A93-45041

AIAA-93-3347-CP

Proven Angular Bounds and Stretched Triangulations With the Frontal Delaunay Method

Jens-Dominik Müller

Von Karman Institute for Fluid Dynamics, 72, Chaussée de Waterloo, 1640 Rhode-St.-Genèse, Belgium,
Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI 48109, U.S.A.

April 1, 1993

Abstract

An unstructured grid generation method based on the Delaunay triangulation is presented that is capable of generating stretched layers along surface lines while retaining strong control over maximum angles. The method requires only minimum user input in form of a set of boundary nodes, the character of each boundary, the thickness of the stretched layer and the maximum aspect ratio desired. Upper and lower angular bounds and an upper bound in surface variation between neighboring cells are proven.

1 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 triangular 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. While 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 currently available unstructured methods that can generate stretched meshes do 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 truncation errors if gradients in the solution exist along the face opposite to a large angle. The latter method also requires additional user input in form of a background grid for the interpolation of scales.

The Frontal Delaunay Method, 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 nodes 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 to make the problem well-posed. However, simple qualitative information about the boundary surfaces will be used to strongly improve grid quality.

The Frontal Delaunay Method [3, 4, 5] is a Steiner triangulation, i.e. starting from an initial triangulation, nodes are inserted incrementally in a frontal manner until a final grid with an appropriate node distribution is produced. The concept is based on the Delaunay triangulation [6] whose construction principle to connect closest nodes is exploited for the automation of the grid generation process. Moreover, Delaunay grids have many interesting properties, including the fact that a Delaunay triangulation maximizes the minimum angle of all possible triangulations with a given set of vertices and is in this sense the smoothest triangulation. An extensive documentation on Delaunay methods can be found in [7].

In FroD, the strong 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 nodes are embedded in a frontal process that introduces the nodes 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 nodes.

2 Generation of the background mesh

The Delaunay triangulation of all boundary nodes is computed as an initial triangulation to begin the node generation process. This triangulation provides at no extra cost a suitable background mesh to interpolate a local value of desired distance between nodes at any point. This "spacing" 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 the nodes of the background triangles is computed as the average distance to its two neighboring nodes on the boundary.

As this initial mesh is a Delaunay triangulation, only closest nodes have been connected. In other terms, the circumcircle of each triangle does not contain any other vertex of the grid. One can exploit this property to find a second set of nodes at a user specified distance, say δ, to define the stretched region. In this region, the interpolated stretching σ decays from the user defined maximum aspect ratio Σ at the solid boundaries to an isotropic value used in the isotropic domain. If the radius of the circumcircle of a triangle with a solid boundary face is less than δ, a node normal to the face at distance δ can be introduced without any interference with other regions. The stretching at this node will be the isotropic value. Two layers interfere if the radius is smaller than δ and the triangle in question connects different solid boundaries. In that case the stretching will decay only to an appropriate fraction of the stretching at the solid boundaries. Interfering layers can be observed in figure 2 between the flaps and the main aerofoil.

In order to have a decay of the aspect ratio with increasing distance from the wall in 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 face 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 from the given values by the computer.

The spacing values assigned to these nodes can be interpolated on the background grid before the nodes that define the stretched layer are introduced. We take the spacing to be the smaller value of the interpolated spacing h and the stretched scale. The stretched scale ν is the length of the longest solid boundary face divided by the local stretching value σ as will be seen in section 3.

We find a monotonic variation between the fine spacing on interior boundaries and the coarse spacing on a farfield boundary if the background triangle connects directly from the interior to the exterior boundary. But along concave contours it may happen that Delaunay connects between finely spaced interior boundaries and the background grid will specify a too large area of fine spacing. Consistent with the philosophy of minimal user input, the program introduces the necessary nodes to break the unwanted connections. The user only specifies which boundaries he does not want to have connected. The procedure will be to detect an illicit liaison and place a background node at the circumcenter of the badly connecting triangle. During a subsequent retriangulation most if not all of the triangles shared between the two bodies will be broken and few extra nodes suffice. Due to its placement, the new node is equidistant from all ill-connected boundaries. The spacing is extrapolated from the more finely discretized boundary using the average spacing gradient of the initial triangulation.

