NAG1-362

Solution of the Euler Equations With Viscous-Inviscid Interaction for High Reynolds Number Transonic Flow Past Wing/Body Configurations

by

Keith Koenig
Associate Professor
Department of Aerospace Engineering
Drawer A
Mississippi State, MS 39762

Final Report
NASA Research Grant
NAG-1-362

Solution of the Euler Equations With Viscous-Inviscid Interaction for High Reynolds Number Transonic Flow Past Wing/Body Configurations

by

Keith Koenig
Associate Professor
Department of Aerospace Engineering
Drawer A
Mississippi State, MS 39762

Final Report
NASA Research Grant
NAG-1-362

Acknowledgements

This work has been supported by the NASA Langley Research Center under the NASA Research Grant NAG-1-362. Mr. Edgar G. Waggoner served as the NASA Technical Officer and has ably directed the focus of this research. Dr. Jim Luckering and Dr. KyleAnderson both of Langley have given valuable help in grid refinements and in mastering the Langley computer system. Dr. David L. Whitfield and Ph.D. candidate Mark Janus, both at Mississippi State University, have very generously provided technical guidance and insight into the workings of the code and have made significant contributions to its refinement. Also at Mississippi State, Lt. Col. Hyun Jin Kim and Mr. Tim Sanford provided valuable assistance in plotting grids and program output. Finally, without the aid of Mrs. Lisa W. Griffin much of this work would not have been possible. For the past three years, as an undergraduate and graduate student, she has struggled with bad phone lines, seemingly endless coding revisions, mysterious operating systems, tedious plotting and strange explicit vector code. Her efforts and abilities are most gratefully appreciated.

List of Symbols

The item in parentheses indicates the equation number or section where the symbol first appears.

a,b,c coefficient matrices of the nonconservative Euler equations, (App. C)

c speed of sound, (9); chord length, (Sect. VI)

e energy per unit volume, (1)

f,g,h flux vectors in Cartesian coordinates, (1)

i,j indices representing ξ,η,ζ directions, (4b)

i,j,k representative curvilinear coordinates, (9)

ℓ,m indices representing elements of the flux Jacobian matrices, (6b)

n normal to cell face, (2c); time level, (5)

p static pressure, (1)

q conserved dependent variables in Cartesian coordinates, (1)

$\hat{q}$ nonconserved dependent variables, (C3)

t time, (1)

u,v,w velocity components in Cartesian coordinates, (1)

u_e^+ boundary layer friction velocity, (Sect. IV)

x,y,z Cartesian coordinates, (1)

x representative element of the dependent variable vector, (23)

A,B,C flux Jacobian matrices, (6b)

A representative element of the coefficient matrices, (21)

CP pressure coefficient with respect to freestream conditions, (Figures)

F,G,H flux vectors in the transformed equations, (2)

I,J,K indices for computational grid lines, (Sect. VI)

H shape factor, (Sect. IV)

I identity matrix, (8b)

J Jacobian matrix of the transformation from Cartesian to curvilinear coordinates, (2)

K representative flux vector (App. C)

$\bar{K}$ representative flux Jacobian matrix (App. C)

M Mach number, (31); $\partial Q/\partial \hat{q}$, (C4)

P eigenvectors of nonconservative Euler equations, (C11)

Q conserved dependent variables in the transformed equations, (2)

R residual, (13)

$Re_{\bar{c}}$ Reynolds number based on mean aerodynamic chord, (Sect. VI)

S cell surface area, (2c)

T transpose, (1); eigenvectors of the conservative Euler equations, (C10)

U,V,W contravariant velocity components, (2)

X representative unknown dependent variable vector, (21)

X^1,X^2 unknown dependent variable vector in the two pass algorithm, (14)

α angle of attack, (Figures)

γ ratio of specific heats, (1)

δ spatial difference operation, (3)

δ^* boundary layer displacement thickness, (Sect. IV)

ξ,η,ζ curvilinear coordinates, (2)

θ contravariant velocity, $k_xu+k_yv+k_zw$, (C7)

κ representative coefficient matrix for the nonconservative Euler equation, (App. C)

λ eigenvalues, (9)

ν cell volume, (2a)

ρ density, (1)

τ time in transformed equations, (2)

ϕ $\frac{\gamma-1}{2}(u^2+v^2+w^2)$, (C7)

