AIAA Journal
Vol. 45, No. 1, January 2007

Multiple Marching Direction Approach to Generate High-Quality Hybrid Meshes

Yasushi Ito,* Alan M. Shih,† and Bharat K. Soni‡
University of Alabama in Birmingham, Birmingham, Alabama 35294
and
Kazuhiro Nakahashi§
Tohoku University, Sendai, Miyagi 980-8579, Japan

DOI: 10.2514/1.23260

This paper describes the method to generate hybrid meshes composed of triangular prisms, pyramids, hexahedra, and tetrahedra for viscous computational fluid dynamics simulations. Surface triangulation is performed using a direct advancing front method or a modified mesh-decimation method. From a surface mesh, a near-field mesh is generated using an advancing layer approach. To generate high-quality meshes and to mesh around singular points, multiple marching directions are prepared from nodes on sharp convex corners. Special placement of elements around the sharp convex corners avoids using generalized elements. Tetrahedral meshing is then performed to fill the rest of the domain using an advancing front method. The hybrid mesh generation method is applied to several models to demonstrate its capability.

Nomenclature

a_i	=	area of triangle i
$a_{i\mathrm{opt}}$	=	area of an equilateral triangle sharing the same circumcircle with triangle i
f_s	=	stretching factor for near-field mesh layers
h_{all}	=	estimated height of the near-field mesh
h_m	=	height of the mth layer
h_{min}	=	initial layer thickness normal to no-slip walls
$\bar{\mathbf{N}}_i$	=	unit normal vector of face i
n_{es}	=	number of connecting edges that belong to sharp convex corners at a node on the surface mesh
n_p	=	number of layers
α_i	=	folding angle at edge i
β_i	=	dihedral angle of quadrangle i
η_i	=	skewness of triangle i
θ	=	angle to specify sharp corners (default value is 60 deg)

I. Introduction

THE continuous progress in computer technologies enables us to perform numerical simulations using large meshes; yet mesh generation, especially for viscous flow simulations, remains a bottleneck of computational fluid dynamics (CFD). Although many researchers have proposed hybrid mesh generation methods for viscous flow simulations [1–4], these methods still lack adequate flexibility for three-dimensional (3-D) complex geometries. Although approaches using anisotropic tetrahedra can resolve this issue [5–8], a mesh having only tetrahedra generally needs more memory than one having only prisms or a hybrid mesh during numerical simulations if these meshes have the same number of nodes. Flow solvers need to be modified beforehand to accept anisotropic elements to avoid numerical errors [9].

Generalized meshes, meshes that contain arbitrary polyhedra with no restrictions placed on the number of edges a face can have or the number of faces an element can have, have become popular in the field of viscous flow simulations, demonstrating a great potential to generate high-quality meshes around complex geometries [10–13]. However, most commercial flow solvers and postprocessors still accept only traditional elements: tetrahedra, triangular prisms, pyramids, and hexahedra. Although a generalized mesh can be converted to an all-tetrahedral mesh, degenerate tetrahedra or extra nodes may be required to eliminate hanging faces. Unfortunately, these processes create extremely low-quality tetrahedra.

The authors have proposed a reliable hybrid mesh generation method starting with the formation of an isotropic tetrahedral mesh to enable robust near-field mesh generation [14]. Prismatic layers are added on no-slip walls, while tetrahedral elements are moved inward. To enhance the quality of the prismatic layers around sharp convex corners, two normals are evaluated for the marching process in these regions. This approach has accomplished robust hybrid mesh generation; nonetheless, it is not very efficient in terms of computational memory. During the process, information of all the tetrahedral elements should be stored in the physical memory during the addition of prismatic layers. Thus, the size of the required physical memory must be greater than the size of the initial tetrahedral mesh all the time. Moreover, the quality of resulting meshes is not always good around concave and convex regions. It is difficult to enable more than two marching directions from nodes on sharp convex corners because of the topological constraint—the existence of the tetrahedral elements. The issue on the placement of semistructured elements around singular points was not resolved [8]. The purpose of this paper is to extend this near-field mesh generation approach to one starting with a surface mesh with significant improvement in mesh quality. It is combined with an isotropic tetrahedral mesh generator [15] to fill the rest of the domain.

In this paper, we propose a robust hybrid mesh generation approach using an advancing layer method to generate a near-field mesh on a surface mesh and an advancing front method to fill the rest of the domain. Our main objective is to create a high-quality hybrid mesh automatically from a surface mesh. Our method is applied to several models to demonstrate its capability.

Presented as Paper 530 at the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 9–12 January 2006; received 16 February 2006; revision received 5 September 2006; accepted for publication 17 September 2006.

*Research Assistant Professor, Department of Mechanical Engineering, 1530 3rd Avenue S, BEC 356B, Birmingham, AL 35294-4461. Member AIAA.

†Research Associate Professor, Department of Mechanical Engineering, 1530 3rd Avenue S, BEC 356C, Birmingham, AL 35294-4461. Senior Member AIAA.

‡Professor and Chair, Department of Mechanical Engineering, 1530 3rd Avenue S, BEC 257, Birmingham, AL 35294-4461. Associate Fellow AIAA.

