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

Unstructured Viscous Layer Insertion Using Linear-Elastic Smoothing

Steve L. Karman Jr.*
University of Tennessee at Chattanooga, Chattanooga, Tennessee 37403

DOI: 10.2514/1.23283

A robust scheme for generating unstructured viscous meshes, using elliptic partial differential smoothing, is presented. Linear-elasticity relationships, borrowed from the structural mechanics discipline, provide the governing partial differential equations for smoothly displacing the existing mesh, making room for new viscous layers. These viscous layers are allowed to advance up adjacent, nonviscous boundaries. A process of inserting layers on alternating boundary groupings enables the creation of high-quality hexahedral elements in juncture regions. Viscous layer height is allowed to vary, depending on the size of the local inviscid mesh. Three-dimensional results are included to illustrate the use of the method to produce high-quality viscous meshes.

Nomenclature

A	=	Jacobian matrix
E	=	Young's modulus of elasticity
f	=	geometric progression factor
g	=	geometric growth rate
$\boldsymbol{n}$	=	normal vector
$\hat{\boldsymbol{n}}$	=	unit normal vector
u, v, w	=	physical perturbation components
x, y, z	=	physical coordinates
α	=	linear-elasticity coefficient, facet angle
β, θ	=	linear-elasticity coefficients
ν	=	Poisson's ratio

I. Introduction

UNSTRUCTURED viscous mesh generation can be challenging for complicated geometries. The goal is to generate unstructured meshes that adequately resolve the viscous region of the flowfield and that exhibit certain quality characteristics. Among these characteristics are a specified normal spacing at the wall, nearly orthogonal mesh edges emanating from the wall, a smoothly increasing normal spacing away from the surface, and a smooth mesh spacing transition to the inviscid portion of the mesh. Achieving some or all these characteristics can be difficult, especially when sharp edges and corners are involved.

Several approaches are currently followed for generating unstructured viscous meshes. One of the more common approaches involves generating the viscous layers of the mesh first, using an extrusion technique, and then discretizing the remaining portion of the domain, using a volume mesh generation technique, such as a Delaunay-based tetrahedral meshing algorithm or advancing layers method [1–3]. The mesh extrusion technique marches away from the boundary in the normal direction, creating prismatic or hexahedral elements from surface triangles or quadrilaterals, respectively. Each marching step adds one layer at a time on top of the existing layers, as the viscous mesh is created. This process requires valid normal vectors at each point of the existing top layer in order to progress to the next layer. Computing these normal vectors can become difficult at sharp edges and in concave corners of the geometry. Care must be taken to avoid collisions of the advancing layer with itself and other advancing fronts. These normal vectors must be recomputed at each new layer and often involve some form of averaging to ensure a smooth transition from layer to layer. The normal spacing is usually specified at the surface and gradually increased from layer to layer following a geometric progression. The spacing of the top layer should approach the element size of the volume mesh elements. Then a tetrahedral volume-meshing algorithm is typically employed to discretize the rest of the domain. This requires the top layer of hexahedral elements to be capped with pyramid elements to produce an all-triangular surface mesh for the tetrahedral meshing routine. Adding these pyramid elements can also introduce complications in concave corners of the domain at which the apex of the pyramids could collide.

Another approach to viscous unstructured meshing constructs the volume mesh by first using a general tetrahedral meshing algorithm and then introducing new grid points at carefully selected locations that result in high aspect ratio tetrahedral elements in the viscous region. Marcum [4] generates the initial volume mesh by tessellating the boundary points only and recovers all boundary edges. Then interior points are added using an advancing front of points that are carefully located to generate near isotropic elements for inviscid meshes or high aspect ratio elements for viscous meshes. The new points are inserted one layer at a time, based on desired spacing criteria, desired growth rate, and local normal vectors of the advancing front. The shape of the domain (specifically, sharp edges and concave corners) affects the choice of the location for the new grid points. New points are rejected that are too close to other new points. As each layer is inserted, a local reconnection is performed to include the new grid points into the mesh. This reconnection process combines the Delaunay criterion with a min–max angle criterion to ensure high-quality elements. A valid tessellation is maintained as the new layers of points are added. The highly packed tetrahedra near the surface can be recombined to form prismatic elements.

Lohner and Cebral [5] published a technique very similar to Marcum's [4]. Lohner's approach also enforced a constrained Delaunay criterion as viscous layer points were introduced into the existing mesh, and so care was taken to ensure the new points resulted in valid, viscous elements in the viscous region. Included in their approach was special treatment of sharp convex corners, using multiple normal vectors, and concave corners, resolving bidirectionally into the corner.

Ito and Nakahashi [6] published a technique whereby the existing inviscid mesh is perturbed away from the surface to make room for new viscous layers. This approach did not enforce the Delaunay criterion as layers were inserted. Instead, valid normal vectors, computed for each viscous surface node, were used to perturb the

Presented as Paper 0531 at the 44th Aerospace Sciences Meeting & Exhibit, Reno, NV, 9–12 January 2006; received 17 February 2006; revision received 1 August 2006; accepted for publication 14 October 2006.

*Research Professor, Graduate School of Computational Engineering. Associate Fellow AIAA.

mesh before creating the new layer. Sharp edge nodes could make use of two normal vectors. Displacements along these normal vectors were restricted to ensure that valid, positive elements were produced. The insertion process could be stopped based on criteria that compared the height of the layers with the connecting tetrahedral mesh.

The current approach, developed independently from [6], generates unstructured viscous meshes by inserting layers of elements into an existing inviscid mesh. To make room for each new layer of elements, the existing mesh is pushed away from the surface, using linear-elasticity relations that are borrowed from the structural mechanics community. These relationships have been used in moving mesh problems with great success [7,8]. This paper, based on a presentation in January 2006, describes a specific use of this mesh-moving technique to generate high-quality viscous meshes [9]. The layer insertion proceeds in reverse order: top layer first, bottom layer last. Multiple normal vectors are possible and the number of layers per column can vary, depending on the size of the local inviscid mesh elements. The current approach does not enforce the Delaunay criterion; instead, the element quality is maintained via the stiffness parameters defined for the linear-elastic relations. The particular linear-elastic formulation will be described. Details of the insertion technique will be provided. Several three-dimensional examples will be shown to demonstrate the generality of the approach.

II. Unstructured Linear-Elastic Mesh Smoothing

Two forms of elliptic smoothing for unstructured meshes were presented in [10,11]. One form used the traditional Winslow equations, but for unstructured meshes. The other form of elliptic smoothing used linear-elasticity equations for performing mesh deformation. The linear-elastic relations provide a robust approach for mesh movement, while maintaining high-quality meshes, especially in the boundary-layer region. The present paper describes details of an insertion technique that uses the linear-elastic smoothing to create viscous meshes.

A. Linear-Elastic Formulation

Linear-elastic theory smoothing is generally used for mesh movement in which boundaries of the mesh are deforming and interior mesh points need to be adjusted to maintain a valid mesh. It is assumed that the mesh obeys the two-dimensional isotropic linear-elastic relations shown next.

$$\nabla^2 u + \frac{1}{1-2\nu}\frac{\partial}{\partial x}\nabla\cdot\boldsymbol{V} = 0 \qquad \nabla^2 v + \frac{1}{1-2\nu}\frac{\partial}{\partial y}\nabla\cdot\boldsymbol{V} = 0 \quad (1)$$

where the nodal displacement vector is given by $\boldsymbol{V} = u\hat{i} + v\hat{j}$. The parameter ν in the denominator is typically manipulated so that the coefficient $1/(1\text{-}2\nu)$ is equal to the aspect ratio of the local cell. This produces stiffness in regions with high aspect ratio cells and ensures that boundary-layer elements track closely with the local boundary as it moves. The solution to these equations is a vector field defining the displacement of each node. The terms in Eqs. (1) can be expanded and cast in the following form:

$$\begin{aligned}\left(1+\frac{1}{1-2\nu}\right)\frac{\partial^2 u}{\partial x^2} + \left(\frac{1}{1-2\nu}\right)\frac{\partial^2 v}{\partial x\partial y} + \frac{\partial^2 u}{\partial y^2} = 0\\ \frac{\partial^2 v}{\partial x^2} + \left(\frac{1}{1-2\nu}\right)\frac{\partial^2 u}{\partial x\partial y} + \left(1+\frac{1}{1-2\nu}\right)\frac{\partial^2 v}{\partial y^2} = 0\end{aligned} \quad (2)$$

Yang and Mavriplis [8] showed how linear-elastic smoothing can be used to perform very large deformations of inviscid and viscous meshes. They chose to use a different form of the linear-elastic relations, for which the modulus of elasticity was allowed to vary and Poisson's ratio was constant. Under these assumptions, their formulation can be transformed to a form similar to Eqs. (2).

$$\begin{aligned}&\frac{\partial}{\partial x}\left[\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial y}\left[\frac{E}{2(1+\nu)}\frac{\partial u}{\partial y}\right]\\ &\quad + \frac{\partial}{\partial x}\left[\frac{E\nu}{(1+\nu)(1-2\nu)}\frac{\partial v}{\partial y}\right] + \frac{\partial}{\partial y}\left[\frac{E}{2(1+\nu)}\frac{\partial v}{\partial x}\right] = 0\\ &\frac{\partial}{\partial x}\left[\frac{E}{2(1+\nu)}\frac{\partial v}{\partial x}\right] + \frac{\partial}{\partial y}\left[\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\frac{\partial v}{\partial y}\right]\\ &\quad + \frac{\partial}{\partial x}\left[\frac{E}{2(1+\nu)}\frac{\partial u}{\partial y}\right] + \frac{\partial}{\partial y}\left[\frac{E\nu}{(1+\nu)(1-2\nu)}\frac{\partial u}{\partial x}\right] = 0\end{aligned} \quad (3)$$

Yang and Mavriplis [8] chose to set E to the inverse of the element volume or to the inverse of the distance from the deforming boundaries. This formulation has been tested by the present author, with E prescribed as the inverse of the cell volume, and is indeed more robust for large deformations than the formulation in Eqs. (1) and (2). Poisson's ratio for Eq. (3) must be specified within the valid range from 0 to 1/2. In structural mechanics, for metals in the elastic range, the value of Poisson's ratio lies between 1/4 and 1/3 [12]. Poisson's ratio is defined as the ratio of lateral strain to axial strain and is always positive in sign. The behavior of the smoothing for varying values of ν was tested on a simple deformation case. An initial square domain with grid lines clustered to the bottom boundary is shown in the upper left of Fig. 1. The bottom boundary is perturbed, resulting in a spike protruding into the middle of the domain. The formulation from Eqs. (2) produced the mesh in the upper right of the figure. The formulation from Eqs. (3) produced the two meshes in the bottom of the figure. The difference in the bottom two images shows that varying the global value of Poisson's ratio affects the degree of penetration of the perturbation into the domain. Smaller values of ν allow larger penetrations, and larger values of ν minimize the penetration.

