Efficient Simulation of Separated Three-Dimensional Viscous Flows Using the Boundary-Layer Equations

William R. Van Dalsem* and Joseph L. Steger†
NASA Ames Research Center, Moffett Field, California

A simple and computationally efficient algorithm for solving the unsteady three-dimensional boundary-layer equations in the time-accurate or relaxation mode is presented. Results of the new algorithm are shown to be in quantitative agreement with detailed experimental data for flow over a swept infinite wing. The separated flow over a 6:1 ellipsoid at angle of attack and the transonic flow over a finite wing with shock-induced "mushroom" separation are also computed and compared with available experimental data. It is concluded that complex, separated, three-dimensional viscous layers can be economically and routinely computed using a time-relaxation boundary-layer algorithm.

Introduction

A GOAL of computational fluid dynamics is to economically predict the viscous flow about three-dimensional aerodynamic configurations. Two distinct approaches toward this goal have evolved. Either the entire flow is resolved by solving the Reynolds-averaged Navier-Stokes equations, or the viscous-inviscid interaction approach is used, satisfying the boundary-layer equations and some inviscid system simultaneously. Currently, the Navier-Stokes approach offers the advantage of being more general, while the viscous-inviscid interaction approach can be more computationally efficient. In other words, by using the viscous-inviscid interaction approach, a given flow can be resolved using less computer resource, or a more complex flow can be computed with a given computer resource.

At this time, it is not clear that either the Navier-Stokes or the viscous-inviscid interaction approach has a clear advantage, or that they will remain distinct approaches. For example, new Navier-Stokes algorithms may incorporate boundary-layer solvers (or features of boundary-layer solvers) in an effort to improve near-wall efficiency. On the other hand, the viscous-inviscid interaction methods are using more complete viscous, inviscid, and interaction algorithms that amount to a different and more efficient method of solving the Navier-Stokes equations. In light of these trends, the purpose of the present work is to develop an efficient, unsteady three-dimensional boundary-layer algorithm that can be used either to help improve the efficiency of the Navier-Stokes methods[1] or in the more general viscous-inviscid interaction algorithms.[2]

One of the major reasons for the recent emphasis on the Navier-Stokes approach is that it is felt to be difficult to apply the viscous-inviscid interaction approach to complex three-dimensional flows. This stems from the lack of reliable viscous-inviscid interaction methods and the perception that boundary-layer algorithms are inadequate when applied to complex flow. The perception is partly due to the saddle-point behavior of the boundary-layer equations at separation. However, this behavior can be entirely avoided by using the inverse mode. Another difficulty in computing complex three-dimensional flows is the need to provide additional flow modeling (e.g., the crossflow velocity profiles) if the popular integral boundary-layer equations are used. However, if the full partial-differential boundary-layer equations are solved, only the same turbulence modeling required by the Navier-Stokes algorithms must be supplied. Finally, past integral and finite-difference three-dimensional boundary-layer algorithms have been space-marching schemes which can be cumbersome and unreliable in complex flow, especially when streamwise and crossflow separation are present.

Here, the unsteady three-dimensional partial-differential boundary-layer equations are solved in the direct and inverse modes using a finite-difference method. By using the inverse mode no difficulties are encountered at separation, whereas the use of the full partial-differential equations results in an algorithm that can be applied to a wide class of problems. The limitations of the standard space-marching approach (when applied to three-dimensional flows) are avoided by using a time-relaxation scheme. By relaxing the boundary-layer equations in time, flow-dependent difference operators that automatically adjust to the flow direction can replace problem-dependent space sweeps. Hence, one relatively simple code can easily be applied to a wide variety of three-dimensional flows without any changes to the algorithm. Relaxation schemes are also more robust and forgiving of approximate boundary conditions than are space-marching schemes. For example, with a space-marching scheme all variables must be specified at in-flow and some side boundaries, whereas zero-gradient boundary conditions can often be used with a relaxation numerical algorithm. On the other hand, compared to space-marching algorithms, relaxation schemes can be relatively computationally expensive when used to simulate simple steady flows. This drawback can be minimized if care is taken to design the relaxation algorithm so that a steady state can be reached in few iterations. Also, since a relaxation boundary-layer scheme can be at least half as expensive as a second-order space-marching scheme on a per-iteration basis, when included in a viscous-inviscid interaction algorithm the relaxation boundary-layer scheme should result in a more efficient code. Finally, the same code can also be used to simulate unsteady flows.

The following two sections present the transformed equations and the relaxation algorithm. Computational results

Presented as Paper 85-4064 at the AIAA Third Applied Aerodynamics Conference, Colorado Springs, CO, Oct. 14-16, 1985; received Nov. 18, 1985; revision received July 29, 1986.

*Research Scientist. Member AIAA.
†Senior Staff Scientist. Associate Fellow AIAA.

are then compared to experimental data for a variety of flowfields. These comparisons are used to verify the present formulation and suggest that the boundary-layer equations can be used on a routine basis to compute complex separated three-dimensional viscous layers.

