AIAA JOURNAL
Vol. 32, No. 12, December 1994

Quality Estimates and Stretched Meshes Based on Delaunay Triangulations

Jens-Dominik Müller*
von Kármán Institute for Fluid Dynamics, 1640 Rhode-St.-Genèse, Belgium

An unstructured grid generation method based on the Delaunay triangulation is presented that is capable of generating stretched layers along surface lines while retaining control over maximum angles. The method requires only minimum user input in the form of a set of boundary vertices, the character of each boundary, the thickness of the stretched layer, and the maximum aspect ratio desired. Upper and lower angular bounds are estimated and the grid quality is analyzed statistically.

I. Introduction

UNSTRUCTURED mesh methods are about to become the workhorse for Euler calculations in the aeronautical industry. Their flexibility dramatically reduces the time to generate a computational mesh around a complex geometry from man-months for a structured multiblocked grid to CPU minutes for a tetrahedral unstructured grid. Besides the gains during the mesh generation, even more significant gains during the calculation of the solution are offered by solution-adaptive grid refinement. Although adaptivity is intrinsic to the concept of an unstructured mesh, it is a rather cumbersome procedure on structured meshes.

However, the solution of the Navier-Stokes equations so far has remained a prime application for structured meshes as grid stretching along shock or shear layers is a trivial task on structured meshes but is relatively hard to achieve on unstructured triangular or tetrahedral grids. The first two unstructured methods that could generate stretched meshes did so either with the additional input of a user-specified structured point cloud[1] or at the expense of losing control of the maximum angles occurring in the mesh.[2] This loss of angular control leads to increased interpolation errors.[3] The latter method also requires additional user input in the form of a background grid for the interpolation of scales. Independently of the present work, methods have been developed recently in two and three dimensions that use some form of hyperbolic structured griding in a viscous layer and fill the remaining domain with an unstructured grid generation technique.[4–6]

The Frontal Delaunay method in two dimensions[7,8] (FroD) presented here is based on a philosophy of truly minimal user input. In the case of isotropic meshes for the Euler equations, minimal user input consists of the set of boundary vertices only. In the case of stretched meshes for Navier-Stokes calculations, the thickness of the stretched layer and the desired maximum aspect ratio are also needed as input. However, simple qualitative information about the boundary surfaces will be used to strongly improve grid quality. Note that FroD is designed as an initial grid generation step that feeds into a solution-adaptive procedure. The goal is to incorporate as much a priori knowledge of the user into the initial mesh. Eventual local faults then will be picked up and fixed by solution adaptation.

FroD is a Steiner triangulation; i.e., starting from an initial triangulation, vertices are inserted incrementally in a frontal manner until a final grid with an appropriate vertex distribution is produced. The concept is based on the Delaunay triangulation[9] whose construction principle that the circumcircle around each triangle does not contain any other vertex is exploited for the automation of the grid generation process. Moreover, Delaunay grids have many interesting properties, an extensive documentation of which can be found in Ref. 10.

In the following, we will first describe the different elements of FroD: the derivation of the background mesh as well as the vertex generation for the viscous and inviscid regions. The latter half of this paper presents the quality measures that can be derived for the isotropic vertex generation part.

II. Frontal Delaunay Method

In FroD, the angular control of ordered rectangular meshes in a stretched layer is combined with the simplicity of the unstructured mesh generation in an isotropic region. The two different ways of generating vertices are embedded in a frontal process that introduces the vertices into an existing Delaunay triangulation. Local isotropic and stretched length scales are interpolated on a background mesh that is derived from an automatically modified triangulation of the boundary vertices.

A frontal technique is employed to construct vertices that refine the existing triangulation. The process of generating and connecting interior vertices for the initial triangulation of the boundary vertices can be distinguished into three different parts: 1) building stretched wedges with as altitude the viscous scale ν in the stretched regions along solid boundaries or wakes, by constructing a new vertex for each existing vertex in the front; 2) building isotropic triangles with the viscous scale ν in the remaining stretched regions by constructing a new vertex for each frontal edge; and 3) building triangles with the isotropic scale h in the rest of the domain.

Although building triangles in the stretched and isotropic regions is essentially the same process with different scales, the building of wedges is a fundamentally different procedure.

A. Generation of the Background Mesh

A constrained Delaunay triangulation of all boundary vertices is computed as an initial triangulation to begin the vertex generation process (Fig. 1). Constrained refers to the fact that possibly edges have to be swapped to insure the boundary conformality of the triangulation. This triangulation provides at no extra cost a background mesh to interpolate a local value of desired distance between vertices at any point. The "spacing" h is interpolated in a possibly nonlinear fashion between the three nodal values of the background triangle that contains the interpolation point. The spacing h at a vertex of the background grid is computed as the average distance to its two neighboring vertices on the boundary.

Since this initial mesh is a constrained Delaunay triangulation, the circumcircle of each triangle does not contain any other vertex of the grid, except for "invisible" vertices that are obscured from view to the algorithm by a required boundary edge. This property is exploited to find a second set of vertices at a user-specified distance, say δ, to define a stretched region around each segment. In this region, the interpolated stretching σ decays from the user-defined

Received April 17, 1993; presented as Paper 93-3347 at the AIAA 11th Computational Fluid Dynamics Conference, Orlando, FL, July 6–9, 1993; revision received March 22, 1994; accepted for publication March 27, 1994.

*Currently Professor, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109.

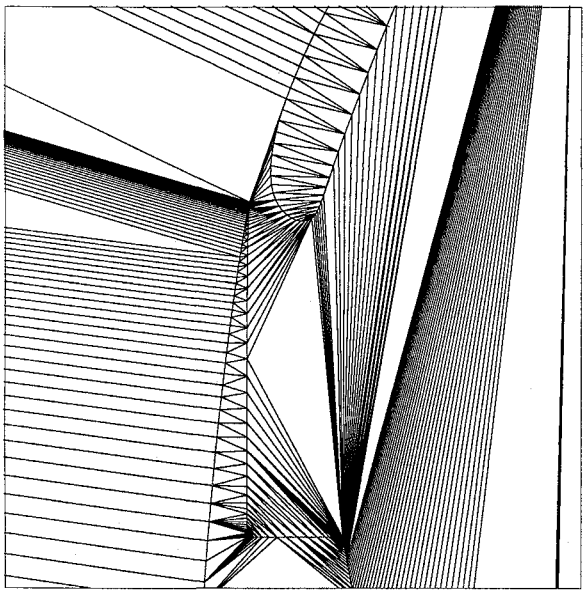

Fig. 1 Initial triangulation of the boundary vertices used to derive a background grid and the interior triangulation. The final triangulation of the detail shown can be seen in Fig. 4, the full configuration in Fig. 3.

