Convergence Acceleration of a Navier–Stokes Solver for Efficient Static Aeroelastic Computations

S. Obayashi and G. P. Guruswamy

Reprinted from

AIAA Journal

Volume 33, Number 6, Pages 1134–1141

AIAA

A publication of the
American Institute of Aeronautics and Astronautics, Inc.
370 L'Enfant Promenade, SW
Washington, DC 20024-2518

AIAA JOURNAL
Vol. 33, No. 6, June 1995

Convergence Acceleration of a Navier–Stokes Solver for Efficient Static Aeroelastic Computations

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

New capabilities have been developed for a Navier–Stokes solver to perform steady-state simulations more efficiently. The flow solver for solving the Navier–Stokes equations is based on a combination of the lower-upper factored symmetric Gauss–Seidel implicit method and the modified Harten–Lax–van Leer–Einfeldt upwind scheme. A numerically stable and efficient pseudo-time-marching method is also developed for computing steady flows over flexible wings. Results are demonstrated for transonic flows over rigid and flexible wings.

Introduction

THIS paper summarizes new developments in the static option of the NASA Ames Research Center's aeroelasticity simulation code ENSAERO, which computes steady-state flowfields and static deformations of structures simultaneously. This code is capable of computing aeroelastic responses by simultaneously integrating the Euler/Navier–Stokes equations and the modal structural equations of motion using aeroelastically adaptive dynamic grids.[1] It was enhanced with the upwind option and applied to transonic flows from small to moderately large angles of attack for fighter wings undergoing unsteady motions.[2,3] Next, it was extended to simulate unsteady flows over a wing with an oscillating trailing-edge flap.[4] The geometric capability of the code has further been extended to handle a full-span wing–body configuration with control surfaces.[5] As an option, the structure can be modeled using shell/plate finite element formulation.[6]

The present research has been performed to improve the static aeroelastic option of the code. The computational fluid dynamics part of the code has been completely rewritten with new algorithms. These new techniques do not change the accuracy of the code, but they make the code more efficient and robust. The lower-upper factored symmetric Gauss–Seidel (LU-SGS) method[7] is employed to reduce the arithmetic operation count in the implicit solver.

The modified Harten–Lax–van Leer–Einfeldt (HLLE) upwind scheme[8] is used to obtain a robust flow solver with reasonable costs. The code originally employed the streamwise upwind algorithm as an upwind option. This algorithm brings multidimensional information to the upwind technique and gives better accuracy than conventional dimensional-split upwind techniques. After the higher order extension is made in a standard manner, however, the improvements in accuracy become small, while the additional computational cost is high. Although it is acceptable for a research code to obtain the best accuracy on a given grid, efficiency is also important for a production code. Therefore, the streamwise upwind algorithm is replaced with the dimensional-split upwind algorithm. In addition, the HLLE upwind scheme is a positively conservative scheme.[9] The present modified HLLE scheme has the same resolution, but it is more robust than the widely used Roe upwind scheme.[10] Its arithmetic operation count still remains comparable to the Roe scheme.

In aeroelastic calculations, structural oscillations are often observed during the initial transient period, which leads to divergence of flow calculations or slow convergence. To reduce such oscillations, the structural dynamics part of the code is also modified to use artificial structural damping along with a pseudo-time-marching method.

To demonstrate the capability of the resulting code, three test cases are shown: computation of a flat-plate boundary layer, transonic flow around the ONERA M6 wing, and transonic flow about an aeroelastic wing.[6,11]

Algorithm Development

ENSAERO computes the three-dimensional Euler or thin-layer Navier–Stokes equations with the Baldwin–Lomax model for the fluid flow part. The description of those equations and the turbulence model can be found in Refs. 1 and 12, for example. The essence of the new algorithms can be described in the one-dimensional equations as shown in the following sections.

Modified HLLE Scheme

The conservation form of the one-dimensional Euler equations is

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

where the conserved quantities Q and flux F are

$$Q = \begin{pmatrix} \rho \\ \rho u \\ e \end{pmatrix}, \qquad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (e + p)u \end{bmatrix} \tag{1b}$$

and where ρ is the density, u is the velocity, and e is the total energy per unit volume. The pressure p is related to the conserved quantities through the equation of state for a perfect gas

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

The cell interface flux F_{LR} can be evaluated by the HLLE scheme[9] as

$$F_{LR} = \tfrac{1}{2}(F_L + F_R - \bar{R}\hat{\Lambda}\bar{L}\Delta Q) \tag{2a}$$

where

$$\Delta Q = Q_R - Q_L \tag{2b}$$

$$\hat{\Lambda} = \frac{b_R^+ + b_L^-}{b_R^+ - b_L^-}\bar{\Lambda} - 2\frac{b_R^+ b_L^-}{b_R^+ - b_L^-}I \tag{2c}$$

$$\bar{\Lambda} = \begin{pmatrix} \bar{u} & & 0 \\ & \bar{u} + \bar{c} & \\ 0 & & \bar{u} - \bar{c} \end{pmatrix} \tag{2d}$$

Presented as Paper 94-2268 at AIAA 25th Fluid Dynamics Conference, Colorado Springs, CO, June 20–23, 1994; received June 27, 1994; revision received Feb. 1, 1995; accepted for publication Feb. 1, 1995.

*Senior Research Scientist, MCAT Institute, San Jose, CA 95127; currently Associate Professor, Tohoku University, Sendai, Japan. Senior Member AIAA.

†Research Scientist, Applied Computational Aerodynamics Branch. Associate Fellow AIAA.

and where $\bar{R}$, $\bar{\Lambda}$, and $\bar{L}$ are the right eigenvector, eigenvalue, and left eigenvector matrices of the Roe-averaged Jacobian,[10] I is the identity matrix, and the subscripts L and R indicate the left and right states. (The overbar indicates Roe-averaged quantities.) The HLLE scheme defines b_R^+ and b_L^- as

$$b_R^+ = \max(\bar{u} + \bar{c}, u_R + c_R, 0) \tag{3a}$$

$$b_L^- = \min(\bar{u} - \bar{c}, u_L - c_L, 0) \tag{3b}$$

This scheme satisfies all of the stability, entropy, and positively conservative conditions required for the nonlinear difference equations.[9]

