# Reasoning Boolean Operation for Modeling, Simulation and Fabrication of Heterogeneous Objects

X. Hu, T. Jiang, F. Lin, and W. Sun
Department of Mechanical Engineering and Mechanics, Drexel University, Philadelphia, PA 19104

## Abstract

An approach using reasoning Boolean operation to model heterogeneous object is presented. Algorithm in the reasoning Boolean operation consists of merging and extraction operation. This algorithm models heterogeneous object at multi-volume level. Due to its CAD-based nature, the model can be implemented with advanced CAD/CAE/CAM software for integrated design, simulation, and prototyping fabrication. Example of using the developed modeling technique to construct the heterogeneous composite unit cells, to perform integrated design and analysis, and to develop a pseudo-processing algorithm for layered fabrication of heterogeneous object is also presented.

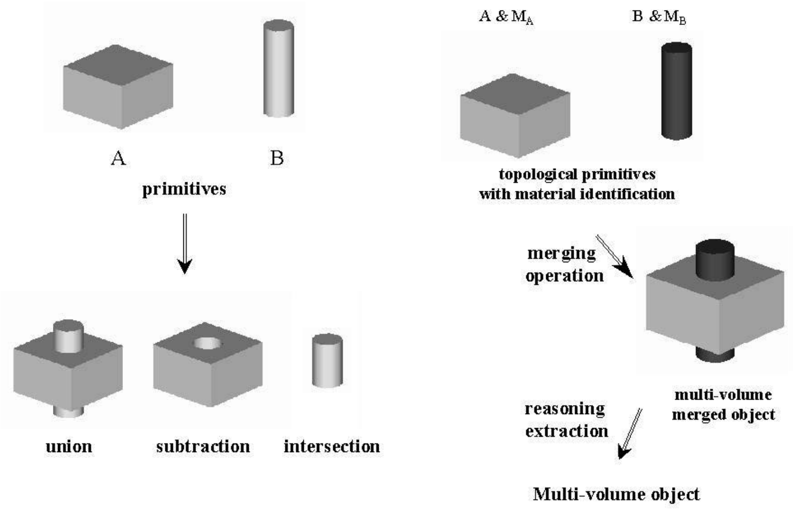
## 1. Introduction

Research on modeling heterogeneous objects has been reported only recently. For example, Cavalcanti, Carvalho and Martha[1] used intersecting multi-loop simple patches and geometric complex completeness for spatial decomposition of heterogeneous objects. Sun and Lau[2, 3] proposed a framework for developing a knowledge-enriched CAD model for solid freeform realization of heterogeneous structures. Kumar and Dutta[4, 5] presented a $r_m$-set modeling approach to construct heterogeneous objects and define processing algorithm for layered manufacturing. Different modeling approaches for constructing heterogeneous solids have also been discussed by Ashok, Kumar, and Wood[6], and Morvan and Fadel[7].

This paper presents a novel computer modeling approach to construct a CAD-based model for heterogeneous objects. The modeling algorithm is developed based on reasoning Boolean operation. The basic principle of the reasoning Boolean operation and its merging and extracting operation is described in Section 2. Application of the modeling technique to construct heterogeneous composite unit cells and to integrate with available CAD/CAE software for finite element analysis is presented in Section 3. Section 4 describes a pseudo-processing algorithm for layered fabrication of heterogeneous object. This processing algorithm is developed based on the STL data exported from the heterogeneous solid assembly. Details of the algorithm development and the generation of the processing path are presented.

## 2. Reasoning Boolean operation based heterogeneous CAD modeling

We consider that heterogeneous object consists of multi-volume topological elements (solids). Its material heterogeneity is accounted according to the material identification assigned within each element (volume). A reasoning Boolean operation algorithm which consists of reasoning merging operation and the extracting operation is developed to construct multi-volume heterogeneous object. Conventional Boolean operation only deals with the geometry of topological solids. As the result of the operation, a single-volume object is constructed through either union, subtraction, or intersection, as shown in Figure 1a. While in the proposed reasoning Boolean operation, the algorithm manipulates multi-volume topological elements through two steps: 1) the reasoning merging operation; and 2) the extracting operation. The reasoning merging operation identifies the material attributes assigned to the topological elements and compares them to decide whether they are identical and need to be merged. The extracting operation follows the Boolean merging operation to generate the needed intersecting surfaces, edges, and/or the splitting of volumes for merged elements. Material identification is consolidated for topological elements with the identical material attribute, and is retained within the new extracting volumes if they are different.



