Design and Implementation of a Multigrid Code for the Euler Equations

DENNIS C. JESPERSEN†

MS 202A-1, NASA Ames Research Center
Moffett Field, CA 94035

The steady-state equations of inviscid fluid flow, the Euler equations, are a nonlinear nonelliptic system of equations admitting solutions with discontinuities (for example, shocks). The efficient numerical solution of these equations poses a strenuous challenge to multigrid methods. A multigrid code has been developed for the numerical solution of the Euler equations. In this paper some of the factors that had to be taken into account in the design and development of the code are reviewed. These factors include the importance of choosing an appropriate difference scheme, the usefulness of local mode analysis as a design tool, and the crucial question of how to treat the nonlinearity. Sample calculations of transonic flow about airfoils will be presented. No claim is made that the particular algorithm presented is optimal.

§1. Introduction.

This paper will discuss the numerical solution of the steady Euler equations using multigrid techniques. The problem studied will be that of steady two-dimensional flow about an airfoil section. This work has as its ultimate motivation the desire to provide an effective tool for aircraft designers. Apart from this motivation, the discussion in this paper should be valuable to workers seeking to implement multigrid methods for complicated problems.

A great deal of effort has been expended in the last decade on devising efficient solution schemes for the transonic full potential equation (a scalar equation describing inviscid fluid flow, derived assuming the flow is irrotational, incompressible, and isentropic). Both multigrid and non-multigrid techniques have been developed (Jameson [1] , Shmilovich and Caughey [2] , and Holst [3]). The recent discovery of possible multiple solutions for the

†Research Scientist, Computational Fluid Dynamics Branch.

transonic full potential equation (Steinhoff and Jameson [4]), the desire to compute flows with shocks stronger than those for which the potential equation is adequate, and the desire ultimately to compute the vortex wake behind a wing without *a priori* modeling it in some fashion have led to a recent surge of interest in efficient computational methods for the Euler equations. Standard solution techniques for the steady Euler equations are based on time-like methods with various artifices employed to speed convergence (see for example Jameson, Schmidt, and Turkel [5] , Steger [6] , and Pulliam, Jespersen, and Childs [7]). This work is an attempt to apply multigrid techniques directly to the steady Euler equations, bypassing any time-like iteration. It thus differs from work of Steger [8] , who employed a multigrid technique as a component of a standard implicit algorithm, and from work of Ni [9] and Johnson [10] , who use "multiple grids" to accelerate an explicit time-marching algorithm.

In the next section of this paper the differential equation problem is described, including the transformation to general curvilinear coordinates. In section 3 a discretization method using flux-vector splitting is described. Section 4 considers the important question of how to deal with the nonlinearity of the equations. The Newton-multigrid and FAS (Full Approximation Scheme) approaches will be described and discussed. Section 5 gives some computed results for flows about airfoils, including transonic flows with shocks.

§2. The Euler Equations

This section describes the differential system that is the object of study. The *unsteady* Euler equations are a nonlinear hyperbolic system of equations that describe the conservation of mass, momentum, and energy for inviscid fluid flow. In two space dimensions they may be written

$$\partial_t Q + \partial_x E(Q) + \partial_y F(Q) = 0, \tag{2.1}$$

where Q is the four-vector $Q = (q_1, q_2, q_3, q_4)^T = (\rho, \rho u, \rho v, e)^T$. Here ρ is density, u and v are Cartesian velocity components, and e is total energy per unit volume. The flux functions $E, F : D \subset \mathbf{R}^4 \to \mathbf{R}^4$ are nonlinear functions given by

$$E(Q) = (\rho u, \rho u^2 + p, \rho uv, u(e+p))^T$$
$$F(Q) = (\rho v, \rho uv, \rho v^2 + p, v(e+p))^T.$$

The pressure p is given by the equation of state

$$p = (\gamma - 1)(e - \rho(u^2 + v^2)/2),$$

where γ is a constant ($\gamma = 1.4$ for air). This paper deals only with the steady Euler equations, which are obtained by setting the time derivative term in (2.1) equal to 0, thus obtaining

$$\partial_x E(Q) + \partial_y F(Q) = 0. \tag{2.2}$$

There is little or no mathematical theory to rely on in studying these equations. They are nonlinear, nonelliptic by the definition of Agmon, Douglis and Nirenberg [11] (even in subsonic flow regions, unlike the potential equation, which is elliptic in regions of subsonic flow), and admit discontinuous solutions (e.g., shocks). Furthermore, it is not clear what are proper boundary conditions for the system (2.2). The lack of an adequate theory makes it virtually impossible to prove rigorous convergence results for solutions of discretizations of (2.2).

The region where we wish to solve (2.2) is a bounded region containing an airfoil section (see Figure 1). Typically the outer boundary will be located 6 chord lengths from the airfoil. It will be desirable to map from the physical (x, y) space with its relatively complicated geometry to a "computational" (ξ, η) space with a simple geometry. The use of such a transformation

simplifies the applications of boundary conditions, allows one to cluster grid points in regions of rapid variation of Q (e.g., near shocks or at the leading edge), and allows all finite difference equations to be written on a uniform mesh. The price paid for this is a more complicated set of equations to solve. Specifically, if $\xi = \xi(x,y)$ and $\eta = \eta(x,y)$ are the components of the map $(x,y) \to (\xi,\eta)$ (note that $(x,y) \to (\xi,\eta)$ is not continuous, or even single-valued, due to the cut in the wake), then the transformed equations may be written

$$\partial_\xi[y_\eta E(Q) - x_\eta F(Q)] + \partial_\eta[-y_\xi E(Q) + x_\xi F(Q)] = 0, \tag{2.3}$$

or

$$\partial_\xi \hat{E}(Q) + \partial_\eta \hat{F}(Q) = 0. \tag{2.4}$$

Now the geometry is simpler but the flux functions $\hat{E}$ and $\hat{F}$ are more complicated. The metric coefficients y_η, etc., in (2.3) can vary over several orders of magnitude. Thus we must consider a nonlinear system with widely varying coefficients.

We will take the image of the physical domain of Figure 1 to be a rectangle in computational space, see Figure 2. The image of the airfoil boundary lies along the line segment BC in the computational plane, the segments AF and DE are the images of the outflow boundary, the segment EF is the image of the inflow and far field boundary, and the segments AB and CD are the images of the cut in the wake. Extra boundary conditions must be prescribed along AB and CD; these boundary conditions say that Q is continuous across the wake cut.

§3. Discretization

The difference scheme will be constructed on a uniform mesh in computational space. We will assume the computational space is a rectangle $1 \le \xi \le$ jmax, $1 \le \eta \le$ kmax. The flow variables at (j,k) will be denoted Q_{jk}.

(Thus we are using a non-staggered grid.) There are many possible difference schemes for (2.4); the simplest is probably central differencing. Simply replace the derivatives in (2.4) by centered differences, obtaining the equation (using $\Delta\xi = \Delta\eta = 1$ and multiplying through by 2)

$$(\hat{E}_{j+1,k} - \hat{E}_{j-1,k}) + (\hat{F}_{j,k+1} - \hat{F}_{j,k-1}) = 0 \tag{3.1}$$

where we write $\hat{E}_{j+1,k} = \hat{E}(Q_{j+1,k})$, etc. Local linearization of these equations would lead to a matrix with 4-by-4 blocks of zeros on the main diagonal. Any form of point Gauss-Seidel would then be impossible. It is true that central differencing with some form of artificial viscosity would lead to a difference scheme to which Gauss-Seidel can be applied, but Gauss-Seidel is a slowly convergent relaxation scheme, and has a poor smoothing rate, for central differencing applied to a convection-dominated problem with a very small viscosity term. (It may be the case that some form of Distributive Gauss-Seidel can be applied, on a staggered grid; see Brandt [12] See also Childs and Pulliam [13] for a centrally-differenced algorithm using artificial viscosity and Euler implicit relaxation.)

Another relaxation scheme that can be applied to a linearization of (3.1) is Kaczmarz relaxation (McCormick [14]). The Kaczmarz scheme is defined for the n-by-n linear system $Au = f$ by

$$u \leftarrow u - \frac{\langle Au - f, e_i \rangle}{\|A^T e_i\|_2^2} A^T e_i, \quad i = 1, \ldots, n, \tag{3.2}$$

where the left arrow denotes replacement, the brackets $\langle \cdot, \cdot \rangle$ denote the l_2 inner product, and e_i are the usual unit vectors. Thus one step of the Kaczmarz scheme consists of n substeps, where the i^{th} substep involves adding a multiple of the i^{th} row of A to the current solution, the multiple being chosen so that after the addition the i^{th} equation is satisfied. It is known that the convergence rate of Kaczmarz relaxation for the matrix A is

the same as the convergence rate of Gauss-Seidel relaxation for the matrix AA^T. It can also be shown that the smoothing rate of Kaczmarz relaxation for the matrix A is the same as the smoothing rate of Gauss-Seidel for the matrix AA^T. Use of these facts and consideration of simple linear scalar problems shows that Kaczmarz relaxation is not suitable as a relaxation scheme for central differencing applied to a first order equation.

The failure of Gauss-Seidel and Kaczmarz forces one to search for a new relaxation scheme, to use a different (staggered?) grid, or to use a different differencing scheme. In this work the third approach will be adopted. We will consider alternatives to central differencing.

