AIAA JOURNAL
Vol. 39, No. 6, June 2001

New Strategy for Unstructured Mesh Generation

C. Y. Liu* and C. J. Hwang†
National Cheng-Kung University, Tainan 70101, Taiwan, Republic of China

A novel strategy, which includes the statements of the no-cavity and nonoverlapping conditions and a mesh generation approach, is presented to create unstructured triangular and tetrahedral meshes. For this strategy, the no-cavity and nonoverlapping conditions provide the theoretical basis for the mesh-generation procedure. In this procedure, a structured background grid is utilized to determine the distribution of characteristic lengths. According to this distribution, nodes are generated first. Then, a characteristic-range method and a hybrid method are proposed for connecting these nodes. By the use of a mesh-removal process, the final mesh is obtained. To illustrate the procedure, a two-dimensional domain with four-element airfoils is considered. For three-dimensional domains, a postprocessing is introduced to improve the mesh quality. To evaluate the strategy, two concentric cubes, a coaxial annual pipe, and two concentric spheres are considered. For the flow region around the ONERA M6 wing, the hybrid method with postprocessing is utilized, and the mesh quality is compared to those previously attained. To demonstrate this strategy further, a mesh is generated for the flow region around an airplane.

Introduction

TWO major unstructured mesh-generation methods, the advancing-front method[1–3] and the Delaunay triangulation method (see Refs. 4–6), are currently available. For the advancing-front method, the boundary of domain is chosen as the initial front first. Then, a new element is generated from the initial front into the empty domain, and the intersections between this new element and the existing elements are checked. After the new element has been formed, the front is updated and is advanced further into the domain. This procedure is repeated and operated on every updated front until the entire domain is meshed. It is known that the boundary integrity is guaranteed and that the point placement strategy results in high-quality meshes. However, the intersection checking technique is expensive.[7] In addition, difficulties may be encountered in regions where the fronts are merged.[8] The Delaunay triangulation method adopts the empty circumcircle/circumsphere property from the computational geometry.[7] When new points are sequentially added to an artificially constructed convex domain, the Delaunay triangulation is operated until the characteristic length criterion is satisfied. Because of its simplicity, the Delaunay triangulation method is more efficient than the advancing-front method. However, the boundary recovery is required. As mentioned in Ref. 7, the main disadvantages of Delaunay triangulation techniques relate to their inability to guarantee boundary integrity. The Delaunay triangulation has the property of maximize minimum angle (max–min) triangulation in two-dimensional cases. However, it could generate sliver tetrahedrons in three-dimensional cases.

To improve these methods, several techniques have been presented. For example, some useful data structures have been used to perform efficiently the necessary search operations during the generation of unstructured grids.[9] To reduce some of the disadvantages of the Delaunay method, an advancing-front Delaunay algorithm has been developed.[10,11] This algorithm has the efficiency of the Delaunay method and provides high-quality meshes usually associated with the advancing-front technique. However, it inherits the boundary integrity difficulties of the Delaunay method.[8] Another technique, which can be used to recover the boundary mesh or to improve the mesh quality, is the swapping algorithm.[7,8,12] When this algorithm is used, convergence to the globally optimum max–min or Delaunay triangulation is assured in two dimensions, but a similar result does not hold in three dimensions.[7] The modified-octree technique,[13] which makes use of octants divided system to separate the domain, is another unstructured mesh-generation algorithm. This algorithm is efficient and robust, but the resulting cell distribution near the boundary is irregular.

In this paper, a new strategy for generating unstructured meshes is presented. First, the no-cavity and nonoverlapping conditions are introduced. Then, a mesh-generation approach is developed. Unlike previous unstructured mesh-generation methods, several overlapping elements are created to satisfy the no-cavity condition. A mesh-removal process is utilized to accomplish the nonoverlapping condition, so that the triangles or tetrahedrons in the final mesh do not overlap each other. A two-dimensional domain with four-element airfoils is adopted to illustrate the present mesh-generation procedure. Because there is no need to check the intersections or to judge the combination of meshes for swapping, the efficiency of the present approach increases greatly, especially for domains with complex configurations. When the postprocessing is operated, bad elements such as the sliver tetrahedrons can be removed. Two concentric cubes, a coaxial annual pipe, and two concentric spheres are adopted to evaluate the present strategy. For the flow region around the ONERA M6 wing, tetrahedral meshes are created. When the present results are compared with those given in Refs. 14 and 15, the mesh quality is studied. To further demonstrate the capability of the present strategy, unstructured tetrahedrons for the flow region around an airplane are generated, and histograms of the mesh qualities are presented.

No-Cavity and Nonoverlapping Conditions

Before descriptions of the no-cavity and nonoverlapping conditions are given, the meanings of the effective edge (segment) or effective face (triangle) and the number function $N(i)$ are explained.

1) For a boundary segment/triangle, there is only one effective edge/face, whose normal direction is toward the computational domain.

2) For a nonboundary segment/triangle, there are two effective edges/faces, whose normal directions are opposite to each other.

3) The number function $N(i)$ represents that there are $N(i)$ triangles or tetrahedrons on the effective edge or effective face i. For example, two triangles or tetrahedrons are located on the effective edge or effective face i, if $N(i)$ is equal to 2.

No-Cavity Condition

If there is one cavity in the domain, some of the effective edges/faces associated with this cavity will have a null value of $N(i)$. In other words, there will be no cavity in the whole domain if

Received 9 December 1998; revision received 2 December 1999; accepted for publication 9 December 2000.

*Graduate Student, Department of Aeronautics and Astronautics; currently Senior Engineer, R&D Division, Core Tech System Co., Ltd., 6F, No. 16, Sec. 2, Jing-Guo Road, Hsinchu 300, Taiwan, Republic of China.

†Professor, Department of Aeronautics and Astronautics. Senior Member AIAA.

the value of $N(i)$ for every effective edge/face is not equal to zero. This statement can be expressed in the following form:

$$\forall i,\ N(i) \geq 1 \Rightarrow \text{no cavity in the domain} \tag{1}$$

It is obvious that the no-cavity condition $[\forall i,\ N(i) \geq 1]$ is the sufficient one.

