DEVELOPMENTS AND TRENDS IN THREE-DIMENSIONAL MESH GENERATION

Timothy J. Baker

Princeton University
Princeton, New Jersey

Abstract

An intense research effort over the last few years has produced several competing and apparently diverse methods for generating meshes. This paper reviews recent progress and emphasizes the central themes that we can expect to form a solid foundation for future developments in mesh generation.

Introduction†

Although long recognized as a major pacing item [24,49], mesh generation has only recently achieved the long sought aim of making possible the calculation of complete aircraft flow fields. Even now mesh generation for complex configurations is not a routine task and a period of maturation is still required before meshes can be quickly and efficiently created for any type of aircraft shape. We are, however, at an important juncture where techniques for tackling the problem are well understood, if not yet fully developed. It is therefore a suitable time to take stock and consider which approaches are likely to stay the course and provide the basis for reliable and efficient flow calculation methods.

Early methods for calculating transonic flow over airfoils and other simple two dimensional shapes were based on conformal mapping techniques [2,3,20,43,70] and simple shearings [5,77]. At that time mesh generation was regarded as a subsidiary part of the flow algorithm. Indeed the problem was essentially that of finding a suitable coordinate transformation, expressing the flow equations in the curvilinear coordinate system and then devising a solution algorithm for the transformed flow equations. The paper of Thompson et al. [88] was a significant break from this tradition. It advocated the idea of using elliptic equation methods to generate meshes around arbitrary shapes. Although this approach had been tried before [104], Thompson et al. were the first to realize its full potential and to exploit the idea as a general technique. The paper had two other important effects. First, it focused attention on mesh generation as a problem in its own right, largely independent of the flow solver. Second, their work also recognized the need to move from a global description in terms of an overall mapping to a local viewpoint based on a mesh defined by a set of points and an ordering of the points corresponding to the coordinate directions. There is, of course, no need to use global mappings, and finite difference formulae can easily be constructed for numerically generated meshes. What is interesting, however, is that the apparently more elegant approach, of using coordinate transformations to achieve a global mapping of the flow equations, did not have sufficient generality to survive as a method for handling complex shapes. At the other extreme, the use of a non–aligned Cartesian mesh [19,65,103] once held the prospect of treating arbitrary geometries at the cost of finding adequate interpolation procedures for applying the solid wall boundary conditions. Unfortunately this cost has proved a formidable obstacle and the non–aligned mesh approach has won few adherents. The lesson,

† This paper refers to research described in references 1–106.

which is often repeated, is that mesh generation eschews the most elegant solution without abandoning elegance altogether. In this case the preferred approach is a surface conforming mesh and a flow solver that uses only information about the mesh point positions. It is a compromise between the beauty of global coordinate mappings which prove too rigid and the excessive generality of a non–aligned mesh which is too difficult to couple with the flow solver. In this context it is important to note another early development, the introduction of the finite volume approach [45,66] that treats each mesh cell as a control volume to approximate the integral form of the flow equations. This is an inherently more accurate approach for discretizing the flow equations, a natural way of ensuring conservation and is particularly well suited to the general surface conforming mesh. These concepts are now so widely accepted that it is easy to forget that there was a time when the choice was not so obvious.

The paper of Thompson et al. [88] stimulated a great deal of work using elliptic equations to generate meshes in both two and three dimensions. The possibility of basing numerical mesh generation on hyperbolic and parabolic equation sets has also been explored. One unfortunate aspect of this emphasis on numerical mesh generation is that relatively little attention has been paid to alternative procedures. Algebraic methods [6,28,32,74], in particular, offer many attractive features. The work of Eiseman [28] is perhaps the best known example of this approach. He constructs a set of coordinate surfaces between two boundary surfaces and then defines an interpolatory function to provide a smooth variation in the third coordinate direction. The greater the number of intermediate surfaces used, the greater the degree of mesh smoothness that can be obtained.

A particularly powerful algebraic method is based on transfinite interpolation. This approach to mesh generation was first introduced by Eriksson [31,32], who has developed the idea into an extremely effective technique. In common with all algebraic methods, this approach is very fast compared with numerical mesh generation. The most significant feature of transfinite interpolation, however, is the ability to exercise a high degree of control over the mesh point distribution, particularly, over the slope of the mesh lines which meet the boundary surfaces.

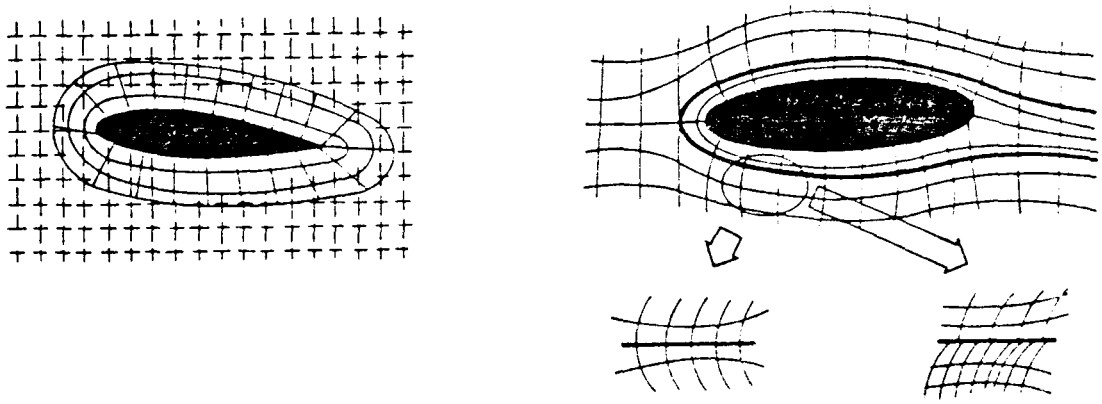


Fig. 1 Multiblock Variants (taken from Kutler, Ref. 49)

The conflicting requirements, of control over mesh quality on the one hand and flexibility to handle arbitrary shapes on the other, have pervaded mesh generation throughout its development. Indeed the essence of mesh generation is the need to maintain as much of both features without seriously degrading either. An important step towards securing a harmonious balance between these two constraints is the use of a multiblock strategy. This concept is to break the flow field into several smaller blocks (essentially an ultra–coarse mesh) and then generate separate meshes in each individual block. This concept was first formulated by Lee et al. [50] and has rapidly gained wide acceptance. There are several variants of this approach (see figure 1), depending on whether there is continuity of mesh lines at the block interfaces (composite), whether the individual blocks are entirely independent and merely overlap one another (overlapped), or block interfaces but no continuity of mesh lines (patched).

At first sight it would seem that a blocked mesh structure is further step away from simplicity and elegance towards messy complexity. However the experience of many researchers who have embraced multiblock in spite of this drawback, provides convincing testimony to its role as an indispensable component of mesh generation for complicated configurations. Again one has been forced to relinquish a global viewpoint (the use of a single mesh) in favor of a local perspective (several blocks each able to accommodate part of the flow field and boundary surface).

The discussion has so far centered on structured meshes composed of quadrilaterals in two dimensions or hexahedra in three dimensions. The mesh is structured in the sense that there is an implied set of coordinate directions within each block. In order to treat complete aircraft, this structure can still prove an impediment unless a large number of blocks are used. One is then faced with the problem of defining the blocks and their interfaces, which becomes increasingly difficult as the number of blocks increases. A radical alternative to a structured mesh is the use of triangles in two dimensions and tetrahedra in three dimensions. There is no longer any inherent regularity in the mesh and one has made a complete transition from the global to a local description. This characteristic gives unstructured meshes maximum flexibility in treating complex geometries while retaining a high degree of control over mesh point distribution. Unfortunately the extreme generality makes the generation of unstructured meshes a singularly difficult task. In two dimensions several triangle mesh generation schemes are available, but in three dimensions tetrahedral mesh generation schemes are scarce. The first report of a tetrahedral mesh around a complete aircraft is given by Bristeau et al. [18] This was a remarkable achievement, which has perhaps not received the recognition it deserves. No details of the mesh generation are given in the publications by this group, an unfortunate omission that has no doubt contributed to the lack of adequate recognition.

There appear to be two main approaches to unstructured mesh generation which have proved successful in three dimensions. The first is based on the Delaunay triangulation and its dual geometric construct, the Voronoi diagram. This idea has been successfully exploited in two dimensions for a variety of applications. For aerodynamic calculations, use of the Delaunay triangulation was initiated by Weatherill [99,100] and developed in three dimensions by Baker [7,8,47,48]. The alternative approach is the moving front technique, successfully applied in two dimensions by Lo [51] and recently developed in three dimensions by Peraire et al. [63] and Löhner [52].

The distinguishing feature of unstructured meshes is their ability to handle three dimensional objects of arbitrary shape and complexity. This is accomplished without the need for introducing blocks, mesh controlling functions and other artifacts, which plague structured mesh generation and prevent its evolution to a fully automated procedure. The criticism that has been leveled at unstructured meshes is the difficulty of constructing efficient flow solvers. It is interesting to observe that the finite element method, which has exploited unstructured

meshes from its inception, has seen little development of the algorithms used to invert the mass matrix and solve the problem. Developments in Computational Fluid Dynamics began with simple structured meshes and this proved to be a stimulus for innovative ideas, which have led to highly efficient flow algorithms. Although it appears unlikely that implicit schemes can be adapted in any simple way to unstructured meshes, advances in explicit schemes (e.g. multi–stage Runge Kutta methods, local time stepping etc.) carry over equally well to triangular meshes. Indeed Jameson's unstructured flow solver [47,48] rivals the best of the structured flow algorithms for computational speed. Recent work by Mavriplis [55] indicates that multigrid will also work very effectively on unstructured meshes.

It is likely that future developments will use combinations of both structured and unstructured meshes. In fact Nakahashi and Obayashi [59] have used such a combination, exploiting a structured mesh near solid boundaries to solve the Navier Stokes equations and a triangular or tetrahedral mesh in the remainder of the flow field. Actually the reverse procedure would be more appropriate for general geometries. It is the near field of closely coupled components such as wing/strut/nacelle and wing/store combinations that prove difficult when one tries to fit hexahedral cells. The likely outcome is that a tetrahedral mesh will be employed in such regions with an structured mesh covering the surrounding space.

In the remainder of this paper we describe some of these ideas in more detail. Several excellent reviews and conference proceedings [30,73,85,86,89] have recently appeared and an AGARDograph containing contributions from several researchers will soon be published. From these publications the reader can get a comprehensive view of the current status of mesh generation. The text of Thompson et al. [90] is an excellent reference for much of the underlying theory. The aim of this paper is to provide a personal view by summarizing the most important developments and by attempting to interpret the many diverse ideas and techniques that have appeared. Also included is a brief discussion of two areas that can be expected to become increasingly important in the future. First, there is the question of mesh quality and, in particular, how changes in cell shape, aspect ratio and mesh stretchings can affect solution accuracy and algorithm performance [36,37,40,64,67,90,92,94,95,98]. These are difficult questions and our present knowledge is rudimentary. But it is likely that this area will receive much more attention, now that the major goal of complete aircraft meshes has been achieved.

The second area is the use of solution adaptive meshes. This is also a field of growing interest and several papers on this subject have already appeared [1,14,17,25,39,41,42,54,57,58,60,61,63,83]. It is evident that the flow field around a complicated object such as a complete aircraft has a complicated structure and widely varying length scales. In order to resolve such features as shed vortices, shock wave patterns and ultimately separated flow, solution adaptive meshes will be needed no matter how many gigawords of computer memory become available. It is against this yardstick that further developments in mesh generation should be measured. How easily can the method cope automatically with an arbitrary shape, first generating an initial mesh and then adapting the mesh to the evolving solution so that all the salient flow features are crisply captured?

2. Mesh Generation Methods

In this section we provide thumbnail sketches of those mesh generation techniques that can be applied to complicated three dimensional shapes. For the purpose of illustration, formulae for the two dimensional implementation will be presented, and we therefore consider a transformation from physical coordinates x,y to another coordinate system defined by ξ,η. A point in the flow field can then be associated with the position vector

$$\underline{r}(\xi,\eta) = (x(\xi,\eta),\ y(\xi,\eta))$$

The reader should consult the references for a detailed derivation of these methods and their generalization to three dimensions.

2.1 Numerical Techniques

2.1.1 Elliptic Systems

In two dimensions the Laplace system

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= 0 \\ \eta_{xx} + \eta_{yy} &= 0 \end{aligned} \tag{1}$$

may be solved to determine the coordinate transformation $x(\xi,\eta)$, $y(\xi,\eta)$. In practice the problem is inverted to solve for x and y as dependent variables [88,90]. Equations (1) then assume the form,

$$\begin{aligned} \alpha x_{\xi\xi} - \beta x_{\xi\eta} + \gamma x_{\eta\eta} &= 0 \\ \alpha y_{\xi\xi} - \beta y_{\xi\eta} + \gamma y_{\eta\eta} &= 0 \end{aligned} \tag{2}$$

where

$$\begin{aligned} \alpha &= x_\xi^2 + y_\xi^2 \\ \beta &= x_\xi x_\eta + y_\xi y_\eta \\ \gamma &= x_\eta^2 + y_\eta^2 \end{aligned}$$

For a boundary conforming mesh, the boundaries are the lines $\xi = \xi_1, \xi_2$ and $\eta = \eta_1, \eta_2$ say, and the solution of equations (2) thus defines a mapping from a rectangle in (ξ,η) space to the required region in physical space. The lines ξ = const. and η = const. correspond to mesh lines in physical space and the extremum principle for harmonic functions ensures that mesh lines do not overlap.

