BMSweep: Locating Interior Nodes During Sweeping

M. L. STATEN[1], S. A. CANANN[1,2], S. J. OWEN[1,2]

[1]*ANSYS Inc. 275 Technology Drive, Canonsburg, Pennsylvania, 15317, U.S.A.*

[2]*Department of Civil and Envirnomental Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A.*

1.0 Abstract

BMSweep is a new algorithm to determine the location of interior nodes while volume sweeping. BMSweep uses background mesh interpolation to locate interior nodes while sweeping two and one half dimensional volumes. Three dimensional volumes can be swept using BMSweep after being decomposed into two and one half dimensional volumes. The interpolation method provides for quality element creation while allowing the volume boundary to vary.

KEYWORDS: mesh generation, sweep, background mesh, hexahedra, mapped meshing, 2.5D.

2.0 Introduction

For a variety of reasons, hexahedral elements are often prefered over tetrahedral elements for use in finite element analysis[1,2]. However, unlike tetrahedral meshes, hexahedral meshes are much more constrained and, therefore, much more difficult to generate[3]. Sweeping is a technology that has received significant research in the past few years and is a method of meshing two and one half dimensional volumes with an all hex mesh [4-9]. A two and one half dimensional volume is a volume that has a topologically constant cross section along a single axis known as the "sweep axis." (Figure 1a).

The procedure used in most sweeping methods is to first identify a source area and a target area at opposite ends of the sweep axis. The source area is meshed with a quadrilateral mesh. This quadrilateral mesh is then swept through the volume towards the target area using a specified number of layers. Each quadrilateral on the source area forms a single hexahedral element in each of the layers during sweeping. If the source area is meshed with triangles, wedges (i.e. triangular prisms) are formed instead of hexahedra.

The method of sweeping can be modified to mesh more complicated geometries by allowing for multiple source and/or multiple targets [4-6]. Blacker[4] presents the Cooper Tool which is a variation of sweeping. The Cooper Tool takes a volume and divides it up into "barrels". Each barrel has a single source area and a single target area. Each barrel is meshed separately, ensuring that the boundary mesh of each barrel conforms to any adjacent barrels. Once each barrel is meshed, the final mesh is simply the combination of the elements in all of the barrels. Liu and Gadh [6] also use sweeping to mesh some of their BLOBs (Basic LOgical Bulk). Similar to the Cooper Tool, Liu and Gadh [6] decompose the volume into simpler shapes which are sent to either a sweeper or a volume mapped mesher.

One problem with the current sweeping technology is how to determine the location of the new nodes created on the interior of each barrel. Shih and Sakurai [7] present a method to determine these interior node locations; however, their method requires that both the source and target areas be planar and parallel to each other. Mingwu and Benzley [5] present another method of determining these interior node locations, but their algorithm requires a time-consuming smoothing operation after the hexahedral elements are formed[10]. In addition, even with smoothing, the quality of the resulting elements can be poor for complex models. Blacker [4] indicates that the Cooper Tool determines the interior node locations using a least square weighted residual method, however, the algorithm is not documented.

This paper presents an alternate method for determining the location of these interior nodes via background mesh interpolation. Hereafter, this algorithm will be called Sweeping via Background Mesh Interpolation (BMSweep). BMSweep provides the following benefits:

- Is General enough to sweep any two and one half dimensional volume.
- Is Orientation insensitive.
- Is computationally inexpensive.
- Does not require flat source and target areas.
- Does not require that the source and target areas be parallel.
- Does not require smoothing after the initial placement of the interior nodes.
- Does not require the source and target areas to have the same shape. (While both the source and target areas must have the same number of loops, the areas can have different surface area and shape.)
- Does not require a constant cross section shape along the sweep direction, however, constant cross section topology is required.

3.0 The Sweeping Algorithm

Before BMSweep begins to sweep through a volume, the following four assumptions are made:

1. The geometry to be swept is a two and one half dimensional volume either unaltered or created by decomposing the volume. Volume decomposition may be done manually or by employing an automated method [4,6,8,13].
2. The source area is meshed with a quadrilateral , quad/tri mixed or triangle mesh [11,12].
3. The target area is meshed with a projection of the source area mesh. Both source and target meshes must have the same mesh topology.
4. All side areas are meshed with a regular, gridded, quadrilateral, mapped mesh [13]. In addition, the mapped meshes on the side areas must conform from one side area to the next. The nodes on the side area mapped meshes are organized into “ribs”. A rib is a continuous line of edges extending from a node on the source area, through a single node on each layer and terminating at a node on the target area [4]. Figure 1a shows a two and one half dimensional volume and Figure 1b shows what the boundary ribs may look like. With the side areas mapped meshed, the boundary ribs are defined before invoking BMSweep. Interior ribs begin at the source area interior nodes and terminate at the corresponding target area nodes. The objective of BMSweep is to determine the location of the nodes on the interior ribs.

With these assumptions, BMSweep proceeds using the following steps:

1. Generate a background mesh. (see section 3.1)
2. Calculate interpolation information using the background mesh. (see section 3.2)
3. Using the background mesh and interpolation information, calculate the location of each interior node and place it on the corresponding rib. (see section 3.3)
4. Connect the nodes created in step 3 to form the new elements. (see section 3.4)

These four steps are described in the following sections.

