JOURNAL OF PROPULSION AND POWER
Vol. 18, No. 3, May–June 2002

Three-Dimensional Hybrid Mesh Generation for Turbomachinery Airfoils

Kyu-sup Kim* and Paul G. A. Cizmas†
Texas A&M University, College Station, Texas 77843-3141

A methodology is introduced for generating three-dimensional hybrid meshes for turbomachinery airfoils. This method creates a volume mesh by stacking layers of area meshes along the blade spanwise direction. The area meshes are hybrid meshes composed of a structured O grid around the airfoil and an unstructured grid in the rest of the domain. Each layer of the stacked mesh is mapped from a single source mesh. Optimization-based smoothing is applied to the mapped area mesh to increase mesh quality. The optimization-based smoothing is also applied to the periodic boundaries to optimize the placement of the grid points. Further enhancement of grid quality is obtained by edge swapping and node insertion. Edge swapping reduces the dependency of mesh quality on the choice of the source mesh. The meshing algorithm is tested on a challenging turbine rotor airfoil whose stagger angle varies approximately 88 deg from hub to tip.

Nomenclature

A	=	cell area
$\boldsymbol{E}_{ij}$	=	edge vector from node N^i to N^j
$\boldsymbol{e}_n$	=	unit vector normal to the triangle cell
$\boldsymbol{G}^i$	=	gradient of the metric at node N^i
$\boldsymbol{n}$	=	unit vector normal to the cell edge
T	=	cell number
$\boldsymbol{X}^i$	=	coordinate of node N^i
γ	=	factor that controls the magnitude of node displacement
τ	=	mesh quality metric

Introduction

HYBRID meshes are widely used in viscous computational fluid dynamics problems because they allow better control of the cell size in the viscous region[1]. Prism cells for the viscous region and tetrahedral cells for the inviscid region are a common combination for a three-dimensional mesh. The advancing-layers method is a popular way to cluster the cells in the viscous region.[2] These advanced methods in unstructured mesh generation techniques, however, require a high implementation cost.

To develop a simple yet effective volume mesh generation tool, the present work uses a mapping method and mesh smoothing techniques. Mapping, which is often referred to as two-and-one-half-dimensional meshing, is a popular volume mesh generation method in finite element method analysis. Mapping is typically used for hexahedral meshing.[3–5] As the name suggests, the method maps a two-dimensional source mesh in sweeping motion to generate a volume mesh. The boundary ribs, that is, a series of ringlike closed boundaries, are initially generated along the sweeping direction to guide the mapping process. The spacing between the layers is specified before the mapping as in a structured mesh generation.

Staten et al.[5] used unstructured background mesh and linear interpolation to map the nodes from the source layer to the target layer. Mapping alone, however, cannot guarantee that the mesh will be high quality. Even though the nodes are always projected within the interior, the edges can be tangled, especially near a concave boundary. Herein, a set of structured background meshes together with linear interpolation are used to map the nodes from the source layer to the target layer. Mesh smoothing is then conducted to increase the quality of the target mesh while maintaining connectivity.

A smoothing method with guaranteed mesh quality improvement is extremely important. Laplacian smoothing is the most widely used smoothing method due to its simplicity and effectiveness in moderate cases. One of the variations of Laplacian smoothing is based on tension spring analogy. This method has been applied to the unsteady flow computation of a pitching blade.[6] The limitation of this method is that the quality of the smoothed mesh is not always improved.[7] Specifically, the Laplacian method and its variant methods can degenerate the mesh near a concave boundary by creating inverted cells or placing a node outside of the valid region. A smart Laplacian smoothing works around the problem by rejecting the displacement of a candidate node computed by a conventional Laplacian method if the new node location decreases the quality measure.

Angle-based smoothing uses a torsion spring analogy and is designed to achieve a smooth variation of angles between neighbor cells.[8] The edge length, however, is not directly controlled, and thus, it appears that this method requires a moderate quality initial mesh. Another class of the mesh smoothing method, which has recently become popular, is based on the direct optimization of the quality measure.[9,10] The optimization can be either a global or a local process, with the latter approach generally being cost effective. The list of the quality measures can be found by Amenta et al.,[11] who suggest that a mixture of quality measures be used. They show that a smoothing method coupled with a quality measure based only on edge length can create collapsed cells.

Optimization-based smoothing for three-dimensional meshes is versatile, but carries with it a high computational cost. The present work uses a two-dimensional mesh smoothing to generate a three-dimensional mesh in a cost effective way. The next section describes the new method. The subsequent sections present the mapping, mesh smoothing, edge swapping, and node insertion. Numerical results of the meshing methodology are then presented for a challenging turbine rotor airfoil that has a large radial variation in shape, size, and stagger angle.

Method

The present method takes advantage of the geometry features that are specific to turbomachinery cascades. Consequently, the cross sections of the flow domain at different radial location are discretized

Presented as Paper 2001-3209 at the AIAA/ASME/SAE/ASEE 37th Joint Propulsion Conference, Salt Lake City, UT, 8–11 July 2001; received 14 August 2001; revision received 18 January 2002; accepted for publication 18 January 2002.

*Research Assistant, Department of Aerospace Engineering. Student Member AIAA.

†Assistant Professor, Department of Aerospace Engineering. Associate Fellow AIAA.

using topologically identical meshes. The method presented in this paper generates the three-dimensional mesh as a sequence of two-dimensional layers stacked along the span of the airfoil.

The two-dimensional grid is a combination of a structured O grid, next to the airfoil surface, and an unstructured grid exterior to the O grid. The O grid is used to resolve the viscous region in the vicinity of the blade. The remainder of the two-dimensional domain is delimited by the outer boundary of the O grid, the periodic boundaries, and the inlet and outlet boundaries. This two-dimensional domain is filled by triangular cells. The unstructured mesh is used to make the mesh generator applicable to a complex geometry for which conventional structured meshes may fail to produce an acceptable result.

The structured O grid is generated first, followed by the source mesh. The source mesh is a two-dimensional unstructured mesh. The source mesh is then projected to target layers at different spanwise locations. As many target layers as necessary are generated to capture blade geometry variation from hub to tip. Edges are then added between the twin nodes of adjacent target layers to create volume cells. Consequently, the three-dimensional cells are either prisms or hexahedra, depending on whether they correspond to triangle or quadrilateral two-dimensional cells.

To improve mesh quality, the grid is smoothed by relocating the interior and periodic boundary nodes to their optimal points. The smoothing procedure uses an optimization-based smoothing technique. The O-grid region is not smoothed. The outer boundary of the O grid serves as a fixed boundary for the smoothing process.

