LM-07K010
March 1, 2007

Converting Boundary Representation Solid Models to Half-Space Representation Models for Monte Carlo Analysis

JE Davis, MJ Eddy, TM Sutton, TJ Altomari

Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)
Monterey, California, April 15-19, 2007, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2007)

Converting Boundary Representation Solid Models to Half-Space Representation Models for Monte Carlo Analysis

J. E. Davis, M. J. Eddy, and T. M. Sutton
Lockheed Martin Corporation
P. O. Box 1072
Schenectady, NY 12301-1072
davije@kapl.gov

T. J. Altomari
Bechtel Bettis, Inc.
P. O. Box 79
West Mifflin, PA 15122

ABSTRACT

Solid modeling computer software systems provide for the design of three-dimensional solid models used in the design and analysis of physical components. The current state-of-the-art in solid modeling representation uses a boundary representation format in which geometry and topology are used to form three-dimensional boundaries of the solid. The geometry representation used in these systems is cubic B-spline curves and surfaces—a network of cubic B-spline functions in three-dimensional Cartesian coordinate space. Many Monte Carlo codes, however, use a geometry representation in which geometry units are specified by intersections and unions of half-spaces. This paper describes an algorithm for converting from a boundary representation to a half-space representation.

Key Words: solid modeling, CAD, boundary representation, BREP, half-space representation

1. INTRODUCTION

Solid modeling computer software systems enable the design of three-dimensional solid models used in the design and analysis of physical components. These systems are essential for the design and manufacture of commercial products, from small portable electronic devices to large mechanical equipment. The current state-of-the-art in solid modeling representation uses a boundary representation (BREP) format in which geometry and topology are used to form three-dimensional boundaries of the solid. The geometry representation used in these systems is cubic B-spline curves and surfaces—a network of cubic B-spline functions in three-dimensional Cartesian coordinate space [1, 2].

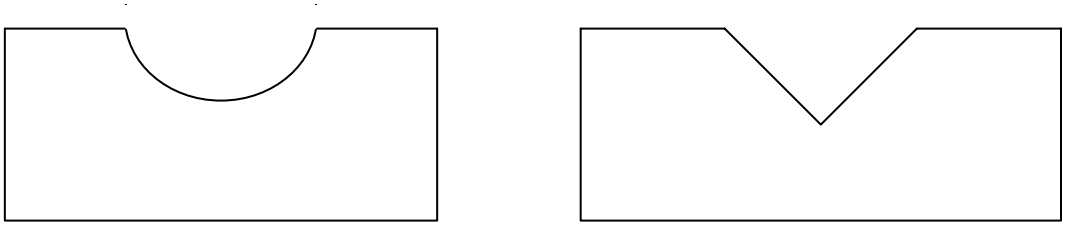
Some Monte Carlo codes [3, 4, 5], however, use a geometry representation in which geometry units are specified by intersections and unions of half-spaces defined by quadric surfaces. A half-space is the set of points defined by a surface and a sense with respect to that surface. The surface separates space into two sections—one on each side. The sense (positive or negative) distinguishes between these two sections. For example, a half-space may be defined as the set of points on one side of an infinite plane. The complementary half-space would be the set of points

on the other side of the plane. A half-space need not be infinite in extent. For example, one could be defined as the set of points within a spherical surface.

This paper describes the generation of boundary representation models using commercially available solid modeling tools, such as Pro/ENGINEER, SolidWorks, or Solid Edge modelers, for applications requiring the use of a half-space model representation. A recursive algorithm for converting a three-dimensional boundary representation to the equivalent half-space representation is presented. This algorithm is specific to the half-space representation used by the RACER Monte Carlo code [3], which has two particular characteristics: the surfaces are constrained to be quadrics, and geometry units are given only as intersections of half-spaces, i.e. unions and negations of half-spaces are not allowed. The intersections-only constraint of the RACER Monte Carlo geometry[1] leads to issues that are not encountered by similar codes [6] that generate geometry for Monte Carlo codes such as MCNP [4] and MC21 [5] that do not have this restriction. In addition to Monte Carlo particle transport, other applications of this type of capability include reverse engineering of mechanical components, image and sensor processing, and image synthesis via ray tracing.

2. THE ALGORITHM

In order to achieve an accurate half-space conversion of a BREP model, a model must be subdivided into discrete sub-models that can be represented solely with half-space boundary surfaces. In order to achieve this objective, any concave edge definition must be removed from the model. In Figure 1, the body on the left can be represented by half space surfaces, but the body on the right cannot be represented due to the concave edge within the concave section of the body.



Concave body without a concave edge *Concave body with a concave edge*

Figure 1: Concave Figures

[1] While these geometry constraints may make RACER model construction more difficult in some cases, it should be noted that the decision to impose these constraints was purposely made to enhance the speed with which neutrons can be tracked.

