NASA Technical Memorandum 107643

P.13

PROGRESS IN UNSTRUCTURED-GRID METHODS DEVELOPMENT FOR UNSTEADY AERODYNAMIC APPLICATIONS

JOHN T. BATINA

JULY 1992

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

PROGRESS IN UNSTRUCTURED-GRID METHODS DEVELOPMENT FOR UNSTEADY AERODYNAMIC APPLICATIONS

John T. Batina
NASA Langley Research Center
Hampton, Virginia 23665–5225 USA

ABSTRACT

The paper summarizes the development by the author of unstructured-grid methods for the solution of the equations of fluid flow and shares what has been learned over the course of the research. The focus of the discussion is on the solution of the time-dependent Euler equations including spatial discretizations, temporal discretizations, and boundary conditions. An example calculation with an implicit upwind method using a CFL number of infinity is presented for the Boeing 747 aircraft. The results were obtained in less than one hour CPU time on a Cray-2 computer, thus demonstrating the speed and robustness of the capability. Additional calculations for the ONERA M6 wing demonstrate the accuracy of the method through the good agreement between calculated results and experimental data for a standard transonic flow case.

INTRODUCTION

The development of unstructured-grid methods for the solution of the equations of fluid flow has become quite popular in recent years because of the advantages that these methods offer in comparison with the more-traditional structured-grid methods [1–24]. For example, with an unstructured grid, it is relatively easy to model very complicated three-dimensional geometries such as a complete aircraft configuration. With a structured grid, it is generally much more difficult to achieve this level of geometrical complexity without resorting to more sophisticated meshing methodologies (such as blocked, patched, chimera, or hybrid type grids), which in turn, significantly complicate the solution algorithm of the governing fluid flow equations. A second advantage is that unstructured grid methods enable, in a natural way, adaptive mesh refinement to more accurately predict the physics of the flow. These methods involve enrichment and coarsening procedures, either to add points in high gradient regions of the flow or to remove points where they are not needed, and thereby to produce solutions of high spatial accuracy at minimal computational cost. Furthermore, analogous to spatial adaption, a temporal adaption technique may be used on an unstructured grid to improve the computational efficiency of explicit time-marching schemes for unsteady aerodynamic applications. Temporal adaption can be thought of as time-accurate local time stepping. With this procedure, each grid cell is integrated according the local flow physics and numerical stability. Time accuracy is maintained by bringing all cells to the same time level as determined by the step size of the largest cell.

The author has been involved over the past five years in the development of unstructured-grid methods for the solution of the time-dependent Euler and Navier-Stokes equations [11–24]. These methods are being developed primarily for unsteady aerodynamic and aeroelastic applications. As a result of developing and testing a variety of methods, much knowledge and experience has been gained. Therefore the purpose of the paper is to summarize this developmental effort and to share what has been learned over the course of the research. It is emphasized, however, that the views presented are those of the author, which are not necessarily consistent with the views

of other researchers. Differences in opinion over the best choice of method, for example, may result due to the unsteady application of the present methods. The focus of the discussion is on the solution of the time-dependent Euler equations including spatial discretizations, temporal discretizations, and boundary conditions. Both central-difference and upwind-type spatial discretizations are described in the context of both cell-vertex and cell-centered implementations. Explicit and implicit temporal discretizations are also discussed along with methods to accelerate convergence to steady state. Strong and weak implementations of the surface boundary conditions and far field conditions are further described. Finally, example applications are presented to illustrate the various methods.

EULER EQUATIONS

In the present study the flow is assumed to be governed by the time-dependent Euler equations which may be written in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega} Q dV + \int_{\partial\Omega} (En_x + Fn_y + Gn_z) dS = 0 \tag{1}$$

where Q is the vector of conserved variables, and E, F, and G are the convective fluxes. Equation (1) has been nondimensionalized by the freestream density and the freestream speed of sound. Also, the second integral in Eq. (1) is a boundary integral resulting from application of the divergence theorem, and n_x, n_y, and n_z are Cartesian components of the unit normal to the boundary surface.

SPATIAL DISCRETIZATIONS

In general, the Euler equations in integral form have been solved numerically using several different finite-volume spatial discretizations on unstructured grids of triangular or tetrahedral cells [11, 13, 16, 18, 21]. These discretizations are either of the central-difference type with explicit artificial dissipation or of the upwind-type which are naturally dissipative.

Central-Difference Algorithms

Central-difference type algorithms may be implemented as either cell-vertex (node-based) or cell-centered discretizations. In the cell-vertex approach, the flow variables are stored at the nodes and the control volume at a given node is typically taken to be all of the neighboring cells which have a vertex at that node, as illustrated in two dimensions by Fig. 1. In the cell-centered approach, the flow variables are stored at the centroids of the cells and the control volume is simply the cell itself. In three dimensions, it is noted that tetrahedral meshes contain between five and six times the number of cells as nodes. Consequently, a cell-centered scheme is inherently five to six times more costly in terms of CPU time and memory than a cell-vertex scheme. However, for a given mesh, the cell-centered scheme is expected to produce a more accurate solution spatially since it is effectively solving the problem with five or six times the number of control volumes as the cell-vertex scheme. In either case, the spatial discretization involves a flux balance where the fluxes are computed along each face that makes up the control volume using flow variables that have been averaged. The averaged flow variables are determined using values at the three nodes which make up a given face for the cell-vertex approach and using values at the centroids of the two tetrahedra that share a given face for the cell-centered approach.

