THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
345 E. 47th St., New York, N.Y. 10017

97-GT-91

Automatic Mesh Generation by a Mixed Advancing-Front Delaunay Approach

Roque Corral†
Dept. of Methods and Technology
Industria de TurboPropulsores, S.A.
28850, Madrid, Spain

Juan Manuel Barroso
Dept. of Aerothermodynamics and Systems
Industria de TurboPropulsores, S.A.
28850, Madrid, Spain

Manuel Antonio Burgos†
Dept. of Methods and Technology
Industria de TurboPropulsores, S.A.
9628850, Madrid, Spain

ABSTRACT

A simple, three-dimensional, unstructured grid generation system is discussed and relevant results for turbomachinery and exhaust systems applications are presented. The method makes special emphasis in algorithm selection based in a small development effort to grid quality ratio. Engineering decisions based on state of the art grid technology knowledge have been adopted to balance fast development progress and turnover time.

The system is based in the sequential execution of Advancing Front and a Delaunay algorithms to produced solution adapted meshes. In a first step a coarse background grid is created by the Advancing-Front method. During this phase boundaries are strictly preserved due to the nature of the approach. At this stage a Delaunay algorithm is employed and the boundary faces are blocked so that they cannot be destroyed in further triangulations. Since both techniques collaborate in the solution of the problem it is possible to write simplified versions of the algorithms that retain nearly all the benefits and eliminate most of the troubles while keeping still reasonable performances.

NOMENCLATURE

AF — Advancing Front
D — Delaunay
K^{AF} — proportionality constant of the complexity of the Advancing Front algorithm
K^{D} — proportionality constant of the complexity of the Delaunay algorithm
N — total number of nodes in the domain
n — number of refinements
s — element size distribution function
$\partial\Omega$ — boundary domain

†Also at the School of Aeronautics, U. Politécnica, Pl. Cardenal Cisneros 3, 28040 Madrid, Spain

INTRODUCTION

It has been acknowledged for a long time that the bottleneck in the numerical simulation of complex configurations is the geometry and grid generation system. With the availability of high-speed supercomputers the computation of flow fields about complex configurations is possible in a matter of hours. However the process of generating a grid using conventional structured grid methods may signify a much larger portion of the total computational time. While in structured grids mesh lines must conform to the boundaries, such restriction does not exist in unstructured grids due to their lack of directionality paving the way for the incorporation of adaptive refinement and local remeshing. Therefore, in spite of the larger requirements of memory and computing time of unstructured methodology caused by the indirect addressing of data (Formaggia et al., 1988), and of the possible accuracy penalty if the grid lacks of enough regularity, the benefits derived from the flexibility and automation possibilities offered by unstructured meshes overcome the above mentioned inconveniences. On the other hand the advantages of a lower development and maintenance cost that stem from the blind character of the unstructured methodology should not be underestimated.

Traditionally two different strategies have been perceived as competing approaches to solve the same problem, Advancing-Front (Peraire et al., 1987) and Delaunay triangulation techniques (Baker 1990). However, while the former constitutes a point placement strategy the latter refers to a special connectivity, and therefore, several attempts to combine the advantages of these methods have appeared in the literature (Muller et al, 1992 and Mavriplis, 1995). In order to understand the development of the present approach, it is useful to analyze the limitations an benefits of the existing algorithms.

Advancing front techniques operate in a edge/face basis, depending on whether we are working on 2D or 3D. The first step is to discretize the boundaries to form the edges/faces of the initial front. Then a particular edge/face of the front is selected and a triangle /

Presented at the International Gas Turbine & Aeroengine Congress & Exhibition
Orlando, Florida — June 2–June 5, 1997

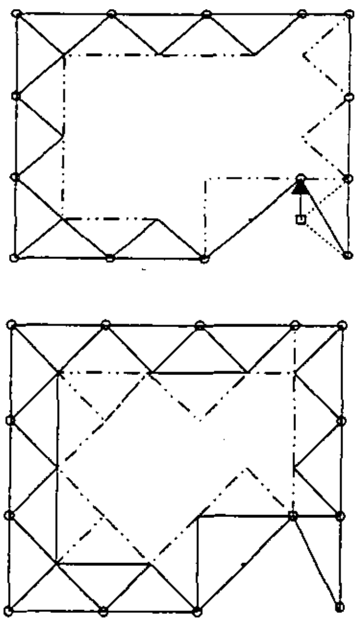

Figure 1: Sketch of two different steps in the triangulation process of the Advancing Front method. Solid lines represent the existing front, dashed lines represent elements to be added to the front and dotted lines portray ideal triangles that do not conform the front and are rearranged during the checking phase.