Half-space model representations use surfaces for three purposes. In this paper, we define *natural* surfaces as those that define the surfaces of the physical component being modeled. In order to define the solid model as intersections of half-spaces, however, it may need to be split into multiple pieces. This splitting is performed using both the natural surfaces and additional *separating* surfaces. Finally, *disambiguation* surfaces are required, for example, to distinguish between two sheets of a hyperboloid. In addition to the topological conversion from a boundary representation to a half-space representation, a geometric conversion is also necessary to replace the cubic B-spline surface representation with a quadric surface representation. We are supporting surface types that are supported by quadric representation, including planes, ellipsoids, cylinders, or other quadric types. The general form of the cubic B-spline free form surface is not supported by this conversion process.

A quadric surface is defined as the locus of zeros of the quadratic polynomial equation

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz + Gx + Hy + Iz + J = 0\,. \tag{1}$$

For implementation of this algorithm, we used Parametric Technology Corporation's Granite One geometry kernel application programming interface. This library of methods and algorithms provided curve and surface analysis, intersection operations, and surface conversion operations to transform the model into the convex sub-models. In particular, we required methods to slice a model into two or more sub-models by a single planar or cylindrical surface. The resulting sub-models would have their own unique boundary representation structure for further analysis during the recursive subdivision of the algorithm.

2.1. Algorithm Outline

1. Place all models to be converted into a list of models.

2. For each model on the list until the list is empty:

    a. Does the model contain one or more concave edges?

        i. If yes, slice the model with one of the natural half-space surfaces that are adjacent to the edge. A planar surface slice is preferred over cylindrical or spherical slices. All resulting models from the slicing operation are placed back onto the list of models.

        ii. If no, move the model onto another list of models for step 3 processing.

3. For each model on the list until the list is empty:

    a. Does the model have a non-planar surface?

        i. If yes, determine if a separating half-space must be introduced based on the location of the edges and the geometry parameterization.

1. If a slice is necessary, slice the model with the separating half-space surface. All models generated from the slicing operation are placed back on the list.

2. If not, move the model to the completed list.

ii. If no, move the model to the completed list.

4. For each sub-model on the completed list:
   a. Convert all sub-model surfaces to quadric coefficient form.

5. Identify and remove duplicate surfaces.

6. For each sub-model on the completed list:
   a. Create a half-space volume based on the surfaces and orientations.

2.2. Concave Edge Evaluation and Natural Half-Space Slicing

The conversion algorithm first examines a model to determine if the model contains one or more concave edges. An edge is determined to be concave based on the edge tangent vector and the adjacent surface normal vectors (see Fig.2). If a concave edge is found, the algorithm slices the model with a natural half-space surface, i.e. one of the planar, cylindrical, or spherical surfaces adjacent to the concave edge. The surface is extended to impact the current model or sub-model under investigation. Our algorithm prefers a planar slicing operation for performance, but either surface could be used.

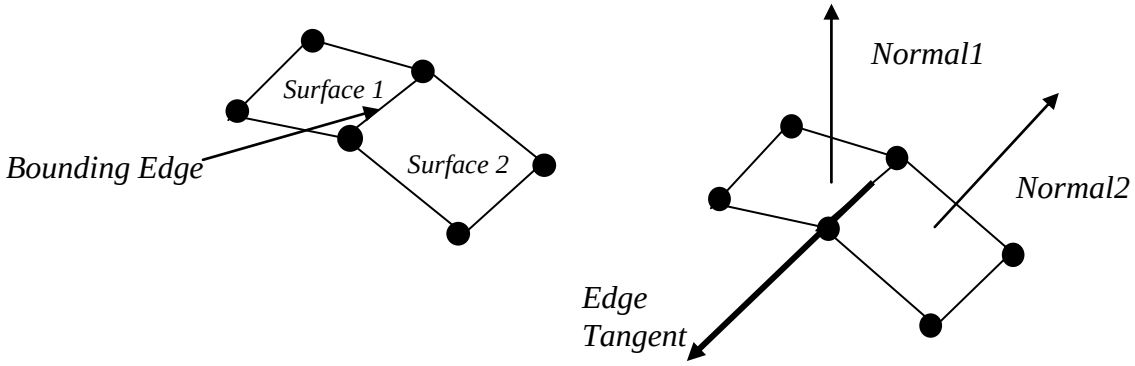


Figure 2: Boundary Edges and Vector Definitions

The slicing operation generates at least two sub-models or subdivisions of the original model. These subdivisions, in turn, are placed back on the list of models for further examination for concave edge evaluation and slicing, if necessary. The algorithm proceeds recursively until no concave edges remain.

At this point in the algorithm, a set of one or more sub-models will exist. Among this set, all edges will be convex. All concave edges have been eliminated through the slicing of natural half-space surfaces.

2.3. Requirements for Creation of Separating Half-spaces

The algorithm then processes the set of convex sub-models one at a time to determine if a separating half-space is needed to fully qualify the volume. If the sub-model contains only planar surfaces, no further evaluation is required. If any non-planar surfaces are encountered, an evaluation is required to determine if the sub-model needs to be sliced.

Figure 3 shows an example of an unambiguous half-space representation with an interior cylindrical half-space surrounded by four planar half-spaces. This model is a top view of rectangular block with a circular hole. The introduction of a separating half-space is not required for this model.