A variety of one-sided discretization methods have been proposed for the (unsteady) Euler equations. The approach that will be adopted in this paper is the flux vector splitting method of Steger and Warming [15] . In this method, use is made of the fact that the flux functions $E(Q)$ and $F(Q)$ (as well as $\hat{E}$ and $\hat{F}$) are homogeneous of degree one in Q, i.e. $E(aQ) = aE(Q)$, $F(aQ) = aF(Q)$ for any scalar a. Using Euler's theorem on homogeneous functions, one can write $E(Q) = AQ$ and $F(Q) = BQ$ where $A = A(Q) = \partial E/\partial Q$ and $B = B(Q) = \partial F/\partial Q$. Then one uses the fact that A and B are diagonalizable with real eigenvalues to write $X^{-1}AX = \Lambda = \Lambda^+ + \Lambda^-$ where $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_4)$, $\Lambda^+ = \mathrm{Diag}(\lambda_1^+, \ldots, \lambda_4^+)$, $\lambda_i^+ = \max(\lambda_i, 0)$, and $\lambda_i^- = \min(\lambda_i, 0)$; the B term is handled similarly. Define $A^\pm = X\Lambda^\pm X^{-1}$ and similarly for $B^\pm$. Finally, define $E^\pm = A^\pm Q$ and $F^\pm = B^\pm Q$. One gets $\hat{E}^\pm$ and $\hat{F}^\pm$ by starting with $\hat{A}$ and $\hat{B}$, which are linear combinations of A and B. The system (2.4) is then equivalent to

$$\frac{\partial \hat{E}^+}{\partial \xi} + \frac{\partial \hat{E}^-}{\partial \xi} + \frac{\partial \hat{F}^+}{\partial \eta} + \frac{\partial \hat{F}^-}{\partial \eta} = 0. \tag{3.3}$$

One problem with this formulation of flux vector splitting is the discontinuity in slope of λ_i^+. Following a suggestion of Steger [16] , an attempt was made

to remedy this by altering the definition of λ_i^+ to $\lambda_i^+ = (\lambda_i + \sqrt{\lambda_i^2 + \epsilon^2})/2$, and $\lambda_i^- = \lambda_i - \lambda_i^+$, where ϵ is a "small" number; in this work $\epsilon = .03$ was used. For the time-dependent problem in one space dimension (where $\partial Q/\partial t$ is added to the left-hand side of Eq. (2.4)), the theory of characteristics indicates that backward differencing is appropriate for the "+" terms, while forward differencing is appropriate for the "—" terms. We will adopt the same spatial differencing scheme given by this approach. For the steady-state problem we will therefore use the one-sided differencing suggested by the theory of characteristics. Defining δ^b and δ^f as backward and forward difference operators, respectively, (both first-order and second-order operators will be considered), the nonlinear discrete equations become

$$\delta_\xi^b \hat{E}^+ + \delta_\xi^f \hat{E}^- + \delta_\eta^b \hat{F}^+ + \delta_\eta^f \hat{F}^- = 0, \tag{3.4}$$

where appropriate analytical and numerical boundary conditions have to be adjoined to complete the specification of the problem. Boundary conditions will be considered later. After a (local or global) linearization of the discrete equations (3.4) we have a linear system. The linear system is of the form (for a global linearization)

$$\begin{aligned} \delta_\xi^b[\tilde{A}^+(Q^*)\Delta Q] + \delta_\xi^f[\tilde{A}^-(Q^*)\Delta Q] \\ + \delta_\eta^b[\tilde{B}^+(Q^*)\Delta Q] + \delta_\eta^f[\tilde{B}^-(Q^*)\Delta Q] \\ = \text{rhs}, \end{aligned} \tag{3.5}$$

where $\tilde{A}^\pm = \partial \hat{E}^\pm / \partial Q$ and similarly for $\tilde{B}^\pm$. Here $Q^* = Q_{jk}^*$ is fixed and $\Delta Q = \Delta Q_{jk}$ is the unknown. The right-hand side is just the negative of the left-hand side of Eq. (3.4) (evaluated at Q^*). (We note in passing that $\tilde{A}^+ \neq A^+$ and similarly for $\tilde{A}^-$ and $\tilde{B}^\pm$, but it is known that the eigenvalues of $\tilde{A}^+$ and $\tilde{B}^+$ are nonnegative while those of $\tilde{A}^-$ and $\tilde{B}^-$ are nonpositive.) A local linearization of (3.4) (replacing Q_{jk}^* by $Q_{jk}^* + \Delta Q_{jk}$ and expanding

in ΔQ_{jk}) gives the equation

$$\text{const} \cdot (|\tilde{A}(Q^*)| + |\tilde{B}(Q^*)|)\Delta Q_{jk} = \text{rhs}, \tag{3.6}$$

where const = 1 for first-order differencing, = 3/2 for second-order differencing, and $|\tilde{A}| = \tilde{A}^+ - \tilde{A}^-$, and similarly for $\tilde{B}$. (The first-order one-sided difference operators are defined by $\delta^b u_j = u_j - u_{j-1}$, $\delta^f u_j = u_{j+1} - u_j$; second-order one-sided difference operators are defined by $\delta^b u_j = (3u_j - 4u_{j-1} + u_{j-2})/2$, $\delta^f u_j = (-3u_j + 4u_{j+1} - u_{j+2})/2$.)

There seems to be no natural way to order the scalar entries of Q to allow the use of a scalar Gauss-Seidel algorithm. On the other hand, it is very natural to consider the use of block Gauss-Seidel (or "collective" Gauss-Seidel), which simply means changing all 4 unknowns q_1, q_2, q_3, q_4 at a given mesh point (j, k) simultaneously. If we consider a block Gauss-Seidel algorithm for (3.5), we find a 4-by-4 linear system with coefficient matrix = const $\cdot$ $(|\tilde{A}(Q)| + |\tilde{B}(Q)|)$. It can be proved (J.-A. Desideri, personal communication) that this matrix is nonsingular provided that $u \neq 0 \neq v$, i.e., provided we are not at a stagnation point. (This also shows that the 4-by-4 linear system of (3.6) is solvable.) Thus (collective) Gauss-Seidel is a feasible relaxation scheme.

The quality of a relaxation scheme as a component of a multigrid algorithm can be evaluated by local mode analysis. This analysis is based on the premise that the most important property of a multigrid relaxation scheme is that it rapidly damp high frequency components of the error. It is assumed that low frequency components of the error will be efficiently reduced on coarser grids. Thus one assumes constant coefficients, ignores boundaries, and expands the error in a discrete Fourier series. High frequencies are defined as those Fourier components which oscillate rapidly in at least one direction.

Consider collective Gauss-Seidel for Eq. (3.5) at a grid point (j, k). If one orders the grid points "lexicographically" (left to right, bottom to top) and

considers first-order difference operators, one gets the relaxation scheme

$$\begin{aligned} &-\tilde{A}^{+}(Q_{j-1,k})\Delta Q^{n+1}_{j-1,k} - \tilde{B}^{+}(Q_{j,k-1})\Delta Q^{n+1}_{j,k-1} \\ &+|\tilde{A}(Q_{j,k})|\Delta Q^{n+1}_{j,k} + |\tilde{B}(Q_{j,k})|\Delta Q^{n+1}_{j,k} \\ &+\tilde{A}^{-}(Q_{j+1,k})\Delta Q^{n}_{j+1,k} + \tilde{B}^{-}(Q_{j,k+1})\Delta Q^{n}_{j,k+1} \\ &\qquad = \text{rhs}, \end{aligned} \tag{3.7}$$

where n is the iteration index. This is a 4-by-4 linear system to be solved for $\Delta Q^{n+1}_{j,k}$, since $\Delta Q^{n+1}_{j,k-1}$ and $\Delta Q^{n+1}_{j-1,k}$ are already known. For local mode analysis one only needs to consider the homogeneous equations, so take rhs $=$ 0. To do local mode analysis, assume that Q is constant, and assume a Fourier mode for ΔQ: take $\Delta Q^{n}_{j,k} = \lambda^{n} e^{ij\theta} e^{ik\phi} U^{0}$ where U^0 is a constant 4-vector. Here θ and ϕ range from $-\pi$ to π, and the high frequencies are defined to be those modes for which $\pi/2 \leq \max(|\theta|, |\phi|) \leq \pi$. Insert this form of ΔQ in Eq. (3.7) and get a generalized eigenvalue problem for λ; solve this problem, obtaining $\lambda_m = \lambda_m(\theta, \phi)$, $m = 1, 2, 3, 4$. The smoothing rate of the relaxation scheme is defined as

$$\mu = \max_{\pi/2 \leq \max(|\theta|,|\phi|) \leq \pi} (\max_{m} |\lambda_m(\theta, \phi)|).$$

The generalized eigenvalue problem can be solved numerically (the matrices $\tilde{A}^{\pm}$ and $\tilde{B}^{\pm}$ are quite complicated). The following results are presented. One must fix some values for the entries of Q. Two cases were considered, corresponding to subsonic and supersonic flow. For the subsonic case, a representative set of flwo variables (corresponding to freestream conditions) was $u = .8$, $v = 0$, $\rho = 1$, $c = 1$. For the supersonic case, a reasonable choice is to take $u = 1.1$ and the other variables were as in the subsonic case. Assuming independent coordinate stretching (i.e., $\xi = \xi(x)$ and $\eta = \eta(y)$), the only remaining parameter is the ratio x_ξ / y_η (which corresponds to a mesh aspect ratio, width over height of mesh cell). We give the smoothing rate μ

as a function of x_ξ/y_η for both first-order and second-order differencing (see Fig. 3).

From Fig. 3 one can conclude that Gauss-Seidel relaxation in conjunction with three-point second-order one-sided differencing is unacceptable as a relaxation scheme (poor smoothing rate, perhaps even divergence), whereas first-order one-sided differencing in conjunction with Gauss-Seidel gives an acceptable smoothing rate. Further investigation revealed that the second-order scheme is acceptable only if $\min(u/c, v/c) > .5$ (roughly). To help understand this result, consider the one-dimensional case, where Gauss-Seidel with the natural ordering is applied to the linear system

$$\delta^b(\tilde{A}^+ \Delta Q) + \delta^f(\tilde{A}^- \Delta Q) = 0. \tag{3.8}$$

Now $\tilde{A}^\pm$ are 3-by-3 matrices. Fig. 4 gives the smoothing rate μ as a function of u/c for first and second-order differencing.