Unsteady Three-Dimensional Boundary-Layer Equations

Instead of solving the boundary-layer equations in similarity variables so as to scale out the viscous-layer growth, a general x,y,z to $\xi(x,z),\eta(x,y,z),\zeta(x,z)$ coordinate transformation is used (see Fig. 1). Hence, a complex similarity transformation is avoided and solution-adaptive gridding is readily implemented as demonstrated in Refs. 3–5. Neglecting surface curvature, the compressible boundary-layer equations for the unsteady, three-dimensional flow of a perfect gas can be written in $\xi(x,z),\eta(x,y,z),\zeta(x,z)$ coordinates as

x-momentum:

$$\rho(u_t + Uu_\xi + Vu_\eta + Wu_\zeta) = -\beta(p_\xi \xi_x + p_\zeta \zeta_x) + (\mu u_\eta \eta_y)_\eta \eta_y \tag{1a}$$

z-momentum:

$$\rho(w_t + Uw_\xi + Vw_\eta + Ww_\zeta) = -\beta(p_\xi \xi_z + p_\zeta \zeta_z) + (\mu w_\eta \eta_y)_\eta \eta_y \tag{1b}$$

Perfect gas relation:

$$p = \rho T \tag{1c}$$

Energy (H = total enthalpy):

$$\rho(H_t + UH_\xi + VH_\eta + WH_\zeta) = \left\{\frac{\mu}{Pr}\left[H_\eta \eta_y + \frac{Pr-1}{2}(u^2+w^2)_\eta \eta_y\right]\right\}_\eta \eta_y \tag{1d}$$

Continuity:

$$\rho_t + (\rho u)_\xi \xi_x + (\rho u)_\eta \eta_x + (\rho u)_\zeta \zeta_x + (\rho v)_\eta \eta_y + (\rho w)_\xi \xi_z + (\rho w)_\eta \eta_z + (\rho w)_\zeta \zeta_z = 0 \tag{1e}$$

where U, V, and W are unscaled contravariant velocities

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} \xi_t + \xi_x u + \xi_z w \\ \eta_t + \eta_x u + \eta_y v + \eta_z w \\ \zeta_t + \zeta_x u + \zeta_z w \end{bmatrix}$$

and $\beta = (p/\rho u^2)_\infty$. Viscosity, pressure, temperature, and density are nondimensionalized by their freestream values. The velocities u and w and the coordinates x and z are nondimensionalized by the freestream velocity and a characteristic length, respectively. The v and y variables are nondimensionalized by the same quantities divided by $\sqrt{Re_\infty}$. For turbulent flows, the viscosity coefficient μ is the sum of the molecular and eddy viscosities.

If no streamwise separation is present, these equations may be solved with the pressure specified (the direct mode). However, streamwise separation may occur, and to avoid saddle-point behavior the equations must be solved in the inverse mode. In this work, the wall shear stress (τ_w) is used as the inverse forcing fuction on a body, while the wake centerline velocity would be used in free-shear layers. It is interesting to note that it is possible to solve the three-dimensional boundary-layer equations in a mixed direct/inverse mode. For example, the x-momentum equation may be solved in the inverse mode, while the z-momentum equation can be solved in the direct mode, or vice versa. The authors have observed that saddle-point behavior at the streamwise separation line may be avoided if just the momentum equation in roughly the freestream direction is solved in the inverse mode (i.e., the other momentum equation can be solved in the direct mode).

The boundary conditions for space-marching three-dimensional boundary-layer computations require a great deal of attention. However, with a relaxation algorithm much simpler boundary conditions can often be used. There are at least two reasons for this. First, with a relaxation algorithm the physical domain of the computation can be quite general. For example, on a three-dimensional wing or body a relaxation algorithm can simply capture the stagnation line or point as part of the computation of the flow about the entire body or wing. Second, as with Navier-Stokes algorithms, the boundary values can be "brought up" with the rest of the solution via zero- or the first-order gradient boundary conditions. For example, it would be very difficult to specify the flow quantities near a wing tip, as may be required by a space-marching scheme. However, with a relaxation algorithm the tip values can be brought up during the iteration process using, for example, a zero-gradient boundary condition. The global impact of this approximate boundary condition can be minimized by refining the grid in the tip region.

Numerical Algorithm

Solution Scheme

A time-like algorithm which can be run in either a time-accurate or iterative mode is used to avoid the need for complex space-marching patterns. The relaxation algorithm was designed to yield quick convergence to a steady-state solution. Because the pressure gradient forcing terms in the momentum equations are treated as given functions, the boundary-layer equations are weakly coupled and can be solved sequentially at each time step. As a result, a semi-implicit algorithm can be used at each time or iterative step, yet only scalar-like uncoupled equations are solved. Spatial derivatives in the momentum (and energy) equations are approximated with implicit second-order-accurate central difference operators in the direction normal to the body, and flow-dependent second-order-accurate operaters in the other two directions. That is, central differencing is used in η, and upwind differencing is used in ξ and ζ. For example, when differencing a term such as u_ξ a backward difference is used if the coefficient ρU is positive and a forward difference is used if the coefficient ρU is negative. In the continuity equation, the ξ and ζ derivatives are approximated with second-order-accurate central-difference operators, and trapezoidal-rule integration is used in the η direction to yield v.