§Professor, Department of Aerospace Engineering, 6-6-01 Aoba, Aramaki, Aoba-ku, Sendai, 980-8579. Associate Fellow AIAA.

II. Surface Mesh Generation

To generate surface meshes for computer-aided design (CAD)-based models, the direct advancing front method is employed [16,17]. A discrete model in stereolithography (STL) format is used as a background mesh, on which a new surface mesh suitable for computational simulations is created. Geometrical features are extracted based on a folding angle at each edge. A user specifies node distributions on the geometrical features, which form an initial front for the advancing front method. The surface triangulation is performed in the physical 3-D space in order to check the quality of triangles easily.

Reverse engineering approaches are also common to generate 3-D models. Commercial 3-D laser scanners can produce triangulated surfaces as output. Moreover, triangulated surfaces can be extracted from medical images using open source libraries, such as the Visualization ToolKit (VTK) [18] and the Insight Segmentation and Registration Toolkit (ITK) [19]. The quality of these triangulated surfaces, however, is usually quite poor and cannot be used as computational meshes. A modified mesh-decimation method combined with an angle-based node-smoothing method and an edge-swapping method based on the Delaunay property enables geometrically adapted, high-quality surface mesh generation semiautomatically [20]. To control the mesh density, local surface curvature and volume thickness (i.e., the distance between two surfaces) are considered. This approach can be coupled with the direct advancing front method to control local mesh density more easily.

III. Near-Field Volume Mesh Generation

Two stages are required for hybrid volume mesh generation: near-field mesh generation to place enough elements in viscous layers and tetrahedral mesh generation to fill the rest of the domain. Near-field mesh generation is based on [14], with many modifications. Starting with a surface mesh results in more flexibility when dealing with sharp convex corners. An alternating digital tree (ADT) [21] is used for the geometric search to avoid creating invalid elements. User inputs to generate near-field mesh layers are three parameters: the number of the layers n_p, an initial layer thickness normal to no-slip walls $h_{\min}$, and a stretching factor f_s. The height of the mth prismatic layer ($m = 1$ to n_p) is calculated as

$$h_m = f_s^{m-1} h_{\min} \tag{1}$$

No user intervention is required during the process. Figure 1 shows an example of a hybrid mesh around a simple wing.

A. Convex Regions

There are two challenges to mesh convex regions mainly using prisms and hexahedra. One is to improve the quality of control volumes around sharp corners. Ill-shaped control volumes may affect solution accuracy, especially when a flow solver employs a cell-vertex method. Control volumes around sharp corners are highly skewed if only a single normal is extruded from each node on no-slip walls as shown in Fig. 2. A special treatment is required to avoid numerical errors there, which can induce unphysical pressure jumps [14]. The other is to fill the domain around singular points, where semistructured elements close to no-slip walls are impossible to achieve using a single normal from each node. Figure 3 gives an example of this problem. An aircraft configuration usually has only a small number of singular points, which prevent generating valid hybrid meshes. These challenges can be resolved with an all-tetrahedral mesh relatively easily. It, however, results in longer computational time for simulations with much more elements required to accurately resolve boundary layers.

Multiple normals should be extruded from a single surface node to solve these problems. Generalized prisms, each of which has a degenerate edge (Fig. 4), can be placed at sharp corners to achieve multiple normals extruded from each node easily. The quality of generalized prisms, however, is difficult to evaluate because each of their two quadrangles cannot be planar. Moreover, most of the

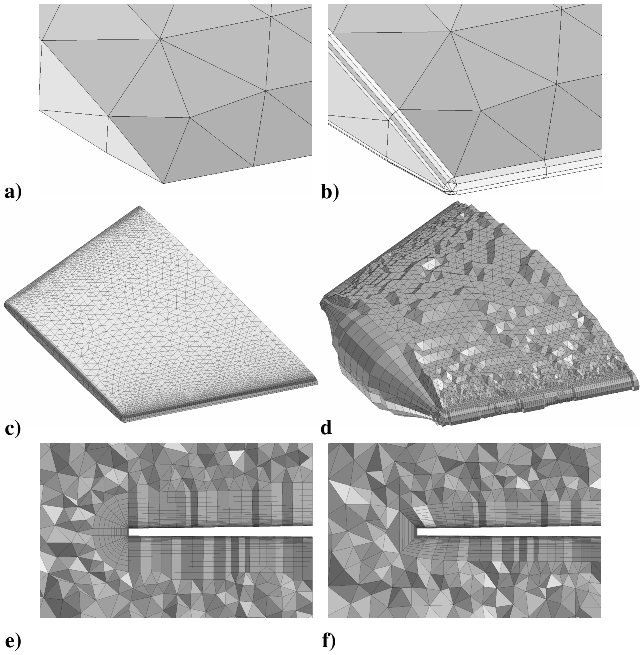

Fig. 1 Hybrid meshes around a wing: a) surface mesh around the wing tip and leading edge; b) 15th, c) 20th, d) 45th layer of near-field mesh using multiple marching directions at sharp corners; volume meshes at a cross-flow cross section e) using multiple marching directions, and f) a single direction.

