AIAA Journal
Vol. 34, No. 2, February 1996

Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flows Around Complex Geometries

Yannis Kallinderis,* Aly Khawaja,† and Harlan McMorris†
University of Texas at Austin, Austin, Texas 78712-1085

A method for the generation of hybrid prismatic/tetrahedral grids is described for complex three-dimensional geometries, including multibody domains, and employed for viscous flows around an aircraft. The prisms cover the region close to each body's surface, while tetrahedra are created elsewhere. A special method is presented that allows the generation of single-block, nonoverlapping prismatic meshes even in complex geometries that contain narrow gaps and cavities. Examples of cases treated by this method include wing–engine configurations as well as multi-element wings. The second development is a combined octree/advancing front method for the generation of the tetrahedra of the hybrid mesh. The main feature of this octree-based tetrahedra generator is that it does not require the creation of a background mesh by the user for the determination of the grid-spacing and stretching parameters. These are determined via an automatically generated octree. The hybrid grid generator is robust, geometry independent, and requires no user intervention. The suitability of hybrid meshes for capturing viscous flow phenomena is demonstrated by simulation of viscous flows around a high-speed civil transport type of aircraft configuration. Reduction in computer resources has been substantial, allowing flow simulations to be performed on workstations rather than supercomputers.

I. Introduction

THERE is an ever-increasing demand to perform flow simulations that incorporate the complete details of geometry as well as sophisticated flow physics. This has led to the development of numerical algorithms that can simulate the actual flow phenomena with greater fidelity. However, the success of these algorithms hinges on the grid that models the geometry. Grid-generation methods for two-dimensional models have long existed, and the general lack of complexity of the simpler two-dimensional models has not quite challenged the efforts in this area. However, demands for generating better three-dimensional geometric models for flow simulations involving complex geometries have completely changed the perspective of grid-generation strategies. As a consequence, grid-generation efforts have earned equal significance as that of numerical solver efforts.

Structured meshes consisting of blocks of hexahedra and unstructured grids consisting of tetrahedra have been the traditional means of discretizing three-dimensional flow domains.[1–4] An emerging technique is that of using semistructured prisms to discretize the viscous region in close proximity to the surface of the geometry.[5–8] The prismatic elements are suitable to capture thin boundary layers. The quadrilateral faces normal to the surface provide good orthogonality and grid-clustering capabilities, whereas the triangulation in the lateral direction allows flexibility in surface modeling. The structured prism layers allow implementation of directional multigrid acceleration schemes.[9,10] The structure can also be exploited when using algebraic turbulence models.[11] Furthermore, implicit numerical schemes can be introduced in the normal direction to alleviate stiffness of viscous flow computations. Finally, the structured nature also leads to reduced memory requirements on the solver.[7,10]

Strongly directional viscous stresses exist in a relatively small region of the flow domain, close to the wall. Prismatic grids alone cannot cover multibody domains. Furthermore, they do not offer the advantages mentioned previously in the regions away from the surface. On the other hand, tetrahedra have the capability to fill a given volume of arbitrary shape. Hence, an effective strategy for grid generation would be to use both the prisms and tetrahedra by generating a hybrid grid.[5,7] The prisms cover the regions close to each of the bodies, whereas tetrahedra fill in the irregular gaps in between the prismatic layers. The tetrahedra may also be needed to capture wakes that extend beyond the thin prismatic layers around the surface.

An important issue arising with the use of semistructured prismatic grids is the treatment of narrow gaps in regions such as wing–engine configurations and in between different bodies in multiply-connected domains.[5,7] In the present work, a special method is developed that allows the generation of single-block, nonoverlapping prismatic meshes by reducing the prism marching distances in such regions. It will be termed the automatic receding method (ARM). A smooth transition is obtained between the thickness of the layers of prisms in the gaps and the thickness elsewhere. It should be noted that it is the effectiveness of the ARM that allows the structure of the prisms to be maintained even in the most complex geometries. The key features of this method are that it is blind to the specifics of the geometry and that no user intervention is required. The cases of a two-element wing and the high-speed civil transport (HSCT) aircraft with engines are considered as tests of the technique.

A current popular method for tetrahedra generation is the advancing front technique.[1,2,12] This method is very suitable for hybrid grid generation since it starts with the triangulation of the outermost prismatic surface. However, advancing front procedures require background meshes that dictate the spacing and stretching parameters for the generated tetrahedra. Generation of the background mesh has been a time-consuming and user-dependent part of the previous advancing front methods. The user specifies sources throughout the domain and Poisson equations needed to be solved. This can be expensive, depending on the number, location, and type of these sources. A new octree/advancing front method for the generation of the tetrahedra of the hybrid mesh is presented. The main feature that is different from previous advancing front generators is that it does not require a user constructed background mesh for determination of the grid spacing and stretching parameters. A special octree is constructed via a divide-and-conquer method of the space outside of the region covered by the prisms. The grid spacing is then determined based on the size of local octants that form the octree. The creation of the octree is robust, automatic, and geometry independent.

Hybrid grids are generated for the HSCT aircraft configuration with and without engines, as well as for a two-element wing configuration. Results of a viscous flow simulation on the HSCT configuration without engines are also presented as a demonstration of the

Received March 2, 1995; revision received Sept. 8, 1995; accepted for publication Sept. 11, 1995. Copyright © 1995 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

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

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

effectiveness and suitability of the generated hybrid grids. The grid required only 175,000 prisms and 170,000 tetrahedra. The structured nature of the prisms and the relatively small number of tetrahedral cells required led to small computer requirements in terms of CPU time and memory storage. As a result, the simulations were run on an IBM RS/6000 model 390 workstation rather than a supercomputer.

II. Generation of Prisms via a Marching Method

An unstructured triangular grid is employed as the starting surface to generate a prismatic mesh. This grid, covering the body surface, is marched away from the body in distinct steps, resulting in generation of semistructured prismatic layers in the marching direction. 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. The process can be visualized as a gradual inflation of the body's volume. There are three main aspects of the algebraic grid-generation process: 1) determination of the directions along which the nodes will march (marching vectors), 2) determination of the distance by which the nodes will march along the marching vectors, and 3) smoothing operations on positioning of the nodes on the new layer. It should be noted that the approach can start from any surface triangulation.

A. Determination of the Marching Vectors

Each node on the marching surface is advanced along a marching vector. The marching direction is based on the node-manifold, which consists of the group of faces surrounding the node to be marched. The primary criterion to be satisfied when marching is that the new node should be visible from all the faces on the manifold (visibility criterion).[6]

The node-normal vector lies on the bisection plane of the two faces on the manifold that form the wedge with the smallest angle. Its location on this plane is determined by bisecting the visibility region on the plane. 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 the node-normal and all of the surrounding face normals. This vector is then used as the marching direction for the nodes on the surface to form the new layer. A more detailed description of the marching procedure can be found in Ref. 6. The aforementioned marching vectors technique has also been adopted by Refs. 13 and 14.