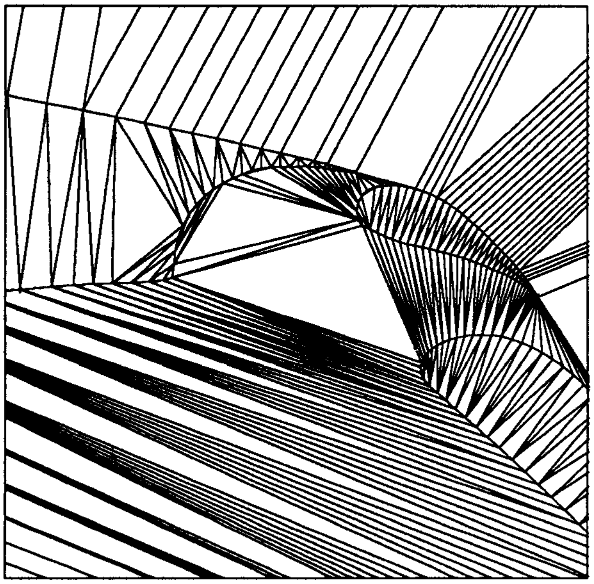

Figure 1: Initial triangulation of the boundary nodes used to derive a background grid and the interior triangulation.

3 Frontal node generation

A frontal technique is employed to construct nodes from frontal edges that refine the existing triangulation. The process of generating and connecting interior nodes for our initial triangulation of the boundaries can be distinguished into three different parts: building wedges with a scale ν in the stretched regions along solid boundaries or wakes, building triangles with the stretched scale ν in the remaining stretched regions, building triangles with the isotropic scale h in the rest of the domain. While 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.

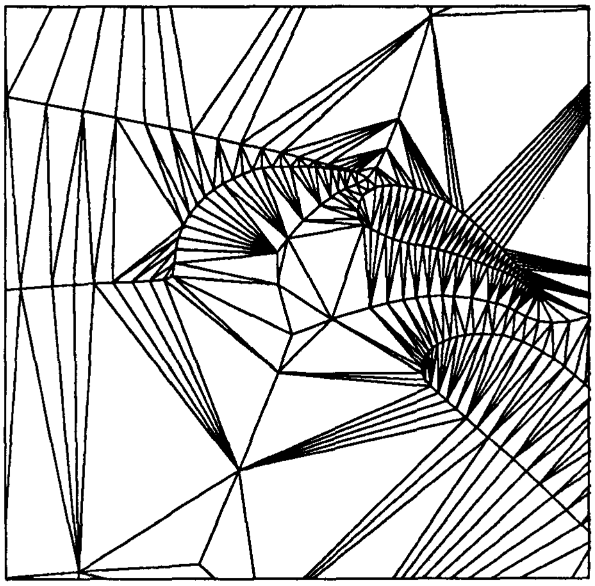

Figure 2: Background grid with $\delta = .1$ and disconnection between the main aerofoil and the main flap.

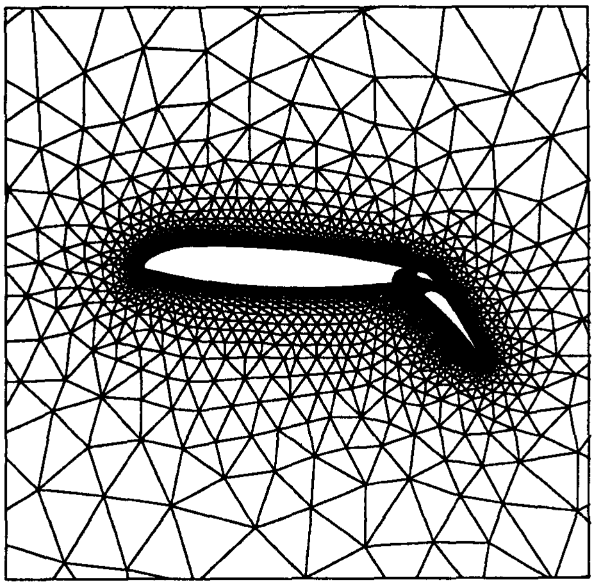

Figure 3: Closeup of a three element aerofoil with a stretched layer around the components, $\Sigma = 1 : 10$.

3.1 Building stretched wedges

Shear layers expose high gradients normal to the layer while 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 or shocks. 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. If the maximum angle approaches 180°, the truncation error in that cell becomes exceedingly high if gradients along the opposite face are present. More or less the only reliable way to avoid large angles while retaining high aspect ratios is to emulate ordered 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.

It thus seems natural to define a stretched scale ν as the length of the longest solid boundary face H divided by the local stretching value σ. Starting from the concatenated strings of nodes that define the closed solid boundaries, a string of nodes is created around each previous string with distance ν between the nodes until the stretched regions are filled. The strings serve to link the nodes such that an average surface normal can be calculated to displace the nodes 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 like e.g. 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 nodes are split around corners in the geometry as the flow features around a corner cannot be assumed a priori and isotropic refinement has to be used. The effects of reduced stretching and reversion to isotropy can be seen in figure 4 around the vane flap and at the lower right corner of the main aerofoil and in figure 5 around the leading edge.