Nonoverlapping Condition

Necessary Condition

If more than one triangle/tetrahedron is simultaneously placed on one effective edge/face, these triangles or tetrahedrons will overlap each other. In other words, the value of $N(i)$ for each effective edge/face is less than or equal to one, if all of the triangles or tetrahedrons in the domain do not overlap each other. Combined with the no-cavity condition, this statement is expressed as follows:

$$\text{nonoverlap} \Rightarrow \forall i,\ N(i) = 1 \tag{2}$$

Sufficient Condition

If some elements in the domain overlap, at least one effective edge/face, on which more than one triangle/tetrahedron is placed, will be observed. This statement can be expressed as follows:

$$\text{overlap} \Rightarrow \exists i,\ N(i) \geq 2 \tag{3}$$

To evaluate statement (3), two triangles/tetrahedrons A_1 and B_1, which overlap each other, are considered. If A_1 and B_1 are placed on the same effective edge/face, statement (3) is established. Otherwise, a set of triangles/tetrahedrons, $\{A_i \mid i = 1, \ldots, n\}$, can be obtained to cover the region occupied by B_1. If an effective edge/face coexists among A_i and B_1, statement (3) is satisfied. If not, a set of triangles/tetrahedrons, $\{B_j \mid j = 1, \ldots, m\}$, is obtained to cover the space occupied by $\{A_i\}$, and statement (3) is checked again. If there is still no coexisting effective edge/face among A_i and B_j, the preceding procedure is repeated. For a finite domain, the aforementioned search procedure will stop on the boundary, and at least one boundary effective edge/face with $N(i) \geq 2$ will be detected. That is, statement (3) is established. The following expression, which is equivalent to statement (3), is written as

$$\forall i,\ N(i) \leq 1 \Rightarrow \text{nonoverlap} \tag{4}$$

Combined with the no-cavity condition, statement (4) is expressed as follows:

$$\forall i,\ N(i) = 1 \Rightarrow \text{nonoverlap} \tag{5}$$

From statements (2) and (5), the nonoverlapping condition ($\forall i$, $N(i) = 1$) is not only necessary but also sufficient.

Mesh-Generation Approach

According to the no-cavity and nonoverlapping conditions, a mesh generation approach is developed by following the procedure given next sequentially:

1) Construct a background grid.
2) Create a boundary mesh.
3) Utilize a node-placement technique to generate nodes.
4) Use a characteristic-range or a hybrid method to connect the nodes into a mesh.
5) Impose the no-cavity condition.
6) Impose the nonoverlapping condition.
7) Use a postprocessing to improve the mesh quality (only for three-dimensional cases).

Steps 1–5 and 7 will be described in the following subsections. First, a mesh-removal process, which is utilized to impose the nonoverlapping condition in the step 6, is discussed. According to the distribution of radii ratio,[16] a heap list is used to sort the elements. The element with the lowest value of radii ratio is deleted. Then, the linked lists are introduced, and the elements having effective edges/faces with $N(i) = 0$ are removed to satisfy the no-cavity condition. If the cancellation of an element causes the boundary elements to be deleted, this element is called a nonremovable element. Under the no-cavity condition, this nonremovable element and the related ones, which are just canceled, must be recovered. In other words, there is no cavity in the whole domain during removal of elements. According to the heap list, the preceding mesh-removal process is repeated and operated on every element until no element can be removed. Then, the nonoverlapping condition is examined. If this condition is satisfied, the mesh-generation procedure is stopped, and a final mesh is obtained. Otherwise, some elements in the domain overlap and couple together. Because the coupling may induce a cavity during removal of the elements, a decoupling operation is necessary. For this operation, only the specified element is removed, and several new elements are patched up around the edges/faces where the values of the number functions are equal to zero. Once the decoupling operation is done, the aforementioned mesh-removal process is applied to cancel the overlapping elements. This procedure is repeated until the nonoverlapping condition is satisfied. Although the decoupling operation is time consuming, numerical experiments indicate that the mesh couplings do not occur when the present characteristic-range or the hybrid mesh-connectivity method is used.

Two-Dimensional Mesh Generation

To illustrate the two-dimensional unstructured mesh-generation procedure, an example of four-element airfoils[17] is presented. The accumulated CPU time T, distributions, and numbers of nodes/segments/triangles at the aforementioned steps 1–6 are indicated in Fig. 1.

Background Grid and Boundary Mesh

A structured background grid (Fig. 1a) is adopted. When a Poisson equation (see Ref. 18) is solved, the distribution of characteristic lengths for this background grid is obtained. According to this distribution and a search/interpolation procedure,[18] the boundary curves are decomposed into segments, which are specified counterclockwise for external boundaries and clockwise for internal boundaries. The distribution of boundary segments for the four-element airfoils is shown in Fig. 1b.

Generation of Nodes

A large rectangle, which contains the computational domain and related bodies, is selected. Every created node must be located inside this large rectangle. Based on the advancing-front concept, a node-placement procedure is used to generate the nodes. At the beginning, all of the boundary segments are designated to be active. This means that these segments are available for generating new nodes. The node-placement steps are described as follows:

1) Choose an active segment as the base. To obtain a better distribution of nodes, the segment with the smallest length is processed first.

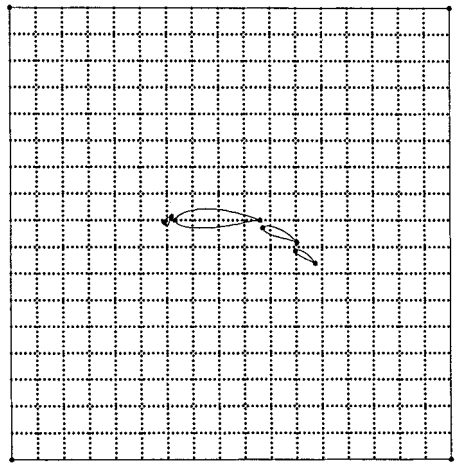
2) From the chosen base, a pseudopoint is located so that a regular element can be obtained. When the quadtrees are used, all nodes within a specified range are searched. This range is bounded by a circle, whose center is located at the pseudopoint and the radius is $\alpha(0.6$–$0.7)$ times the characteristic length at the pseudopoint.

3) If no node is found, the pseudopoint becomes a new node, and it is to be connected with the nodes of the chosen base to form the new segments. When the chosen base is replaced with the new segments, the active set is updated. If only one node is found, this node could be connected with the nodes on the chosen base to form the new segments. When the chosen base and the old active segments related to the node founded with the new segments are replaced, the active set is renewed. If two or more nodes are found, the chosen base is removed from the active set.