3.1 Generation of the Background Mesh

BMSweep requires a background mesh to provide a framework for computing the locations of the nodes on the interior ribs for each layer of the mesh. The background mesh is generated by tesselating the source area's boundary nodes. Since tesselation is more conveniently and robustly done in 2D, the parametric coordinates of the source area boundary nodes may be used. For this implementation, a 2D Bowyer/Watson[16] Delaunay triangulation algorithm is used to tesselate the nodes. For example, Figure 1c shows the mesh on the source area of the volume in Figure 1a. Figure 1d shows the source area boundary nodes projected to parametric space and tesselated. For the volume in Figure 1a, Figure 1d serves as the background mesh.

The background mesh must be created using only the boundary nodes of the source area. No interior points should be added to the background mesh. Although there are interior nodes on the source area that could be inserted into the background mesh, these nodes should not be inserted. This is because the background mesh will elevated to each layer during the sweep (see section 3.3) and none of the intermediate layers have nodes yet that correspond to the source area interior nodes.

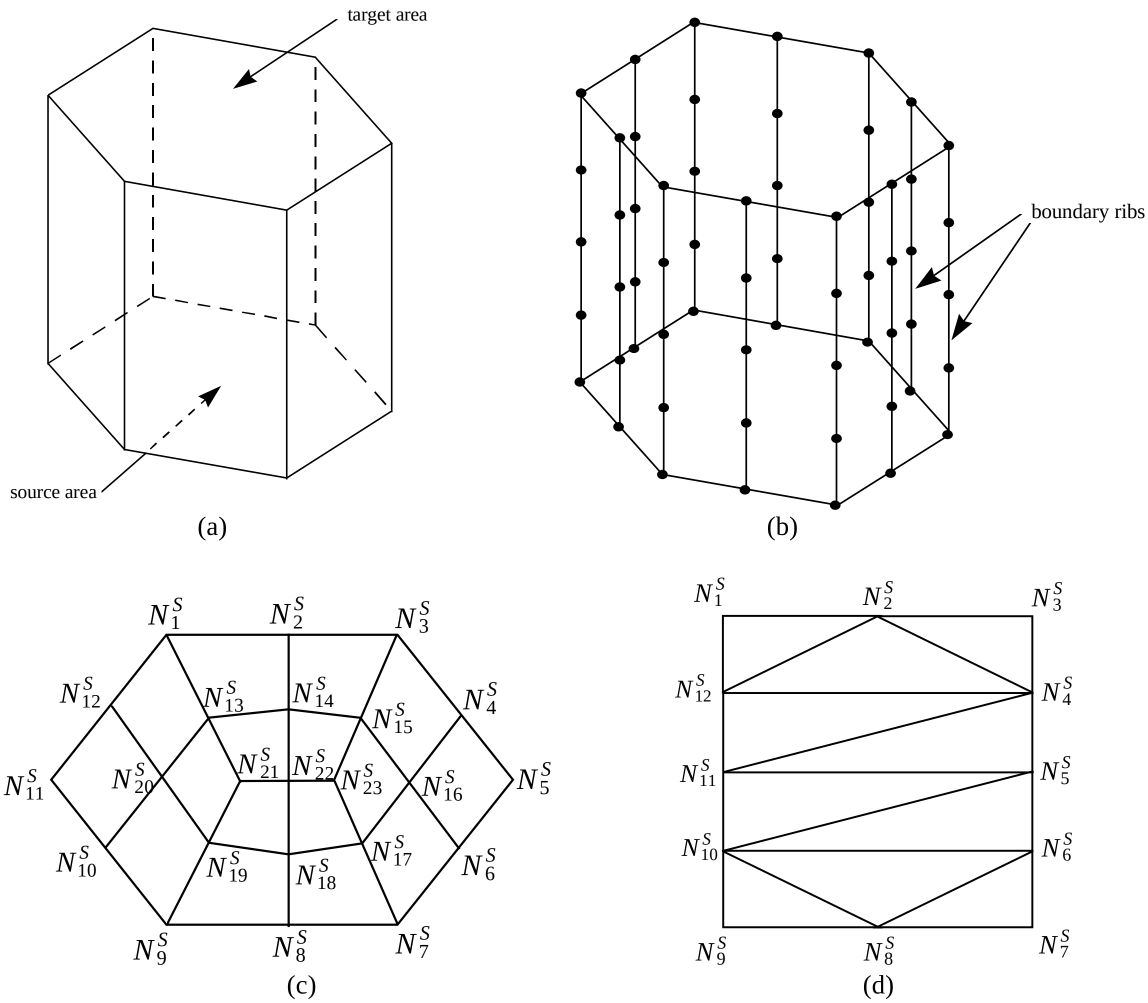

Figure 1 Example of background mesh for simple 2 ½ dimensional volume

3.2 Calculating Interpolation Information

Interpolation data is needed at each interior source area node, including:

1. Which triangle in the background mesh contains it (see section 3.2.1),
2. Its Barycentric coordinates in both the source and target areas (see section 3.2.2), and
3. Its offset distance with respect to the source and target areas. (The offset distance is the distance from an interior node to the background mesh triangle containing the node in 3d space. The offset distance is zero if both the source and target areas are flat.) (see section 3.2.3)

3.2.1 Triangle determination

With the background mesh in 2D parametric space, the 2D parametric coordinates of each interior source area node N_i^S is used to determine which background mesh triangle, T_i contains it. Once T_i is determined, a pointer to T_i is stored with N_i^S for future reference. It is likely that a single background mesh triangle will contain more than one interior source area node. In this case, each N_i^S contained in a particular triangle stores a pointer to the same background mesh triangle.