A Delaunay triangulation maximizes the minimum angles in the mesh. On the other hand, the solvers impose constraints on the maximum cell angle rather than the minimum angle [7]. While the differences between Min-Max and Max-Min triangulations are minor in isotropic grids, the differences are fundamental in stretched meshes. Consequently, one has to apply constraints to the triangulation while introducing nodes in the stretched regions. As the possible connections between the different strings of nodes are known, the triangles 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 nodes
  open a new string of nodes
  for each node in the string
```

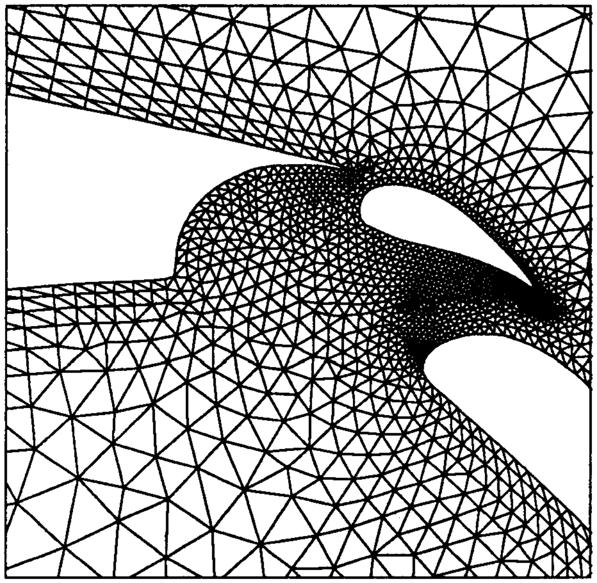

Figure 4: Detail of the three element aerofoil between the main aerofoil and the main flap, $\Sigma = 1:10$.

```
    evaluate the local stretching value
      on the background grid,
    build a new node at distance $\nu$ to
      the node normal to the string,
    check whether that node lies within
      the proper stretched region,
    check whether that node is properly
      spaced with the nodes in the grid.
    if properly spaced and located
      append the new node to the new string,
      introduce the node into the constrained
        triangulation,
      protect the newly formed wedges
        between strings.
    fi
  rof
  close the new string of nodes
rof
```

The actual implementation is slightly more complicated to properly evaluate the normal at the ends of the strings and to allow for an interruption of the string in case one of the checks fails or a corner is detected.

3.2 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 [4]. The only difference between stretched and isotropic regions is the evaluation of the local length scale. In order to match the short sides of the wedges, the length scale for triangles in the stretched region has to be ν.

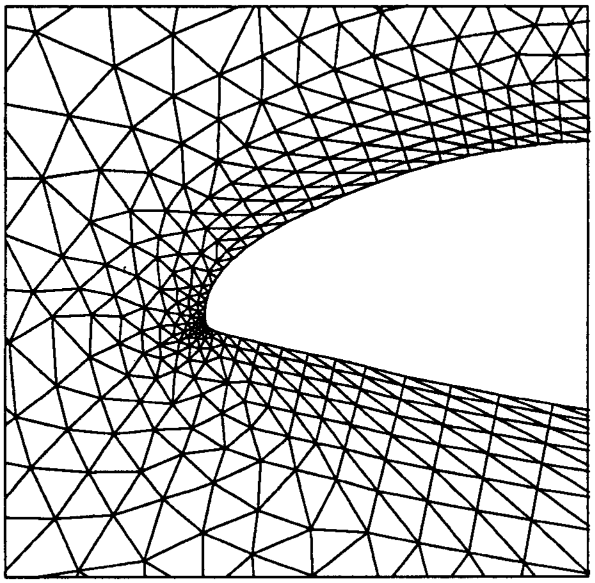

Figure 5: Detail of a three element aerofoil at the leading edge, $\Sigma = 1:10$.

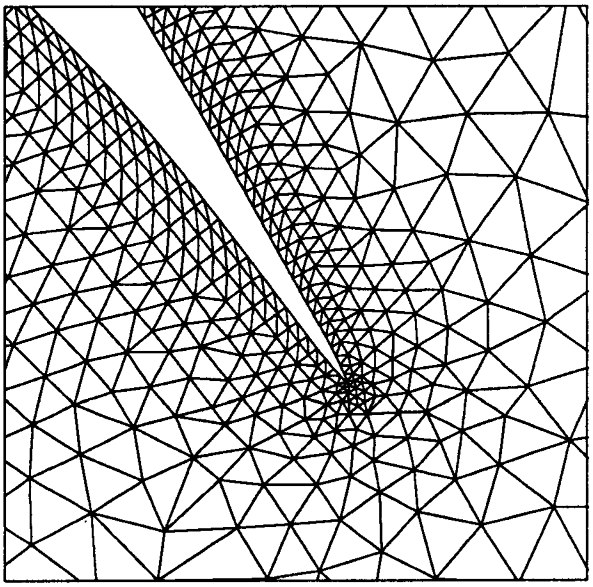

Figure 6: Detail of a three element aerofoil at the trailing edge, $\Sigma = 1:10$.

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 guaranteed by the Delaunay principle.

3.3 Building triangles in the isotropic region

In contrast to Advancing Front techniques, tracking of the front is not required for the generation of isotropic triangles as we provide over a closed Delaunay triangulation at any stage of the process. A frontal edge is defined as a face shared between a well shaped cell with all three sides of similar length and a badly shaped cell with an obtuse or acute angle opposite to the shared edge. For each frontal edge a new node is constructed on the median into the badly shaped cell such that the distances between the new node and the two vertices forming the edge approximate h, the isotropic length scale. A further check is required to make sure that the new node is sufficiently distant from the remaining nodes in the grid and from the other new nodes. For the distance check with the nodes that are already introduced into the structure, the Delaunay properties can be used to limit the search area to a few triangles that surround the one containing the new node. Once a spacing violation with old nodes is detected, the new node is discarded. The check with the other nodes waiting to be introduced is extensive, unless an efficient bucketing data structure is used. New nodes that are found to be too close to each other are merged. With these new nodes in place, the Delaunay algorithm is re-run and the process can be repeated until all bad triangles have vanished. More details about the isotropic node generation process and examples can be found in [3, 4]. Hence, the algorithm can be cast into the following steps:

```
do while new nodes are found
  for each triangle in the grid
    for each face of the triangle
      if this face is frontal
        find a node to form a triangle
          with the face,
        check whether the new node is not
          too close to any other old node,
        check whether the new node is not
          too close to any other new node,
        if the node is well spaced
          append the node to the list
            of new nodes
        fi
      fi
    rof
  rof
  for each new node
    introduce the node into the structure
  rof
od
```

The computational cost of the isotropic node generation process can be shown to be asymptotically optimal as the number of operations required is of $O((p+1)N\log N)$, where $0 \leq p \leq 1$ is the fraction of new nodes created at a particular frontal step. With a tree data structure one finds $p=0$ as the cost of searching through the list of new nodes reverts to $O(N\log N)$. For an Euler grid around the three element aerofoil configuration in figure 3, 1943 nodes were created in 10.1 seconds on a DEC 5000. The Navier-Stokes grid around a NACA 0012 aerofoil with an ensuing wake given in figs 7,8 contains 2526 built to form wedges and 759 nodes built isotropically. Constructing the nodes for the wedges took 5.4 seconds, constructing the isotropic nodes took 4.5 seconds on a DEC 5000. Thus the stretched generation process runs three times as fast, although the implementation is not yet optimal. Further improvements in efficiency can be achieved with the use of an tree data structure.

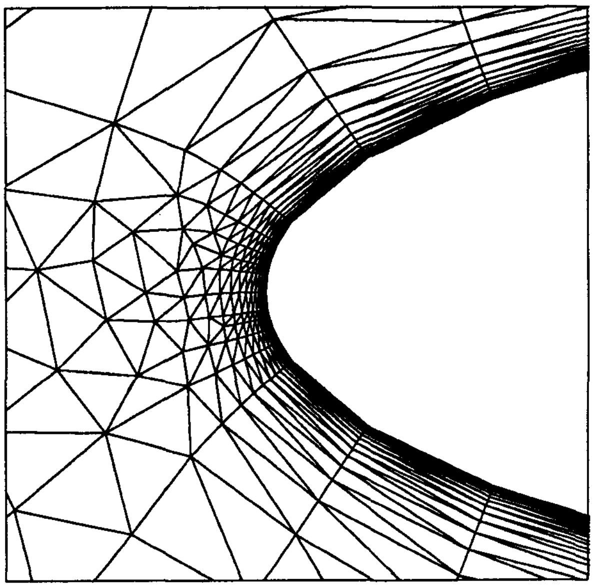

Figure 7: Detail of a NACA 0012 aerofoil at the leading edge, maximum aspect ratio 1:1000.

4 Examples

A classic case for an unstructured grid generator is the grid around a multi-element aerofoil. Structured grid generation already requires sophisticated extensions to deal with this problem. The background grid for the aerofoil given in figure 2 was modified from the intitial triangulation shown in figure 1 by the automatic insertion of 97 nodes to form the stretched layer and one node to disconnect the main aerofoil from the main flap. The thickness

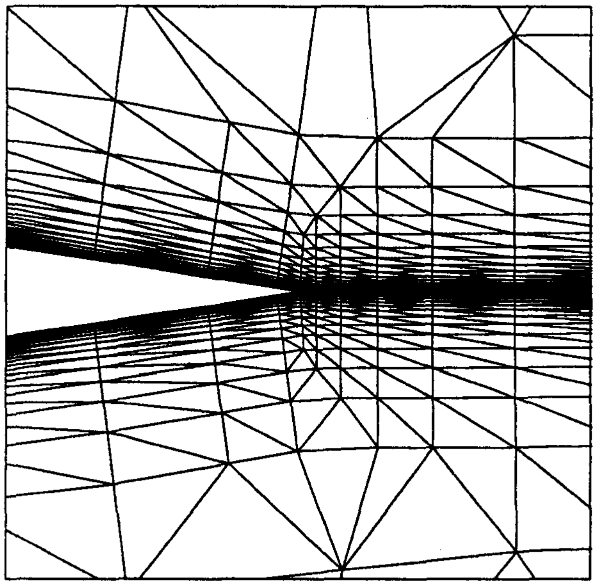

Figure 8: Detail of a NACA 0012 aerofoil with a wake, maximum aspect ratio 1:1000.

of the stretched layer δ is given as .1 chordlengths, the maximum aspect ratio Σ is $1 : 10$. At the midsides of the main aerofoil, where the maximum face length H is found, 8 layers of wedges with constantly decreasing aspect ratio have been generated. The "viscous" process was interrupted at the lower right corner due to the geometry jump (figure 4). No wedges were generated around the vane flap and only