2.1.2 Control Functions

Unfortunately the Laplace system has a strong smoothing effect and often produces an undesirable mesh point distribution. This is particularly evident in regions of high curvature on the boundaries. Control of mesh point distribution can be accomplished by the introduction of controlling functions to generate a system of Poisson equations

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= P(\xi,\eta) \\ \eta_{xx} + \eta_{yy} &= Q(\xi,\eta) \end{aligned} \tag{3}$$

The source terms P and Q are constructed to either attract or repel points about a certain position or line. For example

$$P(\xi,\eta) = -a\ \mathrm{sgn}(\xi-\xi_o)e^{-c|\xi-\xi_o|}, \quad a > 0,\ c > 0 \tag{4a}$$

attracts ξ–lines towards the line $\xi=\xi_o$ while

$$P(\xi,\eta) = -a\ \mathrm{sgn}(\xi-\xi_o)e^{-c[(\xi-\xi_o)^2 + (\eta-\eta_o)^2]^{1/2}} \tag{4b}$$

attracts ξ–lines to the point (ξ_o,η_o). Ways of automating the choice of the amplitude a and decay factor c have been suggested, and more general forms of the source terms are available [56,79,90]. However, non–overlapping of mesh lines can no longer be guaranteed and the specification of control functions requires some care.

2.1.3 Hyperbolic Systems

Other sets of partial differential equations might be expected to produce satisfactory meshes. An interesting development of Steger and Sorenson [80] along these lines is the solution of the following equations

$$\begin{aligned} x_\xi\, x_\eta + y_\xi y_\eta &= 0 \\ x_\xi y_\eta - x_\eta y_\xi &= V(\xi,\eta) \end{aligned} \tag{5}$$

which can be shown to define a hyperbolic system. The first equation of this pair is the orthogonality condition; the second determines the local cell area according to a specified area distribution $V(\xi,\eta)$. If the x and y coordinates of one boundary surface are defined on $\eta=\eta_1$ say, then the system can be marched out in the direction of increasing η. The outer boundary surface cannot, of course, be specified for a hyperbolic system. This is often quite acceptable for external flow problems, but makes a hyperbolic system unsuitable for generating meshes for some geometries, typically those used in internal flow problems.

2.2 Algebraic Methods

2.2.1 Multi–Surface Fitting

An interesting approach that was introduced by Eiseman [28] is based on the idea of interpolating curves along specified directions rather than through specific points. Consider two boundary surfaces $r_1(\xi)$ and $r_N(\xi)$ corresponding to the lines $\eta=\eta_1$ and $\eta=\eta_N$. Let $\underline{r}(\xi,\eta) = (x(\xi,\eta), y(\xi,\eta))$ be the position vector of the point (x,y) and introduce intermediate surfaces $r_i(\xi)$, i=2,...,N–1 which are constructed to direct the η–lines from the boundary surface $\eta=\eta_1$ to the opposite boundary surface $\eta=\eta_2$. We now construct a curve that is tangent to the direction $\underline{r}_{i+1}(\xi) - \underline{r}_i(\xi)$ at the position $\eta=\eta_i$. Thus we require

$$\frac{\partial \underline{r}}{\partial \eta} = \sum_{i=1}^{N-1} A_i \psi_i(\eta)(\underline{r}_{i+1}(\xi) - \underline{r}_i(\xi)) \tag{6}$$

where A_i are suitable normalizing constants and the $\psi_i(\eta)$ are univariate interpolating functions subject to the constraints

$$\psi_i(\eta_j) = \delta_{ij} \qquad i,j=1,2,...,N-1$$

On integrating eqn (6) we obtain

$$\underline{r}(\xi,\eta) = \underline{r}_1(\xi) + \sum_{i=1}^{N-1} A_i\, G_i(\eta)\, (\underline{r}_{i+1}(\xi) - \underline{r}_i(\xi))$$

where

$$G_i(\eta) = \int_{\eta_1}^{\eta} \psi_i(s)ds \, .$$

The choice $r_1(\xi)$ for the integration constant ensures that $r(\xi,\eta)$ coincides with $r_1(\xi)$ when $\eta=\eta_1$. We can now choose $A_i = 1/G_i\,(\eta_N)$ to ensure that $r(\xi,\eta)$ coincides with the outer boundary surface $r_N(\xi)$ at $\eta=\eta_N$.

This approach offers great generality and a high degree of control over mesh point distribution. From a practical viewpoint it must be difficult to specify the intermediate controlling surfaces for regions contained by highly complicated boundaries (e.g. the space around a nacelle/strut/wing). On the other hand, once one has accepted the need for a blocked mesh, a more specific interpolation procedure such as transfinite interpolation is probably adequate.

2.2.2 Transfinite Interpolation

The theory of multivariate interpolation, developed in general terms by Gordon [38], has been successfully exploited by Eriksson [31,32] as a means of generating meshes exhibiting a high degree of control over the mesh point distribution. Unlike Eiseman's multisurface approach, only information about the distribution of boundary points and the slope of mesh lines meeting the boundaries is needed. However, a careful choice of the interpolating functions is required to obtain a smooth mesh when one or more of the boundary surfaces contains a slope discontinuity. This aspect is discussed further in reference 31 which provides a clear description of Eriksson's work. A particularly attractive aspect of this approach is the way in which an interpolating function in two or three dimensions can easily be constructed from univariate interpolants. To illustrate this in two dimensions we again consider a mapping of the rectangular region in ξ,η space defined by the boundaries $\xi=\xi_1$, $\xi=\xi_2$, $\eta=\eta_1$ and $\eta=\eta_2$. We require the vector function $\underline{r}(\xi,\eta) = (x(\xi,\eta), y(\xi,\eta))$ given a prescribed functional variation $r(\xi_1,\eta)$, $r(\xi_2,\eta)$, $\eta_1\le\eta\le\eta_2$ and $r(\xi,\eta_1)$, $r(\xi,\eta_2)$, $\xi_1\le\xi\le\xi_2$ on the boundaries. First, we define two projections that correspond to a pair of univariate interpolation schemes,

$$\begin{aligned} \pi_\xi \underline{r} &= \underline{r}(\xi_1,\eta)\,\phi_1(\xi) + \underline{r}(\xi_2,\eta)\,\phi_2(\xi) \\ \pi_\eta \underline{r} &= \underline{r}(\xi,\eta_1)\,\psi_1(\eta) + \underline{r}(\xi,\eta_2)\,\psi_2(\eta) \end{aligned} \qquad (7)$$

where

$$\begin{aligned} &\phi_1(\xi_1) = 1\,, && \phi_1(\xi_2) = 0 \\ &\phi_2(\xi_1) = 0\,, && \phi_2(\xi_2) = 1 \\ &\psi_1(\eta_1) = 1\,, && \psi_1(\eta_2) = 0 \\ &\psi_2(\eta_1) = 0\,, && \psi_2(\eta_2) = 1 \end{aligned} \qquad (8)$$

We now define the transfinite interpolating function as the Boolean sum of these projections

$$\begin{aligned}\pi_\xi \oplus \pi_\eta \, \underline{r} &= (\pi_\xi + \pi_\eta - \pi_\xi \pi_\eta)\, \underline{r} \\ &= \underline{r}(\xi_1,\eta)\, \phi_1(\xi) + \underline{r}(\xi_2,\eta)\, \phi_2(\xi) + \underline{r}(\xi,\eta_1)\, \psi_1(\eta) + \underline{r}(\xi,\eta_2)\, \psi_2(\eta) \\ &\quad - \underline{r}(\xi_1,\eta_1)\, \phi_1(\xi)\, \psi_1(\eta) - \underline{r}(\xi_2,\eta_1)\, \phi_2(\xi)\, \psi_1(\eta) \\ &\quad - \underline{r}(\xi_1,\eta_2)\, \phi_1(\xi)\, \psi_2(\eta) - \underline{r}(\xi_2,\eta_2)\, \phi_2(\xi)\, \psi_2(\eta)\end{aligned} \tag{9}$$

Along the boundaries the interpolating function reduces to the prescribed form. The functions ϕ_1, ϕ_2, ψ_1 and ψ_2 can be chosen arbitrarily provided they satisfy the conditions (8). The simplest possible choice corresponding to linear interpolation in ξ and η is given by,

$$\phi_1(\xi) = \frac{\xi_2 - \xi}{\xi_2 - \xi_1}\,, \qquad \phi_2(\xi) = \frac{\xi - \xi_1}{\xi_2 - \xi_1}$$

$$\psi_1(\eta) = \frac{\eta_2 - \eta}{\eta_2 - \eta_1}\,, \qquad \psi_2(\eta) = \frac{\eta - \eta_1}{\eta_2 - \eta_1}$$

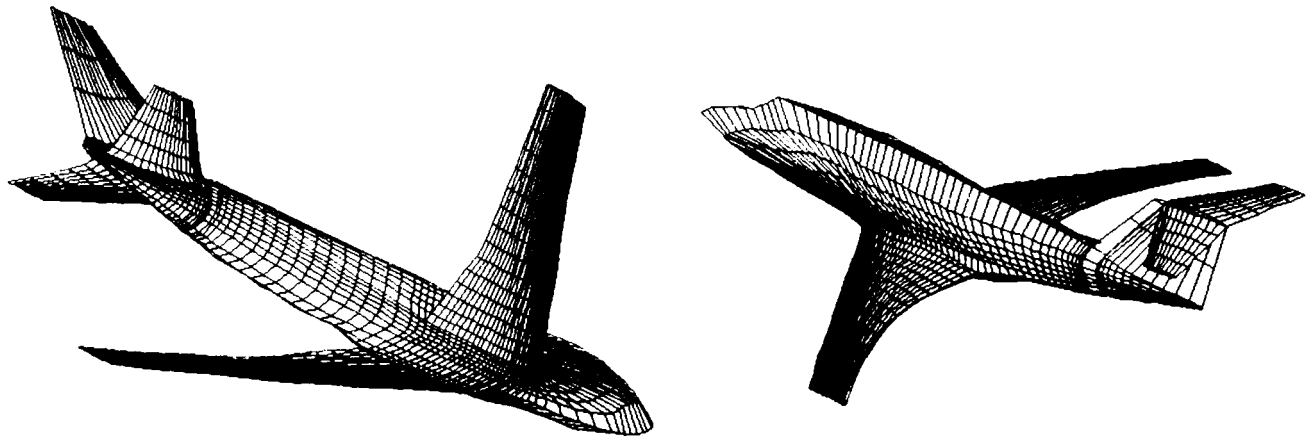
The projections (7) can be generalized to allow for a specification of the derivative of $\underline{r}$ normal to the boundaries. The univariate interpolants ϕ and ψ must then be osculatory functions that satisfy conditions (8) together with a further set of constraints on their derivatives at the end points.

2.2.3 Sequential Mapping

Another algebraic method that retains something of the spirit of the original coordinate transformation methods based on conformal mappings and shearings is the use of a sequence of mappings [6,44]. This approach has the virtue of retaining a high degree of control over mesh point distribution. By applying several simple mappings it is possible to reduce a relatively complicated configuration to a simple generic shape.

For example, a wing/fuselage/tail combination can be mapped into a pair of surfaces plus part of the symmetry plane. A Joukowski mapping followed by a shearing will take the fuselage into the symmetry plane. Further combinations of conformal mappings and shearings can be constructed to reduce the tail to a single sheet and unwrap the wing. It is then fairly straightforward to interpolate a set of coordinate surfaces which conform with the mapped wing surface and contain the tail sheet. In addition the coordinate surfaces are required to coincide with the mapped fuselage crown line on the symmetry plane. The set of mesh points in mapped space is now passed through the inverse of the original mapping sequence. This restores the aircraft geometry and produces a mesh that conforms with all boundary surfaces. A very similar approach was pursued independently by Shmilovich and Caughey [72].

The technique is fairly powerful, and the surface mesh for two examples of meshes generated by Baker's method are presented in figure 2. This mesh generator was linked to Jameson's finite volume scheme to produce the first published Euler flow solutions over wing/fuselage/tail combinations [44,46]. Although this concept could, in principle, be extended to include engine nacelles, it becomes increasingly difficult to fit the coordinate surfaces in mapped space, and some degree of user intervention would then be required to treat different geometries. As we mentioned earlier, the use of global coordinate transformations is not sufficiently flexible to deal with complete aircraft configurations, leading one to accept that a multiblock approach is unavoidable.