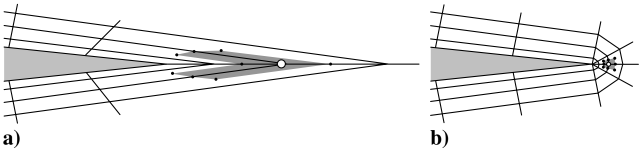

Fig. 2 Elements near wing trailing edges and control volumes around white points using a typical cell-vertex method: a) single normal from a sharp corner creating a highly skewed control volume; b) multiple normals improving the quality of the corresponding control volume.

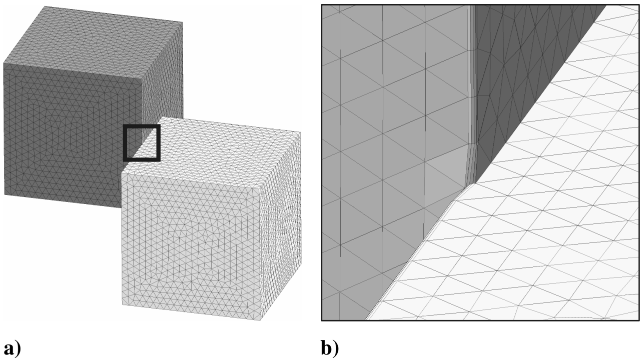

Fig. 3 Intersection of two cubes: a) singular point; b) a layer extruded from the initial surface using multiple normals at nodes on sharp corners.

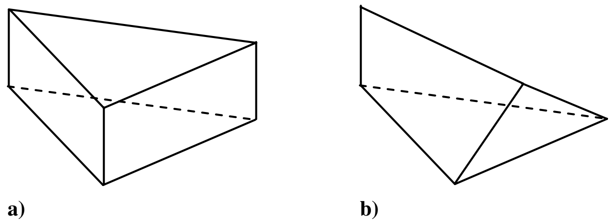

Fig. 4 a) Traditional and b) generalized prisms.

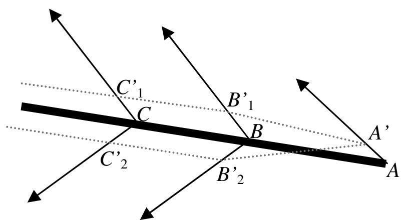

Fig. 5 Resulting gap at an end of a sharp corner: the primes denote new nodes in the next layer. Region $AA'BB'_1B'_2$ cannot be filed with a single traditional element.

existing commercial flow solvers and postprocessing software systems do not accept arbitrary polyhedra. Sharov et al. [22] modified surface meshes beforehand to generate high-quality near-field meshes. This approach, however, may require more user interventions during the surface meshing process and may result in many additional nodes around sharp corners.

Garimella and Shephard [23] proposed a generalized advancing layer method, which allows multiple normals to emanate from each surface node. First, sharp convex corners are identified on a surface mesh, on which each node is assigned multiple marching directions. Second, triangles on the surface boundaries are advanced to create a near-field mesh (*prism creation*). Third, the gap between the prismatic layers at sharp corners are filled (*blend creation*). This approach can provide better control volumes near sharp corners. If semistructured layers need to be maintained as the near-field mesh, however, the blend creation process may not be carried out using only prisms and hexahedra or may need sliver elements at ends of sharp corners, as shown in Fig. 5. Furthermore, the blend creation process may create low-quality elements if the region to be filled is occupied by other no-slip walls. The front to be advanced should cover entire no-slip walls so that the tops of layers close to each other can be detected easily.

Our approach requires the preparation of the next front based on the initial front (the surface mesh). New nodes and faces are added to the next front at sharp convex corners in order to set multiple normals there. The new nodes and faces are degenerated when they are prepared on the initial front. When the degenerate nodes are extruded in their normal directions to create the first layer, degenerate faces are unfolded and have positive area. In our approach, the front covers the entire no-slip walls all the time. Sharp corners are defined at each edge i, the two neighboring faces of which are f_{i0} and f_{i1}, as follows:

$$\alpha_i = \cos^{-1}\left(\bar{\mathbf{N}}_{fio} \cdot \bar{\mathbf{N}}_{fi1}\right) > \theta \qquad (2)$$

The process to add the degenerate faces is as follows:

1) Count the connecting edges n_{es} that belong to sharp convex corners at each node (defined as sharp edges).

2) If $n_{\text{es}} = 1$, the node is at the end of a sharp convex corner, such as the intersection of a wing trailing edge and a fuselage. One more edge that connects to the node may be flagged as a sharp edge to create better elements. For example, Fig. 6 illustrates three boundary surfaces: S_A, S_B, and S_C, which can be a wing upper surface, a blunt trailing edge, and a fuselage, respectively. Suppose line BC is considered as a sharp convex corner and node B is located at the end of the sharp corner. At node B, the average normal of boundary surfaces S_A and S_B, which connect to the sharp convex corner, can be projected to the other boundary surface, S_C, vector BA. Vector BA represents the optimal position of the additional sharp edge. n_{es} is updated if new sharp edges are added.