The spanwise variation of the airfoil shape results in a deformation of the elements of the mapped meshes. This deformation reduces the quality of the mesh. The expression of the grid quality used herein is provided in Eq. (1) The mesh quality is further reduced by the fixed periodic boundaries that limit the redistribution of the nodes close to the boundaries. Unlike typical boundaries in solid modeling, the periodic boundaries of turbomachinery cascade do not have to be fixed. The only restriction is that the circumferential distance between two periodic nodes on the periodic boundaries must be constant. The method proposed allows that the nodes of the periodic boundaries move. The position of the nodes on the periodic boundaries results from the smoothing of the neighboring internal nodes. This additional degree of freedom for the nodes on the periodic boundary yields a better quality mesh.

A major advantage of the technique developed in this study is that the complexity of a three-dimensional algorithm is reduced to that of a two-dimensional algorithm. The structured nature of the mesh in the spanwise direction simplifies mesh generation. In addition, flow computation benefits from the simple partition of the mesh. Because the connectivity is identical among the layers, the communication across them is simplified. This makes the mapped mesh attractive to parallel flow computation.

Mapping

The initial step in the mapped mesh generation is the definition of the boundary ribs. The blade surface is cross sectioned at a predetermined locations along the spanwise direction. An equal number of nodes is placed around the airfoil at each spanwise location. The outer boundaries, that is, the inlet, outlet, and periodic boundaries, are defined at every radial location. An equal number of nodes is used for the outer boundaries at each spanwise location.

An O-grid mesh is generated to discretize the viscous region around the blade at each spanwise location. The hexahedral volume cells in the O-grid region are constructed by connecting the topologically identical quadrilateral cells of adjacent layers. These hexahedral cells are fixed permanently for the remainder of the mesh generation steps. The outer boundary of the O-grid block defines the inner boundary of the unstructured mesh.

The mesh of the source layer is generated first. The interior of the domain is bounded by the outer boundaries (inlet, outlet, and periodic) and the outer boundaries of the O grid. This interior domain is tessellated to produce a mesh with triangular cells. A two-dimensional Delaunay triangulator based on the divide-and conquer method is used herein to generate the two-dimensional unstructured

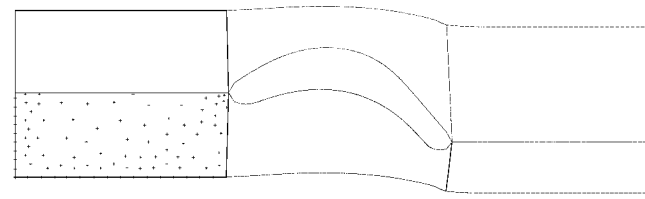


Fig. 1 Boundaries of background meshes with nodes in the lower inlet corner.

mesh (see Ref. 12). This mesh serves as the source to be mapped on the target layers.

The domain to be discretized by the unstructured mesh is divided into six blocks, as shown in Fig. 1. Structured meshes are algebraically generated for each block. These additional meshes are called the background meshes. Similar background meshes are constructed on the target layers so that a cell-to-cell mapping is established between the source and the target background meshes. The source nodes are then projected on the target layer into the corresponding cell of the background mesh. Bilinear interpolation is performed to find the mapped location of the node within the cell.

Mesh Smoothing

A mapped mesh may not be acceptable for flow computation due to the lack of explicit quality control in the mapping procedure. Unless the variation from the source to target layer is small, the mapping can produce low-quality cells or even tangled elements. One of the most effective techniques for increasing the quality of a three-dimensional mesh is optimization-based smoothing. The computational cost of this technique, however, can be very high. The majority of the computational cost is due to the computation of quality measures. Gradient computation can also contribute significantly to the computational cost in the steepest descent method optimization.

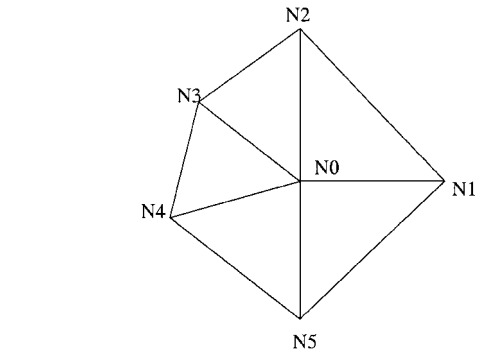
The present study uses a two-dimensional quality measure rather than a three-dimensional quality measure. The underlying assumption is that the variation from one layer to the adjacent layer is small. This is true for turbomachinery airfoils, which usually have a continuous variation of the cross section in the spanwise direction. Consequently, the overall quality of a prism cell, which has two triangular faces on the two adjacent layers, is dominated by the quality of the triangular cells. The mesh generation essentially becomes a series of two-dimensional mesh operations of area mesh smoothing. This approach not only reduces the computational cost significantly, but also simplifies the implementation.

Submesh

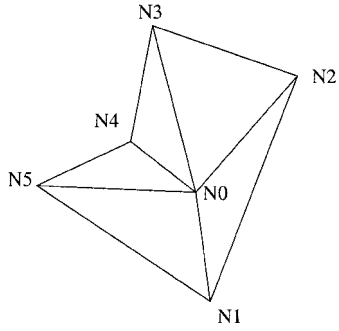
A submesh is an entity used for local submesh smoothing. A submesh is defined by a set of cells that share a common node. Typical convex and star submeshes are shown in Figs. 2a and 2b. All of the corners of a convex submesh are convex. At least one corner is concave in a star submesh. The type of the submesh is important in the smoothing process. Mesh smoothing is applied iteratively to submeshes. Given a submesh as an input, local mesh optimization is then reduced to finding the optimal location of the common node while the other nodes on the boundary of the submesh are stationary.

For a node on the periodic boundary, a submesh is defined as shown in Fig. 2c. Figure 2c shows a submesh for node N_0, which lies on the periodic boundary. Note that nodes $N_{1'}$ and $N_{2'}$ are the pairs of nodes N_1 and N_2, respectively. Nodes $N_{5''}$ and $N_{6''}$ are the pairs of nodes N_5 and N_6, respectively. The submesh for node N_0 can be assembled by using the boundary nodes N_5 and N_6 of the submesh of node $N_{0'}$, the pair to node N_0. Once the optimized location of N_0 is found, the location of its pair node $N_{0'}$ is updated using the new location of N_0 plus the pitch. For inlet and outlet boundaries the submesh is generated by mirroring the half-moon sub-mesh with respect to the boundary face.

