Numerical Solutions of Forward-Flight Rotor Flow Using an Upwind Method

C. L. Chen*
Rockwell International Science Center, Thousand Oaks, California 91360
and
W. J. McCroskey† and S. Obayashi‡
NASA Ames Research Center, Moffett Field, California 94035

A finite-volume upwind algorithm for solving the three-dimensional Euler equations with a moving grid has been developed for computing helicopter forward-flight rotor flows. The computed pressure distributions and shock positions of high-speed rotor flow are compared with various experimental data as well as with other numerical results, and the agreement is encouraging. A comparison of quasisteady solutions with unsteady solutions reveals that when a shock occurs in the flowfield, the assumption of quasisteady flow may fail due to the time lag of the shock motion. Similarly, three-dimensional effects cannot be neglected. Sufficient subiterations for each time step are required to avoid numerical lag effects in using the present method. The redistribution of the residual due to the coordinate transformation is discussed. For high-order Monotone Upstream-Centered Conservation Law (MUSCL)-type schemes, a coordinate-independent solution can be obtained by interpolating primitive variables.

Introduction

A NUMBER of unsteady Euler/Navier-Stokes solvers have been developed for helicopter rotor flow, including central-difference methods and partial flux-splitting methods.[1-8] The latter consists of upwind differencing in the mainstream direction and central differencing in the other directions. Upwind algorithms in all of the directions, based on Roe's scheme, have been widely and successfully used for various complex fluid flow computations.[9-12] Such schemes not only capture shocks well but also predict accurately boundary-layer quantities in viscous calculations.[13] Therefore, it seems useful to extend the scheme to unsteady rotor flows with a moving-grid system. Unfortunately, for implicit schemes, the true flux Jacobian of Roe's scheme is too complicated and the direct solver is inefficient for practical use. Thus, an implicit iterative procedure, using Jameson and Yoon's simplified flux Jacobian, is adopted in the present approach. The LU-SGS method[12,14] is used along with quasi-Newton iterations to ensure time-accurate solutions. The finite-volume cell-averaged discretization procedure is unchanged from previous work.[7,8]

The primary objectives of this paper are to demonstrate the ability of a finite-volume upwind algorithm based on an upwind-biased flux-difference-splitting approach to calculate rotor flows and to improve upon the efficiency of previous work.[7,8] This raises some numerical issues regarding coordinate dependency between the inertial and wing-fixed frame. Thus, the residual redistribution for each cell due to the coordinate transformation is examined.

In the process of validating the approach, three-dimensional and unsteady time-lag shock motion effects for forward-flight rotor flow are examined. Both of the nonlifting rectangular untwisted rotors studied have NACA 0012 profiles with constant chord.

Governing Equations

The basic equations under consideration are the unsteady Euler equations with general grid motions. The computational domain consists of structure-wise hexahedral cells. Each cell surface can be identified in the ξ, η, or ζ direction, respectively. The general form of a conservation law applied to an arbitrary grid cell is

$$\int_{V(t_2)} \boldsymbol{Q}\,\mathrm{d}V - \int_{V(t_1)} \boldsymbol{Q}\,\mathrm{d}V + \int_{t_1}^{t_2} \oint_{S(t)} \boldsymbol{n}\cdot\boldsymbol{F}\,\mathrm{d}S\,\mathrm{d}t = 0 \qquad (1)$$

where

$$\boldsymbol{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \qquad F_n = \boldsymbol{n}\cdot\boldsymbol{F} = \begin{bmatrix} \rho\bar{u} \\ \rho u\bar{u} + pn_x \\ \rho v\bar{u} + pn_y \\ \rho w\bar{u} + pn_z \\ e\bar{u} + pu_n \end{bmatrix}$$

where $V(t)$ is the cell volume, $\boldsymbol{n}\mathrm{d}S(t)$ is a vector element of surface area with outward unit normal vector $\boldsymbol{n}(n_x,n_y,n_z)$, $\boldsymbol{Q}$ is a vector of conservative variables per unit volume, ρ and u, v, w are the density and the fluid velocity components in the x, y, z directions, respectively, and e is the total internal energy per unit volume. In addition, $\boldsymbol{F}$ is the flux of $\boldsymbol{Q}$ per unit area per unit time, p is the pressure, $u_n = \boldsymbol{n}\cdot\boldsymbol{u}$ is the normal velocity component of the fluid, $v_n = \boldsymbol{n}\cdot\boldsymbol{v}$ where $\boldsymbol{v}$ is the grid velocity, and $\bar{u} = \boldsymbol{n}\cdot(\boldsymbol{u}-\boldsymbol{v}) = u_n - v_n$ is the normal relative velocity component. The term v_n is not a function of time and is fixed at each cell for the grid with constant rigid-body rotation.

The preceding Euler equations are written using the absolute variables in the inertial frame. In our calculation of forward-flight rotor flow, this formulation was based mainly on two reasons: 1) the far-field boundary conditions are easily

Presented as Paper 89-1846 at the AIAA 20th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Buffalo, NY, June 12-14, 1989; received Jan. 20, 1990; revision received July 9, 1990; accepted for publication Aug. 26, 1990.