The unsteady Euler equations are a set of nondissipative hyperbolic conservation laws whose numerical solution requires some form of artificial dissipation to prevent oscillations near shock waves and to damp high frequency uncoupled error modes. Generally, a combination of harmonic and biharmonic operators is used, corresponding to second and fourth differences of the conserved variables, respectively. The simplest way to compute a harmonic operator in the cell-vertex approach, for example, is to sum the difference in flow variables over all edges which

have a common vertex at node i, given by

$$\nabla^2 Q_i \approx \sum_{m=1}^{M} (Q_m - Q_i) \tag{2}$$

The biharmonic operator can be approximated similarly by applying Eq. (2) twice. An example application of this form of artificial dissipation in the cell-vertex approach is shown in Fig. 2 for the NACA 0012 airfoil at a freestream Mach number M_∞ of 0.8 and an angle of attack α_0 of 1.25° [11]. This simple treatment of the artificial dissipation works reasonably well for meshes that are "regular" (smooth meshes with little distortion, and a similar number of edges share each node at nearly equal angles, etc.) although the shock waves are smeared typically over several cells. For meshes that are not regular or when spatial adaption is used (naturally leading to highly irregular meshes), however, Eq. (2) does not work well. In these cases a more accurate treatment of the dissipative operators is required such as that given by

$$\nabla^2 Q_i \approx \sum_{m=1}^{M} w_m (Q_m - Q_i) \tag{3}$$

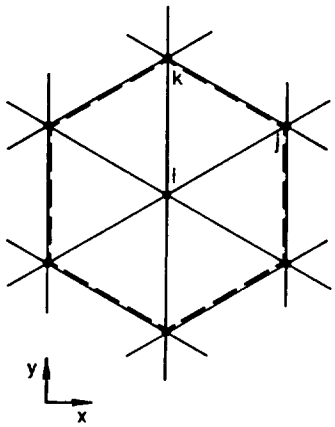

Fig. 1 Definition of the control volume in the two-dimensional cell-vertex version of the central-difference algorithm.

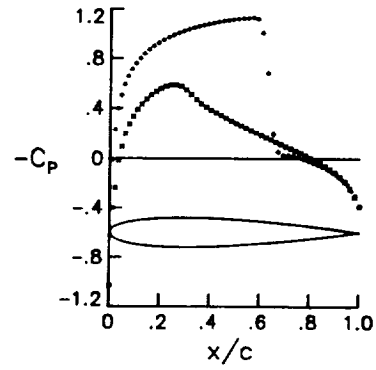

Fig. 2 Steady pressure coefficient distribution for the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha_0 = 1.25°$ computed using the cell-vertex central-difference algorithm.

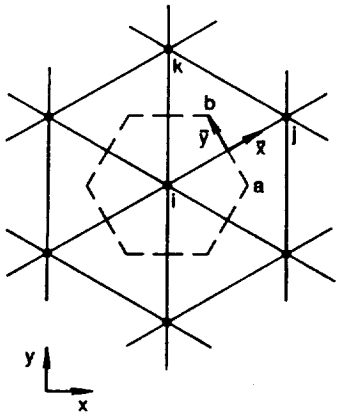

Fig. 3 Definition of the control volume in the two-dimensional cell-vertex version of the upwind-difference algorithm.

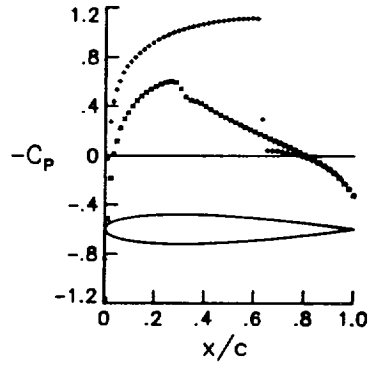

Fig. 4 Steady pressure coefficient distribution for the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha_0 = 1.25°$ computed using the cell-vertex upwind-difference algorithm.

where the weights w_m have been derived by a constrained optimization so that Eq. (3) is exact for a linearly varying Q (thus implying second-order spatial accuracy) [25].

Advantages of the central-difference type discretization are that it is easier to code and requires less memory than an upwind discretization. However, the difficulties in calculating accurately the artificial dissipation on irregular unstructured grids suggest that an upwind discretization may be more desirable. Also, the cell-centered approach has advantages in comparison with the cell-vertex approach. For example, the cell-centered scheme vectorizes naturally (no special coding is required) whereas the flux balance in a cell-vertex scheme involves vector recurrences which inhibit vectorization. To avoid vector recurrences and allow vectorization, cell-vertex schemes require that the cells be sorted into groups, such that not more than one cell in a group accesses a given node. The additional coding that is required to accomplish this is somewhat unwieldy and increases the computational cost of the cell-vertex approach. Consequently, the cost of a cell-vertex method is actually closer to the cost of a cell-centered method than the factor of five to six difference in control volumes would suggest.

