Three-Dimensional Space-Marching Algorithm on Unstructured Grids

W. D. McGrory*
AeroSoft Inc., Blacksburg, Virginia 24062
R. W. Walters†
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061
and
R. Löhner‡
George Washington University, Washington, D.C. 20052

A three-dimensional space-marching algorithm using an unstructured discretization is proposed. The method utilizes a two-dimensional unstructured grid generator to construct grids in a crossflow plane while maintaining structure in the marching direction. The spatial discretization is obtained by applying a characteristic-based, upwind, finite volume scheme for the solution of the Euler equations. Solutions are presented for several different geometries comparing the results with existing numerical techniques and experiment.

Introduction

THE current availability of high-speed computers has enabled the solution of complex three-dimensional fluid flows. Structured algorithms for solutions in the supersonic and hypersonic regime are particularly efficient.[1-3] However, current structured algorithms are restricted in their geometric flexibility. There has been recent interest in the development of finite element and finite volume algorithms in the field of computational fluid dynamics (CFD). These algorithms are not restricted in their geometric flexibility, but the lack of inherent connectivity causes a decrease in efficiency over the structured grid algorithms.[4] There is a definite need to combine the geometric flexibility of the unstructured finite volume algorithms with the numerical efficiency of the structured algorithms, especially for high-speed flows.

Many algorithms for structured discretizations are capable of space marching certain high-speed problems by taking advantage of characteristic theory. In particular, conservative algorithms of the type developed by Walters and Dwoyer[1,2] and Newsome et al.,[3] which retain the time-dependent terms, are capable of producing second-order-accurate solutions in the marching direction. It has been shown that the computational work can be reduced by as much as a factor of 10 as compared to popular iterative techniques common in solving subsonic problems. However, it can be very difficult to construct a single structured grid about three-dimensional bodies that are geometrically complex. Often, these grids contain highly skewed volumes that introduce errors into the solution (Fig. 1). Employing an unstructured grid eliminates the constraint of structured connectivity such that these highly skewed cells may be avoided. However, this lack of structure eliminates the existence of any meaningful marching direction. Current three-dimensional unstructured grid generators generate tetrahedra, which have no particular order.[5] A method for generating a quasiunstructured grid has thus been sought in order to gain most of the flexibility of an unstructured mesh while maintaining the benefits of a space-marching algorithm.

A technique has been chosen to generate two-dimensional unstructured grids in the crossflow plane and connect the consecutive grids to create control volumes. A constraint to this method is that one must generate consecutive grids with the same number of nodes and identical connectivity. In order to apply this technique, it is necessary to extend a conventional two-dimensional grid generator to perform the task of generating consecutive grids of identical connectivity. A second approach would be to generate a completely unstructured grid within the bounds of two adjacent crossflow planes. The first approach was chosen due to the availability of a two-dimensional unstructured grid generator.[6]

Because of the streamwise connectivity incorporated into discretization, streamwise variations of the geometry are somewhat limited. In order to describe more complex geometries, a zonal grid technique (popular among structured algorithms) has been employed.[7-9] The domain may be split into two or more separate zones such that each zone will have two-dimensional grids created in the aforementioned manner.

A discussion of the governing equations and the spatial discretization of the domain follows. The interpolation from one arbitrary unstructured mesh to another is discussed followed by a description of the grid generation technique. Finally, several solutions are presented to demonstrate the feasibility of this method.

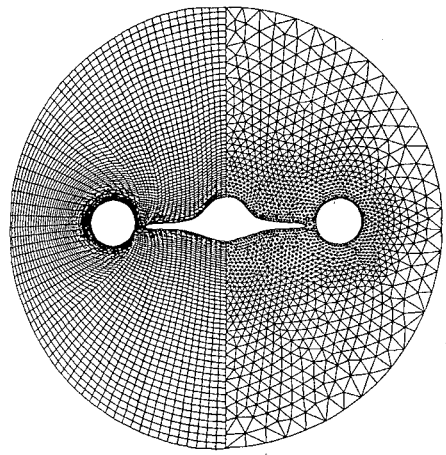

Fig. 1 Comparison of structured and unstructured grids about a single crossflow plane.

Received July 19, 1990; revision received Nov. 23, 1990; accepted for publication Nov. 26, 1990.

