Proceedings of Global 2015
September 20-24, 2015 - Paris (France)
Paper 5261

TRIAD, AN INNOVATIVE VISUALIZATION TOOL FOR CRITICALITY SAFETY AND SHIELDING STUDIES

Gilles MARCHAUD, Hervé AUBERT
Independent
FRANCE
Email: triad.physics@orange.fr

Abstract – *Modelling nuclear facility equipment as well as transport and storage casks from criticality-safety and shielding point of view is a hard and time-consuming task for a nuclear analyst. Indeed, such studies involve complex geometries which lead to model a huge number of volumes. Generally, these geometries are modelled using 3D Monte Carlo codes.*

Several software tools have been developed to help create and visualize geometries. Most of them are specialized for a given code and their Graphical User Interface (GUI) reflects this fact. The recent trend is toward unified simulation platforms, where different physical models share the same geometrical model. In this respect, TRIAD is a free, user-friendly visualization software solution. It has been developed since 2006 from the idea that the syntax of one code or another should not impose its specificities on the GUI. Since then, TRIAD has significantly evolved to account for users' feedback.

TRIAD reads text files that describe geometries and material-to-volume assignments with a simple, code-independent syntax. The task of converting models into TRIAD input files is carried out by importation plugins that can be freely implemented by passionate users. Presently, the available plugins can import TRIPOLI 4, MORET 4, KENO V.a and MCNP 5 models.

TRIAD provides numerous powerful capabilities for the nuclear analyst such as checking the validity of a model or displaying a mesh tally.

This paper presents an overview of TRIAD capabilities which can be helpful for criticality-safety and shielding studies.

I. INTRODUCTION

Modelling nuclear facility equipment as well as transport and storage casks from criticality-safety and shielding point of view is a hard and time-consuming task for a nuclear analyst. Indeed, such studies involve complex geometries which lead to model a huge number of volumes. Generally, these geometries are modelled using 3D Monte Carlo codes.

Several software tools have been developed to help create and visualize geometries. Most of them are specialized for a given code and their Graphical User Interface (GUI) reflects this fact. The recent trend is toward unified simulation platforms (such as SALOME[1]), where different physical models share the same geometrical model. In this respect, TRIAD is a free, user-friendly visualization software solution. It has been developed since 2006 from the idea that the syntax of one code or another should not impose its specificities on the GUI. Since then, TRIAD has significantly evolved to account for users' feedback.

This paper presents an overview of TRIAD capabilities which can be helpful for criticality-safety and shielding studies.

II. TRIAD GUI

The main window of TRIAD (see Fig. 1) consists of:

- a menu bar (File, Display, Units, Help);
- two toolbars (lines of buttons used to open a model, modify the 3D display, interact with the model…);
- one or several pages, accessible through a tab, each page being dedicated to one model: a 3D view on the left, tabbed pages on the right (Geometry, Compos., X-sections, Dimens., Tags, Fields, Misc.).

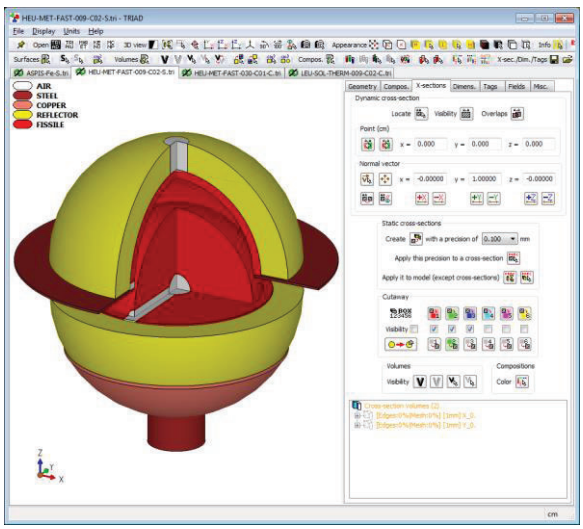

Fig. 1. TRIAD main window.

III. READING MODELS

TRIAD reads text files that describe geometries and material-to-volume assignments with a simple, code-independent syntax. TRIAD can handle surface-bounded volumes as well as combinatory volumes (with Boolean operations) and modular geometries. The task of converting models into TRIAD input files is carried out by importation plugins that can be freely implemented by passionate users. Presently, the available plugins can import TRIPOLI 4[2], MORET 4[3], KENO V.a[4] and MCNP 5[5] models (Fig. 2).

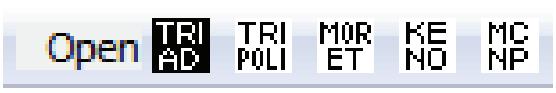

Fig. 2. Toolbar buttons available for opening a model.

TRIAD reads the ".trd" TRIAD input file and calculates the edges and faces of each volume.