Further investigation has revealed that setting E to the inverse of the volume or the inverse of the distance can have detrimental effects in tight corner regions in which elements are naturally small. Corners are regions in which element flexibility could be advantageous. An alternative is to set E equal to the aspect ratio of the element. This would provide the rigidity needed in boundary-layer elements, in which the aspect ratios are high, and the flexibility desired in the far field, in which the aspect ratios are generally small. Low aspect ratio, small elements in corners are then allowed to flex as necessary.

Another option is to set E equal to the element condition number. Knupp [13,14] argues that the condition number is similar to the aspect ratio, but uses three lengths and face areas (the aspect ratio uses six lengths and no face areas). Mathematically, the inverse of the condition number is a measure of the distance of the matrix from the set of singular matrices. Therefore, the condition number, computed using the following equations for a tetrahedron, varies from one for an ideal tetrahedron to large values for compressed tetrahedra. The Jacobian matrix A is constructed using the components of the three edge vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$, emanating from the corner of the tetrahedron. Boundary-layer elements will have a large value for the condition number and therefore have a large value for Young's modulus. Small elements in corners could conceivably have low condition numbers and low stiffness.

$$CN = \frac{\|A\|\|A^{-1}\|}{3} \qquad A = \begin{bmatrix} e1_x & e2_x & e3_x \\ e1_y & e2_y & e3_y \\ e1_z & e2_z & e3_z \end{bmatrix} \quad (4)$$
$$\|A\|^2 = \mathrm{trace}(A^T A)$$

The construction of the discrete linear-elastic system of equations requires the value of Young's modulus for each elemental control volume surrounding a node. When the element is not a tetrahedron, the portion of the element that contributes to the node control volume is the tetrahedron formed by the corner node and the three edges emanating from the corner. So the tetrahedral condition number can be used for Young's modulus for all element types: tetrahedra, pyramids, prisms, and hexahedra.

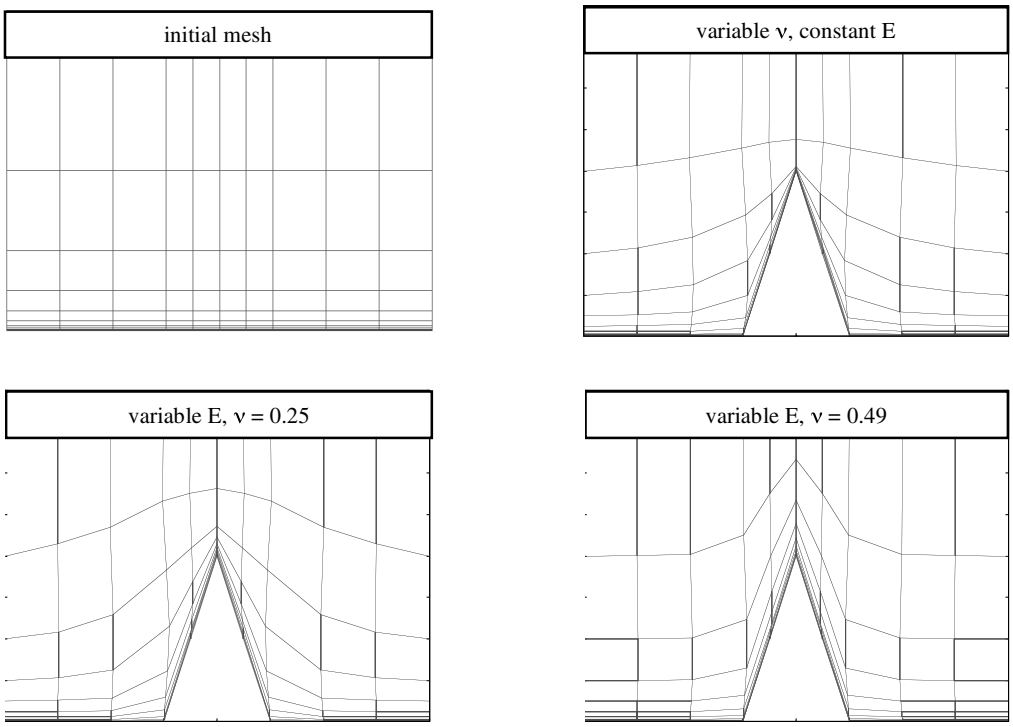

Fig. 1 Spike deformation using variable Poisson's ratio and variable Young's modulus.

Both the element aspect ratio and the element condition number exhibit a barrier against inverting elements, which translates into high stiffness in the linear-elastic relations. Either can be used to define the modulus of elasticity. The element condition number was selected for E and Poisson's ratio was set to 0.25–0.35 for the cases presented in this paper.

B. GMRES Solver

The algorithm described in [10,11] used a point-implicit Gauss–Seidel scheme with subiterations to solve for the nodal displacements. During each subiteration, the most recent information from the neighboring nodes was used to compute the displacement of the local node. With sufficient subiterations, the nonlinear system would converge to the final displacements for the entire field. This technique proved to be unstable as the spacing between layers decreased and required smaller values of the under-relaxation factor in order to compute a solution. This stiffness was overcome by using a generalized minimum residual (GMRES) solver instead. The implementation of the GMRES solver allows the user to specify the number of search directions, which indirectly controls the memory requirements of the system. The GMRES solver also has a restart capability, and so subiterations could be used to converge the global system, with GMRES restarts used at the beginning of each subiteration. The combination resulted in an efficient approach for computing the solution of the nonlinear system, in two and three dimensions.

C. Sidewall Boundaries

The boundary conditions for linear-elastic smoothing include three types: moving boundaries for which the perturbation vectors are specified, fixed boundaries for which the perturbation vectors are zero, and adjacent or sidewall boundaries for which the perturbation vectors are allowed to float freely, tangential to the surface. The shape of all boundaries is provided to the viscous insertion program as a collection of triangulated surface patches. This triangulation could be extracted from the initial inviscid mesh. Alternately, a coarser or finer resolution triangulation could be provided to the program, if desired.

The user identifies moving boundaries and the perturbation vectors are prescribed. For viscous layer insertion, these perturbation vectors are computed with the algorithm described later in this paper. For regular moving mesh problems, the perturbation vectors would be defined according to the motion of the boundary. The same linear-elastic smoother is used for mesh moving problems and viscous layer insertion problems.

Knowing which boundaries are moving boundaries, the solver can identify the sidewall boundaries that will float and the nonadjacent boundaries that remain fixed. The sidewall boundaries need not be planar surfaces. The solver knows the shape of all boundaries; therefore, the solver can perform projections of points to the surface and rotations of vectors on the surface. Because the points on the sidewall boundaries are allowed to float, the perturbation vectors on these boundaries are computed along with the interior points. Implicit boundary conditions loosely impose a tangential perturbation vector for these sidewall boundary points in the system of equations. After each iteration of the solver, the updated perturbation vector for sidewall boundary points is rotated to be tangent with the local boundary. After the iterations are completed, the sidewall boundary points are moved along with the interior points and then projected onto the local surface geometry.

III. Robust Generation of Viscous Meshes

The linear-elastic smoother can be used to prepare for the insertion of viscous layers near the surface of an existing mesh. The approach is to insert new layers of points at the boundary surface after pushing the existing mesh away from the boundary, using linear-elastic smoothing. As the existing mesh is pushed away from the boundary, the linear-elastic smoothing maintains good element shapes and prevents inverted elements. Details of the major steps required for this approach are provided in the following sections.

A. Surface Normal Vectors

The insertion procedure requires valid normal vectors at each viscous surface grid point. Most surface grid points will use a single normal vector. Some surface grid points, such as corners and sharp trailing edge points, may benefit from using multiple normal vectors. Details of computing single normal vectors in two and three dimensions and multiple normal vectors for two dimensions are described next.

The single normal vector procedure is not unlike one that would be used in typical viscous extrusion methods. The difference is in the application of the procedure. Traditional extrusion methods must compute valid normal vectors at each new viscous layer before extruding another layer. The top extruded layer could become very irregular and wrinkly unless care is taken to control the quality, using some kind of averaging of the normal vectors. The present method always inserts layers at the boundary, the surface of which is assumed smooth, avoiding the requirement of computing normal vectors on a wrinkled surface away from the boundary.