The HLLE scheme approximates the solution of the Riemann problem with two waves propagating with speed of $b_R = \max(\bar{u} + \bar{c}, u_R + c_R)$ and $b_L = \min(\bar{u} - \bar{c}, u_L - c_L)$ and a state Q_{LR} between those waves. Compared with the Roe scheme, the HLLE scheme introduces large numerical dissipation at contact discontinuities. A newly proposed modification[8] improves the resolution at contact discontinuities by replacing $\hat{\Lambda} = \text{diag}[\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3]$ in Eq. (2a) with

$$\tilde{\Lambda} = \text{diag}[\hat{\lambda}_1 - 2\delta\min(b_R^+, b_L^-), \hat{\lambda}_2, \hat{\lambda}_3] \tag{4a}$$

where

$$\delta = \min\left(\frac{\rho_{LR}}{|\sigma_1|}, \frac{1}{2}\right) \tag{4b}$$

and where $\sigma_1 = \Delta\rho - \Delta p/\bar{c}^2$. Thus $\delta = 1/2$. The resulting scheme reduces to the Roe scheme when $\delta = 1/2(\rho_{LR}/|\sigma_1| \to \infty$ as $|\sigma_1| \to 0)$, and to the scheme when $\delta = 0(\rho_{LR}/|\sigma_1| \to 0$ as $|\sigma_1| \to \infty)$. Because σ_1 represents a jump in entropy, it is zero for isentropic flows. Then the present scheme results in the Roe scheme. As the jump in entropy becomes large, the present scheme turns into the HLLE scheme.

To compute the present modified HLLE flux, Eq. (2), there is no need to do a matrix computation as suggested for the Roe scheme.[13] In the actual implementation, Eq. (2) is rewritten as

$$F_{LR} = \frac{1}{2}\left[f_{1_L}\begin{pmatrix}1\\u\\H\end{pmatrix}_L + f_{1_R}\begin{pmatrix}1\\u\\H\end{pmatrix}_R + \begin{pmatrix}0\\p_L + p_R + \delta_2\\\delta_3\end{pmatrix}\right] \tag{5a}$$

where

$$f_{1_L} = \rho_L(u_L + \hat{\lambda}_1) + \delta_1 \tag{5b}$$

$$f_{1_R} = \rho_R(u_R - \hat{\lambda}_1) + \delta_1 \tag{5c}$$

$$\delta_1 = -\left(\hat{\lambda}^+\frac{\Delta p}{\bar{c}} + \hat{\lambda}^-\bar{\rho}\Delta u\right)\Big/2\bar{c} \tag{5d}$$

$$\delta_2 = -\left(\hat{\lambda}^+\bar{\rho}\Delta u + \hat{\lambda}^-\frac{\Delta p}{\bar{c}}\right) \tag{5e}$$

$$\delta_3 = -\left(\hat{\lambda}_1\Delta p + \bar{u}\delta_2\right) \tag{5f}$$

$$\hat{\lambda}^+ = \frac{\hat{\lambda}_2 + \hat{\lambda}_3}{2} - \hat{\lambda}_1 \tag{5g}$$

$$\hat{\lambda}^- = \frac{\hat{\lambda}_2 - \hat{\lambda}_3}{2} \tag{5h}$$

Again by replacing $\hat{\lambda}_1$ with $\tilde{\lambda}_1$ by using Eq. (4), the present algorithm is obtained. The extension to the three dimensions is straightforward by dimensional splitting.

Modified Differentiable Limiter

Higher order numerical fluxes are obtained from higher order interpolation of the primitive variables for the left and right states at the cell interface.[14] For example, the interpolated pressure are given as

$$p_{Li+\frac{1}{2}} = \left\{1 + \frac{\phi}{4}[(1-\kappa)\nabla + (1+\kappa)\Delta]\right\}p_i \tag{6a}$$

$$p_{Ri+\frac{1}{2}} = \left\{1 - \frac{\phi}{4}[(1+\kappa)\nabla + (1-\kappa)\Delta]\right\}p_{i+1} \tag{6b}$$

where ∇ and Δ are backward and forward difference operators, respectively. For a one-dimensional or Cartesian grid case, third-order interpolation can be obtained from $\kappa = 1/3$. One limiter function ϕ from Koren[15] is given as

$$\phi(p_i) = \frac{3\nabla p_i \cdot \Delta p_i}{2(\Delta p_i - \nabla p_i)^2 + 3\nabla p_i \cdot \Delta p_i} \tag{7}$$

Since this limiter reacts to even small oscillations in smooth regions, convergence to steady state is often stalled. To avoid clipping smooth extrema, a modification to the limiter was proposed in Refs. 16 and 17 as

$$\tilde{\phi}(p_i) = \frac{3\nabla p_i \cdot \Delta p_i + \varepsilon_i^2}{2(\Delta p_i - \nabla p_i)^2 + 3\nabla p_i \cdot \Delta p_i + \varepsilon_i^2} \tag{8}$$

In this paper, the threshold ε_i^2 is given by $\varepsilon_i^2 = \max(3.0|\nabla\xi|_i^{-3}, 10^{-12})$ where ξ denotes the generalized coordinate.

LU-SGS Method

Discretizing Eq. (1), the LU-SGS method[7] can be described as

$$\{I + h[\chi\rho(A)_i I - A_{i-1}^+]\}[I + h\chi\rho(A)I]_i^{-1} \times \{I + h[\chi\rho(A)_i I + A_{i+1}^-]\}\Delta Q = -h\left(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right) \tag{9a}$$

where $h = \Delta t/\Delta x$, $A = \partial F/\partial Q$, $\rho(A)$ denotes the spectral radius of A,

$$A^\pm = \tfrac{1}{2}[A \pm \chi\rho(A)I)] \tag{9b}$$

and $\chi = 1.01$ typically. Its extension to three dimensions is straightforward. Note that this is a two-factored scheme so that the algorithm can be written as

Forward sweep:

$$\Delta Q_i' = \left[-h\left(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right) + hA^+\Delta Q_{i-1}'\right]/[1 + h\chi\rho(A_i)] \tag{10a}$$

Backward sweep:

$$\Delta Q_i = \Delta Q_i' - hA^-\Delta Q_{i+1}/[1 + h\chi\rho(A_i)] \tag{10b}$$

where $1 + h\chi\rho(A_i)$ is only a scalar quantity. Thus, it requires no block-matrix inversion.