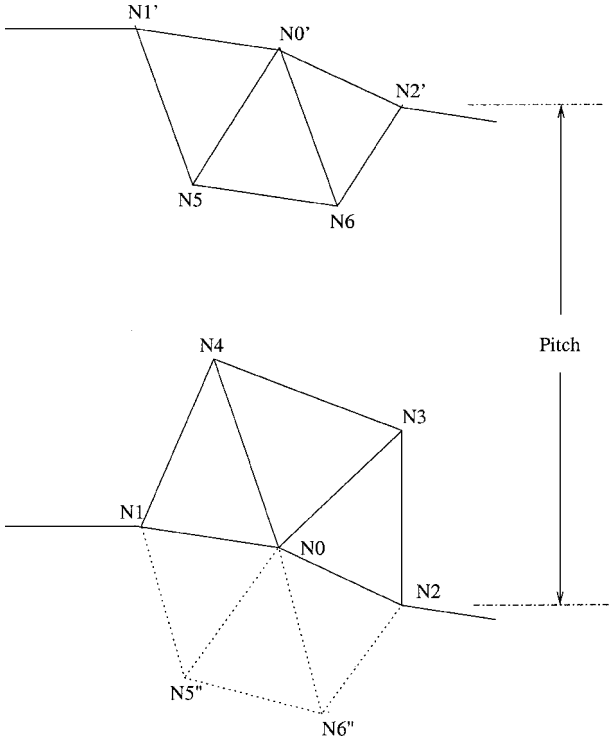
Figure 3 shows the types of submesh boundaries. The boundary of a submesh can be simple, nonsimple (tangled), or inverted simple. The boundary is considered positive if the nodes on the boundary are ordered in a counterclockwise direction. For the nonsimple case, the



a) Convex submesh



b) Star submesh



c) Periodic boundary N_1–N_0–N_2

Fig. 2 Submesh for node N_0.

edge segments cross each other and the cells overlap. The resulting mesh is unusable. The inverted simple submesh, shown in Fig. 3d, occurs in extreme cases. This mesh is unusable because its area is negative.

Centroid Smoothing

Centroid smoothing places the common node N_0 at the centroid of the valid region. The valid region for the common node is defined as the area where the node can be placed such that all of the resultant cells of the submesh have positive areas. Tangled edges are caused by placing the common node outside of the valid region. For the nonsimple case, a valid region cannot be defined.

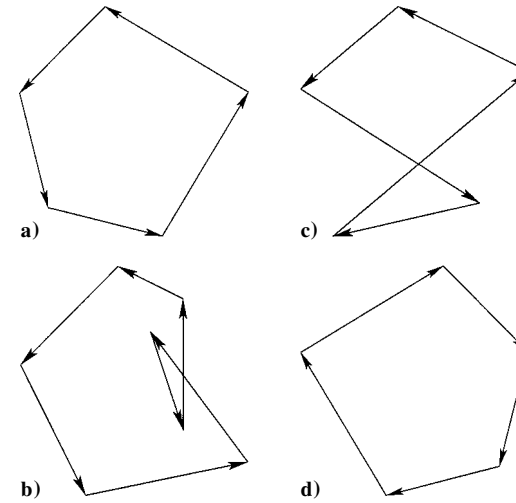


Fig. 3 Types of submesh boundaries: a) simple, b) non-simple due to the convex turns; and c) non-simple due to the reflexive turns, and d) inverted simple.

The valid region depends on the type of submesh. The common node of a convex submesh can be placed anywhere inside the boundary to produce a valid triangulation. Unlike the case of the convex submesh, only part of the interior of a star submesh is a valid region. The valid region of a star submesh is determined by the kernel, which is defined next.

The shadow is defined as the set of points from which a line can be drawn to any point in the interior without crossing the boundary. A shadow is associated with each concave corner. For a submesh with multiple concave corners, the kernel is defined as the intersection of the shadows.[13] For a star submesh with only one concave corner, the kernel is identical to the shadow.

To construct a shadow, half-spaces are defined with respect to each boundary segment that forms the concave corner. Figure 4a shows the half-space formed by the line running through the nodes N_1 and N_2. The second half-space is defined by the nodes N_2 and N_3, as shown in Fig. 4b. The intersection of the two half-spaces defines the shadow of the concave corner at N_2 as shown in Fig. 4c. There is a second concave corner in the submesh shown in Fig. 4, so that the shadow associated with the second corner has to be determined to obtain the kernel. The kernel of the submesh is shown in Fig. 5a.

Near a concave region with high curvature, such as at the airfoil leading or trailing edge, an intersection of shadows may not be found. In this case, the kernel is empty, as shown in Fig. 5b. In such a case, the valid location of the common node cannot be determined locally. In most cases, the smoothing of the neighboring submeshes changes the shape of the empty kernel submesh so that a valid region can be found in later iterations. The prolonged presence of the empty kernel during the iterations, however, indicates the limitations of mesh smoothing. A valid triangulation can be obtained for the case of the empty kernel shown in Fig. 5b by introducing a new node at one of the two shadows and placing the existing common node in the other shadow. Further details on node insertion are given in the next section.

Optimization-Based Smoothing

Optimization-based smoothing is a technique based on the steepest descent optimization method.[10] This smoothing method operates on the submesh and computes the new location of the internal node to increase the local minimum quality metric. The quality metric used throughout this paper is based on the ratio of the triangular cell area to the sum of the squared length of its edges. For a triangle cell with nodes at $\boldsymbol{X}^1$, $\boldsymbol{X}^2$, and $\boldsymbol{X}^3$, the metric τ is defined by

$$\tau = C\frac{(\boldsymbol{E}_{12} \times \boldsymbol{E}_{13}) \cdot \boldsymbol{e}_n}{\boldsymbol{E}_{12} \cdot \boldsymbol{E}_{12} + \boldsymbol{E}_{13} \cdot \boldsymbol{E}_{13} + \boldsymbol{E}_{23} \cdot \boldsymbol{E}_{23}} \tag{1}$$

where C is a constant and $\boldsymbol{E}_{ij}$ is the edge vector from $\boldsymbol{X}^i$ to $\boldsymbol{X}^j$. The unit vector $\boldsymbol{e}_n$ is normal to the triangle cell. Consequently, the vector

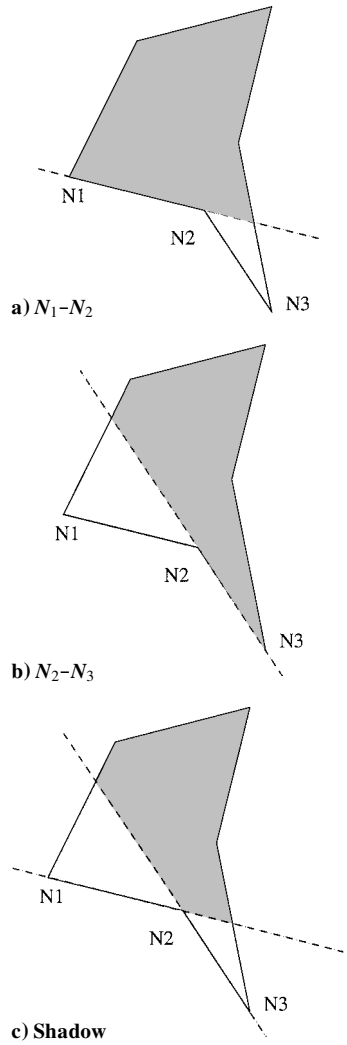


Fig. 4 Construction of shadow at concave corner N_2.