1. Single Normal Vectors

A normal vector at a 3-D surface node can be attached to an arbitrary number of triangular and/or quadrilateral faces. A simple averaging of the normal vectors from each face that shares the surface node may result in a valid normal vector, but it may be biased in the direction of an unbalanced distribution of faces on one side of the node. If, however, the averaging is weighted by the angle subtended at the node for each face, then the normal vector will tend not to be biased to one side or the other side. The angle-weighted averaging for a surface node with nf surrounding faces is given in Eq. (5), in which the angle subtended from the node is given by α.

$$\boldsymbol{n} = \frac{\sum_{f=1}^{nf} \alpha_f \hat{\boldsymbol{n}}_f}{\sum_{f=1}^{nf} \alpha_f} \qquad \hat{\boldsymbol{n}} = \frac{\boldsymbol{n}}{|\boldsymbol{n}|} \tag{5}$$

Geometry with sharp edges could result in violation of a visibility condition. The visibility condition is violated if the dot product of the node normal vector and any surrounding face normal vector becomes negative. A normal vector not visible to all surrounding faces could result in the generation of poor-quality prisms or hexahedra. To ensure that the node visibility condition is enforced, the normal vectors for nodes along sharp edges are forced to lie on the bisector plane [15]. The bisector plane is defined as the plane that bisects the smallest angle between any two faces surrounding the node, irrespective of whether it is a concave or convex edge. Sharp edges in the mesh are identified as edges between faces that contain a concave or convex turning angle of 30 deg or more. The bisector planes for nodes along these sharp edges are computed and stored for later use. These bisector planes will be used in the angle-weighted averaging just described and the normal vector smoothing process described next.

A normal vector smoothing process has been developed that respects sharp edges in the geometry and provides smoothly varying normal vectors across the surface. The initial normal vectors are computed using angle-weighted averages as previously described, with the necessary bisector planes to enforce the visibility condition. A sequence of smoothing passes is then performed to produce smoothly varying normal vectors, especially in corner regions. For sharp edge nodes, the smoothing passes perturb the node normal vectors based on weighted averages of neighboring face normal vectors. The modified face normal vectors $\tilde{\boldsymbol{n}}_f$ are the unit normal vectors for face f, but restricted to lie in the bisector plane for node i. The weighting factor, however, uses the current node normal vector and the unmodified face normal vector. The perturbation vector for sharp edge nodes is given in Eq. (6).

$$\Delta \boldsymbol{n}_i = \frac{\sum_{f=1}^{nf} (1 - \hat{\boldsymbol{n}}_i \hat{\boldsymbol{n}}_f)(\tilde{\boldsymbol{n}}_f - \hat{\boldsymbol{n}}_i)}{\sum_{f=1}^{nf} (1 - \hat{\boldsymbol{n}}_i \hat{\boldsymbol{n}}_f)} \qquad \tilde{\boldsymbol{n}}_f = \hat{\boldsymbol{n}}_f - (\hat{\boldsymbol{n}}_f \cdot \hat{\boldsymbol{b}}_i)\hat{\boldsymbol{b}}_i \tag{6}$$

The smoothing passes for other nodes perturb the normal vectors based on weighted averages of neighboring node normal vectors. The weighting is based on the dot product of the two vectors and the inverse of the distance d_{ij} between the surface nodes. The perturbation vector for all other surface nodes is given in Eq. (7).

$$\Delta \boldsymbol{n}_i = \frac{\sum_{j=1}^{nn} \frac{(1-\hat{\boldsymbol{n}}_i \hat{\boldsymbol{n}}_j)}{d_{ij}} (\hat{\boldsymbol{n}}_j - \hat{\boldsymbol{n}}_i)}{\sum_{j=1}^{nn} \frac{(1-\hat{\boldsymbol{n}}_i \hat{\boldsymbol{n}}_j)}{d_{ij}}} \tag{7}$$

The dot-product weighting factor will favor the neighboring nodes for which normal vectors are not aligned with node i, whereas the inverse-distance weighting factor will favor the nearest nodes. Combined, the weighting will provide smoothly varying normal vectors out of corner regions. The user sets the number of smoothing passes, typically on the order of 10–50. The smoothing passes will alter the normal vectors for all nodes, including those identified as sharp edge nodes. But the perturbation of sharp edge nodes will be in the bisector plane and will not violate the visibility condition. In addition, the visibility condition is monitored at all other nodes. Any nodes with normal vectors that violate the visibility condition are retagged as sharp edge nodes and computed using the sharp edge formula in Eq. (6). The bisector planes for these retagged nodes are initialized to a zero vector, and so the logic in Eq. (6) is unaltered. The node normal vectors are updated each pass, using Eq. (8). The relaxation factor ω is typically set to 0.25.

$$\boldsymbol{n}_i^{k+1} = \boldsymbol{n}_i^k + \omega \Delta \boldsymbol{n}_i \qquad \hat{\boldsymbol{n}}_i^{k+1} = \frac{\boldsymbol{n}_i^{k+1}}{\|\boldsymbol{n}_i^{k+1}\|} \tag{8}$$

One additional restriction is imposed on the normal vectors. Viscous layers can be inserted on boundaries that are adjacent to nonviscous boundaries, such as symmetry planes. The viscous node normal vectors along these interfaces are forced to lie in the plane of the nonviscous boundary. The implementation is general in that the adjacent boundary need not be a planar surface. This allows for normal vectors to follow the contour of an adjacent boundary, such as the wing/body juncture.

2. Multiple Normal Vectors

Multiple normal vectors can be used to improve the quality of the mesh at sharp corners. An example is shown in Fig. 2 for the trailing edge of an airfoil. The single normal vector case, shown on the left side of the figure, produces skewed elements at the trailing edge. The multiple normal vector case, shown on the right side of the figure, produces nonskewed elements and a higher-quality mesh. This capability has been incorporated in two dimensions only, thus far. The three-dimensional implementation is under way.

When multiple node normal vectors are used, the defined surface node normal vector is taken as the direction vector from the surface node to the interior nodes of the existing mesh. These directions are valid, because it is assumed that the existing mesh is valid. The smoothing process does not modify the normal vectors for nodes with multiple normal vectors.

B. Existing Normal Spacing and Viscous Layer Height Control

The insertion process examines the existing inviscid mesh size and allows new layers to be added only where the desired layer height is less than the existing inviscid mesh size in the normal direction. This distance is defined as the minimum distance, in the direction of the defined surface normal vector, to the opposite side of any element containing the node. This is depicted in two dimensions in Fig. 3 for triangular and quadrilateral elements. The displacement vector is tested for intersection with the opposite sides of the element. For triangles, there is one opposite side. For quadrilaterals, there are two

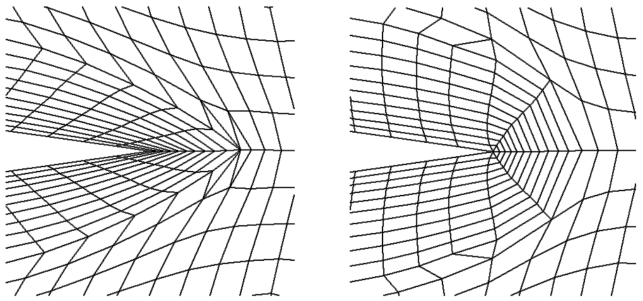

Fig. 2 Mesh near trailing edge of an airfoil with single and multiple normal vectors.

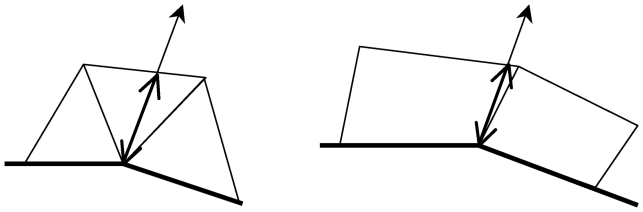

Fig. 3 Existing normal spacing for triangular and quadrilateral boundary elements.

opposite sides. In three dimensions, the existing mesh can contain tetrahedra, pyramids, prisms, and hexahedra.

A special case arises when the opposite face of the element is also on a boundary, as shown on the left side of Fig. 4. The region shown is a narrow gap between a tire geometry intersecting with a ground plane. It is obvious that the element shown cannot be pushed out of the way. The perturbation vector will force the element to be flattened completely. This situation is remedied by redefining the existing spacing as a fraction of the actual distance across the element, shown on the right side of the figure. The fractional amount was arbitrarily chosen as 25%.

Viscous layer height is dictated by the desired layer spacing compared with the existing normal spacing. If the desired layer spacing along the node normal direction is less than the existing mesh height, a new node will be inserted at the boundary and the existing node will be pushed into the interior. So only nodes meeting this criteria will have new nodes introduced. In two dimensions, the surface edges will produce triangles if only one node is perturbed and quadrilaterals if both nodes are perturbed. Three-dimensional surface triangular elements will produce prisms, pyramids, or tetrahedra, depending on how many of the three surface nodes are perturbed. Three-dimensional quadrilateral surface elements have restrictions imposed so that only two cases are allowed: all nodes perturbed, resulting in hexahedral elements, or two nodes on an edge perturbed, resulting in pyramid elements.