IV. DISPLAYING AND EXPLORING VOLUMES

The imported volumes are displayed in 3D with the colors determined by their materials.

The nuclear analyst can then modify the appearance of any volume to better understand the model:

- volumes can be made semi-transparent to show internal volumes (Fig. 3 and 4);
- volumes can be quickly displayed or hidden depending on their materials, by double-clicking on any color pastille of the material legend in the top left corner of the 3D view. It is a straightforward way of detecting errors in material-to-volume assignments;
- colors attributed to materials by the plugins can be changed and selected among more than 300 named colors (Fig. 5).

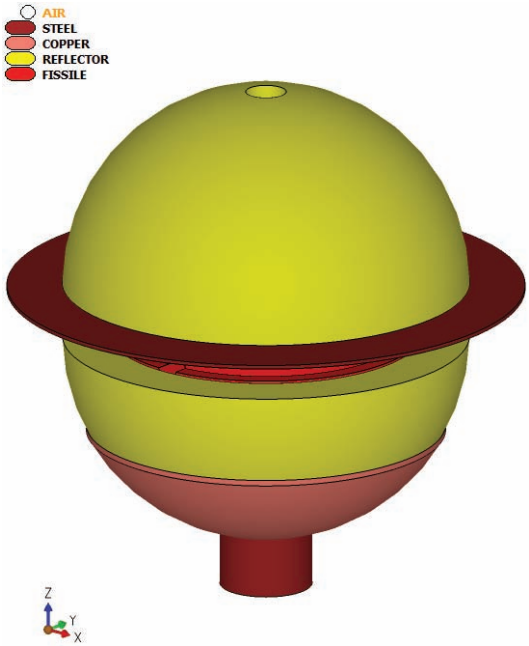

Fig. 3. Opaque volumes.

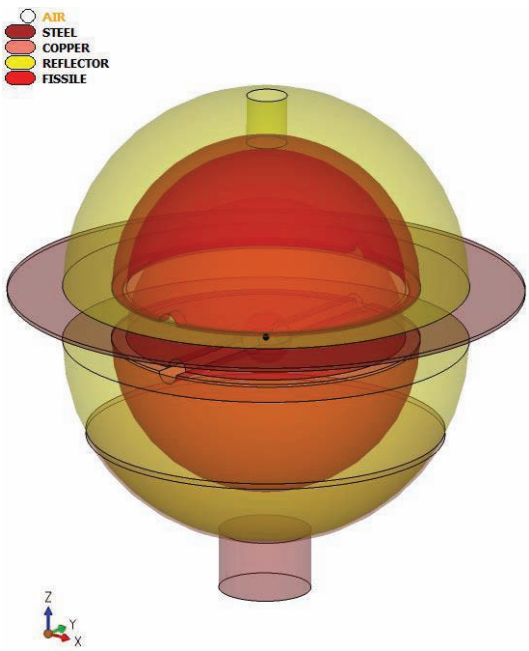

Fig. 4. Semi-transparent volumes.

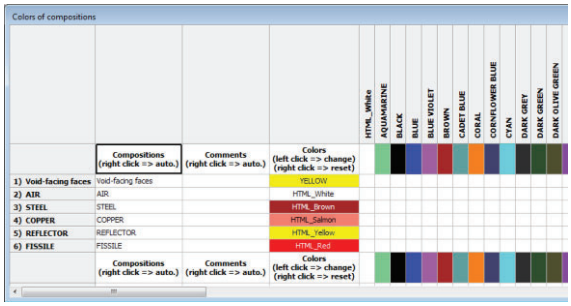

Fig. 5. Dialog box for color selection.

Lattices can be displayed with a variable level of detail: they can use either simplified sub-volumes (Fig. 6) or exact sub-volumes (Fig. 7).

Information can easily be obtained about the way pattern volumes are repeated to make up a given lattice (Fig. 8 and 9).

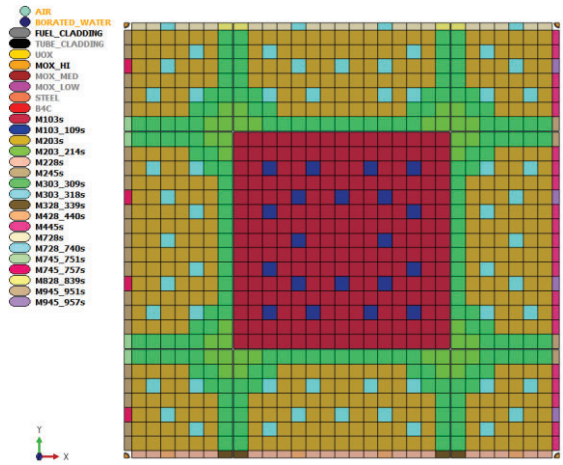

Fig. 6. Fuel assembly surrounded by halved fuel assemblies. Each assembly is a lattice using simplified sub-volumes.