tetrahedron, from here on referred to as elements, is formed by joining the base, that is in the front, with a node that is either one of the existing in the front or a new one that is added if there is not a suitable node in the front that satisfies some prescribed optimal criteria (based usually in a element size function and equiangularity). Finally an intersection checking process is carried out to verify that the newly formed element does not interfere with the already existing (see Fig. 1 for an outlook of the method). Once the element is accepted the front is updated and the process goes on till the front is empty. The phase of intersection checking is rather time consuming since involve non local searching procedures and therefore the main limitation of this sort of algorithms is its lack of efficiency. However they have a built in point placement strategy, the final quality of the meshes is high and the boundaries of the domain are automatically preserved.

Given a set of points a Delaunay construction represents a unique, among the many possible, mode of triangulate these nodes that holds a number of well-defined properties like the empty circumcircle property, which states that no other point of the triangulation lies in the circle/sphere defined for the 3/4 points that form an element. This fact has been successfully used as the basis of the Bower/Watson algorithm (Bowyer 1981 and Watson 1981). Therefore, given an initial triangulation, a new point may be inserted by first deleting all the elements whose circumcircles contain the new node and connecting this to all the vertices of the created cavity. (see Fig. 2). A summary of other interesting properties can be found in Barth (1992). The Delaunay approach seems to be attractive because of its strong mathematical basis and freedom from excessive global search procedures, however it does not guarantee that the edges/faces of the domain will be contained in the final triangulation. In the 2D case a constrained Delaunay triangulation, i.e. a triangulation where some edges are prescribed *apriori*, always exist, however the same it is not true in 3D.

Grid generation without an *apriori* knowledge of the field solution can not be accomplished without recurring to spacing functions provided by the solver, actually although a stand-alone unstructured grid generation system be used, the boundary and internal point distribution relies in a implicit understanding of the physics of the problem. As a result several attempts of coupling error indicators provided by the solvers with grid generation systems have been done. The most popular approach in the unstructured grid context is based on grid enrichment., in this case, the problem is addressed by adding new points in areas of high gradients and computing the steady solution in a sequence of denser grids (Jennions, 1994). As we will see this fact can be taken into account in the design of the grid generation system.

This paper presents the use and the design criteria of a novel grid generation system, based in the sequential operation of an Advancing Front and a Delaunay algorithm, able to produce solution adapted meshes about three dimensional configurations. At the beginning a coarse background grid is created using a simplified Advancing-Front method. In this phase boundaries are strictly preserved due to the nature of the algorithm. At this stage boundary faces are blocked so that they cannot be destroyed in further Delaunay triangulations. This procedure guarantees strict boundary preservation without surface recovery and ensures periodic meshes for 3D turbomachinery computations

TWO-DIMENSIONAL GRID GENERATION

At present constrained Delaunay triangulations in two dimensions do not pose notable difficulties, either from the point of view of algorithmic understanding or its efficient implementation. It is the authors perspective that this sort of algorithms are simpler and easier to develop than their Advancing-Front equivalents. Taken into account that they are faster as well, these methods do not have a true competitor. A number of different algorithms have been proposed for Delaunay tessellation; Green and Simpson (1977) and Lawson (1977) emphasize on constructions based in edge swappings while, Bowyer (1981) and Watson (1981) rely on the incircle property. Enforcing the prescribed geometry is usually done in two post procesing steps. First the edges that define the geometry are required to exist. We have followed here the approach of Sloan (1993) where an arbitrary set of edges, non necessarily conforming a boundary, is forced to exist by executing a sequence of swappings. This feature can be used to secure the existence on the mesh of some prescribed curves of engineering interest (e.g. the cascade throat) though similar results could be obtained in a number of less direct ways. In a second step all the triangles placed outside of the defined domain are flagged making use of the orientation associated to the closed circuits that define the different domains and of a consistent ordering of the nodes forming the triangles

In practice, each time a point is introduced in the existing triangulation, a search must be performed to find the intersecting triangles. In order to speed up the process we make use of search techniques based on "walking" algorithms and in these cases it is important to retain the mentioned exterior cells in order to guarantee that any element can be reached walking from any other. This type