C. Specifying Normal Spacing Distribution

Because this viscous mesh generation process always inserts layers at the boundary, the top layer is inserted first and the bottom layer is inserted last. The bottom, or last, layer will be inserted at the desired surface normal spacing. The spacing for each layer is defined as the spacing for the layer closer to the boundary multiplied by a geometric progression factor. Typical values for f are between 1 and 1.5. The geometric progression factor can be a constant or it can be modified by multiplying by a geometric growth rate. This can significantly reduce the number of layers required to match the spacing in the existing mesh. The growth rate can range from 0.5 to 1.5. Typical values of g are in the range of 1.0 to 1.05. The equations describing this spacing distribution are given next.

$$\Delta d^i = \Delta d^{i-1} f^i \qquad f^i = f^{i-1} g \quad \text{for } i = 2, \text{ no. of layers} \tag{9}$$

The geometric progression factor is limited to a minimum value of 0.5–0.75 and a maximum value of 1.75–2. With these relations and restrictions, the desired spacing Δd at any given layer can be computed. The perturbation vector for each boundary node is given by the following equation.

$$\boldsymbol{p} = \hat{\boldsymbol{n}} * \Delta d^i \tag{10}$$

The insertion process continues to add layers until the desired number of layers has been inserted or the desired normal spacing has been achieved. If the current normal spacing at the boundary of an insertion group is less than or equal to the desired normal spacing at all points in the group, no more layers are inserted for that boundary insertion group. Other boundary insertion groups are processed as usual. If all boundary insertion groups satisfy the desired normal spacing, the insertion process stops and the viscous mesh is considered complete. Otherwise, the insertion process continues to add layers until the specified number of layers has been inserted.

D. Inserted and Perturbed Element Quality Check

Once the normal vector and normal spacing are defined, the quality of the perturbed elements and the new elements can be evaluated. This is necessary because the linear-elastic smoothing is not controlling the displacement of the boundary nodes. Those displacements are boundary conditions for the linear-elastic smoothing. So the defined boundary node perturbations must not deform the current boundary elements to the point at which negative volumes or inverted corners result. The new elements, formed after the linear-elastic smoothing, must also be checked for skewness. If a negative volume is detected for the perturbed elements, the perturbation distances for the boundary nodes used to form the element are iteratively reduced to the point at which a positive volume is recovered. Likewise, if a negative Jacobian is detected for any corner of new elements, the perturbation distance for that corner node is iteratively reduced to achieve a positive Jacobian. If any perturbations are modified as a result of the element quality check, a series of smoothing passes are performed to blend the newly computed normal spacing. This smoothing only allows local spacing values to decrease, to maintain the positive volumes and Jacobians.

E. Treatment of Sidewall Boundaries

The surface nodes on adjacent walls on which layers are not inserted are allowed to float freely on that surface. This is necessary to produce a high-quality mesh at symmetry boundaries, for instance. It also allows the method to be used to construct viscous layers in corner junctures. This is demonstrated in Fig. 5, in which viscous layers are inserted separately on five different groups of boundaries. Four of the walls are shown in the figure. The top and bottom faces of the box are grouped as one boundary. All other faces of the box are defined as individual boundaries. The layer insertion process alternates through the boundary insertion groupings and inserts a layer on one group at a time. After a layer has been inserted on each insertion group, the process repeats for the next layer. This alternating insertion approach is similar to the insertion process followed by Lohner and Cebral [5], except the current technique uses the linear-elastic relations to control the placement of the points on the adjacent boundary.

Although layers were inserted on each boundary for this demonstration, not all boundaries have to have layers inserted. The magnified views of a cutting plane through the interior of the mesh show the benefit of this alternating insertion process for the corner region. A viscous mesh constructed with only one insertion group would have edges coming out of the corners at 45-deg angles and would not resolve the corner region as well.

The elements introduced under triangular boundary faces are prisms, whereas the elements inserted under quadrilateral boundary

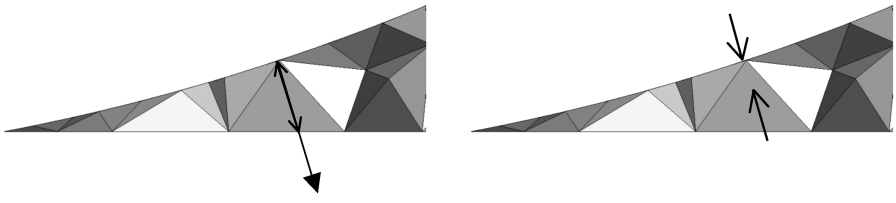

Fig. 4 Existing spacing for tetrahedra positioned between two boundaries.

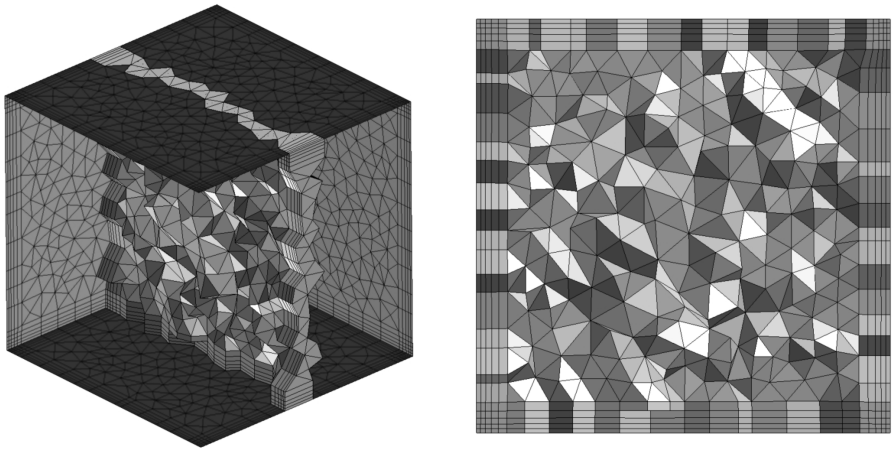

Fig. 5 Viscous layers inserted on multiple walls.

faces are hexahedra. Elements inserted next to adjacent, nonviscous walls or walls in another boundary insertion group create new quadrilateral faces on the adjacent boundary. These new boundary faces are added to the list of faces for those adjacent boundaries. As layers are inserted in alternating boundary groups, these new quadrilateral faces in the corner result in hexahedral elements. For this particular case, the initial mesh was all tetrahedral elements. The insertion process added prisms and hexahedra.

F. Mesh Quality

After each layer insertion, several quality measures are computed and printed for the user to examine. These quality measures were presented in [16] and include element condition number, element aspect ratio, corner solid angle, corner Jacobians, and statistical averaging of normal and tangential spacing. Some of the quality measures will naturally increase in value, due to the definition of the measure. For instance, the maximum condition number and aspect ratio of the prismatic layers will increase, because these metrics measure element shape, and the goal of the insertion process is to creating highly packed elements in the viscous region. Other measures, such as solid angle and corner Jacobian, should remain low, because they measure orthogonality of the mesh and the desire is to create nearly orthogonal elements in the viscous layers. In all of the cases shown next, no element corners became inverted, due to the perturbations defined by the linear-elastic relations.

IV. Results

Several examples are included to demonstrate the geometric complexity that can be modeled using linear-elastic mesh smoothing to generate viscous meshes. All of the examples are three-dimensional cases. Some of the cases are simple configurations with one insertion group. Other cases are much more complicated configurations with multiple insertion groups. All inviscid meshes were generated using Gridgen. All mesh images in this paper were created using Fieldview [17].

A. Onera M6 Wing

A classic three-dimensional CFD validation case is the Onera M6 wing [18]. The geometry is a simple single element wing abutting a plane of symmetry. The initial mesh, shown on the upper left of Fig. 6, was generated using Gridgen and contained 185,060 nodes and 1,011,739 tetrahedra. The surface includes the upper surface, lower surface, and wing tip. Thirty layers were requested in one insertion group that included all three of these boundaries. The geometry progression factor was 1.15 and the growth rate was 1.01. The logic controlling the layer height delayed the first nodes inserted until the sixth layer, where only 9 nodes were inserted, resulting in 23 tetrahedra and 2 pyramids. The normal vectors were smoothed using 20 passes. The linear-elastic smoother was allowed to run 20 subiterations with 10 GMRES search directions, if necessary. The final mesh, shown on the top right side and bottom of Fig. 6, contained 763,392 nodes, 1,005,746 tetrahedra, 8,384 pyramids, and 1,148,391 prisms. The complete viscous insertion process ran on a 2.4 GHz AMD Opteron-280 Linux machine in 82 min. The inviscid and viscous meshes on the symmetry plane for the wing leading edge are shown on the top of Fig. 7. The meshes near the wing tip at an axial station of 0.8 are shown on the bottom of the figure. The variable layer height can be seen in the magnified views of the viscous mesh near the leading edge and the tip. Even though some of the elements get squeezed, good element quality in the inviscid portion of the mesh is maintained after the layers have been inserted. The maximum tetrahedral aspect ratio only increased from 5.56 to 9.55. The maximum tetrahedral condition number increased from 4.40 to 9.28. The maximum tetrahedral solid angle increased from 2.99 to 3.73 rad (an orthogonal corner would have a value of $\pi/2$ and the maximum attainable value is $\pi/2$).

B. Generic High-Lift Configuration

A generic, three-dimensional high-lift configuration was created to demonstrate the viscous insertion process on a more complicated geometry. A two-dimensional, multi-element airfoil was extruded and tapered to create the wing geometry. The inviscid mesh was generated using Gridgen and contained 364,136 nodes and 1,992,846 tetrahedra. The geometry and the symmetry plane mesh are shown in Fig. 8. An initial geometric progression factor of 1.15 was used, with a growth rate of 1.01. Twenty-five total layers were requested. However, due to the logic controlling the layer height, no layers were actually inserted until the tenth layer. A total of 67 nodes were inserted in the tenth layer, resulting in 407 new tetrahedra and 6 new pyramids. The last layer introduced new nodes at all surface points and created 112,258 new prisms. The final mesh contained 1,110,623 nodes, 2,021,146 tetrahedra, 17,510 pyramids, and 1,464,794 prisms. The complete insertion process ran on the same Linux machine as the previous case, in 139 min. Views of the viscous mesh at the symmetry plane are shown in Fig. 9. The right side of the figure shows how the viscous layers advanced up the symmetry plane, creating new quadrilateral faces on the symmetry plane boundary.

