Streamwise Upwind Algorithm for Computing Unsteady Transonic Flows Past Oscillating Wings

Shigeru Obayashi,* Guru P. Guruswamy,† and Peter M. Goorjian†
NASA Ames Research Center, Moffett Field, California 94035

A streamwise upwind algorithm has been extended to compute unsteady flows with a moving grid system and applied to compute flows over oscillating wings at transonic Mach numbers. Comparisons have been made between results obtained from this upwind algorithm, using both temporally nonconservative- and conservative-implicit methods, with the results obtained from a central-difference method, and also with experimental data. The efficiency and practicality of the temporally nonconservative implicit solver are illustrated. Results to show the robustness and accuracy of the upwind method for unsteady computations compared to the central-difference method are presented.

Introduction

IN the last two decades, there have been extensive developments in computational aerodynamics, which constitutes a major part of the general area of computational fluid dynamics. Such developments are essential to advance the understanding of the physics of complex flows, to complement expensive wind-tunnel tests, and to reduce the overall design cost of an aircraft, particularly in the area of aeroelasticity.

Aeroelasticity plays an important role in the design and development of aircraft, particularly modern aircraft, which tend to be more flexible. Several phenomena that can be dangerous and limit the performance of an aircraft occur because of the interaction of the flow with flexible components. For example, an aircraft with highly swept wings may experience vortex-induced aeroelastic oscillations.[1] Also, undesirable aeroelastic phenomena due to the presence and movement of shock waves occur in the transonic range. Aeroelastically critical phenomena, such as a low transonic flutter speed, have been known to occur through limited wind-tunnel tests and flight tests.

Aeroelastic tests require extensive cost and risk. An aeroelastic wind-tunnel experiment is an order of magnitude more expensive than a parallel experiment involving only aerodynamics. By complementing the wind-tunnel experiments with numerical simulations, the overall cost of the development of aircraft can be considerably reduced. To accurately compute aeroelastic phenomenon it is necessary to solve the unsteady Euler/Navier-Stokes equations simultaneously with the structural equations of motion.

At Ames a code, ENSAERO, is being developed for computing the unsteady aerodynamics and aeroelasticity of aircraft, and it solves the Euler/Navier-Stokes equations. The capability of the code has been demonstrated by computing vortical and transonic flows over flexible swept wings,[2,3] The flowfields were calculated by a time-accurate, finite-difference scheme based on central differencing.

The motivation of this study is to enhance the algorithm capability of the present code. Toward this goal, the use of an upwind scheme is investigated in comparison with the current central-difference (CD) scheme. The CD scheme requires artificial dissipation to stabilize computations. In general, such artificial-dissipation models lead to more dissipative, and thus less accurate, solutions than upwind schemes. In addition, the CD scheme is sensitive to the amount of dissipation and it is necessary to specify a dissipation coefficient on a case-by-case basis. On the other hand, upwind schemes do not require any coefficient to be specified.

Upwind algorithms are important in computational fluid dynamics for calculations of flows containing shock waves[4] and also for accurately resolving shear layers.[5,6] Most multidimensional upwind algorithms are first constructed in one dimension and then extended to multidimensions by applying the one-dimensional procedure to each coordinate direction. The resulting algorithms are less accurate for resolving shock waves that are inclined with respect to grid lines. They also have difficulties for solving flows with embedded shear layers. To increase the accuracy, an upwind algorithm should model the physical features of the flow by applying upwind formulas along the streamwise direction. Then, the switching of flux evaluations would take place at sonic values, as is done in algorithms for the full-potential equation.[7]

A streamwise upwind algorithm was previously developed and applied to transonic and vortical flowfields.[8-11] The reader may refer to Refs. 8 and 11 for the derivation of the formulas, which are generalizations to the Euler equations of the algorithm for the full-potential equation. The main features of the algorithm are the use of rotated differencing to enforce upwinding in the streamwise direction and the use of flux-vector splitting (FVS) formulas for the upwinding that smoothly switch between subsonic and supersonic regions of the flow. The latter feature is analogous to the differentiability of Osher's upwind algorithm.[12]

Another way of constructing upwind techniques for the Euler equations is called flux-difference splitting (FDS), one example being Roe's upwind algorithm.[4] In general, the FVS has an advantage over the FDS because of its simplicity. On the other hand, for the crossflow direction or contact discontinuities, the FDS has better resolution than the FVS.[5] Now, Goorjian's FVS[8] can capture shock waves as sharply as Roe's FDS. In addition, Goorjian's formula does not require an entropy correction that is required by Roe's formula. The streamwise upwind algorithm with Goorjian's FVS formula does not smear solutions for either contact discontinuities[8] or shear flows.

The original streamwise upwind algorithm may cause solution decoupling in the supersonic region or instability at

Presented as Paper 90-3103 at the AIAA 8th Applied Aerodynamics Conference, Portland, OR, Aug. 20-22, 1990; received July 2, 1990; revision received Dec. 14, 1990; accepted for publication Dec. 26, 1990.

*Research Scientist, MCAT Institute. Member AIAA.
†Research Scientist. AIAA Associate Fellow.

contact discontinuities because it uses central differencing for the crossflow direction. Further development of the algorithm was made to prevent these instabilities.[10,11] It was found that the enforcement of physical conditions at a contact discontinuity, i.e., pressure and velocity continuity across the discontinuity, leads to a sufficient dissipation model. The additional terms based on pressure and velocity continuity can be obtained by simplifying the FDS algorithm. Therefore, an improved streamwise upwind algorithm has been developed that combines important features of both FVS and FDS.

The streamwise upwind algorithm has an advantage in capturing normal shock waves independently of their alignment with grid lines because the formulas switch depending on the Mach number only, not on the coordinate Mach number. For sharply capturing oblique shock waves in supersonic regions, the algorithm requires the use of a modified rotation angle, which is determined by taking into account the region of the Mach cone in addition to the cosine of the stream direction.[11] The additional operations required to compute the rotated differencing are compensated by the simplicity of the FVS formula. The resulting algorithm requires arithmetic operations similar to those of the FDS methods.

In this work, the streamwise upwind algorithm has been further extended to compute unsteady flows associated with moving grids. The resulting algorithm has been implemented into the code using a finite-volume spatial discretization. This method and the CD method are both options in the present version of the code. The new option of the code has been successfully applied for computing unsteady transonic flows over oscillating wings. The results demonstrate that the upwind scheme improves the accuracy of both steady and unsteady pressure predictions with shock waves.

Governing Equations and Numerical Algorithms

The present streamwise upwind algorithm consists of three key features: FVS, FDS, and the rotated differencing. The first two features are the basic upwind techniques and those concepts can be explained in the one dimension. The third one is associated with the multidimensionality. In the following, the basic upwind concepts are described by using the one-dimensional Euler equations first. Then the complete three-dimensional formulas are presented with a discussion about the rotated differencing. Finally, the implicit methods are discussed. At the end, the CD method is briefly described for comparison purposes.

FVS and FDS Algorithm

The one-dimensional Euler equations can be written as

$$Q_t + F_x = 0 \tag{1}$$

where $Q = (\rho,\rho u,e)^T$ with density ρ, velocity u, and total energy per unit volume e; and where $F = (\rho u,\rho u^2 + p,\rho uH)^T$ with pressure p and total enthalpy H. The pressure is related to Q through the equation of state for a perfect gas:

$$p = (\gamma - 1)(e - \rho u^2/2) \tag{2}$$

where $\gamma = 1.4$ is the ratio of specific heats. The cell-centered space-discretized form of Eq. (3) can be written as

$$\partial_t Q + \frac{F_{j+1/2} - F_{j-1/2}}{\Delta x} = 0 \tag{3}$$

where $F_{j\pm 1/2}$ are numerical fluxes at cell interfaces. Upwind algorithms can be described in the way of evaluating the cell-interface fluxes (for example, see Ref. 4). For example, if $u > a$, where a is the speed of sound, a first-order accurate cell-interface flux is simply given by $F_{j+1/2} = F_j$. This results in the backward differencing of the entire flux F. However, upwind formulas become more complicated when $|u| < a$.

For subsonic flows, Goorjian's FVS algorithm[8] can be written for $F_{j+1/2} = F(Q_l,Q_r)$ as

$$F(Q_l,Q_r) = \frac{1}{2}\left\{F_r + F_l - \int_{Q_l}^{Q_r} \frac{\partial^* F}{\partial Q}\, dQ\right\} \tag{4}$$

where Q_l and Q_r are left and right states across the cell interface, respectively, and $l = j$ and $r = j + 1$ give the basic first-order formula. Higher-order extension can be done by using the standard MUSCL approach.[13] The integral part was derived as follows for $u \geq 0$:

$$\int_{Q_l}^{Q_r} \frac{\partial^* F}{\partial Q}\, dQ = \{F + \Delta^* F\}_r - \{F + \Delta^* F\}_l \tag{5}$$

where "*" indicates local sonic values and $\Delta^* F = F^* - F$. Using the local isentropic relations, $\Delta^* F$ can be simplified as

$$\Delta^* F = \Delta^*(\rho u)\boldsymbol{e}_s \tag{6}$$

where $\boldsymbol{e}_s = (1,u,H)^T$ is a sum of two eigenvectors of $\partial F/\partial Q$. Note $\Delta^*(\rho u) > 0$ for $M < 1$ and $\Delta^*(\rho u) = 0$ for $M = 1$. This formula is equivalent to the FDS formula, if the flow is isentropic. The formula is much simpler than other FDS formulas because the flux difference is represented by the scalar difference. If the flow is not isentropic, then the algorithm can be interpreted as an FVS algorithm. Even in the latter case, the use of isentropic relations does not restrict the flowfields because the isentropic relation is used only locally at each grid point (or cell) to compute local sonic values in space and time. This formula can be directly extended to the three-dimensional case along the streamwise direction.