Upwind Algorithms

Upwind type algorithms also have been implemented as either cell-vertex or cell-centered discretizations. In the cell-vertex approach, the flow variables are stored at the nodes and the control volume is typically taken to be part of the neighboring cells which have a vertex at that node. In two dimensions, the part of the cells taken is determined by connecting the centroid of the cell with the midpoints of the two edges which share the node as shown in Fig. 3. An example application of this form of upwind-difference type algorithm is shown in Fig. 4 for the NACA 0012 airfoil at the same freestream flow conditions as before [11]. Here though, the shock waves are more sharply captured because of the upwind discretization of the governing flow equations. For the cell-vertex approach in three dimensions, the part of the cells taken for the control volume is determined by a surface that is constructed similarly. However, this is quite complicated geometrically to do in three dimensions, and since these surfaces (and their corresponding unit normals and areas) have to be recomputed at each time step when the mesh deforms for unsteady cases, the cell-vertex approach is not the preferred method. In the cell-centered approach, the flow variables are stored at the centroids of the cells and the control volume is simply the cell itself.

Cell-centered, upwind, spatial discretizations have been developed [16, 18, 22] based on either the flux-vector splitting (FVS) of van Leer [26] or the flux-difference splitting (FDS) of Roe [27]. The FDS approach is generally preferable since it is less dissipative than FVS and is easier to code. Similar to the central-difference algorithms, the upwind algorithms involve a flux balance where the fluxes along the four faces of a given tetrahedron are summed as

$$\sum_{m=1}^{4} H\Delta S = \sum_{m=1}^{4} (En_x + Fn_y + Gn_z)\Delta S \tag{4}$$

where ΔS is the area of the face. With FDS the flux vector H is approximated by

$$H = \frac{1}{2}[H(Q^+) + H(Q^-) - \left|\tilde{A}\right|(Q^+ - Q^-)] \tag{5}$$

where Q^- and Q^+ are the state variables to the left and right of the cell face and A is the flux Jacobian matrix given by $\partial H/\partial Q$. Also the tilde and the absolute value sign indicate that the flux Jacobian is evaluated using the so-called Roe-averaged flow variables and the absolute value of the characteristic speeds. The left and right states Q^- and Q^+ are determined by upwind-biased interpolations of the primitive variables q, similar to that which is done on structured meshes.

A significant advantage of the upwind type discretization is that it is naturally dissipative, in comparison with central difference discretizations, and consequently does not require adjustment of free parameters to control the dissipation. The upwinding accounts for the local wave-propagation characteristics of the flow and captures shock waves sharply. To demonstrate the sharp shock-capturing features of the cell-centered upwind approach, steady and unsteady results are presented for the NACA 0012 airfoil. The steady calculation was performed for $M_\infty = 0.8$

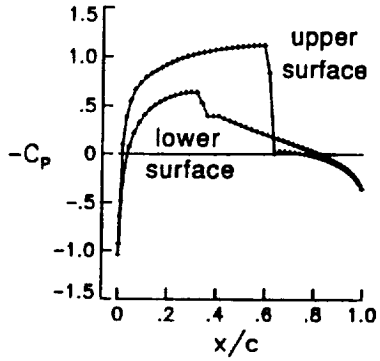

Fig. 5 Steady pressure coefficient distribution for the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha_0 = 1.25°$ computed using the cell-centered FDS algorithm.

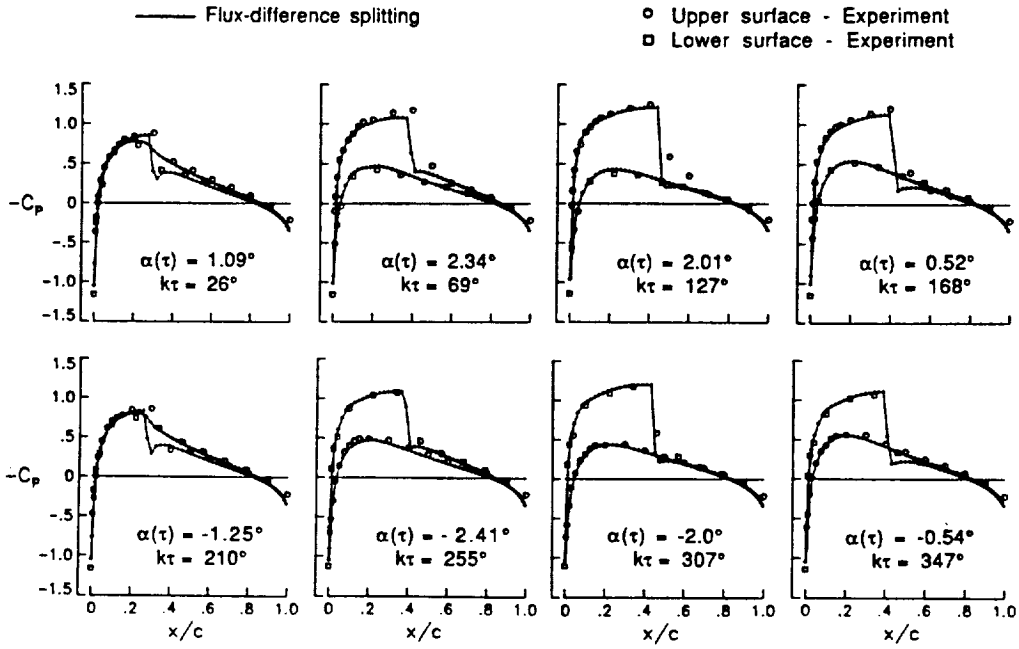

Fig. 6 Instantaneous pressure distributions for the NACA 0012 airfoil pitching harmonically at $M_\infty = 0.755$, $\alpha_0 = 0.016°, \alpha_1 = 2.51°$, and $k = 0.0814$ computed using the cell-centered FDS algorithm.