3) Add degenerate nodes on the sharp edges.

a) If $n_{\text{es}} = 2$, the user-specified number of degenerate nodes n_n are added at the node (default value is 4). They are degenerated because they have the same coordinates in this stage. The faces connected to it can be locally classified into two zones by the two sharp edges. For example, at node B in Fig. 5, vectors BB'_1 and BB'_2 represent the average normals of the faces on the upper side of sharp corner A-B-C and those on the lower side of sharp corner A-B-C, respectively. $(n_n - 1)$ additional normals are added between the two normals, which should equally divide the angle defined by vectors BB'_1 and BB'_2. For example, a degenerate node with normal vector BB'_2 is added at node B if $n_n = 1$. The normal vector of node B is set to vector BB'_1.

b) If $n_{\text{es}} > 2$, $n_{\text{es}}\, n_n$ nodes are added. Figure 7 shows an example at node A when $n_{\text{es}} = 3$. The initial normal at node A (vector AA'_1, the double-line arrow) is preserved, while vectors AA'_2, AA'_3, and AA'_4 are estimated based on the three local face zones. $(n_n - 1)$ additional normals are added between every two of the three vectors.

4) Create degenerate faces. They will be unfolded and have positive area when the front is advanced. A unit normal vector of each degenerate face is estimated by averaging the normals of its nodes as if the geometry is chamfered.

a) At each sharp edge i, n_n quadrangles are added if $n_{\text{es}} \geq 2$ at each of the two nodes of edge i (e.g., edge BC in Fig. 5) or n_n triangles are added if $n_{\text{es}} = 1$ at one of the two nodes (e.g., edge AB in Fig. 5).

b) At each node i where $n_{\text{es}} > 2$, connectivity of the newly added nodes in step 3b and node i is determined by 2-D Delaunay triangulation. Note that these nodes have the same coordinates in this stage (e.g., nodes A'_1, A'_2, A'_3, and A'_4 are at node A in Fig. 7) and that they will be nondegenerate nodes when the front is advanced. They are temporarily extruded in their normal directions by small amounts because degenerate nodes cannot be triangulated. They are moved back once the connectivity is determined.

When the front is advanced, the degenerate faces have positive area as shown in Fig. 1b. The next step is creating elements under the front. At nodes where $n_{\text{es}} > 2$, tetrahedra are added in the stage of the first layer generation (e.g., three tetrahedra $AA'_2A'_3A'_1$, $AA'_3A'_4A'_1$, and $AA'_4A'_2A'_1$ in Fig. 7). Figure 8 shows a strategy to fill the space at an end of a sharp corner. Every node where $n_{\text{es}} = 2$ has two normals in this case to simplify the explanation. At the stage of the first layer generation, node A is not extruded and is assigned a tetrahedron defined by $BB'_1B'_2A$ and pyramids. After that, node A is extruded as usual. From edge BC, prism $BB'_1B'_2CC'_1C'_2$ is created in the first stage and hexahedra $B'_1B''_1B''_2B'_2C'_1C''_1C''_2C'_2$ in the second stage. The multiple marching direction approach creates a small number of low-quality elements, such as tetrahedron $BB'_1B'_2A$ and prism $AB'_2B'_1A''B''_2B''_1$, if there are nodes on no-slip walls where $n_{\text{es}} = 1$.

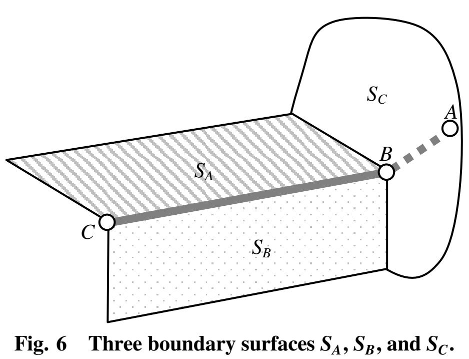

Fig. 6 Three boundary surfaces S_A, S_B, and S_C.

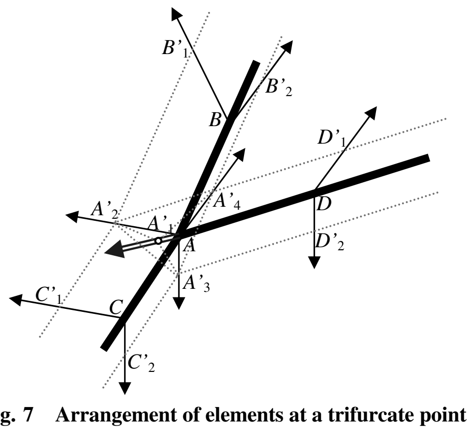

Fig. 7 Arrangement of elements at a trifurcate point.

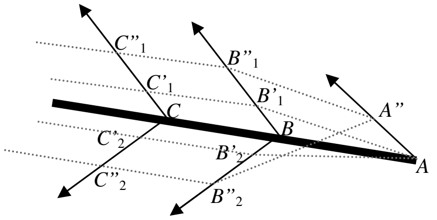

Fig. 8 Arrangement of elements at an end of a sharp corner.