The above FVS formula is simple and accurate for sonic points but cannot handle contact discontinuities properly. To construct a counterpart flux formula, take the FDS algorithm of Roe[4-6]

$$F(Q_l,Q_r) = \tfrac{1}{2}\{F_r + F_l - |A|(Q_r - Q_l)\} \tag{7}$$

where $A = A(Q_l,Q_r)$ is the Roe-averaged Jacobian matrix. Assuming that u, Δp, and Δu are small and neglecting the resultant higher-order terms in the product of the wave strength and its speed for each wave,[11] Eq. (7) can be simplified as

$$|A|\Delta Q \simeq |u|\Delta Q + (a - |u|)\left\{\frac{\Delta p}{a^2}\boldsymbol{e}_s + \rho\Delta u\, \boldsymbol{e}_u\right\} \tag{8}$$

where $\Delta \cdot = \cdot_r - \cdot_l$ and $\boldsymbol{e}_u = (0,1,u)^T$. The variables are averaged unless defined as a difference between left and right states. A simple arithmetic average is taken for ρ, u, and H instead of Roe's average because these terms are not used across normal shock waves. Note that p and u are to be continuous for contact discontinuities ($u = 0$) and that Eq. (8) enforces those continuities.

The present formula is written in a combination of FVS and FDS formulas as

$$\begin{aligned} F(Q_l,Q_r,S_{j+1/2}) = \tfrac{1}{2}\{&F_l + F_r - \vartheta[F_r\text{sign}(u_r) \\ &+ s_r\Delta^*(\rho u)_r\boldsymbol{e}_{sr} - F_l\text{sign}(u_l) - s_l\Delta^*(\rho u)_l\boldsymbol{e}_{sl}] \\ &- (1 - \vartheta)[|u|\Delta Q + (c - |u|)\{(\Delta p/c^2)\boldsymbol{e}_s + \rho\Delta u\boldsymbol{e}_u\}]\} \end{aligned} \tag{9}$$

where sign(u) gives the sign of velocity and ϑ is a weight of the FVS formula. In the one-dimensional case, the square of the Mach number, $M^2 = u^2/a^2$, at the averaged state can be used to determine the parameter ϑ as $\vartheta = \min(M^2,1)$. In the multidimensional case, the parameter θ is replaced with $\cos^2\theta$ where θ is the rotation angle, and $1 - \vartheta$ is replaced with $\sin^2\theta$. This rotation angle will be discussed later. Although Eq. (9) has a lengthy expression, it has only two additional

vectors, e_s and e_d, besides Q and F. Thus the formula is not very expensive computationally.

Following Refs. 7 and 8, the switches s_l and s_r are defined in the manner of Godunov's method as follows. For $u \geq 0$,

$$s_l = 1 - \varepsilon_l \varepsilon_m \qquad s_r = (1 - \varepsilon_m)(1 - \varepsilon_r) \tag{10}$$

where $\varepsilon_{l,m,r} = \{1 + \text{sign}(M^2_{l,m,r} - 1)\}/2$ and M_m denotes the Mach number of the averaged state. Considering a special case of a supersonic flow with $u > 0$, one obtains $s_l = s_r = 1$, $\vartheta = 1$, and $\text{sign}(u_l) = \text{sign}(u_r) = 1$. Then the formula results in $F(Q_l,Q_r) = F_l$. Applying this to Eq. (3) with $l = j$, the present formula coincides with the standard upwind formula that uses the backward differencing of the entire flux F. Note that the switches, Eqs. (10), will use the Mach number even in the multidimensional case, rather than the coordinates Mach number. Thus, with the use of the rotated differencing, this streamwise upwind algorithm can capture normal shock waves independently of their alignment to grid lines.

Three-Dimensional Navier-Stokes Equations

The nondimensionalized thin-layer Navier-Stokes equations used in this study can be written in conservation-law form in a generalized body-conforming curvilinear coordinate system as follows:

$$\partial_\tau \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} + \partial_\zeta \hat{G} = (1/Re)\partial_\zeta \hat{G}^v \tag{11}$$

where $\tau = t$, $\xi = \xi(x,y,z,t)$, $\eta = \eta(x,y,z,t)$, and $\zeta = \zeta(x,y,z,t)$. In the present paper, the ξ and η directions are along the streamwise and spanwise directions of a wing, respectively. The viscous derivatives associated with these directions are dropped. In contrast, the ζ direction is normal to the wing surface, and thus the viscous derivatives are retained.

The vector of conserved quantities $\hat{Q}$ and the inviscid flux vector $\hat{F}$ are

$$\hat{Q} = \frac{1}{J}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \qquad \hat{F} = \frac{1}{J}\begin{bmatrix} \rho\hat{V} \\ \rho u\hat{V} + \eta_x p \\ \rho v\hat{V} + \eta_y p \\ \rho w\hat{V} + \eta_z p \\ \rho H\hat{V} - \eta_t p \end{bmatrix} \tag{12a}$$

where H is the total enthalpy and the contravariant velocity component $\hat{V}$ is defined as $\hat{V} = \eta_t + \eta_x u + \eta_y v + \eta_z w$. The Cartesian velocity components u, v, and w are nondimensionalized by the freestream speed of sound a_∞; the density ρ is nondimensionalized by the freestream density ρ_∞; the total energy per unit volume e is nondimensionalized by $\rho_\infty a_\infty^2$. For the ξ and ζ directions, $\hat{E}$ and $\hat{G}$ can be defined similarly. The viscous flux vector $\hat{G}^v$ is given by

$$\hat{G}^v = \frac{1}{J}\begin{bmatrix} 0 \\ \mu m_1 u_\zeta + \frac{\mu}{3} m_2 \zeta_x \\ \mu m_1 v_\zeta + \frac{\mu}{3} m_2 \zeta_y \\ \mu m_1 w_\zeta + \frac{\mu}{3} m_2 \zeta_z \\ \mu m_1 m_3 + \frac{\mu}{3} m_2(\zeta_x u + \zeta_y v + \zeta_z w) \end{bmatrix} \tag{12b}$$

with

$$m_1 = \zeta_x^2 + \zeta_y^2 + \zeta_z^2 \tag{12c}$$

$$m_2 = \zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta \tag{12d}$$

$$m_3 = \tfrac{1}{2}(u^2 + v^2 + w^2)_\zeta + \frac{1}{Pr(\gamma - 1)}(a^2)_\zeta \tag{12e}$$

where Re is the Reynolds number, Pr is the Prandtl number, a is the speed of sound, and J is the transformation Jacobian. Pressure is related to the conservative flow variables $\hat{Q}$ through the equation of state for a perfect gas:

$$p = (\gamma - 1)\{e - (\rho/2)(u^2 + v^2 + w^2)\} \tag{13}$$

where ρ is the fluid density and e is total energy per unit of volume of the fluid. See Ref. 3 for detailed definitions.

For inviscid flows, the viscous flux $\hat{G}^v$ is replaced by 0. For viscous flows, the viscosity coefficient μ in $\hat{G}^v$ is computed as the sum of $(\mu_l + \mu_t)$, where the laminar viscosity μ_l is taken from the freestream laminar viscosity, assumed to be constant for transonic flows, and the turbulent viscosity, μ_t is evaluated by the Bladwin-Lomax algebraic eddy-viscosity model.[14]

Space Discretization

The space-discretized form of Eq. (11) can be written as

$$\partial_\tau \hat{Q} = -\frac{\hat{E}_{i+1/2} - \hat{E}_{i-1/2}}{\Delta\xi} - \frac{\hat{F}_{j+1/2} - \hat{F}_{j-1/2}}{\Delta\eta} - \frac{\hat{G}_{k+1/2} - \hat{G}_{k-1/2}}{\Delta\zeta} + \frac{1}{Re}\frac{\hat{G}^v_{k+1/2} - \hat{G}^v_{k-1/2}}{\Delta\zeta} \tag{14}$$

where a second-order central-difference evaluation is applied to the viscous term.

The evaluation of the inviscid fluxes is based on the finite-volume cell-centered scheme. To be consistent with the finite-difference scheme in ENSAERO, the metrics are defined at each grid point where the flow variables are stored. The surface vector of each cell interface, which is necessary for the finite-volume formulation, can be obtained by averaging the metrics at the adjoining points. The freestream preservation of this metric evaluation was shown for fixed grids in Ref. 15.

For a general motion of the grid, the time-metric terms used here will not preserve the freestream.[16] The error due to the present time-metric terms, however, was found to be completely negligible in the present applications by comparing the computed results obtained with and without the freestream subtraction.

Now the streamwise upwind algorithm is described for the inviscid fluxes at cell interfaces.

