Generation Of Quad- And Hex-Dominant, Semistructured Meshes Using An Advancing Layer Scheme

Article · November 1999

Source: CiteSeer

2 authors, including:

Bharat Soni
Tennessee Technological University
101 PUBLICATIONS **1,065** CITATIONS

GENERATION OF QUAD- AND HEX-DOMINANT, SEMISTRUCTURED MESHES USING AN ADVANCING LAYER SCHEME

David S. Thompson[1], Bharat K. Soni[2]

Mississippi State University, Mississippi State, MS., U.S.A
[1]dst@erc.msstate.edu, [2]bsoni@erc.msstate.edu

ABSTRACT

A novel algorithm to generate quad- and hex-dominant, semistructured meshes is presented. The algorithm utilizes a three-step, advancing layer scheme to generate the mesh in structured layers starting from an initial data surface. First, a locally orthogonal reference mesh is algebraically generated. The reference mesh is then smoothed using a Poisson-type mesh generation equation. Finally, a line deletion/insertion algorithm based on cell geometry is applied as the initial data surface for the next layer is defined. Typically, for meshes appropriate for viscous flow calculations, more than 95% of the cells are quadrilaterals or hexahedra. The resulting mesh offers a potentially attractive alternative for near-body regions of hybrid meshes. Several preliminary examples are included to demonstrate the efficacy of the approach.

Keywords: mesh generation, semistructured meshes, quad and hex dominant, advancing layer

1. INTRODUCTION

An emerging technique for generating meshes for complex geometries results in the so-called hybrid mesh [1]-[4]. Hybrid meshes combine elements of structured and unstructured mesh generation. Typically, prismatic or hexahedral cells are used in regions of the domain dominated by viscous effects. Away from viscous-dominated regions, tetrahedral or Cartesian meshes are used. Hybrid topologies are unstructured in the sense that they require an explicit connectivity table. Hybrid meshes offer more flexibility than structured meshes for geometrically complex problems and offer improved feature resolution when compared to unstructured meshes through the use of anisotropic elements. Additionally, hybrid topologies require significantly fewer elements than unstructured meshes to achieve the same resolution [1].

In this paper we present a novel algorithm to generate near-body, hybrid meshes using an advancing layer scheme. The meshes generated using this method can be categorized as hybrid meshes since they combine elements of both structured and unstructured topologies. They can perhaps best be described as semistructured. The mesh is generated in layers advancing from the initial data surface. In each layer, a variant of the "parabolic" mesh generation strategies developed for structured meshes in the late 1980's [5]-[7] is used to generate the mesh. A line deletion/insertion algorithm is applied at the interface between the current layer and the next layer as the initial data surface for the next layer is defined. Using the strategies employed here, a line existing in the layer below the interface may terminate at the interface. Additionally, a line not existing in the previous layer may emanate from the interface in the case of insertion. This approach produces mesh points that traditionally have been called hanging nodes. In two dimensions, the deletion or insertion corresponds to a local modification of the mesh. In three dimensions, the modification is no longer local in that the line is deleted/inserted along its entire length. Finite-volume based hybrid flow solvers such as HYBFL3D [8] and COBALT [9] both allow for grid topologies of this type.

The approach used here to generate the mesh in each layer is similar to the hybrid grid generation scheme of Huang [4] in which several layers of a reference mesh are generated and smoothed using an elliptic Poisson equation. It also shares similarities with the marching scheme of Chan and Steger [10] which uses a hyperbolic system with added dissipation to generate fully structured meshes. Additionally, it is similar in philosophy to the marching method described by

Kallinderis [3] for algebraically generating prismatic meshes. In reference [3], the mesh point locations are smoothed indirectly via a smoothing of the surface normals. However, none of these methods utilize an insertion/deletion algorithm. In this respect, the semistructured grid generation algorithm is similar to an unstructured advancing-front algorithm with a "clean-up" step to improve mesh quality [11].

In this paper, the basics of the approach are discussed. First the algorithm is discussed with an emphasis placed on the grid generation procedure. Then the insertion/deletion strategy is discussed in detail. Finally, sample meshes are used to demonstrate the efficacy of the approach.

