THE CLEAVE AND FILL TOOL: AN ALL-HEXAHEDRAL REFINEMENT ALGORITHM FOR SWEPT MESHES

Michael Borden[1], Steven Benzley[2], Scott A. Mitchell[3], David R. White[4], Ray Meyers[5]

[1]Brigham Young University, Provo, UT, U.S.A. bordmic@et.byu.edu
[2]BrighamYoung University, Provo, UT, U.S.A. seb@byu.edu

ABSTRACT

Sweeping algorithms provide the ability to generate all hexahedral meshes on a wide variety of three-dimensional bodies. The work presented here provides a method to refine these meshes by first defining a path through either the source or the target mesh and next by locating the sweeping layer to initiate the refinement. A major contribution of this work is the ability to automatically find a minimal distance path through the target or source mesh. The refinement is accomplished by using the pillowing procedure as proposed by Mitchell. [1]

Keywords: mesh generation, hexahedral meshing, refinement, sweeping, 2 ½ -D

1. INTRODUCTION

Three-dimensional finite element analysis (FEA) is an important design tool for scientists and engineers. Before the analysis can begin, a mesh must be created for the model. There is currently significant research being devoted to the generation of such meshes. Tetrahedral meshes are well developed and have been incorporated in numerous software packages. Hexahedral meshes provide some advantages over tetrahedral meshes but are currently more restrictive in the geometrical shapes they can fill. [2][3][4]

Sweeping algorithms provide the ability to generate all hexahedral meshes on a wide variety of three-dimensional bodies. Significant advances have been made in such algorithms. Sweeping schemes generally project a two-dimensional unstructured quadrilateral mesh from a source surface to a target surface. These algorithms can accommodate non-planar, non-parallel source and target surface and variable cross-sectional areas [5] as well as multiple source and target surfaces.[6][7] Recently, a novel technique to attach additional bodies to linking surfaces has been developed. [8]

Currently, however, most sweeping algorithms require that the mesh on linking surfaces be structured. For some volumes, particularly those with non-parallel source and target surfaces and variable cross-sectional areas, this constraint can lead to a large difference in interval size on opposing curves of the linking surfaces. In some cases this will lead to a mesh with undesirable quality. This paper presents a new algorithm that allows a swept mesh to be modified in a way so that the linking surfaces take on an unstructured configuration. This allows the mesh to be modified, thereby, improving quality.

[3] samitch@sandia.gov
[4] drwhite@sandia.gov
[5] rjmeyer@sandia.gov
Scott Mitchell, David White, and Ray Meyers work at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL8500.

2. DEFINITION OF TERMS

A sketch of a simple all-hexahedral swept three-dimensional mesh is given in Figure 1.

The defining terms of this process include the *Source and Target Surfaces*, which are respectively the upper and lower surfaces in the figure. To allow sweeping, these surfaces must have topologically similar meshes. Though these meshes must be similar, they can be either structured or unstructured (as shown).

The *linking surfaces* are the often referred to as "side-walls." They form the connecting surfaces between the source and target. The *trajectory* as shown in Figure 1 is the defined direction of sweep.

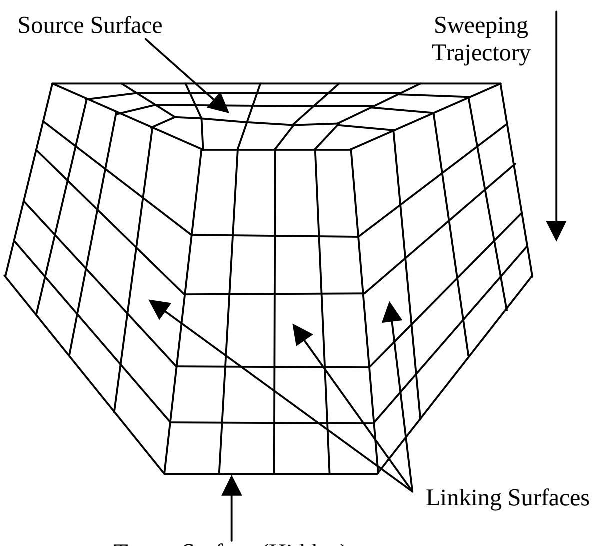


Figure 1. Definition of Terms for Sweeping

Cleaving, as used in this work, means the separation of the original mesh, on a specified plane, between existing element faces. This process is depicted in Figure 2.

