Visualization of Simulated 3d–Geometry in Transfer Equation Solution Problems Using Monte-Carlo Technique

Elena Klass, Sergey Ulyanov
Central Research and Development Institute of Chemistry and Mechanics
Moscow, Russia
elenaklass@yandex.ru, ulyanov.sergey@gmail.com

Abstract

Standard 3d-visualization programs assume object surface representation as a set of polygons. When the ionizing radiation transfer equation is solved in 3d-geometry using Monte-Carlo technique an object geometry is usually introduced as a set of second order surfaces. The current work is focused on 3d modeling of an object, where the second order surfaces are visualized using marching cubes algorithm, representing the surfaces as a set of cubic blocks. The results of the proposed method application for visualization of a specific simulated geometry are presented.

Keywords: *3d-visualization, second order surfaces, quadrics, polygonization, triangulation.*

1. INTRODUCTION

Originally, the practice of geometrical modeling applied to solving transfer equations using Monte-Carlo technique, has adopted an approach that utilized second order surfaces and body-primitives to represent real object designs. [1, 3-5, 9]. To increase the accuracy of geometry description the simulation model can be represented also by a big number of voluminous cubic elements (voxels) with a size of some mm^3. For example, the problems of radiation protection are actively dealt with using mathematical models describing a human body by a set of cubic blocks, based on computer tomography data [7, 10]. Similar geometry representation is applied as well in CAD-systems. While solving transfer problems using Monte-Carlo technique the principle drawback of the voxel representation of an object under study is that geometrical models consisting of numerous cells require significant increase in run time. This is connected with peculiarities of particle tracking using the method of Monte-Carlo. In particular, after generation of an interaction point and the particle free path length it is necessary to search for the cell boundary to be crossed by the generated track. After the boundary is found similar search is undertaken for the next cell, crossed by the track. If the cell size is small and the track length is big the run time to model one trajectory explodes by orders of magnitude. Due to that in practice an approach is preserved based on a simplified representation of object geometry by second order surfaces.

The validation of the geometrical description is performed by special programs. Standard approach to monitor the geometry is to visualize 2d cross sections parallel to the reference system axes. [4]. In recent years visualization packages in 3d have been produced for the geometrical module of Monte-Carlo programs MCNP5 and MCNPX [3, 11, 12], using ray tracing as well as 3d geometry in the form of second order surfaces. This makes it possible to obtain more obvious idea of the object model. The goal of the current work was to create an algorithm, making it possible to produce a 3d object model for the geometrical module of national program “РОБОТ” [14].

2. METHOD OF GEOMETRICAL MODEL REPRESENTATION IN “РОБОТ” PROGRAM

To introduce the problem geometry a user [14] performs division of the object and surrounding space by so called zones. A zone is a region in space, limited by first and second order surfaces. Necessary condition here is that the zone should not be crossed by surfaces, which elements constitute its boundaries. The zone is called simple if under the problem conditions it can be considered uniform in composition and compound, if this requirement is not met. The division by zones is performed in sequential allocations of regions in space using the above mentioned surfaces specifying the surface sign. The following parameters are introduced in the main geometrical array for each zone:

- numerical ID of the surface equation with corresponding sign
- material ID of the zone
- IDs of all equations, limiting the zone and ID of the equation determining the zone.

The latter equation ID list is necessary to estimate path lengths from interaction points up to the zone boundaries. All geometrical zones, identified in the first stage, are described similarly. In the next stage the description of each zone, identified as “compound” or “composite”, is performed. The division is continued until all the simulated region is covered with simple zones. Description principle of geometrical zones is maintained during all stages. The final geometrical zones description array is composed of a series of sequential division arrays of the simulation region.

In the process of the problem solution in order to find which zone contains particular point its coordinates (*x*,*y*,*z*) are sequentially introduced in equations, determining 1st, 2nd, 3d, etc. zones accounting for sign of the equation. Point (*x*,*y*,*z*) belongs to zone *v*, described by equation *j* if the following condition is true:

$$sign(v_j)(c_{1j}x^2+c_{2j}y^2+c_{3j}z^2+c_{4j}xy+c_{5j}xz+c_{6j}yz+c_{7j}x+c_{8j}y+c_{9j}z+c_{10j}) > 0,$$

where $sign(v_j)$ is the sign of equation, describing the zone. In case the point appears to belong to a compound zone the search is continued starting with its first subzone. As a results, the geometrical module contains points in space with corresponding two numbers: material ID and zone ID.

3. GEOMETRICAL OBJECT 3D-MODELING ALGORITHM

As specified above, each geometrical region in the object description follow the format of РОБОТ program using first and second order boundary equations and represents a connected set. The proposed visualization algorithm depends on the complexity of the surface equations bounding the zone. If the zone is bounded by first order equations polygonization of the zone is performed. If second order equations exist in the description, triangulation of the zone is done using marching cubes algorithm [6].

Visualization algorithm proposed in the current work contains the following.