2. ALGORITHM OVERVIEW

In the mesh generation algorithm described here, the mesh is generated in layers starting at the initial data surface – typically the body. Each layer consists of two surfaces – the initial data surface for that layer and the generated surface. The mesh within each layer is structured. Generation of the grid within each layer is accomplished using techniques based on the "parabolic" mesh generation strategies developed for structured meshes [5]-[7]. Then a line insertion/deletion strategy is applied at the layer interface. The mesh generation algorithm described here is basically a three-step process:

1. A structured, locally orthogonal reference mesh is algebraically generated. This reference mesh consists of the initial data surface plus two generated layers.
2. The middle surface of the reference mesh is then smoothed using a Poisson-type equation typically used in structured grid generation. Note that the outer surface is updated after each iteration of the Poisson mesh equations and discarded once the desired level of smoothing is achieved.
3. Using criteria defined by the algorithm, a line deletion/insertion stencil is created. The use of line stencils maintains the structured nature of the mesh within layers. The stencils are then used to define the initial data surface for the next layer and the process is repeated.

It should also be noted that the resulting smoothed mesh still exhibits many characteristics of the reference mesh. In this respect, it is fair to categorize the resulting mesh as "nearly orthogonal." The greatest deviation from orthogonality occurs where the smoother has done the most work.

In practice, it has been found that reducing the mesh point movement induced by the smoothing by two orders of magnitude results in a mesh that is sufficiently smooth. Typically, this requires the equivalent of fewer than ten iterations of an elliptic solver. Domains containing strongly non-convex regions require additional smoothing iterations.

Although developed for growing a volume grid starting from a structured surface mesh, the work of Knupp [12] suggests that it may be possible to extend the approach presented here to growing prismatic volume grids from unstructured surface meshes. The insertion/deletion strategies would be local in nature for this type of algorithm.

3. PARABOLIC GRID GENERATION

The mesh within each layer is generated using the parabolic mesh generation method for structured meshes [5]-[7]. As is customary, a transformation of the form

$$\begin{aligned} \xi &= \xi(x,y,z) \\ \eta &= \eta(x,y,z) \\ \zeta &= \zeta(x,y,z) \end{aligned} \qquad \text{(Eq. 1)}$$

is defined ***for each layer*** and the inverse transformation

$$\begin{aligned} x &= x(\xi,\eta,\zeta) \\ y &= y(\xi,\eta,\zeta) \\ z &= z(\xi,\eta,\zeta) \end{aligned} \qquad \text{(Eq. 2)}$$

is assumed to exist. The initial data surface is assumed to be defined by a ζ=*constant* surface and ζ is taken to be the marching direction. In two dimensions, the initial data curve is defined by an η=*constant* surface and η is taken to be the marching direction. The variation in ζ is taken to be zero.

3.1 Reference Mesh Definition

The reference mesh is generated algebraically to be locally orthogonal. The first surface of the reference mesh in layer *k* is defined using

$$\boldsymbol{r}^{k}_{i,j,1} = \boldsymbol{r}^{k}_{i,j,0} + \delta^{k}_{i,j}\boldsymbol{n}^{k}_{i,j,0} \qquad \text{(Eq. 3)}$$

where $\boldsymbol{r}_{i,j,0}$ is the position vector to the point located at *(i,j)* on the initial data surface (0 subscript), $\boldsymbol{r}_{i,j,1}$ is the position vector to the point *(i,j)* on the first surface of the reference mesh, $\delta_{i,j}$ is the specified distance distribution, and $\boldsymbol{n}_{i,j,0}$ is the unit surface normal. The second surface of the reference mesh in layer *k* is generated using

$$\boldsymbol{r}^{k}_{i,j,2} = \boldsymbol{r}^{k}_{i,j,1} + \delta^{k+1}_{i,j}\boldsymbol{n}^{k}_{i,j,1} \qquad \text{(Eq. 4)}$$

Although not discussed here, it is possible to adjust the reference mesh as the mesh marches away from the initial data surface to meet specified outer boundary points [6],[7].