and $\alpha_0 = 1.25°$ with the resulting pressure distribution shown in Fig. 5. The shocks are sharply captured with only one grid point within the shock structure on either surface. Additionally, these sharp shock capturing features of the upwind method carry over to unsteady cases as well. For example, for the NACA 0012 airfoil pitching harmonically at $M_\infty = 0.755$, $\alpha_0 = 0.016°$, with an oscillation amplitude of $\alpha_1 = 2.51°$ at a reduced frequency of $k = 0.0814$, instantaneous pressure distributions at eight points in time during a cycle of motion are shown in Fig. 6. This is a very interesting case since the shock waves on the upper and lower surfaces of the airfoil periodically appear and disappear during the cycle of motion. It is clear from the results of Fig. 6 that, similar to the steady-state example, the calculated shock waves are captured sharply with only one point within the shock structure.

TEMPORAL DISCRETIZATIONS

The Euler equations have been integrated in time numerically using several different temporal discretizations [11, 13, 16, 18, 21]. These discretizations are either of the explicit type, along with several techniques to accelerate

convergence to steady state, or of the implicit type, which generally do not require such additional techniques for rapid convergence to steady state.

Explicit Algorithm

The explicit algorithm that has been developed for use on unstructured meshes is the standard multi-stage Runge-Kutta time-integration scheme. In this scheme the flux balance is determined at each stage and, for computational efficiency, the artificial dissipation of the central-difference spatial discretization is evaluated only at the first stage. The scheme is second-order-accurate in time and includes the necessary terms to account for changes in cell areas or volumes due to a moving or deforming mesh. Furthermore, this explicit-scheme has a step size that is limited by the Courant-Friedricks-Lewy (CFL) condition. To accelerate convergence to steady-state, the CFL number may be increased by averaging implicitly the residual with values at neighboring grid points. These implicit equations are solved approximately using several Jacobi iterations. Convergence to steady-state is further accelerated by local time stepping and, when using a central-difference type spatial discretization, by enthalpy damping. Enthalpy damping is normally not used with upwind spatial discretizations since these discretizations generally do not preserve enthalpy.

Advantages of the explicit type time integration are that it is simple to code and requires far less memory than an implicit time integration. Furthermore, when a central-difference spatial discretization is also employed, the convergence to steady state is fairly rapid and solutions are obtained with minimal computational resources. However, when finer meshes are used, either globally fine or locally fine due to spatial adaption, the rate of convergence slows dramatically. When an upwind spatial discretization is employed, the rate of convergence also is poor. What is required typically for rapid convergence on fine meshes is a multigrid strategy. Alternatively, an implicit temporal discretization may be used, especially since an implicit time integration is generally desirable for unsteady applications.

Implicit Algorithm

The implicit algorithm that has been developed for use on unstructured meshes is based on the cell-centered, upwind, spatial discretization. The algorithm is derived in general by first linearizing the flux vector H according to

$$H^{n+1} = H^n + \frac{\partial H}{\partial Q}\Delta Q \tag{6}$$

where $\partial H/\partial Q$ is the flux Jacobian A, as discussed before, and $\Delta Q = Q^{n+1} - Q^n$. Linearizing both flux terms on the right hand side of Eq. (5) using Eq. (6), and ignoring the tilde on the flux Jacobian, results in

$$\left[\frac{\text{vol}}{\Delta t}I + \sum_{m=1}^{4} A^+(Q_j)\Delta S\right]\Delta Q_j + \sum_{m=1}^{4} A^-(Q_m)\Delta S\Delta Q_m = -\frac{1}{2}\sum_{m=1}^{4}\left[H(Q^+) + H(Q^-) - \left|\tilde{A}\right|(Q^+ - Q^-)\right]^n \Delta S \tag{7}$$

where I is the identity matrix, "vol" is the volume of the tetrahedron j, and ΔQ_m is the change in flow variables in each of the four tetrahedra adjacent to tetrahedron j. Also in Eq. (7) A^+ and A^- are forward and backward flux Jacobians, respectively. For flux-difference splitting, the exact Jacobian A (derivative of the right-hand side of Eq. (5) with respect to Q) is too expensive to compute and thus an approximate Jacobian is normally used. This is accomplished by constructing the Jacobians making use of the fact that the forward and backward Jacobians should have non-negative and non-positive eigenvalues (characteristic speeds), respectively.

A similar implicit temporal discretization, that is more efficient than the discretization of Eq. (7), is derived by linearizing the flux vector of the quasi-linear form of the Euler equations with respect to the primitive flow variables. This approach results in an equation similar to that of Eq. (7) given by

$$\left[\frac{\text{vol}}{\Delta t}I + \sum_{m=1}^{4} a^+(q_j)\Delta S\right]\Delta q_j + \sum_{m=1}^{4} a^-(q_m)\Delta S\Delta q_m = -\frac{B^{-1}}{2}\sum_{m=1}^{4}\left[H(Q^+) + H(Q^-) - \left|\tilde{A}\right|(Q^+ - Q^-)\right]^n \Delta S \tag{8}$$

where the matrix B relates the time derivative of Q to the time derivative of q as simply

$$\frac{\partial Q}{\partial t} = B\frac{\partial q}{\partial t} \tag{9}$$

The discretization represented by Eq. (8) is more efficient than the discretization represented by Eq. (7) because the flux Jacobians a^+ and a^- are simpler mathematically and therefore faster to compute than the flux Jacobians A^+ and A^-.

