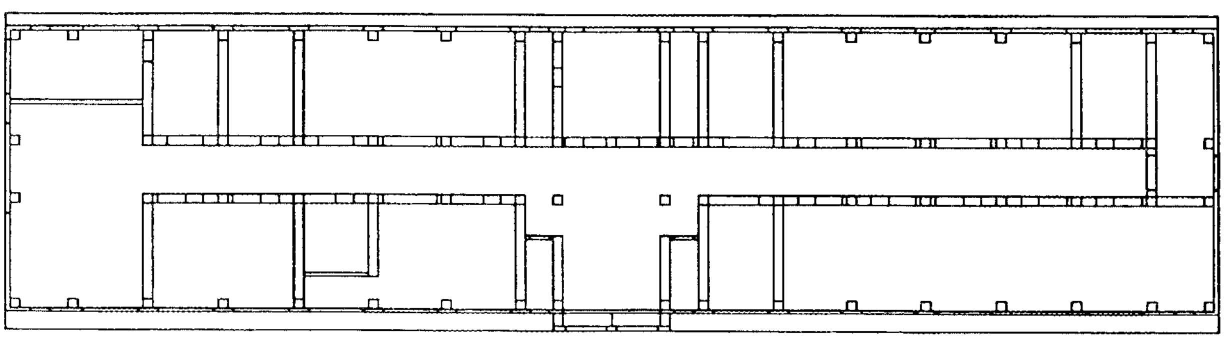

Fig. 4. Chinzei School Building - First Floor Plan.

viewpoint to be directly overhead, and cutting away a half space. The result is a depiction of the first floor plan of the building.

A second mechanism for displaying the internal details of a complex set of objects is to use the selective invisibility feature of the geometry debugger. Figure 5 depicts an isometric of a Soviet BMP (armored personnel carrier)[8]. The model is described in GIFT4 geometry for use with MASH code. By selectively flagging specific zones regions, and materials as invisible, Figure 5 can be converted to Figure 6, i.e. the gun turret and top armor can be removed to show the internal details of the model. Using the selective invisibility feature for multiple views permits views such as Figure 7 to be constructed. The invisibility flags utilized in Figure 6 were reversed for a second view using the same viewpoint and scaling parameters. The second view was then overlaid (and displaced slightly) on Figure 6. The result is an "exploded" view of the Soviet BMP.

Although the use of color greatly facilitates the debugging of combinatorial geometries (e.g. by mapping materials to colors), the wireframe output can be very useful in locating errors or oversights in the geometric model. As an example, consider the BMP model depicted in Figure 8. Even in wireframe view, it is clear that a crucial piece of armor has been omitted from the model. This figure dramatically illustrates the power of a graphical representation of the combinatorial geometry. The model consists of 658 bodies combined into 449 zones. As such, the determination that a single piece of armor had been omitted from the input data would have been a formidable task. Similarly, Figure 9 illustrates another view of the BMP. In this particular view, there are two errors present but the wireframe depiction is not particularly helpful in locating them. However, when this figure is redrawn in color (mapped by materials) the errors become immediately obvious.

Although the initial purpose of the geometry debugger was to permit the analyst to validate and correct the *input* for the radiation transport codes, it readily became apparent that the images produced could also play a significant role in displaying the output of the radiation transport codes as well. Both the discrete ordinates codes (TORT) and the stochastic codes (MORSE and MASH) can produce significant quantities of output data. The images which can be generated via the geometry debugger can provide a mechanism for visualizing what can be an enormous amount of data. For example, combining views such as Figure 4 with existing output manipulation codes like the DOGS[9] system would permit the overlay of flux contours directly on the floor plan of the building. Moreover, it should also be possible to map color coded response values on views such as Figure 3, in effect, mapping the response values onto the exterior of the building as well as onto the floors, walls, and ceilings of model. In a similar vein, the output of MASH (typically a leakage flux as a function of energy and position) could be color coded and mapped onto an image such as Figure 5 to display the geometry vulnerability to radiation effects.