*Member Technical Staff. Member AIAA.
†Senior Staff Scientist. Associate Fellow AIAA.
‡NRC Research Associate. Member AIAA.

implemented, and 2) the formulation is in conservation form. However, this formulation can be recast in terms of absolute variables written in a rotational frame.[15,16] In other words, the absolute fluid velocity, which is the velocity relative to an inertial frame, can be represented as (u^*, v^*, w^*) in the rotational frame. Similarly, the surface unit vector $\boldsymbol{n}$ can be represented as (n_x^*, n_y^*, n_z^*) and calculated only once without the rotational matrix being involved.[8] The equations remain in exactly the same form as in the original formulation, except for an additional source term R, which is introduced into the right side of the equation as

$$\int_{V(t_2)} Q^* \mathrm{d}V - \int_{V(t_1)} Q^* \mathrm{d}V + \int_{t_1}^{t_2} \oint_{S(t)} F_n^* \, \mathrm{d}S \, \mathrm{d}t = R \tag{2}$$

where

$$Q^* = \begin{bmatrix} \rho \\ \rho u^* \\ \rho v^* \\ \rho w^* \\ e \end{bmatrix}, \quad F_n^* = \begin{bmatrix} \rho\bar{u} \\ \rho u^*\bar{u} + pn_x^* \\ \rho v^*\bar{u} + pn_y^* \\ \rho w^*\bar{u} + pn_z^* \\ e\bar{u} + pu_n \end{bmatrix}, \quad R = V \begin{bmatrix} 0 \\ \Omega\rho v^* \\ -\Omega\rho u^* \\ 0 \\ 0 \end{bmatrix}$$

and Ω is the angular velocity in the z direction relative to the inertial frame. The next section shows how easily the present algorithm can be adapted to this formulation. The advantages of the latter formulation are shown for the case of a hovering rotor with constant rotating speed and constant ascent (or descent) speed. The far-field boundary conditions are still very easy to implement for this specific case, and the flow may be treated as independent of time (if unsteadiness in the wake can be ignored). The source term in the preceding discussion is nonzero only in the momentum equations. Thus, for the full-potential equation, where only the continuity equation is considered, it is unnecessary to discuss the absolute variables written in either the inertial or the rotational frame.

Numerical Algorithm

Upwind Method

A finite-volume upwind method developed especially for helicopter rotor flow is described here. The convective terms are differenced with the upwind-biased flux-difference splitting of Roe for the explicit side, while Jameson and Yoon's[14] LU-SGS method is used for the implicit operator. In addition, quasi-Newton iterations are used at each time step to maintain time accuracy.

A numerical flux of the inviscid terms, due to Roe's upwinding in the locally one-dimensional form, can be written for the ξ (index j) direction as

$$\hat{F}(Q_L, Q_R, S_{j+\frac{1}{2}}) = \tfrac{1}{2}[\hat{F}(Q_R, S_{j+\frac{1}{2}}) + \hat{F}(Q_L, S_{j+\frac{1}{2}}) - |\tilde{A}(Q_L, Q_R, S_{j+\frac{1}{2}})|(Q_R - Q_L)] \tag{3}$$

where $\tilde{A}$ is the Roe-averaged Jacobian matrix and Q_L and Q_R are the state variables to the left and right of the cell interface, and

$$|\tilde{A}|(Q_R - Q_L) = \begin{bmatrix} \alpha_4 \\ \tilde{u}\alpha_4 + n_x\alpha_5 + \alpha_6 \\ \tilde{v}\alpha_4 + n_y\alpha_5 + \alpha_7 \\ \tilde{w}\alpha_4 + n_z\alpha_5 + \alpha_8 \\ \tilde{H}\alpha_4 + \tilde{u}_n\alpha_5 + \tilde{u}\alpha_6 + \tilde{v}\alpha_7 + \tilde{w}\alpha_8 - \dfrac{(\tilde{c}^2)}{(\gamma-1)}\alpha_1 \end{bmatrix} \tag{4}$$

where

$$\alpha_1 = |S|\,|\tilde{\bar{u}}|(\Delta\rho - \Delta p/\tilde{c}^2)$$

$$\alpha_2 = |S|\,|\tilde{\bar{u}} + \tilde{c}|(\Delta p + \tilde{\rho}\tilde{c}\Delta\bar{u})/2\tilde{c}^2$$

$$\alpha_3 = |S|\,|\tilde{\bar{u}} - \tilde{c}|(\Delta p - \tilde{\rho}\tilde{c}\Delta\bar{u})/2\tilde{c}^2$$

$$\alpha_4 = \alpha_1 + \alpha_2 + \alpha_3$$

$$\alpha_5 = \tilde{c}(\alpha_2 - \alpha_3)$$

$$\alpha_6 = |S|\,|\tilde{\bar{u}}|(\tilde{\rho}\Delta u - n_x\tilde{\rho}\Delta\bar{u})$$

$$\alpha_7 = |S|\,|\tilde{\bar{u}}|(\tilde{\rho}\Delta v - n_y\tilde{\rho}\Delta\bar{u})$$

$$\alpha_8 = |S|\,|\tilde{\bar{u}}|(\tilde{\rho}\Delta w - n_z\tilde{\rho}\Delta\bar{u})$$

where $\Delta\rho = \rho_R - \rho_L$, and so on, H is the total enthalpy, c is the sound speed, and superscript $\tilde{}$ denotes Roe-averaged quantities. The area magnitude of the cell interface S is $|S|$, $\boldsymbol{n} = S/|S|$, and $\bar{u} = \boldsymbol{n}\cdot(\boldsymbol{u} - \boldsymbol{v}) = u_n - v_n$. The so-called time metrics for the grid motion are included in the evaluations of $\hat{A}$, $\hat{F}(Q_R, S)$, and $\hat{F}(Q_L, S)$. This is considered to be more straightforward than the extension of Van Leer flux splitting to a moving-grid system.[17]

The scheme yields first-order accuracy if $Q_L = Q_j$ and $Q_R = Q_{j+1}$. Higher order schemes are constructed from a one-parameter family of interpolations of the primitive variables $q = (\rho, u, v, w, p)^T$:

$$q_L = q_j + (s/4)[(1-\kappa)\nabla_j + (1+\kappa)\Delta_j] \tag{5a}$$

$$q_R = q_{j+1} - (s/4)[(1+\kappa)\nabla_{j+1} + (1-\kappa)\Delta_{j+1}] \tag{5b}$$

where $\kappa = (1/3)$ is used in the following for the third-order computations, and $\kappa = -1$ for the second-order fully upwind method. The symbols ∇ and Δ are backward and forward difference operators, respectively, which are limited to ensure the monotone interpolation[18] by

$$s = \frac{3\Delta\nabla + \epsilon}{2(\nabla - \Delta)^2 + 3\Delta\nabla + \epsilon} \tag{6}$$

with $\epsilon \le 1.0E-6$ for $\kappa = (1/3)$. The reasons for choosing primitive variables for interpolation will be explained later.