B. Marching Step Size

Determination of marching distances is based on the characteristic angle β_{av} of the manifold of each node to be marched. This angle is computed using the average dot product between the pairs of faces forming the manifold. The marching distance is a linear function of β_{av}. It yields relatively large marching distances in the concave regions and small distances in the convex areas of the marching surface. Specifically, the distance Δn is

$$\Delta n = (1 + \alpha)\Delta n_{\text{av}} \tag{1}$$

where Δn_{av} is the average marching step for that layer, and α is a linear function of the manifold angle β_{av}. The sign of α is positive for concave regions and negative for convex regions.

The average marching step for each layer (j), Δn_{av} is computed based on a user-specified initial marching step Δn_0 and a stretching factor st, as follows:

$$\Delta n_{\text{av}}^{j} = \Delta n_0 \times st^{(j-1)} \tag{2}$$

C. Smoothing Steps

The initial marching vectors are the normal vectors. 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 directions of the marching vectors must be altered. Altering of the directions should not end abruptly in the local neighborhood of the nodes involved, since this may cause overlapping in nearby regions. A gradual reduction of the magnitude of the change in the vector direction is accomplished via a number of weighted Laplacian type smoothing operations over the marching vectors of all nodes. Typically, 10 smoothing passes are performed. These smoothing steps rotate each original marching vector based on the normal vectors of its surrounding manifold nodes as follows:

$$V_i = \omega V_i' + \frac{(1-\omega)}{\sum_j 1/d_{ij}} \sum_j (1/d_{ij}) V_j \tag{3}$$

where V_i' and V_i are the initial and final marching vectors of node i and V_j are the marching vectors of the surrounding nodes j that belong to the manifold of node i. The weighting factor ω is a function of the manifold characteristic angle β_{av}. It has small values in concave regions, and relatively large ones in convex areas. The averaging of the marching vectors of the neighboring nodes is distance weighted with d_{ij} denoting the distance between nodes i and j.

A similar procedure is employed for the smoothing of the marching steps Δn to eliminate abrupt changes in cell sizes.

D. Constraints Imposed to Enhance Quality

Typical Navier–Stokes integration methods impose restrictions on the spacing of the points along the marching lines, as well as on smoothness of these lines. In other words, the prismatic grid should not be excessively stretched or skewed. In the present work, constraints are imposed on the lateral and normal distribution of marching step sizes, as well as on the deviation of the direction of the marching vectors from one layer to the next.

The lateral distributions of cell sizes are constrained so that any node on the current marching surface cannot have a step size that is less than half of any of the surrounding nodes or more than twice the size of any node on its manifold. That is,

$$0.5 \times \Delta n_j < \Delta n_i < 2.0 \times \Delta n_j \tag{4}$$

where i is the node in question and j denotes the surrounding nodes on the manifold.

The constraints on the step size variation along each marching line are applied in a similar manner. A node on layer j cannot have a step size that is smaller than that on the previous layer ($j - 1$). Also, it cannot exceed the size of the previous step by more than a factor of $st_{\max}$ (usually set to 1.3). Specifically,

$$\Delta n_{j-1} < \Delta n_j < st_{\max} \times \Delta n_{j-1} \tag{5}$$

In the first few layers of generation (approximately five), the marching vectors are not allowed to deviate from the original normals. That is, no smoothing is allowed. This is done to maximize orthogonality since it is important in the layers close to the surface. For all subsequent layers, the deviation between two consecutive marching vectors V_{j-1} and V_j should be less than a specified angle of 30 deg.

The aforementioned constraints eliminate kinks in the marching vector directions as well as abrupt changes in step sizes, thus providing a smooth mesh suitable for viscous flow computations. If excessive quality is desired, the parameters $st_{\max}$ and the maximum deviation angle can be varied. Since the visibility criterion is the ultimate test for the validity of the mesh, this is the final constraint that is imposed on the grid.

E. Application to HSCT Aircraft Configuration

A HSCT-type of aircraft configuration was chosen to demonstrate the robustness of the prismatic grid generator. The complexity and singularities of the surface provide a suitable test for the method. The surface triangulation consists of 4412 faces and 2275 nodes. A symmetry plane is considered that divides the body. The time required to generate the prismatic grid around the HSCT was approximately 90 s for 40 layers of prisms on an IBM 390 workstation.

Two wire frame views of the prismatic grid are illustrated in Fig. 1. The shaded regions correspond to the surface of the aircraft. It can be seen that the growth of the prisms is substantial enough to be able to capture the entire boundary-layer development. The effect of the marching process is similar to inflating the original body volume. The curvature of the initial surface has been reduced substantially on the final prismatic surface. Additional views of the resulting hybrid mesh for the same HSCT configuration are presented in Sec. V.A.

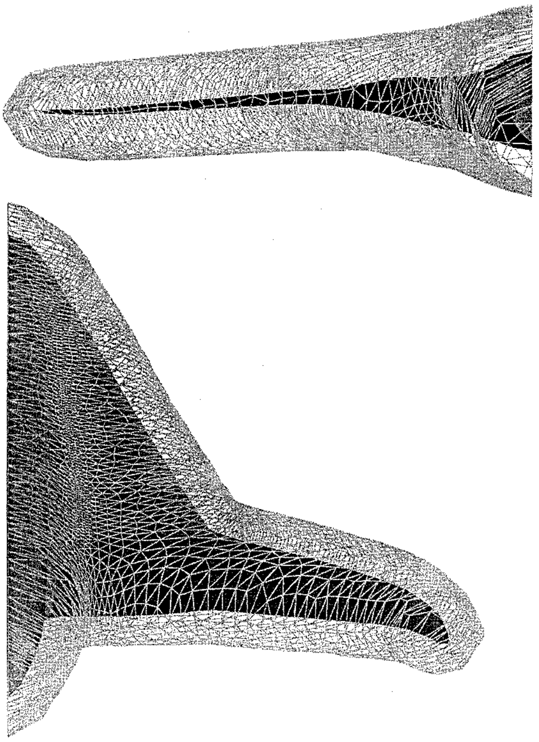

Fig. 1 Wire frame views of the grown prismatic mesh around the HSCT aircraft configuration (the shaded area denotes the surface of the aircraft).

III. ARM for Narrow Gaps and Cavities

Treatment of narrow gaps and cavities in regions such as wing–engine configurations, as well as in between different bodies in multiply-connected domains, has been a major concern for structured and semistructured mesh generators. The structured nature of prisms prohibits filling such complex geometries without overlapping layers if special measures are not taken. A novel method is presented that adjusts the marching step of the prism layers for the treatment of such gaps. The technique allows entirely automatic generation of single-block, nonoverlapping prismatic meshes. Two key features of this ARM are that no user intervention is required, as well as its universality of application to different geometries. The nodes in the vicinity of a cavity are detected by a special algorithm. The marching distances of these flagged nodes are reduced so that the mesh does not overlap. This may result in prismatic meshes of significantly varying local thickness. Smooth variation of the thickness is attained via lateral smoothing of the size of the marching steps.