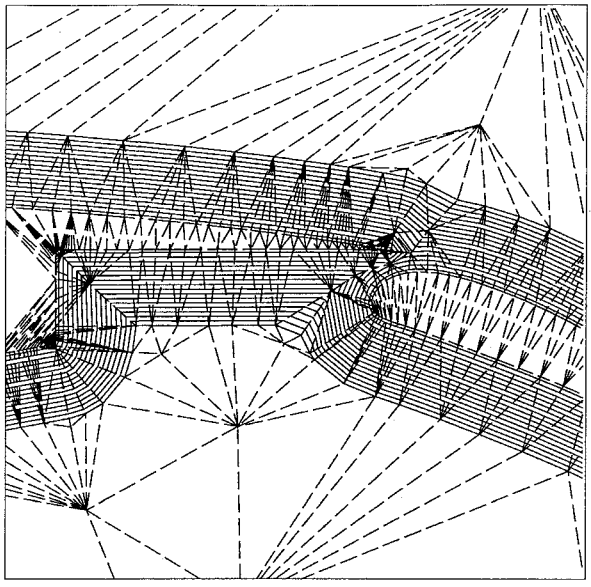

Fig. 2 Background grid with a thickness of the stretched layer $\delta = 0.04$. The isolines of stretching magnitude are shown as solid, the triangulation of the grid as dashed lines.

maximum aspect ratio Σ at the solid boundary to an isotropic value used in the isotropic domain. If the radius of the circumcircle of a triangle with a solid boundary edge is larger than δ, a vertex normal to the edge at distance δ can be introduced without any interference with other regions. The circumcircle criterion guarantees that for any point along the median between the midpoint of the edge and the circumcenter there is no closer vertex to that point than the two vertices that form the edge. The stretching at this vertex will be the isotropic value. Two layers interfere if the radius is less than δ and the triangle in question connects different solid boundaries. In that case the vertex that delimits the stretching layer will be placed at the circumcenter, and the stretching magnitude at that vertex will decay only to an appropriate fraction of the stretching at the solid boundaries. Interfering layers can be observed in Fig. 2 between the flap and the main airfoil. Besides defining the local stretching σ, these extra vertices also serve to define a rim around each segment of the geometry. No check for intersection of different fronts emanating from different segments is necessary if the stretched hyperbolic grid around each segment is constructed only within the perimeter of the rim that is assigned to the segment.

To have a decay of the aspect ratio with increasing distance from the wall in the form of a geometric series, the linearly interpolated stretching value is mapped with a nonlinear function. Given the thickness of the stretched layer δ, the maximum edge length at the wall H, and the maximum aspect ratio Σ, we can calculate a constant ratio of the two aspect ratios of successive cells and the number of cells needed to fill the layer. The equation of a supercircle, $x^q + y^q = r^q$, has been selected to approximate this decay, and the exponent q can be calculated by the computer from the given values. The spacing values assigned to these vertices can be interpolated on the background grid before the set of vertices that defines the stretched layer is introduced.

On grids with a strong spacing gradation, such as, for example, grids used for airfoil calculations, we find a monotonic variation between the fine spacing on interior boundaries and the coarse spacing on a far-field boundary if the background triangle connects directly from the interior to the far-field boundary. But along concave contours it may happen that the Delaunay criterion connects between finely spaced interior boundaries and the background grid will specify an area with a very low spacing gradient that would produce a too-fine grid. Consistent with the philosophy of minimal user input, the program introduces the necessary vertices to break the unwanted connections. Unwanted connections can be, for example, connections between components that the user has specified not to be connected or connections between boundary vertices that are nonconsecutive.

The procedure will be to detect an illicit liaison and to place a vertex at the circumcenter of the badly connecting triangle. During a subsequent retriangulation, most if not all of the triangles exhibiting an unwanted connection will be broken, and thus very few extra vertices suffice. Because of its placement, the new vertex is equidistant from all ill-connected boundaries. The spacing is extrapolated from the most finely discretized boundary using some average spacing gradient of the initial triangulation. The vertices for disconnection are constructed before the vertices that define the stretching layer. At the end of the modifications of the background grid, all of those disconnection vertices that do not fall into any stretched region are reintroduced. Note that all added vertices are only present in the background grid.

B. Building Stretched Wedges

Shear layers expose high gradients normal to the layer, whereas the gradients tangential to the layer remain low. Refining isotropically in both directions is not affordable. In the framework of minimum user input, grid stretching can be aligned with attached boundary layers or user-specified wakes. Once the layer separates from the surface, solution-adaptive refinement procedures have to be used to modify the grid. Similarly, the grid ought to be less stretched in regions where these "boundary-layer assumptions" don't hold.

An important problem of stretching in triangular meshes is the control of the angles in the cells.[3] More or less the only reliable way to avoid large angles while retaining high-aspect ratios is to emulate structured quadrilateral grids in the region where the shear layer is to be expected and choose a dividing diagonal. Isotropic meshing should take place in the regions far from the surfaces or where the layers begin to develop around stagnation points.

It thus seems natural to define a stretched scale ν as the length of the longest solid boundary edge H divided by the local stretching value σ. Starting from the concatenated strings of vertices that define the closed solid boundaries, a string of vertices is created around each previous string with distance ν between the vertices until the stretched regions are filled. The strings serve to link the vertices such that an average surface normal can be calculated to displace the vertices by ν. This will create cells with large aspect ratios where mesh spacing h is coarse and more isotropic cells where the user decided anyhow to have finer spacing, such as, at a corner.

