Methods and Applications of Generalized Sheet Insertion for Hexahedral Meshing

Karl Merkley[1], Corey Ernst[2], Jason F. Shepherd[3], and Michael J. Borden[4]

[1] Computational Simulation Software, LLC., American Fork, UT karl@csimsoft.com
[2] Elemental Technologies, Inc., American Fork, UT corey@elemtech.com
[3] Sandia National Laboratories, Albuquerque, NM jfsheph@sandia.gov
[4] Sandia National Laboratories, Albuquerque, NM mborden@sandia.gov

Summary. This paper presents methods and applications of sheet insertion in a hexahedral mesh. A hexahedral sheet is dual to a layer of hexahedra in a hexahedral mesh. Because of symmetries within a hexahedral element, every hexahedral mesh can be viewed as a collection of these sheets. It is possible to insert new sheets into an existing mesh, and these new sheets can be used to define new mesh boundaries, refine the mesh, or in some cases can be used to improve quality in an existing mesh. Sheet insertion has a broad range of possible applications including mesh generation, boundary refinement, R-adaptivity and joining existing meshes. Examples of each of these applications are demonstrated.

Key words: hexahedra, meshing, dual, boundary layers, refinement

1 Introduction

Sheet insertion is a technique for modifying the topology of a hexahedral mesh and introducing new elements which geometrically correlate with the shape of the sheet. These topological changes and the new elements introduced by the insertion provide methods for defining, refining, and improving the quality of a mesh. This paper will review several applications of sheet insertion including pillowing, dicing, refinement, grafting, and mesh cutting and show how these methods are related by the sheet insertion technique.

Every hexahedral mesh is defined as a collection of hex sheets. A hexahedral sheet is dual to a layer of hexahedra and can be geometrically correlated to the shape of the hexahedral sheet. A sheet of hexahedra can be easily visualized by examining an equivalent structure in two dimensions. Given a quadrilateral mesh, a layer of quadrilaterals is found by starting at the center of a single quadrilateral and traversing through the opposite edge pairs of connected quadrilaterals in both directions. The layer will always end by returning to the original quadrilateral or reaching the boundary of the mesh, as shown in figure 1. A continuous line drawn through this layer of quadrilaterals can be used visually to represent all of the quadrilaterals in the layer. This line segment is known as a chord, and is dual to the layer of quadrilateral elements.

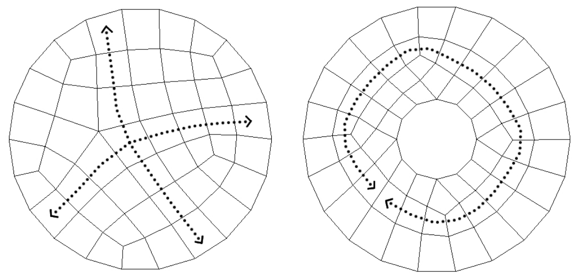

Fig. 1. Rows of quadrilaterals.

This description can be extended to hexahedra in three-dimensions. A layer of hexahedra can be visualized as a single manifold surface. This surface is known as a 'sheet', where a sheet is dual to a layer of hexahedra. Each hexahedron can be defined as a parametric object with 3 sets of 4 edges, where all edges in a set are normal to the same parametric coordinate direction, either *i, j,* or *k*. Starting from any hexahedron, we can create a sheet of hexahedra by first obtaining a set of edges in one of the three parametric directions. Next, we collect all neighboring hexahedra sharing these edges. For each neighboring hexahedron obtain the set of edges in the same parametric direction as the initial hexahedron. Using these edges, continue to propagate outwards, collecting neighboring hexahedra. When all adjacent hexahedra are collected in this manner, the result will be a layer of hexahedra that is manifold within the boundary of the mesh (that is the layer will either terminate at the boundary of the mesh, or will form a closed boundary within the mesh). This layer of hexahedra can be visualized as a single manifold surface, known as a sheet, where the sheet is dual to a layer of hexahedra (see figure 2).

Introduction of new sheets in a hexahedral mesh modifies the topology of the existing mesh and introduces new layers of elements within the mesh. As long as the sheets intersect according to a set of topological constraints for a hexeahedral mesh [1, 2], the resulting mesh with the new sheet will still be a hexahedral mesh. Using this principle, a hexahedral mesh may be modified by inserting new hex sheets in specific manners depending upon the application to realize new meshes, define new mesh boundaries, refine an existing mesh, or improve the quality of a hexahedral mesh.

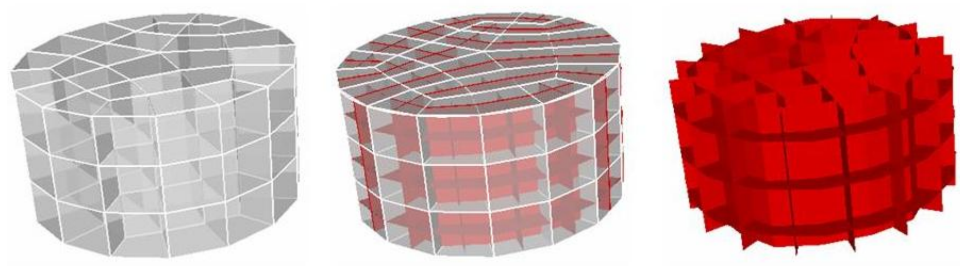

Fig. 2. Sheets of hexahedra.

