SAND2007-6343C

APCOM'07 in conjunction with EPMESC XI, *December 3-6, 2007, Kyoto, JAPAN*

New Strategies for Unstructured All-Hexahedral Mesh Generation

Steven J. Owen[1]*, Matthew L. Staten[1], Michael Borden[1] and Jason Shepherd[1]

[1] *Computational Modeling Sciences, Sandia National Laboratories*[†]*, PO Box 5800 M.S. 0376, Albuquerque, New mexico, U.S.A.*
e-mail: sjowen@sandia.gov, mlstate@sandia.gov, mborden@sandia.gov, jfsheph@sandia.gov

Abstract Four specific strategies for generating all-hexahedral meshes for computational simulation are presented. These range from fully automatic approaches to user interactive. They include 1) Many to many sweeping: an extension of the traditional sweeping; 2) Unconstrained Plastering: A fully automatic hexahedral decomposition tool; 3) Sheet Insertion: An approach to direct hexahedral topology manipulation and 4) Immersive Topology Environment for Meshing (ITEM): A user interactive environment built on automation algorithms. These represent the current technical research and development strategy Sandia National Laboratories has employed to help meet the needs of it computational analysts today and into future.

Key words: hexahedra, mesh generation, sweeping, sheet insertion, unconstrained plastering

INTRODUCTION

Automatic mesh generation is a key component of modeling and simulation. Although tetrahedral methods seems to have flourished in the industry, there is still a significant sector that continues to demand high quality hexahedral meshes. In spite of this, there continues to be a conspicuous lack of fully-automatic hexahedral methods for arbitrary volumes. Because of the benefits afforded by hexahedral meshes, analysts will often expend tremendous time and effort in their construction.

Sandia National Laboratories has a rich history of research and development of all-hexahedral mesh generation. While not all techniques have been successful, significant understanding of the challenges presented by hexahedral meshing have been gained. Drawing on the successes and lessons learned, Sandia is currently persuing several strategies in parallel, most of which are built on the CUBIT Geometry and Meshing Toolkit[1]. This includes a toolbox approach to mesh generation, where it is understood that a one-size-fits-all algorithm is unlikely to be developed nor would such a tool be acceptable for all applications. Instead, a variety of strategies focused on building tools that will be most effective for generating all-hexahedral meshes are currently under development. An outline of four specific strategies, currently at various levels of maturity, will be presented. These include, many-to-many sweeping, unconstrained plastering, sheet insertion and an immersive topology environment for meshing (ITEM).

Each strategy under development involves different levels of user interaction and generality. Many-to-many sweeping [2] involves an automatic decomposition technique that generalizes the traditional sweeping approach and allows for multiple source and multiple target topology. It is however limited to single axis geometric extrusions which may require initial user decomposition to separate uniaxial domains. The more general approach of sheet insertion [3] begins with a base mesh, typically a Cartesian grid or swept mesh, and inserts topological sheets or layers of hexes to enforce geometric conformity where needed. This technique is completely general in the geometries it may address, but currently may require user interaction to provide the best base mesh and locations for sheet insertions. Unconstrained plastering, also a geometry decomposition technique, inserts topological layers beginning from the boundary of the solid, that result in simplified regions that can be more easily meshed with traditional hexahedral techniques. While still under active development, unconstrained plastering [4,5] is

[†] 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

intended to address the general geometry problem with no user interaction. Finally, the immersive topology environment for meshing (ITEM) [6] is a wizard-like tool built on an extensive infrastructure of geometry preparation and meshing tools. ITEM provides an interactive environment where through a series of diagnostic tests, specific and intelligent solutions for preparing and decomposing geometry to admit a hex-meshable topology are presented to the user.

Taken together, these four strategies constitute a significant strategic investment in all-hexahedral meshing technology. They include robust techniques used on a daily basis in a demanding engineering environment. They also include more visionary algorithms that attempt to ultimately reduce the tedious work and effort currently required for all-hexahedral mesh generation. While not intended to be a full description of each strategy, this paper provides a basic overview of each technique along with representative examples to illustrate strengths and weaknesses of each approach.

MANY-TO-MANY SWEEPING

The pave and sweep approach [7,8,9,10] is currently the dominant tool for generating complex, unstructured all-hexahedral meshes. This approach, however, often requires extensive interactive geometry decomposition and user input to create sweepable sub-volumes from the original geometry. It is also difficult to recognize a sweep compatible decomposition by visual inspection---especially as the complexity of the original model increases. This can lead to a time-consuming trial and error manual decomposition approach until the correct sub-volumes are created.

When generating a hexahedral mesh with a sweeping algorithm the mesh from a set of source surfaces is extruded through the volume onto a target surface. For example, *Figure 1* shows a simple many-to-one sweepable volume. The sweep begins with the mesh on the surface labeled S1 and is extruded until it reaches the surface labeled S2. The meshes of these two surfaces are then combined and the extrusion continues until it reaches the surface labeled T1.

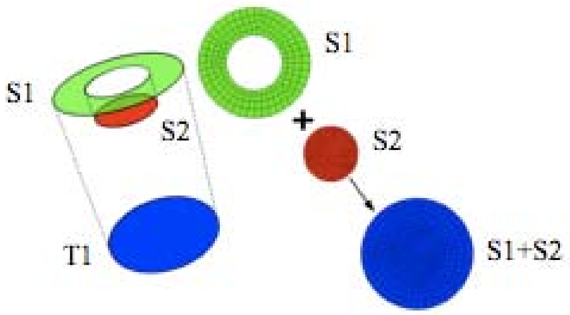


Figure 1. The source surfaces (S1 and S2) of a many-to-one sweep are combined as they are extruded through the volume to create the mesh for the target surface (T1).

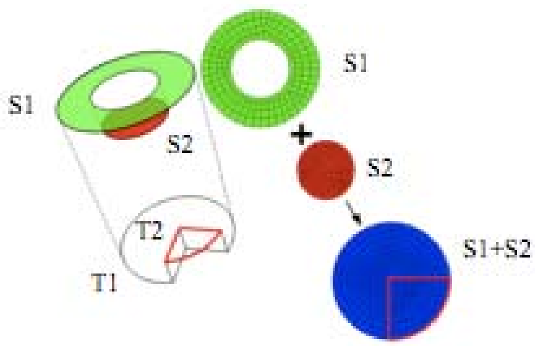


Figure 2. If the source surfaces (S1 and S2) of a many-to-many sweep do not take into account the topology of the target surface (T2) when they are meshed, the mesh and target surface will not match.