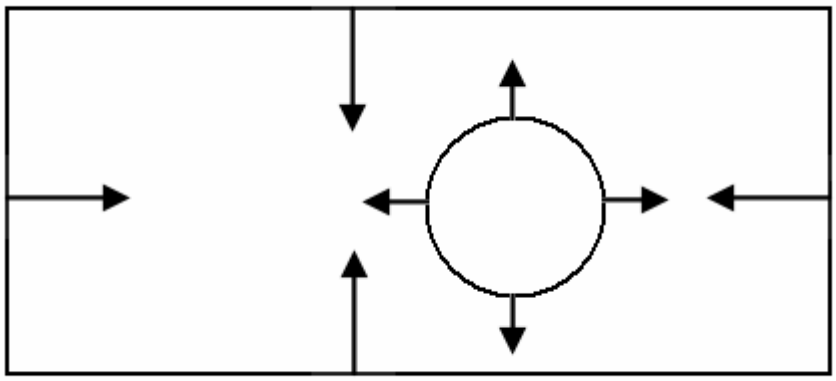


Figure 3: Unambiguous Model

Figure 4 illustrates the need for a separating half-space and model subdivision. Points A and B are contained within the planar and cylindrical boundaries. However, the mathematical definition of the cylindrical surface forms a closed surface (See Fig. 5).

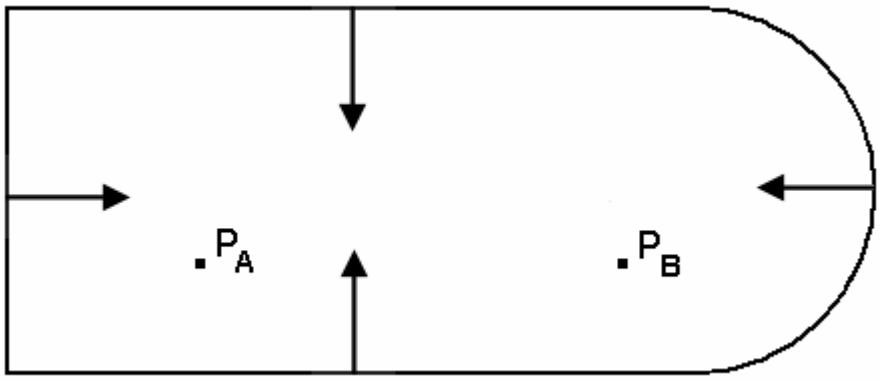


Figure 4: Ambiguous Model

Point A is contained within the three planar half-spaces, but it is not contained within the cylindrical half-space, h_1. Point B, does satisfy all the half-space volume constraints.

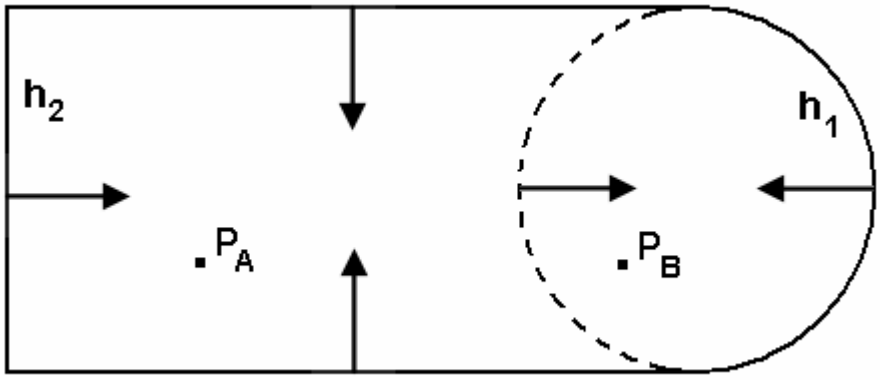


Figure 5: Ambiguous Model – Cylindrical Half-Space Issues

By subdivision of this model into two sub-models (see Fig. 6), the ambiguity of the volume definition is resolved. The planar separating half-space, h_3, isolates the influence of the cylindrical half-space, h_1, into the smaller semi-circle. The remaining volume is accounted for within four planar half-spaces, three natural half-spaces defined by the model geometry and the fourth separating half-space.