A. Automatic Detection of Gaps and Cavities

Detecting gaps and cavities on the initial surface is an integral part of the developed ARM. All of the nodes in the cavity need to be identified beforehand so that special measures can be taken to avoid overlapping prism layers. The algorithm is based on the intersection of node-normals with the triangular faces on the surface of the body. It should be noted that gap detection is a preprocessing step and is performed only once for each new surface geometry.

An outward normal is created for each node on the surface. The intersection of this outward normal is then checked with faces on the surface that are close to the node in question. The list of close faces is composed of all faces that lie within a cube centered at the node. The size of the cube is based on the total prismatic layer thickness to be achieved and the local edge lengths of the surface triangulation. If an intersection occurs, the distance between the intersecting face and the node is computed. If this distance is less than a certain threshold specified by the user, the node is flagged, and the distance of intersection is stored in an array. The usual grid-generation procedure then continues, with only slight modifications to account for the flagged nodes.

B. Single-Block, Nonoverlapping Grids via Reduction of Marching Distance

The marching distances of the nodes in the cavity or gap region are reduced to avoid overlapping prism layers. This is done by recomputing the initial marching distance Δn_0 for all of the flagged nodes according to the following equation:

$$\Delta n_0 = \frac{C_1 \times d_G}{\sum_j st^{(j-1)}} \tag{6}$$

where d_G denotes the gap distance computed by the gap detection algorithm, C_1 is a user-specified constant controlling the extent of reduction (usually chosen to be 0.25), st is the stretching factor, and j is the prism layer index. Thus the total thickness of the prism layers in the vicinity of the gap is approximately $C_1 \times d_G$, with slight variations depending on the local curvature of the marching surface. The exact step size for every node on each layer is then determined by Eqs. (1) and (2).

C. Smoothing Steps

To avoid abrupt changes in the thickness of the prism layers due to the local receding, the unflagged nodes in the neighborhood of the cavity are also receded to a certain extent. This extent gradually reduces to zero as the nodes get farther away from the cavity or gap. The procedure that enforces this smooth transition is the following:

1) For each flagged node, find its edge neighbors that are not flagged.

2) For each such unflagged node, calculate the new marching distance, Δn_{new}, according to

$$\Delta n_{\text{new}} = \min(\Delta n_{\text{orig}}, C_2 \times \Delta n_{\text{flagged}}) \tag{7}$$

where Δn_{orig} is the original marching step size for the unflagged node, $\Delta n_{\text{flagged}}$ is the marching step size of the flagged node, and C_2 is a user-specified constant controlling how gradual the transition is (usually 1.5).

3) Flag all of the edge neighbors.

4) Repeat the process for all flagged nodes.

It should be noted that the quality constraints of Sec.II.D are obeyed during the preceding procedure.

D. Resulting Prismatic Meshes

Two test cases were chosen to illustrate the effectiveness, robustness, and generality of the ARM. The first was the HSCT aircraft with engines, shown in Fig. 2. A cross section of the wing–engine configuration shows how the ARM recedes nodes in the vicinity of the cavity region (Fig. 3). It is observed that there is a substantial difference in the total prismatic layer thickness of the nodes within the cavity compared with nodes away from it. The ratio between the thickest and the thinnest prismatic layers is 85:1. Even with such a drastic difference in thicknesses, no abrupt changes are observed in the prism cell sizes. Note that the structure of the prisms is not destroyed by the ARM. Thus the ARM has been successful in both, reducing the step size within the cavity, as well as providing a smooth transition to the unaffected area. The final prisms layer can now be used as the starting surface for the advancing front tetrahedral grid generator to fill the remaining volume.

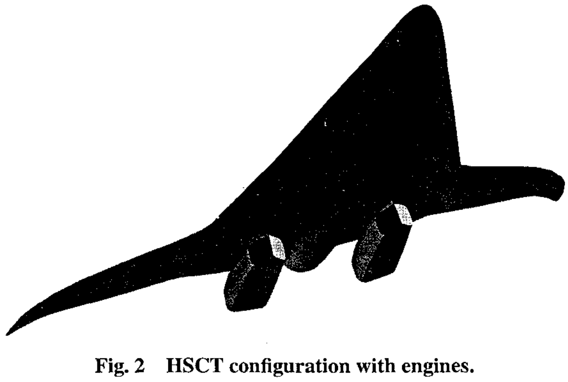

Fig. 2 HSCT configuration with engines.

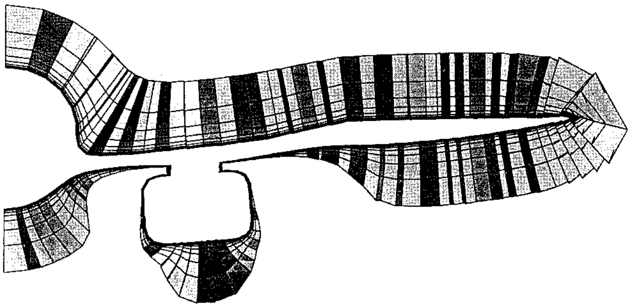

Fig. 3 Field cut of the HSCT along the fuselage showing the effect of the ARM. The automatic local reduction of the marching step size in the cavity prevents overlapping of the marching lines. A smooth transition is observed in the layer thickness from the cavity region to the area away from it.

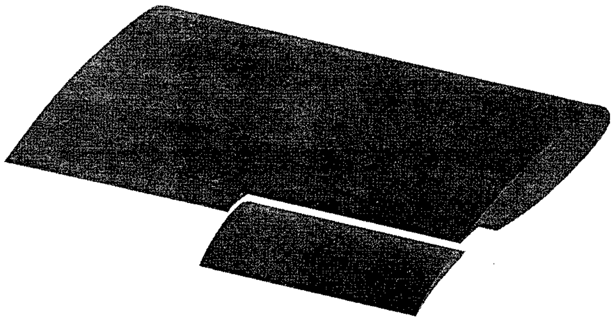

Fig. 4 View of the partial-flap high-lift wing. Note that there are actually two gaps: the gap between the wing and the flap and the much smaller gap inboard of the flap separating it from the unflapped portion of the wing.

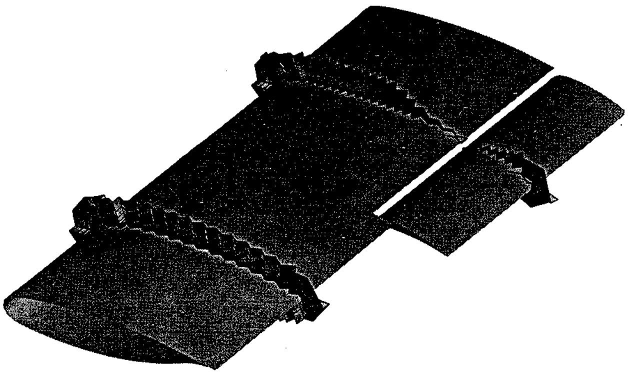

Fig. 5 View of the prism layers at two field cuts along the span of the high-lift wing. The first cut is taken at midspan of the unflapped region, and the second cut is taken at the midspan of the flapped region. The difference in prism layer thickness illustrates the effect of the ARM.