Many-to-many sweeps are difficult because multiple target surfaces add additional constraints to the source mesh as shown in *Figure 2*. In this case the volume is similar to the one shown in *Figure 1* except that an additional target surface, labeled T2, has been added. This surface adds the constraint that the combined mesh for S1 and S2 must align with the target boundary on T2. The difficulty is caused because information about the topology of T2 is needed when meshing S1 and S2. Creating this information is the challenge of many-to-many sweeping and has been the focus of considerable research and development over the years [2,11,12].

An example of a volume that requires decomposition before meshing is shown in *Figure 3*. Notice that this volume has a cylindrical-like topology that suggests a potential sweep direction. However, because there are multiple cap surfaces on both the top and bottom of the volume it cannot be meshed with a one-to-one or many-to-one algorithm. To mesh this volume manually, the process outlined in *Figure 4* is used. This decomposition results in six one-to-one or many-to-one sub-volumes. Each volume is then meshed individually with care being taken to insure that the mesh between shared volumes is conformal.

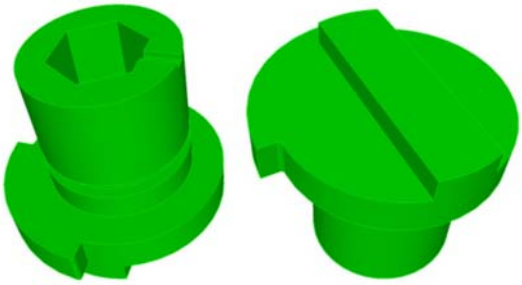


Figure 3. Top and bottom view of a volume that requires decomposition before meshing

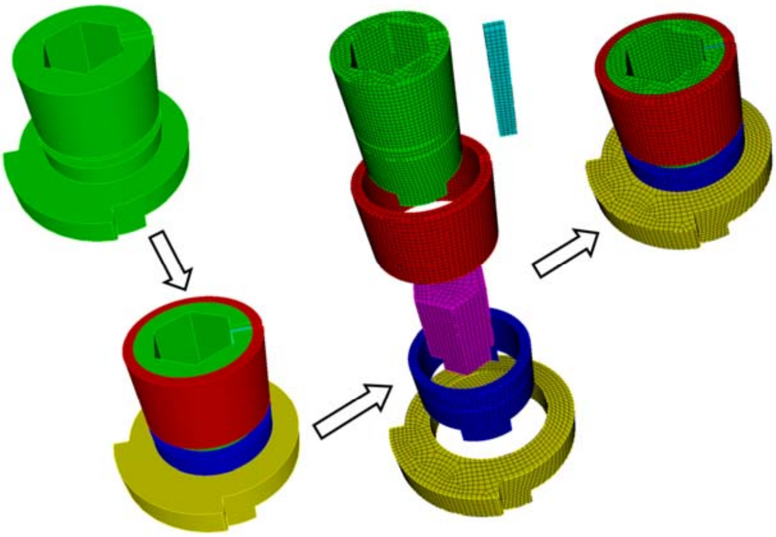
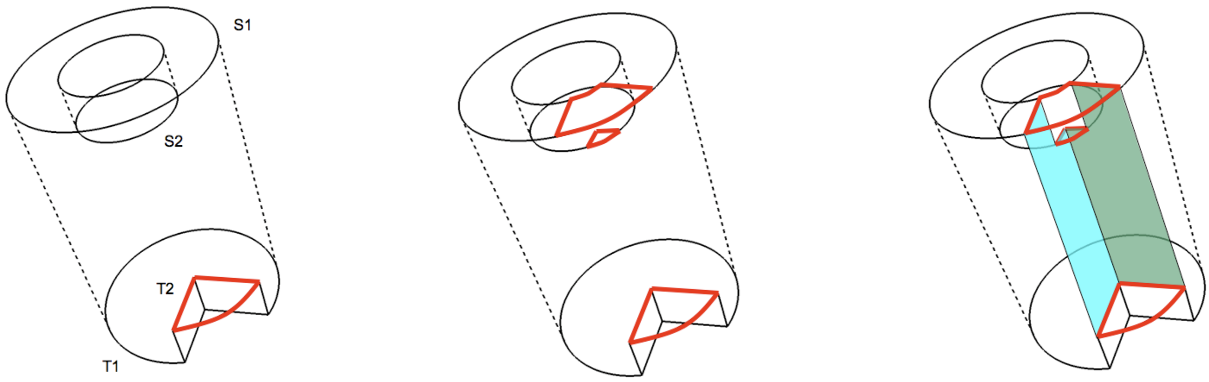


Figure 4. Illustration of the decomposition required to manually mesh a many-to-many sweepable volume.

To reduce user input, many-to-many sweeping attempts to automatically decompose many-to-many sweepable volumes into one-to-one or many-to-one sub-volumes. The process is illustrated in *Figure 5*. In order to create a valid mesh for this volume the mesh on surface S1 and S2 will need to align with the boundary of surface T2 (see *Figure 5a*). To do this, the bounding curves of T2 are projected onto the S1 and S2 as shown in *Figure 5b*. Surfaces S1 and S2 are then imprinted with the projected curves. Partitions that are internal to the volume are then created between the original boundary and the projected imprints as shown in *Figure 5c*. These internal partitions are used to decompose the volume into the two many-to-one volumes shown in Figure 6.

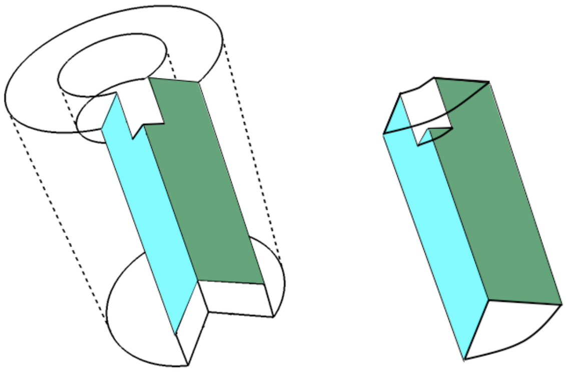


a *b* *c*

Figure 5. Steps to automatic decomposition: a) identify target features, b) project target topology and imprint with sources, c) create internal partitions.