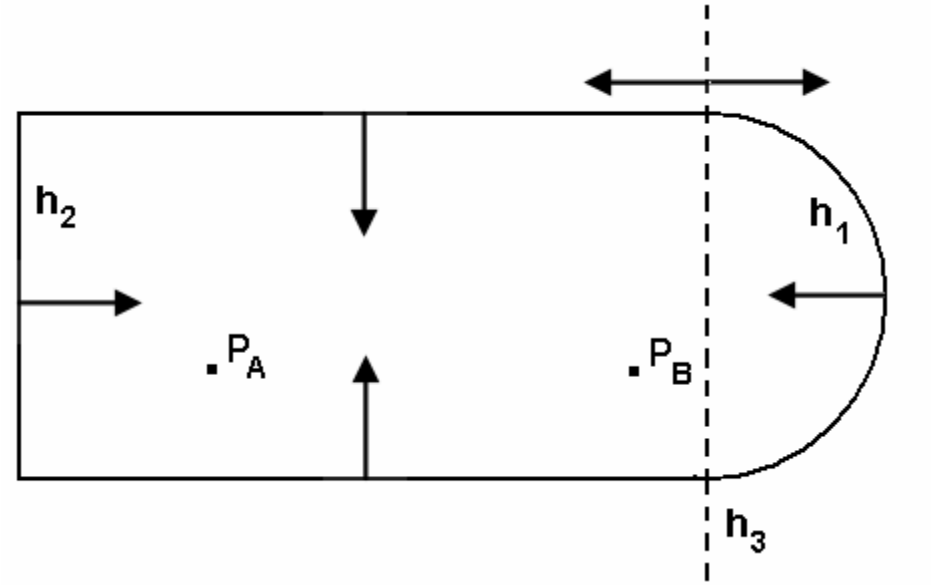


Figure 6: Ambiguous Model – Separating Half-Space Resolution

The construction of the half-space, h_3, was done based on the cylindrical surface's edge locations at the interface with the two planar surfaces. The resulting models will be placed back on the model list to be reevaluated.

Figure 7 demonstrates a different ambiguity. In this example, Point B satisfies all the half-space equations for volume containment, but it does not lie within the physical geometry model. A separating half-space is required to isolate the "infinite" volume on the right side of the figure.

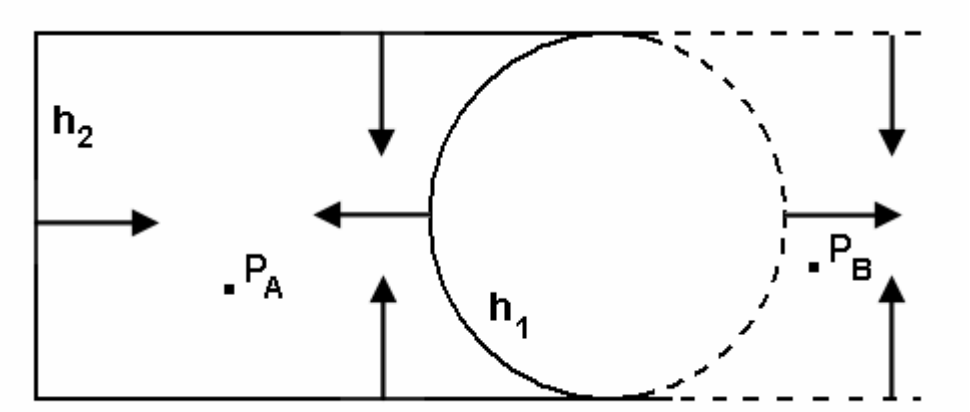


Figure 7: Ambiguous Model

3. EXAMPLE AND ILLUSTRATIONS

The following illustrations describe the algorithm in sequence. A simple three dimensional model is shown in Figure 8 with one cylindrical surface and three linear concave edges.

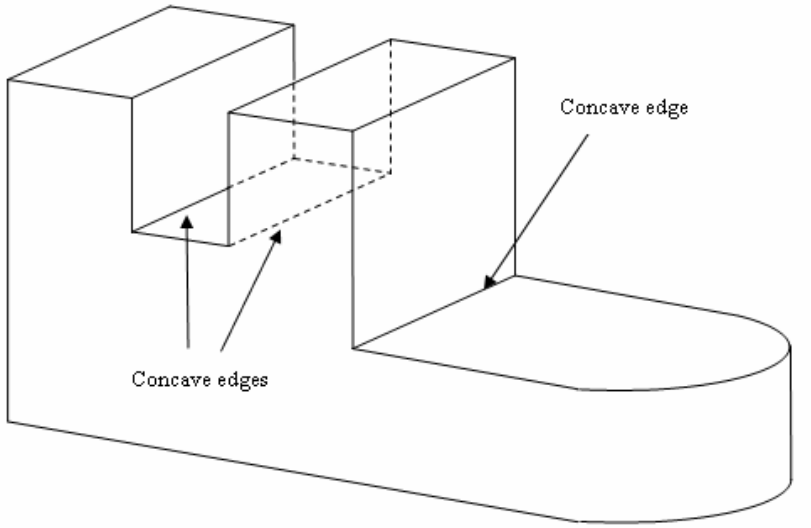


Figure 8: Initial Model with Concave Edges Identified

One concave edge is selected for resolution. One of its adjacent planar surfaces is chosen for the slice operation (See Fig. 9).