Streamwise Upwind Algorithm on Moving Coordinates

To extend the streamwise upwind algorithm from a fixed grid system to a moving grid system, the flow velocity relative to the fixed grid (inertial frame), $\boldsymbol{q} = (u,v,w)$, is redefined as the flow velocity measured relative to the moving grid, $\tilde{\boldsymbol{q}} = (u - x_t, v - y_t, w - z_t)$. This is consistent with the modification of the definition of the contravariant velocity from a fixed system. For example,

$$\begin{aligned} \hat{V}_{\text{fixed}} &= \eta_x u + \eta_y v + \eta_z w \\ &= \nabla\eta \cdot \boldsymbol{q} \end{aligned} \tag{15}$$

For a moving system,

$$\begin{aligned} \hat{V}_{\text{moving}} &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\ &= \eta_x(u - x_t) + \eta_y(v - y_t) + \eta_z(w - z_t) \\ &= \nabla\eta \cdot \tilde{\boldsymbol{q}} \end{aligned} \tag{16}$$

where $\eta_t = -\eta_x x_t - \eta_y y_t - \eta_z z_t$. Then the formulas for a fixed system can be rewritten for a moving system. Note that the present algorithm uses x_t, y_t, and z_t to obtain the flow velocity for upwinding in the streamwise direction, instead of ξ_t, η_t, and ζ_t to compute the contravariant velocity for upwinding in the coordinate direction.

The complete algorithm can be summarized in component form with a surface vector $S = \nabla\eta/J$ and a motion of its centroid, $x_t = (x_t, y_t, z_t)$, as

$$\hat{F}(Q_l, Q_r, S_{j+1/2}, x_{\eta_{j+1/2}}) = (|\nabla\eta|/J)(\tfrac{1}{2}[F_l^+ + F_r^-] - k_t Q_\infty) \quad (17)$$

where

$$F_{l,r}^{\pm} = \begin{bmatrix} f_{l,r}^{\pm} \\ f_{l,r}^{\pm} u_{l,r} + k_x p_{l,r}^{\pm} \\ f_{l,r}^{\pm} v_{l,r} + k_y p_{l,r}^{\pm} \\ f_{l,r}^{\pm} w_{l,r} + k_z p_{l,r}^{\pm} \\ f_{l,r}^{\pm} H_{l,r} - k_t p_{l,r}^{\pm} + \tfrac{1}{2}(|\tilde{V}_m|\Delta p - \tilde{V}_m \Delta_2)\sin^2\theta \end{bmatrix} \quad (18a)$$

with

$$f_{l,r}^{\pm} = (\rho\tilde{V})_{l,r}\{1 \pm \text{sign}(\tilde{V}_{l,r})\cos^2\theta_{l,r}\}$$
$$\pm s_{l,r}\Delta^*(\rho\tilde{q})_{l,r}\cos^2\theta_{l,r} \pm \rho_{l,r}|\tilde{V}_m|\sin^2\theta - \tfrac{1}{2}\Delta_1\sin^2\theta \quad (18b)$$

$$p_{l,r}^{\pm} = p_{l,r}\{1 \pm \text{sign}(\tilde{V}_{l,r})\cos^2\theta_{l,r}\} - \tfrac{1}{2}\Delta_2\sin^2\theta \quad (18c)$$

$$\tilde{V}_{l,m,r} = k_x(u_{l,m,r} - x_t) + k_y(v_{l,m,r} - y_t) + k_z(w_{l,m,r} - z_t) \quad (18d)$$

where $\tilde{q}^2 = (u - x_t)^2 + (v - y_t)^2 + (w - z_t)^2$ and $\Delta^*(\rho\tilde{q}) = \rho^*\tilde{q}^* - \rho\tilde{q}$ = local sonic values,

$$(\tilde{q}^*)^2 = [2/(\gamma + 1)]\{a^2 + [(\gamma - 1)/2]\tilde{q}^2\} \quad (18e)$$

$$\rho^* = \rho[(\tilde{q}^*)^2/a^2]^{1/(\gamma-1)} \quad (18f)$$

and where

$$\Delta_1 = (a_m - |\tilde{V}_m|)(\Delta\rho/a_m^2) \quad (18g)$$

$$\Delta_2 = (a_m - |\tilde{V}_m|)\rho_m\Delta V \quad (18h)$$

with $\Delta\cdot = \cdot_r - \cdot_l$, $V = k_x u + k_y v + k_z w$, $k_t = -k_x x_t - k_y y_t - k_z z_t$, $k_x = \eta_x/|\nabla\eta|$, and so on. The switches s_l and s_r are defined as in Eqs. (10) by using the modified Mach number, $\tilde{M} = \tilde{q}/a$. The term, $k_t Q_\infty$, subtracts the freestream for time metrics. The subscripts l, m, and r denote the left, averaged, and right state of the flow variables. The averaged state is defined for ρ, u, v, w, and H by the arithmetic average of the left and right states. The evaluation of $\theta_{l,r}$ will be discussed in the next section.

The basic scheme is first-order accurate with $l = j$ and $r = j + 1$. Higher-order schemes are constructed from a one-parameter family κ of interpolations of the primitive variables, ρ, u, v, w, and p. For example,

$$p_l = \{1 + (\psi_j/4)[(1 - \kappa)\nabla + (1 + \kappa)\Delta]\}p_j \quad (19a)$$

$$p_r = \{1 - (\psi_{j+1}/4)[(1 + \kappa)\nabla + (1 - \kappa)\Delta]\}p_{j+1} \quad (19b)$$

where ∇ and Δ are backward and forward difference operators, respectively.[13] For the third-order scheme, $\kappa = \frac{1}{3}$, Koren's differentiable limiter[17] is used in this paper. The limiter ψ is calculated as

$$\psi_j = \frac{3\nabla p_j \Delta p_j + \varepsilon}{2(\Delta p_j - \nabla p_j)^2 + 3\nabla p_j \Delta p_j + \varepsilon} \quad (20)$$

where a small constant ε, $\varepsilon = 10^{-6}$ typically, is added to prevent the division by zero. The same formulas are used for the other primitive variables.

Rotated Differencing

A simple example of the rotation angle is the use of cosine of velocity as $\cos\theta = \tilde{V}/\tilde{q}$ because the FVS part has the term associated with $\tilde{q}$. In supersonic flowfields, however, it is important to detect whether the velocity projected to the grid line is beyond the Mach cone. Thus, $\tilde{V}/\tilde{q}$ is replaced by $\tilde{M}\cdot\tilde{V}/\tilde{q} = \tilde{V}/a$. If $\tilde{V}/a$ becomes larger than one, $\cos\theta$ is frozen at one. The resulting formula is similar to the one-dimensional case. This enhances the ability to capture oblique shock waves.[11] Note that this form prevents dividing by zero when $\tilde{q} = 0$.

This evaluation of the rotated angle makes the present formula similar to the FDS formula in certain cases. For example, let us assume supersonic flow ($s_{l,r} = 0$) and $\tilde{U} > \tilde{V} > a$. Then even for the $\tilde{V}$ direction, the present formula gives the complete upwind differencing of the entire flux, as FDS does, because $\cos^2\theta = 1$. This feature leads to a favorable resolution of bow and crossflow shocks. However, it allows crossflow expansion shocks because the FVS part does not contribute ($s_{l,r} = 0$).

To avoid expansion shocks, the rotation angle is determined by a mixture of averaged (m) and pointwise (l,r) values:

$$\cos^2\theta_{l,r} = \min[(1 - \phi)\frac{\tilde{V}_m^2}{a_m^2} + \phi\frac{\tilde{V}_{l,r}^2}{a_{l,r}^2}, 1] \quad (21)$$

The following is used for evaluating ϕ in this paper:

$$\phi = \max\left[\frac{2\gamma}{\gamma + 1}\left(1 - \frac{1}{2\gamma}\left\{\gamma - 1 + (\gamma + 1)\frac{p_2}{p_1}\right\}\right), 0\right] \quad (22)$$

where $p_{1,2}$ denote upstream and downstream pressures, respectively. Note that the formula, $(1/2\gamma)\{\gamma - 1 + (\gamma + 1)(p_2/p_1)\}$, gives the square of the Mach number ahead of the shock wave (or the normal component of the Mach number ahead of the oblique shock).[18] The parameter ϕ is fixed at 0 for the compression $p_2/p_1 > 1$. For the expansion, ϕ increases as p_2/p_1 decreases. Finally, it reaches 1 for the expansion to the vacuum ($p_2 = 0$).

The sine is determined by an arithmetic average of the cosines: $\sin^2\theta = 1 - \frac{1}{2}(\cos^2\theta_l + \cos^2\theta_r)$.

Implicit Method

Implicit methods for upwind algorithms are under strong constraint in computational efficiency.[12,19,20] The main difficulty lies in the linearization of the numerical fluxes. An exact linearization is computationally formidable for time-marching methods, although it can be used for specific purposes, such as to achieve a quadratic convergence with a direct solver.[21] Another difficulty originates in the use of flux limiters for higher-order spatial accuracy. Yet implicit methods are preferred especially for viscous computations. Thus, practical implicit procedures for upwind algorithms can be constructed with the use of approximate linearizations (so-called approximate Jacobians) in the first-order accuracy in time.