The second case used to test the robustness as well as the universality of the ARM is a partial-flap two-element wing configuration. The narrow gaps between the wing and the flap provide a good test for the method. Figure 4 shows a view of the wing surface. Note that there are actually two gaps for the ARM to treat: the gap between the wing and the flap and an even smaller gap inboard of the flap separating it from the unflapped portion of the wing.

A view of two field cuts along the span of the high-lift wing is shown in Fig. 5. A field cut is a surface consisting of all of the faces that intersect a given cutting plane. The cuts have been taken at midspan of both the unflapped and flapped regions. The irregularity of the cuts is due to the fact that the quadrilateral faces of the prisms are not coplanar with the cuts. In the first cut (the unflapped wing portion), it is observed that the prismatic layers have uniform thickness all around, since the ARM has not taken effect in this region. The effect of the ARM in the gap is clear in the second cut. A better view is shown in the planar cut in Fig. 6. The marching step size in the gap has been reduced to avoid overlap. Also, there are no abrupt changes in the prismatic layer thickness.

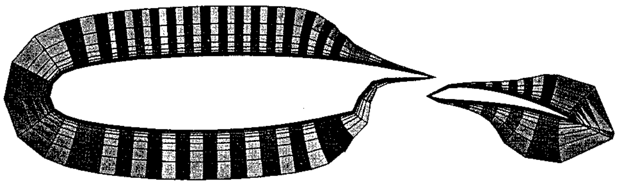

Fig. 6 Planar view of the prism layers at a field cut corresponding to the midspan of the flapped region of the high-lift wing. The ARM reduces the marching step size in the gap to avoid overlap. Note that there are no abrupt changes in the prismatic layer thickness.

It is important to note that the resulting single-block, nonoverlapping grids for the HSCT and the two-element wing were obtained without any user intervention. The ARM is automatic as well as blind to the specifics of the geometry.

IV. Octree/Advancing Front Tetrahedra Generation

A combined octree/advancing front method is used to generate the tetrahedra for the hybrid grid.[5,7,15] Advancing front type of methods require specification by the user of the distribution of three parameters over the entire domain to be gridded. These field functions are 1) the node spacing, 2) the grid stretching, and 3) the direction of the stretching. Using the octree/advancing front method, these parameters do not need to be specified. Instead, they are determined via an automatically generated octree. There is no need for a special background mesh that has been the backbone of previous advancing front generators.[1,2,12] Construction of this background mesh requires solution of the Poisson equation, which can be very expensive. The octree, on the other hand, is an efficient data structure to use to extract the mesh parameters.

Generation starts from the outermost surface of the layer of prisms surrounding the body. The triangular faces of this surface form the initial front list. A face from this list is chosen to start the tetrahedra generation. Then a list of points is created that consists of a new node, as well as of nearby existing points of the front. One of these points is chosen to connect to the vertices of the face. Following the choice of the point to connect to, a new tetrahedron is formed. The list of the faces, edges, and points of the front is updated by adding and/or removing elements. The algorithm followed in the present work is the one presented in Refs. 2 and 16. The method requires a data structure that allows for efficient addition/removal of faces, edges, and points, as well as for fast identification of faces and edges that intersect a certain region. The alternating digital tree (ADT) algorithm is employed for these tasks.[17]

The tetrahedra that are generated using this method grow in size as the front advances away from the original surface. Their size and the rate of increase of their size, as well as the direction of the increase, are all given from an octree consisting of cubes that is generated automatically via a divide-and-conquer method. This process generates octants that are progressively larger with distance away from the body. Their sizes determine the characteristic size of the tetrahedra that are generated in their vicinity. This method is flexible and can be used to generate tetrahedra around any type of geometry.

A. Special Octree for Tetrahedral Grid Spacing Control

The divide-and-conquer process starts with a master hexahedron that contains the body. This hexahedron is recursively subdivided into eight smaller hexahedra called octants. Any octant that intersects the body is a boundary octant and is subdivided further (inward refinement). The subdivision of a boundary octant ceases when its size matches the local thickness of the outermost prismatic layer.

Then the hexahedral grid is further refined in a balancing process (outward refinement) to prevent neighboring octants whose depth differs by more than one. Outward refinement is performed to ensure that the final octree varies smoothly in size away from the original surface. The sole criterion for outward refinement is a depth difference greater than one between the octant itself and any of its neighbors. The outward refinement continues until no octants

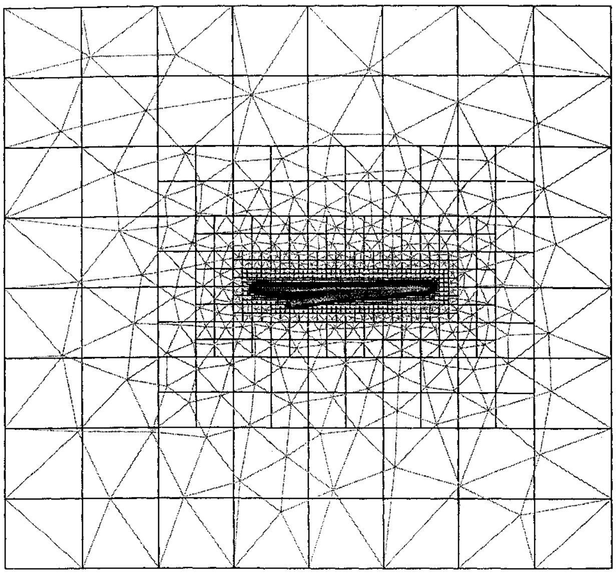

Fig. 7 Effect of the octree on growth of the tetrahedra for the HSCT geometry. View of the octants (quadrilateral faces), as well as of the tetrahedra (triangular faces) on the symmetry plane. Growth of the tetrahedra away from the outermost prisms surface follows growth of the octree quite faithfully.

meet the refinement criterion. Typically five sweeps are performed to produce a balanced octree.

B. Automatic Determination of Size of the Tetrahedra

The advancing front method creates a new tetrahedron by connecting each face of the current front to either a new or an existing node. This point is found by using a characteristic distance δ that is calculated from the size of the local octant to which the face of the front belongs. Specifically,

$$\delta = \alpha st^{(l_t - l)} \qquad (8)$$

where α is a scaling factor, st is the stretching parameter, l_t is the total number of octant levels, and l is the level of the local octant in the data structure. The value of st controls are rate of growth of the tetrahedra. The lower the value of st, the less the tetrahedra's increase in size away from the body. A typical value of the stretching parameter st is 1.8. The level l of the local octant is the number of subdivisions of the master octant required to get to the size of the local octant.