However, it enables one to place high-quality traditional elements around sharp corners in general as shown in Fig. 2.

B. Concave Regions

The adoption of generalized elements enables the edges to collapse easily to thin marching directions at concave regions. Because only traditional elements are used in this paper, edge collapse can create lower quality elements. When the number of layers that are already created becomes larger than $\min(n_p/2, 5)$, Laplacian smoothing revises the positions of new nodes at concave regions. If the creation of low-quality elements remains a possibility, the near-field mesh generation is locally stopped based on the criteria that are discussed in Sec. III.D.

C. Boundary Surfaces to Be Modified

The surface mesh on no-slip walls (i.e., the initial front) and far-field boundaries are preserved during the hybrid mesh generation process. There are boundary surfaces that are not required to be advanced but that are adjacent to the no-slip walls, for example, inlets, exits, symmetry planes, and periodic boundaries [2]. They are defined as adjacent surfaces and are assumed to be flat surfaces in this paper. A node on the border between an adjacent surface and a no-slip wall (defined as a border node) can be moved on the adjacent surface, but should always be on it. The normal of the border node is revised so that the normal is always tangent to the adjacent surface.

Faces on the adjacent surfaces need to be modified so that the border nodes can be advanced. The nodes on the adjacent surfaces might be moved outward, while the connectivity of them was kept during the process. However, this approach usually creates poor quality faces on the adjacent surfaces even with the help of a node-smoothing method and they may prevent the border nodes advancing sufficiently. In our approach, faces on the adjacent surfaces are removed if they are close to the no-slip walls. A face is considered to be close to a no-slip wall if the distance between them is smaller than $3h_{\text{all}}$. Figure 9 shows an example. The resulting holes are temporarily filled by a Delaunay triangulation algorithm without using additional points to define the closed domain again and to prevent adding nodes outside the domain when the front is advanced (Fig. 9b). The Delaunay triangles are replaced with suitable triangles using the advancing front method when the near-field mesh generation is finished (Fig. 9d).

D. Criteria for Locally Stopping Layer Generation

The addition of a new layer is continued until the user-specified number of layers is obtained. Layer generation is locally stopped when one of the following conditions is satisfied at a node on the front:

1) The node is too close to another surface on the front ($<3h_m$).

2) The node will intersect faces.

3) The marching direction of the node is not visible from the faces that connect to it and cannot be fixed.

4) The square root of the average face area at a node is larger than the height of a layer to be added. If the initial surface mesh has isotropic faces, the area of a quadrangle is weighted by 0.5.

5) The quality of a face on the front is low, and it becomes worse if the front is advanced.

6) The quality of quadrangles to be created as sides of prisms and hexahedra is low.

7) All the neighboring nodes are no more advanced.

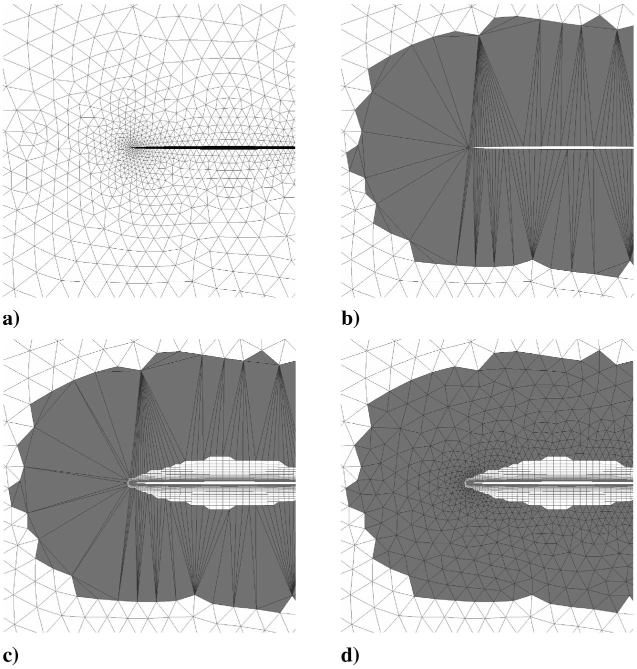

Fig. 9 Symmetry plane of the hybrid mesh around the wing shown in Fig. 1: a) initial surface mesh; b) Delaunay triangles around the wing indicated by gray; c) after the near-field mesh generation; d) after applying the advancing front method.

The quality of triangle i is evaluated by the skewness defined as follows:

$$\eta_i = \frac{a_{i\text{opt}} - a_i}{a_{i\text{opt}}} \tag{3}$$

Because the skewness is the ratio of the area of triangle i to the area of an equilateral triangle sharing the same circumcircle with triangle i, it

