NASA Technical Memorandum 102815

A Comparison of Two Central Difference Schemes for Solving the Navier-Stokes Equations

C. M. Maksymiuk, Ames Research Center, Moffett Field, California
R. C. Swanson, Langley Research Center, Hampton, Virginia
T. H. Pulliam, Ames Research Center, Moffett Field, California

July 1990

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

SUMMARY

Five viscous transonic airfoil cases have been computed by two significantly different computational fluid dynamics codes: an explicit finite-volume algorithm with multigrid, and an implicit finite-difference approximate-factorization method with eigenvector diagonalization. Both methods are described in detail, and their performance on the test cases is compared. The codes utilized the same grids, turbulence model, and computer to provide the truest test of the algorithms. The two approaches produce very similar results, which, for attached flows, also agree well with experimental results; however, the explicit code is considerably faster.

INTRODUCTION

The development of algorithms for the Navier-Stokes equations has proceeded along two distinct but intertwined paths: explicit methods, which are less computationally intensive but are restricted by stability considerations to small time-steps, and implicit methods, which trade a larger operation count for greater stability. Jameson et al. pioneered the use of an explicit modified Runge-Kutta method for the Euler equations (ref. 1). With the enhancements of multigrid, implicit residual smoothing, and local time-stepping, the method was practical for application to the Navier-Stokes equations. Analogously, Beam and Warming developed an implicit approximate-factorization scheme which was useful for reducing the work of a two- or three-dimensional implicit operator to a series of one-dimensional operators (ref. 2). Enhanced by local time-stepping, eigenvector diagonalization, and grid sequencing, which applies some of the ideas of multigrid, this algorithm also proved applicable to the Navier-Stokes equations.

Both methods have been rigorously tested against experimental data for a variety of flow conditions, and both are in widespread use today in research and practical applications. The purpose of this study is to compare the methods with regard to their capability to compute transonic viscous airfoil flows. Computations were performed using two codes: FLOMG, which is based on Jameson's multigrid algorithm, and ARC2D, which uses the Beam-Warming implicit method. Since computational results can be sensitive to various features of the grids, both codes used the same grids, three of which were generated for each airfoil. To provide the most controlled test of the methods, both codes also used the same turbulence model and computer. Five cases, with conditions ranging from subcritical attached flow to strong shock-induced separation, were computed. Results include plots of pressure coefficient, skin-friction coefficient, boundary-layer profiles, and displacement thickness, as well as tabulated force coefficients and convergence times. Comparisons with experimental data are made when data are available.

NAVIER-STOKES EQUATIONS

The nondimensional, strong-conservation-law form of the two-dimensional Navier-Stokes equations in Cartesian coordinates is

$$\partial_t Q + \partial_x E + \partial_y F = Re^{-1}(\partial_x E_v + \partial_y F_v) \tag{1}$$

where

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \quad E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{bmatrix}, \quad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{bmatrix}, \tag{2a}$$

$$E_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ f_4 \end{bmatrix}, \quad F_v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ g_4 \end{bmatrix} \tag{2b}$$

and Re is the Reynolds number, ρ is the density, u and v are the Cartesian velocities, and e is the total energy, and where

$$\begin{aligned} \tau_{xx} &= \mu(4u_x - 2v_y)/3 \\ \tau_{xy} &= \mu(u_y + v_x) \\ \tau_{yy} &= \mu(-2u_x + 4v_y)/3 \\ f_4 &= u\tau_{xx} + v\tau_{xy} + \mu Pr^{-1}(\gamma - 1)^{-1}\partial_x a^2 \\ g_4 &= u\tau_{xy} + v\tau_{yy} + \mu Pr^{-1}(\gamma - 1)^{-1}\partial_y a^2 \end{aligned} \tag{2c}$$

Here Pr is the Prandtl number, and a is the speed of sound ($a^2 = \gamma p/\rho$ for ideal fluids).

Pressure is related to the conservative flow variables Q by the equation of state,

$$p = (\gamma - 1)\left[e - \frac{1}{2}\rho(u^2 + v^2)\right] \tag{3}$$

where γ is the ratio of specific heats, generally taken as 1.4, and μ is the dynamic viscosity, which is made up of the molecular viscosity plus a computed turbulent eddy viscosity.

The choice of nondimensional parameters is arbitrary. Here we have chosen to scale the variables as

$$\tilde{\rho} = \frac{\rho}{\rho_\infty}, \quad \tilde{u} = \frac{u}{a_\infty}, \quad \tilde{v} = \frac{v}{a_\infty}, \quad \tilde{e} = \frac{e}{\rho_\infty a_\infty^2} \tag{4a}$$

where ∞ refers to free-stream quantities. Assuming a reference length l (usually taken as some characteristic physical length such as chord of an airfoil), time t scales as $\tilde{t} = ta/l$. The viscous coefficients scale as

$$\tilde{\mu} = \frac{\mu}{\mu_\infty}, \quad Re = \frac{\rho_\infty l a_\infty}{\mu_\infty} \tag{4b}$$

Note that Re uses a_∞ and, therefore, Re based on u_∞ (the usual case for experimentally given Reynolds number) must be scaled by $M_\infty = u_\infty/a_\infty$. For the remainder of this development, the overtilde will be dropped for simplicity.

Coordinate Transformation

For a finite-difference solution on a body-fitted grid, the equations are transformed as described in references 3 and 4. The Navier-Stokes equations written in generalized curvilinear coordinates (ξ, η, t) are

$$\partial_\tau \widehat{Q} + \partial_\xi \widehat{E} + \partial_\eta \widehat{F} = Re^{-1}[\partial_\xi \widehat{E}_v + \partial_\eta \widehat{F}_v] \tag{5}$$

where

$$\widehat{Q} = J^{-1}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ e\end{bmatrix},\ \widehat{E} = J^{-1}\begin{bmatrix}\rho U\\ \rho uU + \xi_x p\\ \rho vU + \xi_y p\\ U(e+p) - \xi_t p\end{bmatrix},\ \widehat{F} = J^{-1}\begin{bmatrix}\rho V\\ \rho uV + \eta_x p\\ \rho vV + \eta_y p\\ V(e+p) - \eta_t p\end{bmatrix} \tag{6}$$

with $J^{-1} = (x_\xi y_\eta - x_\eta y_\xi)$ representing the metric Jacobian and $U = \xi_t + \xi_x u + \xi_y v$ and $V = \eta_t + \eta_x u + \eta_y v$ representing the contravariant velocities. The viscous flux terms are $\widehat{E}_v = J^{-1}(\xi_x E_v + \xi_y F_v)$ and $\widehat{F}_v = J^{-1}(\eta_x E_v + \eta_y F_v)$.

The stress terms, such as τ_{xx}, are also transformed in terms of the ξ- and η-derivatives where

$$\begin{aligned}
\tau_{xx} &= \mu[4(\xi_x u_\xi + \eta_x u_\eta) - 2(\xi_y v_\xi + \eta_y v_\eta)]/3\\
\tau_{xy} &= \mu(\xi_y u_\xi + \eta_y u_\eta + \xi_x v_\xi + \eta_x v_\eta)\\
\tau_{yy} &= \mu[-2(\xi_x u_\xi + \eta_x u_\eta) + 4(\xi_y v_\xi + \eta_y v_\eta)]/3\\
f_4 &= u\tau_{xx} + v\tau_{xy} + \mu Pr^{-1}(\gamma-1)^{-1}(\xi_x \partial_\xi a^2 + \eta_x \partial_\eta a^2)\\
g_4 &= u\tau_{xy} + v\tau_{yy} + \mu Pr^{-1}(\gamma-1)^{-1}(\xi_y \partial_\xi a^2 + \eta_y \partial_\eta a^2)
\end{aligned} \tag{7}$$

Thin-Layer Approximation

In high-Reynolds-number viscous flows, the effects of viscosity are concentrated near rigid boundaries and in wake regions. Typical grids have finer grid spacing in directions nearly normal to the surfaces and coarser grid spacing along the surface. On such grids, the viscous terms associated with derivatives along the body will not be resolved, and in most cases for attached and mildly separated flows these terms are negligible. The terms associated with the normal direction will be resolved for sufficiently fine-grid spacing, and these are substantial terms.

The thin-layer approximation is similar in philosophy to, but not the same as, the boundary-layer theory, for which appropriate scaling arguments show that streamwise components of the viscous terms can be neglected relative to the normal terms. In the thin-layer approximation, the normal momentum equation is solved and pressure can vary through the boundary layer.

Applying the thin-layer approximation to equations (5)-(7), in which all the viscous terms associated with ξ-derivatives are neglected, the following system of equations is obtained:

$$\partial_\tau \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} = Re^{-1} \partial_\eta \hat{S} \tag{8}$$

where

$$\hat{S} = J^{-1} \begin{bmatrix} 0 \\ \eta_x m_1 + \eta_y m_2 \\ \eta_x m_2 + \eta_y m_3 \\ \eta_x(um_1 + vm_2 + m_4) + \eta_y(um_2 + vm_3 + m_5) \end{bmatrix} \tag{9a}$$

with

$$\begin{aligned} m_1 &= \mu(4\eta_x u_\eta - 2\eta_y v_\eta)/3 \\ m_2 &= \mu(\eta_y u_\eta + \eta_x v_\eta) \\ m_3 &= \mu(-2\eta_x u_\eta + 4\eta_y v_\eta)/3 \\ m_4 &= \mu Pr^{-1}(\gamma - 1)^{-1}\eta_x \partial_\eta(a^2) \\ m_5 &= \mu Pr^{-1}(\gamma - 1)^{-1}\eta_y \partial_\eta(a^2) \end{aligned} \tag{9b}$$

Turbulence Model

The algebraic mixing length model of Baldwin and Lomax (ref. 5) is included to approximate the effects of turbulence. It is a two-layer model in which the inner layer is governed by the Prandtl mixing length with Van Driest damping; the outer layer follows the Clauser approximation. Computed vorticity is used in defining the reference mixing length required for the outer layer. In order to improve the numerical compatibility of the Baldwin-Lomax model, two simple modifications are made. First, in the Van Driest damping factor, the shear stress at the wall is replaced with the maximum local laminar shear stress. This generally prevents the numerical difficulties caused by vanishing shear stress at separation. Second, the constant C_{wk}, which is used in the outer-layer viscosity formulation, is changed from 0.25 to 1.0. For some transonic flows, convergence may not be possible if this value is too low, a result of motion of the shock wave.

NUMERICAL ALGORITHM FOR FLOMG RESULTS

The FLOMG computations were performed using a Navier-Stokes code developed by Swanson and Turkel, which is based on the explicit multistage time-stepping schemes of references 1 and 6. This class of schemes is currently in widespread use for solving the Euler equations of gas dynamics. In references 7–10, these schemes were extended to allow the solution of the compressible Navier-Stokes equations. Significant improvements in numerical efficiency were introduced in references 11–13. The basic ideas developed in two dimensions were extended to three dimensions in references 14–16. In the code of Swanson and Turkel, a cell-centered, finite-volume method is employed to obtain difference approximations for the flow equations. Such a method provides flexibility in treating arbitrary

geometries and different grid topologies, since no special treatment is required in the vicinity of singular points or lines. The scheme is second-order accurate in space for sufficiently smooth meshes (see ref. 13 for a definition of "sufficiently smooth"). Artificial dissipation terms with adaptive coefficients are included for stability, convergence, and shock capturing. The spatial and temporal differencing are decoupled. Thus, the numerical scheme is independent of time-step and amenable to steady-state convergence-acceleration techniques. These methods include local time-stepping (a preconditioning for the system of difference equations); variable coefficient, implicit residual smoothing; and multigrid. The latter two methods were designed to be effective on the highly stretched meshes encountered in viscous flow problems; they also work quite well on typical meshes for inviscid flows.

Spatial Discretization

A finite-volume approach is applied to discretize the equations of motion. The computational domain is divided into quadrilateral cells, fixed in time, and for each cell the governing equations are written in integral form as follows,

$$\frac{\partial}{\partial t}\iint_{\Omega} Q\,dx\,dy + \int_{\partial\Omega}(E\,dy - F\,dx) = Re^{-1}\int_{\partial\Omega}(E_v\,dy - F_v\,dx) \tag{10}$$

where Ω is a generic cell and $\partial\Omega$ its boundary.

A cell-centered type of discretization is used, and the line integral of equation (10) is approximated with the midpoint rule. Thus, taking $Q_{i,j}$ as a cell-averaged solution vector, equation (10) can be written in semidiscrete form as

$$\frac{d}{dt}(\Omega_{i,j}Q_{i,j}) + \mathcal{L}Q_{i,j} = 0 \tag{11}$$

where $\mathcal{L}$ is a discrete operator defined by

$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_D - \mathcal{L}_{AD}$$

with the subscripts C, D, and AD referring to convection, diffusion, and artificial dissipation, respectively. The convective fluxes at the cell faces are obtained by an averaging process. Moreover, summing over the cell faces,

$$\mathcal{L}_C Q_{i,j} = \sum_{l=1}^{4} \mathcal{F}_l \cdot \vec{S}_l \tag{12}$$

with the flux tensor, which is associated with convection, given by

$$\mathcal{F}_l = E_l\vec{e}_x + F_l\vec{e}_y,$$

and for each cell face l, the directed area $\vec{S}_l$ is expressed as $\vec{S}_l = (\Delta y)_l\,\vec{e}_x - (\Delta x)_l\,\vec{e}_y$, where the proper signs of $(\Delta x)_l$ and $(\Delta y)_l$ produce an outward normal to the cell face. The augmented convective flux tensor is evaluated as

$$\mathcal{F}_l = \frac{1}{2}(Q^-\vec{V}^- + Q^+\vec{V}^+)_l + P_l \tag{13}$$

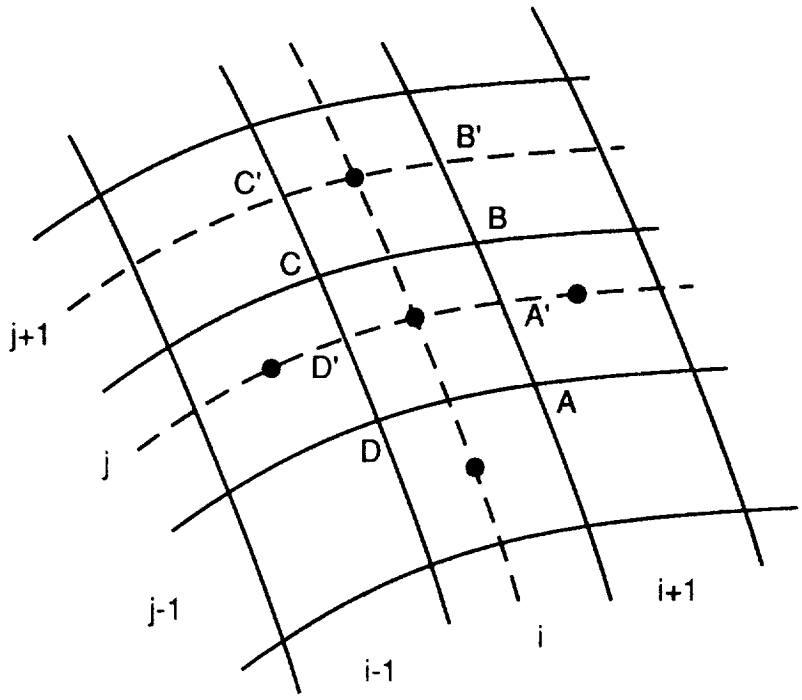


Figure 1. Finite-volume discretization.

$$P_l = \begin{bmatrix} 0 & (p_{avg})_l \vec{e}_x & (p_{avg})_l \vec{e}_y & ((p\vec{V})_{avg})_l \end{bmatrix}^T$$

$$(p_{avg})_l = \frac{1}{2}(p^- + p^+)_l, \qquad ((p\vec{V})_{avg})_l = \frac{1}{2}(p^-\vec{V}^- + p^+\vec{V}^+)_l$$

and the minus and plus superscripts mean that one quantity is determined at the center of the cell of interest and the other at the center of the adjacent cell on edge l, and $\vec{V}$ is the velocity vector. The spatial derivatives necessary to compute the viscous terms are evaluated by means of Green's theorem. For example, consider the arbitrary cell Ω $(ABCD)$ in figure 1. The contributions u_x and u_y to the viscous flux across each cell face (i.e., BC) are approximated with their mean values using

$$\iint_{\Omega'} u_x d\Omega' = \int_{\partial\Omega'} u dy$$

$$\iint_{\Omega'} u_y d\Omega' = -\int_{\partial\Omega'} u dx \qquad (14)$$

where Ω' is an auxiliary cell ($A'B'C'D'$ for face BC in fig. 1). The values of u at B and C required for the cell-boundary integrals in equation (14) are obtained by a simple average of the values of u at the four surrounding cells. In a similar manner, v_x and v_y are computed. Additional details for finite-volume treatment of viscous stresses and heat-conduction terms are given in reference 7, which also gives a thin-layer formulation, and in reference 9.

Artificial Dissipation

For inviscid flows the basic finite-volume scheme just described contains no dissipative terms. In order to prevent odd-even point decoupling and oscillations near shock waves or stagnation points, artificial dissipation terms are added to the governing discrete equations. Moreover, the introduction of appropriate dissipation in the vicinity of shocks allows an entropy condition to be satisfied, and thus guarantees the uniqueness of weak solutions. For viscous flows, dissipative properties are present owing to diffusive terms; however, due to nonlinear effects the physical dissipation may not be sufficient to guarantee stability, especially in the case of the highly stretched meshes generally used to resolve the steep gradients in shear layers. Thus, to maintain the stability and robustness of the numerical procedure, artificial dissipation is also included in viscous regions.

The artificial dissipation model used in the scheme is basically the one developed by Jameson et al. (ref. 1). This nonlinear model is a blending of second and fourth differences. The quantity $\mathcal{L}_{AD}Q_{i,j}$ in equation (11) is defined as

$$\mathcal{L}_{AD}Q_{i,j} = (D_\xi^2 - D_\xi^4 + D_\eta^2 - D_\eta^4)Q_{i,j} \tag{15}$$

where (ξ, η) are arbitrary curvilinear coordinates,

$$D_\xi^2 Q_{i,j} = \nabla_\xi(\lambda_{i+\frac{1}{2},j}\, \varepsilon^{(2)}_{i+\frac{1}{2},j})\Delta_\xi Q_{i,j} \tag{16}$$

$$D_\xi^4 Q_{i,j} = \nabla_\xi(\lambda_{i+\frac{1}{2},j}\, \varepsilon^{(4)}_{i+\frac{1}{2},j})\Delta_\xi \nabla_\xi \Delta_\xi Q_{i,j} \tag{17}$$

and where i, j are indices (for a cell center) associated with the ξ- and η-directions, and Δ_ξ, ∇_ξ are forward and backward difference operators in the ξ-direction. Following references 11 and 12, the variable scaling factor is defined as

$$\lambda_{i+\frac{1}{2},j} = \frac{1}{2}[(\bar{\lambda}_\xi)_{i,j} + (\bar{\lambda}_\xi)_{i+1,j}] \tag{18}$$

where

$$\begin{aligned} (\bar{\lambda}_\xi)_{i,j} &= \phi_{i,j}(r)\ (\lambda_\xi)_{i,j} \\ \phi_{i,j}(r) &= 1 + r^\zeta_{i,j} \end{aligned} \tag{19}$$

and where r is the ratio λ_η/λ_ξ, λ_ξ and λ_η are the scaled spectral radii of the flux Jacobian matrices (associated with the ξ- and η-directions) for the Euler equations, and the exponent ζ is generally taken to be 2/3. The spectral radii for the ξ- and η-directions are given by

$$\lambda_\xi = |uy_\eta - vx_\eta| + a\sqrt{y_\eta^2 + x_\eta^2} \tag{20}$$

$$\lambda_\eta = |vx_\xi - uy_\xi| + a\sqrt{x_\xi^2 + y_\xi^2} \tag{21}$$

and a is the speed of sound. The coefficients $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ use the pressure as a sensor for shocks and stagnation points, and they are defined as

$$\varepsilon^{(2)}_{i+\frac{1}{2},j} = \kappa^{(2)} \max(\nu_{i-1,j}, \nu_{i,j}, \nu_{i+1,j}, \nu_{i+2,j}) \tag{22}$$

$$\nu_{i,j} = \left| \frac{p_{i-1,j} - 2p_{i,j} + p_{i+1,j}}{p_{i-1,j} + 2p_{i,j} + p_{i+1,j}} \right| \tag{23}$$

$$\varepsilon^{(4)}_{i+\frac{1}{2},j} = \max[0, (\kappa^{(4)} - \varepsilon^{(2)}_{i+\frac{1}{2},j})] \tag{24}$$

where typical values for the constants $\kappa^{(2)}$ and $\kappa^{(4)}$ are in the ranges 1/4 to 1/2 and 1/64 to 1/32, respectively. For the normal direction (η), the dissipation contributions are defined in a similar way, except

$$(\bar{\lambda}_\eta)_{i,j} = \phi_{i,j}(r^{-1})\ (\lambda_\eta)_{i,j} \tag{25}$$

The treatment of the artificial dissipation must be modified at the boundaries of the physical domain. In the case of the fourth-difference dissipation, the standard five-point difference stencil must be replaced at the first two interior mesh cells. This means that one-sided or one-sided biased stencils are used at these cells. The dissipative character of the artificial terms is important because it influences both stability and accuracy. For example, if the dissipation is too large at a solid boundary, an artificial boundary layer is created in an inviscid flow, and the effective Reynolds number for a viscous flow is altered. To improve accuracy at the wall boundaries of viscous flows, where gradients are steep because of physical boundary layers, the usual fourth-difference stencils are changed in this dissipation model.