Smooth transition in size from the prisms to the tetrahedra is important for accuracy of the numerical method. The value of the scaling factor α is calculated so that the initial marching size δ of the tetrahedra equals the local thickness of the outermost prismatic layer. This feature allows smooth transition in size from the last layer of prisms into the tetrahedral region.

C. Universality of the Octree/Advancing Front Method

Figure 7 illustrates the symmetry plane of the HSCT configuration without the engine. The quadrilaterals (dark lines) correspond to the faces of the octants on this plane, whereas the triangles (light lines) correspond to the faces of the tetrahedra. It is observed that the size of the tetrahedra, as well as the stretching of the mesh and the direction of stretching, is guided quite accurately by the octree.

Figure 8 shows a plane cut of the octree around the engine of the HSCT. The lower portion of the figure shows an enlarged view of the engine area. The octants in the cavity between the wing and the engine are much smaller, and they match the local thickness of the outer prismatic layer. The prisms are constrained in that region by the ARM to avoid overlapping prismatic layers. Note that only every third layer for the prismatic mesh is shown for clarity of the figure.

Figure 9 shows a plane cut of the octree around the two-element wing configuration taken at midspan of the flapped region. The lower portion of the figure shows a closeup of the flap area. The octants in the gap between the flap and the main airfoil are smaller than those out of the gap and match the local thickness of the outer prismatic layer accurately. Note that only every third layer of the prismatic mesh is shown for clarity of the figure.

It is important to note that the octree for each of these cases was generated with minimal user interaction. The same octree/advancing front method automatically generated the octree mesh and then the tetrahedral mesh for each of the three geometries.

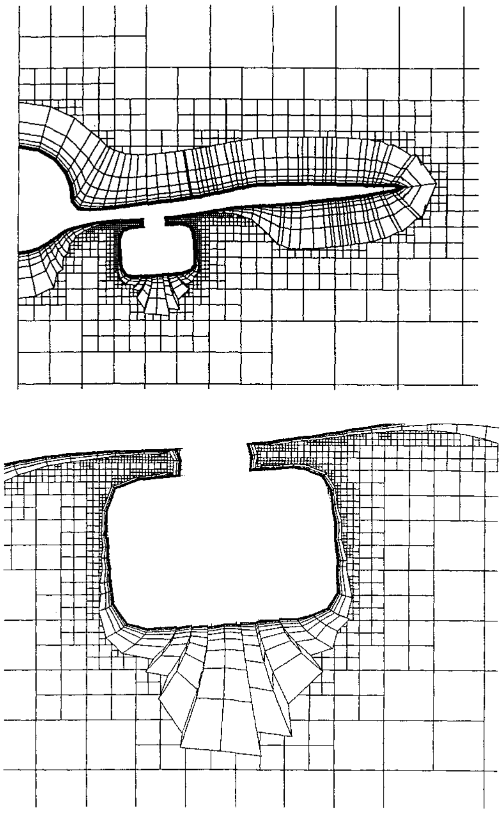

Fig. 8 Plane cut of an octree mesh for the HSCT. The lower portion shows an enlarged view of the engine area. Note that only every third layer of the prismatic mesh is shown for clarity of the figure.

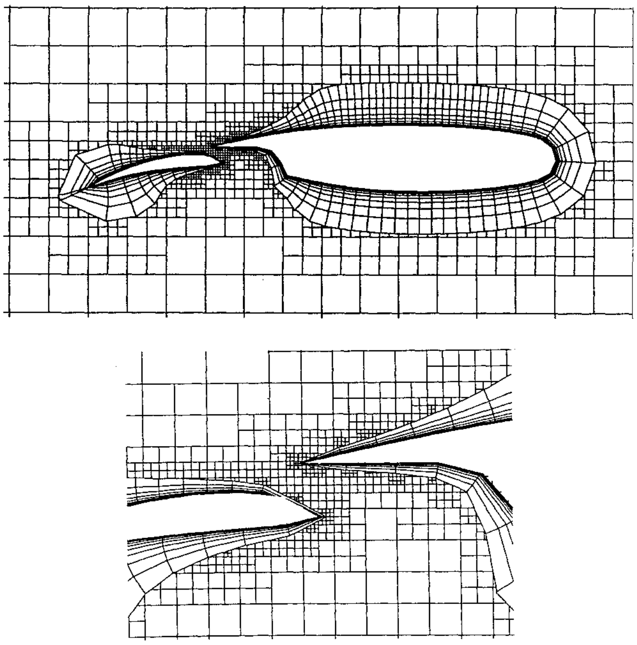

Fig. 9 Plane cut of an octree mesh for the two-element wing at midspan of the flapped region. The lower portion shows an enlarged view of the gap region.

V. Universality of Hybrid Grids Application

Hybrid grids were generated for a variety of geometries to demonstrate the effectiveness of the developed grid generator. The geometries presented here include HSCT aircraft configurations with and without engines, as well as a partial-flap two-element wing. All surface geometries were generated via the advancing front method. A viscous flow simulation is presented over the HSCT configuration without engines to demonstrate the validity of the hybrid grids. The finite volume scheme used for the numerical simulations uses central space differencing and Lax–Wendroff time marching. Scheme operations are cast in edge-based form. The details regarding the Navier–Stokes solver are presented in Ref. 10.

A. Hybrid Grid for the HSCT Aircraft Configuration Without Engines

The final hybrid grid for the HSCT without engines consists of 176,480 prisms and 171,483 tetrahedra. Generation of the prisms took about 90 s on an IBM 390 workstation, whereas generation of the tetrahedra took about 45 min on the same machine. Figure 10 illustrates the hybrid mesh on two different planes that are perpendicular to each other. The symmetry plane shows the signature of the prisms and tetrahedra in two dimensions. The quadrilaterals close to the surface represent the prisms, whereas the triangles away from the surface correspond to the tetrahedra. The second plane is a field cut intersecting the fuselage. The field cut clearly brings out the structured nature of the prisms and the unstructured tesselation of the tetrahedra. The good orthogonality and clustering properties of the prisms are clear in the figure. A smooth cell size variation is observed in the transition from prisms to tetrahedra. The quadrilateral signature of the prisms also indicates that the thickness of the prism cells is enough to capture the boundary layer.

1. Significant Reduction in Required Memory Storage

The advantage of the hybrid grid over an all tetrahedral mesh can be demonstrated by the savings in computer resources. Because of the two-dimensional nature of the pointers required to store the prismatic cell information and the small number of tetrahedral cells required, the total memory required by the solver for the HSCT simulation was only 18 Mwords. If, on the other hand, an all tetrahedral mesh of the same resolution was used to discretize the flow domain, the total number of the cells required would be on the order of 750,000. Furthermore, indirect addressing would be required for all of these cells. This would lead to a total memory requirement for the solver of approximately 30 Mwords.

