JOURNAL OF AIRCRAFT
Vol. 32, No. 6, November–December 1995

Conservative Multiblock Navier–Stokes Solver for Arbitrarily Deforming Geometries

Jaakko Hoffren,* Timo Siikonen,† and Seppo Laine‡
Helsinki University of Technology, 02150 Espoo, Finland

A new finite volume based thin-layer Navier–Stokes solver for time-dependent compressible flows is described. The implicit, temporally and spatially second-order accurate formulation facilitates the simulation of flows at high Reynolds numbers. Complex, arbitrarily deforming geometries can be handled using fully conservative multiblock structured grids. An iterative time-stepping utilizing a multigrid technique maintains the temporal and spatial accuracy of the basic scheme also at grid block interfaces and solid boundaries. Several test cases have been computed to verify the proper function of the developed code, and the simulations of a transonic buffet and an oscillating supercritical airfoil are presented here. The solver employing the three-level fully implicit time discretization scheme is found to work well in dynamic grids within the limitations of the original formulation. The new code has the desired capabilities, although there is room for improvement in computational efficiency.

Nomenclature

C_p = pressure coefficient
$C_{p,\mathrm{Im}}$ = imaginary part of scaled unsteady pressure coefficient, $2\int_0^T \cos(\omega t)C_p(t)\,dt/(\alpha_0 T)$
$C_{p,\mathrm{Re}}$ = real part of scaled unsteady pressure coefficient, $2\int_0^T \sin(\omega t)C_p(t)\,dt/(\alpha_0 T)$
c = airfoil chord
c_d, c_l = drag and lift coefficients
e = total energy per unit volume
$\hat{F}$ = flux vector in a curvilinear grid
k = reduced frequency, $\omega c/(2U_\infty)$
M = Mach number
n_x, n_y = Cartesian unit normal components of a cell face
p = pressure
R = residual
Re = Reynolds number
S = cell face area
T = cycle period
t = time
U = vector of conservative variables
U_∞ = freestream velocity
$\tilde{u}$ = convective velocity component
u, v = Cartesian velocity components
u_g, v_g = Cartesian velocity components of the grid
u_n = normal velocity component of the flow
V = cell volume
v_n = normal velocity component of the grid
x, y = spatial coordinates
α = instantaneous angle of attack, $\bar{\alpha} + \alpha_0 \sin(\omega t)$
$\bar{\alpha}$ = average angle of attack
α_0 = angle-of-attack oscillation amplitude
β = parameter defining the implicitness of the temporal discretization
γ = parameter controlling the time levels in the temporal discretization
Δt = physical time step, nondimensionalized by the freestream sonic speed and the airfoil chord
$\Delta\mathcal{V}$ = volume swept by a cell face during a time step
$\Delta\tau$ = local pseudo-time-step
δU = change in conservative variables in an iteration cycle or a pseudo-time-step
ρ = density
ω = angular velocity

Subscripts
i, j = spatial indices
k = summing index
mod = modified expression
v = viscous
1, 2 = cell face endpoints

Superscripts
k = intermediate solution between n and $n+1$
n = time level with the latest known solution
$n+1$ = new time level with the unknown solution
$n-1$ = time level one step before n

Received April 27, 1994; revision received Feb. 23, 1995; accepted for publication May 16, 1995.
*Research Scientist, Laboratory of Aerodynamics.
†Senior Research Scientist, Laboratory of Aerodynamics.
‡Professor in Aerodynamics. Member AIAA.

Introduction

TIME-ACCURATE flow simulations using the Euler or the Navier–Stokes equations have been conducted as long as the basic numerical methods and sufficiently powerful computers have been available. Fairly sophisticated cases had been studied already during the early development stage of the numerical flow solvers. Emery[1] published transient Euler calculations in a shock channel. Oscillating geometries were involved in the simulation of a pitching airfoil by Magnus and Yoshihara[2] and in the aileron buzz study of Steger and Bailey.[3] Other classic examples are the calculations of transonic shock-boundary-layer interactions on a biconvex airfoil by Levy[4] and Steger.[5]

Owing to their relative simplicity, explicit time-integration methods have been popular from the early days of numerical flow simulations. The Lax–Wendroff scheme was used in Ref. 2 and the explicit MacCormack scheme was applied in Ref. 4. The latter scheme was employed also in the inlet calculations of Newsome[6] and in the study of an axisymmetric combustor by Scott and Hankey.[7] The Adams–Bashfort scheme[8,9] has gained some favor too, but different versions of the Runge–Kutta scheme have become the most popular explicit time-integration methods. Runge–Kutta schemes for oscillating inviscid flows have been applied, e.g., in the airfoil calculations of Smith et al.[10] and Jameson and Venkatakrishnan,[11,12] and in the study of counter-rotating cascades by Engel et al.[13]

The Runge–Kutta methods are second-order accurate in time, easy to code, and straightforward to vectorize, but they suffer from the serious time-step stability limitations, as do all of the explicit schemes. This deficiency makes them poorly suited to simulations of viscous flow at high Reynolds numbers requiring very dense grids. The time-step limitation can be avoided by using implicit schemes that have been applied parallel to the explicit schemes as long as the numerical flow simulation has been feasible.

By far, the most popular implicit time-integration method has been the first-order accurate backward-Euler scheme, usually combined with an approximate factorization to facilitate the solution at each time step. This method was applied in Refs. 3 and 5, and it has been utilized in numerous other viscous and inviscid simulations of oscillating airfoils and wings[14–24] or other types of unsteady flow cases.[25–30] The method is robust, relatively efficient, and appears to be accurate enough for slowly varying situations.

Second-order accurate implicit schemes have found some applications. The three-level fully implicit scheme has been utilized in the simulation of dynamic stall by Visbal[31] and Shida et al.[32] The same method has been applied by Simpson[33] and Belk,[34] who report an improvement in accuracy compared with the first-order scheme. Thomadakis and Tsangaris[35] have come to similar conclusions in their studies of an oscillating airfoil. Some attempts with using the trapezoidal rule or the Crank–Nicolson (C–N) scheme have also been published,[3,36] but there have been stability problems.

The approximate factorization frequently applied with the implicit formulations produces additional errors into the solution. A proper implicit treatment of boundary conditions causes difficulties, too. Especially problematic are the block interfaces in multiblock grids. To circumvent these difficulties and to allow for enlarged physical time steps while retaining the accuracy of the basic discretization, iterative solution methods may be employed. Several different iteration schemes have been studied. Simpson[33] and Steithorsson et al.[37] have coupled a Newton-type iteration with a pseudotime integration within physical time steps. A similar method has been used by Rogers and Kwak[38] for solving incompressible flows applying the pseudocompressibility concept during iterations. Ridder et al.,[36] Brenneis and Eberle,[39] Batina,[40] and Chen et al.[41] have implemented variants of Gauss–Seidel iterations. Relaxation schemes for incompressible flows have been pursued by Hegna[42,43] and Tuncer et al.[44] Recently, the generalized minimal residual (GMRES) method studied, e.g., by Sankar and Hixon[45] has been arousing interest.

The longer time steps of the iterative methods combined with a higher-order basic temporal discretization offer a possibility to improve the efficiency of the computations compared with low-order factored schemes. If an iteration could be made to converge faster than it takes to perform several short steps of the direct solution, savings would be achieved. However, few studies addressing the convergence acceleration within physical time steps have been published. The multigrid technique, highly successful in steady-state calculations, has been tested,[46–48] but the results are inconclusive.