An alternative approach is to use iterative methods.[12] However, iterations required in each time step can be compensated with the use of relatively small time-step sizes in noniterative methods. Also, the additional memory requirement could be critical for three-dimensional applications. Therefore, a noniterative first-order implicit method for the present upwind algorithm is considered in this research.

The computational efficiency of the method is critical for expensive unsteady calculations. Thus, the method chosen here is the lower-upper factored, alternating direction implicit (LU-ADI) method[15] that requires only scalar bidiagonal matrix inversions. In addition, this method uses the approximate Jacobian of the Steger-Warming flux-vector splitting[22] which is very robust when used in the implicit procedure. For the viscous computations, only the diagonal elements of the viscous Jacobians are taken into account. See Ref. 23 for further details.

The LU-ADI method is nonconservative in time due to the diagonalization. To investigate the significance of this tem-

poral nonconservativeness, an alternate approach is also considered and compared with the LU-ADI method. A time-conservative method can be constructed using a block tridiagonal solver.[12,19] Since true Jacobians of the numerical fluxes of the present upwind algorithm are expensive to compute, approximate Jacobians are used here. Reference 24 describes the formulations for the approximate Jacobians.

Similar to the results discussed in Ref. 20, this approach yields a method that is less stable than the LU-ADI method. In addition, this method requires twice the computational time used by the LU-ADI method. Note that both methods are first-order accurate in time even though the resulting block-ADI method is temporally conservative.

CD Method

To compare with the upwind algorithm, the second-order-accurate CD scheme with the artificial dissipation terms[25] is shown here as

$$\hat{F}(Q_j,Q_{j+1},S_{j+1/2},x_{t_{j+1/2}}) = \tfrac{1}{2}\{\hat{F}(Q_j,S_j,x_{t_j}) + \hat{F}(Q_{j+1},S_{j+1},x_{t_{j+1}})\} - \left(\frac{\sigma}{J}\right)_{j+1/2}\{\kappa_j^{(2)}(Q_{j+1} - Q_j) - \kappa_j^{(4)}(\delta^2 Q_{j+1} - \delta^2 Q_j)\} \quad (23)$$

where δ^2 is a central second-difference, σ is a sum of spectral radii of the Jacobian matrices of the inviscid fluxes, and $(\ )_{j+1/2} = \{(\ )_{j+1} + (\ )_j\}/2$. The parameters, $\kappa^{(2)}$ and $\kappa^{(4)}$, control the strength of the second- and fourth-order dissipation terms;

$$\kappa_j^{(2)} = (\varepsilon^{(2)}/4)(|\bar{\delta}^2 p_{j-1}| + 2|\bar{\delta}^2 p_j| + |\bar{\delta}^2 p_{j+1}|) \quad (24)$$

where

$$\bar{\delta}^2 p_j = \left(\frac{p_{j+1} - 2p_j + p_{j-1}}{p_{j+1} + 2p_j + p_{j-1}}\right) \quad (25)$$

and $\kappa_j^{(4)}$ and $\varepsilon^{(4)} - \min(\varepsilon^{(4)},\kappa_j^{(2)})$. Typical values for dissipation coefficients, $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$, are 0.25 and 0.01, respectively.[2,3,25,26] These values have been carefully tested for both steady and unsteady flow computations and thus are used here.

Note that Eq. (23) uses the difference of the vector of the conservative variables directly, while the present upwind formula, Eq. (17), uses the combination of differences, such as total mass flux, pressure, and contravariant velocity, as illustrated in Eq. (9).

The implicit part of the CD method uses the diagonal form of Beam-Warming method which is first-order accurate and nonconservative in time similar to the LU-ADI method. See Ref. 25 for further details.

Results

In this work, only the first-order time-accurate methods are considered because of computational efficiency. However, time accuracy is an essential requirement for aeroelastic computations. Numerical schemes used for flow calculations in aeroelasticity must guarantee the correct calculation of amplitude and phase of unsteady pressures. In order to verify the time accuracy of the present code, unsteady flows over typical rectangular and fighter wings undergoing prescribed oscillatory motions are computed.

The grids used here are generated algebraically in the C-H topology. The ξ, η, and ζ coordinates represent the chordwise, spanwise, and normal (to the wing surface) directions, respectively. Coarse and fine grids are used for both rectangular and F-5 wing cases.

Rectangular Wing

The first test considers inviscid and viscous unsteady flows over a rectangular wing with a NACA 64A010 airfoil section and an aspect ratio of 4. For this rectangular wing case, the coarse grid contains 91 × 25 × 25 points in the ξ, η, and ζ coordinate directions, respectively, while the fine grid contains 151 × 25 × 34 points. References 24 and 26 discuss the computed results for the same test case. The unsteady measured data from wind tunnel tests are given in Ref. 27.

The unsteady data are given for the case when the rigid wing is oscillating in the pitching mode, $\alpha(t) = \alpha_m - \bar{\alpha}\sin(\omega t)$, about an axis at $x/c = 0.5$, where c is the chord length and ω is the pitching frequency in radians per second. The flow is computed at $M_\infty = 0.8$ with a mean angle of attack $\alpha_m = 0$ deg, a pitch amplitude $\bar{\alpha} = 1$ deg, and a reduced frequency $k = 0.27$ ($k = \omega c/U_\infty$).

Unsteady computations are started from the corresponding steady-state solution. The convergence of the unsteady computations to a periodic flow is verified by comparing the results between cycles. For all cases presented here, the third-cycle results give identical pressure profiles to those of the second-cycle results. Thus, the numerical transient is confirmed to disappear within two cycles.

First, the LU-ADI upwind method was applied to compute this inviscid flow on the coarse (91 × 25 × 25 point) grid using 360, 720, 1080, 1440, and 1800 time steps per cycle of oscillation (steps/cycle). The unsteady pressure profiles on the wing surface that were obtained using 1800 steps/cycle coincided with those using 1440 steps/cycle. The computation converged with respect to time-step sizes at 1440 steps/cycle.

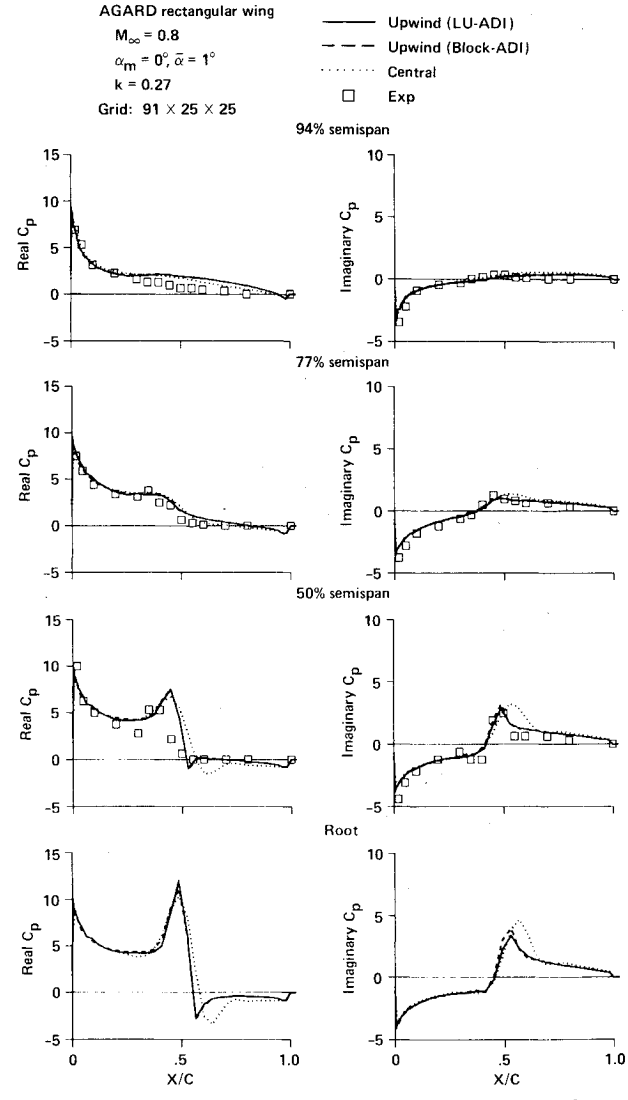

Fig. 1 Comparison of computed inviscid upper surface unsteady pressures among the LU-ADI upwind, block-ADI upwind, and central-difference methods over the rectangular wing with the coarse grid.