Direct solution of the system of simultaneous equations which results from application of Eq. (8) for all tetrahedra in the mesh requires the inversion of a large matrix with large bandwidth which is computationally expensive. Instead, a Gauss-Seidel relaxation approach is used to solve the equations whereby the summation involving Δq_m is moved to the right hand side of Eq. (8). The terms in this summation are then evaluated for a given time step using the most recently computed values for Δq_m. The solution procedure then involves only the inversion of a 5×5 matrix (represented by the terms in square brackets on the left hand side of Eq. (8)) for each tetrahedron in the mesh. The method is implemented by re-ordering the list of tetrahedra that make up the unstructured mesh from upstream to downstream, and the solution is obtained by sweeping two times through the mesh as dictated by stability considerations. The first sweep is performed in the direction from upstream to downstream and the second sweep is from downstream to upstream.

Advantages of the implicit type time integration are that it is numerically stable for very large CFL numbers, even on very fine meshes, and consequently enables rapid convergence to steady-state. Also, for unsteady applications the implicit discretization allows the selection of the step size based on the physical problem rather than on numerical stability considerations. This is in contrast with an explicit time integration which has a restrictive step size for unsteady applications which is more severe on finer meshes. A disadvantage of the implicit Gauss-Seidel relaxation procedure is that it requires twice the memory of the explicit Runge-Kutta integration, primarily due to having to store the backward flux Jacobian A^- (or a^-). Also, the linearization and relaxation errors associated with the implicit method may be too large for a given step size and thus contaminate the solution. To illustrate this problem, Fig. 7 shows the effects of step size on instantaneous pressure distributions using the Gauss-Seidel relaxation procedure. The calculations are for the same pitching NACA 0012 airfoil case presented in Fig. 6. Here though, three sets of results were obtained corresponding to using 250, 1000, and 2500 steps per cycle of motion. The figure shows the instantaneous pressure distribution at one instant of time (instantaneous angle of attack $\alpha(\tau) = 2.34°$ which is $69°$ ($k\tau$) into a cycle of motion). These results indicate that large errors in the strengths and locations of the shock waves on the upper and lower surfaces of the airfoil can occur when too large a time step is used (corresponding to 250 steps per cycle). However, when an appropriately small step size is used (corresponding to 2500 steps per cycle), the correct solution is obtained with a shock of moderate strength on the upper surface and subcritical flow (no shock wave) about the lower surface.

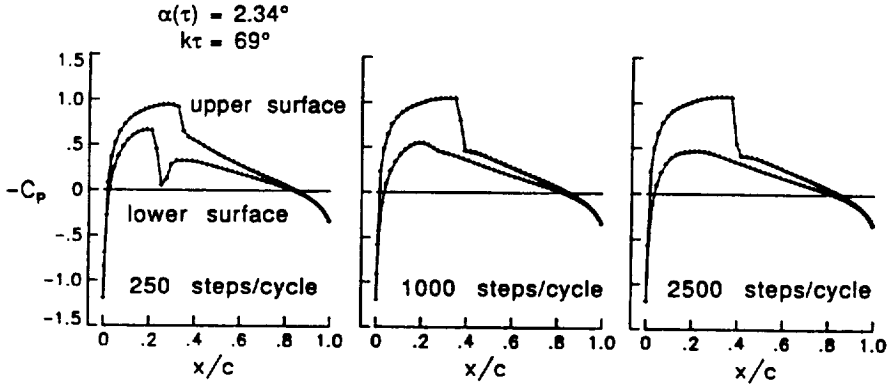

Fig. 7 Effects of step size on the instantaneous pressure distribution at $k\tau = 69°$ (corresponding to $\alpha(\tau) = 2.34°$) for the NACA 0012 airfoil pitching harmonically at $M_\infty = 0.755$, $\alpha_0 = 0.016°$, $\alpha_1 = 2.51°$, and $k = 0.0814$ computed using the cell-centered FVS algorithm.

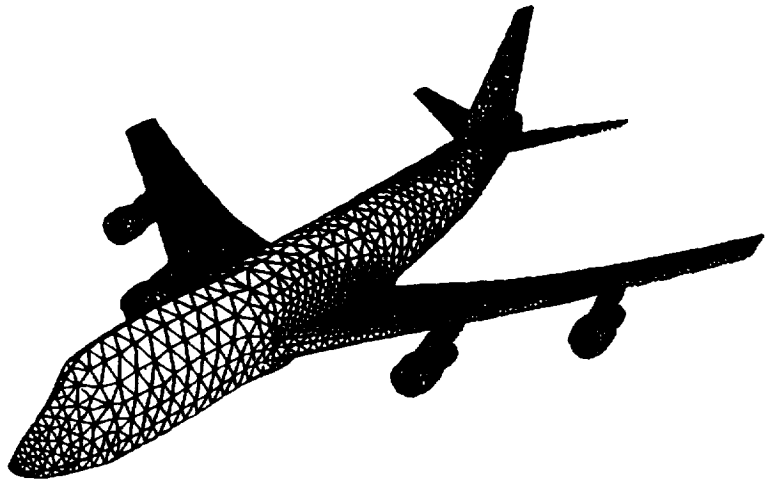

Fig. 8 Surface mesh of triangles for the Boeing 747 aircraft.