*Research Scientist. Student Member AIAA.
†Associate Professor, Department of Aerospace and Ocean Engineering. Senior Member AIAA.
‡Research Scientist. Member AIAA.

Spatial Discretization

The Euler equations are the governing equations considered in this study. However, the algorithm presented here is also applicable to the parabolized Navier-Stokes (PNS) equations. The integral form of the Euler equations may be written as

$$\frac{\partial}{\partial t}\iiint_V Q\,dV + \iint_S \boldsymbol{F}\cdot\hat{n}\,dS = 0 \qquad (1)$$

where $Q = [\rho, \rho u, \rho v, \rho w, \rho e_0]^T$ is the vector of conserved variables; ρ the density; u, v, w the velocities in the x, y, z directions; and e_0 the total energy per unit mass. The vector $\boldsymbol{F}$ is the flux of mass, x, y, z momentum, and energy. The quantity $\boldsymbol{F}\cdot\hat{n}$ represents the flux out of the volume V, through the surface S when $\hat{n}$ is the outward pointing normal of S. This system of equations is closed by the perfect gas equation of state

$$p = (\gamma - 1)\left[\rho e_0 - \rho\frac{(u^2 + v^2 + w^2)}{2}\right]$$

where p is the pressure, and γ is the ratio of specific heats.

The domain is discretized in a finite volume sense. Each control volume has two logical indices, (i, n), where i specifies the current crossflow plane and n distinguishes different elements within a single crossflow plane. In addition, a cell vertex discretization has been chosen within the crossflow plane (Fig. 2) similar to the approach taken by Stoufflet et al.[10]; a cell centered scheme could just as easily have been chosen. In the cell vertex method, the vertices of the triangular elements form the approximate centroid of the resulting volumes. The control volume (Fig. 3) is enclosed by two distinct types of faces. The first are n-sided polygons at the inflow and outflow of the crossflow plane, where n is the number of vertices surrounding the current vertex. These faces separate the volumes among neighboring crossflow planes. The second type of face is that separating the volumes within a single crossflow plane. These faces are quadrilaterals formed by connecting the coinciding edges of the inflow and outflow faces.

With this discretization, Eq. (1) may be approximated as

$$V_{i,n}\frac{\partial\langle Q_{i,n}\rangle}{\partial t} = -\sum_{j=1}^{\text{faces}}\langle\hat{F}_{i,j}\rangle\,\Delta A_{i,j}$$

where $V_{i,n}$ is the volume of element i, n, and $\langle Q_{i,n}\rangle$ is the volume average of Q over the element i, n. $\langle\hat{F}_{i,j}\rangle$ is the flux out of element i, n from face j and is approximated as the value of $\boldsymbol{F}\cdot\hat{n}$ averaged over the area $\Delta A_{i,j}$ of face i, j. The values of $\langle\hat{F}\rangle$ for the faces separating volumes within a plane are constructed using Roe's characteristic-based flux difference

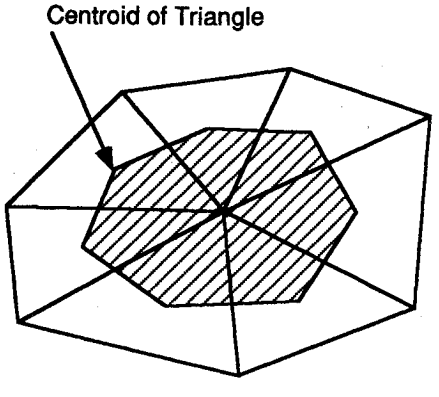

Fig. 2 Cell vertex discretization of an unstructured grid.

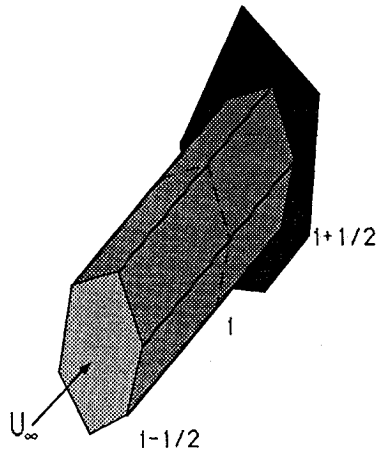

Fig. 3 Typical control volume on *i*th plane.

splitting (FDS) scheme.[11,12] A left and right state, Q^- and Q^+, are necessary for this scheme and can be obtained to first-order accuracy from the average values of the two elements adjacent to each face. Since this is a space-marching algorithm, the full, unsplit fluxes are used for the inflow and outflow faces to the plane. These fluxes are evaluated using values of Q interpolated from upstream.