From Fig. 4 we see that with first-order differencing, the smoothing rate is less than 1 for $u > -c$, whereas for second-order differencing the smoothing rate is greater than 1 unless $u > c/2$, approximately. One concludes from these results that Gauss-Seidel with second-order differencing is unacceptable as a relaxation scheme. Consider a general one-dimensional constant-coefficient problem $(AQ)_x = 0$ where A is a diagonalizable n-by-n matrix. Local mode analysis applied to first-order flux vector splitting leads to the equation

$$\det((e^{i\theta} - \lambda)A^- + \lambda(1 - e^{-i\theta})A^+) = 0.$$

If A is diagonal and $A^- \neq 0$, then $\lambda = -e^{i\theta}$ is a solution for any value of θ, hence the smoothing rate of the relaxation scheme is at least 1. Local mode analysis applied to second-order flux vector splitting leads to

$$\det\Big((2e^{i\theta} - \frac{1}{2}e^{2i\theta} - \frac{3}{2}\lambda)A^- + \lambda(\frac{3}{2} - 2e^{-i\theta} + \frac{1}{2}e^{-2i\theta})A^+\Big) = 0.$$

Again, if A is diagonal and $A^- \neq 0$, one solution is $\lambda = \frac{1}{3}e^{i\theta}(4 - e^{i\theta})$, which has modulus greater than 1 for $0 < \theta < 2\pi$, and so local mode analysis gives a smoothing rate greater than 1.

Another analysis technique can be used to study the behavior of a proposed relaxation scheme. The idea for this technique goes back to Garabedian [17] and is based on the concept of artificial time. One considers the relaxation scheme as a time-accurate approximation (with iteration index as time) to some time-dependent operator L, writes down the operator L, and studies its properties. If all solutions of the partial differential equation $Lu = f$ approach steady state one expects the relaxation scheme to converge, while if the partial differential equation has a nondecaying solution then the relaxation scheme will likely not converge. When this analysis technique was applied to the one-dimensional Euler equations with first-order spatial differencing, (assuming constant spatial coefficients) it was found that L was given by

$$LQ = -\frac{1}{2}\Delta t^2 \tilde{A}^- Q_{tt} - \Delta t \tilde{A}^- Q_t + \Delta t \Delta x \tilde{A} Q_{tx} - \frac{1}{2}\Delta x^2 |\tilde{A}| Q_{xx} + \Delta x A Q_x.$$

The mode $Q = e^{\lambda t} e^{i\omega x}\hat{q}$ satisfies $LQ = 0$ if

$$\det(-\frac{1}{2}\Delta t^2 \lambda^2 \tilde{A}^- - \Delta t \tilde{A}^- \lambda + \Delta t \Delta x \lambda i\omega \tilde{A} + \frac{1}{2}\Delta x^2 \omega^2 |\tilde{A}| + i\omega \Delta x A) = 0.$$

We see that if $\tilde{A}^- = 0$ then $\mathrm{Re}(\lambda) = 0$, while if $\tilde{A}^+ = 0$ then there are two roots λ, one with real part 0, the other with negative real part. (The case $\tilde{A}^- \neq 0 \neq \tilde{A}^+$ is too messy to do by inspection.) For second-order space differencing, the operator L turns out to be

$$LQ = -\frac{3}{2}\Delta t \tilde{A}^- Q_t + \Delta x A Q_x - \frac{3}{4}\Delta t^2 \tilde{A}^- Q_{tt} + \Delta x \Delta t \tilde{A}^+ Q_{xt}.$$

If $\tilde{A}^+ = 0$ there are modes $Q = e^{\lambda t} e^{i\omega x}\hat{q}$ with $LQ = 0$ and $\mathrm{Re}(\lambda) > 0$, hence there are growing solutions, and one does not expect the difference scheme to be convergent.

In sum, both local mode analysis and the Garabedian artificial time analysis predict Gauss-Seidel will not converge for second order spatial differencing. Gauss-Seidel must then be rejected as the relaxation scheme.

For fluid flow problems it is natural to attempt to align the relaxation direction with the flow direction. In problems with systems of equations or where the flow direction is not known *a priori* one can try a symmetric relaxation scheme, so consider symmetric Gauss-Seidel (SGS). For SGS each relaxation step is composed of two half-steps, the first being a Gauss-Seidel sweep in the direction of increasing j and k, the second a Gauss-Seidel sweep in the direction of decreasing j and k. For the one-dimensional problem with first-order differencing the artificial-time analysis leads to the operator

$$LQ = -\Delta t \tilde{A}^{+} |\tilde{A}|^{-1} \tilde{A}^{-} (Q_t + \frac{1}{2}\Delta t Q_{tt}) + \Delta x |\tilde{A}| (Q_x + \Delta t Q_{xt} - \frac{1}{2}\Delta x Q_{xx}).$$

Now one can show that for either of the cases $\tilde{A}^{+} = 0$ or $\tilde{A}^{-} = 0$, any solution $Q = e^{\lambda t} e^{i\omega x} \hat{q}$ of $LQ = 0$ satisfies $\mathrm{Re}(\lambda) = 0$. Thus one expects SGS to be convergent. A similar result holds for the algebraically messier case of second-order spatial differencing. Local mode analysis applied to SGS gives results shown in Fig. 5. These results indicate that SGS has reasonably good smoothing behavior (the smoothing deteriorating as the mesh stretching becomes more severe).

One could attempt to improve the smoothing rate by adopting a line relaxation algorithm (say, symmetric line-Gauss-Seidel, SLGS). This indeed will improve the smoothing rate but gives a linear system with block pentadiagonal (in the case of second-order space differencing) matrix to solve on each line. The expense of the block pentadiagonal solver almost outweighs the gain due to better smoothing rate. Specifically, for the Newton-multigrid algorithm to be discussed in the next section, the SGS algorithm cost (with non-optimal coding) about 1000 floating point operations per grid point per

sweep. The SLGS algorithm cost about 4000 floating point operations per grid point per sweep, and gave a smoothing rate about 5 times better than SGS. Thus SLGS will give only a marginal gain in total computing time over SGS. One can attempt to make further modifications, such as approximating the block pentadiagonal matrix by a block tridiagonal matrix. There does not seem to be much to be gained, however, by so doing. Thus the SGS relaxation scheme was adopted.

§4. Nonlinearity.

The nonlinearity of the system (2.4) and its discretization (3.4) poses a difficult challenge to numerical methods. From a practical viewpoint, how well one handles the nonlinearity is perhaps the most important criterion in determining the success or failure of an algorithm. Two algorithms will be discussed in this section, the Newton-multigrid algorithm and the FAS (Full Approximation Storage)-Newton algorithm.

The Newton-multigrid algorithm is: apply Newton's method to (3.4), using multigrid to solve the linear problems that arise at each stage of Newton's method. If (3.4) is written as $\mathcal{F}(Q) = 0$ Newton's method is, with an initial guess Q^0,

$$\begin{aligned} D\mathcal{F}(Q^n)(\Delta Q^n) &= -\mathcal{F}(Q^n) \\ Q^{n+1} &= Q^n + \Delta Q^n. \end{aligned} \tag{4.1}$$

In (4.1), $D\mathcal{F}(Q^n)$ is the Jacobian matrix of the nonlinear function $\mathcal{F}$. It is the linear system $D\mathcal{F}(Q^n)(\Delta Q^n) = -\mathcal{F}(Q^n)$ which is solved via multigrid.

Some of the favorable aspects of this algorithm are:

1. Newton's method has been intensively studied and is well-understood;
2. Convergence is guaranteed provided the initial guess is sufficiently close to the (numerical) solution;
3. Multigrid is only used to solve linear problems (and multigrid theory is better understood for linear problems);
4. The matrices $\tilde{A}^{\pm}$ and $\tilde{B}^{\pm}$, which are expensive to compute, are held con-

stant in each Newton step, thus the expense of recomputing them is avoided;
5. For problems with shocks, the solution ΔQ^n to the linear problem may be smooth after the shock has set up, so multigrid should work well;
6. "Global Newton" methods exist, which guarantee convergence from an arbitrary initial guess;
7. A good theory exists for "nice" elliptic problems [18] , which indicates that with the nested multigrid algorithm (or full multigrid, FMG), one Newton step per level is sufficient (asymptotically).

Some of the disadvantages of the Newton-multigrid algorithm are:
1. Full Newton (solving each linear problem to completion) is too expensive, and one must resort to a modified Newton algorithm, which raises the thorny question of stopping criteria for the linear problems;
2. Any time spent on solving linear problems too accurately is wasted;
3. Newton's method may not converge (if the starting guess is not sufficiently accurate–examples of this have arisen in practice);
4. The solution of the linear problem may be singular (not smooth) even if the coefficients are smooth, which can cause trouble for both the multigrid solver and the Newton method;
5. For the problem at hand, some sort of underrelaxation must be employed, or else the pressure or density at a grid point can become negative after the Newton update and the code cannot continue;
6. In practice, quadratic convergence is irrelevant, since if one is "close enough" for quadratic convergence to take over then one already has a good enough answer (for engineering purposes);
7. The proper (global) linearization of the equation and boundary conditions can be quite difficult, especially if the boundary conditions are nonlocal and/or nonlinear.

The FAS-Newton algorithm is: attack the nonlinear system (3.4) directly,

using multigrid in the FAS mode (Brandt [19]) as the solver. For the relaxation sweeps, linearize locally and take one step of Newton's method at each grid point (see (3.6)). This method is easier to program than the Newton-multigrid method and uses less arithmetic in the relaxation scheme (there are fewer matrix-vector multiplies).