very few around the main flap due to the fine discretization. The region between the three aerofoils is filled isotropically. At the leading edge (figure 5) the node generation process reverts to isotropy as the node spacing on the boundary was chosen very fine and, hence, the stretched scale ν exceeded the local mesh spacing h. A fully stretched leading edge would be obtained for a higher Σ as can be seen in figure 7.

The regularity of the grid is entirely due to the frontal insertion, no smoothing filter was applied. Overall, the cell surface varies very smoothly with a factor of about 100 000 from the smallest cells at the trailing edge of the vane flap (figure 4) to the largest cells at the outer boundary. The algorithm proves to be very robust, as can be seen from the regularity of the triangulation of the lower rear corner and the trailing edge of the main aerofoil in figure 4 and the trailing edge of the main flap in figure 6.

An example of a higher maximum aspect ratio is given in figures 8 and 7, a NACA 0012 profile with an ensuing wake and $\Sigma = 1 : 1000$. As the wake picks up the stretching of the main aerofoil, the spacing of the boundary nodes at the trailing edge does not have to revert to isotropy. The node distribution has been refined at the leading edge in a rather blunt, binary way to show that the smoothness of the grids does not depend crucially on a smooth boundary point distribution as the isotropic node generation process combined with a Delaunay insertion shows some smoothing properties.

5 Angular bounds for the isotropic frontal node insertion

As the generation and introduction of nodes in FroD is embedded in the rather rigorous mathematical framework of the Delaunay triangulation, the algorithm can be analysed mathematically. On the other hand, for the more heuristic Advancing Front method [2], no derivation of angular bounds or bounds in surface variation is known. The smoothness of the grids created with the Advancing Front method relies on aposteriori smoothing.

Upper and lower bounds for the angles in the interior isotropic triangulation obtained with FroD can be derived on the assumption of vanishing spacing differences at the nodes involved in the construction process. Although this simplification seems rather restrictive, it only comes into effect in the outer flowfield when the cell size is large and, hence, the solution gradients are small and large angles are permissible. The other constraint actually is more important. To ensure boundary conformality and to preserve the connectivity of the stretched triangulation, the Delaunay principle is violated locally. In other words, close to these interfaces there may be triangles that contain other vertices than their own in their circumcircle and the proofs given underneath may fail. However, the Delaunay property will be respceted if the local boundary discretization reverts to the isotropic scale ν at the ends of the solid surfaces and to the isotropic spacing h at the outer edge of the stretched layers.