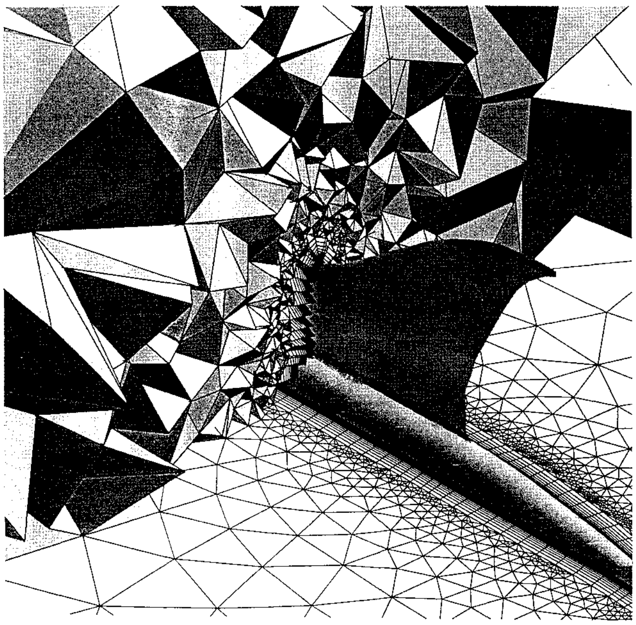

Fig. 10 View of the hybrid mesh around the HSCT (without engines) on two different planes that are perpendicular to each other. The first plane is that of the symmetry, whereas the second is a field cut intersecting the fuselage.

In short, the substantial reduction in memory storage is achieved due to the following factors: 1) the significantly reduced number of elements required to cover the viscous region using prisms rather than tetrahedra, 2) the semistructured nature of the prisms that require only two-dimensional data structures, and 3) the reduced number of tetrahedral cells employed over the domain. It is these savings in computer resources that allowed viscous flow simulations around aircraft configurations to be performed on workstations rather than supercomputers.

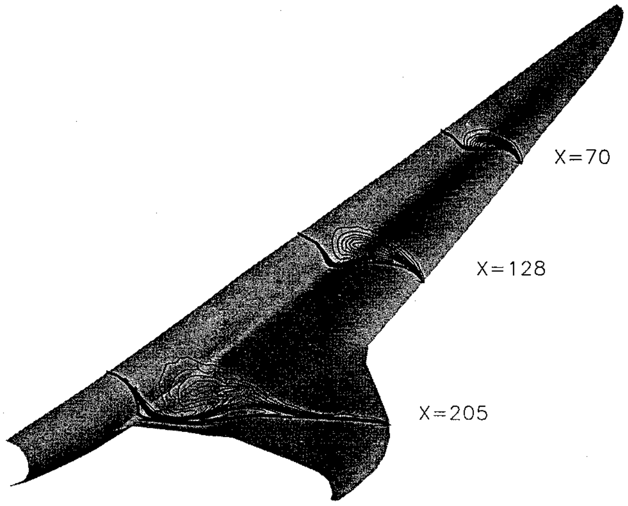

Fig. 11 Isometric view of the HSCT showing entropy contours at x = 70, 128, and 205. The conical growth of the primary vortex and the recirculation region between the wing and fuselage are clearly visible. Case of supersonic flow (M_∞ = 3.0, Re = 10^6, and α = 5.0 deg) around an HSCT configuration.

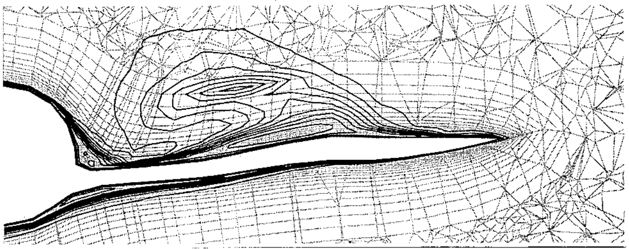

Fig. 12 Plane cut along the fuselage showing the hybrid mesh as well as the corresponding entropy contours. The hybrid grid interface has no adverse effects on the accuracy of the solution. Case of supersonic HSCT flow (M_∞ = 3.0, Re = 10^6, and α = 5.0 deg) around the HSCT configuration.

2. Suitability for Capturing Complex Flow Phenomena

Supersonic flow at a Mach number of 3.0 and a Reynolds number of 10^6 is considered. Entropy contours and velocity vector plots are obtained to investigate vortex formation. The suitability of hybrid grids to capture complex flow physics is demonstrated, and it is shown that the presence of the prism/tetrahedra interface does not affect the solutions.

Figure 11 shows entropy contours at three different planes along the fuselage, namely $x = 70$, 128, and 205. These cuts show the progression of the primary vortex that forms at the nose. It is observed that the vortex grows conically as the flow traversed past the aircraft. A recirculation region is observed at the wing/fuselage junction on the plane of the cut at $x = 205$.

A view of the entropy contours along with the hybrid mesh for the HSCT is shown in Fig. 12. The cut has been taken along the axis perpendicular to the fuselage ($x = 180$). Note that the irregularity of the mesh is due to the fact that the grid is not coplanar with the cut. The recirculation region is clearly visible at the junction between the wing and the fuselage. Another separation area is observed slightly outboard of the primary vortex. The clustering of the prismatic grid

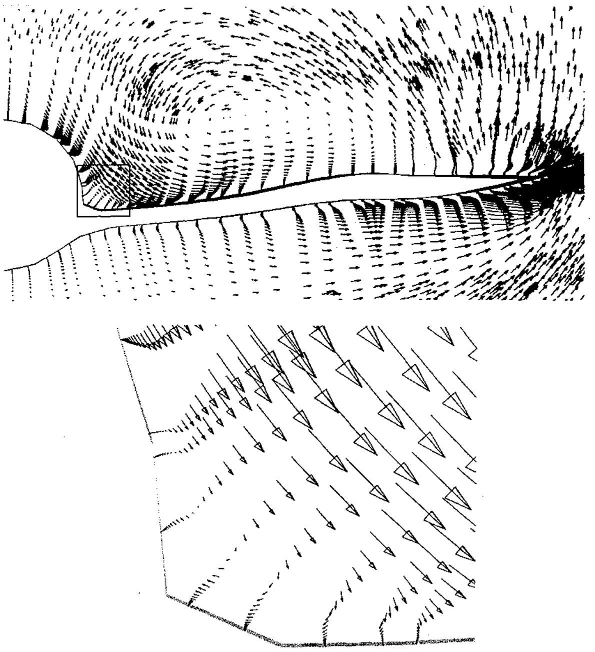

Fig. 13 Predicted crossflow velocity vectors at a cut along the fuselage (x = 180). The lower portion shows an enlarged view of the wing–fuselage junction where a small recirculation region exists. Case of supersonic flow (M_∞ = 3.0, Re = 10^6, and α = 5.0 deg) around the HSCT configuration.

close to the wall thus seems to be very effective in capturing small-scale physics. Furthermore, a qualitative assessment of the contour lines indicates that the numerical solution is not affected by the transition from prisms to tetrahedra since there are no abrupt changes in the contours at the interface.