Isotropic meshing with nearly equilateral triangles is more appropriate wherever the user chose such a fine boundary discretization that ν is larger than the length of the frontal edge. In a similar way the strings of vertices are split around corners in the geometry, since the flow features around a corner cannot be assumed a priori and isotropic refinement has to be used. The effects of reduced stretch-

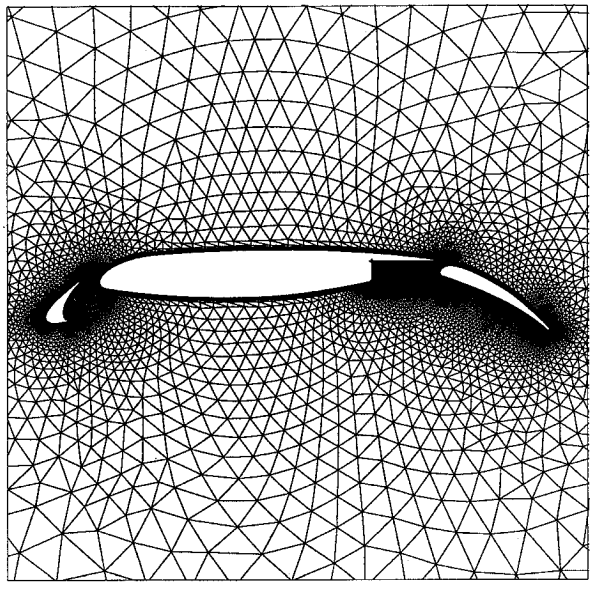

Fig. 3 Close-up of a three-element airfoil with a stretched layer around the components, maximum aspect ratio Σ = 10,000.

ing and reversion to isotropy can be seen in Fig. 6 around the trailing edge of the slat.

A Delaunay triangulation maximizes the minimum angles in the mesh.[11] On the other hand, we are interested in minimizing the maximum angle. Although the differences between MinMax and MaxMin triangulations are minor in isotropic grids, the differences are fundamental in stretched meshes. Consequently, one has to apply constraints to the triangulation while introducing vertices in the stretched regions. Since the desired connections that form the wedges between the different strings of vertices are known, the cells that properly connect between strings can be flagged and exempted from retriangulation. Hence, the algorithm to build layers of wedges can be cast in the following steps:

```
for each string of vertices:
 open a new string of vertices.
 for each vertex in the string:
  evaluate the local stretching value
   on the background grid,
  build a new vertex at distance ν to
   the vertex normal to the string,
  check whether that vertex lies within
   the proper stretched region,
  check whether that vertex is properly
   spaced with the vertices in the grid.
  if properly spaced and located:
   append the new vertex to the new string,
   introduce the vertex into the constrained
    triangulation,
   protect the newly formed wedges
    between strings.
  end if
 end for
 close the new string of vertices.
end for
```

The actual implementation allows for the proper evaluation of the normal at the ends of the strings and for an interruption of the string in case the aspect ratio check fails or a corner is detected.

C. Building Triangles in Stretched Regions

Once the stretched layers have been filled with wedges, the remaining areas in the stretched regions around corners and edges and the rest of the computational domain are filled with isotropic triangles.[7] The only difference between stretched and isotropic regions is the evaluation of the local length scale. To match the short sides of the wedges, the length scale for triangles in the stretched region has to be ν.

As a consequence of this reversion to isotropy, the discretization of the solid boundaries has to approach the stretched scale ν at corners or open ends. Then a smooth transition from isotropic quadrilaterals with a dividing diagonal to isotropic triangles is obtained (Fig. 6).

D. Building Triangles in the Isotropic Region

In contrast to the advancing front method (AFM)[2] and Rebay's frontal Delaunay method,[12] tracking of the front is not necessary in FroD for the generation of isotropic triangles as we provide over a closed Delaunay triangulation at any stage of the process. A frontal edge is a short edge in a cell with disparate edge lengths that is shared by a well-shaped or unrefinable cell. For each frontal edge a new vertex is constructed on the median into the badly shaped cell such that the distances between the new vertex and the two vertices forming the edge approximate h, the isotropic length scale.[7]

A check insures that the new vertex is sufficiently distant from any other vertex. The underlying Delaunay triangulation is used as an efficient search structure for the distance check, since the dual of the Delaunay triangulation, the Voronoi diagram, actually solves the closest point problem.[13] Once the closest vertex has been found by walking along the Voronoi diagram, the distance can be compared with the spacing required by the background grid at the midpoint between the two vertices in question. If the distance is too small, the new vertex is either rejected if it belongs to a previous row of vertices or is merged with the conflicting one if both are on the same row. Note that all other Delaunay refinement methods[12,14–16] are "vertex greedy" in the sense that once a vertex has been constructed, it is never removed from the triangulation. The vertex merging is key to the regularity of the obtained meshes. The isotropic vertex generation loop can be summarized as follows:

```
do while new vertices are found:
 for each triangle in the grid:
  for each edge of the triangle:
   if this edge is frontal:
    build a vertex on the median.
   end if
  end for
 end for
 for each new vertex:
  find the closest vertex in the mesh,
  if the new vertex is properly spaced:
   introduce the vertex into the structure
  else if the conflicting vertex is on a lower row:
   discard the new vertex,
  else
   extract the conflicting vertex from the grid,
   merge it with the new vertex,
   reintroduce the merged vertex.
  end if
 end for
end do
```

The number of operations necessary to introduce a vertex in the isotropic process is of $\mathcal{O}(1)$ for the construction and of $\mathcal{O}(\log N)$ for the distance check if a tree structure for the searches on the background mesh was used.[17] Hence, the cost of generating a mesh with N vertices, $\mathcal{O}(N \log N)$, is asymptotically optimal.[13]

The mesh in Figs. 3–8 is created with a thickness of the stretched layer $\delta = 0.04$ on a total chord from slat to flap of about 1.3; 14,681 vertices are created in the viscous region in 27.5 s and 16,247 vertices with the isotropic procedure in 119.4 s on a DEC-α 3000. Note that there is no accelerating tree structure in the current implementation and timings increase above the factor $N \log N$ with the mesh size.