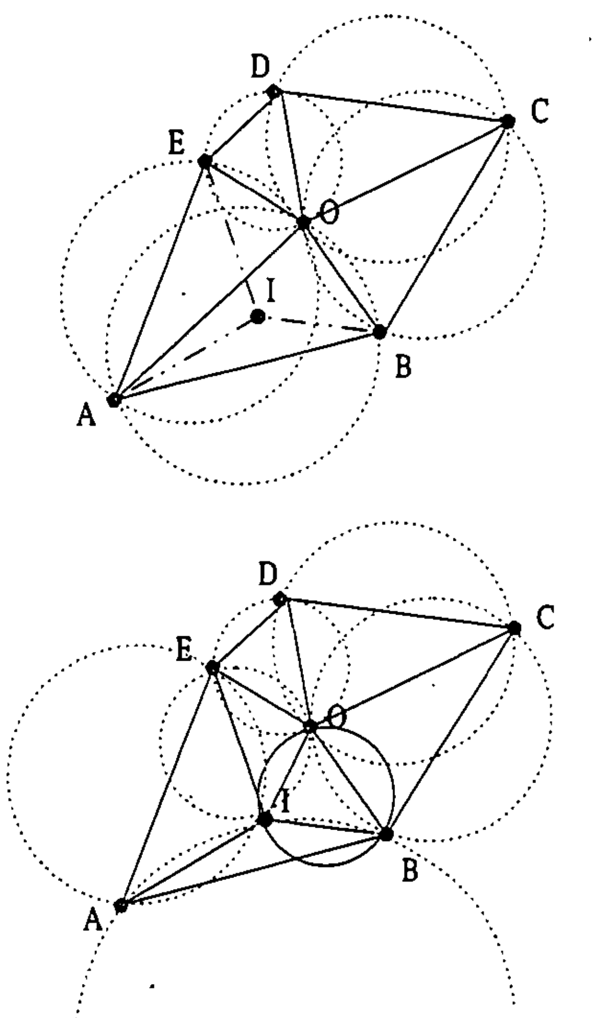

Figure 2: Point insertion sequence in a Delaunay algorithm. The insertion of node I is performed in two steps. First the triangles that do not fulfill the incircle property (ABO and AEO) are found and destroyed (top). Then new triangles are formed by joining the new point, I, with the vertices of the created cavity, A, B ,O and E. (bottom)

of techniques are very efficient and for randomly distributed cells the average point insertion requires $O(N^{1/2})$ walks that results in an average complexity of $O(N^{3/2})$. In practice, however, it is easy to observe complexities of $O(N)$ if the initial guess is close enough to the sought triangle.

Interior points of the domain are computed making use of a function that provides a measure of the desired size of the triangles. This element-size function is built just from the information contained in the point distribution of the boundaries and as far as we know this is a novel concept. The first step is to compute the function in the nodes of the boundaries as the mean of the distances of the two edges that meet in the node and to mesh the domain making use of the boundary nodes only. Then new points are added in the centroid of the triangles. This operation is performed twice in order to obtain at least 3 nodes between two opposite boundaries. The element-size function, s, is initialized by weighing its value in the nodes of the

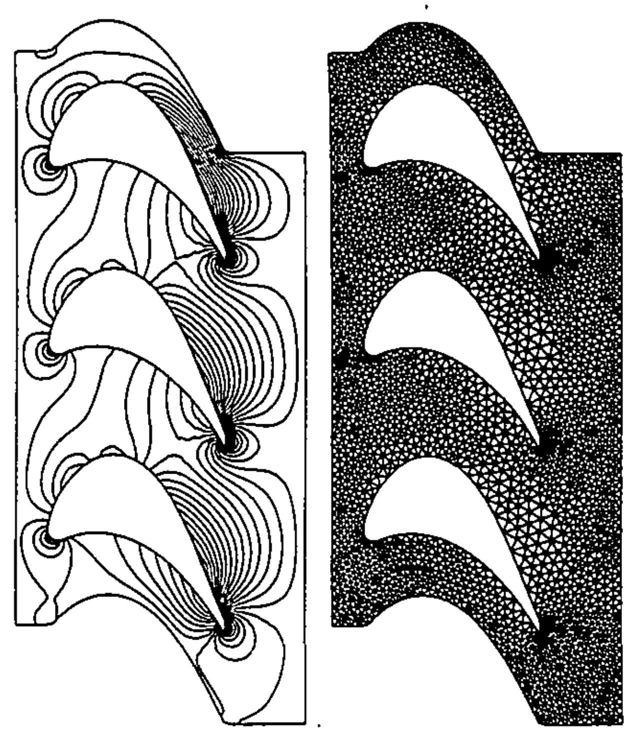

Figure 3: Final element size distribution (left) and inviscid grid (right) in a typical configuration.