1a) Conventional Boolean operations 1b) Reasoning Boolean operations

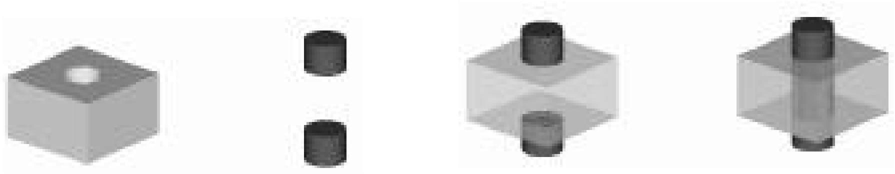
Figure 1: Comparison of conventional and reasoning Boolean operations

The algorithm of the reasoning Boolean operation for manipulating a heterogeneous object C constituted by two-volume topological elements A and B with the material identification $M_A$ and $M_B$, as shown in Figure 1b, can be described as follows:

```
If $M_A = M_B$, Then (single volume object with identical material identification)
   $C = A(M_A) + B(M_A)$                                  (conventional *union*)
      OR $C = A(M_A) - B(M_A)$                            (conventional *subtract*)
          OR $C = A(M_A) \cap B(M_A)$                     (conventional *intersect*)
Otherwise (multi-volume object with different material identification)
   $C = A(M_A) - B(M_B)$                                  ($M_A$ dominant *subtract*)
      or $C = B(M_B) - A(M_A)$                            ($M_B$ dominant *subtract*)
   OR $C = A(M_A) + \{B(M_B) - A(M_A)\}$                  ($M_A$ dominant *union*)
      or $C = B(M_B) + \{A(M_A) - B(M_B)\}$               ($M_B$ dominant *union*)
   *OR* $C = A(M_A) \cap B(M_B)$                          ($M_A$ dominant *intersect*)
      or $C = B(M_B) \cap A(M_A)$                         ($M_B$ dominant *intersect*)
   *OR* $C = \{B(M_B) - A(M_A)\} + \{A(M_A) \cap B(M_B)\}$     ($M_A$ dominant *complex_union*)
      or $C = \{A(M_A) - B(M_B)\} + \{B(M_B) \cap A(M_A)\}$    ($M_B$ dominant *complex_union*)
end if
```

Unlike the conventional Boolean operation, the reasoning Boolean operation needs to be executed according to the material-dominant information. In the example shown in Figure 1b, the Boolean operation is defined as either $M_A$-dominant or $M_B$-dominant *union*, *subtract*, and *intersect*. Two new set operations need to be introduced in the reasoning Boolean operation: 1) $M_A$-dominant *complex_union*; and 2) $M_B$-dominant *complex_union*. The operation of *complex_union* "assembles" the results of the *intersect* operation and the *subtract* operation to form a heterogeneous assembly. In the database of the assembled heterogeneous model, each solid element retains its original material identification in the resulting multi-volume model.

Results of the reasoning extracting operation for manipulating two-volume objects with different material identifications are shown in Figure 2 (for *subtract* and *union*) and Figure 3 (for *intersection* and *complex_union*).



$M_A$ dominant
***subtraction:***
**$A(M_A)$ - $B(M_B)$**

$M_B$ dominant
***subtraction:***
**$B(M_B)$ - $A(M_A)$**

$M_A$ dominant
***union:***
**$A(M_A)$ + {$B(M_B)$ - $A(M_A)$}**

$M_B$ dominant
***union:***
**$B(M_B)$ + {$A(M_A)$ - $B(M_B)$}**

Figure 2: Reasoning extracting operation for *subtract* and *union*

$M_A$ dominant
***intersection:***
**$A(M_A)$    $B(M_B)$**

$M_B$ dominant
***intersection:***
**$B(M_B)$    $A(M_A)$**

$M_A$ dominant
***complex_union:***
**{$B(M_B)$ - $A(M_A)$} + {$A(M_A)$    $B(M_B)$}**

$M_B$ dominant
***complex_union:***
**{$A(M_A)$ - $B(M_B)$} + {$B(M_B)$    $A(M_A)$}**

Figure 3: Reasoning extracting operation for *intersect* and *complex_union*

## 3. Modeling and simulation of heterogeneous composite unit cells

The reasoning Boolean operation algorithm is applied to a construct CAD-based heterogeneous composite unit cell model. Advanced CAD software, Pro/Engineer 2000 (Pro/E) by Parametric Technology Corporation[8], was used in the model construction. The procedure of generating the "assembled" heterogeneous composite unit cell model from its matrix (A and $M_A$) and fiber (B and $M_B$ ) solid elements is described in Figure 4. As shown in the figure, the matrix-dominant *subtract* cuts the fiber element from the matrix to form a matrix-dominant subtraction (matrix with cavity), while the fiber-dominant *intersect* produces a geometrically fitted fiber elements (fitted reinforcement). The fiber-dominant Boolean *complex_union* operation assembles the results of the matrix-dominant subtraction and the fiber-dominant intersection to form a heterogeneous composite unit cell model.

One of the advantages for using the presented modeling approach is that the geometry and the orientation of fiber architecture can be designed in the CAD model so that both material and geometry of composite constituents can be characterized in the composite design. Using computer rendering techniques, the design engineer can also exam the composite internal structure, pattern of the reinforcement, and fiber architecture in the design unit cell CAD-based model. This modeling technique and the ability to visual the design model is particularly useful in design complicated composite structures, such as three-dimensional textile composites with complex fiber architectures as shown in the Figure 5 for 2D woven fabric, 2D basket weave, 3D tri-axial braided, and 3D uni-directional fiber-reinforced composite structures.