3.2 Mesh Smoothing

The equations used to smooth the reference mesh are the standard Poisson equations [13] used in mesh generation with the assumption of mesh orthogonality, i.e., g_{12}, g_{13}, g_{23}, are zero:

$$\begin{aligned} &\frac{g_{22}g_{33}}{g^2}(\boldsymbol{r}_{\xi\xi} + \phi\boldsymbol{r}_{\xi}) + \frac{g_{11}g_{33}}{g^2}(\boldsymbol{r}_{\eta\eta} + \psi\boldsymbol{r}_{\eta}) \\ &+ \frac{g_{11}g_{22}}{g^2}(\boldsymbol{r}_{\zeta\zeta} + \theta\boldsymbol{r}_{\zeta}) = 0 \end{aligned} \qquad \text{(Eq. 5)}$$

where

$$g_{11} = x_\xi^2 + y_\xi^2 + z_\xi^2$$
$$g_{22} = x_\eta^2 + y_\eta^2 + z_\eta^2$$
$$g_{33} = x_\zeta^2 + y_\zeta^2 + z_\zeta^2 \qquad \text{(Eq. 6)}$$

$$g = \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix}$$

The control functions $\phi, \psi,$ and θ play a critical role in determining the quality of the mesh. In particular, they are important for maintaining an acceptable spacing when line deletion or insertion occurs. The form of the control functions used here includes curvature effects [14] and is given by

$$\phi = -\frac{\partial}{\partial\xi}\ln\left(\sqrt{\frac{g_{11}}{g_{22}g_{33}}}\right)$$
$$\psi = -\frac{\partial}{\partial\eta}\ln\left(\sqrt{\frac{g_{22}}{g_{11}g_{33}}}\right) \qquad \text{(Eq. 7)}$$
$$\theta = -\frac{\partial}{\partial\zeta}\ln\left(\sqrt{\frac{g_{33}}{g_{11}g_{22}}}\right)$$

The control functions given in Eq. 7 are computed for each layer using values from the initial data surface to define ϕ and ψ and values from all three surfaces to define θ.

The partial derivatives in Eq. 5 are approximated using second-order central differences at the initial data surface. The resulting system of equations is solved using alternating direction line relaxation. Based on the topology of the mesh, the matrices may be regular or periodic.

On occasion, it has been found that adding dissipation to the Poisson smoothing equation (Eq. 5) is necessary, particularly when generating meshes in strongly nonconvex regions. The approach employed here is similar in spirit to the approach taken by Chan and Steger [10] and is shown in detail for the ξ direction. An analogous approach is used for the η direction.

In regions where $g_{11} >> g_{33}$, the dominant term in Eq. 5 is the term containing the ζ derivatives. In nonconvex regions, the resulting lack of smoothing along ζ=constant lines can lead to mesh line crossing. The approach taken here is to modify the first term of Eq. 5 in nonconvex regions. This is accomplished as follows:

$$\frac{g_{22}g_{33}}{g^2}((1+\nu_\xi)\boldsymbol{r}_{\xi\xi} + \phi\boldsymbol{r}_\xi) \qquad \text{(Eq. 8)}$$

where

$$\nu_\xi = \sqrt{\frac{\max(g_{11}, g_{33})}{g_{33}}} \times f(\theta_\xi)$$

$$f(\theta_\xi) = \begin{cases} 1 & 0 \le \theta_\xi < \dfrac{\pi}{2} \\ \sin\theta_\xi & \dfrac{\pi}{2} \le \theta_\xi < \pi \\ 0 & \pi \le \theta_\xi \end{cases} \qquad \text{(Eq. 9)}$$

and θ_ξ is the angle between $\boldsymbol{r}_{i+1,j}$-$\boldsymbol{r}_{i,j}$ and $\boldsymbol{r}_{i-1,j}$-$\boldsymbol{r}_{i,j}$.

4. LINE DELETION/INSERTION ALGORITHM

The line deletion/insertion algorithm employed here is based on cell geometry. The procedure used for ξ line deletion will be presented in detail with the extension to η lines by analogy. It is assumed there are $j=1,j_{max}$ points located on each ξ line.

The line deletion/insertion scheme described below produces cell edges having one, two, or three segments. The cell faces can have four to seven edges and the resulting volumes may have six to 17 faces. However, for viscous-type meshes in particular, a majority of the cells (typically greater than 95%) are hexahedra. It should be noted that other deletion/insertion schemes produce different cell topologies. It should be emphasized that, in three dimensions, the insertion/deletion process is global in nature in that a line is inserted/deleted along its entire length. In two dimensions, it is a localized process only.

There is great flexibility in the design of the deletion/insertion algorithm. It is also possible to define insertion and deletion algorithms based on flow field properties as well. Coupled with the weight functions described in [15], it would be possible to develop a solution adaptive mesh using both redistribution and refinement.