5.1 Upper angular bound

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

In this way, one has to consider as a worst case a triangle with non-desirable properties that cannot be refined because the local point density is already too high. Figure 9 shows this case: the obtuse triangle ABC with a circumcircle O of radius r. As this triangle is Delaunay, there is no other node of the triangulation in O and we can

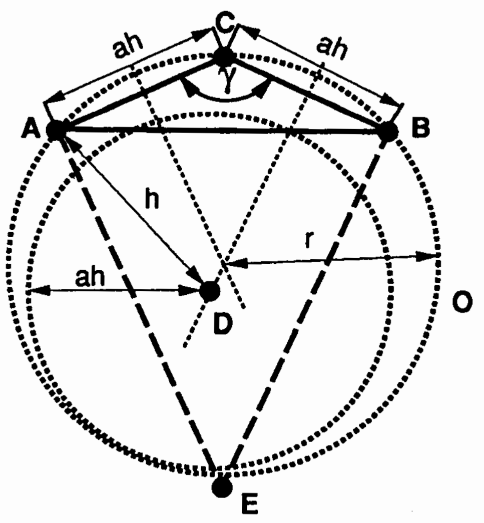

Figure 9: Obtuse triangle with maximum angle.

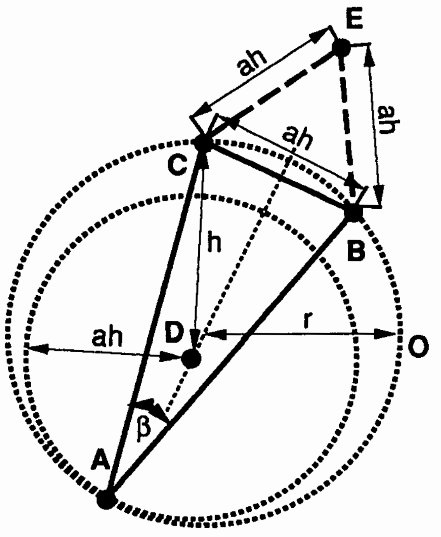

Figure 10: Acute triangle with minimum angle.

always introduce the new node D if its spacing disc with a radius ah around D is contained in O. Consequently, the worst case occurs when the two short faces of the triangle ABC have the minimum permissible length of ah. If we were to increase this length, r was to increase as well and the disc around D was included in O already at a smaller angle γ. Hence, given the tolerance a we can calculate the maximum angle γ:

$$\frac{r}{h} = \frac{1}{2}\sqrt{\frac{a^2h^2}{4} + \frac{h^2}{4}\left(\sqrt{1-\frac{a^2}{4}} + a - \frac{a^2}{4\left(\sqrt{1-\frac{a^2}{4}}+a\right)}\right)^2} \tag{1}$$

$$\gamma = 2\arccos\frac{ah}{2r}. \tag{2}$$

The lower limit is obtained for $a = 1$ with $\gamma = 120°$. However, this restrictive tolerance will inhibit the insertion of D if $\nabla h = 0$ as D then is too close to C and B. The maximum angle increases monotonically to 141° for $a = .5$.

5.2 Lower angular bound

Similarly, one can derive a lower angular bound when refining acute triangles with one short face. This case is depicted in figure 10. As long as the circumcircle O is large enough to contain the spacing disc around D, refinement of the triangle ABC will always be possible and the acute angle β will increase after the insertion of D. Again, as one further decreases β in ABC, O will shrink until the limiting case is reached where O and the rim of the spacing disc around D become tangent. In this case we find

$$\beta = 2\arctan\frac{1}{\sqrt{\frac{4}{a^2}-1}+2}. \tag{3}$$

The smallest minimum angle is found for $a = 1.$ as $\beta = 30°$ and decreases monotonically to $\beta = 19.3°$ for $a = .5$.

5.3 Upper bound in surface variation