The *cleaved path* is a directed line that partially defines an edge of the plane on which the separation will take place. This path is generated along existing element edges within the meshed volume. An example of such a path that has been projected onto a source surface is given in Figure 3. Note that there are numerous paths that can be defined. Also note that the path has the same connectivity on any element layer within the sweeping volume. The *cleaved plane* is defined by specifying the layer within the sweep where the plane begins and the number of layers that will be cleaved as shown in Figure 3. Separating the elements that lie on the cleaved plane from the surrounding elements creates a cleaved volume.

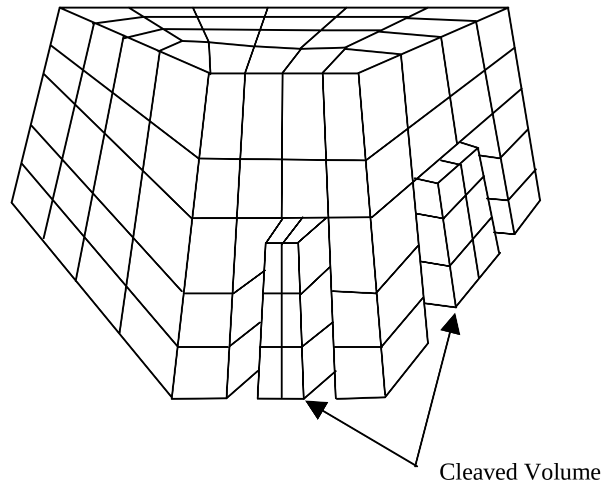


Figure 2. The cleaving process

The void between the newly created *cleaved volume* and the existing mesh is then filled using a process called *pillowing.* Mitchell introduced pillowing previously. [1] This process fills the cleaved volume with conforming hexahedral elements as shown in Figure 4.

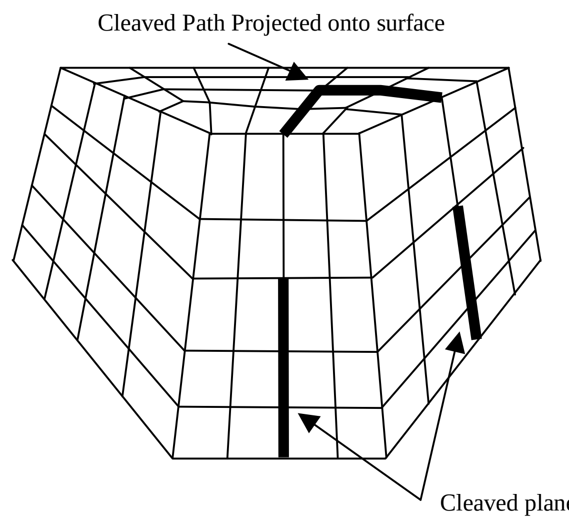


Figure 3. Cleaved plane definition

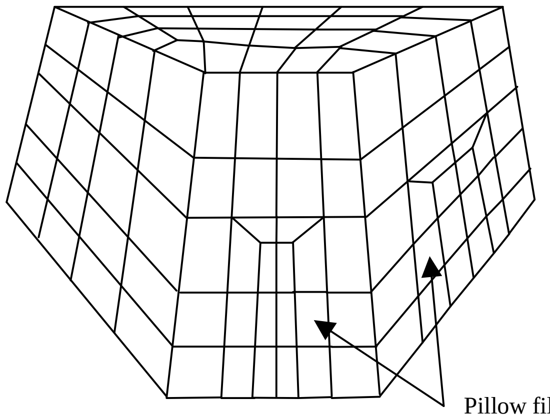


Figure 4. Cleaved volume filled with a "pillow"