For example, Figure 2 shows the background mesh in parametric space from Figure 1 with the source area mesh projected onto it (shown with dotted lines). The triangles that contain each of the interior source area nodes (N_{13}^S - N_{23}^S) are determined using a 2D triangle node location algorithm [14,15] . For example, triangle T_{13} contains node N_{13}^S, triangle T_{17} contains node N_{17}^S, and so forth.

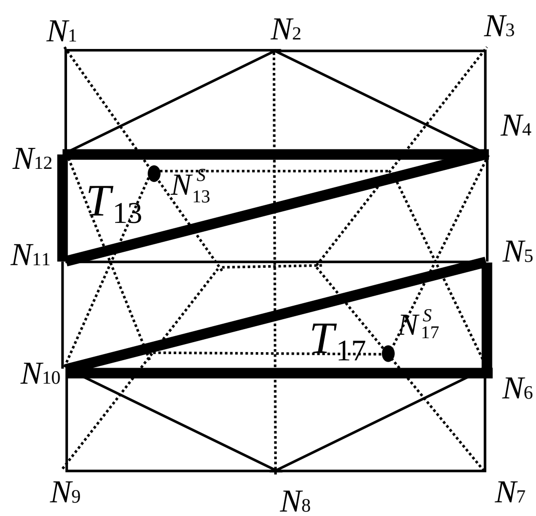

Figure 2 Background mesh with source area mesh connectivity (dotted lines) projected to parametric space

3.2.2 Barycentric Coordinates

Barycentric coordinates with respect to the background mesh triangles are used to locate each interior rib node. Two sets of barycentric coordinates are needed for each source area interior node N_i^S:

1. $\{a_i^S, b_i^S, b_i^S\}$: Barycentric coordinate of the uv location of N_i^S with respect to the <u>source</u> area uv locations of the corner nodes of the background mesh triangle T_i , and
2. $\{a_i^T, b_i^T, c_i^T\}$: Barycentric coordinates of the uv location of the cooresponding target area node N_i^T with respect to the <u>target</u> area uv locations of the corner nodes of the background mesh triangle T_i .

At each layer, the barycentric coordinates used to locate the new interior point are a linear interpolation of $\{a_i^S, b_i^S, c_i^S\}$ and $\{a_i^T, b_i^T, c_i^T\}$.

If the shape of the source and target areas are identical, then these two sets of barycentric coordinates will be identical. If, however, the shape of the source area varies at all from the shape of the target area, these barycentric coordinates could be quite different. In fact, if the source and target areas have drastically different shapes, then the background mesh triangle, T_i , after being elevated to the target area, may not contain N_i^T at all, causing one or more of the barycentric coordinates $\{a_i^T, b_i^T, c_i^T\}$ to be negative. This is perfectly acceptable. The barycentric coordinates of the N_i^S and N_i^T must be with respect to the same background mesh triangle, T_i since each interior node is located using a linear interpolation of $\{a_i^S, b_i^S, c_i^S\}$ and $\{a_i^T, b_i^T, c_i^T\}$.

3.2.3 Offset Distance

If either the source area or target area is not flat, then offset distances are needed to locate the interior nodes. Offset distance interpolation data is needed to catch source and target area curvature that is not captured by the background mesh. Since only boundary nodes are used to generate the background mesh, the background mesh only catches the curvature that is present on the boundary of the source and target areas (see Figure 9).

The offset distance is the distance, in 3d space, from an interior node to the background mesh triangle, T_i , containing the node. Two offset distances, d_i^S and d_i^T , are needed for each interior source area node. Let $\mathbf{P}_i^S$ be the location of the source area interior node, N_i^S , and $\mathbf{P}_i^T$ be the location of the target area node, N_i^T , corresponding to N_i^S . Let $(\mathbf{P}_i^{S1}, \mathbf{P}_i^{S2}, \mathbf{P}_i^{S3})$ be the source area xyz corner node coordinates of the background mesh triangle T_i . Let $(\mathbf{P}_i^{T1}, \mathbf{P}_i^{T2}, \mathbf{P}_i^{T3})$ be the corresponding target area xyz corner node coordinates of the background mesh triangle T_i . Offset distances d_i^S and d_i^T are determined using Equations 1 & 2. If the source area is flat, d_i^S will be zero. Likewise if the target area is flat, d_i^T will be zero.

$$d_i^S = \left\| (a_i^S \bullet \mathbf{P}_i^{S1} + b_i^S \bullet \mathbf{P}_i^{S2} + c_i^S \bullet \mathbf{P}_i^{S3}) - \mathbf{P}_i^S \right\| \quad [1]$$

$$d_i^T = \left\| (a_i^T \bullet \mathbf{P}_i^{T1} + b_i^T \bullet \mathbf{P}_i^{T2} + c_i^T \bullet \mathbf{P}_i^{T3}) - \mathbf{P}_i^T \right\| \quad [2]$$

3.3 Locating the Nodes on each Layer

Once the interpolation information is calculated, the location on each layer, $\mathbf{P}_i^L$, of each interior node, N_i^L, can be determined. For each new interior node N_i^L, four steps are required to determine its location. The first step is to linearly interpolate the Barycentric coordinates and offset distances calculated from the source and target areas. That is, let $\{a_i^L, b_i^L, c_i^L\}$ be the linear interpolation of $\{a_i^S, b_i^S, c_i^S\}$ and $\{a_i^T, b_i^T, c_i^T\}$ for node *i* on layer *L*. Also, let d_i^L be the linear interpolation of d_i^S and d_i^T. If the layers during the sweep are evenly spaced, the interpolation parameter can simply be the layer number. Otherwise a distance along the sweep path over the total sweep distance can be used as the interpolation parameter.