Unsteady flows often involve moving or deforming bodies. A variety of methods has been devised to take the body dynamics into account. Magnus and Yoshihara[1] as well as Ruo and Sankar[18] have circumvented the problem of an unsteady grid by specifying time-dependent transpiration velocities to simulate airfoil oscillations. Rigid airfoil oscillations have been studied with a rigidly moving grid[49] and with a deforming grid.[50] A rigidly moving grid does not cause any practical computational problems, but grid deformations may be difficult to calculate efficiently. Different interpolation or stretching methods may be used, or the grid can be regenerated at each time step. The grids used in the time-dependent simulations are generally structured, although Batina and coworkers[40,51–56] have pioneered the use of unstructured grids with dynamic bodies. An advantage of this approach is that complex geometries and large deformations are easier to handle than with structured grids. On the other hand, the concept is poorly suited to high-Reynolds-number viscous flows, which limits its applicability.

The conservation form of the flow equations has been found preferable in numerical simulations. However, the geometrical conservation law has often been neglected. This relationship connects the grid cell face velocities and volume changes in such a way that the grid deformations do not bring any additional numerical errors into the solution. The early calculations were invariably performed without concern about the geometrical conservation law, but later its importance has been addressed in a few theoretical papers[57–59] and documents including test calculations.[35,51,52] However, Guruswamy[60] and Obayashi et al.[61] claim that in practical calculations with small deformations, it is not necessary to respect the geometrical conservation in order to get accurate results. Although the issue is somewhat controversial, it can be assumed that the geometrical conservation law is an important factor in cases involving large deformations.

Based on the survey of the available literature, it appears that most of the published time-accurate flow solvers are somewhat restricted in applicability, or their efficiency has not been addressed. The time integration itself may be approximate and inaccurate, the geometry to be studied may have to be simple, or the treatment of the grid deformations is not well defined. Computer power has apparently limited the development work. However, recent advances in computer technology have alleviated the problem significantly, and at present it seems reasonable to direct the effort towards the development of more general and accurate simulation methods for time-dependent flows.

In this article, a new time-accurate flow simulation method based on the iterative solution of the thin-layer Navier–Stokes equations is studied. The aim of the work is to produce a general, accurate, and efficient time-integration code for the time-dependent flow equations in arbitrarily deforming, complex computing domains. The flow solver developed from a steady-state code[62] is described in the following chapter, and some test cases used to verify the correct operation of the system are presented subsequently. Also, a search for an optimal code operation mode is discussed. At the end of this article, the results of the work are evaluated and some conclusions are drawn.

Description of the Numerical Method

Finite Volume Formulation in a Deforming Grid

In the finite volume method applied, the conservation laws for mass, momentum, and energy in a compressible flow are written directly for each deformable, discrete computational cell as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t} VU + \sum_{k=1}^{m} \boldsymbol{F}_k(U_k)\cdot\boldsymbol{S}_k = 0 \tag{1}$$

Here, $\boldsymbol{S}_k$ are the vectorial cell surface segment areas. In two dimensions, U and $\boldsymbol{F}(U)\cdot\boldsymbol{S} = \hat{\boldsymbol{F}}S$ for each cell face can be written as

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \qquad \hat{\boldsymbol{F}}S = \begin{pmatrix} \rho\bar{u} \\ \rho u\bar{u} + n_x p \\ \rho v\bar{u} + n_y p \\ e\bar{u} + u_n p \end{pmatrix} S + \hat{\boldsymbol{F}}_v S \tag{2}$$

Here, u and v are defined in a fixed frame of reference. For brevity, the viscous fluxes represented by $\hat{\boldsymbol{F}}_v$ are not written explicitly here, since their basic forms are not changed by the grid deformation. In the given fluxes, n_x and n_y are the unit normal components of S, p is the pressure obtained from the

equation of state for a perfect gas, u_n the velocity component normal to the cell face, and $\bar{u} = u_n - v_n$, the relative flow velocity through the face. The only additional term due to grid dynamics as compared with a fixed geometry is the grid normal velocity v_n.

For the following discussion, a shortened form

$$\frac{\mathrm{d}}{\mathrm{d}t} VU = R \tag{3}$$

for the flow equations is adopted, applying the abbreviation $R = -\sum_{k=1}^{m} \boldsymbol{F}_k(U_k) \cdot \boldsymbol{S}_k$.

Temporal Discretization

In order to be able to simulate high-Reynolds-number viscous flows with reasonable calculation times, an implicit formulation is necessary. Second-order temporal accuracy is also desirable to achieve an efficient scheme suitable for an iterative time-stepping method. Since different discretizations may be optimal for different types of problems, the following family of two- or three-level schemes is utilized in the present work:

$$(1 + \gamma)(VU)^{n+1} - (1 + 2\gamma)(VU)^n + \gamma(VU)^{n-1} = \Delta t[(1 - \beta)R^n + \beta R^{n+1}] \tag{4}$$

Here, the superscripts relate to the time levels involved. The parameter γ controls the levels to be employed and β defines the extent of the implicitness. For example, the combination $\gamma = 0.5, \beta = 1.0$ specifies the three-level fully implicit scheme (3-LFI). The drawback of this formulation is the relatively large computer memory requirement, since the solution at three time levels and R at two time levels must be storable.

Treatment of Grid Deformations

To avoid any unnecessary errors introduced by the grid deformations, the exact fulfillment of the conservation of geometry is enforced regardless of the discretization applied. A suitable mathematical formulation for the geometrical conservation law is obtained from Eq. (4) by requiring exact mass conservation in a freestream. By taking into account the specified constant density and flow velocity an equation connecting the cell volume changes and the cell face velocities is obtained. From this basic formulation, a useful condition applicable to each cell face separately can be deduced by understanding the volume changes to mean the volumes $\Delta\mathcal{V}$ swept by the face within the time steps involved. With this reasoning, the geometrical conservation law in its final form is written as

$$(1 + \gamma)\Delta\mathcal{V}^{n+1} - \gamma\Delta\mathcal{V}^n = \Delta t[(1 - \beta)v_n^n S^n + \beta v_n^{n+1} S^{n+1}] \tag{5}$$

A different notation $\mathcal{V}$ for the swept volumes is used here to distinguish them from V.

In addition to the cell face normal velocity, the cell face tangential velocity component is needed for the calculation of viscous fluxes at the solid walls. The necessary extra condition follows from the direction in which the cell face is moving. Applying this idea, a unique grid velocity vector fulfilling the geometrical conservation law can be computed.

In the practical implementation of the code, Cartesian velocity components are used. The geometrical quantities involved in the definition of the velocities for a cell face at the time level $n + 1$ are illustrated in Fig. 1. After the generation of the grid node points for the new time level, the only unknowns are the cell face velocities u_g^{n+1} and v_g^{n+1}. From the geometrical conservation law and the specification of the face

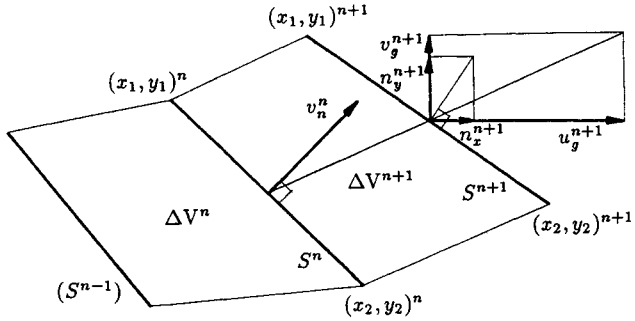

Fig. 1 Grid properties involved in the calculation of cell face velocity components.

movement direction, the following pair of equations is obtained:

$$u_g^{n+1} n_x^{n+1} + v_g^{n+1} n_y^{n+1} = \frac{1}{\beta S^{n+1}} \times \left[\frac{(1 + \gamma)\Delta\mathcal{V}^{n+1} - \gamma\Delta\mathcal{V}^n}{\Delta t} - (1 - \beta)v_n^n S^n \right] \tag{6a}$$

$$\frac{u_g^{n+1}}{v_g^{n+1}} = \frac{x_1^{n+1} + x_2^{n+1} - x_1^n - x_2^n}{y_1^{n+1} + y_2^{n+1} - y_1^n - y_2^n} \tag{6b}$$

The final values for the new cell face velocities are solved from these equations. For explicit calculations, Eqs. (6) are replaced by simpler conditions for the grid velocities at time level n.

