AIAA JOURNAL
Vol. 31, No. 10, October 1993

Prismatic Grid Generation for Three-Dimensional Complex Geometries

Y. Kallinderis* and S. Ward†
University of Texas at Austin, Austin, Texas 78712

The paper describes generation of semiunstructured prismatic grids for three-dimensional complex geometries. A new method of generation is presented, which couples both algebraic and elliptic steps. This combination yields low computation cost, as well as smoothness of the grid. Main advantages of the developed grid generator are its direct control of grid orthogonality and spacing, as well as its generality for treatment of three-dimensional geometries. The method marches a surface consisting of triangular faces away from the body surface. Employment of prismatic elements is a relatively new approach toward three-dimensional, complex geometry numerical simulations of viscous flows. A body surface is covered with triangles, which gives the geometric flexibility, whereas the structure of the mesh in the direction normal to the surface provides better resolution of the viscous regions. An F-16A aircraft geometry was included in the applications to investigate efficiency and to demonstrate robustness of the method in handling relatively complex topologies. Grid generation time for the entire aircraft domain was of the order of minutes on a workstation.

I. Introduction

SIMULATION of flows around three-dimensional bodies is a major issue in computational fluid mechanics. Complexity of the geometry and flowfields makes three-dimensional computations a pacing item. Generation of a body-conforming grid proves to be a difficult task.[1,2] In cases of structured meshes, a large number of separate blocks need to be defined, and hexahedral grids are generated within each block. In addition, a special algorithm is required to match the grids of neighboring blocks. A radical alternative to a structured mesh is the use of tetrahedra. Tetrahedral grids provide flexibility in three-dimensional grid generation since they can cover complicated topologies more easily compared to the hexahedra meshes.[1,3-6] However, generation of tetrahedral cells for boundary layers is quite difficult. In those regions the main solution gradients occur in the direction normal to the surface, which requires cells of very large aspect ratio. It appears that structured grids are superior in capturing the directionality of the flowfield over viscous regions.

A compromise between the two different types of grids are the prismatic grids. Such grids consist of triangular faces that cover the body surfaces, while quadrilateral faces extend in the direction normal to the surface. The cells can have very high aspect ratio, while providing geometric flexibility in the lateral directions. An example of a prism grid is shown in Fig. 1. Employment of directionally structured grids has three other important advantages. First, an implicit numerical scheme that would alleviate stiffness of computations, due to a larger allowable time step compared to corresponding explicit schemes, is possible in the "structured" direction. Second, implementation of algebraic turbulence models on semiunstructured grids is much simpler when compared to a fully unstructured grid case.[7] Third, in comparison to tetrahedral grids, the prismatic grid requires significantly less memory, which is essential to three-dimensional Navier-Stokes computations. The structure of the prismatic grid in one of the directions can be exploited to reduce storage to the amount required for two-dimensional Navier-Stokes solvers with triangles.[8]

There are two main approaches related to grid generation. The first requires solution of partial differential equations, which can be either elliptic[9] or hyperbolic.[10-12] Elliptic grid generators usually yield smooth grids. However, it can be difficult to impose grid orthogonality and high boundary surface resolution. In addition, the setup of boundary points can be time consuming. Finally, numerical solution of the elliptic differential equations can be quite expensive. On the other hand, hyperbolic grid generators can produce nearly orthogonal grids with very good clustering control requiring one or two orders of magnitude less computing time compared to the elliptic generators. However, hyperbolic grid generation methods are less robust, and they tend to propagate discontinuities in the mesh. The second approach includes the algebraic generators, which generate grids without solving any differential equations. They are the least expensive to use since no differential equations need be solved, and they control grid quality directly. However, they require a relatively large amount of logic to generate the grid. Such methods are frequently of the hyperbolic type, which means that they propagate an initial grid on the body surface away from the body.[4,5]

Up to the present, only one grid generator of prisms has been under development.[13] The developed method combines

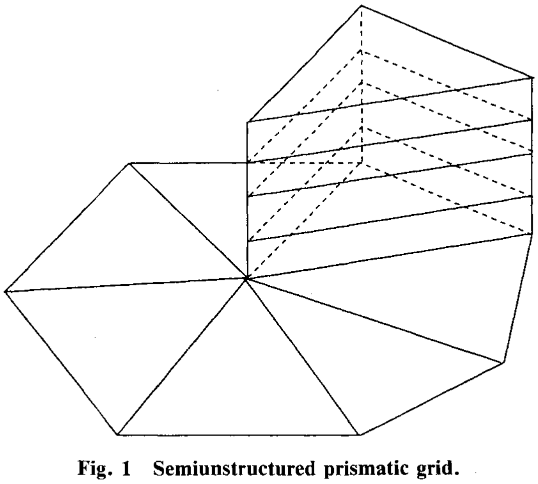


Fig. 1 Semiunstructured prismatic grid.

Received May 4, 1992; revision received Feb. 7, 1993; accepted for publication March 11, 1993. Copyright © 1993 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics. Member AIAA.

†Graduate Research Assistant, Department of Aerospace Engineering and Engineering Mechanics.

the efficiency and good grid control of algebraic generators with the smoothing effect of elliptic methods. The algebraic technique marches a surface consisting of triangular faces away from the body surface.

