AIAA JOURNAL
Vol. 34, No. 10, October 1996

Unsteady Flow Calculations with a Multigrid Navier–Stokes Method

Feng Liu* and Shanhong Ji†
University of California, Irvine, Irvine, California 92717-3975

A multigrid Navier–Stokes method with a two-equation k–ω turbulence model is developed for calculating unsteady flows. The method uses a staggered finite volume scheme to discretize the Navier–Stokes and the turbulence model equations in space. A fully implicit time-accurate scheme is reformulated so that an efficient explicit multistage method with local time stepping, residual smoothing, and multigrid can be used to accelerate convergence within each unsteady time step. The method combines the advantages of implicit and explicit schemes for unsteady flow calculations. Solutions for the forced oscillation of an airfoil are obtained with the Euler and the Navier–Stokes equations. Results are compared with experimental data.

I. Introduction

UNSTEADINESS is of major importance for many engineering flow problems, such as aeroelasticity and the flow through turbomachinery blade rows. As methods for steady flow calculations become more mature and computers become more powerful, there is a strong interest and need to develop methods for unsteady flow calculations.

Unsteady flow calculations with the Euler and Navier–Stokes equations were carried out by Steger[1] and Chyu et al.[2] with implicit approximate-factorization methods. Approximate-factorization methods were also reviewed recently by Pulliam.[3] Such methods relieve the stringent restrictions on time steps allowable of an explicit method as a result of the Courant–Friedrichs–Lewy (CFL) condition requirement. However, approximate-factorization schemes also introduce factorization errors in time, which in practice limit the maximum allowable time steps.

An explicit finite volume scheme originally developed by Jameson et al.[4] has been very successful for steady flow calculations with both the Euler and the Navier–Stokes equations. Fast convergence can be achieved with the use of local time stepping, residual smoothing, and multigrid. Unfortunately, these techniques cannot be applied directly to time accurate calculations. Jameson[5] extended the use of these techniques in an implicit time stepping scheme by reformulating the implicit Euler equations for each unsteady time step into a pseudotime marching problem. Local time stepping, residual averaging, and multigrid can then be used for accelerating convergence for this pseudo-time-marching problem. Once it converges, one gets a time-accurate solution of the original implicit equations for one time step. Jameson[5] obtained the solution of the Euler equations for oscillating airfoils and wings. Alonso and Jameson[6] used this method to solve an aeroelastic problem where the two-dimensional Euler equations are coupled with structural modal equations simulating a section of a wing with bending and torsion. Arnone et al.[7,8] applied this method to Navier–Stokes calculations for laminar and turbulent flows with the Baldwin–Lomax turbulence model. Most recently, Alonso et al.[9] presented results for the Navier–Stokes equations with the Baldwin–Lomax turbulence model and a new H-cusp dissipation scheme.

In this paper we will present results of the present authors' independent work in using the multigrid method for unsteady Navier–Stokes flow calculations. The major contribution of this paper is the use of the implicit multigrid scheme for unsteady flows in combination with a novel staggered finite volume scheme for the Navier–Stokes and the k–ω turbulence model equations initially developed by Liu and Zheng[10–12] for steady-state flows. In the next sections we will describe the reformulation of an implicit time-stepping scheme and our spatial discretization for the Navier–Stokes and k–ω equations. We will then present results for the unsteady flow over a NACA 64A010 airfoil oscillating around its quarter-chord. Results of the Euler and the Navier–Stokes equations with the k–ω equations will be compared with experimental data.

II. Governing Equations

Consider a moving and possibly deforming control volume Ω in two dimensions whose boundary $\partial\Omega$ moves at velocity $\boldsymbol{V}_b = (u_b, v_b)$. The governing equations for an unsteady compressible turbulent flow with a k–ω model by Wilcox[13] can be written in integral form for such a control volume as

$$\frac{\partial}{\partial t}\iint_\Omega \boldsymbol{w}\,\mathrm{d}\Omega + \oint \boldsymbol{f}\,\mathrm{d}S_x + \boldsymbol{g}\,\mathrm{d}S_y = \oint \boldsymbol{f}_\mu\,\mathrm{d}S_x + \boldsymbol{g}_\mu\,\mathrm{d}S_y + \iint_\Omega \boldsymbol{S}\,\mathrm{d}\Omega \tag{1}$$

where

$$\boldsymbol{w} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ \rho k \\ \rho\omega \end{bmatrix} \tag{2}$$

$$\boldsymbol{f} = \begin{bmatrix} \rho\bar{u} \\ \rho u\bar{u} + p \\ \rho v\bar{u} \\ \rho E\bar{u} + pu \\ \rho k\bar{u} \\ \rho\omega\bar{u} \end{bmatrix}, \qquad \boldsymbol{g} = \begin{bmatrix} \rho\bar{v} \\ \rho u\bar{v} \\ \rho v\bar{v} + p \\ \rho E\bar{v} + pv \\ \rho k\bar{v} \\ \rho\omega\bar{v} \end{bmatrix} \tag{3}$$

$$\boldsymbol{f}_\mu = \begin{bmatrix} 0 \\ \hat{\tau}_{xx} \\ \hat{\tau}_{yx} \\ u\hat{\tau}_{xx} + v\hat{\tau}_{yx} + (\mu + \sigma^*\mu_T)\dfrac{\partial k}{\partial x} - q_x \\ (\mu + \sigma^*\mu_T)\dfrac{\partial k}{\partial x} \\ (\mu + \sigma\mu_T)\dfrac{\partial \omega}{\partial x} \end{bmatrix} \tag{4}$$

Received Sept. 22, 1995; revision received March 28, 1996; accepted for publication April 2, 1996.