3.1 Validation of geometrical zones

In the first stage the validation of the geometry of each zone is performed, including search for unbounded regions in space as well as the correctness of the defined boundaries and their signs. The zone geometry is considered to be defined correctly if the space, bounded by first and second order surfaces, is closed, i.e. represents some volume. For this purpose during processing (sequential or parallel) of each zone a search problem is solved of minimization of the enclosing rectangular parallelepiped. This subproblem is reduced to search for the extrema in the degrees of freedom, satisfying the inequality system of the surface equations, Such problem statement can be easily reduced to the problem of mathematical programming:

$$f(\vec{x}) \to min,\ \vec{x} \subseteq \mathrm{E}^3,$$

subject to:

$$g_i(\vec{x}) \leq 0,\ i=[1,m],$$

where:

$\vec{x}$ – coordinates vector in space $\vec{x} = (x, y, z)^T$;

$f(\vec{x})$ – objective function, represented by three degrees of freedom $f(\vec{x}) = x$, $f(\vec{x}) = y$ *and* $f(\vec{x}) = z$;

$g_i(\vec{x})$ – bounding functions, corresponding to 1st and 2nd order surface equations;

m – number of boundary conditions.

As a result, to find six bounding faces of the parallelepiped it is necessary to solve the problem of mathematical programming six times in order to find global minima and maxima of each degree of freedom. Since boundary conditions are generally nonlinear functions, to solve the defined problem it is necessary to use one of the nonlinear optimization methods. In our solution we applied the flexible tolerance algorithm [15] due to the fact that this method is direct (do not use derivatives), and not very sensitive to the choice of initial feasible solution. One can utilize linear programming technique in the specific case when all boundary conditions are first order equations.

In case the described parallelepiped exists, i.e. finite limits were found for all reference system axes and it has not degenerated in a point or plane, then it is decided that the zone was defined correctly (represents a bounded region in space) and is subject to visualization (Figure 1).

3.2 Zone visualization

Based on surface types describing a specific zone it is possible to identify two major cases: only 1st order equations and combined description. Such classification can drastically increase the visualization algorithm efficiency by simplified polygonization of zones, described only by 1st order equations.

3.2.1 Polygonization of zones, defined by 1st order equations

If a zone is described only by 1st order equations, then each of its bounding surfaces corresponds to maximum one polygon, lying on its surface. For each plane, bounding a zone, points of intersections with pairs of other such planes are found. The resulting points are filtered to select those, belonging to the zone surface (i.e. satisfy unstrict inequalities, defining the zone). As a result vertex coordinates of the polygon lying on the zone surface are obtained. Then, a normal to the plane under consideration is built, directed outside (Figure 1) and obtained points are sorted anti-clockwise with respect to this normal, which is necessary to correct mapping of the polygon.

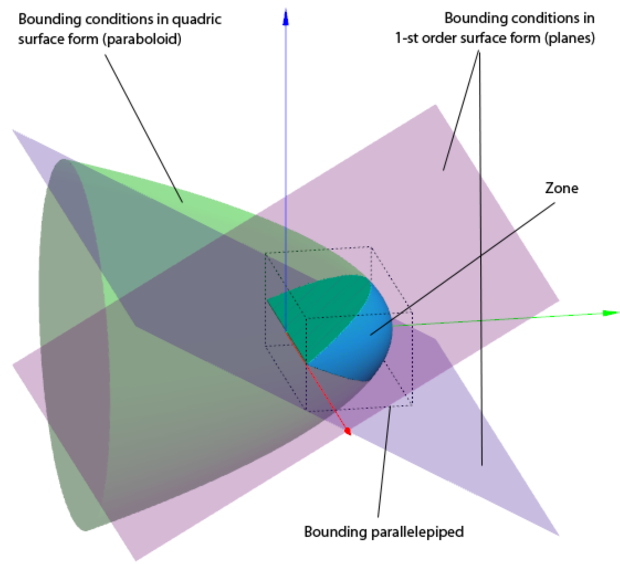

Figure 1: Zone, bounding conditions and parallelepiped example.

3.2.2 Triangulation of zones, defined by 2nd order equations

The process of visualization of the zones, defined by 2nd order equations, consists of dividing the three dimensional space into cubic blocks of predefined size, each of which experience triangulation.

To perform triangulation of 2nd order equation zones marching cubes algorithm is used [6].

It was mentioned above, that during validation of geometry definition an enclosing parallelepiped is found for each zone. In the process of zone triangulation the space bounded by its faces is split into cubes of different sizes. Then each cube is processed to find intersections with bounding surfaces of the zone.

Each cube vertex is probed for belonging to the geometrical zone under search. Eight Boolean quantities are obtained or one byte of information. This number in a pre-calculated table corresponds to a set of triangles, forming part of the image. The triangles are defined by indexes of cube edges, containing their vertexes. Precise location of each such vertex is determined by the solution of the problem of cube edge intersection with corresponding zone surface. If one vertex of the cube edge lies within the zone and the other is outside, then it is assumed that this cube edge intersects the zone surface only ones. If both vertexes either belong or do not belong to the zone, then the intersection is missing at all. If the distance between the zone boundaries is less than the cube edge length, the zone can remain invisible. That is why when marching cubes algorithm is used the cube sizes are selected based on characteristic sizes of the geometry.