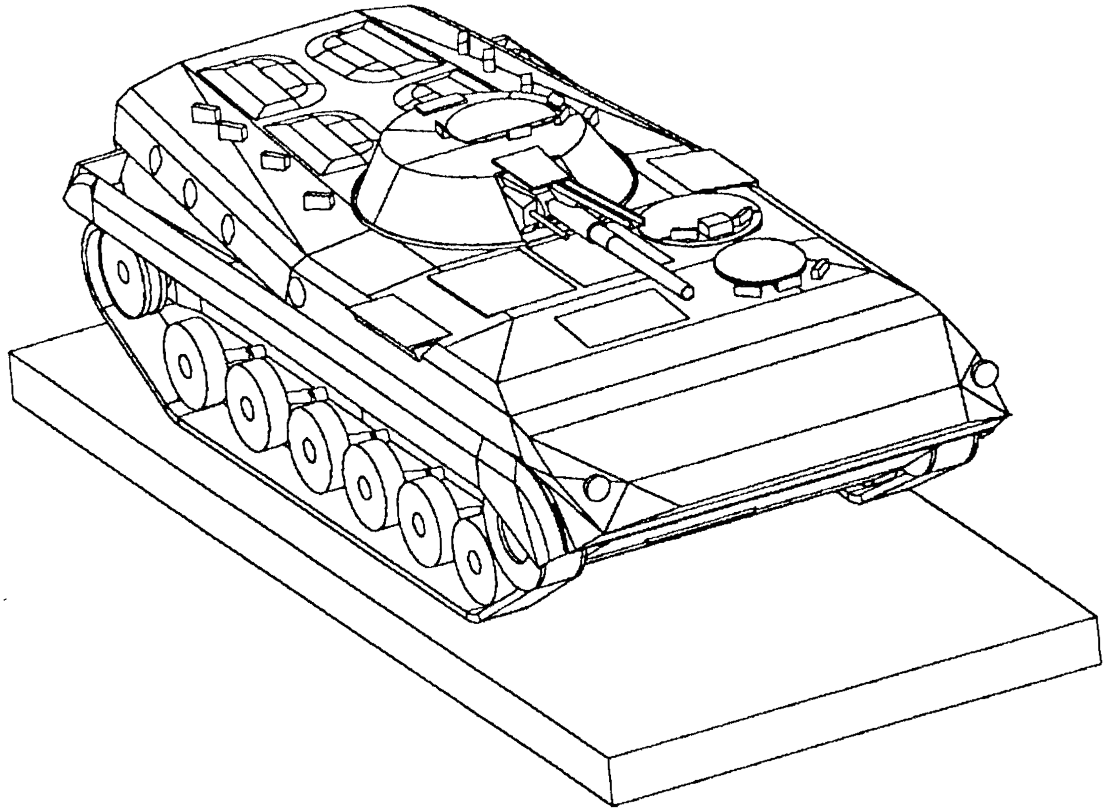

Fig. 5. Soviet BMP Model.