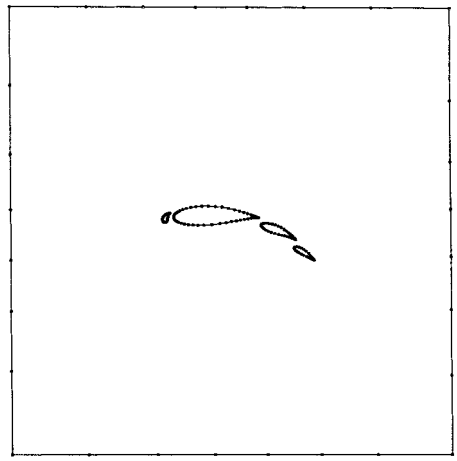
The preceding node-placement process is repeated until the active set becomes empty. Unlike the approach given in Ref. 3, the present method does not need to check whether the node is located inside or outside the bodies in the selected rectangle. Therefore, the numerical effort can be reduced, especially for cases with complex domain configurations. As shown in Fig. 1c, it is observed that some of the created nodes are located inside the largest airfoil.

Mesh Connectivity

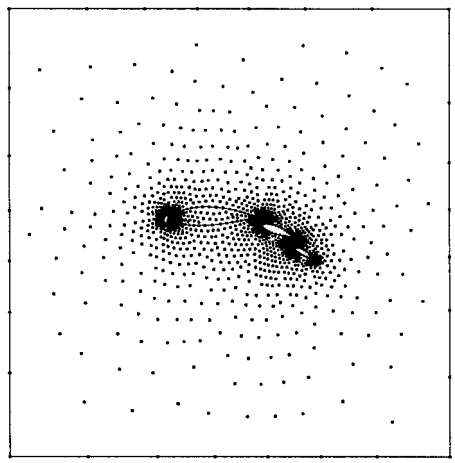
A characteristic-range method is introduced to create triangles. The characteristic range for a node, for example, node A, is defined as a circle, whose center is node A and whose radius is equal to



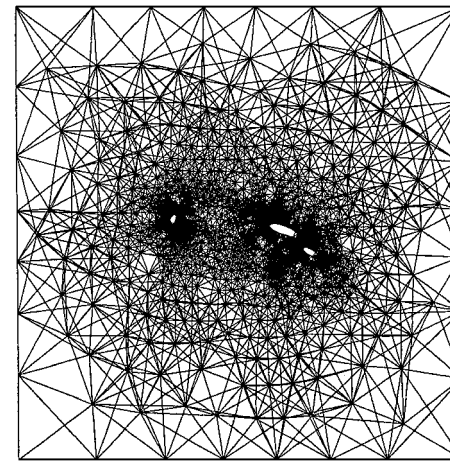
a) $T = 0.01$ s (0 node, 0 segment, 0 triangle)



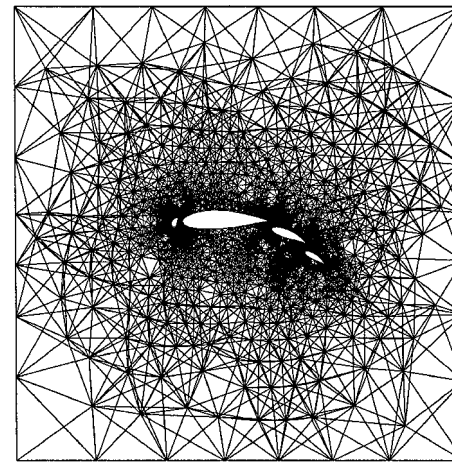
b) $T = 0.02$ s (161 nodes, 161 segments, 0 triangle)



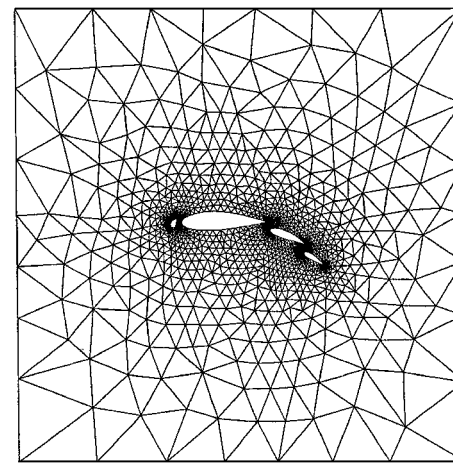
c) $T = 0.1$ s (1425 nodes, 161 segments, 0 triangle)



d) $T = 0.61$ s (1425 nodes, 10,188 segments, 20,303 triangles)



e) $T = 1.14$ s (1374 nodes, 9624 segments, 19,150 triangles)



f) $T = 1.42$ s (1374 nodes, 3970 segments, 2593 triangles)

Fig. 1 Mesh-generation procedure for four-element airfoils.

the characteristic length at node A. By the use of the quadtrees, the nodes are selected if their characteristic ranges intersect with that at node A. Then, these nodes are connected with A. For any two nodes, for example, B and C, the angle $\angle BAC$ is checked. If the angle is smaller than a threshold angle α, the longer side of $\overline{AB}$ and $\overline{AC}$ is deleted to reduce the number of segments. For example, $\overline{AB}$ is deleted if $\angle BAC$ is smaller than α and $\overline{AB}$ is longer than $\overline{AC}$. By considering the mesh quality and the required memory size, a specified value of the threshold angle α of 15 deg is adopted. The preceding procedure is repeated for all of the nodes to generate segments. When the linked lists are adopted, these segments are used to create all of the possible triangles, where no node is located inside. Because the created triangles are permitted to overlap each other or be located inside/across the bodies (Fig. 1d, four-element airfoils), the intersection-checking operations utilized in the advancing-front method are not needed.