The time-accurate algorithm is presented below using conventional operators defined in terms of the shift operator $E_\xi^{\pm 1} u_j = u_{j\pm 1}$

$$\nabla_\xi = (1 - E_\xi^{-1})/(\Delta\xi)$$

$$\Delta_\xi = (E_\xi^{+1} - 1)/(\Delta\xi)$$

$$\delta_\xi = (E_\xi^{+1} - E_\xi^{-1})/(2\Delta\xi)$$

$$\bar{\delta}_\xi = (E_\xi^{+1/2} - E_\xi^{-1/2})/(\Delta\xi)$$

and the second-order-accurate upwind operator

$$\hat{\alpha}\hat{\delta}_\xi = \left(\frac{\alpha + |\alpha|}{2}\right)\nabla_\xi\left(\frac{3 - E_\xi^{-1}}{2}\right) + \left(\frac{\alpha - |\alpha|}{2}\right)\Delta_\xi\left(\frac{3 - E_\xi^{+1}}{2}\right)$$

where α is the convective coefficient (e.g., ρU). Using a notation that space-time indices are not written unless changed (e.g., $u=u^n_{j,k,l}$, $u^{n+1}=u^{n+1}_{j,k,l}$, etc.), the semi-implicit scheme is written as follows.

Update u at the new time step from the x-momentum equation ($U=\xi_x u+\xi_z w$, etc.):

$$\rho(\nabla_t u^{n+1}+\hat{U}\hat{\delta}_\xi u^{n+1}+V\delta_\eta u^{n+1}+\hat{W}\hat{\delta}_\zeta u)$$
$$=-\beta(\xi_x\delta_\xi p+\zeta_x\delta_\zeta p)+\eta_y\bar{\delta}_\eta(\mu\eta_y\delta_\eta u^{n+1}) \quad (2a)$$

where if $\hat{U}<0$, $\hat{\delta}_\xi$ will actually operate on explicit data, and as noted below, care is taken to maintain stability of terms that are explicitly three-point differenced. This system of equations is easily solved by inverting scalar-tridiagonal matrices in η. In a similar way, w is updated from the z-momentum equation ($U=\xi_x u^{n+1}+\xi_z w$, etc.):

$$\rho(\nabla_t w^{n+1}+\hat{U}\hat{\delta}_\xi w^{n+1}+V\delta_\eta w^{n+1}+\hat{W}\hat{\delta}_\zeta w)$$
$$=-\beta(\xi_z\delta_\xi p+\zeta_z\delta_\zeta p)+\eta_y\bar{\delta}_\eta(\mu\eta_y\delta_\eta w^{n+1}) \quad (2b)$$

and H is updated from the energy equation ($U=\xi_x u^{n+1}+\xi_z w^{n+1}$, etc.):

$$\rho(\nabla_t H^{n+1}+\hat{U}\hat{\delta}_\xi H^{n+1}+V\delta_\eta H^{n+1}+\hat{W}\hat{\delta}_\zeta H)$$
$$=\eta_y\bar{\delta}_y\left\{\frac{\mu}{Pr}\left[\eta_y\delta_\eta H^{n+1}+\frac{Pr-1}{2}\eta_y\bar{\delta}_\eta(u^2+w^2)^{n+1}\right]\right\} \quad (2c)$$

Finally, obtain $\rho=p/T$ and integrate the continuity equation for v using updated values of u, w, and ρ:

$$\nabla_\eta(\rho v)^{n+1}=-[(1+E_\eta^{-1})/2][\xi_x\delta_\xi(\rho u)+\eta_x\delta_\eta(\rho u)$$
$$+\zeta_x\delta_\zeta(\rho u)+\xi_z\delta_\xi(\rho w)+\eta_z\delta_\eta(\rho w)$$
$$+\zeta_z\delta_\zeta(\rho w)+\nabla_t\rho]^{n+1} \quad (2d)$$

The coefficients μ and Pr are then evaluated using either the Cebeci algebraic turbulence model[6] (generalized to three dimensions) or the Baldwin-Lomax model.[7]

The algorithm presented above is second-order-accurate in space and first-order-accurate in time. A simple corrector sequence can be used to achieve second-order time accuracy and provide additional stability of the explicitly differenced terms, following the same technique employed in the authors' two-dimensional space marching algorithm.[3-5] However, if only a steady-state solution is required, enhanced stability and faster convergence is obtained by implicitly including the diagonal contribution of any explicit operator. In this case, the algorithm takes on a similarity to a SLOR scheme. Finally, in many steady-state cases an assumption of constant total enthalpy can replace the partial-differential energy equation.