*Assistant Professor, Department of Mechanical and Aerospace Engineering. Member AIAA.
†Graduate Student Researcher, Department of Mechanical and Aerospace Engineering. Student Member AIAA.

$$g_\mu = \begin{bmatrix} 0 \\ \hat{\tau}_{xy} \\ \hat{\tau}_{yy} \\ u\hat{\tau}_{xy} + v\hat{\tau}_{yy} + (\mu + \sigma^*\mu_T)\dfrac{\partial k}{\partial y} - q_y \\ (\mu + \sigma^*\mu_T)\dfrac{\partial k}{\partial y} \\ (\mu + \sigma\mu_T)\dfrac{\partial \omega}{\partial y} \end{bmatrix} \tag{5}$$

$$S = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \tau_{ij}\dfrac{\partial u_i}{\partial x_j} - \beta^*\rho\omega k \\ \dfrac{\alpha\omega}{k}\tau_{ij}\dfrac{\partial u_i}{\partial x_j} - \beta\rho\omega^2 \end{bmatrix} \tag{6}$$

In the preceding equations t is time, ρ density, p pressure, μ molecular viscosity, k turbulent mixing energy, and ω the specific dissipation rate; $i = 1, 2$ indicates the two coordinate directions. The terms x_i, or x and y, stand for the position vectors; u_i, or u and v, are the flow velocity components in the x and y directions; and $(\bar{u}, \bar{v}) = (u - u_b, v - v_b)$ stands for the local convective velocity relative to the control surface.

The total energy and enthalpy are $E = e + k + u_i u_i/2$ and $H = h + k + u_i u_i/2$, respectively, with $h = e + p/\rho$, and $e = p/[(\gamma - 1)\rho]$. The term γ is the ratio of specific heats. Other quantities are defined in the following equations:

$$\mu_T = \alpha^*(\rho k/\omega) \tag{7}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{8}$$

$$\tau_{ij} = 2\mu_T\left(S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho k\delta_{ij} \tag{9}$$

$$\hat{\tau}_{ij} = 2\mu\left(S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) + \tau_{ij} \tag{10}$$

$$q_j = -\left(\frac{\mu}{Pr_L} + \frac{\mu_T}{Pr_T}\right)\frac{\partial h}{\partial x_j} \tag{11}$$

where Pr_L and Pr_T are the laminar and turbulent Prandtl numbers, respectively. The molecular viscosity μ is calculated by Sutherland's law. The closure coefficients are

$$\begin{aligned} &\beta = \tfrac{3}{40}, \qquad \beta^* = \tfrac{9}{100}, \qquad \alpha = \tfrac{5}{9} \\ &\alpha^* = 1, \qquad \sigma = \tfrac{1}{2}, \qquad \sigma^* = \tfrac{1}{2} \end{aligned} \tag{12}$$

When the Baldwin–Lomax algebraic turbulence model is used, the last two equations (1) are not solved and μ_T is replaced by that obtained with the Baldwin–Lomax model. Implementation of the Baldwin–Lomax model used in this paper is described in Liu and Jameson.[14]

III. Multigrid Method for Unsteady Flows

After the spatial discretization, Eq. (1) can be written in the following semidiscrete form:

$$\frac{d\boldsymbol{w}}{dt} + R(\boldsymbol{w}) = 0 \tag{13}$$

where R is the vector of residuals, consisting of the spatially discretized flux balance of Eq. (1).

The semidiscrete equations in Eq. (13) describe the local evolution of the system in the neighborhood of each mesh point. The underlying idea of a multigrid time-stepping scheme is to transfer some of the task of tracking the evolution of the system to a sequence of successively coarse meshes. This has two advantages. First, the amount of computation needed on a coarse mesh per time step is less. Second, the evolution of the system takes place over larger computational volumes and, therefore, may reach global equilibrium at a faster rate. If the equations in Eq. (13) are solved by an explicit multistage method, this manifests itself through the possibility of allowing larger time steps on the coarse meshes.

A multistage time-stepping scheme proposed by Jameson[15] was found to be particularly suitable for use with a multigrid method[12,14,15] to calculate steady flows. This explicit time-stepping scheme can be written as

$$\begin{aligned} \boldsymbol{w}^{(n+1,0)} &= \boldsymbol{w}^n \\ &\cdots \\ \boldsymbol{w}^{(n+1,q)} &= \boldsymbol{w}^n - \alpha_q \Delta t R\left[\boldsymbol{w}^{(n+1,q-1)}\right] \\ &\cdots \\ \boldsymbol{w}^{n+1} &= \boldsymbol{w}^{(n+1,m)} \end{aligned}$$

where m is the total number of stages to be used and Δt is the time step.

Multigrid methods in its original format, however, are only applicable to steady flow calculations because time accuracy is lost through the use of different time steps on the different levels of grids. To take advantage of the fast convergence property of the multigrid method without sacrificing time accuracy, Jameson[5] reformulated Eq. (13) as follows.