3. THE CLEAVE AND FILL ALGORITHM

A principle objective of the cleave and fill algorithm is to allow the refinement of a swept mesh by adding additional elements. The algorithm has three major steps. The first is to define the cleaved path in an efficient manner. Having the path defined the next step is to generate the cleaved volume. The final step is to fill the cleaved volume with well-formed elements. The algorithm, as reported here, has the ability to automatically generate a minimum distance path across a given unstructured two-dimensional source or target mesh. The cleaved volume is filled using an adapted "pillow fill" process. The algorithm is also capable of refining the mesh without modifying a linking surface as well as refining the mesh both parallel and perpendicular to the sweeping trajectory. These procedures are described in the next sections.

3.1 Defining the path through a meshed volume

The first step of the cleave and fill algorithm is to find a path of nodes through the mesh between two given nodes that are owned by a linking surface. Our method to accomplish this task is based on a modified form of "Dijkstra's algorithm." [9] This algorithm finds the shortest weighted distance between a given start node and every other node in the search group. This is done using a breadth first search.

There are three objectives to this step:

1) Minimize the number of nodes in the path. This will minimize the number of hexes in the transition area and improve the resulting quality of the mesh.

2) Minimize the number of turns made by the path. A turn is defined as three consecutive nodes in the path that are owned by the same hex.

3) Move the path away from surfaces. By keeping the path toward the center of the volume there is more room to smooth the new hexes to increase the quality of the resulting mesh.

The weighted distance of a given node is defined as:

$$\text{distance} = (\text{previous distance} + 1) + \text{turns} + (\text{max weight} - \text{node weight})$$

Where

previous distance = distance of previous node in the path.

turns = 0 if no turn between node and previous node, = 1 if turn between node and previous node.

node weight = weight of node depending on position in volume. The nodes are weighted starting with 0 on any surface of the volume and increasing towards the center of the volume.

max weight = maximum weight of all weighted nodes.

The algorithm used to find the shortest weighted distance proceeds through the following steps:

1) Define the start and end nodes as well as the group of nodes that will be in the search.

2) Weight all nodes in the search group.

3) Remove all nodes with weight of zero except the start and end nodes.

4) Set start node distance equal to zero and the distance of all other nodes equal to the maximum integer value.

5) Loop through the search group until the end node is reached.

A) If the search group is empty, then the layer mesh is disjoint and the end node cannot be reached from the start node. Return Failure.

B) If N is the end node exit loop, otherwise proceed with loop.

C) Remove N from search group.

D) Update the weighted distances and the shortest paths to the nodes adjacent to N using the triangle inequality:

For each adjacent node M

If
distance to N + distance from N to M <
current distance to M;

Then
  distance to M = distance to N +
  distance from N to M;

  shortest path to M = path to N +
  path from N to M;

6) From end node step to previous node and continue stepping to previous nodes until the start node is reached.

Figures 5 through 10 show examples of this algorithm. Figure 5 shows a simple path with no turns and only one choice with a minimum number of nodes. This path also runs through the nodes with the maximum weight value. Figure 6 shows a simple NxN rectangular mesh with a start and end node chosen one node from the edge. In this case the shortest path with no turns is chosen over a path that passes through the highest weighted nodes.