Smoothness of the generated grid is attained by applying Laplacian-type smoothing steps. In addition, redistribution of the nodes along the marching direction is employed to attain specific grid resolution and uniformity requirements. Two main features of the prisms generation scheme are a new method for calculating the normal to surface vectors at nodes and employment of *voxels* for generating smooth surfaces around bodies. Other aspects of the developed grid generator are its direct control of grid orthogonality and spacing, as well as its robustness for treatment of relatively complex three-dimensional aircraft geometries.

In the following, the prismatic grid generation method is described. Then, applications of the grid generator to different geometries, including a version of the F-16A aircraft, are presented and discussed.

II. Generation of Prismatic Grid

An unstructured triangular grid is employed as the starting surface to generate a prismatic grid. This grid, covering the body surface, is marched away from the body in distinct steps, resulting in the generation of structured prismatic layers in the marching direction (Fig. 1). The process can be visualized as a gradual inflation of the body's volume. There are two main stages in the algebraic grid generation process. In the first, the destination of the marching surface is determined; in the second, the nodes are positioned on that surface. The marching surface is determined by employing a new technique based on voxels. The nodes are positioned by determining the marching vectors corresponding to the nodes of the previous surface.

Direct control of grid orthogonality and spacing is a main advantage of the algebraic method since the accuracy of Navier-Stokes solvers is critically dependent on these properties.[14,15] Algebraic grid generators may yield a grid that is nonsmooth and that may overlap. A grid generator must produce a grid with no overlapping faces, which may occur especially in concave regions. In both stages of the present algebraic method, elliptic-type steps are employed in the form of Laplacian smoothing.

Each node on the marching surface is advanced along a marching vector. If strict orthogonality is imposed, the marching vector is the surface normal vector at the node. However, the resulting grid may overlap in concave regions. Furthermore, the grid lines in the normal direction will have kinks from one layer of prisms to the next.

The goal of the marching scheme is to reduce the curvature of the previous marching surface at each step while ensuring smooth grid spacing to avoid surface overlap. At the same time, a specified degree of grid orthogonality must be maintained. These two conditions define a valid grid. The developed method reduces surface curvature by marching the nodes to a known, smooth surface which encloses the current marching surface. A novel method for the construction of this conformal target surface is presented. The marching surface is modeled by its voxel representation, which is then smoothed.

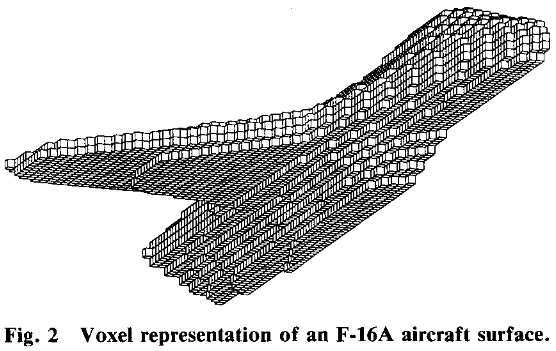


Fig. 2 Voxel representation of an F-16A aircraft surface.

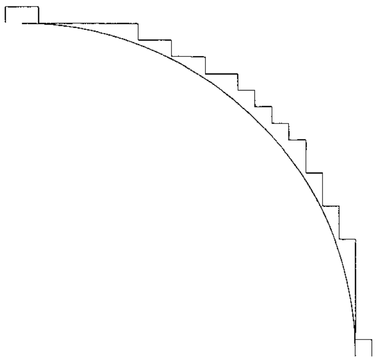


Fig. 3 Outer surface of voxels representing a sphere (two-dimensional cut).

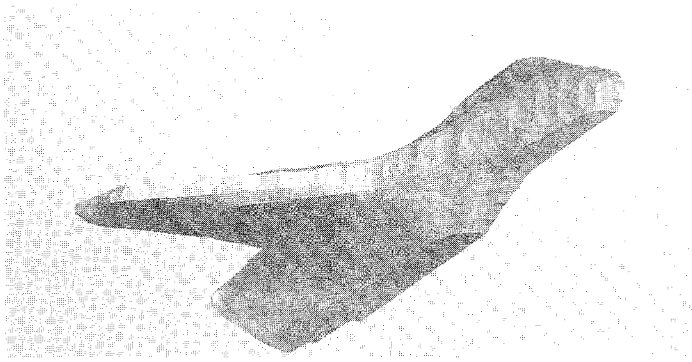


Fig. 4 Smoothed voxel representation of the F-16A aircraft surface.

A. Smooth Voxel Generation

A voxel is a three-dimensional element used to approximate a point in space. It may have any shape which will entirely fill a domain when voxels are placed adjacent to each other, thus conserving the volume. In the present work, parallelepiped voxels are used.[16] The voxel representation of an object comprises all voxels partially occupying any part of the object. Figure 2 shows the voxel representation of the surface of an F-16A aircraft.

In general, the voxel representation is externally bounded by quadrilateral faces which do not intersect the object. It is this aspect of voxel representations which allows them to be used to generate a conformal target surface. Figure 3 shows a two-dimensional section of a spherical surface, as well as its voxel representation.