III. Examples

The regularity of the grid is entirely due to the frontal insertion, since no smoothing filter was applied. The cell surface area varies very smoothly from the smallest cells at the trailing edges and corners to the largest cells at the outer boundary with a ratio of over 2×10^7 in the Euler grid for the shown configuration and of over 4×10^{11} in the Navier-Stokes grid shown. The robustness and the

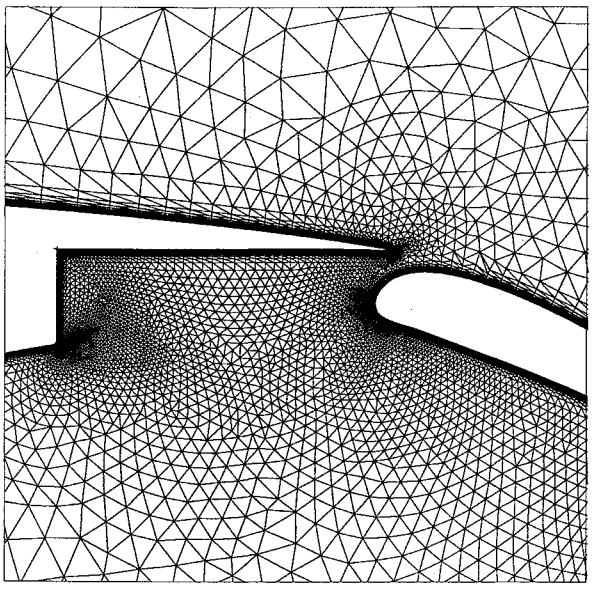

Fig. 4 Detail of the three-element airfoil between the main airfoil and the main flap.

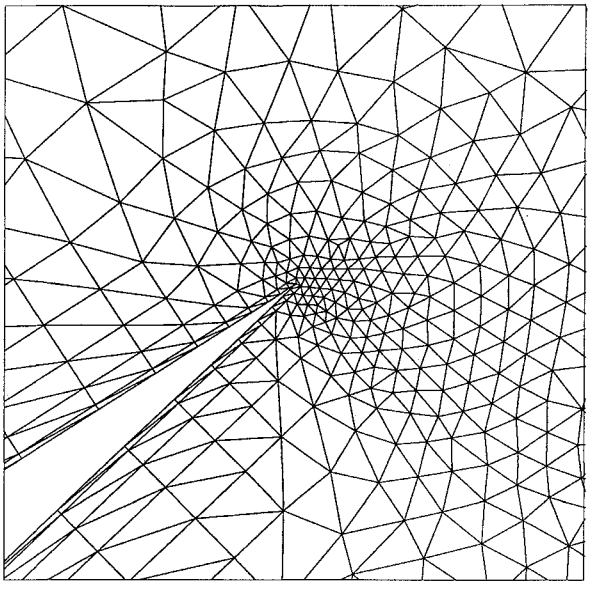

Fig. 6 Detail of a three-element airfoil, upper trailing edge of the slat.

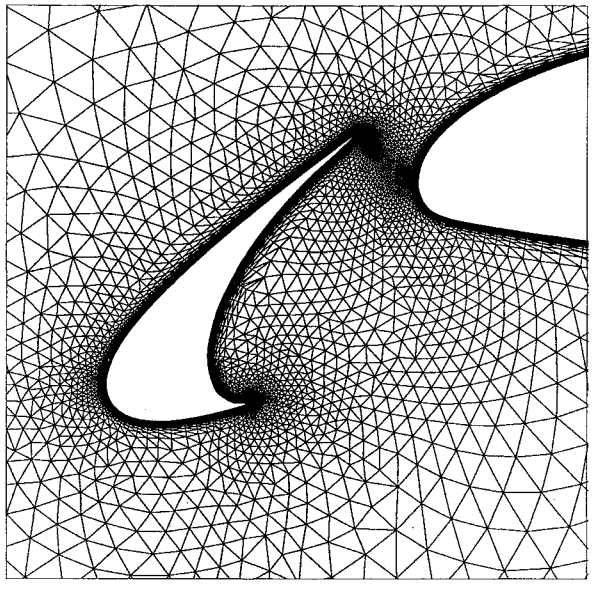

Fig. 5 Detail of a three-element airfoil around the slat.

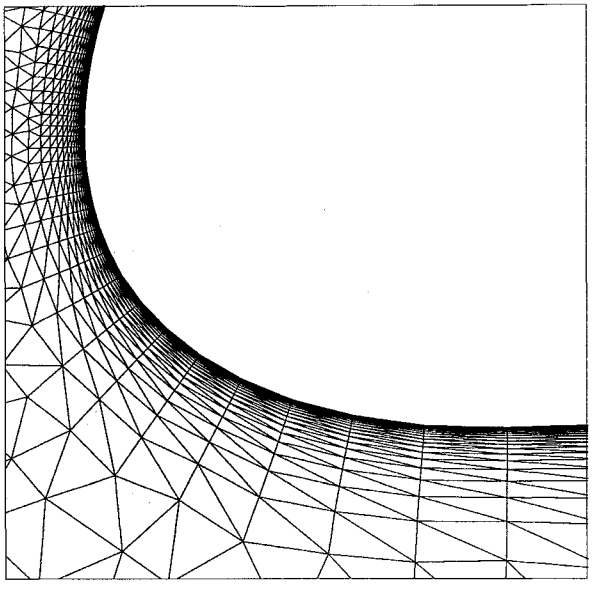

Fig. 7 Detail of the slat around the leading edge.

smoothness of the isotropic part of the algorithm can be seen in Figs. 4 and 5 where different fronts coalesce between components and the smoothness and regularity of the grid is barely affected.

The viscous grid generation part proves less robust: irregularities can be observed in Figs. 5 and 8. If the gradient of the stretching, $\nabla\sigma$, is different for two consecutive stacks of vertices in the viscous layer, the stack with the larger gradient will grow faster, leading to the tilt in the layer that can be observed. This phenomenon is amplified by the highly nonlinear way of interpolation of the stretching, although a postprocessing step that swaps edges for a smallest maximum angle (MinMax) usually removes the problem. More severe grid distortions occur when the stretching magnitude does not decay to a value close to isotropy, as would be the case if a wake line was traced out downstream from the trailing edge of the slat, very close to the upper surface of the main airfoil and grided. A frontal mechanism that does not depend on the locally interpolated stretching should be adopted here.

A turbulent high-Reynolds-number flow calculation ($Ma = 0.2$, $\alpha = 8.2$ deg, $Re = 9 \times 10^6$) on the presented grid (Fig. 9) has been performed by T. J. Barth using a Finite Volume scheme with the Baldwin-Barth one-equation turbulence model.[18] It compares well with Barth's calculation,[18] although a discretization of the wakes in a solution-adaptive way is necessary for a full resolution of the viscous features.