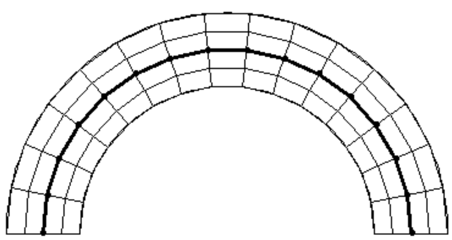


Figure 5. Simple path, no turns

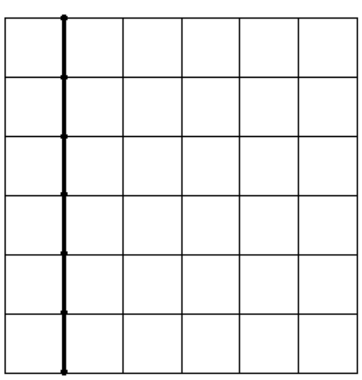


Figure 6. Simple path, no turns

The path in Figure 7 is more complex. With this path there are a number of routes that would qualify as containing the fewest number of nodes. The path that was chosen has the fewest number of turns—since the path along the outer surfaces was not allowed in the search—and passes through the nodes with the highest weights.

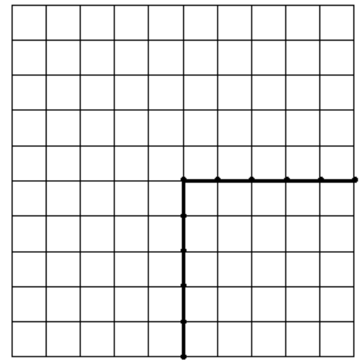


Figure 7. Simple path, one turn

The path that is shown in Figure 8 shows the effect of node weight in path selection. A path with two fewer nodes can be found by following the nodes that are only one interval from the surface. However, since these nodes are weighted less than the nodes along the center path, the center path was chosen.

Figures 9 and 10 show paths that were found through a complex geometry.

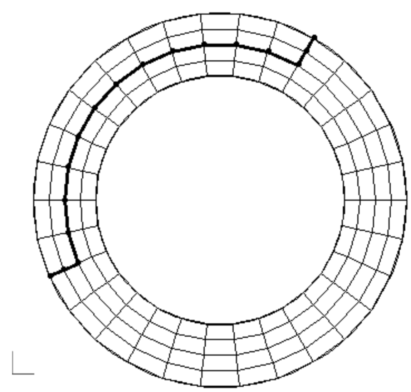


Figure 8. Simple path with two turns

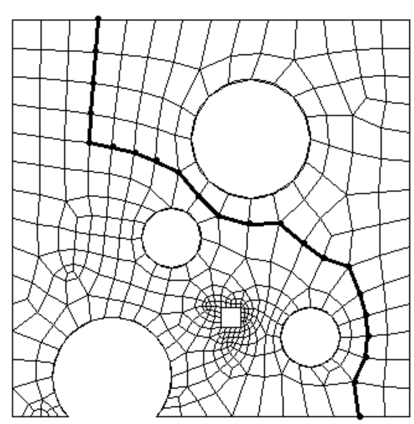


Figure 9. Complex path

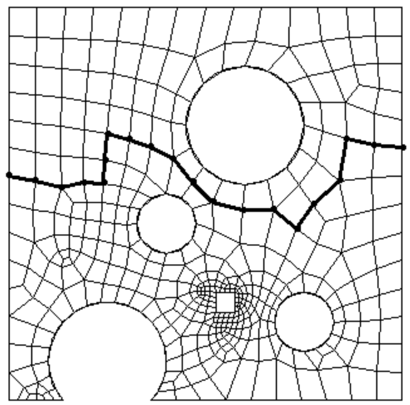


Figure 10. Complex path

3.2 Inserting a Pillow

Once the initial cleaved path has been defined, it is then projected in the cleave direction until it reaches its target destination. The projection of the cleaved path at each sweep layer is used to define the cleaving plane. Once the cleaving plane has been defined, all hexahedral elements that contain a node that is in the cleaving plane are found. The sides of these hexahedral elements make up the boundary of the new cleaved volume.

Pillow insertion is accomplished after the cleaved volume has been defined. A pillow is inserted around all the hexahedral elements that compose the cleaved volume. The pillow volume has the identical connectivity that exists on the cleaved surfaces. The element configuration of the pillow is shown in Figure 11.