Various views of the inviscid and viscous meshes for airfoil segments are shown in Figs. 10–14. Figure 15 shows an axial cut through the main element near the wing tip. The maximum tetrahedral aspect ratio increased from 4.305 to 8687. This may seem high, but it is probably due to tetrahedral elements that were introduced within the viscous layer region in which the layer height was restricted and some nodes of triangular surface elements did not march. This results in short edges in the normal direction and does not introduce large solid angles. The maximum tetrahedral condition number increased from 3.515 to 183.0.

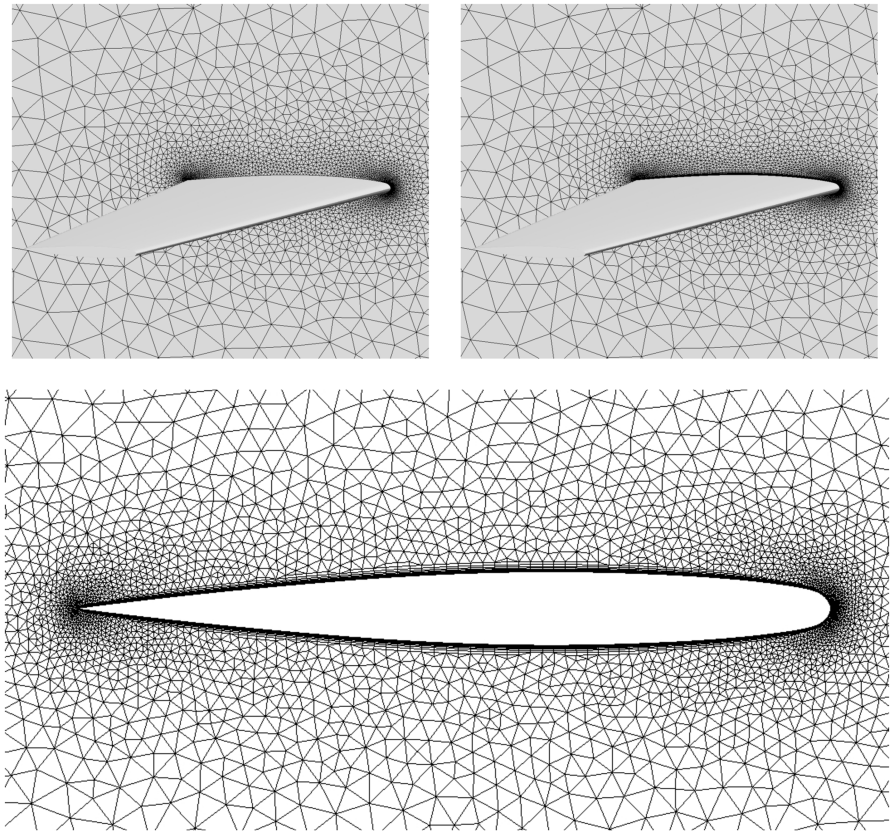

Fig. 6 Onera M6 geometry with symmetry plane meshes.

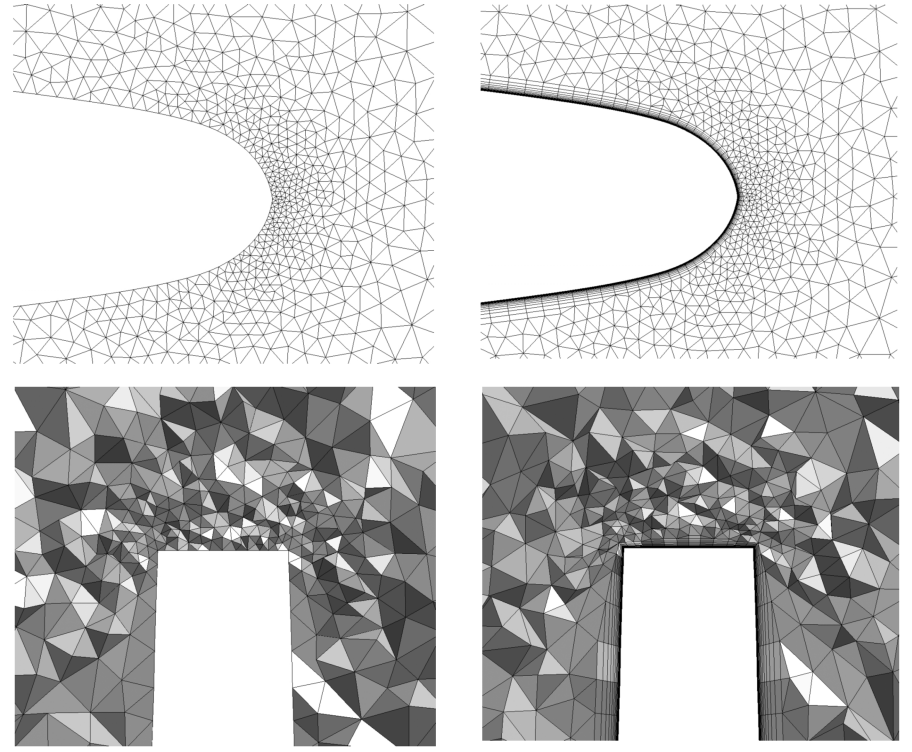

Fig. 7 Inviscid and viscous meshes near the wing leading edge and the wing tip.

C. Generic Conventional Model Tractor

Viscous layers were inserted into a tetrahedral mesh for a generic conventional model tractor and trailer. The inviscid mesh was generated using Gridgen, containing 3,894,285 nodes, 18,785,775 tetrahedra, 35,859 pyramids, 179,448 prisms, and 521,905 hexahedra. The mesh included a ground plane and contained a symmetry plane located at the middle of the tractor-trailer assembly. Thirty layers were inserted, with a geometric progression factor of 1.15 and a geometric growth rate of 1.01. This case was performed before incorporating the layer height control logic, so that layers were inserted at each surface node for each layer. The linear-elastic solver used the point-implicit scheme described in , instead of the

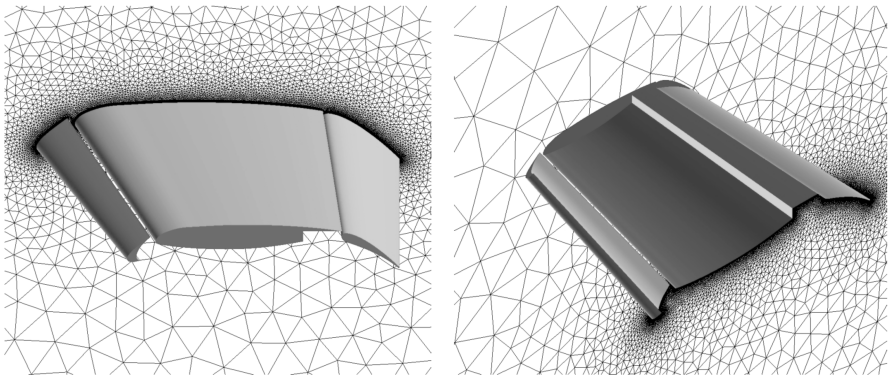

Fig. 8 Generic high-lift wing geometry and symmetry plane mesh.

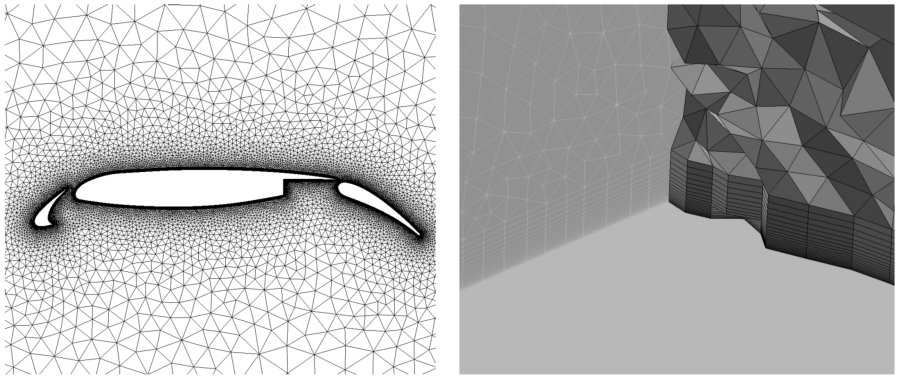

Fig. 9 Symmetry plane viscous mesh and magnified view of viscous elements near the symmetry plane.

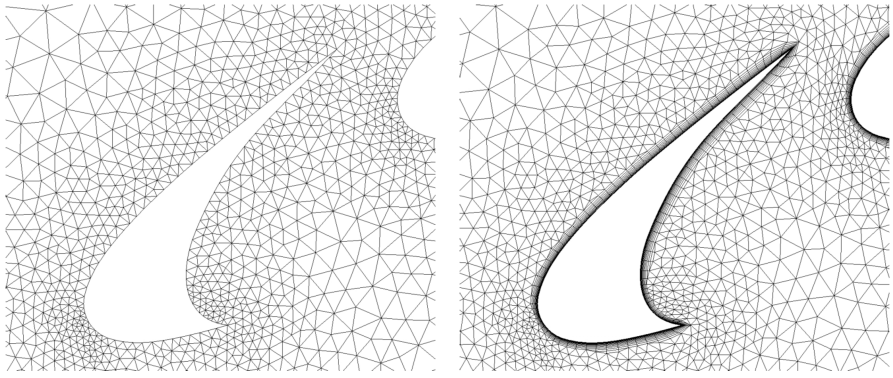

Fig. 10 Inviscid and viscous meshes of the slat at the symmetry plane.