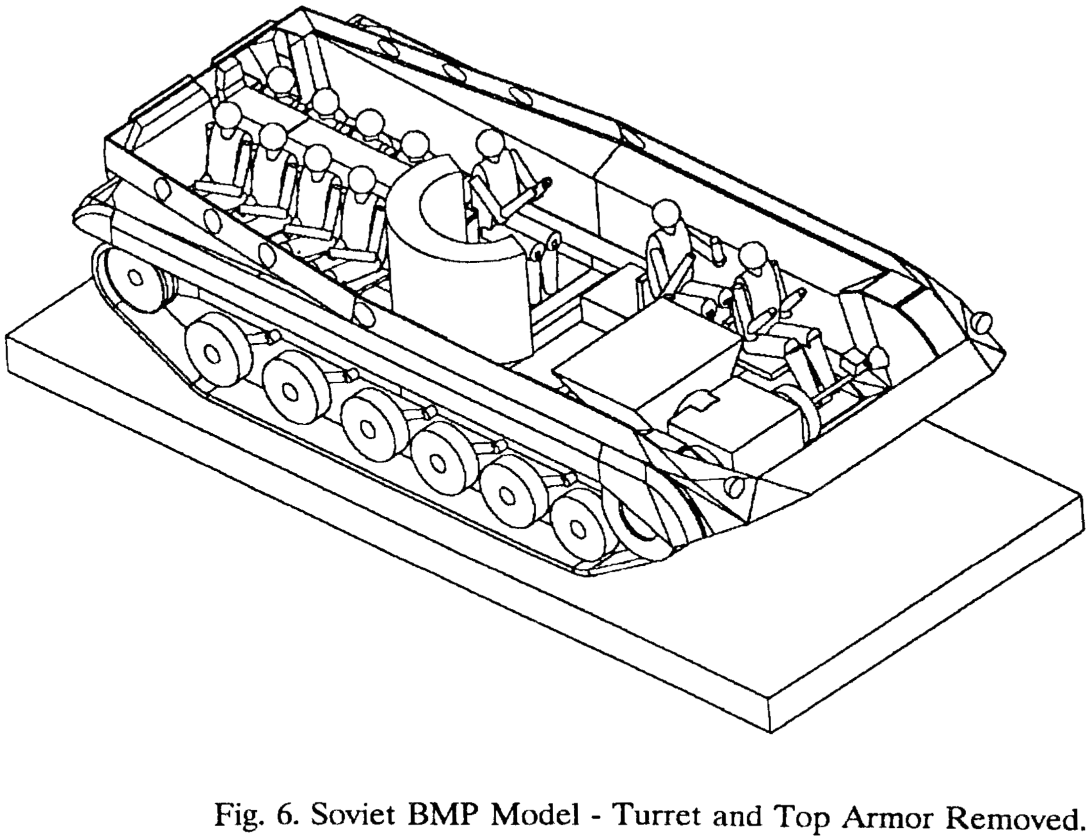

Fig. 6. Soviet BMP Model - Turret and Top Armor Removed.