Once the automatic decomposition has completed traditional sweeping algorithms can then be used to generate the mesh on the individual sub-volumes. The mesh can then be transferred back to the original volume as a final step so that the entire process becomes transparent. Some benefits of this approach are:

1) less user input and expertise is required, 2) shorter time to mesh, and 3) potential quality improvement because the final mesh is not constrained by the interior surfaces created during decomposition.



a *b*

Figure 6. Final many-to-one sweepable volumes after decomposition from Figure 5

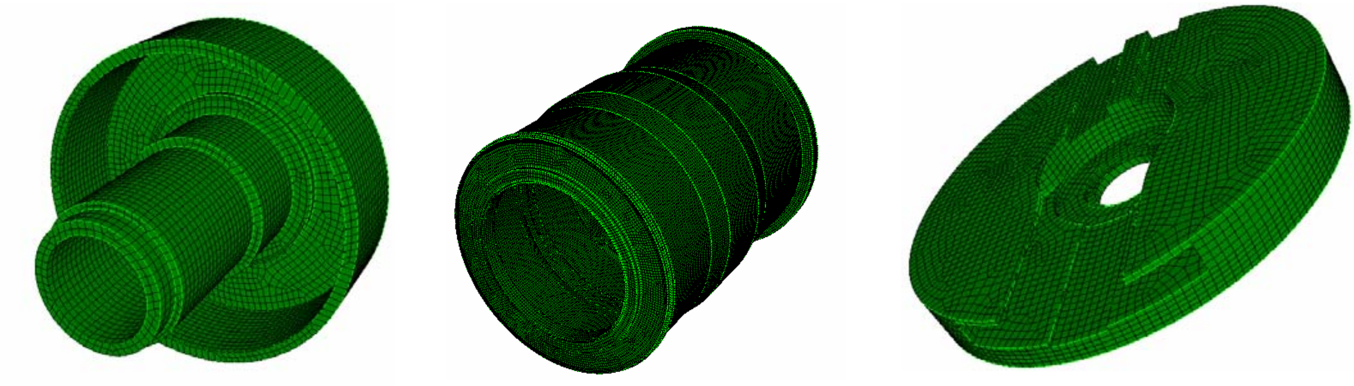
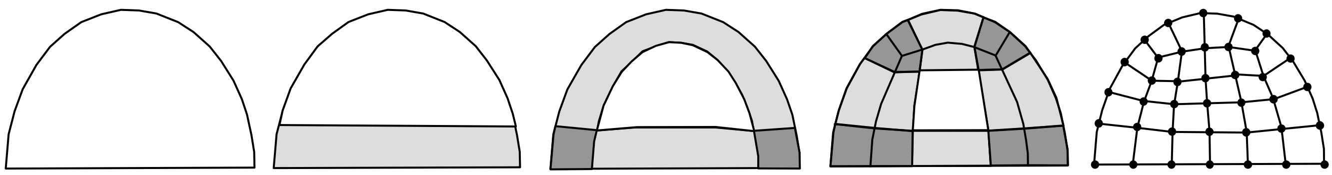


Figure 7. Examples of mechanical parts automatically meshed using many-to-many sweeping

With the deployment of many-to-many sweeping, the generality of the topologies that can now be meshed using the pave and sweep approach has increased significantly (see *Figure 7*). Notwithstanding, the technique is limited to a single sweep direction and may still require user intervention to decompose the volume. Recognizing this fact, there is still a strong need to provide a fully automated hexahedral meshing solution for arbitrary topologies. The Unconstrained Plastering research is one approach to solving this problem.

UNCONSTRAINED PLASTERING

Unconstrained Plastering [4,5] is an automatic advancing front geometry decomposition method which advances fronts from an unmeshed volume boundary. Prior research on a general hexahedral meshing technique, such as Whisker Weaving[13], traditional Plastering[14], and Many-to-Many Sweeping[2], require the boundary to first be meshed with a completely defined quadrilateral mesh. This boundary quadrilateral mesh serves as a boundary constraint to which the interior generated hexahedra must conform. In contrast, Unconstrained Plastering starts from an unmeshed volume boundary. The quadrilateral connectivity of the final mesh is a by-product of the interior meshing process rather than a pre-defined constraint. This gives Unconstrained Plastering significantly more freedom (i.e. unconstraining the problem) as it generates the interior hexahedra.



a) One row advanced *b) Intermediate step* *c) Intermediate step* *d) Intermediate step* *e) Final mesh*

Figure 8. Unconstrained Paving of a simple geometric surface.

The 2D corollary to Unconstrained Plastering is Unconstrained Paving, which generates all-quadrilateral meshes on general geometric surfaces. *Figure 8* illustrates the Unconstrained Paving process on a simple surface. The process begins in Figure 1a with a surface with an unmeshed boundary. In surface meshing, the entire domain has two degrees of freedom, corresponding to the two inherent directions in a quadrilateral element. In *Figure 8b* a single front is advanced. The front is advanced using a simple geometric offset and results in a constraint to one of the degrees of freedom behind the advanced front. This means that the domain behind the front advance will eventually be a single row of quadrilateral elements, however, at this point, the number of quadrilaterals in this row is undetermined. In *Figure 8c* and *d*, additional fronts are advanced. In the corners of the surface, two fronts advance over the same domain coming from opposite directions, both of which constrain one of the two degrees of freedom. As a result, the domain where two advancing fronts intersect create a full-quadrilateral element in the final mesh. Fronts are advanced until the unconstrained region becomes small enough to fill with a simple templated mesh yielding the final mesh for this model in *Figure 8e*.

In 3D, Unconstrained Plastering uses the same procedure of advancing unconstrained fronts from an unmeshed volume boundary using discrete geometric surface offsets. In 3D, the domain has three degrees of freedom corresponding to the three inherent directions in a hexahedral element. As in 2D, advancing a a single front constrains only one of these degrees of freedom, resulting in what will eventually be a sheet of hexahedral elements. Additionally, advancing a single front constrains only the direction perpendicular to the front, leaving the size, orientation, and number of elements in the sheet undefined at this point. In corners, where two fronts intersect, a column of hexahedral elements is created. In corners, where three fronts intersect, a single hexahedral element is created in the final mesh. Advancing each of the fronts is accomplished using a discrete geometric offset approach similar to the one described in [15].