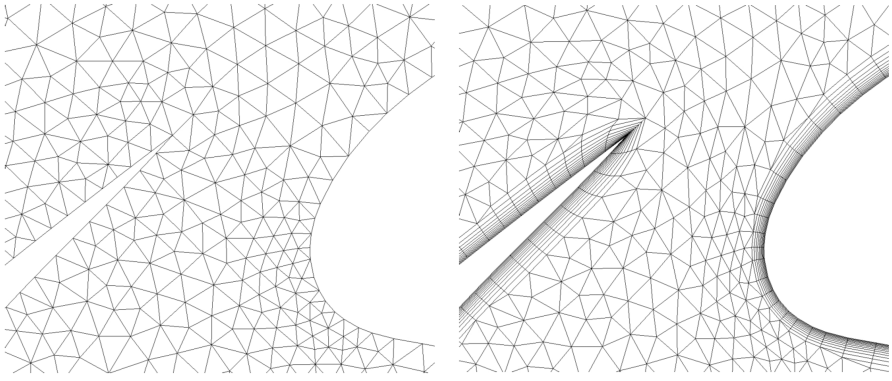

Fig. 11 Inviscid and viscous meshes of the slat; main element gap at the symmetry plane.

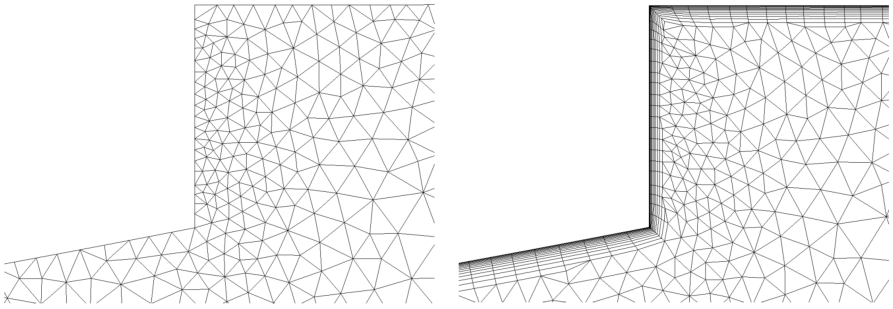

Fig. 12 Inviscid and viscous meshes of the rear-facing step of the main element at the symmetry plane.

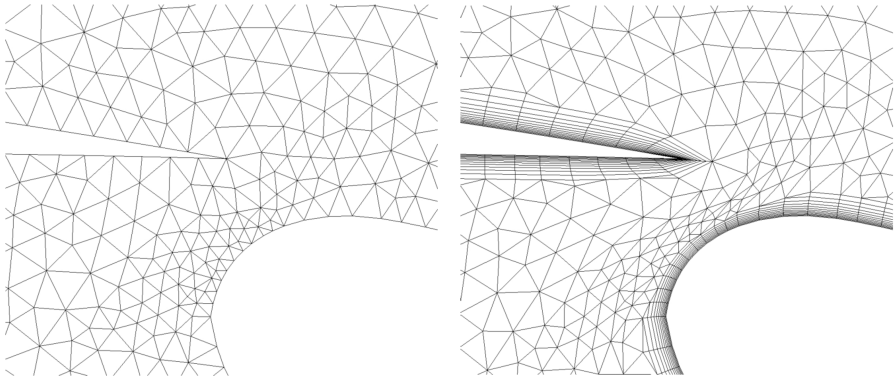

Fig. 13 Inviscid and viscous meshes of the main element; flap gap at the symmetry plane.

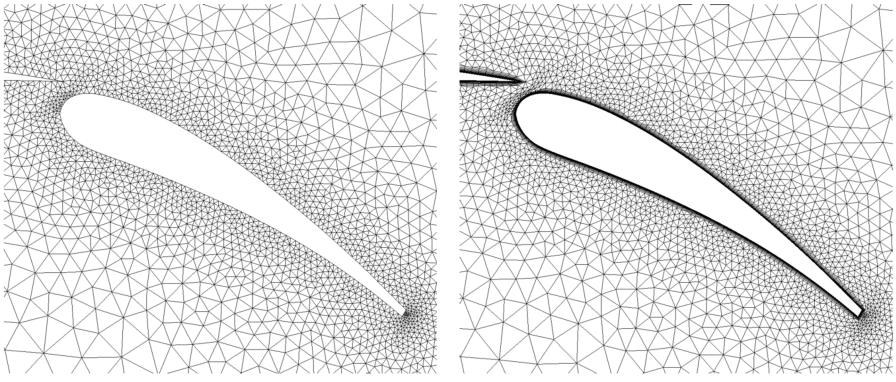

Fig. 14 Inviscid and viscous meshes of the flap at the symmetry plane.

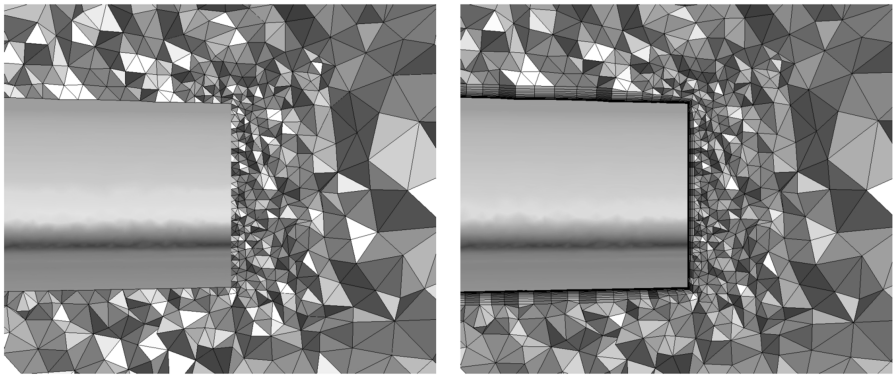

Fig. 15 Inviscid and viscous meshes of axial cut through the main element wing tip.

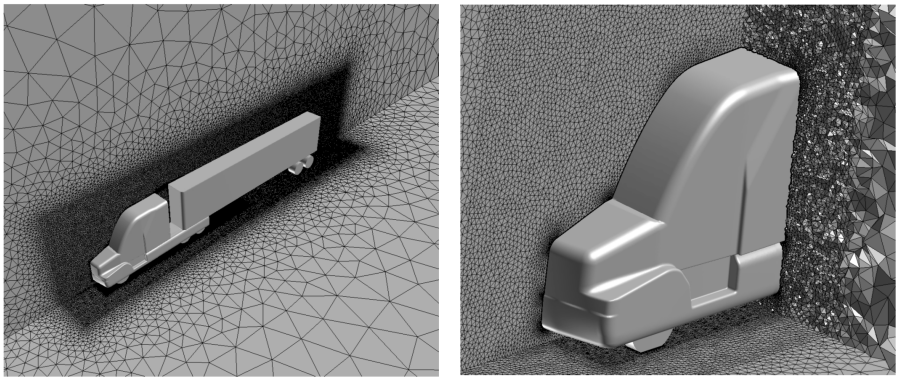

Fig. 16 Generic model tractor and trailer mesh.

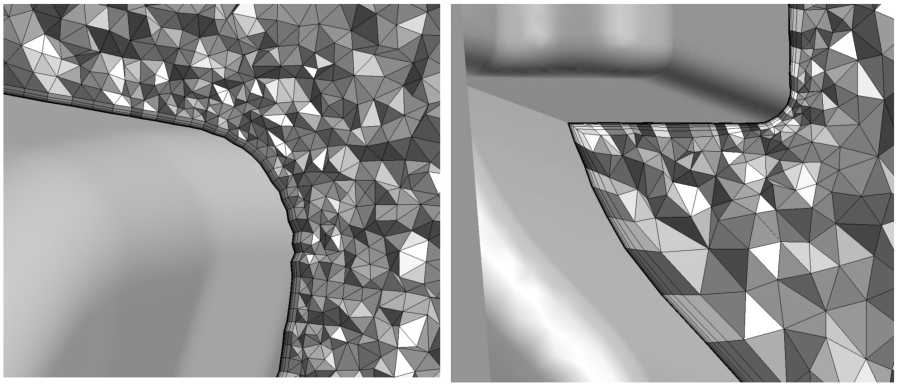

Fig. 17 Magnified views axial cut through the cab.

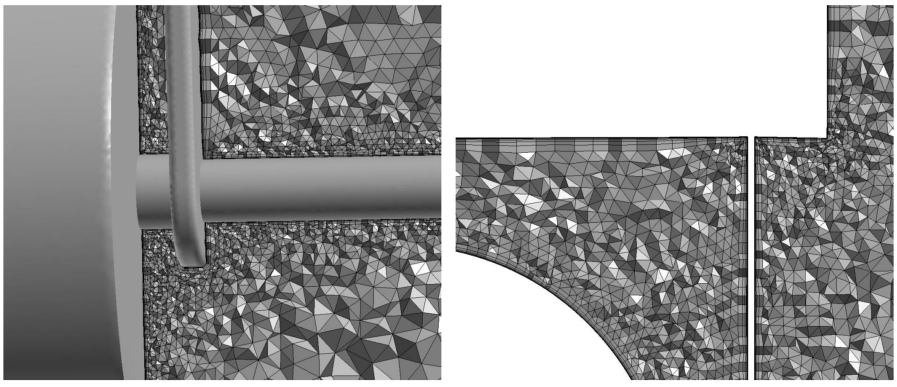

Fig. 18 Axial cut through mesh near the rear axle and side view of cut through mesh near the mid flap.