Consider a fully implicit scheme for Eq. (13):

$$D_t(\boldsymbol{w}^{n+1}V^{n+1}) + R(\boldsymbol{w}^{n+1}) = 0 \tag{14}$$

where V is the volume of a computational cell. The term D_t is a kth-order accurate backward difference operator,

$$D_t = \frac{1}{\Delta t}\sum_{q=1}^{k}\frac{1}{q}(\Delta^-)^q$$

where

$$\Delta^-\boldsymbol{w}^{n+1} = \boldsymbol{w}^{n+1} - \boldsymbol{w}^n$$

This implicit scheme is A-stable for $k = 1, 2$. In the present work, the second-order scheme corresponding to $k = 2$ is used. In this case, Eq. (13) is reformulated into

$$\frac{d\boldsymbol{w}}{dt^*} + R^*(\boldsymbol{w}) = 0 \tag{15}$$

where

$$\begin{aligned} R^*(\boldsymbol{w}) &= R(\boldsymbol{w}) + (3/2\Delta t)(\boldsymbol{w}V^{n+1}) \\ &\quad - (2/\Delta t)(\boldsymbol{w}^n V^n) + (1/2\Delta t)(\boldsymbol{w}^{n-1}V^{n-1}) \end{aligned}$$

where t^* is a pseudotime. The solution of the implicit equation (14) is now made equivalent to the steady-state solution of Eq. (15) with the pseudotime t^*. We can then apply our efficient multigrid method to solving Eq. (15).

With the preceding reformulation, techniques such as local time stepping and residual smoothing can still be used to accelerate convergence of the subiterations. Each time step needed about only 50 multigrid pseudotime cycles to reach a converged solution. Being an explicit pointwise scheme within each implicit time step, methods along this line also have the advantage that they are best suitable for optimized performance on parallel computers as was demonstrated by Alonso et al.[9]

IV. Spatial Discretization

The major contribution of this paper is the use of the preceding time-accurate implicit multigrid method in combination with a novel staggered finite volume solver for the Navier–Stokes and the two-equation k–ω turbulence model equations. In the staggered finite volume scheme, the computational domain is discretized into quadrilateral cells as shown in Fig. 1. A regular cell-centered scheme is used for the Navier–Stokes equations in which the flow variables ρ, ρu_i, and ρE are defined at the cell centers marked by the circles in Fig. 1. Velocity gradients and viscous stresses are calculated at the cell vertices by using Gauss's formula as detailed in Liu and

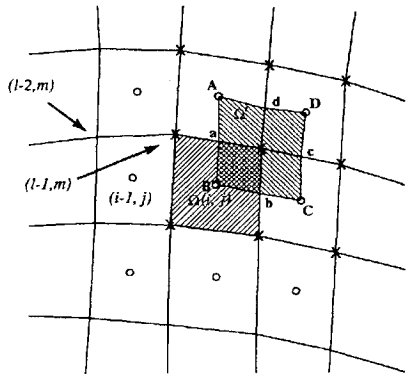

Fig. 1 Staggered control volumes for a mixed cell-center and cell-vertex scheme. ○, cell-centers: location for ρ, ρu_i, ρE, $\partial k/\partial x_j$, $\partial \omega/\partial x_j$; ×, cell-vertex: location for k, ω, $\partial u_i/\partial x_j$, $\partial h/\partial x_j$; Ω, control volume for the Navier–Stokes equations; and Ω', control volume for the k–ω equations.

Zheng.[10] The diffusive flux across a cell face is then obtained by using the averages of the nodal points of the corresponding diffusive terms. This yields a discretization stencil involving nine points with minimum spatial extent as shown by the circles in Fig. 1.

To solve the k–ω equations one could either define k and ω at the cell centers or the cell vertices. If k and ω were defined at the cell centers, one would have to interpolate the strain tensor calculated at the cell vertices to the cell center of Ω so that the production terms for the control volume can be evaluated. On the other hand, the eddy viscosity μ_T calculated from k and ω at the cell centers must be translated to the cell vertices to calculate the turbulent stresses there. This double averaging process would broaden the final discretization stencil for the coupled Navier–Stokes and the k–ω equations and thus reduce the accuracy and increase the likelihood of uninhibited growing modes.

Alternatively, one can define k and ω at the cell vertices and use the staggered control volume Ω' to integrate the k–ω equations. The discretization is done in a similar fashion as that for the Navier–Stokes equations, but in the reverse order of using the original and the auxiliary cells. Because the variables k and ω are defined at the cell vertices marked by the crosses, we will no longer need the excessive averaging steps for the strain tensor and the eddy viscosity. The production terms are evaluated at exactly the same locations, namely the cell vertices, where the stress and strain tensors are calculated, and the eddy viscosity calculated from the k and ω at these cell vertices are directly used to calculate the turbulent stress tensors. In this way, the Navier–Stokes equations and the k–ω equations in their discrete forms are coupled as closely as possible. The discretization of each set of the equations involves a stencil of only nine points: those for the Navier–Stokes equations are shown by the circles and those for the k–ω equation by the crosses in Fig. 1. This staggered finite volume approach has been successfully tested for steady flow calculations.[10,11]

For unsteady flow calculations, care must be taken in constructing the convective fluxes to take into account the effect of grid motion. Let (i, j) be the index numbers for the cell centers and (l, m) be the index numbers for the cell vertices, as shown in Fig. 1. In a central-differencing type of scheme, the Euler flux through the cell face $(i-\frac{1}{2}, j)$ is calculated as

$$F_{i-\frac{1}{2},j} = \frac{1}{2}(\boldsymbol{w}_L + \boldsymbol{w}_R)U + \frac{p_L + p_R}{2}\begin{bmatrix} 0 \\ S_x \\ S_y \\ \hat{u}S_x + \hat{v}S_y \end{bmatrix}$$

where the subscripts L and R denote the flow variables at the cell centers on the left- and right-hand sides of the cell interface, respectively;

$$U = (\hat{u} - u_b)S_x + (\hat{v} - v_b)S_y$$

$$\hat{u} = \frac{(\rho u)_L + (\rho u)_R}{\rho_L + \rho_R}, \qquad \hat{v} = \frac{(\rho v)_L + (\rho v)_R}{\rho_L + \rho_R}$$

where S_x and S_y are the x and y components of the cell face area vector that point to the direction of increasing grid index number i. Artificial dissipation of blended second and fourth differences of the flow variables $\boldsymbol{w}$ are formulated in the same way as for the steady-state calculations with fixed grids. However, the maximum eigenvalue used to scale the dissipation terms must include the grid moving velocity, i.e.,

$$|\lambda_{\max}| = |U| + c\sqrt{S_x^2 + S_y^2}$$

where c is the sound speed at the cell interface.