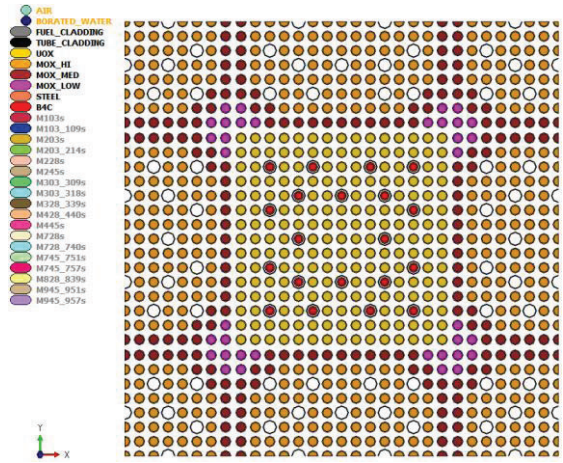

Fig. 7. Lattices using exact sub-volumes.

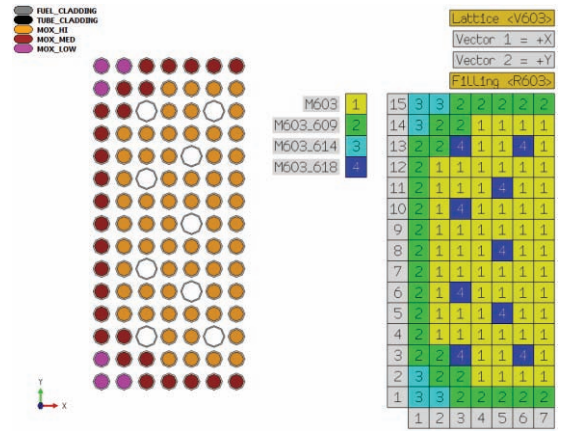

Fig. 8. Information about lattice “V603” in the 3D view: it uses lattice filling “R603” with 4 pattern volumes “M603”,“M603_609”,“M603_614” and“M603_618”.

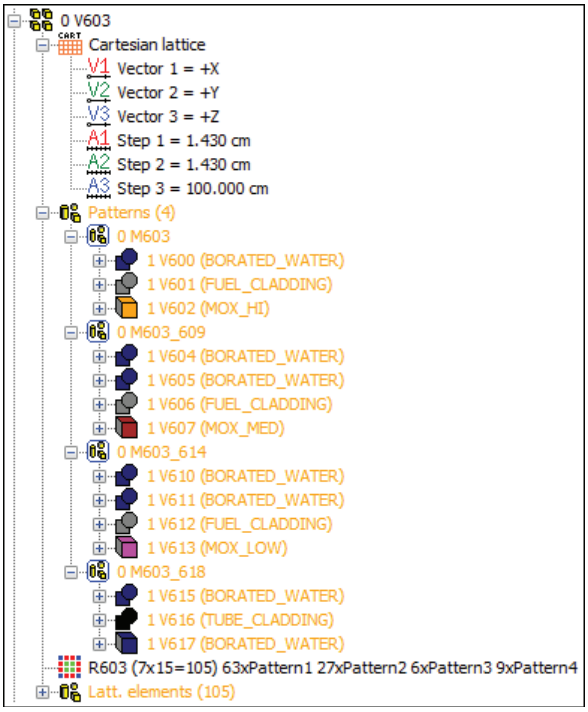

Fig. 9. Information about lattice "V603" in the geometry tree.

Hexagonal lattices are also implemented (Fig. 10).

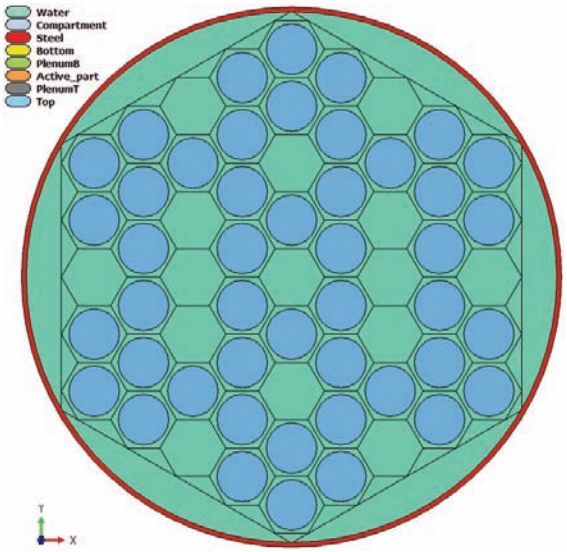

Fig. 10. Hexagonal lattice.

Volumes can be probed with a mouse-controlled exploration ray that gives a list of volume names, materials and thicknesses (see Fig. 11).