IV. Angular Bounds for the Isotropic Insertion

As the generation and introduction of vertices in FroD are embedded in the rather rigorous mathematical framework of the Delaunay triangulation, the algorithm can be analyzed mathematically. Ruppert[16] gave an algorithm that guarantees a minimum angle but produces grids with too few triangles in the field to be suitable for flow calculations. Chew[15] presented an algorithm that guarantees maximum angles of 120 deg but that does not allow specification of a spacing gradation. Both methods produce meshes that are much more irregular than the ones produced by Rebay's method,[12] the AFM,[2] or the present method. On the other hand, no derivation of angular bounds is known for the more heuristic AFM. The smoothness of the grids created with the AFM relies on a posteriori smoothing.

In the following, angular bounds for FroD are derived, assuming that the triangulation is Delaunay and that the gradient is constant. Thus, the analysis can fail in the vicinity of boundaries and for configurations that cross edges of the background grid where the spacing gradient exhibits a jump.

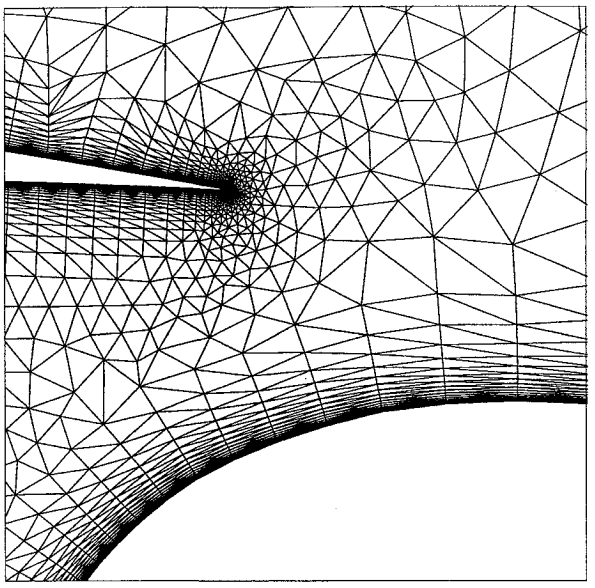

Fig. 8 Detail of the three-element airfoil at the gap between main airfoil and flap. An instability of the viscous procedure can be seen in the upper left corner.

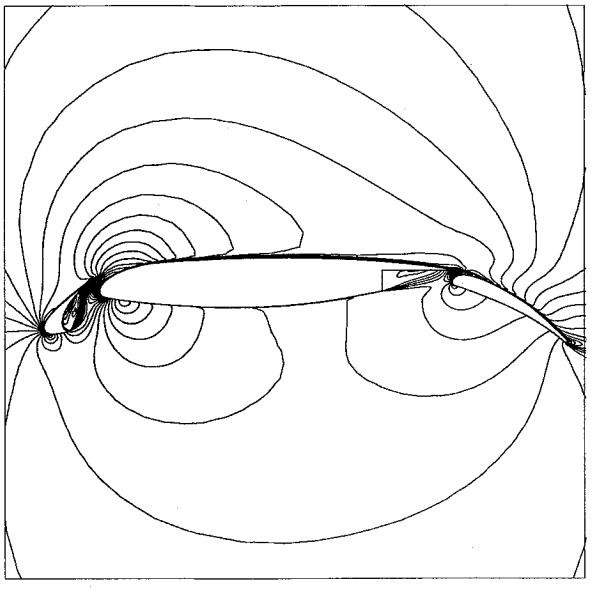

Fig. 9 Iso-Mach contours around the three-element airfoil, Ma = 0.2, α = 8.2 deg, $Re = 9 \times 10^6$. Calculation by Barth.

A. Upper Angular Bound

FroD will detect short edges in triangles that are considered too obtuse or too acute and will construct a vertex on the median of that frontal edge to refine the triangulation locally and to improve the grid. The new vertex is placed approximately at a distance h from the two vertices that form the frontal edge.[7] The introduction of the vertex is subject to a distance check: if the new vertex is located too close to any existing vertex, it cannot be introduced, and the refinement will not take place. The vertex is too close if the distance to the closest vertex is lower than a tolerance a times the local mesh spacing h that is interpolated at the midpoint of the two vertices in question.

In this way, we have to consider as a worst case a triangle with nondesirable properties that cannot be refined because the local vertex density is already too high. Figure 10 shows the limit case: the obtuse triangle ABC with a circumcircle O of radius r. As this triangle is Delaunay, there is no other vertex of the triangulation in O, and we can always introduce the new vertex D if its spacing disk

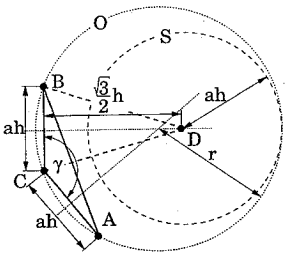

Fig. 10 Obtuse triangle ABC with maximum angle γ. The circumcircle O contains the spacing disk S around D.

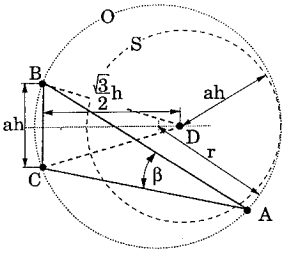

Fig. 11 Acute triangle ABC with minimum angle β. The circumcircle O contains the spacing disk S around D.

S with a radius ah around D is contained in O. Consequently, the worst case occurs when the two short edges of the triangle ABC have the minimum permissible length of ah. If we were to increase this length, r has to increase as well, and the disk around D is included in O already at a smaller angle γ. Hence, given the tolerance a, we can calculate the maximum angle γ for $\nabla h = 0$:

$$\gamma = 2 \arctan\left(s - \frac{1}{4s}\right), \qquad s = \frac{\sqrt{3}}{2a} + 1$$

The narrower limit is obtained for $a = 1$ with $\gamma = 120$ deg. However, this restrictive tolerance will inhibit the insertion of D as D then is too close to C and B. The maximum angle increases monotonically to 139 deg for $a = 0.5$ (Fig. 12).

In the case of $\nabla h \neq 0$, the construction is executed numerically. All orientations of ∇h are tested since the length of the edge CA, the placement of D, and the shape of S depend on the gradient. The deformed spacing disk S is traced out and tested for inclusion in O. The curves for $\nabla h = 0.0, 0.31$, and 0.5 are presented in Fig. 12. The curves are drawn for all values a for which S does not contain B or C for any gradient.