LU-SGS Method

The Jameson and Yoon LU-SGS implicit operator is used in the present study. Let $\hat{A}$, $\hat{B}$, $\hat{C}$ be Jacobian matrices in the ξ, η, and ζ directions, respectively, and let the indices j, k, and l correspond to these directions. The LU-SGS implicit operator consists of L, D, U, as follows:

$$L = VI + \Delta t(-\hat{A}^-|_{j,k,l} + \nabla_\xi\hat{A}^+ - \hat{B}^-|_{j,k,l} + \nabla_\eta\hat{B}^+ - \hat{C}^-|_{j,k,l} + \nabla_\zeta\hat{C}^+) \tag{7a}$$

$$D = [VI + \Delta t(\hat{A}^+ - \hat{A}^- + \hat{B}^+ - \hat{B}^- + \hat{C}^+ - C^-)|_{j,k,l}]^{-1} \tag{7b}$$

$$U = VI + \Delta t(\hat{A}^+|_{j,k,l} + \Delta_\xi\hat{A}^- + \hat{B}^+|_{j,k,l} + \Delta_\eta\hat{B}^- + \hat{C}^+|_{j,k,l} + \Delta_\zeta\hat{C}^-) \tag{7c}$$

with $\hat{A}^\pm = \tfrac{1}{2}(\hat{A} \pm \sigma_\xi I)$ and $\sigma_\xi \ge (\bar{u} + c)|S|$, and similiarly for $\hat{B}^\pm$ and $\hat{C}^\pm$. Because of the simplified form of the Jacobian terms $\hat{A}^\pm$, $\hat{B}^\pm$, and $\hat{C}^\pm$, the matrix D is a scalar diagonal matrix. Thus, this method requires only two (forward and backward) sweeps with scalar inversions and leads to reduced factorization error.

The whole algorithm applied to Eq. (1) may be symbolically described as

$$LDU\Delta Q^p = -\Delta t RHS^p \tag{8a}$$

where RHS^p is the residual of the unsteady Euler equations:

$$RHS^p = \delta_\xi \hat{E}^p + \delta_\eta \hat{F}^p + \delta_\zeta \hat{G}^p + [Q^n \Delta V + V(Q^p - Q^n)]/\Delta t \tag{8b}$$

with numerical fluxes $\hat{E}$, $\hat{F}$, and $\hat{G}$ in the ξ, η, and ζ directions, respectively. At any time n, a few iterations on p may be required. In general, p decreases as Δt is reduced.

If absolute variables are written in the rotational frame as Q^* in Eq. (2), in addition to a right side source term, a source term Jacobian can be added in either the L or U operator. This source term Jacobian is

$$\frac{\partial R}{\partial Q^*} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Omega V & 0 & 0 \\ 0 & -\Omega V & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

Slightly more work is then required for 2×2 matrix inversions.

All of the boundary conditions are implemented explicitly, as shown in Ref. 8. For inviscid flow, no flow normal to the wall boundary is allowed by imposing $\boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{v} \cdot \boldsymbol{n}$ on the solid wall, where $\boldsymbol{u}$, $\boldsymbol{v}$ are fluid velocity and grid velocity and $\boldsymbol{n}$ is surface normal unit vector of wall. The wall pressure is obtained by solving the normal momentum equation.

Coordinate-Independent Scheme

The numerical scheme used here is a finite-volume scheme.[7,8] It captures freestream exactly. Another feature of the present scheme is the interpolations with primitive variables for high-order schemes, as in Refs. 11 and 12. The numerical solutions satisfy exactly the linear transformation of the velocity field between the inertial frame with moving grid and the wing-fixed frame. In other words, if a set of solutions ρ, u, v, w, p is obtained for freestream past an object in the wing-fixed frame, it is equivalent to a set of solutions $\rho, u - U_\infty, v, w, p$ for the object moving in the negative x direction in the inertial frame. The relation mentioned is obvious for differential equations, but it may not be exactly true for arbitrarily discretized equations, as will be shown in the following discussion. A scheme that satisfies this relation will be called a coordinate-independent scheme.

For simplicity, a one-dimensional flux-balance is considered for each cell; however, the following discussions are still true for multidimensional problems. Suppose the solution ρ, u, p is obtained by solving the equations in the wing-fixed coordinate. The velocity u is then linearly transformed to $u - U_\infty$, and ρ and p remain the same as in the inertial frame. For a first-order scheme, the interpolation results in $\boldsymbol{Q}_L = \boldsymbol{Q}_j$ and $\boldsymbol{Q}_R = \boldsymbol{Q}_{j+1}$, where $\boldsymbol{Q}$ may be either a primitive or a conservative vector. Therefore, at a cell interface, $u_L(u_R)$ will become $u_L(u_R) - U_\infty$, and ρ and p will remain unchanged. Suppose the residuals in the wing-fixed coordinates are $(\epsilon_1, \epsilon_2, \epsilon_3)$ for the mass, momentum, and energy equations, respectively. By casting the transformed variables into the numerical-flux evaluation [Eq. (3)] and applying the conservation law at each cell, the residuals are redistributed in the inertial coordinate system as follows:

1) Residual ϵ_1 remains the same as in the continuity equation.
2) Residual ϵ_2 is changed to $\epsilon_2 - U_\infty \epsilon_1$.
3) Residual ϵ_3 is changed to $\epsilon_3 + (U_\infty^2/2)\epsilon_1 - U_\infty \epsilon_2$.