The steps in generating a smooth, conformal, voxel surface are as follows: 1) Construct all of the voxels intersecting the current marching surface. 2) Cull all of the interior voxel faces and nodes. 3) Smooth the resulting quadrilateral face structure. Figure 4 shows the corresponding smoothed voxel representation of the same aircraft surface as in Fig. 2.

Three important facets of this method should be mentioned:

1) The smoothing of the voxel face structure must be done in a manner which prevents voxel nodes from moving toward the marching surface. A voxel node is allowed to move only if its new position is visible from the neighboring faces. Laplacian-type smoothing is applied a number of times (typically 10–20). The resulting smooth surface will be termed the target surface. Nodes forming the triangular faces of the prismatic grid need to be placed on this surface.

2) The dimensions of the voxel should be such that surface details are captured. Excessively large voxels will "ignore" small-scale surface features. Figure 5 illustrates a complex corner topology and an acceptable voxel dimension for representing it. Figure 6 illustrates an unacceptably large voxel dimension to represent the same corner topology. The main features of the initial corner topology are essentially lost.

3) If a large number of smoothing operations is performed, the voxel nodes move far from their original positions. This

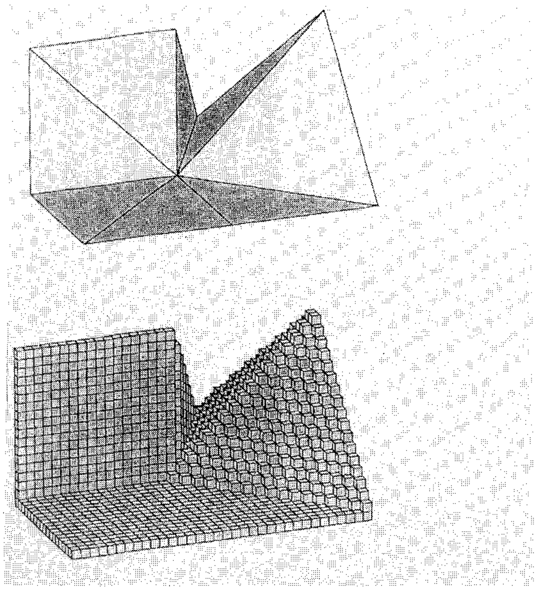


Fig. 5 Small voxel sizes capture complex corner surface topology.

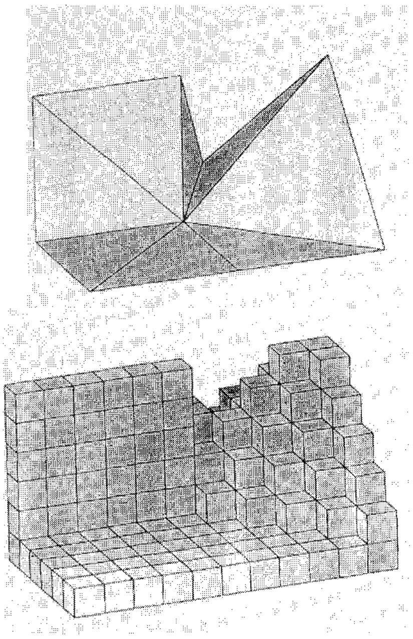


Fig. 6 Large voxel sizes ignore complex corner surface topology.

has the same effect as choosing a voxel dimension too large to capture surface detail.

It should be emphasized that the resulting target surface has reduced surface curvature compared to the marching surface. Concave regions are "filled in," whereas convex areas are smoother.

B. Node Normals and Marching Vectors

One of the main issues of the present method is determination of the vectors along which each node of the triangular surface of the previous layer of prisms will march (marching vectors).[16] The initial step is calculation of the vectors that are normal to the surface. For a continuous analytic surface, the calculation of the surface normals is trivial. However, the grid marching surface comprises discrete faces of discontinuous slope. The group of faces sharing a common node will be termed manifold (Fig. 7). Each manifold on the marching surface has a unique configuration, which may have singular topology. Therefore, application of a general algorithm to compute the surface normal vector at the nodes is very difficult. Furthermore, nodes that lie on concave surfaces must march away from the concavity. As the node marches to the target surface, it must not intersect the current marching surface prior to intersecting the target surface.

This situation is avoided by enforcing a visibility condition, which constrains the marching vectors such that the new node position is visible from all faces of the manifold. This region is the visibility region. It has the shape of a polyhedral cone extending outward from the node as shown in Fig. 7. To simplify the constraints, a visibility cone with a circular cross section and half-cone angle β centered on the normal vector is constructed at the node. This cone lies completely within the visibility region (see Fig. 7).

An apparent approach to calculate surface normals at the nodes is to employ an averaging of the manifold face normals. One such formula is:

$$\hat{N}_i = \frac{\Sigma_k W_k \hat{n}_k}{\Sigma_k W_k} \tag{1}$$

where $\hat{n}_k$ is the normal vectors of the faces forming manifold i, and W_k is a weighting factor equal to the angle between edge k and edge $k-1$ as illustrated in Fig. 7. The formula has also been used with $W_k = 1$.