triangle with the distances from the new point to them, then the node positions and the element size function itself are smoothed by a Laplacian filter, in this manner the initial coarse grid is improved and s tends to verify the Laplace equation in the current grid. At this stage the nodes are allowed to be reconnected by the Delaunay algorithm to improve the mesh connectivity. The process of inserting, smoothing and reconnecting new points is repeated until convergence. When nodes are moved due to the effect of the filter, s is convected with them and actually this value is used as a the new initial condition in the process of solving the Laplace equation by successive relaxations or smoothing steps. It can be shown that this procedure is equivalent, to first order, to solve the Laplace equation for s with Dirichlet boundary conditions,

$$\nabla^2 s = 0 \text{ with } s = s_0 \text{ at } \partial\Omega. \quad (1)$$

A typical result is shown in Fig. 3. where the final element size function is depicted. The node concentration on the boundaries is proportional to the local curvature weighted with a uniform distribution to obtain a more homogeneous spacing in zones of very high gradients (e.g. corners, ...). Note that a smooth transition between zones of different size is obtained with a minimum user input. The resulting element size field contour lines correlates somehow with the isocontours of the modulus of the expected velocity far away of the boundaries. The method is readily extensible to 3D and eliminates the need of using background grids. This process is inexpensive, easy to implement and the results are good enough for engineering purposes. Figure 4 shows the statistics obtained for this configuration. Its angle distribution compares well with other 2D Delaunay results (Marcum and Weatherill, 1995), however, it is well known that point-placement strategies based on AF are superior in terms of grid quality and the present methodology is not an exception to this fact.

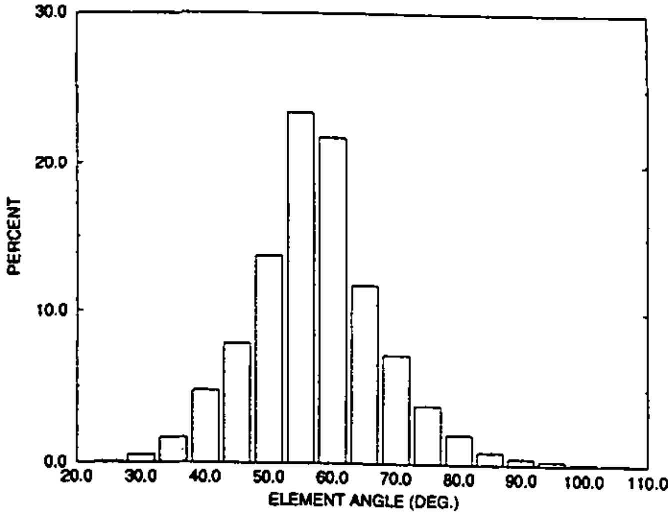

Figure 4: Cell angle distribution for the configuration of Fig. 3

THREE-DIMENSIONAL GRID GENERATION

On the contrary that in the case of the solvers, the task of developing a unstructured 3D mesh generator is at least an order of magnitude more complex than the two-dimensional case. The number of possible configurations and options is much greater and thus, the likelihood of committing a mistake either in the conception or the programming of the algorithm is also much higher. It is our personal experience that a 3D mesh is harder to attain following Delaunay than Advancing Front techniques. The origin of the problem is that while in 2D there exists a clear concept of a constrained Delaunay triangulation and clean and efficient algorithms, that make use of edge swapping, are readily available (Sloan, 1993) the same is not true in 3D and the situation is somewhat more complex. Here the boundary integrity has to be recuperated in two phases: First the recovery of the edges, and then the recovery of the boundary faces (George et al., 1988, and Weatherill et al, 1993), however the insertion of additional points is often required to complete the procedure. Once the surface has been recovered, the new boundary points can be removed, but in some cases, this phase can give rise to instabilities. Another possibility is the iterative insertion of points on the destroyed boundary to force the existence of the desired edge or face. One of the latest attempts in this direction is due to Dawes (1996). The main disadvantage of the latter approach is that if several length scales are involved in the vicinity of the broken boundary, a number of points proportional to the size of the minimum of the present lengths may be required.

In order to avoid these difficulties we have wind up using an Advancing Front approach based in the ideas of Peraire et al. (1987). The aim of this idea will become clear in the following lines. It is well known that the main disadvantage of the Advancing Front approach is related with its lack of efficiency in the intersection checking phase, that is computationally very intensive although advanced algorithms for searching and sorting be used. Bonet and Peraire (1990) have shown that even in 3D the problem can be solved in Nlog(N) operations by using alternating digital tree data structures. However, the computational times obtained with this technique are still high when compared to the ones obtained by Delaunay triangulations. Using the information provided by Marcum and Weatherill (1995), Weatherill (1992) and Pirzadeh (1994), extrapolating all the data to a common architecture and assuming that for the same number of elements a 3D configuration is between twice and three times more expensive to grid than a 2D configuration, it can be inferred that though both approaches behave linearly in practice, the slope of the curves is at least an order of magnitude greater in the Advancing Front than in the Delaunay approach and therefore:

$$K^{AF} \cong 10K^{D}, \quad (2)$$

however, if the ultimate goal is to perform grid adaptation based in the solution provided by a solver of the Reynolds Averaged Navier-Stokes equations (RANS) by means of successive enrichments, the initial grid could be very coarse and become denser as the solution is formed, the areas of high gradients develop and the resolution is increased in these zones. In this case, how the initial coarse grid is generated is not crucial from the point of view of the efficiency of the global method since this mesh can be very rough.