(a) Boeing 747–200

(b) Lear Jet

Fig. 2. Surface Mesh from Baker, Ref. 6 and 44

2.2.4 Grid Blending

Recently Steinhoff [82] introduced the idea of blending a collection of meshes, each one of which is generated for a separate region, to form a smooth global mesh. In some respects this idea has some of the features of a multiblock method in so far as it is based on defining several relatively simple, separate meshes rather than a global transformation. Suppose, for example, that there are N meshes and that the vector

$$\underline{r}_{ijk}(m) = (x_{ijk}(m), y_{ijk}(m), z_{ijk}(m))$$

represents the point with indices (i,j,k) generated by the mth mesh where m=1,2,...,N. The set of points represented by $\underline{r}_{ijk}(1)$ might be the mesh around the wing, $\underline{r}_{ijk}(2)$ the mesh around the fuselage, $r_{ijk}(3)$ the mesh around the tail and so on. The actual point $\bar{\underline{r}}_{ijk}$ corresponding to the indices (i,j,k) is then formed by the weighted combination

$$\bar{\underline{r}}_{ijk} = \frac{\sum_{m=1}^{N} P_{ijk}(m)\ r_{ijk}(m)}{\sum_{m=1}^{N} P_{ijk}(m)} \qquad (10)$$

where the $P_{ijk}(m)$, m=1,...,N are weighting functions constructed so that the mth mesh dominates in the region where it is needed. The weights for the other meshes should be small in this region, and decay to zero near the boundary surface where the mesh is determined entirely by the component mesh $r_{ijk}(m)$.

Although this approach is attractive, the choice of the weighting functions is by no means easy and will presumably have to be done on a trial and error basis for each new configuration. How difficult or easy this task is compared with the construction of a blocked mesh will determine whether the blending mesh approach is a viable technique for treating arbitrary complex configurations.

3. Multiblock

The need to handle complicated geometries while maintaining adequate control over mesh point distribution and cell shape can perhaps be best achieved by introducing a multiblock structure. The flow field is broken up into a number of blocks by defining a set of surfaces which will represent the block boundaries. The individual blocks can then be meshed and the flow algorithm constructed to exploit the blocked structure. In fact the flow solution can proceed in each block independently, with information between contiguous blocks being passed at the common interface of two block boundaries. This makes multiblock particularly suitable for computer architectures with a parallel processing capability. Various possibilities arise depending on what degree of continuity is required at the block interfaces (see figure 1).

3.1 Overlapped

If we do not define a precise interface but introduce separate meshes for each component, we obtain a system of overlapped meshes. This approach was originally considered by Atta [4] and has been extensively developed by Benek et al [10–12]. The lack of any constraint at the block boundaries makes mesh generation for the individual blocks much easier. However the penalty for this facility is the problem of transferring information from each component mesh to its neighbor. This difficulty has some similarity with the problem of applying solid wall boundary conditions on a non–aligned mesh. Similar reservations about accuracy of the interpolation procedures and the ease of maintaining conservation apply to overlapped meshes. Recent work by Berger [13] shows that conservation can usually be achieved though with some difficulty, particularly in three dimensions, and it is by no means clear whether stability could be assured in the general case.

3.2 Patched

If we define a block structure and require the separate meshes to conform with the surfaces of their respective block boundaries, we obtain a patched mesh. There will, in general, be no continuity of mesh lines from neighboring blocks at the block interfaces. However, interpolation at the interfaces is now less demanding than that required by an overlapped system, and this approach retains the advantage of allowing a highly refined mesh in specific regions without imposing unnecessary refinement elsewhere. A patched mesh was used by Baker et al. [9] to obtain a refined mesh in the vicinity of the tail for a wing/fuselage/tail combination. This technique has also been successfully applied to solve the Navier–Stokes equations around an F–16 [34,75,76] (see figure 3).

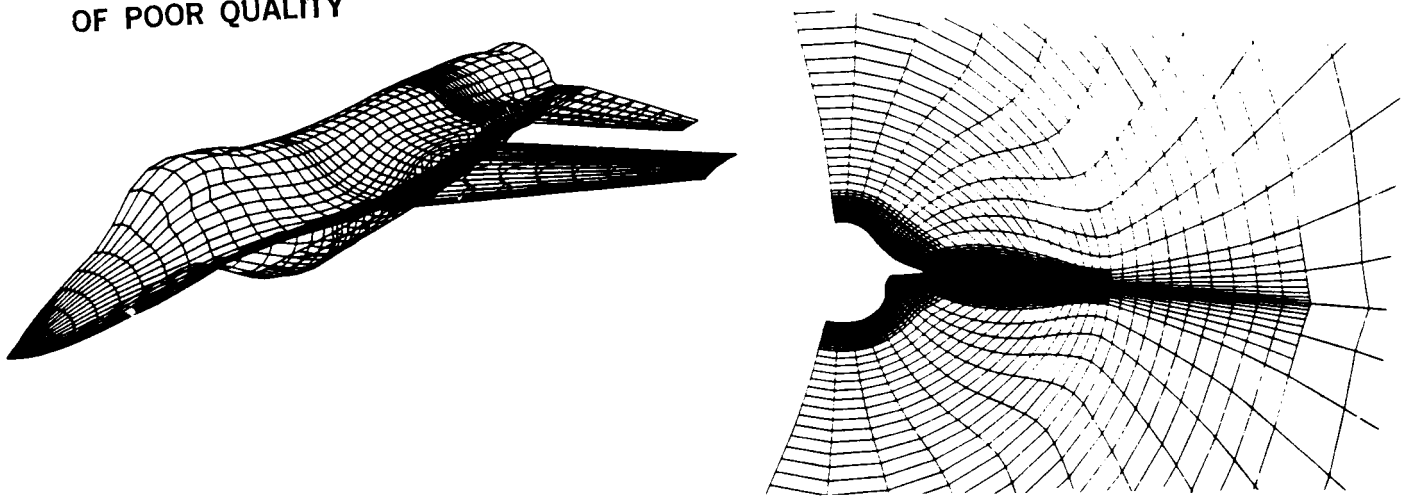


Fig. 3. Mesh from Sorenson, Ref. 75

3.3 Composite

The composite method can be regarded as a special case of the patched approach in which mesh lines are required to be continuous across block interfaces This has the drawback that mesh refinement in one block, involving an increase in the number of mesh points on block boundaries will induce a corresponding refinement in neighboring blocks and so on throughout the entire mesh. However, it is probable that the advantage of mesh line continuity at the block interfaces outweighs the disadvantage of requiring an unduly fine mesh in certain regions. Mesh smoothness is further enhanced by requiring slope continuity as well, and the extra burden this places on the mesh generation within each block is amply justified by improved accuracy that results.

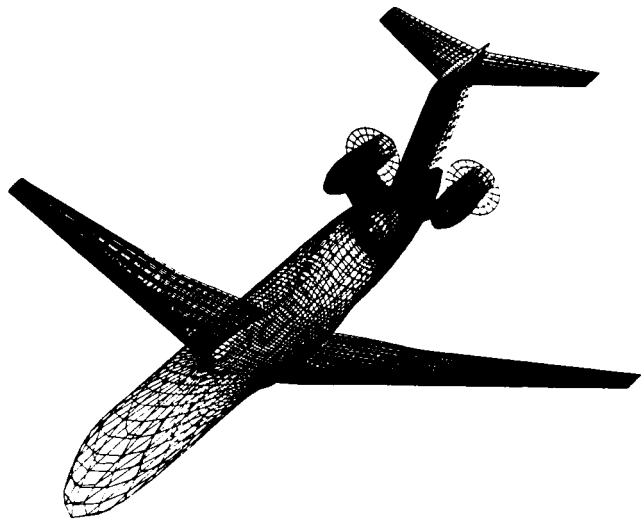


Fig. 4. Surface Mesh

from Yu et al., Ref. 106

3.4 Miscellaneous Results

Over the last two years several examples have appeared in the literature showing flow calculations on structured meshes for aircraft geometries. Yu et al. [105,106] have used a two block Poisson solver to obtain a C–H mesh around a high tail/aft mounted propfan configuration (figure 4). Eberle and Schwarz [27] have generated a single block H–H mesh around a wing/canard/fuselage/tail combination (figure 5). They used an elliptic mesh generator that solves the biharmonic equation, thus gaining a higher degree of smoothness than would be obtained from the Laplace system. A similar configuration, but without a vertical tail, has been treated by Eriksson et al. [33], who used transfinite interpolation to produce a two block H–O mesh (figure 6).

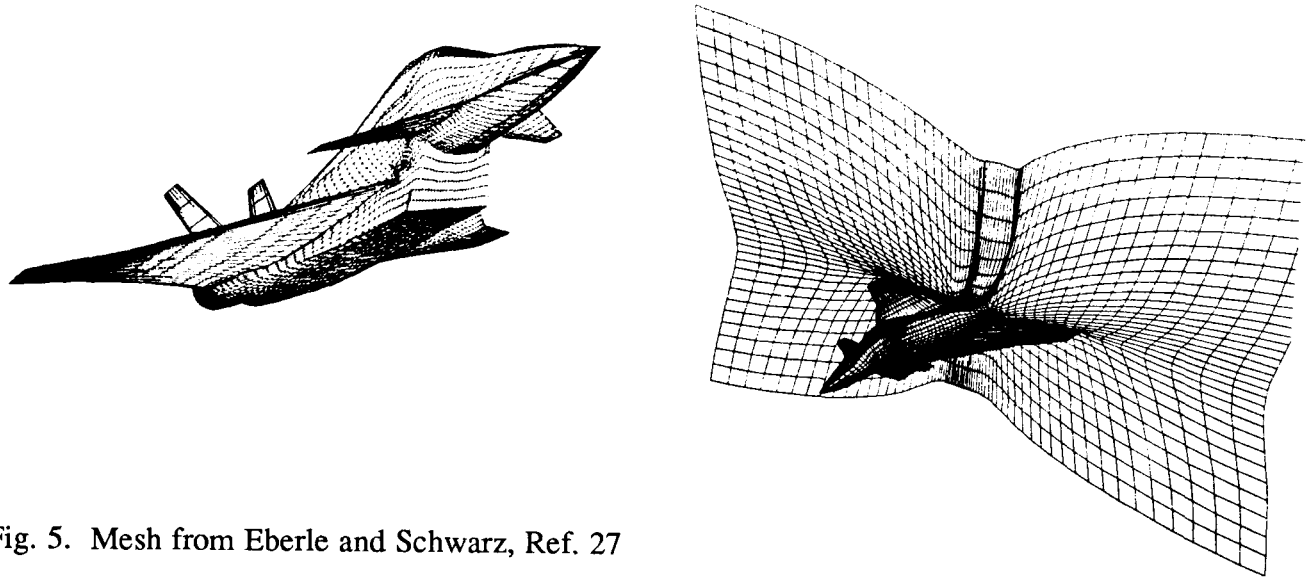


Fig. 5. Mesh from Eberle and Schwarz, Ref. 27

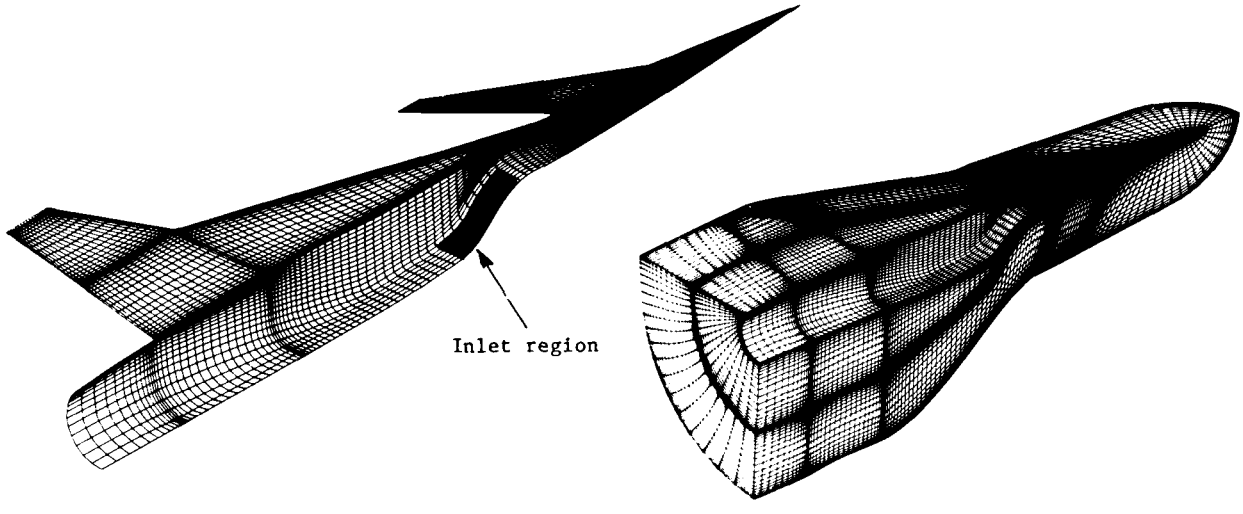


Fig. 6. Mesh from Eriksson et al., Ref. 33

A mesh generator that is based on only one or two blocks must necessarily be restricted to configurations of one generic type. Other types of configuration could be handled only after extensive recoding of the mesh generation software. More general methods based on several blocks have been developed by Weatherill and Forsey [101] and Shaw et al. [71], who use a Poisson solver in each block (figure 7), and also by Fritz et al. [35] and Seibert [69] (figure 8), who generate an initial mesh algebraically and then iterate with a Poisson solver to obtain a smooth point distribution.