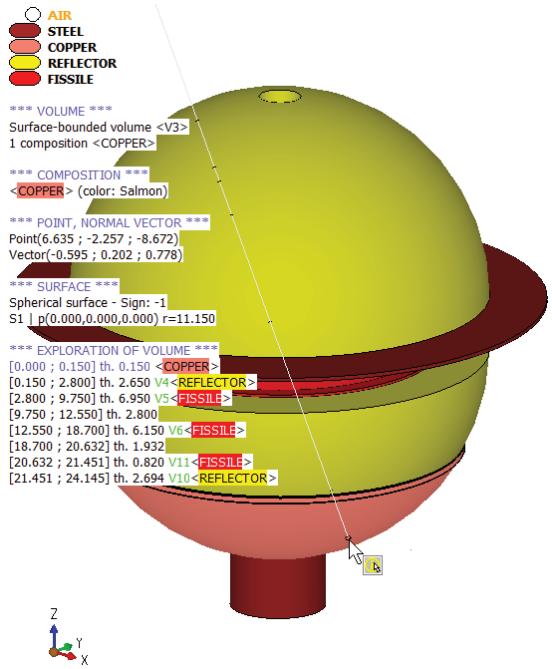

Fig. 11. Using the exploration ray.

The exploration ray can also detect overlapping volumes (Fig. 12). In this case, the ray will be lined with red lines in the overlap area(s).

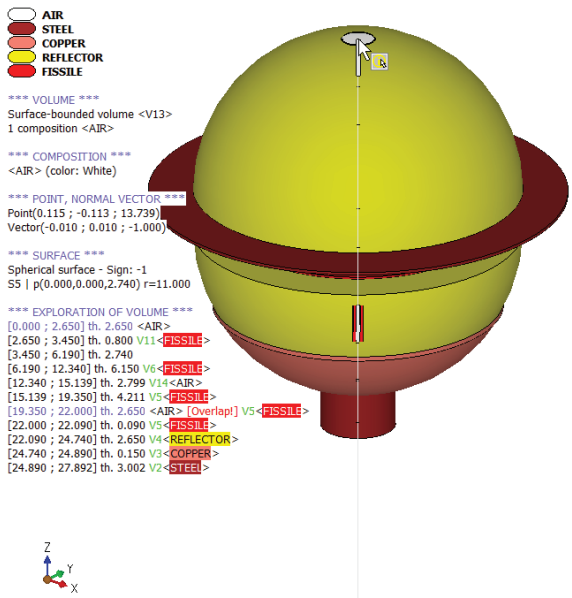

Fig. 12. Detecting overlapping volumes with the exploration ray: volume "V13" (air) overlaps volume "V5" (fissile).

A dynamic cross-section plane, powered by hardware graphics acceleration, can be dragged with the mouse through the model and offers a view of all internal details (Fig. 13), with the option of emphasizing the areas where volumes overlap (Fig. 14).

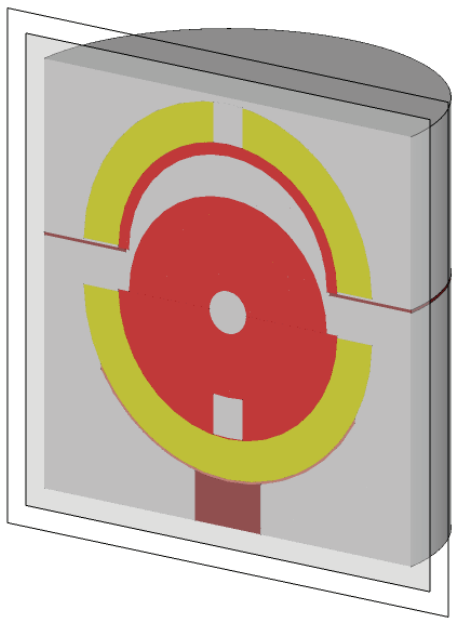

Fig. 13. Dynamic cross-section.

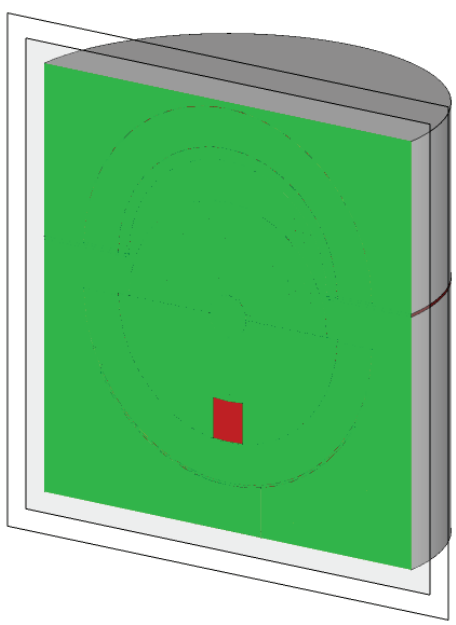

Fig. 14. Dynamic cross-section with overlap area in red.