To vectorize these sweeps in three dimensions, a hyperplane is necessary. Suppose i, j, and k denote the indices of a grid point, a hyperplane is expressed as $i + j + k = \text{const}$. The three-dimensional array (i, j, k) is then converted to two-dimensional array (l, m) where l denotes the point in the hyperplane and $m = i + j + k$ represents the hyperplane. This enjoys a very long vector length, but the permutation requires additional memory space. Instead, the existing three-dimensional array can be used as it is by vectorizing only in the i index. The outer DO loop is for m. The next loop is for k. The inner DO loop is for i, and j is determined as $j = m - k - i$. Its vector length is shorter than the former, but it requires no additional memory space for vectorization.

The split Jacobian matrices $A^\pm$ mimic the upwind Jacobians. For a natural upwinding, $A^+ = A$ and $A^- = 0$ when $u > c$. However, the definition of $A^\pm$ in Eq. (9b) gives $A^- \neq 0$. A simple modification to the split Jacobian was tried as

$$A^\pm = \frac{s^\pm}{2}[A \pm \rho(A)I] \pm \frac{(\chi - 1)}{2}\rho(A)I \tag{11a}$$

where

$$s^\pm = \begin{cases} 1, & \text{if } \pm(u \pm c) \geq 0 \\ 0, & \text{if } \pm(u \pm c) < 0 \end{cases} \tag{11b}$$

In the authors' experience, it gives slightly better convergence than the unmodified case. For the flat-plate boundary-layer case, however, it did not make the convergence any better or worse.

For the thin-layer Navier–Stokes computations, the split Jacobians in the viscous direction are further modified as

$$A^{\pm} = \frac{s^{\pm}}{2}[A \pm \rho(A)I] \pm \frac{(\chi - 1)}{2}\rho(A)I \pm \frac{\mu + \mu_T}{Re \cdot \rho}|\nabla\xi|^2 I \quad (12)$$

where μ is the laminar viscosity and μ_T is the turbulent viscosity.

In addition, Eq. (9b) introduces large dissipation through the time integration. Only when the solution converges to a steady state, its effect vanishes. However, the LU-SGS method which is first-order accurate in time similar to other implicit methods, can be used for unsteady calculations by using special methods. For example, the Newton iteration[18,19] can be used for unsteady computations, which not only removes the excess dissipation but also gives the second-order accuracy in time.

In the following calculations, a locally varying time step was taken as

$$\Delta t_{i,j,k} = \Delta t_{\text{global}} \frac{1 + 0.0005\sqrt{J_{i,j,k}}}{1 + \sqrt{J_{i,j,k}}} \quad (13)$$

where J is the Jacobian of transformation and $\Delta t_{\text{global}} = 1$–5, depending on the problem. The Jacobian scaling in the denominator has been used commonly (for example, see Ref. 20). The additional scaling in the numerator has been also used widely without documentation. It prevents the time step from becoming too small as the Jacobian becomes very large. With the LU-SGS scheme, this Jacobian scaling for the time stepping might not be necessary since the scheme is unconditionally stable. However, it was used to make fair comparisons with the previous version of the code.

Aeroelastic Equation of Motion

The governing aeroelastic equations of motion of a flexible wing are solved using the Rayleigh–Ritz method. It is assumed that the deformed shape of the wing can be represented by a set of discrete displacements at selected nodes. From the modal analysis, the displacement vector $\{d\}$ can be expressed as

$$\{d\} = [\phi]\{q\} \quad (14)$$

where $[\phi]$ is the modal matrix and $\{q\}$ is the generalized displacement vector. The final matrix form of the aeroelastic equations of motion is

$$[M]\{\ddot{q}\} + [D]\{\dot{q}\} + [K]\{q\} = \{Z\} \quad (15)$$

where $[M]$, $[D]$, and $[K]$ are modal mass, damping, and stiffness matrices, respectively. The term $\{Z\}$ is the aerodynamic force vector defined as $\rho U_\infty^2[\phi]^T[L]\{\Delta C_p\}/2$, and $[L]$ is the diagonal area matrix of the aerodynamic control points. The aeroelastic equation of motion, Eq. (15), is solved by a numerical integration technique based on the linear acceleration method.[21]

Since $\{\ddot{q}\} = \{\dot{q}\} = 0$ at a steady state, a primary static option uses

$$[K]\{q\} = \{Z\} \quad (16)$$

However, this often causes significant numerical oscillations because there is no damping in the system. These oscillations are initiated by numerical transients. To obtain a smooth numerical transition, a pseudo-time-marching method can be taken by using Eq. (15) where the time step is set as $\Delta t_{\text{structure}} = 0.02\Delta t_{\text{global}}$ typically. The damping coefficients $[D]$ are set so that all aeroelastic modes damp out quickly. The damping matrix is computed using the relation

$$[D] = \alpha[M] + \beta[K] \quad (16a)$$

where α and β are functions viscous damping coefficients of each mode. More details about the computation of $[D]$ can be found in Chap. 9 of Ref. 22. In this work, viscous damping coefficient values for all modes are set to a high value of 0.80. This large value for the damping coefficient gives an artificial "shock absorber" for structural oscillations caused by flow transients.

Results

Flat-Plate Boundary Layer

First, performances of the new algorithms are examined. Figure 1 shows a comparison of computed u-velocity profiles of a laminar layer on a flat plate. The freestream Mach number is 2, the Reynolds number is 1×10^6, and the freestream temperature is 222 K. The Sutherland law was used to compute the laminar viscosity. This computation was performed on a $111 \times 5 \times 69$ grid by enforcing two dimensionality through five planes. The grid was generated algebraically. The time step Δt_{global} used was 2.6 for all cases.

The upwind (modified HLLE) solution coincides with the similarity solution perfectly, whereas the central difference (CD) solution gives slight smearing at the outer edge of the boundary layer. The CD option uses the combination of the second- and fourth-order dissipation with the pressure switch.[20] The smoothing coefficients were set to 0.25 and 0.005 for the second- and fourth-order dissipation terms, respectively. The upwind algorithm performs better as expected. Since obtaining grid convergence is not practical for most cases, the upwind option is generally recommended. Figure 2 shows a comparison of the convergence history (in terms of the averaged residual for all five equations multiplied by the Jacobian) between unmodified and modified limiters. The modified limiter, Eq. (8), shows a clear improvement over the original limiter.

Figure 3 shows a comparison of convergence history among four different schemes. The original ENSAERO has the CD option with the diagonal approximate factorized alternating direction implicit (ADI) solver[20] and the upwind option using the streamwise upwind algorithm with the LU-ADI solver.[2] The new version of the code employs the LU-SGS solver for both CD and upwind options. The computational time per grid point per iteration in microseconds was 4.0 for the CD option with the LU-SGS solver, 6.0 for the CD option with the diagonal ADI solver, 9.5 for the modified HLLE upwind option with the LU-SGS solver, and 13.8 for the streamwise upwind option with the LU-ADI solver, on a single Cray C90 processor. It should be noted that there is a disadvantage in the vectorization for the ADI solver in this comparison because the grid has only five planes in one direction.