The two cases shown in figure 9 and 10 also represent the two limiting cases in surface variation between two neighboring triangles. In both cases an enlargement of the bigger triangle would enable refinement of the configuration. The case A of figure 9 of an obtuse triangle next to a slightly acute one is the case that will produce larger variations in size:

$$\frac{S(AEB)}{S(ABC)} = \frac{2r}{ah\cos\frac{\gamma}{2}} - 1. \tag{4}$$

The maximum surface ratio for this case ranges from 3. for $a = 1.$ to 8.1 at $a = .5$.

Case B, as given in figure 10, is formed with a larger triangle that has an acute angle with the minimum β. As the face AB has the minimum allowable length in a triangulation of ah, the triangle ABE must be equilateral. Hence, the ratio in surfaces in case B is

$$\frac{S(ACB)}{S(ABE)} = \frac{1}{\sqrt{3}\tan\frac{d}{2}} \tag{5}$$

The ratio for case B ranges from 2.15 for $a = 1.$ to 3.39 for $a = .5$.

5.4 Measured bounds of angles and surface ratio

The numerical experiments with various a performed with isotropic grids around the three element aerofoil of figure 3 strongly confirm the upper angular bound as can be seen in figure 11. However, the maximum angle found in the grid does not increase monotonically with decreasing a: a local minimum at $a = .6$ is found.

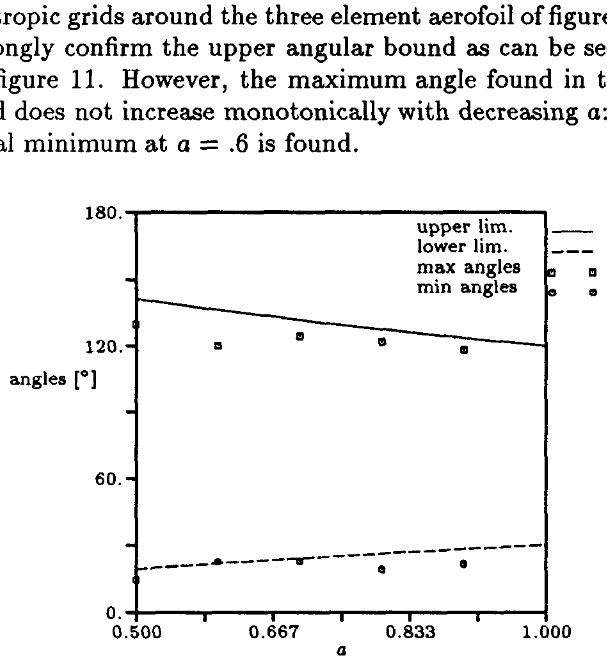

Figure 11: Measured maximum and minimum angles and lower and upper bounds in function of the tolerance a.

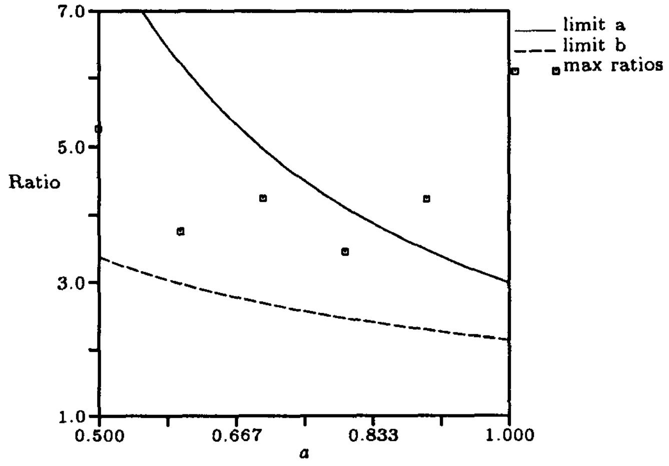

Figure 12: Measured surface variation and bounds of surface variation for the case A and B in function of a.

The lower angular bound is not as well observed as the upper angular bound. The minimum angle in the triangulation remains at roughly 20° except for the lowest $a = .5$. Again, a rather favorable value is found for $a = .6$. The disparity might be due to the fact that in the case of an acute triangle with two long faces the assumption of negligible gradients in spacing does not hold as well. Still, the lower bound derived gives a good estimate for the minimum angles.

The limit in surface variation of case A is well observed except for the case with the narrowest tolerance of $a = .9$ (figure 12). Again, a local minimum in surface variation is found for $a = .6$.

5.5 Distribution of angles and surface ratios

More important than maximum values of the above parameters are their distributions over the grids. As to be expected, a narrower tolerance might decrease the largest angle of the grid, but will not improve grid quality as a whole. Distributions of minimum and maximum angles and surface ratios over the entire grid can show the grid quality better. As the figures 13-15 show, a narrow peak close to the optimum values is found for a rather large tolerance $a = .6$ with a very small number of cells with excessive angles $\gamma \geq 90°$ or a surface variation larger than 2. On the other hand a small tolerance of $a = .9$ will produce a broader variation with a larger number of undesirable cells. Still, even the distributions with $a = .9$ are very satisfactory.