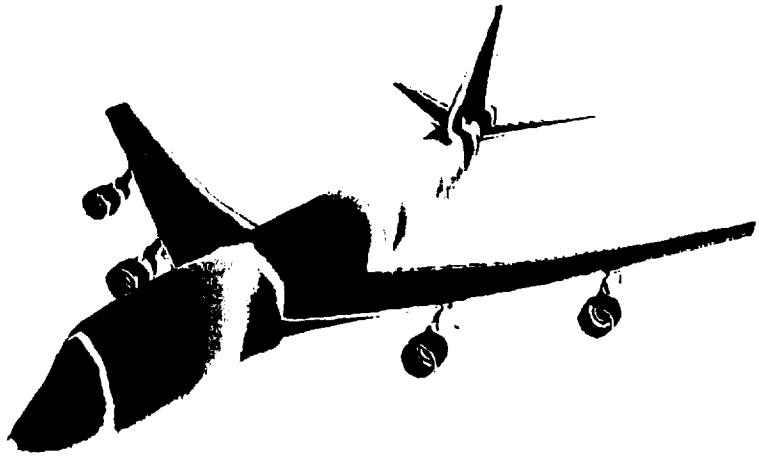

Fig. 9 Steady pressure coefficient contours on the Boeing 747 aircraft at $M_\infty = 0.84$ and $\alpha_0 = 2.73°$ computed using the implicit cell-centered FDS algorithm.

BOUNDARY CONDITIONS

To impose the flow tangency boundary conditions along solid surfaces in the cell-centered, upwind spatial discretization method, the flow variables are set within dummy cells that are effectively inside the geometry being considered. The velocity components within the dummy cells are determined from the values in the cell adjacent to the surface. This is accomplished by first rotating the components into a coordinate system that has a coordinate direction normal to the boundary face. The sign of the velocity component in this direction is changed (hence imposing no flow through the face) and the three velocity components are then rotated back into the original x, y, z coordinate system. Also, pressure and density within the dummy cell are set equal to the values in the cell adjacent to the surface.

After application of the upwind-biased interpolation formula to determine q^- and q^+ at each face, the velocity components are corrected to give a "strong" implementation of the surface boundary condition according to

$$
\begin{aligned}
u_{corrected} &= u - n_x(un_x + vn_y + wn_z) \\
v_{corrected} &= v - n_y(un_x + vn_y + wn_z) \\
w_{corrected} &= w - n_z(un_x + vn_y + wn_z)
\end{aligned}
\tag{10}
$$

In the far field a characteristic analysis based on Riemann invariants is used to determine the values of the flow variables on the outer boundary of the grid. This analysis correctly accounts for wave propagation in the far field, which is important for rapid convergence to steady state and serves as a "nonreflecting" boundary condition for unsteady applications.

RESULTS AND DISCUSSION

To demonstrate the speed and robustness of the cell-centered, upwind, implicit solution algorithm, calculations were performed for the Boeing 747 aircraft [21]. The results were obtained using the unstructured mesh shown in Fig. 8. The geometry includes the fuselage, the wing, horizontal and vertical tails, underwing pylons, and flow-through engine nacelles. The mesh contains 101,475 tetrahedra and 19,055 nodes for the half-span airplane. Also there are 4,159 nodes and 8,330 triangles on the boundaries of the mesh which include the airplane, the symmetry plane, and the far field. Steady-state calculations were performed for the aircraft at $M_\infty = 0.84$ and $\alpha_0 = 2.73°$. The results were obtained using a CFL number of infinity and the flux Jacobians were updated only every twenty iterations. The solution required approximately 3,420 CPU seconds (662 iterations) on the Cray-2 computer (Navier) at the Numerical Aerodynamic Simulation Facility, NASA Ames Research Center, to converge the solution to plotting accuracy (taken to be a four order-of-magnitude reduction in the L_2-norm of the density residual). The resulting steady pressure coefficient contours on the surface of the aircraft are shown in Fig. 9. The contours indicate that there is a significant amount of flow compression on the nose of the aircraft, along the inboard leading edge of the wing, and inside the cowl of the engine nacelles. There is flow expansion on the forward fuselage, on the horizontal and vertical tail surfaces, and on the upper surface of the wing terminated by a shock wave.

To demonstrate the accuracy of the cell-centered, upwind, implicit solution algorithm, calculations were performed for the ONERA M6 wing [22]. The results were obtained using three meshes of increasing fineness as shown in Fig. 10. The meshes, referred to as Mesh 1, 2, and 3, have 47,344, 98,317, and 288,170 tetrahedra, respectively. Steady-state calculations were performed for the wing at $M_\infty = 0.84$ and $\alpha_0 = 3.06°$ using a CFL number of infinity. The resulting steady pressure coefficients are shown in Fig. 11 including comparisons with experimental data [28]. For this case, a double shock wave occurs on the upper surface of the wing which coalesces into a single, relatively strong shock near the wing tip. Results for all three meshes show good agreement with each other aft of the 70 percent chord line on the upper surface, and aft of the 25 percent chord line on the lower surface. As is expected, the pressure coefficients obtained using Mesh 3 show the sharpest resolution of the shock waves and agree best with the experimental data.