Computation of Fluxes

For the calculation of the inviscid fluxes, Roe's method is used. The fluxes are computed in a locally one-dimensional manner applying rotation matrices and the scheme is adapted to moving grids by including the grid speeds in the convective velocities.[63] To exclude nonphysical expansion shocks allowed by the basic method, the entropy fix of Yee[64] ensuring nonzero values for the relative acoustic wave speeds is implemented. In the evaluation of the required variables at cell faces in the structured grid used, MUSCL-type differencing with an optional flux limiter is applied. The discretization is thus formally second-order accurate in smooth regions of the flowfield.

For the viscous and heat fluxes, the thin-layer approximation with conventional central differences is utilized. The effects of turbulence are taken into account by algebraic turbulence models that can be activated in predetermined flow regions to simulate partially laminar flow. At present, the Baldwin–Lomax model and the Cebeci–Smith model adapted to Navier–Stokes calculations by Stock and Haase[65] are available.

To be able to study complex flow regions, multiblock grids are used. The blocks must be nonoverlapping with continuous cell distributions across the interfaces to facilitate a complete continuity and conservation of the solution. However, the blocks to be joined can be of different size to ease the grid generation and to enhance the solution efficiency. For the calculation of the fluxes at common boundaries of the blocks, states in the two cell rows adjacent to the interface are transferred by utilizing two layers of ghost cells outside each block overlapping the neighboring block.

For external flow calculations, freestream conditions are specified at the grid outer boundary. At solid viscous walls the flow velocities are set equal to the grid velocity and a flow tangency condition is applied at inviscid walls and symmetry planes. The implementation of the solid wall conditions relies on a second-order extrapolation of pressure and viscosity.

Time-Stepping Method

With the implicit temporal discretizations, an iterative solution facilitates the elimination of factorization errors and eases the treatment of boundary conditions. However, the most important factor necessitating the iterations is the multiblock structure of the grid, since the correct, conservative implicit data transfer between the blocks in a noniterative system is practically impossible to achieve.

The iteration scheme utilized is derived from Eq. (4) by linearizing it at an intermediate state k. When the new solution is written as $(VU)^{n+1} = V^{n+1}U^k + V^{n+1}\delta U$ and R^{n+1} is linearized, too, an equation for the corrections δU is obtained. When everything except the terms containing δU are written on the right-hand side (RHS) and a division by V^{n+1} is conducted, the equation applicable to each cell takes the form:

$$\left[1 + \gamma - \beta \frac{\Delta t}{V^{n+1}} \left(\frac{\partial R}{\partial U}\right)^k\right] \delta U = \frac{1}{V^{n+1}} \times \{-(1+\gamma)V^{n+1}U^k + (1+2\gamma)(VU)^n - \gamma(VU)^{n-1} + \Delta t[(1-\beta)R^n + \beta R^k]\} \tag{7}$$

From this equation U^{n+1} can be iterated by computing successive corrections for U^k applying a matrix solver required by the left-hand side (LHS) until the process is sufficiently converged. However, because of the approximations made in the adopted solver that will be described shortly, Eq. (7) must be modified to ensure stability of the iteration at arbitrary physical time steps. This can be accomplished without affecting the solution accuracy, because at the converged situation δU is zero. The stabilization is conducted by replacing γ on the diagonal of the LHS with a variable weight $\Delta t/\Delta\tau$, where $\Delta\tau$ is a stability-limited local pseudo-time-step related to the solver employed.

After the stabilizing modification and a subsequent multiplication by $\Delta\tau/(\Delta t + \Delta\tau)$, Eq. (7) can be converted to

$$\left[1 - \beta \frac{\Delta\tau_{\text{mod}}}{V^{n+1}} \left(\frac{\partial R}{\partial U}\right)^k\right] \delta U = \frac{\Delta\tau_{\text{mod}}}{V^{n+1}} R^k_{\text{mod}} \tag{8}$$

where $\Delta\tau_{\text{mod}} = \Delta t\Delta\tau/(\Delta t + \Delta\tau)$ and the modified residual R^k_{mod} is the braced expression of the RHS of Eq. (7) divided by Δt.

This final equation is of exactly the same form as that used in typical implicit steady-state flow solvers employing pseudotime integration. In this work, the LU-factorization scheme of Ref. 62, following the ideas of Obayashi and Kuwahara,[66] is used to obtain the corrections δU. The scheme is based on the approximate factorization and on the splitting of the Jacobians $(\partial R/\partial U)^k$. The tridiagonal equations resulting from the factorization are further factored into bidiagonal sweeps that can be vectorized for efficient computations. The starting values of the bidiagonal sweeps at the boundaries of the computational domain are set to zero for simplicity. In addition to the factorizations, a further approximation is that the split Jacobians used in the bidiagonal sweeps are not the true Jacobians of the fluxes used on the RHS, but simpler expressions based on the splitting of Steger and Warming.[67] These factors necessitate the limitation of the local pseudo-time steps $\Delta\tau$ as discussed previously. The values to be used are based on a user-defined Courant number and a diffusion stability criterion. By selecting an appropriate Courant number the stability and efficiency of the iteration can be controlled.

To speed up the convergence within physical time steps, a multigrid algorithm based on the work of Jameson and Yoon[68] can be utilized. The actual implementation is described in detail in Refs. 69–71. In our steady-state applications, this system enhances the convergence rate significantly, the attainable speed-up factors being 5–10.

During the solution process, the blocks are handled sequentially and independently of each other within each pseudotime-step (iteration cycle), keeping all of the boundary conditions fixed. For each block, the spatial discretizations and the number of multigrid levels to be used are defined independently. After all of the blocks have been computed, the block boundaries are updated by setting the new solution at the interfaces into the ghost cells of the neighboring blocks to form new boundary conditions for the next cycle. Although the grid block boundaries lag the solution elsewhere, the final result is completely continuous when the calculation is sufficiently converged. As a result, the grid block interfaces and boundary conditions cause no deterioration in the basic spatial or temporal accuracy.

Three different iteration termination criteria have been used. The first alternative compares the computed maximum change in density to a user-defined limit. The second system computes the density correction L_2 norm at each iteration and divides it by the value obtained at the first iteration in the time step involved. This ratio is the monitored quantity. Because the test runs have shown that it is very difficult to specify suitable values for these criteria ensuring sufficient convergence without unnecessary calculations, a predetermined number of iterations within time steps is specified in the following tests.

Test Calculations

Initial Studies

To validate the basic operation of the developed code, relatively simple, well-defined test cases were computed. Firstly, the inviscid, supersonic, transient shock-channel case defined in Ref. 1 was studied using a two-block grid having a total of 8191 cells. Two noteworthy features were observed in this case. The entropy fix of the Roe's scheme turned out to be necessary and a simple explicit Euler time integration was computationally about as efficient as implicit second-order methods. As the second test case, laminar flow around a rigid NACA 0012 airfoil at $Re = 1 \times 10^4$, $M = 0.85$, and $\alpha = 0$ was simulated in a 192×64 c-grid applying just the implicit schemes. In this GAMM workshop test case,[72] the shocks on the airfoil remained stationary, but periodic oscillations developed in the wake. The behavior obtained is in agreement with the available reference results.[72] The initial calculations are documented in detail in Refs. 48 and 63.

Transonic Buffet on a Biconvex Airfoil

A test case with turbulent flow at a high Reynolds number is necessary to validate the computation method under study for practical applications. A suitable situation is the transonic buffet on an 18% thick biconvex airfoil studied experimentally at $Re = 11 \times 10^6$ in Ref. 73. In this case, a strong oscillatory interaction takes place between the boundary layer and shock waves in a certain Mach number range. The experiments and some reference calculations[4,74] include the effects of the wind-tunnel walls, but calculations have also been made simulating the airfoil in a freestream[5] with comparable results. In this work, it was decided to follow the latter setup because of the simplicity of the grid generation.