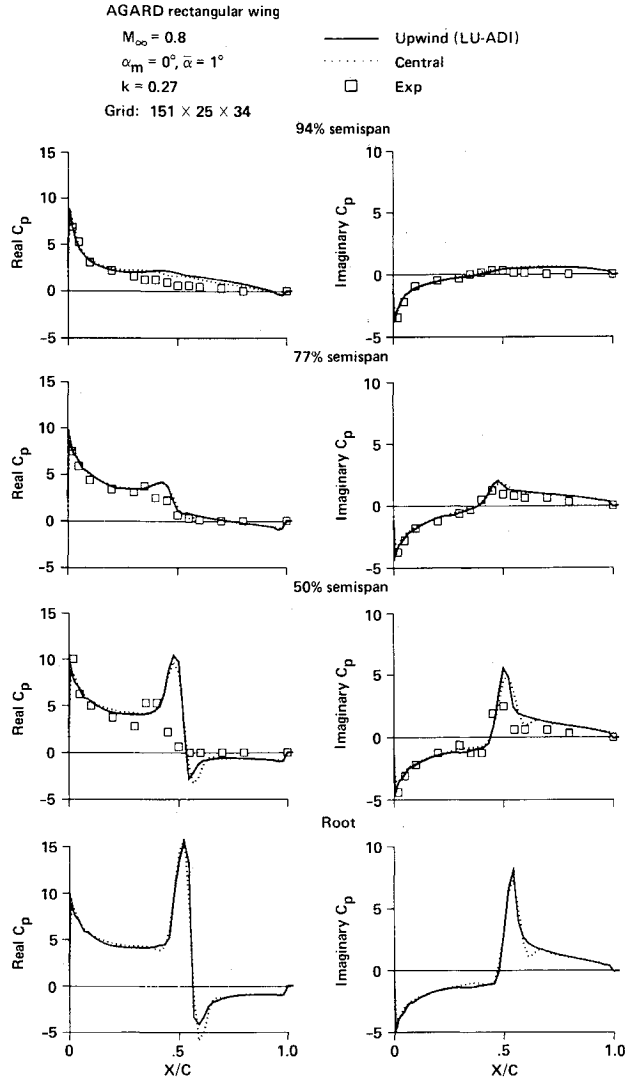

Fig. 2 Comparison of computed inviscid upper surface unsteady pressures between the LU-ADI upwind and central-difference method over the rectangular wing with the fine grid.

There was at most 4% difference in the unsteady pressure plots between the results using 720 and 1440 steps/cycle. Thus, 720 steps/cycle gives the practically converged solution in this case.

The CD method showed less dependence on time-step sizes than the LU-ADI upwind method. The CD computation converged at 720 steps/cycle. The block-ADI upwind method showed more dependence on time-step size. The solution profiles obtained with 3600 steps/cycle finally gave a good agreement with the LU-ADI upwind result obtained at 1440 steps/cycle. The block-ADI computations were not stable for 360 and 720 steps/cycle because of the use of the approximate Jacobians as mentioned before.

Figure 1 shows the comparison of real and imaginary parts of the first Fourier component between the computed and measured unsteady upper surface pressure coefficients of the wing at various spanwise locations, using the three methods with 1440 steps/cycle. Both upwind results give similar profiles of the shock motion, although the LU-ADI upwind method is nonconservative in time. Both upwind results give crisper profiles (narrower peaks) at the region of the shock motion than the CD result.

These results indicate that the temporally nonconservative LU-ADI method can be used for unsteady computations even with moving shock waves when a large enough number of time steps (i.e., small Δt) is used per cycle. In the present case, any number greater than 720 will give practical results.

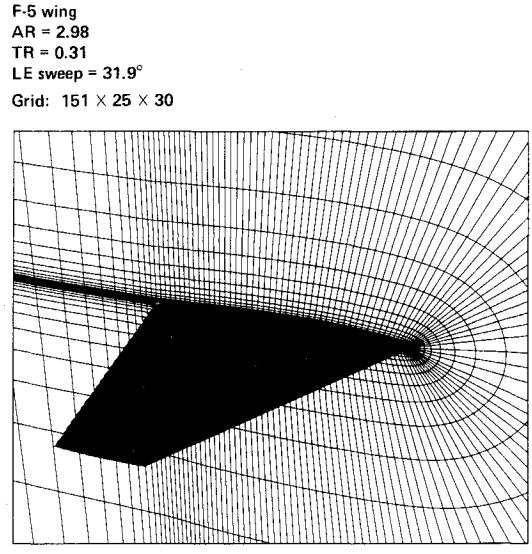

Fig. 3 F-5 wing and grid distributions at root section.

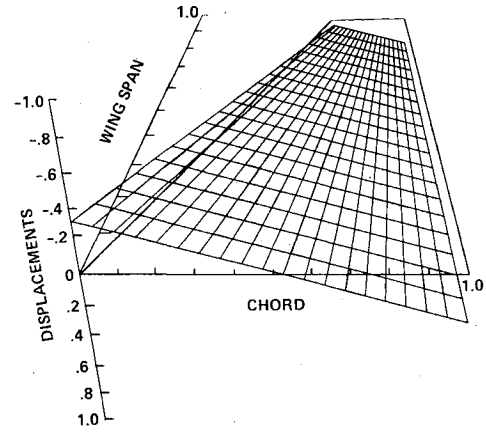

Fig. 4 Unsteady modal motion of F-5 wing.

The CD method also uses the temporally nonconservative diagonal form. However, the result differs from both upwind results. This indicates that the solution depends on the numerical dissipation more than the time-conservative properties of the methods. The block-ADI method requires twice as much CPU time as the LU-ADI method, but its accuracy does not appear to compensate for the increased computational time. Thus, the block-ADI method will be dropped in the following computations.

Next, the inviscid computations were repeated on a finer (151 × 25 × 34 point) grid to check the grid dependency. Figure 2 shows the comparison of unsteady pressures using the LU-ADI upwind and CD methods. Both computations use 720 steps/cycle, as suggested in the coarse-grid case. Comparing with the coarse-grid solution in Fig. 1, the CD solution tends to converge to the upwind solution due to the grid refinement.

Although there is a discrepancy between the computed and measured data especially at 50% semispan location, it can be explained by the difference between the computational and experimental models. In the computation, the wing root section was assumed to be a symmetry plane. However, in the experiment,[27] the wing was mounted on a half-model of a circular body to avoid the effect of side-wall boundary layer. This increased the aspect ratio from 4.0 to 4.76. Also, there was a clearance gap between the wing root and the fixed circular body, which was found to affect the inboard loading. In addition, the experimental wing had a fixed transition, while the computation assumed a fully turbulent flow.

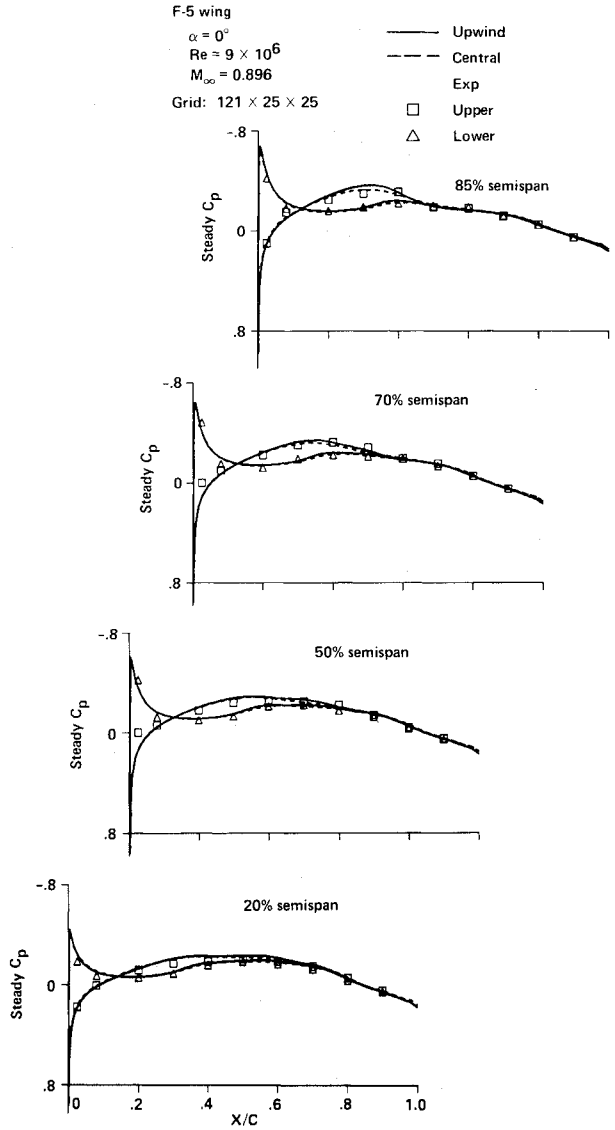

Fig. 5 Comparison of computed viscous steady pressures between the upwind and central-difference methods at M_∞ = 0.896 over the F-5 wing with the coarse grid.