4.1 Example – Node Deletion at Layer Interface

The node deletion process is illustrated for a two-dimensional mesh in Figure 1. The points 1, 2, and 3 form the initial data surface for layer k. Points 4, 5, and 6 are generated using the parabolic marching algorithm. Note that points 4, 5, and 6 are not collinear. Based on the deletion/insertion algorithm, it is determined that point 5 should not be used to generate the mesh in the next layer k+1. Therefore, the initial data surface for layer k+1 contains the points 4 and 6 but not point 5. The marching algorithm is then used to generate the surface containing points 7 and 8. Note, however, that the cell in layer k+1 is actually defined using points 4, 5, 6, 8, and 7 resulting in a five-sided cell even though point 5 was not used to compute points 7 and 8. The extension to three dimensions occurs through the deletion of the line passing through point 5 rather than point 5 alone.

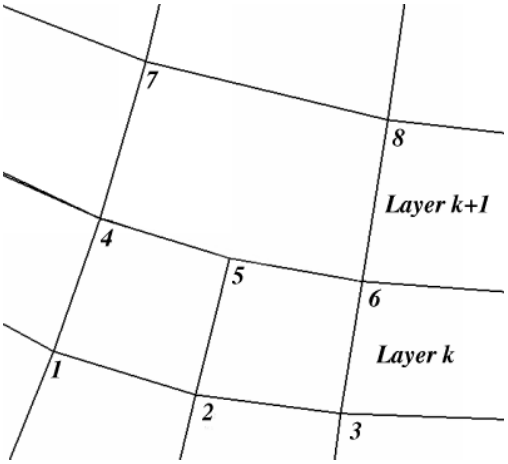

Figure 1. Semistructured mesh topology

4.2 Line Deletion/Insertion Strategy

An ξ line is tagged for deletion if it meets either of two criteria. The first criterion is based on the average "slenderness" of the cells on an ξ line. An ξ line is marked for deletion if the average value of the ratio of the ξ arc length to the ζ arc length is greater than a specified tolerance or

$$\frac{1}{j_{\max}} \sum_{j=1}^{j_{\max}} \left(\sqrt{\frac{g_{33}}{g_{11}}} \right)_j > \alpha_{avg} \qquad \text{(Eq. 10)}$$

where j_{max} is the number of points on each ξ line. The second criterion is based on a maximum "slenderness" of the cells on each ξ line. An ξ line is marked for deletion if the value of the ratio of the ξ arc length to the ζ arc length is greater than a specified tolerance at any point on the line or

$$\max_j \left(\sqrt{\frac{g_{33}}{g_{11}}} \right)_j > \alpha_{\max} \qquad \text{(Eq. 11)}$$

Note that α_{avg} and α_{max} are user-specified constants with $\alpha_{max} > \alpha_{avg}$.

It should be noted that the criteria described in Eq. 10 and Eq. 11 are defined on a line-by-line basis without regard to the status of adjacent lines. An acceptable insertion/deletion stencil can be defined only by including information about adjacent lines. The first step is tagging contiguous groups of ξ lines that are selected for deletion and identifying whether the groups contain an even number or an odd number of elements. Then, based on the configuration, one of two strategies described below can be employed to define the appropriate stencil.

4.2.1 Odd Number of Contiguous Nodes Tagged for Deletion

In the case of an odd number of contiguous ξ lines selected for deletion, the first line of the group and every other succeeding line of the group are deleted. The case of a single line selected for deletion is trivial.

4.2.2 – Even Number of Contiguous Nodes Tagged for Deletion

The case of an even number of contiguous ξ lines selected for deletion is treated using a delete two/insert one strategy. For each consecutive pair of lines tagged for deletion, both are deleted and replaced by the line defined by connecting the midpoints between the two lines tagged for deletion.

