Numerical Grid Generation in
Computational Fluid Dynamics and Related Fields
A.S.-Arcilla, J. Häuser, P.R. Eiseman, J.F. Thompson (editors)

Generation and Adaptive Alteration of Unstructured Three-Dimensional Meshes

Todd J. Mitty
Timothy J. Baker
Antony Jameson

Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, NJ 08544 USA

Abstract

Adaptive mesh refinement on unstructured meshes in three-dimensions is applied to obtain a sharp resolution of oblique shock waves. Meshes are generated through the application of Bowyer's Algorithm to yield a Delaunay tessellation of the space. The Euler Equations are solved by a finite-volume time-stepping scheme to reach a steady state.

1 Introduction

The accuracy of computational solutions to analytical equations is strongly influenced by the discretization of the space in which a solution is sought. As the spacing between the points which define the computational space is reduced, the discrete domain approaches a continuum. Since each point is associated with a flow state, resolution of "interesting" phenomena is dictated by point placement and concentration. Thus, the ability to determine where phenomena are occurring, what type of computational modifications are needed and how to implement them, is a paramount concern for effective computations.

In theory, the introduction of a highly dense distribution of points throughout a computational domain will yield a more accurate answer than a coarse distribution. However, limitations in computer processing speed and accessible memory prohibit such a scenario. An appropriate alternative would be to increase the accuracy of the computations where needed. If the method of discrete approximation remains unchanged within the computational domain, then altering the concentration of points in critical regions by either adding or shifting points is a possible resolution of this dilemma.

There are essentially two means by which a numerical calculation can be altered to allow for an improved solution. One is to alter the computational mesh; either by changing the number of points (h-method) or by altering the distribution of points (r-method) in a prescribed zone. The other alternative is to improve the accuracy of the numerical scheme (p-method). The p-method may yield favorable results. However, it is quite cumbersome to implement for a general class of problems — particularly if the type of equations being solved varies between problems. The r-method is attractive because it allows one to fix the total number of points (i.e., a fixed amount of requisite computer memory and processing time per cycle) while concentrating points in critical regions [1]. A major drawback of this method is that not only must regions for increased point density be determined, but it must also be determined where imposing a less dense distribution will not corrupt the solution. In addition, the procedures to generate these distributions may be computationally expensive (especially in three-dimensions) as well as problematic in that they can produce computational elements which possess a "poor" aspect-ratio. Finally, there is the h-method which can *locally* improve the accuracy of *any* well-posed numerical scheme. [2]

All alteration methods can have two modes of operation — either to refine or de-refine a given region [3]. The advantage of de-refinement is to reduce the number of necessary points, thereby reducing the amount of required storage and increasing the rate of convergence. However, as previously discussed, a great deal of caution must be taken when removing points since this makes the computational domain more coarse — allowing for the possibility of decreased accuracy.

The aforementioned considerations are well recognized, and an appreciation of them is essential for successful computations. In the past, researchers involved with adaptive techniques have implemented these notions in various roles. Initially, adapted computational meshes were generated with *a priori* knowledge of the location and nature of occurring phenomena. Later, schemes were devised which re-generated a mesh based upon previously calculated results — adaptively altering the mesh.

To facilitate many of the aspirations which researchers in computational methods have, unstructured meshing techniques have become the canvas upon which a great deal of research emphasis has been placed [4]. The reasons why an unstructured philosophy is attractive are numerous and varied. Although computational techniques performed under the auspice of unstructured meshes have their limitations, their suitability for adaptive meshing seems to be inherent since unstructered meshing permits the insertion (or removal) of points anywhere within the computational domain of complex multi-component surfaces. As such, unstructured meshes are the tools used in this work.