Let the total dissipation for a mesh cell, in the direction represented by the index j, be denoted by d_j. For simplicity, assume that $\lambda\varepsilon^{(4)} = 1$. Then,

$$d_j = df_{j+\frac{1}{2}} - df_{j-\frac{1}{2}} \tag{26}$$

where the dissipative flux

$$df_{j+\frac{1}{2}} = (\Delta Q)_{j+\frac{3}{2}} - 2(\Delta Q)_{j+\frac{1}{2}} + (\Delta Q)_{j-\frac{1}{2}} \tag{27}$$

and thus

$$d_j = (\Delta Q)_{j+\frac{3}{2}} - 3(\Delta Q)_{j+\frac{1}{2}} + 3(\Delta Q)_{j-\frac{1}{2}} - (\Delta Q)_{j-\frac{3}{2}} \tag{28}$$

with the index i for ΔQ suppressed for convenience. Consider the first two interior cells adjacent to a solid boundary, as depicted in figure 2. If

$$(\Delta Q)_{\frac{1}{2}} = (\Delta Q)_{\frac{3}{2}} = (\Delta Q)_{\frac{5}{2}} \tag{29}$$

then equation (28) gives

$$d_2 = Q_4 - 2Q_3 + Q_2 \tag{30}$$

$$d_3 = Q_5 - 4Q_4 + 5Q_3 - 2Q_2 \tag{31}$$

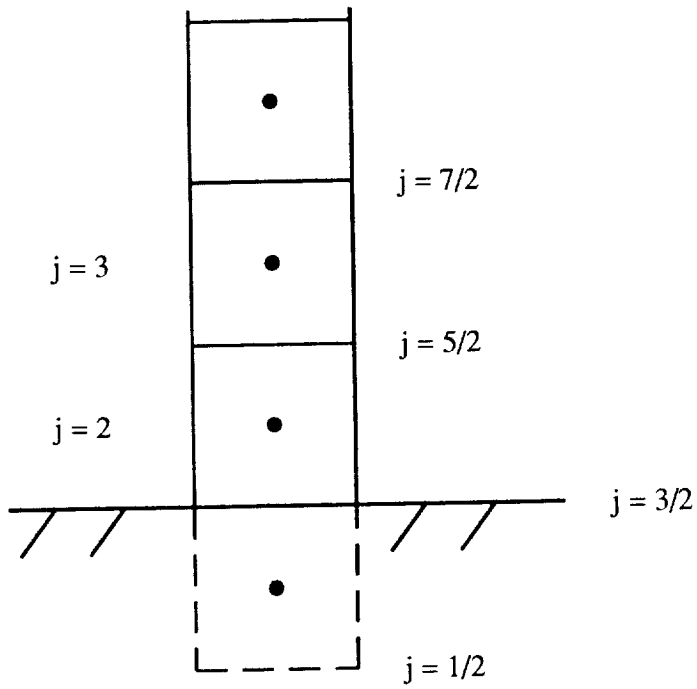


Figure 2. Boundary point dissipation.

These boundary stencils are fairly standard ones, and they result in a nonpositive definite dissipation matrix for the system of difference equations (ref. 17). An alternative form, which has reduced the sensitivity to solid surface normal mesh spacing for turbulent flow calculations without compromising stability or convergence, is given by

$$(\Delta Q)_{\frac{1}{2}} = 2(\Delta Q)_{\frac{3}{2}} - (\Delta Q)_{\frac{5}{2}} \tag{32}$$

and

$$d_2 = Q_4 - 3Q_3 + 3Q_2 - Q_1 \tag{33}$$

$$d_3 = Q_5 - 4Q_4 + 6Q_3 - 4Q_2 + Q_1 \tag{34}$$

Time-Stepping Scheme

The system of ordinary differential equations of equation (11) is advanced in time toward the steady-state solution with a five-stage Runge–Kutta scheme. This scheme is second-order accurate in time. At the $(m + 1)st$ stage,

$$Q_{i,j}^{(m+1)} = Q_{i,j}^{(0)} - \alpha_{m+1} \frac{\Delta t_{i,j}}{\Omega_{i,j}} [\mathcal{L}_C Q_{i,j}^{(m)} + \mathcal{L}_D Q_{i,j}^{(0)} - \sum_{n=0}^{m} \gamma_{mn} \mathcal{L}_{AD} Q_{i,j}^{(n)}] \tag{35}$$

where $Q_{i,j}^{(0)} = Q_{i,j}^{n}$ is the discrete solution at time-level n; $Q_{i,j}^{n+1} = Q_{i,j}^{(5)}$ is the solution at the new time-level, $n + 1$; α_{m+1} are the coefficients of the scheme; Δt is the time-step; Ω is the area of the

mesh cell; and γ_{mn} are the weighting factors of the artificial dissipation. The coefficients α_{m+1} are determined such that the scheme has the largest possible hyperbolic stability limit. An appropriate set of coefficients is given by

$$\alpha_1 = \frac{1}{4}, \quad \alpha_2 = \frac{1}{6}, \quad \alpha_3 = \frac{3}{8}, \quad \alpha_4 = \frac{1}{2}, \quad \alpha_5 = 1 \tag{36}$$

This scheme also exhibits good high-frequency damping behavior, which is crucial for a rapidly convergent multigrid method (to be described in a subsequent section). In order to establish a good parabolic stability, the artificial dissipation terms are evaluated on the first, third, and fifth stages. The weighting factors γ_{mn} must satisfy the condition

$$\sum \gamma_{mn} = 1 \tag{37}$$

They are defined as follows:

$$\begin{aligned}
&\gamma_{00} = 1, \\
&\gamma_{10} = 1, \quad \gamma_{11} = 0, \\
&\gamma_{20} = (1 - \overline{\gamma}_3), \quad \gamma_{21} = 0, \quad \gamma_{22} = \overline{\gamma}_3, \\
&\gamma_{30} = (1 - \overline{\gamma}_3), \quad \gamma_{31} = 0, \quad \gamma_{32} = \overline{\gamma}_3, \quad \gamma_{33} = 0, \\
&\gamma_{40} = (1 - \overline{\gamma}_3)(1 - \overline{\gamma}_5), \quad \gamma_{41} = 0, \quad \gamma_{42} = \overline{\gamma}_3(1 - \overline{\gamma}_5), \quad \gamma_{43} = 0, \quad \gamma_{44} = \overline{\gamma}_5
\end{aligned} \tag{38}$$

where $\overline{\gamma}_3 = 0.56$, and $\overline{\gamma}_5 = 0.44$. As indicated in equation (35), the physical viscous terms are computed only on the first stage and are frozen for the remaining stages. The single evaluation appears to have no significant effect on the stability of the scheme and allows a reduction in computational effort.

Convergence Acceleration Techniques

Three methods are employed to accelerate convergence of the basic explicit time-stepping scheme. These techniques are: (1) local time-stepping; (2) residual smoothing; and (3) multigrid. With local time-stepping, the solution at each mesh point is advanced at the maximum Δt allowed by stability. Both convection and diffusion limits are included in the Δt computation (see ref. 18). Implicit smoothing of the residuals is used to extend the stability range of the basic time-stepping scheme. For two-dimensional problems, the residual smoothing can be applied in the form

$$(1 - \beta_\xi \nabla_\xi \Delta_\xi)(1 - \beta_\eta \nabla_\eta \Delta_\eta) \bar{\mathcal{R}}_{i,j}^{(m)} = \mathcal{R}_{i,j}^{(m)} \tag{39}$$

where the residual $\mathcal{R}_{i,j}^{(m)}$ is defined by

$$\mathcal{R}_{i,j}^{(m)} = \alpha_m \frac{\Delta t_{i,j}}{\Omega_{i,j}} (\mathcal{L}_C Q_{i,j}^{(m-1)} + \mathcal{L}_D Q_{i,j}^{(0)} - AD^{(m)}) \quad , \quad m = 1,5 \tag{40}$$

and is computed in the Runge-Kutta stage m, $AD^{(m)}$ is the total artificial dissipation at stage m, and $\bar{\mathcal{R}}_{i,j}^{(m)}$ is the final residual at stage m after the sequence of smoothings in the ξ- and η-directions. The

coefficients β_ξ and β_η are variable and are functions of the spectral radii λ_ξ and λ_η. Based on the unpublished work of Swanson, they can be written as follows:

$$\beta_\xi = \max\left\{\frac{1}{4}\left[\left(\frac{N}{N^*}\,\frac{1}{1+\psi\,r_{\eta\xi}}\right)^2 - 1\right], 0\right\}$$

$$\beta_\eta = \max\left\{\frac{1}{4}\left[\left(\frac{N}{N^*}\,\frac{1}{1+\psi\,r_{\eta\xi}^{-1}}\right)^2 - 1\right], 0\right\} \tag{41}$$

where the ratio $r_{\eta\xi} = \lambda_\eta/\lambda_\xi$, and the quantity N/N^* is the ratio of the Courant-Friedrichs-Lewy number of the smoothed scheme to that of the basic explicit scheme (usually having a value of 2). From a linear stability analysis, the scheme with these coefficients is stable for all mesh-cell aspect ratios when the parameter $\psi \approx 0.125$ and N/N^* is sufficiently large. The practical limitation on the Courant number is due to the requirement for effective high-frequency damping. For large N/N^*, the high-frequency damping of the scheme vanishes. An alternative form for these coefficients was introduced by Martinelli and Jameson (ref. 12). Similar performance of the two forms has been observed for typical viscous flow meshes. However, significant improvement in convergence rate has been obtained with standard inviscid meshes using the form of equation (41).

The multigrid method used with the Runge-Kutta time-stepping algorithm is based on the work of Jameson (ref. 6). The basic idea of the method is to use coarser grids to speed up the propagation of the fine-grid corrections. Coarser meshes are obtained by eliminating every other mesh line in each coordinate direction. On the auxiliary meshes, the solution is initialized as

$$Q_{2h}^{(0)} = \frac{\sum \Omega_h Q_h}{\Omega_{2h}} \tag{42}$$

where the subscript refers to the mesh-spacing value, the sum is over the four fine-grid cells that compose the $2h$ grid cell, and, again, Ω is a cell volume. This rule conserves mass, momentum, and energy. On a coarse grid, a forcing function P is added to the governing discrete equations in order to impose the fine-grid approximation. After the initialization of the coarse-grid solution, this function is computed as follows:

$$P_{2h} = \sum R_h(Q_h) - R_{2h}(Q_{2h}^{(0)}) \tag{43}$$

where $R_h(Q_h) = \mathcal{L}_h Q_h$. Then, the time-stepping scheme on the $(m+1)st$ stage becomes

$$Q_{2h}^{(m+1)} = Q_{2h}^{(0)} - \alpha_{m+1}\frac{\Delta t}{\Omega}[R_{2h}(Q_{2h}^{(m)}) + P_{2h}^{(0)}] \tag{44}$$

We can also define a new value R^* for the residual as

$$R_{2h}^* = R_{2h}(Q_{2h}) + P_{2h} \tag{45}$$

This value can be collected, the solution Q_{2h} restricted to the next coarser grid, and the process repeated. The corrections computed on a coarse grid are transferred back to a finer grid with bilinear

interpolation. A fixed W-type cycle is used to execute the multigrid strategy (see ref. 18 for a diagram of a cycle). In order to make this strategy effective for a wide range of flow conditions, the resultant coarse-grid corrections are smoothed before they are passed to the finest mesh. The factored scheme of equation (39) with constant coefficients ($\beta_\xi = \beta_\eta = 0.2$) is used for this smoothing. Also, the application of a full multigrid (FMG) method provides a well-conditioned starting solution for the finest mesh being considered. Additional details of the multigrid procedure are discussed in reference 18.

Discrete Boundary Conditions

At a solid boundary, a row of auxiliary cells is created exterior to the domain of the flow. The no-slip condition is imposed by treating the Cartesian velocity components as antisymmetric functions with respect to the solid surface. Thus,

$$u_{i,1} = -u_{i,2}$$
$$v_{i,1} = -v_{i,2} \tag{46}$$

where the indices $(i, 1)$ and $(i, 2)$ refer to the centers of the auxiliary and the first interior cells, respectively. The surface values of pressure p and temperature T are computed using the reduced normal momentum and energy equations,

$$\frac{\partial p}{\partial \eta} = 0 \quad , \quad \frac{\partial T}{\partial \eta} = 0 \tag{47}$$

where η is the coordinate normal to the surface.

At the outer boundary of "C" meshes, the boundary points are treated in essentially the same way as described in the discussion of boundary conditions in the next section. Simple extrapolation is used to compute the flow quantities at the downstream boundary of the C-type mesh topology.

NUMERICAL ALGORITHM FOR ARC2D RESULTS

The ARC2D computations were performed using a code based on the implicit approximate factorization algorithm of Beam and Warming (ref. 2). The code was originally developed by Steger (ref. 19) and has been steadily improved and modified (refs. 20–22). The algorithm is an implicit approximate-factorization finite-difference scheme which is first-order accurate in time. Local time-linearizations are applied to the nonlinear terms and an approximate factorization of the two-dimensional implicit operator is used to produce one-dimensional operators. The spatial derivative terms are approximated with second-order central differences. A diagonal form of the algorithm that produces a computationally efficient modification of the standard algorithm in which the diagonalization results in scalar pentadiagonal operators in place of the block operators is also used. Explicit and implicit artificial dissipation terms are added to achieve nonlinear stability. Spatially variable time-steps and mesh sequencing are used to accelerate convergence.

Implicit Time-Differencing and Local Time-Linearization

Applying the first-order Euler implicit time-differencing scheme to equation (8) results in

$$\widehat{Q}^{n+1} - \widehat{Q}^{n} + h\left(\widehat{E}_{\xi}^{n+1} + \widehat{F}_{\eta}^{n+1} - Re^{-1}\widehat{S}_{\eta}^{n+1}\right) = 0 \tag{48}$$

with $h = \Delta t$. The nonlinear terms are linearized in time about $\widehat{Q}^{n}$ by a Taylor series such that

$$\begin{aligned}
\widehat{E}^{n+1} &= \widehat{E}^{n} + \widehat{A}^{n}\Delta\widehat{Q}^{n} + O(h^2) \\
\widehat{F}^{n+1} &= \widehat{F}^{n} + \widehat{B}^{n}\Delta\widehat{Q}^{n} + O(h^2) \\
Re^{-1}\widehat{S}^{n+1} &= Re^{-1}\left[\widehat{S}^{n} + J^{-1}\widehat{M}^{n}\Delta\widehat{Q}^{n})\right] + O(h^2)
\end{aligned} \tag{49}$$

where $\widehat{A} = \partial\widehat{E}/\partial\widehat{Q}$, $\widehat{B} = \partial\widehat{F}/\partial\widehat{Q}$, and $\widehat{M} = \partial\widehat{S}/\partial\widehat{Q}$ are the flux Jacobians and $\Delta\widehat{Q}^{n}$ is $O(h)$. The linearizations are second-order accurate.

The Jacobian matrices are $\widehat{A}$ or $\widehat{B}$ =

$$\begin{bmatrix}
\kappa_t & \kappa_x & \kappa_y & 0 \\
-u\theta + \kappa_x\phi^2 & \kappa_t + \theta - (\gamma-2)\kappa_x u & \kappa_y u - (\gamma-1)\kappa_x v & (\gamma-1)\kappa_x \\
-v\theta + \kappa_y\phi^2 & \kappa_x v - (\gamma-1)\kappa_y u & \kappa_t + \theta - (\gamma-2)\kappa_y v & (\gamma-1)\kappa_y \\
\theta[\phi^2 - a_1] & \kappa_x a_1 - (\gamma-1)u\theta & \kappa_y a_1 - (\gamma-1)v\theta & \gamma\theta + \kappa_t
\end{bmatrix} \tag{50}$$

with $a_1 = \gamma(e/\rho) - \phi^2$, $\theta = \kappa_x u + \kappa_y v$, $\phi^2 = \frac{1}{2}(\gamma-1)(u^2+v^2)$, and $\kappa = \xi$ or η for $\widehat{A}$ or $\widehat{B}$, respectively.

The viscous flux Jacobian is

$$\widehat{M} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
m_{21} & \alpha_1\partial_\eta(\rho^{-1}) & \alpha_2\partial_\eta(\rho^{-1}) & 0 \\
m_{31} & \alpha_2\partial_\eta(\rho^{-1}) & \alpha_3\partial_\eta(\rho^{-1}) & 0 \\
m_{41} & m_{42} & m_{43} & m_{44}
\end{bmatrix} J \tag{51a}$$

where

$$\begin{aligned}
m_{21} &= -\alpha_1\partial_\eta(u/\rho) - \alpha_2\partial_\eta(v/\rho) \\
m_{31} &= -\alpha_2\partial_\eta(u/\rho) - \alpha_3\partial_\eta(v/\rho) \\
m_{41} &= \alpha_4\partial_\eta\left[-(e/\rho^2) + (u^2+v^2)/\rho\right] \\
&\quad - \alpha_1\partial_\eta(u^2/\rho) - 2\alpha_2\partial_\eta(uv/\rho) \\
&\quad - \alpha_3\partial_\eta(v^2/\rho) \\
m_{42} &= -\alpha_4\partial_\eta(u/\rho) - m_{21} \\
m_{43} &= -\alpha_4\partial_\eta(v/\rho) - m_{31} \\
m_{44} &= \alpha_4\partial_\eta(\rho^{-1}) \\
\alpha_1 &= \mu[(4/3)\eta_x^2 + \eta_y^2], \quad \alpha_2 = (\mu/3)\eta_x\eta_y \\
\alpha_3 &= \mu[\eta_x^2 + (4/3)\eta_y^2], \quad \alpha_4 = \gamma\mu Pr^{-1}(\eta_x^2 + \eta_y^2)
\end{aligned} \tag{51b}$$

Applying equations (49) to equation (48) and combining the $\Delta\widehat{Q}^n$ terms produces the unfactored block algorithm in delta form,

$$\begin{aligned}\left[I + h\partial_\xi \widehat{A}^n + h\partial_\eta \widehat{B}^n - Re^{-1} hJ^{-1}\partial_\eta \widehat{M}\right]\Delta\widehat{Q}^n = \\ -h\left(\partial_\xi \widehat{E}^n + \partial_\eta \widehat{F}^n - Re^{-1}\partial_\eta \widehat{S}^n\right)\end{aligned} \tag{52}$$

Space Differencing

The next step is to take the continuous differential operators ∂_ξ and ∂_η and approximate them with finite-difference operators on a discrete mesh. Introducing a grid of mesh points (j, k), variables are defined at mesh points as $u_{j,k} = u(j\Delta\xi, k\Delta\eta)$. The grid spacing in the computational domain is chosen to be unity so that $\Delta\xi,\ \Delta\eta = 1$. Second-order central-difference operators are used, where, for example,

$$\delta_\xi u_{j,k} = \left(u_{j+1,k} - u_{j-1,k}\right)/2 \quad \text{and} \quad \delta_\eta u_{j,k} = \left(u_{j,k+1} - u_{j,k-1}\right)/2 \tag{53a}$$

For the viscous derivatives, the terms take the form

$$\partial_\eta\left(\alpha_{j,k}\partial_\eta \beta_{j,k}\right) \tag{53b}$$

which is differenced in the compact three-point form as

$$\left[\left(\alpha_{j,k+1} + \alpha_{j,k}\right)\left(\beta_{j,k+1} - \beta_{j,k}\right) - \left(\alpha_{j,k} + \alpha_{j,k-1}\right)\left(\beta_{j,k} - \beta_{j,k-1}\right)\right]/2 \tag{53c}$$

Approximate Factorization

The implicit (left-hand) side of equation (52) can be written as

$$\begin{aligned}\left(I + h\delta_\xi \widehat{A}^n + h\delta_\eta \widehat{B}^n - hRe^{-1}\delta_\eta J^{-1}\widehat{M}^n\right)\Delta\widehat{Q}^n = \\ \left(I + h\delta_\xi \widehat{A}^n\right)\left(I + h\delta_\eta \widehat{B}^n - hRe^{-1}\delta_\eta J^{-1}\widehat{M}^n\right)\Delta\widehat{Q}^n \\ -h^2\delta_\xi \widehat{A}^n\delta_\eta \widehat{B}^n\Delta\widehat{Q}^n + h^2 Re^{-1}\delta_\xi \widehat{A}^n\delta_\eta J^{-1}\widehat{M}^n\Delta\widehat{Q}^n\end{aligned} \tag{54}$$

The cross term is second-order accurate, since $\Delta\widehat{Q}^n$ is $O(h)$. It can therefore be neglected without degrading the time-accuracy of any first- or second-order scheme.

The resulting factored form of the algorithm is

$$\begin{aligned}\left(I + h\delta_\xi \widehat{A}^n\right)\left(I + h\delta_\eta \widehat{B}^n - hRe^{-1}\delta_\eta J^{-1}\widehat{M}^n\right)\Delta\widehat{Q}^n = \\ -h\left(\delta_\xi \widehat{E}^n + \delta_\eta \widehat{F}^n - Re^{-1}\delta_\eta \widehat{S}^n\right)\end{aligned} \tag{55}$$

with two implicit operators, each of which is block tridiagonal. The solution algorithm now consists of two one-dimensional sweeps, one in the ξ- and one in the η-direction. Each sweep requires the solution of a linear block-tridiagonal system by block LU (lower-upper) decomposition.

Diagonal Form of Implicit Algorithm

One way to improve the efficiency of this method is to reduce the operation count by introducing a diagonalization of the blocks in the implicit operators, as developed by Pulliam and Chaussee (ref. 23). The eigensystems of the flux Jacobians $\widehat{A}$ and $\widehat{B}$ are used in this construction. Initially, the discussion is restricted to the Euler equations; the application to Navier-Stokes is considered later.

Each of the flux Jacobians $\widehat{A}$ and $\widehat{B}$ has real eigenvalues and a complete set of eigenvectors. Therefore, the Jacobian matrices can be diagonalized (see ref. 24)

$$\Lambda_\xi = T_\xi^{-1} \widehat{A} T_\xi \quad \text{and} \quad \Lambda_\eta = T_\eta^{-1} \widehat{B} T_\eta$$

with T_ξ the matrix whose columns are the eigenvectors of $\widehat{A}$ and T_η the corresponding eigenvector matrix for $\widehat{B}$.

Replacing $\widehat{A}$ and $\widehat{B}$ in equation (52) with their eigensystem decompositions produces

$$\left[T_\xi T_\xi^{-1} + h\,\delta_\xi \left(T_\xi \Lambda_\xi T_\xi^{-1}\right)\right] \left[T_\eta T_\eta^{-1} + h\,\delta_\eta \left(T_\eta \Lambda_\eta T_\eta^{-1}\right)\right] \Delta Q^n$$

$$= \quad \text{the explicit right-hand side of equation (52)} = \widehat{R}^n \tag{56}$$

At this point, equation (56) and the inviscid form of equation (55) are exactly equivalent. A modified form of equation (56) can be obtained by factoring the T_ξ and T_η eigenvector matrices outside the spatial derivative terms δ_ξ and δ_η. The eigenvector matrices are functions of ξ and η and, therefore, this modification reduces the time-accuracy to at most first-order. The resulting equations are

$$T_\xi \left(I + h\,\delta_\xi \Lambda_\xi\right) \widehat{N} \left(I + h\,\delta_\eta \Lambda_\eta\right) T_\eta^{-1} \Delta \widehat{Q}^n = \widehat{R}^n \tag{57}$$

where $\widehat{N} = T_\xi^{-1} T_\eta$.

The diagonal algorithm as presented in equation (57) is only rigorously valid for the Euler equations. The implicit linearization of the viscous flux $\widehat{S}^n$ in the implicit operator for the η-direction was neglected, because the viscous flux Jacobian $\widehat{M}^n$ is not simultaneously diagonalizable with the inviscid flux Jacobian $\widehat{B}^n$. For stability and robustness, it is desirable to include the viscous term on the implicit side, but retaining the block operator in the η-direction requires more computation time. Adding an approximate viscous eigenvalue,

$$\lambda_v(\eta) = \rho \mu Re^{-1} J^{-1} (\eta_x^2 + \eta_y^2) \tag{58}$$

to the implicit operator has proved to be an appropriate compromise between accuracy and speed.

The explicit side of the diagonal algorithm is exactly the same as in the original algorithm, equation (52). The modifications are restricted to the implicit side and so, if the diagonal algorithm converges, the steady-state solution will be identical to one obtained with the unmodified algorithm. The diagonal algorithm reduces the block-tridiagonal inversion to 4×4 matrix multiplies and scalar tridiagonal inversions. The overall savings in computational work can be as high as 40%.