Disadvantages of this method are:

1. Updating the "tilde" matrices $\tilde{A}^{\pm}$ and $\tilde{B}^{\pm}$ is expensive (it costs about 814 floating point operations and 18 square roots to compute $\tilde{A}^{\pm}$, $\tilde{B}^{\pm}$, $\hat{E}^{\pm}$, and $\hat{F}^{\pm}$ at a given grid point);
2. There is not much theory to guide in the construction of a good multigrid algorithm (for example, when to switch grids);
3. Coarse grids don't do a good job of representing solutions with shocks;
4. It is not clear that the relaxation scheme (one Newton step at each grid point) smooths the error.

In sum, both the Newton-multigrid and FAS-Newton algorithms have advantages and disadvantages. The Newton-multigrid algorithm as coded cost about 1020 floating point operations per grid point per relaxation step (not counting the work required to update the tilde matrices), while the FAS-Newton algorithm cost about 2114 floating point operations plus 36 square roots per grid point per relaxation step. Results for both algorithms will be given in the next section.

§5. Sample Calculations.

A code was written using SGS relaxation and either Newton-multigrid or FAS-Newton. For the Newton-multigrid algorithm, the restriction operator was taken to be straight (unweighted) injection, while for FAS-Newton the restriction operator was full-weighted restriction. In both cases the interpolation operator was bilinear interpolation. Useful assistance in coding was provided by the code of Brandt and Cryer [20] . The unknowns were taken to be $\{Q_{jk} : 2 \leq j \leq \text{jmax} - 1,\ 2 \leq k \leq \text{kmax} - 1\}$. Values for jmax

ranged from 65 to 161, values for kmax ranged from 25 to 33. For the results to be shown, jmax was 145 and kmax was 33. Fig. 6 shows the grid.

Boundary conditions for the numerical scheme were as follows: all quantities (q_1, q_2, q_3, q_4) held fixed at inflow; all quantities extrapolated on outflow (using zeroth order extrapolation); on the airfoil, normal component of velocity held equal to 0, while density, tangential velocity component, and pressure were extrapolated using zeroth order extrapolation; on the wake cut, all variables defined by linear averaging. Notice that the boundary condition involving the pressure is nonlinear in terms of the variables (q_1, q_2, q_3, q_4), while the boundary condition in the wake is nonlocal in (ξ, η) space. Both these factors require that great care be taken in the linearization (either global or local) near the airfoil and wake cut.

Three-point one-sided second-order differencing was used everywhere except near the boundaries, where two-point first-order one-sided differencing was employed in the direction normal to the boundary.

Figure 7 shows a sequence of plots of pressure coefficient as the Newton-multigrid iteration proceeded. The Newton-multigrid iteration had to be modified in order to produce a convergent process. The update step of (4.1) was modified to

$$Q_{jk}^{n+1} = Q_{jk}^{n} + \omega_{jk} \Delta Q_{jk}^{n}$$

where ω_{jk} was chosen to be the largest number (less than or equal to 1) of the form 2^{-m} such that the density and pressure at grid point (j, k) and iteration index $n+1$ were positive (thus the Newton update was point-underrelaxed). Also, the strategy for solving the linear problems was modified to force the ℓ_2 norm of the dynamic residuals to decrease by a given factor (4 in this case) before the linear problem was regarded as solved. This has the effect of ensuring that the linear problem is solved reasonably accurately at each Newton step. The Newton-multigrid process then converged, with initial

guess of freestream values for all variables. Multigrid was used in the "Coarse Grid Correction" mode. The sequence shown in Figure 7 seems to be typical (and has been noted for other flow conditions): the flow field sets up fairly rapidly away from the shock, and near the shock strong oscillations appear, which gradually die away. The computations were done on a VAX 11-780; total CPU time was about 3.25 hours for 11 steps. A comparison of the final solution with a solution obtained by a central difference algorithm showed good agreement in the shock locations and shock strengths.

Figure 8 shows a sequence of plots of pressure coefficient for the FAS-Newton algorithm. The flow field is slower to set up, and strong oscillations do not appear near the shock. CPU time for this case was about 2.5 minutes per work unit (1 work unit is defined as 1 relaxation sweep on the finest grid, thus 1 relaxation sweep on the next-finest grid is .25 work units, etc.). Total time for a converged solution was about 1.5 hours.

In conclusion, it has been demonstrated that multigrid methods can solve transonic flow problems with shocks. The choice of differencing scheme is important; a flux-vector splitting scheme was used. A symmetric collective Gauss-Seidel relaxation scheme was adopted. Local mode analysis is a useful tool in analyzing relaxation schemes. Both the Newton-multigrid algorithm and the FAS-Newton algorithm were shown to be viable. It is too early to make a judgment as to the ultimate impact of multigrid on steady transonic Euler equation calculations.

References

1. A. Jameson, "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method", Proc. of AIAA 4th Computational Fluid Dynamics Conference, 122–146, 1979.
2. A. Shmilovich and D. Caughey, "Application of the Multi-grid Method to Calculations of Transonic Potential Flow about Wing-Fuselage Combinations", in **Multigrid Methods**, NASA CP 2202, October 1981.
3. T.L. Holst, "Implicit Algorithm for the Conservative Transonic Full-Potential Equation Using an Arbitrary Mesh", **AIAA J.**, Vol. 17, Oct. 1979, pp. 1038–1045.
4. J. Steinhoff and A. Jameson, "Multiple Solutions of the Transonic Potential Flow Equation", Fifth AIAA Computational Fluid Dynamics Conference, Palo Alto, 1981.
5. A. Jameson, W. Schmidt, and E. Turkel, "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes", AIAA paper 81-1259, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, 1981.
6. J. L. Steger, "Implicit Finite-difference Simulation of Flow about Arbitrary Two-Dimensional Geometries", **J. Comp. Phys.**, vol. 16, 1978, pp. 679–686.
7. T. Pulliam, D. Jespersen, and R. Childs, "An Enhanced Version of an Implicit Code for the Euler Equations", AIAA paper 83-0344, AIAA 21st Aerospace Sciences Meetings, 1983, Reno.
8. J. L. Steger, "A Preliminary Study of Relaxation Methods for the Inviscid Conservative Gasdynamics Equations Using Flux Splitting", NASA Contractor Report 3415, March 1981.
9. R.-H. Ni, "A Multiple Grid Scheme for Solving the Euler Equations", AIAA Paper 81-1025, June 1981.
10. G. Johnson, "Multiple Grid Acceleration of Lax-Wendroff Algorithms", NASA Technical Memorandum 82843, March 1982.
11. S. Agmon, A. Douglis, and L. Nirenberg, "Estimates Near the Boundary for Solutions of Elliptic Partial Differential Equations Satisfying General

Boundary Conditions II", **Comm. Pure and Appl. Math.**, vol. 17, 1964, pp. 35–92.

12. A. Brandt, "Multigrid Solutions to Steady-State Compressible Navier-Stokes Equations", in Proc. 5th Int. Symp. Computing Meth. in Applied Science and Engineering, ed. R. Glowinski and J.L. Lions, 1981.
13. R. E. Childs and T. H. Pulliam, "An Implicit Multigrid Method Applied to the Centrally Differenced Euler Equations", to be presented at the AIAA Computational Fluid Dynamics Conference, Danvers, Mass., July 1983.
14. S. McCormick, The Methods of Kaczmarz and Row Orthogonalization for Solving Linear Equations and Least Squares Problems in Hilbert Space, **Indiana Univ. Math. J.**, vol. 26, pp. 1137–1150, 1977.
15. J. Steger and R. Warming, "Flux Vector Splitting of the Inviscid Gas Dynamic Equations with Applications to Finite-Difference Methods", **J. Comp. Phys.** 40, 1981, pp. 263–293.
16. J.L. Steger, "Implicit Finite Difference Simulation of Inviscid and Viscous Compressible Flow", in **Transonic, Shock, and Multidimensional Flows**, ed. R. E. Meyer, Academic Press, New York, 1982.
17. P. R. Garabedian, "Estimation of the Relaxation Factor for Small Mesh Size", **Math. Tables Aids Comp.**, vol. 10, 1956, pp. 183–185.
18. R. E. Bank and D. J. Rose, "Analysis of a Multilevel Iterative Method for Nonlinear Finite Element Equations", **Math. Comp. 39**, 1982, pp. 453–465.
19. A. Brandt, "Guide to Multigrid Development", in **Multigrid Methods**, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag Lecture Notes in Mathematics 960, 1982.
20. A. Brandt and C. Cryer, "Multigrid Algorithms for the Solution of Linear Complementarity Problems Arising from Free Boundary Problems", Mathematics Research Center Report 2131, 1980.