The impetus for this research stems from the computational solution of a supersonic shock wave/boundary layer interaction flow for which there is a great deal of experimental data. Basically, the flow results from a two-dimensional shock wave (generated from a wedge normal to a flat plate) interacting with a boundary layer in the third dimension. (See Settles and Dolling [5] for a comprehensive overview of the problem). Although the geometry is simple, its resultant flow is complex. Some of the characteristic features which are present include a vortex, multiple intersecting shock waves, and high compression zones. Since the location of these phenomena is dependent on upstream conditions, it is difficult to generate an appropriate mesh consistently and *a priori*. By using adaptive alteration, enhanced meshing can be applied directly at locations where these phenomena are occurring, resulting in sharp capture of them. Although solutions of this flow have been computed in the past, significant improvements would be achieved with the application of mesh adaption.

A geometry identical to the shock wave/boundary layer interaction problem is used.

However, for illustrative purposes and initial testing, only a two-dimensional supersonic flow with no boundary layer is considered.

2 Discretization and Tessellation

The geometric model is a basic one — comprised of a ramp at a given inclination (i.e., a wedge) and bounding planes. The generation of these bounding surfaces and points between them is obtained through simple algebraic expressions. From this array of points, a tessellation of the space must be imposed to provide the mesh connectivity. Any connectivity will do as long as computations conducted over it are capable of recognizing the features (e.g., cell geometry) of the governing mesh. However, if one desires a robust and efficient computational solver, then a tessellation of the space by a single cell type is beneficial.

Tetrahedral tessellation of the computational domain is the convention of most unstructured meshings, and it was exploited in this research as well. Formation of a set of tetrahedra can be performed in any of a number of ways. Two quite favorable methods are the so-called "advancing-front" [6] and Delaunay [7] tessellations. Both of these approaches have their advantages, but Delaunay tessellation is particularly useful for refinement methods because of its ability to accept points in arbitrary locations at any time.

The Delaunay tessellation of the computational space is obtained through application of Bowyer's Algorithm which may be stated as follows from [7]:

Given a set V_n of n points in k-space, the Delaunay tessellation is the unique tessellation of V_n such that no point $P_i \in V_n$ lies inside the circumsphere of any k-simplex.

In this work, the implementation of Bowyer's Algorithm on a computer is similar to the methods used in [7] and [8]. A bounding-box comprised of eight corner points which encloses all points in the domain is constructed. This box is then divided into Delaunay tetrahedra. Points to be included in the computational domain are then injected one at a time, and any tetrahedral subspace breeched according to Bowyer's rule is updated to make that subspace Delaunay once again. To enhance the restrictions imposed by irregular data storage, an octree data structure is utilized for quick and efficient searches. After all desired points have been inserted, tetrahedra not belonging to the the computational domain (i.e., those between the bounding box and computational domain) are removed.

In an effort to retain surface integrity, a system incorporating the notion of protected tetrahedra is enforced. The first group of points to be inserted into the bounding box are the initial surface points. Once these tetrahedra are formed, a sweep of the space is made and any tetrahedra which have three forming points on a surface and the fourth on the bounding box are flagged. All of the remaining points are then injected, and if an injected point breeches a flagged tetrahedron, the point is rejected unless it is compatible with the surface face(s) of that tetrahedron.

Currently, the adaptively altered meshes used are produced from a Delaunay remeshing of the whole space based upon a modified set of points. However, it is a simple task to alter this procedure so that updated meshes are obtained directly from their predecessor without the need for redundant and computationally costly re-meshings. In fact, this is the essence of why a Delaunay tessellation is so amenable to adaptive techniques.

3 Mathematical Model and Numerical Description

The adaptive alteration method of this paper is applied to the solution of the conservation equations of fluid mechanics which govern the flow of compressible fluids. In there inviscid form they are referred to as the Euler Equations and may be drafted in compact integral form for a domain Ω with bounding surfaces $\partial\Omega$ as follows:

$$\frac{\partial}{\partial t}\iiint_{\Omega} \mathbf{w}\, d\Omega + \iint_{\partial\Omega} \mathbf{F}\,\mathbf{n}\, d\partial\Omega = 0, \tag{1}$$

where $\mathbf{w}$ is a vector of dependent variables; $\mathbf{F}$ is a flux matrix; and $\mathbf{n}$ is a surface unit normal. All of these are defined as:

$$\mathbf{w} = \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_o \end{array} \right\}, \quad \mathbf{F} = \left[\begin{array}{ccc} \rho u & \rho v & \rho w \\ \rho u^2 + p & \rho v u & \rho w u \\ \rho u v & \rho v^2 + p & \rho w v \\ \rho u w & \rho v w & \rho w^2 + p \\ \rho u h_o & \rho v h_o & \rho w h_o \end{array} \right], \quad \mathbf{n} = \left\{ \begin{array}{c} n_x \\ n_y \\ n_z \end{array} \right\},$$

where the primative variables ρ, p, e_o, h_o, u, v, and w are the thermodynamic density, pressure, total internal energy, and total enthalpy, and three Cartesian velocity components respectively. Furthermore, we restrict our analysis to that of a perfect gas, and are obliged with an equation of state of the form (γ is the ratio of specific heats)

$$e_o = \frac{p}{(\gamma - 1)\rho} + \frac{1}{2}\left(u^2 + v^2 + w^2\right). \tag{2}$$

Equations 1 and 2 form a complete set which is used to solve for $\mathbf{w}$. However, these equations apply to a continuum and must be translated into a form appropriate for use in a discrete space. For the sake of brevity, only the resulting equations of the discretization will be stated. For a more complete derivation, see Jameson and Baker [8].

It is assumed that the computational domain is spanned by *tetrahedra* and that the dependent variables are stored at the *nodes* of the tetrahedra. Applying Galerkin's Method to Equation 1 using a piecewise linear test function, and after some algebraic manipulation, a discrete approximation is arrived at. From [8], it may be written as:

$$\frac{d}{dt}\left[\left(\sum_k \Omega_k\right)\mathbf{w}_i\right] + \sum_k \left(\tilde{\mathbf{F}}\mathbf{n}\partial\Omega\right)_k = 0, \tag{3}$$

$$\tilde{\mathbf{F}} = \frac{1}{3}\left(\mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3\right).$$

The sums in Equation 3 are over all tetrahedra which meet at a common node i to form a control volume surrounding that node. $\tilde{\mathbf{F}}$ is the average of the flux matrix from the three nodes which form the face of the tetrahedra on the bounding control surface with $\mathbf{n}$ as the normal of that plane. In order to aid conceptualization, Equation 3 is directly interpretable as a finite volume approximation of the conservation laws (Equation 1). It may be re-written as a first-order ordinary differential equation in time to yield

$$\frac{d\mathbf{w}_i}{dt} + \mathcal{R}(\mathbf{w}_i) = 0, \tag{4}$$

where $\mathcal{R}(\mathbf{w}_i)$ represents a numerical flux-residual at node i.

Numerical dissipation must be added to this discrete approximation for the purpose of removing undamped and/or lightly damped error modes, as well as to prevent solution oscillation near large gradients (e.g., shock waves) for more precise resolution. Reference [8] presents the form of these dissipative terms which will be stated here as a dissipative residual $\mathcal{D}(\mathbf{w}_i)$, and is lumped with $\mathcal{R}(\mathbf{w}_i)$.

A Runge-Kutta multi-stage time-stepping scheme is used to integrate Equation 4 (with dissipation included) forward in time. Only steady flows are considered here, and it follows that the unsteady term in Equation 4 should approach zero in the limit of a steady state. Since a steady state is sought, the time accuracy of the integration is not of concern and convergence acceleration techniques such as residual averaging and enthalpy damping can be applied. In this work, a five-stage time-stepping scheme was employed with no residual averaging and no enthalpy damping.

4 Adaptive Technique

4.1 Criteria for Adaption

In order to *adaptively* alter a mesh, an ability to determine which parts of the computational field require special treatment is necessary. If an accurate understanding of the local discretization error were available, then this would serve as an ideal monitor for alteration. In fact, it would also allow for a more justified application of de-refinement techniques since the spacing between points could be maximized according to restrictions dictated by the order of the error. Unfortunately, it is not always a simple matter to determine what the discretization error of a given numerical scheme is. This is especially true of finite-volume/finite-element approaches.

