AIAA JOURNAL
Vol. 33, No. 7, July 1995

Compressible Flowfield Solutions with Unstructured Grids Generated by Delaunay Triangulation

N. P. Weatherill*
University of Wales Swansea, Swansea SA2 8PP, Wales, United Kingdom
O. Hassan†
Computational Dynamics Research, Swansea SA2 8PP, Wales, United Kingdom
and
D. L. Marcum‡
Mississippi State University, Mississippi State, Mississippi 35762

A method is described that constructs three-dimensional unstructured tetrahedral meshes using the Delaunay triangulation criterion. The approach includes an automatic point creation technique and ensures that, given an initial surface triangulation which bounds a domain, a valid boundary conforming assembly of tetrahedra is produced. The efficiency of the proposed procedure reduces the computer time for the generation of realistic unstructured tetrahedral grids to the order of minutes on workstations of modest computational capabilities. The grids generated are used with two finite element algorithms to simulate inviscid transonic compressible flows. Flow solutions for aerospace configurations are presented and computed results compared with experimental data.

I. Introduction

UNSTRUCTURED grid methods with finite element or finite volume flow solvers have proved particularly successful in applications in computational aerodynamics. The grid generation method based on the advancing front concept has been used widely for engineering applications.[1–3] In this approach, given an initial surface of triangles, points are created within the field and tetrahedra formed which grow, or advance, into the domain. An alternative method for tetrahedral generation is to utilize the geometrical construction which is attributable to Delaunay.[4] In this technique, points can be connected using a geometrical criterion to form a topologically valid nonintersecting set of tetrahedra.[5–11] Although the Delaunay geometrical construction provides a well-defined method with which to connect points, it does not provide a method of generating points within a domain. Furthermore, the triangulation may not be boundary conforming.

In this paper a method will be described which, given a set of points and connectivity information for the boundary points, performs a triangulation of the points, automatically creates points in the interior of the domain, and ensures that the bounding surface of the domain is contained in the triangulation. One major feature of the proposed method is the computational speed with which three-dimensional unstructured grids can be generated. It will be shown that it is possible to generate the order of 1×10^6 tetrahedral elements in just minutes on workstations of modest computational capabilities. The method will be demonstrated for realistic shapes, including a complex aircraft configuration.

Flow algorithms successfully used with the proposed grid method will be highlighted. These solvers are based on a second-order accurate space discretization of the Euler equations obtained from a Galerkin weighted residual approximation. Time discretization is obtained as either a two-step Lax–Wendroff scheme or a multistep Runge–Kutta scheme. A new formulation for artificial dissipation is discussed.

Received Feb. 22, 1994; revision received Dec. 6, 1994; accepted for publication Dec. 16, 1994.

*Reader, Department of Civil Engineering, Singleton Park. Member AIAA.
†Research Engineer, Innovation Centre.
‡Associate Professor, Mechanical Engineering and National Science Foundation Engineering Research Center for Computational Field Simulation. Member AIAA.

II. Grid Generation

Three problems must be overcome in the development of a Delaunay triangulation grid generator. Firstly, there is the problem of how to connect the points, secondly, how to generate interior grid points, and thirdly, how to ensure that the resulting triangulation is boundary conforming. Before discussing each of these problems the global procedure I for the generation of grids will be outlined.

1) Input boundary points $\{P_i\}$, $i = 1, I$ and boundary point connectivities of the faces $\{C_j\}$, $j = 1, N$.

2) Derive boundary edges $\{E_k\}$, $k = 1, M$ from boundary face connectivities $\{C_j\}$.

3) Perform the Delaunay triangulation of $\{P_i\}$ to obtain tetrahedra $\{\tau_m\}$, $m = 1, R$.

4) Create interior field points and connect using the Delaunay algorithm to form tetrahedra $\{\tau_l\}$, $l = 1, T$.

5) Ensure that the surface triangulation $\{C_j\}$, $j = 1, N$ is contained in the volume triangulation. Recover any missing faces by following the steps a) recover boundary edges $\{E_k\}$ in $\{\tau_l\}$ and b) recover boundary faces $\{C_j\}$ in $\{\tau_l\}$.

6) Identify all tetrahedra outside the domain of interest $\{\tau o_h\}$, $h = 1, L$.

7) Delete tetrahedra $\{\tau o_h\}$, $h = 1, L$ to give the final grid $\{\tau_l, l = 1, S\}$.

Following the grid construction, postprocessing can be applied to smooth the spatial position of the grid points, check element topology consistencies, and derive grid quality statistics prior to use with an analysis module. In the following sections, details will be given of the steps outlined in the general procedure.

A. Delaunay Triangulation

The mechanism by which an arbitrary set of points can be connected was first proposed by Dirichlet[12] in 1850. For a given set of points in three-dimensional space, $\{P_k\}$, $k = 1, \ldots, K$, the regions $\{V_k\}$, $k = 1, \ldots, K$, are the territories that can be assigned to each point P_k, such that V_k represents the space closer to P_k than to any other point in the set. Clearly, these regions satisfy

$$V_k = \{P_i : |p - P_i| < |p - P_j|, \qquad \forall j \neq i \qquad (1)$$

This geometrical construction of tiles is known as the Dirichlet tessellation or Voronoi[13] diagram. This tessellation of a closed domain results in a set of nonoverlapping convex polyhedra, called Voronoi regions, covering the entire domain. If all point pairs that have some segment of a Voronoi boundary in common are joined,

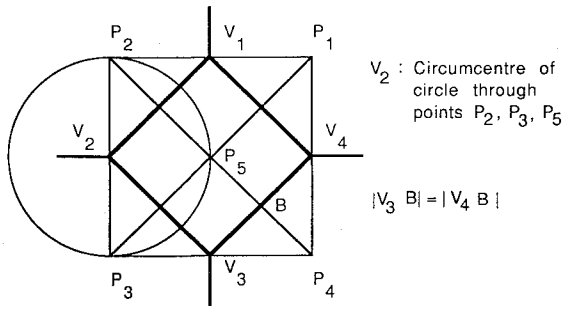


Fig. 1 Delaunay and Voronoi diagram showing some geometrical properties; points and Voronoi vertices P_i, $i = 1, 5$ and V_j, $j = 1, 4$, respectively.

the result is a triangulation of the convex hull of the set of points $\{P_k\}$. This triangulation is known as the Delaunay[4] triangulation. The definition is valid for n-dimensional space.