If a fully converged solution ($\epsilon_i = 0$) is obtained for wing-fixed coordinates, the linearly transformed velocity would also be a fully converged solution in the inertial frame. Thus, the first-order scheme is a coordinate-independent scheme.

For higher order schemes, using Eq. (5), flow variables $\boldsymbol{Q}_L$ and $\boldsymbol{Q}_R$ can always be represented by $\Sigma \alpha_j \boldsymbol{Q}_j$, with $\Sigma \alpha_j = 1$ for each flow variable interpolation. Three equations have to be solved at each cell for a one-dimensional problem. Here, the $\boldsymbol{Q}_j$ are the neighboring cell flow variables. Without any limiter, the α_j are the same for all the equations. Then $u_L(u_R)$ will become $u_L(u_R) - U_\infty$ at each cell interface, regardless of whether the interpolation is performed on the conservative or primitive variables. Again, a coordinate-independent scheme is easily obtained.

Once a slope limiter is introduced into each flow variable, α_j may not be the same for all the equations. If conservative variables are used for the interpolation to obtain cell interface state variables, a linear transformation of the velocity field may not be obtained that guarantees the vanishing of the residual. However, if primitive variables are used for the interpolation/extrapolation, a linear velocity transformation relation is maintained. In addition, the interpolation relation for ρ and p remains the same for both reference frames. This guarantees that the scheme is coordinate-independent. Hence, from this point of view, the interpolation of the primitive variables would be preferable.

Numerical Results

In the process of validating this algorithm, the present algorithm based on Roe's scheme is first applied to a transonic flow past an airfoil, and the results are compared with those obtained with the previous method. Subsequently, two forward-flight rotor flows were examined; the results are discussed and compared with experimental data. Both of the nonlifting rectangular untwisted rotors studied have NACA 0012 profiles with constant chord.

NACA 0012 Airfoil

A NACA 0012 airfoil at $M_\infty = 0.85$, $\alpha = 1.0$ deg was calculated using the present method, with $\kappa = (1/3)$. The C-type grid uses 181×31 points. The results are compared in Fig. 1 with the solutions from a partial flux-splitting method and $\kappa = -1$ without limiter.[7,8] The partial flux-splitting method consists of upwind differencing in the mainstream direction and central differencing in the other directions. The shock locations agree to within one grid point. The leading-edge suction peak is predicted slightly better with the present code. The result also agrees very well with Yee's calculation,[19] which is not included here. Roe's scheme with Koren's differential limiter can predict a one-point shock transition without strong

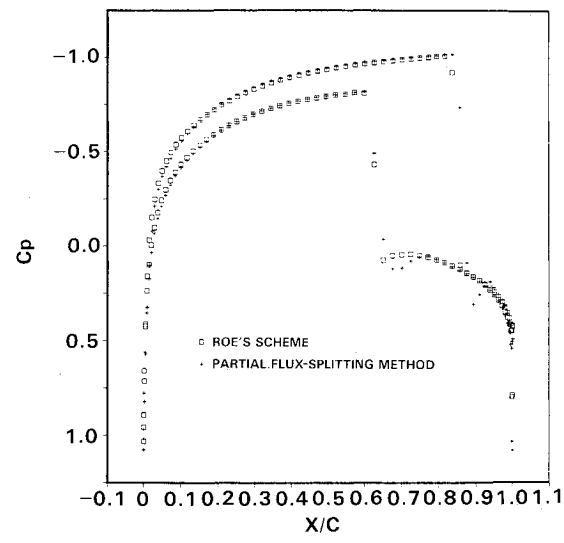

Fig. 1 Surface pressure comparison with various right sides for a NACA 0012 airfoil at $M_\infty = 0.85$, $\alpha = 1.0$.

postshock overshoots. However, for this transonic flow, no limiter was required for the partial flux-splitting scheme with the fully upwind Monotone Upstream-Centered Conservation Law (MUSCL) approach ($\kappa = -1$). Conversely, without a limiter, numerical oscillations occurred with Roe's approach using $\kappa = -1$.

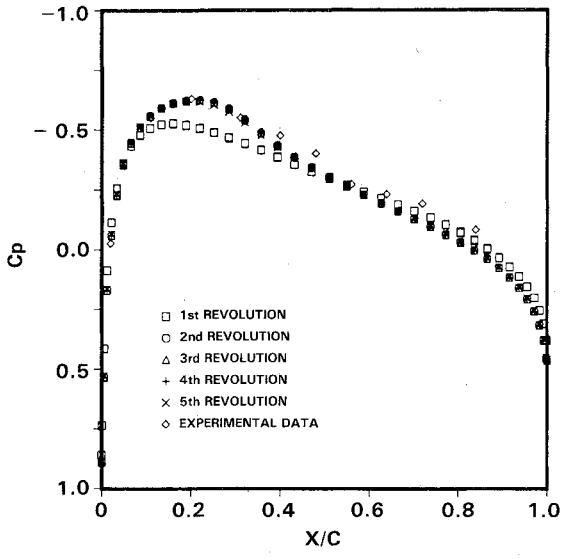

Fig. 2 Surface pressure at $\Psi = 30$ deg with various revolutions for a forward-flight rotor; $M_T = 0.8$, $\mu = 0.2$, aspect ratio = 7, nonlifting untwisted untapered NACA 0012 blade.