The second step is to elevate the background mesh triangle, T_i, containing the corresponding source area interior node to the current layer *L*. This is done by retrieving the corresponding boundary nodes at layer *L* from the boundary ribs that originate at the nodes defining T_i on the source area (see Figure 3). Let $(\mathbf{P}_i^{L1}, \mathbf{P}_i^{L2}, \mathbf{P}_i^{L3})$ be the locations of these three nodes defining the triangle T_i at layer L.

The third step is to define $\vec{\mathbf{V}}_i^L$ which is used with the offset distance d_i^L to specify the location of N_i^L to compensate for source and target area curvature. $\vec{\mathbf{V}}_i^L$ is calculated using the following:

$$\vec{\mathbf{V}}_i^L = a_i^L \bullet \vec{\mathbf{V}}_i^{L1} + b_i^L \bullet \vec{\mathbf{V}}_i^{L2} + c_i^L \bullet \vec{\mathbf{V}}_i^{L3} \quad [3]$$

where $\left[\vec{\mathbf{V}}_i^{L1}, \vec{\mathbf{V}}_i^{L2}, \vec{\mathbf{V}}_i^{L3}\right]$ are unit vectors in the direction of the sweep at $(\mathbf{P}_i^{L1}, \mathbf{P}_i^{L2}, \mathbf{P}_i^{L3})$ (see Figure 3).

The final step is to use the following equation to determine the location, P_i^L, of interior node, N_i^L, on layer *L* of an interior rib.

$$\mathbf{P}_i^L = \left(a_i^L \bullet \mathbf{P}_i^{L1} + b_i^L \bullet \mathbf{P}_i^{L2} + c_i^L \bullet \mathbf{P}_i^{L3}\right) + d_i^L \bullet \vec{\mathbf{V}}_i^L \quad [4]$$

Equation 4 calculates the final location of the interior nodes. No smoothing is required. Equation 4 interpolates the location of the interior nodes directly from the location of the boundary nodes on the current layer *L*. The locations of the nodes on the layers directly above and below a particular layer have no effect on the node locations of that layer. As a result, the locations of the nodes on each layer twist, turn, and rotate along with the boundary nodes of that layer completely independent of the volume's global orientation. This allows BMSweep to sweep volumes that have dramatic twists, turns and changes in cross-sectional area (see examples in section 4.0).

As the interior nodes are created, they are stored in order of increasing layers on each interior rib so as to facilitate element creation.