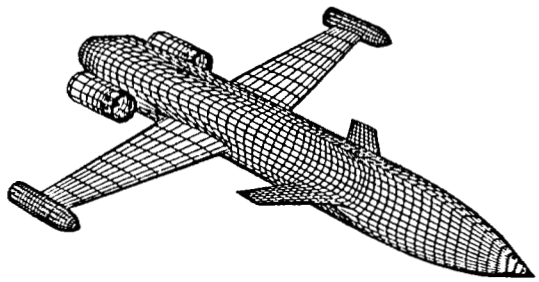


Fig. 7. Mesh from Weatherill and Forsey, Ref. 101

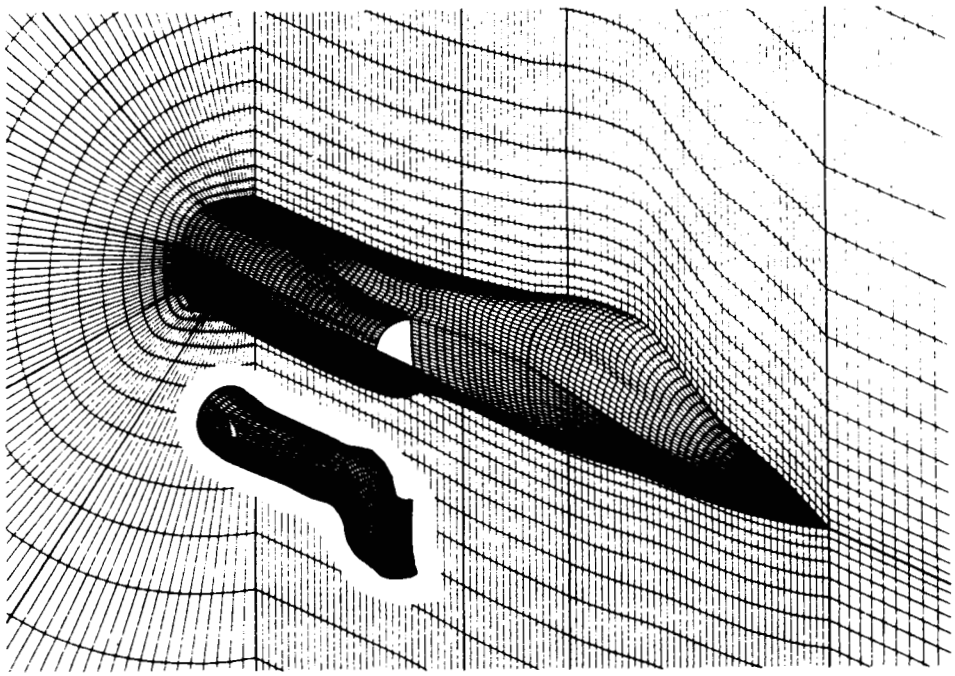


Fig. 8. Mesh from Seibert, Ref. 69

The inclusion of nacelles, struts and stores introduces an added degree of difficulty. The work of Vigneron et al. [96,97] (figure 9), who used algebraic generation followed by a Poisson type smoothing is therefore particularly impressive. Another outstanding contribution is the paper by Sawada and Takanashi [68]. They used transfinite interpolation to produce a mesh around a complete aircraft with over–wing nacelles (figure 10).

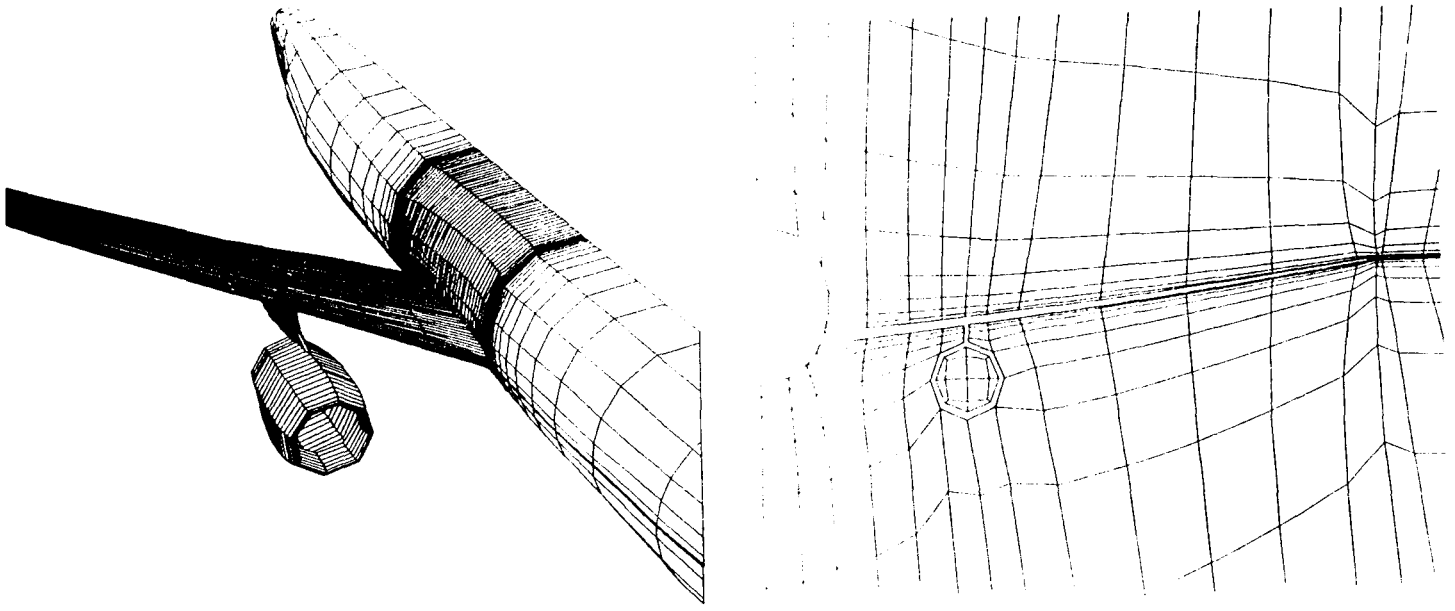
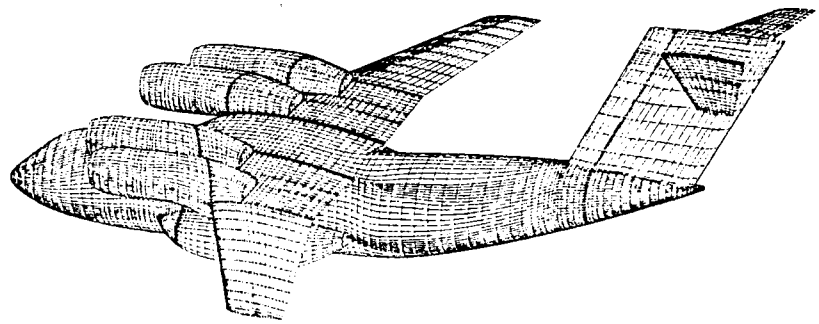


Fig. 9. Mesh from Vigneron et al., Ref. 97

3.5 Further Developments

Once one has accepted the need for a multiblock scheme and the extra data structure that this entails, it makes sense to keep the individual blocks fairly simple (i.e. as close to rectangular as possible) in order to simplify the mesh generation process within each block. To treat a complete aircraft configuration may require around 100 blocks, but this can certainly be accommodated if the data structure is sufficiently general and contains complete information about the block face and edge contiguities and the relative orientation of neighboring blocks [15,16,81,87,101,102]. Transfinite interpolation, with prescribed mesh line slopes at the block boundaries would appear to be a perfectly adequate method of mesh generation within each block. This could be followed by a few iterations of a Laplace solver if further smoothness were needed. The remaining problem is the decomposition of the flowfield to give the block structure. Ideally this should be completely automated, but this may be difficult to achieve for arbitrary aircraft shapes. The consequent need for good interactive graphics has been mentioned by several authors and it appears that structured mesh generation for highly complex geometries will continue to rely on the intervention of a skilled user.



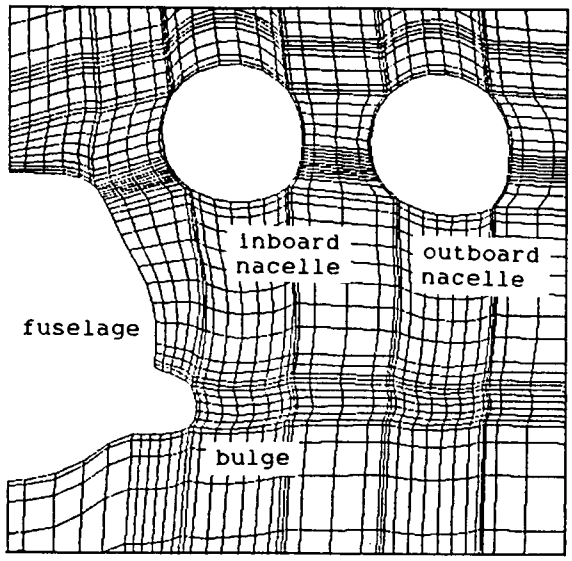


Fig. 10. Mesh from Sawada and Takanashi, Ref. 68

4. Triangulations

A review of unstructured mesh generators prior to 1980 has been given by Thacker [84]. Significant developments, however, have taken place since then and efficient triangulation methods are now available for two and three dimensional probiems. It is worth noting that any planar triangulation possesses a degree of structure that is not present in three dimensions. First we observe that given a set of V points in a plane such that B lie on the convex hull, the number of edges E and the number of triangles (or faces) F are specified by the formulae

$$E = 3(V - 1) - B$$
$$F = 2(V - 1) - B$$

In particular, since $B = 0(V^{1/2})$ we see that for a large number of points,

$$E \sim 3V \quad \text{and} \quad F \sim 2V$$

Moreover, if k denotes the average number of edges meeting at a point then, since each edge is associated with exactly two points, the number of edges is also given by

$$E = \frac{kV}{2}$$

Combining this with the above expression for the number of edges leads to the following expression for k,

$$k = 6 - \frac{6}{V} - \frac{2B}{V}$$

and so for large V we have

$$k \sim 6.$$

The average number of edges meeting at a point is therefore six, and since at least three edges must meet at any point, this places a fairly rigid constraint on the expected variation in the number of edges meeting at any point. In other words, we do not expect too much variation between triangulations for a given set of points in two dimensions. In particular, a high degree of regularity, in terms of the number of edges meeting at any point, is to be expected no matter how we triangulate the points.

There is no such invariance in the number of tetrahedra, triangular faces and edges in three dimensions. It is well known, for example, that a cube can be cut into either five or six tetrahedra. In fact, it is possible to find triangulations of N points in three space which contain 0(N) points and also triangulations which contain $0(N^2)$. Intuitively one would expect that an 0(N) triangulation would be likely to contain better shaped tetrahedra and lead to a greater degree of mesh regularity. The important requirement is, therefore, to find a triangulation scheme and mesh point distribution that will achieve this aim.

4.1 Delaunay Triangulation

The Delaunay triangulation of a set of points and the dual geometric construct, the Voronoi diagram, are extremely fertile concepts that have been the subject of considerable theoretical investigation and have found numerous practical applications. The Voronoi diagram marks off the region of space that lies closer to each point than the other points. This is illustrated for the planar case in figure 11. The solid lines make up the Voronoi diagram which form a tessellation of the space surrounding the points. Each Voronoi tile (e.g. the hatched area around point P) consists of the region of the plane that is closer to that point than any other. The edges of the Voronoi diagram are formed from the perpendicular bisectors of the lines connecting neighboring points. In general three edges will meet at a vertex which must be equidistant from three forming points (e.g. points P, Q_3, Q_4 in figure 11) and hence each vertex is the circumcenter of the triangle formed by three points. This determines a unique triangulation known as the Delaunay triangulation and is such that the circumcircle through each triangle contains no points other than its forming points.

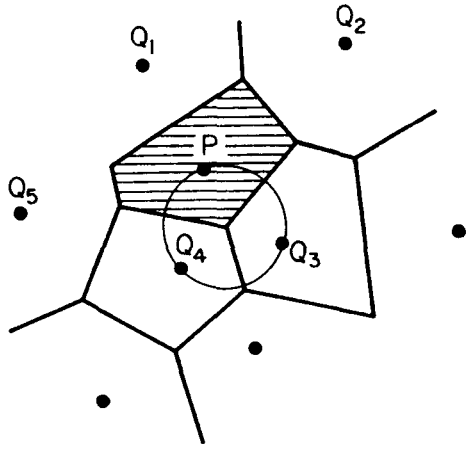


Fig. 11. Voronoi Diagram

The complete mesh is generated by the triangulation of a cloud of points surrounding the aircraft. The point distribution can be defined in any way whatsoever, and it is this generality which makes the Delaunay approach so powerful. The surface geometry for the F-15 was based on a network of points originally generated for use by a panel method. Flow field points were introduced in three sets corresponding to a farfield, midfield and nearfield distribution. The farfield points were generated by defining a Cartesian box with a relatively coarse distribution covering the entire field. A finer distribution of points, again in a regular array, was added in a region extending a few wing chords around the aircraft. Finally, the nearfield points were generated by placing points along normals directed outward from the surface points. Any points which fall inside the aircraft structure are detected and removed. The remaining points are triangulated to create a mesh which varies from a fine distribution of cells near the aircraft to a coarser distribution in the farfield.