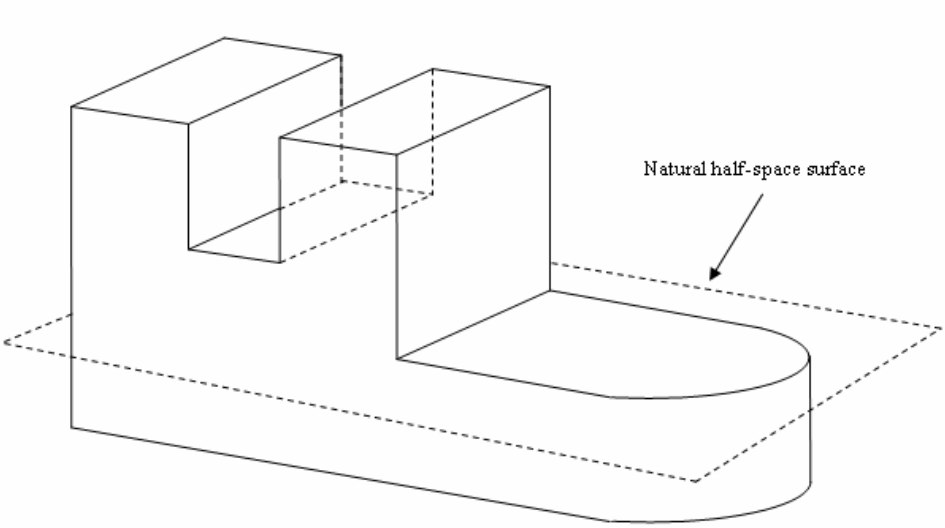


Figure 9: Identification of a Natural Half-Space Surface for Slicing

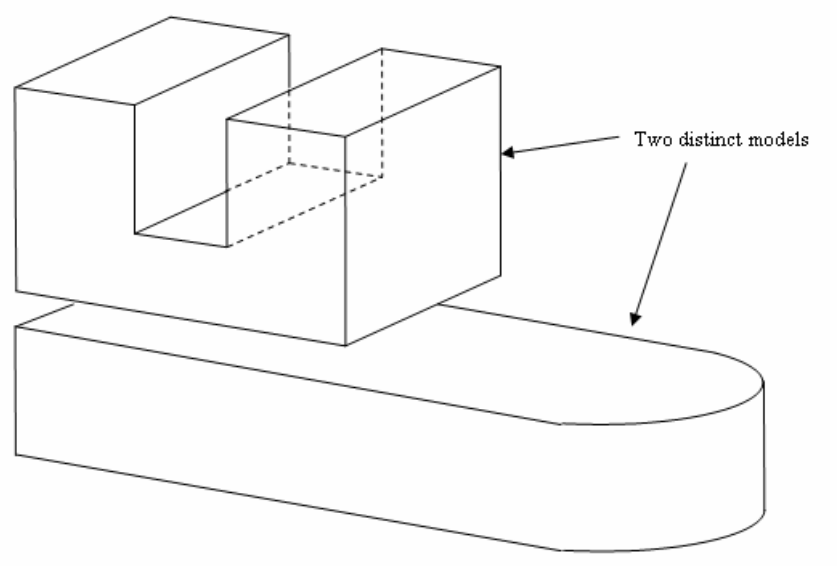


Figure 10: Results of the Slicing Operation

As a result of the planar slice, two sub-models are created. One sub-model contains two concave edges, while the second body has only convex edges (see Fig. 10).

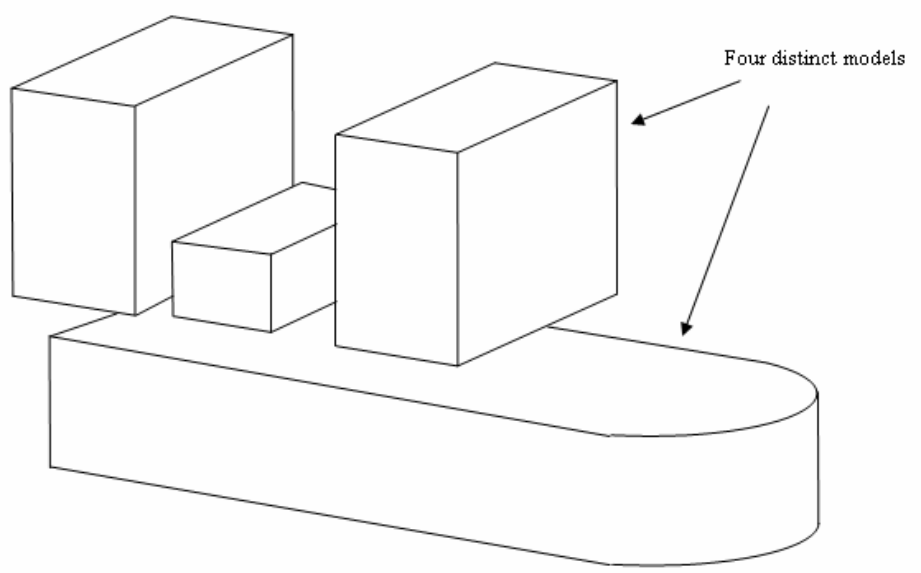


Figure 11: Additional Concave Edge and Natural Half-Space Slicing

The slicing operations are repeating for the remaining two concave edges (see Fig. 11). These operations create a total of four smaller sub-models from the original model presented to the

algorithm for conversion. Note that the larger sub-model generated in Figure 10 was not subdivided by subsequent planar slicing operations. Once a model has been cleared of all concave edges, no additional slicing operations are required and it can avoid potentially many additional slice operations.