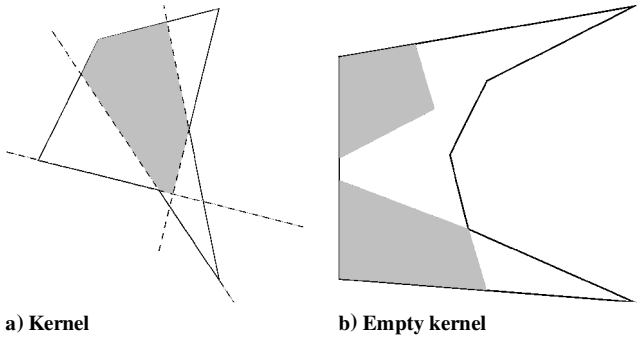


Fig. 5 Types of kernels.

$\boldsymbol{E}_{12} \times \boldsymbol{E}_{13}$ has the same direction and orientation as the unit vector $\boldsymbol{e}_n$ if the nodes of the cell T_{123} are arranged in a counterclockwise order. For an inverted cell, the numerator should be negative. The metric τ is bounded,

$$-1 \le \tau(T) \le 1 \tag{2}$$

by setting $C = 2\sqrt{3}$. The metric for an equilateral triangle is either -1 or 1 depending on the orientation.

With a submesh about a node N^i, the new location $\hat{\boldsymbol{X}}^i$ of the node N^i is determined by

$$\hat{\boldsymbol{X}}^i = \boldsymbol{X}^i + \gamma^i \boldsymbol{G}^i \tag{3}$$

where $\boldsymbol{G}^i$ is the gradient of the metric at node N^i and γ^i is a factor to control the magnitude of the node displacement along $\boldsymbol{G}^i$. The new location of the node changes the metrics of the cells. The new minimum of the quality metric should be greater than the old minimum, that is,

$$\min\left[\hat{\tau}_j(\hat{\boldsymbol{X}}^i)\right] > \min\left[\tau_j(\boldsymbol{X}^i)\right] \tag{4}$$

Note that the subscripts are associated with the cells and the superscripts are associated with the nodes.

The major task in the optimization is the gradient computation, which requires repeated computations of the quality measures. A cost-effective way to compute the gradient is desired. The gradient computation based on the divergence theorem is used to approximate the uphill direction of the quality metric at the common node. The method is applicable only for simple submesh boundaries.

A closed path connecting the centroids of adjacent cells in a counterclockwise direction is defined. This closed path generates a centroid dual cell, as shown in Fig. 6. The divergence theorem applied for τ on the centroid dual cells,

$$\int_A \nabla\tau \, \mathrm{d}A = \oint_{\partial A} \tau \boldsymbol{n} \, \mathrm{d}l \tag{5}$$

is used to approximate the gradient at the common node N^i,

$$\nabla\tau^i \approx \boldsymbol{G}^i = \frac{1}{A} \sum_{j=1}^{m^i} \tau_{j'} \boldsymbol{n}_{j'} l_{j'} \tag{6}$$

where A is the interior area of the integral path, j' is the average between j and $j+1$, and m^i is the number of cells of the submesh i. Herein, the quality metric for each cell is assumed to be associated with the centroid of the cell. The gradient is first-order accurate and, thus, exact for linearly varying τ. The procedure for calculating the gradient is similar to flux summation in the finite volume method.[14]

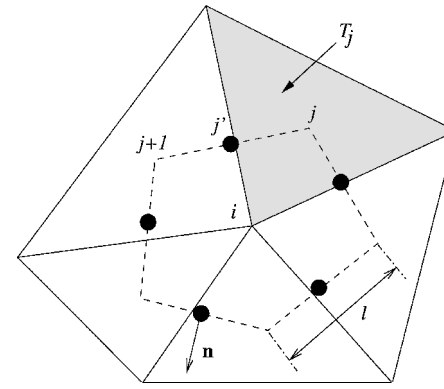


Fig. 6 Centroid dual cell.

If the centroid path cannot be defined, as happens with tangled and inverted submesh cases, the gradient is computed using the perturbed τ (Ref. 9). For each cell T_j of the submesh, the gradient is approximated by

$$\nabla\tau_j \approx \frac{\tau_j^{\delta x} - \tau_j}{\delta x}\boldsymbol{e}_x + \frac{\tau_j^{\delta y} - \tau_j}{\delta y}\boldsymbol{e}_y \tag{7}$$

where $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ are the unit vectors in the x and y directions. The perturbed metrics $\tau^{\delta x}$ and $\tau^{\delta y}$ are computed with the location of the common node perturbed by δx and δy, respectively. The magnitudes of δx and δy are adjusted to the local scale, that is, 1% of the breadth of the submesh. The gradient at the common node $\boldsymbol{G}^i$ is set to

$$\boldsymbol{G}^i = \nabla\tau_{j*} \tag{8}$$

where cell $j*$ has the minimum quality metric value,

$$\tau_{j*} = \min\{\tau_j\}, \qquad j = 1, \ldots, m^i \tag{9}$$

Once the uphill direction, $\boldsymbol{G}^i$, of the quality metric is determined, the displacement of the common node along the uphill direction is specified by $\gamma^i \boldsymbol{G}^i$. Two procedures to compute γ^i are presented. The selection of the procedure depends on the type of the submesh.

For a simple submesh case, centroid smoothing sets the initial location of the common node at the centroid of the valid region. The centroid is not the optimal point in most cases, but fairly close to it. Thus, the search for the optimal node location is limited to a small region near the current position.

The search for the new node location is accomplished as follows. A line is drawn from the current node location in the uphill direction given by Eq. (6). The line intersects the boundary of the valid region. The length between the current location of the node and the intersection point determine the maximum $d\boldsymbol{X}^i$. An initial γ^i can be set to a value such that $\gamma^i \boldsymbol{G}^i$ becomes a fraction of the maximum $d\boldsymbol{X}^i$. In the present implementation the value of the fraction is 10%.

If the initial translation of the node along $\boldsymbol{G}^i$ does not improve the minimum τ, the search distance is reduced by halving γ^i. The procedure repeats until the new minimum is greater than the current minimum. The number of trials, however, is limited to a fixed number of iterations.