Nonlinear Artificial Dissipation Model

A mixed second- and fourth-difference dissipation model with appropriate coefficients produces a scheme with good shock-capturing capabilities. Jameson et al. have employed a dissipative model of such a form in which second- and fourth-difference dissipation are combined (ref. 1). The model, written in our notation, is

$$\nabla_\xi \left(\psi_{j+1,k} J^{-1}_{j+1,k} + \psi_{j,k} J^{-1}_{j,k} \right) \left(\epsilon^{(2)}_{j,k} \Delta_\xi Q_{j,k} - \epsilon^{(4)}_{j,k} \Delta_\xi \nabla_\xi \Delta_\xi Q_{j,k} \right) \tag{59}$$

with

$$\begin{aligned} \epsilon^{(2)}_{j,k} &= \kappa_2 \Delta t \max(\mu_{j+1,k}, \mu_{j-1,k}) \\ \mu_{j,k} &= \frac{|p_{j+1,k} - 2p_{j,k} + p_{j-1,k}|}{|p_{j+1,k} + 2p_{j,k} + p_{j-1,k}|} \\ \epsilon^{(4)}_{j,k} &= \max(0, \kappa_4 \Delta t - \epsilon^{(2)}_{j,k}) \end{aligned} \tag{60}$$

where for these calculations values of the constants are $\kappa_2 = 1$ and $\kappa_4 = 1/100$. Similar terms are used in the η-direction. The term $\psi_{j,k}$ is a spectral radius scaling factor and is defined as

$$\begin{aligned} \psi_{j,k} &= \psi_x + \psi_y \\ \psi_x &= |U| + a\sqrt{\xi_x^2 + \xi_y^2} \\ \psi_y &= |V| + a\sqrt{\eta_x^2 + \eta_y^2} \end{aligned} \tag{61}$$

which is the sum of the spectral radii of $\widehat{A}$ and $\widehat{B}$.

The first term of equation (58) is a second-difference dissipation term with an extra pressure gradient coefficient to increase its value near shocks. The second term is a fourth-difference term where the logic to compute $\epsilon^{(4)}_{j,k}$ switches it off when the second-difference nonlinear coefficient is larger then the constant fourth-difference coefficient. This occurs near a shock. This model is added to the right-hand explicit side of the algorithm.

The implicit dissipation used with equation (58) is the linearization of the model, treating the pressure coefficient μ and the spectral radius ψ as space-varying functions but ignoring their functional dependency on Q. Then the dissipation is linear in $Q_{j,k}$ and is added to the diagonal algorithm, necessitating scalar pentadiagonal solvers. This produces the most efficient, stable, and convergent form of the implicit algorithm.

Near computational boundaries, the fourth-difference dissipation is modified so as to eliminate the five-point stencil. A one-side biased second-difference term is used instead.

Convergence Acceleration

In solving these steady-state problems, two techniques were used to accelerate the algorithm, the goal being to eliminate the transient as quickly as possible. Note that for the delta form of the algorithm (either factored or unfactored) the steady-state solution is independent of the time-step, h. Therefore, the time-step path to the steady-state does not affect the final solution and we can use time-step sequences or spatially variable time-steps to accelerate convergence. The particular form of spatially variable time-step used here is to replace h with

$$\Delta t_0 \left(\frac{1}{1 + \sqrt{J}} \right) \tag{62}$$

with Δt_0 chosen to be $O(1)$. The use of the local time-step can improve convergence time by a factor of 2 to 3.

Another way to accelerate convergence to a steady state is to obtain a good initial guess for a fine mesh by first iterating on a sequence of coarse grids and then interpolating the solution up to the next refined grid. A coarsened grid is cut from each previous grid by halving the number of points in the ξ-direction and by regenerating a new η-distribution using fewer points. The η-distribution is regenerated because in viscous flows the spacing near the wall would be too coarse if the halving procedure were used. A small number of iterations (a few hundred) are carried out on each coarsened grid, then the approximate solution is interpolated onto a more refined grid. The finest grid is then iterated to convergence. This can reduce the practical convergence time by as much as a factor of 2.

Boundary Conditions

At a rigid-body surface, the no-slip condition is satisfied with the Cartesian velocities u, v set to zero. The pressure is obtained through extrapolation such that $\partial P / \partial \eta = 0$. A zeroth-order extrapolation is used to compute density. At subsonic free-stream conditions, the outer boundary condition is based on locally one-dimensional Riemann invariants. At a boundary, local normal and tangential velocity components are computed. The normal component is the scaled contravariant velocity,

$$V_n = \frac{\eta_x u + \eta_y v}{\sqrt{(\eta_x^2 + \eta_y^2)}} \tag{63}$$

and the tangential component is

$$V_t = \frac{\eta_y u - \eta_x v}{\sqrt{(\eta_x^2 + \eta_y^2)}} \tag{64}$$

The locally one-dimensional Riemann invariants are

$$R_1 = V_n - 2a/(\gamma - 1) \quad \text{and} \quad R_2 = V_n + 2a/(\gamma - 1) \tag{65}$$

Two other equations are needed so that four unknowns (the four flow variables) can be calculated; V_t and a quantity $S = p/\rho^\gamma$, which has the same functional dependence as entropy, are used. For an inflow

boundary, $V_n < 0$ and then three quantities can be specified. The Riemann invariant R_1, V_t and S are all set to free-stream values. The other variable R_2 is extrapolated from the interior flow variables. On outflow, $V_n > 0$, only R_1 is fixed to free stream and R_2, V_t, and S are extrapolated. Once these four variables are available at the boundary the four flow variables Q can be obtained.

For these computations, the wake cut in the "C" meshes is handled implicitly. The array storage is shuffled so that integration in the η-direction proceeds directly from one outer boundary to the other in the wake region, and from the body surface outward to the far-field boundary in the rest of the grid.

RESULTS

Computational results for the following five viscous transonic airfoil cases (from ref. 25) were compared:

1. (A1) NACA 0012 at $M_\infty = 0.7$, $\alpha_{exp} = 1.86°$, $\alpha_{corr} = 1.49°$, $Re = 9 \times 10^6$, transition at 0.05 chords.

2. (A2) NACA 0012 at $M_\infty = 0.55$, $\alpha_{exp} = 9.86°$, $\alpha_{corr} = 8.34°$, $Re = 9 \times 10^6$, transition at 0.05 chords.

3. (A4) NACA 0012 at $M_\infty = 0.7$, $\alpha = 3.0°$, $Re = 9 \times 10^6$, transition at 0.05 chords.

4. (6) RAE 2822 at $M_{\infty exp} = 0.725$, $\alpha_{exp} = 2.92°$, $M_{\infty corr} = 0.729$, $\alpha_{corr} = 2.31°$, $Re = 6.5 \times 10^6$, transition at 0.03 chords.

5. (10) RAE 2822 at $M_{\infty exp} = 0.750$, $\alpha_{exp} = 3.19°$, $M_{\infty corr} = 0.754$, $\alpha_{corr} = 2.57°$, $Re = 6.2 \times 10^6$, transition at 0.03 chords.

The angle-of-attack corrections for the NACA 0012 cases were obtained from reference 26. The corrections for the RAE 2822 cases were obtained from Hall (private communication from M. G. Hall, Royal Aircraft Establishment, Farnborough, UK, 1988). In those cases, the airfoil coordinates were measured coordinates with a camber correction, rather than the usual design coordinates, as proposed by Hall. Experimental data for cases A1 and A2 are from reference 26, and those for cases 6 and 10 are from reference 27.

Each of the two codes—ARC2D and FLOMG—were used to compute each of the five cases on three different grids generated by Maksymiuk, Swanson, and Hall. All grids were large C-meshes with leading-edge and trailing-edge clustering and very fine spacing in the normal direction near the body surface. The two grids (one for the NACA 0012 and one for the RAE 2822) designated "Maksymiuk" were generated with a hyperbolic solver (ref. 28). These grids are 385×65 points, with 321 points on the body (streamwise) and 32 points in the wake. The only clustering of points is at the leading and trailing edges, where the spacing is 0.001. Normal spacing at the body is 0.00001. The outer boundary is at 25 chords. The grids designated "Swanson" are also 385×65 points, with 257 points on the body and 64 in the wake. The leading- and trailing-edge spacing is 0.001 on the NACA airfoil and 0.002

on the RAE airfoil. Normal spacing at the body is 0.000004 on the NACA airfoil and 0.00001 on the RAE. The distance to the outer boundary is about 20 chords. The "Hall" grids are slightly less resolved at 321 $\times$ 65 points, with 257 points on the body and 32 points in the wake. The outer boundary is at 8 chords. The NACA 0012 grid has a normal spacing of 0.0000127 at the wall and clustering at the leading and trailing edges, where the spacing is 0.00066 and 0.00481, respectively. The RAE 2822 grid has a normal spacing of 0.0000555, a leading-edge spacing of 0.00064, and a trailing-edge spacing of 0.00457.

All of the figures showing computational results are found at the end of the paper. Figures 3–7 show the pressure coefficients on the airfoil surfaces; results from each grid are shown separately. For case A1 (fig. 3), the computations show a weak shock near the leading edge. Despite the apparently larger amount of second-difference dissipation, the ARC2D gives a slightly sharper shock than the FLOMG. This behavior of the two codes is a consequence of the several differences in the numerical dissipation employed. In particular, there are differences in the form of the eigenvalue scaling, the magnitude of the constant coefficient for the fourth-difference term, and implementation of the switching operator. In case A2 (fig. 4), there is a strong shock at about 20% chord. On the highly resolved Maksymiuk grid (fig. 4(a)), the ARC2D nearly captures the lambda shock structure; however, the main part of the shock is slightly farther aft than the experiment indicates. The FLOMG exhibits a simple strong shock in this region. On the coarser grids (figs. 4(b) and 4(c)), both codes compute simple shocks, with the ARC2D locating it slightly farther aft on both grids. The same behavior is observed for case A4 (fig. 5). On two of the grids (figs. 5(a) and 5(c)), the FLOMG results show a small overcompression at the shock. Results for both cases 6 and 10 are similar. On the finest grid (figs. 6(a) and 7(a)), there is again a slight difference in shock location. On the coarser grids (figs. 6(b), 6(c), 7(b), and 7(c)), the two codes show slight disagreement along the upper surface, as well as at the shock. The large error in shock location relative to experiment in case 10 has been attributed to a shortcoming in the turbulence model for cases with large separation zones (ref. 25).

Figures 8–12 show skin-friction computations for each case. In the plots for case A1 (fig. 8), the ARC2D results show a bump at 20% chord, corresponding to the sharper shock produced there. The FLOMG results for cases A1, A2, and A4 on the Hall grid (figs. 8(c), 9(c), and 10(c)) have a sharp increase at the trailing edge. Figure 11 illustrates how important grid resolution is to the proper calculation of skin friction. Experimental results are closely matched by both codes on the Maksymiuk and Swanson grids (figs. 11(a) and 11(b)); however, on the Hall grid (fig. 11(c)), both codes underpredict the friction coefficient. The Hall grid does not have adequate grid resolution in the normal direction at the body surface. Experimental results are less successfully matched in case 10 (fig. 12), but both codes still compute somewhat lower friction on the Hall grid.

Selected boundary-layer profiles from cases 6 and 10 are shown in figures 13–16. Case 6 results are in figures 13 and 14. Both codes are in good agreement with each other and with experiment. Some of the dissimilarities might reflect difficulties in pinpointing the edge of the boundary layer. The results for case 10 (figs. 15 and 16) do not match those of the experiment, since the turbulence model precludes computing the post-shock separation accurately.

Displacement thickness is plotted in figures 17 and 18. The results of FLOMG for case 6 are quite good, but those of the ARC2D underpredict the experiment downstream of the shock. This could be a result of the greater dissipation used in its computations.

Table 1 displays lift, drag, and convergence data, as well as CPU time in seconds needed to achieve a given percentage of the converged value of lift. It is evident that FLOMG, which can get a solution to 0.1% of the converged value of the lift coefficient in just over a minute for most cases, is much faster than ARC2D, which takes 5–11 min. For computing steady-state solutions, the extra work of an implicit method is not needed. However, the explicit scheme is significantly less efficient for unsteady flows than for steady ones, a result of the inapplicability of local time-stepping and the low effectiveness of the current form of the multigrid method. Improvements are being investigated. Some of ARC2D's convergence times are unusually high because the runs were not optimized to converge quickly. Also, implicit wake integration operates more efficiently on the Swanson meshes, which have twice as many points in the wake as the Maksymiuk meshes, so there are more long vectors. Consequently, convergence times are generally shorter on the Swanson grids than on the Ames grids. It is interesting that FLOMG requires about the same amount of time in all cases, whereas ARC2D is quite variable (especially for case 10, which was difficult to converge because of slight motions of the shock on the finely spaced grids); moreover, ARC2D usually requires proportionately more time for the final stage of convergence (from 1% of C_l to 0.1% of C_l) than FLOMG.

Some additional results obtained with FLOMG are presented in the appendix. They indicate the influence of artificial dissipation on several of the computations.

CONCLUDING REMARKS

ARC2D and FLOMG produce essentially the same results. Pressure coefficients are nearly identical, except in some cases near shocks; some of these differences are attributable to different dissipation coefficients. Their agreement with experimental data is excellent, except for shock location in flows with large separation zones. Skin-friction coefficients, boundary-layer profiles, and displacement-thickness results are very similar. Satisfactory agreement with experimental data is observed for grids with sufficiently fine spacing near the body.

Lift coefficients calculated by the two codes are generally within 1% of each other, and drag coefficients generally fall within 5% of each other. For attached cases, both codes predict lift within 5% of the experimental value, and drag within 10%. This accuracy is not achieved for flows with separation, where deficiencies in the turbulence model are apparent.

For the steady-state cases computed here, the explicit multigrid method demonstrated superior convergence characteristics, requiring only about 1 min of CPU time on a Cray 2, compared with 5–10 min for the implicit method.

APPENDIX

THE EFFECT OF DISSIPATION COEFFICIENTS ON FLOMG RESULTS

In FLOMG computations, the constants for the second-difference and fourth-difference artificial dissipation terms were selected because they had worked quite well for a wide variety of two-dimensional external and internal flow problems. However, at least for the airfoil cases considered herein, which include the difficult RAE airfoil case 10, the dissipation constants can be reduced by a factor of 2 without causing serious deterioration in convergence behavior. Since differences occurred in the vicinity of shock waves between some of the results obtained with FLOMG and ARC2D, several computations were repeated to determine the effect of the smaller dissipation factors.

Very small changes occurred in the sharpness of the shocks for the cases considered. As seen in table 2, the maximum increase in lift was less than 3%. Except for case 10, in which the total drag increased by 5 counts, the drag increased by 2 counts or less. The computer times required to reach the different levels of lift-coefficient convergence are essentially the same as those of the higher dissipation constants. This again demonstrates the robustness of the scheme. Pressure coefficients for the three cases shown in table 2 are presented in figures 19–21; skin-friction coefficients are presented in figures 22–24.

REFERENCES

1. Jameson, A.; Schmidt, W.; and Turkel, E.: Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes. AIAA Paper 81–1259, Palo Alto, Calif., 1981.

2. Beam, R.; and Warming, R. F.: An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form. J. Comp. Phys., vol. 22, 1976, pp. 87–110.

3. Viviand, H.: Conservative Forms of Gas Dynamics Equations. La Recherche Aerospatiale, 1974, p. 65.

4. Vinokur, M.: Conservation Equations of Gas-Dynamics in Curvilinear Coordinate Systems. J. Comp. Phys., vol. 14, 1974, pp. 105-125.

5. Baldwin, B. S.; and Lomax, H.: Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows. AIAA Paper 78–257, 1978.

6. Jameson, A.: Multigrid Algorithms for Compressible Flow Calculations. Mechanical and Aeronautical Engineering Report 1743, Princeton U., Princeton, N.J., 1985.

7. Swanson, R. C.; and Turkel, E.: A Multistage Time-Stepping Scheme for the Navier-Stokes Equations. AIAA Paper 85–0035, 1985.

8. Agarwal, R. K.; and Deese, J. E.: Computation of Transonic Viscous Airfoil, Inlet, and Wing Flowfields. AIAA Paper 84–1551, 1984.

9. Martinelli, L.; Jameson, A.; and Grasso, F.: A Multigrid Method for the Navier-Stokes Equations. AIAA Paper 86–0208, 1986.

10. Vatsa, V. N.: Accurate Solutions for Transonic Viscous Flow Over Finite Wings. AIAA Paper 86–1052, 1986.

11. Swanson, R. C.; and Turkel, E.: Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations. AIAA Paper 87–1107, Honolulu, Hawaii, 1987.

12. Martinelli, L.; and Jameson, A.: Validation of a Multigrid Method for the Reynolds-Averaged Equations. AIAA Paper 88–0414, 1988.

13. Radespiel, R.; and Swanson, R. C.: An Investigation of Cell-Centered and Cell-Vertex Multigrid Schemes for the Navier-Stokes Equations. AIAA Paper 89–0548, 1989.

14. Jayaram, M.; and Jameson, A.: Multigrid Solution of the Navier-Stokes Equations for Flows Over Wings. AIAA Paper 88–0705, 1988.

15. Radespiel, R.; Rossow, C.; and Swanson, R. C.: An Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations. AIAA Paper 89–1953, Buffalo, N.Y., 1989.

16. Vatsa, V. N.; and Wedan, B. W.: Development of an Efficient Multigrid Code for 3-D Navier-Stokes Equations. AIAA Paper 89–1791, 1989.

17. Pulliam, T. H.: Artificial Dissipation for the Euler Equations. AIAA Paper 85–0438, 1985.

18. Arnone, A.; and Swanson, R. C.: A Navier-Stokes Solver for Cascade Flows. NASA CR-181682, 1988.

19. Steger, J. L.: Implicit Finite Difference Simulation of Flow about Arbitrary Geometries with Application to Airfoils. AIAA Paper 77–665, 1977.

20. Pulliam, T. H.; Jespersen, D. C.; and Childs, R. E.: An Enhanced Version of an Implicit Code for the Euler Equations. AIAA Paper 83-0344, Reno, Nev., 1983.

21. Pulliam, T. H.; and Steger, J. L.: Recent Improvements in Efficiency, Accuracy, and Convergence for Implicit Approximate Factorization Algorithms. AIAA Paper 85-0360, Reno, Nev., 1985.

22. Pulliam, T. H.: Efficient Solution Methods for The Navier-Stokes Equations. Lecture Notes for the von Kármán Institute For Fluid Dynamics Lecture Series: Numerical Techniques for Viscous Flow Computation In Turbomachinery Bladings, von Kármán Institute, Rhode-St-Genese, Belgium, 1985.

23. Pulliam, T. H.; and Chaussee, D. S.: A Diagonal Form of an Implicit Approximate Factorization Algorithm. J. Comp. Phys., vol. 39, 1981, p. 347.

24. Warming, R. F.; Beam, R.; and Hyett, B. J.: Diagonalization and Simultaneous Symmetrization of the Gas-Dynamic Matrices. Math. of Comp., vol. 29, 1975, p. 1037.

25. Holst, T. L.: Viscous Transonic Airfoil Workshop Compendium of Results. AIAA Paper 87–1460, Honolulu, Hawaii, 1987.

26. Harris, C. D.: Two-Dimensional Aerodynamic Characteristics of the NACA 0012 Airfoil in the Langley 8-Foot Transonic Pressure Tunnel. NASA TM-81927, 1981.

27. Cook, P.; McDonald, M.; and Firmin, M.: Aerofoil RAE 2822 – Pressure Distributions, and Boundary Layer and Wake Measurements. AGARD–AR–138, 1979.

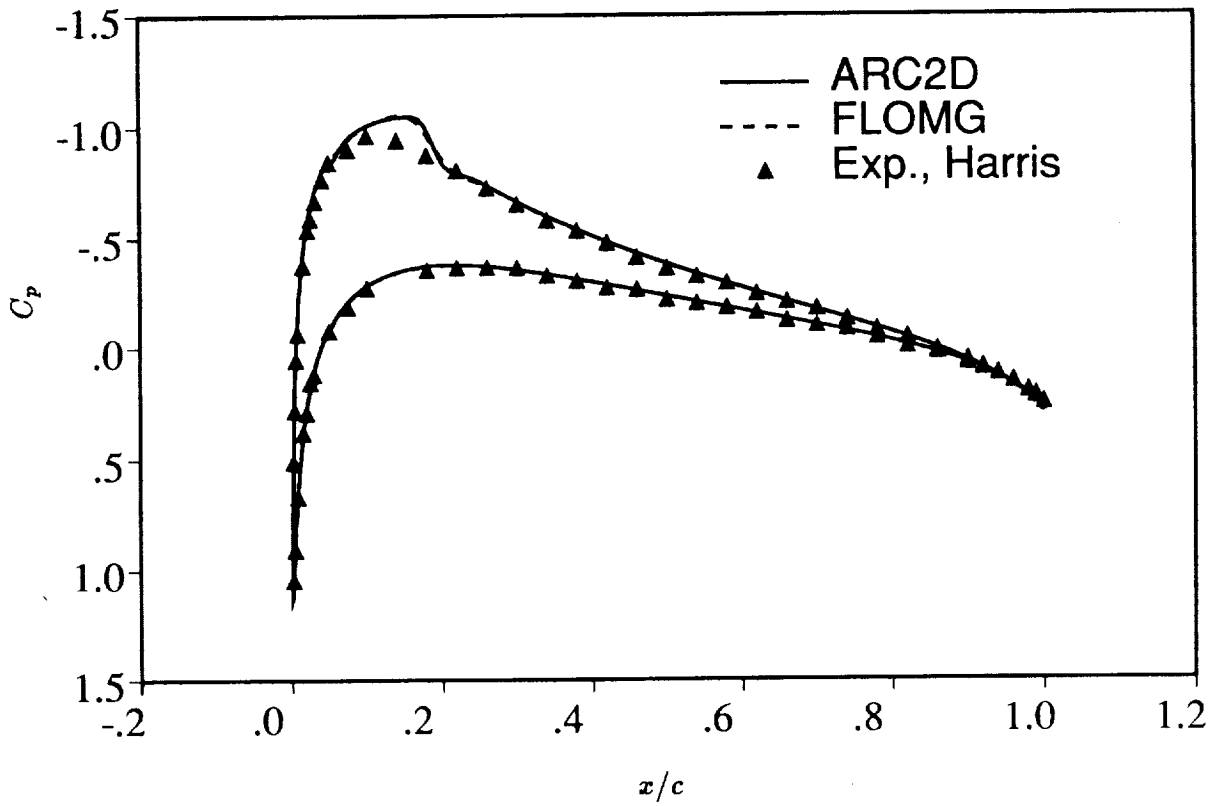
28. Barth, T. J.; Pulliam, T. H.; and Buning, P. G.: Navier-Stokes Computations for Exotic Airfoils. AIAA Paper 85–0171, 1985.

Table 1. Lift coefficients, drag coefficients, and convergence times.