This method produces reasonable surface normals for relatively smooth manifolds. However, it may result in a violation of the visibility condition for manifolds which discretize complex topologies such as wing-fuselage junctions, sharp edges, and combined acute/obtuse junctions. A new method of determining the normal vectors has been developed in the present work and tested for a variety of complex topologies. A unit normal vector $\hat{N}_i$ and a visibility half-cone angle β_i are calculated for each node i with position vector $\boldsymbol{P}_i$ on the current marching surface.

The steps involved in the determination of these quantities are described with the aid of Fig. 8.

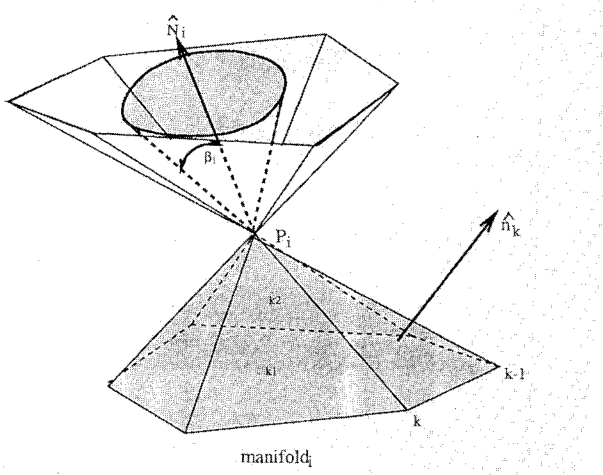


Fig. 7 Marching of point P_i constrained within the shaded area (visibility condition). Manifold surface formed by faces surrounding a node (lower shaded area).

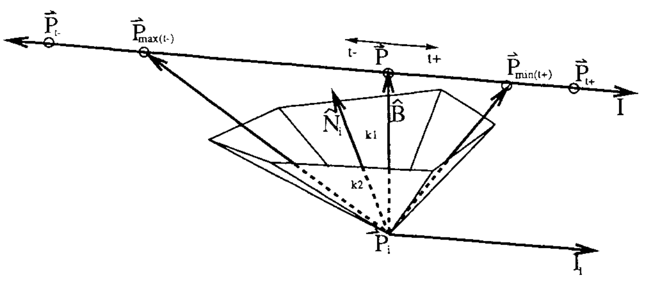


Fig. 8 Determination of normal vector at the center node of a manifold surface.

1) Determine which face planes (f_{k1}, f_{k2}) of manifold i intersect to form the most acute wedge by finding $k1$, $k2 \in K$ such that

$$\hat{n}_{k1} \cdot \hat{n}_{k2} \leq \hat{n}_i \cdot \hat{n}_j \tag{2}$$

for all i, $j \in K$.

2) Construct the intersection vector of the face planes f_{k1}, f_{k2}:

$$\boldsymbol{I}_1 = \hat{n}_{k1} \times \hat{n}_{k2} \tag{3}$$

3) Construct the vector $\hat{\boldsymbol{B}}$ lying on the bisection plane of the faces f_{k1}, f_{k2} and which is perpendicular to $\boldsymbol{I}_1$:

$$\hat{\boldsymbol{B}} = (\hat{n}_{k1} + \hat{n}_{k2})/2 \tag{4}$$

where $\hat{n}_{k1}$, $\hat{n}_{k2}$ are the unit normal vectors to faces f_{k1}, f_{k2}, respectively.

4) Construct a parametric line $\boldsymbol{I}$, which is parallel to $\boldsymbol{I}_1$ and passing through a point with position vector $\boldsymbol{P}$, which is unit distance from the node $\boldsymbol{P}_i$.

$$\boldsymbol{I} = \boldsymbol{P} + t\boldsymbol{I}_1 \tag{5}$$

where

$$\boldsymbol{P} = \boldsymbol{P}_i + \hat{\boldsymbol{B}} \tag{6}$$

5) Find all intersections $\boldsymbol{P}_{t\pm}$ of the remaining f_k face planes with the line $\boldsymbol{I}$ in terms of the parameter t, where $t^{\pm}$ denotes positive and negative values of t for points lying on either side of point $\boldsymbol{P}_i$ (Fig. 8).

6) The node normal $\hat{N}_i$ is the bisector of the vectors

$$(\boldsymbol{P}_{\max(t^-)} - \boldsymbol{P}_i) \text{ and } (\boldsymbol{P}_{\min(t^+)} - \boldsymbol{P}_i) \tag{7}$$

and the visibility half-cone angle is the angle between $\hat{N}_i$ and face f_{k1}

$$\beta_i = \pi/2 - \cos^{-1}(\hat{N}_i \cdot \hat{n}_{k1}) \tag{8}$$

This process has yielded consistently valid normal vectors at the nodes by constructing the vector most normal to the most acute face planes. Essentially, it does this by maximizing the minimum angle between $\hat{N}_i$ and all $\hat{n}_k$. Figure 9 shows a singular manifold located at the junction between the wing trailing edge and the fuselage. It is observed that the new technique yields reasonable normal vectors.