CONCLUDING REMARKS

The paper summarized the development by the author of unstructured-grid methods for the solution of the equations of fluid flow and shared what has been learned over the course of the research. The focus of the discussion was on the solution of the time-dependent Euler equations including central-difference and upwind-type spatial discretizations, explicit and implicit temporal discretizations, and strong and weak implementations of boundary conditions. A best choice of methods, in the opinion of the author, is a cell-centered, upwind, implicit solution algorithm. An example calculation with this algorithm, using a CFL number of infinity, was presented for the Boeing 747 aircraft. The results were obtained in less than one hour CPU time on a Cray-2 computer, thus demonstrating the speed and robustness of the capability. Additional calculations for the ONERA M6 wing, also obtained using a CFL number of infinity, demonstrated the accuracy of the method through the good agreement between the calculated results and experimental data for a standard transonic flow case. Further results for unsteady

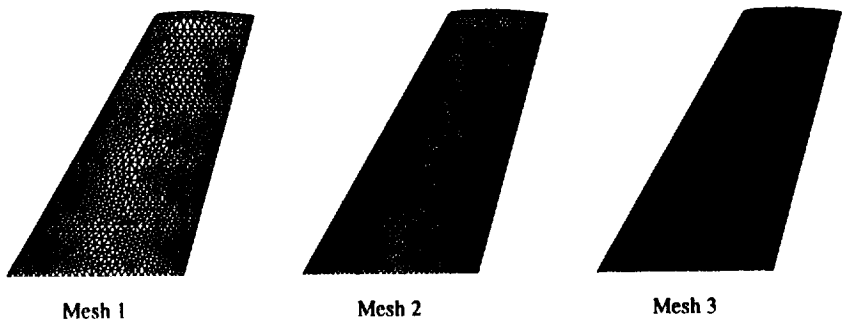

Fig. 10 Surface triangularizations from the ONERA M6 wing meshes.

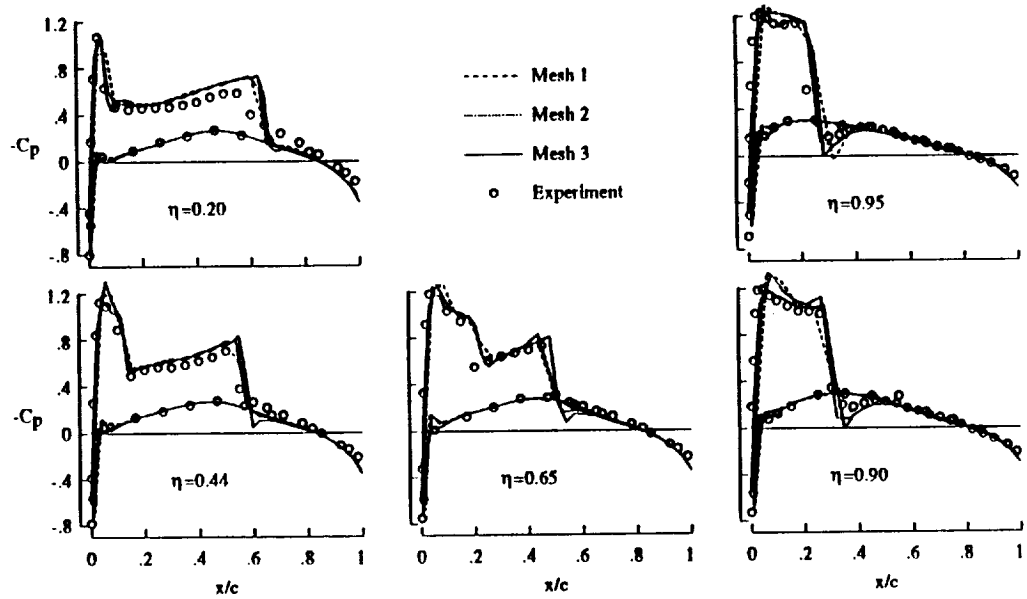

Fig. 11 Comparison of calculated and measured steady pressure coefficients on the ONERA M6 wing at $M_\infty = 0.84$ and $\alpha_0 = 3.06°$ computed using the implicit cell-centered FDS algorithm.

aerodynamic and aeroelastic cases which demonstrate time-dependent applications of the methods are presented in refs. [11–16, 19, 20, 23, 24].

REFERENCES

1. A. Jameson, Successes and Challenges in Computational Aerodynamics, AIAA Paper No. 87–1184, January 1987.
2. J. W. Edwards, and J. L. Thomas, Computational Methods for Unsteady Transonic Flows, AIAA Paper No. 87–0107, January 1987.
3. A. Jameson, and D. J. Mavriplis, Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh, AIAA Journal, vol. 24, April 1986, pp. 611–618.
4. D. J. Mavriplis, Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes, AIAA Journal, vol. 26, July 1988, pp. 824–831.
5. A. Jameson, T. J. Baker, and N. P. Weatherill, Calculation of Inviscid Transonic Flow Over a Complete Aircraft, AIAA Paper No. 86–0103, January 1986.
6. K. Morgan, and J. Peraire, Finite Element Methods for Compressible Flow, Von Karman Institute for Fluid Dynamics Lecture Series 1987–04, Computational Fluid Dynamics, March 2–6, 1987.