2 Motivation

Hexahedral meshing has been an active research topic for many years [3]. There have been a number of attempts at generalized meshing algorithms such as plastering [4], whisker-weaving [5, 6, 7] and unconstrained plastering [8]. While significant progress as been made and some of these algorithms show promise for specific geometric types, hex meshing is still considered an open problem [9].

The most successful algorithms for hexahedral mesh generation are based on geometric primitives. For example, a mapping algorithm is based on a parametric cube [10]. Blocking algorithms wrap a geometry in a known hexahedral primitive [11, 12, 13, 14]. Submapping algorithms virtually decompose a geometry into cube-shaped primitives [15]. Sweeping algorithms are based on extrusions of 2D geometric shapes [16, 17, 18].

Sheet insertion offers techniques that can extend the capabilities of primitive based algorithms. Because sheet insertion provides methods for modifying hexahedral meshes, it can be considered as somewhat analogous to tetrahedral mesh modifications such as edge swapping. Tetrahedral mesh modification operations tend to be local and only affect the elements in the immediate area of the change. However, due to the topological structure of hexahedral meshes, modification of a hexahedral mesh affects the mesh in a non-local manner. Sheet insertion quantifies the scope of the non-local changes and provides effective tools for hexahedral mesh modification.

The remainder of this paper provides an overview of several algorithms utilizing sheet insertion. The scope of each algorithm is different, depending on the application for which the algorithm was initially intended. The common thread running through these algorithms is that they introduce new sheets into an existing hexahedral mesh. The new sheets are then utilized to affect changes to the original mesh including geometric boundary modification, refinement, quality improvement, and mesh adaptivity. We provide a survey of these sheet insertion algorithms, and then offer several examples of how sheet insertion can be utilized to generate hexahedral meshes in complex geometries, adaptively conform a mesh topology to specific supposed analytic

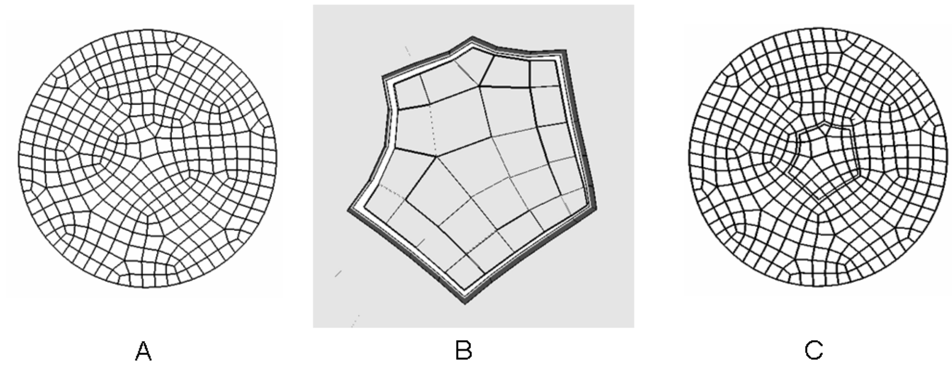

Fig. 3. A basic pillowing operation starts with an initial mesh (A) from which a subset of elements is defined to create a shrink set. The shrink set is separated from the original mesh and ‘shrunk’ (B), and a new layer of elements (i.e., a dual sheet) is inserted (C) to fill the void left by the shrinking process.

properties within a simulation, improve the mesh quality within an existing mesh, or control mesh sizing.

3 Methods

This section provides an overview of hexahedral mesh modification methods that rely on sheet insertion, including pillowing, dicing, refinement, mesh cutting and grafting.

3.1 Pillowing.

Pillowing is a method for inserting a single sheet into an existing hexahedral mesh. Initially proposed by Mitchell et al. [19], a pillowing operation provides a means for eliminating problematic mesh topology.

The basic pillowing algorithm is as follows:

1. *Define a shrink set* - Divide the existing mesh into two sets of elements: one set for each of the half-spaces defined by the sheet to be inserted. Choose one set as the shrink set. The set with the fewest number of elements is typically chosen as the shrink set.
2. *Shrink the shrink set* - Create a gap region between the two element sets (see figure 3).
3. *Connect with a layer of elements* - Create a fully-conformal mesh with the new sheet inserted between the original two element sets.

It is often desirable to smooth the resulting mesh after the sheet insertion to obtain better nodal placement and higher quality elements. The speed of the pillowing algorithm is largely dependent on the time needed to find the shrink set. The number of new hexahedra created will be equal to the number of quadrilaterals on the boundary of the shrink set.

Pillowing is an operation for inserting a single sheet into an existing mesh. It is a useful multipurpose, foundational tool for operations on hexahedral meshes and is the basis for other mesh modification tools, including refinement, grafting, and mesh-cutting.

3.2 Dicing.

The dicing algorithm [20] was created to efficiently generate large, refined meshes from existing coarse meshes by duplicating sheets within the coarse mesh. Each sheet is copied and placed in a parallel configuration to the sheet being copied. The basic method for dicing is as follows:

1. *Define the sheet to be diced* - See sheet definition in introduction.
2. *Dice the edges* - Split or dice the list of edges found in the previous step the specified number of times.
3. *Form the new sheets* - Redefine the hexahedra based on the split edges.

Utilizing the dicing method, the number of elements increases as the cube of the dicing value. For instance, if an existing mesh is diced four times (i.e., each of the sheets in the existing mesh is split four times), the resulting mesh would have 64X as many elements as the original mesh. Because all search operations can be performed directly, the dicing algorithm can produce large meshes at very efficient speeds (see figure 4).