Higher-Order Spatial Accuracy

The spatial accuracy may be improved by constructing higher-order approximations of $Q^{\pm}$ used in the FDS scheme. This can be accomplished through the use of the kappa formulation common among structured algorithms,

$$Q^-_{i+1/2} = Q_i + \frac{1}{4}\{(1-\kappa)\overline{\nabla} + (1+\kappa)\overline{\Delta}\}\,Q_i$$

$$Q^+_{i+1/2} = Q_{i+1} + \frac{1}{4}\{(1+\kappa)\overline{\nabla}; + (1-\kappa)\overline{\Delta}\}\,Q_{i+1}$$

The + and − superscripts denote right- and left-biased interpolations, respectively. κ controls the biasing and order of the interpolation and varies between $[-1,1]$. This formulation includes the min-mod limiter where

$$\overline{\Delta}Q = \text{min-mod}\,(\Delta Q, \beta\nabla Q)$$

$$\overline{\nabla}Q = \text{min-mod}\,(\nabla Q, \beta\Delta Q)$$

and

$$\text{min-mod}(x,y) = \text{sgn}(x)*\max(0,\min[x\ \text{sgn}(y), y\ \text{sgn}(x)])$$

with

$$\beta = (3-\kappa)/(1-\kappa)$$

The Δ and ∇ operators represent forward and backward differences, respectively. The use of a limiter reduces the order of accuracy of the interpolation to first order near regions of large gradients in order to suppress oscillations near discontinuities.

This method assumes a one-dimensional uniform distribution of points that lends itself well to structured discretizations. With the present discretization, it would only be possible to directly implement this formulation in the streamwise direction. A fully upwind, second-order interpolation ($\kappa = -1$) is performed in the streamwise direction as dictated by the marching algorithm. Higher-order interpolations in the crossflow plane are constructed in a manner similar to the two-dimensional formulation of Whitaker et al.[13] The interpolations are performed along vectors passing through the

vertices directly to either side of the current face labeled points P_{n1} and P_{n2} in Fig. 4.

Forward and backward differences are needed along this vector for the kappa formulation. The difference $Q_{n1'} - Q_{n1}$ is constructed by dotting the two-dimensional gradient of Q with the vector $\boldsymbol{P}_{n1}\boldsymbol{P}_{n1'}$. The point $P_{n1'}$ is constructed such that

$$\boldsymbol{P}_{n1}\boldsymbol{P}_{n1'} = \boldsymbol{P}_{n2}\boldsymbol{P}_{n1}$$

The gradient of Q is obtained by using Gauss theorem and integrating about the triangular cell containing the point P_{n1} and the vector $\boldsymbol{P}_{n1}\boldsymbol{P}_{n1'}$. The difference $Q_{n2'} - Q_{n2}$ is derived in a similar fashion to that of $Q_{n1'} - Q_{n1}$.

Zonal Interpolation

When the the use of two or more zones of different cross-flow plane connectivity is dictated, it is necessary to interpolate values from one zone to another. The use of multiple zones is demonstrated in Figs. 5 where the two grids shown discretize the same plane of the model SR71 described later. This station is located just forward of the engine diffuser cones. The grid in Fig. 5a is an appropriate discretization for the forebody of the aircraft, whereas the grid in Fig. 5b contains a clustering of points near the diffusers necessary to resolve the gradients that will occur at that location. It is clear from this picture that the faces of one zone do not coincide with those of the other and that a method for constructing the fluxes at the inflow of the second zone is required.

An interpolation procedure that is efficient to program and is conservative to within the truncation error of the discretization scheme has been chosen. The approach is to interpolate to find values of Q^2 at planes i and $i-1$ where plane i is the last plane of zone 1 (superscripts indicate the zone to which the values refer). Note that only values of Q^1 are known at planes i and $i-1$. Once these values are obtained in zone two, it is possible to perform the upwind interpolation necessary for the construction of the fluxes at the inflow faces of the first plane in zone 2. The two-dimensional grids at the zonal interface are used to obtain the necessary interpolation coefficients. A quad-tree search algorithm is employed to find the triangular element from zone 1 that contains the vertex P_n^2 in zone 2 (Fig. 6).