7. R. Lohner, Finite Elements in CFD: What Lies Ahead, International Journal of Numerical Methods in Engineering, vol. 24, 1987, pp. 1741–1756.
8. K. Morgan, J. Peraire, R. R. Thareja, and J. R. Stewart, An Adaptive Finite Element Scheme for the Euler and Navier-Stokes Equations, AIAA Paper No. 87–1172, 1987.
9. J. Peraire, J. Peiro, L. Formaggia, and K. Morgan, Finite Element Euler Computations in Three Dimensions, AIAA Paper No. 88–0032, January 1988.
10. N. T. Frink, P. Parikh, and S. Pirzadeh, A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes, AIAA Paper No. 91–0102, January 1991.
11. J. T. Batina, Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes, AIAA Paper No. 89–0115, January 1989.
12. R. D. Rausch, J. T. Batina, and H. T. Y. Yang, Euler Flutter Analysis of Airfoils Using Unstructured Dynamic Meshes, AIAA Paper No. 89–1384, April 1989.
13. J. T. Batina, Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aeroelastic Analysis, AIAA Paper No. 89–1189, April 1989.
14. J. T. Batina, Vortex-Dominated Conical-Flow Computations Using Unstructured Adaptively-Refined Meshes, AIAA Paper No. 89–1816, June 1989.
15. E. M. Lee, and J. T. Batina, Conical Euler Solution for a Highly-Swept Delta Wing Undergoing Wing-Rock Motion, NASA TM–102609, March 1990.
16. J. T. Batina, Implicit Flux-Split Euler Schemes for Unsteady Aerodynamic Analysis Involving Unstructured Dynamic Meshes, AIAA Paper No. 90–0936, April 1990.
17. J. T. Batina, Accuracy of an Unstructured-Grid Upwind-Euler Algorithm for the ONERA M6 Wing, presented at the Accuracy of Unstructured Grid Techniques Workshop, NASA Langley Research Center, Hampton, Virginia, January 16–17, 1990.
18. J. T. Batina, Three-Dimensional Flux-Split Euler Schemes Involving Unstructured Dynamic Meshes, AIAA Paper No. 90–1649, June 1990.
19. W. L. Kleb, J. T. Batina, and M. H. Williams, Temporal Adaptive Euler/Navier Stokes Algorithm for Unsteady Aerodynamic Analysis of Airfoils Using Unstructured Dynamic Meshes, AIAA Paper No. 90–1650, June 1990.
20. R. D. Rausch, J. T. Batina, and H. T. Y. Yang, Spatial Adaption Procedures on Unstructured Meshes for Accurate Unsteady Aerodynamic Flow Computation, AIAA Paper No. 91–1106, April 1991.
21. J. T. Batina, A Fast Implicit Upwind Solution Algorithm for Three-Dimensional Unstructured Dynamic Meshes, AIAA Paper No. 92–0447, January 1992.
22. P. R. Woodard, J. T. Batina, and H. T. Y. Yang, Quality Assessment of Two- and Three-Dimensional Unstructured Meshes and Validation of an Upwind Euler Flow Solver, AIAA Paper No. 92–0444, January 1992.
23. R. D. Rausch, J. T. Batina, and H. T. Y. Yang, Three-Dimensional Time-Marching Aeroelastic Analyses Using an Unstructured-Grid Euler Method, AIAA Paper No. 92–2506, April 1992.
24. J. R. Hooker, J. T. Batina, and M. H. Williams, Spatial and Temporal Adaptive Procedures for the Unsteady Aerodynamic Analysis of Airfoils Using Unstructured Meshes, AIAA Paper No. 92–2694, June 1992.
25. D. G. Holmes, and S. D. Connell, Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids, AIAA Paper No. 89–1392, June 1989.
26. B. Van Leer, Flux-Vector Splitting for the Euler Equations, Lecture Notes in Physics, vol. 170, 1982, pp. 507–512.
27. P. L. Roe, Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, Journal of Computational Physics, vol. 43, 1981, pp. 357–372.
28. V. Schmidt, and F. Charpin, Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers, Appendix B1 in AGARD-AR-138, Experimental Data Base for Computer Program Assessment, May 1979.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1992	Technical Memorandum

4. TITLE AND SUBTITLE
Progress in Unstructured-Grid Methods Development for Unsteady Aerodynamic Applications

5. FUNDING NUMBERS
505-63-50-12

6. AUTHOR(S) John T. Batina

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23665-5225

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA TM 107643

11. SUPPLEMENTARY NOTES
Invited Paper Presented at the 7th IMACS International Conference on Computer Methods for Partial Differential Equations, Rutgers University, New Brunswick, New Jersey, June 22-24, 1992

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 02

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)* The paper summarizes the development by the author of unstructured-grid methods for the solution of the equations of fluid flow and shares what has been learned over the course of the research. The focus of the discussion is on the solution of the time-dependent Euler equations including spatial discretizations, temporal discretizations, and boundary conditions. An example calculation with an implicit upwind method using a CFL number of infinity is presented for the Boeing 747 aircraft. The results were obtained in less than one hour CPU time on a Cray-2 computer, thus demonstrating the speed and robustness of the capability. Additional calculations for the ONERA M6 wing demonstrate the accuracy of the method through the good agreement between calculated results and experimental data for a standard transonic flow case.

14. SUBJECT TERMS
Computational Fluid Dynamics
Unsteady Aerodynamics
Transonic Aerodynamics

15. NUMBER OF PAGES
12

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	