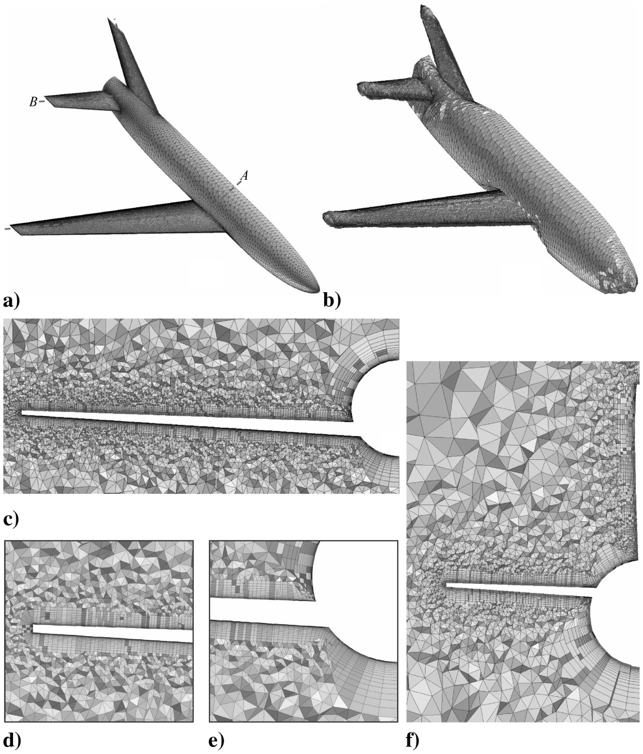

Fig. 10 Hybrid mesh around an ONERA M5 model: a) surface mesh; b) top of the near-field mesh; (c–e) cross section of the hybrid mesh at *A*; f) cross section at *B*.

indicates low-quality triangles efficiently. Triangle i is considered to be low quality if $\eta_i > 0.8$. The quality of quadrangle i, which has four nodes, A, B, C, and D, is defined by a dihedral angle between triangles ABC and CDA, and ABD and BCD:

$$\beta_i = \max\left[\cos^{-1}\left(\bar{\mathbf{N}}_{ABC} \cdot \bar{\mathbf{N}}_{CDA}\right), \cos^{-1}\left(\bar{\mathbf{N}}_{ABD} \cdot \bar{\mathbf{N}}_{BCD}\right)\right] \quad (4)$$

β_i should be smaller than 30 deg.

IV. Far-Field Volume Mesh Generation

After the near-field mesh generation on the no-slip walls, tetrahedral elements are generated to fill the rest of the domain using an advancing front method [15]. The quality of the tetrahedral elements is enhanced using angle-based node smoothing, face swapping based on the Delaunay property, and removal of nodes that have an insufficient number of tetrahedra. A user can specify a stretching factor to control the mesh density.

V. Applications

In this section, a few models are shown to demonstrate the hybrid mesh generation method. All the meshes are generated using a single processor on a 64-bit Linux machine with two 3.2 GHz Pentium 4-based Xeon processors, with 4 GB of physical memory.

A. ONERA M5

Figure 10 illustrates surface and hybrid meshes around an ONERA M5 semispan model, which has 2,359,775 nodes, 2,118,342 tetrahedra, 3,578,219 prisms, 45,088 pyramids, and 151,246 hexahedra. Although the surface mesh only has triangles as shown in Fig. 10a, hexahedra are used around the wing trailing edges. The maximum number of layers is 40, the initial layer thickness is 5.0×10^{-6} based on the overall length, and the stretching factor is 1.2. The required time for the volume mesh generation is 72 min from the surface mesh: 44 min for the near-field mesh generation, 26 min for the far-field mesh generation, and 2 min for the smoothing of tetrahedral elements. The required time is probably reasonable because an advancing front based meshing approach typically needs more CPU time than a Delaunay based approach to create a mesh. The maximum memory usage of 1 GB is observed. Another hybrid mesh with 2,111,977 nodes is created using our previous hybrid mesh generation approach as a reference [14]. Figure 11 shows the dihedral angle distribution of the two hybrid meshes. The two peaks around the dihedral angle of an equilateral tetrahedron [$\cos^{-1}(1/3) = 70.5$ deg] and 90 deg indicate that the mesh is in good quality. The approach proposed in this paper creates much better quality elements.

B. Human Sinus Model

Figure 12 shows a hybrid mesh for a human sinus model, which has 516,673 nodes, 889,861 tetrahedra, 654,608 prisms, and 10,509 pyramids. The original discrete surface is extracted from magnetic resonance imaging (MRI) data, and then surface retriangulation is performed using the modified mesh-decimation method [20]. Although the surface of the model is smooth, the geometry is complicated. Prismatic layers are added up to five layers and the stretching factor is 1.2. The required time for the volume mesh generation is 23 min: 10 min for the near-filed mesh generation, 12 min for the far-field mesh generation, and 40 s for the smoothing of tetrahedral elements.

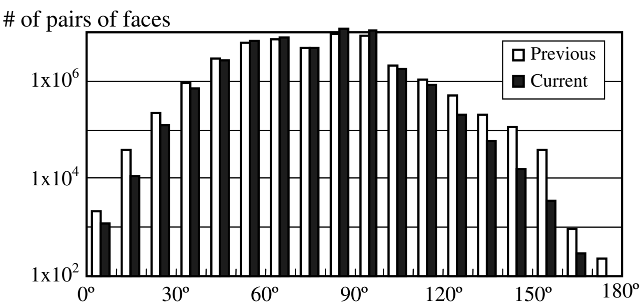

Fig. 11 Dihedral angle distribution for the ONERA M5 meshes using the previous hybrid mesh generation approach (previous) and the approach proposed in this paper (current).