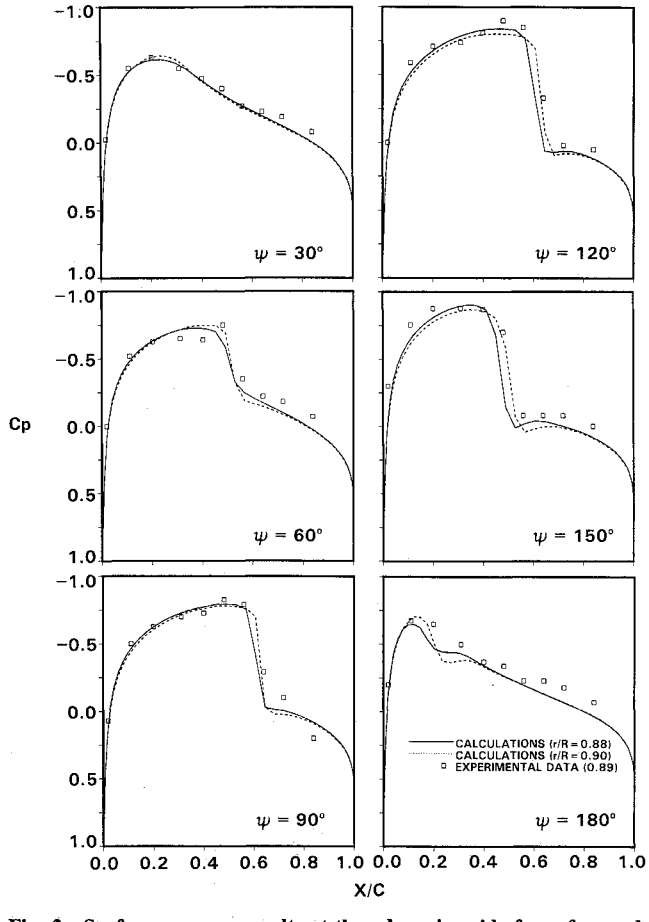

Fig. 3 Surface pressure results at the advancing side for a forward-flight two-blade rotor; $M_T = 0.8$, $\mu = 0.2$, aspect ratio = 7, nonlifting untwisted untapered NACA 0012 blade.

Two Forward-Flight Rotors

The aspect ratio of the first rotor blade is 7.0 and the flight conditions consist of tip Mach number $M_T = 0.8$ and advance ratio $\mu = 0.2$.[20] The grid size used was 81 × 25 × 35 points in an O-H-type topology. For this forward-flight rotor, the initial solution was a hovering solution obtained by solving Eq. (2). Then, the forward-flight speed was imposed at the farfield, and the time-accurate calculation of Eq. (1) was started at azimuthal angle $\Psi = 0.0$ deg. The solution from the first to the fifth revolution at $\Psi = 30$ deg is shown in Fig. 2. The numerical result shows a recursive surface pressure distribution beginning at the second revolution, indicating the establishment of a periodic flow. The solutions for spanwise station $r/R = 0.88$ and 0.90 at various azimuthal locations are shown in Fig. 3 and compared with experimental data at $r/R = 0.89$ ($r/R = 1.0$ at blade tip). The location of the shock is predicted well. Shock locations may differ in one point with different methods. The partial flux-splitting method occasionally predicts shock locations a point farther toward the trailing edge, as shown in Fig. 4. The leading-edge suction peak is slightly lower than that obtained from Roe's scheme with limiter. These effects have also been observed in the airfoil calculation described earlier. Overall, from accuracy consider-

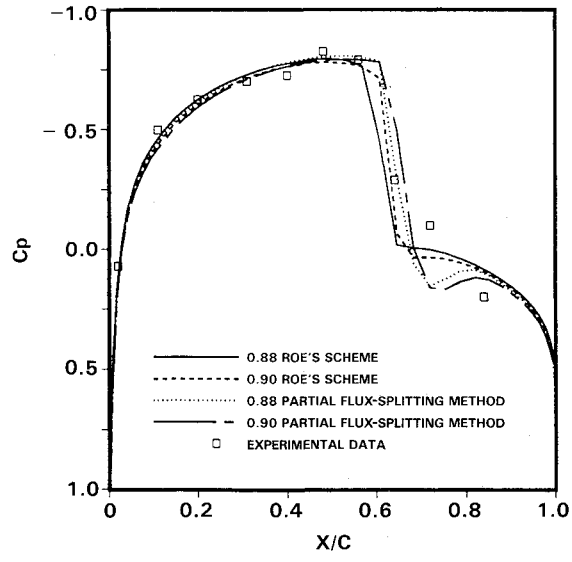

Fig. 4 Surface pressure results with various right sides for a forward-flight rotor at $\Psi = 90$ deg, $M_T = 0.8$, $\mu = 0.2$, aspect ratio = 7, nonlifting untwisted untapered NACA 0012 blade.

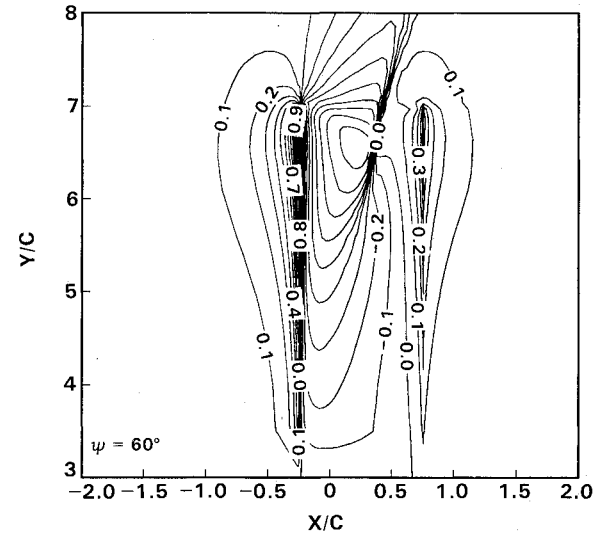

Fig. 5 Quasisteady pressure contours at blade plane for $\Psi = 60$ deg.