Table 1 Final and maximum numbers of tetrahedrons and CPU times

Cases	Methods	Final no. of tetrahedrons	Maximum no. of tetrahedrons	CPU time, s
Concentric cubes	Characteristic range	6,892	226,233	11
	Hybrid	7,115	22,087	8
	Hybrid with postprocessing	6,914	22,087	54
Coaxial annular pipe	Characteristic range	4,721	209,411	10
	Hybrid	4,914	14,859	6
	Hybrid with postprocessing	4,718	14,859	53
Concentric spheres	Characteristic range	3,467	141,041	6
	Hybrid	3,656	13,242	5
	Hybrid with postprocessing	3,483	13,242	46
ONERA M6 wing	Hybrid with postprocessing	283,670	808,766	1,412
Airplane	Hybrid with postprocessing	913,447	1,931,745	4,121

No-Cavity and Nonoverlapping Conditions

Before the no-cavity condition is examined, those boundary triangles with the effective edges, whose normal directions are not toward the computational domain, are deleted first. Then the linked lists are adopted, and those triangles having the effective edges with $N(i) = 0$ are canceled, so that the no-cavity condition is satisfied. For the domain with four-element airfoils, a temporary mesh (Fig. 1e) is obtained. As indicated in Fig. 1e, the numbers of nodes, segments, and triangles are less than the corresponding ones given in Fig. 1d. Finally, a mesh-removal process is adopted to satisfy the nonoverlapping condition. The details of this process are described in the step 6 of the aforementioned mesh-generation procedure. After the mesh-removal process is applied, a final mesh for the domain with four-element airfoils is obtained (Fig. 1f). When the results given in Fig. 1e are compared, the numbers of segments and triangles in Fig. 1f are significantly reduced. Because all of the boundary segments are retained during the generation/cancellation of the triangles, the boundary integrity is warranted after the no-cavity and nonoverlapping conditions are satisfied. No other techniques, such as the swapping algorithm, are needed to recover the boundary mesh.

Three-Dimensional Mesh Generation

Background Grid and Boundary Mesh

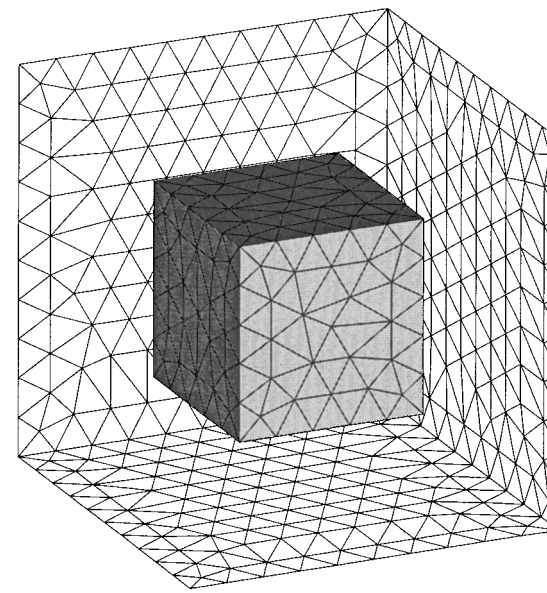
A three-dimensional structured background grid is adopted, and the distribution of characteristic lengths for this background grid is obtained by solving a three-dimensional Poisson equation (see Ref. 18). To construct the boundary mesh, the boundary surfaces are decomposed into several patches. For a patch, there are a series of cross-sectional profiles along a principal axis. By choosing X (distance along the principal axis) and S (arc length along the cross-sectional profile) as the surface coordinates, the coordinates x, y, and z of data points on a surface patch can be mapped to the two-parameter space in X and S. According to the preceding distribution of characteristic lengths and the two-dimensional mesh-generation approach, the triangulation is done in this parametric space first. Then, a transformation back to the physical space is performed to obtain the final triangulation on the boundary surface.

Generation of Nodes

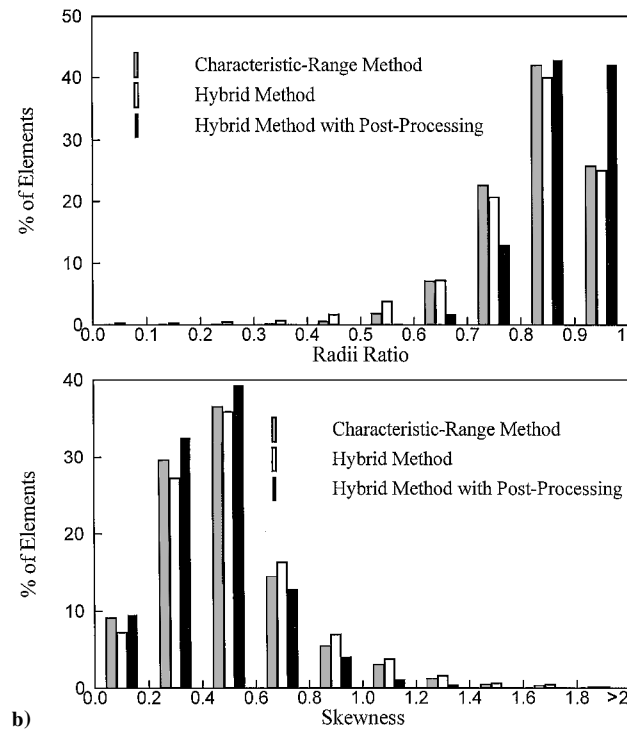
A large hexahedron, which contains the computational domain and related bodies, is selected. When all of the boundary triangles are allowed to be active, an active triangle is chosen as the base, and the octrees are used, the aforementioned two-dimensional node-placement procedure is utilized to generate the nodes. These nodes must be placed inside the selected hexahedron, but they may be located inside or outside the bodies. In other words, the generation of nodes is not restricted by the configurations of bodies.

Mesh Connectivity

To achieve the mesh connectivity, a characteristic-range method is presented. For a node, for example, node A, the characteristic range is defined as a sphere, whose center is node A and whose radius is equal to the characteristic length at node A. By the use of the octrees and a threshold angle α, for example, 15–30 deg, the aforementioned two-dimensional mesh-connectivity process is adopted to generate

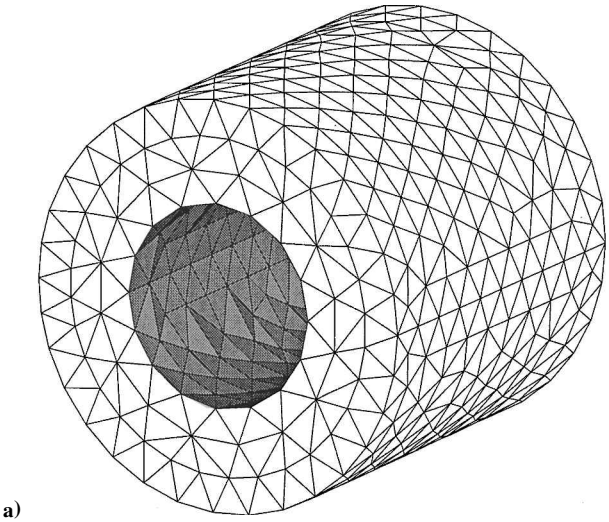


a)