The C-type grid employed has 192×64 cells. At the leading edge, the thickness of the first cell layer is 5×10^{-6}, increasing to 20×10^{-6} at the trailing edge, c being of unit length. As the spatial discretization, a third-order upwind-biased scheme without the flux limiter was applied. The transition was fixed at the 5% chord point owing to the lack of any specific data.

Initially, steady flows at subcritical Mach numbers were simulated to check that the solver does not give erratically time-dependent solutions. When the Mach number was increased from 0.754 to 0.783, a periodically oscillating flow with a roughly constant amplitude was obtained, as in Ref. 5. The computed lift and drag coefficients vs time are shown in Fig. 2, and the flowfield is illustrated in Fig. 3. Two different

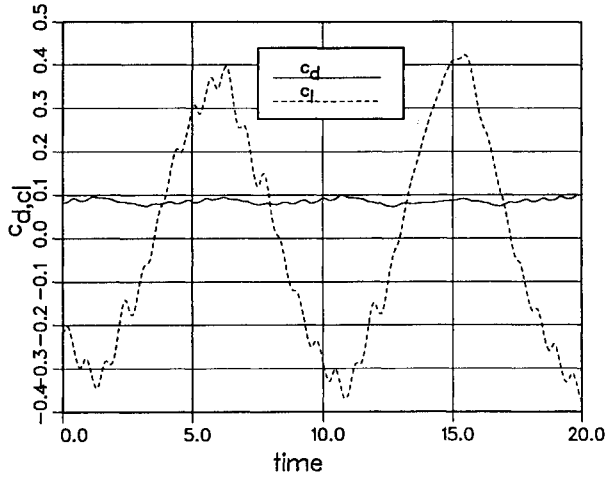

Fig. 2 Computed lift and drag coefficient histories during the fully developed oscillations at $M = 0.783$.

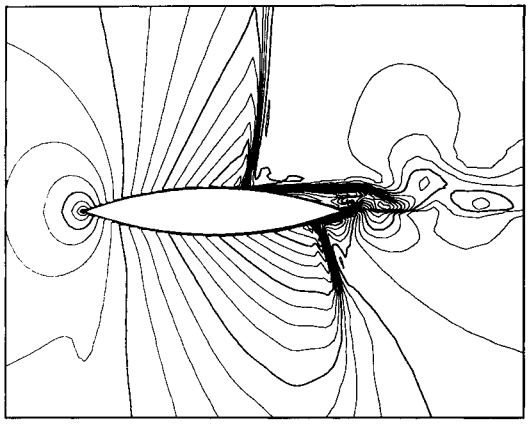

Fig. 3 Mach number distributions around the 18% thick biconvex airfoil at $M = 0.783$ at a lift minimum.

modes of oscillation can be distinguished. The long-period regular oscillation is related to the movement of the shocks, and the high-frequency, small-amplitude irregular oscillation is caused by the wake vortices. The reduced frequency of the shock oscillation is $k = \omega c/(2U_\infty) = 0.43$, which is somewhat lower than the value of 0.49 measured in the tunnel. Nevertheless, the present result is closer to the experimental value than the computed values of 0.40–0.41 given in Refs. 4, 5, and 74. It is interesting to note that none of the references include the wake vortex oscillation mode. However, it can be expected that the present calculations on a relatively dense grid model finer details than the coarse-grid, temporally first-order calculations of the references.

In this context, the efficiency of the solver was studied, and an efficient operating mode was sought by computing a certain period of time with more than 50 different combinations of computational parameters. In the tests, the parameters were the time discretization, the length of time steps, the number of iterations within time steps, the number of multigrid levels, and the iteration Courant numbers. The results obtained were evaluated against an accurate reference solution computed with the following attributes: 3-LFI, $\Delta t = 0.005$ (nondimensionalized by the freestream sonic speed and the airfoil chord), 20 iterations per time step, 2 multigrid levels, Courant–Friedrichs–Lewy (CFL) = 4 on the fine and CFL = 6 on the coarse grid level.

Only implicit time discretizations were tested since in this type of flow the explicit schemes are impractical. The three-level fully implicit scheme produced accurate results without computational problems and is not critical about the time-step lengths as long as they are reasonable. The Crank–Nicolson calculations turned out to be unstable, as could be expected on the basis of Refs. 3 and 36. The scheme is apparently inapplicable to strongly nonlinear problems. The implicit Euler scheme (IE) performed smoothly, but the results were inaccurate unless very short time steps were applied. The method is undoubtedly more inefficient than the 3-LFI scheme. The computed lift histories shown in Fig. 4 were obtained with a constant time step and the computational effort within iterations was the same in each run.

The effects of the time-step lengths and the number of iterations within time steps studied with the 3-LFI scheme turned out to be strongly interrelated. At a certain total effort, equivalent results could be obtained with short time steps and few iterations, or with long time steps and a large number of iterations. This observation sets pressures to enhance the convergence of the iterations. The iteration is necessary in any case, as was proved in additional test runs with very short time steps and no iterations giving poor results.

It was hoped that the multigrid system would speed up the convergence to such an extent that the iterative solution would become highly efficient. However, in the multigrid efficiency tests, the employment of additional grid levels after the third level had no effect on convergence, and the difference between two and three levels was minimal. This behavior differs markedly from the steady-state calculation mode where five grid levels normally lead to a very rapid convergence.[69–71] In

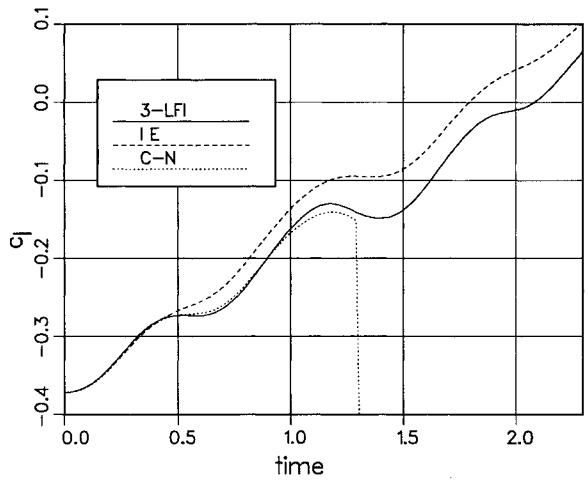

Fig. 4 Behavior of lift of the biconvex airfoil at $M = 0.783$ during the test period computed with different time discretizations.

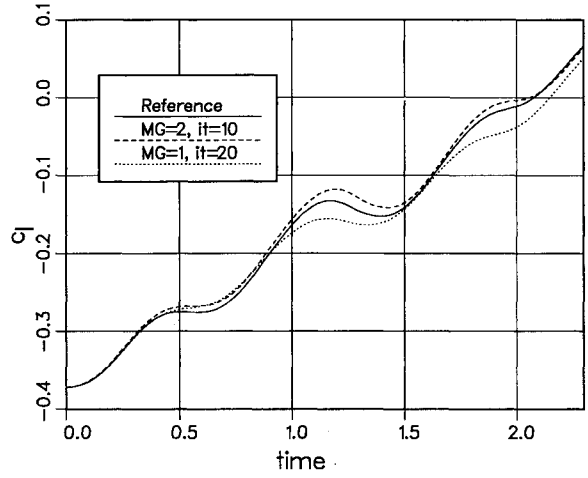

Fig. 5 Behavior of the lift of the biconvex airfoil at $M = 0.783$ during the test period computed with constant time steps and variable numbers of iterations and multigrid levels.

the time-accurate mode no more than two grid levels should be used because the additional levels just consume extra computing effort to no effect. Figure 5 presents results computed with one and two grid levels in such a way that the CPU time consumption of the two-level calculation is about 15% lower, although the result is better. It is seen that the multigrid algorithm with two grid levels improves efficiency, although the benefit is modest. The reason for the relatively poor improvement in convergence when applying the multigrid scheme is apparently the fact that in time-accurate calculations no disturbance waves are to be spread rapidly through the computing domain within time steps.