The two photographic plates show the surface mesh and a flow solution for the F-15. The mesh was generated for one half of the aircraft; the mesh shown in plate 1 and the surface contours shown in plate 2 have been reflected about the aircraft plane of symmetry. The mesh surrounds one half of the full configuration, including the interior of the duct which extends from the fuselage inlet to the nozzle at the rear. Of the 77,000 points in the mesh, approximately 5,000 are on the aircraft surface, and the complete set of points has been connected to form a collection of 460,000 cells. The surface triangulation shown in plate 1 provides a good indication of the mesh resolution, and plate 2 shows the pressure distribution for a case run at a freestream Mach number of 0.81 and an angle of attack of 4.84 degrees. In the color coded picture (plate 2), red represents low pressure and blue represents high pressure with the fringe bar indicating the graduations in between these extremes.

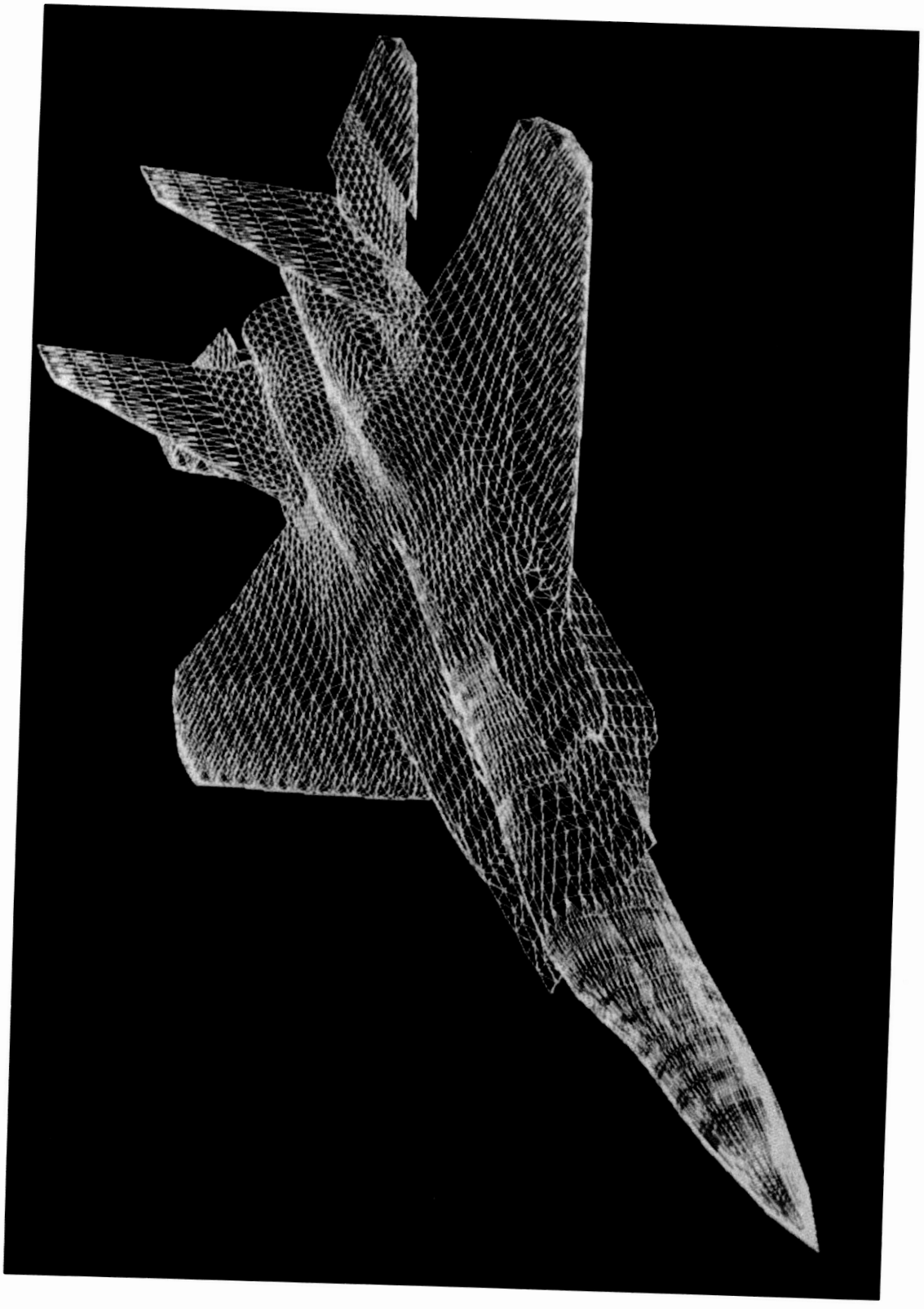


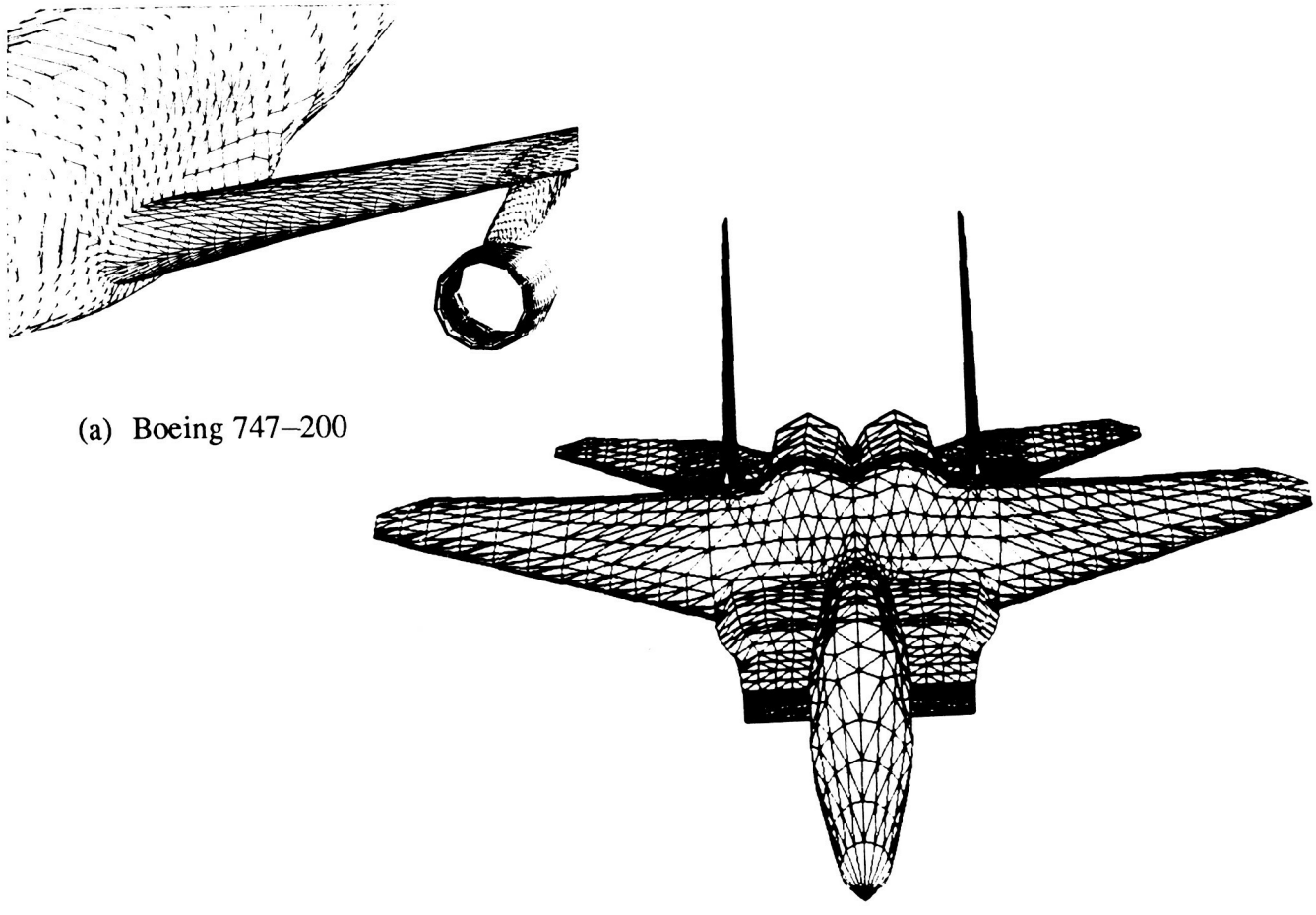
Plate 1

ORIGINAL PAGE
COLOR PHOTOGRAPH

Plate 2

These concepts generalize to higher dimensions. In particular, the Delaunay triangulation of three space is the unique triangulation such that the circumsphere through each tetrahedron contains no points other than its forming points. In two dimensions this circle criterion can be shown to be equivalent to the equiangular property that selects the triangulation which maximizes the minimum of the six angles in any pair of two triangles which make up a convex quadrilateral. No equivalent characterization is known in three dimensions but the circle criterion can still be regarded as selecting a good triangulation for the given set of points.

The Delaunay triangulation has been applied to mesh generation for structural problems by Cavendish et al. [21] and for semi–conductor device simulation by Cendes [22,23] et al. A three dimensional, Delaunay based, mesh generator developed by Baker [7] has been linked to the finite element flow solver of Jameson and reference 47 contains the first published example of an Euler flow calculation over a complete aircraft. Detail of the surface mesh around the inner nacelle/strut/wing area of the Boeing 747–200 is presented in figure 12(a) and figure 12(b) shows the surface triangulation associated with a tetrahedral mesh that was generated for the F–15. The generality and flexibility of this method is clearly evident and the technique is capable of generating tetrahedral meshes around any three dimensional object.



(a) Boeing 747–200

(b) McDonnell Douglas F–15

Fig. 12. Surface Triangulation from Baker, Ref. 7 and 47

4.2 Moving Front

An alternative approach to the triangulation problem is the moving front technique first suggested by Lo [51]. This idea is illustrated in figure 13, where a layer of triangles has already been placed around an internal boundary surface. The outer edges of this layer forms the front and a new triangle is constructed on each edge by connecting the edge to the nearest point in such a way that the shape of the new triangle is acceptable. When triangles have been formed on all front edges, a second layer of triangles will be in place. The procedure is repeated with the new front until the whole set of points has been triangulated.

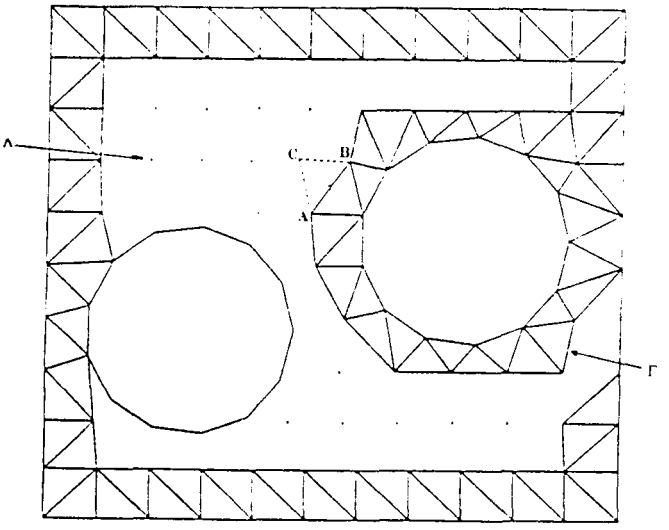


Fig. 13. Moving Front Method (taken from Lo, Ref. 51)

The extension of this idea to three dimensions is not an easy task since the front, formed by a layer of tetrahedra, is now made up of a surface of triangular faces. It is presumably necessary to take care when determining the intersections of planar faces, and to ensure that no overlapping of tetrahedra occurs or that holes are left when the front folds over on itself. Significant progress has been made in this area, however, by Peraire et al. [62,63] and also by Löhner [52]. Figure 14 shows detail of the surface triangulation associated with a tetrahedral mesh that Peraire generated around a wing/canard/fuselage/tail combination.

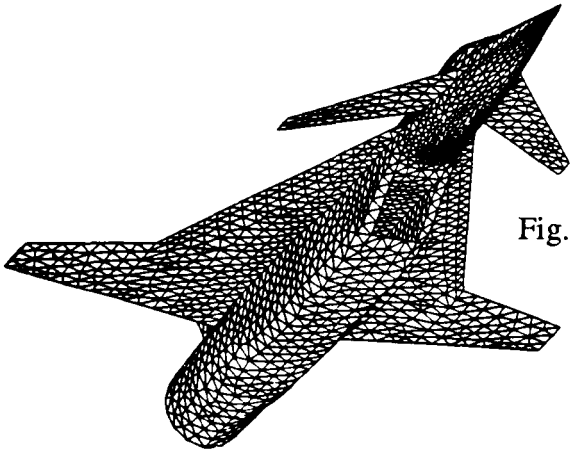


Fig. 14. Surface Triangulation for Wing/Canard/Fuselage/Tail Configuration (taken from Peraire et al., Ref. 63)

5. Mesh Type and Solution Accuracy

An inevitable consequence of creating a surface conforming mesh is the introduction of mesh stretching. Some degree of mesh stretching is clearly desirable in order to achieve an efficient distribution of mesh points throughout the flow field. However, the stretching should be smooth to prevent spurious, numerically generated effects from contaminating the flow solution. Investigation of this subject is still in its infancy and an improved understanding of this important area is clearly necessary. Two aspects of mesh distortion are considered below.