The dynamic cross-section plane can be used to create a static cross-section (Fig. 15) that has more accurate edges and is intended to be copied in documents for illustration purposes.

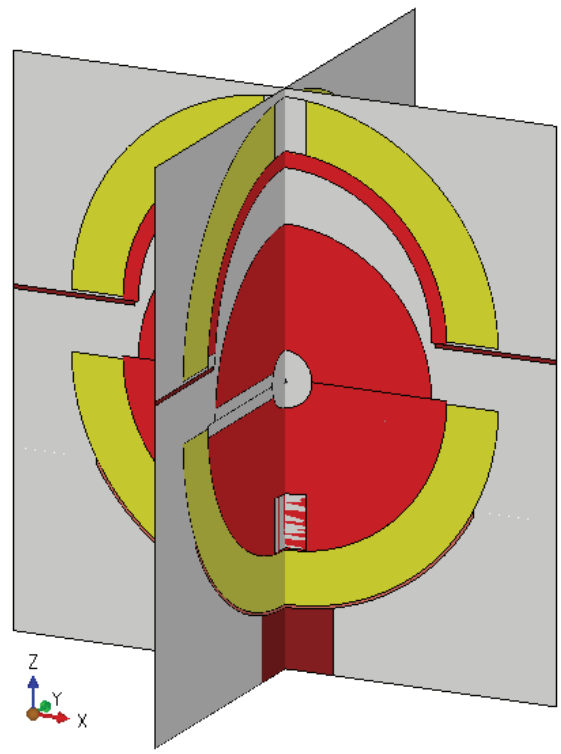

Fig. 15. Two static cross-sections.

A cutaway view of the model can be created by removing a box or a W-shaped volume (Fig. 16 and 17).

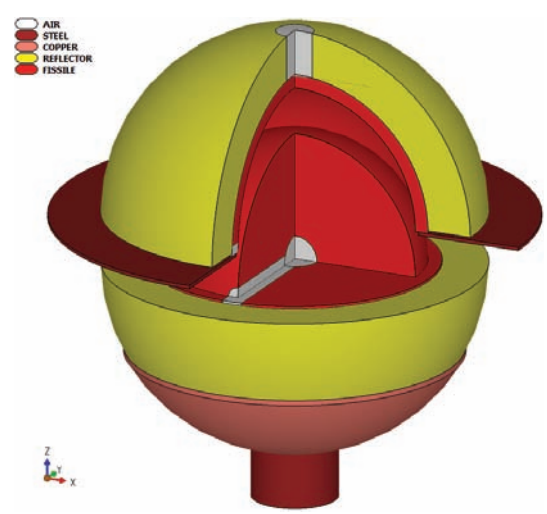

Fig. 16. Cutaway view obtained by removing a box.

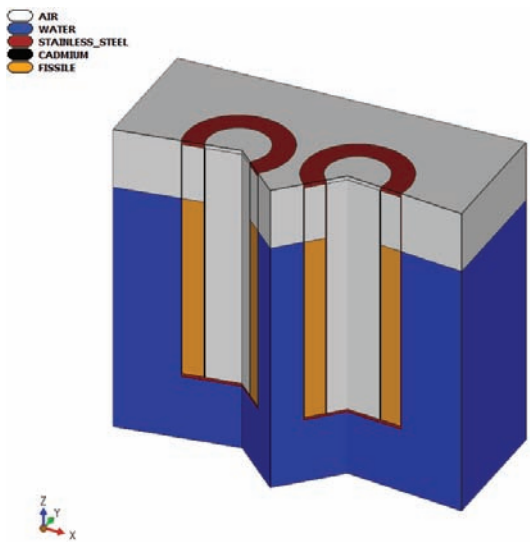

Fig. 17. Cutaway view obtained by removing a “W”.

The different entities handled by TRIAD can be accessed through the trees on the right side of the main window (see Fig. 18 and 19). The nuclear analyst can easily obtain the answer to the question: “Which volumes are using this surface?”

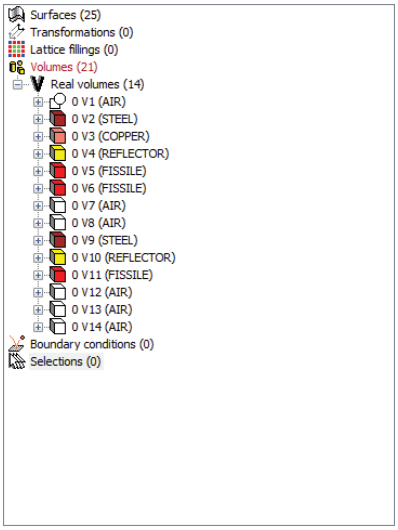

Fig. 18. Geometry tree, showing volumes.