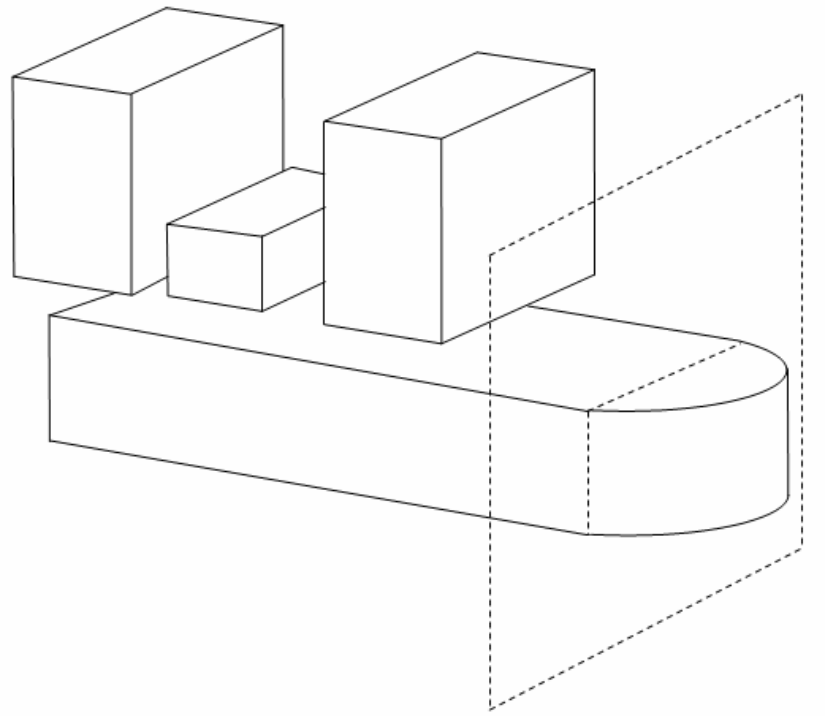


Figure 12: Identification of a Necessary Separating Half-Space

As was shown in Figures 4-6, the cylindrical surface in Figure 12 will require a separating half-space to resolve the half-space volumes ambiguities. The planar surface is constructed and slices the sub-model into an extruded semi-circle and a rectangular parallelepiped.

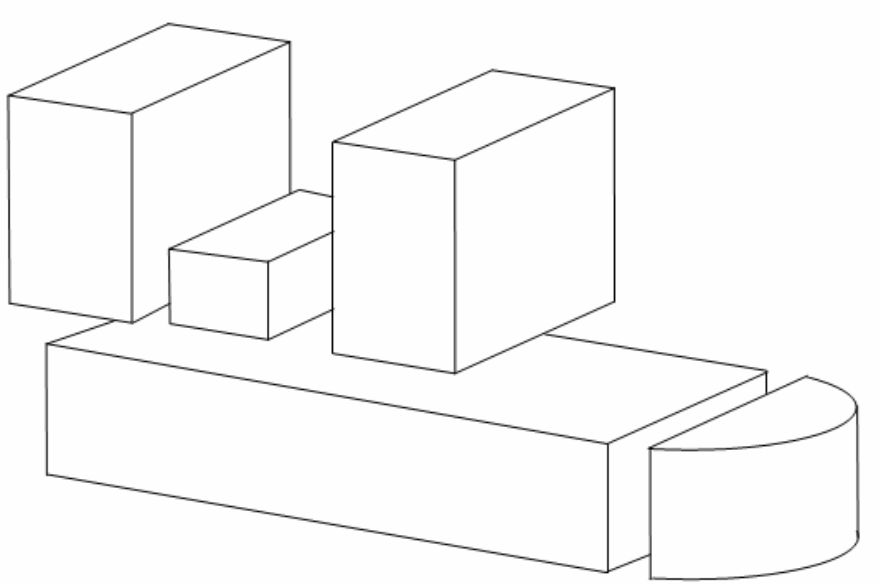


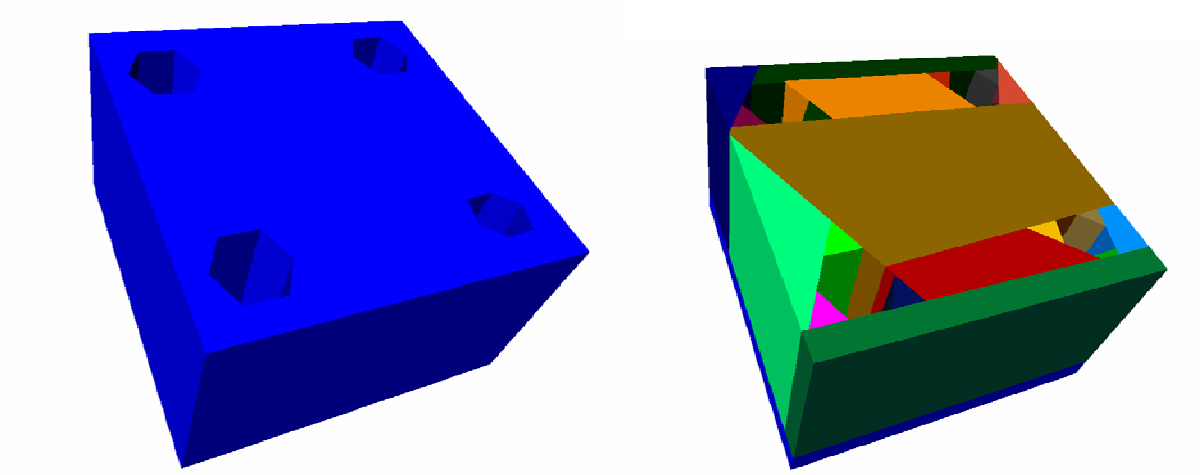
Figure 13: Completed Model Subdivision for Half-Space Representation

The algorithm terminates when all edge and ambiguous conditions are resolved. The final model representation contains five sub-models that can be successfully represented as half-space volumes (See Fig. 13).