From the discussion, it is apparent that in two dimensions a line segment of the Voronoi diagram is equidistant from the two points it separates. Hence, the vertices of the Voronoi diagram must be equidistant from each of the three nodes that form the Delaunay triangles. Clearly, it is possible to construct a circle, centered at a Voronoi vertex, which passes through the three points which form a triangle. Furthermore, it is evident that, given the definition of Voronoi line segments and regions, no circle can contain any point. This latter condition is referred to as the in-circle criterion. Figure 1 shows some of the geometrical properties of the construction. In three dimensions a vertex of a Voronoi diagram is at the circumcenter of a sphere that passes through four points which form a tetrahedron, and no other point in the construction can lie within the sphere.

Several algorithms have been proposed for the construction of the Delaunay triangulation.[14,15] The algorithm used here follows the same steps as for the two-dimensional construction[5,7] and is based on the work of Bowyer.[14] This algorithm is based on the in-circle criterion and is a sequential process; each point is introduced into an existing Delaunay satisfying structure, which is broken and then reconnected to form a new Delaunay triangulation. The algorithm is relatively easy to implement. The major issue is to use an efficient data structure. In our work the data required are the element connectivity matrix and the data structure that relates each element to its four neighbor elements. Further details of the method adopted can be found in other papers.[16,17]

B. Automatic Point Creation

Points for connection by the Delaunay algorithm can be derived in many ways. Two ways that have been used include superposition methods and points generated from an independent technique (e.g., structured grid methods[18]). The former approach gives rise to good quality grids in the interior of regions, but grid quality can deteriorate where the tetrahedra and the connections are constrained by the boundaries. The latter approach is restrictive for general geometries. Here, a new method will be presented that is flexible, easy and efficient to implement, requires minimal manual user input, and readily extends for use in grid adaptation with sources and a background mesh.

Point Creation Driven by the Boundary Point Distribution

For grid generation purposes the boundary of the domain is defined by points and associated connectivities. It will be assumed that the grid points on the surface reflect appropriate variations in surface slope and curvature. Ideally any method that automatically creates points should ensure that the boundary point distribution is extended into the domain in a spatially smooth manner.

To explain the motivation behind the approach adopted, consider, in two dimensions, nonintersecting boundary line segments on which points have been distributed that enclose a domain. It is required to distribute points within the region so as to construct a smooth distribution of points. For each point on the boundary, a typical length scale for the point can be computed as the average of the two lengths of the connected edges. No points should be placed within a distance comparable to the defined length scale since this would inevitably define a badly formed triangle. Hence, for each point i, it is appropriate to define a region Γ_i within which no interior point should be placed. In the Delaunay triangulation algorithm, the surface or boundary points are connected together to form an initial triangulation. Points can be placed anywhere within the interior but not inside any of the regions Γ_i already identified. Hence, points are placed at the centroid of each of the formed triangles and then a test is performed to determine if any of the points lie within any Γ_i. If a point lies within the region Γ_i it must be rejected, if it does not then it can be included and connected using the Delaunay triangulation algorithm. Once a point has been inserted, it too must have associated with it a length scale which defines an effective region Γ_i for point exclusion. A newly inserted point takes a length scale from interpolation of the length scales from the nodes that formed the triangle from which it was created. In this way a smooth transition between boundaries of interior points can be ensured. This process of point insertion continues until no point can be added because the union of all Γ_i covers the entire interior domain.

The details of the implementation of procedure II in three dimensions are as follows.

1) Compute the point distribution function for each boundary point $r_o = (x, y, z)$ i.e., for point o

$$dp_o = \frac{1}{M} \sum_{i=1}^{M} |r_i - r_o|$$

where $|\ |$ is the Euclidean distance and it is assumed that point o is surrounded by M points, $i = 1, M$.
2) Generate the Delaunay triangulation of the boundary points.
3) Initialize the number of interior field points created, $N = 0$.
4) For all tetrahedra within the domain, complete the following steps.
   a) Define a prospective point Q to be at the centroid of the tetrahedron.
   b) Derive the point distribution dp_Q for the point Q by interpolating the point distribution function from the nodes of the tetrahedron, dp_m, $m = 1, \ldots, 4$.
   c) Compute the distances d_m, $m = 1, \ldots, 4$, from the prospective point Q to each of the four points of the tetrahedron.

      If $\{dp_m < \alpha dp_Q\}$ for any $m = 1, \ldots, 4$ then reject the point; return to the beginning of step 4.

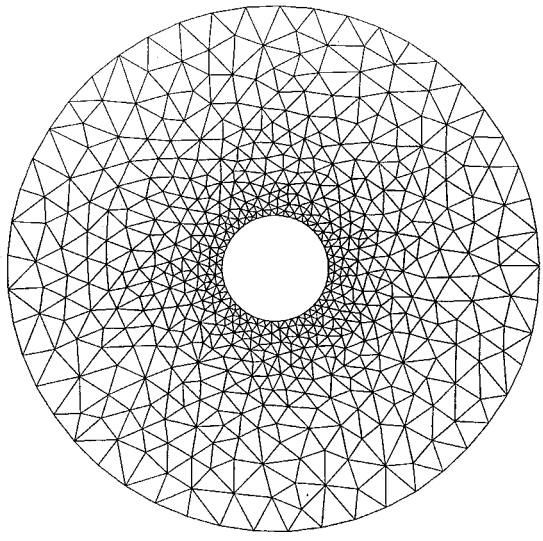
      If $\{dp_m > \alpha dp_Q\}$ for all $m = 1, \ldots, 4$ then compute the distance s_j, $(j = 1, \ldots, N)$, from the prospective point Q, to other points to be inserted, P_j, $j = 1, N$.

         If $\{s_j < \beta dp_Q\}$ then reject the point; return to the beginning of step 4.

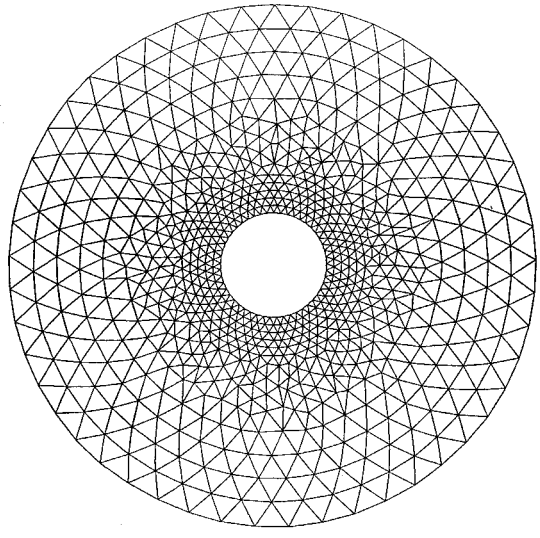
         If $\{s_j > \beta dp_Q\}$ then accept the point Q for insertion by the Delaunay triangulation algorithm and include Q in the list P_j, $j = 1, \ldots, N$.

   d) Assign the interpolated value of the point distribution function, dp_Q, to the new node P_N.
   e) Next tetrahedra.
5) If $N = 0$ go to step 7.
6) Perform the Delaunay triangulation of the derived points, P_j, $j = 1, \ldots, N$. Go to step 3.
7) Smooth the mesh.