Although the discretization error describes the deviation from the analytical solution of a set of equations to be solved, other measures of deviation can be used as close approximations to this error. Qualitatively, these error indicators, or *sensors*, serve to signal where interesting phenomena may be occurring, thereby indicating a need for alteration. Since alteration is occurring according to results from computations, it is referred to as *adaptive alteration*.

There are many types of sensors which can be used [9, 3, 10]. In fact, those sensors which are used for shock capturing can also be incorporated to interrogate the computational domain for adaption determination. In this work, the simple sensor as given by Holmes [9] which examines the pressure of a node of a cell with respect to the average pressure in that cell is used. If the difference of the node's pressure to the average pressure is greater than some percentage change, then that cell is marked for adaption.

4.2 Mesh Alteration

There are several ways in which a cell can be altered for adaption purposes. Ultimately, the goal is to generate cells of an appropriate size such that they follow suit with the type of improvement to the mesh which is being sought. Identifying the existing set of n-points and its tessellation as V_n and T_n, the set of adaptively obtained points $\hat{V}_m$ should be such that modified tessellation T_{n+m} has cells with a "good" aspect-ratio (e.g., no long thin tetrahedra). Thus, even if m is large, if the resulting mesh quality is poor, the implemented procedure will prove to have been somewhat futile.

It is difficult to state explicitly conditions which must be adhered to for successful modified tessellations. This is particularly true for tessellations such as Delaunay, where the tessellation of an introduced point is dependent upon the pre-determined tessellation surrounding that point. However, in general, if $\hat{V}_m$ is chosen such that V_{n+m} has a distribution of points which gradually changes density from point-to-point, then an acceptable tessellation is usually achieved.

In this work, the following types of cell divisions were employed for those cells marked for alteration:

- Each cell had a point placed at its centroid (the arithmetic average of the positions of the forming nodes).
- Surface cells had an additional point placed at the centroid of the surface face(s).
- Cells containing a boundary edge(s) had a point placed at the midpoint of that edge.

This type of division was necessary because it was found that refinement of the boundary was essential for proper point distributions and successful convergence to a solution. Divisions based upon the circumcenter were also tried and were successful. It was found that depending upon the nature of the cell being altered, one method would be more advantageous than the other. Thus, the form of the initial mesh can have a significant effect on subsequent alterations. At present, no consistent policy exists for obtaining the most appropriate division, and further research on this topic is required.

5 Results

A supersonic Mach 2 flow over a 20° wedge was calculated. Imposed boundary conditions were as follows: *i*) Side bounding surfaces, wedge surface, and floor surface ahead of the wedge were treated as zero normal flux walls (with slip); *ii*) The inlet surface was set at freestream Mach 2 values; and *iii*) The remaining two planes ("top-surface" and "exit-surface") were set according to theoretical oblique shock wave jump conditions and