A linear interpolation of Q_{n1}^1, Q_{n2}^1, and Q_{n3}^1 between the vertices P_{n1}^1, P_{n2}^1, and P_{n3}^1 is used to determine the value of Q_n^2 at P_n^2. An efficient search algorithm has been sought in order to reduce the cost associated with the zonal interpolation. Quad-tree technology[6] has been employed to locate rapidly the elements from zone 2 that are closest to the vertex for which a value is being interpolated. A description of the quad-tree structure and associated search algorithms may be found in Ref. 6. This search algorithm greatly improves the interpolation performance as compared to a brute force type search algorithm.

Time Integration

A time-dependent space-marching algorithm is used in the integration to the steady-state solution.[1-3] With this formulation, a single plane is integrated in time. Since the characteristics do not travel opposite to the marching direction, the solution for the current plane does not impose itself upon the solution of the upstream planes. Equivalently, the current plane is not affected by the solution downstream of the marching direction. This means that individual planes may be integrated in time separately. The flux terms on the right side due to the upstream terms will be constant homogeneous terms to the equations; there will be no terms due to the downstream values. A four-stage Runge-Kutta scheme[14,15] has been chosen for the time integration of the individual planes

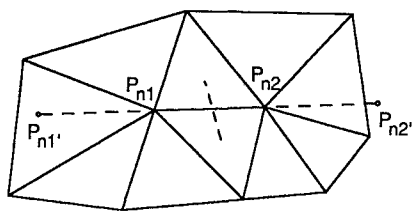

Fig. 4 Construction of higher-order interpolation.

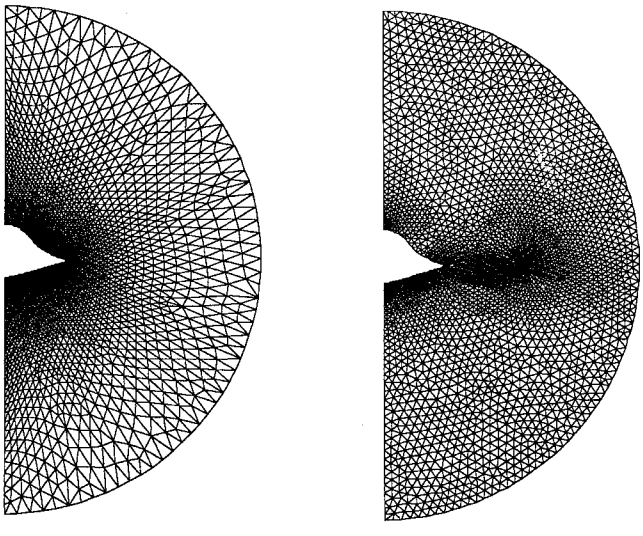

Fig. 5 Two-dimensional grids at a zonal interface.

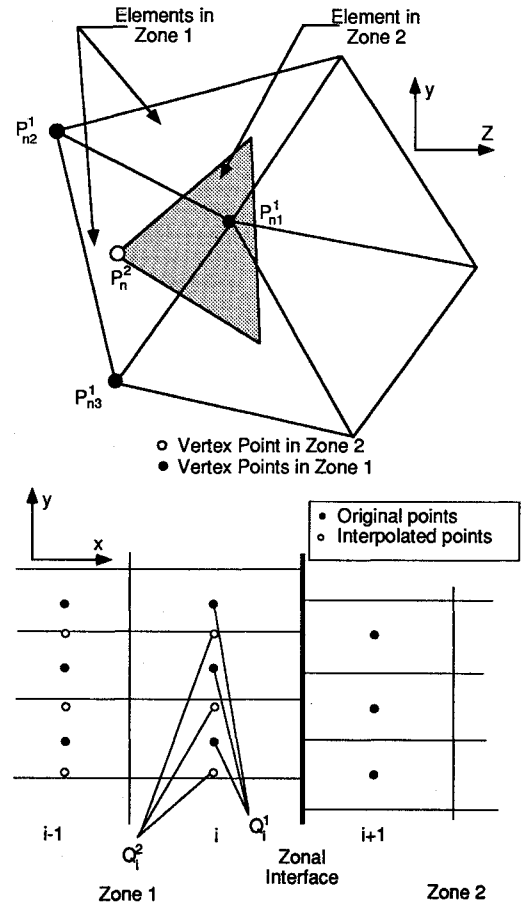

Fig. 6 Zonal interpolation.

that requires the storage of only two levels of Q. This scheme has the advantage of being very simple and efficient to implement. On the other hand, being an explicit scheme, this method requires many more iterations to converge than an implicit time integration scheme. It is known that such implicit schemes are capable of being implemented with this algorithm, but an explicit scheme is considered sufficient to demonstrate the feasibility of the spatial discretization, which is the purpose of this work. In addition, one of the objectives of this project is to enable the use of this algorithm on engineering workstations where efficient use of memory plays an extremely important role in the selection of algorithms.