The coefficient α controls the grid point density by changing the allowable shape of formed triangles, whereas β has an influence on the regularity of the triangulation by not allowing points within a specified distance of each other to be inserted in the same sweep of the triangles within the field. The effects of the parameters α and β are demonstrated in Fig. 2.



a)



b)

Fig. 2 Two grids generated with the combination of parameters α and β: a) α = 1.0 and β = 10.0, and b) α = 1.0 and β = 0.1.

The algorithm presented has some aspects in common with the approach used to generate unstructured grids using the advancing front technique.[1-3] In that approach a background grid is defined by the user which covers the domain to be gridded. At each node of this grid the required point spacing is specified. During the generation process the points are created from a linear interpolation of the background grid point spacing. In the approach advocated here, it is assumed that the boundary points have been suitably distributed. A length scale is then computed for each boundary point as the average of the lengths of the edges connected to that point. The Delaunay triangulation is then used to connect all boundary points to form, in effect, a background mesh in the sense used in the advancing front method. In this approach, however, the user does not have to define the rather intricate background mesh. Further work in combining concepts from the advancing front method and point insertion strategies has recently been reported.[19]

The approach has been found to work well for geometries in two and three dimensions. Examples will be given in the results section.

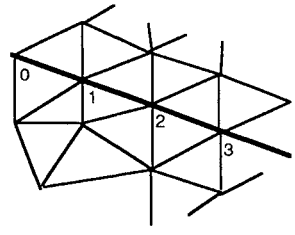
It should be noted that this point creation algorithm can be implemented very efficiently within the Delaunay triangulation procedure. In particular, if a point is accepted for insertion then in the Delaunay algorithm a tetrahedron is known that contains this point, since by the very nature of the procedure the tetrahedron from which the point was created is known. However, after the insertion of one point the tetrahedron numbering can be changed and if the tetrahedra formed from the inserted points overlap then the tetrahedron numbers which have been flagged for each new point can be then incorrect. However, the exclusion zone, controlled by the parameter β, ensures that the points created from one sweep through the tetrahedra are sufficiently spatially separated that on the insertion of each point the resulting tetrahedra do not overlap and, hence, the original tetrahedron numbers associated with each new point are

Table 1 Grid statistics before and after grid quality enhancement

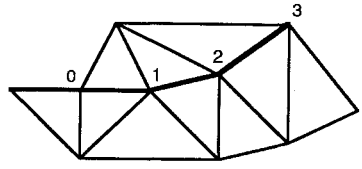
Volume distributions				
Interval no.	Volume interval		No. of elements improved grid	No. of elements initial grid
1	0.3662×10^{-3}	0.6277×10^{-3}	0	1
2	0.6277×10^{-3}	0.1076×10^{-3}	0	2
3	0.1076×10^{-2}	0.1844×10^{-2}	0	2
4	0.1844×10^{-2}	0.3160×10^{-2}	0	4
5	0.3160×10^{-2}	0.5415×10^{-2}	0	1
6	0.5415×10^{-2}	0.9280×10^{-2}	0	1
7	0.9280×10^{-1}	0.1591×10^{-1}	0	0
8	0.1591×10^{-1}	0.2726×10^{-1}	1	0
9	0.2726×10^{-1}	0.4672×10^{-1}	2	1
10	0.4672×10^{-1}	0.8006×10^{-1}	16	4
11	0.8006×10^{-1}	0.1372×10^{0}	48	28
12	0.1372×10^{0}	0.2352×10^{0}	199	179
13	0.2352×10^{0}	0.4030×10^{0}	221	208
14	0.4030×10^{0}	0.6907×10^{0}	203	197
15	0.6907×10^{0}	0.1184×10^{1}	216	223
16	0.1184×10^{1}	0.2029×10^{1}	123	132
17	0.2029×10^{1}	0.3477×10^{1}	64	63
18	0.3477×10^{1}	0.5959×10^{1}	28	28
19	0.5959×10^{1}	0.1021×10^{2}	12	12
20	0.1021×10^{2}	0.1000×10^{2}	1	1
Maximum value of the element distortion parameter α:			1.24	185.91

Table 2 Computational times for the generation of a grid with 1×10^6 elements: times include I/O, consistency checks, and postprocessing for the flow solvers

Computer	CPU, min
Cray Y-MP	4.96
IBM Risc 6000 550	12.28
SGI PI	28.84
SGI Indigo Elan	25.28
SGI Indigo XS 24/4000	14.75
SGI Crimson	14.36
SUN Sparc II	26.98



a)



b)

Fig. 3 Edge data structure for the artificial dissipations: a) smooth, regular, unstructured grid showing edges aligned unidirectionally and b) unstructured grid showing edges not aligned unidirectionally.

valid. Hence, in this way β improves the regularity of the tetrahedra and also ensures that no search is required to find a sphere which includes the point.

The procedure outlined creates points consistent with the point distribution on the boundaries. However, in many problems, information is known about features within the domain that require a suitable special discretization. It proves possible to modify procedure II to take such effects into account. Two techniques can be readily implemented. The first utilizes the idea of point, line, and planar sources, whereas the second uses the concept of a background mesh.[20,21] Details of the implementation of these approaches have been given elsewhere.[16,22]

C. Surface Boundary Integrity

The generation of boundary conforming grids using the Delaunay triangulation has long been recognized as a major problem, and several approaches to its solution have been proposed.[23,24] After the connection of boundary and interior points there is no guarantee that the resulting assembly of tetrahedra will recover the initial boundary surface triangulation. The procedure followed here to ensure boundary integrity is an extension of earlier work[10] and is closely related to the work of George and Hermeline.[24] Both these approaches involve the addition of points to "block" the penetration of tetrahedra through the boundary surface. Here, an approach has been devised which involves a local modification to the tetrahedral construction so as to recover a given set of triangular bounding faces. A finite number of direct transformations can be formulated for all types of tetrahedral penetration of a surface and, hence, in the proposed procedure the recovery of the surface can be guaranteed.[16]

D. Element Grid Quality

Badly formed elements in time-dependent flow analysis using explicit flow solvers can have a dramatic effect in limiting the allowable time step. Hence, some effort has been expended in improving grid quality, in particular, removing worst case elements from the grid.