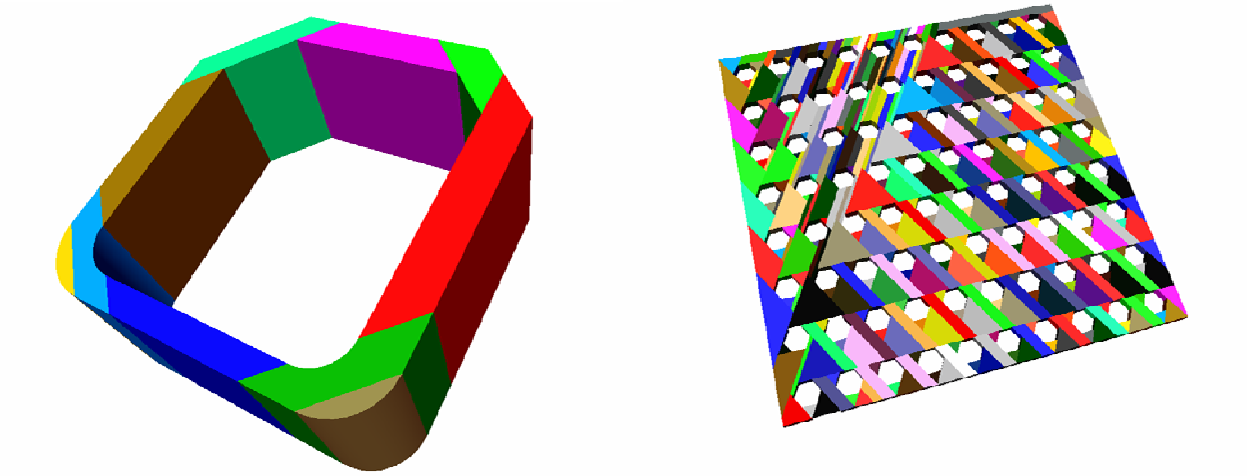
4. CONCLUSION

The algorithm described in this paper has been effective for converting a variety of modeling examples containing planar, cylindrical, and spherical surfaces for Monte Carlo analysis. Surface definitions are currently limited to those surfaces represented by quadric surface parameters, including planes, arbitrary circular and elliptical cylinders, spheres, ellipsoids, and other quadric representations.

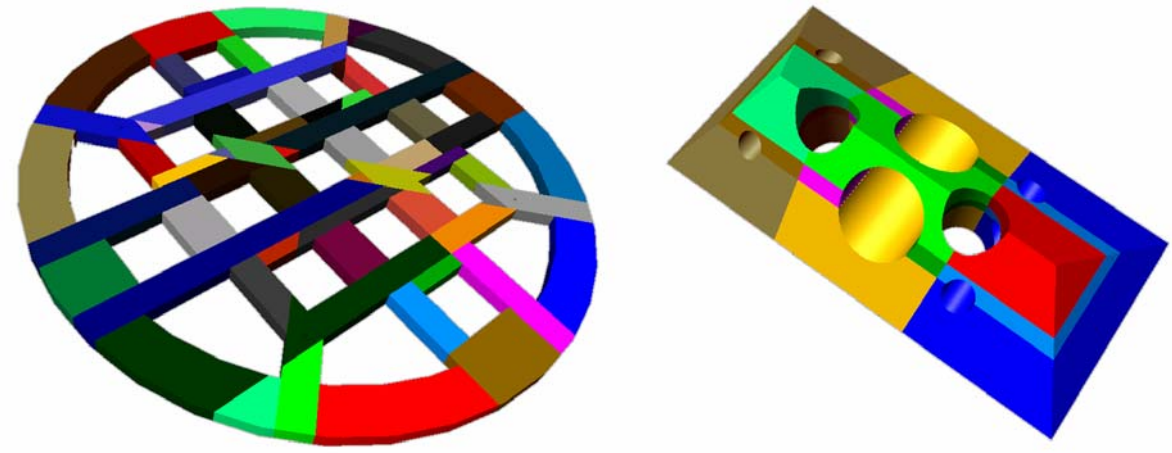
Other modeling examples from the algorithm are shown in Figures 14 – 23. Figure 14 shows an initial solid boundary representation model with four hexagonal holes in a block. Figure 15 shows the model after algorithm processing.