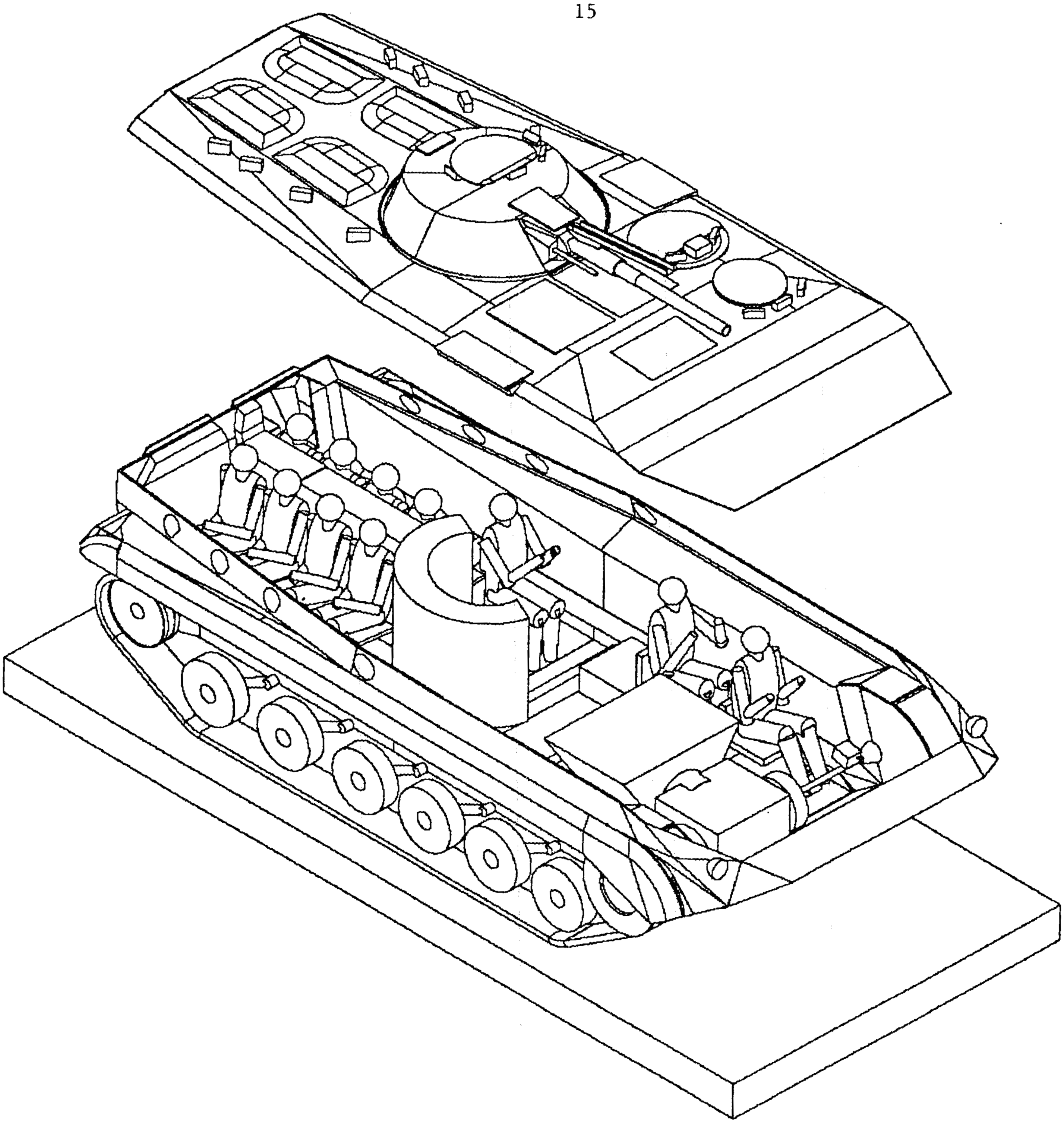

Fig. 7. Soviet BMP Model - Exploded View.