In the upwind-type scheme with Roe splitting, the left and right states are obtained by second-order MUSCL interpolation.[11] The interface flux is then obtained as

$$F_{i-\frac{1}{2},j} = \tfrac{1}{2}[F(\boldsymbol{w}_L) + F(\boldsymbol{w}_R)] - \tfrac{1}{2}|\tilde{A}|(\boldsymbol{w}_R - \boldsymbol{w}_L) \tag{16}$$

where $\tilde{A}$ is the Jacobian of F evaluated with Roe-averaged flow variables (denoted by a $\tilde{}$) so that

$$F(\boldsymbol{w}_R) - F(\boldsymbol{w}_L) = \tilde{A}(\boldsymbol{w}_R - \boldsymbol{w}_L) = \tilde{T}\tilde{\Lambda}\tilde{T}^{-1}(\boldsymbol{w}_R - \boldsymbol{w}_L)$$

In the case of a moving grid

$$F(\boldsymbol{w}_L) = \boldsymbol{w}_L[(u_L - u_b)S_x + (v_L - v_b)S_y] + p_L\begin{bmatrix} 0 \\ S_x \\ S_y \\ u_L S_x + v_L S_y \end{bmatrix}$$

$$F(\boldsymbol{w}_R) = \boldsymbol{w}_R[(u_R - u_b)S_x + (v_R - v_b)S_y] + p_R\begin{bmatrix} 0 \\ S_x \\ S_y \\ u_R S_x + v_R S_y \end{bmatrix}$$

Notice that u_b, v_b, S_x, and S_y in the preceding equations are defined at the cell interface and are independent of $\boldsymbol{w}$. Thus, Eq. (16) will still hold if the diagonal matrix $\tilde{\Lambda}$ consisting the eigenvalues of $\tilde{A}$ includes the grid velocity, i.e.,

$$\tilde{\Lambda} = \text{diag}\left\{\tilde{U}, \tilde{U}, \tilde{U} + \tilde{c}\sqrt{S_x^2 + S_y^2}, \tilde{U} - \tilde{c}\sqrt{S_x^2 + S_y^2}\right\}$$

where

$$\tilde{U} = (\tilde{u} - u_b)S_x + (\tilde{v} - v_b)S_y$$

Therefore, all of the formulas derived for the fixed grid apply to a moving grid as long as the grid velocity is included in the calculation of F_L and F_R and also the eigenvalues of $\tilde{A}$.

For the convective terms in the k and ω equations, a second-order simple flux-splitting scheme is used because the k and ω equations have very simple wave structures that consist of essentially the flow convective velocities in the two coordinate directions. If the estimated normal convective velocity $\bar{u}S_x + \bar{v}S_y$ at the interface AaB in Fig. 1 is positive, a second-order upwind interpolation formula may be used to obtain the values k and ω at the interface midpoint a,

$$(k)_a = \tfrac{1}{2}[3(k)_{l-1,m} - (k)_{l-2,m}] \tag{17}$$

$$(\omega)_a = \tfrac{1}{2}[3(\omega)_{l-1,m} - (\omega)_{l-2,m}] \tag{18}$$

These values are then used to form the convective fluxes through the interface AaB. For example, the flux for ρk can be calculated as

$$f = \rho(\bar{u}S_x + \bar{u}S_y)(k)_a$$

In the unsteady calculations presented in this paper, a C mesh is generated over an airfoil. The grid is rigidly fixed to and moves with the airfoil. The motion of the airfoil is specified analytically. At every time instant, the coordinates and grid velocities (u_b, v_b) of the grid vertices and the midpoints of the cell interfaces are calculated according to the analytically specified airfoil movement.

After spatial discretization, the reformulated pseudo-time-dependent system of equations, which includes both the Navier–

Stokes equations and the k–ω equations, is marched in the pseudotime with a strongly coupled multigrid method and the multistage time-stepping scheme. The strongly coupled multigrid method described in Ref. 12 integrates the Navier–Stokes and the k–ω turbulence model equations simultaneously in time with the same multigrid cycles without any lagging of the velocity or the k and ω values in a typical loosely coupled approach. Fast convergence to steady-state solution was achieved with this strongly coupled approach for both airfoil and turbomachinery flows. This method is particularly suitable for the current multigrid strategy for unsteady flows since within each time step the equations are formulated as a steady-state problem with pseudotime.

V. Results and Discussions

The method has been used to calculate the flow over a NACA 64A010 airfoil pitching around its quarter-chord. Experimental data were provided in Ref. 16. The pitching motion of the airfoil is described by the following equation:

$$\alpha(t) = \alpha_m + \alpha_0 \sin \omega t \tag{19}$$

where $\alpha(t)$ is the instantaneous angle of attack, α_0 is the pitching range, and α_m is the mean angle of attack. The angular frequency ω is related to the reduced frequency defined as

$$\kappa = \omega c / 2U_\infty \tag{20}$$

where c is the chord of the airfoil and U_∞ is the freestream speed of the flow. A C mesh is generated using a hyperbolic grid generator. Test case CT6 is studied. The freestream Mach number is $M_\infty = 0.796$, and $\alpha_0 = 1.01$ deg, $\alpha_m = 0$ deg, and $\kappa = 0.202$. The flow Reynolds number is 1.3×10^7.