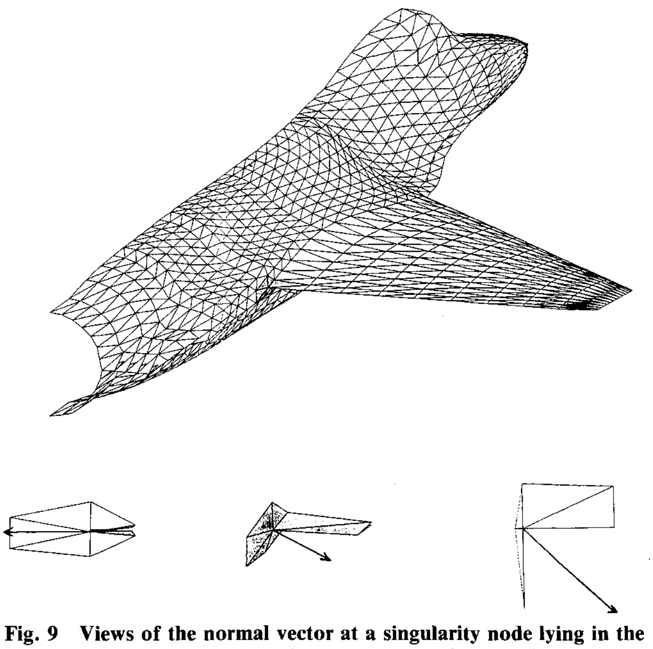


Fig. 9 Views of the normal vector at a singularity node lying in the junction between the wing trailing edge and the fuselage (shaded area on aircraft surface).

A specified degree of orthogonality can be enforced by restricting the value of the half-cone angle β to a specified value

$$\beta_i = \min(\beta_i, \beta_{\max}) \tag{9}$$

where $\beta_{\max}$ is the maximum allowable deviation angle from orthogonality. This has the effect of limiting the valid deviation regime of the marching vector. The visibility constraint is still met, but excessively skewed marching of the nodes is avoided.

The initial marching vectors $\hat{m}_i$ are the normal vectors $\hat{N}_i$. However, this may not provide a valid grid since overlapping may occur, especially in regions of the grid with closely spaced nodes. To prevent overlapping, the marching vectors must be altered to increase the divergence of all $\hat{m}_k$ on each manifold i. Therefore, a number of smoothing passes (typically five) are performed over all the nodes on the marching surface:

$$\hat{m}_i = \Sigma_k \hat{m}_k / K \tag{10}$$

After each step, a check is performed to determine if $\hat{m}_i$ lies outside the visibility cone. The following equation is used:

$$(\hat{m}_i \cdot \hat{N}_i) \leq \cos(\beta_i) \tag{11}$$

If $\hat{m}_i$ does lie outside the visibility cone, then it is projected onto the cone and held fixed for subsequent operations.

C. Advancement and Smoothing of Marching Surface

The nodes on the previous surface are connected to the points of intersection of the target surface and the marching vectors $\hat{m}_i$. A special situation arises if a valid intersection cannot be found. The marching vector is then allowed to deviate within the visibility cone until a valid intersection with the target surface is found. If this fails, the degree of orthogonality constraint is successively relaxed and a search is made over the target faces to find the closest point to node i, while maintaining the visibility condition. If a valid intersection point still cannot be found, the target surface is not sufficiently conformal and the voxel dimension is reduced.

Marching to a point on the target surface may result in a reduction in face area caused by "convergence" of the marching grid lines. This will yield a nonuniform mesh; eventually, overlapping may occur. This situation may arise during growth of concave regions due to a decrease in the surface area available for node placement. The nodes on the target surface are redistributed by applying a Laplacian-type operator so that the surface elements change their areas and curvature smoothly. A distance-weighted averaging procedure is performed on all manifolds on the grown surface to obtain an initial correction vector $\boldsymbol{c}_i$ defined as:

$$\boldsymbol{c}_i = \frac{\Sigma_k W_k \boldsymbol{P}_k^1}{\Sigma_k W_k} - \boldsymbol{P}_i \tag{12}$$

where k denotes each one of the K points that are directly connected to node i with an edge and W_k is the length of that edge. This averaging process is repeated one or two times. Figure 10 shows the correction and marching vectors when advancing a node.

D. Mesh Adaptivity

1. *Redistribution of Grid Nodes*

Flexibility in specifying grid spacing along the marching lines is crucial for accuracy of Navier-Stokes computations. A scheme is employed that redistributes the nodes along each one of the marching lines emanating from points on the body surface. In other words, the marching directions are maintained but the marching distances (Δn) are modified. This is

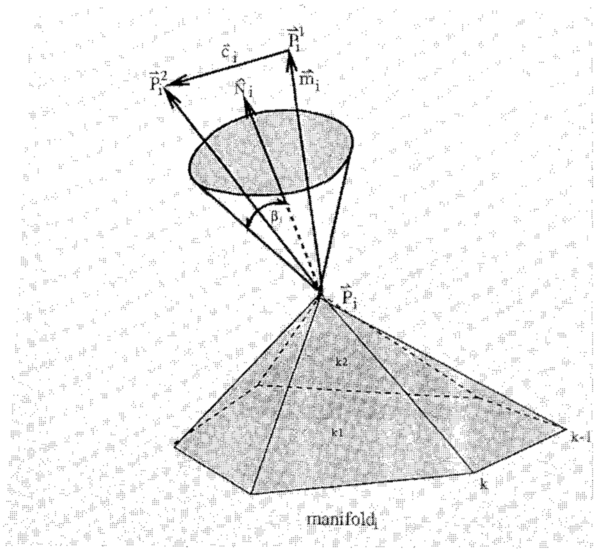


Fig. 10 Initial P_i^1 and corrected P_i^2 positions of grid point P_i on the new marching surface.