In addition to advancing simple fronts, proximity of the front with other fronts must be continually computed and updated. Proximity between nearby fronts is resolved prior to advancing by fitting a constrained sheet in between the fronts involved. This proximity resolution is the most time-consuming part of the Unconstrained Plastering process, requiring significant amount of care and precision. However, the fact that the boundaries of the unconstrained region are themselves unconstrained gives significant amounts of freedom to fit a proper mesh in the unmeshed voids.

In 3D, fronts are advanced until the unmeshed region is recognizable as either mappable [16], sweepable [10], or midpoint-subdividable [17]. *Figure 9* illustrates three models meshed with Unconstrained Plastering. The quality of the meshes generated is similar to that of pave-and-sweep meshes, where typically, the minimum scaled Jacobian is >0.5.

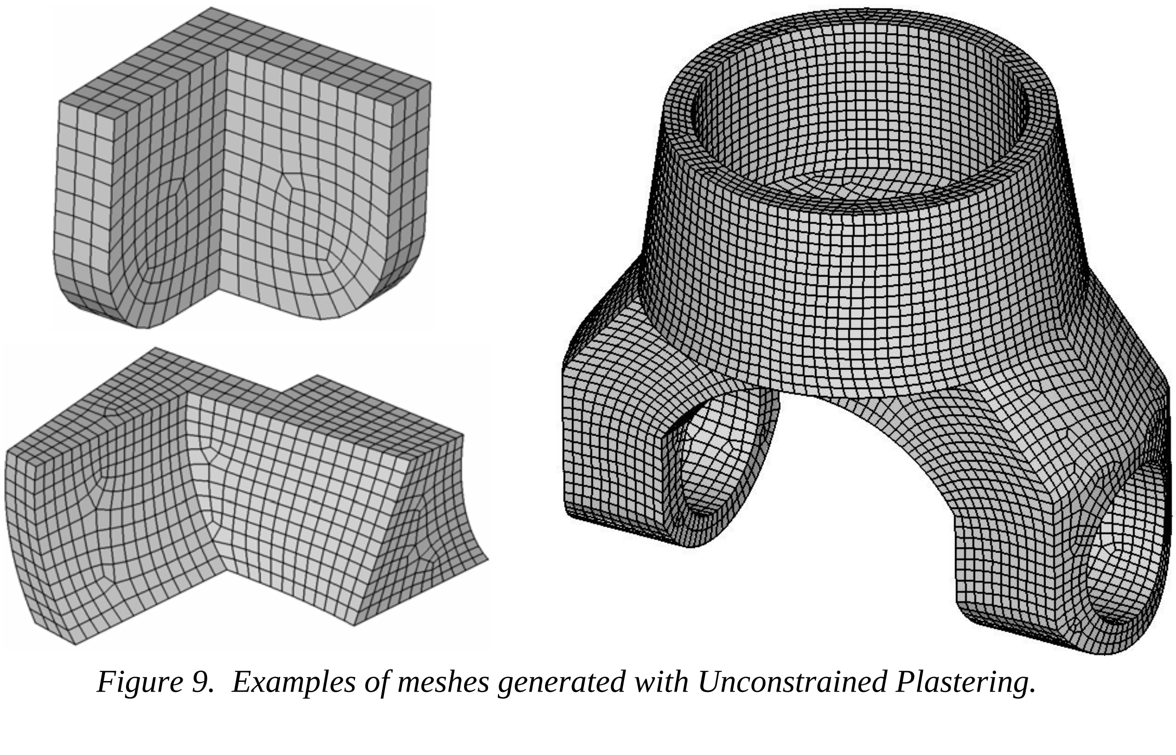


Figure 9. Examples of meshes generated with Unconstrained Plastering.

Unconstrained plastering remains an open area of research rather than a customer ready tool. Research on the resolution of proximities at complex intersections, extension to assembly models, and sensitivity to floating point round-off remains active areas of research.

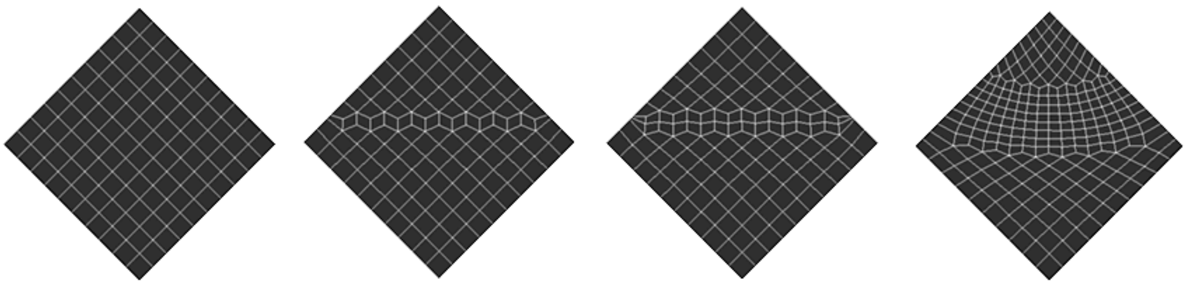
One of the attractive characteristics of unconstrained plastering is the ability to produce a high quality mesh at the boundaries. While this can be an advantage, it can also increase the complexity of the geometric problem being solved. Interior regions must be effectively resolved into primitive regions that can be meshed with a known technique. This problem has yet to be solved in a completely robust and reliable manner. A more general hexahedral technique, that takes a different approach to resolving geometric boundaries is that of *sheet insertion*.

SHEET INSERTION

Sheet insertion [18] begins with a base mesh, typically a Cartesian grid or swept mesh, and inserts topological sheets or layers of hexes to enforce geometric conformity where needed. In the past, traditional grid-based methods such as those described in [19,20] relied only a base Cartesian grid where nodes and elements were snapped and smoothed to conform to surface features, often leaving poor quality elements at the boundary. Sheet insertion improves on this technique by generalizing the type of base mesh that can be utilized and improving the boundary element quality. In addition, sheet insertion can also be used to capture analytic features in hexahedral meshes, including boundary layers. This technique is completely general in the geometries it may address, but currently may require user interaction to provide the best base mesh and locations for sheet insertions.

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 to the mesh and the new elements introduced by the insertion provide methods for defining, refining, and improving the quality of a mesh. Pillowing[21] is the most common type of generalized sheet insertion operation; however, other methods exist for specialized sheet insertion including dicing[22], refinement[23], grafting[24], and mesh cutting[25].