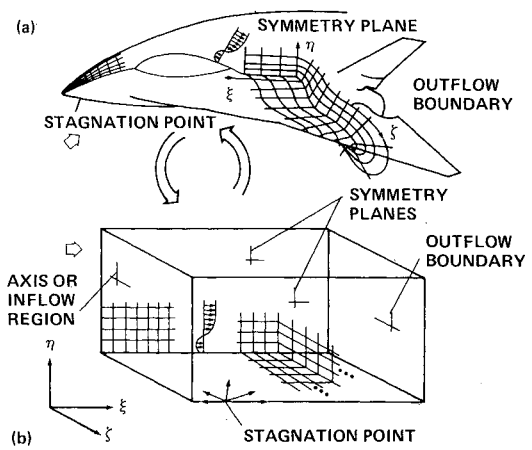

Fig. 1 Physical to computational domain coordinate transformations: a) physical domain; b) computational domain.

Direct and Inverse Modes

Near and in reversed flow regions, the boundary-layer equations are solved in the inverse mode to avoid saddle-point behavior at the separation point. To solve the boundary-layer equations in the inverse mode, the algorithm must be modified so that the inverse forcing functions can be specified. As in Refs. 3-5, the wall shear stress τ_w (on a solid body) and the wake centerline velocity (in a wake) are used as the inverse forcing functions. This is achieved by replacing the pressure terms in the momentum equations with expressions containing the inverse forcing functions. These relations are obtained by applying the momentum equations at the wall and wake centerline. For example, the x-momentum equation evaluated at a wall yields

$$\beta(p_\xi\xi_x+p_\zeta\zeta_x)=(\mu u_\eta\eta_y)_\eta\eta_y|_w$$
$$=2\left(\frac{\mu_2+\mu_1}{2}\frac{u_2-u_1}{y_2-y_1}-\tau_x|_w\right)\Big/(y_2-y_1) \quad (3)$$

These expressions allow the elimination of the pressure terms from the momentum equations. However, if the entire inverse forcing term is treated implicitly, the central differenced η operators do not generate a simple tridiagonal matrix. Instead, a tridiagonal-like matrix with an additional column of nonzero coefficients of the following form must be inverted:

$$\begin{bmatrix} b_2 & c_2 & & & & & \\ f_3 & b_3 & c_3 & & & & \\ f_4 & a_4 & b_4 & c_4 & & & \\ \cdot & & \cdot & \cdot & \cdot & & \\ \cdot & & & \cdot & \cdot & c_{k\max-1} & \\ f_{k\max} & & & & a_{k\max} & b_{k\max} & \end{bmatrix} u=d \quad (4)$$

This augmented scalar tridiagonal matrix system is efficiently solved using an algorithm developed specifically for this application. The algorithm utilizes an LU decomposition to solve the augmented scalar tridiagonal system (Refs. 3 and 4) and costs only 40% more than the standard scalar tridiagonal inversion. Because the inverse forcing functions are implemented implicitly, no iteration is required at each time step to obtain the desired wall shear or wake centerline velocity.

Vectorization

This algorithm readily lends itself to vector computer architecture by arranging the computation of the inversion matrix coefficients and the actual inversion in a vectorized loop. To do this, one spatial direction must be chosen along which to create the vectors. It is convenient to choose the direction along the body roughly normal to the freestream. Then at each time step, along each $\xi=$ const plane, all the coefficient matrices are computed, stacked together, and fed to a vectorized version of a scalar tridiagonal (or augmented tridiagonal) solver. Using this approach, the algorithm requires approximately 2 μs/point/iteration on a Cray-XMP processor.

Results

A variety of flows have been computed with the new algorithm, but only three flows will be discussed here. First, quantitative agreement between computed results and detailed experimental data for separated flow over a swept infinite wing will be shown. The efficiency of the algorithm will also be indicated. Then results of computing the separated flows over a 6:1 prolate spheroid at angle of attack and the separated transonic flow over a swept finite wing will be presented to indicate some of the complex flows which can be simulated with the present algorithm.

Van den Berg and Elsenaar[8] acquired detailed experimental data for a flow which mimics the incompressible flow over a wing with a 35 deg sweep and large aspect ratio. The boundary layer on this wing experiences an adverse pressure gradient (see Fig. 2) which eventually results in turbulent three-dimensional separation at approximately 90% of chord, as shown in Fig. 3. This flow was computed in the inverse mode on a 50 (streamwise) × 50 (normal to the body) × 20 (span) mesh. In all computations, the first grid point off the surface is placed at a distance of less than $y^+ = 1$. The large wing aspect ratio condition was modeled with an aspect ratio of 10 and extrapolation boundary condition at the two wing tips. In Fig. 4 some of the computed boundary-layer integral and edge-flow parameters are compared with experimental data. Overall, the agreement is quite good, especially considering that this is a separated turbulent three-dimensional flow.