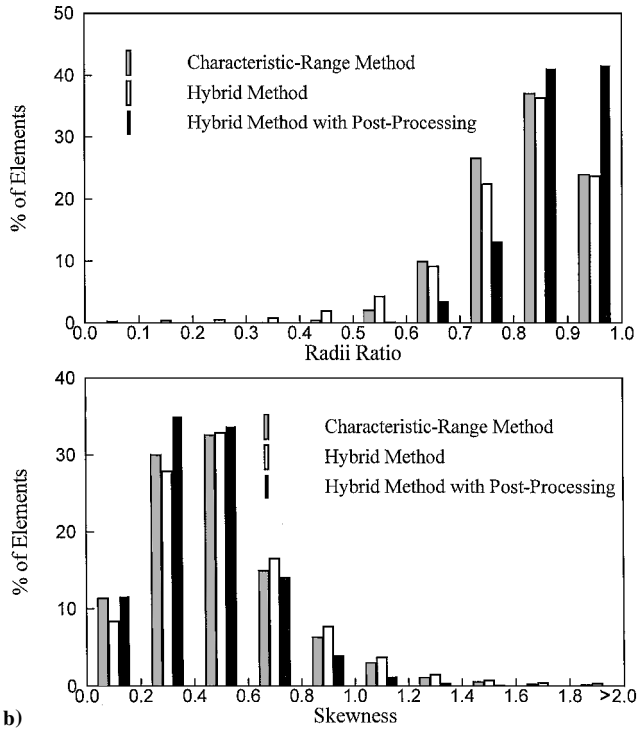


b)

Fig. 2 Concentric cubes: a) partial view of boundary mesh and b) histograms of mesh qualities.



a)



b)

Fig. 3 Coaxial annular pipe: a) partial view of boundary mesh and b) histograms of mesh qualities.

segments. These segments and the related linked lists are utilized to create all of the possible triangles. Then, these triangles are introduced to generate all of the possible tetrahedrons, where no node is located inside. It is similar to the two-dimensional case in that the created tetrahedrons are likely to overlap each other, and some of them may be located inside/across the bodies. Because the preceding characteristic-range method generates too many tetrahedrons in the three-dimensional domain, a hybrid method is developed. For this hybrid method, the empty circumsphere property is adopted to accomplish the mesh connectivity. When this empty circumsphere method causes the recovery problem on some boundary elements, the preceding characteristic-range method is utilized to create the tetrahedrons around those boundary triangles.

No-Cavity and Nonoverlapping Conditions

After the mesh connectivity is done, several overlapping tetrahedrons are created. Before the no-cavity condition is examined, those boundary tetrahedrons having the effective faces with normal directions being outward to the computational domain are deleted first. Then, the linked lists are introduced, and the tetrahedrons having the effective faces with $N(i)=0$ are canceled to satisfy the no-cavity condition. Finally, the nonoverlapping condition is satisfied by utilizing a mesh-removal process described in the step 6 of the mesh-generation procedure.

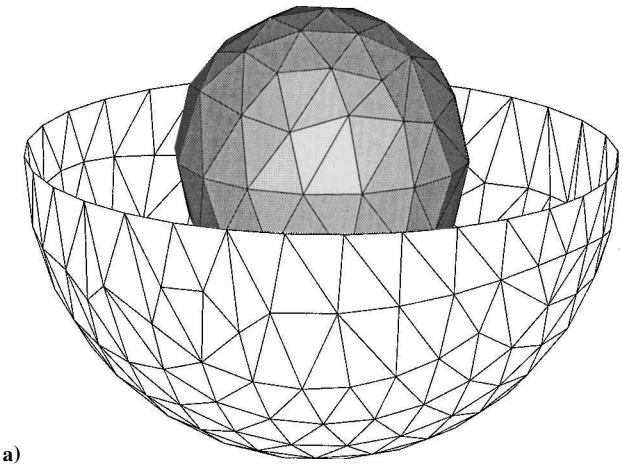
Postprocessing

When the hybrid method is used, several sliver tetrahedrons are created. To remove these sliver tetrahedrons and enhance the mesh quality, a postprocessing is developed.

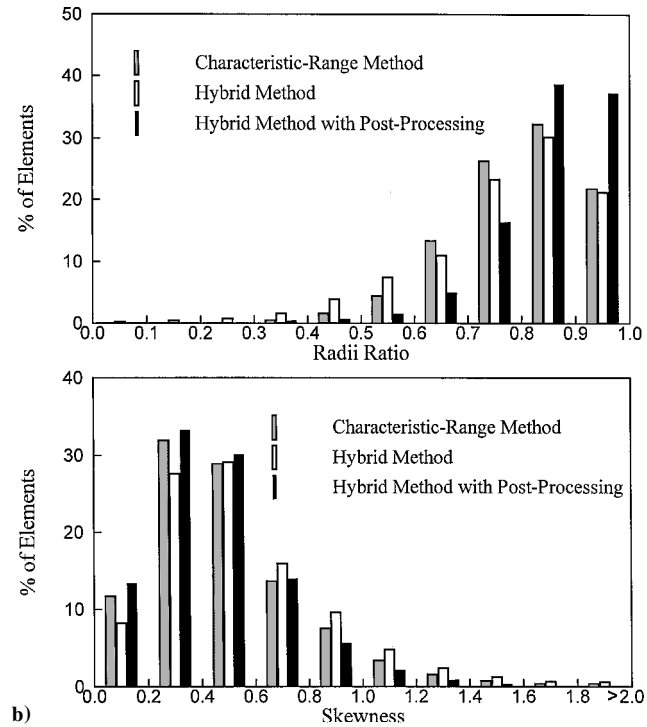
1) To identify tetrahedrons for postprocessing, a parameter $\tilde{R}$ is introduced and calculated by the following equation:

$$\tilde{R} = \bar{R} - \beta \times \sigma \tag{6}$$

where $\bar{R}$ and σ are the average and standard deviation of the radii ratios, respectively. The parameter β is an adjustable factor, and it is set to be 2 in this paper. The value of $\tilde{R}$ is between zero and one. For a tetrahedron A with the value of radii ratio less than or equal to the value of $\tilde{R}$, several new segments are generated by connecting the four nodes of A with nodes of the neighboring tetrahedrons. These segments are utilized to create all of the possible triangles, which are then used to generate all of the possible tetrahedrons. For these tetrahedrons, no node is located inside. It is obvious that the number of tetrahedrons in the mesh increases. To reduce the



a)



b)

Fig. 4 Concentric spheres: a) partial view of boundary mesh and b) histograms of mesh qualities.

computer memory, the maximum number of tetrahedrons obtained by the hybrid method without postprocessing is chosen as a threshold value. When the number of tetrahedrons in the mesh is equal to this threshold value, the preceding mesh connecting is stopped. Then, the aforementioned procedures, which are used to satisfy the no-cavity and nonoverlapping conditions, are utilized. The preceding process is applied to those tetrahedrons with values of radii ratio less than or equal to the value of $\tilde{R}$.