Case	grid	code	C_l	C_d	C_{d_p}	C_{d_f}	CPU time required to reach designated value of C_l convergence, sec.			
							5%	1%	0.25%	0.1%
A1	Maksymiuk	ARC2D	0.253	0.0082	0.0025	0.0057	92	396	572	678
		FLOMG	0.252	0.0081	0.0024	0.0056		52		72
	Swanson	ARC2D	0.256	0.0088	0.0023	0.0065	25	250	350	415
		FLOMG	0.253	0.0081	0.0024	0.0057		52		60
	Hall	ARC2D	0.257	0.0073	0.0022	0.0051	214	380	493	556
		FLOMG	0.258	0.0077	0.0023	0.0052		39		51
	experiment		0.241	0.0079						
A2	Maksymiuk	ARC2D	0.978	0.0357	0.0319	0.0038	59	254	335	356
		FLOMG	0.984	0.0352	0.0314	0.0038		53		81
	Swanson	ARC2D	0.994	0.0368	0.0321	0.0047	56	260	310	450
		FLOMG	0.981	0.0353	0.0316	0.0037		59		78
	Hall	ARC2D	0.991	0.0360	0.0326	0.0035	35	250	350	384
		FLOMG	0.949	0.0341	0.0308	0.0033		62		79
	experiment		0.983	0.0253						
A4	Maksymiuk	ARC2D	0.503	0.0143	0.0089	0.0054	54	369	529	661
		FLOMG	0.496	0.0138	0.0085	0.0052		52		72
	Swanson	ARC2D	0.508	0.0149	0.0089	0.0060	23	230	317	385
		FLOMG	0.501	0.0139	0.0086	0.0053		54		60
	Hall	ARC2D	0.509	0.0135	0.0088	0.0047	46	112	293	318
		FLOMG	0.493	0.0132	0.0083	0.0049		51		56
6	Maksymiuk	ARC2D	0.778	0.0143	0.0084	0.0059	101	406	572	695
		FLOMG	0.772	0.0136	0.0080	0.0056		52		70
	Swanson	ARC2D	0.774	0.0141	0.0081	0.0060	108	401	621	745
		FLOMG	0.783	0.0138	0.0082	0.0056		53		70
	Hall	ARC2D	0.773	0.0132	0.0082	0.0050	86	282	388	502
		FLOMG	0.782	0.0134	0.0081	0.0054		45		71
	experiment		0.743	0.0127						
10	Maksymiuk	ARC2D	0.805	0.0292	0.0236	0.0056	66	1147	1478	1511
		FLOMG	0.811	0.0289	0.0235	0.0054		52		72
	Swanson	ARC2D	0.800	0.0288	0.0231	0.0057	59	325	379	935
		FLOMG	0.814	0.0287	0.0233	0.0054		59		99
	Hall	ARC2D	0.804	0.0279	0.0231	0.0048	47	114	126	474
		FLOMG	0.807	0.0276	0.0224	0.0052		69		83
	experiment		0.743	0.0242						

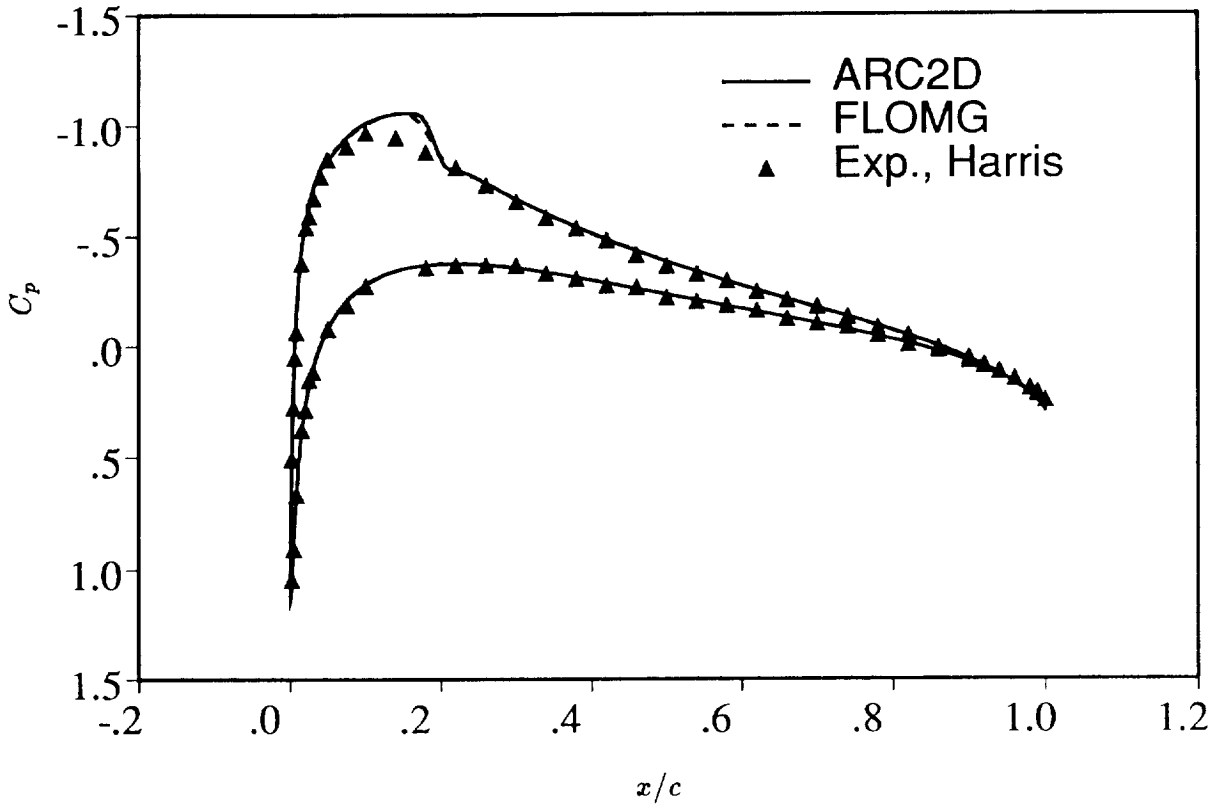
Table 2. Lift coefficients, drag coefficients, and convergence times for FLOMG results with reduced artificial dissipation.

Case	grid	C_l	C_d	C_{d_p}	C_{d_f}	CPU time required to reach designated value of C_l convergence, sec.	
						1%	0.1%
A2	Maksymiuk	0.990	0.0352	0.0312	0.0039	53	81
	Hall	0.963	0.0340	0.0305	0.0034	62	79
6	Maksymiuk	0.779	0.0137	0.0080	0.0056	54	70
	Hall	0.794	0.0136	0.0082	0.0054	45	74
10	Maksymiuk	0.820	0.0294	0.0240	0.0054	52	72



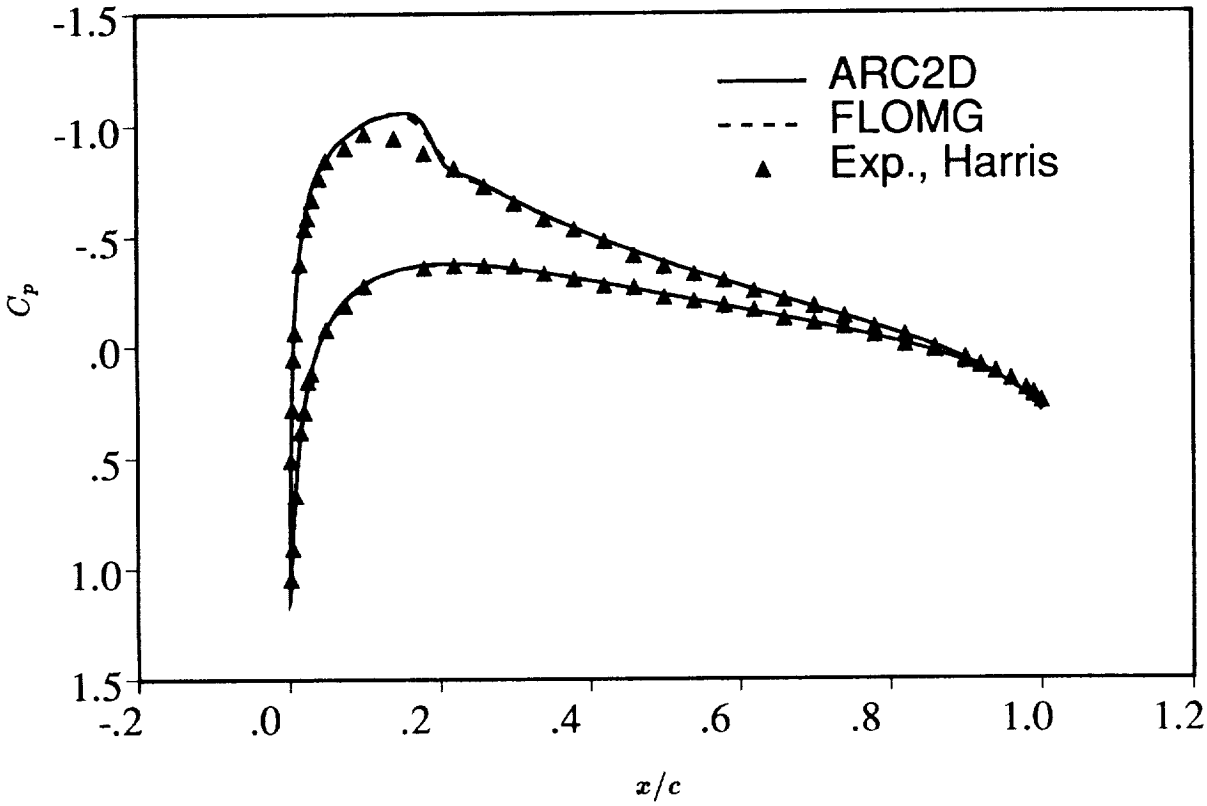
(a) Maksymiuk grid.

Figure 3.–Pressure coefficient for NACA case A1: $M_\infty = 0.7$, $\alpha_{exp} = 1.86°$, $\alpha_{corr} = 1.49°$.



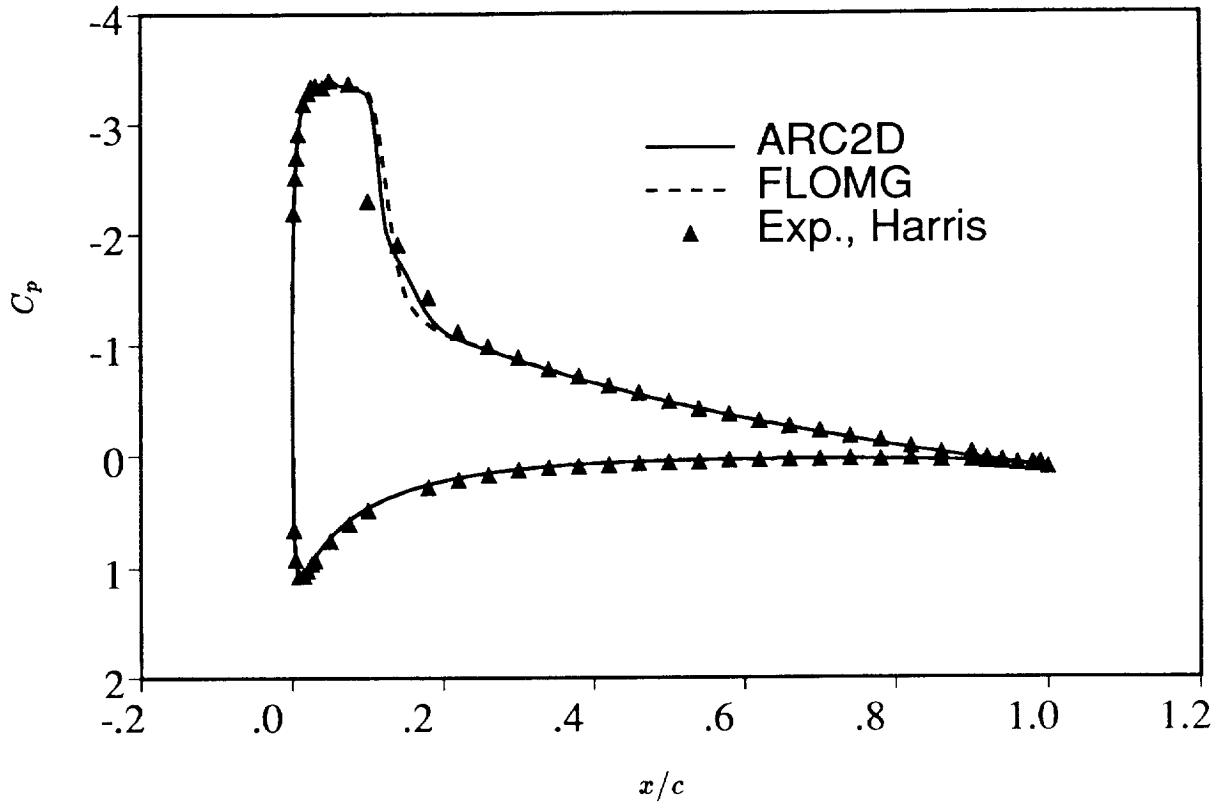
(b) Swanson grid.

Figure 3.–Continued.



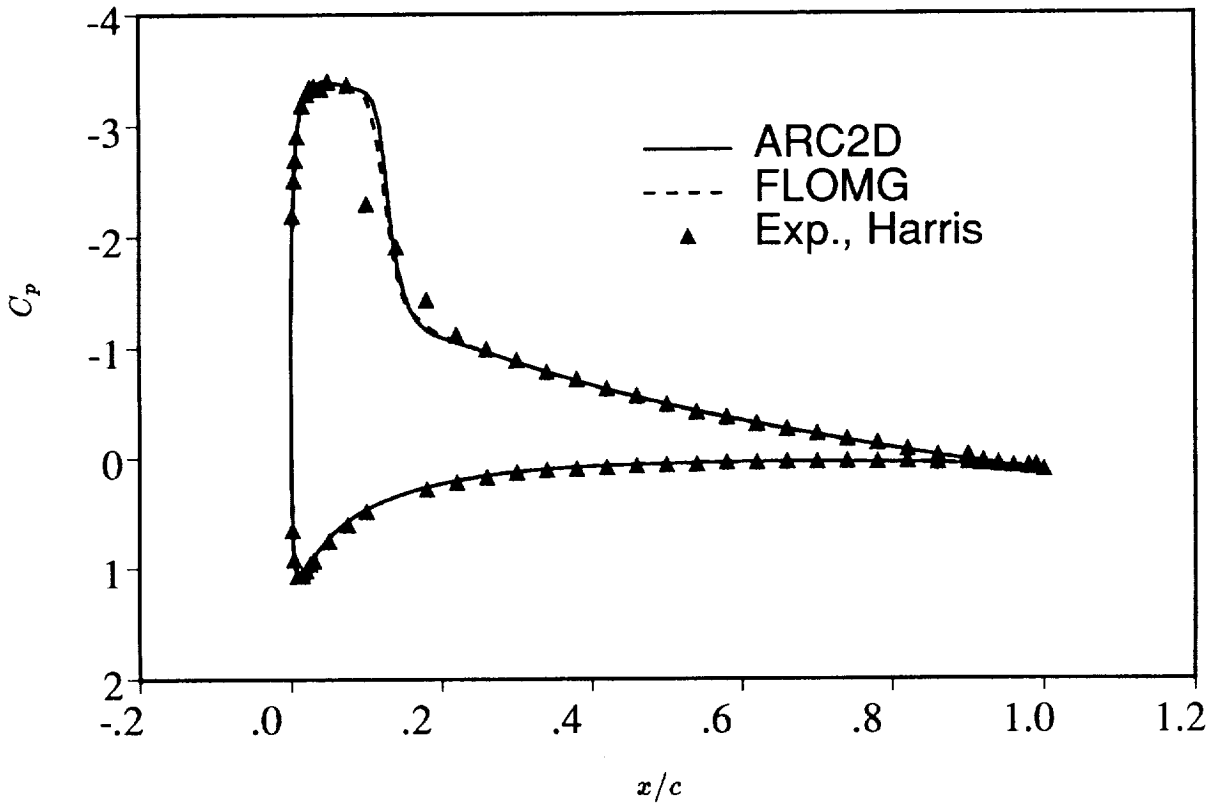
(c) Hall grid.

Figure 3.–Concluded.



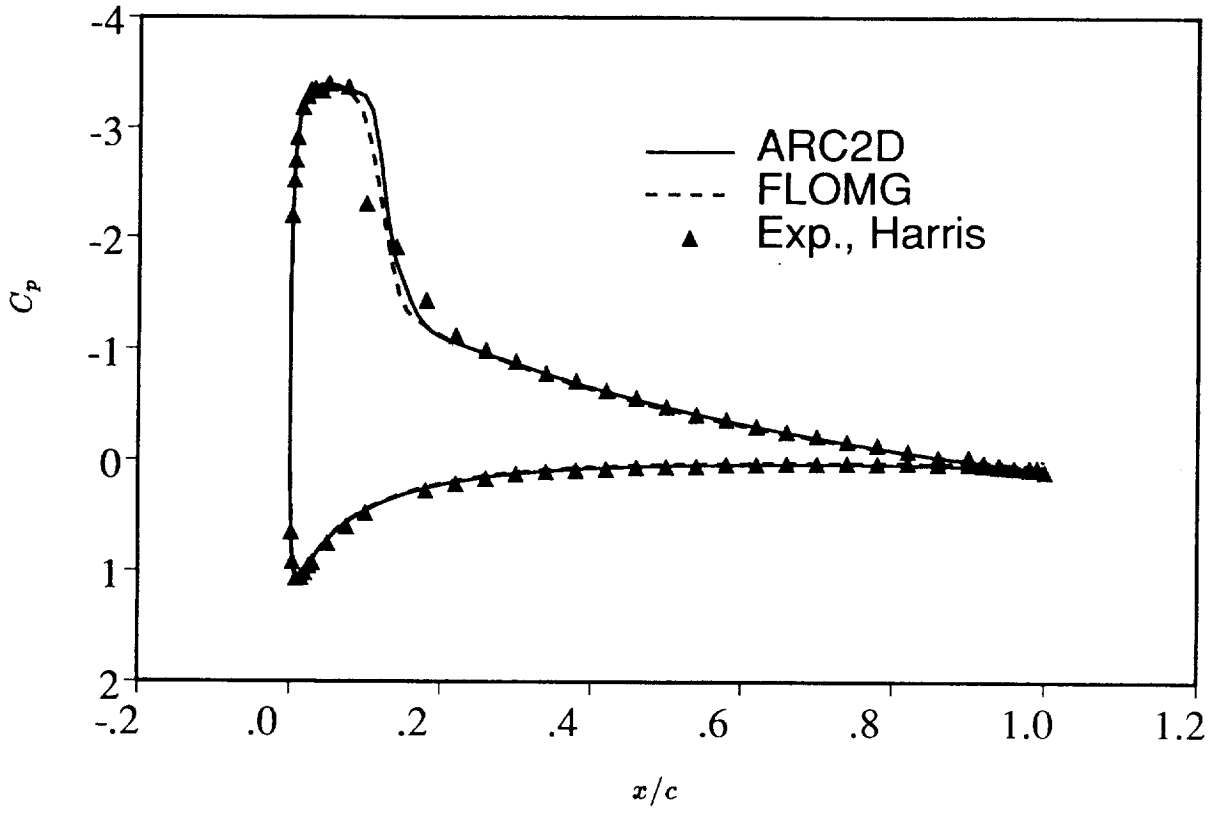
(a) Maksymiuk grid.

Figure 4.–Pressure coefficient for NACA case A2: $M_\infty = 0.55$, $\alpha_{exp} = 9.86°$, $\alpha_{corr} = 8.34°$.



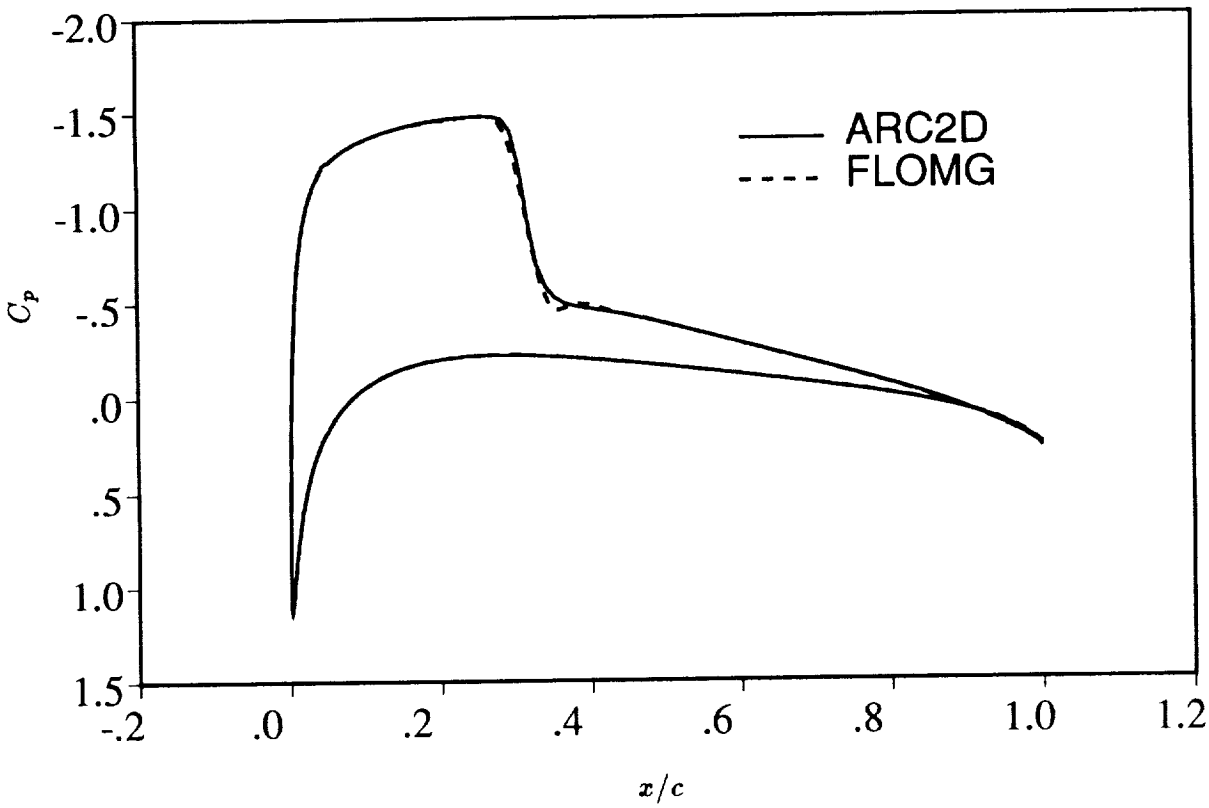
(b) Swanson grid.

Figure 4.–Continued.