GMRES solver, and was allowed to run 100 iterations to converge two orders of magnitude. The final mesh contained 12,138,015 nodes, 18,785,775 tetrahedra, 35,859 pyramids, 15,948,768 prisms, and 831,505 hexahedra. It ran on a 64-bit, 3.6-GHz Intel Zeon, Linux machine with 32 GB of memory. The serial job completed in 45.87 hr and used approximately 15 GB of memory. The left side of Fig. 16 shows the truck surface, the ground plane, and the symmetry plane. The right side of the figure shows an axial cut through the cab.

A magnified views of the viscous layers inserted on the top corner of the cab are shown on the left side of Fig. 17. A concave region near the doorstep is shown on the right side of the figure. The surface normal vectors near the corner vary smoothly. The normal vectors in the corner were labeled sharp and the smoothing process restricted those vectors to lie in the bisector planes.

Two boundary insertion groups were used to generate this mesh. One group included the rear axles and the mud flap shown in Fig. 18. The remaining truck surfaces were included in the second insertion group. Alternating through these two groups created hexahedral elements in the corners.

D. Suboff

The final example is a viscous mesh for a fully appended Suboff configuration [19]. Like the previous case, this was performed before incorporating the layer height control logic and the GMRES solver. The geometry, shown in Fig. 19, contained the sail, stern appendages (rears horizontal and vertical fins), and propulsor shroud, or ring wing, with support struts. The initial Gridgen mesh contained 2,702,290 nodes and 15,106,590 tetrahedra. Twenty layers were inserted, with an initial geometric progression factor of 1.2 and a geometric growth rate of 1.02. The final mesh contained 12,319,910 nodes, 15,106,590 tetrahedra, 15,739,400 prisms, and 1,664,800 hexahedra. The linear-elastic solver was allowed to run 100 iterations in order to converge two orders of magnitude. The final layer required only 45 iterations to converge. This case was also run on the 64-bit Linux machine, but the total run time was not recorded.

Two boundary insertion groups were used for this case. One group contained the main body and ring wing. The other group contained the sail stern appendages and ring wing struts. The result of this grouping combination is hexahedral elements at the sail/main body

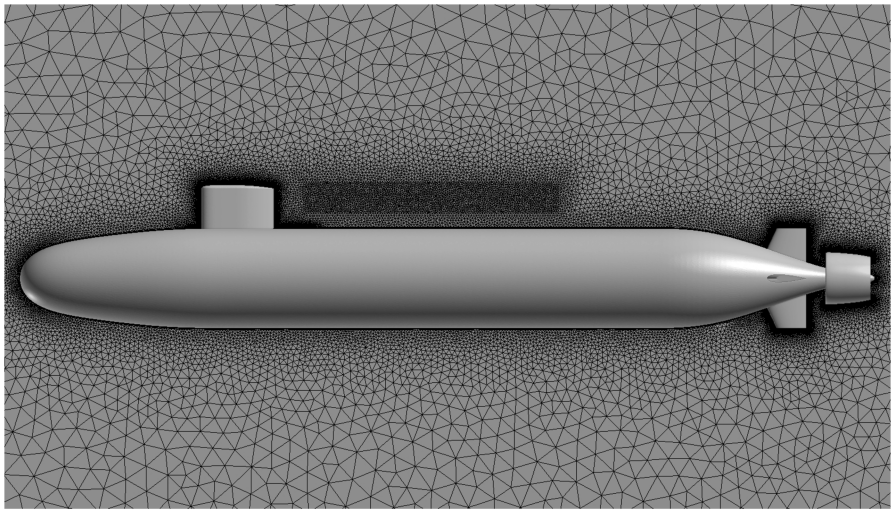

Fig. 19 Side view of the Suboff configuration.

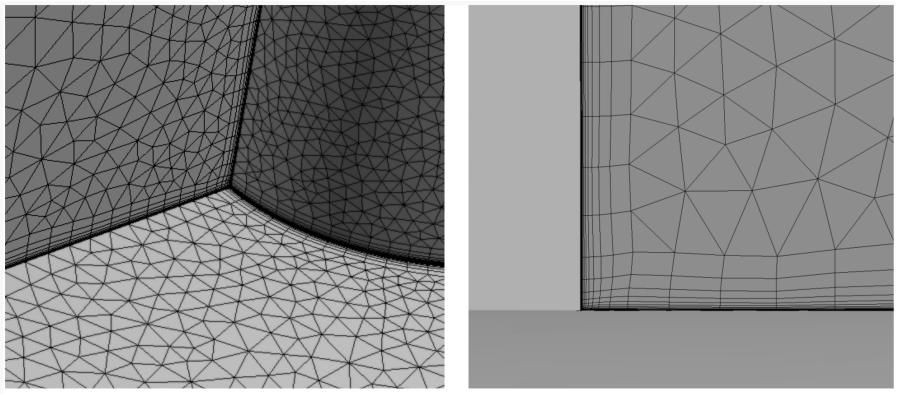

Fig. 20 Main body/sail juncture region containing hexahedral elements.

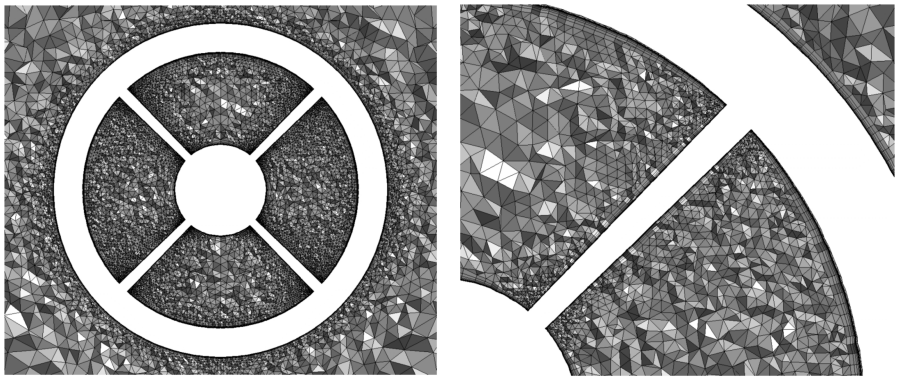

Fig. 21 Aft view of the propulsor shroud.

juncture, the stern appendages and main body juncture, and the strut/ring wing juncture. Views of the sail/main body juncture are shown in Fig. 20. Views of the strut/ring wing juncture are shown in Fig. 21.

V. Conclusions

A method for generating unstructured viscous meshes using layer insertion was described. This method uses linear-elastic smoothing to insert new layers at the boundary of an existing mesh and pushes the interior points away from the surface in a very robust manner. Layers of points are inserted in reverse order, with the final layer positioned at the desired normal spacing. Layers need not be inserted at each surface point for each layer. Instead, layers are inserted only where the desired layer height is smaller than the existing mesh normal distance. Mesh points on sidewall boundaries are allowed to float freely as the viscous boundary points are pushed into the interior. Inserting layers on alternating groups of boundaries can be used to generate tightly packed elements off both walls in the corner regions. Several three-dimensional cases were described. The first couple of cases were relatively simple geometries with one boundary insertion group. The latter cases were much larger, more complex configurations that used two insertion groups to properly resolve corner regions in the geometry.

All cases were run in serial on Linux workstations. The smaller cases were generated on a Linux machine that contained 16 GB of RAM. Neither case occupied more than 2 GB of memory. The last two cases were run on a 64-bit Linux workstation that contained 32 GB of RAM. Neither case occupied more than 15 GB of RAM

during the calculation. Run times were reported for the smaller cases and required between 1 and 3 hr to complete. The run time for the third case was 45.87 hr.

VI. Future Work

The implementation of the multiple normal vectors capability for sharp edges will be completed in 3-D. This will allow better resolution of sharp edge regions. The linear-elastic solver will be ported to a distributed memory, parallel Linux cluster. This will reduce the run times considerably and will also enable larger cases to be run.

Appendix A: Three-Dimensional Equations

The three-dimensional equations for Winslow smoothing and linear-elasticity smoothing can be combined into a generalized set of equations and are included for reference. The set of equations shown here includes the form of linear-elasticity that allows for variable Young's modulus, whereas [10] included the form with constant Young's modulus and variable Poisson's ratio.

$$\frac{\partial}{\partial x}\left[\alpha_{11}\frac{\partial u}{\partial x}\right]+\frac{\partial}{\partial y}\left[\alpha_{12}\frac{\partial u}{\partial y}\right]+\frac{\partial}{\partial z}\left[\alpha_{13}\frac{\partial u}{\partial z}\right]+\alpha_{11}\Phi\frac{\partial u}{\partial x}+\alpha_{12}\Psi\frac{\partial u}{\partial y}+\alpha_{13}\Omega\frac{\partial u}{\partial z}+2\left[\beta_1\frac{\partial^2 u}{\partial x\partial y}+\beta_2\frac{\partial^2 u}{\partial y\partial z}+\beta_3\frac{\partial^2 u}{\partial x\partial z}\right]+\frac{\partial}{\partial x}\left[\theta_{11}\left(\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)\right]+\frac{\partial}{\partial y}\left[\theta_{12}\frac{\partial v}{\partial x}\right]+\frac{\partial}{\partial z}\left[\theta_{13}\frac{\partial w}{\partial x}\right]=0$$