2) To further enhance the mesh quality, a Laplacian-like technique[15] is adopted to smooth the coordinates of nodes.

3) Steps 1 and 2 are processed alternately, and they are operated three times each for the present cases.

Results and Discussion

To evaluate the present strategy, three cases, which include two concentric cubes, a coaxial annular pipe, and two concentric spheres, are studied. The background grid with a uniform characteristic length of 0.1 is introduced. To examine and demonstrate the present strategy, the ONERA M6 wing and an airplane configuration are adopted. For all of the cases except the ONERA M6 wing, the radii ratio[16] and the skewness[14] are calculated to evaluate the mesh qualities. Table 1 lists the final and maximum numbers of tetrahedrons and the CPU times for implementation of the mesh-generation procedure. In this work, the operations for all of the cases are implemented on a compatible personal computer (Pentium III-450, 512-MB RAM).

Evaluation of the Present Mesh-Generation Strategy

Concentric Cubes

For the concentric cubes shown in Fig. 2a, the edge lengths of the larger and smaller cubes are equal to 1 and 0.5, respectively. After the boundary mesh is constructed (Fig. 2a), the region between these two cubes is decomposed into tetrahedrons. The histograms of the mesh qualities are shown in Fig. 2b. When the characteristic-range method is used, the CPU time is 11 s, and about 32 times extra tetrahedrons are created during the mesh connectivity. From histograms

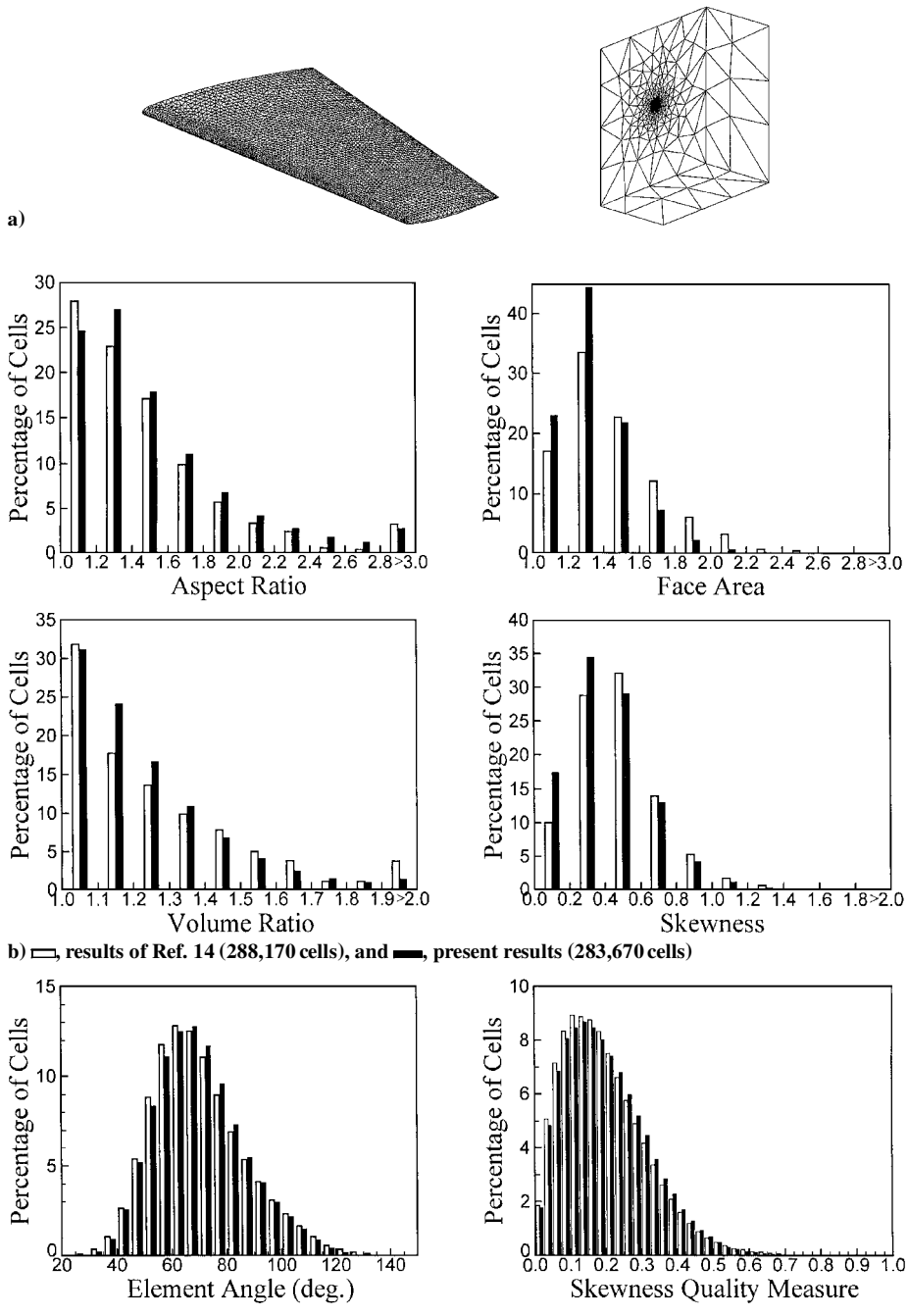


Fig. 5 ONERA M6 wing: a) partial view of boundary mesh, b) mesh quality comparison between the present results and those of Woodard et al.,[14] and c) mesh quality comparison between the present results and those of Marcum and Weatherill.[15]

of the mesh qualities, there are more than 60% of the tetrahedrons with their values of radii ratio greater than or equal to 0.8 and more than 70% of the elements with their values of skewness equal to or less than 0.6. When the hybrid method is applied, the ratio of the maximum number of tetrahedrons to the final number of tetrahedrons is reduced to 3.1. Also, the required CPU time decreases to 8 s. However, the mesh quality is worse than that obtained by the characteristic-range method (Fig. 2b). When the hybrid method is incorporated with postprocessing, the ratio of the maximum number of tetrahedrons to the final number of tetrahedrons is equal to 3.2. The postprocessing does not significantly affect the final number of tetrahedrons. Although the required CPU time increases to 54 s, the mesh quality is enhanced. Note that almost all tetrahedrons with values of radii ratio lower than 0.6 are removed (Fig. 2b). Compared with the characteristic-range method, the hybrid method with postprocessing utilizes less memory and provides better mesh quality at the expense of CPU time.