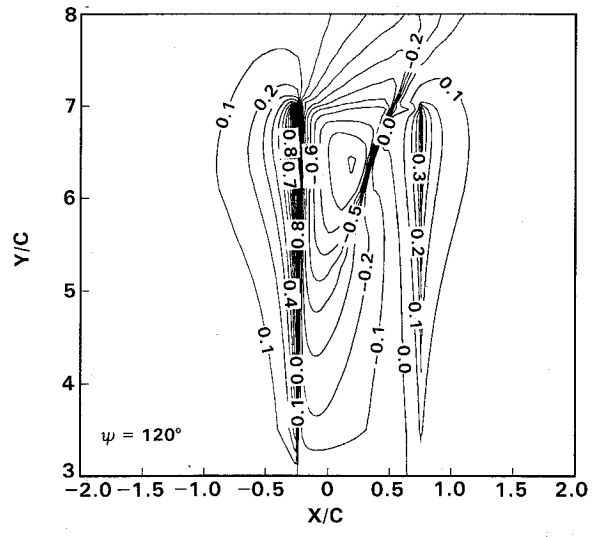

Fig. 6 Quasisteady pressure contours at blade plane for $\Psi = 120$ deg.

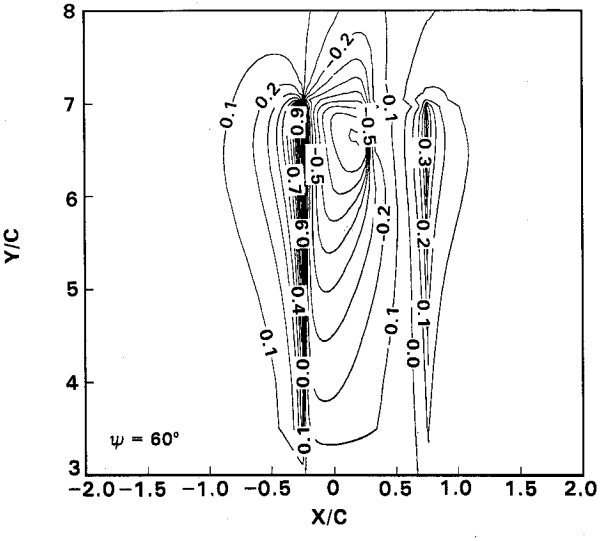

Fig. 7 Unsteady pressure contours at blade plane for $\Psi = 60$ deg.

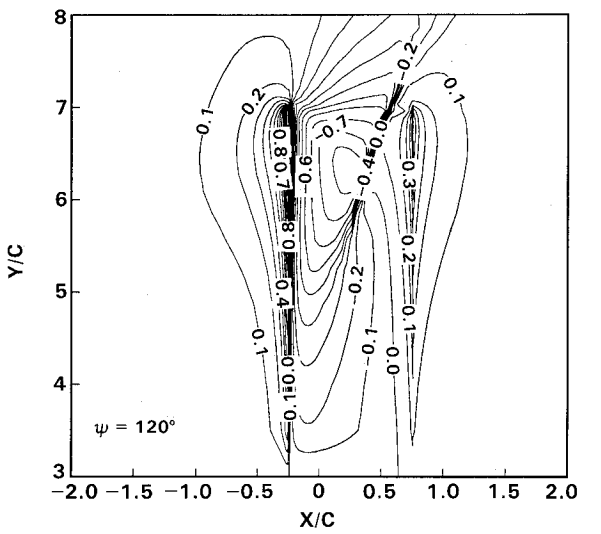

Fig. 8 Unsteady pressure contours at blade plane for $\Psi = 120$ deg.

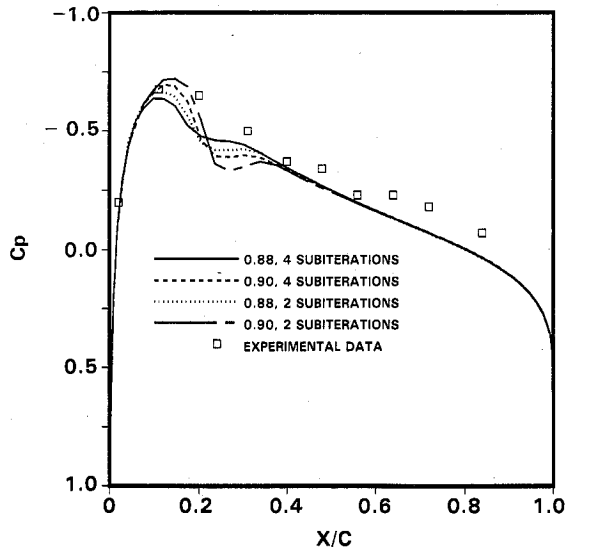

Fig. 9 Surface pressure at $\Psi = 180$ deg with various amounts of subiteration for the rotor in Fig. 2.

ation, the advantage of the present scheme over the partial flux-splitting method is not dramatic for the nonlifting inviscid rotor flow calculation, but it avoids specification of explicit dissipation in the crossflow direction.

Three-dimensional radial-flow effects are examined by comparing quasisteady solutions at $\Psi = 60$ deg (outward radial flow) and $\Psi = 120$ deg (inward radial flow). The pressure contours at the plane of the rotor blade are shown in Figs. 5 and 6. The quasisteady solution neglects the unsteady terms and solves the resulting equations using the instantaneous grid velocity associated with the specified azimuthal angle. This quasisteady flow cannot be investigated experimentally but is easily computed numerically. Comparing the two plots, certain differences in the pressure on the blade surface are observed. The inward radial flow at $\Psi = 120$ deg causes the suction peak to decrease near the tip (supercritical region) and increase near the root (subcritical region). In addition, the shock near the tip is moved slightly aft toward the trailing edge.

The differences in the surface pressure contours between the quasisteady flow and the unsteady flow solutions are seen by comparing Fig. 5 with Fig. 7 and Fig. 6 with Fig. 8. The unsteady solution predicts a weaker shock at $\Psi = 60$ deg than that predicted by the quasisteady solution, whereas the opposite occurs at $\Psi = 120$ deg. The time-lag effects observed in these Euler calculations indicate that the quasisteady approach may not be sufficient to model rotor flows with unsteady shock motions. The preceding observations of three-dimensional and time-lag effects agree well with the full-potential results in Ref. 21.