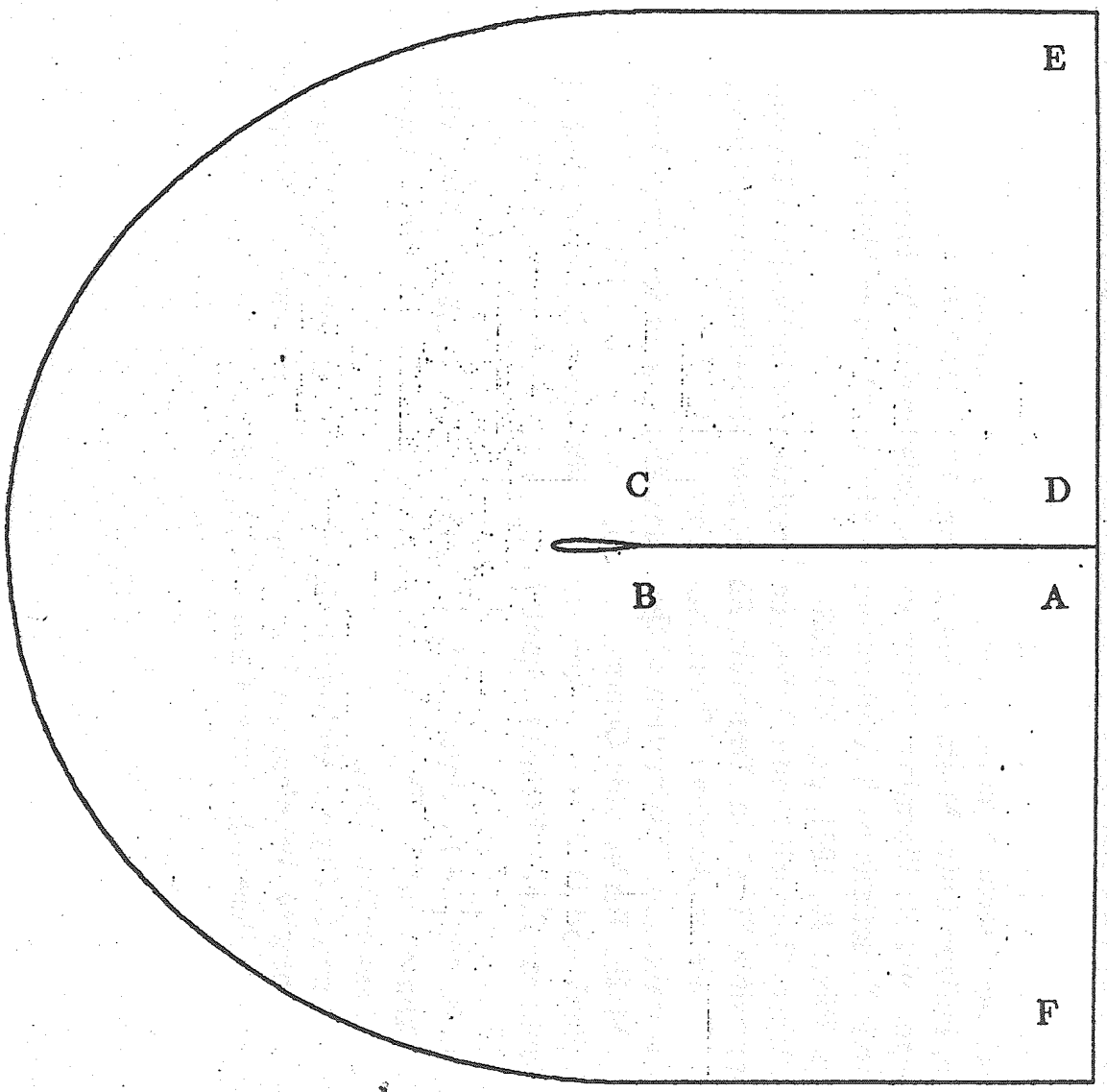


Figure 1.

Physical Domain

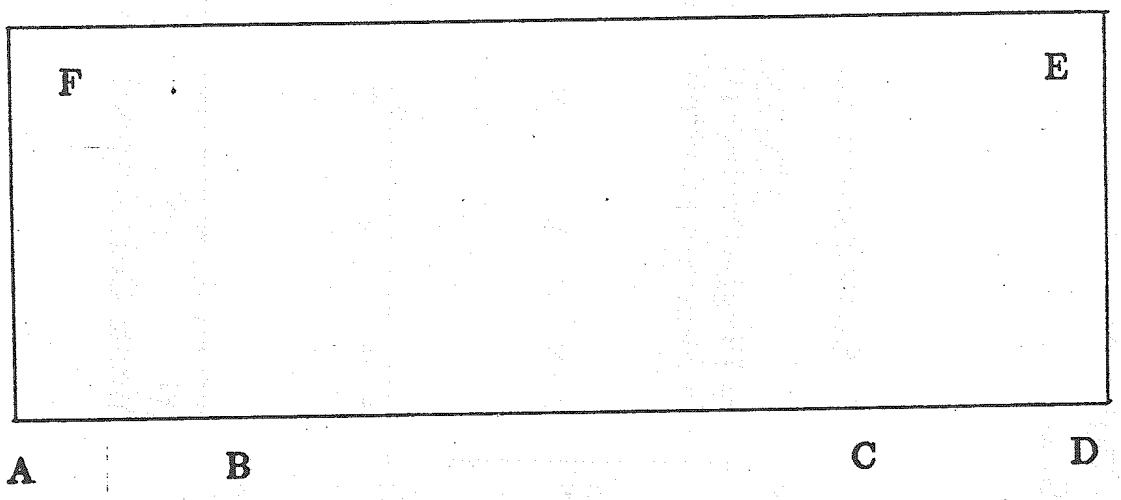


Figure 2.

Computational Domain

Smoothing Rate from Local Mode Analysis
Two–Dimensional Euler Equations
Flux Vector Splitting, Gauss–Seidel Relaxation

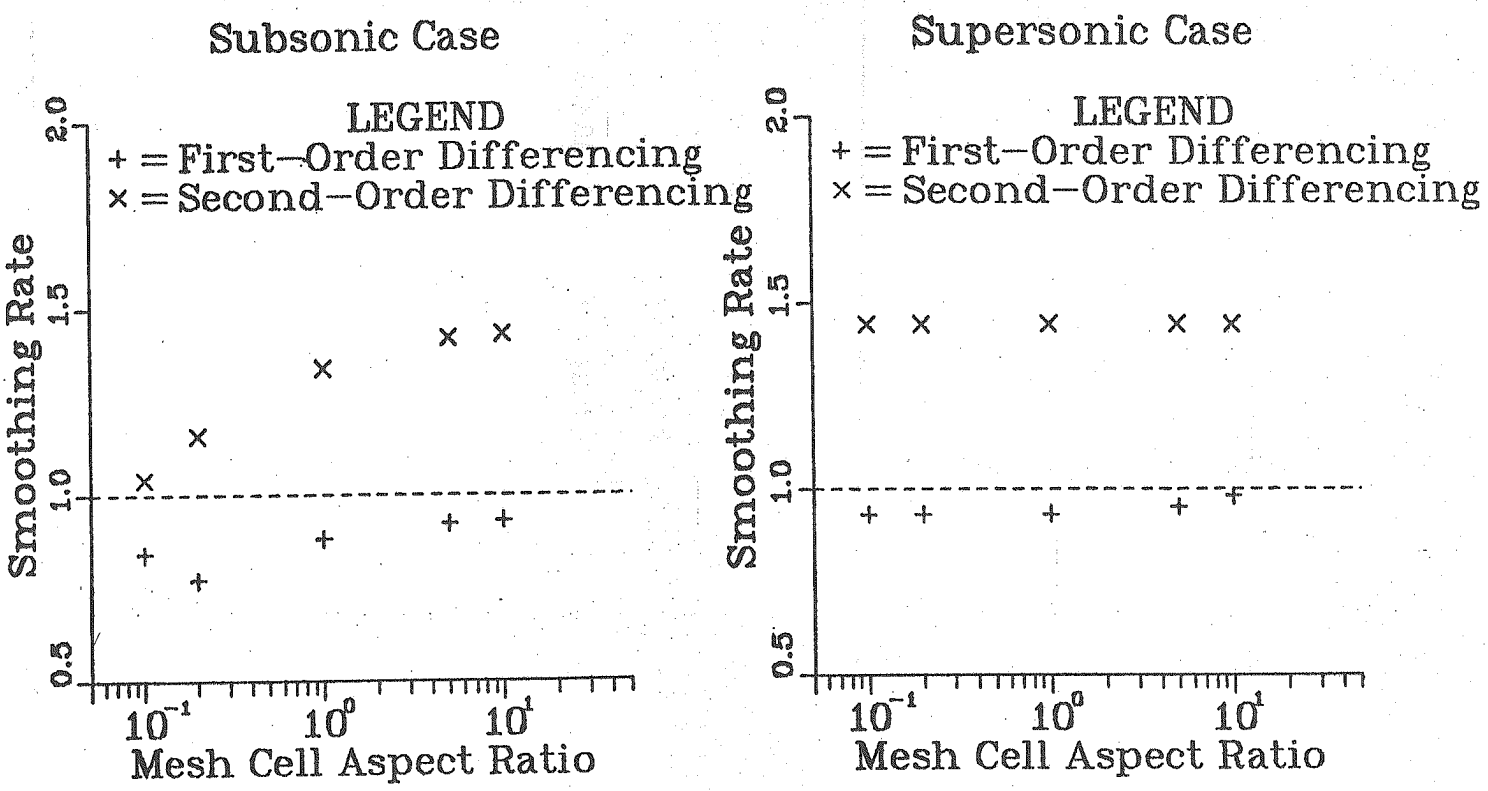


FIGURE 3

Smoothing Rate from Local Mode Analysis
One–Dimensional Euler Equations
Flux Vector Splitting, Gauss–Seidel Relaxation
Smoothing Rate as a Function of u/c