Our approach consists in using the Advancing Front technique to generate an initial grid. Once a valid mesh is generated we have built a Delaunay algorithm based on edge and face swaps that guarantees that the boundaries are preserved during the next steps. In this way a rather simple, not sophisticated Advancing Front grid generator can be use instead the complex 3D boundary recovery algorithm of Weatherill et al. (1993), or other less efficient techniques.

Strategy

Let us imagine that a coarse grid already exists for a certain domain and that we are interested in refining the grid in some prescribed cells. Several strategies can then be devised with the present tools at hand.

If the final goal were to obtain a homogeneously refined grid we would h-refine the whole mesh subdividing each cell in eight tethaedra and then we would force the new grid to be Delaunay in a sequence of swaps and smoothings that could be regarded as a post processing step. The quality of this mesh would be nearly as good as the original, since if the new tethaedra are obtained placing nodes in the mid-edges, half of these cells will be similar to the previous and the statistics of the refined grid will be close (it would be the same in a 2D case) to the one of the coarse grid that was generated with the Advancing-Front method. It should be recalled here that meshes generated in this manner are known to possess a high quality in terms of angle and skewness distribution and that this characteristic is somehow inherited by the refined grid. The quality of the surface mesh, measured in terms of angle distribution, obtained by this procedure, is exactly the same than the original. The operation count for the generation of the volume grid will be:

$$t_{cpu} \propto K^{AF}N_C + 8^n K^D N_C \quad (3)$$

and therefore is enough that the number of refinements, n, be greater than one to neglect the computational time associated to the AF part of the algorithm. It is necessary however to consider that since the AF has not been optimized the ratio between both, the Delaunay and the AF method, is no longer linear and Eq. (2) does not make sense anymore. Another possibility is to use the AF mesh generator preventing him to insert any point, in this functioning mode the algorithm uses only the boundary points as nodes of the front and the cpu time associated to the generation of the initial grid would be lowered to $K^{AF}N_C^{2/3}$, although we have experienced robustness problems in completing such meshes in some cases. However in

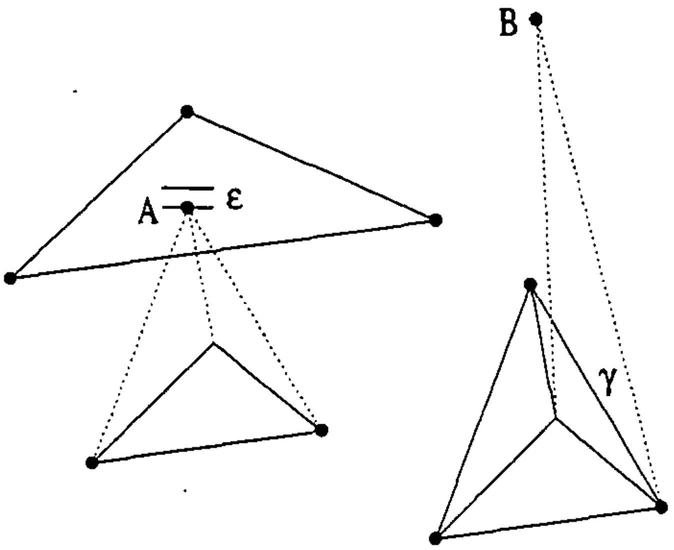

Figure 5: Sketch of configurations considered as intersections by the AF algorithm if ε and γ fall below a given tolerance.

configurations as the one shown in Fig. 6 we have been able to generate such initial grids. Nevertheless, we consider that the algorithm, at the moment, is not reliable enough to be used in this way, though this possibility remains as a source for potential improvement.

On the contrary, if the objective is to perform a series of local refinements based on the field solution gradients, or any other criterion, the simplest and fastest way is to produce an h-refined surface grid with the corresponding refinement in the inner neighbor cells. The faces created on the domain boundaries can be easily blocked, since have been created by subdivision and are perfectly identified. At this stage the Delaunay algorithm is ready to be used again. With this technique the penalty induced by the Advancing Front part of the algorithm is negligible, since it is used just once in initial mesh, when the number of elements of the grid is several orders of magnitude smaller than the final. However the quality of some transitional elements created on the surface and volume grids with this methodology is poor though is widely found and admitted in the literature.