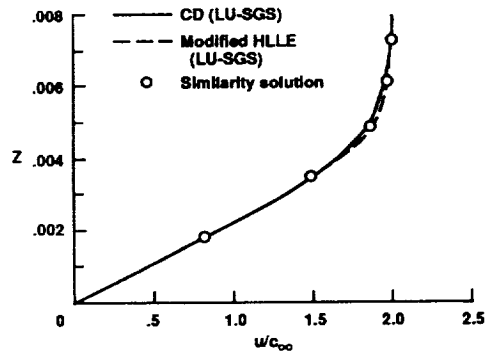

Fig. 1 Comparison of computed laminar boundary-layer profiles on a flat plate between the central difference and the upwind algorithms, $M_\infty = 2.0$ and $Re = 10^6$.

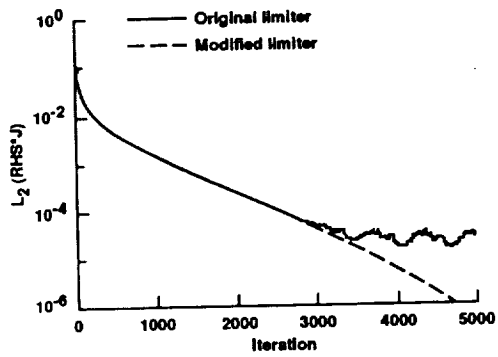

Fig. 2 Comparison of convergence history between the original and modified differentiable limiters, $M_\infty = 2.0$ and $Re = 10^6$.

The LU-SGS solver shows better convergence for both CD and upwind options than the ADI solvers. The advantage of using the LU-SGS solver is more apparent when the residual is replotted against CPU time in seconds using a single Cray C90 processor as shown in Fig. 4. Because of fewer operations and better convergence rates, the LU-SGS solver is about four times faster than the diagonal ADI solver with the CD option. Even the new upwind option using the modified HLLE scheme is faster than the original CD option with the diagonal ADI solver. The original upwind option is obviously the slowest because of the multidimensional upwinding. To obtain further acceleration for the LU-SGS solver, one might consider implementing the multigrid technique.[23]

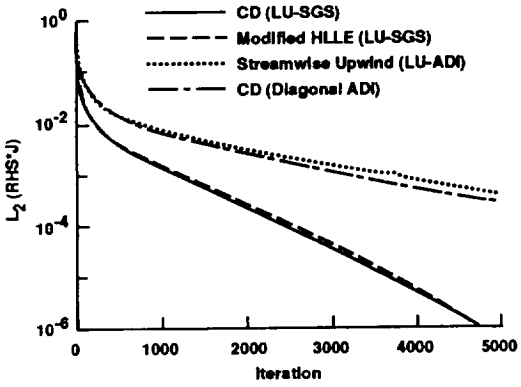

Fig. 3 Comparison of convergence history among four methods using the present and the previous version of the code, $M_\infty = 2.0$ and $Re = 10^6$.

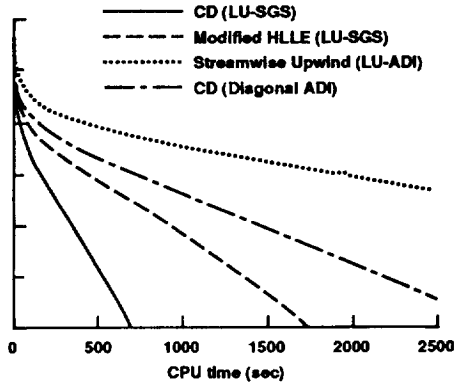

Fig. 4 Comparison of convergence history in CPU seconds among four methods using the present and the previous versions of the code, $M_\infty = 2.0$ and $Re = 10^6$.

ONERA M6 Wing

To further validate the flow solver, transonic flows over the ONERA M6 wing have been computed by using the new upwind option of the code. A series of experimental data can be found in Ref. 24. Following a recent report of computational results,[25] the two cases chosen were one for Mach number 0.8395 and angle of attack of 3.06 deg and the other for Mach number of 0.8447 and angle of attack of 5.06 deg. The Reynolds number was set to 11×10^6 for both cases. A C-O grid of size $193 \times 34 \times 49$ was used. This is the coarser grid used in Ref. 25. The nondimensional time step was set to 2.0.

High angle of attack cases with a large flow separation present a severe test for turbulence models. Even the Baldwin–Lomax model was found to be sensitive for a particular function evaluation.[26] The model requires the definition of $\bar{u}_{\max}$, $(\bar{u}^2_{\mathrm{DIF}} = \bar{u}^2_{\max} - \bar{u}^2_{\min})$, which is supposed to be $\bar{u}_{\max} \equiv \bar{u}|_{F_{\max}}$, not the maximum of $\bar{u}$ along the grid line normal to the wall. For the flat-plate boundary layer with a uniform outer flow, for example, it does not matter how $\bar{u}_{\max}$ is defined. However, $\bar{u}_{\max}$ should be defined at the $F_{\max}$ location.

The 5.06-deg angle of attack case of ONERA M6 wing was sensitive to this definition of $\bar{u}_{\max}$. The computation didn't converge by using the true definition of $\bar{u}_{\max}$, whereas the computation did converge by using the nominal $\bar{u}_{\max}$, the maximum of $\bar{u}$ along the grid line normal to the wing surface. Since turbulent viscosity is increased by overestimating $\bar{u}_{\max}$, more numerical viscosity is introduced into the computation. It probably helped stabilize the computation. In the following calculations, the nominal $\bar{u}_{\max}$ is used in the Baldwin–Lomax model.