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, a layer of hexahedra can be identified by obtaining a set of edges in one of the three parametric directions. Next, all neighboring hexahedra sharing these edges are collected. This process continues iteratively using each neighboring hexahedron to obtain the set of edges in the same parametric direction as the initial hexahedron and collecting all hexahedra sharing these edges. 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.



a *b* *c* *d*

Figure 10. Inserting layers of hexahedra, or sheets, into a mesh.

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 correct set of topological constraints for a hexahedral mesh [3], the resulting mesh with the new sheet will still be a conformal hexahedral mesh. Using this principle, a hexahedral mesh may be modified by inserting new hexahedral sheets to create new meshes, define new mesh boundaries, refine an existing mesh, or improve the quality of a hexahedral mesh. In *Figure 10* below, we demonstrate sheet insertion using a pillowing technique [21] demonstrating the changes resulting from insertion of one to many layers of hexahedra. In *Figure 10b*, a single layer of hexahedra has been inserted. The inserted layer results in the edges within that layer being aligned orthogonally to the new sheet, i.e., the edges defining the sheet are geometrically adaptive to the insertion. *Figure 10c* shows the insertion of two layers of hexahedra in the mesh. The shared quadrilaterals between the two layers of hexahedra can be aligned to geometrically approximate a surface where the sheet is inserted. Because the shared quadrilaterals can approximate a surface, it is possible to use double sheet insertion as a method for performing Boolean-like cutting of the mesh [25,26]. *Figure 10d* demonstrates multiple layer insertion which improves the topological structure in the vicinity of the inserted sheets which may be useful in capturing analytic features within existing meshes, including boundary layers and shock fronts. Several example meshes utilizing sheet insertion to capture geometric or analytic features are shown in *Figure 11*, to *Figure 13*.

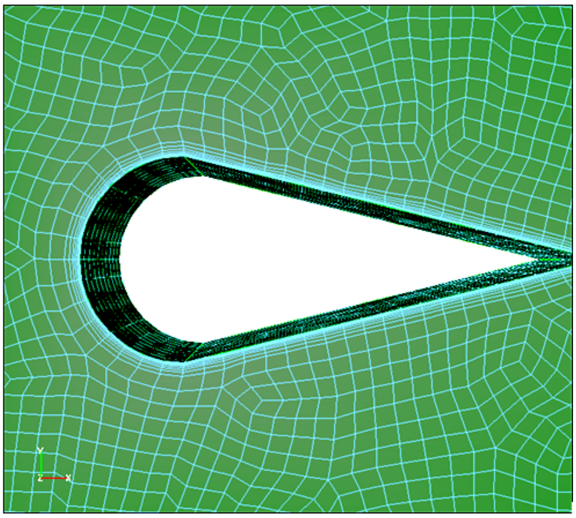


Figure 11. Layers of hexahedra inserted around wing producing boundary layers within the mesh.

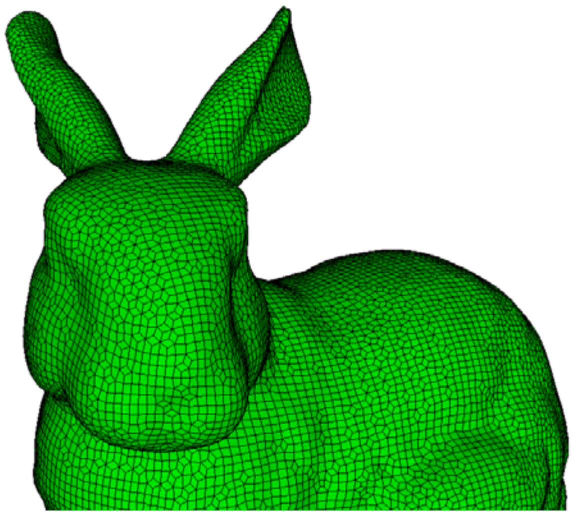


Figure 12. Hexahedral mesh of an organic model created using sheet insertion techniques.

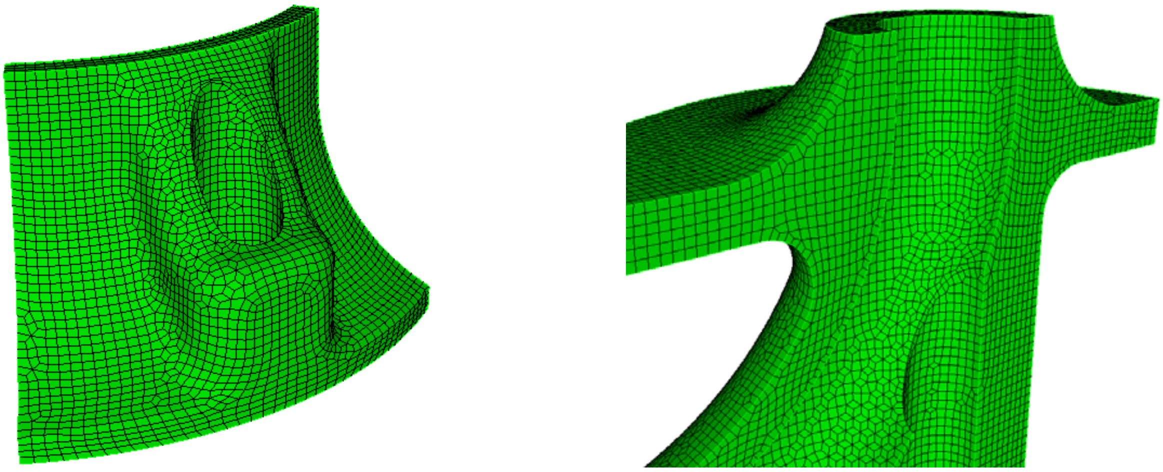


Figure 13. Hexahedral meshes of mechanical parts created by inserting sheets to capture geometric features of the boundary.

The sheet insertion methods described here, provide a distinct advantage on the generality of the models that can be meshed over those that can currently be addressed by sweeping and even unconstrained plastering. Research is however ongoing to automate the procedures. For example, the selection and orientation of the base mesh can significantly change the characteristics and quality of the mesh. The user must currently define a base mesh on which sheet insertion operations will be performed. Additionally, automatic selection of the appropriate sheet insertion tool to best capture geometry and topology is a future research objective. Recognizing the inherent manual nature of many of the current hexahedral meshing technologies available today, the final approach described in this paper attempts to utilize existing robust tools, but provides assistance to the user through a smart *diagnostic-solution* approach.