Δ	difference operator (usually first order), (2a)
Λ	diagonal matrix of eigenvalues, (C15)
∇	del operator, (2a)
$(\ )_\infty$	freestream conditions, (31)
$(\tilde{\ })$	$(\ )/\lvert\nabla k\rvert$, (C17)
$(\ )_e$	boundary layer edge conditions, (Sect. IV)
$(\vec{\ })$	vector quantity
$(\ )^\pm$	terms associated with positive or negative eigenvalues, (10c)
$(\tilde{\ })$	equivalent incompressible boundary layer terms, (Sect. IV)

I. Introduction

During the mid 1970's a computer program for the solution of the Euler equations was developed by researchers at Sverdrup Technology, Inc. for transonic and supersonic applications. This program, known as ARO-1,[15,16] was an explicit, finite volume formulation based on McCormack's predictor-corrector algorithm. From this code has evolved a long line of Euler equation solvers whose current members now share only the finite volume formulation with the prototype. One of these current programs is the subject of the following presentation.

The code of interest here goes by the (rather colorful) name BROWN MULE. It is a three-dimensional, time accurate, finite volume Euler solver (as its progenitors) with extensive refinements.[1,2,3] The solution scheme is an implicit, upwind, split-flux vector formulation in which the flux vectors are divided into subvectors based on the signs of the eigenvalues. Prudent approximate factorization then leaves only a system of block bidiagonal equations to be solved which is readily accomplished. The upwind scheme used here requires no artificial dissipation and is conditionally stable in three-dimensions. Furthermore, because the present interest is in steady state solutions, local time stepping can be used thereby improving the convergence rate. The resulting program is easy to use, requiring only a minimum of input variables and parameters, and achieves engineering quality answers in reasonably short machine time. A disadvantage is the relatively large memory needed to store the flow variables and split flux vectors, although with the latest generation of supercomputers this is not a serious limitation. Further history and background concerning BROWN MULE and its family may be found in References 1 and 3.

As part of the present research, a version of BROWN MULE was developed to include calculation of viscous effects, including effects of moderate flow separation. This is accomplished through an unique inverse integral boundary layer solution which uses an analytical description of the velocity profile.(10) This profile and other supporting relationships are based on curve fits to experimentally determined compressible turbulent boundary layers including separated layers. The viscous effects are imposed upon the flow through a surface source model which is simply implemented as a modified solid wall boundary condition. These viscous calculations are quite efficient and impose negligible time and storage penalties on the overall program so that the capability of the program is significantly enhanced at little cost.

This paper explains the theoretical and numerical bases of the program with emphasis on the logic behind the equation development. In addition, the program is fully detailed so that a user can quickly become familiar with its operation.

Because this code is intended for computation of complex flow fields, an application to transonic flow past a wing/body configuration representative of a modern wide body turbofan transport is also presented. A companion paper(17) describes in detail the explicit vectorization of this program on the NASA Langley Research Center VPS-32.

II. Formulation of the Numerical Procedure

Governing Equations

The governing equations for inviscid flow, the Euler equations, take the form

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial q}{\partial y} + \frac{\partial h}{\partial z} = 0 \tag{1}$$

where

$$q = [\rho, \rho u, \rho v, \rho w, e]^T$$

$$f = [\rho u, \rho u^2 + p, \rho uv, \rho uw, u(e+p)]^T$$

$$g = [\rho v, \rho uv, \rho v^2 + p, \rho vw, v(e+p)]^T$$

$$h = [\rho w, \rho uw, \rho uv, \rho w^2 + p, w\,(e+p)]^T$$

$$p = (\gamma-1)[e - \tfrac{1}{2}\rho(u^2 + v^2 + w^2)]$$

This form of the equations (other forms are possible with algebraic manipulation) is said to be in strong conservation form (1) because the fundamental properties conserved in nature - mass (ρ), momentum (ρu, ρv, ρw) and energy (e) - always appear as distinct entities in the equations. The transformation (2) (outlined in Appendix A) using curvilinear coordinates,

$$\xi = \xi(x,y,z,\tau)$$

$$\eta = \eta(x,y,z,\tau)$$

$$\zeta = \zeta(x,y,z,\tau)$$

$$\tau = \tau$$

leads to

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0 \qquad (2)$$

where

$$Q = J\ [\rho,\ \rho u,\ \rho v,\ \rho w,\ e]^T$$

$$F = J\ [\rho U,\ \rho uU + \xi_x p,\ \rho vU + \xi_y p,\ \rho wU + \xi_z p,\ U(e + p)]^T$$