Finally, the effect of the Courant numbers in the iterations (pseudo-time-steps) was studied. A 10-fold increase of the Courant numbers from the basic values produced an equivalent result at half the computational effort, which means that the effect is substantial.

A code-operating mode devised according to the guidelines obtained from the tests gives practically as good results as the reference system, but consumes only 25% of the reference CPU time. However, even the enhanced simulation is still a heavy process, requiring about 12,500 CPU seconds per one long cycle on an IBM 3090VF-200J.

Pitching Supercritical Airfoil

To validate the code for moving grids, oscillating airfoils were studied. The first case was the NACA 0012 at $M = 0.755$ pitching around its quarter-chord point, as defined by a standard AGARD test case.[75] The angle of attack varied as $\alpha = (0.016 + 2.51)$ deg $\sin(\omega t)$ with $k = 0.0814$. The calculations proceeded smoothly and the results[49,63] agree well with the numerous reference calculations,[14,21,33–35,40,46,50,51,53,76,77] confirming the correct basic operation of the grid movement algorithm. However, the case is apparently relatively easy to handle, requiring just about 100 time steps including five iterations to compute one oscillation cycle accurately.

To test the capabilities of the code in a more challenging situation, the supercritical NLR 7301 airfoil was studied near its shock-free design point corresponding to the AGARD test case CT 8.[75] In this case, the airfoil pitches sinusoidally around the 40% chord point at the reduced frequency of 0.2 with the amplitude of 0.5 deg. The experiments conducted at NASA Ames,[75] where the nominal conditions are $M = 0.751$, $Re = 11.4 \times 10^6$, $\bar{\alpha} = 0.37$ deg, and $k = 0.201$, were simulated in this study.

Initially, the calculations were performed employing a rigidly moving 192 × 64 C-grid similar to the one used with the biconvex airfoil. The grid points were moved analytically, but the grid speeds were evaluated using the general numerical method for deforming grids. The third-order upwind-biased spatial discretization and the three-level fully implicit time integration with five iterations and two multigrid levels were applied in all of the unsteady runs, and the effects of turbulence were simulated using the Baldwin–Lomax model. A problematic issue is the transition that had not been fixed in the experiments. However, since the transition must be specified for the solver, the calculations were initiated with the transition fixed at the 5% chord.

The agreement between the computed and measured steady pressure distributions in the nominal conditions turned out to be rather poor. The computed shock position was clearly too far aft, and the pressure levels in front of the shock disagreed, too. Nevertheless, the airfoil pitching motion was simulated to see whether the unsteady results would be any better. Not surprisingly, the results computed using the time step of about 1/200 of the oscillation cycle were bad. Calculations were repeated using shorter time steps to assess the temporal accuracy, and the criticality of the transition was studied by moving it to the 20% chord. The results of these computations are shown in Fig. 6 depicting the distributions of the real and

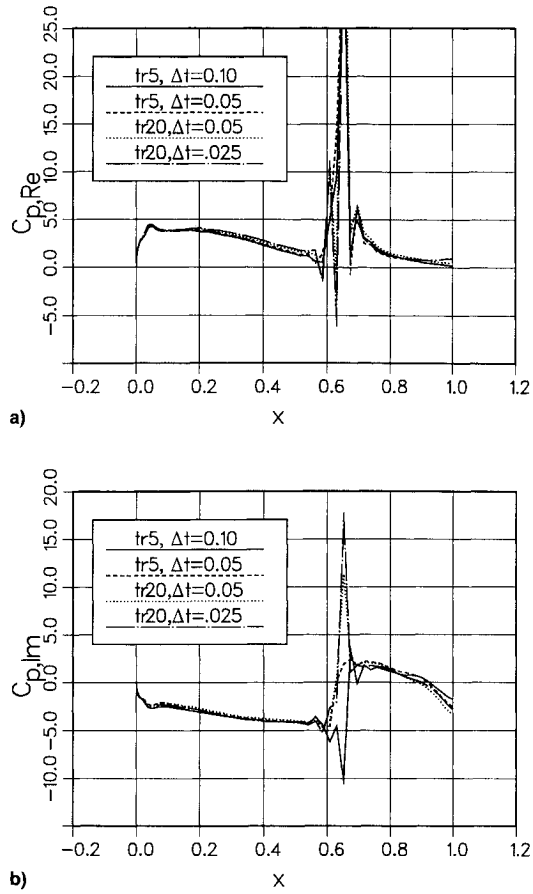

Fig. 6 Distributions of the reduced a) real and b) imaginary parts of C_p on the airfoil upper surface at nominal conditions $M = 0.751$ and $\bar{\alpha} = 0.37$ deg computed with different transition points (tr) and Δt.

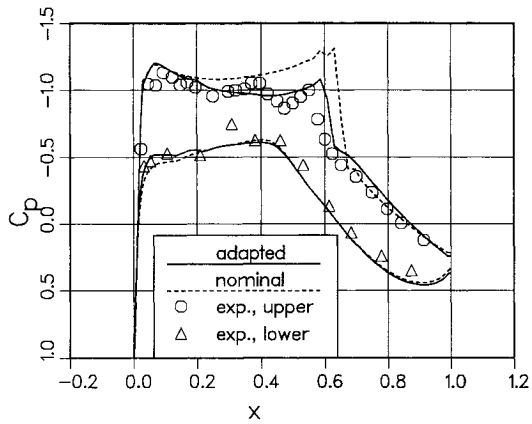

Fig. 7 Steady pressure coefficient distributions computed at the adapted conditions ($M = 0.735$, $\bar{\alpha} = -0.15$ deg) and at the nominal conditions ($M = 0.751$, $\bar{\alpha} = 0.37$ deg) compared with the experimental data (nominal conditions).

imaginary part of C_p on the upper surface divided by the oscillation amplitude in radians.

From Fig. 6a it is seen that moving the transition rearwards induces a downward spike in the real part of C_p at the shock, but the effect of the time step is minimal. In the imaginary part of Fig. 6b, the transition movement brings about an upward spike at the shock. The longest, initially applied time step of 0.1 is apparently too long, causing spurious oscillations. Although the location of the transition and the length of the time step have some effect on the results, it is evident that the solution is mainly governed by other factors.

The calculations were continued by attempts to match the computed steady pressure distribution to the measured one by perturbing the Mach number and the angle of attack while maintaining the transition at the 20% chord. When the combination $M = 0.735$ and $\alpha = -0.15$ deg was specified, a reasonable agreement was obtained. The result is shown in Fig. 7, which also contains the distribution computed at the nominal conditions and the experimental data.[75] The need for adaptation is probably largely caused by the unknown wind-tunnel interferences, which demonstrates the sensitivity and difficulties in transonic experiments. The irregularities in the measured points are related to the imperfections in the manufacture of the model,[75] leading to inevitable deviations from the results computed with the smooth design contour.

The subsequent unsteady calculations at the adapted conditions applying the time step 0.05 led to a significant improvement in the real and imaginary parts of C_p as illustrated in Fig. 8. The agreement of the real part with the measurements is quite good except for the downward spike just after the shock, whereas the result computed at the nominal conditions is far from the experimental data. The differences in the solutions are even larger in the imaginary part where marked differences between the adapted calculation and the experiments still exist before the shock. There are probably two main reasons for this discrepancy. Firstly, the thickness of the experimental model is slightly greater than the design value,[75] which may lead to an increased sensitivity to flow unsteadiness. The second factor is the transition that is fixed in the calculations but free in the experiments. In addition, the primitive turbulence model may have a contribution.