IMMERSIVE TOPOLOGY ENVIRONMENT FOR MESHING (ITEM)

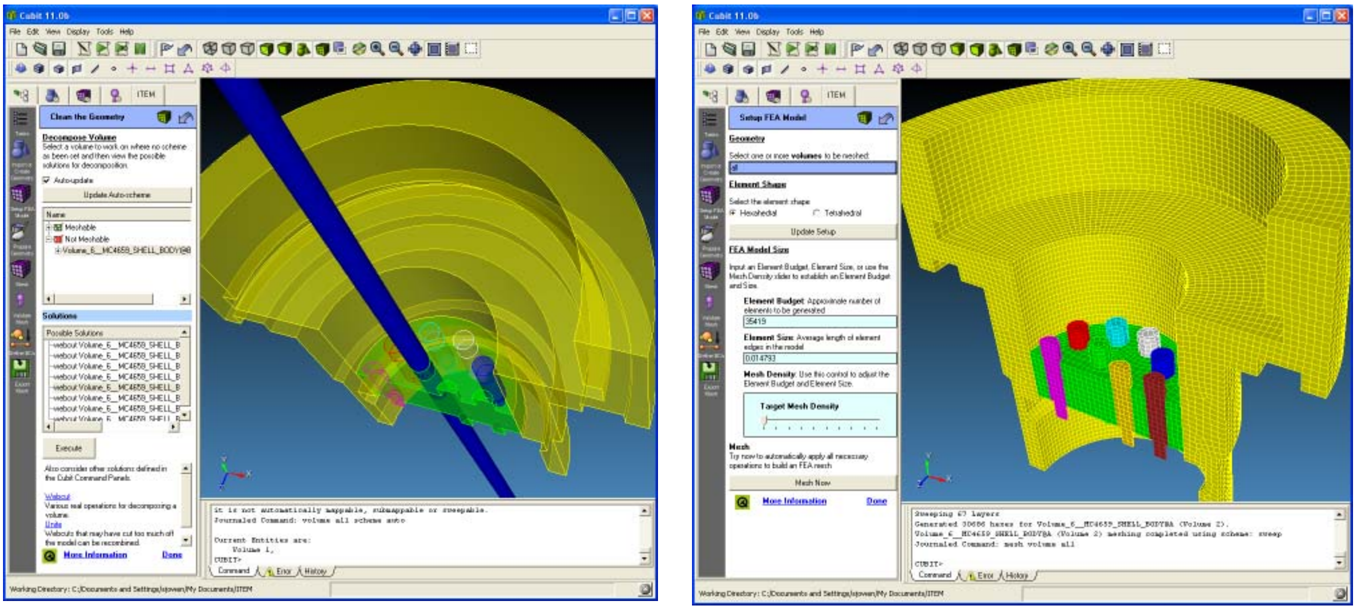
Many of the tools currently available for hexahedral meshing can produce high quality meshes ideally suited for simulation. The shortcoming of these tools however is their non-generality for arbitrary topologies, often requiring extensive expertise to prepare the geometry prior to applying an appropriate meshing algorithm. Preparation may involve simplifying, repairing or removing small features from a CAD model as well as decomposing or cutting the model to suit the requirements of the meshing algorithm as well as assigning specific meshing schemes and intervals. All of this may be overwhelming to a user, particularly if the meshing task is performed only on a intermittent basis.

With the ultimate goal of reducing the time to generate a mesh for simulation, The Immersive Topology Environment for Meshing (ITEM) [6] has been developed within the CUBIT Geometry and Meshing Toolkit [1] to take advantage of its extensive tool suite. Built on top of these tools it attempts to improve the user experience by accomplishing three main objectives:

1. Guiding the user through the workflow: As shown in *Figure 14*, ITEM provides a wizard-like environment that steps the user through the geometry and meshing process. For someone unfamiliar with the software, it provides an interactive, step-by-step set of tools for accomplishing the major tasks in the process.

2. Providing the user with smart options: Based on the current state of the model, ITEM will automatically run diagnostics and determine potential solutions that the user may consider. For example, where unwanted small features may exist in the model, ITEM will direct the user to these features and provide a range of geometric solutions to the problem.

3. Automating geometry and meshing tasks: For various characteristic geometric problems that are encountered in a solid model, ITEM can determine from the potential geometric solutions, which may be most applicable and apply that solution without any user intervention. For many configurations of geometry, a completely automated solution may be available.



a) ITEM detects and previews an option for decomposing the model for hexahedral mesh.

b) The final hexahedral mesh after using ITEM to help prepare the CAD model.

Figure 14. An example of the ITEM user interface within CUBIT used to mesh a CAD model

In addition to providing diagnostics and solutions to cleanup and prepare a CAD model, ITEM provides a number of tools to assist the user with the hexahedral mesh generation problem. These include detecting and suggesting decomposition operations, recognizing nearly sweepable topologies and suggesting source-target pairs and detecting and compositing surfaces to force a sweep topology.

The ITEM algorithms determine possible decompositions and suggest these to the user. The user can then make the decision as to whether a particular cut is actually useful. This process helps guide new users by demonstrating the types of decompositions that may be useful. It also aids experienced users by reducing the amount of time required to set up decomposition commands. The current algorithm for suggesting webcuts is based upon an auto scheme selection technique [27] and geometric reasoning for decomposition [28] but provides additional interactive alternatives to the user.

Figure 15 shows an example scenario for using this tool. The simple model at the top is analyzed using the above algorithm. This results in several different solutions being offered to the user, three of which are illustrated here. As each of the options is selected, the extended cutting surface is displayed providing rapid feedback to the user as to the utility of the given option. The user can then quickly apply the decomposition and the model will be updated to reflect the changes.