$$G = J\ [\rho V,\ \rho uV + \eta_x p,\ \rho vV + \eta_y p,\ \rho wV + \eta_z p,\ V(e + p)]^T$$

$$H = J\ [\rho W,\ \rho uW + \zeta_x p,\ \rho vW + \zeta_y p + \rho wW + \zeta_z p,\ W(e + p)]^T$$

$$J = x_\xi(y_\eta z_\zeta - z_\eta y_\zeta) - y_\xi(x_\eta z_\zeta - z_\eta x_\zeta) + z_\zeta(x_\eta y_\zeta - y_\eta x_\zeta)$$

$$U = \xi_x u + \xi_y v + \xi_z w + \xi_t$$

$$V = \eta_x u + \eta_y v + \eta_z w + \eta_t$$

$$W = \zeta_x u + \zeta_y v + \zeta_z w + \zeta_t$$

$$\xi_x = J^{-1}\,(y_\eta z_\zeta - z_\eta y_\zeta) \qquad \eta_x = J^{-1}\,(z_\xi y_\zeta - y_\xi z_\zeta)$$

$$\xi_y = J^{-1}\,(z_\eta x_\zeta - x_\eta z_\zeta) \qquad \eta_y = J^{-1}\,(x_\xi z_\zeta - z_\xi x_\zeta)$$

$$\xi_z = J^{-1}\,(x_\eta y_\zeta - y_\eta x_\zeta) \qquad \eta_z = J^{-1}\,(x_\zeta y_\xi - y_\zeta x_\xi)$$

$$\xi_t = -\,x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z \qquad \eta_t = -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z$$

$$\zeta_x = J^{-1}\,(y_\xi z_\eta - z_\xi y_\eta)$$

$$\zeta_y = J^{-1}\,(x_\eta z_\xi - z_\eta x_\xi)$$

$$\zeta_z = J^{-1}\,(x_\xi y_\eta - y_\xi x_\eta)$$

$$\zeta_t = -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z$$

Discretization

To discretize (2), let us integrate equation (2) over the volume of a computational cell.

$$\int_\nu \left(\frac{\partial Q}{\partial \tau} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta}\right) d\nu = 0 \tag{2a}$$

The mean value theorem of calculus permits us to write

$$\int_\nu \frac{\partial Q}{\partial t}\, d\nu = \nu\left(\frac{\Delta Q}{\Delta \tau}\right) \tag{2b}$$

where $\left(\frac{\Delta Q}{\Delta \tau}\right)$ is an effective average value of $\frac{\partial Q}{\partial \tau}$ within the cell volume ν.

The flux terms combine to give $\int \nabla \cdot \vec{F} d\nu$, where ∇ is the divergence operator (in the curvilinear coordinate frame) and F is a vector with components F, G and H.

The divergence theorem then permits us to write

$$\int_S \nabla\cdot\vec{F}d = \int_S \vec{n}\cdot\vec{F}dS \qquad (2c)$$

where S is the cell surface and $\vec{n}$ is the unit outward normal.

The flux terms - F, G, H - transport fluid properties across ξ,η,ζ faces of a cell, respectively, so that (2c) becomes

$$\int_S \vec{n}.\vec{F}dS = (F_{out} - F_{in})\Delta\eta\Delta\zeta + (G_{out} - G_{in})\Delta\xi\Delta\zeta + (H_{out} - H_{in})\Delta\xi\Delta\eta \qquad (2d)$$

where effective values of F, G and H are used on the cell faces. Using (2b) and (2d) in (2a), letting $\delta = (\ )_{out} - (\ )_{in}$, realizing $\Delta\nu = \Delta\xi\Delta\eta\Delta\zeta$, and dividing through by $\Delta\nu$ gives

$$\frac{\Delta Q}{\Delta\tau} + \frac{\delta F}{\Delta\xi} + \frac{\delta G}{\Delta\eta} + \frac{\delta H}{\Delta\zeta} = 0 \qquad (3)$$

Equation (3) might have been obtained from equation (2) by simply approximating the partial derivatives in (2) with suitable differences. However, the procedure [(2a) through (2d)] here shows more clearly the finite volume nature of equation (3) and, therefore, of the entire problem. The volume approach is, in general, a more physically meaningful approach and is recommended whenever the option for this approach arises. It is now necessary to precisely define the individual operators and terms in equation (3) and to formulate the precise numerical problem to be solved. The δ operator is the central difference operator notation and means, e.g.