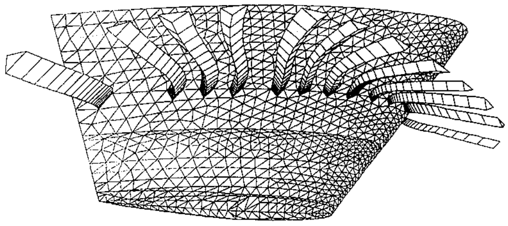


Fig. 11 Marching lines emanating from points of the Onera M6 wing surface.

accomplished by performing a cubic-spline fit to each of the marching lines using the prism node locations for the spline knots. The nodes are then redistributed along the splined lines. The distribution is such that the new node positions satisfy certain grid spacing requirements. In the present work, the spacing of the first point off the body surface is specified along with a constant stretching factor ω. Then, the n positions of the points are given by the formula

$$n_{j+1} = n_j + \omega(n_j - n_{j-1}) \tag{13}$$

A typical value of ω is 1.1.

Adaptive redistribution of the grid points can also be employed with the present technique. The evolving flow solution can be monitored, and the nodes can be moved along the marching lines so that the boundary layer is resolved sufficiently. For example, the points can be moved so that the y^+ values at the wall are smaller than a specified number.[17]

2. *Frame mesh*

The prisms generated by marching the surface form a skeleton mesh of valid prisms within which any number of new cells may be generated. Generation of this frame mesh is robust since the triangular faces on the marching surfaces are relatively large and overlapping of the marching lines can be avoided. The marching process generates a relatively small number of prisms occupying a relatively large volume and is therefore an undesirable grid for Navier-Stokes computations. The new prisms are generated by subdividing the original prisms. This subdivision can be applied along either the normal or the lateral directions, or along both. In this way, smaller grid cells are created that have the required dimensions. This technique can be included in an adaptive grid scheme which employs local grid embedding to optimize the mesh to the solution.

III. Applications

The developed algebraic prismatic grid generator was applied as an initial case to the Onera M6 wing. Robustness, efficiency, and computer requirements of the method were investigated through generation of a grid around a version of the F-16A aircraft. A Sun workstation with speed of 2 Mflops was employed for the computations. The required generation time was found to be 0.018 CPU s/cell. The F-16 grid consists of 96,320 prisms and it took 1742 s of computing time on the Sun station to generate it.

A. Onera M6 Wing

The Onera M6 wing has a leading-edge sweep of 30 deg, an aspect ratio of 3.8, a taper ratio of 0.56, and symmetrical airfoil sections of 10% maximum thickness-to-chord ratio. The wing surface consists of 3239 triangular faces and 1654 points.

Marching lines that emanate from different points of the wing surface are shown in Fig. 11. In addition, the figure shows the surface triangulation, which constitutes the initial surface that is marched outward away from the wing along the direction of the marching vectors.

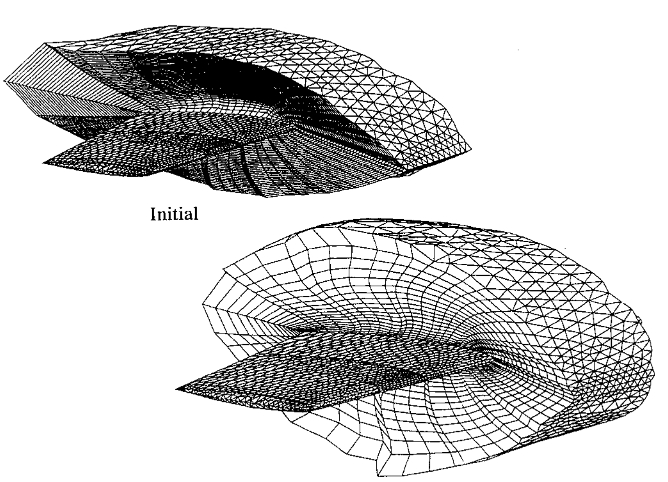


Fig. 12 Effect of grid redistribution along the marching lines; initial and redistributed grid around Onera M6 wing.