Grid Generation

The generation of the unstructured two-dimensional grids in the crossflow planes is essential to the success of this algorithm. A two-dimensional unstructured grid generator developed by Löhner[6] using the advancing front technique is employed to generate a single grid for each zone. This method gives control over grid point spacing as well as providing a technique for adaptive remeshing. If the geometry does not radically vary, it is possible to generate a second grid at an adjacent plane by forcing the boundary points of the original grid to align themselves with the boundary of the second plane. In this way, the resulting grid has the same number of nodes as the first while maintaining the original connectivity. An example of such an initial grid is shown in Fig. 7a. This is a course grid generated about a crossflow plane of the model SR71 problem and is suitable only for demonstration. This plane includes the midsection of the fuselage, the leading edge of the delta wing, and a section of the engine diffuser. Figures 7b and 7c denote the two steps required to generate a second two-dimensional grid about a crossflow plane forward of the first grid shown in Fig. 7a. The second plane also includes the midsection of the fuselage, leading edge of the wing, and diffuser. However, the span of the wing is less than that of the first station, and the radius of the diffuser is less at this location than the first.

The first step shown in Fig. 7b is to redistribute all of the boundary points of the grid in Fig. 7a such that they all fall on the boundary of the second plane. This step is accomplished while ensuring that the same number of points are maintained on each individual segment of the boundary. Thus, the same number of points are kept on the top surface of the body and wing, as well as the top symmetry line, etc. In addition, the distribution is kept the same along each boundary segment. For instance, if there were clustering near the tip of the wing, this clustering would be maintained from one grid to the next. Examining Fig. 7b, it is apparent that stretching has occurred along the body and diffuser. In order to reduce this stretching, the resulting grid is smoothed with the algorithm used in the advancing front generator (Fig. 7c). This algorithm attempts to place each point at the center of the polygon formed by connecting all of the surrounding triangular elements. In this way, grid points are more uniformly distributed such that they better describe the geometry of the current plane.

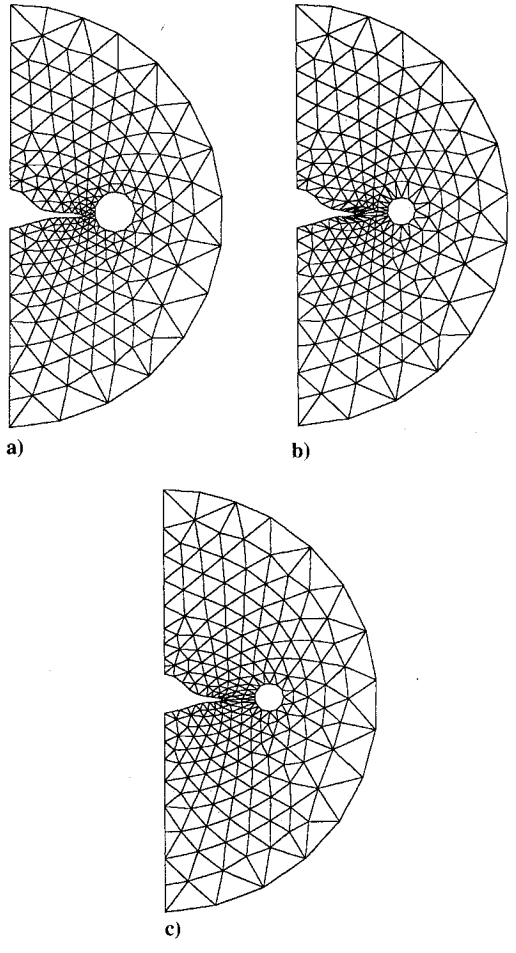

Fig. 7 Construction of two-dimensional grids.