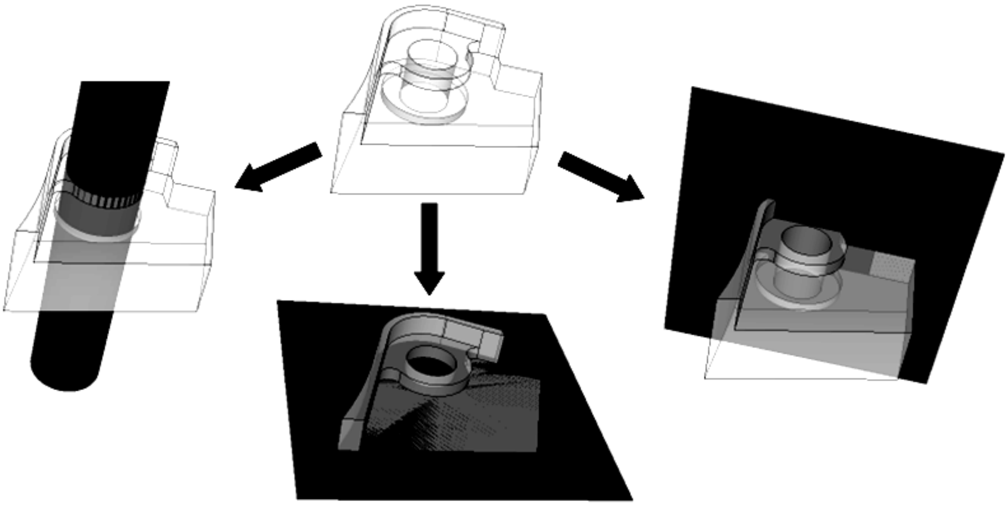
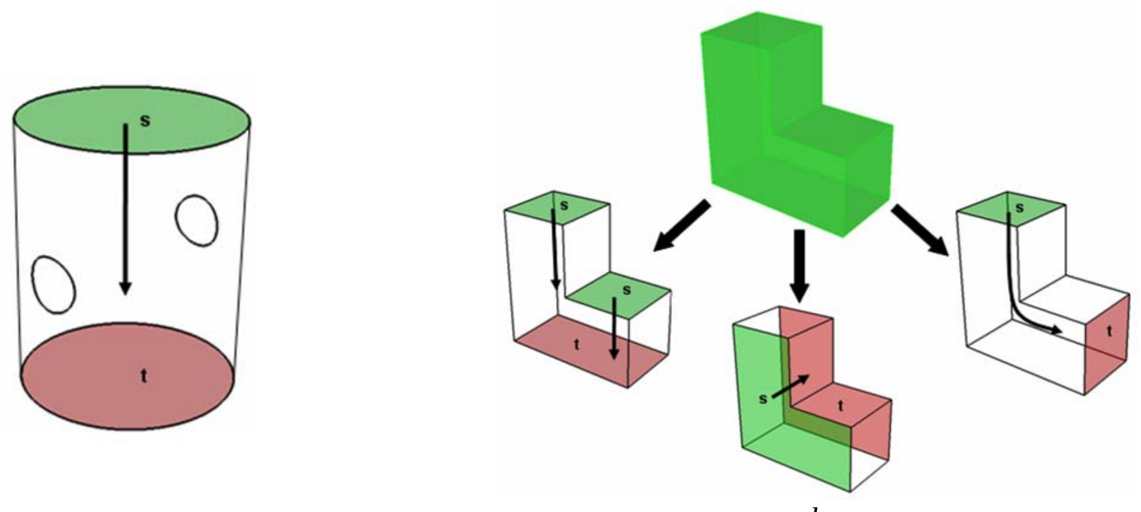


Figure 15. ITEM decomposition tool shows 3 of the several solutions generated that can be selected to decompose the model for hex meshing



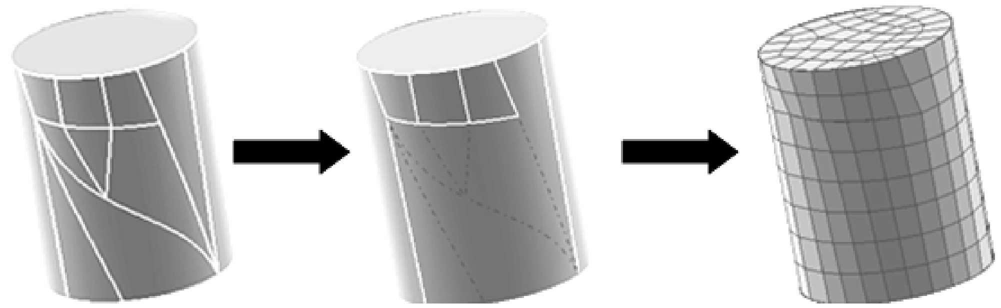
a *b*

Figure 16. (a) ITEM displays the source and target of a geometry that is nearly sweepable. The region is not currently sweepable due to circular imprints on the side of the cylinder. (b) Alternative feasible sweep directions are also computed

The purpose of geometry operations such as decomposition is to transform an unmeshable region into one or more meshable regions. However, even the operations suggested by the decomposition tool can degenerate into guesswork if they are not performed with a specific purpose in mind. Without a geometric goal to work toward, it can be difficult to recognize whether a particular operation will be useful.

Incorporated within the ITEM environment are algorithms that are able to detect geometry that is nearly sweepable, but which are not fully sweepable due to some geometric feature or due to incompatible constraints between adjacent sections of geometry. By presenting potential sweeping configurations to the user, ITEM provides suggested goals to work towards, enabling the user to make informed decisions while preparing geometry for meshing. *Figure 16* shows two different examples of scenarios that the algorithm is designed to detect and present "sweep suggestion" alternatives to the user.

In some cases, decomposition alone is not sufficient to provide the necessary topology for sweeping. The forced sweepability capability included in ITEM attempts to force a model to have sweepable topology given a set of source and target surfaces. To use this tool the user identifies a source-target pair. This pair may have been selected manually by the user, or defined as one the solutions from the sweep suggestion algorithm described above. The force sweep tool will then combine or *composite* all of the surfaces between the identified source and target pair, with the objective of creating mappable or submappable regions. These composited *linking* surfaces, as they are referred to, can then provide the appropriate topology to apply the pave and sweep method. *Figure 17* shows an example of where the forced sweep capability is used.



a *b* *c*

Figure 17. Non-submappable linking surface topology is composited out to force a sweepable volume topology. (a) Initial topology does not admit a sweep, (b) linking surfaces automatically composited, (c) final swept mesh ignoring some of the original surface topology.

The Immersive Topology Environment for Meshing (ITEM) addresses a wide range of problems and issues commonly encountered during process of preparing a model for simulation. Its intent is to reduce the learning, and re-learning often associated with complex software tools and to ultimately reduce the time to mesh. Although the ultimate objective will be to eventually provide high quality hexahedral meshing tools that will have little or no human intervention, providing more effcicient and smart diagnostics achieves part of the objective by providing more efficient tools allowing faster turn around on modeling and simulation tasks.