A relatively simple, but effective way of improving grid quality is to smooth the spatial position of grid points using a Laplacian operator. The final grid can be smoothed N times according to the operation

$$r_o^{n+1} = r_o^n + \frac{\omega}{M}\sum_{i=1}^{M}(r_o - r_i) \tag{2}$$

where r_o^{n+1} is the new position of the point r_o after $n+1$ iterations, M is the number of neighboring points with coordinates r_i and ω is

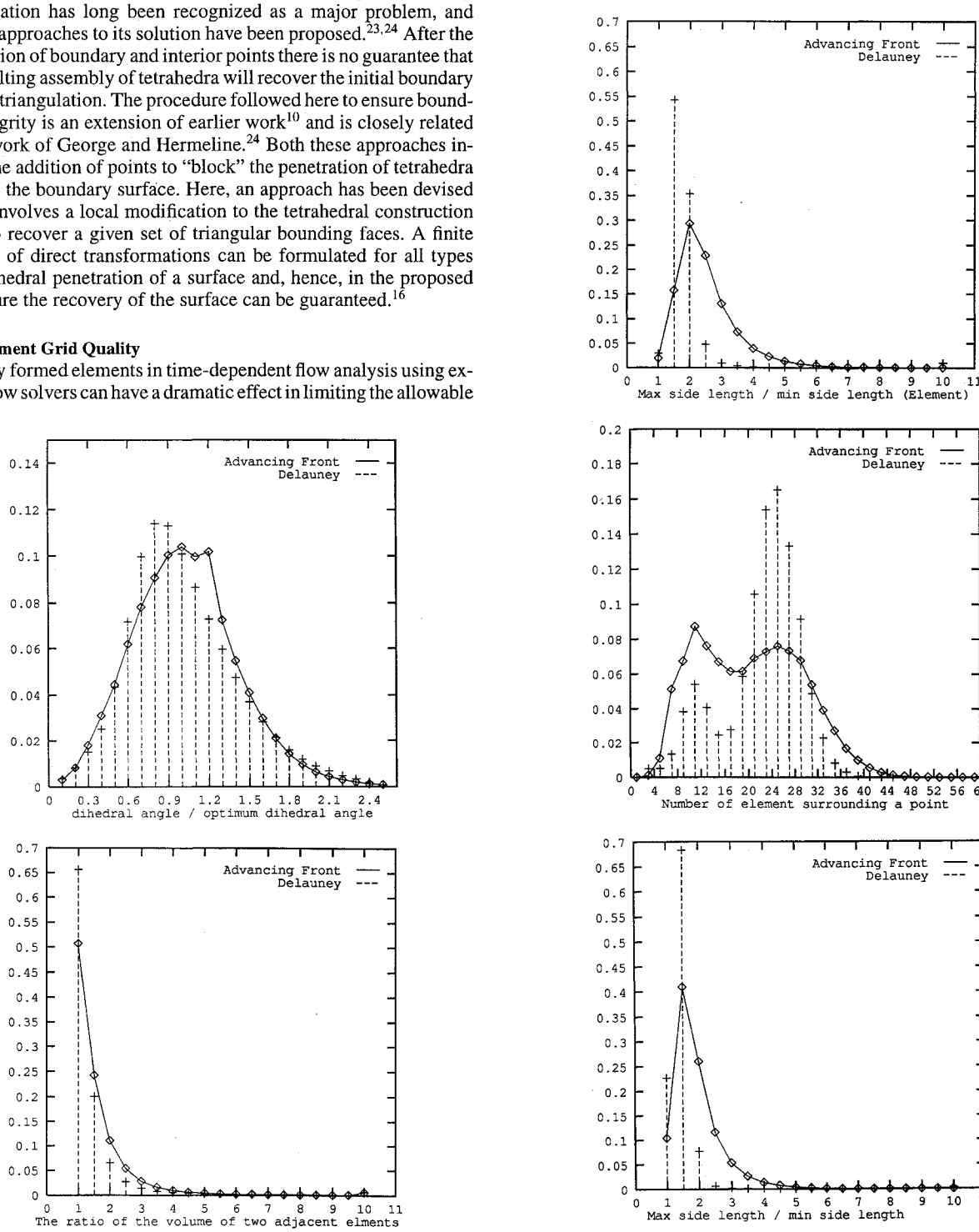
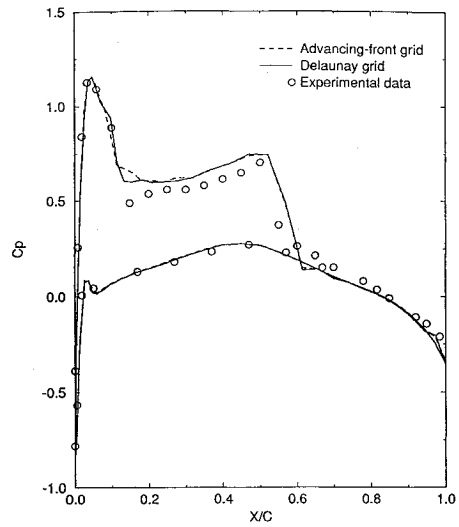
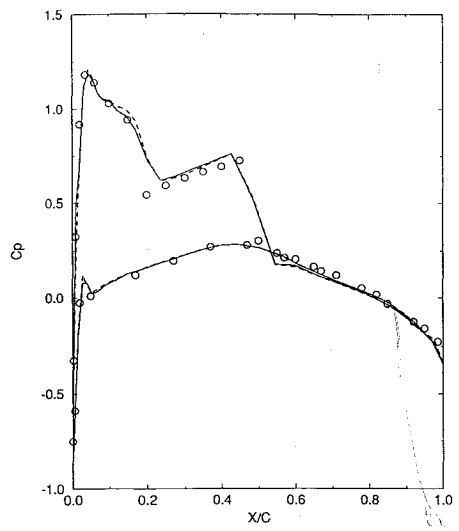