To indicate the basic efficiency of the algorithm, the residual history for the previous case is presented in Fig. 5. On a 50,000-point mesh a practical solution is obtained after a three-order-of-magnitude drop in the residual, or roughly 30 iterations and 30 s on a Cray-XMP processor. (The residual can be dropped to machine round-off error in approximately 100 iterations.) When strongly interacting flows are computed using the viscous-inviscid interaction approach, roughly this number of viscous-inviscid iterations (and viscous-flow evaluations) are required to resolve the interaction. So, since the present algorithm is at least twice as efficient on a per-iteration basis as a second-order space-marching scheme, its use should result in more general and possibly more efficient viscous-inviscid interaction codes.

Detailed experimental data have been obtained for the flow over a 6:1 prolate spheroid at angle of attack.[9-12] The measured flowfields vary in angle of attack, Reynolds number, and method of transition. For this paper, the authors computed the essentially transitional flow (i.e., $u_\infty = 45$ m/s) over the ellipsoid at 30 deg angle of attack. The

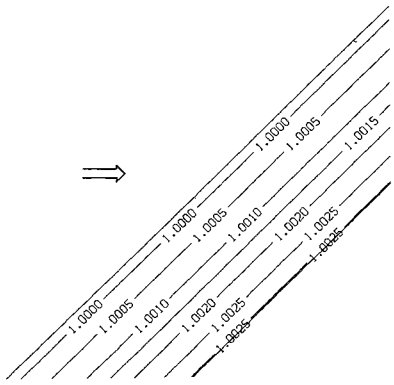

Fig. 2 Pressure contours for the incompressible turbulent flow over the swept infinite wing of Van den Berg and Elsenaar.[8]

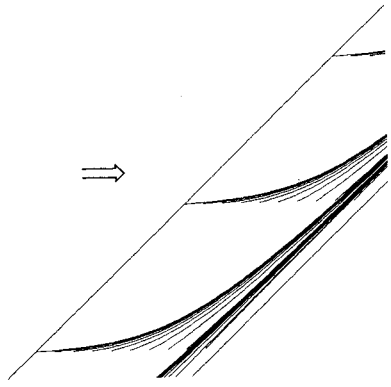

Fig. 3 Computed near-surface particle paths on the Van den Berg and Elsenaar wing.

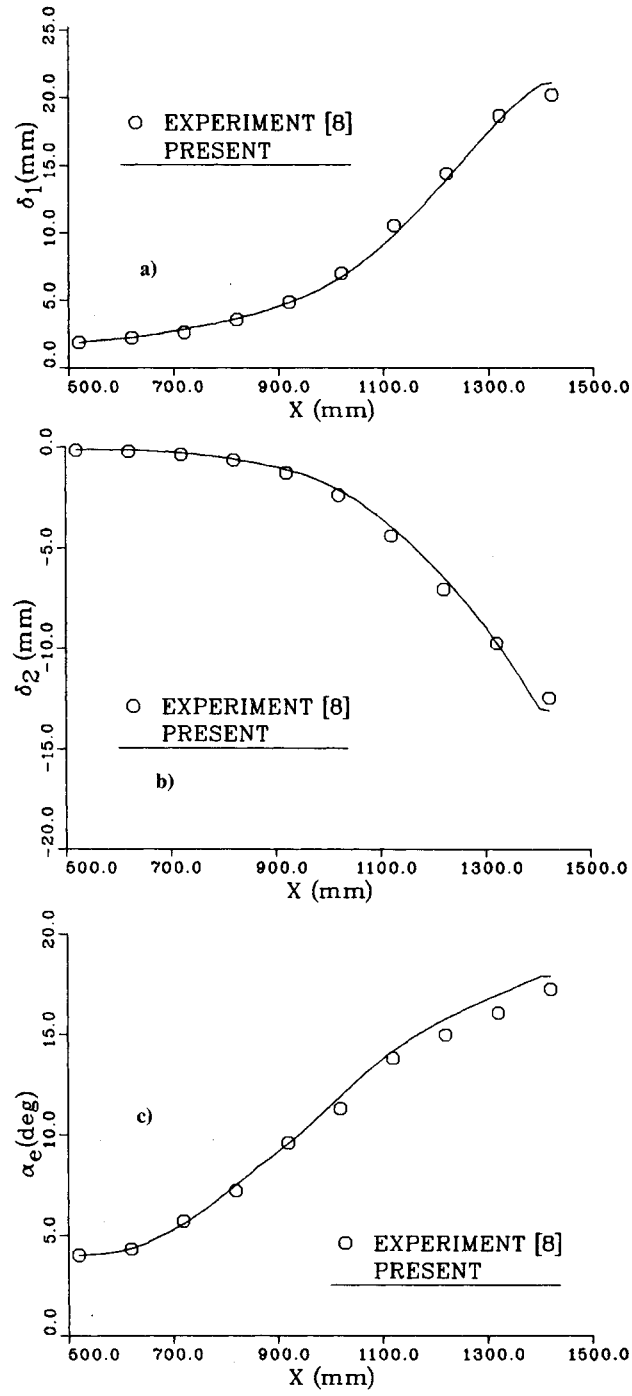

Fig. 4 Comparison of computed and experimental boundary-layer parameters for the Van den Berg and Elsenaar wing: a) streamwise displacement-thickness distribution; b) crosswise displacement-thickness distribution; c) boundary-layer edge-flow angle (relative to tunnel axis) distribution.