5.1 Truncation Error

To illustrate ideas we consider a one dimensional stretching from the interval $0 \le \xi \le 1$ to physical space defined by the interval $x_o \le x \le x_N$. In most cases there will not be an explicit mapping from ξ to x. The variation in mesh width can, however, be regarded as the result of an implied transformation $x(\xi)$. We assume that the mapped interval is divided into N equally spaced increments $\Delta\xi = 1/N$ and write $x_j = x(j\Delta\xi)$ with the further requirement

$$x_o < x_1 < x_2 < \ldots < x_j < \ldots < x_N$$

We also write $h_j = x_j - x_{j-1}$
and let $h = \max_j h_j$

As Thompson et al. [90] point out, it is necessary to distinguish between two senses of order. First there is the behavior of the error as the number of points in the field is increased while maintaining the same relative point distribution over the field. Consider the approximation to the first derivative f_x given by

$$\frac{f_{j+1} - f_{j-1}}{x_{j+1} - x_{j-1}} = \frac{f_\xi}{x_\xi} + \frac{\Delta\xi^2}{6x_\xi}\left(f_{\xi\xi\xi} - \frac{x_{\xi\xi\xi}}{x_\xi} f_\xi\right) + 0(\Delta\xi^4) \qquad (11)$$

The first term is, of course, the required derivative f_x; the second term is the principal part of the truncation error, which is formally second order. An alternative definition of order is based on the behavior of the error as the relative point distribution is changed so as to reduce the spacing locally with a fixed number of points in the field. Expanding f_{j+1} and f_{j-1} as functions of x about the position x_j we obtain

$$\frac{f_{j+1} - f_{j-1}}{x_{j+1} - x_{j-1}} = f_x + \frac{1}{2}(h_{j+1} - h_j)f_{xx} + \frac{1}{6}[h_{j+1}^2 - h_{j+1}h_j + h_j^2]f_{xxx} + \text{HOT} \qquad (12)$$

But

$$h_{j+1} - h_j = x_{\xi\xi}\Delta\xi^2 + 0(\Delta\xi^4)$$

$$h_{j+1}^2 - h_{j+1}h_j + h_j^2 = x_\xi^2\Delta\xi^2 + 0(\Delta\xi^4)$$

Hence

$$\frac{f_{j+1} - f_{j-1}}{x_{j+1} - x_{j-1}} = f_x + \frac{\Delta\xi^2}{2}\left(x_{\xi\xi} f_{xx} + \frac{x_\xi^2}{3} f_{xxx}\right) + \text{HOT} \tag{13}$$

Now

$$h_j = x_\xi \, \Delta\xi + 0(\Delta\xi^2)$$

so we have

$$\frac{f_{j+1} - f_{j-1}}{x_{j+1} - x_{j-1}} = f_x + \frac{h_j^2}{2}\left(\frac{x_{\xi\xi}}{x_\xi^2} f_{xx} + \frac{1}{3} f_{xxx}\right) + \text{HOT} \tag{14}$$

It is apparent that a necessary condition for the approximation to be second order in the second sense is that the term

$$\frac{x_{\xi\xi}}{x_\xi^2} = 0(1) \tag{15}$$

This is discussed by Thompson et al. [90], who point out that a similar condition is required to maintain local order in two dimensions. A further condition must also be imposed to limit the rate at which the Jacobian approaches zero. This extra condition is essentially a limit on the degree of nonorthogonality that can be tolerated.

It should be noted in passing that expression (13) or (14) is exact for linear f. This is not the case if we use the exact mapping derivative x_ξ in place of $x_{j+1} - x_{j-1}$. For then, we have

$$\frac{f_{j+1} - f_{j-1}}{2\Delta\xi x_\xi} = f_x + \frac{\Delta\xi^2}{2}\left(\frac{x_{\xi\xi\xi}}{3x_\xi} f_x + x_{\xi\xi} f_{xx} + \frac{x_\xi^2}{3} f_{xx}\right) + \text{HOT}$$

It is therefore preferable to evaluate the metric coefficients numerically by the same difference representation as is used for the dependent variable. This conclusion was probably first stated by Steger [78].

The issue of local accuracy and stability on nonuniform meshes has been examined in some detail by Turkel [92]. He first defines a local stretching factor $r_j = \frac{h_{j+1}}{h_j}$ and then classifies three groups of stretchings:

(a) Quasi–uniform (or algebraic) if

$$r_j = 1 + 0(h^p), \quad p > 0$$

(b) Exponential if (a) is not valid but

$$\frac{r_{j+1}}{r_j} = 1 + 0(h) \qquad \text{and}$$

(c) faster than exponential if neither (a) nor (b) is true.

Now $h_{j+1} - h_j = h_j(r_j - 1)$ and it follows that if the stretching is quasi–uniform then from expression (12), the approximation remains locally second order (i.e. second order in the second sense described above). A quasi–uniform stretching must therefore satisfy condition (15). Further we can write

$$r_j = \frac{h_{j+1}}{h_j} = \frac{x_\xi \Delta\xi + \frac{\Delta\xi^2}{2} x_{\xi\xi}}{x_\xi \Delta\xi - \frac{\Delta\xi^2}{2} x_{\xi\xi}} + \text{HOT}$$

$$= 1 + \Delta\xi \frac{x_{\xi\xi}}{x_\xi} + \text{HOT}$$

or

$$r_j = 1 + h_j \frac{x_{\xi\xi}}{x_\xi} + \text{HOT}$$

Thus condition (15) implies that the stretching is quasi–uniform.

Turkel considers several finite difference and quadrilateral based finite volume formulations and his main conclusions are that with quasi–uniform stretchings all second order techniques retain their accuracy locally. If the stretchings are exponential or faster, central difference approximations will usually deteriorate to first order accuracy, and in the case of a cell centered approximation, the finite volume approach can yield an inconsistent scheme.

Another illuminating investigation of truncation error has been provided by Roe [67]. He considers vertex based finite volume schemes and examines the error that arises from using a trapezoidal integration rule to evaluate the flux integral around different cell types. He concludes that only a few specific cell types retain second order accuracy locally. In particular, among quadrilaterals only a parallelogram can admit a locally second order accurate approximation. Furthermore he shows that a triangular mesh can only have first order accuracy.

This rather alarming conclusion would appear to put unstructured meshes at a serious disadvantage. However, recent work by Giles [36] suggests that although the truncation error on an unstructured mesh is first order, the solution error remains second order. Further investigation is required to shed more light on this difficult area. It is clear, however, that future investigations of accuracy must attempt to follow Giles' lead in estimating the solution error and not basing conclusions on the order of the truncation error alone.

5.2 Wave Propagation

Consider the scalar wave equation

$$u_t + cu_x = 0\,, \qquad c > 0 \tag{16}$$

which represents the propagation of disturbance traveling to the right with constant velocity c. We now discretize the spatial terms to obtain the semi–discrete form

$$\frac{du_j}{dt} = -c \frac{(u_{j+1} - u_{j-1})}{x_{j+1} - x_{j-1}} \tag{17}$$

On a uniform mesh $x_{j+1} - x_j = h = \text{const.}$ and this equation reduces to

$$\frac{du_j}{dt} = -c \frac{(u_{j+1} - u_{j-1})}{2h} \tag{18}$$

Equation (16) admits traveling wave solutions of the form

$$e^{i(kx - \omega t)} \quad \text{where} \quad k = \frac{\omega}{c}$$

If we look for similar solutions of the finite difference approximation (18) we obtain a dispersion relation

$$\omega = \frac{c}{h} \sin(kh), \qquad -\pi \le kh \le \pi$$

In other words, the mesh acts like a dispersive medium so that a component with wave number k will travel with a phase velocity

$$c^* = c \frac{\sin(kh)}{kh}$$

and a group velocity

$$\frac{d\omega}{dk} = c \cos(kh)$$

We observe that for small values of the wave number k, the group velocity is close to the continuum solution velocity c. For higher wave numbers this is no longer the case. In fact, when $kh = \pi/2$ corresponding to a wave length

$$\lambda \equiv \frac{2\pi}{k} = 4h$$

the component has zero group velocity, and at the highest wave number $kh = \pi$, which corresponds to the shortest wavelength

$$\lambda = 2h$$

the component travels upstream at speed c.

An investigation of the wave propagation characteristics for different time integration schemes has been pursued by Giles and Thompkins [37], Trefethen [91] and Vichnevetsky [93–95]. Of particular interest is recent work by Vichnevetsky [95] who considers a non–uniform mesh and looks for solutions of the form

$$u = A_\omega(x,t)\, e^{i(\phi(x) - \omega t)}$$

The wave number k and amplitude A_ω are now slowly varying functions of x. As the wave number changes on an expanding mesh, it is possible for the group velocity to decrease, reach zero and become negative (see figure 15). A solution component can therefore undergo a reflection and, if the mesh expands in all directions, it is possible for the wave to become trapped.

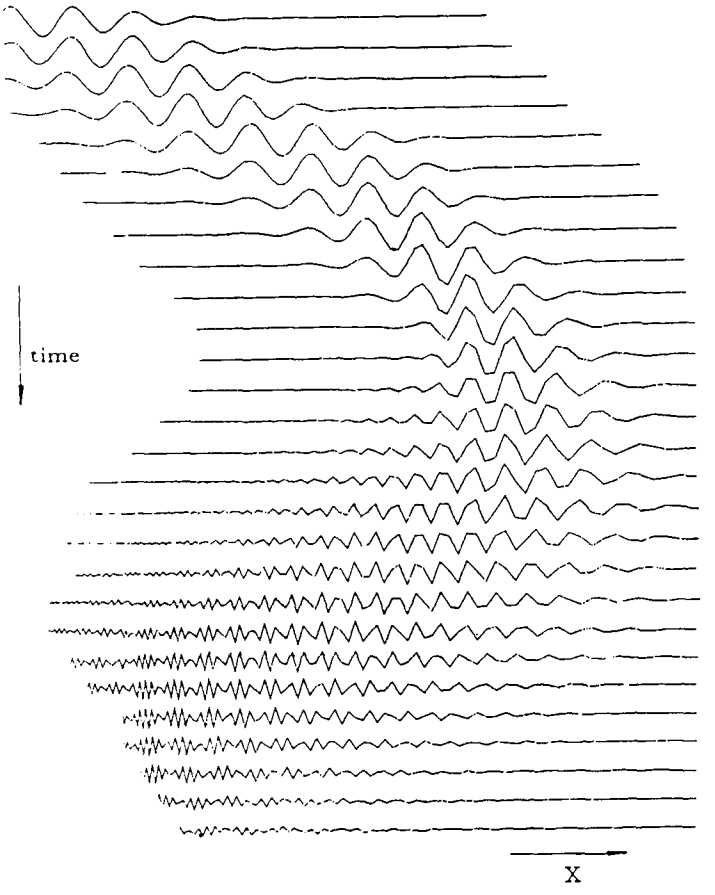


Fig. 15. Wave Reflection on an Expanding Mesh
(taken from Vichnevetsky, Ref. 95)

Vichnevetsky's analysis and indeed the results of all the investigators in this area is restricted to non–dissipative approximations. When artificial dissipation is present, these waves are damped and no undesired transients are left trapped in the mesh. However, the wave propagation analysis could have important implications for unsteady flow problems where the artificial dissipation should be kept small and not interfere with the true time accurate behavior.

6. Adaptive Meshing

It has long been recognized that a mesh determined independently of the flow solution is unlikely to resolve all the flow features. Specific regions of the flow field will be subject to rapid variations in flow variables. Solutions of the Euler equations require the accurate capture of shockwaves and the resolution of vortices that can emanate from sharp edges in three dimensional computations. With the advent of Navier Stokes calculations it becomes necessary to ensure a fine mesh in wake regions and areas of separated flow. It is clearly not possible to achieve the necessary mesh resolution a priori without introducing an unacceptably large number of points throughout the flow field. The use of an adaptive mesh that evolves with the flow solution is therefore an unavoidable requirement. In fact, it is anticipated that an adaptive meshing capability will be an integral part of all three dimensional flow codes within a few years.

6.1 Mesh Redistribution

An option that received early attention is use of mesh redistribution to achieve the required density of mesh points. One dimensional mesh stretchings that bunch the mesh lines have been considered, and stretchings in more than one direction have also received attention [29,39,58,90]. Within the framework of numerical generation methods, it is possible to link the source and repulsion terms to some measure of the truncation error or variation in flow variables. Other numerical procedures based on a variational approach have been tried [17,90]. Here some flow property is included in the functional to be minimized. The Euler–Lagrange equation that one obtains can then be solved numerically to generate the new mesh. A common difficulty of all the methods based on mesh redistribution is the need to ensure that the mesh cells do not become too distorted and, in particular, do not cross over each other. It is therefore necessary to monitor very closely changes in cell size and shape, and to halt the mesh adaptation if tolerances on these constraints are exceeded.