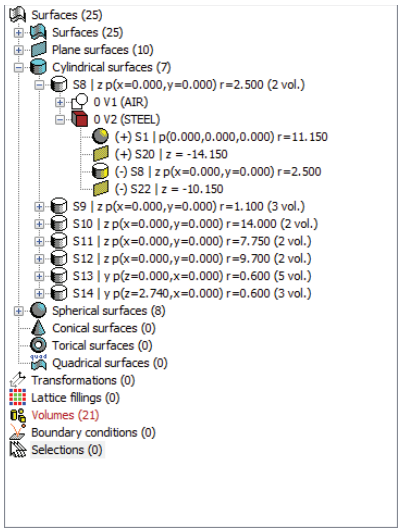

Fig. 19. Geometry tree, showing the 4 surfaces “S1”, “S20”, “S8” and “S22” used by volume “V2”.

Some buttons allow the user to click onto a material or surface or volume and immediately open the imported file with an advanced editor such as Notepad++[6] at the line where the material/surface/volume is defined. Once the file has been edited, the model can be imported again.

V. CHECKING THE MODEL VALIDITY

To check the validity of a model, the user can run an enhanced Monte Carlo diagnostic of volume overlaps and holes (Fig. 20). The display of volumes is then modified to show the overlapping volumes and the faces close to holes (Fig. 21).

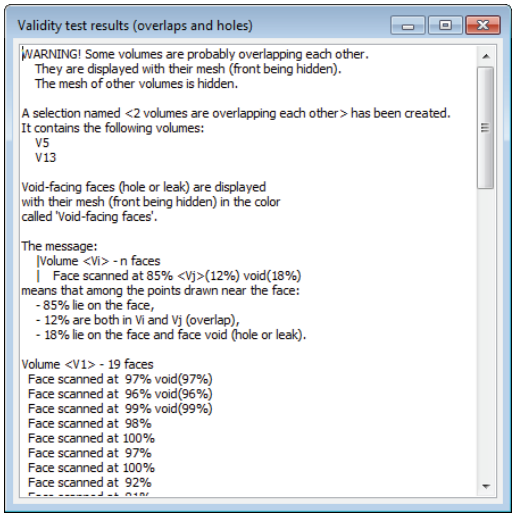

Fig. 20. Dialog box giving the result of a model diagnostic.

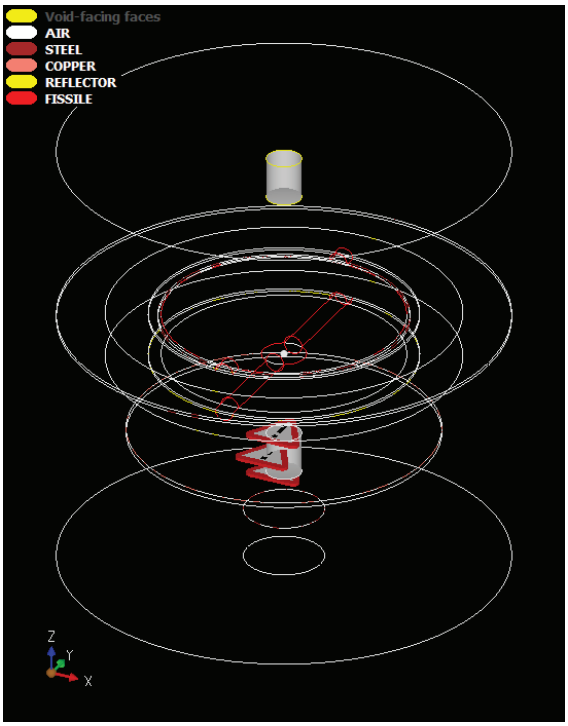

Fig. 21. 3D view resulting from running a model diagnostic. Triangular warning signs are displayed in the overlap area.

Regarding the diagnostic algorithm, TRIAD draws points randomly on each volume face and checks whether they happen to belong to other volumes (overlap) or to be close to an undefined space (hole or outer surface of model). This algorithm has proven to be far more effective than simply drawing points in a box bounding the model: if the model has to be checked with a given space resolution (such as 1 mm), the number of points needed on 2D faces is far less than the number of points needed in the 3D box.

VI. ANNOTATING THE MODEL

It is possible to display dimensions (Fig. 22), whether lengths or angles, by simply clicking on faces, edges or vertices.

This feature, typical of CAD software, has been developed to provide more accuracy than the ruler that can be sometimes seen in cross-sectional plots generated by Monte Carlo codes.