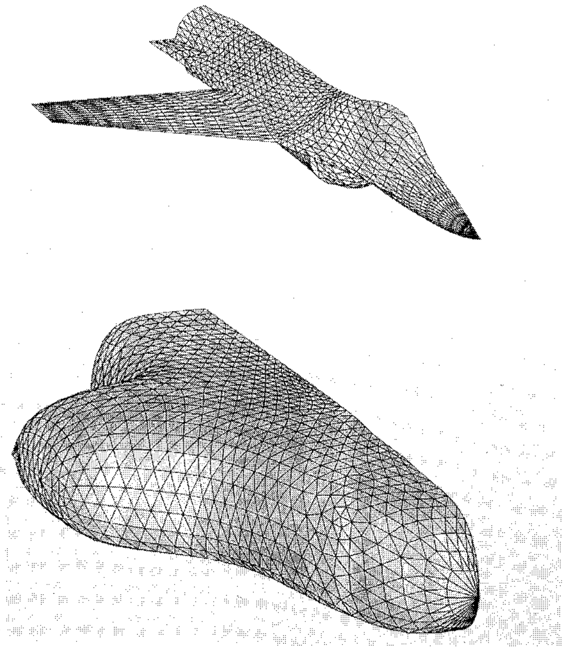


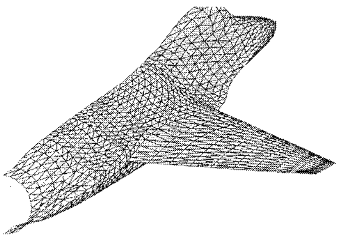
Fig. 13 Initial and grown surfaces of the prismatic grid around F-16A body (unstructured part of grid); fore view.

The prismatic grid can be quite flexible in covering complex topologies since both bending and twisting of the prisms is allowed during the marching outward, as seen in Fig. 11. Twisting is allowed, since the marching vectors are generally not parallel to each other. An example of grid point redistribution is illustrated in Fig. 12, which shows the initial grid before redistribution, as well as the final grid that is obtained after moving the points along the marching lines. The figure includes part of the wing triangulated surface (unstructured), as well as the surface of the quadrilateral faces (structured).

The spacing of the initial grid along the marching lines is quite large close to the wing, which is unacceptable for accurate high Reynolds number Navier-Stokes computations. The final redistributed grid has a specified minimum spacing at the wall equal to 0.005, and constant stretching factor $\omega = 1.1$. The final number of prismatic cells is 129,560, whereas the number of grid nodes is 67,814.

B. F16-A Aircraft

An F-16A aircraft geometry was chosen as a case for the developed grid generator, since the complexity and singularities of the surface are a severe test for the method. The aircraft geometry was supplied by Flores et al.[18] and is shown in the first part of Fig. 13. The main features of the configuration are the forebody, canopy, leading-edge strake, wing, shelf regions, and inlet. The surface triangulation consists of 2408 triangular faces and 1259 points. Generation of a grid around this geometry is quite complex, especially so at the junctions between the different aircraft components. Two parts of the generated prismatic grid require examination in terms of quality. The first is the grid formed by the triangular faces of the prisms (unstructured part), whereas the second is the grid formed by the quadrilateral faces (structured part).



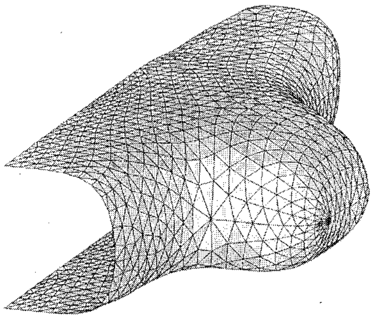


Fig. 14 Initial and grown surfaces of the prismatic grid around F-16A body (unstructured part of grid); aft view.

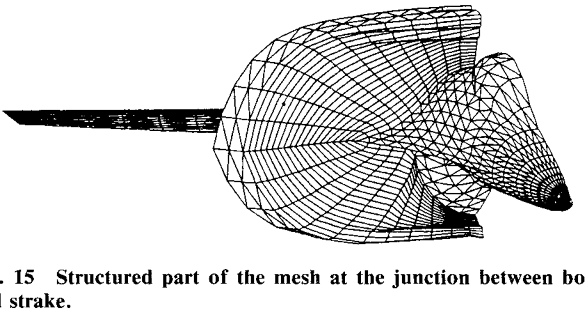


Fig. 15 Structured part of the mesh at the junction between body and strake.

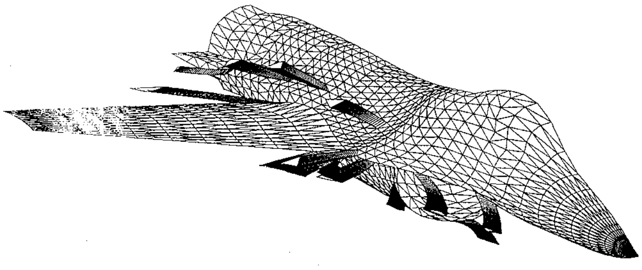


Fig. 16 Piles of prisms corresponding to different faces of the F-16A aircraft surface.

Two views of the initial and grown surfaces are shown in Figs. 13 and 14. The growth of the grid is illustrated after 15 marching steps. The effect of the marching process is similar to inflation of the original body volume. It is observed that the distribution of points on the grown surface is quite smooth. The highly singular regions at the aircraft nose, the wing leading and trailing edges, the wing tip, the canopy, as well as the inlet have been smoothed out on the grown surface. Furthermore, the grid spacing on the grown triangular surface is smoother compared to the initial triangulated body surface. The singular concave regions at the junctions between the wing and the fuselage, as well as between the engine inlet and the body, have been "filled in," and the grid is more uniform over those regions compared to the initial grid on the body.