Figure 5 shows the surface pressure distributions compared with the experiment at the 44, 65, 80, and 90% spanwise sections for the 3.06-deg case. The computational results agree reasonably well with the experiment except on the upper surface of the 80% spanwise section where a finer grid may be required to resolve the merger of

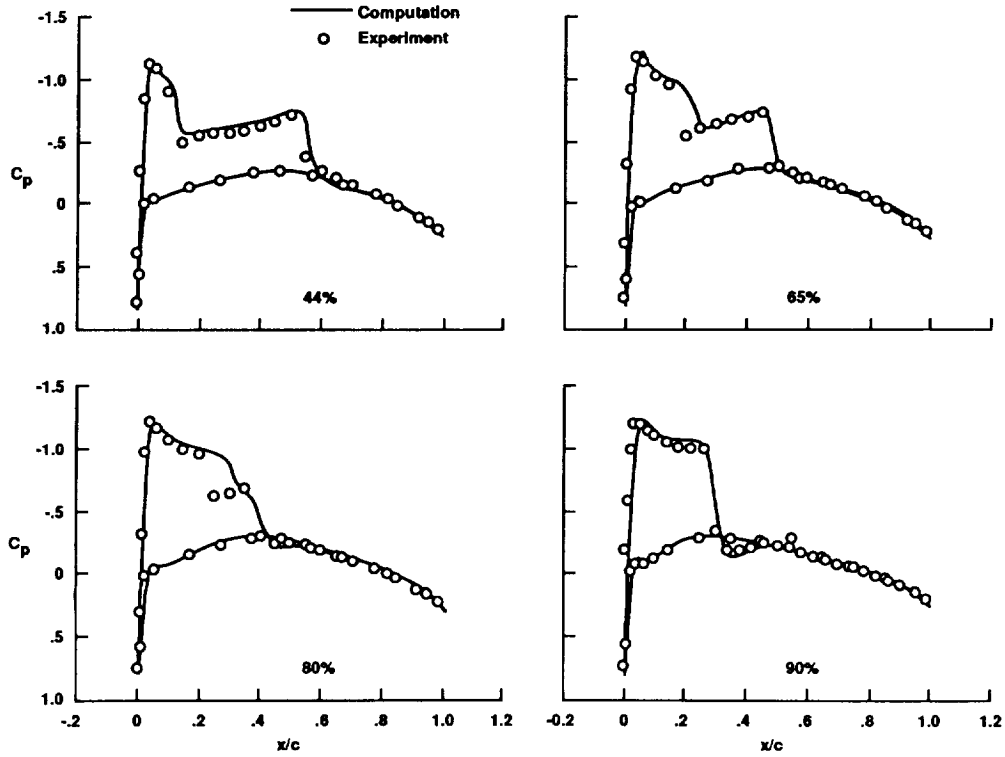

Fig. 5 Comparison of computed pressure distributions on the ONERA M6 wing with experiment, $M_\infty = 0.84$, $Re_c = 11 \times 10^6$, and $\alpha = 3$ deg.

two shock waves. Figure 6 shows the comparison of the convergence history between the new upwind option (modified HLLE + LU-SGS) and the previous CD option (diagonal ADI) in terms of both the residual and lift coefficient. The new upwind option shows a better convergence history. It took roughly 4500 iterations to converge to three digits of accuracy in the lift coefficient for the new upwind option. The CD computation was performed with the time step 1.0, since it diverged with the time step 2.0. The smoothing coefficients in this case were also set to 0.25 and 0.01 for the second- and fourth-order dissipation terms, respectively, since the computation diverged with the fourth-order dissipation coefficient 0.005. The CD option required roughly 8700 iterations to converge to three digits in the lift coefficients.

The computational time per grid point per iteration using the upwind option was 7.7 μs on a C90 single processor. This number is better than the flat-plate case (9.5 μs) because the grid is truly three dimensional. On the other hand, the diagonal ADI CD option requires 5.9 μs. To obtain the three-digit convergence in the lift coefficient, the new upwind option required two-thirds of the total computational time of the previous CD option. Although the time per grid point per iteration is reasonably fast for both cases, the convergence is still slow because of the more complicated flow-field in this case than that in the flat-plate case. To improve the convergence rate further, the multigrid technique may be useful as demonstrated on the same grid in Ref. 25.

Figure 7 shows the corresponding surface pressure distributions for the 5.06-deg case. The Baldwin–Lomax model predicts a stronger shock wave and less flow separation, similar to the results in Ref. 25.

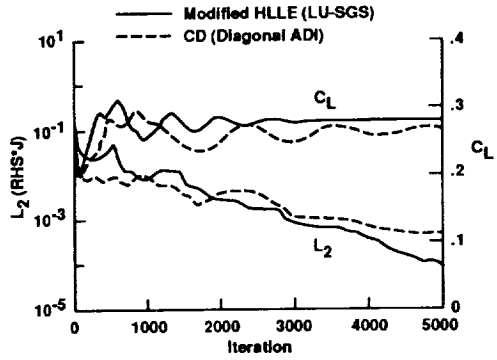

Fig. 6 Comparison of convergence history between the new upwind option and the previous CD option of the code, ONERA M6 wing, $M_\infty = 0.84$, $Re_c = 11 \times 10^6$, and $\alpha = 3$ deg.

Aeroelastic Wing

To demonstrate static aeroelastic computations, a swept wing of aspect ratio 3 and taper ratio 1/7 with the NACA 65A006 airfoil section was selected. The planform of the wing is shown in Fig. 8. Experimental and numerical studies can be found in Ref. 11 and 6, respectively. A C-H grid of size 151 × 44 × 44 was used. The time step was set to 6.25 for all of the steady cases. Only the upwind (the modified HLLE and LU-SGS option) results are shown here. The computational time per grid point per iteration was 8.0 μs for the fluid part and 0.7 μs for the structure part, including computations for grid movement.

In the following calculations, the freestream Mach number was set to 0.854, the Reynolds number was 0.597×10^6, and the ratio of specific heats was 1.135 because the experimental fluid was Freon. The Baldwin–Lomax model was used to compute turbulent viscosity. The dynamic pressure was set to 0.7 psi. The natural vibration modes of the wing were calculated by the finite element plate model.[6] The computed frequencies for first three modes are 21.8, 78.1, and 126 Hz, and the corresponding measured values are 21.6, 79.7, and 121 Hz. Following Ref. 22, the values obtained for α and β are equal to 137.5 and 2.912×10^{-3}, respectively.