Coaxial Annular Pipe

In this case, the length of the pipe shown in Fig. 3a is equal to 1, and the diameters of two coaxial cylinders are set to be 1 and 0.5, individually. After a boundary mesh (Fig. 3a) is created, the region between these two cylinders is decomposed into tetrahedrons. The histograms of the mesh qualities are plotted in Fig. 3b. When the characteristic-range method is used, the final and maximum numbers of tetrahedrons are 4721 and 209,411, respectively. The CPU time is 10 s. It is the same as the preceding case in that the ratio of the maximum number of tetrahedrons to the final number of tetrahedrons is large. As shown in Fig. 3b, the values of radii ratio for most of the tetrahedrons are equal to or greater than 0.6, and there are more than 70% of the elements with their values of skewness equal to or less than 0.6. When the hybrid method is utilized, the final and maximum numbers of tetrahedrons are 4914 and 14,859, respectively, and the required CPU time is 6 s. Although the CPU time and maximum number of tetrahedrons are reduced, the mesh quality is worse than that obtained by the characteristic-range method. When the postprocessing is operated with the hybrid method, the final and maximum numbers of tetrahedrons are 4718 and 14,859, respectively, and the required CPU time is 53 s. The mesh quality is better than that obtained by the characteristic-range method (Fig. 3b). Based on the preceding discussion, the conclusions about these three methods in the study of concentric cubes are drawn again in this case.

Concentric Spheres

For the concentric spheres shown in Fig. 4a, the diameters of the spheres are equal to 1 and 0.5, individually. After the boundary triangles are created (Fig. 4a), the region between these two spheres is decomposed into tetrahedrons. The histograms of the mesh qualities are shown in Fig. 4b. When the characteristic-range method is utilized, the final and maximum numbers of tetrahedrons are 3467 and 141,041, respectively, and the CPU time is 6 s. It is the same as the preceding two cases in that the ratio of the maximum number of tetrahedrons to the final number of tetrahedrons is large. From the results shown in Fig. 4b, about 80% of the tetrahedrons have the values of radii ratio equal to or greater than 0.7. Also, there are more than 70% of the elements with their values of skewness equal to or less than 0.6. When the hybrid method is utilized, the final and maximum numbers of tetrahedrons are 3656 and 13,242, respectively, and the CPU time is 5 s. When a comparison with the results obtained by the characteristic-range method is made, it is found that the number of extra tetrahedrons is significantly reduced, but the mesh quality deteriorates. It is the same as the preceding two cases in that the hybrid method with postprocessing consumes longer computer time, but improves the mesh quality (Table 1 and Fig. 4b). Based on the preceding discussion, the following two cases are investigated by using the hybrid method with postprocessing.

Generation of Meshes for the Flow Regions Around ONERA M6 Wing and Airplane

ONERA M6 Wing

The aspect and taper ratios of the ONERA M6 wing[19] (Fig. 5a) are 3.8 and 0.56, respectively. For the computational domain, the ONERA M6 wing with a semispan of 1.0, a root chord of 0.6775, and a leading-edge sweep of 30 deg is enclosed by a symmetrical plane and several far-field planes that are at least 10 times the semispan away from the wing (Fig. 5a). For this case, the maximum and final numbers of tetrahedrons are 808,766 and 283,670, respectively, and the CPU time is 1412 s. Therefore, about 12,054 tetrahedrons are generated in 1 min. When the data given in Refs. 14 and 15 are considered as the reference values, the results shown in Figs. 5b and 5c indicate that the hybrid method with postprocessing provides a mesh with satisfactory quality, even though the distribution of boundary mesh and the number of tetrahedrons are different from those in these two references.

Airplane

As shown in Fig. 6a, the length of fuselage and wing span for an airplane are 10 and 15 ft, respectively. The cross section of the wing corresponds to the NACA 23012 airfoil. The aspect and taper ratios of the wing are 8.0 and 0.5, respectively. This wing has an incidence angle of 3 deg, a dihedral angle from 0 to 5 deg, and a twist angle from 3 to 1 deg. The cross sections of vertical and horizontal wings are NACA 0012 airfoil types. The incidence, dihedral, and twist angles of the horizontal wing are similar to those of the main wing. For the computational domain, the far-field boundaries are about 10 semispans away from the airplane (Fig. 6a). The final and maximum numbers of tetrahedrons are 913,447 and 1,931,745, respectively, and the CPU time is about 69 min. Therefore, more than 13,000 tetrahedrons are generated in 1 min. In this case, about 400-MB RAM is used. It is the same as the earlier cases in that the hybrid method with postprocessing produces a mesh with satisfactory quality (Fig. 6b). From the preceding discussion, the present