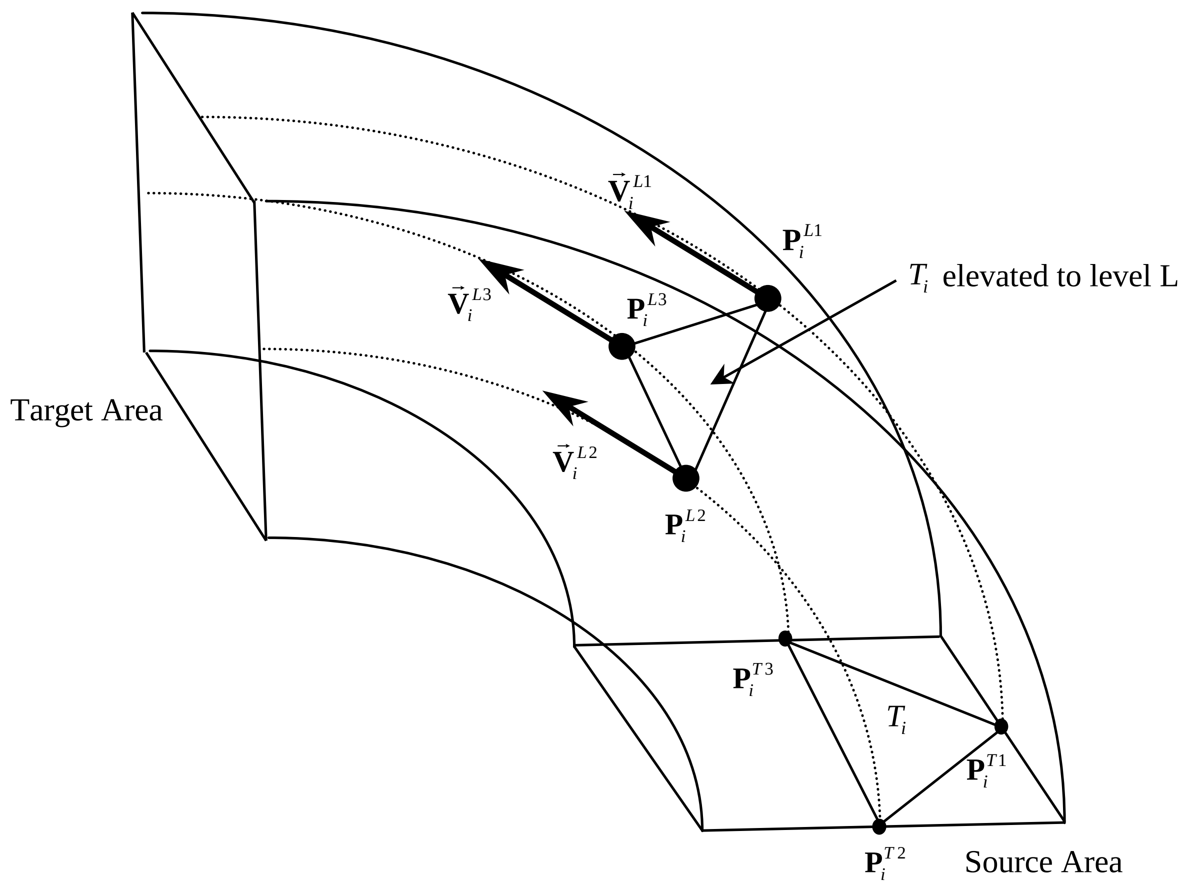

Figure 3 Vectors used to calculate node locations at layer L

3.4 Creating the new Elements.

Once all of the interior nodes have been located and placed on the interior ribs, the formation of the elements can begin. Each facet on the source area is individually processed by determining the nodes on the next layer that are on the same rib as each facet node. This process is repeated for each source area facet on each layer until all volume elements are created.

4.0 Examples

Figure 4 through Figure 11 are examples of volumes swept using BMSweep.

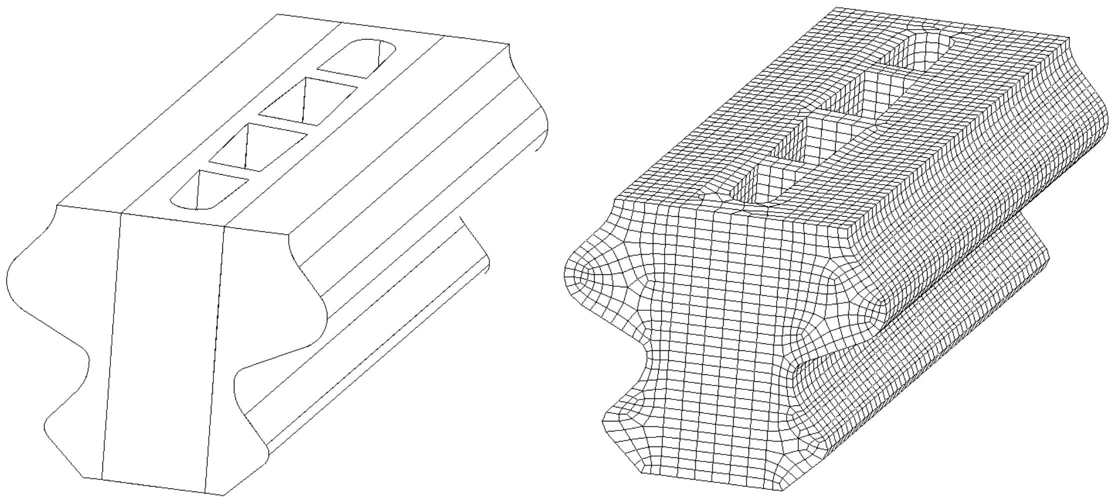

Figure 4 *BMSweep sweeping three adjacent volumes in different directions. CAD model courtesy of Pratt & Whitney*

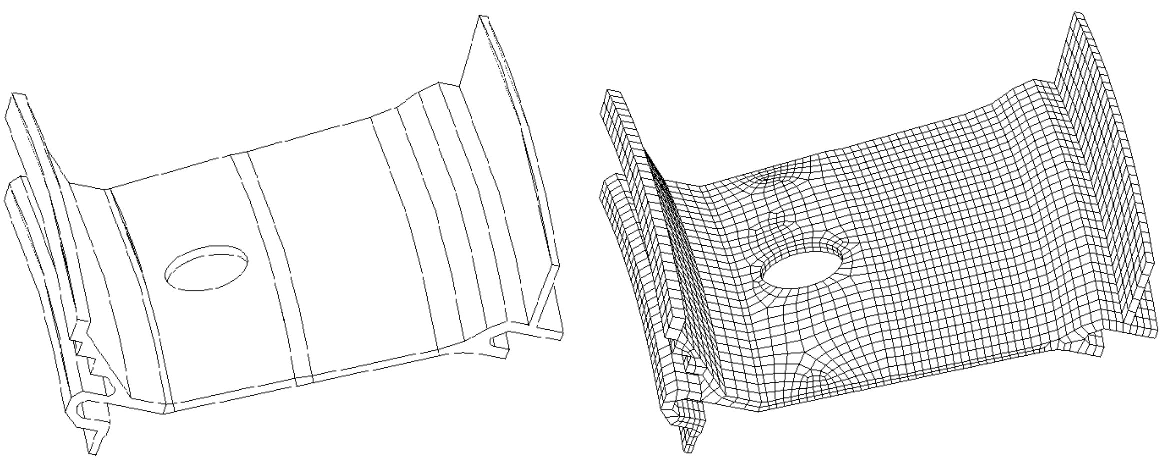

Figure 5 *BMSweep sweeping six adjacent volumes in different directions . CAD model courtesy of Pratt & Whitney*