Figure 9 shows responses of the leading-edge displacement at the wing tip. The displacement is plotted in inches, and 5000 iterations correspond to about 0.05 s in the unsteady computations. Small generalized accelerations were given as initial conditions for the structure. Starting from the steady-state flow solution at 1-deg angle of attack, three aeroelastic computations were performed. Two of them are steady-state calculations using the locally varying time stepping in the fluid part. One of them uses the pseudo-time-marching method for the structure using Eq. (15) (denoted as "pseudodynamic"), and the other uses the static equation, Eq. (16) (denoted as "static"). The third case is an unsteady computation using the time-accurate

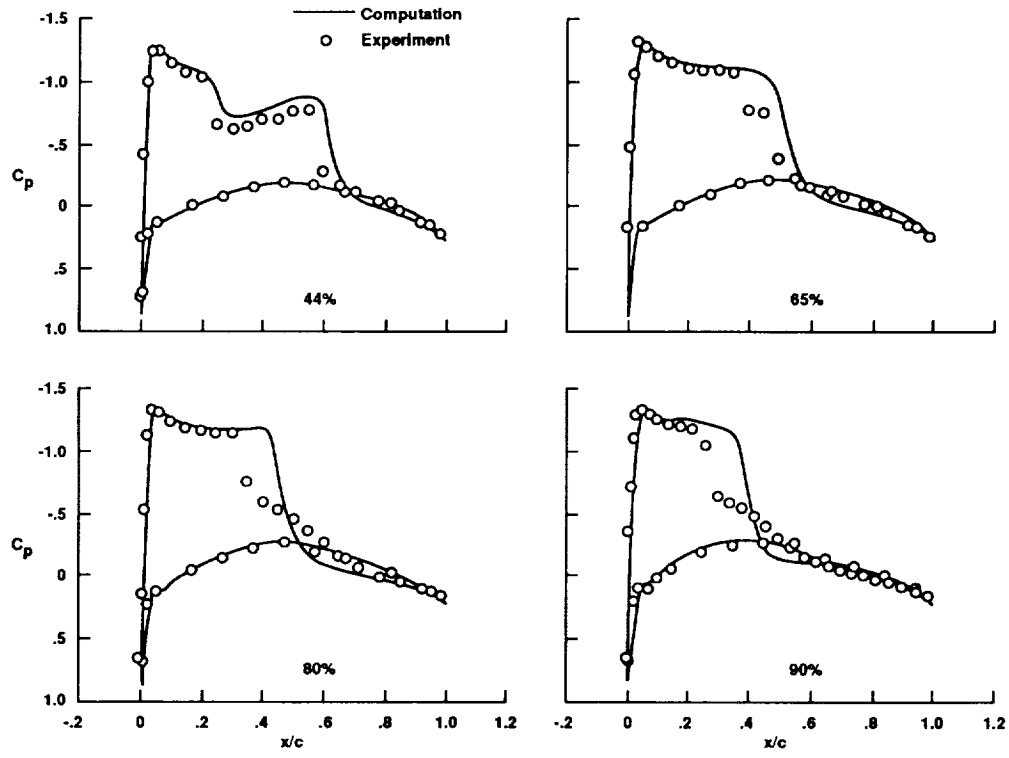

Fig. 7 Comparison of computed pressure distributions on the ONERA M6 wing with experiment, $M_\infty = 0.84$, $Re_c = 11 \times 10^6$, and $\alpha = 5$ deg.

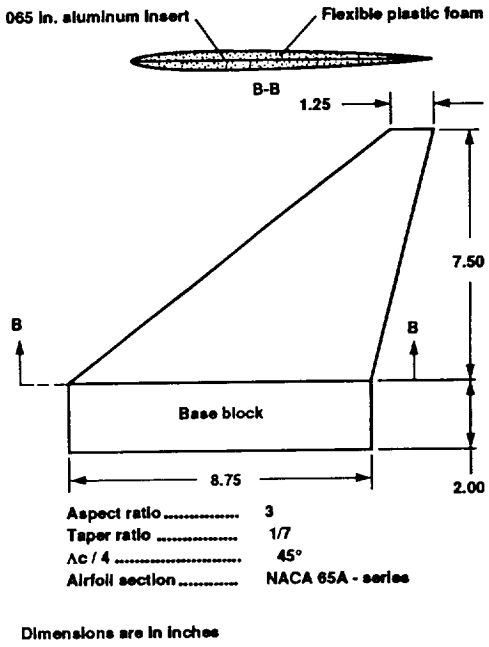

Fig. 8 Planform of an experimental swept wing.

option of the code where the time step is constant for both fluid and structure part. It uses high damping coefficients[6] in Eq. (15) (denoted as "dynamic with damping").

The two steady-state computations converged within 1000 iterations. The unsteady result shows that the solution approaches the same steady state slowly. The unsteady computation with the damping was continued up to 10,000 iterations and confirmed to converge to a displacement of 0.279 in. However, this option is too slow to be used for a production code. Figure 10 shows the magnified view of Fig. 9 for the first 1000 iterations. Both of the steady-state calculations go through the initial transient in the first 200 iterations and converge to 0.279 (three digits) in about 1000 iterations. The main difference of the two is the behavior during the initial transient. The pseudodynamic case shows very smooth transient, whereas the static case shows significant oscillations. Although it damps out quickly in this case, such oscillations often cause numerical instability in the fluid part. Thus, the pseudo-time-marching method is very favorable for static aeroelastic computations.

Since the pseudo-time-marching method for the structure gives a smooth transient, and computations can be started from the freestream condition impulsively. Note that the previous static aeroelastic calculations required 1000 iterations in addition to computing the initial steady-state flow solution without the structural dynamics. Figure 11 shows responses of the leading-edge displacement at the wing tip for this impulsive start case. The initial transient of the 1-deg angle of attack case was damped out in 2000 iterations. It converged to 0.279 (three digits) in about 3400 iterations. This converged value coincides with that obtained in Fig. 10. The freestream condition was also set to 3-deg angle of attack. The convergence was slower than that for the 1-deg case, but it reached steady state roughly at 3500 iterations.

Figure 12 shows the corresponding responses of the lift coefficients. The lift coefficients behave similarly to the leading-edge displacements shown in Fig. 11. The rigid wing cases without any structural dynamics are included for the comparison. The comparison between the flexible and rigid cases indicates that there is no penalty for having structural dynamics on convergence. In fact, the convergence history is almost identical for aeroelastic and rigid cases to a certain degree as shown in Fig. 13. Since the aeroelastic option does not slow down the convergence, it is more efficient and much simpler to perform static aeroelastic simulations directly starting from freestream conditions.