The calculations at the adapted conditions were repeated using a double-density grid to check the effect of spatial resolution. The 384 × 128 grid was generated by dividing each cell of the coarse grid into four smaller cells. The steady pressure distribution obtained using the dense grid agrees with the coarse grid solution with negligible differences. Only the shock is slightly sharper. Also, the unsteady results compared in Fig. 9 are very similar and the reduction of the time step does not change the situation appreciably.

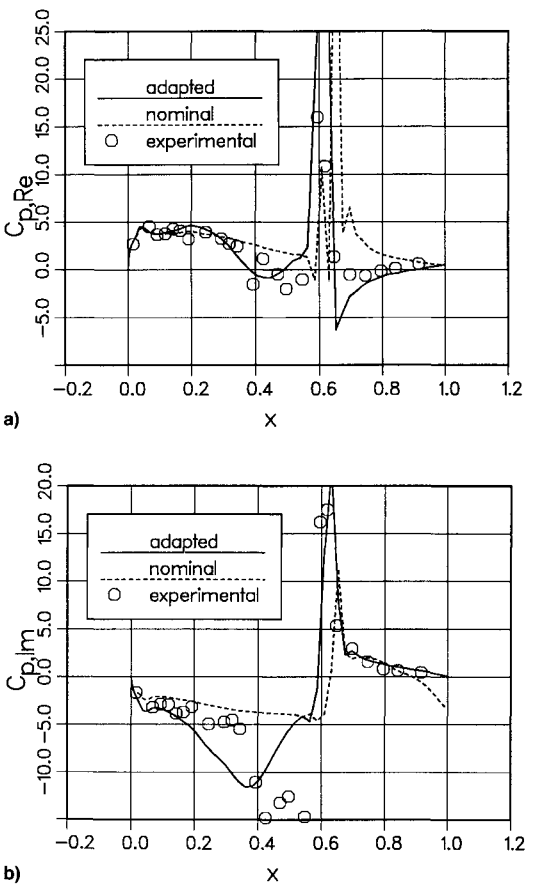

Fig. 8 Distributions of the reduced a) real and b) imaginary parts of C_p on the airfoil upper surface computed at the adapted ($M = 0.735$, $\bar{\alpha} = -0.15$ deg) and nominal conditions ($M = 0.751$, $\bar{\alpha} = 0.37$ deg) compared with the experimental data (nominal conditions).

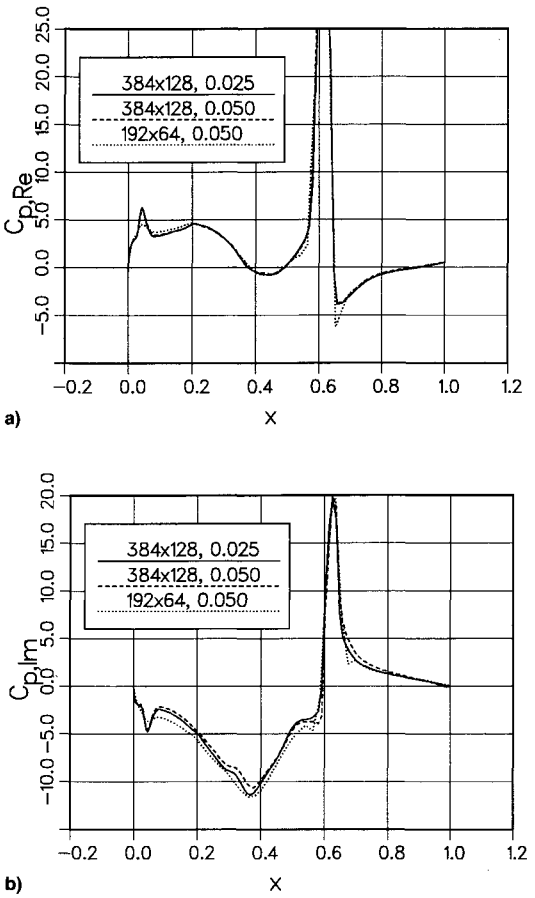

Fig. 9 Distributions of the reduced a) real and b) imaginary parts of C_p computed with the 384 × 128 grid using different time steps and with the 192 × 64 grid at the adapted conditions ($M = 0.735$, $\bar{\alpha} = -0.15$ deg).

The test calculations were conducted on the same IBM machine as used in the buffet calculations. The computation of one full oscillation cycle with the 192 × 64 grid and $\Delta t = 0.05$ giving sufficient spatial and temporal accuracy took about 3300 CPU seconds, a few percent of which was consumed by the grid movement algorithm. To obtain a stabilized periodic solution, three cycles were generally necessary after a start from a steady condition. The unsteady results presented are taken from the final cycle of each calculation.

Based on the calculations, several conclusions can be drawn. The sensitivity of transonic flow around supercritical airfoil designs is once more demonstrated. Clearly, the average steady pressure distribution must be predicted accurately in order to obtain a correct unsteady behavior, as suggested in Ref. 75. The wind-tunnel interference effects that are hard to evaluate at transonic speeds are important, which makes the validation of a computation method very difficult. In this particular case, the modeling of the free transition is at present beyond the capabilities of typical Navier–Stokes codes, which further adds to the uncertainties of the comparisons. However, the tests revealed that the discrepancies between the computed and measured results are not of numerical, but of physical origin, which confirms the proper function of the time-accurate solver under study. The code limitations are related to the basic physical modeling and not the time integration.

Conclusions

A new thin-layer Navier–Stokes solver for time-dependent compressible flows has been developed. The implicit, second-order accurate formulation facilitates the simulation of flows at high Reynolds numbers. Complex, arbitrarily deforming geometries can be handled using multiblock structured grids, and the grid deformation algorithm conserves the freestream exactly. An iterative time-stepping maintains the temporal and spatial accuracy of the basic scheme also at grid block interfaces and solid boundaries.

Several different types of flow cases have been run to validate the code, including turbulent flows around stationary and oscillating airfoils. In addition, a parametric study for the computational variables has been conducted to find an efficient code operating mode. All of the essential functions of the solver are found to work properly within the limitations of the basic formulation, mainly the simple transition and turbulence modeling. Of the time discretizations tested, the three-level fully implicit scheme is clearly the best choice, improving the accuracy over the implicit Euler scheme at negligible cost. Iterations within time steps are necessary to obtain accurate results, but the division of the computational effort between the time steps and iterations is not critical. This suggests that the efficiency of the current iteration method is not optimal. The use of a multigrid method with two grid levels speeds up the convergence within time steps, but the observed benefit is modest.

In its current status, the developed time-accurate flow solver can be applied to a wide variety of flow problems as designed, but the system demands substantial computer resources both in terms of execution time and memory. The CPU time consumption can probably be reduced to a certain extent by improving the efficiency of iteration within time steps and by finding suitable convergence criteria to prevent unnecessary iterations. Evidently, more work and test calculations are needed to find the best obtainable performance.

References

[1]Emery, A. E., "An Evaluation of Several Differencing Methods for Inviscid Fluid Flow Problems," *Journal of Computational Physics*, Vol. 2, 1968, p. 306.

[2]Magnus, R., and Yoshihara, H., "Unsteady Transonic Flows over an Airfoil," *AIAA Journal*, Vol. 13, No. 12, 1975, pp. 1622–1627.

[3]Steger, J. L., and Bailey, H. E., "Calculation of Transonic Aileron Buzz," *AIAA Journal*, Vol. 18, No. 3, 1980, pp. 249–255.

[4]Levy, L. L., "Experimental and Computational Steady and Unsteady Transonic Flows About a Thick Airfoil," *AIAA Journal*, Vol. 16, No. 6, 1978, pp. 564–572.

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

[6]Newsome, R. W., "Numerical Simulation of Near-Critical and Unsteady Subcritical Inlet Flow," *AIAA Journal*, Vol. 22, No. 10, 1984, pp. 1375–1379.