6.2 Mesh Enrichment

An alternative approach which does not suffer from these difficulties is mesh enrichment. Here the mesh is successively refined whenever the truncation error, or some convenient measure of the flow variables, becomes too large. Notable work in this area is that of Berger and Jameson [14] and Dannenhoffer and Baron [25,26]. The main difficulty of this approach is the need to keep track of the interfaces where refinement occurs. This requirement adds an overhead to the data handling and complicates the flow solver. It is also necessary to ensure conservation and stability at the interfaces where refinement occurs. These problems are difficult to treat in two dimensions; in three dimensions the complexity of this task is increased considerably.

It is in the context of mesh adaption that unstructured meshes enjoy a distinct advantage over structured meshes. A triangular mesh can easily be refined without the need to introduce interfaces [1,23,41,42,53,54,57,60,61,63,83]. In other words the mesh remains entirely transparent to the flow solver. The addition of mesh points can be carried out either by explicitly refining triangles, or alternatively by retriangulating the mesh to include the extra mesh points. This latter approach is particularly well suited to the Delaunay triangulation technique and has been demonstrated very effectively by Holmes et al. [42]. Figure 16 shows the mesh in a channel with a bump for a supersonic onset flow. A Delaunay triangulator was used with adaptive mesh enrichment, and the concentration of triangles which formed around the shockwaves is clearly evident. An example of the former approach has been provided very recently for a three dimensional triangulation by Periaire [63].

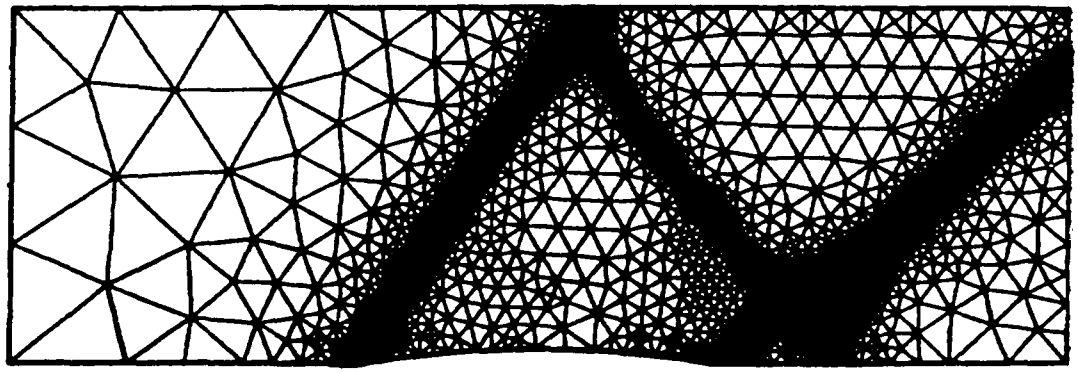


Fig. 16. Mesh Enrichment on a Triangular Mesh
(taken from Holmes et al., Ref. 42)

7. Whither Mesh Generation ?

It is a reckless though irresistible prospect to speculate on the future development of mesh generation methods. This is particularly apposite since we are now at a stage where the primary goal of treating complete aircraft configurations has been achieved. What remains is to bring these 'primitive' methods to a stage where it is possible to handle shapes of arbitrary generality with ease, without the need for special treatment, or undue interaction from the user whenever a different configuration arises. Of particular importance will be the need to achieve the most accurate solution possible for a given number of points and, if feasible, provide an estimate of the accuracy obtained. The development of efficient and robust adaptive meshing techniques is therefore a mandatory requirement. Not only should extra points be added to refine the mesh where needed, but it is also necessary to ensure that the shape and, in particular, aspect ratio of the cells remains acceptable. This may require the addition of points in some areas and the deletion of points in other regions. A deeper understanding of the dependence of the solution on mesh stretching and distortion is therefore required. Techniques that will either enrich or prune the mesh in selected regions must also be implemented as part of an adaptive mesh generation package. It is the author's opinion that it will be the unstructured meshes that best meet these stringent requirements for generality and adaptability.

However, much effort has been devoted to the development of structured mesh generation schemes and it is unlikely that they will be abandoned in the near future. Indeed, the future vigor of research into mesh generation can best be served by the competitive pursuit of both approaches.

References

1. F. Angrand, V. Billey, J. Periaux, C. Pouletty and J.P. Rosenblum, "2–D and 3–D Euler Computations around Lifting Bodies on Self Adapted Finite Element Meshes", Sixth International Symposium on Finite Element Methods in Flow Problems, Antibes, France, June 1986

2. B.G. Arlinger, "Calculation of Transonic Flow around Axisymmetric Inlets", AIAA 13th Aerospace Sciences Meetings, Pasadena, CA, AIAA Paper 75–80, January 1975.

3. B.G. Arlinger and W. Schmidt, "Design and Analysis of Slat Systems in Transonic Flow", 11th ICAS Congress, Lisbon, Portugal, September 1978.

4. E. Atta, "Component–Adaptive Grid Interfacing", AIAA 19th Aerospace Sciences Meeting, AIAA Paper 81–0382, January 1981.

5. T.J. Baker, "A Numerical Method to Compute Inviscid Flow around Axisymmetric Ducted Bodies", IUTAM Symposium Transsonicum II, Grottingen, September 1975, Proc. pub Springer Verlag, 1976.

6. T.J. Baker, "Mesh Generation by a Sequence of Transformations", Appl. Num. Math., Vol. 2, No. 1, December 1986.

7. T.J. Baker, "Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets", AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, AIAA Paper 87–1124, June 1987.

8. T.J. Baker and A. Jameson, "A Novel Finite Element Method for the Calculation of Inviscid Flow over a Complete Aircraft", Sixth International Symposium on Finite Element Methods in Flow Problems, Antibes, France, June 1986.

9. T.J. Baker, A. Jameson and R.E. Vermeland, "Three–Dimensional Euler Solutions with Grid Embedding", AIAA 23rd Aerospace Sciences Meeting, Reno, AIAA Paper 85–0121, January 1985.

10. J.A. Benek, J.L. Steger and F.C. Dougherty, "A Flexible Grid Embedding Technique with Application to the Euler Equations", AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, AIAA Paper 83–1944, July 1983.

11. J.A. Benek, P.G. Buning and J.L. Steger, "A 3–D Chimera Grid Embedding Technique", AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, Ohio, AIAA Paper 85–1523, June 1985.

12. J.A. Benek, T.L. Donegan and N.E. Suhs, "Extended Chimera Grid Embedding Scheme with Application to Viscous Flows", AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, AIAA Paper 87–1126, June 1987.

13. M.J. Berger, "On Conservation at Grid Interfaces", SIAM J. Num. Anal., Vol. 24, No. 5, pp. 967–984, October 1987.

14. M.J. Berger and A. Jameson, "Automatic Adaptive Grid Refinement for the Euler Equations", AIAA J., Vol. 23, No. 4, pp. 561–568, April, 1985.

15. J.W. Boerstoel, "Problem and Solution Formulations for the Generation of 3D

Block–Structured Grids", International Conference on Numerical Grid Generation in Computational Fluid Dynamics, Landshut, FRG, July 1986 (Proc. ed. J. Hauser and C. Taylor, pub. Pineridge Press).

16. J. W. Boerstoel, "Preliminary Design and Analysis of Procedures for the Numerical Generation of 3D Block–Structured Grids", NLR Tech. Report No. 86102L.

17. J.U. Brackbill and J.S. Saltzman, "Adaptive Zoning for Singular Problems in Two Dimensions", JCP, Vol. 46, pp 342–368, 1982.

18. M.O. Bristeau, O. Pironneau, R. Glowinski, J. Periaux, P. Perrier and G. Poirier, "On the Numerical Solution of Nonlinear Problems in Fluid Dynamics by Least Squares and Finite Element Methods (II). Application to Transonic Flow Simulations", Proc. 3rd International Conference on Finite Elements in Nonlinear Mechanics, FENOMECH 84, Stuttgart, 1984, edited by J. St. Dottsinis, North Holland 1985, pp. 363–394.

19. L.A. Carlson, "Transonic Airfoil Analysis and Design Using Cartesian Coordinates", AIAA 2nd Computational Fluid Dynamics Conference, Hartford, June 1975.

20. D.A. Caughey, "A Systematic Procedure for Generating useful Conformal Mappings", Int. J. Num. Meth. Eng., Vol. 12, pp. 1651–1657.

21. J.C. Cavendish, D.A. Field and W.H. Frey, "An Approach to Automatic Three–Dimensional Finite Element Mesh Generation", Int. J. Num. Meth. Eng., Vol. 21, pp. 329–347, 1985.

22. Z.J. Cendes, D. Shenton and H. Shahnasser, "Magnetic Field Computations using Delaunay Triangulations and Complementary Finite Element Methods", IEEE Trans. Magnetics, Vol. MAG–19, No. 6, pp. 2551–2554, Nov. 1983.

23. Z.J. Cendes and D.N. Shenton, "Adaptive Mesh Refinement in the Finite Element Computation of Magnetic Fields", IEEE Trans. Magnetics, Vol. MAG–21, No. 5, pp. 1811–1816, Sept. 1985.

24. D.R. Chapman, "Trends and Pacing Items in Computational Aerodynamics", Lecture Notes in Physics, Vol. 141, (ed. Reynolds and MacCormack), Springer–Verlag, pp. 1–12, 1980.

25. J.F. Dannenhoffer III and J.R. Baron, "Grid Adaptation for the 2–D Euler Equations", AIAA Paper 85–0484, January 1985.

26. J.F. Dannenhoffer III and J.R. Baron, "Adaptive Procedure for Steady State Solution of Hyperbolic Equations", AIAA 22nd Aerospace Sciences Meeting, Reno, AIAA Paper 84–0005, January 1984.

27. A. Eberle and W. Schwarz, "Grid Generation for an Advanced Fighter Aircraft", in Three Dimensional Grid Generation for Complex Configurations – Recent Progress (compiled by Steger and Thompson) AGARDograph No. 309.

28. P.R. Eiseman, "A Multi–Surface Method of Coordinate Generation", JCP, Vol. 33, pp. 118–150, 1979.

29. P.R. Eiseman, "Alternating Direction Adaptive Grid Generation", AIAA J., Vol. 23, No. 4, pp. 551–560, April 1985.

30. P.R. Eiseman and G. Erlebacher, "Grid Generation for the Solution of Partial Differential Equations", ICASE Report No. 87–57.

31. L.E. Eriksson, "Practical Three–Dimensional Mesh Generation Using Transfinite Interpolation", Lecture Series Notes 1983–04, von Karman Institute for Fluid Dynamics, Brussels, Belgium, 1983.

32. L.E. Eriksson, "Generation of Boundary–Conforming Grids Around Wing–Body Configurations using Transfinite Interpolation", AIAA J., Vol. 20, No. 10, pp. 1313–1320, 1982.

33. L.E. Eriksson, R.E. Smith, M.R. Wiese and N. Farr, "Grid Generation and Inviscid Flow Computation about Cranked–Winged Airplane Geometries", AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, AIAA Paper 87–1125, June 1987.

34. J. Flores, S.G. Reznick, T.L. Holst and K. Gundy, "Transonic Navier–Stokes Solutions for a Fighter–Like Configuration", AIAA 25th Aerospace Sciences Meeting, Reno, NV, AIAA Paper 87–0032, January 1987.

35. W. Fritz, W. Haase and W. Seibert, "Mesh Generation for Industrial Application of Euler and Navier Stokes Solvers", in Three Dimensional Grid Generation for Complex Configurations – Recent Progress (compiled by Steger and Thompson) AGARDograph No. 309, to appear 1988.

36. M.B. Giles, "Accuracy of Node–Based Solutions on Irregular Meshes", submitted to the 11th International Conference on Numerical Methods in Fluid Dynamics, Williamsburg, VA, June 1988.

37. M.B. Giles and W.T. Thompkins, Jr., "Propagation and Stability of Wavelike Solutions of Finite Difference Equations with Variable Coefficients", JCP, Vol. 58, pp. 349–360, 1985.

38. W.J. Gordon, "Blending–Function Methods of Bivariate and Multivariable Interpolation and Approximation", SIAM J. Num. Anal., Vol. 8, No. 1, pp 158–177, 1971.

39. P.A. Gnoffo, "A Finite–Volume, Adaptive Grid Algorithm Applied to Planetary Entry Flowfields", AIAA J., Vol. 21, No. 9, pp. 1249–1254, Sept. 1983.

40. J.D. Hoffman, "Relationship between the Truncation Errors of Centered Finite–Difference Approximations on Uniform and Nonuniform Meshes", JCP, Vol. 46, pp 469–474, 1982.

41. D.G. Holmes and S.H. Lamson, "Adaptive Triangular Meshes for Compressible Flow Solutions", International Conference on Numerical Grid Generation in Computational Fluid Dynamics, Landshut, FRG, July 1986 (Proc. ed. J. Hauser and C. Taylor, pub. Pineridge Press).

42. D.G. Holmes, S.H. Lamson and S.D. Connell, "Quasi–3D Solutions for Transonic, Inviscid Flows by Adaptive Triangulation", to be presented at ASME Gas Turbine Meeting, June 1988.