The aeroelastic case seems to fall into the limit cycle earlier than the rigid case, although the 3-deg cases go into the limit cycle before they depart from each other. For the fluid part, the threshold in Eq. (8)

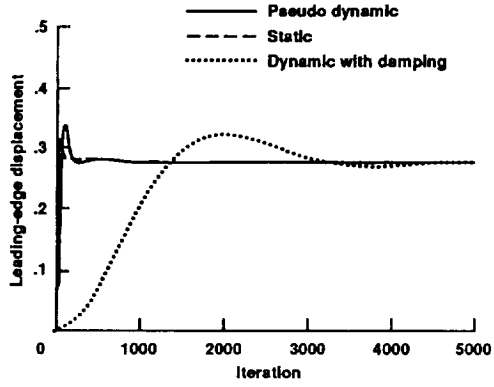

Fig. 9 Comparison of computed responses of leading-edge displacements at wing tip with steady and unsteady options of the code, M_∞ = 0.85, $Re_c = 0.6 \times 10^6$, and α = 1 deg.

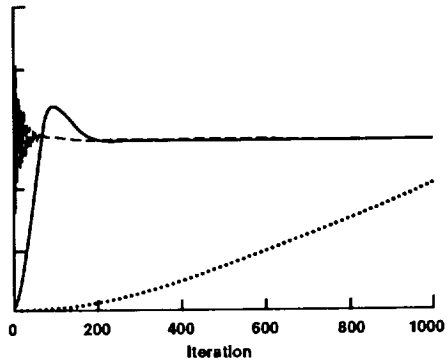

Fig. 10 Close-up view of computed responses of leading-edge displacements at wing tip.

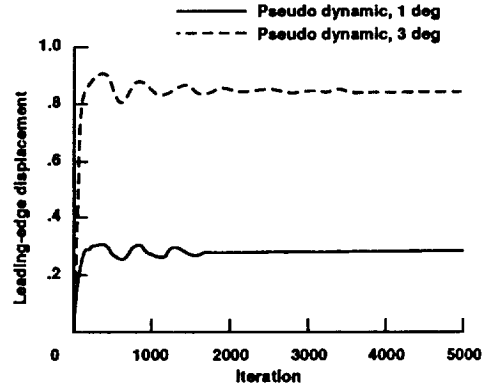

Fig. 11 Computed responses of leading-edge displacements at wing tip with the pseudodynamic options of the code, $M_\infty = 0.85$, $Re_c = 0.6 \times 10^6$, and α = 1 and 3 deg.

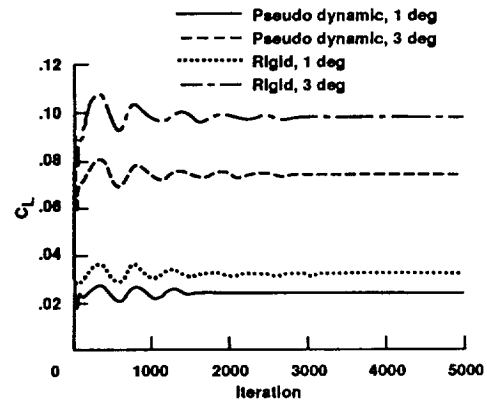

Fig. 12 Comparison of computed lift responses for flexible and rigid wing cases, M_∞ = 0.85, $Re_c = 0.6 \times 10^6$, and α = 1 and 3 deg.

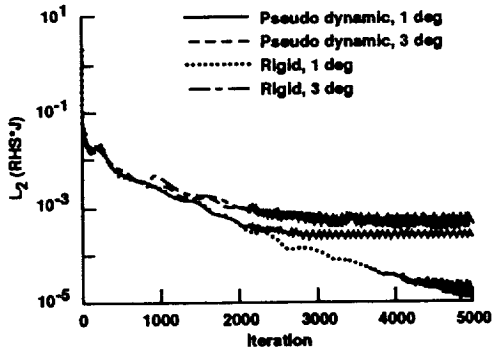

Fig. 13 Comparison of convergence history for flexible and rigid wing cases, $M_\infty = 0.85$, $Re_c = 0.6 \times 10^6$, and $\alpha = 1$ and 3 deg.

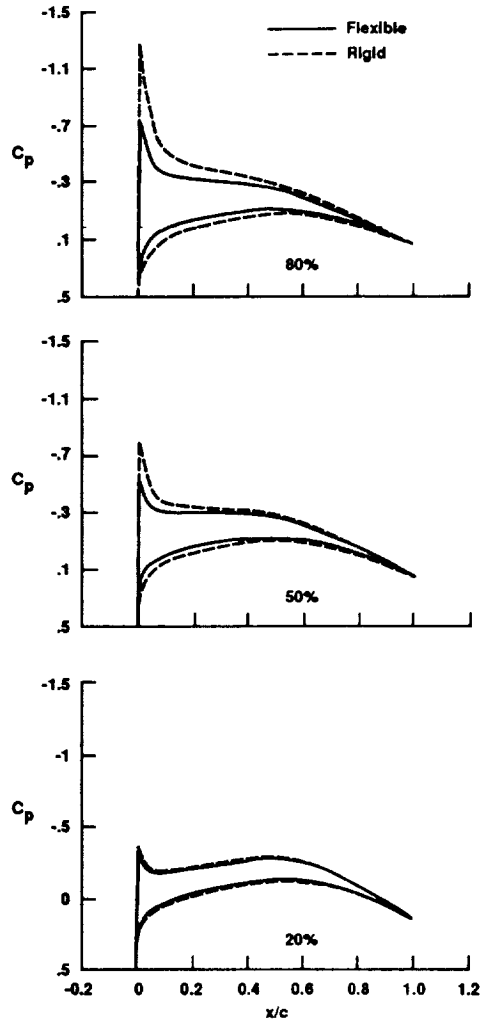

Fig. 14 Comparison of computed surface pressures between flexible and rigid wing cases, $M_\infty = 0.85$, $Re_c = 0.6 \times 10^6$, and $\alpha = 3$ deg.

can be tuned case by case to obtain better convergence. However, it is simply more practical to accept such limit cycles in convergence history for general applications. In such cases, one may choose either lift coefficient or leading-edge displacement as an index for convergence.