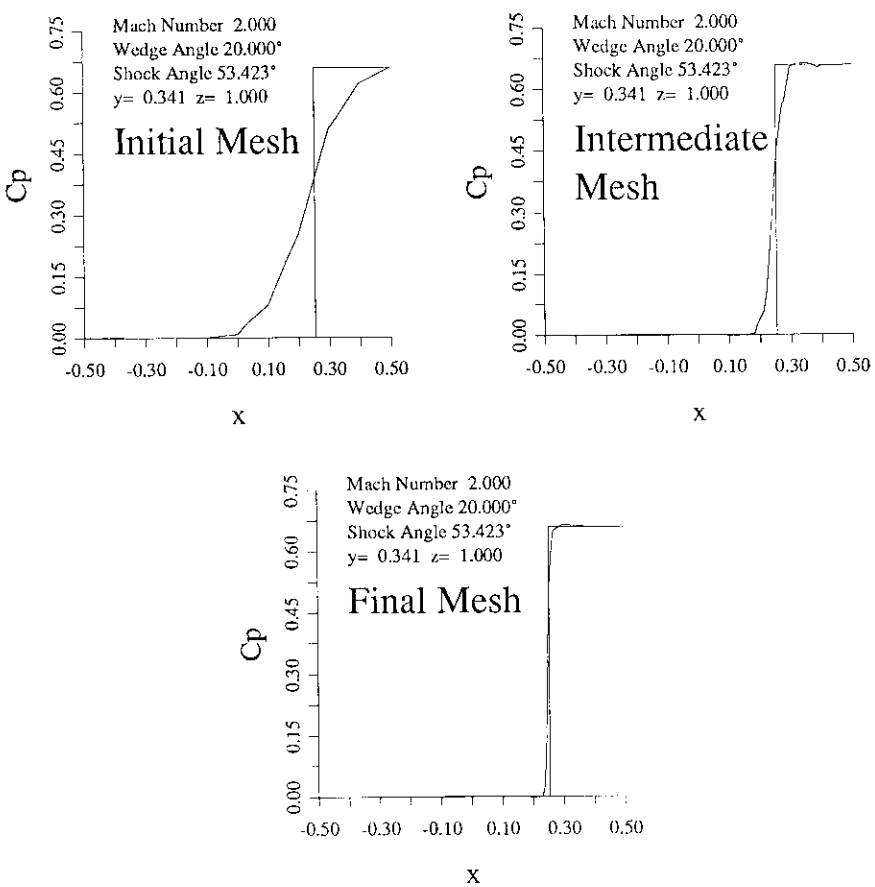

Figure 1: Pressure Coefficient Distribution at Various Adaption Stages

shock inclination. Condition *iii* was imposed as a means of decoupling boundary condition concerns from the research. The initial condition was uniform freestream values everywhere.

Six mesh alterations were calculated. Results obtained from the initial mesh, an intermediate altered mesh, and the final altered mesh are presented in Figures 1 and 2. In Figure 1, values are given along a line on a side-surface through the middle of the exit-surface. Figure 2 is the side-surface images of Figure 1 with a correspondence between levels of alteration. The evidence is compelling that adaptive alteration significantly enhances the flow resolution.