CONCLUSION

High quality hexahedral mesh generation continues to be demanded by many sectors of the modeling and simulation community. While there have been significant advances in automation of hexahedral meshing, a fully automatic method for arbitrary geometry and topology has yet to be deployed. The all-hex problem, as it has become known, is the focus of ongoing research and development at Sandia National Laboratories. Presented in this paper, are four strategies currently being actively researched and developed. Each technology, while providing many demonstrated advantages, has not yet proved to be the one ultimate solution. From long experience, we have recognized that a single strategy is unlikely to address all modeling scenarios. As a result, a strategic investment in each of these technologies will likely prove to be the most prudent course of action for today and the future.

REFERENCES

[1] CUBIT Geometry and Meshing Toolkit, Sandia National Laboratories, Albuquerque, New Mexico, U.S.A. http://cubit.sandia.gov

[2] D. R. White, S. Saigal, S. J. Owen, CCSweep: *Automatic decomposition of multi-sweep volumes*, Engineering with Computers, 20, (2004), 222-236

[3] J. F. Shepherd, *Topologic and Geometric Constraint-Based Hexahedral Mesh Generation*, Dissertation, University of Utah (2007)

[4] M. L. Staten, S. J. Owen, T. D. Blacker, *Unconstrained Paving & Plastering: A New Idea for All Hexahedral Mesh Generation*, Proceedings, 14th International Meshing Roundtable, (2005), 399-416

[5] M. L. Staten, R. A. Kerr, S. J. Owen, T. D. Blacker, *Unconstrained Paving and Plastering: Progress Update,* 15th International Meshing (2006), 469-486

[6] S. J. Owen, B. W. Clark, D, J. Melander, M. Brewer, J. F. Shepherd, K. Merkley, C. Ernst, R. Morris, *An Immersive Topology Engine for Meshing*, accepted to 16th International Meshing Roundtable (2007)

[7] M. L. Staten, S. A. Canann, S. J. Owen, *BMSWEEP: Locating Interior Nodes During Sweeping*, 7th International Meshing Roundtable (1998), 7-18

[8] P. M. Knupp, *Next-Generation Sweep Tool: A Method For Generating All-Hex Meshes On Two-And-One-Half Dimensional Geomtries*, Proceedings, 7th International Meshing Roundtable, (1998), 505-513

[9] X. Roca, J. Sarrate, *An Automatic and General Least-Squares Projection Procedure for Sweep Meshing*, Proceedings, 15th International Meshing Roundtable, (2006), 487-506

[10] M. A. Scott, M. N. Earp, S. E. Benzley, M. B. Stephenson, *Adaptive Sweeping Techniques*, Proceedings, 14th International Meshing Roundtable, (2005), 417-432

[11] T. D. Blacker, *The Cooper Tool*, Proceedings, 5th International Meshing Roundtable, (1996), 13-30

[12] M. Lai, S. E. Benzley, D. R. White, *Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping*, International Journal for Numerical Methods in Engineering, 49, (2000), 261-275

[13] T. J. Tautges, T. D. Blacker, S. A. Mitchell, *The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes*, International Journal for Numerical Methods in Engineering, 39, (1996), 3327-3349

[14] T. D. Blacker, R. J. Meyers, *Seams and Wedges in Plastering: A 3D Hexahedral Mesh Generation Algorithm*, Engineering with Computers, 2, (1993) 83-93

[15] X. Jiao, *Facet Offsetting: A Unified approach for explicit moving interfaces*, Journal of Computational Physics, 220(2): 612-625

[16] W. A. Cook, W. R. Oakes, *Mapping Methods for Generating Three-Dimensional Meshes*, Computers In Mechanical Engineering, CIME Research Supplement, (August 1982) 67-72

[17] T. S. Li, R. M. McKeag, C. G. Armstrong, *Hexahedral Meshing Using Midpoint Subdivision and Integer Programming*, Computer Methods in Applied Mechanics and Engineering, 124, (1995) 171-193

[18] K. Merkley, C. D. Ernst, J. F. Shepherd, M. Borden, *Methods and Applications of Generalized Sheet Insertion for Hexahedral Meshing*, accepted to the 16th International Meshing Roundtable, (2007)

[19] R. Schneiders, R. Bunten, *Automatic Generation of Hexahedral Finite Element Meshes*, Computer Aided Geometric Design, Elsevier, 12, (1995) 693-707

[20] Dhondt, Guido D. *Unstructured 20- node brick element meshing*, Computer Aided Design, 33, (2001) 233-249

[21] S. A. Mitchell, T. J. Tautges, *Pillowing doublets: Refining a mesh to ensure that faces share at most one edge*. Proceedings, 4th International Meshing Roundtable, (1995), 231–240

[22] D. J. Melander, T. J. Tautges, S. E. Benzley, *Generation of multi-million element meshes for solid model-based geometries: The dicer algorithm*. AMD - Trends in Unstructured Mesh Generation, 220 (1997), 131–135

[23] N. Harris, S. E. Benzley, S. J. Owen. *Conformal refinement of all-hexahedral meshes based on multiple twist plane insertion*. Proceedings, 13th International Meshing Roundtable, (2004),157–168

[24] S. R. Jankovich, S. E. Benzley, J. F. Shepherd, S. A. Mitchell, *The graft tool: An all-hexahedral transition algorithm for creating multi-directional swept volume mesh*, Proceedings, 8th International Meshing Roundtable, (1999) 387–392

[25] M. J. Borden, J. F. Shepherd, S. E. Benzley, *Mesh cutting: Fitting simple all-hexahedral meshes to complex geometries*. Proceedings, 8th International Society of Grid Generation, (2002)

[26] J. F. Shepherd, Y. Zhang, C. Tuttle, C. Silva, *Quality Improvement and Boolean-like Cutting Operations in Hexahedral Meshes*, 10th Conference of the International Society of Grid Generation, (2007)

[27] D. R. White, T. J. Tautges, *Automatic scheme selection for toolkit hex meshing*, International Journal for Numerical Methods in Engineering, 49, (2000),127-144

[28] Y. Lu, R. Gadh, T. J. Tautges, *Volume decomposition and feature recognition for hexahedral mesh generation*, Proceedings, 8th International Meshing Roundtable, (1999), 269-280