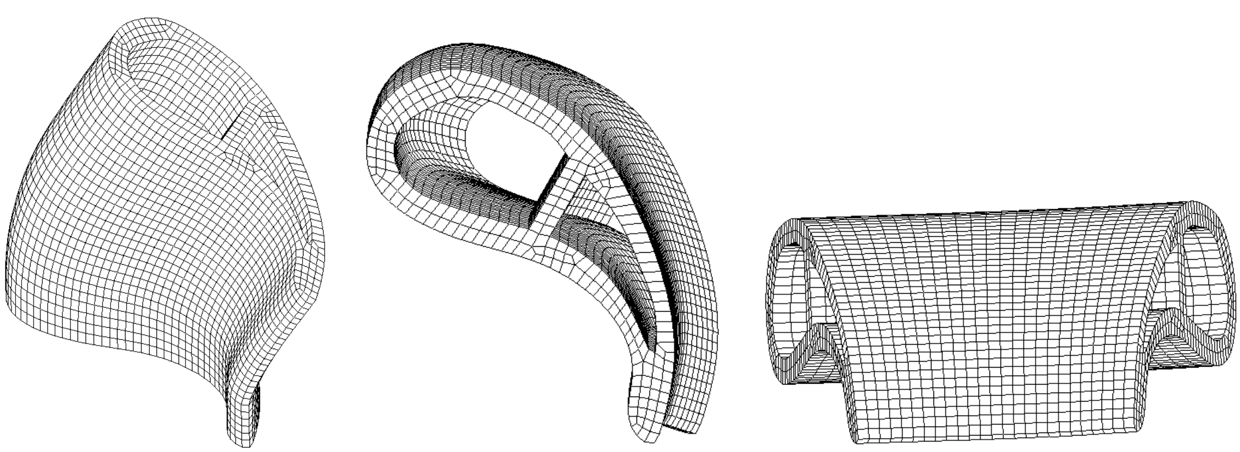

Figure 6 *BMSweep sweeping volume with irregular sweep direction*

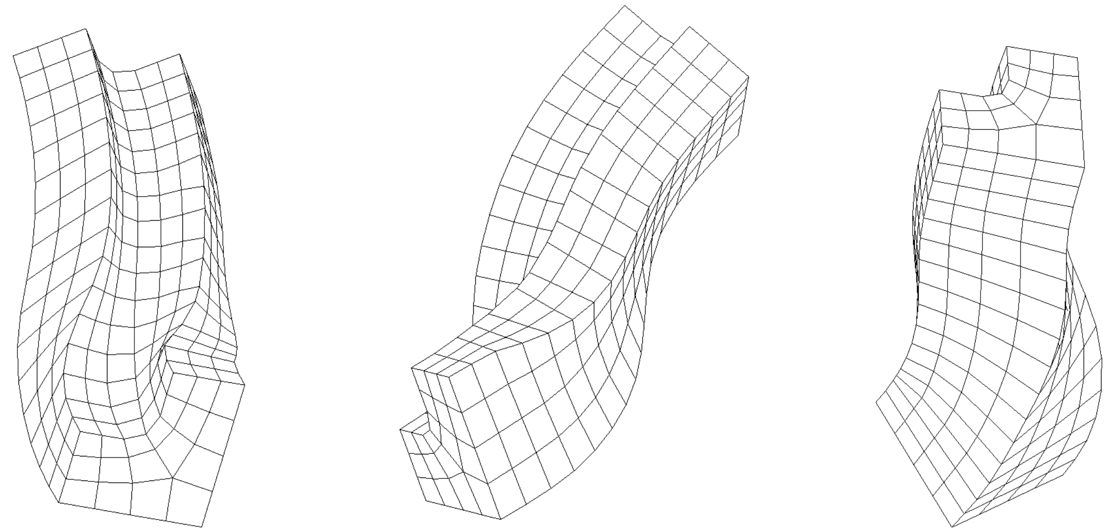

Figure 7   BMSweep sweeping volume with twisted and curved sweep direction

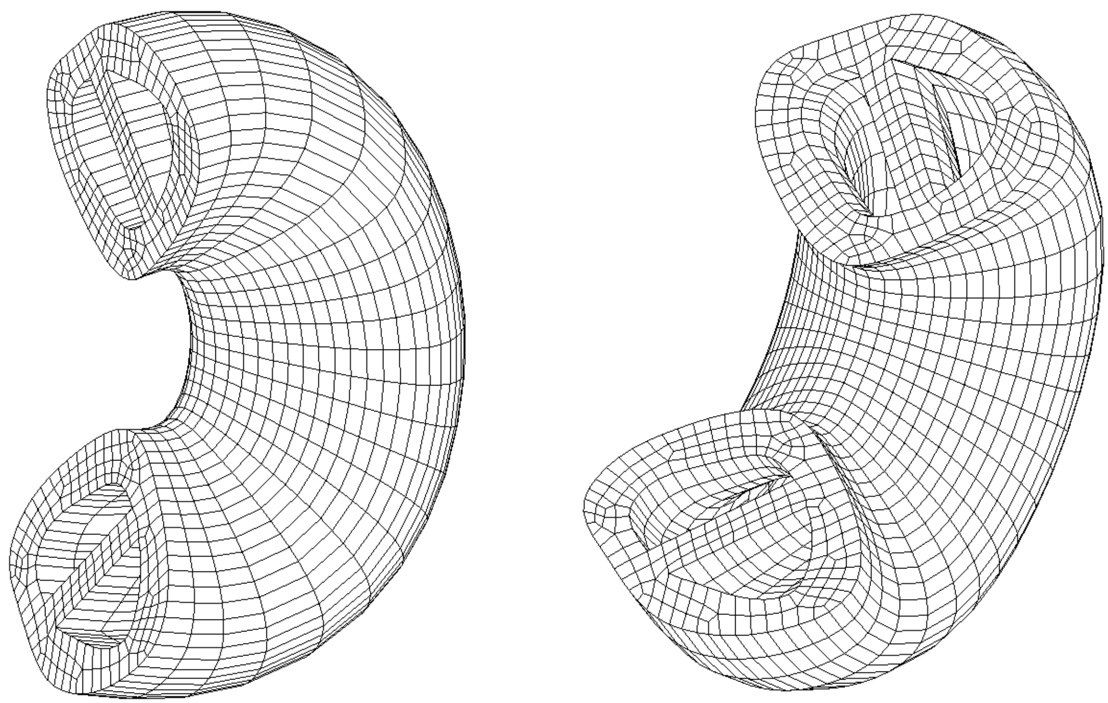