If a higher quality is pursued, the new points on the surface have to be allowed to reconnect by the surface grid generator. This causes the information of the cells that had been previously blocked to be lost and therefore it needs to be reconstructed. In this case, some layers of cells adjacent to the boundaries are removed, and this small volume is gridded again with the Advancing Front technique. If the interest is in doubling approximately the number of nodes in each direction and N is number of points before refinement, then the number of nodes on the updated surface can be estimated as $O(4N^{2/3})$. Since the implementation of the algorithm is not optimum, the complexity of gridding this slice close to the surface should be estimated as $O(16K^{AF} N^{4/3})$ and the operation count to grid the rest of the domain as $O(8K^{D}N)$. Because we have estimate $K^{AF} \cong 10\ K^{D}$, the extra cost associated to the computation of the grid near the boundaries is $O(20\ N^{1/3})$. However if a efficient procedure is implemented to check intersections in the Advancing Front part of the algorithm the overhead associated to the boundaries can be reduced to $O(5\ N^{-1/3})$ and the procedure recovers part of its attractive. This latter technique of meshing, though possible, is not advisable since an inefficient AF is forced to grid a narrow volume with a large number of nodes.

Figure 6 shows a configuration triangulated with the primitive algorithm

Robustness

The critical phase of Delaunay algorithms takes place during the early stages of the gridding process, when elements of very high aspect ratio are likely to be present and give rise to a lack of accuracy in the computations due to round off errors. The symptoms are usually the appearance of negative cells generated because of the inability of the algorithm to decide whether a domain is convex or not. This malfunction occurs even if the final mesh does contain any of those intermediate and elongated cells. However, since the initial meshing is performed by the AF algorithm, when the Delaunay triangulation is employed the aspect ratio of the existing grid is usually low and we have not experienced special problems in this sense. Therefore, to speak about the robustness of the present method we shall concentrate on the AF part of the algorithm.

The more serious concern with the AF is the closure problem (i.e. the difficulty in completing the volume triangulation), as one of the reviewers pointed out. It is our opinion that this problem is somehow independent of how is the initial surface grid and that sooner or later a set of slivers, that has to be closed, is formed, independently of how smooth is the initial front. Two types of problems can be distinguished, the merging of fronts of different size and the closure of the last, and usually small, remaining volumes. The former is related with grid quality issues, since quite distorted cell are prone to appear, while the latter has to do with the completeness and robustness of the method. We have paid attention to both problems but we have concentrated our effort in the latter because the topic of grid quality may be addressed as a post processing step in the refinement process.

The first step taken to ensure that very distorted elements, are not formed, is to consider the face intersection in a broad sense. This is done to prevent robustness problems originated by the presence in the grid of this type of elements. Figure 5 depicts two configurations that can give rise, in subsequent steps, to non desirable cells if the parameters ε and γ fall below a certain tolerance. It can be seen that we do not just pay attention to the nodes of a certain vicinity of the point that is inserted, but also to its proximity to the faces of the front (left sketch) and the possible parallelism between the new and already existing faces (right sketch). In this fashion a number of potential problems due to round off errors or very elongated cavities is avoided. Note that the tethaedra that are about to be formed are not bad in themselves (they may possess a good aspect ratio) but in the context that they are created.

The second measure taken to increase the reliability of the method consists in performing a cleaning processes in the cavities that are causing problems. Most of the times these troubles are originated by an incorrect ordering in the generation of the front and in this way non-tetralizeable cavities are usually removed without recurring to point insertion. Roughly speaking any time difficulties are encountered a cleaning process of the cavity is carried out hoping not to find the same difficulty in the new polyhedron. To avoid oscillations in the process of removing and adding new cells, once a new front is created by removing existing cells the new tethaedra that are formed from this front are flagged to avoid being removed again.