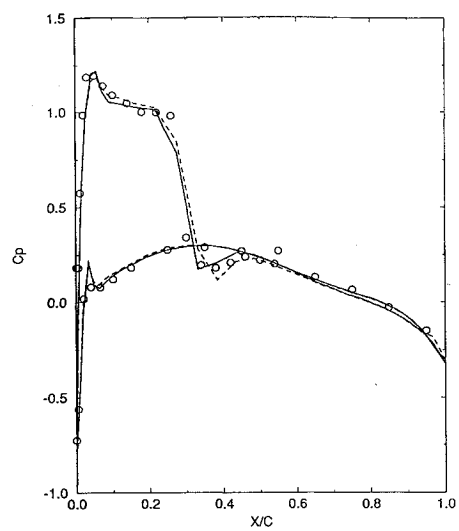
Fig. 4a Grid and flow solutions for the ONERA M6 wing, freestream Mach number = 0.84, incidence = 3.06 deg: grid quality statistics for the Delaunay and advancing front grids.



a) 44% semispan



b) 65% semispan



c) 90% semispan

Fig. 4b Grid and flow solutions for the ONERA M6 wing, freestream Mach number = 0.84, incidence = 3.06 deg: computed coefficients on the wing surface.

the relaxation parameter. The number of iterations should be within the range 20–100 and the relaxation parameter 0.05–0.25.

In two dimensions there are several operations which can be performed to enhance grid quality. In particular, interior field nodes which are connected to just three points can be removed so that the three triangles connected to the node are removed leaving the one triangle. It is also possible to consider local diagonal swapping to improve the regularity of the triangles.[19] In three dimensions, it is possible, although more difficult, to enhance grid quality through the implementation of such operations. The required tetrahedral transformations to remove particular elements or element connectivities are presently an area of study[19] and the improvement of tetrahedral grid quality is likely to be a major issue in the future, particularly as grid generation techniques mature. The measure α,

$$\alpha = \frac{(\text{average element edge length})^3}{\text{volume}}$$

is very successful in identifying flat or near flat elements since in such cases the volume degenerates and tends to zero. In fact, in a flat element with four points all planar the volume is zero and the distortion measure α is infinite. For an equilateral element the measure takes the value of $\alpha_{\text{equilateral}} = 8.479$.

A typical example of the improvement in grid quality which can be achieved using grid coordinate smoothing and element transformations is given in Table 1. It is apparent that the badly formed elements with small volumes and high values of the citerion α can be removed from the mesh.

III. Flow Algorithms

A. Governing Equations

The Euler equations for time-dependent, three-dimensional, compressible, inviscid, nonconducting flow of a simple system in thermodynamic equilibrium in the absence of body forces can be expressed in conservation form:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}^i}{\partial x^i} = 0 \qquad i = 1, 2, 3 \tag{3}$$

where $\boldsymbol{U}$ is the solution vector and $\boldsymbol{F}^i$ is the advection flux vector in the i direction. The solution and advective flux vectors are given by

$$\boldsymbol{U} = [\rho, \rho V^1, \rho V^2, \rho V^3, \rho E]^T \tag{4}$$

and

$$\boldsymbol{F}^i = \begin{bmatrix} \rho V^i \\ \rho V^1 V^i + \delta^{1i} p \\ \rho V^2 V^i + \delta^{2i} p \\ \rho V^3 V^i + \delta^{3i} p \\ (\rho E + p) V^i \end{bmatrix} \tag{5}$$

respectively, where ρ is the density, V^i is the velocity component in the i direction, E is the total energy, and p is the static pressure. For the present investigation, a thermally and calorically perfect gas is assumed and the equation of state for pressure is given by

$$p = (\gamma - 1)\left[\rho E - \tfrac{1}{2}\rho\left(V^{1^2} + V^{2^2} + V^{3^2}\right)\right] \tag{6}$$

where γ is the specific heat ratio, which is taken for this work to be 1.4.

The governing equations, Eq. (3), are solved using either an explicit multistage Runge–Kutta scheme[6,25,26] or an explicit two-step Lax–Wendroff scheme.[1,2,25,26] In both implementations, characteristic boundary conditions are applied.[27] A new key feature of our

implementation of these procedures involves the formulation of the artificial diffusion.

B. Artificial Diffusion

In regions containing severe gradients, e.g., near shock waves and stagnation points, artificial diffusion is required to filter out oscillations. Also, background dissipation is required with the Runge–Kutta solver to prevent decoupling. In the present work, a blended, adaptive first-/third-order dissipation model is used. The dissipation is given by

$$D_{12} = \sum_{\text{edge}} [\max(c_1, c_2, G)\, S(U_2 - U_1) - c_3\, S(U_3 - 3U_2 + 3U_1 - U_0)] \tag{7}$$

where D_{12} is the dissipation residual term which is added to the right-hand side of Eq. (3) for point 1 and subtracted for point 2; U_0, U_1, U_2, and U_3 are the solution vectors at 0, 1, 2, and 3; c_1 is the first-order background constant, c_2 is the gradient switch constant, c_3 is the third-order background constant, G is the gradient switch, and S is the directional scaling factor. The gradient switch is given by

$$G = \max\left\{ \left| \frac{p_0 + p_2 - 2p_1}{p_0 + p_2 + 2p_1} \right|, \left| \frac{p_1 + p_3 - 2p_2}{p_1 + p_3 + 2p_2} \right| \right\} \tag{8}$$

and where p is typically the pressure. The gradient switch is also normalized so that it varies between 0 and 1. At high Mach numbers, this method fails and another scheme is used. A flux-limited dissipation model developed for structured solvers by Jameson[28] and enhanced by Yoon and Kwak[29] can be used successfully at high Mach numbers. The directional scaling factor is obtained using edge variables and is given by

$$S = V_i \cdot \Gamma^i + a|\Gamma^i| \tag{9}$$

where a is the speed of sound and Γ^i is edge area vector. For the present work, an edge-based data structure that includes a local edge connectivity is used.

Most finite element and finite volume flow solvers can be implemented using either an edge-, element- or face-based data structure. The edge-based data structure can offer memory and speed advantages. It can also be used to exploit local structure in the grid. An edge-based data structure that utilizes a local structured edge connectivity has been developed and implemented for the present work. Neighboring edges and their connectivity for this data structure are shown in Fig. 3a for a two-dimensional unstructured grid. For an edge given by points 1 and 2, two closely aligned neighboring edges given by points 0 and 1 and points 2 and 3 are selected. Points 0, 1, 2, and 3 are analogous to locations $i-1, i, i+1$ and $i+2$ in a structured grid. Ideally, the connected edges should form a very smooth line, as shown in Fig. 3a. For a typical grid, however, with varying element volumes, there will be some connected edges which do not form a smooth line, as shown in Fig. 3b. Based on results obtained with such connections, this does not create any numerical problems. The local structured edge connectivity allows direct and efficient implementation of many upwind high-resolution structured schemes. In the present work, a structured flux-limited dissipation model is implemented using this connectivity. The localized structured connectivity allows this scheme, including limiters, to be implemented exactly as it is in structured solvers.