Figure 13 shows a velocity vector plot for the same location (x = 180) at which the previous entropy contours were shown. Note that these vectors represent projections on the plane of the cut. It is seen that the vortex that forms at the nose of the delta-shaped wing of the HSCT is captured quite well. An enlarged view of the area of the junction between the wing and the fuselage is shown in the lower portion of the figure. The small-scale recirculation region is clearly visible here.

B. Hybrid Grid for the Two-Element Wing Configuration

A hybrid grid is also generated for the two-element wing configuration shown in Fig. 4. The surface geometry consists of 4144 nodes and 8226 triangular faces. The final hybrid grid for the wing consists of 18 layers of prisms (a total of 148,000 prism cells), and 276,000 tetrahedral cells.

A field cut of the hybrid mesh at midspan of the flapped portion of the wing is shown in Fig. 14. This figure illustrates the effect of the ARM on the gap. As can be seen, the prisms have been receded to avoid overlap. The small prism step size in the gap region results in a dense tetrahedral mesh in this area. It should be noted that even though the prism layer thickness varies along the wing and the flap, the transition of thickness between prisms and tetrahedra is still smooth. A better view of the gap between the wing and the flap is shown in Fig. 15. The thickness of the prism cells gradually increases with distance from the gap, and so does the tetrahedral cell size.

C. Hybrid Grid for the HSCT Configuration with Engines

The final case that is considered to test the robustness and universality of the developed hybrid grid generator is the HSCT configuration with engines. Figure 2 shows a view of the initial surface. The surface was composed of 10,578 nodes and 20,784 triangular faces. The hybrid grid consisted of 39 layers of prisms (810,576 prismatic cells), and 487,854 tetrahedral cells.

An isometric view of the hybrid grid is shown in Fig. 16. The figure shows the mesh on two planes that are perpendicular to each other. The first plane is that of the symmetry, which shows the quadrilateral signature of the prisms along with the triangular signature of the tetrahedra. The second plane is a field cut along the fuselage that illustrates the three-dimensional hybrid nature of the grid. The structure of the prisms and the smooth transition from prisms to tetrahedra are worth observing. The field cut in this view has been taken through the engine to show the effect of the ARM. It can be seen that the mesh is much more dense in the engine cavity than elsewhere.

Figure 17 is a field cut along the fuselage. It shows a close-up of the cavity region between the wing and the engine. This view better illustrates the cavity region and the smooth variation in prismatic cell thickness away from the cavities. It is seen that the tetrahedra match the local prism cell sizes relatively well even in the cavity.

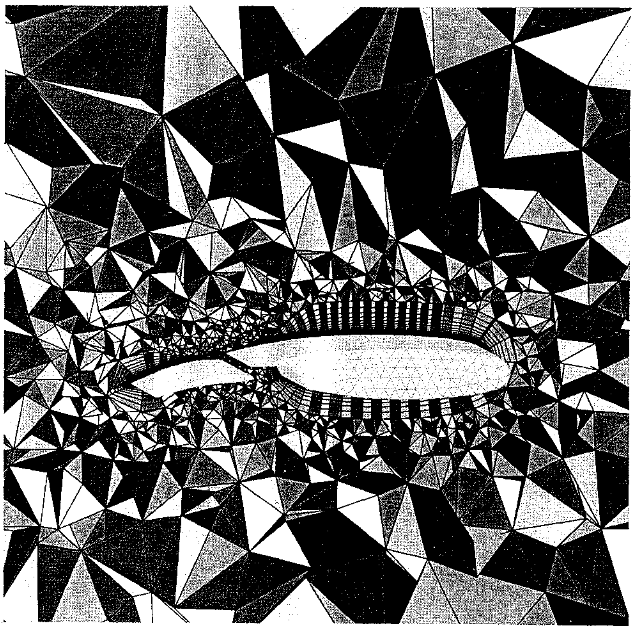

Fig. 14 Field cut of the hybrid grid for the two-element wing configuration at midspan of the flapped portion. The ARM has receded the prisms in the gap between the wing and the flap, and the tetrahedra fill the remaining region.

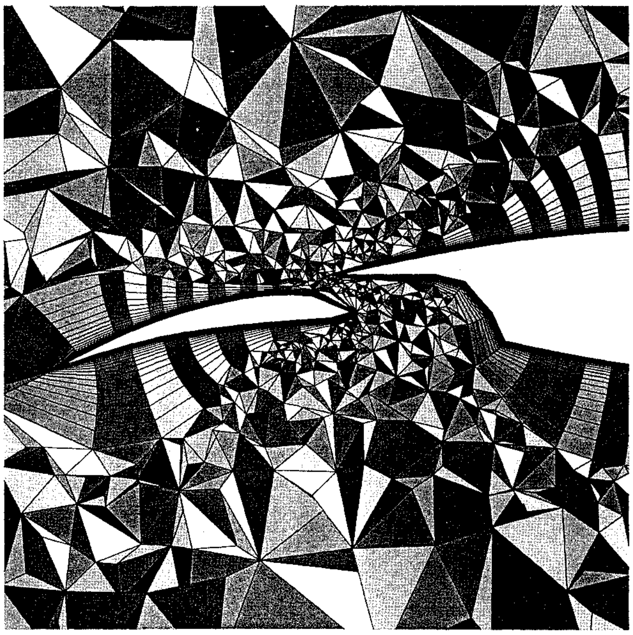

Fig. 15 Close-up of the hybrid grid in the region of the gap between the wing and the flap of the two-element wing geometry. The size of the tetrahedra in the gap is smaller than that away from the gap so as to match the local prism cell size.

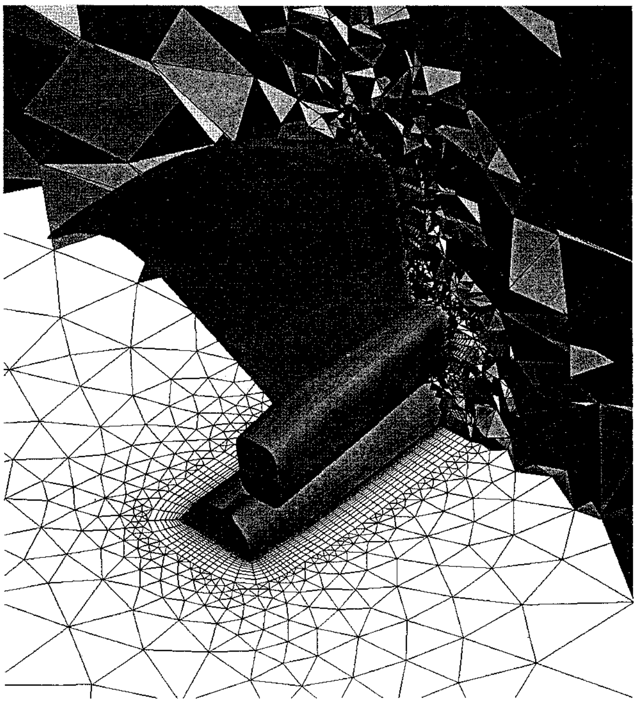

Fig. 16 View of the hybrid mesh around the HSCT with engines on two different planes that are perpendicular to each other. The first plane is that of the symmetry, whereas the second is a field cut intersecting the fuselage and engine.