Applications

5-Degree Cone

The first test case is the flow about a 5-deg half-angle cone in a Mach 5 freestream. This is a simple problem not meant to test the geomeric flexibility but to verify the solution algorithm. An exact inviscid analytic solution to this problem is presented for comparison to the numerical solution. A structured triangular grid containing 1691 elements has been generated about the cone by dividing the quadrilaterals of a polar mesh into two triangular elements. A value of $\kappa = 1/3$ is used in addition to the min-mod limiter in this and all of the remaining test cases, resulting in a higher-order upwind biased interpolation in the crossflow plane. Figure 8 shows a comparison of normalized pressure along a radial line emanating from the cone axis. In Fig. 8, phi represents the angle with the cone vertex along a radial line. The figure shows good agreement between the numerical and analytical solutions. The computed solution has correctly captured the oblique shock as well as predicted the correct value of pressure on the surface of the cone.

Analytic Forebody

Two distinct discretizations have been used to compute the solution about an analytic forebody. In both cases, a fully upwind second-order interpolation has been used in the freestream direction with no limiting. A freestream Mach number of 1.7 along with a zero angle of attach have been chosen to

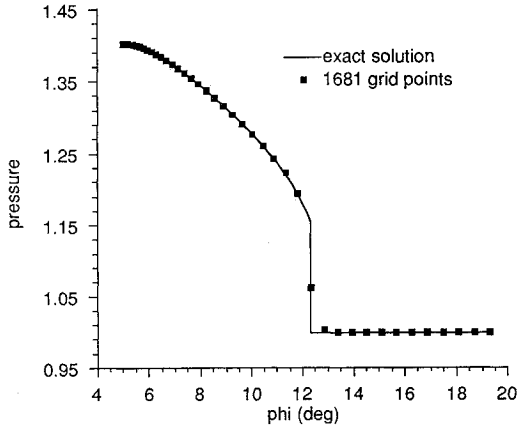

Fig. 8 Pressure along radial line from 5-deg cone at Mach 5.

run this problem. Experimental data about this configuration are available at a Reynolds number of 3×10^6 based upon the body length.[16] The first discretization contains a single zone of 40 crossflow planes with 961 grid points in each plane. Like the cone discretization, this grid originated from a structured quadrilateral mesh with each cell being divided into two triangular elements. This discretization may then be compared to that of a structured marching algorithm with the same number of grid points and distribution.[17] In addition, a two-zone grid has been constructed employing the advancing front grid generation technique. The first zone consists of 22 planes with 1330 cells per plane, whereas the second zone consists of 18 planes with 1797 cells per plane. The second zone is necessary to ensure sufficient points near the surface of the aft portion of the forebody. Figure 9 shows pressure contours about the symmetry plane and last crossflow plane for the two-zone unstructured discretization. In addition, the grid has been superimposed on the surface of the body and the last crossflow plane. Note that the zonal interpolation yields smooth variation in contours from one zone to the next. The pressure coefficient C_p is shown along the top and bottom surface of the body in Fig. 10. All numerical solutions show good agreement with experimental data. Most important, note that there is good agreement between the three numerical solutions. Thus, the solution accuracy of the unstructured algorithm is similar to that of the structured algorithm with similar grid point distribution.

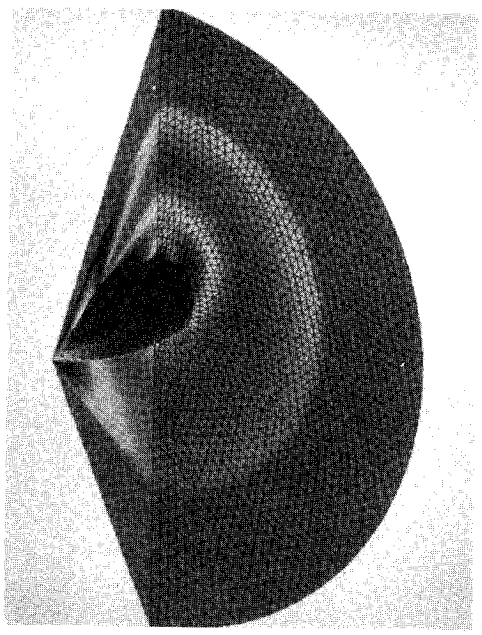

Fig. 9 Pressure contours and surface grid of analytic forebody.