However, when subcritical flow occurs at certain azimuthal locations (for example, $\Psi = 0.0$ deg), the quasisteady solutions are very close to the unsteady solutions. Therefore, it is possible to start with a quasisteady solution as the initial condition at $\Psi = 0.0$ deg for an unsteady calculation as others have stated.[21,22]

The present code with LU-SGS implicit operator enables marching with two to three times larger time steps than the previous code.[7,8] It requires about 30 μs per grid point per iteration on a CRAY 2, whereas the previous code[7,8] required 75 μs. For the calculations performed here, two or three internal iteration per reasonable time step were usually sufficient to obtain accurate solutions. (A reasonable time step is about one-third of a degree of rotation.) However, the number of internal iterations at each time step may need to be increased when large time steps are used, otherwise the numerical error may play an important role in the solution. This numerical error is due to the increase in linearization and factorization error with increasing time step size and is reduced during the iteration process. In Fig. 9, the solution at $\Psi = 180$ deg with a time step size of half a degree is compared

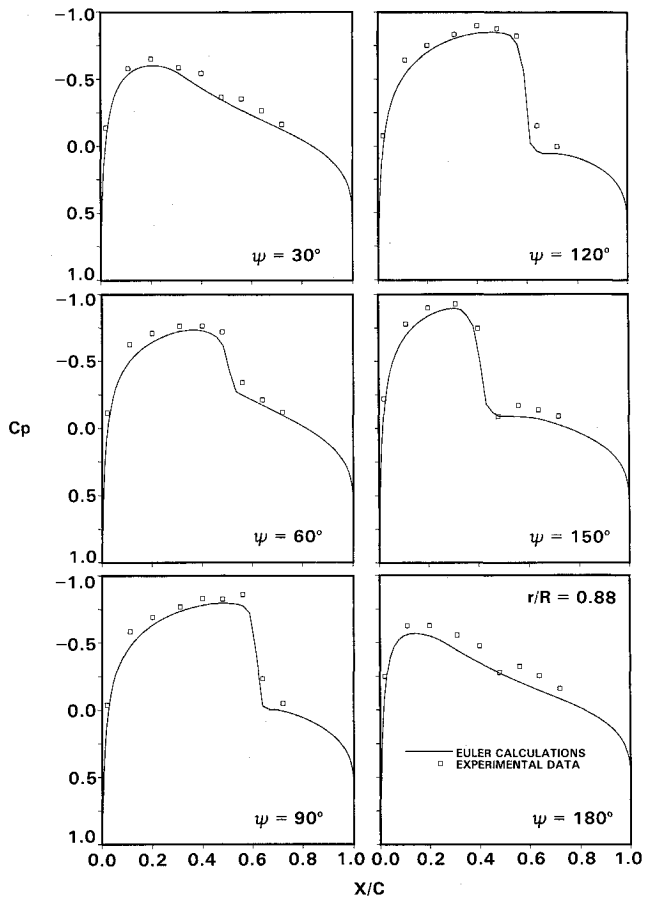

Fig. 10 Surface pressure results at the advancing side for a forward-flight two-blade rotor, $M_T = 0.7634$, $\mu = 0.25$, spanwise location $r/R = 0.88$, aspect ratio = 7.125, nonlifting untwisted untapered NACA 0012 blade.

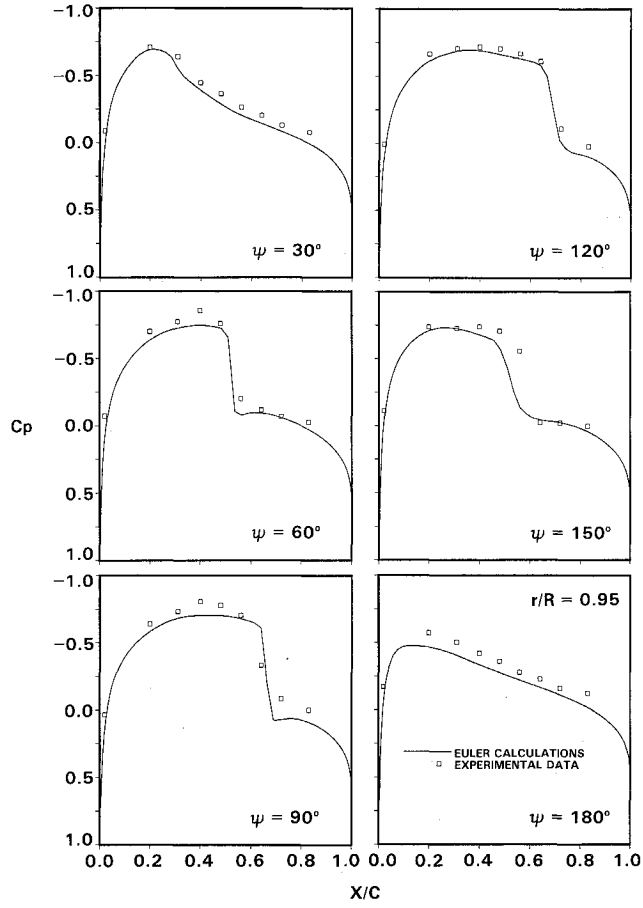

Fig. 11 Surface pressure results at the advancing side for a forward-flight two-blade rotor, $M_T = 0.7634$, $\mu = 0.25$, spanwise location $r/R = 0.95$, aspect ratio = 7.125, nonlifting untwisted untapered NACA 0012 blade.

for two and four subiterations at each time step. The solution is converged after four iterations, but not after two. The comparison in Fig. 9 indicates that the present scheme requires enough subiterations to avoid numerical time lag. Overall, the present code is twice as fast as the previous code[7] for each run. The speed-up is mainly due to the efficiency of the LU-SGS implicit operator.