IV. Results

A. Efficiency of Grid Generation

Table 2 shows the computational time required to generate a grid of 1×10^6 tetrahedral elements on several types of computers. It should be emphasized that no form of optimization has been performed on the code for use with any particular machine. Furthermore, the times given are for the generation and input of the surface grid and output of the final grid, consistency checks and postprocessing such that the final grid is output in a form which can be used directly, without further processing, with the Lax–Wendroff and Runge–Kutta flow solvers.

B. Flow Results

The first example of the performance of the grid and flow methods described is chosen to be one of the standard test case problems for three-dimensional aerospace applications, namely, the flow over an ONERA M6 wing.[30] The freestream Mach number is 0.84, and the angle of attack is 3.06 deg. To demonstrate how the results of the new approach compare with other methods, two grids were generated both using the same surface grid; one using the Delaunay approach described and the other using the advancing front grid used for a NASA Langley unstructured grid workshop in 1990.[3] For both grids, the total number of grid points and the field grid point distribution is very similar. The advancing front grid contained 231,507 elements and 42,410 points and the Delaunay grid contained 233,182 elements and 40,442 points. Grid quality statistics of the two generated grids are shown in Fig. 4a. The two grids are comparable in the element and point measures chosen. Using the Lax–Wendroff flow solver on both grids, the flow solutions were obtained. The computed pressure coefficient distributions on the wing surface for the Delaunay and advancing front grids are shown in Fig. 4b. The computed flow results are in good agreement, which, together with the grid quality statistics, provides some justification of the validity and quality of the Delaunay grid approach.

As an example of the application of the methods to a complex aerospace configuration, Fig. 5 shows the simulation of the flow around the B60 configuration with wing/fuselage/pylon and nacelle with powered engines. The freestream Mach number was 0.801, and the angle of attack 2.738 deg. For the simulation, the engine conditions imposed were a jet pressure ratio of 2.477, an engine mass flow ratio of 2.733 lb/s, and a jet total temperature of 370.04 K. Shown in

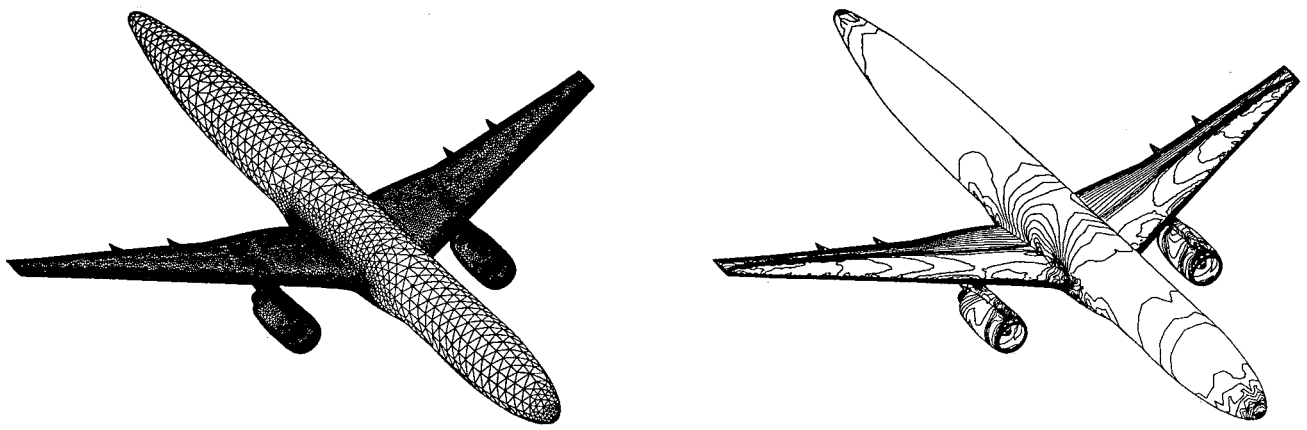


Fig. 5a Grid generation and inviscid flow computation around a wing/body/pylon/engine configuration, freestream Mach number = 0.84, incidence = 3.06 deg: surface grid and surface solution contours of pressure.

Fig. 5a are the grid and pressure contours on the surface. Figure 5b indicates planar cuts through the field grid, and pressure coefficient distributions, compared with experiment, along the wing and at a section through the nacelle, are shown in Fig. 5c. The Runge–Kutta algorithm was used to obtain the results, and the residual, measured as the rms of the change in density per time step, was reduced by 5 orders of magnitude. In general, reasonable agreement is achieved between the computed and experimental data.