4.3 Line Insertion Strategy

The criteria used to determine if an ξ line should be inserted is based on the divergence of the ξ=constant faces of the cell. Unlike the deletion criteria, the insertion criteria apply to a cell face rather than a line. Here, the divergence of the cell is defined as

$$\Delta = \frac{1}{\sqrt{g_{33}}} \frac{\partial}{\partial \zeta} \left(\sqrt{g_{11}} \right) \qquad \text{(Eq. 12)}$$

As above, average and maximum values of the divergence are used in the algorithm. A surface of ξ=constant and ζ=constant cell faces is tagged for ξ line insertion if the average divergence exceeds a specified tolerance

$$\frac{1}{j_{\max}} \sum_{j=1}^{j_{\max}} (\Delta)_j > \beta_{avg} \qquad \text{(Eq. 13)}$$

Similarly, a surface of ξ=constant and ζ=constant cell faces is tagged for ξ line insertion if the maximum divergence exceeds a specified tolerance

$$\max_j (\Delta)_j > \beta_{\max} \qquad \text{(Eq. 14)}$$

where $\beta_{max} > \beta_{avg}$. The line insertion occurs at the midpoints of the cell faces.

4.4 – Two-Dimensional Meshes

In two-dimensions, the equivalent procedure is a point insertion/deletion scheme. The maximum and average values on the line reduce to the value at the point in question.

5. EXAMPLE MESHES

To provide a demonstration of the efficacy of this approach, several preliminary two- and three-dimensional meshes are included. These examples were chosen to illustrate the characteristics of meshes generated using this approach.

5.1 Interior of Unit Square (2D)

The first case presented is the mesh generated by marching into the interior of the unit square with the initial data curve being an equally-spaced distribution of points on the square. The distribution function δ was taken to be uniform. Figures 2-4 show the mesh generated for this case using different forms of the control functions. In each of the figures, point deletion occurs near each corner in regions of coordinate line convergence. The mesh in Figure 2 was generated using zero control functions. This is equivalent to using Winslow smoothing [16]. The coordinate lines exhibit discontinuities in the first derivative where node deletion occurs. This is a direct result of the control functions being zero. With zero control functions, the smoothing process attempts to enforce equal spacing along the surface in each layer.

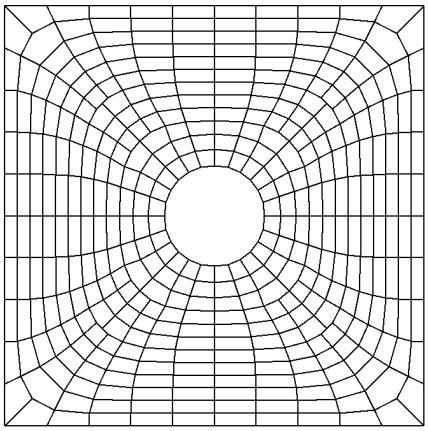

Figure 2. Interior of unit square – zero control functions

Figure 3 shows the mesh that was generated using the control functions without curvature effects, i.e., Eq. 7 without the metric terms in the denominator. Including nonzero control functions improves the smoothness of the grid by maintaining the point spacing within each layer.

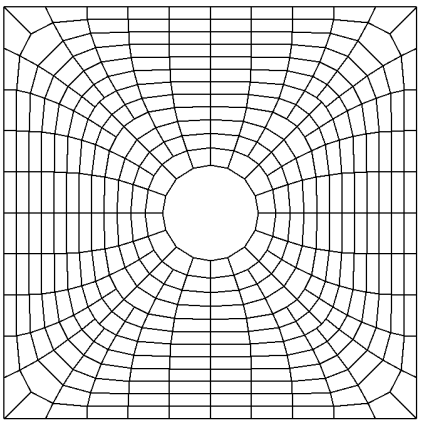

Figure 3. Interior of unit square – control functions without curvature effects

Finally, Figure 4 shows the mesh generated using the full control functions as defined in Eq. 7. The primary effect is to force the distribution in the marching direction to be more nearly uniform. In Figures 2 and 3, an apparent increase in the marching step size occurs as the grid grows toward the center of the domain. This is particularly evident in Figure 3. Additionally, the corners of the square are significantly rounded. Including curvature effects in the control functions significantly reduces both these effects.

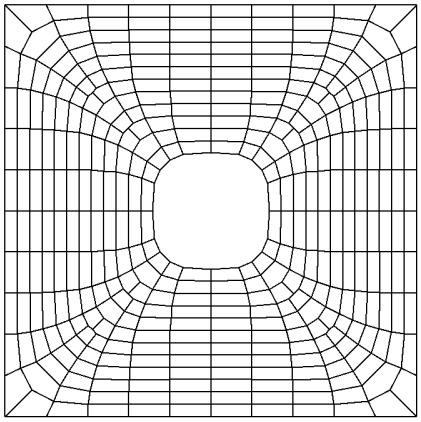

Figure 4. Interior of unit square – full control functions

5.2 Non-Convex Region (2D)

Figure 5 shows the mesh generated about a unit circle with the first quadrant "removed." The distance distribution was defined using a hyperbolic tangent stretching function to provide a mesh with a viscous-type spacing near the body. The mesh shows smooth transitions even near regions where point deletion has occurred. The mesh transitions over 40 layers from 65 points on the body surface to 28 points at the outer boundary.