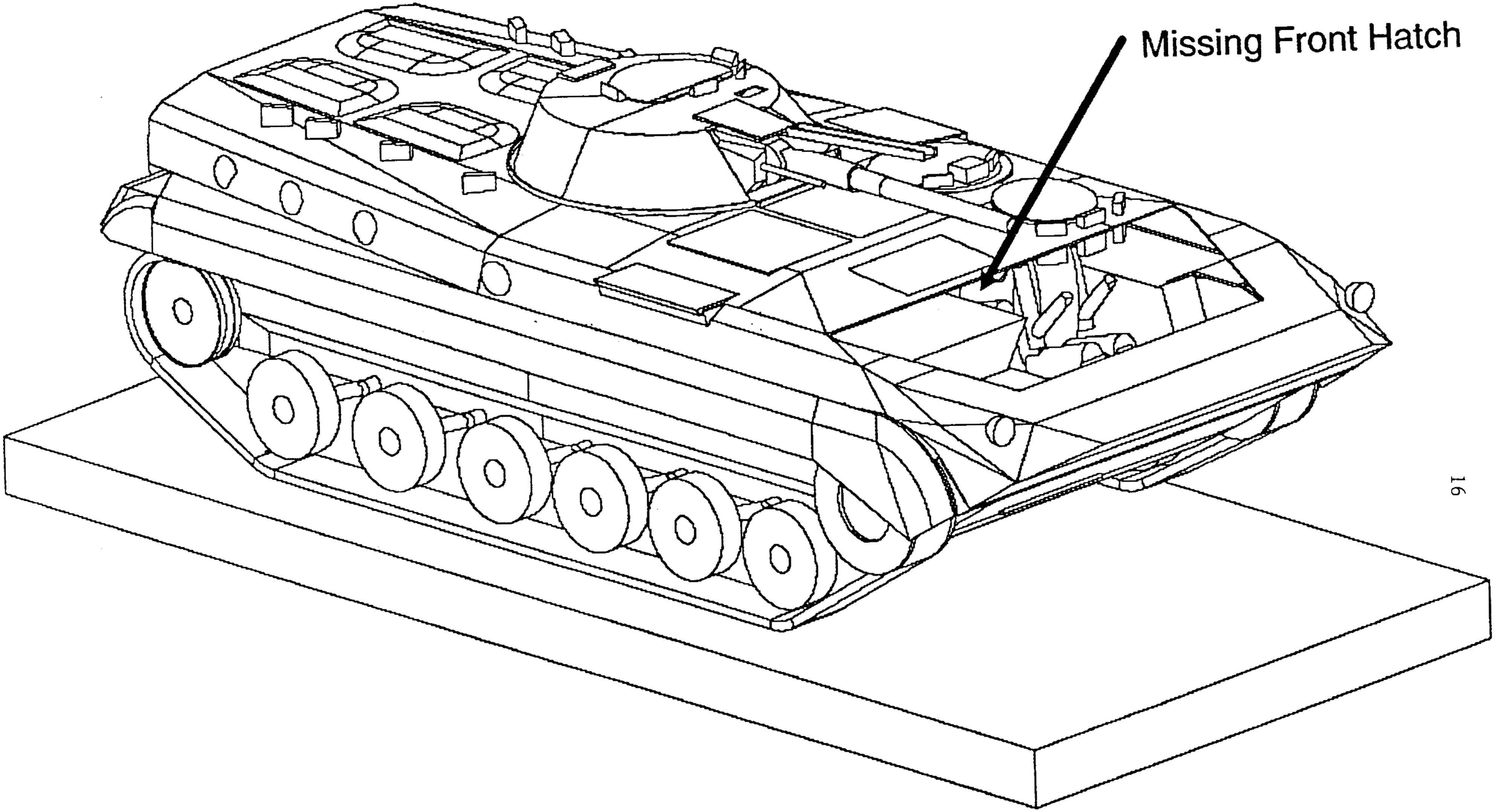

Fig. 8. Soviet BMP Model with Missing Front Hatch.