A different methodology is used to calculate γ^i for nonsimple or inverted submeshes. The presence of a nonsimple or inverted submesh voids the validity of the mesh. Therefore, it is critical to find the proper new location of the common node, which will transform such a submesh to a valid simple submesh. Furthermore, when a submesh is not simple, the variations of the quality metrics appear to be extremely sensitive to the displacement magnitude of the common node.

A new quality metric for a cell T_j can be approximated using Taylor series (see Ref. 9):

$$\tau_j(\hat{\boldsymbol{X}}^i) = \tau_j(\boldsymbol{X}^i + \gamma^i \boldsymbol{G}^i) = \tau_j(\boldsymbol{X}^i) + \gamma^i \boldsymbol{G}^i \cdot \left.\frac{\partial \tau_j}{\partial \boldsymbol{X}}\right|_{X^i} + \text{HOT} \quad (10)$$

where $\hat{\boldsymbol{X}}^i$ is the new location of node N^i. When the gradient term is substituted by the gradient approximation $\boldsymbol{G}_j$ for a cell T_j, where $\boldsymbol{G}_j \equiv \nabla \tau_j$, Eq. (10) becomes

$$\tau_j(\hat{\boldsymbol{X}}^i) \approx \tau_j(\boldsymbol{X}^i) + \gamma^i \boldsymbol{G}^i \cdot \boldsymbol{G}_j \quad (11)$$

The value of the gradient $\nabla \tau_j$ is approximated by the perturbed quality metric, as shown in Eq. (7). For the cell T_{j*} with the current minimum τ_{j*}, Eq. (11) can be written as

$$\tau_{j*}(\hat{\boldsymbol{X}}^i) \approx \tau_{j*}(\boldsymbol{X}^i) + \gamma^i \boldsymbol{G}^i \cdot \boldsymbol{G}^i \quad (12)$$

where the second $\boldsymbol{G}^i$ term recalls the definition of the gradient for the cell with the minimum quality measure, as shown in Eq. (8).

Equation (12) shows that the metric of the cell with the old minimum τ_{j*} will always improve because the inner product of $\boldsymbol{G}^i$ and $\boldsymbol{G}^i$ is positive. The product $\boldsymbol{G}^i \cdot \boldsymbol{G}_j$ in Eq. (11), however, can be negative. When this occurs, the new τ_j decreases for a positive γ^i. Therefore, when the product is negative, the decreasing metric should be restricted to be equal to or greater than the improved value of the old minimum metric. For a submesh with m^i cells, this condition implies

$$\tau_{j*}(\hat{\boldsymbol{X}}^i) \le \tau(\hat{\boldsymbol{X}}^i), \qquad 1 \le j \le m^i, \qquad j \ne j* \quad (13)$$

which can be expanded to

$$\tau_{j*} + \gamma^i \boldsymbol{G}^i \cdot \boldsymbol{G}^i \le \tau_j + \gamma^i \boldsymbol{G}^i \cdot \boldsymbol{G}_j \quad (14)$$

Finally, when Eq. (14) is rearranged, the minimum γ^i is

$$\gamma^i = \min\left\{\frac{\tau_j - \tau_{j*}}{\boldsymbol{G}^i \cdot \boldsymbol{G}^i - \boldsymbol{G}^i \cdot \boldsymbol{G}_j}\right\}, \qquad 1 \le j \le m^i, \quad j \ne j* \quad (15)$$

where $\boldsymbol{G}^i \cdot \boldsymbol{G}_j$ is negative. For negative $\boldsymbol{G}^i \cdot \boldsymbol{G}_j$, Eq. (15) yields a positive γ^i. When $\boldsymbol{G}^i \cdot \boldsymbol{G}_j$ is positive, Eq. (11) shows that a positive γ^i is sufficient for the increase of the quality metric.

Note that the minimum γ^i can be close to zero in some extreme cases. In such a case, the magnitude of $\gamma^i \boldsymbol{G}^i$ is compared to the submesh length scale. If the magnitude of the displacement given by the $\gamma^i \boldsymbol{G}^i$ is smaller than a predefined threshold, the next smallest γ^i can be used instead.[9]

Edge Swapping and Node Insertion

As shown in the preceding section, there are limitations to using mesh smoothing for star submeshes. Specifically, when an empty kernel case occurs, mesh smoothing alone cannot guarantee a valid triangulation. Edge swapping and node insertion are, thus, introduced to overcome the limitation of the mesh smoothing and to further improve mesh quality.

Edge swapping is used to improve the quality of the unstructured mesh. Unlike the mesh smoothing methods, edge swapping alters the connectivity. The present study uses a slightly modified version of Lawson's[15] edge swapping method. The Lawson's edge swapping method compares the minimum angles of two adjacent triangle cells before and after the common edge is swapped. The edge is swapped only if the new minimum angle is greater than the old minimum angle. Otherwise, the current edge connectivity is kept.

The two-dimensional edge swapping technique is extended to the mapped mesh considered in this study in the following manner. The mapped mesh has a single set of connectivity information shared by all of the layers. Specifically, given an edge from the connectivity table, the corresponding edge can be found on every layer. Thus, two neighboring triangle cells that share the edge can also be formed on every layer. These two neighboring triangle cells form a "quad-tube" when they are stacked in the spanwise direction. The decision to swap the edge is then based on whether swapping improves the minimum measure among the quad-cells in the quad-tube.

Here Lawson's method[15] is modified by replacing the maximization of the minimum angle criteria with the quality measure τ. When the quality measure τ is used, edge swapping becomes consistent with mesh smoothing in the sense that both procedures maximize the minimum quality measure. Note that the quality measure τ includes information on the mesh angle values. The criteria based on the quality measure τ are more general than the criteria based on the maximization of the minimum angle because the former also include information on the cell area and edge length.

Mesh smoothing improves the mesh greatly, but the improvement is limited by the connectivity restriction between layers. Edge swapping, however, modifies the connectivity of the smoothed meshes. A swapped edge alters the submeshes associated with all four nodes of the quad-cell, and further mesh smoothing may become necessary. Because mesh smoothing and edge swapping are coupled, they can be applied alternatively until 1) edge swapping is no longer necessary and 2) the node movement and/or the variations in the quality measure are smaller than some given values.

When a source mesh is generated using a certain airfoil cross section, the connectivity of the resultant unstructured mesh is optimal with respect to the criteria of the mesh generator. For example, if Delaunay triangulation is used, the connectivity corresponds to the best max–min angle possible for the given distribution of nodes. When nodes are forced to move to new locations, as occurs with the mapping and smoothing procedures, the connectivity is no longer optimal. Edge swapping reduces the dependency of the overall mesh quality on the choice of the source mesh. As a result, the influence of the spanwise location of the source layer on the final global mesh quality is diminished.