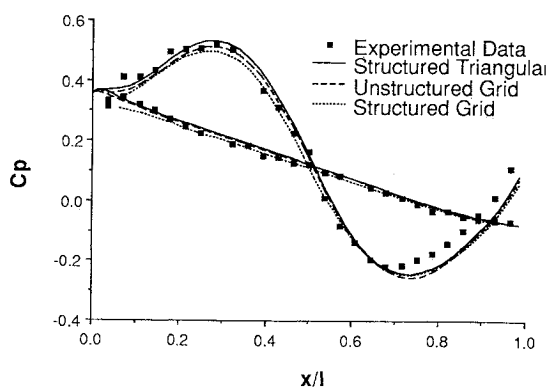

Fig. 10 Longitudinal C_p distributions in the symmetry plane of the analytic forebody.

Model SR71

To demonstrate the geometric flexibility of this algorithm, the solution about a simplified model of the SR71 reconnaissance aircraft has been calculated. This model includes a region of multiple elements in the streamwise direction at the start of the engine inlets, as well as multiple vertical tails. Both of these geometries cause difficulties with a structured discretization, whereas they impose no restrictions with an unstructured discretization. The solution is calculated in a Mach 3.5 freestream at a 0-deg angle of incidence relative to the root chord. This solution was obtained on a grid with a total of 42 crossflow planes in 10 separate zones: 4 in the forebody region, 2 in the multiple element inlet region, 2 in the region forward of the vertical tail, and 2 to resolve the vertical tail section. The number of nodes in each plane varied from 2338 in the first zone to a maximum of 5589 in the next to last zone. Figure 11 shows the grid on the surface of the body as well as in the last crossflow plane. Note that there are discontinuities in the grid along the body at the zonal interfaces. Figure 12 presents pressure contours as well as the grid along the body. In this figure, the bow shock and oblique shocks from the nacelles may be seen.

Concluding Remarks

It has been demonstrated that the use of an unstructured discretization along with the use of a space-marching algo-

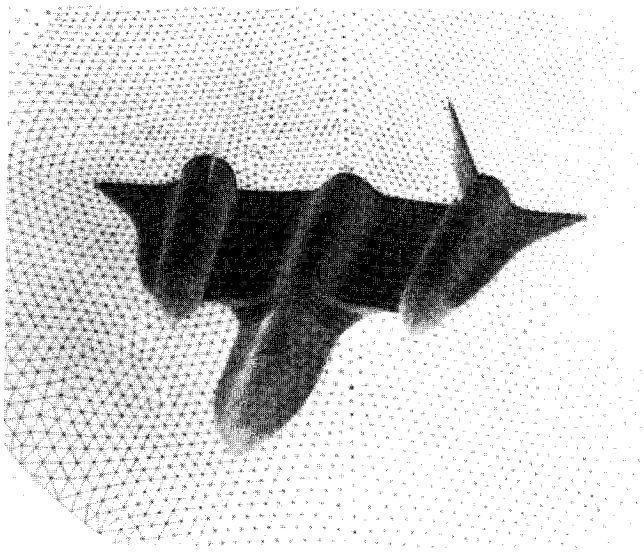

Fig. 11 Grid on the surface and in the exit plane of model SR71.

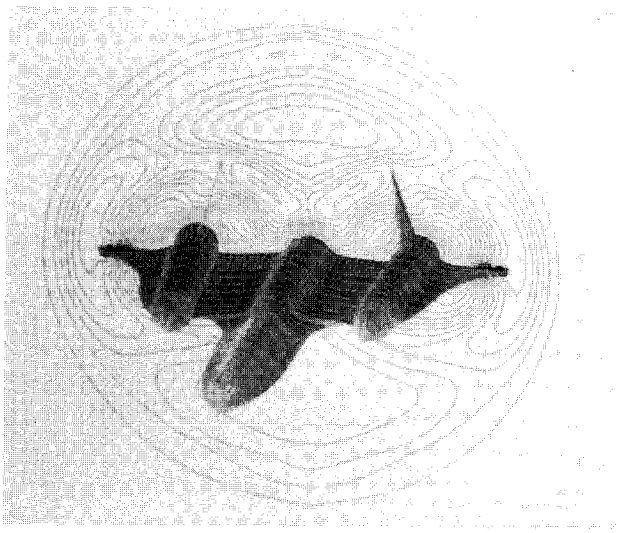

Fig. 12 Line contours in the exit plane of model SR71.