B. Lower Angular Bound

Similarly, we can derive a lower angular bound when refining triangles with one short edge BC opposite the acute angle β by inserting D, as depicted in Fig. 11. As long as the circumcircle O is large enough to contain the spacing disk S around D, refinement of the triangle ABC will always be possible, and β will increase after the insertion of D. Again, as β is further decreased, O will shrink until the limiting case is reached where O and S become tangent. In this case we find for $\nabla h = 0$

$$\beta = 2 \arctan \frac{1}{2s}$$

The smallest minimum angle is found for $a = 1$ as $\beta = 30$ deg and decreases monotonically to $\beta = 21$ deg for $a = 0.5$. The construction for $\nabla h \neq 0$ is executed numerically as in the obtuse case. The curves are given in Fig. 12.

C. Measured Angular Bounds

Maximum and minimum angles for two configurations have been measured using different tolerances a for the generation of Euler meshes. The finer configuration is the three-element configuration used for the viscous grids in Figs. 1–8. The gradient of 90% of the

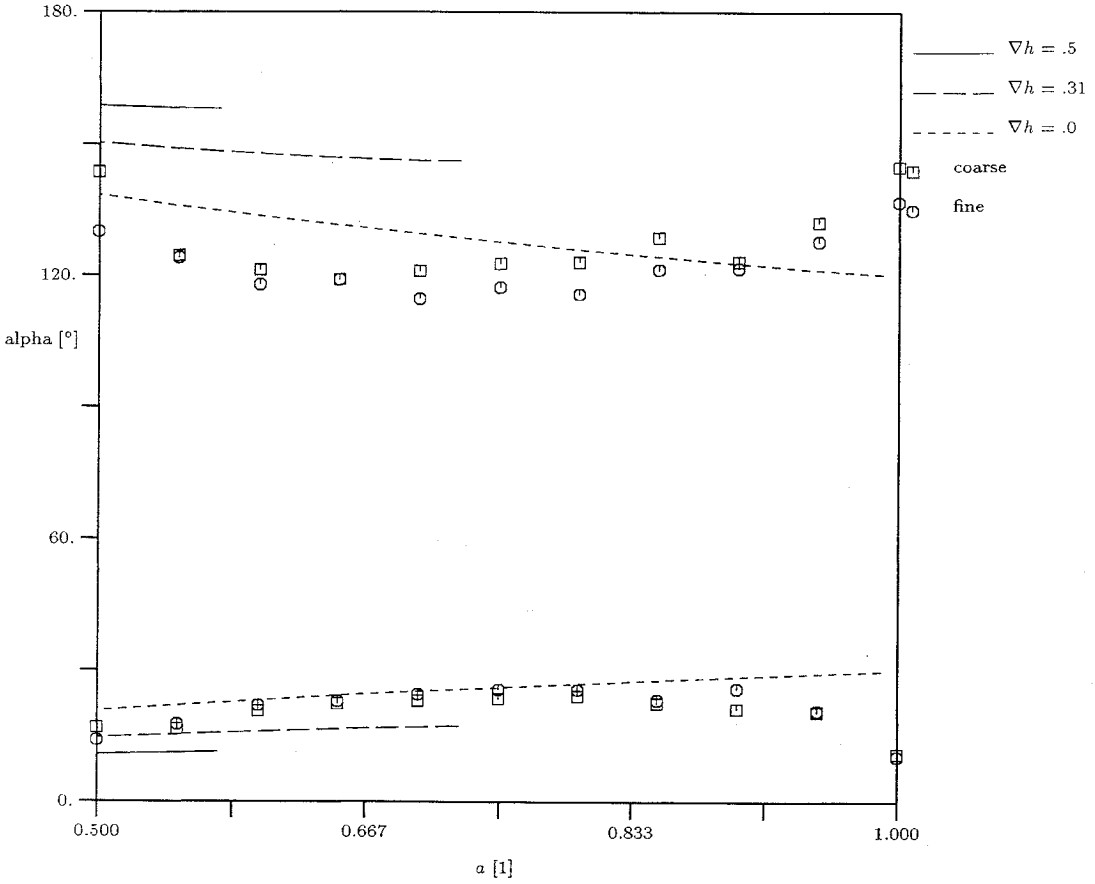

Fig. 12 Upper and lower angular bounds for different ∇h and measured maximum and minimum angles of two configurations in function of the tolerance a.