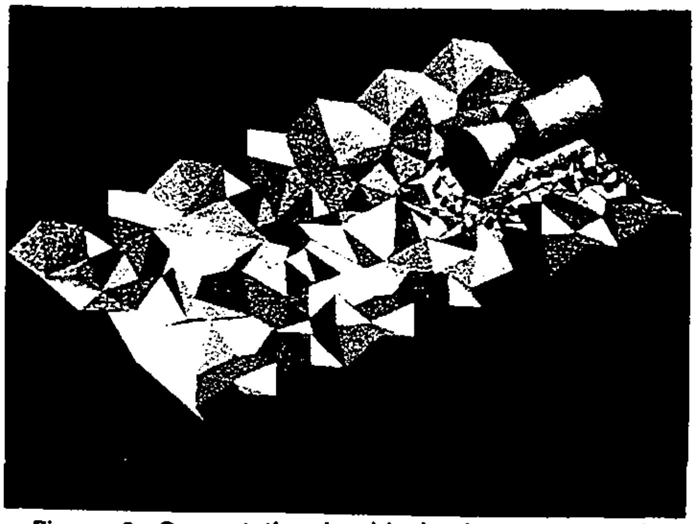

Figure 6: Computational grid about a convergent divergent nozzle with the external field. Fairing flaps are not shown but are included in the computation.

Finally, if every thing fails, a point is allowed to be inserted to let the whole process go on.

EXAMPLES

We have concentrated in gridding volumes confined by a convergent-divergent nozzle and its corresponding external field. The surface grid is formed of quasi-equilateral triangles and has been generated by means of a program developed ad hoc for this type of configuration. The coarse surface mesh contains 1100 faces and the corresponding volume grid about 6000 tethaedra that are inserted at a ratio of 100 tethaedra/s in a IBM 39H. Then a sequence of smoothing steps and swaps is performed till the mesh reaches a stationary state. Should be noted that in this case the quality of the mesh provided by the AF is not as good as the one that can be seen in other works. This is due to the fact that a significant part of the grid is involved in the merging of fronts and therefore the number of high quality elements is low. The grid is then h-refined and the postprocessing step is repeated again till convergence. Both meshes and their statistics can be seen in Figs. 7-9.

SUMMARY

The main advantages and limitations of two competing grid generation approaches have been revisited. Emphasis has been given to the possibility of producing adequate meshes with a small investment in development. The conclusion is that the goal can be fulfilled as long as the surface grids are produced by successive refinement from a background mesh. Efficiency improvement is possible at the expense of further investment in the development of the Advancing Front part of the algorithm.

ACKNOWLEDGMENTS

The authors wish to thank ITP for the permission to publish this paper and for its support to this work. The coding of the 2D Delaunay algorithm is due to Raquel Gómez and her contribution is greatly appreciated. We want also to thank Pedro de la Calzada for his help in developing some of the graphics routines used in this paper.

REFERENCES

Babuska, I., and Aziz, A.K., 1976 "On the Angle Condition in the Finite Element Method" SIAM Journal of Numerical Analysis, Vol. 13, no. 2.

Baker, T.J., 1990, "Unstructured Mesh Generation by a Generalized Delaunay Algorithm" in *Applications of Mesh Generation to Complex 3-D Configurations* AGARD CP-464.

Barth, T.J., 1992, "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations", in *Unstructured Grid Methods for Advection Dominated Flows*, AGARD R-787.

Bonet J, and Peraire, J.,1990 "An Alternating Digital Tree (ADT) Algorithm for Geometric Searching and Intersection Problems", Int. J. Num. Meth. Engng.

Bowyer, A., 1981, "Computing Dirichlet Tesselations", The Computer Journal, Vol. 24, No. 2, pp 162-166.

Dawes, W.N., 1996, "The Generation of 3D, Stretched, Viscous Unstructured Meshes for Arbitrary Domains", ASME 96-GT-55.

Formaggia, L., Peraire, J., Morgan, K. and Peiró, J. 1988, "Implementation of a 3D Explicit Euler Solver on a Cray Computer", in *Proc. 4th Int. Symposium on Science and Engineering on CRAY Supercomputers*, Minneapolis, pp. 45-65.

George, P.I., Hetch, F., and Saltel,E. , 1988, "Constraint of Boundary and Automatic Grid Generation", in *Proc. 2nd Numerical Grid Conference*, Miami

Green, P.J., and Sibson, R., 1977, "Computing the Dirichlet Tesselation in the Plane", The Computer Journal Vol 21, no. 2, pp 168-173.

Jennions,I.K., 1994, "The Role of CFD in the Design Process", in Turbomachinery Design using CFD, AGARD LS-195, pp 8-1,8-34.

Lawson, C.L., 1977, "Software for C^1 Surface Interpolation", Mathematical software III, (Ed. John R. Rice) Academic Press, New York.

Marcum, D.L. and Weatherill, N.P., 1995, "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection", AIAA Journal Vol. 33, no. 9, pp. 1619-1625.

Mavriplis, D.J., 1995, "An Advancing Front Delaunay Triangulation Algorithm Design for Robustness", Journal of Computational Physics, no. 117, pp 90-101.