With an explicit scheme, the minimum allowable time step as a result of stability over the complete computational domain must be used for time accuracy for small κ. Let the minimum grid size in the computational domain be Δx. Since the maximum wave speed $u + a$ is at least the sound speed a even within the boundary layer in a viscous flow calculation and that $a \sim U_\infty$ for transonic flows, the allowable time step for an explicit scheme with CFL number of order 1 can be estimated as

$$\Delta t_e \sim (\Delta x / U_\infty) = (c/U_\infty)(\Delta x / c)$$

The preceding estimate is based on only time step limits for the convective part of the Euler or Navier–Stokes equations; time step limits for the diffusive part of the Navier–Stokes equations may be more stringent within thin boundary layers.

For an A-stable implicit scheme the time steps are not governed by a numerical stability limit but rather by the time accuracy needed

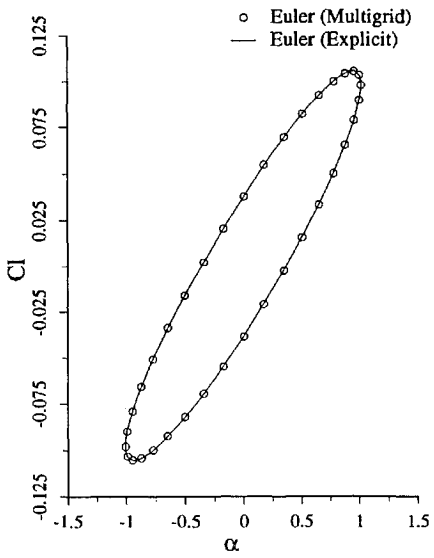

Fig. 2 C_l vs α computed with explicit and implicit multigrid methods.

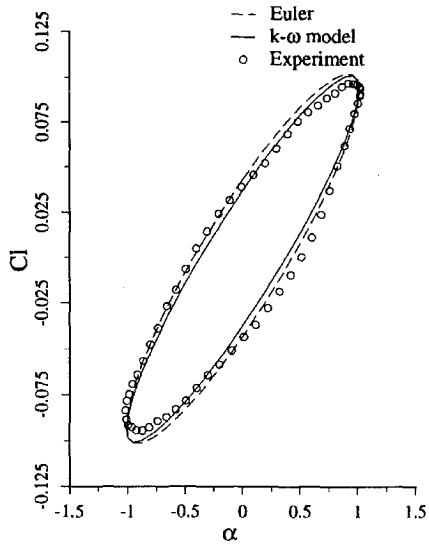

Fig. 3 Comparison of computed C_l vs α with experimental data.

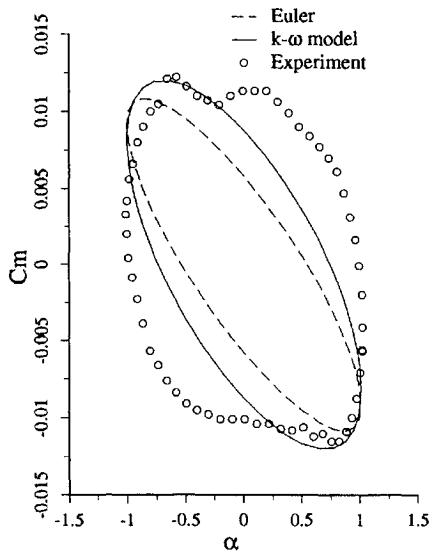

Fig. 4 Comparison of computed C_m vs α with experimental data.

for resolving the physical unsteadiness of the flow, in this case, the reduced frequency κ given in Eq. (20) or the period

$$T = 2\pi/\omega = \pi c / U_\infty \kappa$$

For a second-order scheme, 32 time steps within one period are enough to resolve the variations. The time step for an implicit scheme is then

$$\Delta t_i = (\pi/32)(c/U_\infty \kappa)$$

If the work needed to march one time step with the implicit scheme is N times of that needed to march one time step with the explicit scheme, the ratio of computational work needed by the two schemes to simulate the same length of time is

$$\frac{\text{work for explicit scheme}}{\text{work for implicit scheme}} = \frac{\Delta t_i}{N \Delta t_e} \sim \frac{1}{10N} \frac{1}{\kappa} \frac{c}{\Delta x}$$

Thus, we can see that the benefit of the implicit scheme relative to an explicit scheme is large when the physical unsteadiness is of low frequency, the grid size needed for spatial resolution is small, and, of course, N is small. The value of N is directly related to the effectiveness of the multigrid scheme in the current approach.