3.3 Refinement

The size of a hexahedral mesh is relative to the sheet density in the mesh. Increasing the element count within a region of the mesh can be accomplished

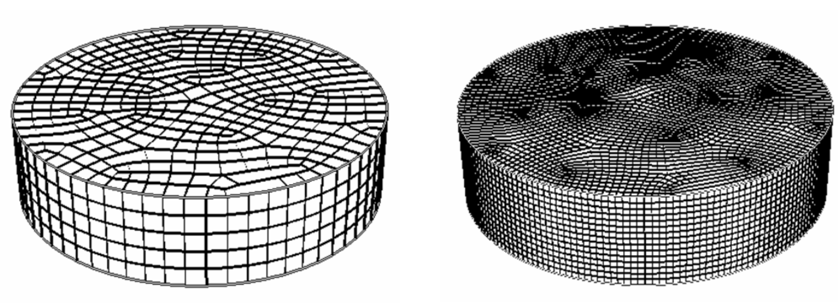

Fig. 4. The original mesh (left) contains 1,805 hex elements before dicing. Each sheet in the original mesh is copied three times resulting in a mesh that is 3^3 larger, with 48,735 hex elements.

by increasing the density of the sheets in the region. Refinement algorithms typically utilize predefined templates to increase the element density. These templates and the algorithms that insert them ensure manifold sheet structures within the mesh to maintain mesh validity [21, 22]. A basic refinement algorithm is constructed as follows:

1. *Define the region or elements to be refined* - Replace the hexahedra in the refinement region with a predefined template of smaller elements.
2. *Cap the boundary of the refinement region to ensure conformity to the rest of the mesh.* - Create a second set of templated elements which transition from the refinement template back to the original mesh size. Figure 5 demonstrates this process.

3.4 Mesh cutting.

The mesh-cutting [23] method is an effective approach for capturing geometric surfaces within an existing mesh topology. The mesh-cutting method utilizes the pillowing and dicing methods mentioned previously to insert two sheets which are geometrically similar to the surface to be captured. By utilizing two sheets, a layer of quadrilaterals results that approximates the desired surface where each of the new quadrilateral is shared between the hexes in the two new sheets. The mesh-cutting method entails the following steps:

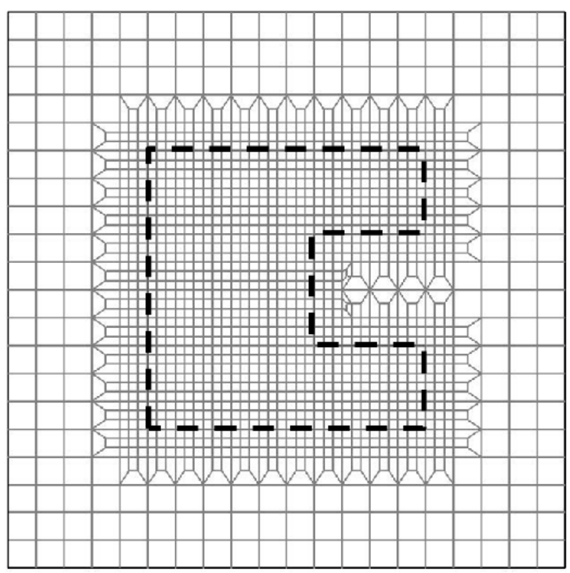

Fig. 5. Refinement of a hex mesh using refinement templates. Refinement region is shown as a dashed line.

1. *Define the pillowing shrink set* - Divide the existing mesh into two sets of elements. One of these element sets will be the shrink set, and a sheet (pillow) is inserted between the two sets of elements.
2. *Dice the new sheet* - Split the newly inserted sheet into two sheets utilizing an approach similar to dicing.
3. *Move the shared quadrilaterals to the surface* - Find all of the quadrilaterals that are shared by the hexes between the two sheets. These quadrilaterals become the mesh on the surface being cut (see figure 6).

Because the quadrilaterals between the two new sheets approximate the surface, the mesh topology must be fine enough to capture the detail of the surface being inserted. Because the resulting quadrilaterals only approximate the inserted surface, if the resulting quadrilateral mesh is too coarse, the surface may not be approximated adequately.

3.5 Grafting.

The term 'grafting' is derived from the process of grafting a branch from one tree into the stem, or trunk, of another tree. In meshing, the grafting method was developed to allow a branch mesh to be inserted into the linking surface of a previously hexahedrally swept volume [24]. The grafting method offers a reasonably generalized approach to multiaxis sweeping. The method for creating a graft (i.e., capturing the geometric curve) can be outlined as follows (see also figure 7):

1. *Create a pillowing shrink set* - The shrink set is defined as the set of hexes which have one quadrilateral on the surface and which are interior to the bounding curves of the graft.

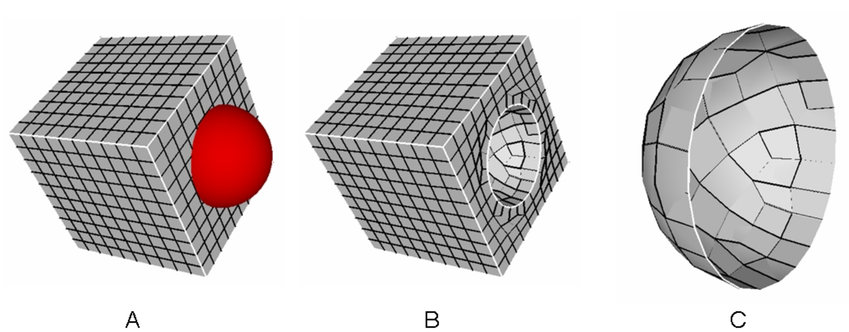

Fig. 6. Mesh cutting utilizes an existing mesh and inserts new sheets to capture a geometric surface (the existing mesh is shown in (A) where the spherical surface is the surface to be captured.) The resulting mesh after mesh cutting is shown in (B), with a close-up of the quadrilaterals on the captured surface being shown in (C).