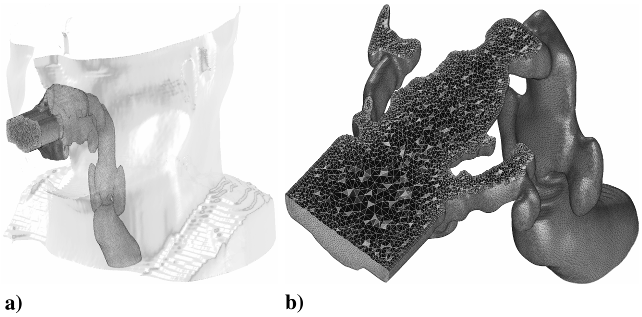

Fig. 12 Hybrid mesh for a human sinus model: a) surface mesh and its location relative to the skin; b) cross section of the hybrid mesh.

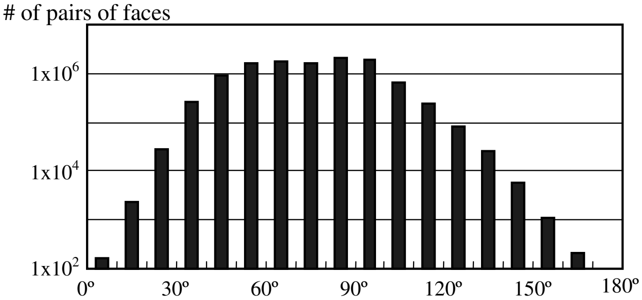

Fig. 13 Dihedral angle distribution for the human sinus mesh.

One approach to adding layers is to keep the total number of layers while varying the height of the individual layers; however, the height of each layer may not be constant. The advantage of this approach is that the required elements are only prisms and tetrahedra. The height of the top layer may become too large or too small for complex geometries. Consequently, the smaller number of nodes may be placed in the direction normal to the no-slip walls, which conflicts with the objective of hybrid meshes. Moreover, the nonconstant height of each layer may affect the computational result.

The alternative approach for adding layers is to keep the height of each layer almost constant everywhere, and the addition of prisms is locally stopped when the height of the prisms becomes sufficient. Although this approach requires pyramids as intermediates between prisms and tetrahedra, the quality of the resulting mesh is better. Because the density of the surface mesh is usually widely changed, the latter approach is more appropriate. A smooth transition is observed from the prismatic layers to the tetrahedral mesh. Figure 13 gives the dihedral angle distribution of the hybrid mesh. There are also two peaks around $\cos^{-1}(1/3)$ and 90 deg.

C. Flow Element of a Thrust Chamber Assembly

Figure 14 shows a hybrid mesh inside a flow element of a thrust chamber assembly. The mesh is $1/6$ of the overall model because it is axisymmetric. Quadrangles are used for the pipe surfaces. Multiple marching directions are defined at pipe inlets, which enable smooth transition of the mesh size there. The hybrid mesh has 364,554 nodes, 870,761 tetrahedra, 21,476 prisms, 33,655 pyramids, and 167,259 hexahedra. The total computational time for the hybrid volume mesh generation is 51 min: 38 min for the near-field mesh generation, 12 min for the far-field mesh generation, and 30 s for the smoothing of the tetrahedral elements.

VI. Conclusion

A hybrid mesh generation method for viscous flow simulations is proposed. An advancing layer method is employed for near-field

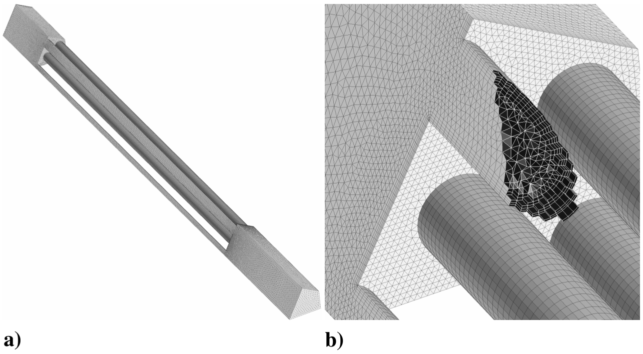

Fig. 14 Hybrid mesh in a flow element of a thrust chamber assembly: a) surface mesh; b) cross section of the hybrid mesh.

mesh generation. To improve the mesh quality near convex regions, multiple marching directions are defined on the surface mesh. The special placement of elements at ends of sharp corners avoids using generalized elements. Those modifications enable higher-quality hybrid mesh generation around sharp convex corners. Multiple normals also help to create semistructured elements around singular points. Tetrahedral meshing is performed using an advancing front method to fill the rest of the domain. The hybrid mesh generation method is applied to a few models. The results represented here demonstrate that the hybrid mesh generation method achieves a smooth transition between the near- and far-field meshes and produces high-quality meshes based on the dihedral angle distributions.

Acknowledgments

This research is supported in part by the NASA Constellation University Institutes Project (CUIP) No. NCC3-994 and the NSF ITR Adaptive Software Project No. ACI-0085969.