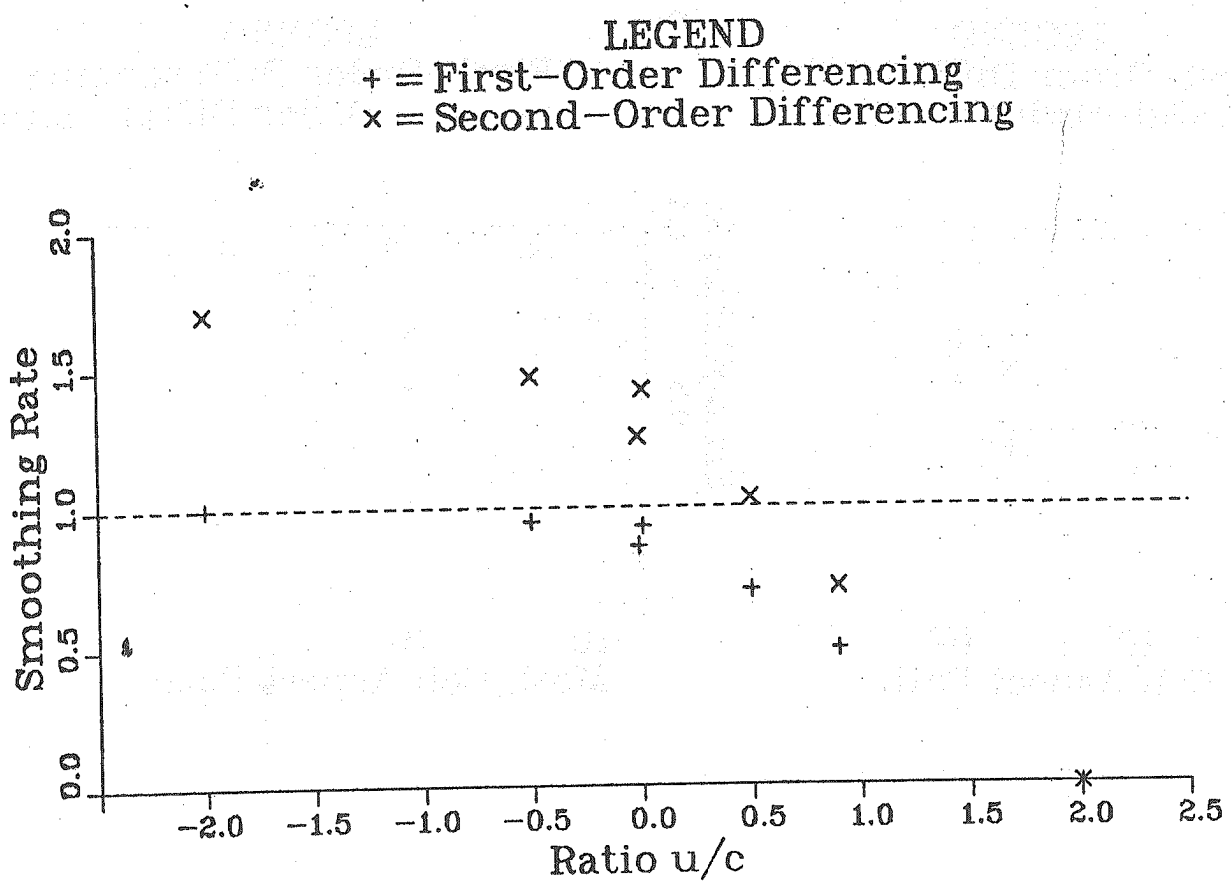


FIGURE 4

Smoothing Rate from Local Mode Analysis
Two–Dimensional Euler Equations
Flux Vector Splitting, Symmetric Gauss–Seidel Relaxation

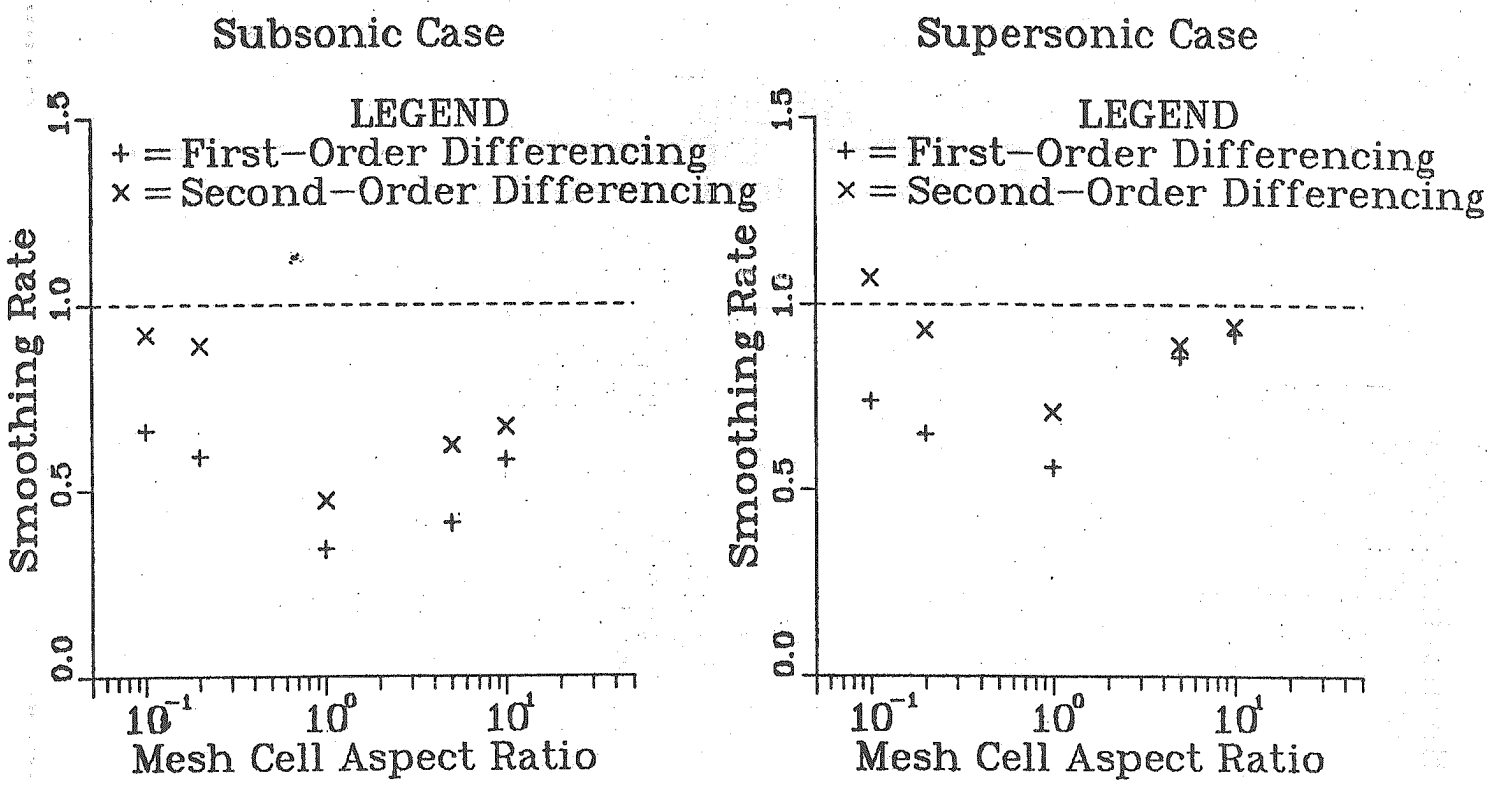


FIGURE 5

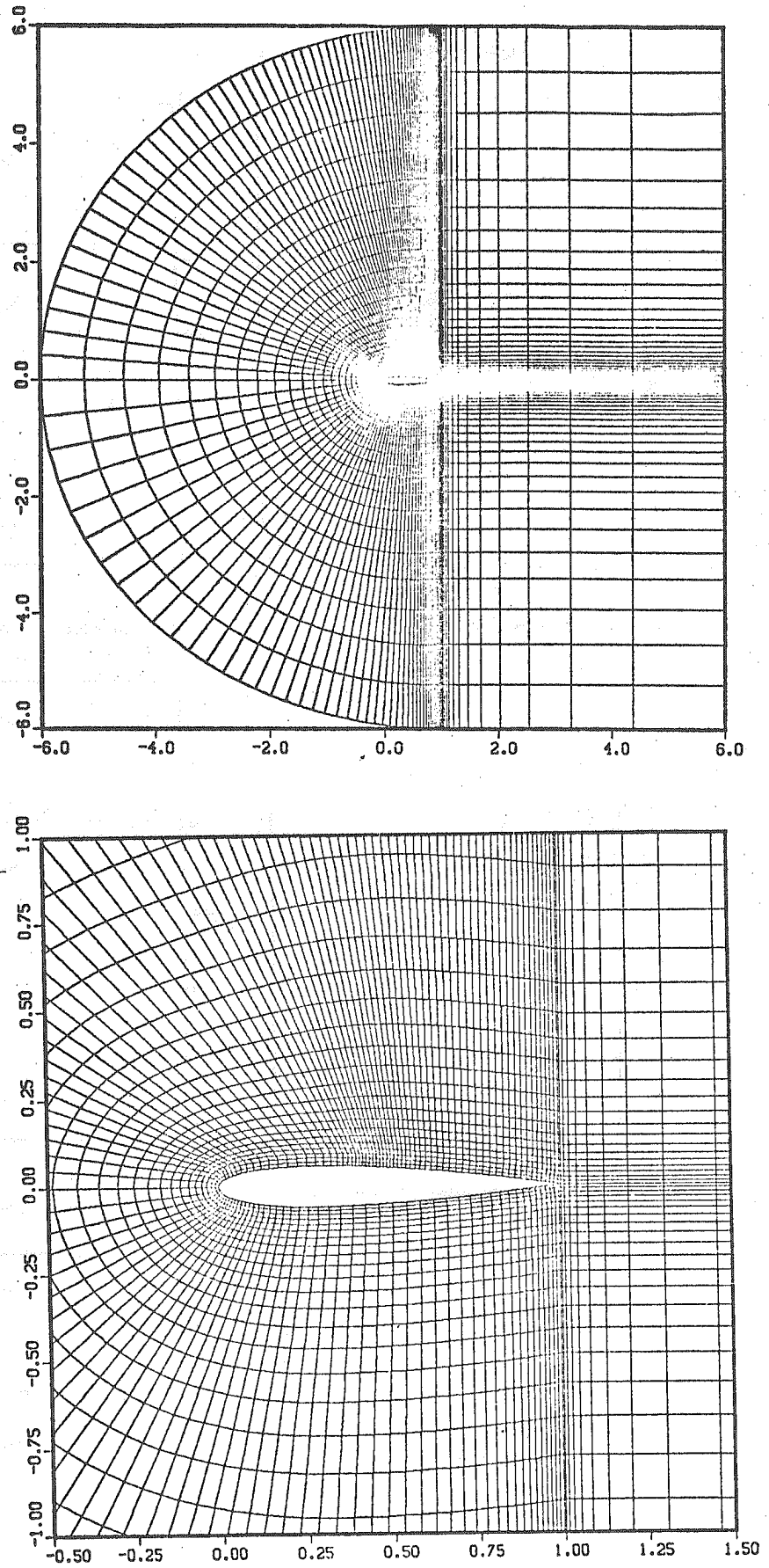
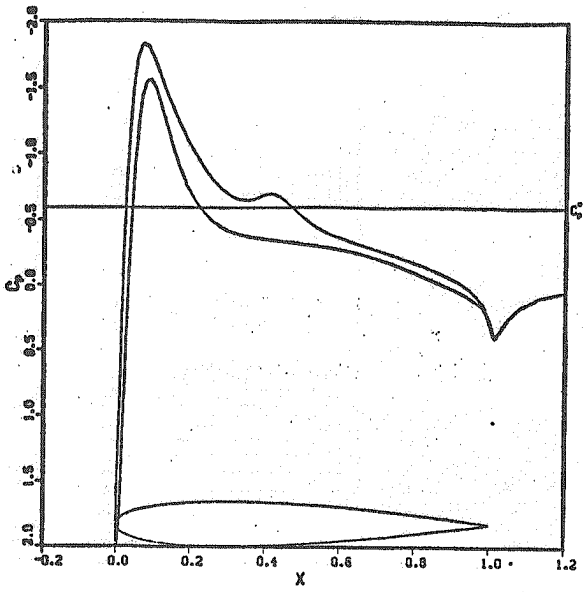