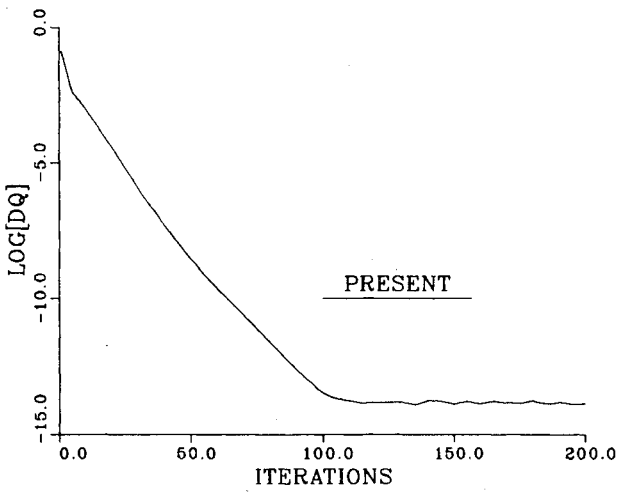

Fig. 5 Typical residual history.

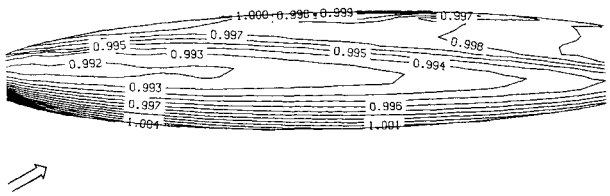

Fig. 6 Measured pressure distribution[12] on a 6:1 prolate spheroid at 30 deg angle of attack, $Re = 7.2 \times 10^6$, and $M_\infty = 0.13$.

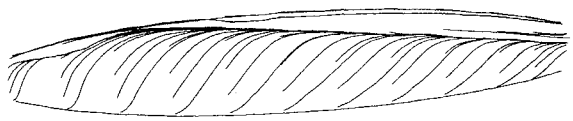

Fig. 7 Measured near-surface particle paths for the flow over the prolate spheroid of Fig. 6.

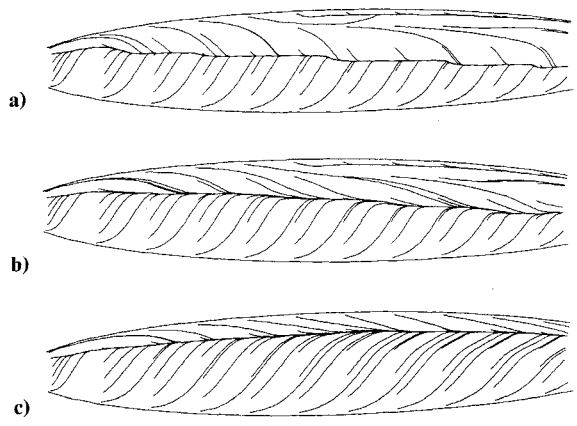

Fig. 8 Computed near-surface particle paths for the flow over the prolate spheroid of Fig. 6: a) completely laminar; b) transition at the crossflow separation line; c) completely turbulent.

experimental pressure and near-surface particle traces are presented in Figs. 6 and 7, respectively. The experimental particle traces were obtained by computing the streamlines of particles released on the body (and required to remain on the body) given the experimental wall skin-friction data.

The flow was computed by solving the boundary-layer equations, including the nonunity geometrical metric coefficients outlined by Wang,[13] Cebeci,[14] and others, on a 45 (axial direction) $\times$ 50 (normal to body) $\times$ 25 (circumferential

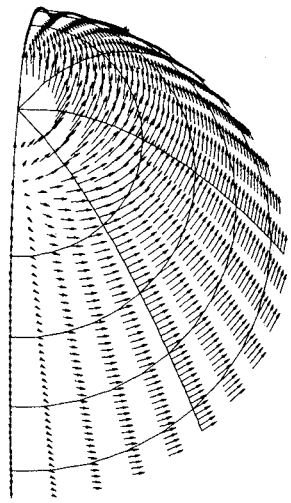

Fig. 9 Close-up of captured stagnation point of the flow over the prolate spheroid of Fig. 6.