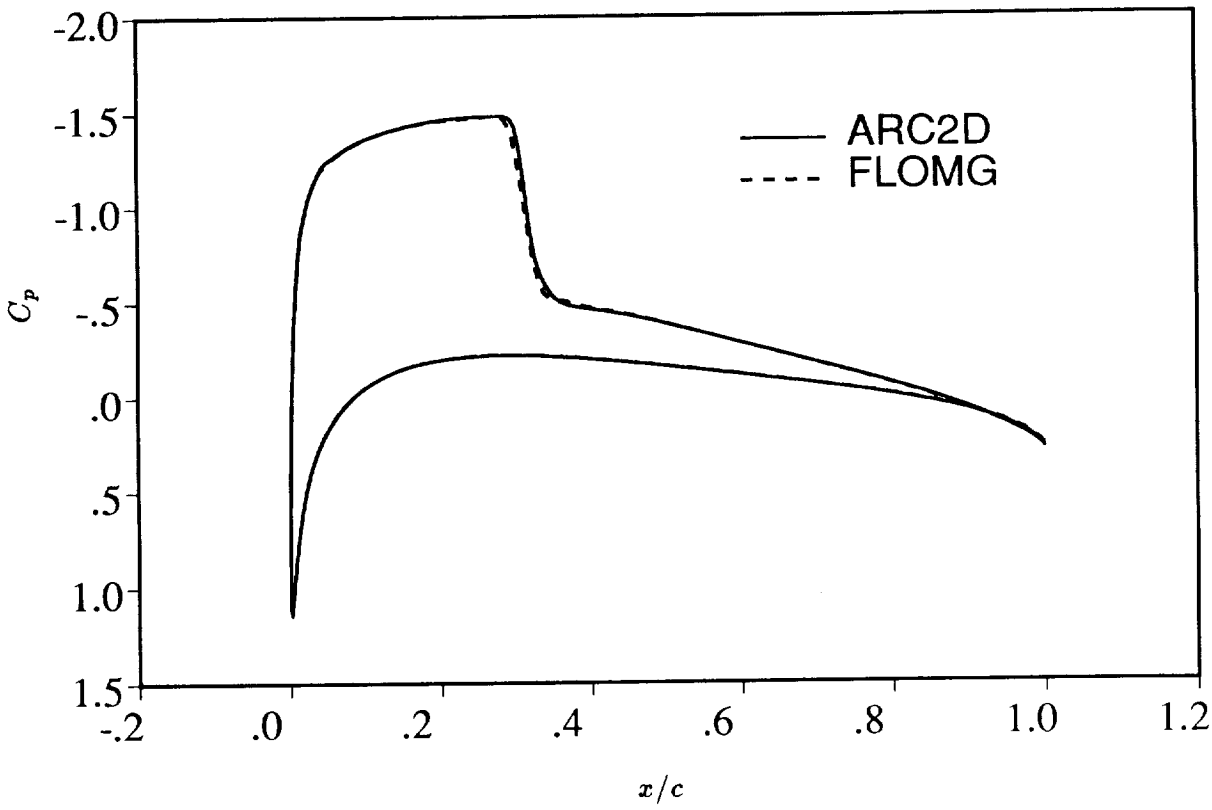
(c) Hall grid.

Figure 4.–Concluded.



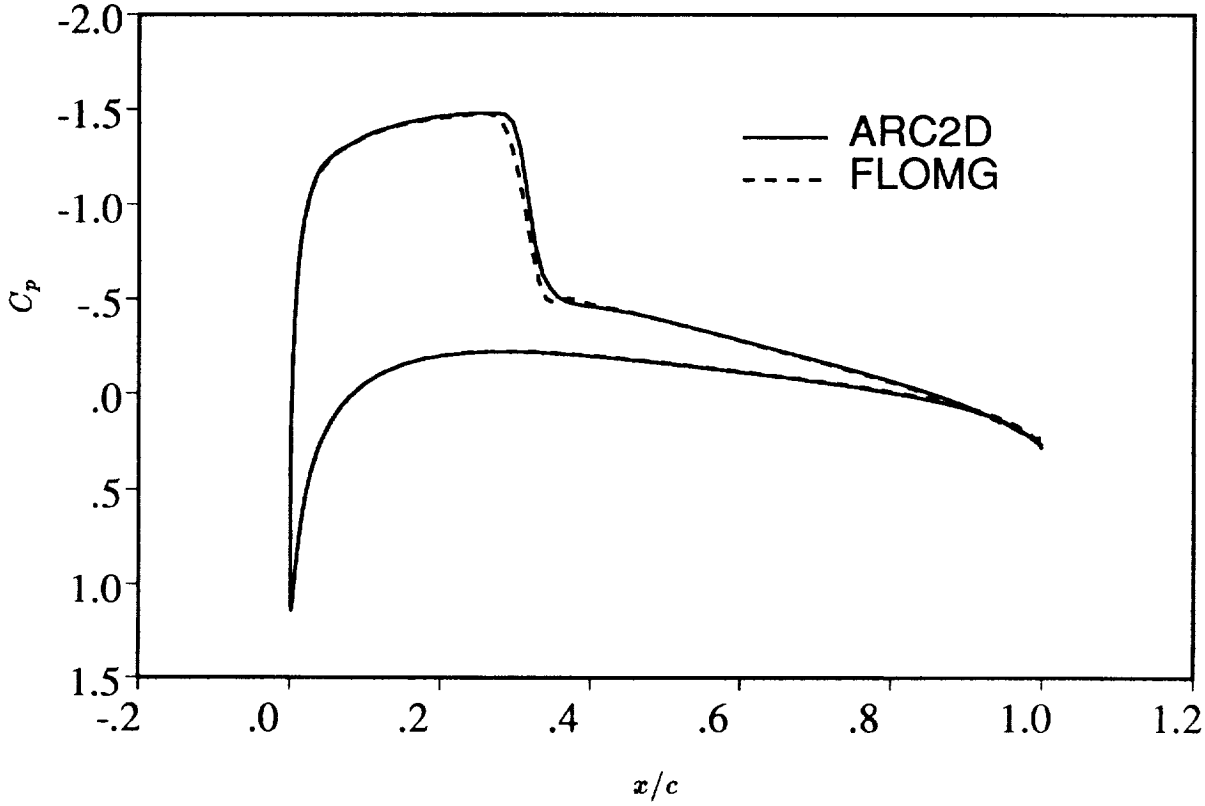
(a) Maksymiuk grid.

Figure 5.–Pressure coefficient for NACA Case A4: $M_\infty = 0.7$, $\alpha = 3.0°$.



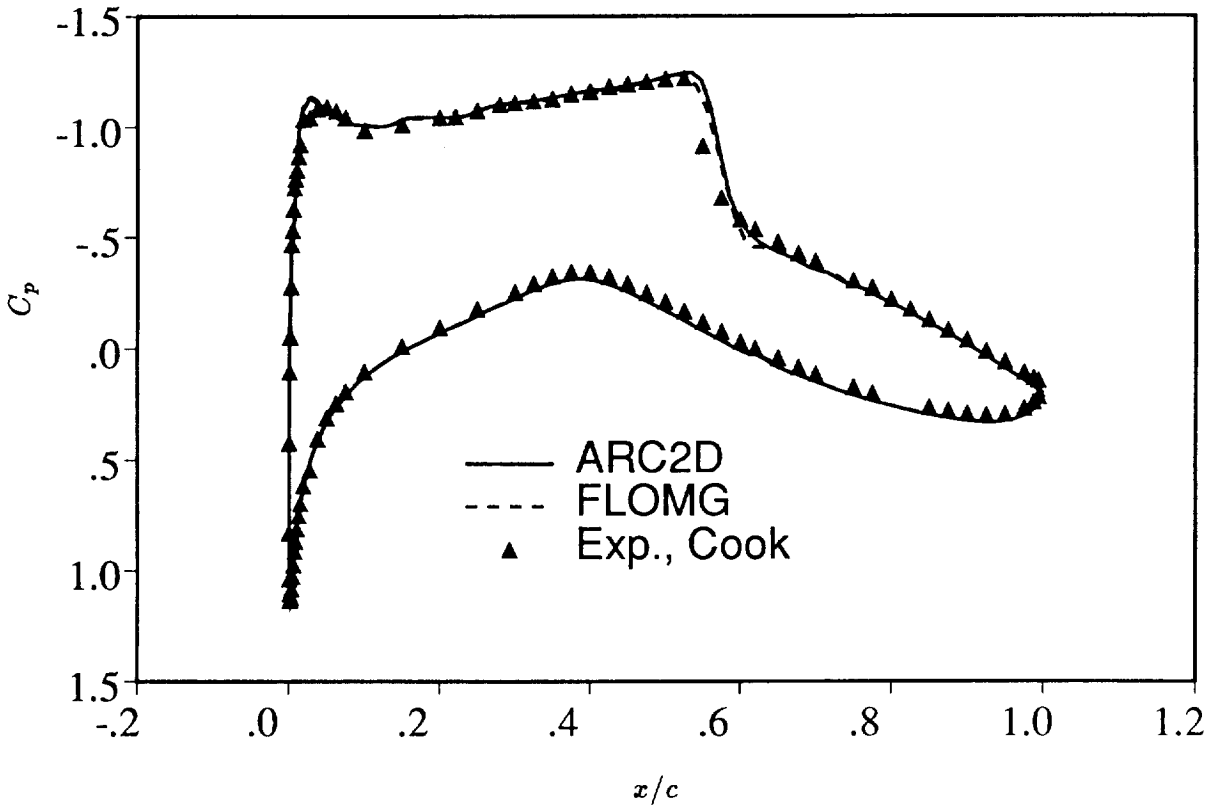
(b) Swanson grid.

Figure 5.–Continued.



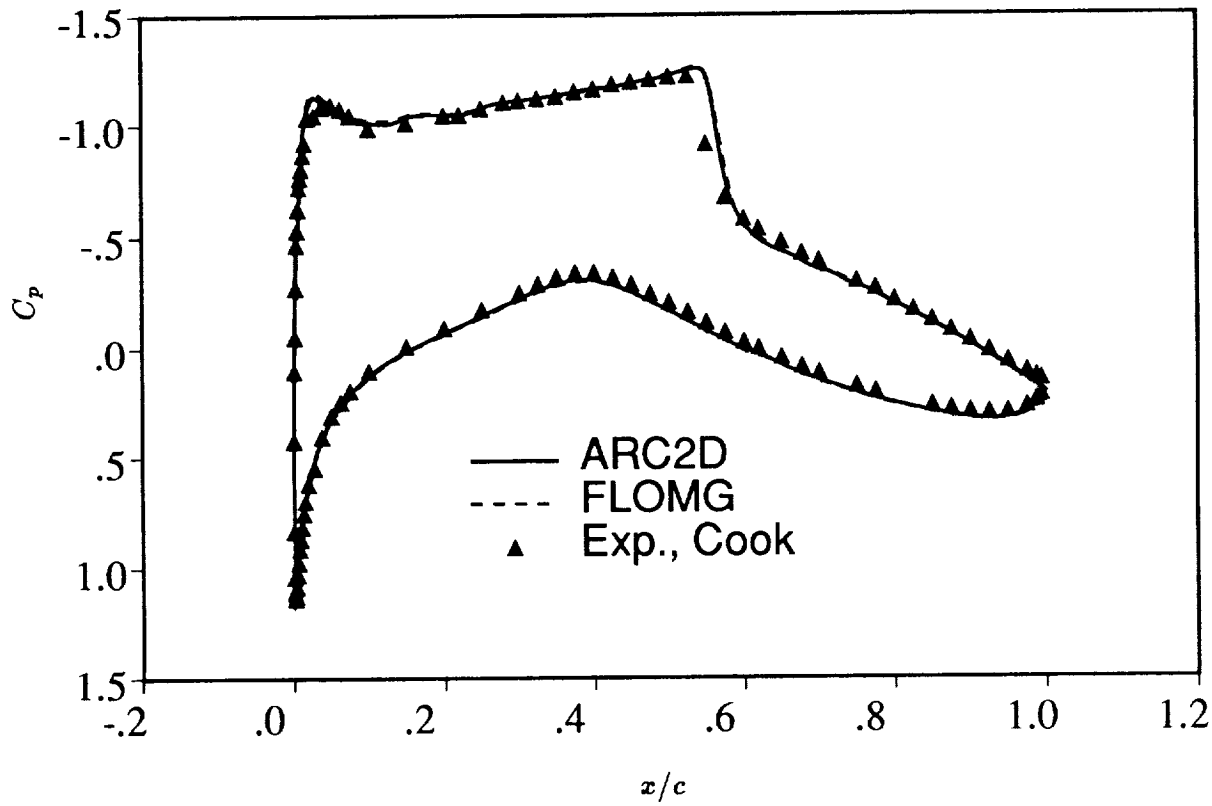
(c) Hall grid.

Figure 5.–Concluded.



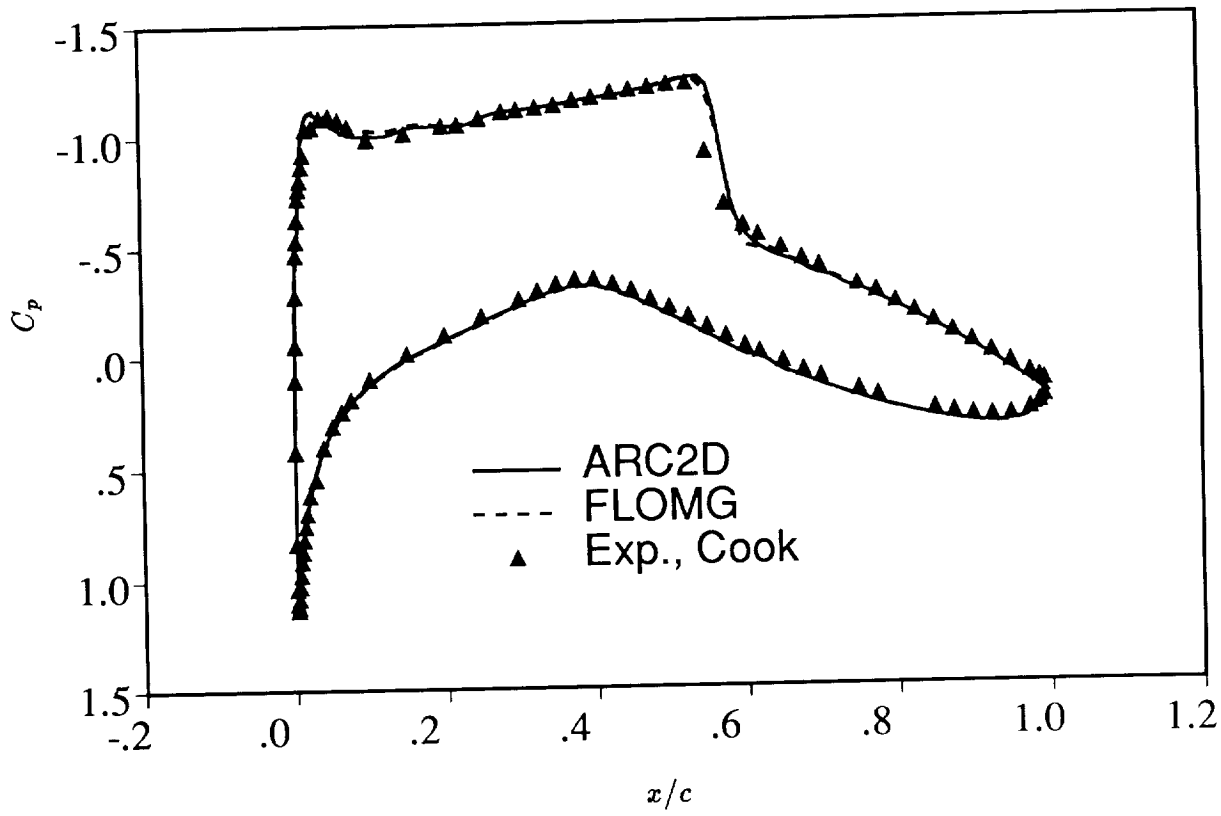
(a) Maksymiuk grid.

Figure 6.–Pressure coefficient for RAE Case 6: $M_{\infty_{exp}} = 0.725$, $M_{\infty_{corr}} = .729$, $\alpha_{exp} = 2.92°$, $\alpha_{corr} = 2.31°$.



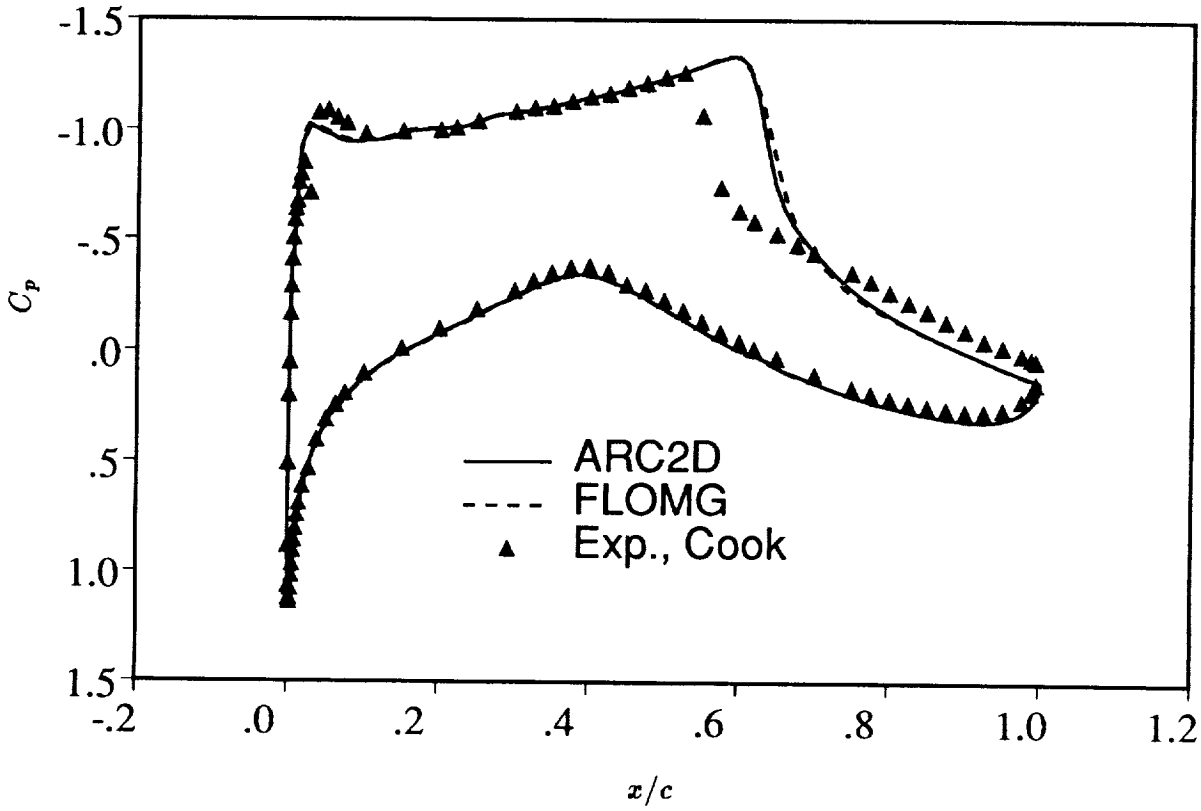
(b) Swanson grid.

Figure 6.–Continued.



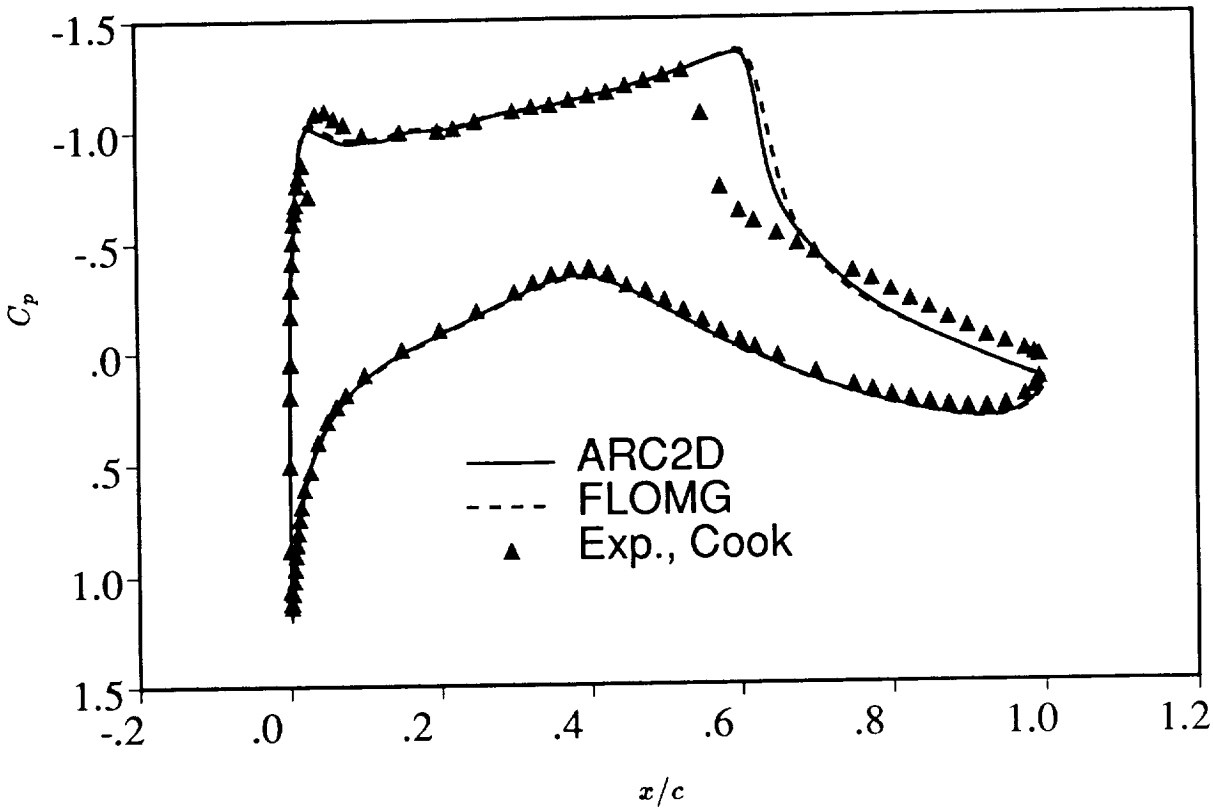
(c) Hall grid.

Figure 6.–Concluded.