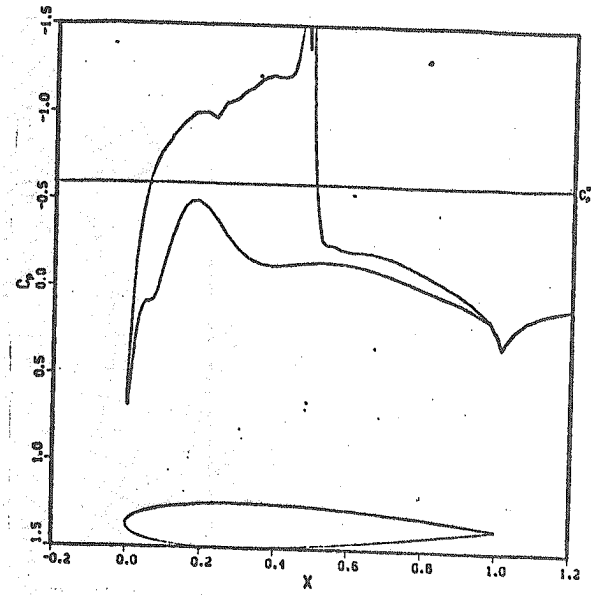
Figure 6

Grid for Airfoil Problem

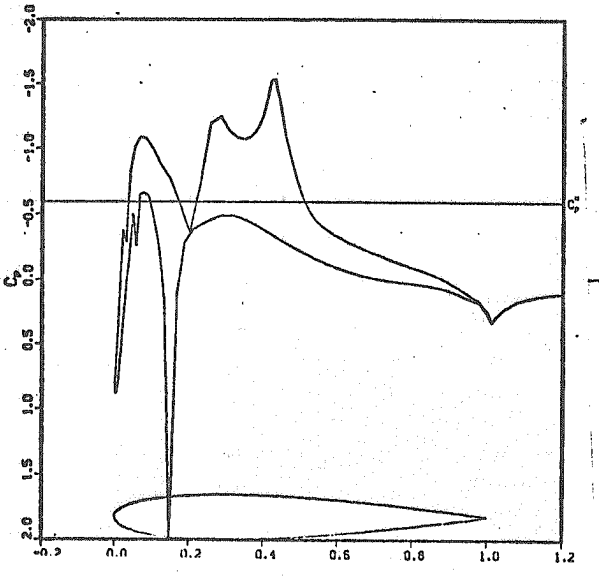
Figure 7: NACA 0012 airfoil, Mach number 0.75, angle of attack 2°



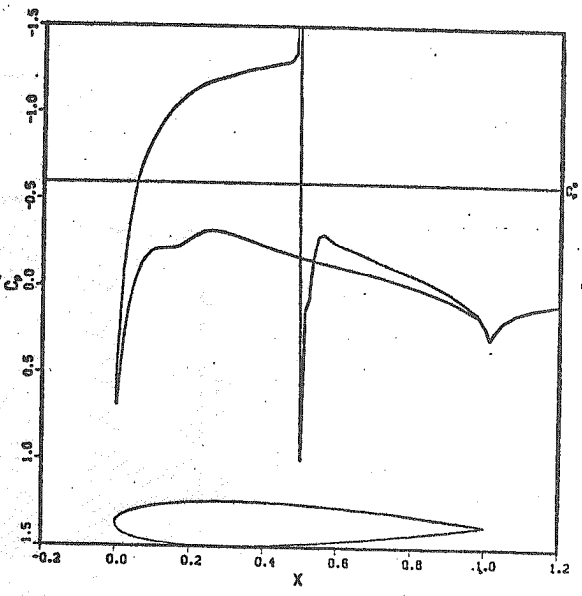
Pressure coefficient after 1 modified Newton step



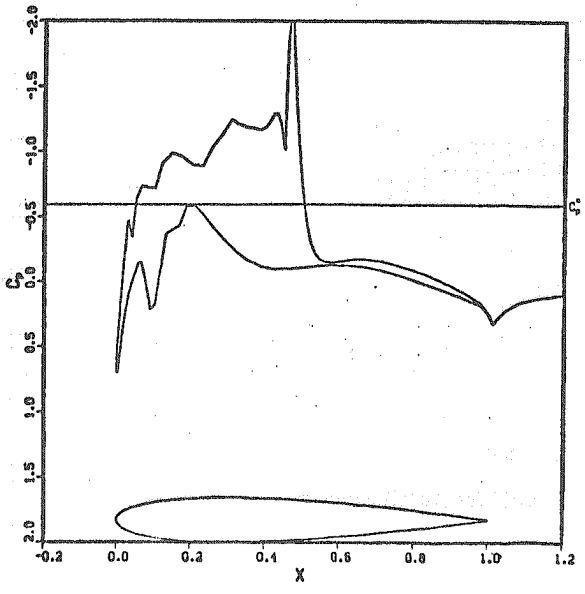
Pressure coefficient after 4 modified Newton steps



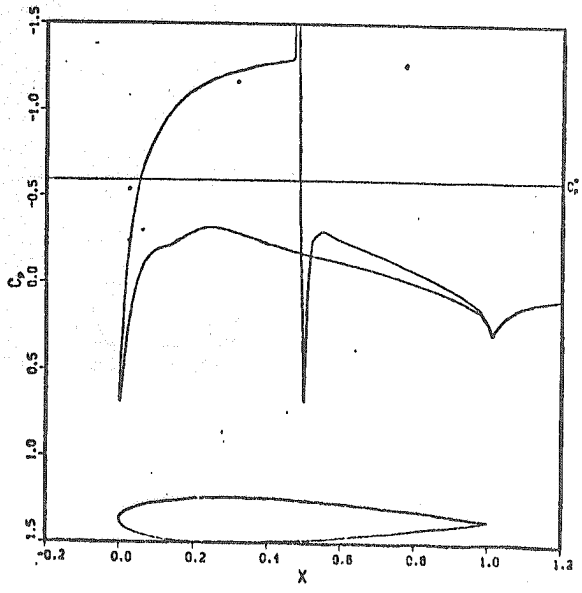
Pressure coefficient after 2 modified Newton steps



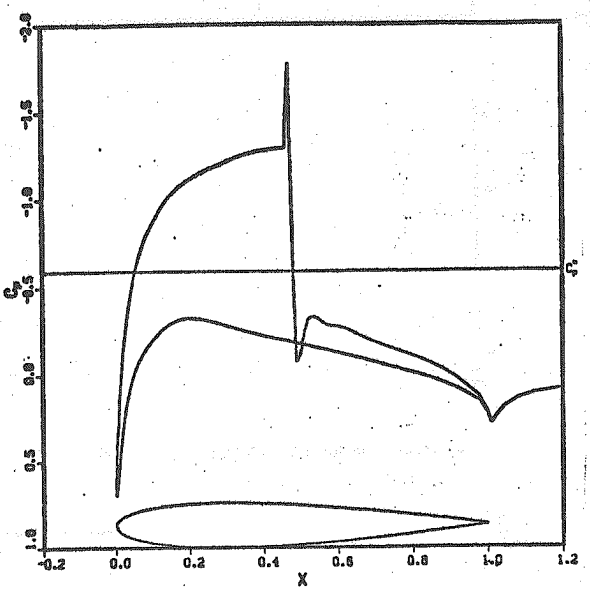
Pressure coefficient after 6 modified Newton steps



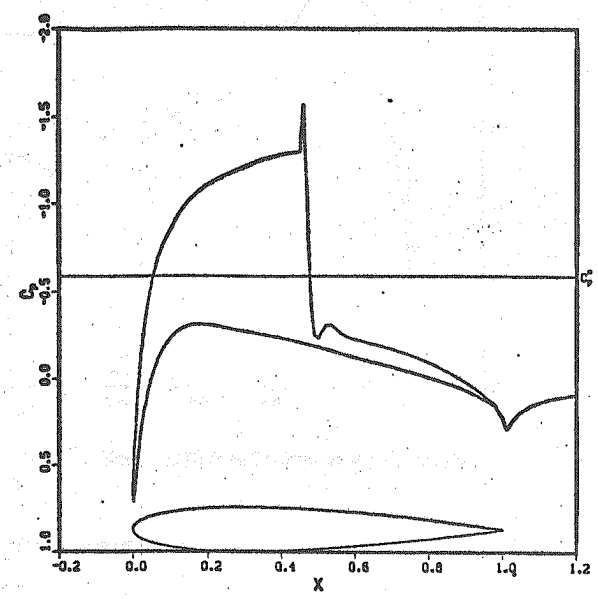
Pressure coefficient after 3 modified Newton steps