Figure 8   BMSweep sweeping thin walled volume with curved sweep direction

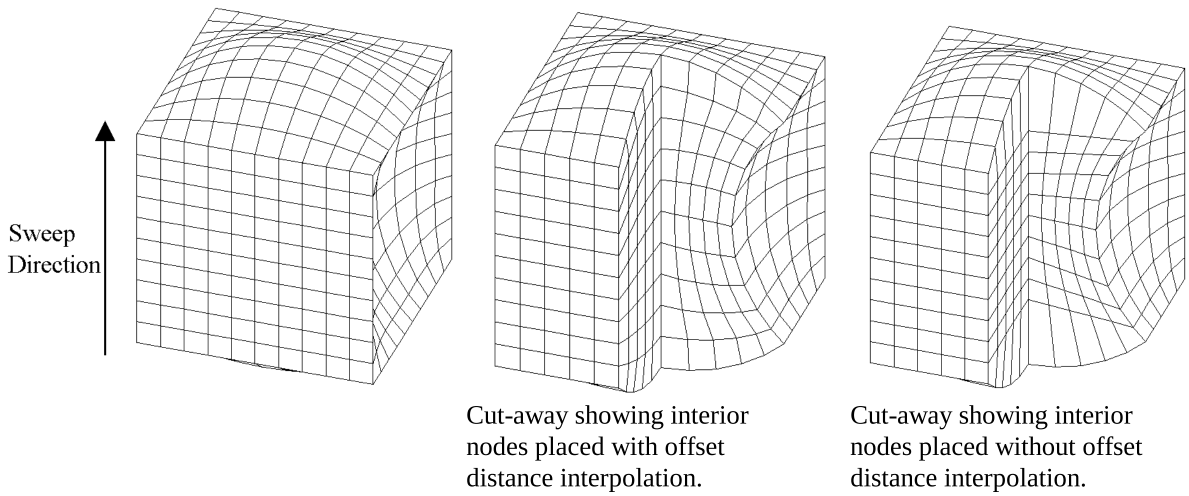

Figure 9   *BMSweep sweeping volume using offset distance interpolation*

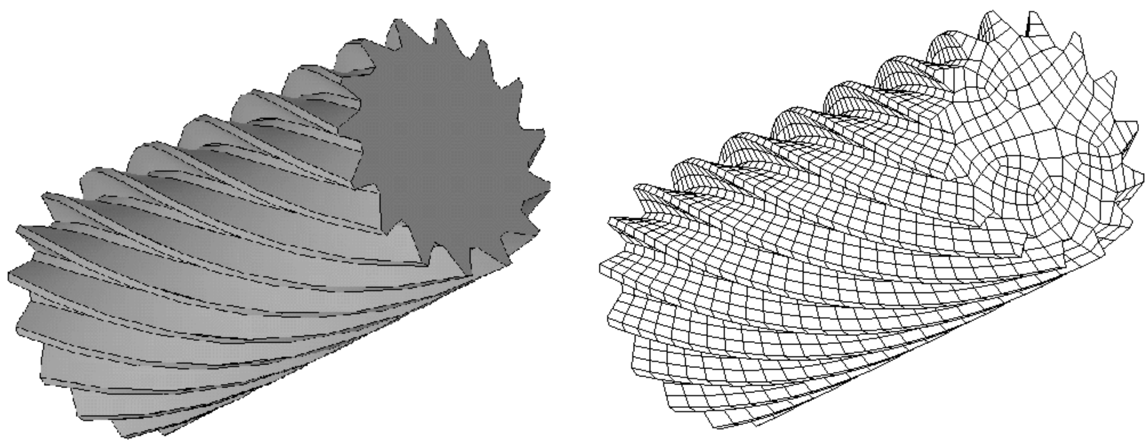

Figure 10   *BMSweep sweeping volume with 150 degree rotation in sweep direction*