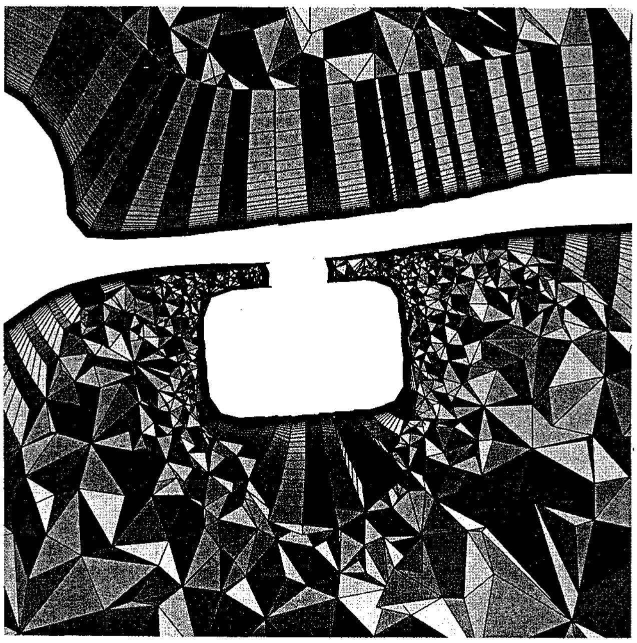

Fig. 17 Close-up of the hybrid grid for the HSCT around the engine cavity. The tetrahedral mesh is very dense here compared with other regions so as to match the thin local prism cell sizes.

VI. Concluding Remarks

An algorithm for generating hybrid prismatic/tetrahedral meshes for arbitrary three-dimensional geometries was developed. The prisms covered the regions close to the body surface, while tetrahedra were created elsewhere. The marching vectors procedure to generate the prisms proved to be robust (in avoiding overlapping of prism layers) and efficient. The smoothing operations and the imposition of constraints eliminated surface ripples and avoided excessively stretched and skewed meshes.

The ARM allowed the generation of a single-block, nonoverlapping prismatic mesh even when the surface geometry contained narrow gaps and cavities. This was demonstrated on configurations such as the HSCT with engines, as well as a two-element wing. The ARM allowed the structure of the prisms to be retained, thus reducing memory requirements on the solver. Also, it was blind to specifics of the geometry.

The generation of tetrahedra was made simpler because a user-dependent background mesh was not required for determination of mesh spacings. The spacing parameters were computed using an automatic octree-based procedure to provide a smooth transition in thickness from prisms to tetrahedra. The octree was universally effective on different geometries without any user intervention.

The hybrid grid generation procedure was automated so that no user expertise was required. The resulting grids were of high quality and were suitable for viscous flow simulations. The validity of the grid was demonstrated by performing supersonic flow simulations over an HSCT aircraft configuration. The structured nature of the prisms allowed significant savings in computer memory required by the solver. This allowed for flow simulations to be performed on workstations rather than supercomputers.

Acknowledgments

This work was supported by NASA Grant NAG1-1459, National Science Foundation Grant ASC-9357677 (NYI Program), and Texas Advanced Technology Program Grant 003658-413. Computing time was provided by the Numerical Aerodynamic Simulation Division of NASA Ames Research Center, as well as by the High Performance Computing Facility of the University of Texas.

References

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

[2] Peraire, J., Morgan, K., and Peiro, J., "Unstructured Finite Element Mesh Generation and Adaptive Procedures for CFD," *Application of Mesh Generation to Complex 3-D Configurations*, AGARD CP No. 464, 1990, pp. 18.1–18.12.

[3] Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W., *Numerical Grid Generation: Foundations and Applications*, Elsevier, New York, 1985.

[4] Yerry, M. A., and Shephard, M., "Automatic 3-D Mesh Generation by the Modified Octree Technique," *International Journal for Numerical Methods in Engineering*, Vol. 20, July 1984, pp. 1965–1990.

[5] Kallinderis, Y., Khawaja, A., and McMorris, H., "Hybrid Prismatic Tetrahedral Grid Generation for Complex Geometries," AIAA Paper 95-0211, Jan. 1995.

[6] Kallinderis, Y., and Ward, S., "Prismatic Grid Generation for 3-D Complex Geometries," *AIAA Journal*, Vol. 31, No. 10, 1993, pp. 1850–1856.

[7] Khawaja, A., McMorris, H., and Kallinderis, Y., "Hybrid Grids for Viscous Flows Around Complex 3-D Geometries Including Multiple Bodies," AIAA Paper 95-1685, June 1995.

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

[9] Parthasarathy, V., and Kallinderis, Y., "Directional Viscous Multigrid Method on Adaptive Prismatic Meshes," *AIAA Journal*, Vol. 33, No. 1, 1995, pp. 69–78.

[10] Parthasarathy, V., Kallinderis, Y., and Nakajima, K., "A Hybrid Adaptation Method and Directional Viscous Multigrid with Prismatic/Tetrahedral Meshes," AIAA Paper 95-0670, Jan. 1995.

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

[12] Pirzadeh, S., "Recent Progress in Unstructured Grid Generation," AIAA Paper 92-0445, Jan. 1992.

[13] Pirzadeh, S., "Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method," AIAA Paper 94-0417, Jan. 1994.

[14] Hassan, O., Probert, E. J., Morgan, K., and Peraire, J., "Unstructured Mesh Generation for Viscous High Speed Flows," *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Proceedings of the 4th International Conf., Pineridge, Swansea, Wales, UK, 1994, pp. 779–793.

[15] McMorris, H., and Kallinderis, Y., "Octree Advancing Front and Automatic Partial Remeshing for Tetrahedra Generation," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference* (San Diego, CA), AIAA, Washington, DC, 1995, pp. 1172, 1173 (AIAA Paper 95-1762).

[16] Peiro, J., Peraire, J., and Morgan, K., "FELISA System Reference Manual Parts 1 and 2," Univ. of Wales, Swansea Repts., CR/821/94 and CR/822/94, Swansea, Wales, UK, Aug. 1994.

[17] Bonet, J., and Peraire, J., "An Alternating Digital Tree (ADT) Algorithm for Geometric Searching and Intersection Problems," *International Journal for Numerical Methods in Engineering*, Vol. 31, No. 1, 1990, pp. 11–19.