References

[1] Kallinderis, Y., and Ward, S., "Prismatic Grid Generation for Three-Dimensional Complex Geometries," *AIAA Journal*, Vol. 31, No. 10, 1993, pp. 1850–1856.

[2] Connell, S. D., and Braaten, M. E., "Semistructured Mesh Generation for Three-Dimensional Navier-Stokes Calculations," *AIAA Journal*, Vol. 33, No. 6, June 1995, pp. 1017–1024.

[3] Sharov, D., and Nakahashi, K., "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flow Applications," *AIAA Journal*, Vol. 36, No. 2, 1998, pp. 157–162.

[4] Khawaja, A., and Kallinderis, Y., "Hybrid Grid Generation for Turbomachinery and Aerospace Applications," *International Journal for Numerical Method in Engineering*, Vol. 49, No. 1–2, 2000, pp. 145–166.

[5] Marcum, D. L., "Generation of Unstructured Grids for Viscous Flow Applications," AIAA Paper 95-0212, 1995.

[6] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layers Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49.

[7] Hassan, O., Morgan, K., Probert, E. J., and Peraire, J., "Unstructured Tetrahedral Mesh Generation for Three-Dimensional Viscous Flows," *International Journal for Numerical Methods in Engineering*, Vol. 39, No. 4, 1996, pp. 549–567.

[8] Löhner, R., and Cebral, J., "Generation of Non-Isotropic Unstructured Grids via Directional Enrichment," *International Journal for Numerical Method in Engineering*, Vol. 49, No. 1–2, 2000, pp. 219–232.

[9] Luo, H., Sharov, D., Baum, J. D., and Löhner, R., "On the Computation of Compressible Turbulent Flows on Unstructured Grids," AIAA Paper 2000-0926, 2000.

[10] Michal, T., and Cary, A., "Unstructured Grid Extrusion for Viscous Flow Simulations," AIAA Paper 2001-0444, 2001.

[11] Shaw, J. A., Stokes, S., and Lucking, M. A., "The Rapid and Robust Generation of Efficient Hybrid Grids for RANS Simulations over Complete Aircraft," *International Journal for Numerical Methods in Fluids*, Vol. 43, No. 6–7, 2003, pp. 785–821.

[12] Chalasani, S., and Thompson, D., "Quality Improvements in Extruded Meshes Using Topologically Adaptive Generalized Elements," *International Journal for Numerical Methods in Engineering*, Vol. 60, No. 6, 2004, pp. 1139–1159.

[13] Athanasiadis, A. N., and Deconinck, H., "A Folding/Unfolding Algorithm for the Construction of Semi-Structured Layers in Hybrid Grid Generation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, No. 48–49, 2005, pp. 5051–5067.

[14] Ito, Y., and Nakahashi, K., "Improvements in the Reliability and Quality of Unstructured Hybrid Mesh Generation," *International Journal for Numerical Methods in Fluids*, Vol. 45, No. 1, 2004, pp. 79–108.

[15] Ito, Y., Shih, A. M., and Soni, B. K., "Reliable Isotropic Tetrahedral Mesh Generation Based on an Advancing Front Method," *Proceedings of the 13th International Meshing Roundtable*, Sandia National Laboratories, Albuquerque, NM, 2004, pp. 95–105.

[16] Ito, Y., and Nakahashi, K., "Direct Surface Triangulation Using Stereolithography Data," *AIAA Journal*, Vol. 40, No. 3, 2002, pp. 490–496.

[17] Ito, Y., and Nakahashi, K., "Surface Triangulation for Polygonal Models Based on CAD Data," *International Journal for Numerical Methods in Fluids*, Vol. 39, No. 1, 2002, pp. 75–96.

[18] Visualization ToolKit (VTK), http://www.vtk.org/ [cited 5 January 2006].

[19] Insight Segmentation and Registration Toolkit (ITK), http://www.itk.org/ [cited 5 January 2006].

[20] Ito, Y., Shum, P. C., Shih, A. M., Soni, B. K., and Nakahashi, K., "Robust Generation of High-Quality Unstructured Meshes on Realistic Biomedical Geometry," *International Journal for Numerical Methods in Engineering*, Vol. 65, No. 6, 2006, pp. 943–973.

[21] Bonet, J., and Peraire, J., "An Alternating Digital Tree (ADT) Algorithm for 3D Geometric Searching and Intersection Problems," *International Journal for Numerical Methods in Engineering*, Vol. 31, No. 1, 1991, pp. 1–17.

[22] Sharov, D., Luo, H., Baum, J. D., and Löhner, R., "Unstructured Navier-Stokes Grid Generation at Corners and Ridges," *International Journal for Numerical Methods in Fluids*, Vol. 43, No. 6–7, 2003, pp. 717–728.

[23] Garimella, R. V., and Shephard, M. S., "Boundary Layer Mesh Generation for Viscous Flow Simulations," *International Journal for Numerical Method in Engineering*, Vol. 49, No. 1–2, 2000, pp. 193–218.

D. Gaitonde
Associate Editor