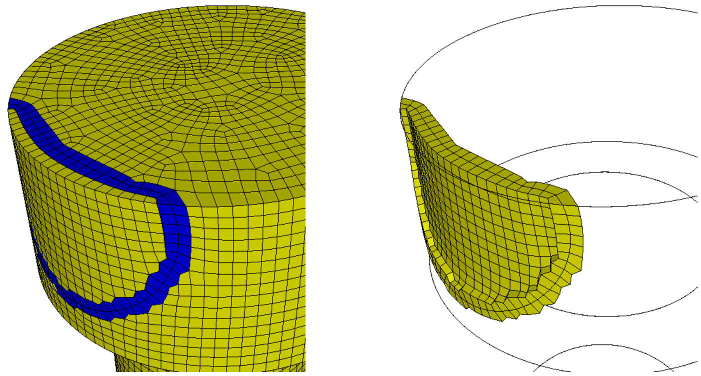

Fig. 7. A view of the hex sheets that were inserted into one of the trunk volumes by the grafting algorithm. The figure on the left highlights the sheets where they intersect the surface of the volume. The figure on the right shows the orientation of the sheets in the interior of the mesh.

2. *Insert the pillow (sheet)* -Inserting the second sheet to satisfy the hexahedral constraint for capturing the geometric curves used in the graft.

At this point, there are often database adjustments needed to ensure that the new mesh entities are associated with the correct geometric entities.

4 Applications

Because hexahedral sheet insertion operations are extremely flexible, there are numerous possible applications. This section demonstrates a few of these applications. The meshes in this section were developed using Cubit [25], developed at Sandia National Laboratories, SCIRun [26], developed at the University of Utah, or a combination of the two tools. The processes for completing these models are not yet fully automated, but there is ongoing development to make these processes robust and usable.

4.1 R-Adaptivity

Sheet insertion can be used to divide a model and refine the mesh in a given region. Figure 8 shows an R-adaptive mesh capturing a shock a wave. An initial sheet is placed that is aligned with the shock wave and mesh-cutting techniques are used to capture the shock wave. Additional sheets are then added to provide the required resolution around the shock wave.

Adaptive smoothing operations can be utilized to control orthogonality and mesh spacing close to the adaptive region. The mesh after several operations of layer smoothing is shown in figure 9.

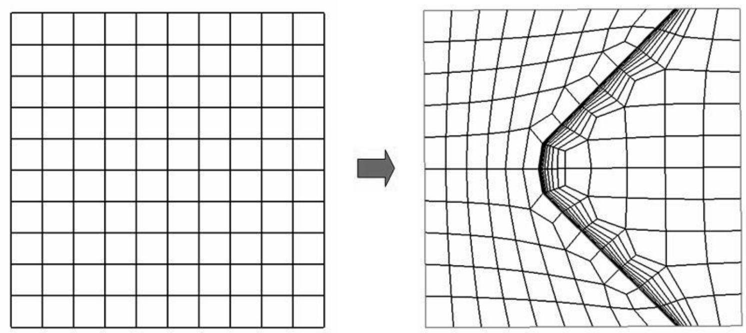

Fig. 8. Sheets can be placed within a mesh to capture analytic features, such as shock waves. The original mesh is shown on the left. After several sheet insertion operations to place layers of elements roughly conforming to a supposed shock, the newly modified mesh is shown on the right. Mesh generated using Cubit.

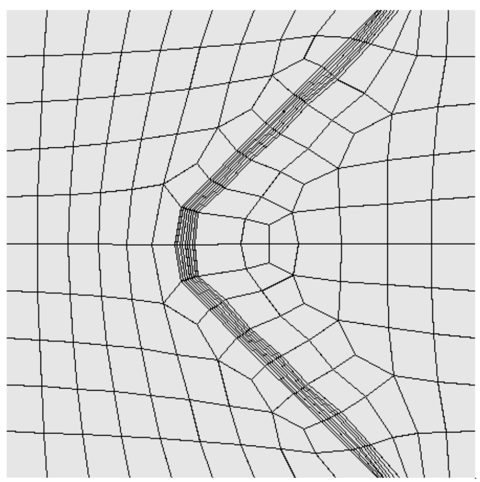

Fig. 9. Smoothing of R-adaptive region using Cubit

4.2 Boundary Layers

Sheet insertion techniques can be used in a similar manner to capture boundary layers as shown in airfoil mesh in figure 10. Inserted sheets can provide additional resolution at the boundary. The number of layers and the distribution of the sheets could be specified according to the requirements of the application.

Additionally, new geometric features can be added by the sheet insertion process. For example, the geometry in figure 11 was split through the center of the airfoil. This model provides a mesh that is more closely aligned with streamlines through the entire model. Because varying applications may pose different requirements on the analysis model, the flexibility provided by the sheet insertion techniques can be very valuable.