After generating the CAD "assembly" of the heterogeneous composite unit cell in Pro/E assembly module, the Pro/E CAD model was transferred seamlessly into a Pro/MECHANICA (Pro/M) model for finite element analysis. Finite element meshes were generated through Pro/MECHANICA's AutoGEM function module. The meshes of fiber and matrix automatically match at their interfaces. This matching requirement is the basic requirement for finite element analysis of heterogeneous object. Often times, this requirement is difficult, if not impossible, to satisfy by using conventional finite element meshing approach in producing meshes for composites with complex three-dimensional fiber architectures. Therefore, the presented modeling technique may be very useful in generating matched meshes for analysis of heterogeneous object.

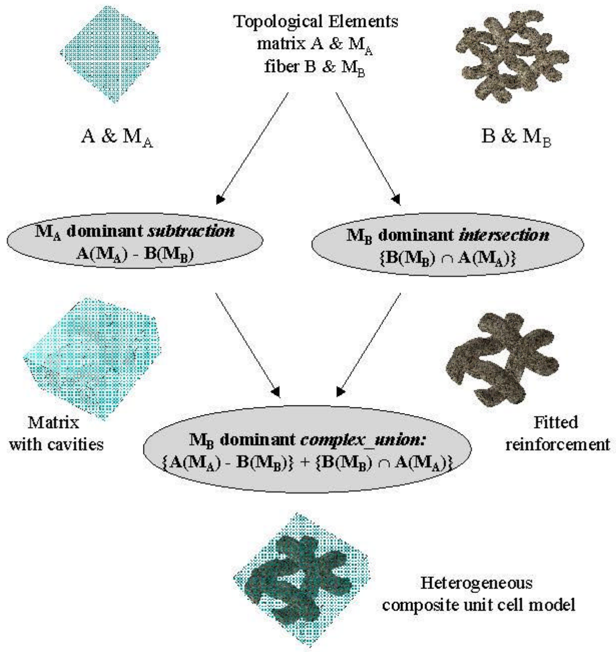


Figure 4: Procedure of constructing heterogeneous composite unit cell model

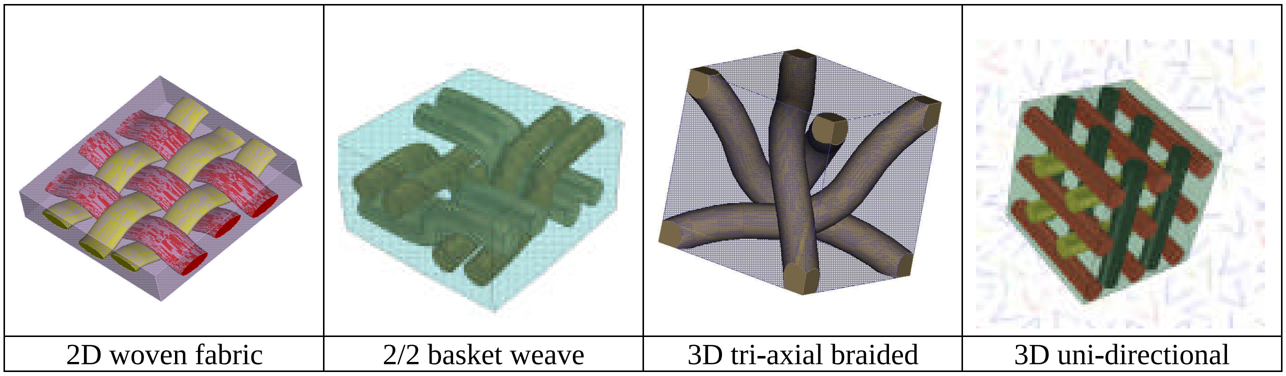
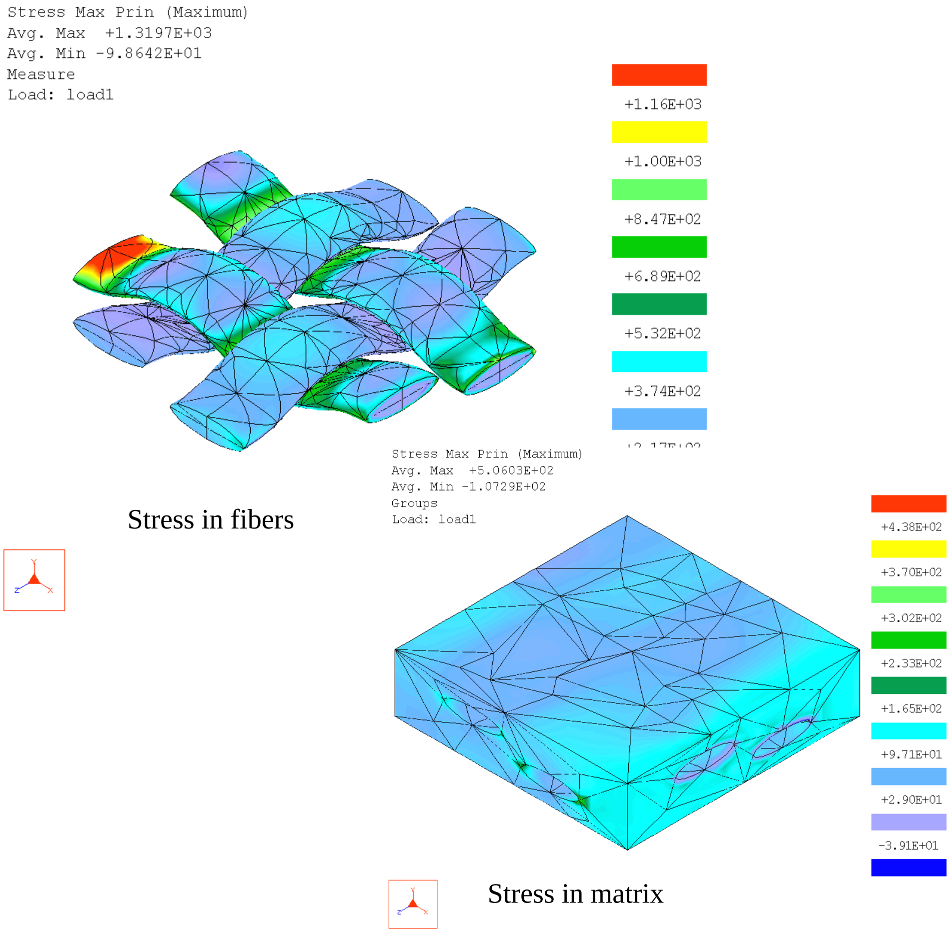