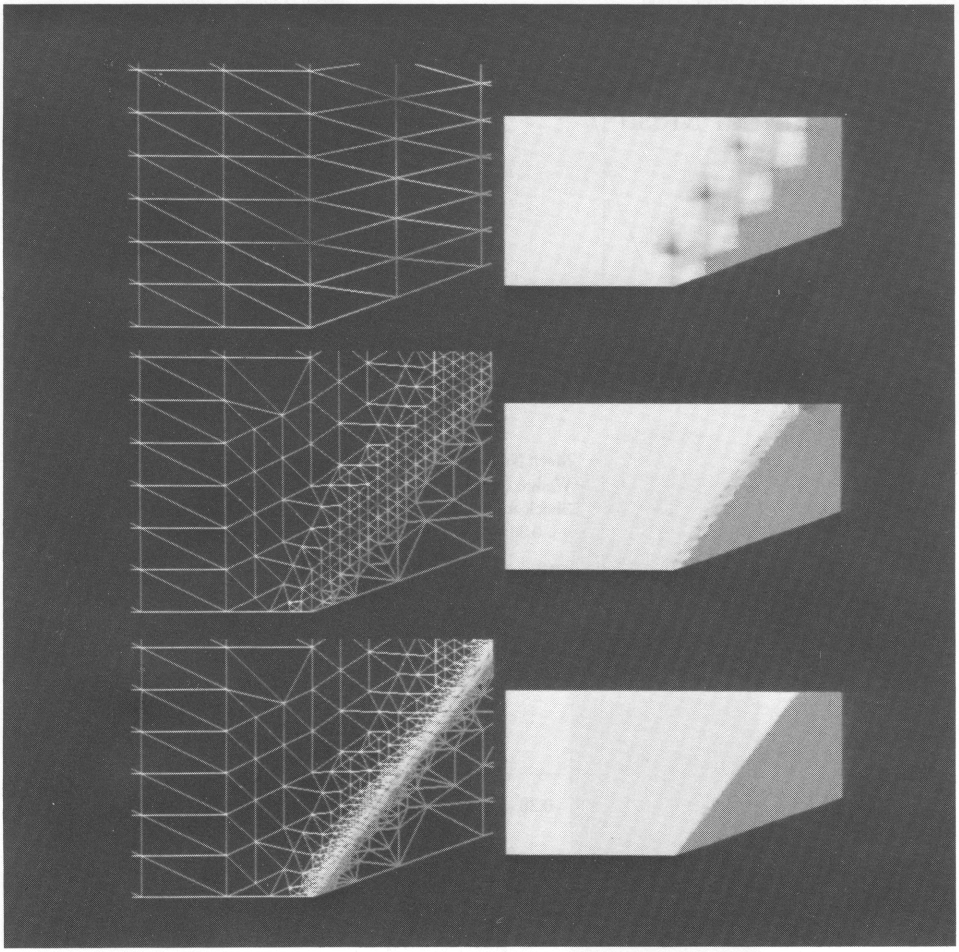

Figure 2: Bounding Plane Mesh and Pressure Field at Various Adaption Stages

---

SEE SPECIAL SECTION AT THE END OF THIS VOLUME FOR COLOUR REPRODUCTION

6 Future Work

It is clear that further applications of adaptive alteration should be explored, and it is certain that its methods will evolve and improve. Eventually, adaptively calculated solutions to the three-dimensional shock wave/boundary layer interaction problem are desired. However, it is presently felt that a concerted effort should be made for the determination of robust criteria which specify the character of the initial mesh and the manner in which this should be modified to yield an appropriate point distribution for an advantageous resultant tessellation.

Acknowledgements

The authors wish to thank Dr. Raul Mendez and the Institute for Supercomputing Research in Tokyo, Japan for supporting the first author in a research internship. A great deal of initial investigation for this work was accomplished at the Institute.

References

[1] Palmerio, B. and Dervieux, A., "2-D and 3-D Unstructured Mesh Adaption Relying on Physical Analogy," in *Numerical Grid Generation in Computational Fluid Mechanics '88*, ed. Sengupta, et al. Pineridge Press Ltd., 1988.

[2] Cheng, J.H., Finnigan, P.M., Hathaway, A.F., Kela, A. and Schroeder, W.J., "Quadtree/Octree Meshing with Adaptive Analysis," in *Numerical Grid Generation in Computational Fluid Mechanics '88*, ed. Sengupta, et al. Pineridge Press Ltd., 1988.

[3] Vemaganti, G.R., Thornton, E.A. and Wieting, A.R., "A Structured and Unstructured Remeshing Method for High Speed Flows," AIAA-90-0401, AIAA 28th Aerospace Sciences Meeting, Reno, Nevada, January 8–11, 1990.

[4] Baker, T.J., "Development and Trends in Three-Dimensional Mesh Generation," *Applied Numerical Mathematics*, Vol. 5, No. 4, July 1989, pp. 275–304.

[5] Settles, G.S. and Dolling, D.S., "Swept Shock/Boundary-Layer Interactions: Tutorial and Update," AIAA-90-0375, AIAA 28th Aerospace Sciences Meeting, Reno, Nevada, January 8–11, 1990.

[6] Löhner, R., "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method," AIAA-88-0515, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 11–14, 1988.

[7] Baker, T.J., "Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation," *Engineering with Computers*, No. 5, 1989, pp. 161–175.

[8] Jameson, A. and Baker, T.J., "Improvements to the Aircraft Euler Method," AIAA-87-0452, AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 12–15, 1987.

[9] Holmes, D.G. and Lamson, S.H., "Adaptive Triangular Meshes for Compressible Flow Solutions," in *Numerical Grid Generation in Computational Fluid Mechanics*, ed. Häuser and Taylor, Pineridge Press Ltd., 1986.

[10] Löhner, R. and Baum, J.D., "Numerical Simulation of Shock Interaction with Complex Geometry Three-Dimensional Structures Using a New Adaptive H-Refinement Scheme on Unstructured Grids," AIAA-90-0700, AIAA 28th Aerospace Sciences Meeting, Reno, Nevada, January 8–11, 1990.