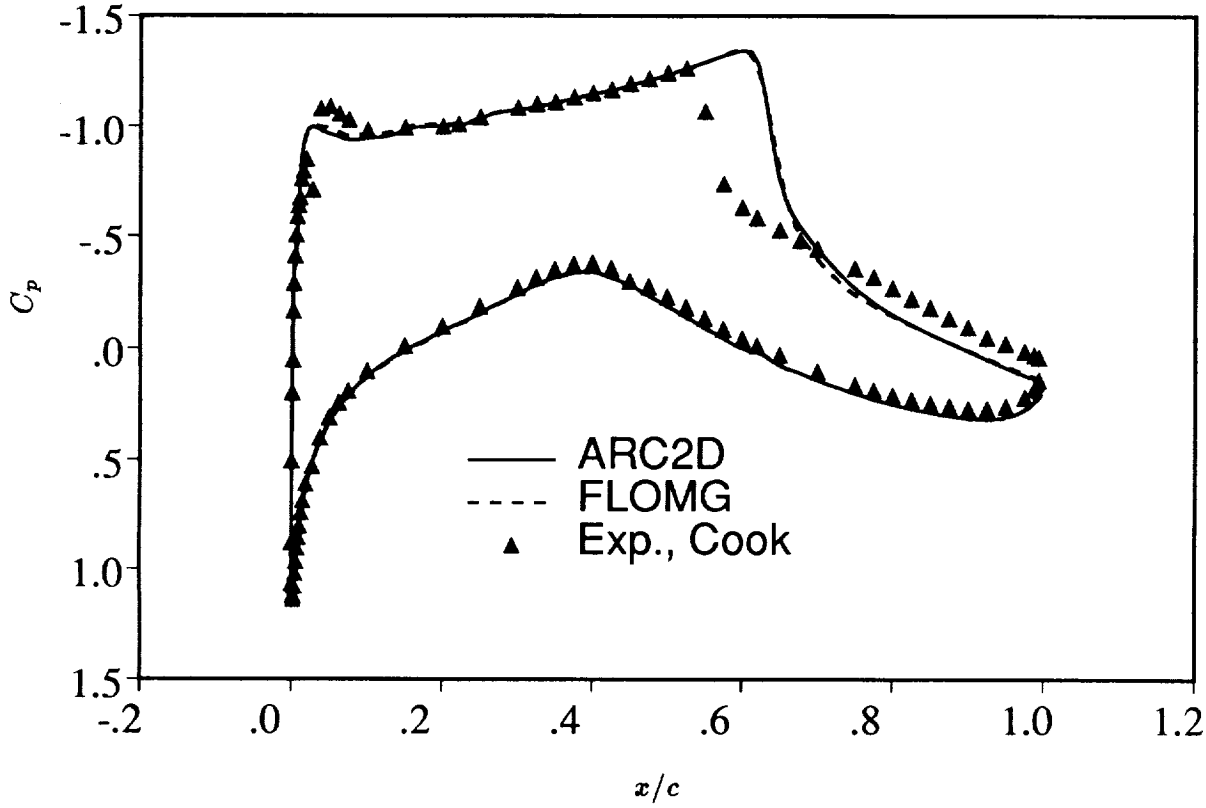
(a) Maksymiuk grid.

Figure 7.–Pressure coefficient for RAE Case 10: $M_{\infty_{exp}} = 0.75$, $M_{\infty_{corr}} = .754$, $\alpha_{exp} = 3.19°$, $\alpha_{corr} = 2.57°$.



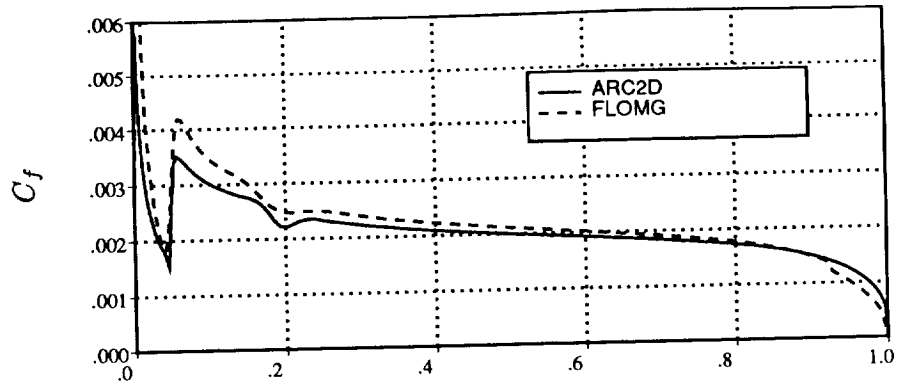
(b) Swanson grid.

Figure 7.–Continued.

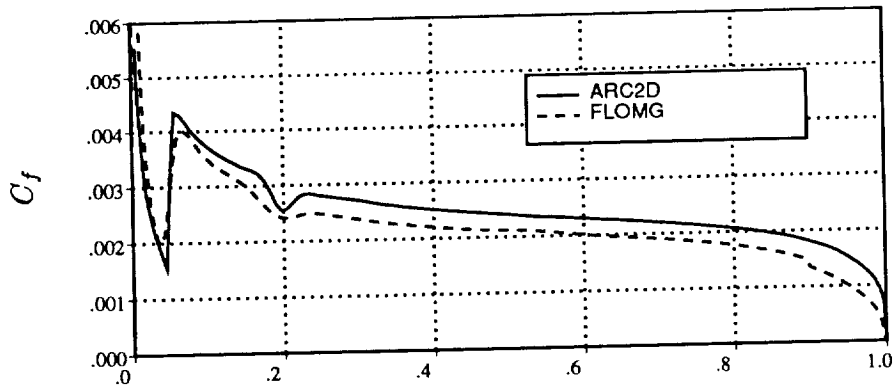


(c) Hall grid.

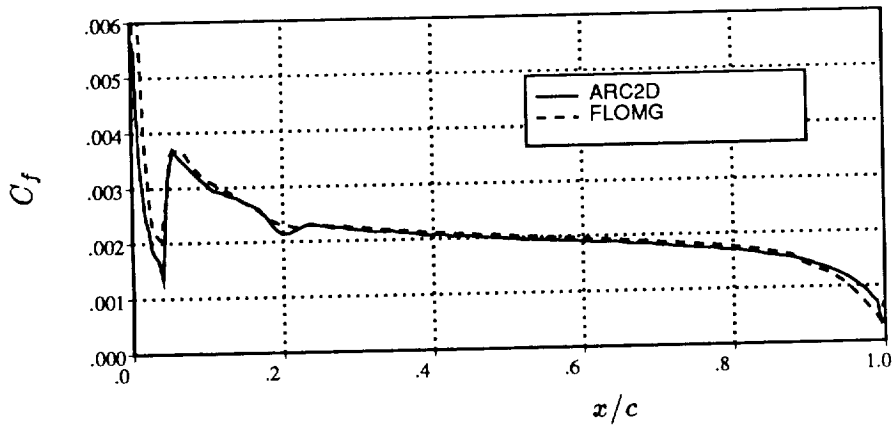
Figure 7.–Concluded.



(a) Maksymiuk grid.

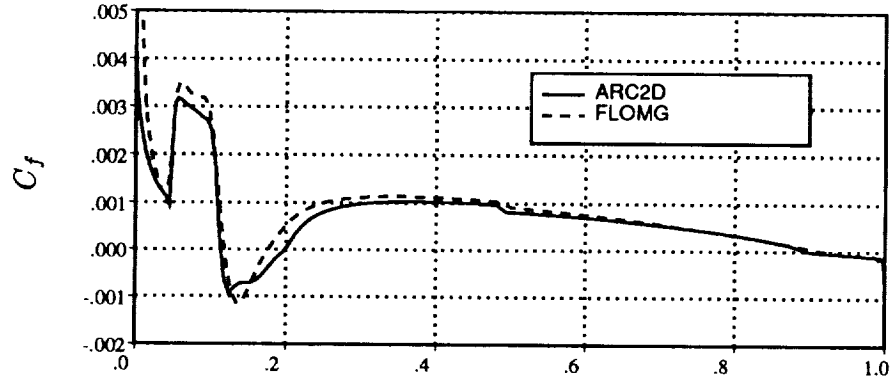


(b) Swanson grid.

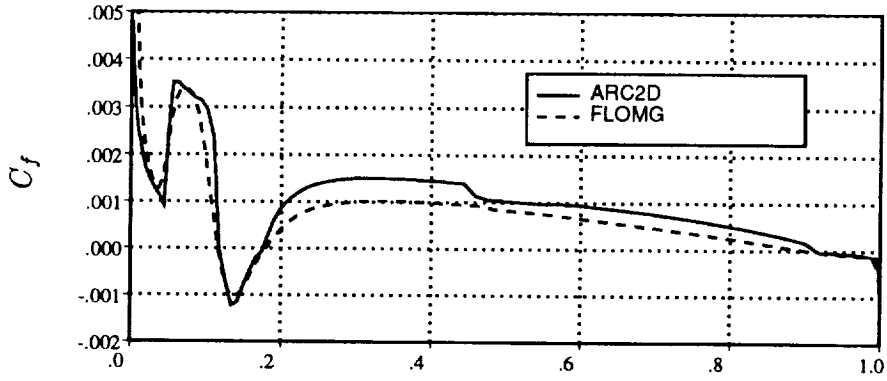


(c) Hall grid.

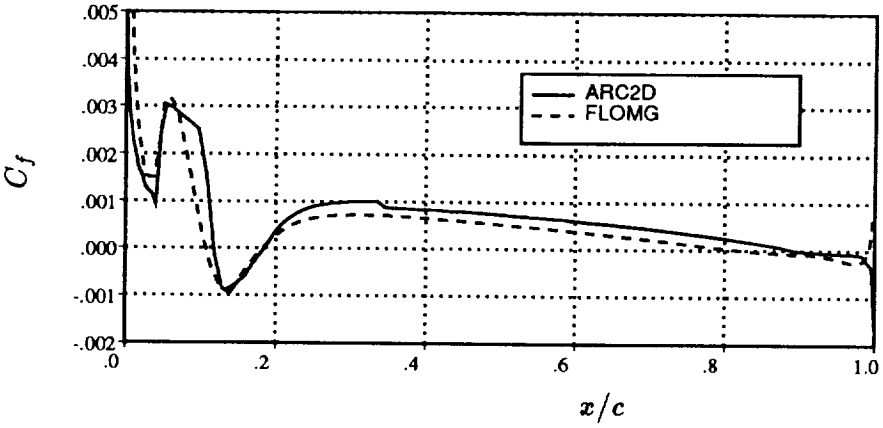
Figure 8.–Skin-friction coefficient for NACA case A1: $M_\infty = 0.7$, $\alpha_{exp} = 1.86°$, $\alpha_{corr} = 1.49°$.



(a) Maksymiuk grid.

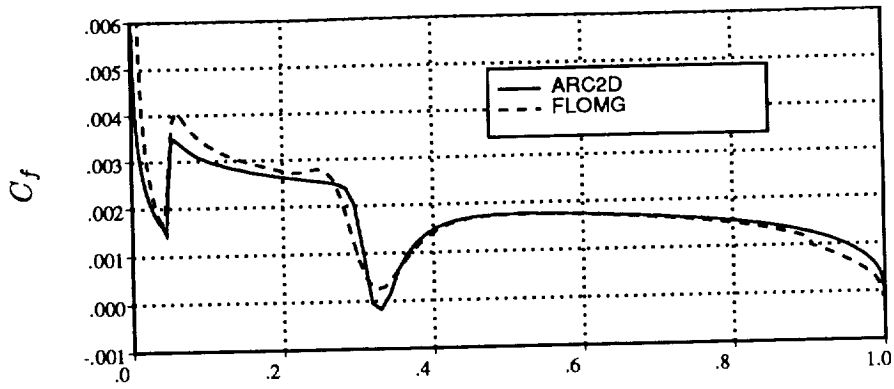


(b) Swanson grid.

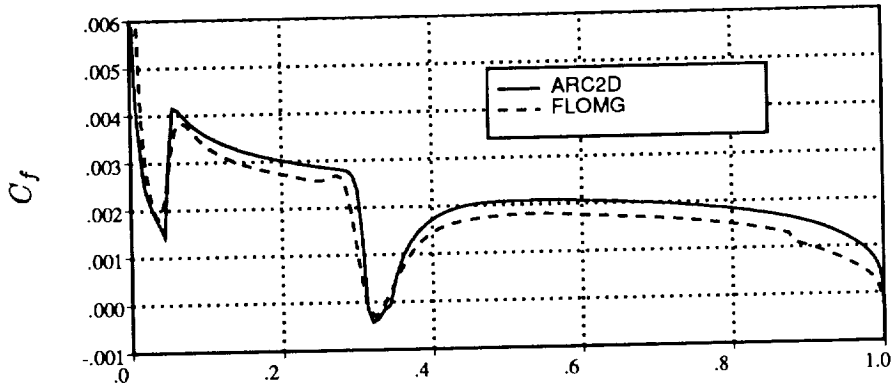


(c) Hall grid.

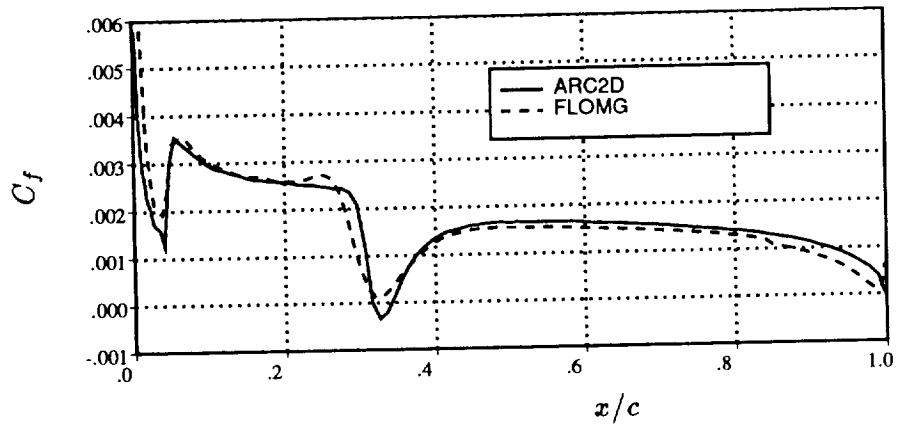
Figure 9.–Skin-friction coefficient for NACA case A2: $M_\infty = 0.55, \alpha_{exp} = 9.86°, \alpha_{corr} = 8.34°$.



(a) Maksymiuk grid.

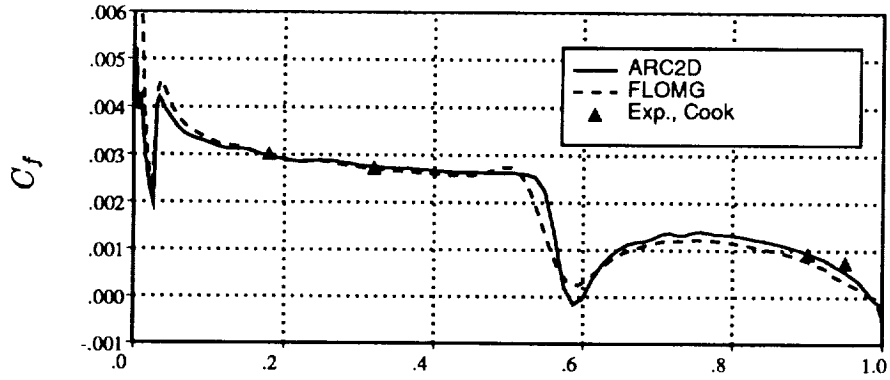


(b) Swanson grid.

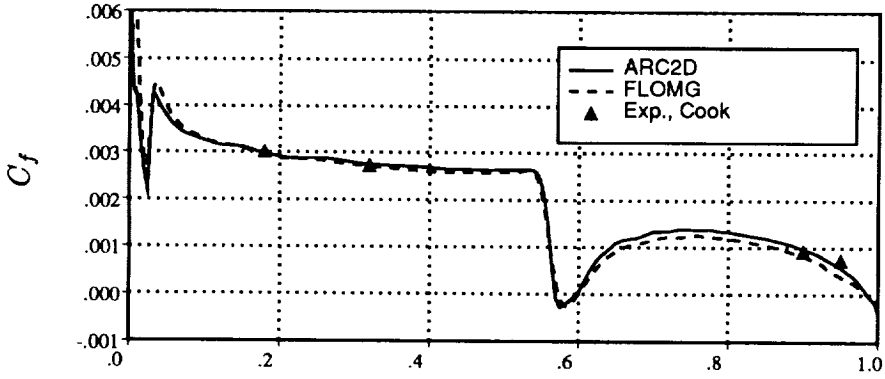


(c) Hall grid.

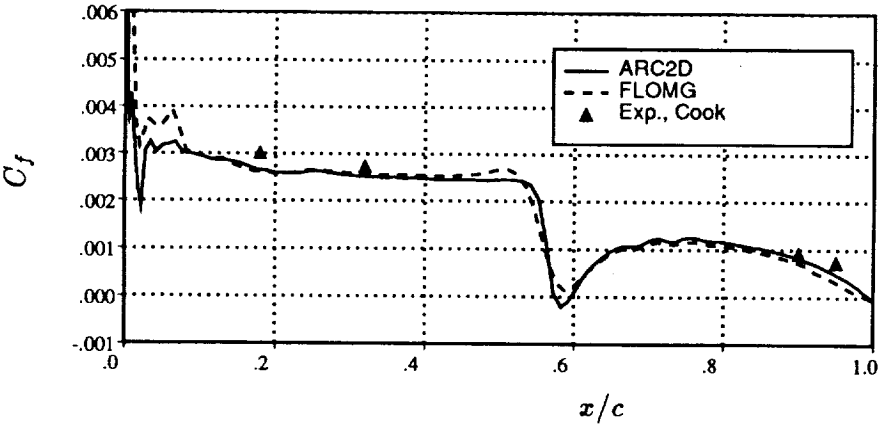
Figure 10.–Skin-friction coefficient for NACA case A4: $M_\infty = 0.7, \alpha = 3.0°$.



(a) Maksymiuk grid.

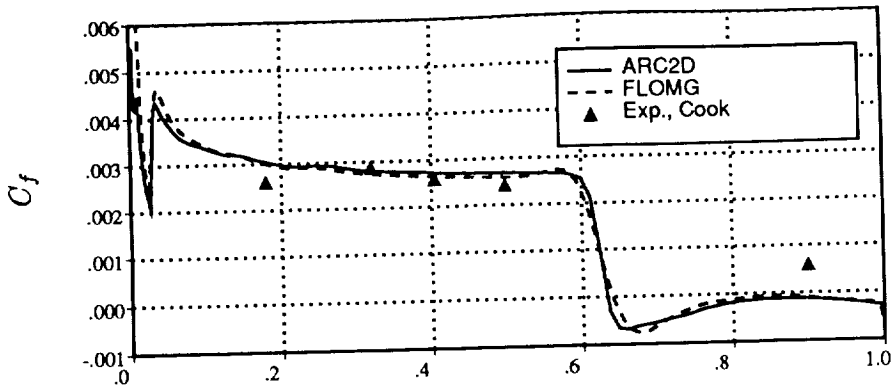


(b) Swanson grid.

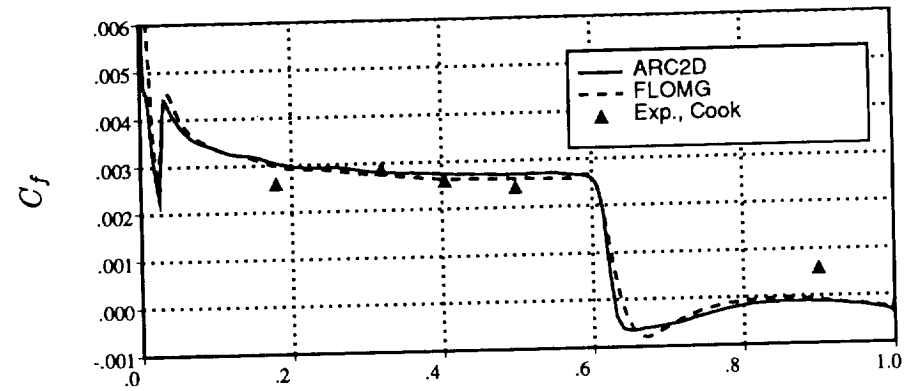


(c) Hall grid.

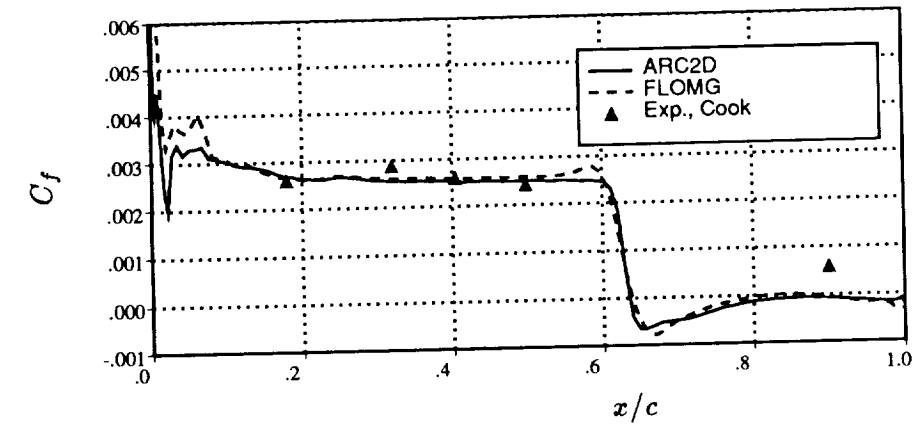
Figure 11.–Skin-friction coefficient on upper surface for RAE case 6: $M_{\infty_{exp}} = 0.725$, $M_{\infty_{corr}} = 0.729$, $\alpha_{exp} = 2.92°$, $\alpha_{corr} = 2.31°$.



(a) Maksymiuk grid.

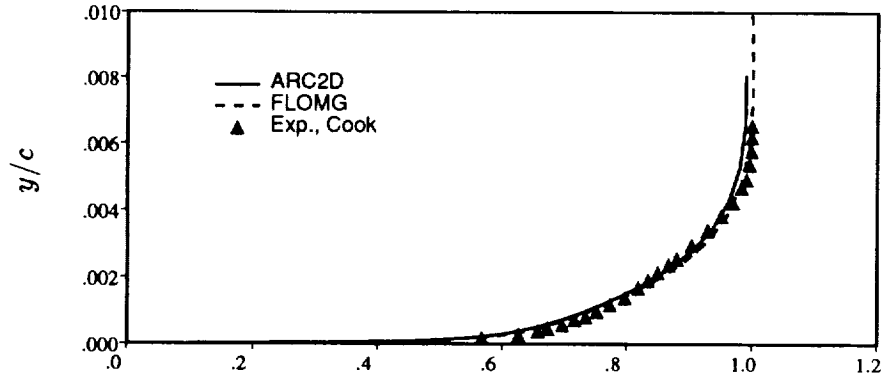


(b) Swanson grid.

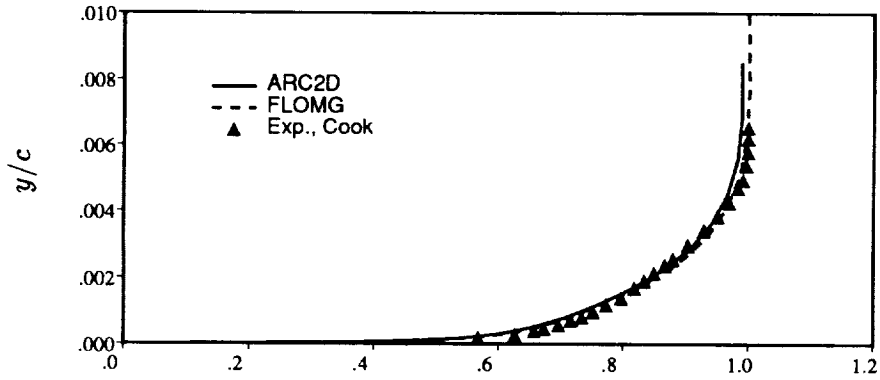


(c) Hall grid.