[7]Scott, J. N., and Hankey, W. L., Jr., "Numerical Simulation of Cold Flow in an Axisymmetric Centerbody Combustor," *AIAA Journal*, Vol. 23, No. 5, 1985, pp. 641–649.

[8]Gerz, T., "Direkte Simulation stabil geschichteter, homogen-turbulenter Scherströmungen," DLR-FB 88-04, Oberpfaffenhofen, Germany, 1988.

[9]Le Thanh, K. C., Troff, B., and Ta Phuoc Loc, "Numerical Study of Unsteady Incompressible Separated Viscous Flows Around an Obstacle," *Recherche Aérospatiale*, No. 1991-1, 1991, pp. 43–58.

[10]Smith, G. E., Whitlow, W., Jr., and Hassan, H. A., "Unsteady Transonic Flows Past Airfoils Using the Euler Equations," *Journal of Aircraft*, Vol. 24, No. 9, 1987, pp. 670–672.

[11]Jameson, A., and Venkatakrishnan, V., "Transonic Flows About Oscillating Airfoils Using the Euler Equations," AIAA Paper 85-1514, July 1985.

[12]Venkatakrishnan, V., and Jameson, A., "Computation of Unsteady Transonic Flows by the Solution of Euler Equations," *AIAA Journal*, Vol. 26, No. 8, 1988, pp. 974–981.

[13]Engel, K., Faden, M., and Pokorny, S., "Numerical Investigation of the Unsteady Flow Through a Counter-Rotating Fan," International Council of the Aeronautical Sciences Paper 92-3.1.2, Beijing, PRC, Sept. 1992.

[14]Kandil, O. A., and Chuang, H. A., "Unsteady Transonic Airfoil Computations Using Implicit Euler Scheme on Body-Fixed Grid," *AIAA Journal*, Vol. 27, No. 8, 1989, pp. 1031–1037.

[15]Kandil, O. A., and Chuang, H. A., "Unsteady Navier-Stokes Computations Past Oscillating Delta Wing at High Incidence," *AIAA Journal*, Vol. 28, No. 9, 1990, pp. 1565–1572.

[16]Kandil, O. A., and Chuang, H. A., "Computation of Vortex-Dominated Flow for a Delta Wing Undergoing Pitching Oscillations," *AIAA Journal*, Vol. 28, No. 9, 1990, pp. 1589–1595.

[17]Kandil, O. A., and Chuang, H. A., "Unsteady Inviscid and Viscous Computations for Vortex-Dominated Flows," *Journal of Aircraft*, Vol. 27, No. 5, 1990, pp. 387, 388.

[18]Ruo, S. Y., and Sankar, L. N., "Euler Calculations for Wing-Alone Configuration," *Journal of Aircraft*, Vol. 25, No. 5, 1988, pp. 436–441.

[19]Sankar, L. N., Malone, J. B., and Schuster, D., "Euler Solutions for Transonic Flow Past a Fighter Wing," *Journal of Aircraft*, Vol. 24, No. 1, 1987, pp. 10–16.

[20]Kandil, O. A., and Salman, A. A., "Prediction and Control of Slender-Wing Rock," International Council of the Aeronautical Sciences Paper 92-4.7.2, Beijing, PRC, Sept. 1992.

[21]Geissler, W., "Instationäres Navier-Stokes-Verfahren für beschleunigt bewegte Profile mit Ablösung," DLR-FB 92-03, Göttingen, Germany, 1992.

[22]Chaderjian, N. M., and Guruswamy, G. P., "Transonic Navier-Stokes Computations for an Oscillating Wing Using Zonal Grids," *Journal of Aircraft*, Vol. 29, No. 3, 1992, pp. 326–335.

[23]Wu, J., Kaza, K. R. V., and Sankar, L. N., "Technique for the Prediction of Airfoil Flutter Characteristics in Separated Flow," *Journal of Aircraft*, Vol. 26, No. 2, 1989, pp. 168–177.

[24]Nakamichi, J., "Some Computations of Unsteady Navier-Stokes Flow Around Oscillating Airfoil/Wing," National Aerospace Lab. TR-1004T, Japan, 1988.

[25]Fejtek, I., and Roberts, L., "Navier-Stokes Computation of Wing/Rotor Interaction for a Tilt Rotor in Hover," *AIAA Journal*, Vol. 30, No. 11, 1992, pp. 2595–2603.

[26]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.

[27]Baysal, O., Fouladi, K., Leung, R. W., and Sheftic, J. S., "Interference Flows Past Cylinder-Fin-Sting-Cavity Assemblies," *Journal of Aircraft*, Vol. 29, No. 2, 1992, pp. 194–202.

[28]Srinivasan, G. R., McCroskey, W. J., and Baeder, J. D., "Aerodynamics of Two-Dimensional Blade-Vortex Interaction," *AIAA Journal*, Vol. 24, No. 10, 1986, pp. 1569–1576.

[29]Sun, Y.-C., Shen, S., and Chen, N., "A Numerical Study of the Response of the Laminar Unsteady Separated Flow About a Circular Cylinder to Changes of Boundary Conditions," DLR-FB 90-10, Gottingen, Germany, 1990.

[30]Azevedo, J. L. F., "Three Dimensional Flow Simulation with Application to Aeroelastic Analysis," International Council of the Aeronautical Sciences Paper 88-4.10.2, Jerusalem, Israel, 1988.

[31]Visbal, M. R., "Dynamic Stall of a Constant-Rate Pitching Airfoil," *Journal of Aircraft*, Vol. 27, No. 5, 1990, pp. 400–407.

[32]Shida, Y., Kuwahara, K., Ono, K., and Takami, H., "Computation of Dynamic Stall of a NACA 0012 Airfoil," *AIAA Journal*, Vol. 25, No. 3, 1987, pp. 408–413.

[33]Simpson, B. L., "Unsteady Three-Dimensional Thin-Layer Navier-Stokes Solutions on Dynamic Blocked Grids," Ph.D. Dissertation, Dept. of Aerospace Engineering, Mississippi State Univ., Mississippi State, MS, Dec. 1988.

[34]Belk, D. M., "Unsteady Three-Dimensional Euler Equations Solutions on Dynamic Blocked Grids," AFATL-TR-86-74, Eglin AFB, FL, Oct. 1986.

[35]Thomadakis, M. P., and Tsangaris, S., "On the Prediction of Transonic Unsteady Flows Using Second Order Time Accuracy," *Proceedings of the 1st European Computational Fluid Dynamics Conference*, edited by C. Hirsch, J. Periaux, and W. Kordulla, Brussels, Belgium, 1992, pp. 711–718.

[36]Ridder, J. P., and Beddini, R. A., "Temporal and Acoustic Accuracy of an Implicit Upwind Method for Ducted Flows," *AIAA Journal*, Vol. 29, No. 11, 1991, pp. 1860–1867.

[37]Steinthorsson, E., Li, Z., Shih, T. I.-P., Nguyen, H. L., and Willis, E. A., "Flux-Vector Splitting Algorithm for Chain-Rule Conservation-Law Form," *AIAA Journal*, Vol. 29, No. 7, 1991, pp. 1101–1107.

[38]Rogers, S. E., and Kwak, D., "Upwind Differencing Scheme for

the Time-Accurate Incompressible Navier-Stokes Equations," *AIAA Journal*, Vol. 28, No. 2, 1990, pp. 253–262.

[39]Brenneis, A., and Eberle, A., "Unsteady Transonic Flows Past Airfoils and Wings Using a Fast Implicit Godunov Type Euler Solver," International Council of the Aeronautical Sciences Paper 88-6.3.1, Jerusalem, Israel, Sept. 1988.

[40]Batina, J. T., "Implicit Flux-Split Euler Schemes for Unsteady Aerodynamic Analysis Involving Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 29, No. 11, 1991, pp. 1836–1843.