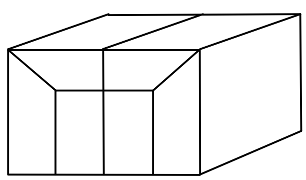


Figure 11. Basic configuration of a pillow region

Note that normal smoothing operations are required after pillow insertion to improve the quality of the final mesh.

3.3 Refining Without Modifying Linking Surfaces

For some models it may be desired to modify the internal mesh without modifying the linking surfaces. This can be done with the cleave and fill tool by modifying the algorithm used to find the cleaving path. Instead of finding a path between two nodes the algorithm finds a path at a given number of intervals from the linking surfaces. The number of intervals, the sweep layer that will contain the path, and the cleave direction must be passed to the tool. The tool then finds all the nodes contained in the given sweep layer and weights them using the same weighting scheme as mentioned above. Using this approach, all the nodes in the sweep layer with a weight equal to the given number of intervals are found. These nodes are then checked to see if they make a valid path. If they do not form a single, closed loop the path is not valid and the operation is stopped. If a single, closed loop is found it becomes the internal cleaved path and a cleaved plane is projected through the volume until it emerges from a surface.

Figure 12 shows the target surface of a brick with an internal path projected onto it. This path has been specified with an internal interval of three.

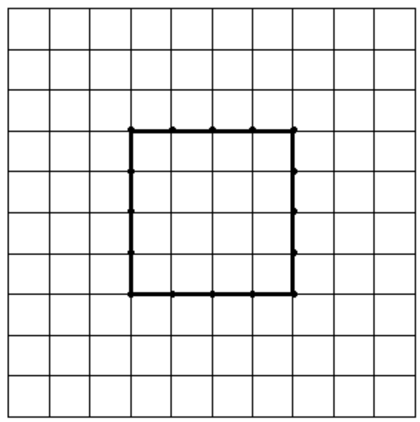


Figure 12. Internal path

3.4 Refining Across the Sweep Direction

The cleave and fill algorithm also has the ability to refine a mesh perpendicular to the sweep direction. To do this, the normal steps are taken to find the cleaved path between two given nodes. Once the path has been found it is then projected along the sweep layer of nodes that contain the path. If a hole that passes through the volume in the direction of the sweep trajectory is encountered the path will be projected across the void and continue on until it reaches an outer linking surface.

An example of a volume where this type of cleaving may be desired is shown in Figure 13. This volume has the cross section shown in Figures 9 and 10 with the cleaved path through the volume highlighted in Figure 10. This path would be projected through the volume along the highlighted edges in Figure 13 to emerge from the linking surface. The results of this cleave and fill operation can be seen in Figure 18.

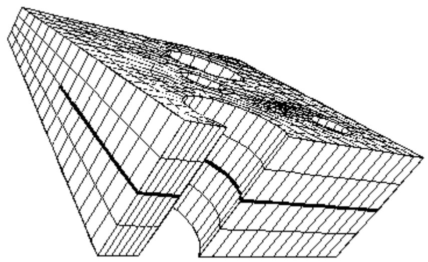
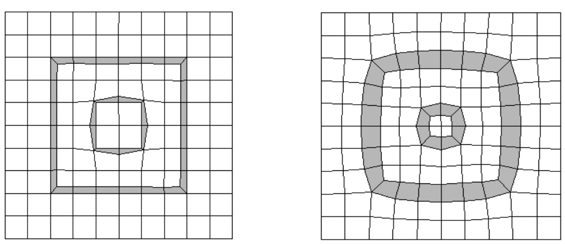


Figure 13. Cleaving normal to sweep trajectory

4. EXAMPLES

Shown below are four examples of the cleave and fill algorithm. For clarity, the elements that have been added by the algorithm are highlighted.