With Euler calculations, the minimum grid size Δx is typically of the order 10^{-3}–10^{-4} chord, and 30 multigrid cycles within each

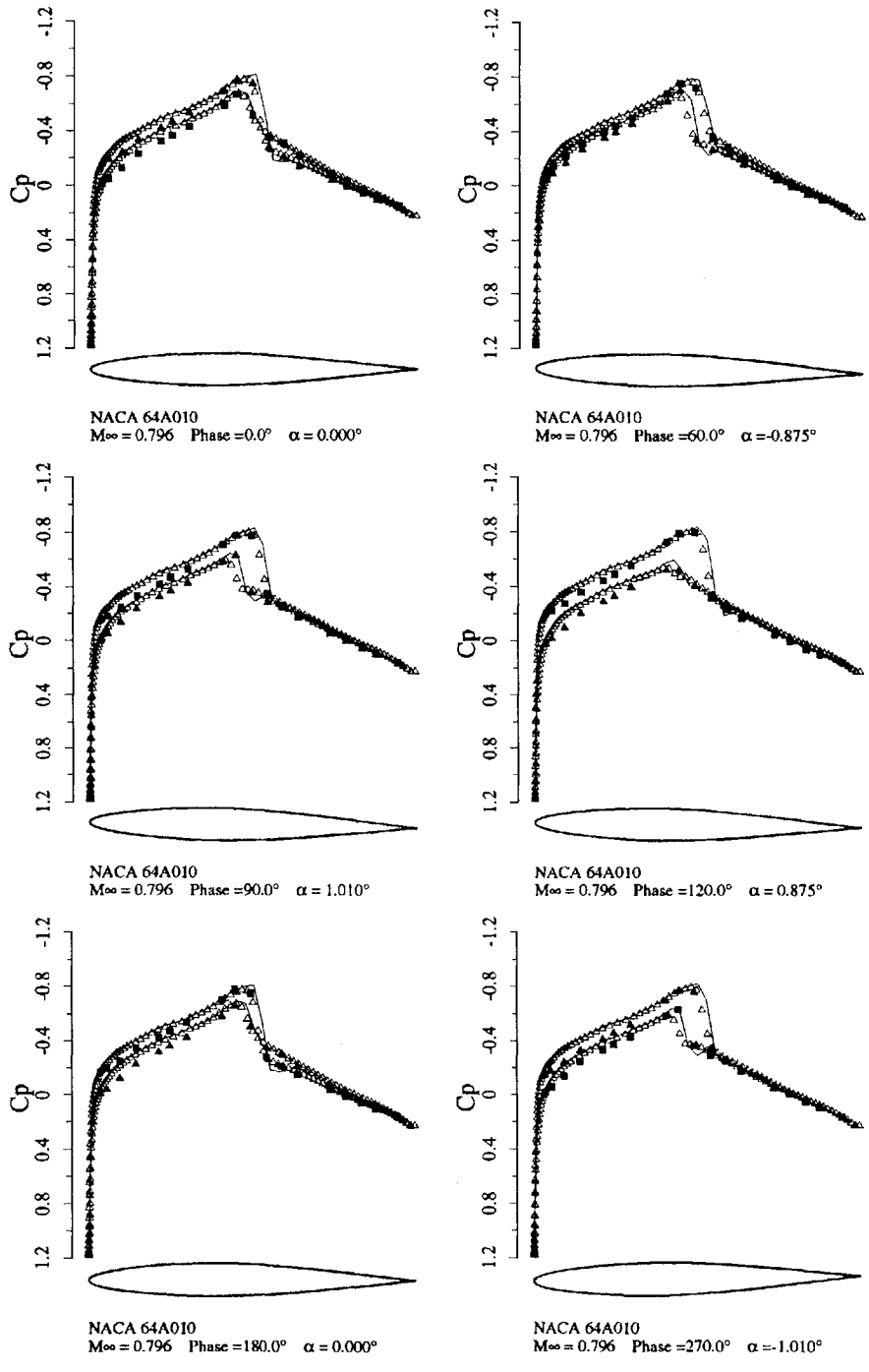

Fig. 5 Time histories of surface pressures. Experiment: △, k–ω model; ——, Euler; ■, upper surface; and ▲, lower surface.

time step are enough to drive the residual of each implicit time step iteration to the level of 10^{-4}, which is more than enough to guarantee a converged solution. From the preceding estimate we see a benefit factor on the order of 100, depending on the actual grid size.

To verify the implicit scheme with the reformulated multigrid method and to demonstrate its efficiency compared with a direct explicit approach, solutions of the Euler equations are obtained. In this case, the central-differencing scheme is used with both approaches on a mesh of 161 × 49 grid points. The computations are started by pitching the airfoil impulsively in a uniform flow according to Eq. (19). After four to five periods, the computed flow reaches a periodic solution. Thirty multigrid cycles are used within each implicit time step, and 36 implicit time steps are used within each period of airfoil oscillation. The implicit scheme takes 75 s on an SGI Indigo 2 to march one real time step with 30 multigrid cycles, whereas the explicit scheme needs 3800 s to march to the same time level. The gain in computational efficiency with the multigrid implicit scheme is more than 50. The computed C_l vs α curves by both schemes are compared in Fig. 2. As can be seen, there is no difference in the final periodic solutions at all, demonstrating the accuracy of the multigrid implicit scheme while achieving high efficiency gains.

For Navier–Stokes calculations, we may need 50–100 multigrid cycles to drive the residuals to an acceptable value (normally 10^{-2}–10^{-3}) because of the use of grids of high stretching and high aspect ratios needed for high-Reynolds-number flows. Notice, however, that the minimum grid size, Δx, needed for Navier–Stokes computations is much smaller, particularly for high-Reynolds-number flows. Typical values are on the order of $10^{-5}c$ or even smaller. The benefit factor will then reach the level of thousands even with $N = 100$.