43. D.C. Ives, "A Modern Look at Conformal Mapping including Doubly Connected Regions", AIAA 8th Fluid and Plasma Dynamics Conference, Hartford, AIAA Paper

75–842, June 1975.

44. A. Jameson and T.J. Baker, "Solution of the Euler Equations for Complex Configurations", AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, AIAA Paper 83–1929, July 1983.

45. A. Jameson and D.A. Caughey, "A Finite Volume Method for Transonic Potential Flow Calculations", AIAA 3rd Computational Fluid Dynamics Conference, Albuquerque, NM, June 1977.

46. A. Jameson and T.J. Baker, "Multigrid Solution of the Euler Equations for Aircraft Configurations", AIAA 22nd Aerospace Sciences Meeting, Reno, AIAA Paper 84–0093, January 1984.

47. A. Jameson, T.J. Baker and N.P. Weatherill, "Calculation of Inviscid Transonic Flow over a Complete Aircraft", AIAA 24th Aerospace Sciences Meeting, Reno, NV, AIAA Paper 86–0103, January 1986.

48. A. Jameson and T.J. Baker, "Improvements to the Aircraft Euler Method", AIAA 25th Aerospace Sciences Meeting, Reno, AIAA Paper 87–0452, January 1987.

49. P. Kutler, "A Perspective of Theoretical and Applied Computational Fluid Dynamics", AIAA J., Vol. 23, No. 3, pp. 328–341, 1985.

50. K.D. Lee, M. Huang, N.J. Yu and P.E. Rubbert, "Grid Generation for General Three–Dimensional Configurations", Proc. NASA Langley Workshop on Numerical Grid Generation Techniques, (ed. Smith) Oct. 1980.

51. S.H. Lo, "A New Mesh Generation Scheme for Arbitrary Planar Domains", Int. J. Num. Meth. Eng., Vol. 21, pp. 1403–1426, 1985.

52. R. Lohner, "Generation of Three–Dimensional Unstructured Grids by the Advancing–Front Method", AIAA 26th Aerospace Sciences Meeting, Reno, AIAA Paper 88–0515, January 1988.

53. R. Lohner, K. Morgan, J. Peraire and O.C. Zienkiewicz, "Finite Element Methods for High Speed Flows", AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, Ohio, AIAA Paper 85–1531, July 1985.

54. R. Lohner and K. Morgan, "Improved Adaptive Refinement Strategies for Finite Element Aerodynamic Computations", AIAA 24th Aerospace Sciences Meeting, Reno, AIAA Paper 86–0499, January 1986.

55. D. Mavriplis and A. Jameson, "Multigrid Solution of the Two–Dimensional Euler Equations on Unstructured Triangular Meshes", AIAA 25th Aerospace Sciences Meeting, Reno, AIAA Paper 87–0353, January 1987.

56. J.F. Middlecoff and P.D. Thomas, "Direct Control of the Grid Point Distribution im Meshes Generated by Elliptic Equations", AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, VA, AIAA Paper 79–1462, July 1979.

57. K. Morgan, J. Peraire, R.R. Thareja and J.R. Stewart, "An Adaptive Finite Element Scheme for the Euler and Navier–Stokes Equations", AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 1987.

58. K. Nakahashi and G.S. Deiwert, "Self–Adaptive–Grid Method with Application to Airfoil Flow", AIAA J., Vol. 25, No. 4, pp. 513–520, April 1987.

59. K. Nakahashi and S. Obayashi, "Viscous Flow Comptuations Using a Composite Grid", AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, AIAA Paper 87–1128, June 1987.

60. B. Palmerio and A. Dervieux, "Application of a FEM Moving Node Adaptive Method to Accurate Shock Capturing", International Conference on Numerical Grid Generation in Computational Fluid Dynamics, Landshut, FRG, July 1986 (Proc. ed. J. Hauser and C. Taylor, pub. Pineridge Press).

61. J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz, "Adaptive Remeshing for Compressible Flow Computations", JCP, Vol. 72, No. 2, pp. 449–466, 1987.

62. J. Peraire and K. Morgan, "A General Triangular Mesh Generator", to appear in Int. J. Num. Meth. Eng., 1988.

63. J. Peraire, J. Peiro, L. Formaggia, K. Morgan and O.C. Zienkiewicz, "Finite Element Euler Computations in Three Dimensions", AIAA 26th Aerospace Sciences Meeting, Reno, AIAA Paper 88–0032, January 1988.

64. J. Pike, "Grid Adaptive Algorithms for the Solution of the Euler Equations on Irregular Grids", JCP, Vol. 71, pp. 194–223, 1987.

65. T.A. Reyhner, "Transonic Potential Flow Computation about Three–Dimensional Inlets, Ducts and Bodies", AIAA 13th Fluid and Plasma Dynamics Conference, Snowmass, AIAA Paper 80–1364, July 1980.

66. A.W. Rizzi and M. Inouye, "Time Split Finite–Volume Method for Three–Dimensional Blunt Body Flow", AIAA J., Vol. 11, pp. 1478–1485, 1973.

67. P.L. Roe, "Error Estimates for Cell–Vertex Solutions of the Compressible Euler Equations", ICASE Report No. 87–6.

68. K. Sawada and S. Takanashi, "A Numerical Investigation on Wing/Nacelle Interferences of USB Configuration", AIAA 25th Aerospace Sciences Meeting, Reno, NV, AIAA Paper 87–0455, January 1987.

69. W. Seibert, "An Approach to the Interactive Generation of Block–Structured Volume Grids Using Computer Graphics Devices", International Conference on Numerical Grid Generation in Computational Fluid Dynamics, Landshut, FRG, July 1986 (Proc. ed. J. Hauser and C. Taylor, pub. Pineridge Press).

70. C.C.L. Sells, "Plane Subcritical Flow Past a Lifting Aerofoil", Proc. Roy. Soc., London, Vol. 308A, pp. 377–401, 1968.

71. J. Shaw, C.R. Forsey, N.P. Weatherill and K.E. Rose, "A Block Structured Mesh Generation Technique for Aerodynamic Geometries", International Conference on Numerical Grid Generation in Computational Fluid Dynamics, Landshut, FRG, July 1986(Proc. ed. J. Hauser and C. Taylor, pub. Pineridge Press).

72. A. Shmilovich and D.A. Caughey, "Grid Generation for Wing–Tail–Fuselage Combinations", ASME Mini–Symposium on Advances in Grid Generation, Houston,

Texas, June 1983.

73. R.E. Smith (Editor), Numerical Grid Generation Techniques, Proc. Conference held at NASA Langley, October 1980.

74. R.E. Smith, "Three–Dimensional Algebraic Grid Generation", AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, AIAA Paper 83–1904, July 1983.

75. R.L. Sorenson, "Elliptic Generation of Compressible Three–Dimensional Grids about Realistic Aircraft", International Conference on Numerical Grid Generation in Computational Fluid Dynamics, Landshut, FRG, July 1986 (Proc. ed. J. Hauser and C. Taylor, pub. Pineridge Press).

76. R.L. Sorenson, "Three–Dimensional Elliptic Grid Generation for an F–16", in Three Dimensional Grid Generation for Complex Configurations – Recent Progress (compiled by Steger and Thompson) AGARDograph No. 309.

77. J.C. South and A. Jameson, "Relaxation Solutions for Inviscid Axisymmetric Transonic Flow over Blunt or Pointed Bodies", AIAA 1st Computational Fluid Dynamics Conference, Palm Springs, CA, pp. 8–17, 1973.

78. J.L. Steger, "Implicit Finite–Difference Simulation of Flow about Arbitrary Two–Dimensional Geometries", AIAA J., Vol. 16, No. 7, pp. 679–686, July 1978.

79. J.L. Steger and R.L. Sorenson, "Automatic Mesh–Point Clustering near a Boundary in Grid Generation with Elliptic Partial Differential Equations", JCP, Vol. 33, No. 3, pp. 405–410, 1979.

80. J.L. Steger and R.L. Sorenson, "Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates", Proc. NASA Langley Workshop on Numerical Grid Generation Techniques (ed. Smith), October 1980.

81. J.P. Steinbrenner, S.L. Karmen and J.R. Chawner, "Generation of Multiple Block Grids for Arbitrary 3D Geometries", in Three Dimensional Grid Generation for Complex Configurations – Recent Progress (compiled by Steger and Thompson) AGARDograph No. 309.

82. J. Steinhoff, "Blending Methods for Grid Generation", JCP, Vol. 65, pp. 370–385, 1986.

83. B Stoufflet, J. Periaux, F. Fezoui and A. Dervieux, "Numerical Simulation of 3–D Hypersonic Euler Flows Around Space Vehicles using Adapted Finite Elements", AIAA Paper 87–0560.

84. W.C. Thacker, "A Brief Review of Techniques for Generating Irregular Computational Grids", Int. J. Num. Meth. Eng., Vol. 15, pp. 1335–1341, 1980.

85. J.F. Thompson, ed., Numerical Grid Generation, Nashville, Tennessee, April 1982, Proc. pub. in Appl. Math. Comp., Vol. 10&11, 1982.

86. J.F. Thompson, "Grid Generation Techniques in Computational Fluid Dynamics", AIAA J., Vol. 22, No. 11, pp. 1505–1523, Nov. 1984.

87. J.F. Thompson, "A Composite Grid Generation Code for General 3–D Regions", AIAA

25th Aerospace Sciences Meeting, Reno, AIAA Paper 87–0275, January 1987.

88. J.F. Thompson, F.C. Thomas and C.W. Mastin, "Automatic Numerical Generation of Body Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two–Dimensional Bodies", JCP, Vol. 15, pp. 299–319, 1974.

89. J.F. Thompson, Z.U.A. Warsi and C.W. Martin, "Boundary–Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations – A Review", JCP, Vol. 47, pp 1–108, 1982.

90. J.F. Thompson, Z.U.A. Warsi and C.W. Mastin, Numerical Grid Generation–Foundations and Applications, pub. North–Holland, 1985.

91. L.N. Trefethen, "Group Velocity Interpretation of the Stability Theory of Gustafsson, Kreiss and Sundstrom", JCP, Vol. 49, pp 199–217, 1983.

92. E. Turkel, "Accuracy of Schemes with Nonuniform Meshes for Compressible Fluid Flows", Appl. Num. Math., Vol. 2, pp 529–550, 1986.

93. R. Vichnevetsky, "Invariance Theorems Concerning Reflection at Numerical Boundaries", JCP, Vol. 63, No. 2, pp 268–282, 1986.

94. R. Vichnevetsky, "Wave Propagation Analysis of Difference Schemes for Hyperbolic Equations: A Review", Int. J. Num. Math. Fluids, Vol. 7, pp 409–452, 1987.

95. R. Vichnevetsky, "Wave Propagation and Reflection in Irregular Grids for Hyperbolic Equations", Appl. Num. Math., Vol. 3, pp. 133–166, 1987.

96. Y. Vigneron and T. Lejal, "Calculation of Transonic Flow around an Aircraft Configuration with Motorized Nacelle", ICAS Paper 84–2.10.2.

97. Y. Vigneron, T. Lejal and R. Collercandy, "Computation of Transonic Flows with Nacelle Simulation", Sixth International Symposium on Finite Element Methods in Flow Problems, Antibes, France, June 1986.

98. M. Vinokur, "On One–Dimensional Stretching Functions for Finite–Difference Calculations", JCP, Vol. 50, pp. 215–234, 1983.

99. N.P. Weatherill, "The Generation of Unstructured Grids Using Dirichlet Tessellations", Princeton Univ. MAE Report, No. 1715, July 1985.

100. N.P. Weatherill, "A Method for Generating Irregular Computational Grids in Multiply Connected Planar Domains", to appear in Int. J. Num. Meth. in Fluids, Vol. 8, 1988.

101. N.P. Weatherill and C.R. Forsey, "Grid Generation and Flow Calculations for Aircraft Geometries", J. Aircraft, Vol. 22, pp. 855–860, 1985.

102. N.P. Weatherill and J.A. Shaw, "Component Adaptive Grid Generation for Aircraft Configurations", in Three Dimensional Grid Generation for Complex Configurations – Recent Progress (Compiled by Steger and Thompson) AGARDograph No. 309,

103. B. Wedan and J.C. South, Jr., "A Method for Solving the Transonic Full–Potential Equation for General Configurations", AIAA 6th Computational Fluid Dynamics

Conference, Danvers, MA, AIAA Paper 83–1889, July 1983.

104. A.M. Winslow, "Numerical Solution of the Quasilinear Poisson Equation in a Non–uniform Triangle Mesh", JCP, Vol. 1, No. 2, pp 149–172, 1967.

105. N.J. Yu, "Grid Generation and Transonic Flow Calculations for Three–Dimensional Configurations", AIAA 13th Fluid and Plasma Dynamics Conference, Snowmass, Colorado, AIAA Paper 80–1391, July 1980.

106. N.J. Yu, K. Kusurose, H.C. Chen and D.M. Sommerfield, "Flow Simulations for a Complex Airplane Configuration using Euler Equations", AIAA 25th Aerospace Sciences Meeting, Reno, NV, AIAA Paper 87–0454, January 1987.