[41]Chen, K.-H., and Pletcher, R. H., "Primitive Variable, Strongly Implicit Calculation Procedure for Viscous Flows at All Speeds," *AIAA Journal*, Vol. 29, No. 8, 1991, pp. 1241–1249.

[42]Hegna, H. A., "Numerical Solution of Incompressible Turbulent Flow over Airfoils near Stall," *AIAA Journal*, Vol. 20, No. 1, 1982, pp. 29, 30.

[43]Hegna, H. A., "Numerical Prediction of Dynamic Forces on Arbitrarily Pitched Airfoils," *AIAA Journal*, Vol. 21, No. 2, 1983, pp. 161, 162.

[44]Tuncer, I. H., Wu, J. C., and Wang, C. M., "Theoretical and Numerical Studies of Oscillating Airfoils," *AIAA Journal*, Vol. 28, No. 9, 1990, pp. 1615–1624.

[45]Sankar, L. N., and Hixon, D., "Development of Iterative Techniques for the Solution of Unsteady Compressible Viscous Flows," NASA-CR-188734, Aug. 1991.

[46]Anderson, W. K., Thomas, J. L., and Rumsey, C. L., "Extension and Applications of Flux-Vector Splitting to Unsteady Calculations on Dynamic Meshes," AIAA Paper 87-1152, June 1987.

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

[48]Hoffren, J., and Siikonen, T., "Time-Accurate Calculations with an Implicit Euler/Navier-Stokes Solver," *Proceedings of the 3rd ICFD Conference on Numerical Methods for Fluid Dynamics*, Reading, England, UK, 1992, pp. 575–581.

[49]Hoffren, J., "Unsteady Navier-Stokes Simulations of Airfoil Flows," *Proceedings of the 8th International Conference on Numerical Methods in Laminar and Turbulent Flow*, Swansea, Wales, UK, 1993, pp. 1065–1076.

[50]Voss, R., and Carstens, V., "Computation of Unsteady Transonic Flows Around Oscillating Airfoils Using Full Potential and Euler Equations," International Symposium of Aeroelasticity and Structural Dynamics, Aachen, Germany, 1989.

[51]Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1381–1388.

[52]Batina, J. T., "Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis," *AIAA Journal*, Vol. 29, No. 3, 1991, pp. 327–333.

[53]Rausch, R. D., Batina, J. T., and Yang, H. T. Y., "Spatial Adaptation of Unstructured Meshes for Unsteady Aerodynamic Flow Computations," *AIAA Journal*, Vol. 30, No. 5, 1992, pp. 1243–1252.

[54]Lee, E. M., and Batina, J. T., "Conical Euler Simulation of Wing Rock for a Delta Wing Planform," *Journal of Aircraft*, Vol. 28, No. 1, 1991, pp. 94–96.

[55]Batina, J. T., "Accuracy of an Unstructured-Grid Upwind-Euler Algorithm for the ONERA M6 Wing," *Journal of Aircraft*, Vol. 28, No. 6, 1991, pp. 397–402.

[56]Robinson, B. A., Batina, J. T., and Yang, H. T. Y., "Aeroelastic Analysis of Wings Using the Euler Equations with a Deforming Mesh," *Journal of Aircraft*, Vol. 28, No. 11, 1991, pp. 781–788.

[57]Hindman, R. G., "Generalized Coordinate Forms of Governing Fluid Equations and Associated Geometrically Induced Errors," *AIAA Journal*, Vol. 20, No. 10, 1982, pp. 1359–1367.

[58]Obayashi, S., "Freestream Capturing for Moving Coordinates in Three Dimensions," *AIAA Journal*, Vol. 30, No. 4, 1992, pp. 1125–1127.

[59]Zhang, H., Reggio, M., Trepanier, J. Y., and Camarero, R., "Discrete Form of the GCL for Moving Meshes and Its Implementation in CFD Schemes," *Computers and Fluids*, Vol. 22, No. 1, 1993, pp. 9–23.

[60]Guruswamy, G. P., "Vortical Flow Computations on a Flexible Blended Wing-Body Configuration," *AIAA Journal*, Vol. 30, No. 10, 1992, pp. 2497–2503.

[61]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.

[62]Siikonen, T., Hoffren, J., and Laine, S., "A Multigrid LU Factorization Scheme for the Thin-Layer Navier-Stokes Equations," International Council of the Aeronautical Sciences Paper 90-6.10.3, Stockholm, Sweden, Sept. 1990.

[63]Hoffren, J., "Time-Accurate Schemes for a Multi-Block Navier-Stokes Solver," Helsinki Univ. of Technology, Lab. of Aerodynamics, Rept. A-14, Helsinki, Finland, Dec. 1992.

[64]Yee, H. C., "A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods," NASA TM 101088, Feb. 1989.

[65]Stock, H. W., and Haase, W., "Determination of Length Scales in Algebraic Turbulence Models for Navier-Stokes Methods," *AIAA Journal*, Vol. 27, No. 1, 1989, pp. 5–14.

[66]Obayashi, S., and Kuwahara, K., "An Approximate LU Factorization Method for the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 63, No. 1, 1986, pp. 157–167.

[67]Steger, J. L., and Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods," *Journal of Computational Physics*, Vol. 40, 1981, pp. 263–293.

[68]Jameson, A., and Yoon, S., "Multigrid Solution of the Euler Equations Using Implicit Schemes," *AIAA Journal*, Vol. 24, No. 11, 1986, pp. 1737–1743.

[69]Siikonen, T., and Hoffren, J., "Multigrid Solution Method for the Euler Equations," Helsinki Univ. of Technology, Lab. of Aerodynamics, Rept. A-10, Helsinki, Finland, Jan. 1989.

[70]Siikonen, T., and Hoffren, J., "Solution of the Thin-Layer Navier-Stokes Equations for Laminar Transonic Flow," Helsinki Univ. of Technology, Lab. of Aerodynamics, Rept. A-11, Helsinki, Finland, Feb. 1989.

[71]Siikonen, T., "A Three-Dimensional Multigrid Algorithm for the Euler and the Thin-Layer Navier-Stokes Equations," Helsinki Univ. of Technology, Lab. of Aerodynamics, Rept. A-12, Helsinki, Finland, March 1991.

[72]Bristeau, M. O., Glowinski, R., Periaux, J., and Viviand, H. (eds.), "Numerical Simulation of Compressible Navier-Stokes Flows, a GAMM Workshop," *Notes on Numerical Fluid Mechanics*, Vol. 18, Vieweg, Brunswick, Germany, 1987.

[73]McDevitt, J. B., Levy, L. L., and Deiwert, G. S., "Transonic Flow About a Thick Circular-Arc Airfoil," *AIAA Journal*, Vol. 14, No. 5, 1976, pp. 606–613.

[74]Edwards, J. W., and Thomas, J. L., "Computational Methods for Unsteady Transonic Flows," *Unsteady Transonic Aerodynamics*, edited by D. Nixon, Vol. 120, Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 1989, pp. 211–261.

[75]Lambourne, N. C., Zwaan, R. J., Davis, S. S., Landon, R. H., and Mabey, D. G., "Compendium of Unsteady Aerodynamic Measurements," AGARD-R-702, 1982.

[76]Borel, C., and Bredif, M., "High Performance Parallelized Implicit Euler Solver for the Analysis of Unsteady Aerodynamic Flows," *Proceedings of the 1st European Computational Fluid Dynamics Conference*, edited by C. Hirsch, J. Periaux, and W. Kordulla, Brussels, Belgium, 1992, pp. 1069–1076.

[77]Simpson, L. B., and Whitfield, D. L., "Flux-Difference Split Algorithm for Unsteady Thin-Layer Navier-Stokes Solutions," *AIAA Journal*, Vol. 30, No. 4, 1992, pp. 914–922.