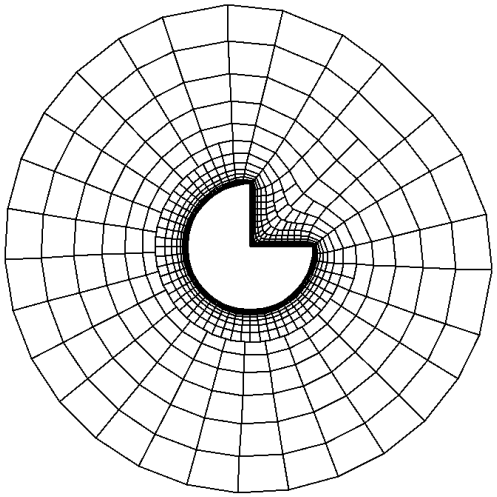

Figure 5. Mesh for non-convex body

5.3 Iced Airfoil (2D)

Figure 6 shows the grid generated about an NLF0414 airfoil with ice accretion (iced airfoil geometry courtesy of NASA Glenn Research Center). Figure 7 shows the mesh in the region near the nose. The distance distribution was defined for 60 layers using a hyperbolic tangent stretching function. Again smooth mesh transitions are evident in regions where point deletion has occurred. Since point deletion occurs naturally as the active layer advances away from the initial surface due to a decrease in cell aspect ratio, the number of points on the initial data line (321) is reduced by more than a factor of ten (28) at the outer boundary.

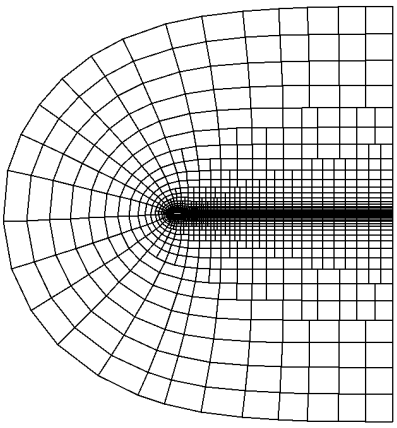

Figure 6. Full mesh for airfoil with ice accretion

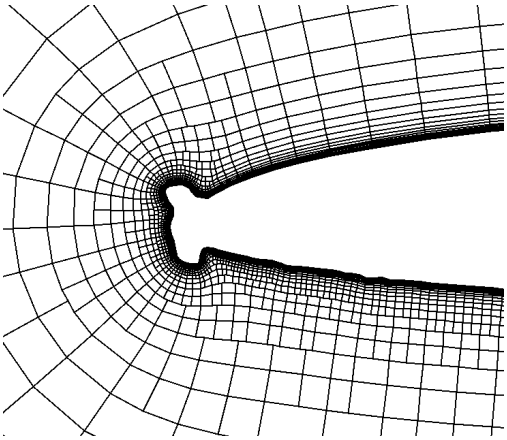

Figure 7. Detail of nose region for iced airfoil

5.4 Body of Revolution (3D)

Figures 8-10 show the grid generated around a body of revolution with a sharp nose, a sharp convex corner, and a significant nonconvex region. The grid dimensions on the body were 121x41 and 71x41 in the 40^{th} layer. Figure 8 shows a view of the grid in an η=*constant* plane. Figure 9 shows the grid in the nonconvex region near the nozzle. There is a reasonable variation in cell size in this region. Again, the cells are mostly orthogonal except in the converging region. The line insertion algorithm created lines at the nose, at the aft sharp convex corner, and at the aft end of the nozzle. The effects of line insertion are best demonstrated in the detail of the region near the sharp corner at the aft end of the body shown in Figure 10. The grid near the corner appears to be of good quality. This example illustrates how line insertion and deletion helps alleviate two of the major problems with marching-type grids - grid line convergence and grid line divergence. This configuration is included for illustrative purposes only.