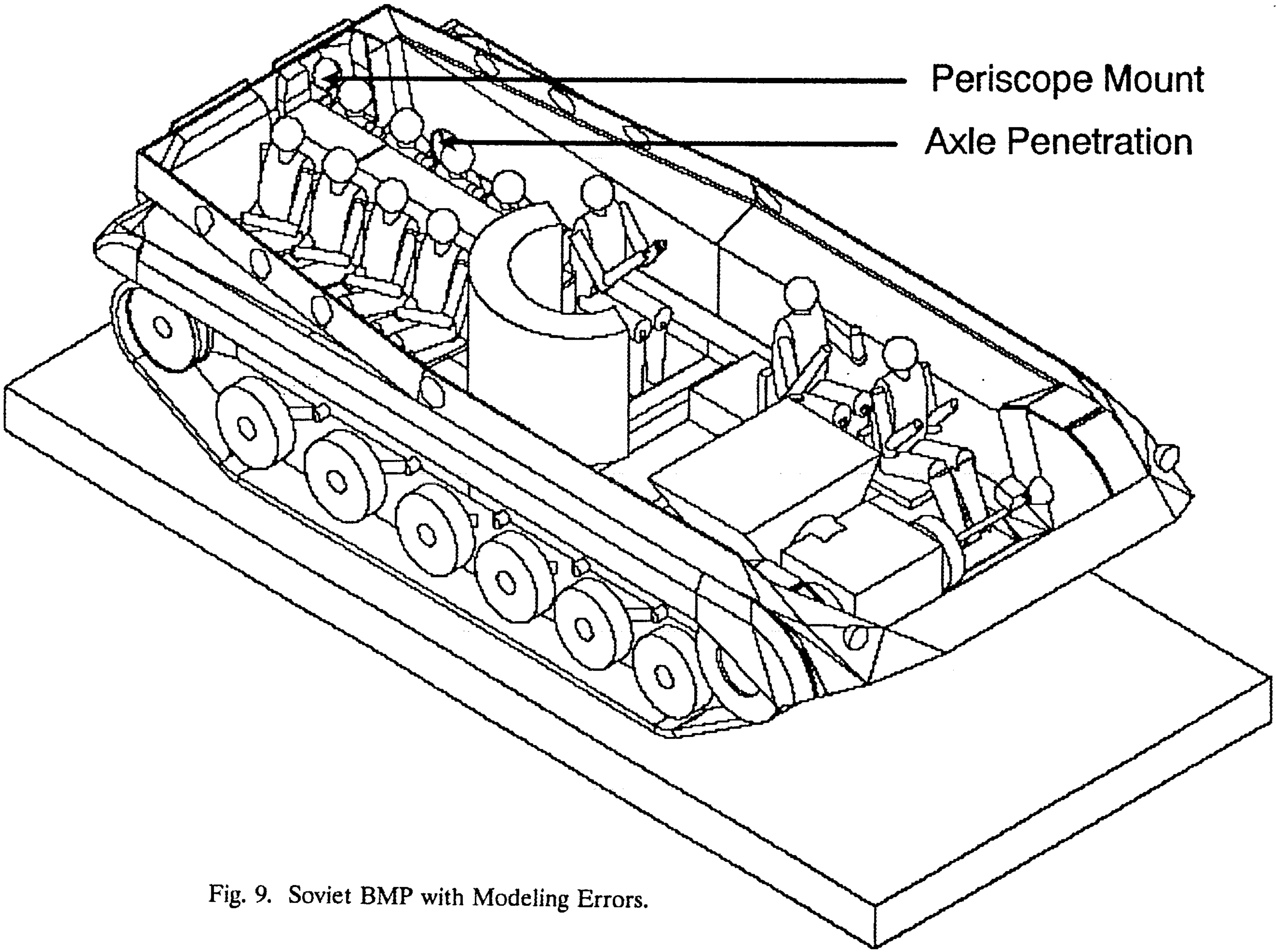

Fig. 9. Soviet BMP with Modeling Errors.

REFERENCES

1. J. O. Johnson et al., "A User's Manual for MASH 1.0 - A Monte Carlo Adjoint Shielding Code System," ORNL/TM-11778, Oak Ridge National Laboratory, (March 1992).

2. M. B. Emmett, "The MORSE Monte Carlo Radiation Transport Code System," ORNL-4972 (1975), ORNL-4972/R1 (1983), ORNL-4972/R2 (1984), Oak Ridge National Laboratory.

3. W. A. Rhoades and R. L. Childs, "The TORT Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code," ORNL/TM-6268, Oak Ridge National Laboratory, (November 1987).

4. Bradford Smith et al., "Initial Graphics Exchange Specification (IGES), Version 4.0," SP-767, Society of Automotive Engineers,(June 1988).

5. T. J. Burns, "ORGBUG - A Windows-based Combinatorial Geometry Debugger", ORNL/TM-12020, Oak Ridge National Laboratory, (To be published).

6. R. T. Santoro et al., "Comparisons of Calculated and Measured Spectral Distributions From a 14-MeV Neutron Source Inside the Tokamak Fusion Test Reactor, ORNL/TM- 9888, Oak Ridge National Laboratory, (December 1985).

7. W. A. Rhoades, R. L. Childs and D. T. Ingersol, "Radiation Exposure Inside Concrete Buildings at Nagasaki", ORNL /TM-10569, Oak Ridge National Laboratory, (May 1989).

8. C. M. Ward, personal communication, Foreign Science and Technology Center, November 15,1989.

9. D. T. Ingersol and C. O. Slater, "DOGS - A Collection of Graphics for Support of Discrete Ordinates Codes", ORNL /TM-7188, Oak Ridge National Laboratory, (March 1980).

INTERNAL DISTRIBUTION

1. B. R. Appleton
2. J. M. Barnes
3-7. T. J. Burns
8. J. D. Drischler
9. K. F. Eckerman
10. M. B. Emmett
11. C. M. Haaland
12. D. T. Ingersoll
13. J. O. Johnson
14. B. L. Kirk
15. J. V. Pace
16. R. T. Primm
17. W. A. Rhoades
18. R. W. Roussin
19. R. T. Santoro
20. R. C. Ward
21-25. EPMD Reports Office
26-27. Laboratory Records Department
28. Laboratory Records ORNL-RC
29. Document Reference Section
30. Central Research Library
31. ORNL Patent Section

EXTERNAL DISTRIBUTION

32. Dr. Roger W. Brockett, Wang Professor of Electrical Engineering and Computer Science, Division of Applied Science, Harvard University, Cambridge, MA 02138

33. Electronics Division, NES, ATTN: Dr. Tom Cousins, Defence Research Establishment Ottawa, Ottawa, Ontario, Canada, K1A OZ4

34. Commander, U. S. Army Nuclear & Chemical Agency, ATTN: MONA-ZB (Dr.Charles Davidson), 7500 Backlick Rd., Bldg. 2073, Springfield, VA 22150-3198

35. Establissment Technique Central de l' Armement, ATTN: Dr. Joel Dhermain, Centre d'Etudes du Boucher, 16 bis Avenue Prieur de la Cote d'or, 94114 Arceueil-Cedex, France