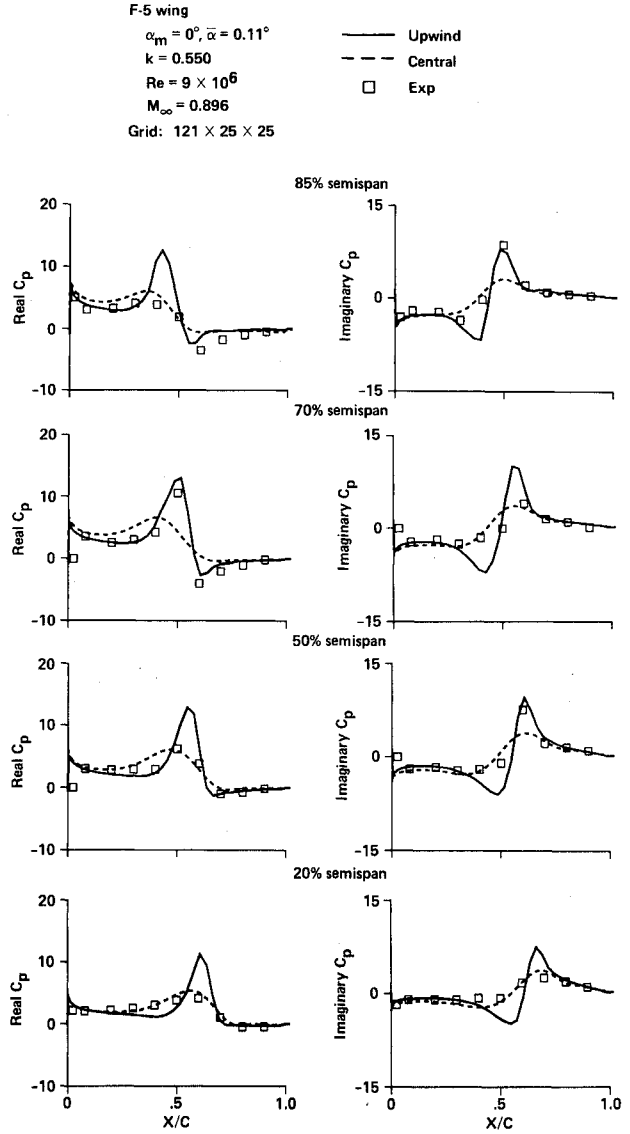

Fig. 6 Comparison of computed viscous unsteady pressures between the upwind and central-difference methods at M_∞ = 0.896 over the F-5 wing with the coarse grid.

To demonstrate the capability of the present method for viscous computations, the same case has been computed with the viscous terms. The viscous grid has the same number of grid points as the inviscid fine grid, but with a smaller spacing in the ζ direction normal to the wing. To obtain a steady-state solution, the grid had to be smoothly rounded at the wing tip due to the stability requirement for the CD method. In contrast, the LU-ADI upwind method is robust enough to allow a finite thickness at the tip.

In the unsteady computations, the time-step size used in the CD computation was limited to at least 1440 steps/cycle even with the smoothly rounded grid because of the stiffness due to clustered viscous grids. In contrast, the LU-ADI upwind method gave a reasonable solution with 720 steps/cycle. The CD computation with 720 steps/cycle was stabilized by doubling the dissipation coefficient, but the resulting pressure distributions were smeared out and thus the result was not satisfactory. The viscous solutions (not shown here) are similar to the inviscid solutions in Fig. 2 because the flowfield is barely viscous.

F-5 Wing

The second test considers unsteady viscous flows over an F-5 wing that has an aspect ratio of 2.98, a taper ratio of 0.31, and a leading-edge sweep angle of 31.92 deg. Computations were made using two grids: the coarse and slightly finer grids containing 121 × 25 × 25 points and 151 × 25 × 30 points, respectively. Figure 3 shows the F-5 wing and the grid distributions at the root section of the fine grid. Figure 4 illustrates the motion used in the experiment conducted at the National Aerospace Laboratory of The Netherlands.[28] The wing is pitching about an axis located at the 50% root chord, and the pitching axis is normal to the wing root. Reference 3 discusses the CD results applied to the same test case. Inviscid results about the F-5 wing can be found in Refs. 29 and 30.

The test cases are chosen at M_∞ = 0.896 and 1.328 where the measured steady and unsteady data are given in Ref. 28. All F-5 wing cases are computed at $Re = 9 \times 10^6$ based on the root chord. At these flow conditions, the coarse-grid solution gave $y^+ \leq 13.3$ at the first shell of points above the wing surface. For both the steady and unsteady cases, the mean angle of attack α_m is 0 deg. The unsteady flows are computed at a reduced frequency k = 0.550 and a pitch amplitude $\bar{\alpha}$ = 0.11 deg for M_∞ = 0.896. Similarly, k = 0.396 and $\bar{\alpha}$ = 0.22 deg for M_∞ = 1.328. The Baldwin-Lomax eddy-viscosity model is used to compute the turbulent viscosity coefficient.

The time-step dependency of the upwind method was checked on both the coarse and fine grids at M_∞ = 0.896. The unsteady

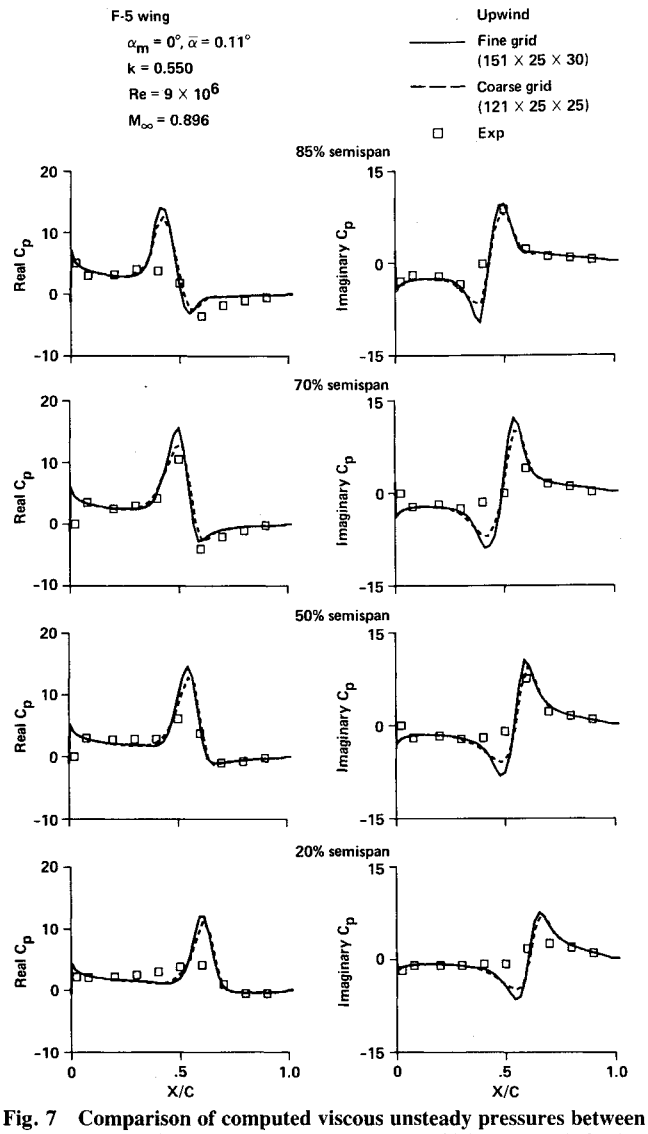

Fig. 7 Comparison of computed viscous unsteady pressures between the coarse- and fine-grid solutions at M_∞ = 0.896 over the F-5 wing using the upwind method.

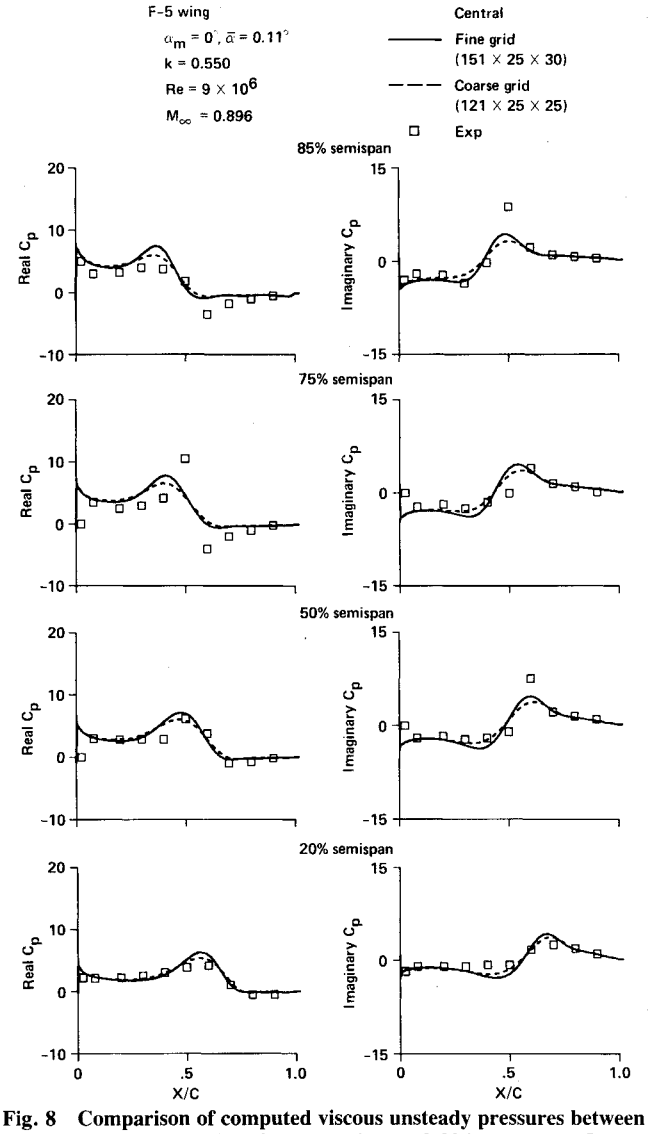

Fig. 8 Comparison of computed viscous unsteady pressures between the coarse- and fine-grid solutions at M_∞ = 0.896 over the F-5 wing using the central-difference method.

pressure profiles converged with respect to time-step sizes at 1800 steps/cycle for the coarse grid and 2160 steps/cycle for the fine grid. For computational efficiency, 1440 and 1800 steps/cycle are deemed acceptable for the coarse and fine grids, respectively. All results shown here were computed with 1800 steps/cycle.