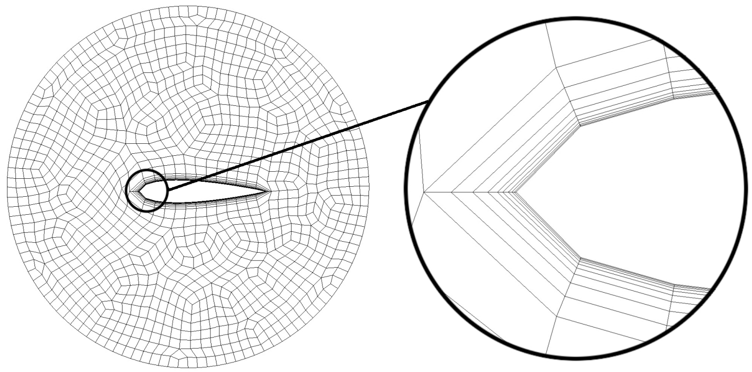

Fig. 10. Sheets of hexahedra can be inserted to capture boundary layers. Mesh generated in Cubit.

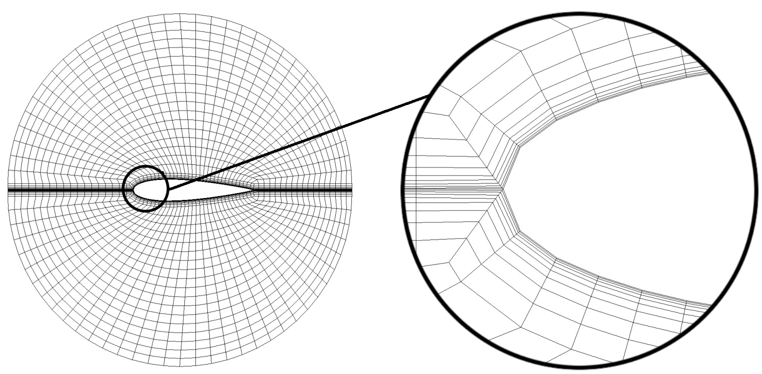

Fig. 11. Sheets of hexahedra added to better resolve streamlines of model. Mesh generated in Cubit.

4.3 Mesh Generation

Hexahedral sheet insertion also provides techniques that can be used to generate complex meshes. The typical mesh-cutting technique involves embedding the desired geometry into an existing hexahedral mesh that envelopes the geometry. One or more sheets are then inserted into the mesh which capture the missing, but desired, geometric features. Additional sheets may be inserted to improve the element quality at the boundary.

Figure 12 shows an all hexahedral mesh of a human hand while and figure 13 shows the internal detail of one finger of this model. This mesh was generated in SCIRun using a sheet insertion operation, similar to mesh cutting. Once the initial mesh was captured as desired, four additional layers of hexahedra were added at the boundary (for a total of five boundary layers) that are oriented orthogonally to the surface. This was accomplished by dicing the original sheet used to capture the surface of the hand. The resulting mesh contains 379,429 hexahedra elements, and the mesh contains all positive Jacobian and well shaped hexahedra. Figure 14 shows the spread of scaled Jacobian element quality.

Fig. 12. All hexahedral mesh of a human hand created by mesh cutting. The original triangle mesh of the hand that was utilized in controlling the placement of the hexahedral boundary layer was provided courtesy of INRIA by the AIM@SHAPE Shapes Repository http://shapes.aim-at-shape.net/index.php). Mesh generated in SCIRun.

Biomedical applications are a source of very difficult modeling problems. Figure 15 shows a model of an all hexahedral mesh of a human skull and cranial cavity. The meshes were formed by first creating a rectilinear mesh that was sized such that the bounding box was slightly larger than the skull. Triangular meshes of the skull and the cranial cavity were then inserted into

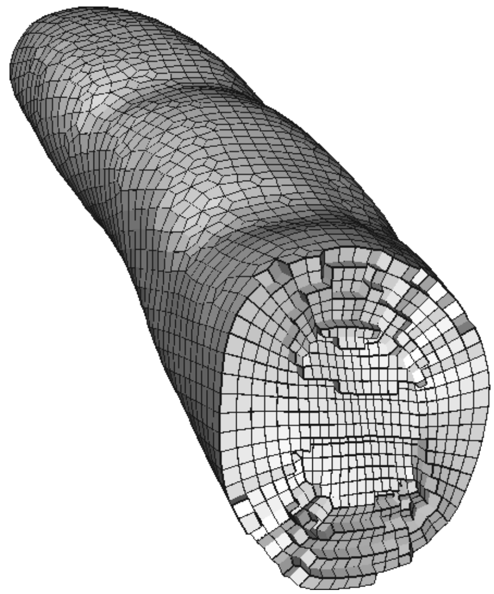

Fig. 13. Cut-away view of one of the fingers for the all-hexahedral mesh of a human hand showing the internal structure of the mesh.