36. Dr. John J. Dorning, Department of Nuclear Engineering & Engineering Physics, Thorton Hall, McCormick Road, University of Virginia, Charlottesville, VA 22901

37. Science Applications International Corporation, ATTN: Dr. Stephen Egbert, 10260 Campus Point Drive, San Diego, CA 92121

38. Commander, U. S. Army Combat Systems Test Activity, ATTN: STECS-NE (Mr. John Gerdes), Aberdeen Proving Ground, MD 21005-5059

39. Ministry of Defence, Atomic Weapons Establishment, ATTN: Dr. Kevin G. Harrison, Building A72, Aldermaston, Reading, Berkshire, United Kingdom, RG7 4PR

40. Commander, U. S. Army Combat Systems Test Activity, ATTN: STECS-NE (Dr. C. Heimbach), Aberdeen Proving Ground, MD 21005-5059

41. Dr. P. B. Hemming, Safety and Physics Branch, Office of Technology Support Programs, Department of Energy, Washington, D.C 20545

42. Science Applications International Corporation, ATTN: Mr. Dean C. Kaul, 10260 Campus Point Drive, San Diego, CA 92121

43. Director, Armed Forces Radiobiology Research Institute, ATTN: MRAD (CDR Kearsly), Bethesda, MD 20814-5145

44-46. Director, Defense Nuclear Agency, ATTN: RARP (MAJ Robert Kehlet), 6801 Telegraph Road, Alexandria, VA 22310-3398

47. Director, Harry Diamond Laboratory, ATTN: SLCHD-NW-P (Mr. Klaus Kerris), 2800 Powder Mill Road, Adelphi, MD 20783-1197

48. Establissment Technique Central de l' Armement, ATTN: Dr. Jacques Laugier, Centre d'Etudes du Boucher, 16 bis Avenue Prieur de la Cote d'or, 94114 Arcueil-Cedex, France

49. Dr. James E. Leiss, Route 2, Box 142C, Broadway, VA 22815

50. Dr. Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street, Urbana, Il 61801

51. Establissment Technique Central de l' Armement, ATTN: Dr. Guy Nurdin, Centre d'Etudes du Boucher, 16 bis Avenue Prieu de la Cote d'or, 94114 Arcueil-Cedex, France

52. Director, Defense Nuclear Agency, ATTN: RARP (Ms. Joan Ma Pierre), 6801 Telegraph Road, Alexandria, VA 22310-3398

53. Naval Surface Warfare Center, ATTN: CODE R41 (Mr. Gordon Reil), New Hampshire Road, White Oak, MD 20903-5000

54. Commander, U.S. Foreign Science & Technology Center, ATTN: UVA (Dr. Roger Rydin), 220 7th Street NE, Charlottesville, VA 22901-5396

55. HEAD, Nuclear Radiation Effects, ATTN: Dr. Ludwig Schaenzler, Wehrwissenschaftliche Dienststelle, Postfach 1320, 3042 Munster, Federal Republic of Germany

56. Commander, U.S. Army Nuclear & Chemical Agency, ATTN: MONA-NU (MAJ Carl Curling), 7500 Backlick Rd., Bldg. 2073, Springfield, VA 22150-3198

57. Naval Surface Warfare Center, ATTN: CODE R41 (Mr. Tim Temple), New Hampshire Road, White Oak, MD 20903-5000

58. Commander, U.S. Army Foreign Science & Technology Center, ATTN: AIFRTA (Mr. Charles Ward), 220 7th Street NE, Charlottesville, VA 22901-5396

59. Dr. Mary Wheeler, Department of Mathematical Sciences, Rice University P.O. Box 1892, Houston, TX 77204-3476

60. Los Alamos National Laboratory, ATTN: Dr. Paul Whalen, C-DOT MS-B281, Los Alamos, NM 87545

61. Commander, U.S. Army Tank Automotive Command, ATTN: AMSTA-RSK (Mr. Greg Wolfe), Bldg. 200, Warren, MI 48317-5000

62. Science Application International Corporation, ATTN: Dr. William Woolson, 10260 Campus Point Drive, San Diego, CA 92121

63. Office of the Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations , P.O. Box 2001, Oak Ridge, TN 37831

64-73. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37830