$$\frac{\partial}{\partial x}\left[\alpha_{21}\frac{\partial v}{\partial x}\right]+\frac{\partial}{\partial y}\left[\alpha_{22}\frac{\partial v}{\partial y}\right]+\frac{\partial}{\partial z}\left[\alpha_{23}\frac{\partial v}{\partial z}\right]+\alpha_{21}\Phi\frac{\partial v}{\partial x}+\alpha_{22}\Psi\frac{\partial v}{\partial y}+\alpha_{23}\Omega\frac{\partial v}{\partial z}+2\left[\beta_1\frac{\partial^2 v}{\partial x\partial y}+\beta_2\frac{\partial^2 v}{\partial y\partial z}+\beta_3\frac{\partial^2 v}{\partial x\partial z}\right]+\frac{\partial}{\partial x}\left[\theta_{21}\frac{\partial u}{\partial y}\right]+\frac{\partial}{\partial y}\left[\theta_{22}\left(\frac{\partial u}{\partial x}+\frac{\partial w}{\partial z}\right)\right]+\frac{\partial}{\partial z}\left[\theta_{23}\frac{\partial w}{\partial y}\right]=0$$

$$\frac{\partial}{\partial x}\left[\alpha_{31}\frac{\partial w}{\partial x}\right]+\frac{\partial}{\partial y}\left[\alpha_{32}\frac{\partial w}{\partial y}\right]+\frac{\partial}{\partial z}\left[\alpha_{33}\frac{\partial w}{\partial z}\right]+\alpha_{31}\Phi\frac{\partial w}{\partial x}+\alpha_{32}\Psi\frac{\partial w}{\partial y}+\alpha_{33}\Omega\frac{\partial w}{\partial z}+2\left[\beta_1\frac{\partial^2 w}{\partial x\partial y}+\beta_2\frac{\partial^2 w}{\partial y\partial z}+\beta_3\frac{\partial^2 w}{\partial x\partial z}\right]+\frac{\partial}{\partial x}\left[\theta_{31}\frac{\partial u}{\partial z}\right]+\frac{\partial}{\partial y}\left[\theta_{32}\frac{\partial v}{\partial z}\right]+\frac{\partial}{\partial z}\left[\theta_{33}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right]=0$$

For linear-elasticity,

$$\beta_1=\beta_2=\beta_3=0,\qquad \Phi=\Psi=\Omega=0$$

$$\alpha_{11}=\frac{E(1-\nu)}{(1+\nu)(1-2\nu)},\qquad \alpha_{12}=\frac{E}{2(1+\nu)}$$

$$\alpha_{13}=\frac{E}{2(1+\nu)},\qquad \theta_{11}=\frac{E\nu}{(1+\nu)(1-2\nu)}$$

$$\theta_{12}=\frac{E}{2(1+\nu)},\qquad \theta_{13}=\frac{E}{2(1+\nu)},\qquad \alpha_{21}=\frac{E}{2(1+\nu)}$$

$$\alpha_{22}=\frac{E(1-\nu)}{(1+\nu)(1-2\nu)},\qquad \alpha_{23}=\frac{E}{2(1+\nu)}$$

$$\theta_{21}=\frac{E}{2(1+\nu)},\qquad \theta_{22}=\frac{E\nu}{(1+\nu)(1-2\nu)}$$

$$\theta_{23}=\frac{E}{2(1+\nu)},\qquad \alpha_{31}=\frac{E}{2(1+\nu)},\qquad \alpha_{32}=\frac{E}{2(1+\nu)}$$

$$\alpha_{33}=\frac{E(1-\nu)}{(1+\nu)(1-2\nu)},\qquad \theta_{31}=\frac{E}{2(1+\nu)}$$

$$\theta_{32}=\frac{E}{2(1+\nu)},\qquad \theta_{33}=\frac{E\nu}{(1+\nu)(1-2\nu)}$$

For Winslow,

$$\theta_{ij}=0,\qquad \boldsymbol{r}=(u,v,w)$$

$$\alpha_{11}=\alpha_{21}=\alpha_{31}=(\boldsymbol{r}_y\cdot\boldsymbol{r}_y)(\boldsymbol{r}_z\cdot\boldsymbol{r}_z)-(\boldsymbol{r}_y\cdot\boldsymbol{r}_z)(\boldsymbol{r}_y\cdot\boldsymbol{r}_z)$$

$$\beta_1=(\boldsymbol{r}_y\cdot\boldsymbol{r}_z)(\boldsymbol{r}_z\cdot\boldsymbol{r}_x)-(\boldsymbol{r}_x\cdot\boldsymbol{r}_y)(\boldsymbol{r}_z\cdot\boldsymbol{r}_z)$$

$$\alpha_{12}=\alpha_{22}=\alpha_{32}=(\boldsymbol{r}_z\cdot\boldsymbol{r}_z)(\boldsymbol{r}_x\cdot\boldsymbol{r}_x)-(\boldsymbol{r}_z\cdot\boldsymbol{r}_x)(\boldsymbol{r}_z\cdot\boldsymbol{r}_x)$$

$$\beta_2=(\boldsymbol{r}_z\cdot\boldsymbol{r}_x)(\boldsymbol{r}_x\cdot\boldsymbol{r}_y)-(\boldsymbol{r}_y\cdot\boldsymbol{r}_z)(\boldsymbol{r}_x\cdot\boldsymbol{r}_x)$$

$$\alpha_{13}=\alpha_{23}=\alpha_{33}=(\boldsymbol{r}_x\cdot\boldsymbol{r}_x)(\boldsymbol{r}_y\cdot\boldsymbol{r}_y)-(\boldsymbol{r}_x\cdot\boldsymbol{r}_y)(\boldsymbol{r}_x\cdot\boldsymbol{r}_y)$$

$$\beta_3=(\boldsymbol{r}_x\cdot\boldsymbol{r}_y)(\boldsymbol{r}_y\cdot\boldsymbol{r}_z)-(\boldsymbol{r}_z\cdot\boldsymbol{r}_x)(\boldsymbol{r}_y\cdot\boldsymbol{r}_y)$$

Acknowledgments

The University of Tennessee at Chattanooga through the Lupton Renaissance Fund sponsored this work. Calmar Research Corporation and NASA Ames Research Center provided the geometry for the tractor-trailer configuration. This support is greatly appreciated.

References

[1] Gridgen Software Package, Ver. 15, Pointwise, Inc., Forth Worth, TX; also available online at http://www.pointwise.com/.

[2] VGRID Software Package, Ver. 2.7, ViGYAN, Inc., Hampton, VA; also available online at http://www.vigyan.com/vgrid.

[3] GAMBIT Geometry and Mesh Generation Software Package, Ver. 2.3, Fluent, Inc., Lebanon, NH; also available online at http://www.fluent.com/software/gambit/index.htm/.

[4] Marcum, D. L., "Unstructured Grid Generation Using Automatic Point Insertion and Local Reconnection," *Handbook of Grid Generation*, edited by J. Thompson, B. Soni, and N. Weatherill, CRC Press, Boca Raton, FL, 1999, Chap. 18.

[5] Lohner, R., and Cebral, J., "Generation of Nonisotropic Unstructured Grids via Directional Enrichment," *International Journal for Numerical Methods in Engineering*, Vol. 49, 2000, pp. 219–232.

[6] Ito, Y., and Nakahashi, K., "Improvements in the Reliability and Quality of Unstructured Hybrid Mesh Generation," *International Journal for Numerical Methods in Fluids*, Vol. 45, 2004, pp. 79–108.

[7] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," AIAA Paper 2001-0596, Jan. 2001.

[8] Yang, Z., and Mavriplis, D. J., "Unstructured Dynamic Meshes with Higher-order Time Integration Schemes for the Unsteady Navier-Stokes Equations," AIAA Paper 2005-1222, Jan. 2005.

[9] Karman, S. L., Jr., "Unstructured Viscous Layer Insertion Using Linear-Elastic Smoothing," AIAA Paper 2006-0531, Jan. 2006.

[10] Karman, S. L., Jr., Anderson, W. K., and Sahasrabudhe, M., "Mesh Generation Using Unstructured Computational Meshes and Elliptic Partial Differential Equation Smoothing," AIAA Paper 2005-0923, Jan. 2005.

[11] Karman, S. L., Jr., Anderson, W. K., and Sahasrabudhe, M., "Mesh Generation Using Unstructured Computational Meshes and Elliptic Partial Differential Equation Smoothing," *AIAA Journal*, Vol. 44, No. 6, June 2006, pp. 1277–1286.

[12] Hornsey, E., McFarland, D., Muhlbauer, K., and Smith, B., *Mechanics of Materials, An Individualized Approach*, Houghton Mifflin, Boston, 1977.

[13] Freitag, L. A., and Knupp, P. M., "Tetrahedral Element Shape Optimization ia the Jacobian Determinant and Condition Number," *8th International Meshing Roundtable*, [online database], http://www.andrew.cmu.edu/user/sowen/imr8.html [retrieved 15 July 2005].

[14] Knupp, P. M., "Matrix Norms and the Condition Number: A General Framework to Improve Mesh Quality via Node-Movement," *8th International Meshing Roundtable* [online database], http://www.andrew.cmu.edu/user/sowen/imr8.html [retrieved 15 July 2005].

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

[16] Sahasrabudhe, M. S., Karman, S. L., Jr., and Anderson, W. K., "Grid Control of Viscous Unstructured Meshes Using Optimization," AIAA Paper 2006-0532, Jan. 2006.

[17] FIELDVIEW Software Package, Ver. 11.1, Intelligent Light, Inc., Rutherford, NJ; also available online at http://www.ilight.com/.
[18] Anon., "Onera M6 Wing," *NPARC Alliance Verification and Validation Archive* [online database], http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html [retrieved 12 Dec. 2005].
[19] Roddy, R. F., "Investigation of the Stability and Control Characteristics of Several Configurations of the DARPA Suboff Model (DTRC Model 5470) from Captive-Model Experiments," David Taylor Research Center, Rept. DTRC/SHD-1298-08, Sept. 1990.

K. Fujii
Associate Editor