To enhance mesh quality, node insertion is also employed. Recall that, for the empty kernel case, introducing a new node or Steiner point can remove the inverted cells. Furthermore, node insertion at a later stage of the mesh generation process makes it possible to start from a coarse source mesh. This reduces the work load for mesh smoothing and edge swapping. Once the mesh is converged, additional nodes can be inserted where the resolution of the mesh does not meet the predefined length scale.

The number of newly added nodes is controlled by calculating a quality measure or score for each cell and then sorting the cells based on these scores. Then a predetermined number of nodes is placed on the worst cells first. As is the case with edge swapping, node insertion also alters the connectivity of the mapped mesh. The

sorting procedure examines the score for each triangle cell in a triangle-tube that spans from hub to tip.

Although the circumcircle center of a candidate triangle cell is often the preferred site for a new node,[16] here, for simplicity, the centroid of a triangle cell is used for a new node location. Only a fraction of the total number of new nodes are added at each step. The whole procedure repeats until all of the intended number of nodes are added and the mesh smoothing and edge swapping are completed.

Results

The mapped mesh generation is tested on an axial turbine rotor blade shown in Fig. 7. The variation of the airfoil cross sections at 20% span increments is shown in Fig. 8. Mesh generation for this airfoil is challenging because of the large radial variation in shape and size. What makes it even more difficult is the extreme variation of the stagger angle, which is approximately 88 deg.

The source mesh is generated at the hub. Figure 9 shows the source mesh where only the unstructured region is displayed. The unstructured portion of the two-dimensional layer contains 480 nodes, excluding the common nodes between the O grid and the unstructured region. The mesh is rather coarse, and additional nodes are to be added after the mapping and smoothing procedures are performed. The O grid has 120 nodes along the airfoil and 10 nodes normal to the airfoil.

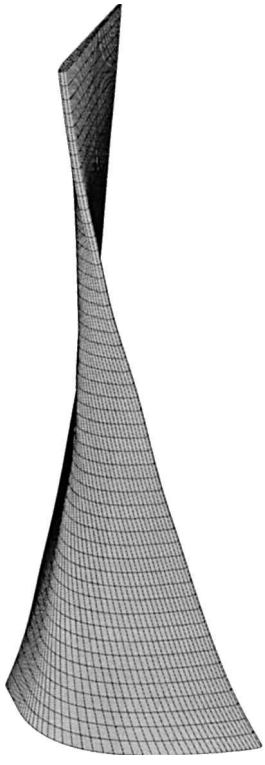


Fig. 7 Test case airfoil.

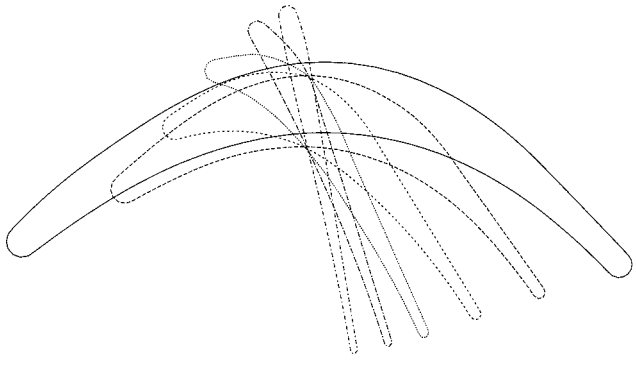


Fig. 8 Variation of airfoil cross sections.

Table 1 Minimum and maximum angle

Case	Minimum angle, deg	Maximum angle, deg
MS	n/a	179.8
MS + ES	3.5	150.0
MS + ES + NI	11.9	147.3

Table 2 Minimum of the quality metric τ

Case	Minimum τ
Mapping only	−0.6994
MS	−0.0643
MS + ES	0.2137
MS + ES + NI	0.2512

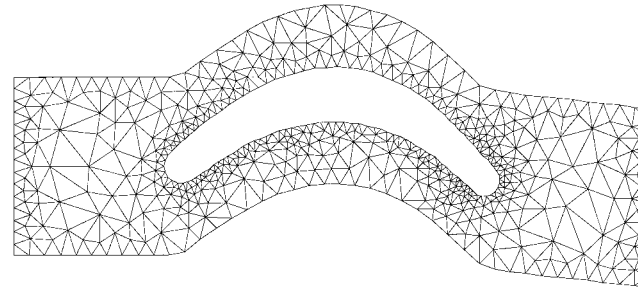


Fig. 9 Source mesh at the hub.

Mesh smoothing alone failed to eliminate the inverted cells on the layers close to the tip. These inverted cells resulted due to the significant change of the airfoil profile from hub to tip. Note that in this case the source layer was located at the hub. The optimal connectivity for the source mesh caused too much restriction of the mesh at the tip. Edge swapping was, thus, necessary to remove the constraints due to the biased connectivity of the source mesh.

The first mapped layer was generated at 10% span from the hub. After mapping and mesh smoothing, edge swapping was done for the hub and the layer at 10% span simultaneously. By advancing the layers by 10% span each time, the connectivity constraints were not as severe as in the initial attempt when the base mesh at the hub was mapped to the target mesh at the tip. This procedure of adding one new layer each time and applying the edge swapping was repeated until the tip layer was reached. As a result, all of the target meshes were successfully mapped and smoothed while connectivity was continuously updated. Note that the 10% span spacing was arbitrarily chosen and that this spacing was not the actual spacing between the layers of the final mesh.

The variation in cell size at the interface between the O grid and the unstructured region was controlled by forcing the unstructured cells on the interface to form equilateral triangles. On the tip layer, the node distribution was not dense enough because the area the nodes had to cover increased significantly compared to the hub layer. The abrupt changes in cell size became problematic in the transition region from the O grid to the unstructured region of the tip layer. To correct this problem, new nodes were added and smoothing and edge swapping were applied. The results are summarized in Tables 1 and 2. Table 1 shows the improvement in the minimum angle, and Table 2 shows the improvement in the minimum quality measure. Because the mesh optimization was based on the quality metric τ rather than on the minimum angle, the variation of τ shown in Table 2 provides a better image of the mesh improvement than the values of the minimum angle shown in Table 1. Note that the minimum angle for mesh smoothing is not defined due to the presence of the inverted cell.