Figure 3 shows the lift coefficient vs angle of attack within one period of oscillation calculated by the Euler and the Navier–Stokes codes with the multigrid implicit method. In this case, we have

used the upwind Roe scheme to achieve better shock and boundary-layer resolution. The Euler solution was obtained on the previously mentioned 161 × 49 mesh, whereas the Navier–Stokes calculations were done on a 257 × 65 mesh with a minimum grid size of 2×10^{-5} chord length; this minimum spacing corresponds to $y+$ values from 1.2 to 4.8 for the unsteady Navier–Stokes calculation. From Fig. 3 it seems that both the Euler and the Navier–Stokes solutions agree well with the experimental data, although it is noticeable that the Navier–Stokes solution with the k–ω model has a slightly smaller tilt angle, leaning closer to the experimental data.

Figure 4 shows the moment coefficient vs angle of attack as compared with the experimental data. For the moment coefficient, the Navier–Stokes solution appears to offer some improvement over the Euler solution. However, the computation misses two pocket-like distributions of the experimental data between the maximum and minimum angles of attack. The present authors are not able to explain this. But aerodynamic moment is very sensitive to shock locations. The details of shock/boundary-layer interaction may affect the moment significantly.

Figure 5 shows the pressure coefficient distribution on the airfoil compared with the experimental data at the phase angles of 0, 60, 90, 120, 180, and 270 deg as the airfoil pitches up and then down. As the airfoil pitches up from zero angle of attack, the shock wave on the top surface moves backwards and becomes stronger. The shock wave on the lower surface moves forward and becomes weaker. The lift coefficient increases as shown in Fig. 3. As the airfoil further pitches up through phase angle 60 towards the maximum angle of attack at phase angle 90, the shock on the top surface becomes stronger and stronger and the shock on the lower surface becomes very weak. The airfoil reaches its maximum lift coefficient after it has slightly passed its maximum angle of attack and already started pitching down as can be seen in Fig. 3. At phase angle 120, the shock on the lower surface has almost disappeared. When the airfoil further pitches down, the shock on the top surface starts to move forward, and the shock on the lower surface reappears, moves backwards, and becomes stronger. The lift coefficient drops down. Because of symmetry, the behavior of the flow repeats itself on the lower surface of the airfoil.

The computed pressure distributions on the airfoil shown in Fig. 5 indicate general agreement with the experimental data with some discrepancy in the forward portion of the airfoil where the computations generally show a higher suction than the experiments. The differences between the Euler and the Navier–Stokes solutions are only significant near the shock waves. The Euler calculations typically predict shock locations further downstream and overpredict the shock strength. The Navier–Stokes calculations predict better shock location and strength compared with the experimental data. Because of the use of Roe flux difference splitting and second-order MUSCL interpolation, the computational shock waves remain sharp with a two-point shock structure. For Euler calculations, a one-point shock structure is generally obtained.

The time variation of the computed and experimental data of surface pressure distributions can be expressed in terms of their Fourier components. Figure 6 shows the comparison between the Navier–Stokes solution and the experimental unsteady pressure for their first three Fourier modes. The modal content decreases rapidly with the increasing of the order of the mode number, except near the shock wave region. It can be seen that good agreement is obtained between the computational result and the experimental data.

The effect of grid density is studied following the approach of Roache.[17] Grid convergence index (GCI) is calculated for both spatial and time step refinement. The GCI is defined as

$$\text{GCI [fine grid]} = 3|\epsilon|/(r^p - 1)$$

$$\text{GCI [coarse grid]} = \frac{3|\epsilon| r^p}{r^p - 1}$$

$$\epsilon = \frac{f_2 - f_1}{\max(f_1)}$$

where f_2 is the coarse grid solution, f_1 is the fine grid solution, p is the order of accuracy of the scheme, and r is the ratio of the number of grid points of the fine grid to that of the coarse grid. In our case

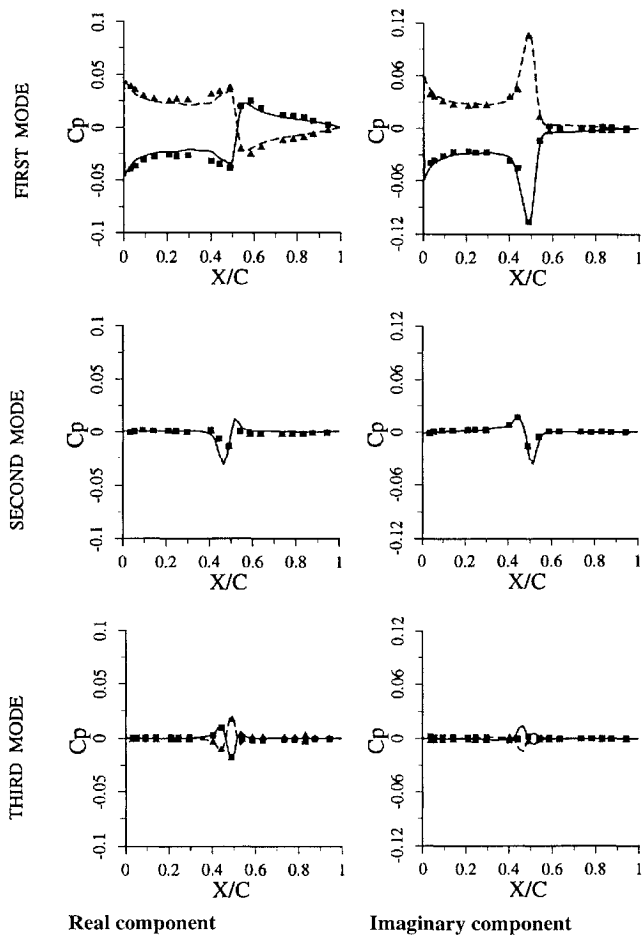

Fig. 6 Complex representation of surface pressure variations. Upper surface: ——, computed and ■, experiment. Lower surface: ——, computed and ▲, experiment.