The flight conditions for the second rotor problem (aspect ratio = 7.125) are $M_T = 0.763$ and $\mu = 0.25$. The grid size used was $121 \times 25 \times 35$ points in an O-H-type grid. The numerical solutions are compared in Figs. 10 and 11 with experimental data at various spanwise locations and at various azimuthal angles. The shock locations are predicted reasonably well. However, the predicted leading-edge suction peaks are not as high as in the experimental data. This phenomenon was not obvious in the previous test case. Some of the discrepancy may be due to tunnel blockage effects (the model occupies about 8% of the test section), mentioned by Bridgeman et al. in Ref. 22.

Concluding Remarks

A finite-volume upwind algorithm for solving the three-dimensional Euler equations with a moving grid has been developed and applied to compute helicopter rotor flows under forward-flight conditions. This code, which is twice as fast as that in Ref. 7, is based on Roe's scheme with limiter, an LU-SGS implicit operator, and quasi-Newton iterations. The numerical results agree well with experimental data as well as with other numerical results. Three-dimensional radial-flow effects, time-lag effects, and some numerical issues regarding the solution for rotor flows are examined using this Euler solver. The residual redistribution due to the coordinate transformation is observed. The primitive variables interpolation is preferred to obtain coordinate-independent solution.

References

[1]Sankar, N. L., Wake, B. E., and Lekoudis, S. G., "Solution of the Unsteady Euler Equations for Fixed and Rotor Wing Configurations," *Journal of Aircraft,* Vol. 23, No. 4, 1986, pp. 283–289.

[2]Chang, I. C., and Tung, C., "Euler Solution of the Transonic Flow for a Helicopter Rotor," AIAA Paper 87-0523, Jan. 1987.

[3]Agarwal, R. K., and Deese, J. E., "An Euler Solver for Calculating the Flowfield of a Helicopter Rotor in Hover and Forward Flight," AIAA Paper 87-1427, June 1987.

[4]Wake, B. E., Sankar, N. L., and Lekoudis, S. G., "Computation of Rotor Blade Flows Using the Euler Equations," *Journal of Aircraft,* Vol. 23, No. 7, 1986, pp. 582–588.

[5]Stahl, H., "Application of a 3D Euler Code to Transonic Blade Tip Blow," *Proceedings of the 12th European Rotorcraft Forum,* Paper No. 29, Garmish-Partenkirchen, Germany, 1986.

[6]Kramer, E., Hertel, J., and Wagner, S., "Computation of Subsonic and Transonic Helicopter Rotor Flow Using Euler Equations," *Proceedings of the 13th European Rotorcraft Forum,* Paper No. 2-14, Arles, France, 1987.

[7]Chen, C. L., McCroskey, W. J., and Ying, S. X., "Euler Solution of Multiblade Rotor Flow," *Vertica,* Vol. 12, Oct. 1988, pp. 303–313.

[8]Chen, C. L., and McCroskey, W. J., "Numerical Simulation of Helicopter Multi-Bladed Rotor Flow," AIAA Paper 88-0046, Jan. 1988.

[9]Chakraverthy, S. R., and Szema, K. Y., "An Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets," AIAA Paper 85-1703, July 1985.

[10]Lawrence, S. L., Tannehill, J. C., and Chaussee, D. S., "An Upwind Algorithm for the Parabolized Navier-Stokes Equations," *AIAA Journal,* Vol. 27, No. 9, 1989, pp. 1175–1183.

[11]Vatsa, V. N., Thomas, J. L., and Wedan, B. W., "Navier-Stokes Computations of Prolate Spheroids at Angle of Attack," AIAA Paper 87-2627, Aug. 1987.

[12]Obayashi, S., "Numerical Simulation of Underexpanded Plumes Using Upwind Methods," AIAA Paper 88-4360, Aug. 1988.

[13]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," AIAA Paper 87-1104, June 1987.

[14]Jameson, A., and Yoon, S., "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA Journal,* Vol. 25, No. 7, 1987, pp. 929-935.

[15]Holmes, D. G., and Tong, S. S., "A Three-Dimensional Euler Solver for Turbomachinery Blade Rows," *Journal of Engineering for Gas Turbines and Power*, Vol. 107, No. 2, 1985, pp. 258-264.

[16]Vinokur, M., "An Analysis of Finite-Difference and Finite-Volume Formulations of Conservation Laws," NASA CR-177416, June 1986.

[17]Parpia, I. H., "Van Leer Flux Vector Splitting in Moving Coordinates," *AIAA Journal,* Vol. 26, No. 1, 1988, pp. 113-115.

[18]Koren, B., "Upwind Schemes, Multigrid and Defect Correction for the Steady Navier-Stokes Equations," *Proceedings of the 11th International Conference on Numerical Methods in Fluid Dynamics,* June 1988, Springer-Verlag, Berlin, 1989, pp. 344-348.

[19]Yee, H., "Implicit TVD Schemes for Hyperbolic Conservation Laws in Curvilinear Coordinates," *AIAA Journal,* Vol. 25, No. 2, 1987, pp. 266-274.

[20]Caradonna, F. X., Laub, G. H., and Tung, C., "An Experimental Investigation of the Parallel Blade-Vortex Interaction," NASA TM-86005, Nov. 1984.

[21]Chang, I. C., "Transonic Flow Analysis for Rotors, Part 2—Three-Dimensional, Unsteady, Full-Potential Calculations," NASA TP-2375, Jan. 1985.

[22]Bridgeman, J. O., Strawn, R. C., and Caradonna, F. X., "An Entropy and Viscosity Corrected Potential Method for Rotor Performance Prediction," *44th Annual Forum of the American Helicopter Society,* American Helicopter Society, Washington, DC, June 1988.