2D woven fabric | 2/2 basket weave | 3D tri-axial braided | 3D uni-directional

Figure 5: CAD models for heterogeneous composite unit cell

In the finite element analyses presented in this paper, the mechanical properties of the fiber and the matrix are assumed to be linear elastic and isotropic. The interface between the fiber and the matrix is assumed to be "perfectly" bonded throughout the model, which means that the traction and displacements are assumed to be continuous across the interface. The P-version high order interpolating polynomials was used in Pro/M for solution convergence in the analysis. In our example, 6 p-loop passes were performed before all elements satisfied the convergence condition. AutoGEM in Pro/M produced 5896 tetrahedral elements for fibers and matrix in the finite element model. Results of the finite element analyses for unit cell under in-plane tension are presented in Figure 6. Contour representations of the maximum principal stress distributed in the unit cell fibers and matrix are presented. Results obtained from the finite element analyses can be used to predict the effective properties of the heterogeneous object through composite homogeneous theory. Also, the stress and deformation predicted in the constituent materials, as shown in Figure 6, may be used in the micromechanics analysis of strength, failure, and interface mechanism of heterogeneous object.



Stress in fibers

Stress in matrix

Figure 6: Maximum principal stress in unit cell fibers and matrix (under in-plane tension)

## 4. Pseudo-processing algorithm for freeform fabrication of heterogeneous object

The general flowchart for layered freeform fabricating of heterogeneous object is shown in Figure 7. This flowchart begins with the generation of material regional files as the input data. The material regional files are generated by assigning material attribute into the STL files which were exported from the heterogeneous solid assembly in Pro/E assembly module. Since the material information can not be introduced into Pro/E's design database in the modeling stage, the material regional file serves as the interface between the heterogeneous modeling and heterogeneous fabrication.

After generating geometrical profiles of heterogeneous object, the process for slicing can be defined. Slicing can be uniform, with a constant layer thickness, or adaptive, with variable layer thickness according to the curvature of the bounding surfaces in the model. Each material region will generate the cross section contours and will form faces during the slicing process. The face is the list of edges that forms a closed loop or polygon. With scanning these polygons at each layer by parallel scan lines, the fabricating path is then generated. The pseudo-processing program is the assembly of the fabricating path after all layers of heterogeneous part are processed. Detail algorithm for the fabrication processes is described in following sub-sections.