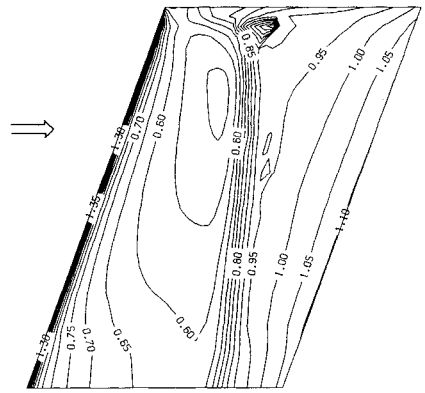

Fig. 10 Computed pressure contours[16] on a NACA-0012 wing with $AR = 3$, 2 deg angle of attack, $Re = 8.0 \times 10^6$, and $M_\infty = 0.826$.

direction) mesh. A cubic spline representation of the experimental pressure distribution was specified as the forcing function. In Fig. 8 computed surface streamlines are presented for three conditions: fully laminar, transition at the crossflow separation line, and fully turbulent. From these three figures it is clear that this flow is very sensitive to the laminar/turbulent state of the boundary layer. Specifically, in the fully laminar and transitional computations the streamlines indicate two counterrotating streamwise vortexes (indicated by the coalescence/dispersion/coalescence streamline patterns in the meridional direction), while in the fully turbulent case only one fairly small streamwise vortex is indicated by the one streamwise coalescence line. By comparison with the experimental particle traces it is apparent that the experimental flow is neither entirely laminar nor turbulent, and that the transitional result is in the closest agreement with the experimental data. The simple transition model used here has underpredicted the extent of turbulent flow, and as a result a slightly larger separated region is predicted than is observed experimentally. However, it is clear that these flows can be computed routinely with the present code and a refined transitional model could be developed using the code. Figure 9 is a close-up of the captured stagnation point. As mentioned earlier, with the present algorithm there is no special treatment of the stagnation region since there is no requirement to march away from the stagnation point.

The transonic flow over a NACA-0012 wing with a 20 deg sweep and an aspect ratio of 3 has been studied by Lockman and Seegmiller[15] over a range of freestream Mach numbers and angles of attack. At $M_\infty = 0.826$, $\alpha = 2$ deg, a strong shock runs across the wing (see Fig. 10) which contributes to the formation of complex streamwise and crossflow separation regions. Unfortunately, the experimental data are not of

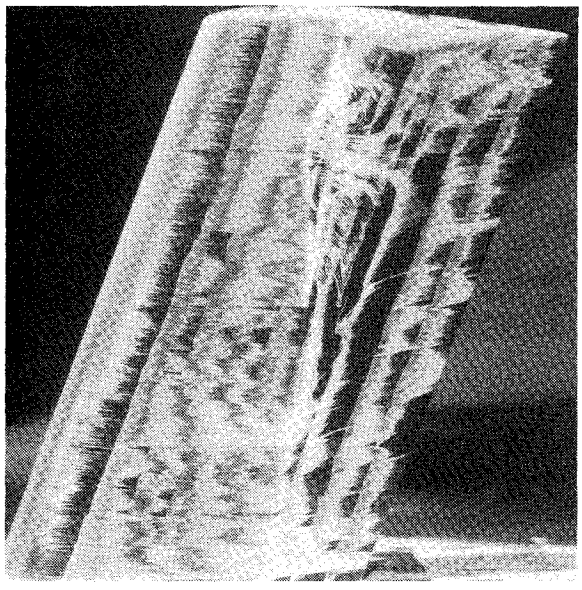

Fig. 11 Experimental oil-flow patterns[15] for the flow over the NACA-0012 wing of Fig. 10.

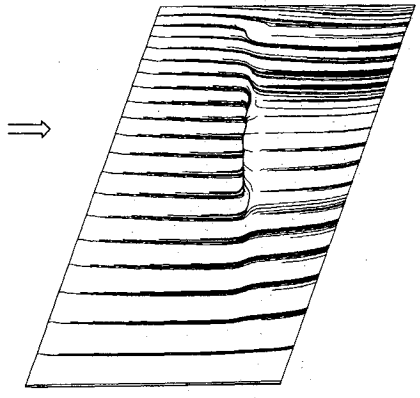

Fig. 12 Computed near-surface particle paths for the flow over the NACA-0112 wing of Fig. 10.

sufficient detail to supply the input for a boundary-layer computation. However, this flow has also been computed using a Navier-Stokes algorithm[16] (called TNS) which does supply detailed results. In general, the pressure distribution generated by the TNS code is in good agreement with the experimental data, while the computed skin-friction distribution is only qualitatively correct. Consequently, only the pressure distribution generated by the TNS code can be used as input for a boundary-layer computation, and this was done by operating in the inverse mode and iterating to the TNS pressure distribution. It is important to note that if this iteration procedure were to converge to exactly the specified pressure distribution, saddle-point behavior would be encountered at any streamwise separation line. To avoid this, the relaxation parameter [ω in $\tau_x^{n+1} = \tau_x^n + \omega(p_{bl} - p_{TNS})$] is made proportional to the absolute value of the computed wall shear stress. The result is that the specified pressure distribution is accurately matched, but the saddle-point behavior is avoided since the iteration procedure never reaches exactly the specified pressure at the separation line. The experimental oil-flow and computed near-surface particle paths are compared in Figs. 11 and 12. As shown, details of the flow near the tip, the "mushroom" separation region, and (farther inboard) the continued deflection of the flow through the shock are all captured. Overall, even for this complex case, the agreement between the experimental and computed boundary layer flows is quite encouraging.