M_∞ = 0.896

Figures 5 and 6 show the comparisons of the computed steady and unsteady pressures with the experimental data at M_∞ = 0.896 using the upwind and CD methods on the coarse (121 × 25 × 25 point) grid. For the steady case, both numerical solutions show shock-free profiles in agreement with the measured data (note that $C_p^* = -0.196$ for $M_\infty = 0.896$). For the oscillatory case, unsteady pressure peaks of the upwind solution indicates that the unsteady motion of the wing produces a shock wave. The measured data also show the peaks except at the 20% semispan location. In contrast, the unsteady pressure peaks of the CD solution are smeared out and the resulting profiles agree less favorably with the experiment except at the 20% semispan location. The computation assumes symmetry at the wing root instead of a side wall which may affect the experimental data at the 20% semispan location.

To see the grid dependency of the numerical solutions, computations were done on the slightly finer grid (151 × 25 × 30 points). Figure 7 shows the comparison of unsteady pressures between the coarse- and fine-grid solutions using the upwind method. The coarse-grid upwind solution is confirmed to predict the shock motion reasonably well compared with the fine-grid result. Figure 8 shows the corresponding plot using the CD method. The unsteady pressure peaks also appear higher with the grid refinement in the CD results. Further grid refinement will be required to obtain the grid convergence because the geometry of the low-aspect-ratio, swept, and tapered F-5 wing is more complicated than the previous rectangular wing.

The unsteady pressure peaks in the upwind result indicate the shock motion, while the smeared peaks in the CD result do not. To check the presence of a shock wave in both numerical solutions, the computed instantaneous pressure coefficient contours at α = 0 deg during pitch down motion are plotted in intervals of 0.02 for both upwind and CD methods (Fig. 9). The bold lines indicate $C_p^* \simeq -0.2$. The upwind result shows the shock wave moving forward, which is captured within the two grid points here. In contrast, the CD result does not show any concentration of contours throughout the cycle. Although the measured unsteady pressure data show the scattering, the data indicate the presence of the shock wave. Therefore, the upwind method captures the important feature of the flowfield better than the CD method.

Since only pressure data were available from the experiment, the computed pressure profiles are compared with the

experimental data. As discussed in Refs. 9 and 11, the upwind method gives better (numerically less dissipative) resolution not only for shock waves but also for boundary layers than the CD method. To indicate this feature, the y^+ values at the first shell of points above the wing surface were averaged over the wing for the steady case at $M_\infty = 0.896$ on the fine grid. The upwind method resulted in $y^+_{ave} = 8.6$, while the CD method resulted in $y^+_{avc} = 7.3$. In general, such y^+ value indicates how well the grid resolves the turbulent boundary layer. In this comparison, the larger y^+ indicates the larger

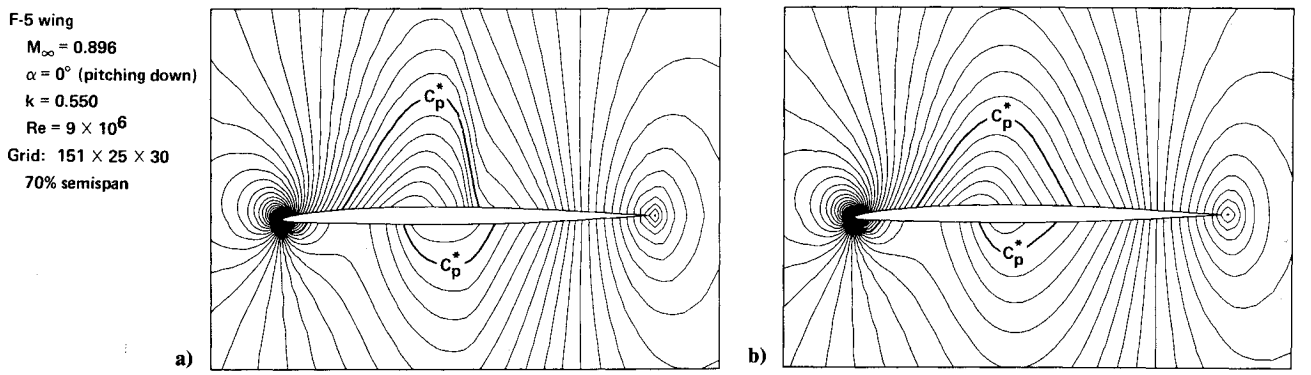

Fig. 9 Comparison of computed instantaneous pressure contours between the upwind and central-difference methods at $M_\infty = 0.896$ over the F-5 wing with the fine grid: a) upwind result; and b) central-difference result.

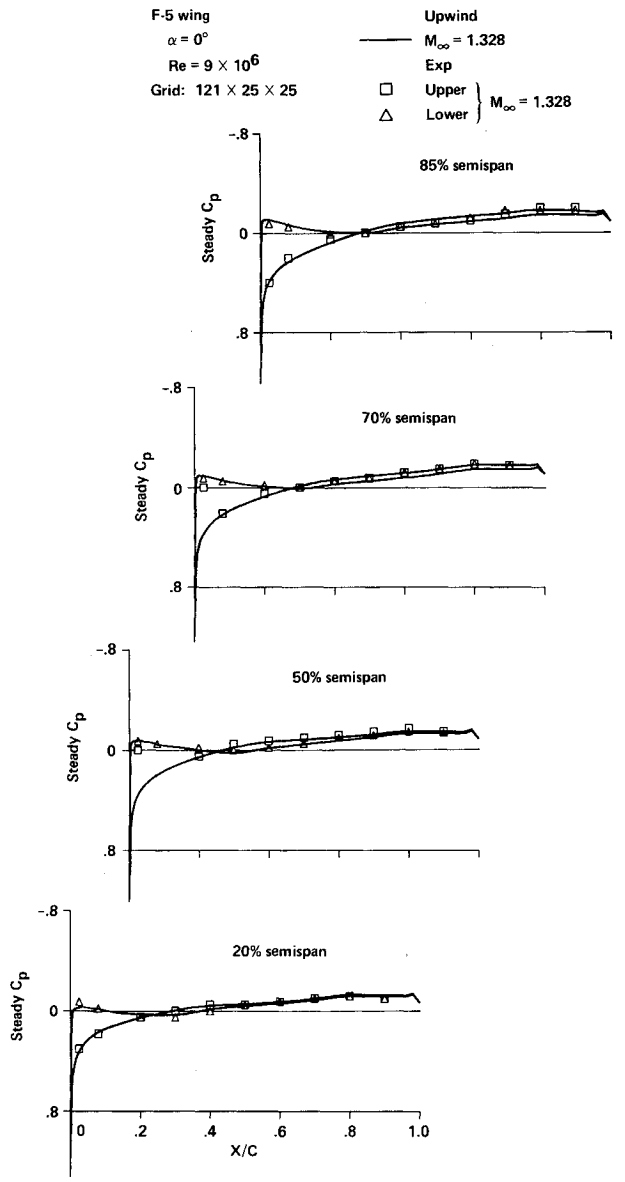

Fig. 10 Comparison of computed viscous steady pressures with experiment at $M_\infty = 1.328$ over the F-5 wing with the coarse grid using the upwind method.

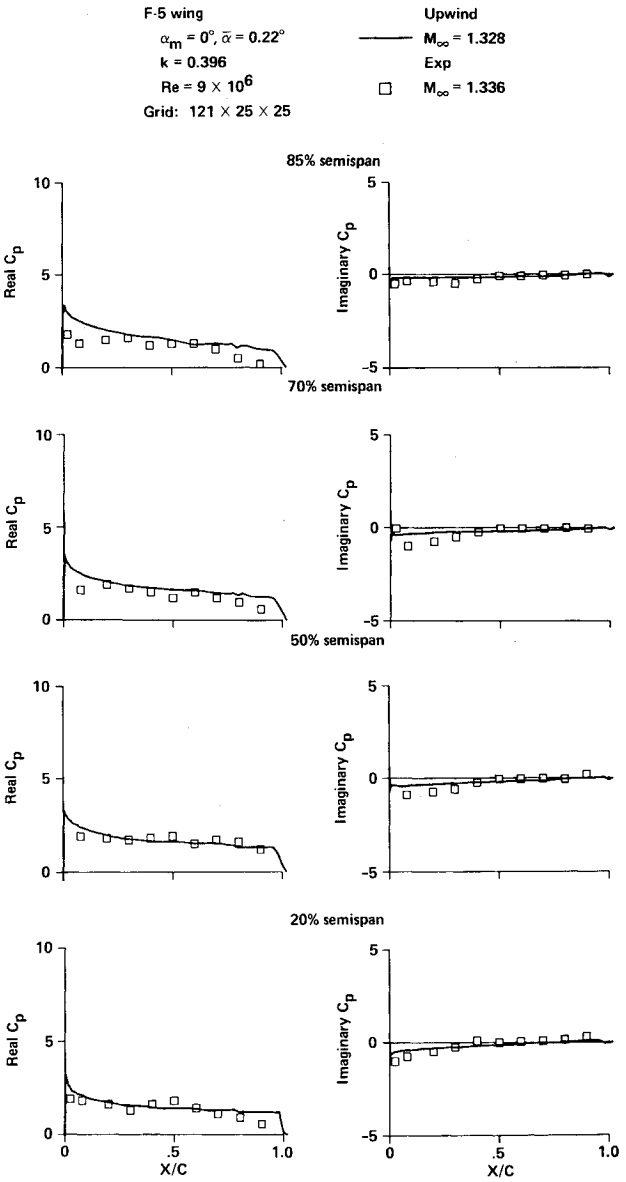

Fig. 11 Comparison of computed viscous unsteady pressures with experiment at $M_\infty = 1.336$ over the F-5 wing with the coarse grid using the upwind method.

shear stress on the wing surface because the same grid, boundary conditions, and turbulence model were used for both methods. In addition, on the fine grid, the upwind and CD methods give identical pressure distributions to those of the coarse-grid upwind results (and thus the results are not shown here). Note that there is no shock wave in the steady-state as shown in Fig. 5. Therefore, the larger y^+ indicates that the upwind method leads to a less dissipative solution because the only difference was in evaluating the inviscid fluxes with Eqs. (17) and (23).