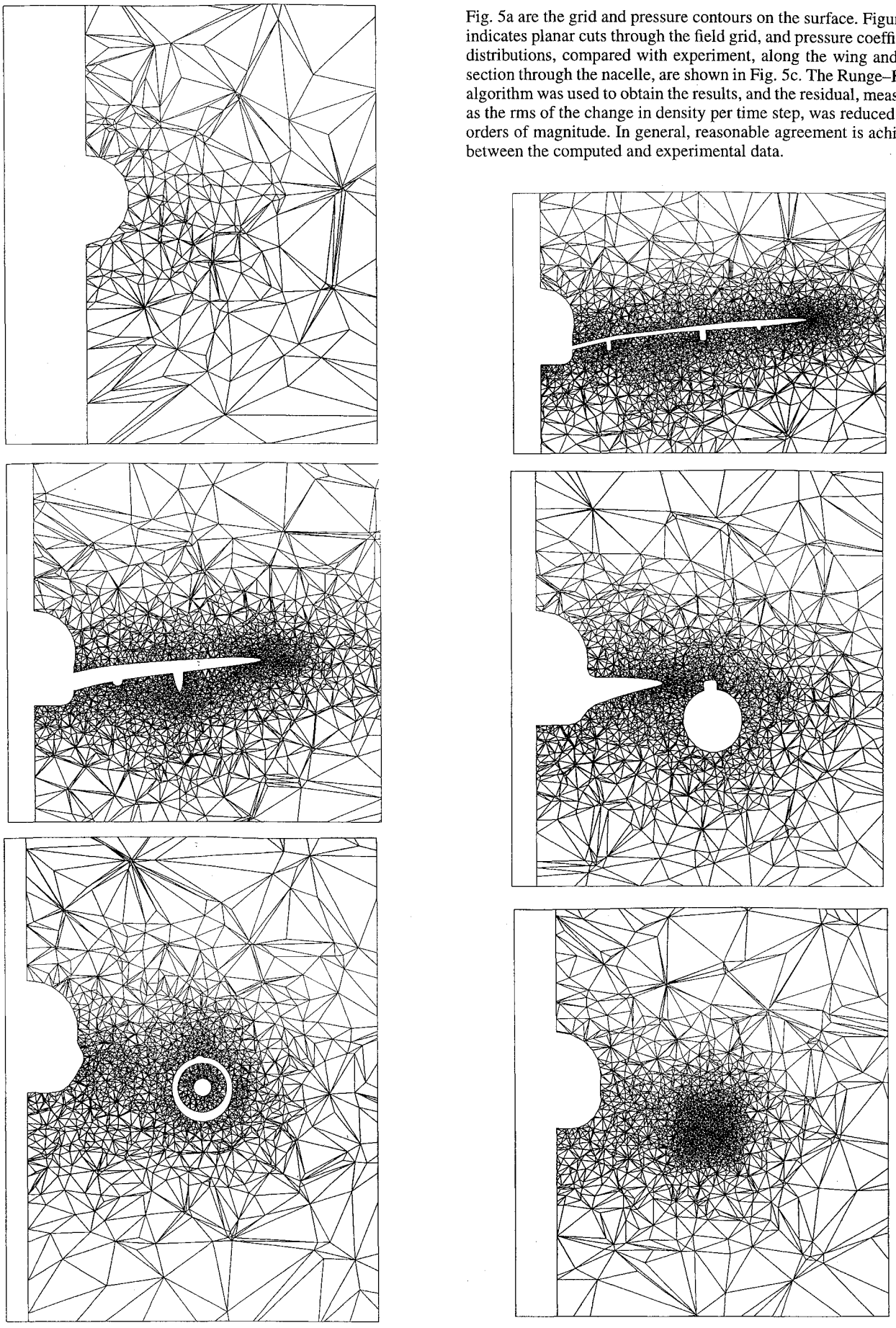


Fig. 5b Grid generation and inviscid flow computation around a wing/body/pylon/engine configuration, freestream Mach number = 0.84, incidence = 3.06 deg: planar cuts through the field grid at sections perpendicular to the plane of symmetry at different stations along the fuselage.

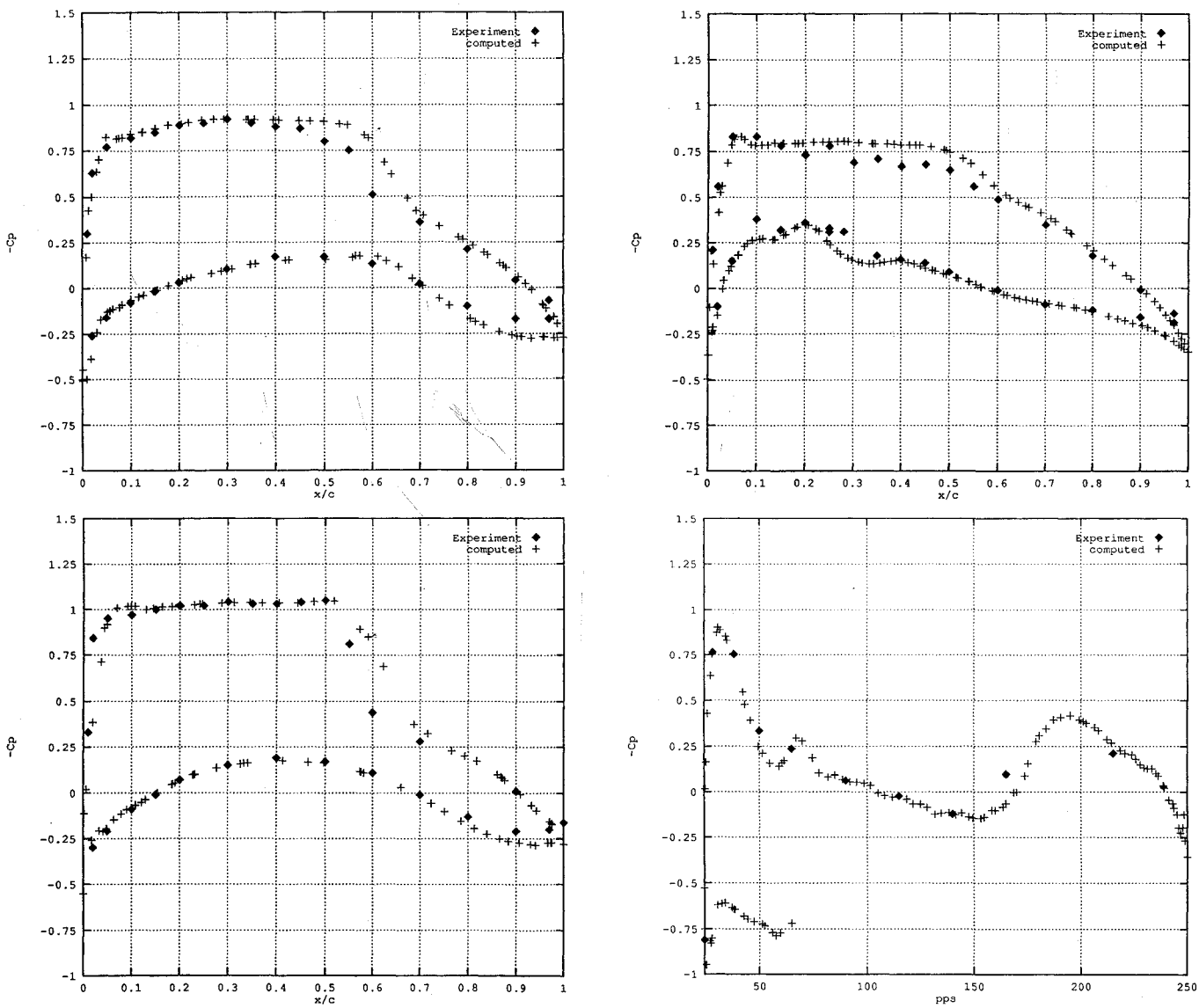


Fig. 5c Grid generation and inviscid flow computation around a wing/body/pylon/engine configuration, freestream Mach number = 0.84, incidence = 3.06 deg: computed pressure coefficients on the wing and nacelle surfaces.