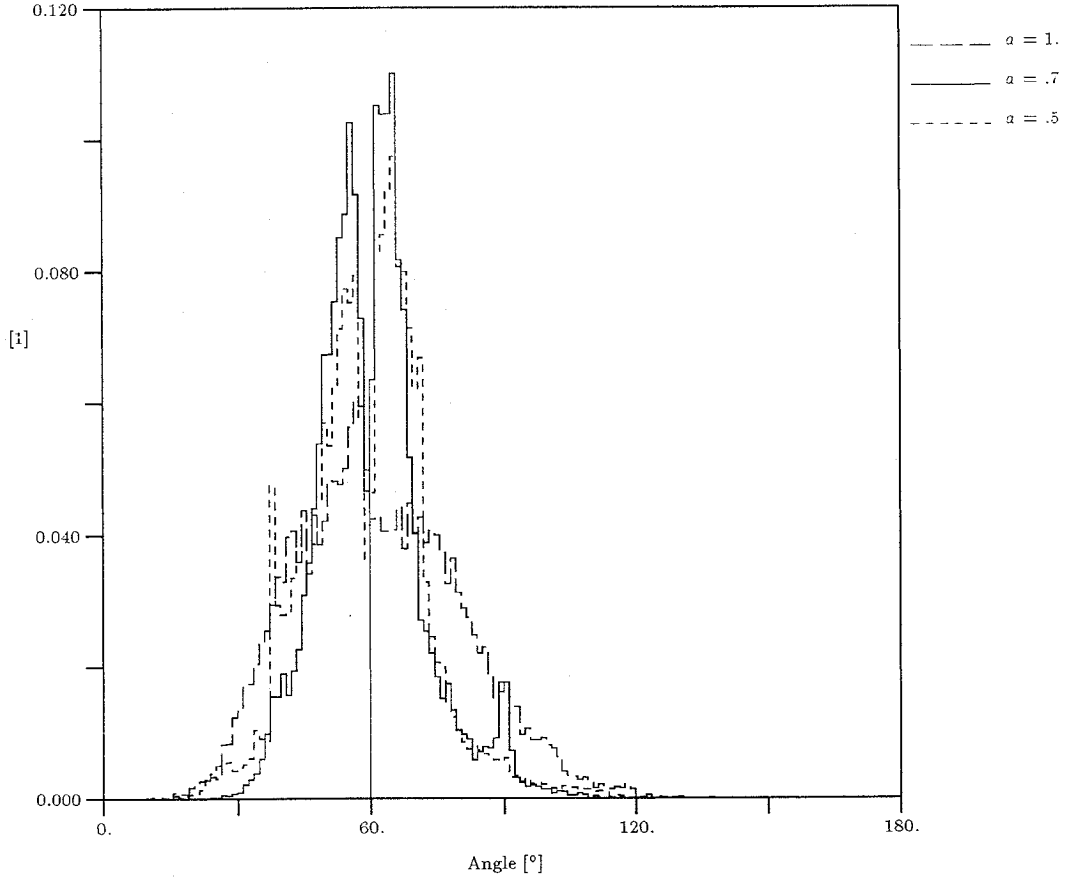

Fig. 13 Distribution of maximum and minimum angles for various a of the three-element configuration of Figs. 1–8.

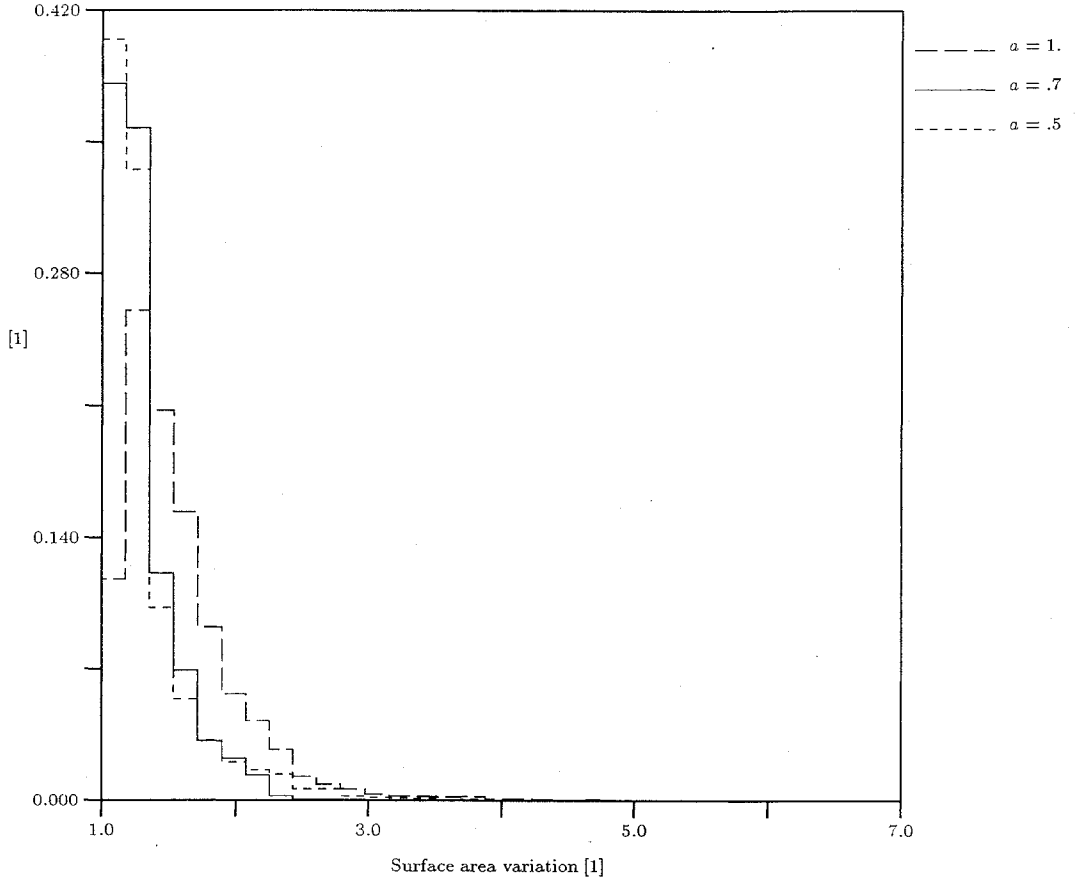

Fig. 14 Distribution of surface area ratios.