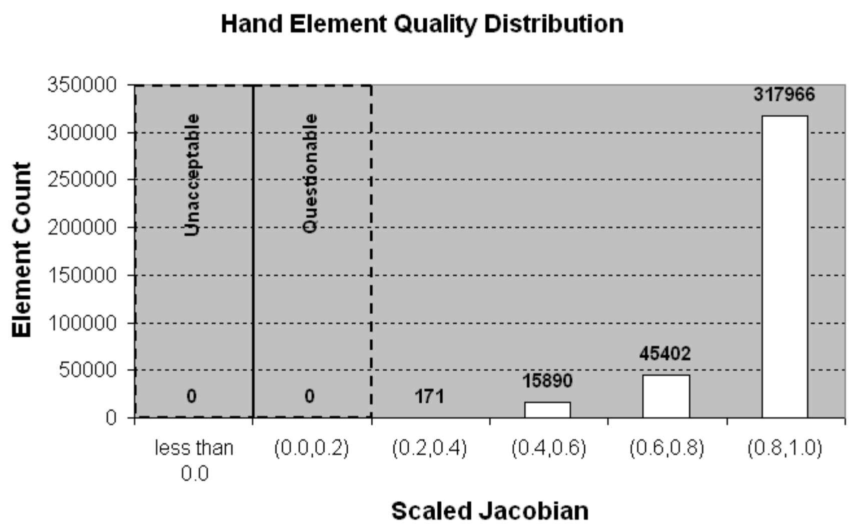

Fig. 14. Distibution of element quality by the scaled Jacobian metric for the hand model.

the rectilinear mesh to capture the surface data. The level of detail captured in this technique is proportional to the level of detail of the triangular surface mesh. Additional sheets where inserted at the surface to improve the quality of the mesh at the surface. Figure 16 shows a cut-away view of the mesh (accomplished again with a mesh-cutting operation) to give a view of the internal structure of the mesh. The mesh of the skull contains 77,775 elements consisting of well-shaped hexahedra with all positive Jacobian measures. The quality distribution for the elements of this mesh is shown in figure 17.

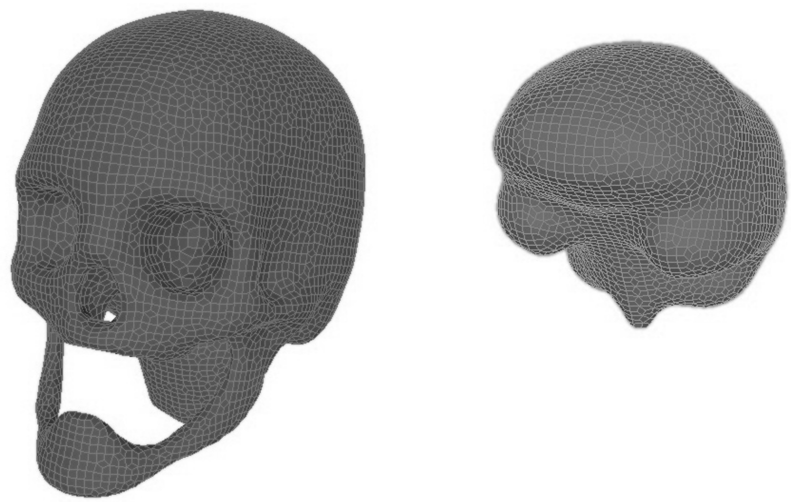

Fig. 15. All hexahedral mesh of a human skull and cranial cavity. The original triangle mesh of the skull that was utilized in controlling the placement of the hexahedral boundary layer was provided courtesy of INRIA by the AIM@SHAPE Shapes Repository http://shapes.aim-at-shape.net/index.php). Mesh generated in SCIRun.

4.4 Grafting

The grafting algorithm [27] was developed to allow a volume to be imprinted and merged with another volume after the initial volume was meshed. The grafting algorithm locally modifies the initial mesh to conform to the geometry of the second, unmeshed volume. One application of this algorithm has been generating swept meshes for models with two independent sweep directions.

For example, suppose an engineer is interested in analyzing the fluid flow through the valve housing shown in figure 18. The fluid region with the valve partially closed is shown in figure 19. Notice that there is an independent sweep direction corresponding to each pipe opening from the valve. To generate a

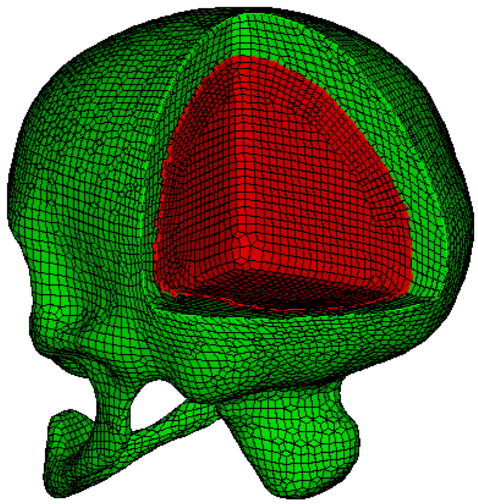

Fig. 16. Cutaway view showing the interior structure of the all hexahedral mesh of a human skull and cranial cavity. Cutaway created in Cubit.

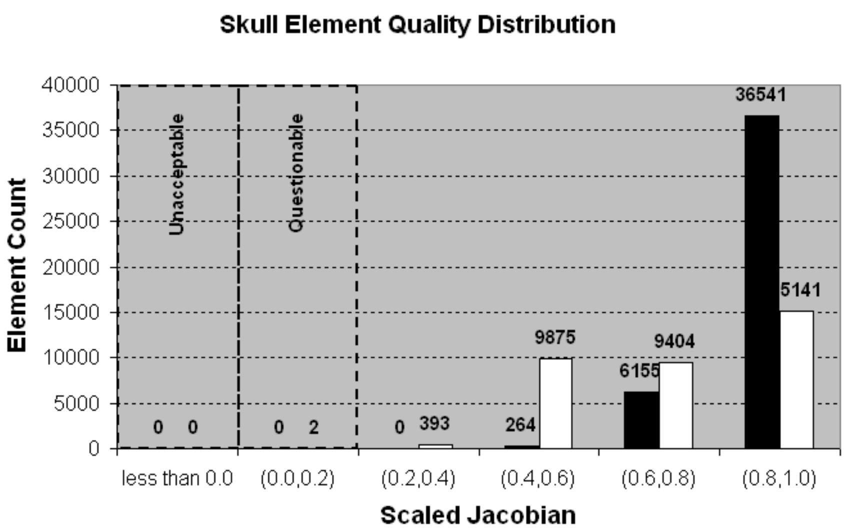

Fig. 17. Distribution of element quality by the scaled Jacobian metric for the skull model. The black bars represent the quality of the interior cranial mesh, while the white bars are representative of the skull bone.