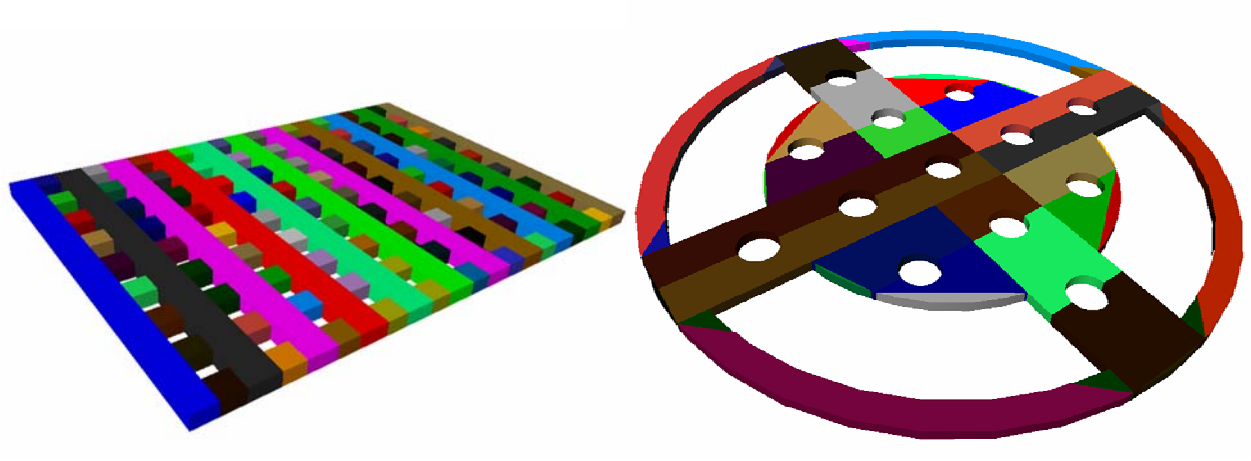
Figures 14-15



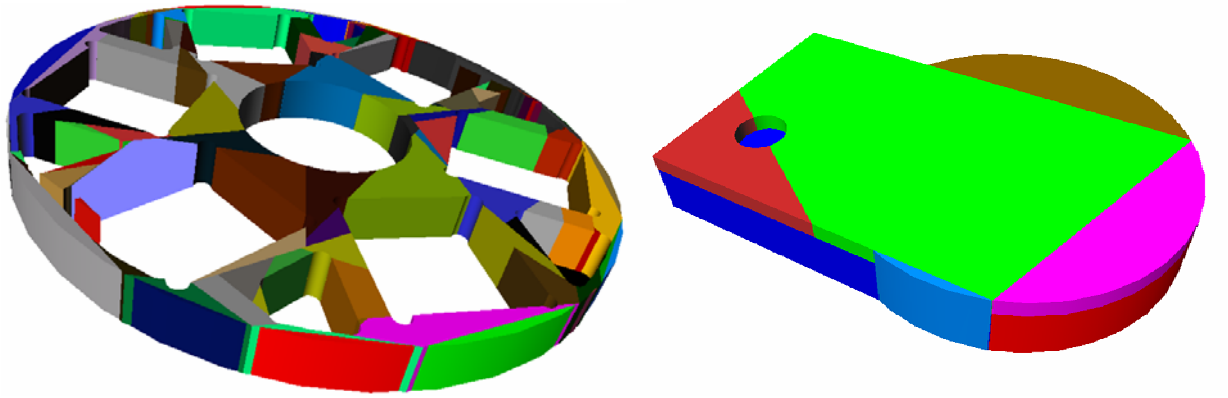
Figures 16-17



Figures 18-19



Figures 20-21



Figures 22-23

Opportunities for additional efficiency can be gained through other strategies applied to the algorithm.

One strategy addresses the reduction of the number of slicing operations with natural half-space surfaces can be obtained by choosing natural half-spaces that slice multiple concave edges simultaneously. These surfaces exist when a common plane contains protruding features that form concave edges.

Another opportunity exists by preprocessing the geometrically complex and simple sections through subdivision before the general algorithm processes the complete model. This strategy reduces the impact of geometry slices occurring in the complex section impacting the simpler sections of the model with undue slicing and subdivision. Heuristics to determine when and where these subdivisions are viable are a matter for additional research.

REFERENCES

1. M. E. Mortenson, *Geometric Modeling*, John Wiley & Sons, (1997).
2. J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, Massachusetts (1996).
3. T. M. Sutton, F. B. Brown, F. G. Bischoff, D. B. MacMillan, C. L. Ellis, J. T. Ward, C. T. Ballinger, D. J. Kelly and L. Schindler, "The Physical Models and Statistical Procedures Used in the RACER Monte Carlo Code", KAPL-4840 (1999).
4. X-5 Monte Carlo Team, "MCNP – A General Monte Carlo N-Particle Transport Code, Version 5", LA-UR-03-1987 (2003).
5. T. M. Sutton, T. J. Donovan, T. H. Trumbull, P. S. Dobreff, E. C. Caro, D. P. Griesheimer, L. J. Tyburski, D. C. Carpenter and H. Joo, "The MC21 Monte Carlo Transport Code", accepted for the *Joint Int. Conf. on Math. & Comp. and Supercomputing in Nuclear Applications*, Monterey, Calif., April 15-19 (2007).
6. N. Shaaban, F. Masuda, H. Nasif, M. Yamada, H. Sawamura, H. Morota, A. Sato, H. Iida and T. Nishitani, "A New Interface for CAD/MCNP Data Conversion", *Proc. ICONE-14 Int. Conf. on Nuc. Eng.*, ICONE14-89742, Miami, Florida, July 17-20 (2006).