$p = 2$ and $r = 2$, corresponding to a double grid refinement with a second-order scheme.

The GCI study was performed on the lift coefficient result. For the spatial refinement, with a 513 × 129 fine grid and a 257 × 65 coarse grid, the mean square root GCI (coarse grid) is 13%. The grid refinement on time steps is based on a 257 × 65 grid. With 36 or 18 time steps per oscillating period, GCI (fine grid) is 3%.

The time needed to simulate one period of the airfoil oscillation with the implicit multigrid Euler method on the 161 × 49 grid with a central-differencing scheme is 70 min. The full Navier–Stokes code with the k–ω two-equation turbulence model and Roe's upwind scheme takes 2930 s on an SGI Indigo 2 with an R4400 CPU to complete 100 multigrid cycles within one real time step on a 257 × 65 mesh, corresponding to a time of 29 h for one period computation with 36 real time steps. To calculate the same flow with an explicit scheme for the Navier–Stokes equations would be practically unfeasible.

VI. Concluding Remarks

A fully implicit time-accurate multigrid algorithm is developed for solving the Euler and the Navier–Stokes equations with a two-equation k–ω turbulence model. The scheme eliminates the CFL stability limit by using a fully implicit time-accurate discretization while the time-required at each time step is small and comparable with that of an explicit time-marching scheme. Local time stepping, residual smoothing, and multigrid techniques can be used to accelerate the convergence for the solution of the implicit time-stepping equations through the introduction of a pseudo-time-marching problem. Unsteady flow over an oscillating airfoil was computed by the Euler and the Navier–Stokes codes. The Navier–Stokes solution agrees well with experimental data. The gain in computational efficiency is significant for flows where the unsteady physical time

scale is much greater than the usually small time scale governed by the stability limit as a result of a finite acoustic wave speed and small grid sizes needed for high-Reynolds-number flows. Further studies should be directed to validation of turbulence models for unsteady separated flows and improvement of multigrid convergence for computations involving highly stretched and high-aspect-ratio grids.

Acknowledgments

This research has been funded by the National Science Foundation (NSF) under Grant CTS-9410800; the technical monitor is Robert Powell. Partial support has also been provided by the University of California Energy Institute under Contract UERG-19244. Computer time was provided by the Office of Academic Computing at the University of California, Irvine, and the NSF San Diego Supercomputer Center. We also want to thank the reviewers and the associate editor for their comments, which helped bring about a better presentation of this paper.

References

[1]Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries," *AIAA Journal*, Vol. 16, No. 7, 1978, pp. 679–686.

[2]Chyu, W. J., Davis, S. S., and Chang, K. S., "Calculation of Unsteady Transonic Flow over an Airfoil," *AIAA Journal*, Vol. 19, No. 6, 1981, pp. 684–690.

[3]Pulliam, T. H., "Time Accuracy and the Use of Implicit Methods," *Proceedings of the AIAA 11th Computational Fluid Dynamics Conference*, Pt. 2, AIAA, Washington, DC, 1993 (AIAA Paper 93-3360).

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

[5]Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA Paper 91-1596, June 1991.

[6]Alonso, J. J., and Jameson, A., "Fully-Implicit Time-Marching Aeroelastic Solutions," AIAA Paper 94-0056, Jan. 1994.

[7]Arnone, A., Liou, M. S., and Povinelli, L. A., "Multigrid Time-Accurate Integration of Navier–Stokes Equations," *Proceedings of the AIAA 11th Computational Fluid Dynamics Conference*, Pt. 2, AIAA, Washington, DC, 1993 (AIAA Paper 93-3361).

[8]Arnone, A., Liou, M. S., and Povinelli, L. A., "Integration of Navier–Stokes Equations Using Dual Time Stepping and a Multigrid Method," *AIAA Journal*, Vol. 33, No. 6, 1995, pp. 985–990.

[9]Alonso, J. J., Martinelli, L., and Jameson, A., "Multigrid Unsteady Navier–Stokes Calculations with Aeroelastic Applications," AIAA Paper 95-0048, Jan. 1995.

[10]Liu, F., and Zheng, X., "A Staggered Finite Volume Scheme for Solving Cascade Flow with a k–ω Turbulence Model," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1589–1597.

[11]Zheng, X., and Liu, F., "Numerical Solution of Navier–Stokes Equations and k–ω Turbulence Model Equations Using a Staggered Upwind Method," *AIAA Journal*, Vol. 33, No. 6, 1995, pp. 991–998.

[12]Liu, F., and Zheng, X., "A Strongly-Coupled Time-Marching Method for Solving the Navier–Stokes and k–ω Turbulence Model Equations with Multigrid," AIAA Paper 94-2389, June 1994.

[13]Wilcox, D. C., "Reassessment of the Scale-Determining Equation for Advanced Turbulence Models," *AIAA Journal*, Vol. 26, No. 11, 1988, pp. 1299–1310.

[14]Liu, F., and Jameson, A., "Multigrid Navier–Stokes Calculations for Three-Dimensional Cascades," AIAA Paper 92-0190, Jan. 1992.

[15]Jameson, A., "Transonic Flow Calculations," Princeton Univ., Dept. of Mechanical and Aerospace Engineering, MAE Rept. 1651, Princeton, NJ, July 1983.

[16]Davis, S. S., "Naca64a010 Oscillatory Pitching," *Compendium of Unsteady Aerodynamics Measurements*, AGARD-R-702, Aug. 1982.

[17]Roache, P. J., "A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, Vol. 116, Sept. 1994, pp. 405–413.