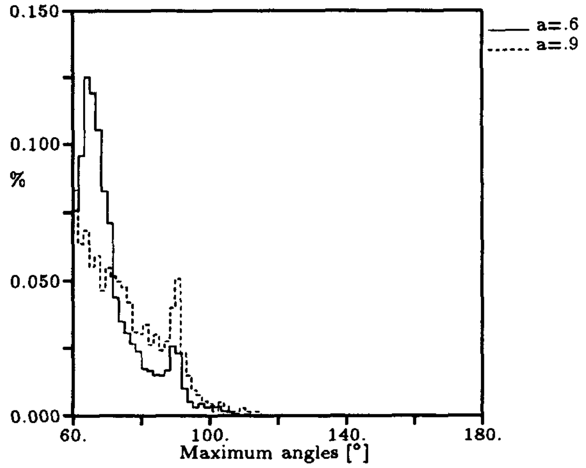

Figure 13: Distribution of maximum angles.

6 Conclusions

The Frontal Delaunay Method, FroD, has been extended to be used with stretching surfaces like shocks or wakes

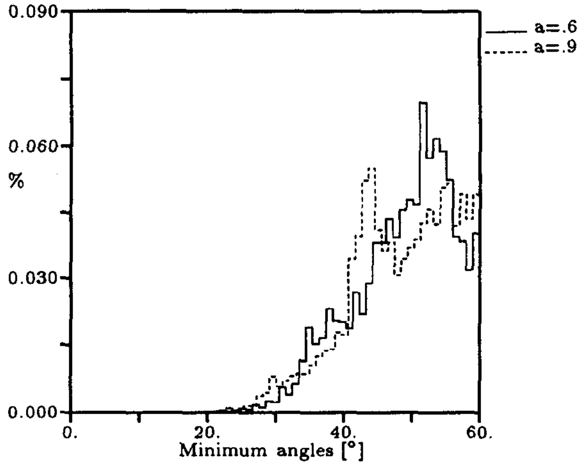

Figure 14: Distribution of minimum angles.

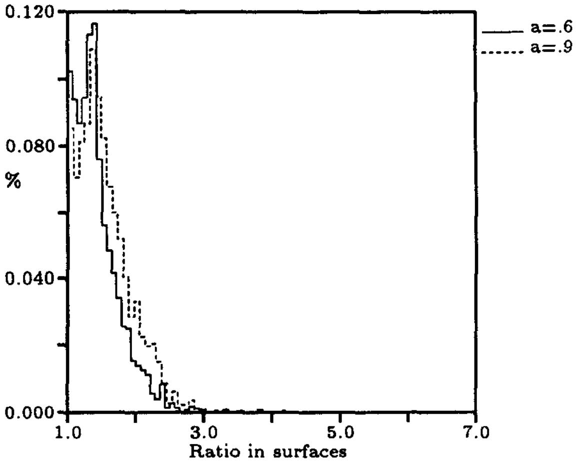

Figure 15: Distribution of surface ratios for $a = .6$.

that are not solid surfaces. Also realistically high stretching values have been applied successfully. With boundary and wake discretizations that revert to isotropy at their ends, a smooth transition to the triangular frontal process is found and the high mesh quality of the isotropic part of the algorithm is retained. Within the stretched layers, strong angular control is guaranteed as the process reverts to isotropic constructions at points of high curvature.

For the isotropic node generation process upper and lower angular bounds and upper bounds in surface variation between adjacent cells were derived and confirmed with sample calculations. Furthermore, statistical measures show that only very few cells approach these bounds while most of the cells exhibit values close to the smallest possible maximum angle and the largest possible minimum angle.

As in the purely isotropic case, only minimum user input is required for the stretched grids. Besides the set of boundary nodes, 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] D. Mavriplis. Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation. *ICASE Report No. 88-47*, 1988.

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

[3] J.-D. Müller, P.L. Roe, H. Deconinck. A Frontal Approach for Internal Node Generation in Delaunay Triangulations. *Int. J. of Num. Meth. in Fluids*, submitted.

[4] J.-D Müller, P.L. Roe and H. Deconinck. Delaunay-based triangulations for the Navier-Stokes equations with minimum user input. *13th ICNMFD, Rome*, June, 1992.

[5] J.-D. Müller, P.L. Roe, H. Deconinck. A Frontal Approach for Node Generation in Delaunay Triangulations. *AGARD R-787*, 1992.

[6] B. Delaunay. Sur la sphère vide. *Bull. Acad. Science USSR VII: Class. Sci. Mat. Nat. 793-800*, 1934.

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

Acknowledgements

The author is very grateful for the generous and selfless support received by Prof. Phil Roe at the University of Michigan and Prof. Herman Deconinck at the Von Karman Institute.