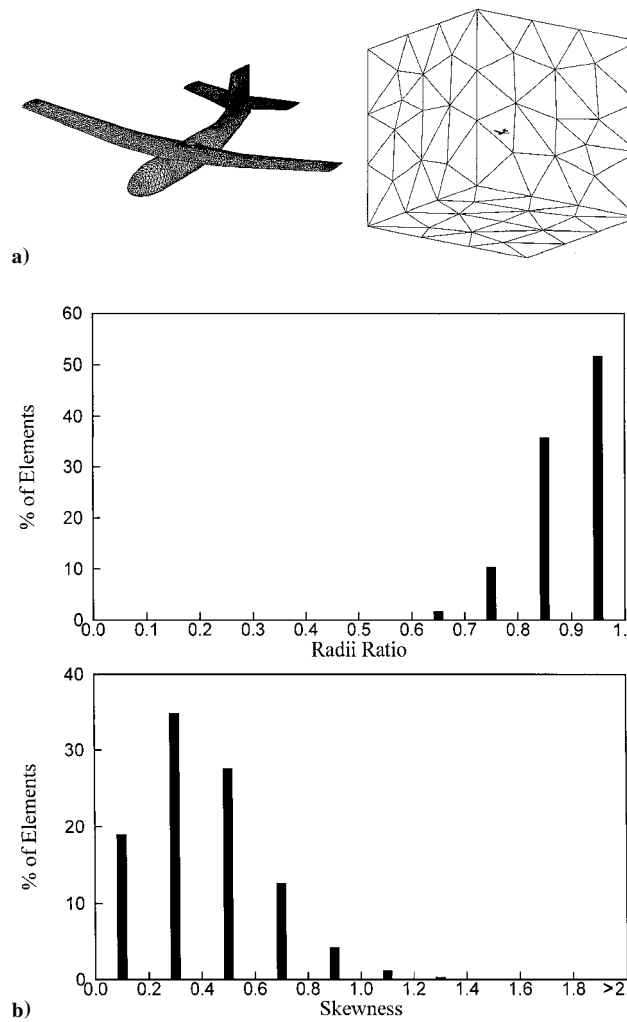


Fig. 6 Airplane: a) partial view of boundary mesh and b) histograms of mesh qualities.

strategy is suitable for generating meshes around bodies with complex configurations.

Conclusions

A new strategy, which includes the statements of the no-cavity and nonoverlapping conditions and a mesh-generation approach, is presented to create unstructured triangular and tetrahedral meshes. This mesh-generation approach is composed of seven steps: construction of a background grid, creation of the boundary mesh, generation of nodes, connection of the nodes into a mesh, achievement of the no-cavity condition, accomplishment of the nonoverlapping condition, and utilization of a postprocessing (only for three-dimensional mesh generation). For a two-dimensional domain with four-element airfoils, the preceding steps are illustrated. Unlike the advancing-front method, nodes generated by the present approach can be placed inside or outside the bodies, and the created elements are permitted to overlap each other or be located inside/across the bodies. Therefore, the intersection checking process is not needed, and this reduces the numerical effort, especially for cases with complex configurations. It is different from the Delaunay triangulation method in that the present approach is capable of guaranteeing boundary integrity because all of the boundary segments/triangles are retained during the generation or cancellation of the triangles/tetrahedrons. Besides the preceding features, the implementation for the no-cavity and nonoverlapping conditions is based on integer operations, which make the checking processes less computational intensive and insensitive to the machine accuracy. For the concentric cubes, coaxial annual pipe, and concentric spheres that are studied, the final and maximum numbers of tetrahedrons, CPU time, and histograms of mesh qualities are obtained. These results indicate that the hybrid method requires less computer memory and that the postprocessing significantly improves the mesh quality. When the results in Refs. 14 and 15 are considered as the reference values, the histograms of mesh qualities indicate that the hybrid method with postprocessing produces a mesh with satisfactory quality for the ONERA M6 wing. The results for the study of the flow region around an airplane show that the present strategy is suitable for generating meshes around bodies with complex configurations.

Acknowledgments

This research was supported by the National Science Council of the Republic of China under Contract NSC89-2612-E-006-010. The authors would like to thank F. M. Yu and D. L. Sheu for providing the geometrical configuration of the airplane.

References

[1]Lo, S. H., "A New Mesh Generation Scheme for Arbitrary Planar Domains," *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 8, 1985, pp. 1403–1426.

[2]Löhner, R., and Parikh, P., "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method," *International Journal for Numerical Methods in Fluids*, Vol. 8, No. 10, 1988, pp. 1135–1149.

[3]Hwang, C. J., and Wu, S. J., "Global and Local Remeshing Algorithms for Compressible Flows," *Journal of Computational Physics*, Vol. 102, No. 1, 1992, pp. 98–113.

[4]Bowyer, A., "Computing Dirichlet Tessellations," *Computer Journal*, Vol. 24, No. 2, 1981, pp. 162–166.

[5]Watson, D. F., "Computing the *n*-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes," *Computer Journal*, Vol. 24, No. 2, 1981, pp. 167–172.

[6]Baker, T. J., "Three-Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets," AIAA Paper 87-1124, June 1987.

[7]Mavriplis, D. J., "Unstructured Mesh Generation and Adaptivity," NASA CR 195069, Inst. for Computer Applications in Science and Engineering, ICASE Rept. 95-26, April 1995.

[8]Mavriplis, D. J., "Unstructured Grid Techniques," *Annual Review of Fluid Mechanics*, Vol. 29, 1997, pp. 473–514.

[9]Löhner, R., "Some Useful Data Structures for the Generation of Unstructured Grids," *Communications in Applied Numerical Methods*, Vol. 4, No. 1, 1988, pp. 123–135.

[10]Merriam, M. L., "An Efficient Advancing Front Algorithm for Delaunay Triangulation," AIAA Paper 91-0792, Jan. 1991.

[11]Rebay, S., "Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer–Watson Algorithm," *Journal of Computational Physics*, Vol. 106, No. 1, 1993, pp. 125–138.

[12]George, P. L., "Improvements on Delaunay-Based Three-Dimensional Automatic Mesh Generator," *Finite Elements in Analysis and Design*, Vol. 25, No. 3–4, 1997, pp. 297–317.

[13]Yerry, M. A., and Shephard, M. S., "Automatic Three-Dimensional Mesh Generation by the Modified-Octree Technique," *International Journal for Numerical Methods in Engineering*, Vol. 20, No. 11, 1984, pp. 1965–1990.

[14]Woodard, P. R., Batina, J. T., and Yang, H. T. Y., "Unstructured Mesh Quality Assessment and Upwind Euler Solution Algorithm Validation," *Journal of Aircraft*, Vol. 31, No. 3, 1994, pp. 644–650.

[15]Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1619–1625.

[16]Zhu, J. Z., and Zienkiewicz, O. C., "A Posteriori Error Estimation and Three-Dimensional Automatic Mesh Generation," *Finite Elements in Analysis and Design*, Vol. 25, No. 1–2, 1997, pp. 167–184.

[17]Suddhoo, A., and Hall, I. M., "Test Cases for the Plane Potential Flow Past Multi-Element Aerofoils," *Aeronautical Journal*, Vol. 89, No. 890, 1985, pp. 403–414.

[18]Pirzadeh, S., "Structured Background Grids for Generation of Unstructured Grids by Advancing-Front Method," *AIAA Journal*, Vol. 31, No. 2, 1993, pp. 257–265.

[19]Schmidt, V., and Charpin, F., "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers," *Experimental Data Base for Computer Program Assessment*, AR-138, AGARD, May 1979, Appendix B1.

P. Givi
Associate Editor