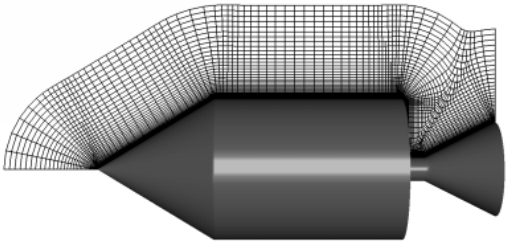

Figure 8. Mesh for body of revolution

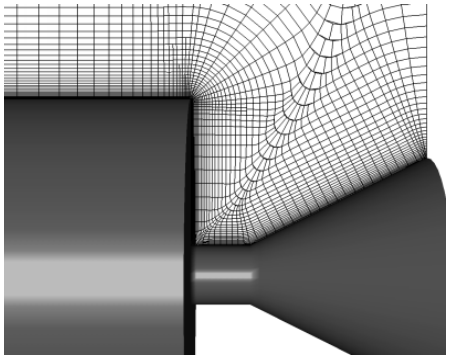

Figure 9. Detail of region near nozzle

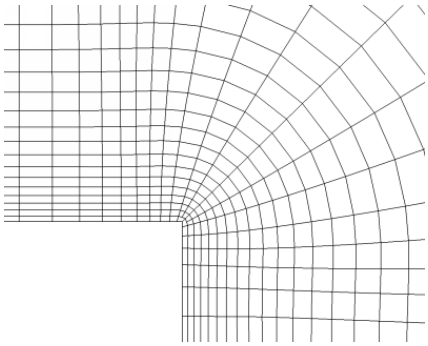

Figure 10. Line insertion at convex corner

5.5 ONERA M6 Wing (3D)

The final example of the capabilities of this grid generation algorithm is a grid around an ONERA M6 wing. It should be noted that the grid has a C-type topology. This was accomplished by including the wake cut in the definition of the initial surface. Figure 11 shows the wing and the grid in the symmetry plane. At present, no smoothing is performed on the boundaries. Figure 12 shows the wing and the grid distribution in a spanwise plane. In both figures, the grid distribution is smooth. There is a slight asymmetry in the grid at the wing tip in Figure 12. Through line deletion/insertion, this asymmetry is propagated back to the symmetry plane. The grid dimensions on the initial data surface were 97x17. In the 50^{th} and final layer, the grid dimensions were 33x10.

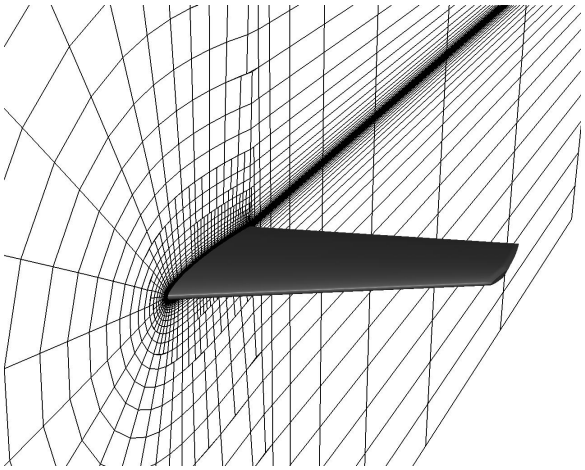

Figure 11. Wing grid in symmetry plane

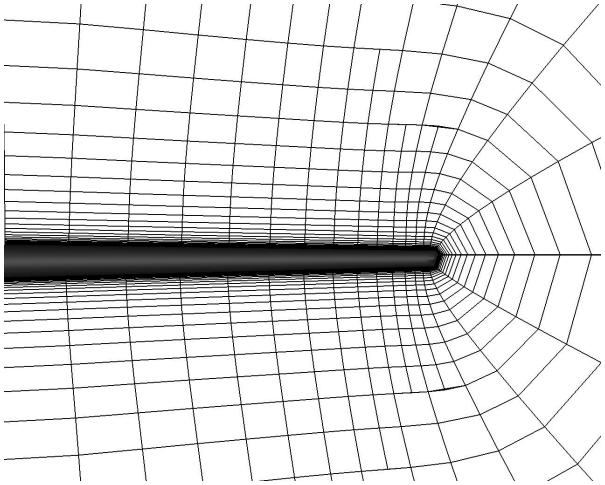

Figure 12. Wing grid in spanwise plane

6. CONCLUSIONS

The capability to generate two-dimensional, quad-dominant and three-dimensional, hex-dominant meshes using a novel semistructured mesh generation algorithm has been demonstrated. These preliminary results indicate the methods may have potential for the generation of near-body regions of hybrid meshes. Further, the flexibility allowed by the insertion/deletion strategy suggests the possibility of including insertion and deletion of points or lines based on the solution resulting in a solution adaptive mesh using both redistribution and refinement.