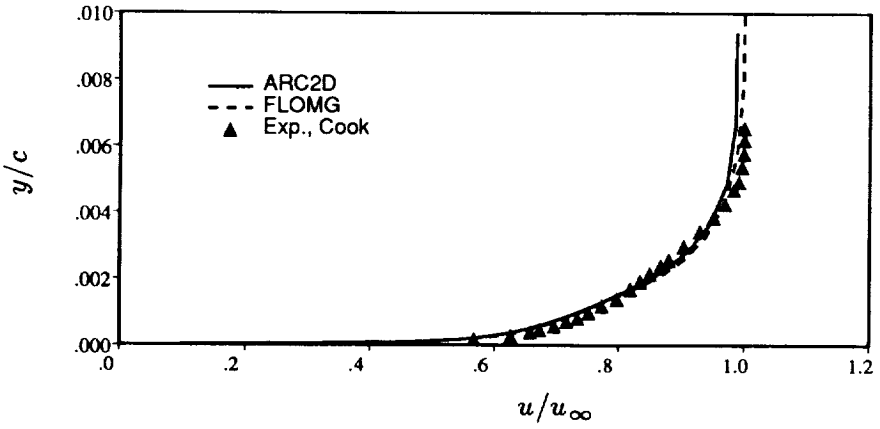
Figure 12.–Skin-friction coefficient on upper surface for RAE case 10: $M_{\infty_{exp}} = 0.75$, $M_{\infty_{corr}} = 0.754$, $\alpha_{exp} = 3.19°$, $\alpha_{corr} = 2.57°$.



(a) Maksymiuk grid.

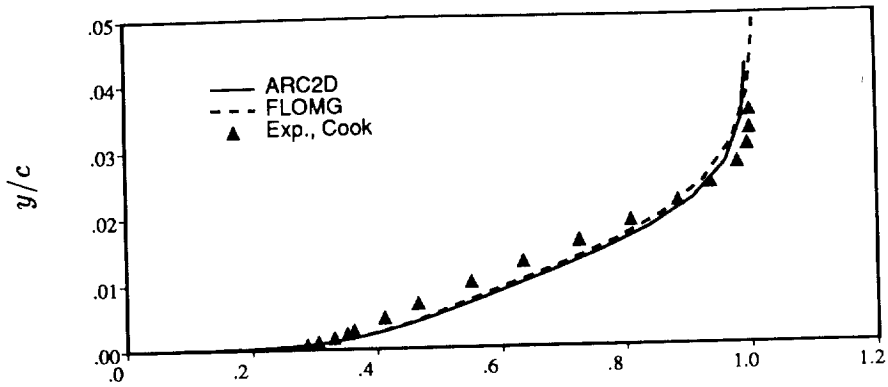


(b) Swanson grid.

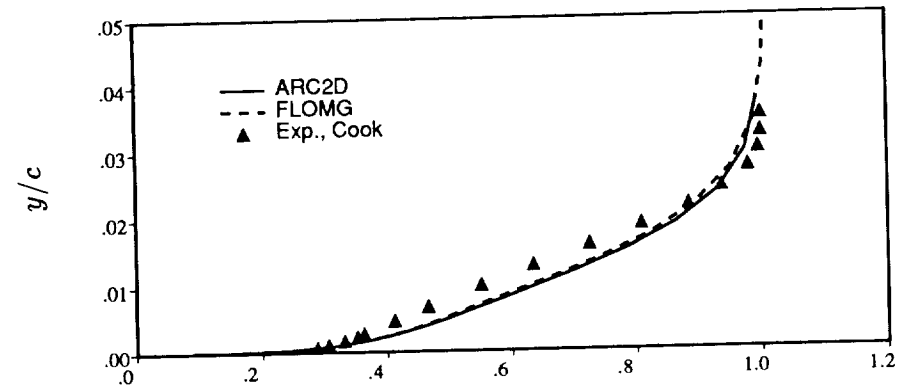


(c) Hall grid.

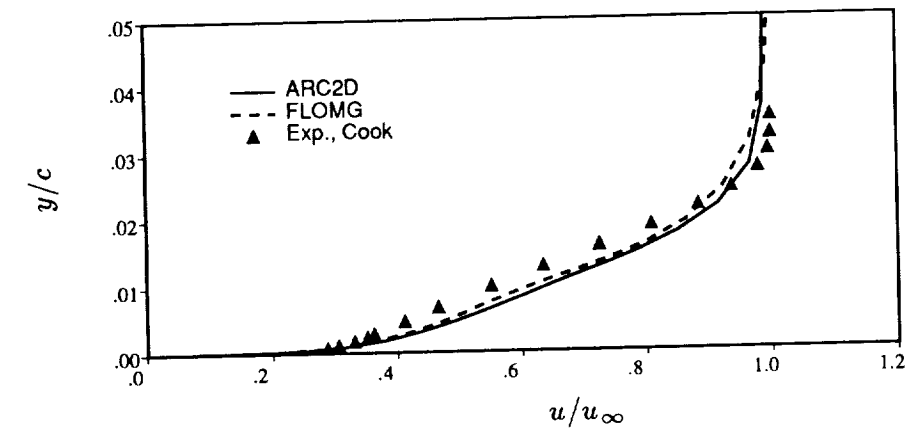
Figure 13.–Boundary-layer profile at $x = 0.319$ for RAE case 6: $M_{\infty_{exp}} = 0.725$, $M_{\infty_{corr}} = 0.729$, $\alpha_{exp} = 2.92°$, $\alpha_{corr} = 2.31°$.



(a) Maksymiuk grid.

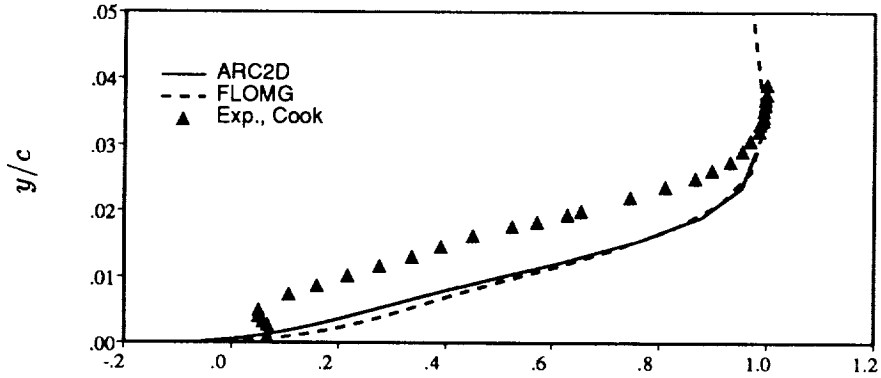


(b) Swanson grid.

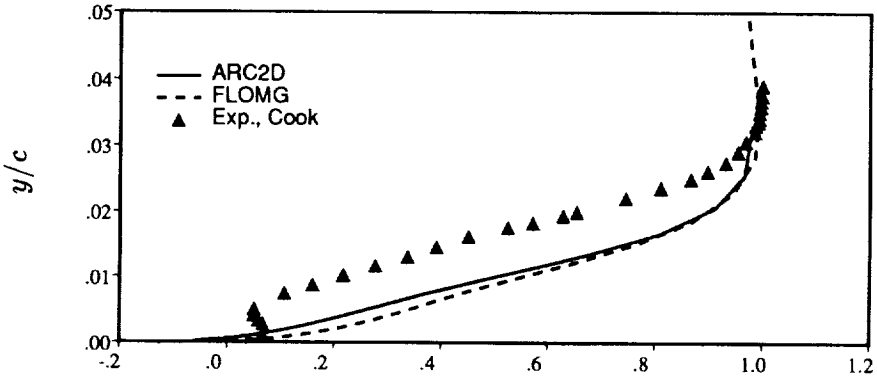


(c) Hall grid.

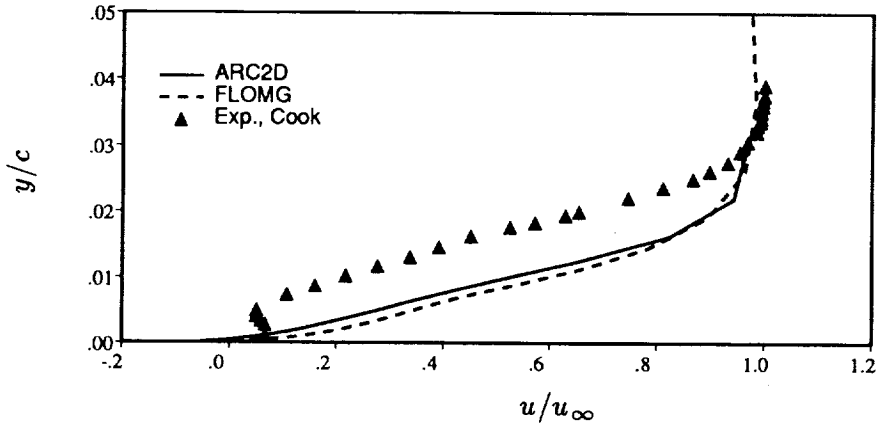
Figure 14.–Boundary-layer profile at $x = 0.95$ for RAE case 6: $M_{\infty_{exp}} = 0.725$, $M_{\infty_{corr}} = 0.729$, $\alpha_{exp} = 2.92°$, $\alpha_{corr} = 2.31°$.



(a) Maksymiuk grid.

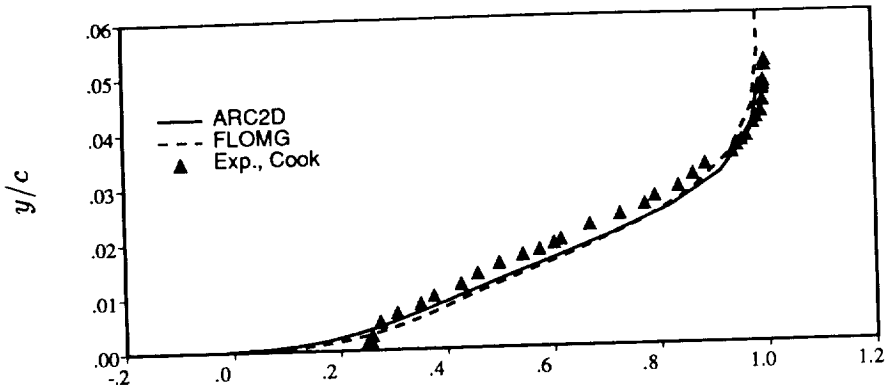


(b) Swanson grid.

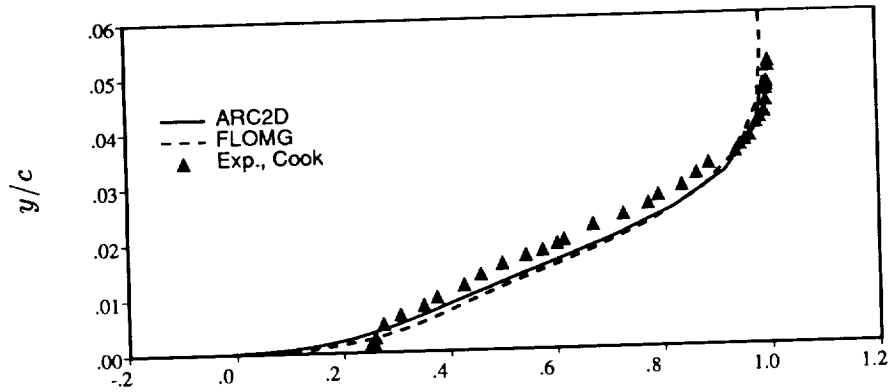


(c) Hall grid.

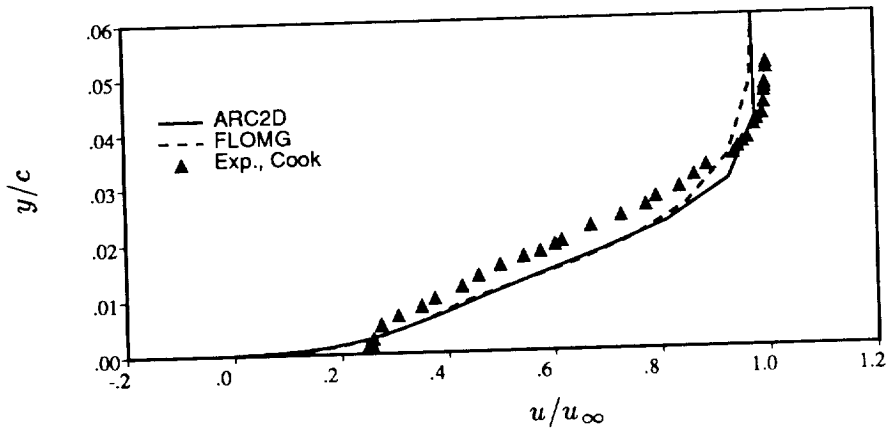
Figure 15.–Boundary-layer profile at $x = 0.75$ for RAE case 10: $M_{\infty_{exp}} = 0.75$, $M_{\infty_{corr}} = 0.754$, $\alpha_{exp} = 3.19°$, $\alpha_{corr} = 2.57°$.



(a) Maksymiuk grid.

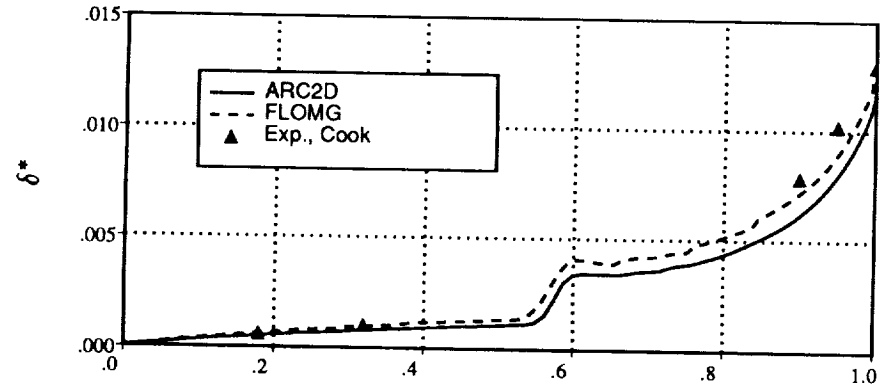


(b) Swanson grid.

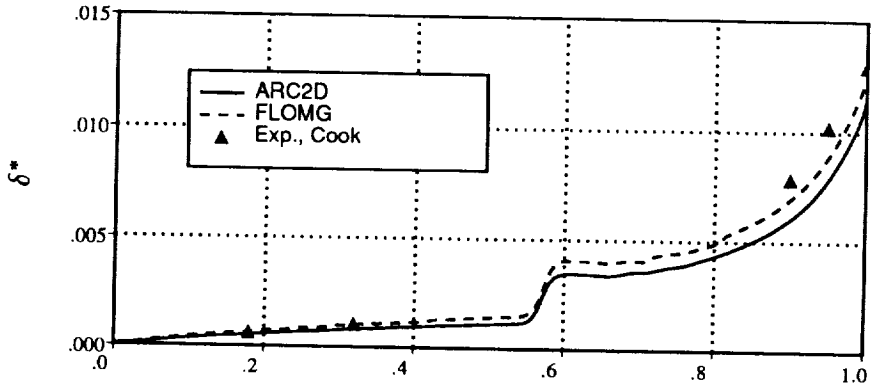


(c) Hall grid.

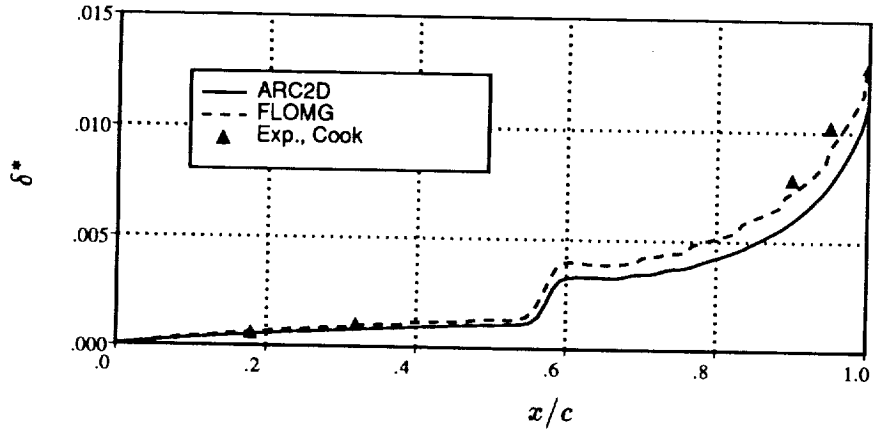
Figure 16.–Boundary-layer profile at $x = 0.90$ for RAE case 10; $M_{\infty_{exp}} = 0.75$, $M_{\infty_{corr}} = 0.754$, $\alpha_{exp} = 3.19°$, $\alpha_{corr} = 2.57°$.



(a) Maksymiuk grid.

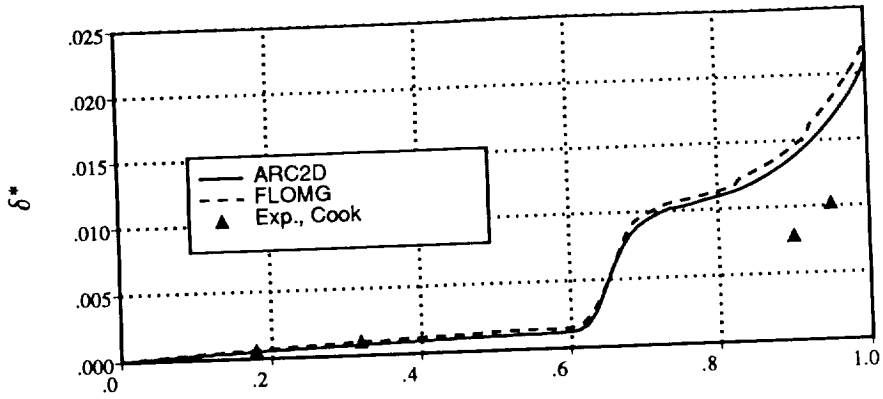


(b) Swanson grid.

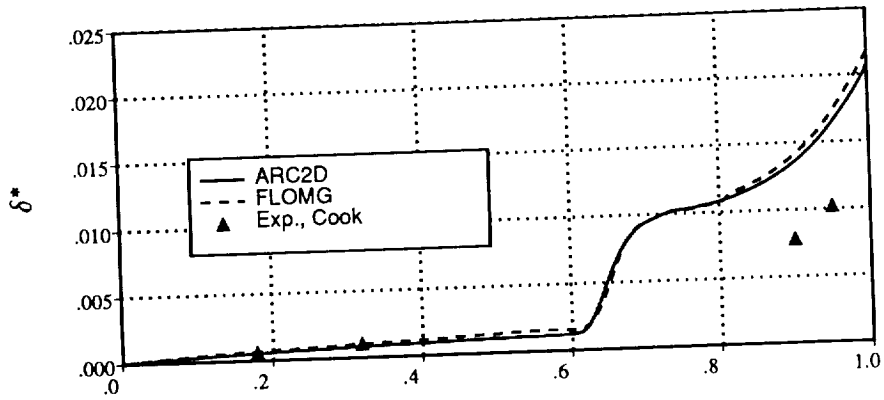


(c) Hall grid.

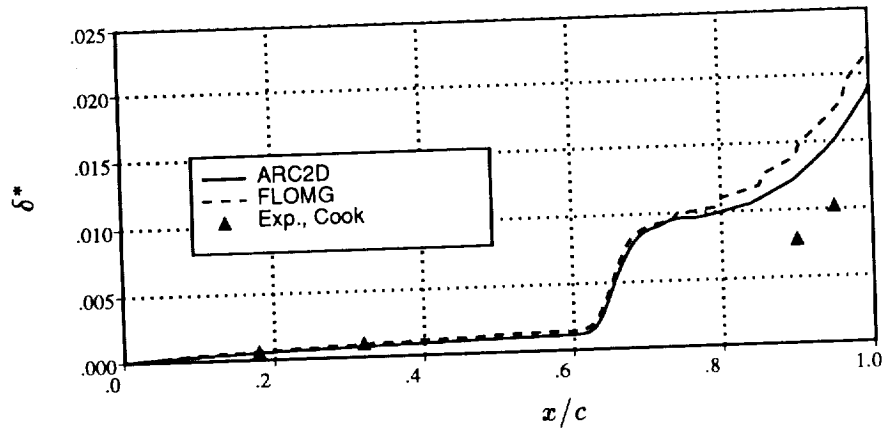
Figure 17.–Displacement thickness on upper surface for RAE case 6: $M_{\infty_{exp}} = 0.725$, $M_{\infty_{corr}} = 0.729$, $\alpha_{exp} = 2.92°$, $\alpha_{corr} = 2.31°$.



(a) Maksymiuk grid.

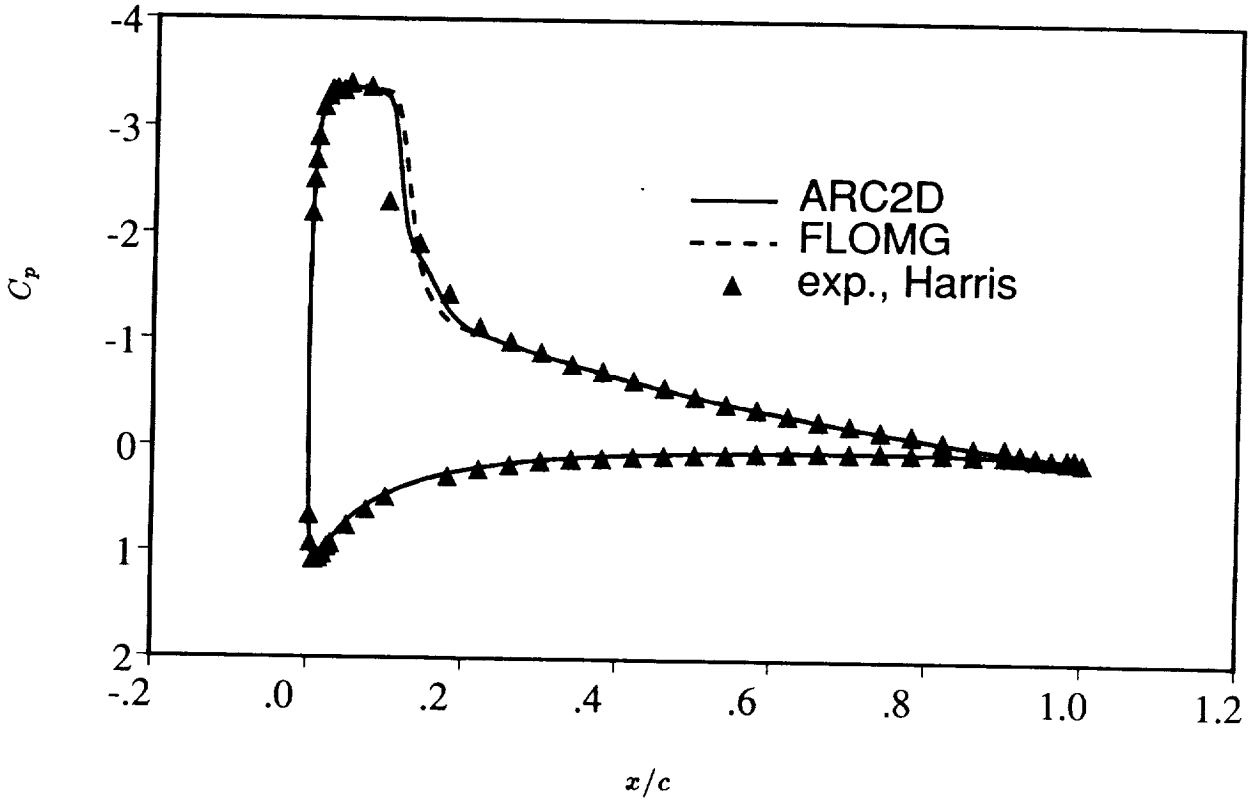


(b) Swanson grid.