The distribution of the quality measure τ as a function of the spanwise location is shown in Fig. 10. Note that τ is 1 for an equilateral triangle and 0 for a collapsed triangle. A larger number of cells exhibited a low τ on the hub and tip layers than on the layers near the midspan. Note the similarity of the minimum τ distribution

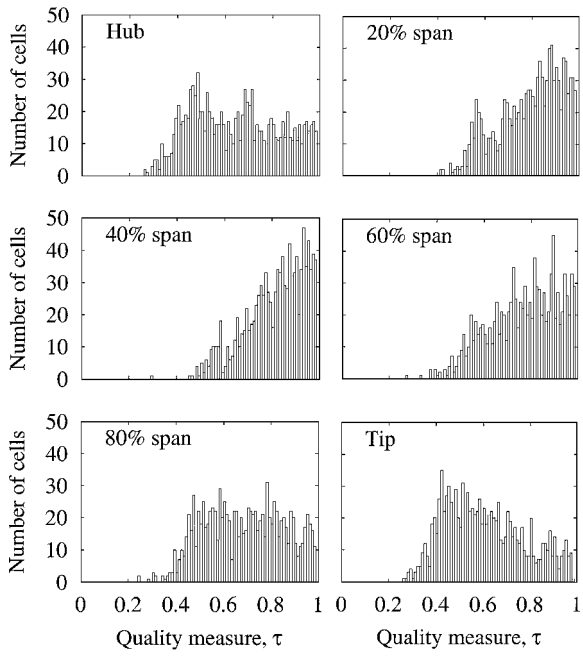


Fig. 10 Number of cells vs quality measure τ.

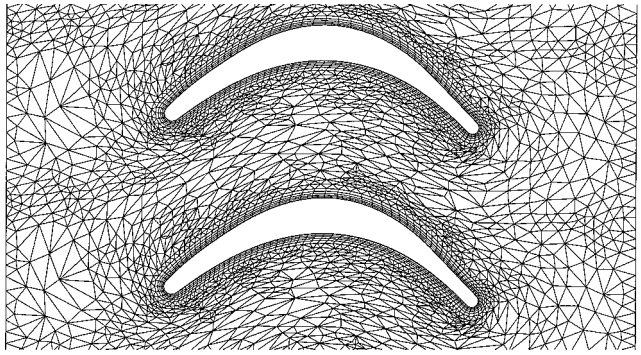


Fig. 11 Hub layer.

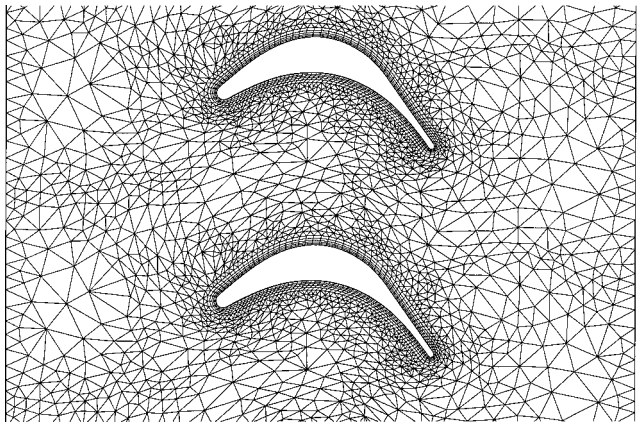


Fig. 12 Layer at 20% span from the hub.

for all of the layers, which proves that edge swapping was effective in eliminating the dependency of the quality of the target meshes on the source mesh. Mesh smoothing maximized the minimum quality measure for every layer. Edge swapping corrected the initial connectivity to an unbiased one that satisfied all of the layers.

Figures 11–16 show the grid at five spanwise locations. Node insertion (NI) removed the abrupt changes of cell size. By adding 400 new nodes, the total number of nodes in the unstructured mesh increased to 880. Additional mesh smoothing (MS) and edge swapping

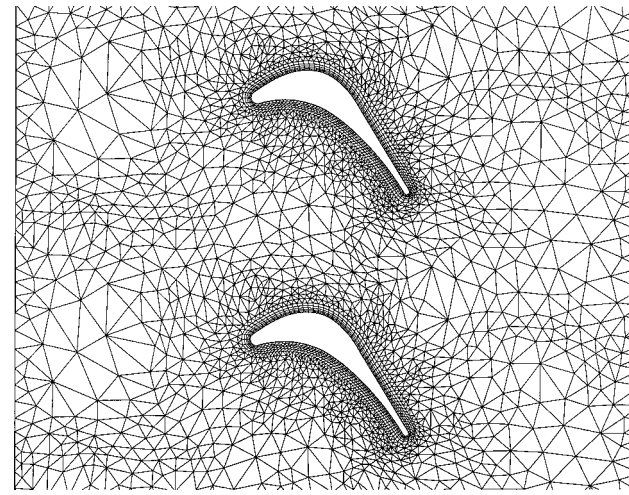


Fig. 13 Layer at 40% span from the hub.

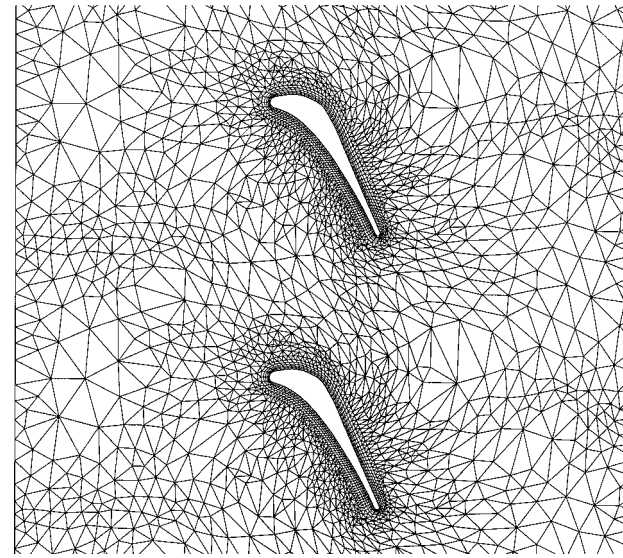


Fig. 14 Layer at 60% span from the hub.

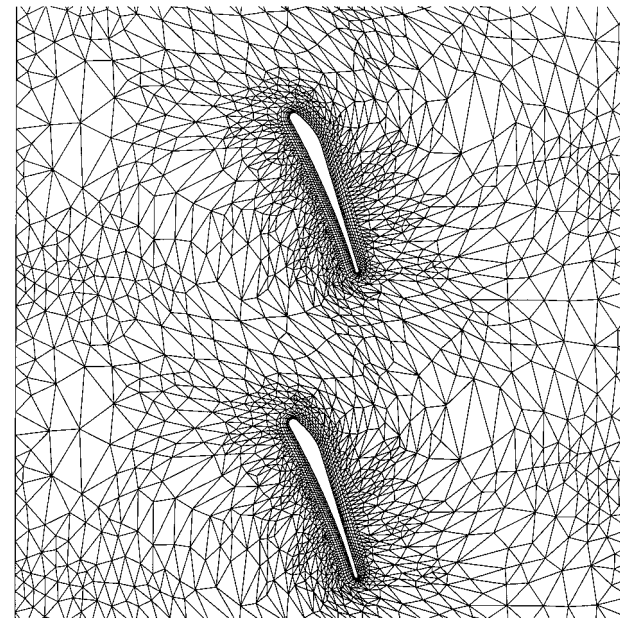


Fig. 15 Layer at 80% span from the hub.