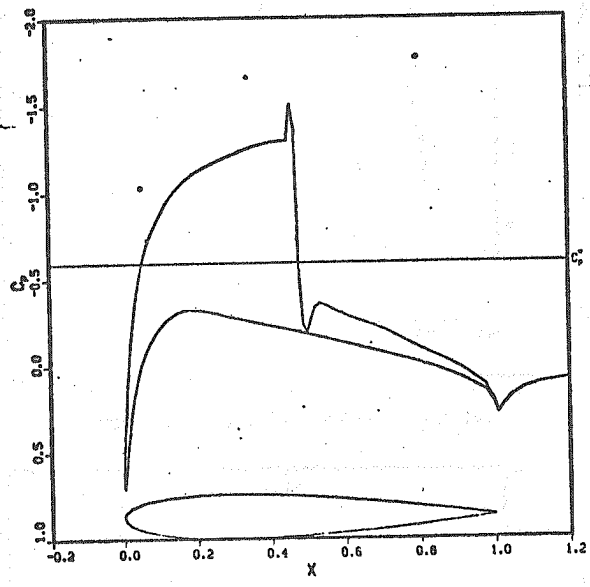
Pressure coefficient after 7 modified Newton st



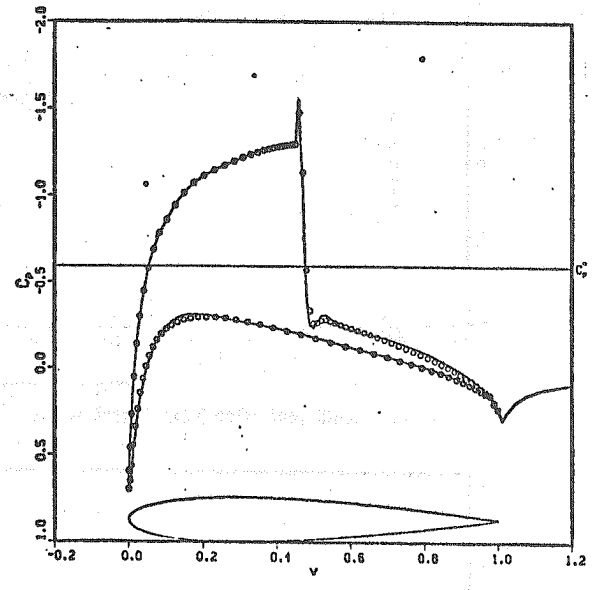
Pressure coefficient after 9 modified Newton steps



Pressure coefficient after 11 modified Newton steps

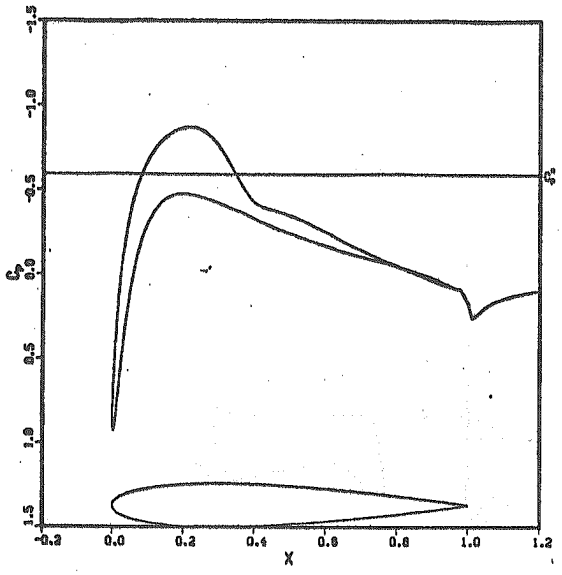


Pressure coefficient after 10 modified Newton steps

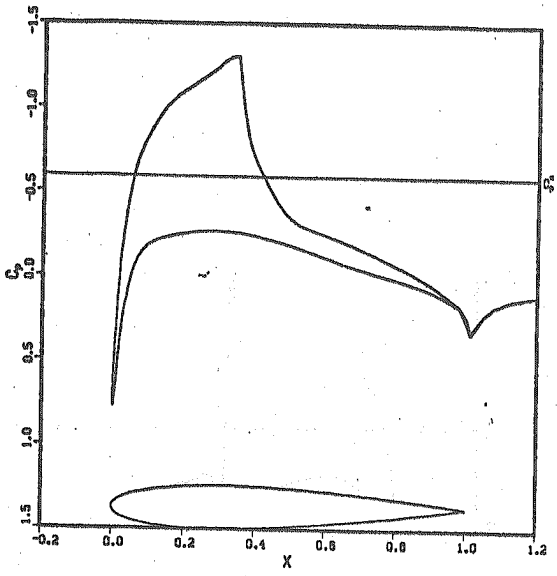


Comparison of pressure coefficient after 11 and after 23 steps

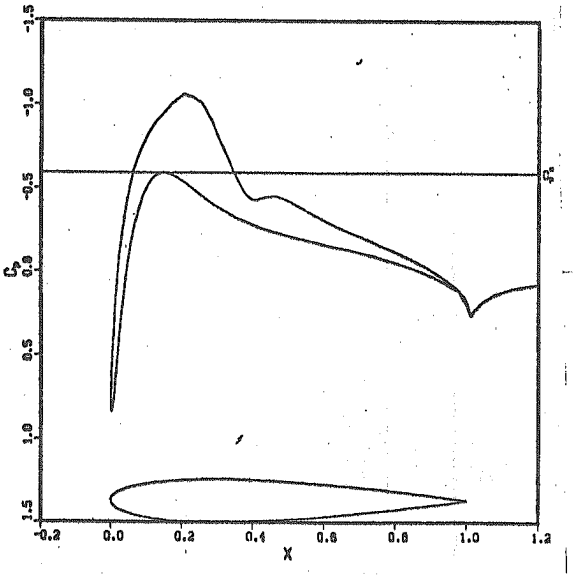
Figure 8: Pressure coefficient for FAS-Newton using 3 grids



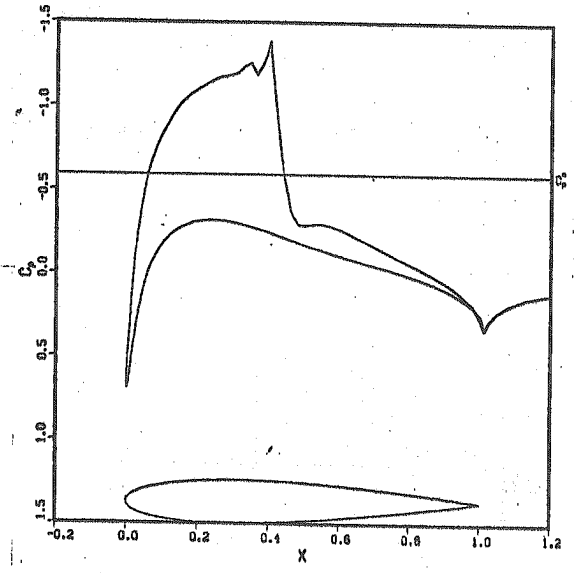
Pressure coefficient after 6.8125 work units



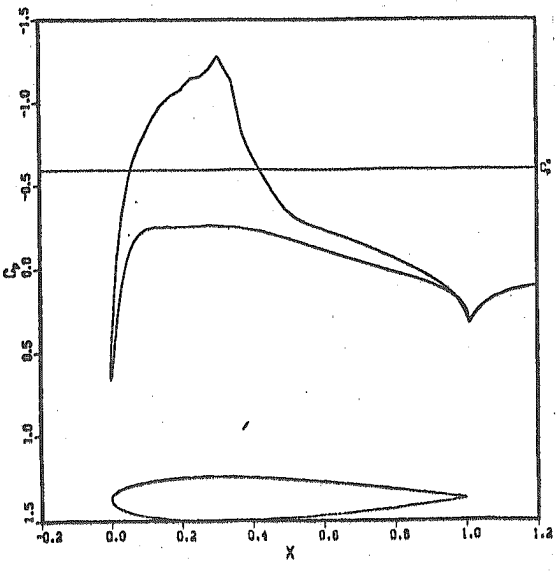
Pressure coefficient after 17.9375 work units



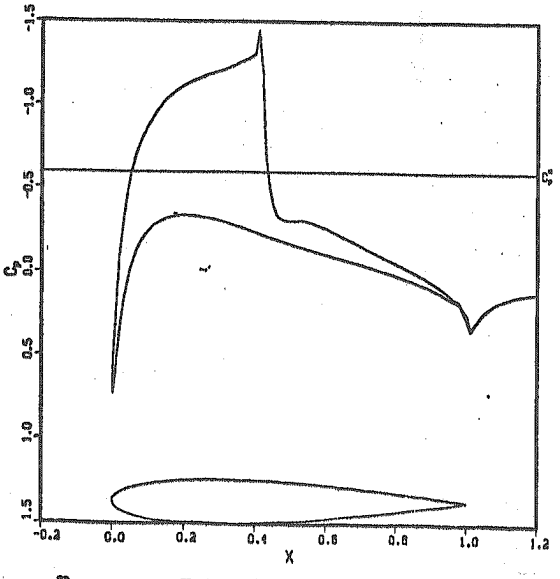
Pressure coefficient after 9.5625 work units



Pressure coefficient after 20.5 work units



Pressure coefficient after 15.9375 work units



Pressure coefficient after 22.5 work units