rithm is technically feasible. Comparison of the one- and two-zone solutions of the analytical forebody show that the interpolation procedure used here is sufficient to ensure a smooth and accurate transition of the solution from one zone to the next. The solution over the model SR71 has demonstrated the ability to model accurately complex three-dimensional bodies. Currently, the grid generation process requires more interaction with the user than does the actual flow simulation. Since all grids in a single zone originate from the same initial grid, it is not possible to discretize a domain with as few zones as is possible with a structured grid. This is evident in the analytic forebody problem where it is possible to generate a single-zone structured grid, while it was necessary to generate a two-zone unstructured grid. In addition, care must be taken in the selection of the initial grid for each zone to ensure adequate discretizations for each plane in the zone. Further investigation of the current grid generation technique might yield a more robust algorithm capable of discretizing more of the domain, with less forethought from the user. However, it is thought that a more productive course of action would be to develop a discretization of tetrahedral elements typical of the completely unstructured algorithms that contains sufficient structure to enable a marching algorithm to be employed. In this way, the need for zonal interpolation would be eliminated while enabling the discretization of domains that are much more complex in the marching direction.

Acknowledgments

This work was supported by National Science Foundation Grant ISI-8861052 and by NASA Grant NAG-1-776.

References

[1]Walters, R. W., and Dwoyer, D. L., "Efficient Solutions to the Euler Equations for Supersonic Flow with Embedded Subsonic Regions," NASA TP 2523, 1987.

[2]Walters, R. W., and Dwoyer, D. L., "An Efficient Iteration Strategy for the Solution of the Euler Equations," AIAA Paper 85-1529, July 1985.

[3]Newsome, R. W., Walters, R. W., and Thomas, J. L., "An Efficient Iteration Strategy for Upwind Relaxation Solutions to the Thin-Layer Navier-Stokes Equations," AIAA Paper 87-1113, June, 1987.

[4]Löhner, R., "Finite Elements in CFD: What Lies Ahead," *International Journal of Numerical Methods Engineering*, No. 24, 1987, pp. 1741–1756.

[5]Löhner, R., Morgan, K., Peraire, J., and Vahdati, M., "Finite Element Flux-Corrected Transport (FEM-FCT) for the Euler and Navier-Stokes Equations," *International Journal of Numerical Methods Fluids*, No. 7, 1987, pp. 1093–1109.

[6]Löhner, R., "Some Useful Data Structures for the Generation of Unstructured Grids," *Communications in Applied Numerical Methods*, Vol. 4, 1988, pp. 123–135.

[7]Rai, M. M., "A Relaxation Approach to Patched-Grid Calculations with the Euler Equations," AIAA Paper 85-0295, Jan. 1985.

[8]Rai, M. M., "An Implicit, Conservative, Zonal-Boundary Scheme for Euler Equation Calculations," AIAA Paper 85-0488, Jan. 1985.

[9]Hessenius, K. A., and Rai, M. M., "Three Dimensional, Conservative, Euler Computations Using Patched-Grid Systems and Explicit Methods," AIAA Paper 86-1081, May 1986.

[10]Stouffet, B., Périaux, J., Fezoui, F., and Dervieux, A., "Numerical Simulation of 3-D Hypersonic Euler Flows Around Space Vehicles Using Adapted Finite Elements," AIAA Paper 87-0560, Jan. 1987.

[11]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[12]Roe, P. L., "Characteristic Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337–365.

[13]Whitaker, D. L., Grossman, B., and Löhner, R., "Two-Dimensional Euler Computations on a Triangular Mesh Using an Upwind, Finite-Volume Scheme," AIAA Paper 89-0470, Jan. 1989.

[14]Van der Houwen, P. J., "Construction of Integration Formulas for Initial Value Problems," North Holland, Amsterdam, 1977.

[15]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions for the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.

[16]Townsend, J. C., Howell, D. T., Collins, I. K., and Hayes, C., "Surface Pressure Data on a Series of Analytic Forebodies at Mach Numbers from 1.70 to 4.50 and Combined Angles of Attack and Sideslip," NASA TM 80062, June 1979.

[17]Walters, R. W., Reu, T., McGrory, W. D., and Thomas, J. L., "A Longitudinally-Patched Grid Approach with Applications to High Speed Flows," AIAA Paper 88-0715, Jan. 1988.