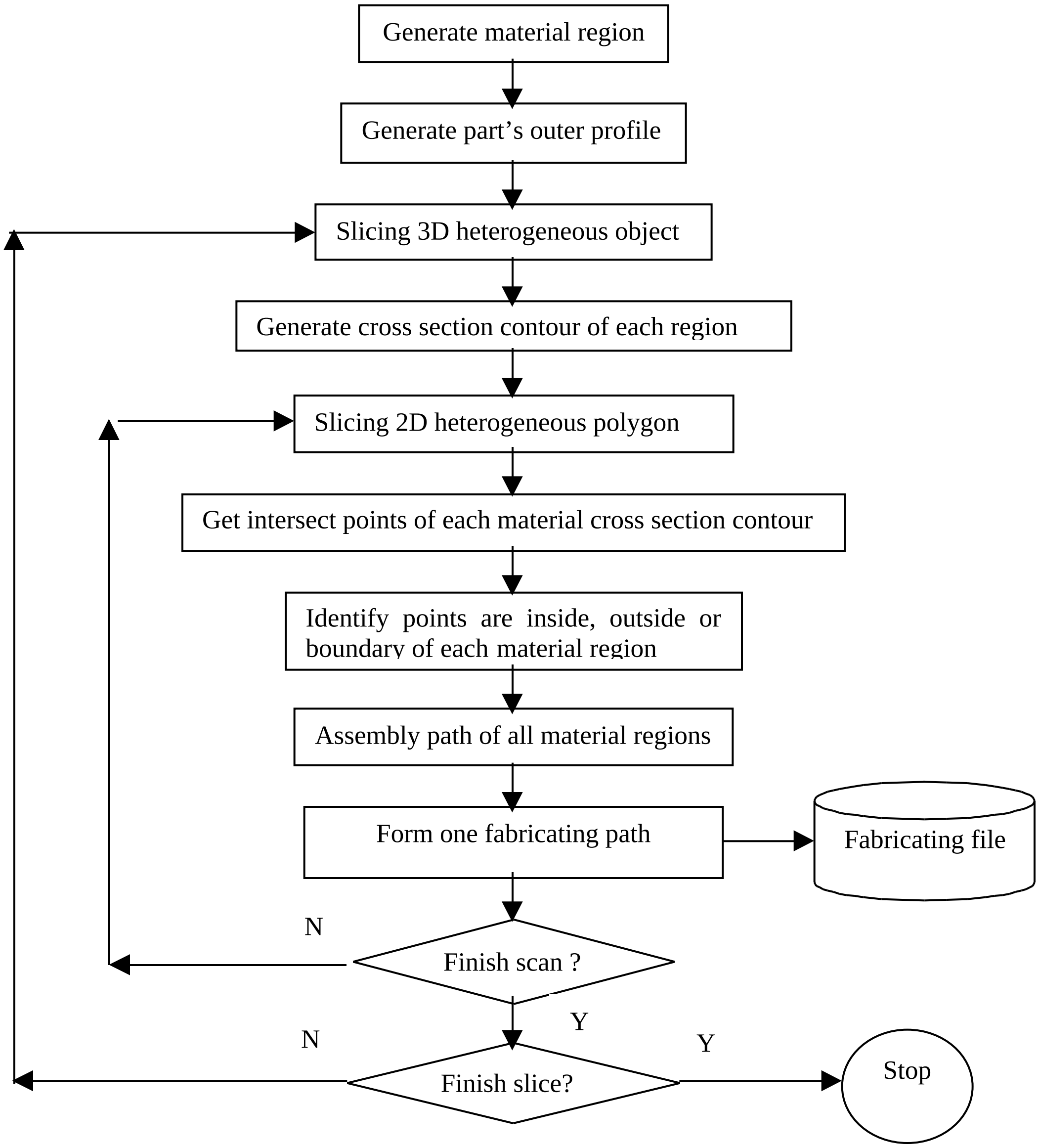


Figure 7: Flowchart of heterogeneous solid models for fabricating

### 4.1 Algorithm for slicing 3D heterogeneous object

To generate the fabrication path for heterogeneous object as shown in Figure 8, the geometrical profile of the heterogeneous object needs to be generated before slicing. The geometrical profile can be determined from the positions of the vertices. A slice is generated as the intersecting plane of the heterogeneous object with a horizontal plane at a given height. In the database, heterogeneous object consists of multi-material volumes, each material volume has a surface boundary. So slicing a heterogeneous object may generate multiple contours, as shown in Figure 8.

The algorithm for slicing heterogeneous object is defined as:

```
function  SLICE-OBJECT(MaterialRegions) returns fabricating path or failure
        inputs:  MaterialRegions,  Material Region files
        local variables:  h-part, a heterogeneous part
                          face-list, a FACE  type planes for slicing surface
                          edge-list, a EDGE type list intersecting lines
                          face, a FACE variable from list

        h-part ← INITIAL-PART(STLfiles)
        face-list ← INITIAL-PLANE(3D_MAX_PROFILE(h-part))
        loop do
                face ← PICKNODE(face-list)
                edge-list ← 3D-INTERSECTION(face, h-part)
                if edge-list is not-null
                        SLICE-POLYGON(edge-list)
        end
```

In above algorithm, Function "INITIAL-PART" reconstructs heterogeneous object based on a prior developed database, while allowing user to assign material identification to the imported STL data file. Material identification is assigned to every vertex through the function "INITIAL-PART". Function "INITIAL-PART" also generates parallel slicing planes based on the geometrical profile of the heterogeneous object. Function "3D-INTERSECTION" calculates the intersection edges of the slicing planes on each surface of the heterogeneous part.