Conclusion

By using the relaxation approach and flow-dependent space-difference operators, the boundary-layer equations have been applied to complex separated three-dimensional viscous layers. By relaxing the unsteady boundary-layer equations in time, a simple algorithm which is both robust and efficient has been developed. Results of the new code have been shown to be in quantitative agreement with detailed experimental data and capable of computing complex transonic flows with streamwise separation and arbitrary crossflows. As a stand alone code, the new algorithm has already proved useful in estimating the impact of transition, turbulence models, and grid resolution, and as a diagnostic tool for improving the skin-friction estimates of Navier-Stokes simulations. The authors have also used this boundary-layer algorithm to improve the efficiency of a Navier-Stokes algorithm using the Fortified Navier-Stokes (FNS) approach first described in Ref. 2 and detailed in Ref. 1.

References

[1]Van Dalsem, W. R. and Steger, J. L., "Using the Boundary-Layer Equations in Three-Dimensional Viscous Flow Simulation," *Proceedings of the AGARD Fluid Dynamics Panel Symposium on Applications of Computational Fluid Dynamics in Aeronautics*, Aix-en-Provence, France, April 1986.

[2]Steger, J. L. and Van Dalsem, W. R., "Developments in the Simulation of Separated Flows Using Finite-Difference Methods," *Proceedings of the Third Symposium on Numerical and Physical Aspects of Aerodynamic Flows*, California State Univ., Long Beach, Jan. 1985.

[3]Van Dalsem, W. R. and Steger, J. L., "Finite-Difference Simulation of Transonic Separated Flow Using a Full-Potential Boundary-Layer Interaction Approach," AIAA Paper 83-1689, July 1983.

[4]Van Dalsem, W. R., "Simulation of Separated Transonic Airfoil Flow by Finite-Difference Viscous-Inviscid Interaction," Ph.D. Thesis, Stanford Univ., June 1984.

[5]Van Dalsem, W. R. and Steger, J. L., "Simulation of Transonic Separated Airfoil Flow by Finite-Difference Viscous-Inviscid Interaction," *Proceedings of the Ninth International Conference on Numerical Methods in Fluid Dynamics*, Saclay, France, June 1984.

[6]Cebeci, T., "Calculation of Compressible Turbulent Boundary Layers with Heat and Mass Transfer," AIAA Paper 70-741, June 1970.

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

[8]Van den Berg, B. and Elsenaar, A., "Measurements in a Three-Dimensional Incompressible Turbulent Boundary Layer in an Adverse Pressure Gradient Under Infinite Swept Wing Conditions," NLR TR-72092, Aug. 1972.

[9]Meier, H. U. and Kreplin, H. P., "Experimental Investigation of the Boundary Layer Transition and Separation on a Body of Revolution," *Zeitschrift f. flugwiss. Weltraumforsh.*, Vol. 4, 1981, pp. 65-71.

[10]Meier, H. U., Kreplin, H. P., and Ming, X., "Problems Associated with Artificial Boundary Layer Transition," AIAA Paper 83-1673, July 1983.

[11]Meier, H. U., Kreplin, H. P., and Vollmers, H., "Development of Boundary Layers and Separation Patterns on a Body of Revolution as Incidence," *Proceedings of the Second Symposium on Numerical and Physical Aspects of Aerodynamic Flows*, California State Univ., Long Beach, Jan. 1983.

[12]Meier, H. U. and Cebeci, T., "Flow Characteristics of a Body of Revolution at Incidence," *Proceedings of the Third Symposium on Numerical and Physical Aspects of Aerodynamic Flows*, California State Univ., Long Beach, Jan. 1985.

[13]Wang, K. C., "Boundary Layer Over a Blunt Body at High Incidence with an Open-Type of Separation," *Proceedings of the Royal Society*, Vol. A340, 1974, pp. 33-55.

[14]Cebeci, T., Khattab, A. K., and Stewartson, K., "On Nose Separation," *Journal of Fluid Mechanics*, 1980, Vol. 97, Part 3, pp. 435-454.

[15]Lockman, W. K. and Seegmiller, H. L., "An Experimental Investigation of the Subcritical and Supercritical Flow About a Swept Semispan Wing," NASA TM-84367, June 1983.

[16]Holst, T. L., Kaynak, U., Gundy, K. L., Thomas, S. D., Flores, J., and Chaderjian, N., "Numerical Solution of Transonic Wing Flows Using an Euler/Navier-Stokes Zonal Approach," AIAA Paper 85-1640, July 1985.