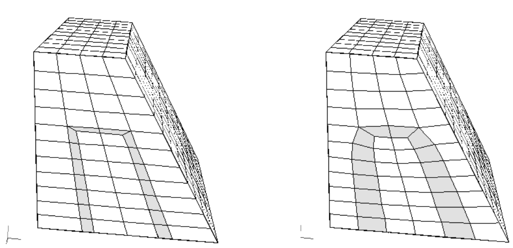
Shown in Figure 14 is the inserted mesh of a totally contained cleave and fill as it appears on a target face of a simple swept rectangular shaped box.



mesh before smooth mesh after smooth

Figure 14. Target surface of a totally contained cleave and fill of a rectangular box

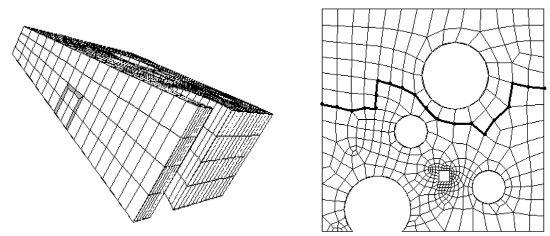
Figure 15 shows the results of applying the algorithm, again on a simple box object where now the operation has begun on one linking surface and proceeded to the directly opposite linking surface.



mesh before smooth mesh after smooth

Figure 15. Linking surface after cleave and fill operation

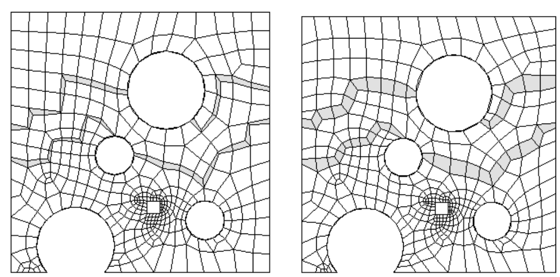
Shown in Figure 16 is a swept mesh using the complex source and target surfaces given in Figure 10. The cleave and fill operation has been applied in the direction of sweep. The cleaving path is shown as the heavy solid line.



Linking surfaces Source and Target surfaces

Figure 16. Cleaving path for complex source and target

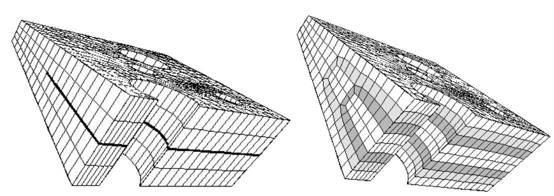
Figure 17 shows the modified mesh on the target surface after the cleave and fill operation of the body shown in Figure 16.



mesh before smooth mesh after smooth

Figure 17 Source mesh after cleave and fill operation

Figure 18 is the same object as shown in Figure 16, however in this instance, the refinement is applied orthogonal to the sweeping trajectory. Note that the cleave and fill algorithm has been applied twice to this volume.



Cleaved path mesh after smooth

Figure 18 Cleave and fill orthogonal to sweeping trajectory

The bodies in Figures 15 and 18 provide good examples of how the cleave and fill algorithm can be used to enhance the quality of a swept mesh. Both of these bodies are tapered such that the initial mesh has elements on one side with aspect ratios near 1.0 and on the opposite side with aspect ratios that may be undesirable. The aspect ratio is defined as the maximum of the six quantities

$$LX/LY,\ LY/LX,\ LX/LZ,\ LZ/LX,\ LY/LZ,\ LZ/LY$$

where *LX* is the distance between two opposite hex faces in the YZ plane, *LY* is the distance between two opposite faces in the XZ plane, etc. [10] Elements with aspect ratios near 1.0 are deemed to be of higher quality.

Tables 1 and 2 show a comparison of the aspect ratio before and after cleaving, filling, and smoothing for Figures 15 and 18 respectively.