Muller,J.D., Roe, P.L., and.Deconnick, H., 1992 ,"A Frontal Approach for node Generation in Delaunay Triangulations", in *Unstructured Grid Methods for Advection Dominated Flows*, AGARD R-787, pp 9-1,9-7.

Pirzadeh, S., 1994, "Unstructured Viscous Grid Generation by the Advancing-Layers Method", AIAA Journal, Vol. 32, no 8, pp. 1735-1737.

Peraire, J,Vahdati,M., Morgan, K. and Zienkiewicz, O.C. 1987 "Adaptative Remeshing for Compressible Flow Computations", J. Comp. Phys. Vol. 72, pp 449-466.

Sloan, S.W., 1993, "A Fast Algorithm for Generating Constrained Delaunay Triangulations", Computers and Structures, Vol. 47, No. 3, pp 441-450.

Watson, D.F., 1981, "Computing the n-dimensional Delaunay Triangulation with Application to Voronoi Polytopes" The Computer Journal, Vol. 24, No. 2, pp 167-171.

Weatherill, N.P., Hassan,O., and Marcum, D.L., 1993, "Calculation of Steady Compressible Flowfields with the finite element method", AIAA paper 93-0341.

APPENDIX

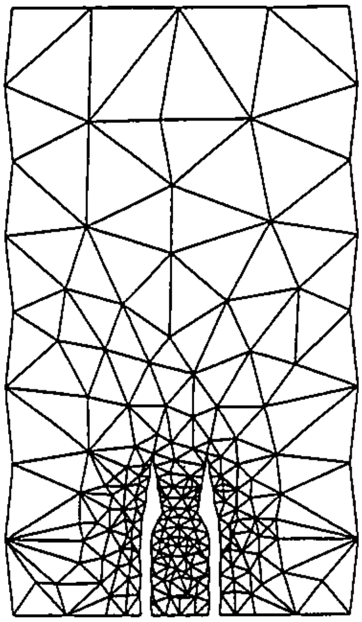

Figure 7: Initial grid before h-refinement

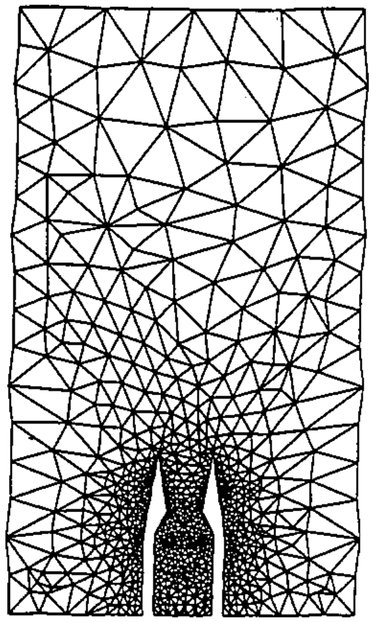

Figure 8: Nozzle after h-refinement and post processing

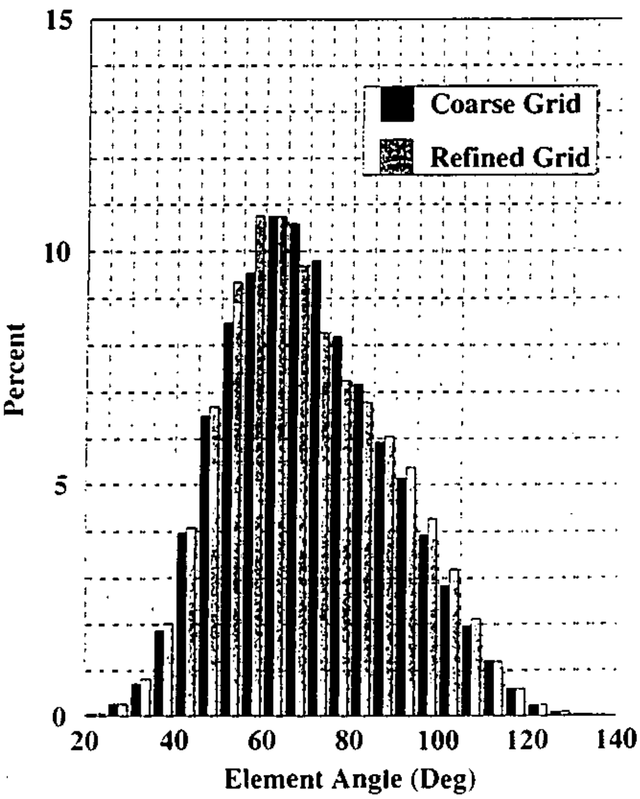

Figure 9: Element angle distribution of the initial and h-refined and postprocessed grids. Refined grid: Max ang.:148°, Min. ang:13°, σ =19°