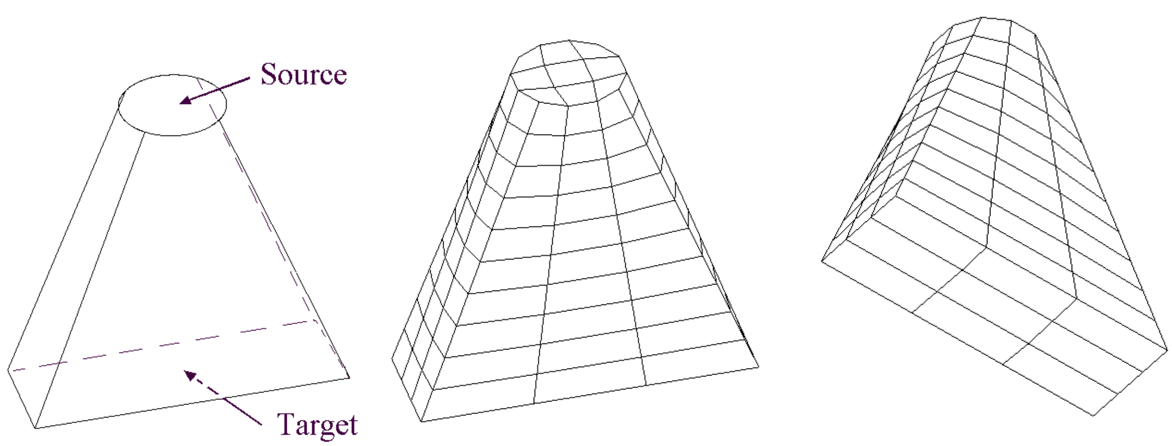

Figure 11   *BMSweep sweeping volume with circular source and rectangular target*

5.0 Conclusions

BMSweep, a method for determining interior node locations during sweeping, has been presented. BMSweep uses background mesh interpolation to locate the interior nodes. BMSweep is orientation insensitive and meshes two and one half dimensional volumes without requiring flat or parallel faces. In addition, the cross section of the sweep can twist, turn, and rotate freely. The shape of the cross section can also vary along the sweep path as long the cross section remains topologically constant. The algorithm has been extensively tested for robustness and has proven able to mesh a large variety of complex models. Coupled with volume decomposition, BMSweep can be used to mesh a large class of three dimensional volumes.

6.0 References

[1] Cifuentes, A. O. and A. Kalbag, 1992, "A Performance Study of Tetrahedral and Hexahedral Elements in 3-D Finite Element Structural Analysis." *Finite Elements in Analysis and Design*, Vol. 12, pp. 313-318.

[2] Benzley, S. E.; Perry, E.; Merkley, Karl; Clark, B. and Sjaardema, G., 1995, "A Comparison of All-Hexahedral and All-Tetrahedral Finite Element Meshes for Elastic and Elasto-Plastic Analysis," *Proceedings, 4th International Meshing Roundtable, Sandia National Labratories*, October 1995, pp. 179-191.

[3] Mitchell, S. A., 1996, "A Characterization of the Quadrilateral Meshes of a Surface Which Admit a Compatible Hexahedral Mesh of the Enclosed Volume." *Proceedings, 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS '96), Lecture Nodes in Computer Science 1046*, Springer, pp. 465-476.

[4] Blacker, T., 1996, "The Cooper Tool," *Proceeding, 5th International Meshing Roundtable 96*, pp. 13-29.

[5] Mingwu, L. and Benzley, S. E., 1996, "A Multiple Source and Target Sweeping Method for Generating All Hexahedral Finite Element Meshes," *Proceedings, 5th International Meshing Roundtable 96*, pp. 217-225.

[6] Liu, S. S. and Gadh, R., 1997, "Automatic Hexahedral Mesh Generation by Recursive Convex and Swept Volume Decomposition," *Proceeding, 6th International Meshing Roundtable 97*, pp. 217-231.

[7] Shih, B. and Sakurai, H., 1997, "Shape Recognition and Shape-Specific Meshing for Generating All Hexahedral Meshes," *Proceedings, 6th International Meshing Roundtable 97*, pp. 197-209.

[8] White, D. W., Mingwu, L., and Benzley, S. E., 1995, "Automated Hexahedral Mesh Generation by Virtual Decomposition," *Proceeding, 4th International Meshing Roundtable 95*, pp. 165-176.

[9] Chiba, N., Nishigaki, I., Yamashita, Y., Takizawa, C., and Fujishiro, K., 1996, "An Automatic Hexahedral Mesh Generation System Based on the Shape-Recognition and Boundary-Fit Methods," *Proceedings, 5th International Meshing Roundtable 96*, pp. 281-290.

[10] Canann, S. A., Tristano, J. R., and Staten, M. L., 1998, "An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes," *Proceedings, 7th International Meshing Roundtable 98*.

[11] Tristano, J. R., Owen, S. J., and Canann, S. A., 1998, "Advancing Front Surface mesh Generation in Parametric Space Using Riemannian Surface Definition," *Proceedings, 7th International Meshing Roundtable 98*.

[12] Owen, S. J., Staten, M. L., Canann, S. A., and Saigal, S., 1998, "Advancing Front Quadrilateral Meshing Using Triangle Transformations," *Proceedings, 7th International Meshing Roundtable 98*.

[13] White, D. R., 1996, "Automatic, Quadrilateral and Hexahedral Meshing of Pseudo-Cartesian Geometries using Virtual Decomposition," Master's Thesis, Brigham Young University, August 1996.

[14] Mirante, A. and Weingarten, N., 1982, "The radial sweep algorithm for constructing triangulated irregular networks," *IEEE Computer Graphics and Applications*, Vol. 2, No. 3, May, pp. 11-21.

[15] Lee, D. T. and Schacter, B. J., 1980, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Computer and Information Sciences*, Plenum Press, New York, London, Vol. 9, No. 3, June, pp. 219-242.

[16] Watson, D. F., 1981, "Computing the Delaunay tesselation with application to Voroni Polytopes," Computer Journal, Vol. 24, pp. 167-172.