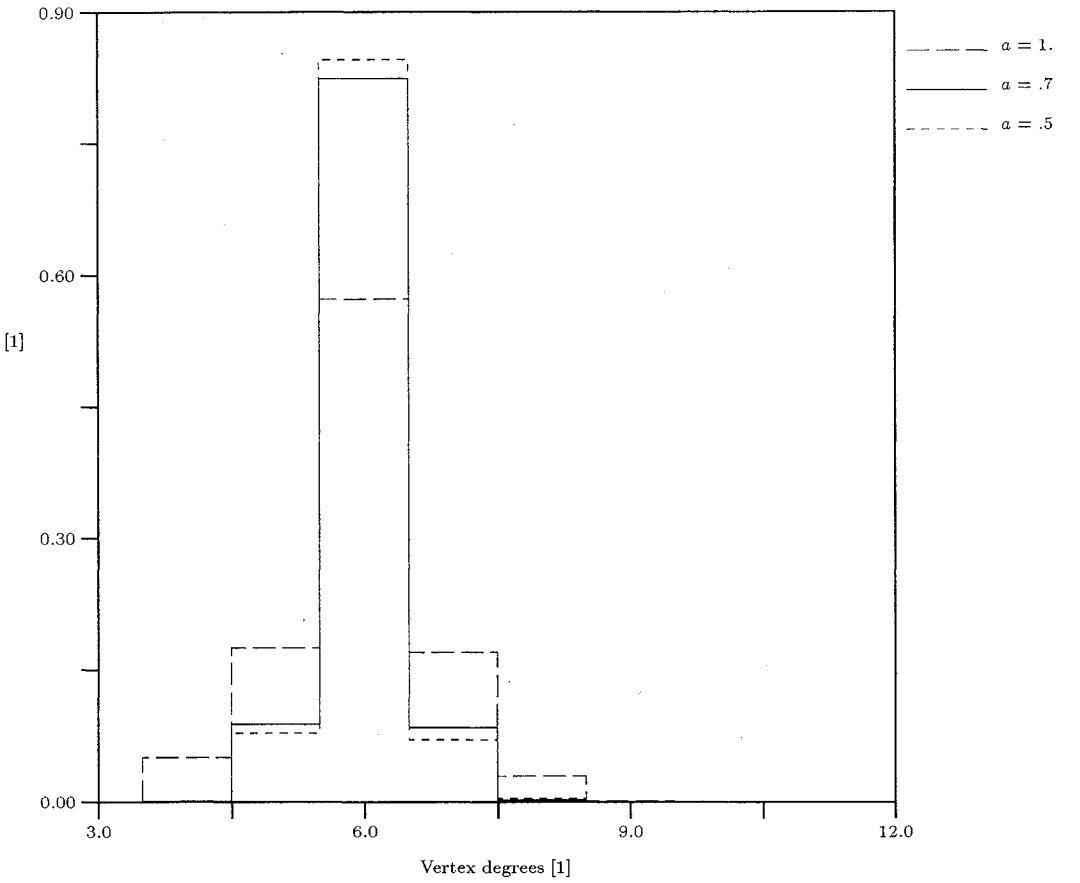

Fig. 15 Distribution of vertex degrees.

cells is inferior to $\nabla h \leq 0.31$. The Euler meshes for this configuration contain between 7581 and 12,739 cells for a between 1 and 0.5. The coarser configuration is the three-element airfoil used in Ref. 7 and has a gradient of less than $\nabla h \leq 0.5$ in 90% of the cells. The Euler meshes for this configuration contain between 2626 and 4708 cells. The upper and lower bounds predicted by the analysis are observed.

D. Distribution of Angles, Surface Area Ratios, and Vertex Degrees

The results in Fig. 12 show the best values for a tolerance around $a = 0.65$. However, distributions of the values for each cell over the entire grid can show the grid quality better than the maximum or minimum values. As Figs. 13–15 show, a narrow peak close to the optimum values is found for a rather large tolerance $a = 0.7$ with a very small number of cells with excessive angles $\gamma \geq 90$ deg or a surface area ratio larger than 2. The surface area ratio plotted for each cell in Fig. 14 is defined as the larger one of the ratio or its inverse of the areas of each triangle compared with its neighbors. On the other hand a stringent tolerance of $a = 1$ will produce a broader variation with a larger number of undesirable cells. Any choice of $0.5 \leq a \leq 0.75$ will produce a satisfactory grid. The distribution of the vertex degrees shows an extremely narrow peak at the optimum value of six edges formed with each vertex. Very few vertices exhibit a lower or higher vertex degree.

V. Conclusions

The frontal Delaunay method has been demonstrated for practical high-Reynolds-number configurations. With a boundary discretization that reverts to isotropy at corners, a smooth transition to the triangular frontal process is found, and the high mesh quality of the isotropic part of the algorithm is retained. The algorithm proves sufficiently robust for viscous layers around solid surfaces that interact at their outer edges where stretching values have decayed close to isotropy.

For the isotropic vertex generation process, upper and lower angular bounds were derived for constant gradients of vertex spacing and confirmed with sample calculations. Furthermore, statistical measures show that only very few cells approach these bounds, whereas most of the cells exhibit values close to the smallest possible maximum angle and the largest possible minimum angle, and most vertices have the optimum vertex degree.

As in the purely isotropic case, only minimum user input is required for the stretched grids. Besides the set of boundary vertices, information about the boundary character as well as the thickness of the stretched layer and the maximum aspect ratio is needed. All other information is generated by the algorithm.

References

[1]Mavriplis, D., "Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation," *Institution for Computer Applications in Science and Engineering*, TR 88-47, Hampton, VA, 1988.

[2]Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, 1987, pp. 449–466.

[3]Babuška, I., and Aziz, A. K., "On the Angle Condition in the Finite Element Method," *SIAM Journal of Numerical Analysis*, Vol. 13, No. 2, 1976, pp. 214–226.

[4]Löhner, R., "Matching Semi-Structured and Unstructured Grids for Navier-Stokes Calculations," AIAA Paper, 93-3348, 1993.

[5]Pirzadeh, S., "'Unstructured Viscous Grid Generation by Advancing-Layers Method," AIAA Paper, 93-3453, 1993.

[6]Weatherill, N., Grid Generation by the Delaunay Triangulation, 3. Unstructured Grids for Viscous Flow, Lecture Series 1994-02, von Kármán Inst., Rhode-St. Genèse, Belgium, 1994.

[7]Müller, J.-D., Roe, P. L., and Deconinck, H., "A Frontal Approach for Internal Node Generation in Delaunay Triangulations," *International Journal of Numerical Methods in Fluids*, Vol. 17, No. 3, 1993, pp. 241–256.

[8]Müller, J.-D, Roe, P. L., and Deconinck, H., "Delaunay-Based Triangulations for the Navier-Stokes Equations with Minimum User Input," 13th ICNMFD Rome, June 1992.

[9]Delaunay, B., "Sur la sphère vide," *Bulletin of the Academy of Sciences of the USSR VII: Class. Sci. Mat. Nat. 793–800*, 1934.

[10]Barth, T. J., "Aspects of Unstructured Grids and Finite Volume Solvers for the Euler and Navier-Stokes Equations," AGARD, TR R-787, 1992.

[11]Lee, D. T., and Schachter, B. J., "Two Algorithms for Constructing a Delaunay Triangulation," *International Journal of Computation and Information Sciences*, Vol. 9, No. 3, 1980, pp. 219–241.

[12]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.

[13]Preparata, F. P., and Shamos, M. I., *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.

[14]Holmes, D. G., and Snyder, D. D., "The Generation of Unstructured Triangular Meshes Using Delaunay Triangulation," *Proceedings of the Second Conference on Grid Generation in Computational Fluid Dynamics*, Pineridge Press, Swansea, Wales, UK, 1988.

[15]Chew, P., "Guaranteed-Quality Triangular Meshes," Cornell Univ., TR 89-98993, Ithaca, NY, 1989.

[16]Ruppert, J., "A New and Simple Algorithm for Quality 2-Dimensional Mesh Generation," UCB, TR UCB/CSD 92/694, June 1992.

[17]Knuth, D. E., *The Art of Computer Programming—Sorting and Searching*, Vol. 3, Addison-Wesley, Reading, MA, 1973.

[18]Barth, T. J., "Recent Developments in High Order k-Exact Reconstruction on Unstructured Meshes," AIAA Paper, 93-0668, 1993.