mesh of this region with grafting, one would first mesh the two volumes that are split by the valve plate as shown in figure 19. The remaining unmeshed volume can then be grafted onto the meshed volumes as shown in figure 20. The grafting operation introduces a minimal, local change to the mesh as shown in figure 7.

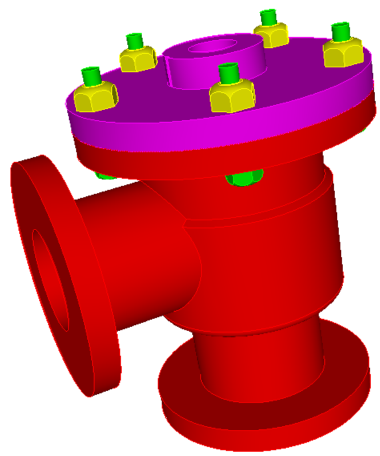

Fig. 18. A valve housing

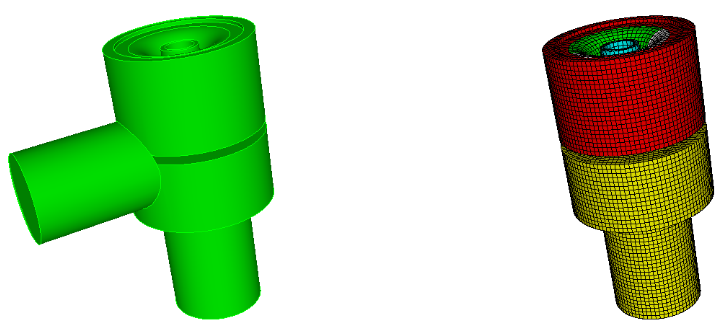

Fig. 19. The fluid region of the valve housing in figure 18 with the valve partially closed (left) and the initial, pre-grafting swept mesh (right)

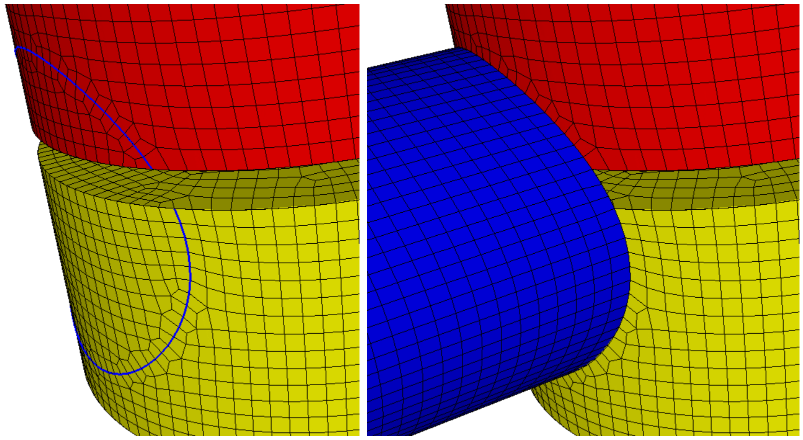

Fig. 20. The area of the mesh that has been modified by grafting. The figure on the left shows the imprint of the branch volume onto the trunk volumes. The figure on the right shows the final mesh on the branch. Mesh created in Cubit.

5 Conclusion

Generalized hexahedral sheet insertion is an effective technique for modifying hexahedral meshes, including defining new mesh boundaries, refining the mesh, and improving the quality of an existing mesh. The technique is very flexible and can be applied in a variety of methods to modify a mesh, including R-adaptivity, refinement, boundary layer insertion, mesh generation, and mesh improvement.

The examples presented here have shown that sheet insertion can greatly decrease the effort required to construct a high quality mesh on difficult models. In particular, sheet insertion as a mesh generation tool for biomedical models shows great promise. This method could easily be extended into other promising areas. Geotechnical analysis requires models similar to those found in biomedical applications—they are often irregularly shaped and possess little or no topology that can be used to define the structure needed by traditional hexahedral meshing schemes. Additionally, sheet insertion methods, like mesh cutting and grafting, can also be utilized to dramatically reduce the level of geometric decomposition that is traditionally required to generate a hexahedral mesh on complex geometries.