$$\delta_i F = F_{i+1/2} - F_{i-1/2} \qquad (4a)$$

To interpret this relation we realize that the cell center is at i, so that i + 1/2 and i - 1/2 signify cell faces, and thus the flux vectors (F, G, H) are evaluated at cell faces as dictated by the finite volume approach. The computational coordinates (ξ, η, ζ) correspond directly to the cell indices (i, j, k) so that $\Delta\xi = \Delta\eta = \Delta\zeta = 1$ and (3) can be written as(3)

$$\Delta Q = -\Delta\tau \ (\delta_i F + \delta_j G + \delta_k H) \qquad (4b)$$

We will be advancing the solution in time. The most recently computed time step is the n step, or level. Equation (4b) might then be written as

$$\Delta Q^n = -\Delta\tau \left(\delta_i F^{n+1} + \delta_j G^{n+1} + \delta_k H^{n+1}\right) \qquad (5)$$

with $\qquad \Delta Q^n = Q^{n+1} - Q^n$

The unknowns are really Q^{n+1}, but it is convenient to work with ΔQ^n as the unknowns because eventually, as we get closer to the final solution, ΔQ^n will go to zero. This fact will be used to improve our computational efficiency.

Linearization

Now, the flux vectors are explicit functions of Q, so that we can write(4,5,6)

$$F^{n+1} = F^n + \left(\frac{\partial F}{\partial Q}\right)^n \left(Q^{n+1} - Q^n\right) + \cdots = F^n + \left(\frac{\partial F}{\partial Q}\right)^n \Delta Q^n + \cdots$$

$$G^{n+1} = G^n + \left(\frac{\partial G}{\partial Q}\right)^n \left(Q^{n+1} - Q^n\right) + \cdots = G^n + \left(\frac{\partial G}{\partial Q}\right)^n \Delta Q^n + \cdots \qquad (6a)$$

$$H^{n+1} = H^n + \left(\frac{\partial H}{\partial Q}\right)^n \left(Q^{n+1} - Q^n\right) + \cdots = H^n + \left(\frac{\partial H}{\partial Q}\right)^n \Delta Q^n + \cdots$$

The flux vectors are nonlinear in the dependent variables, Q or ΔQ. For a manageable numerical solution they must be linearized, which is accomplished by dropping terms of order $(\Delta Q^n)^2$ and higher. The derivatives such as $\partial F/\partial Q$, which appear in the expressions for the flux vectors, form matrices known as the Jacobian matrices for each flux vector. Because equation (1) or (5) actually represents 5 distinct equations, there are 5 elements to each of Q, F, G and H and the flux Jacobian matrices therefore each contain 25 elements,

$$A^n_{\ell m} = \frac{\partial F^n_\ell}{\partial Q_m} , \; B^n_{\ell m} = \frac{\partial G^n_\ell}{\partial Q_m} , \; C^n_{\ell m} = \frac{\partial H^n_\ell}{\partial Q_m} , \; \ell = 1, 5, \; m = 1,$$

With this notation, the linearized flux vectors become,

$$\begin{aligned}
F^{n+1}_1 &= F^n_1 + A^n_{11} \Delta Q^n_1 + A^n_{12}\Delta Q^n_2 + A^n_{13}\Delta Q^n_3 + A^n_{14}\Delta Q^n_4 + A^n_{15}\Delta Q^n_5 \\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (6b)\\
F^{n+1}_5 &= F^n_5 + A^n_{51}\Delta Q^n_1 + A^n_{52}\Delta Q^n_2 + \cdots \qquad\qquad + A^n_{55}\Delta Q^n_5
\end{aligned}$$

with similar expressions for G^{n+1} and H^{n+1}.

Using (6b), and the companion expressions for G and H, in (5) gives

$$\begin{aligned}
\Delta Q^n_1 &= -\Delta\tau(\delta_i F^n_1 + \delta_i \; (A^n_{11}\Delta Q_1 + \ldots + A^n_{15}\Delta Q^n_5)) + \delta_j G^n_1 + \delta^n_j(B^n_{11}\Delta Q_1 + \ldots + B^n_{15}\Delta Q^n_5) \\
&\vdots \qquad\qquad\qquad\qquad + \delta_k H^n_1 + \delta_k(C^n_{11}\Delta Q^n_1 + \ldots + C^n_{15}\Delta Q^n_5) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (7)\\
\Delta Q^n_5 &= -\Delta\tau(\delta_i F^n_5 + \delta_i \; (A^n_{51}\Delta Q^n_1 + \ldots + A^n_{55}\Delta Q^n_5) + \delta_j G^n_5 + \delta_j(B^n_{51}\Delta Q^n_1 + \ldots + B^n_{55}\Delta Q^n_5) \\
& \qquad\qquad\qquad\qquad + \delta_k H^n_5 + \delta_k(C^n_{51}\Delta Q^n_1 + \ldots + C^n_{55}\Delta Q^n_5)
\end{aligned}$$