4. RESULTS

Based on the described algorithm a program application was developed, making it possible to perform validation and visualization of object geometry of defined complexity (Figure 2, 3).

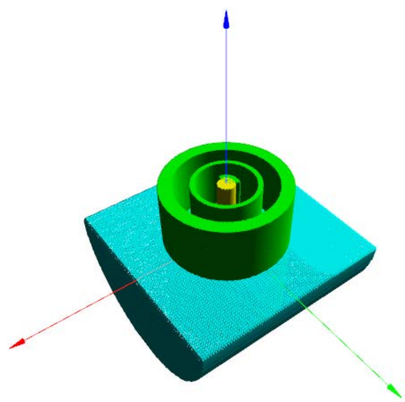

Figure 2: Example of visualized object.

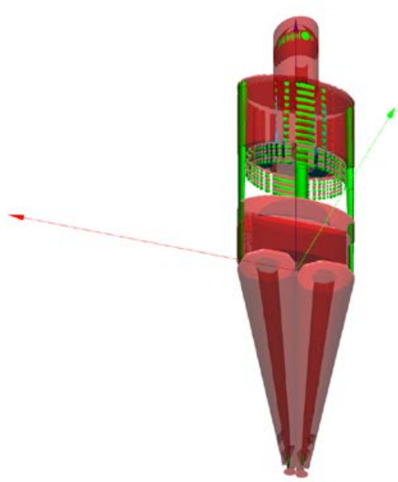

Figure 3: Visualization of the components of a heterogeneous model of the human body (MIRD-phantom [8, 13]).

5. CONCLUSION

Algorithm of 3d-visualization of geometrical file, containing object description in the form of 2^{nd} order surfaces, widely used in radiation fields Monte-Carlo simulation programs is developed.

6. REFERENCES

[1] *GEANT. Detector Description and Simulation Tool. CERN, Geneva, Switzerland, 1993*.

[2] *Geant4 User's Documents Version: Geant4 3.2 June 2001*.

[3] *Graphical User Interface for High Energy Multi-Particle Transport. Final Report November 30th 2008 Home Page: http://www.mcnpvised.com*.

[4] *MCNP4C: Monte Carlo N-Particle Transport Code System. ORNL RSICC Computer Code Collection, CCC-700, April, 1990*.

[5] *MCNP – A General Monte Carlo N-Particle Transport Code, Version 5, X-5 Monte Carlo Team, LA-UR-03-1987, April 24, 2003*.

[6] Paul Bourke. *Polygonising A Scalar Field* http://paulbourke.net/geometry/polygonise/

[7] L. de Carlan, P. Roch, E. Blanchardon and D. Franck. *New method of voxel phantom creation: application for whole-body counting calibration and perspectives in individual internal dose assessment. Radiation Protection Dosimetry, 2005, 116(1-4):160-*164.

[8] Cristy M. *Mathematical phantoms representing children of various ages for use in estimates of internal dose. U.S. Nuclear Regulatory Commission Rep. NUREG/CR-1159 (also Oak Ridge National Laboratory Rep. ORNL/NUREG/TM-367).*

[9] Emmett M.B. *The MORSE Monte-Carlo Radiation Transport Code System. Report-ORNL-4972, 1975*.

[10] J.F. Evans, T. E. Blue, N. Gupta. *"Absorbed dose estimates to structures of the brain and head using a high-resolution voxel-based head phantom." Med Phys. 2001 May;28(5):780-6.*

[11] R.A. Schwarz, L.L. Carter, W Brown. *"Particle Track Visualization Using the MCNP Visual Editor, "Proc. Am. Nucl. Soc. Topical Radiation Protection for Our National Priorities Medicine, the Environment and, the Legacy, 324-331, 2000, Spokane, Washington.*

[12] A.L. Schwarz, R.A. Schwarz, L.L. Carter. *"3D Plotting Capabilities in the Visual Editor for Release 5 of MCNP," Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew, April 6-11, 2003 on CD-ROM, American Nuclear Society, La Grange Park, Illinois (2003).*

[13] Snyder W.S., Ford M.R.,Warner G.G. and Fischer H.I. *MIRD. Pamphlet 5, Medical Internal Radiation Commitee, J. Nucl. Med., 1969, Suppl. 3, p. 7-12, 46-52.*

[14] Класс Е.В., Вязьмин С.О., Шаховский В.В. и др. *Состояние и возможности комплекса программ РОБОТ по расчету трехмерной защиты методом Монте-Карло. Тез. докл. // Защита от ионизирующих излучений ядерно-технических установок. ФЭИ, Обнинск, 1994. С. 106-108. (6-я Всесоюзн. конф.).*

[15] Химмельблау Д. *Прикладное нелинейное программирование, «Мир», Москва, 1974*