	Before Cleaving	After Cleaving and Smoothing
Average	1.956	1.441
Std. Deviation	0.4570	0.1564
Min.	1.232	1.111
Max.	2.683	1.857

Table 1 Aspect ratio of wedge model

	Before Cleaving	After Cleaving and Smoothing
Average	2.390	1.550
Std. Deviation	1.261	0.4971
Min.	1.009	1.004
Max.	7.197	4.144

Table 2 Aspect ratio of wedge with holes model

5. FUTURE WORK

Although the cleave and fill algorithm has been seen to improve mesh quality in many cases, there are some issues that need to be addressed in order to make this algorithm more widely applicable to the 3-D all-hexahedral meshing problem.

- Generalize the cleave and fill algorithm so that is can be applied to any meshed volume. Currently, the algorithm will only work on swept meshes with the start and end node of the cleaved path lying on a linking surface. These constraints stem from the assumption that the surfaces containing the start and end node are mesh with a structured mesh. By removing this assumption, the start and end node could be located on any surface and the volume meshed with a scheme other than sweep.

- Add the ability to coarsen a mesh. Currently, the cleave and fill algorithm only refines a mesh by adding a pillow around the cleaved volume. In some cases it may be desirable to remove the cleaved volume from the mesh and then stitch the sides of the remaining void together, thereby, coarsening the mesh.

- Improve internal cleaving. The complexity of volumes that internal cleaving can be applied to is limited. In order for internal cleaving to be more useful it needs to work on volumes that contain holes. Another limitation of the current implementation of internal cleaving is the requirement that the internal cleaved path be contained in a single sweep layer, thus, limiting it to cleaving only in the direction of sweep. Developing a new weighting scheme that is not dependent on the surfaces of the volume would solve both these problems. One way this could be done is by specifying a center point—either a location or a node—and an approximate radius around that point that the desired cleave path should lie on.

6. CONCLUSIONS

This paper has presented a method to refine three-dimensional all hexahedral swept meshes. This refinement can be accomplished both parallel and perpendicular to the sweeping trajectory. An important feature of the method is an adaptation of “Dijkstra’s” algorithm that is used to define a minimum path across a source or target surface.

The method is very useful in allowing the modification of a traditionally created swept mesh. It can be easily incorporated into an existing three-dimensional all hexahedral meshing schemes. High quality refined elements are a feature of this algorithm.

REFERENCES

[1] Mitchell, S.A., and Tautges, T.J., “Pillowing doublets: refining a mesh to ensure that faces share at most one edge,” *Proceedings, 4th International Meshing Roundtable 95*, pp 231-240, October 1995.

[2] Benzley, S.E., Perry, E., Merkley, K., Clark, B, and Sjaardema. G., “A Comparison of All-Hexahedral and All-Tetrahedral Finite Element Meshes for Elastic and Elasto-Plastic Analysis,” *Proceedings, 4th International Meshing Roundtable, Sandia National Laboratories, 95*, pp. 179-191, October 1995.

[3] Cifuentes, A.O., and Kalbag, 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, 1992.

[4] Mitchell, S.A., "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, 1996.

[5] Staten, M.L., Canann S.A., and Owen, S.J., "BMSweep: Locating Interior Nodes During Sweeping," *Proceedings, 7th International Meshing Roundtable 98,* pp. 7-18, October 1998.

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

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

[8] Jankovich, S.R., Benzley, S.E., Shepherd, J.F., and Mitchell, S.A., "The Graft Tool: An All-Hexahedral Transition Algorithm for Creating a Multi-directional Swept Volume Mesh," *Proceedings, 8th International Meshing Roundtable 99*, pp. 387-392, October 1999.

[9] Weiss, M.A., *Data Structures and Algorithm Analysis,* Benjamin/Cummings Publishing, Redwood City, CA, 1995, pp 289-297.

[10] Sandia National Laboratories, *CUBIT Mesh Generation Environment Volume 1: Users Manual*, Sandia National Laboratories, 2000