In Fig. 12, the flexible cases consistently show lower lift at both 1 and 3-deg angles of attack. Figure 14 shows comparison of surface pressure coefficients between the flexible and rigid wing cases at 3-deg angle of attack. The plots confirm the lower lift for the flexible wing. Figure 15 illustrates the corresponding tip displacement. The wing tip is bent up, and its leading edge is twisted down. The aeroelastic deformation reduces the effective angle of attack locally

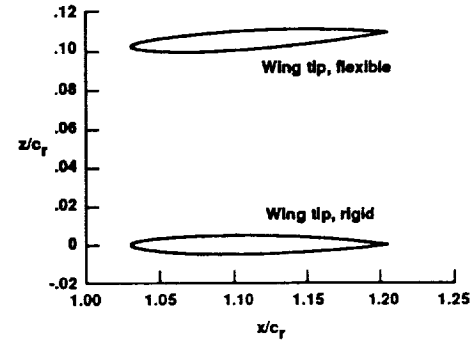

Fig. 15 Comparison of wing tip locations between flexible and rigid wing cases, $M_\infty = 0.85$, $Re_c = 0.6 \times 10^6$, and $\alpha = 3$ deg (the vertical scale is exaggerated).

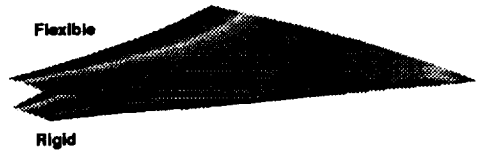

Fig. 16 Comparison of wing shapes between flexible and rigid wing cases, $M_\infty = 0.85$, $Re_c = 0.6 \times 10^6$, and $\alpha = 3$ deg.

near the tip, which results in the lower lift. Figure 16 shows the overall view of the wing deformation.

Conclusion

New capabilities to compute static aeroelasticity more efficiently have been added to ENSAERO code. The test case of computing a boundary layer on a flat plate demonstrates that the combination of the LU-SGS implicit method and the modified HLLE upwind scheme gives a good compromise of accuracy and efficiency for solving the Navier–Stokes equations. This option computes a steady-state solution faster than the original central difference option of the code. The capability of this new flow solver has been demonstrated successfully for computing transonic flows over wings.

The existing procedure for static aeroelastic option requires a good initial guess, specifically a converged steady-state flow solution. In this study, it is found that aeroelastic computations can be started from freestream conditions with the use of a pseudo-time-marching method for the structural equations of motion. The test case of computing transonic flows over a flexible wing indicates that static aeroelastic simulations can be performed at computational effort similar to the cost of single-discipline steady-state flow simulations.

Acknowledgments

The first author's work was supported by NASA Grant NCC 2-605. The computer time was partly provided by the Numerical Aerodynamic Simulator program located at the NASA Ames Research Center.

References

[1]Guruswamy, G. P., "ENSAERO—A Multidisciplinary Program for Fluid/Structural Interaction Studies of Aerospace Vehicles," *Computing System Engineering*, Vol. 1, Nos. 2–4, 1990, pp 237–256.

[2]Obayashi, S., Guruswamy, G. P., and Goorjian, P. M., "Streamwise Upwind Algorithm for Computing Unsteady Transonic Flows Past Oscillating Wings," *AIAA Journal*, Vol. 29, No. 10, 1991, pp. 1668–1677 (see also "Errata," *AIAA Journal*, Vol. 30, No. 2, 1992, p. 569).

[3]Obayashi, S., and Guruswamy, G. P., "Unsteady Shock-Vortex Interaction on a Flexible Delta Wing," *Journal of Aircraft*, Vol. 29, No. 5, 1992, pp. 790–798.

[4]Obayashi, S., and Guruswamy, G. P., "Navier–Stokes Computations for Oscillating Control Surfaces," *Journal of Aircraft*, Vol. 31, No. 3, 1994, pp. 631–636.

[5]Obayashi, S., Chiu, I.-T., and Guruswamy, G. P., "Navier–Stokes Computations on Full-Span Wing-Body Configuration with Oscillating Control Surfaces," AIAA Paper 93-3687, Aug. 1993.

[6]Guruswamy, G. P., and Byun, C., "Fluid-Structural Interactions Using Navier–Stokes Flow Equations Coupled with Shell Finite Element Structures," AIAA Paper 93-3087, July 1993.

[7]Yoon, S., and Jameson, A., "Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier–Stokes Equations," *AIAA Journal*, Vol. 26, No. 9, 1988, pp. 1025–1026.

[8]Obayashi, S., and Wada, Y., "Practical Formulation of a Positively Conservative Scheme," *AIAA Journal*, Vol. 31, No. 5, 1994, pp. 1093–1095.

[9]Einfeldt, B., Munz, C. D., Roe, P. L., and Sjögreen, B., "On Godunov-Type Methods near Low Densities," *Journal of Computational Physics*, Vol. 92, 1991, pp. 273–295.

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

[11]Doggett, R. V., Jr., Rainey, A. G., and Morgan, H. G., "An Experimental Investigation of Aerodynamic Effects of Airfoil Thickness on Transonic Flutter Characteristics," NASA TM X-79, Nov. 1959.

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

[13]Liu, Y., and Vinokur, M., "Upwind Algorithms for General Thermo-Chemical Nonequilibrium flows," AIAA Paper 89-0201, Jan. 1989.

[14]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.

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

[16]Venkatakrishnan, V., "Preconditioned Conjugate Gradient Methods for the Compressible Navier–Stokes Equations," *AIAA Journal*, Vol. 29, No. 7, 1991, pp. 1092–1100.

[17]Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA Paper 93-0880, Jan. 1993.

[18]Chen, C. L., McCroskey, W. J., and Obayashi, S., "Numerical Solutions of Forward-Flight Rotor Flow Using an Upwind Method," *Journal of Aircraft*, Vol. 28, No. 6, 1991, pp. 374–380.

[19]Pulliam, T. H., "Time Accuracy and the Use of Implicit Methods," AIAA Paper 93-3360, July 1993.

[20]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.

[21]Guruswamy, P., and Yang, T. Y., "Aeroelastic Time Response Analysis of Thin Airfoils by Transonic Code LTRANS2," *Computers and Fluids*, Vol. 9, No. 4, 1980, pp. 409–425.

[22]Bathe, K. J., *Finite Element Procedures in Engineering Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1982, Chap. 9, pp. 528–530.

[23]Yoon, S., and Kwak, D., "Multigrid Convergence of an Implicit Symmetric Relaxation Scheme," AIAA Paper 93-3357, July 1993.

[24]Schmitt, V., and Charpin, F., "Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers," AGARD-AR-138, B1, May 1979.

[25]Rumsey, C. L., and Vatsa, V. N., "A Comparison of the Predictive Capabilities of Several Turbulence Models Using Upwind and Central-Difference Computer Code," AIAA Paper 93-0192, Jan. 1993.

[26]Flores, J., private communication, NASA Ames Research Center.