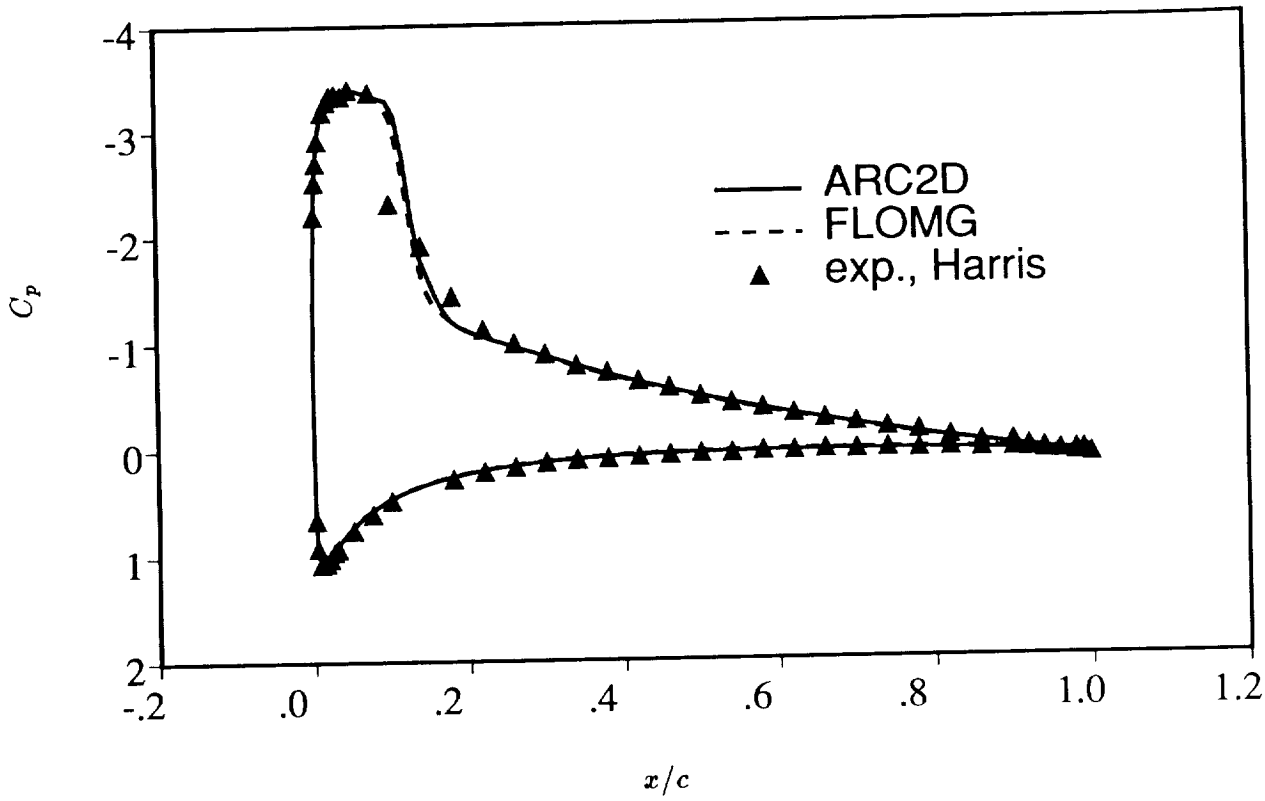
(c) Hall grid.

Figure 18.–Displacement thickness on upper surface for RAE case 10: $M_{\infty_{exp}} = 0.75$, $M_{\infty_{corr}} = 0.754$, $\alpha_{exp} = 3.19°$, $\alpha_{corr} = 2.57°$.



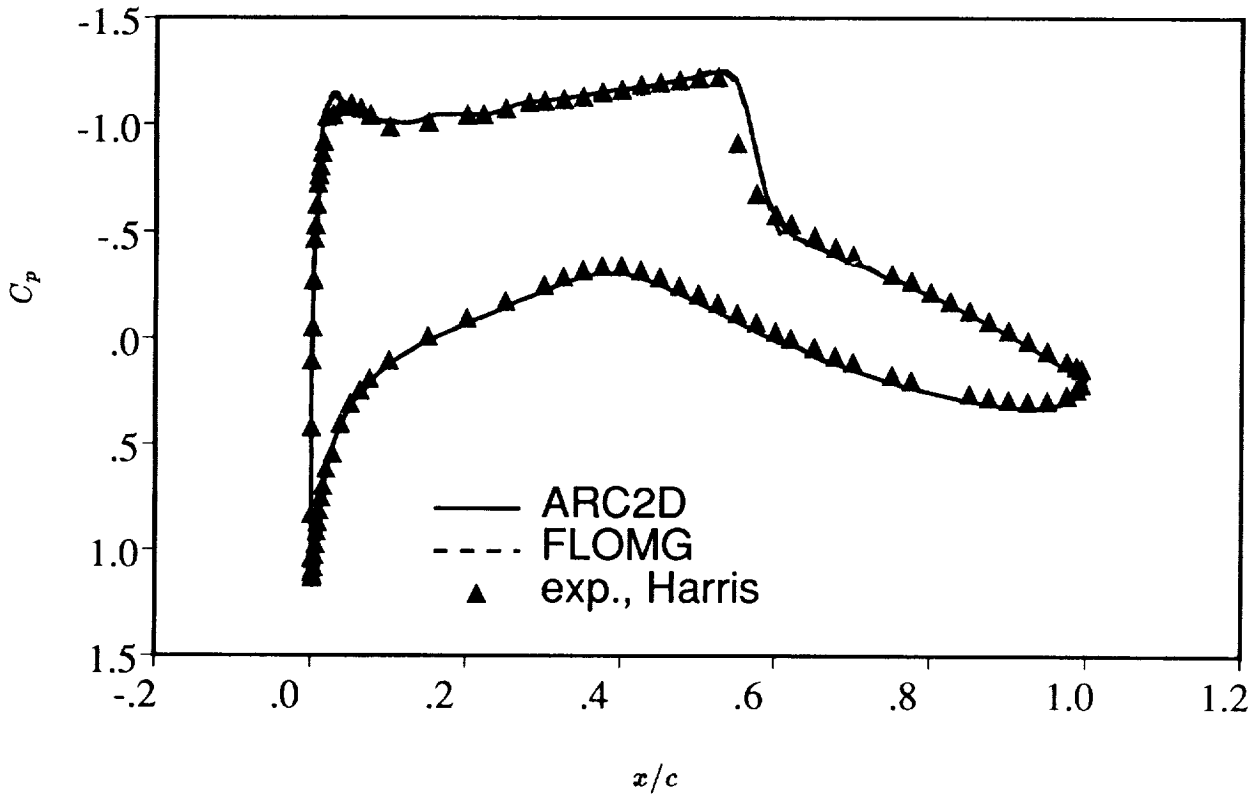
(a) Maksymiuk grid.

Figure 19.–Pressure coefficient for NACA case A2: $M_\infty = 0.55$, $\alpha_{exp} = 9.86°$, $\alpha_{corr} = 8.34°$.



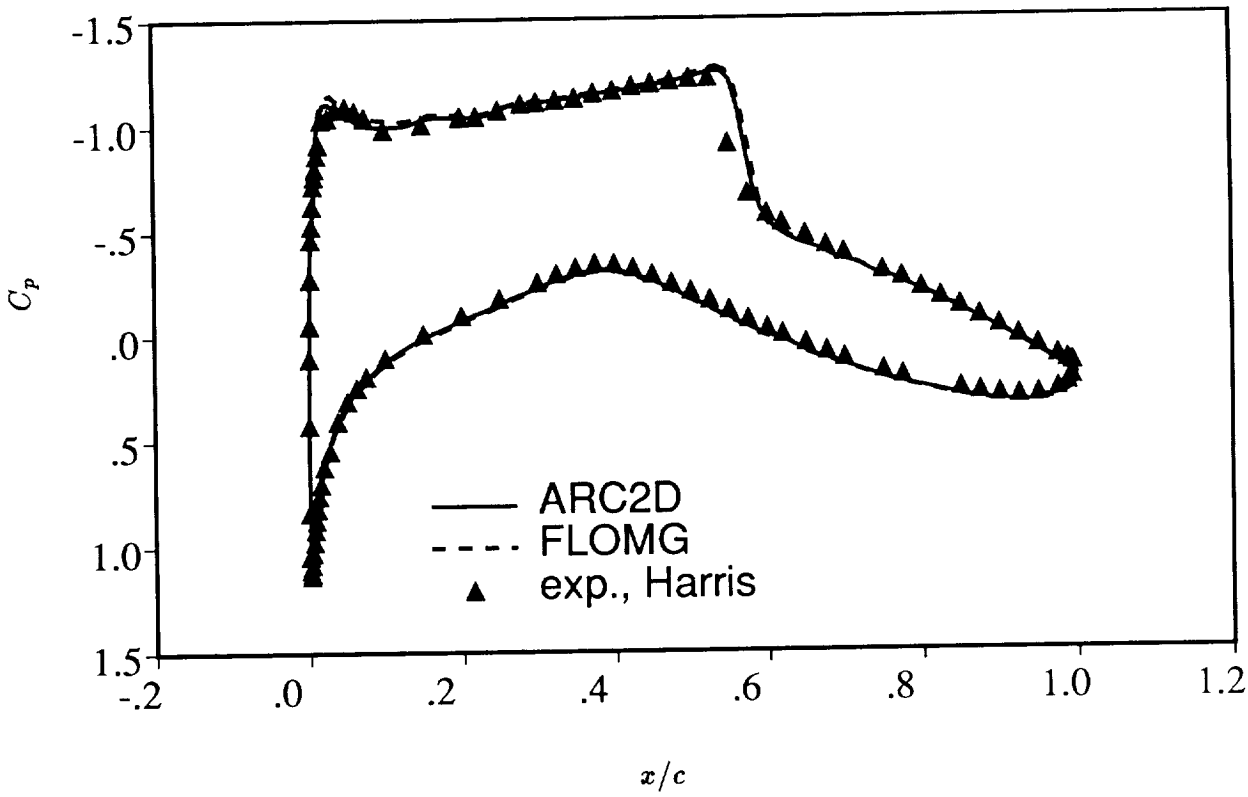
(b) Hall grid.

Figure 19.–Concluded.



(a) Maksymiuk grid.

Figure 20.–Pressure coefficient for RAE case 6: $M_{\infty_{exp}} = 0.725$, $M_{\infty_{corr}} = .729$, $\alpha_{exp} = 2.92°$, $\alpha_{corr} = 2.31°$.



(b) Hall grid.

Figure 20.–Concluded.

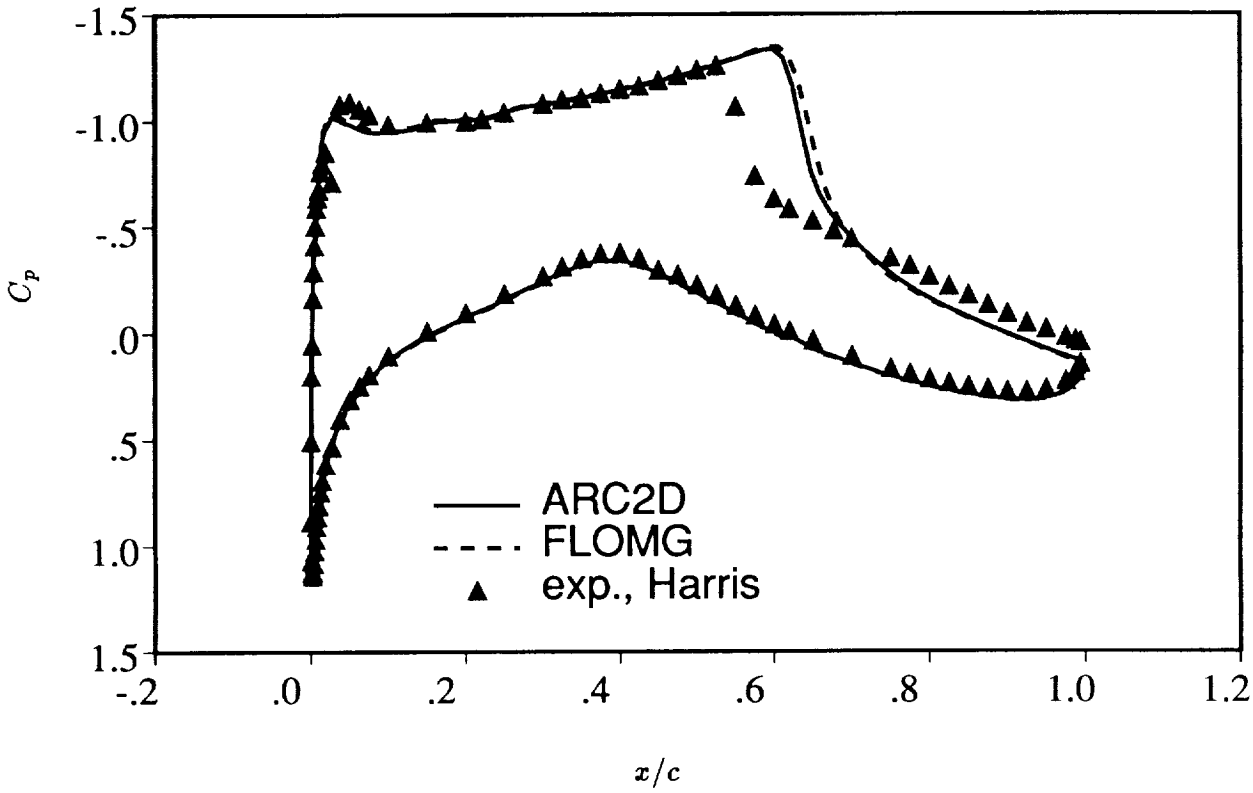
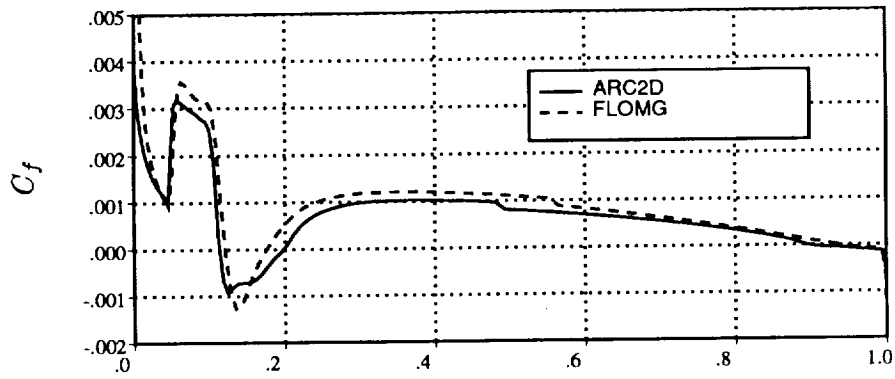
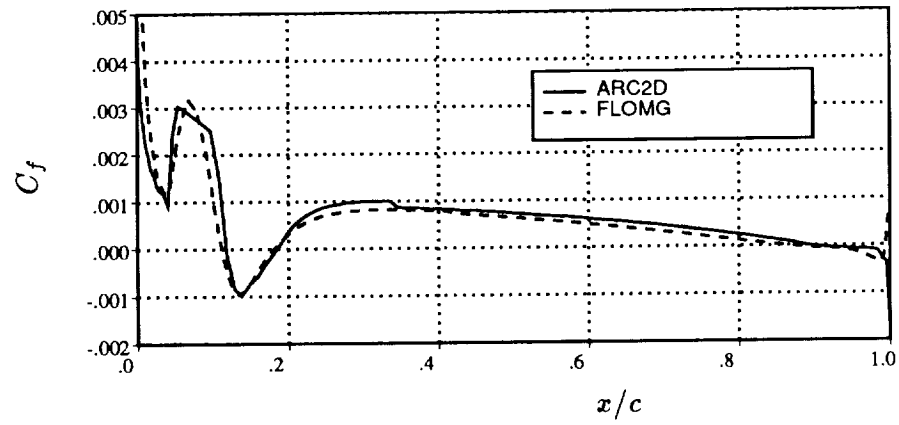


Figure 21.–Pressure coefficient for RAE case 10 on Maksymiuk grid: $M_{\infty_{exp}} = 0.75$, $M_{\infty_{corr}} = .754$, $\alpha_{exp} = 3.19°$, $\alpha_{corr} = 2.57°$.

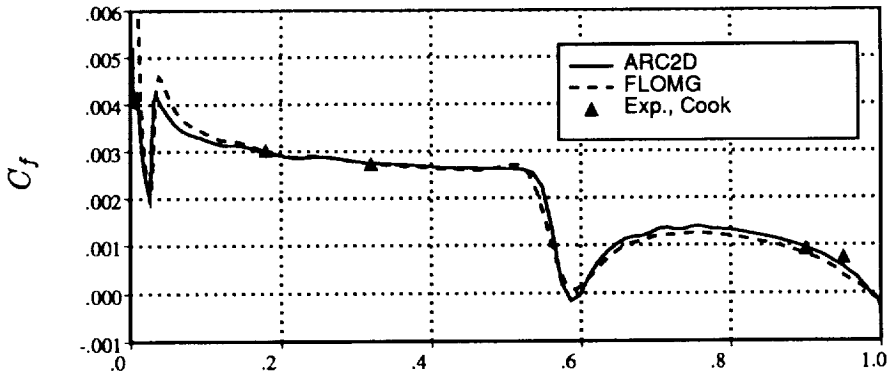


(a) Maksymiuk grid.

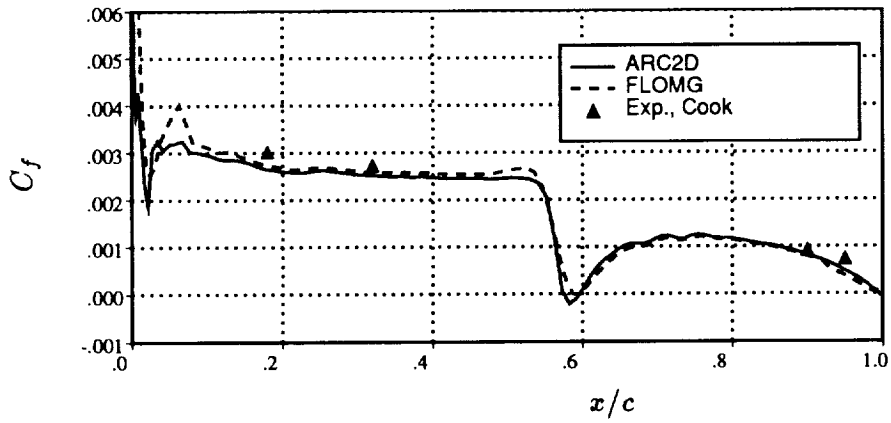


(b) Hall grid.

Figure 22.–Skin-friction coefficient for NACA case 2: $M_\infty = 0.55, \alpha_{exp} = 9.86°, \alpha_{corr} = 8.34°$.



(a) Maksymiuk grid.



(b) Hall grid.

Figure 23.–Skin-friction coefficient on upper surface for RAE case 6: $M_{\infty_{exp}} = 0.725$, $M_{\infty_{corr}} = 0.729$, $\alpha_{exp} = 2.92°$, $\alpha_{corr} = 2.31°$.

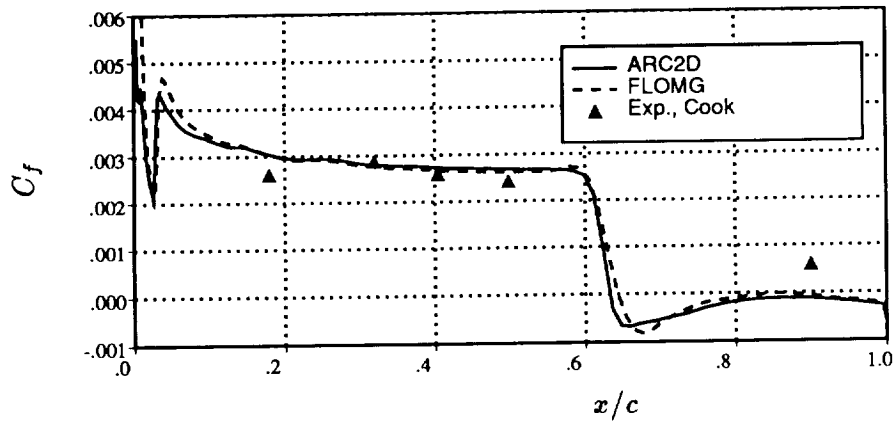


Figure 24.–Skin-friction coefficient on upper surface for RAE Case 10 on Maksymiuk grid: $M_{\infty_{exp}} = 0.75$, $M_{\infty_{corr}} = 0.754$, $\alpha_{exp} = 3.19°$, $\alpha_{corr} = 2.57°$.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-102815		

4. Title and Subtitle

A Comparison of Two Central Difference Schemes for Solving the Navier-Stokes Equations

5. Report Date

July 1990

6. Performing Organization Code

7. Author(s)

C. M. Maksymiuk, R. C. Swanson (Langley Research Center), and T. H. Pulliam

8. Performing Organization Report No.

A-90141

10. Work Unit No.

505-60

9. Performing Organization Name and Address

Ames Research Center
Moffett Field, CA 94035-1000

11. Contract or Grant No.

13. Type of Report and Period Covered

Technical Memorandum

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: Catherine M. Maksymiuk, Ames Research Center, MS 202A-1,
Moffett Field, CA 94035-1000
(415)604-4737 or FTS 464-4737

16. Abstract

Five viscous transonic airfoil cases have been computed by two significantly different computational fluid dynamics codes: an explicit finite-volume algorithm with multigrid, and an implicit finite-difference approximate-factorization method with eigenvector diagonalization. Both methods are described in detail, and their performance on the test cases is compared. The codes utilized the same grids, turbulence model, and computer to provide the truest test of the algorithms. The two approaches produce very similar results, which, for attached flows, also agree well with experimental results; however, the explicit code is considerably faster.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computational fluid dynamics Navier-Stokes equations Airfoils	Unclassified–Unlimited Subject Category–02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	62	A04

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, Virginia 22161