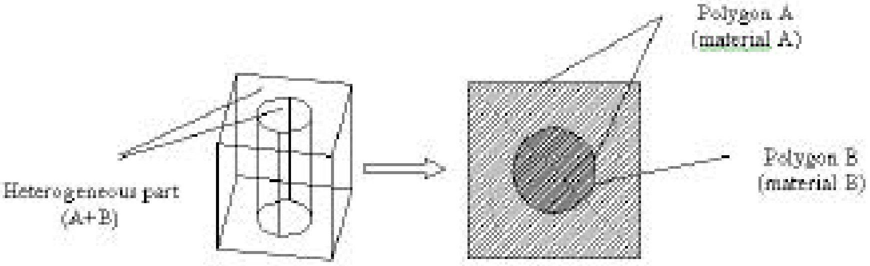


Figure 8: Slice 3D heterogeneous part

Example shown in Figure 8 illustrates that the center polygon is the interface of two material volumes, part A and part B. A polygon is the face that is constituted by a sequential of ordered edges. Edge contains the face information that is important to be used in identifying the direction of fabricating path when the edge is sliced. For instance, starting a bottom up slicing operation, the edges of the first slice contain the four edges of the cubic. Each of these four edges is the shared edge of two faces and each face has its normal direction. So if an edge is defined, the faces that the edge belong to can be retrieved. With the normal direction of these faces and the direction of scan line, the "inoutDirection" states of the intersecting vertices can be determined. In addition, since the material information is introduce through Function "INITIAL-PART" and stored in the edges, if an edge generates a new intersection vertex, the vertex will inherit the material identification of the edge. 2D polygons are then defined after slicing a 3D heterogeneous model.

### 4.2 Algorithm for slicing 2D heterogeneous polygon

In order to generate fabricating path, the polygons need to be sliced by parallel scan lines. The intersecting vertices of the scan lines with the polygons are defined as the end points of the fabricating paths. In the algorithm defined below, Function "INITIAL-LINE" will generate parallel scan lines based on the maximum and minimum coordinate of the input edge list. Function "2D-INTERSECTION" calculates the intersection vertices of the scan line and edge list. Function “SORTBYCOORDINATE” will sort these intersection vertices according to the direction of the scan line. Function “IDENTIFY-POINT” identifies if two vertices on the scan line can generate a valid fabricating path. It will end up with “true” or “false”. The path will be generated if it is “true”. The path will not be generated if it is “false”.

```
function SLICE-POLYGON(edge-list) returns success or failure
    inputs: edge-list, result from intersection of plane and heterogeneous part
    local variables: scan-line-list, a EDGE type list
                     scan-line, a EDGE type
                     intersect-vertex-list, a VERTEX type list
                     sort-point-list, a VERTEX type
                     first-point, a VERTEX type
                     second-point, a VERTEX type

    scan-line-list ← INITIAL-LINE(2D_MAX_PROFILE(edge-list))
    loop do
        scan-line ← PICKNODE(scan-line-list)
        intersect-vertex-list ← INTERSECTION(scan-line, edge-list)
        sort-point-list ← SORTBYCOORDINATE(intersect-vertex-list)
        first-point ← PICKNODE(sort-point-list)
        loop do
            second-point ← PICKNEXT(first-point)
            if IDENTIFY-POINT(intersect-vertex-list, scan-line) is true
                then GENERATE-PATH(first-point, second-point)
            first-point ← second-point
        end
    end
```

As shown in Figure 9, six intersecting vertices 1-6 will be generated after a scan line intersects polygon A and B. Because there is a member “face” in the database to store the face list of the edge where it should belong to, so it is easy to know how many faces the edge should belong to and their normal direction of the faces. As example shown in Figure 10, face 1 and face 2 are two faces of a heterogeneous object sliced by two planes, for the two edges generated on these faces, edge 1 belongs to the face 1 and edge 2 belongs to both face 1 and face 2.

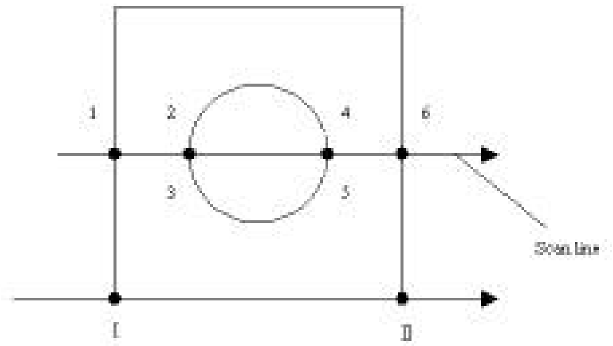


Figure 9: Slice 2D heterogeneous polygon

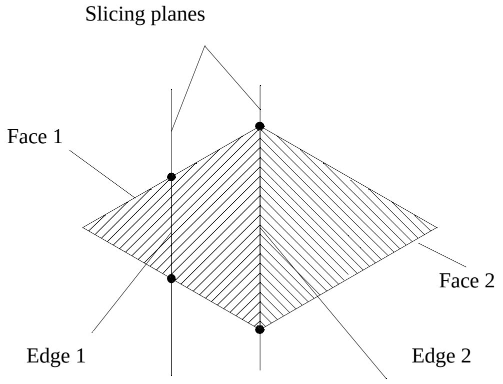


Figure 10: The relationship of edges and faces

According to the normal direction of a face that the edge belongs to, and according to the direction of the scan line, the angle between the normal direction and the direction of the scan line can be calculated. Following criteria define the direction state of the intersecting vertices:

Criterion 1: For edge lying inside the object, if the angle is greater than 90 degree, the scan line enters into the region; if angle is equal to 90 degree, the scan line is on the boundary of region; and if angle is smaller than 90 degree, the scan line goes out of the region. This criterion can be schematically explained in Figure 11. The edge v4-v5 belongs to the face v1-v2-v3, the angle of scan line and the normal direction is equal to 180, so the scan line is entering into the block from the intersection vertex of scan line and edge v4-v5. For the example shown in Figure 9, when the intersecting vertex 1 belongs to material A and its state is “in”, we define it as A1(in). Analogously, the other vertices can be defined as A2(out), A4(in), A6(out), B3(in), and B5(out), respectively.

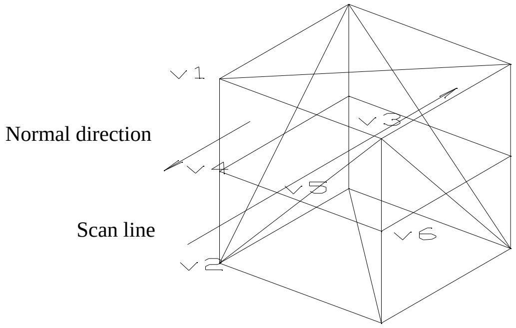


Figure 11: Identify the direction state of intersection vertices

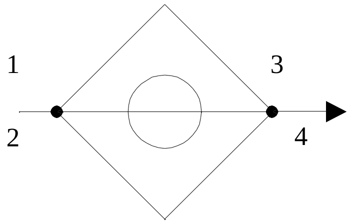


Figure 12: Redundant vertices with the same geometry and material identification

Criterion 2: For edge lying at the boundary of the object, there will be several possible solutions. Since we have stored the face with the edge information in the database, it is possible to retrieve all the faces as well as their normal directions. In the case that the angles are all greater than, or smaller than, or equal to 90 degree, the criterion 1 will apply. Otherwise, we define the intersecting vertex as a boundary vertex which its “inoutDirection” state is determined by other intersection vertexes on the scan line. As shown in Figure 9, the vertices I and II belong to material A and both are boundary vertices, so the fabricating path is generated from I to II. The path is for the boundary of the material volume A. Along the scan line direction, if the first vertex is a boundary vertex and another vertex goes out of the object, then the first vertex’s state should be changed into “in” before to generate the fabricating path. Similarly, if the first vertex is “in” and the second vertex is a boundary vertex, then the state of the second vertex will be changed into “out”.

Criterion 3: Delete the redundant vertices. As shown in Figure 12, vertices 1 and 2 (and vertices 3 and 4) have the same geometry and material identification. So only one vertex of each pair is needed in the course of fabricating path generation.

According to the above criteria, we obtain following fabrication path information in the database for the example shown in Figure 9:

Table 1: Data representation for generating fabrication path

| Vertices | V1 | V2 | V3 | V4 | V5 | V6 |
|---|---|---|---|---|---|---|
| Coordinate | X1 | X2 | X3 | X4 | X5 | X6 |
| Material | A1 | A2 | B3 | A4 | B5 | A6 |
| inoutDirection | in | out | in | in | out | out |
| … | … | … | … | … | … | … |

In above table, X is assigned with real coordinate of each vertex, the material is assigned with either “A” or “B”, and the “inoutDirection” is assigned with “in” or “out” definition.

### 4.3 Algorithm for generating fabricating path

We can assemble all vertices on a scan line to generate the fabricating path according to the information stored in Table 1. The fabricating path for heterogeneous object can be graphically illustrated in Figure 13.

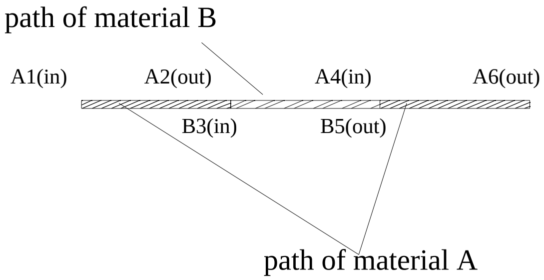


Figure 13: A fabricating path for heterogeneous object

The algorithm for layered fabrication process, as shown in Figure 14, can be defined as:

1. For the given layer $Z_i$, i=1,2,3,…k, there are heterogeneous polygons in X-Y domain.
2. For the given $Y_i$, i=1,2,3,…n, in X-Y domain, assume m number of intersection vertices $X_i$, i=1,2,3,…,m for multi-material region.
3. For the starting $X_i$, check material identification “materialID” and “inoutDirection”, fabricating always starts from entering a part.
4. If materialID=$M_i$, and inoutDirection=in, check the next vertex alone $X_i$.
5. If materialID=$M_i$, and inoutDirection=out, identify processing for material I.
6. Record SFF process path.
7. Repeat step 3 to step 6 for $X_i$, till $X_i$=$X_m$.
8. Repeat step 2 to step 7 for $Y_i$, till $Y_i$=$Y_n$
9. Repeat step 1 to step 8 for $Z_i$, till $Z_i$=$Z_k$.
10. Optimization of process path.

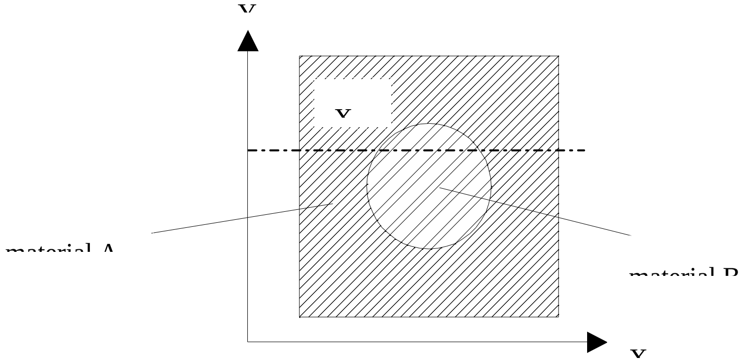


Figure 14: Layered processing algorithm

If all layered processing paths have been assembled together, we obtain the processing paths for freeform fabrication of 3D heterogeneous object as shown in Figure 10. The multi-layer path representation is shown in Figure 15.

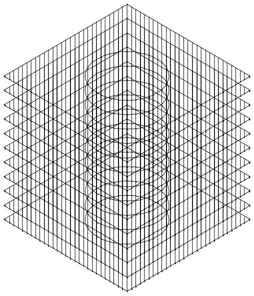


Figure 15: SFF fabricating path for heterogeneous object

## 5. Conclusion and discussions

This paper presents an approach for constructing a CAD-based heterogeneous model and a pseudo-processing algorithm for layered manufacturing of heterogeneous object. In the modeling construction, a reasoning Boolean operation algorithm is developed to manipulate heterogeneous topological elements and to construct heterogeneous CAD model. Example of using the developed modeling technique to construct heterogeneous composite is given. The implementation of the developed model with structural finite element analysis is presented. The processing algorithm for layered manufacturing of heterogeneous object is developed. The fabricating algorithm is developed based on the database of multi-volume CAD assembly, which can be generated by using the enabling CAD software, such as Pro/Engineer, I-DEAS, or UniGraphics Solutions. This offers the advantage that the developed fabrication algorithm may be implemented into the available CAD software to perform an integrated design and freeform fabrication for heterogeneous object.

## 6. Acknowledgement

We gratefully acknowledge the support from NSF 9980298 project funding to graduate student T. Jiang during the course of this research.

## References

1. Cavalcanti, P.R., Carvalho, P.C.P and Martha, L.F., “Non-manifold Modeling: An Approach Based on Spatial Subdivision,” *Computer-Aided Design*, Vol. 29, No. 3, 1997, pp. 209 – 220.
2. Sun, W. and Lau, A., "A Knowledge-enriched CAD Modeling and Solid Free-form Realization for Heterogeneous Material Structures”, Proceeding of *The Seventh International Conference on Rapid Prototyping*, March 31 - April 3, 1997, San Francisco, CA, pp. 79 – 87.
3. Sun, W., “Multi-volume CAD Modeling for Heterogeneous Object Design and Fabrication,” *Journal of Computer Science and Technology*, Vol. 15, No. 1, 2000, pp. 27 - 36.
4. Kumar, V. and Dutta, D., “An Approach to Modeling and Representation of Hetrogeneous Objects,” *J. Mechanical Design*, Vol. 120, 1998, pp. 659 – 667.
5. Kumar, V., “Solid Modeling and Algorithm for Heterogeneous Objects,” Ph.D. Dissertation, Department of Mechanical Engineering, The university of Michigan, 1999.
6. Ashok, V. Kumar and Asron Wood. "Representation and design of heterogeneous components". *Proceedings of the Solid Freeform Fabrication Symposium*, 1999, Austin, TX, pp. 179 – 186.
7. Stephane M.Morvan, Georges M.Fadel. "Heterogeneous Solids: Possible Representation Schemes". *Proceedings of the Solid Freeform Fabrication Symposium*, 1999, Austin, TX, pp. 187 – 197.
8. Pro/Engineer 2000, Parametric Technology Corporation, Waltham, Massachusetts, 2000.