A view of the structured part of the prismatic grid is shown in Fig. 15. It should be noted that the depicted surface is not planar. The grid spacing on the quadrilateral surface is quite uniform. Furthermore, the marching lines emanating from the aircraft surface are quite smooth, including the ones that correspond to singular points. The lines do not have the "kinks" that appear frequently with algebraic grid generators. The structured part of the grid is also shown in Fig. 16 by means of piles of prisms corresponding to different triangular faces on the aircraft surface. It should be noted that the degree of twist and bending of the piles is quite minimal even in regions of surface singularities.

The final number of prismatic cells is 96,320, whereas the number of grid nodes is 50,360. The required computing time was 444 s on a Sun workstation (2 Mflops speed), which implies generation of 217 prisms/s.

IV. Summary

Generation of semiunstructured prismatic grids was accomplished for three-dimensional geometries including an aircraft configuration. A new method that couples both algebraic and elliptic steps was introduced. The method yielded smooth grids at low computation cost. The technique marches a surface consisting of triangular faces away from the body surface, so that a semiunstructured grid is created. The developed method of calculating the marching vectors at the nodes proved to be general and yielded grids of good quality even in regions of severe geometric singularities. Employment of voxels to represent the marching surfaces proved to be a robust method of eliminating concave regions and smoothing out convex areas, achieving "inflation" of the volume of the body.

Main advantages of the developed grid generator are its low computation cost, its direct control of grid orthogonality and spacing, as well as its generality for treatment of complex three-dimensional geometries. The required computing time was found to be 0.0046 s per generated grid cell. A relatively complex version of the F-16A aircraft was included to investigate efficiency, as well as robustness of the method to handle complex topologies. The required computing time for the aircraft geometry was 444 s on a Sun workstation of approximately 2 Mflops speed.

Acknowledgment

This work was supported by AFOSR Grant 91-0022 and monitored by L. Sakell.

References

[1] Baker, T. J., "Developments and Trends in Three Dimensional Mesh Generation," *Applied Numerical Mathematics,* Vol. 5, 1989, pp. 275-304.

[2] Eiseman, P. R., and Erlebacher, G., "Grid Generation for the Solution of Partial Differential Equations," ICASE Rept. 87-57.

[3] Baker, T. J., "Unstructured Meshes and Surface Fidelity for Complex Shapes," AIAA Paper 91-1591-CP, 1991.

[4] Lo, S. H., "A New Mesh Generation Scheme for Arbitrary Planar Domains," *International Journal for Numerical Methods in Engineering,* Vol. 21, 1985, pp. 1403-1426.

[5] Lohner, R., and Parikh, P., "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method," AIAA Paper 88-0515, Jan. 1988.

[6] Peraire, J., Peiro, J., Formaggia, L., Morgan, K., and Zienkiewicz, O. C., "Finite Element Euler Computations in Three Dimensions," AIAA Paper 88-0032, AIAA 26th Aerospace Sciences Meeting, Reno, NV, Jan. 1988.

[7] Kallinderis, Y., "Algebraic Turbulence Modeling for Adaptive Unstructured Grids," *AIAA Journal,* Vol. 30, No. 3, 1992, pp. 631-639.

[8] Kallinderis, Y., "A New Finite-Volume Navier-Stokes Scheme on Three-Dimensional Semi-Unstructured Prismatic Elements," AIAA Paper 92-2697, Palo Alto, CA, June 1992.

[9] Thompson, J. F., Warsi, Z., and Mastin, C. W., *Numerical Grid Generation,* Elsevier Science, Amsterdam, 1985.

[10] Chan, W. M., and Steger, J. L., "A Generalized Scheme for Three-Dimensional Hyperbolic Grid Generation," AIAA Paper 91-1588-CP, 1991.

[11] Steger, J. L., and Rizk, Y. M., "Generation of Three-Dimensional Body-Fitted Coordinates Using Hyperbolic-Partial Differential Equations," NASA TM 86753, April 1985.

[12] Steger, J. L., "Grid Generation with Hyperbolic Partial Differential Equations for Application to Complex Configurations," *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields,* Elsevier Science, B.V., North-Holland, 1991, pp. 871-886.

[13] Nakahashi, K., "Optimum Spacing Control of the Marching Grid Generation," AIAA Paper 91-0103, Jan. 1991.

[14] Kallinderis, Y., "Adaptation Methods for Viscous Flows," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, CFDL-TR-89-5, Cambridge, MA, May 1989.

[15] Kallinderis, Y., "A Finite-Volume Navier-Stokes Algorithm for Adaptive Grids," *International Journal for Numerical Methods in Fluids,* Vol. 15, 1992, pp. 193-217.

[16] Ward, S., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex Three-Dimensional Geometries," M.S. Thesis, Dept. of Aerospace Engineering and Engineering Mechanics, Univ. of Texas at Austin, Rept. CAR 92-7, Austin, TX, Aug. 1992.

[17] Kallinderis, Y., and Baron, J. R., "A New Adaptive Algorithm for Turbulent Flows," *Computers and Fluids Journal,* Vol. 21, No. 1, 1992, pp. 77-96.

[18] Flores, J., Reznick, S. G., Holst, T. L., and Gundy, K., "Transonic Navier-Stokes Solutions for a Fighter-Like Configuration," AIAA Paper 87-0032, 1987.