V. Summary

This paper has outlined a method for constructing unstructured grids of tetrahedra. The geometrical criterion on which the method is based is due to Delaunay. Automatic point creation within the Delaunay triangulation which utilizes boundary point spacing with an option for the use of sources and the background mesh has been discussed. Two finite element flow algorithms have been described. These methods have been demonstrated on realistic three-dimensional configurations.

Acknowledgments

Computing resources were provided in part by the Numerical Aerodynamic Simulation program in the U.S. and the Science and Engineering Research Council in the U.K. The authors would like to thank J. Peiro, J. Peraire, and K. Morgan for providing the surface grids shown in the paper. The authors wish to thank Rolls Royce and, in particular, P. Stow, for permission to show the results for the B60 with engines and to the Ministry of Defence/Royal Aerospace Establishment for permission to publish the corresponding experimental data.

References

[1]Peraire, J., Peiro, J., Formaggia, L., Morgan, K., and Zienkiewicz, O. C., "Finite Element Euler Computations in Three Dimensions," *International Journal for Numerical Methods in Engineering*, Vol. 26, 1988, pp. 2135–2159.

[2]Morgan, K., Peraire, J., Peiro, J., and Hassan, O., "The Computation of Three-Dimensional Flows Using Unstructured Grids," *Computational Methods in Applied Mechanics and Engineering*, Vol. 87, 1991, pp. 335–352.

[3]Lohner, R., and Parikh, P., "Three-Dimensional Grid Generation by the Advancing Front Method," *International Journal for Numerical Methods in Fluids*, Vol. 8, 1988, pp. 1135–1149.

[4]Delaunay, B., "Sur la sphere vide," *Bulletin of Academy of the Sciences of the USSR, Class. Sci. Nat.*, 1934, pp. 793–800.

[5]Weatherill, N. P., "The Generation of Unstructured Grids Using Dirichlet Tessellations," Princeton Univ., Mechanical and Aerospace Engineering Rept. 1715, Princeton, NJ, July 1985.

[6]Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft," AIAA Paper 86-0103, Jan. 1986.

[7]Weatherill, N. P., "A Method for Generating Irregular Computational Grids in Multiply Connected Planar Domains," *International Journal for Numerical Methods in Fluids*, Vol. 8, 1988, pp. 181–197.

[8]Baker, T. J., "Three-Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets," *Proceedings of the AIAA 8th Computational Fluid Dynamics Conference*, AIAA, Washington, DC, 1987.

[9]Childs, P. N., and Weatherill, N. P., "Generation of Unstructured Grids Within a Hybrid Multi-Block Environment," *Proceedings of the 3rd International Conference on Numerical Grid Generation* (Barcelona, Spain), edited by A. S. Arcilla, Elsevier, Amsterdam, 1991, pp. 899–911.

[10]Weatherill, N. P., "Delaunay Triangulation in Computational Fluid Dynamics," *Computers and Mathematics with Applications*, Vol. 24, No. 5/6, 1992, pp. 129–150.

[11]Weatherill, N. P., and Hassan, O., "Efficient Three-Dimensional Grid Generation Using the Delaunay Triangulation," *Proceedings of the 1st*

European Computational Fluid Dynamics Conference (Brussels, Belgium), edited by C. Hirsch, J. Periaux, and W. Kordulla, Elsevier, Amsterdam, 1992.

[12]Dirichlet, G. L., "Uber die reduction der positiven quadratischen formen mit drei underestimmten ganzen zahlen," *Journal Reine Angewandte Mathematik*, Vol. 40, No. 3, 1850, pp. 209–227.

[13]Voronoi, G., "Nouvelles applicatons des parametre continus a la theorie des formes quadratiques. Recherches sur les parallelloedres primitifs," *Journal Reine Angewandte Mathematik*, Vol. 134, 1908.

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

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

[16]Weatherill, N. P., and Hassan, O., "Efficient Three-Dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints," *International Journal for Numerical Methods in Engineering*, Vol. 37, 1994, pp. 2005–2039.

[17]Weatherill, N. P., "Grid Generation by the Delaunay Triangulation," *Lecture Notes on Grid Generation*, von Kármán Institute, Brussels, Belgium, Jan. 1994.

[18]Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W., *Numerical Grid Generation: Foundations and Applications*, North-Holland, Amsterdam, 1985.

[19]Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," AIAA Paper 94-1926, June 1994.

[20]Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466.

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

[22]Weatherill, N. P., Hassan, O., Marchant, M. J., and Marcum, D. L., "Adaptive Inviscid Flow Solutions for Aerospace Geometries on Efficiently Generated Unstructured Tetrahedral Meshes," AIAA Paper 93-3393, June 1993.

[23]Baker, T. J., "Shape Reconstruction and Volume Meshing for Complex Solids," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 4, 1991, pp. 665–677.

[24]George, P. L., and Hermeline, F., "Delaunay's Mesh of a Convex Polyhedron in Dimension d: Application for Arbitrary Polyhedra," *International Journal for Numerical Methods in Engineering*, Vol. 33, 1992, pp. 975–995.

[25]Marcum, D. L., and Agarwal, R. K., "A Three-Dimensional Finite Element Navier–Stokes Solver with k-ε Turbulence Model for Unstructured Grids," AIAA Paper 90-1652, June 1990.

[26]Marcum, D. L., "Accuracy of a Finite Element Euler Solver for Unstructured Grids," NASA Workshop: Accuracy of Unstructured Grid Techniques, Hampton, VA, Jan. 1990.

[27]Marcum, D. L., and Hoffman, J. D., "Numerical Boundary Condition Procedures for Euler Solvers," *AIAA Journal*, Vol. 25, No. 8, 1987, pp. 1054–1062.

[28]Jameson, A., "A Nonoscillatory Shock Capturing Scheme Using Flux Limited Dissipation," *Lectures in Applied Mathematics*, Vol. 22, Pt. I, Applied Mathematics Series, 1985, pp. 345–370.

[29]Yoon, S., and Kwak, D., "Artificial Dissipation Models for Hypersonic External Flow," AIAA Paper 88-3708, July 1988.

[30]Schmidt, V., and Charpin, F., "Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers," AGARD-AR-138, Chap. B-1, 1979.