M_∞ = 1.328

Figures 10 and 11 show the comparisons of the computed steady and unsteady pressures with the experimental data at $M_\infty = 1.328$ using the upwind method with the coarse grid. The computed pressures are in reasonable agreement with the experiment for both steady and unsteady supersonic flow cases.

For the viscous case, the upwind computation requires 18.9 μs per grid point per time step at a speed of 163 MFLOPS on a CRAY-YMP computer using a single processor, while the CD computation requires 17.1 μs at a speed of 150 MFLOPS. There is an 11% increase in CPU time while using the upwind option of the code.

Conclusions

A streamwise upwind algorithm has been developed to compute unsteady flows associated with a moving grid system. The resulting code has been applied to compute flows over oscillating wings at transonic Mach numbers. Comparisons have been made between results obtained from the upwind algorithm, using both temporally nonconservative- and conservative-implicit methods, with the results obtained from the existing central-difference (CD) method, and also with experimental data.

A comparison of the temporally nonconservative LU-ADI method with a time-conservative block-ADI version of the upwind scheme indicates that the solutions are insensitive to the time conservativeness of the implicit solvers when practical time-step sizes are used for a moderate motion of shock wave in a transonic flowfield. The LU-ADI method was found to allow larger time steps and require 50% less computational time per iteration than the block-ADI method.

Based on the coarse-grid solutions, the LU-ADI upwind method gives less dissipative solutions and thus predicts the shock motion better than the CD method. The present method is more robust than the CD method, since it does not require any explicitly added numerical dissipation. The present work illustrates the improvements in robustness and accuracy by using the streamwise upwind scheme compared to the CD scheme.

References

[1]Dobbs, S. K., and Miller, G. D., "Self-Induced Oscillation Wind-Tunnel Test of a Variable Sweep Wing," AIAA Paper 85-0739, April 1985.

[2]Guruswamy, G. P., "Unsteady Aerodynamic and Aeroelastic Calculations of Wings Using Euler Equations," *AIAA Journal*, Vol. 28, No. 3, 1990, pp. 461–469.

[3]Guruswamy, G. P., "Navier-Stokes Computations on Swept-Tapered Wings, Including Flexibility," AIAA Paper 90-1152, April 1990.

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

[5]Van Leer, B., Thomas, J. L., Roe, P. L., and Newsome, R. W., "A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations," *Proceedings of the AIAA 8th Computational Fluid Dynamics Conference*, 1987, pp. 36–41.

[6]Vatsa, V. N., Thomas, J. L., and Wedan, B. W., "Navier-Stokes Computations of Prolate Spheroids at Angle of Attack," *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, 1987, pp. 488–506.

[7]Goorjian, P. M., Meagher, M. E., and van Buskirk, R., "Monotone Switches in Implicit Algorithms for Potential Equations Applied to Transonic Flows," *AIAA Journal*, Vol. 23, No. 4, 1985, pp. 492–498.

[8]Goorjian, P. M., "Algorithm Development for the Euler Equations with Calculations of Transonic Flows," AIAA Paper 87-0536, Jan. 1987.

[9]Goorjian, P. M., "A New Algorithm for the Navier-Stokes Equations Applied to Transonic Flows Over Wing," *Proceedings of the AIAA 8th Computational Fluid Dynamics Conference*, 1987, pp. 230–236.

[10]Goorjian, P. M., "A Streamwise Upwind Algorithm for the Euler and Navier-Stokes Equations Applied to Transonic Flows," *Numerical Methods for Fluid Dynamics III*, edited by K. W. Morton and Baines, Inst. of Mathematics and Its Applications Conference Series, New Series No. 17, Clarendon Press, Oxford, England, UK, 1988, pp. 318–324; also NASA TM 101019, Aug. 1988.

[11]Obayashi, S., and Goorjian, P. M., "Improvements and Applications of a Streamwise Upwind Algorithm," *Proceedings of the AIAA 9th Computational Fluid Dynamics Conference*, 1989, pp. 292–302.

[12]Rai, M. M., and Chakravarthy, S. R., "An Implicit Form for the Osher Upwind Scheme," *AIAA Journal*, Vol. 24, No. 5, 1986, pp. 735–743.

[13]Anderson, W. K., Thomas, J. L., and van Leer, B., "A Comparison of Finite-Volume Flux Vector Splittings for the Euler Equations," AIAA Paper 85-0122, Jan. 1985.

[14]Baldwin, B. S., and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, Jan. 1978.

[15]Obayashi, S., "Numerical Simulation of Underexpanded Plumes Using Upwind Algorithms," *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, 1988, pp. 284–299.

[16]Vinokur, M., "An Analysis of Finite-Difference and Finite-Volume Formulations of Conservation Laws," *Journal of Computational Physics*, Vol. 81, No. 1, 1989, pp. 1–52.

[17]Koren B., "Upwind Schemes, Multigrid and Defect Correction for the Steady Navier-Stokes Equations," *11th International Conference on Numerical Methods in Fluid Dynamics*, Lecture Notes in Physics, No. 323, Springer-Verlag, New York, 1989, pp. 344–348.

[18]Liepmann, H. W., and Roshko, A., *Elements of Gasdynamics*, 4th Printing, Wiley, New York, 1962, pp. 57–61 and 85–86.

[19]Barth, T. J., "Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms," AIAA Paper 87-0595, Jan. 1987.

[20]Liou, M. S., and van Leer, B., "Choice of Implicit and Explicit Operators for the Upwind Differencing Method," AIAA Paper 88-0624, Jan. 1988.

[21]Venkatakrishnan, V., "Viscous Computations Using a Direct Solver," *Computers and Fluids*, Vol. 18, No. 2, 1990, pp. 191–204.

[22]Steger, J. L., and Warming, R. F., "Flux-Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods," *Journal of Computational Physics*, Vol. 40, No. 2, 1981, pp. 263–293.

[23]Obayashi, S., Matsushima, K., Fujii, K., and Kuwahara, K., "Improvements in Efficiency and Reliability for Navier-Stokes Computations using the LU-ADI Factorization Algorithm," AIAA Paper 86-0338, Jan. 1986.

[24]Obayashi, S., Goorjian, P. M., and Guruswamy, G. P., "Extension of a Streamwise Upwind Algorithm to a Moving Grid System," NASA TM-102800, April 1990.

[25]Pulliam, T. H., and Steger, J. L., "Recent Improvement in Efficiency, Accuracy, and Convergence for Implicit Approximate Factorization Algorithms," AIAA Paper 85-360, Jan. 1985.

[26]Chaderjian, N. M., and Guruswamy, G. P., "Unsteady Transonic Navier-Stokes Computations for an Oscillating Wing Using Single and Multiple Zones," AIAA Paper 90-0313, Jan. 1990.

[27]Mabey, D. G., Welsh, B. L., and Pyne, C. R., "A Summary of Measurements of Steady and Oscillatory Pressures on a Rectangular Wing," *The Aeronautical Journal of the Royal Aeronautical Society*, Vol. 92, No. 911, Jan. 1988, pp. 10–28.

[28]Tijdeman, J., van Hunen, J. W. G., Krann, A. N., Persoon, A. J., Poestkoke, R., Roos, R., Schippers, P., and Siebert, C. M., "Transonic Wind-Tunnel Tests on an Oscillating Wing with External Stores; Pt. II—The Clean Wing," AFFDL-TR-78-194, Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH, March 1979.

[29]Guruswamy, G. P., and Goorjian, P. M., "Efficient Algorithm for Unsteady Transonic Aerodynamics of Low-Aspect-Ratio Wings," *Journal of Aircraft*, Vol. 22, No. 3, 1985, pp. 193–199.

[30]Anderson, W. K., Thomas, J. L., and Rumsey, C. L., "Extension and Application of Flux-Vector Splitting to Calculations on Dynamic Meshes," *AIAA Journal*, Vol. 27, No. 6, 1989, pp. 673–674.