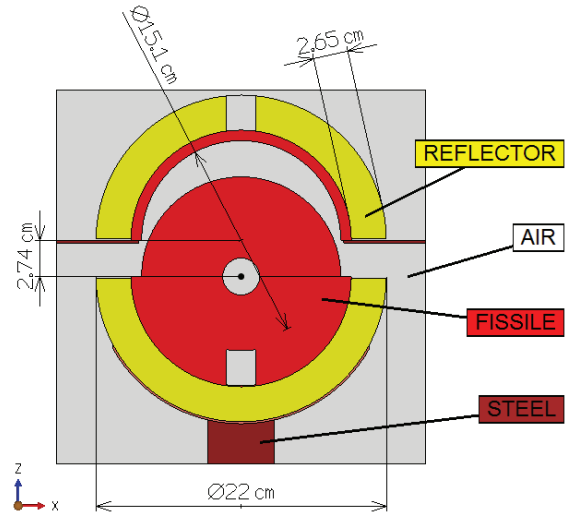

Fig. 22. Dimensions and tags created on a static cross-section.

The model can also be annotated with text rectangles called “tags” (Fig. 22).

Tags contain formulas that can be edited to display additional information such as volume names (Fig. 23).

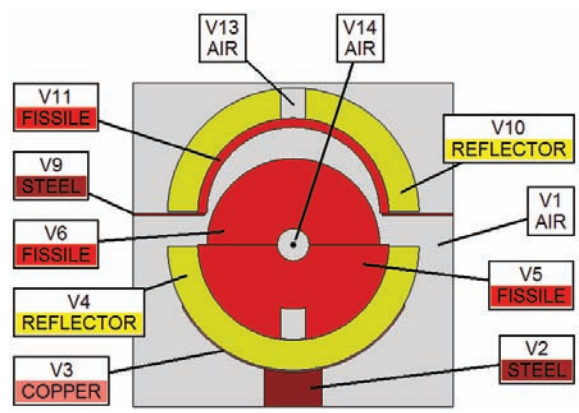

Fig. 23. Tags showing volume names and materials.

VII. DISPLAYING FIELDS

Scalar fields such as dose rates can be displayed in the 3D view (Fig. 24). A simple syntax can be used by the plugin to define field values on volumes, surfaces or a mesh.

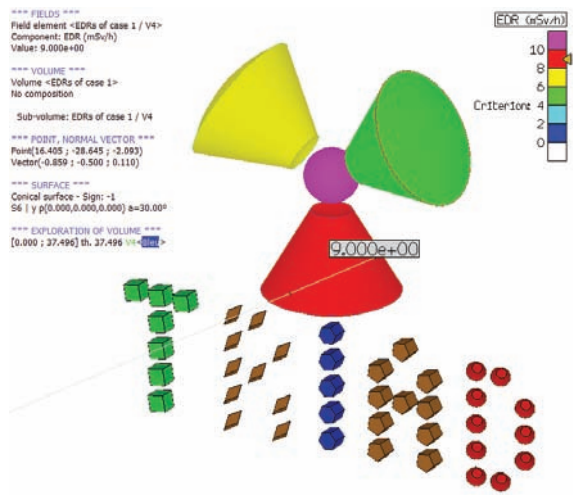

Fig. 24. Example of scalar field defined on volumes.

VIII. PERFORMANCES AND LIMITATIONS

TRIAD can import several thousands of volumes in a reasonable amount of time (no more than a few minutes). Importing simple geometries consisting of tens of volumes is only a matter of seconds (Fig 25).

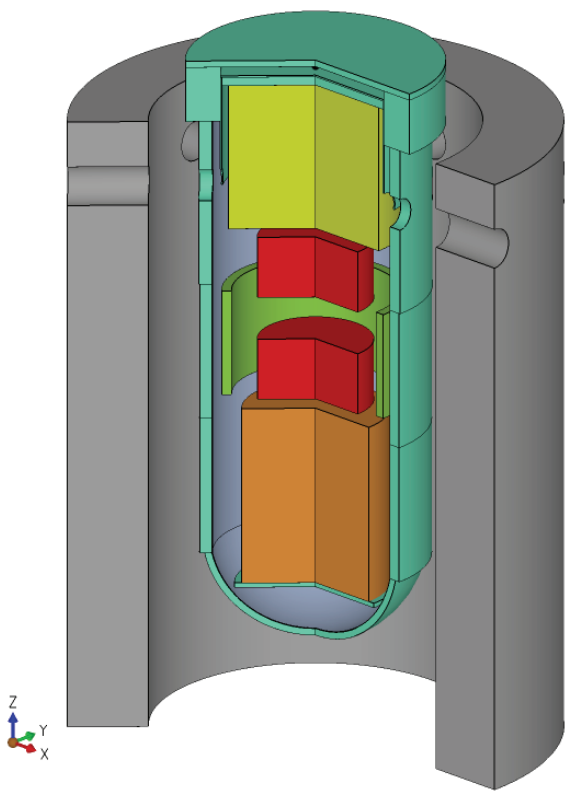

Fig. 25. Reactor vessel: a simple geometry.

The highest performances are reached for models composed with surface-bounded volumes only, without Boolean operations.

TRIAD being currently compiled as a 32-bit executable program, it will not be able to use more than 2 GB of RAM memory.