Acknowledgment

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

1. Scott A. Mitchell. A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volumes. In *13th Annual Symposium on Theoretical Aspects of Computer Science*, volume Lecture Notes in Computer Science: 1046, pages 465–476, 1996.
2. Jason F. Shepherd. *Topologic and Geometric Constraint-Based Hexahedral Mesh Generation*. Published Doctoral Dissertation, University of Utah, May 2007.
3. Steven J. Owen. A survey of unstructured mesh generation technology. `http://www.andrew.cmu.edu/user/sowen/survey/index.html`, September 1998.
4. Ted D. Blacker and Ray J. Meyers. Seams and wedges in plastering: A 3D hexahedral mesh generation algorithm. *Engineering With Computers*, 2(9):83–93, 1993.
5. Timothy J. Tautges, Ted D. Blacker, and Scott A. Mitchell. Whisker weaving: A connectivity-based method for constructing all-hexahedral finite element meshes. *International Journal for Numerical Methods in Engineering*, 39:3327–3349, 1996.
6. Matthias Muller-Hannemann. Hexahedral mesh generation by successive dual cycle elimination. In *Proceedings, 7th International Meshing Roundtable*, pages 365–378. Sandia National Laboratories, October 1998.
7. Nestor A. Calvo and Sergio R. Idelsohn. All-hexahedral element meshing: Generation of the dual mesh by recurrent subdivision. *Computer Methods in Applied Mechanics and Engineering*, 182:371–378, 2000.
8. Matthew L. Staten, Steven J. Owen, and Ted D. Blacker. Unconstrained paving and plastering: A new idea for all hexahedral mesh generation. In *Proceedings, 14th International Meshing Roundtable*, pages 399–416. Sandia National Laboratories, September 2005.
9. Erik D. Demain, Joseph S.B. Mitchell, and Joseph O'Rourke. `http://maven.smith.edu/~orourke/TOPP/P27.html#Problem.27`, 2006. The open problems project - problem 27: Hexahedral meshing.
10. W. A. Cook and W. R. Oakes. Mapping methods for generating three-dimensional meshes. *Computers In Mechanical Engineering*, CIME Research Supplement:67–72, August 1982.
11. John F. Dannenhoffer(III). A block-structuring technique for general geometries. In *29th Aerospace Sciences Meeting and Exhibit*, volume AIAA-91-0145, January 1991.
12. XYZ Scientific Applications, Inc. Truegrid: High quality hexahedral grid and mesh generation for fluids and structures. `http://www.truegrid.com`, October 2006.
13. Pointwise. Gridgen - reliable CFD meshing. `http://www.pointwise.com/gridgen`, October 2006.
14. ANSYS. ANSYS ICEM CFD, http://www.ansys.com/products/icemcfd.asp, October 2006.
15. David R. White, Mingwu Lai, Steven E. Benzley, and Gregory D. Sjaardema. Automated hexahedral mesh generation by virtual decomposition. In *Proceedings, 4th International Meshing Roundtable*, pages 165–176. Sandia National Laboratories, October 1995.
16. Patrick Knupp. Next-generation sweep tool: A method for generating all-hex meshes on two-and-one-half dimensional geometries. In *Proceedings, 7th In-*

ternational Meshing Roundtable, pages 505–513. Sandia National Laboratories, October 1998.
17. Xevi Roca, Josep Sarrate, and Antonio Huerta. Surface mesh projection for hexahedral mesh generation by sweeping. In *Proceedings, 13th International Meshing Roundtable*, volume SAND 2004-3765C, pages 169–180. Sandia National Laboratories, September 2004.
18. Mingwu Lai, Steven E. Benzley, and David R. White. Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping. *International Journal for Numerical Methods in Engineering*, 49(1):261–275, September 2000.
19. Scott A. Mitchell and Timothy J. Tautges. Pillowing doublets: Refining a mesh to ensure that faces share at most one edge. In *Proceedings, 4th International Meshing Roundtable*, pages 231–240. Sandia National Laboratories, October 1995.
20. Darryl J. Melander, Timothy J. Tautges, and Steven E. Benzley. Generation of multi-million element meshes for solid model-based geometries: The dicer algorithm. *AMD - Trends in Unstructured Mesh Generation*, 220:131–135, July 1997.
21. Ko-Foa Tchon, Julien Dompierre, and Ricardo Camarero. Conformal refinement of all-quadrilateral and all-hexahedral meshes according to an anisotropic metric. In *Proceedings, 11th International Meshing Roundtable*, pages 231–242. Sandia National Laboratories, September 2002.
22. Nathan Harris, Steven E. Benzley, and Steven J. Owen. Conformal refinement of all-hexahedral meshes based on multiple twist plane insertion. In *Proceedings, 13th International Meshing Roundtable*, pages 157–168. Sandia National Laboratories, September 2004.
23. Michael J. Borden, Jason F. Shepherd, and Steven E. Benzley. Mesh cutting: Fitting simple all-hexahedral meshes to complex geometries. In *Proceedings, 8th International Society of Grid Generation Conference*, 2002.
24. Steven R. Jankovich, Steven E. Benzley, Jason F. Shepherd, and Scott A. Mitchell. The graft tool: An all-hexahedral transition algorithm for creating multi-directional swept volume mesh. In *Proceedings, 8th International Meshing Roundtable*, pages 387–392. Sandia National Laboratories, October 1999.
25. Sandia National Laboratories. The CUBIT Geometry and Mesh Generation Toolkit. `http://cubit.sandia.gov`, 2007.
26. Scientific Computing and Imaging Institute (SCI). SCIRun: A Scientific Computing Problem Solving Environment. `http://software.sci.utah.edu/scirun.html`, 2002.
27. S.R. Jankovich, S.E. Benzley, J.F. Shepherd, and S.A. Mithcell. The graft tool: an all-hexahedral transition algorithm for creating a mult-directional swep t volume mesh. In *Proceedings, 8^{th} International Meshing Roundtable '99*, pages 387–392. Sandia National Laboratories, 1999.