REFERENCES

1. J. A. Shaw, "Hybrid Meshes," *Handbook of Mesh Generation*, Eds. J. F Thompson, B. K. Soni, and N. P. Weatherill, CRC Press, Boca Raton, FL, 1998.
2. J. A. Chappell, J. A. Shaw, M. Leatham, "The Generation of Hybrid Meshes Incorporating Prismatic Regions for Viscous Flow Calculations," *Numerical Mesh Generation in Computational Field Simulation*, Eds. M. Cross, B. K. Soni, J. F. Thompson, , J. Hauser, and P. R. Eisman, NSF Engineering Research Center for Computational Field Simulation, Mississippi State, MS, 1996.
3. Y. Kallinderis, "Hybrid Meshes and Their Applications," *Handbook of Mesh Generation*, Eds. J. F Thompson, B. K. Soni, and N. P. Weatherill, CRC Press, Boca Raton, FL, 1998.
4. C.-T. Huang, "Hybrid Mesh Generation System," M.S. Thesis, Department of Aerospace Engineering, Mississippi State University, 1996.
5. S. Nakamura, ``Noniterative Mesh Generation Using Parabolic Partial Differential Equations," *Numerical Mesh Generation*, Ed. J. F. Thompson, Elsevier Science Publishing Company, Inc., New York, 1982.
6. R. W. Noack and D. A. Anderson, "Solution-Adaptive Mesh Generation using Parabolic Partial Differential Equations," *AIAA J.*, Vol. 28, pp.1016-1023, 1990.
7. R. W. Noack, and I. H. Parpia, ``Solution Adaptive Parabolic Mesh Generation in Two and Three Dimensions", *Numerical Mesh Generation in Computational Fluid Dynamics and Related Fields*, Eds. A. S.-Arcilla, J. Hauser, P. R. Eiseman, and J. F. Thompson, Elsevier Science Publishing Company, New York, 1991.
8. R. P. Koomullil and B. K. Soni, "Generalized Mesh Techniques in Computational Field Simulation," *Numerical Mesh Generation in Computational Field Simulation*, Eds. M. Cross, P. R. Eisman, J. Hauser, B. K. Soni, and J. F. Thompson, International Society for Mesh Generation, Mississippi State, MS, 1998.
9. R. F. Tomaro, W. Z. Strang, and L. N. Sankar, "An Implicit Algorithm for Solving Time Dependent Flows on Unstructured Meshes," AIAA Paper 97-0333, Presented at 35^{th} Aerospace Sciences Meeting, Reno, NV, January 1997.
10. W. M. Chan and J. L. Steger, "Enhancements of a Three-Dimensional Hyperbolic Grid Generation

Scheme," *Appl. Math. Comput.*, Vol. 51, pp. 181-205, 1992.
11. S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal, "Advancing Front Quadrilateral Meshing Using Triangle Transformation," *Proc. 7th Int. Meshing Roundtable '98*, Sandia National Laboratories, pp. 409-428, 1998.
12. P. M. Knupp, "Winslow Smoothing on Two-Dimensional Unstructured Meshes," *Proc. 7th Int. Meshing Roundtable '98*, Sandia National Laboratories, pp. 449-457, 1998.
13. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation*, Elsevier Science Publishing Company, New York, 1985.
14. B. K. Soni, "Elliptic Mesh Generation Systems: Control Functions Revisited," *App. Math. Comp.*, Vol. 59, pp. 151-164, 1993.
15. H. J. Thornburg and B. K. Soni, "Weight Functions in Grid Adaption," *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Eds. N. P. Weatherill, P. R. Eisman, J. Hauser, and J. F. Thompson, Pineridge Press Ltd., Swansea, U.K., 1994.
16. A. Winslow, "Numerical Solution of the Quasilinear Poisson Equations in a Nonuniform Triangle Mesh, *J. Comp. Phys.*, Vol. 2, pp. 149-172, 1967.