The surfaces bounding the volumes can be planes, cylinders, spheres or cones, but not general quadrics or tori. Moreover, if a volume utilizes two cones that are not coaxial, their intersection will not be displayed properly.

IX. DEVELOPING TRIAD

TRIAD is written in C++ and makes the most of object-oriented programming. The interface of TRIAD is implemented with wxWidgets[7], a free, open-source cross-platform GUI library.

The 3D display is implemented with OpenGL[8], the cross-language, multi-platform application programming interface (API). This API is used to interact with the graphics processing unit (GPU) and perform hardware-accelerated rendering.

The GPU is fed with segments and triangles that have been calculated by TRIAD's main algorithm. In the case of a volume defined as the intersection of algebraic half-spaces HSi = (surface Si, sign εi):

- TRIAD builds a list of segments that approximates the intersection of surfaces Si and Sj with a given maximum error (default value: 1mm);
- the segments are trimmed by the other half-spaces HSk = (Sk,εk) and make up the edge Eij(Si, Sj) of the volume;
- the other edges are determined in the same way;
- for each surface Si, the end points of the segments of the edges Eij(Si, Sj) are projected onto the 2D domain in the coordinate system relative to the surface (e.g. (theta, z) for a cylindrical surface);
- the abscissas and ordinates of the points are sorted to define a Cartesian grid;
- additional abscissas or ordinates are created in the direction(s) where the surface is curved (e.g. theta for a cylindrical surface);
- triangles are built to cover the 2D grid and transformed to the 3D space;
- some triangles are discarded if a certain number of their internal points is not found on the volume face.

The available importation plugins have been developed with Python, an interpreted, high-level programming language[9]. The Python plugins have been converted into executable Windows® programs that can be run without requiring a Python installation.

X. INSTALLING TRIAD

Installing TRIAD does not require any administrator right. TRIAD package (10 Mo) contains:

- a TRIAD 32-bit executable program;
- a directory containing plugins and associated help text files;
- a directory with TRIAD input file examples, one of which is fully commented with TRIAD input syntax.

XI. CONCLUSION AND OUTLOOK

TRIAD has been developed to alleviate the nuclear analysts' task of managing a huge number of volumes in their models.

Tens of users are already utilizing TRIAD in their everyday work.

Numerous capabilities enable them to explore all volumes and to check the model validity in terms of material distribution, overlaps, holes and dimensions. They can also display scalar fields such as mesh tallies on the imported geometry.

Future development efforts will include:

- issuing a 64-bit executable program for importing huge models;
- implementing quadric surfaces and torical surfaces (useful for displaying pipe elbows);
- plugins for other codes such as NARMER[10], MORET 5[11], Keno VI[12]...

ACKNOWLEDGMENTS

We wish to thank all TRIAD users for their priceless feedback and their timeless support.

REFERENCES

1. www.salome-platform.org

2. E. BRUN, F. Damian, C.M. Diop, E. Dumonteil, F.X. Hugot, C. Jouanne, Y.K. Lee, F. Malvagi, A. Mazzolo, O. Petit, J.C. Trama, T. Visonneau, A. Zola, "TRIPOLI-4®, CEA, EDF and AREVA reference Monte Carlo code," *Annals of Nuclear Energy*, **82,** pp. 151-160 (2015).

3. O. JACQUET, J. Miss, G. Courtois, "MORET: Version 4.B, a Multigroup Monte Carlo Criticality Code," *Proc. of ICNC'2003*, Tokai-mura, Japan (2003).

4. *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluations*, ORNL/TM-2005/39, Version 5, Vols. I–III, Oak Ridge National Laboratory, USA (April 2005). Available from Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-725.

5. X-5 Monte Carlo Team, *MCNP – A General N-Particle Transport Code, Version 5*, Vol. I: Overview and Theory, LA-UR-03-1987, Los Alamos National Laboratory, USA (April 2003).

6. www.notepad-plus-plus.org

7. www.wxwidgets.org

8. www.opengl.org

9. www.python.org

10. C. SUTEAU, M. Chiron, G. Arnaud, "Improvement of MERCURE-6's General Formalism for Calculating Gamma-Ray Buildup Factors in Multilayer Shields," *Nucl. Sci. & Eng.*, **147,** pp. 419-424 (2004).

11. B. COCHET, A. Jinaphanh, L. Heulers, O. Jacquet, "Capabilities overview of the MORET 5 Monte Carlo code," *Annals of Nuclear Energy*, **82,** pp. 74-84 (2015).

12. B.T. REARDEN, L.M. Petrie, D.E. Peplow, K.B. Bekar, D. Wiarda, C. Celik, C.M. Perfetti, A.M. Ibrahim, S.W.D. Hart, M.E. Dunn, W.J. Marshall, "Monte Carlo capabilities of the SCALE code system," *Annals of Nuclear Energy*, **82,** pp.130-141 (2015).