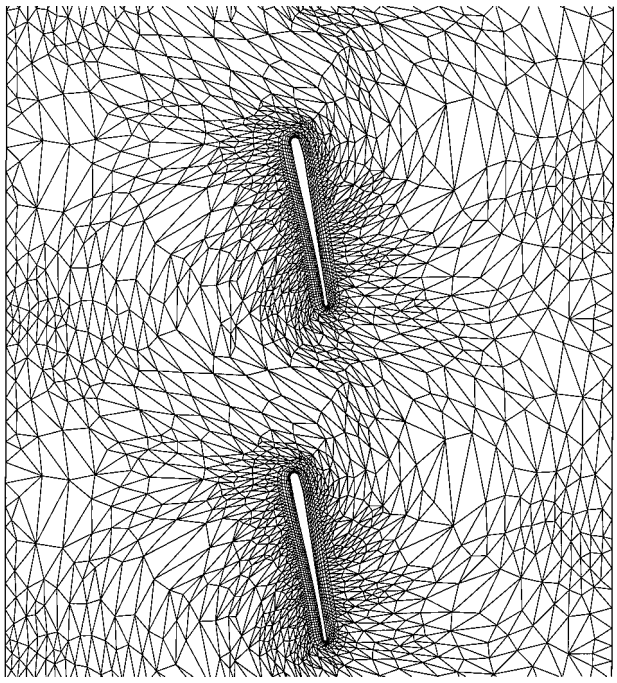


Fig. 16 Tip layer.

(ES) increased the minimum angle to 11.9 deg from 3.5 deg. The computational time necessary to build a three-dimensional hybrid mesh containing 106,080 nodes was approximately 5 min on a 600-MHz personal computer running Linux.

The current ES method was applied to the interior edges only. ES, however, could also be applied to the periodic boundaries using the ghost-cell approach. This could further enhance mesh quality.

Conclusions

The hybrid mesh generation algorithm for turbomachinery airfoils developed here took advantage of the flexibility of the two-dimensional unstructured mesh and the efficiency of the structured mesh. The hybrid grid with topologically identical layers is ideal for parallel flow computation. The grid generation method is robust and can handle efficiently challenging airfoils, as shown in the "Results" section.

The method is based on two-dimensional algorithms, which include a mapping technique and an optimization-based smoothing method. Further enhancement of grid quality was possible by ES and NI. ES reduced the dependency of mesh quality on the choice of the source mesh. MS was extended to include the nodes on the periodic boundaries by using the ghost-cell approach. This approach relaxed the restrictions imposed to the nodes on the periodic boundaries, and as a consequence, mesh quality was improved.

Lawson's method[15] was extended in two ways. First, the maximization of the minimum angle was replaced by a more general quality measure. The purpose of this modification was to provide a unified quality measure for smoothing and edge swapping. Second, Lawson's method was extended from a two-dimensional to a three-dimensional algorithm by applying it to a quad-tube. The decision to swap the edges of the tube was based on whether swapping improved the minimum quality measure of the quad-cells in the quad-tube. The methodology developed herein provides an integrated system for mapping, smoothing, ES, and NI.

Acknowledgment

This research was supported by the Texas Advanced Research Program under Grant 000512-0116-1999.

References

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

[2]Pizadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layers Method," *AIAA Journal*, Vol. 24, No. 1, 1996, pp. 43–49.

[3]Owen, S., "A Survey of Unstructured Mesh Generation Technology," *Proceedings of the 7th International Meshing Roundtable*, Sandia National Lab., NM, 1998, pp. 239–267.

[4]Mingwu, L., Benzley, S., Sjaardema, G., and Tautges, T., "A Multiple Source and Target Sweeping Method for Generating All Hexahedral Finite Element Meshes," *Proceedings of the 5th International Meshing Roundtable*, Sandia National Lab., NM, 1996, pp. 217–225.

[5]Staten, M., Canann, S., and Owen, S., "BMSWEEP: Locating Interior Nodes During Sweeping," *Engineering with Computers*, Vol. 15, No. 3, 1999, pp. 212–218.

[6]Batina, J., "Unsteady Euler Airfoil Solution Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1381–1388.

[7]Bern, M. W., and Eppstein, D., "Mesh Generation and Optimal Triangulation," Technical Xerox Palo Alto Research Center, Rept. CSL-92-1, Palo Alto, CA, Jan. 1992.

[8]Zhou, T., and Shimada, K., "An Angle-Based Approach to Two-Dimensional Mesh Smoothing," *9th International Meshing Roundtable, Sandia National Laboratories*, Sandia National Lab., NM, 2000, pp. 373–384.

[9]Canann, S., Tristano, J., and Staten, M., "An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes," *Proceedings of the 7th International Meshing Roundtable*, Sandia National Lab., NM, 1998, pp. 479–494.

[10]Freitag, L., and Plassmann, P., "Local Optimization-based Simplicial Mesh Untangling and Improvement," *International Journal of Numerical Methods in Engineering*, Vol. 49, No. 1–2, 2000, pp. 109–125.

[11]Amenta, A. B., Bern, M. W., and Eppstein, D., "Optimal Point Placement for Mesh Smoothing," *Journal of Algorithms*, Vol. 30, No. 2, 1999, pp. 302–322.

[12]Shewchuk, J. R., "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," *Applied Computational Geometry: Towards Geometric Engineering*, edited by M. C. Lin and D. Manocha, Vol. 1148, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1996, pp. 203–222.

[13]Preparata, F. P., and Shamos, M. I., *Computational Geometry*, Texts and Monographs in Computer Science, Springer-Verlag, Berlin, 1985, pp. 299–306.

[14]Barth, T., "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier–Stokes Equations," von Karman Inst. Lecture Series 1994-05, Rhode Saint Genèse, Belgium, 1994, pp. 28–34.

[15]Lawson, C. L., "Software for C^1 Surface Interpolation," NASA CR-155047, Jet Propulsion Lab., California Inst. of Technology, Pasadena, CA, 1977.

[16]Holmes, D. G., and Snyder, D. D., "The Generation of Unstructured Triangular Meshes Using Delaunay Triangulation," *Numerical Grid Generation in Computational Fluid Mechanics*, Pineridge, Swansea, Wales, U.K., 1988, pp. 643–652.