Equations (7) are an implicit set of linear equations for the unknowns ΔQ^n. They are implicit in two ways. First, each ΔQ^n depends on all five ΔQ^n, giving a system implicit with respect to the dependent variables at a given location in the grid. Second, the difference operator introduces neighboring values of the unknowns (recall, for example,

$$\delta F_i^{n+1} = F_{i+\frac{1}{2}}^{n+1} - F_{i-\frac{1}{2}}^{n+1} \quad \text{thus giving a}$$

spatially implicit system as well. Equations (7) can be algebraically rearranged into the following matrix equation,

$$\left[\begin{array}{l|l|l}
1+\Delta\tau(\delta_i A_{11}^n+\delta_j B_{11}^n+\delta_k C_{11}^n) & \Delta\tau(\delta_i A_{12}^n+\delta_j B_{12}^n+\delta_k C_{12}^n) & \ldots\Delta\tau(\delta_i A_{15}^n+\delta_j B_{15}^n+\delta_k C_{15}^n) \\
\cdot & & \\
\cdot & & \\
\cdot & & \\
\Delta\tau(\delta_i A_{51}^n+\delta_j B_{51}^n+\delta_k C_{51}^n) & \Delta\tau(\delta_i A_{52}^n+\delta_j B_{52}^n+\delta_k C_{52}^n) & \ldots 1+\Delta\tau(\delta_i A_{55}^n+\delta_j B_{55}^n+\delta_k C_{55}^n)
\end{array}\right]
\begin{array}{l}
\Delta Q_1^n \\ \Delta Q_2^n \\ \Delta Q_3^n \\ \Delta Q_4^n \\ \Delta Q^n{}_5
\end{array} \tag{8a}$$

$$= -\Delta\tau\{(\delta_i F_1^n+\delta_j G_1^n+\delta_1 H_1^n)\ (\delta_i F_2^n+\ \delta_j G_2^n+\delta_k H_2^n)\ldots(\delta_i F_5^n+\delta_j G_5^n+\delta_k H_5^n)\}^T$$

or, in compact notation

$$(I+\Delta\tau\delta_i A\cdot+\Delta\tau\delta_j B\cdot+\Delta\tau\delta_k C\cdot)\ \Delta Q^n = -\Delta\tau(\delta_i F^n+\delta_j G^n+\delta_k H^n) \tag{8b}$$

The dot indicates that the difference operator acts on the products $A\Delta Q^n$, $B\Delta Q^n$ and $C\Delta Q^n$.

Equations (8) are the expression of the Euler equations which are to be numerically solved. It might be wise, at this point, to review the basic steps leading to (8). We start with the Euler equations in conservative differential form in Cartesian coordinates, equation (1). These are transformed to curvilinear coordinates, equation (2). The transformed equations are discretized by integrating over a cell volume to yield equation (3). Details of the discretization are introduced and finally the flux vectors are linearized with respect to the dependent variables. The result of these last operations is equation (8). Thus we see that (8) is a discretized, linearized, finite volume formulation of the Euler equations.

Eigenvalues

The problem now becomes one of solving the system of equations in (8). Various schemes are possible; the scheme used here is centered on the notion that information is propagated through a flow field in certain preferred, or characteristic directions, and with characteristic velocities. (The word 'characteristic', as used here, has a double meaning. The first meaning may be taken as 'particular' and is motivated by physical arguments, while the second meaning stems from the mathematical properties of systems of partial differential equations.) Physically, information is propagated via bulk fluid motion and via acoustic waves. Acoustic waves travel in all directions between molecules at the local sound speed; the molecules, meanwhile, are being convected, on the average, at the bulk flow velocity. Thus, the net acoustic wave speed is the sum of the sound speed and bulk velocity. The fluid velocity and the net acoustic wave speeds are the characteristic velocities. These concepts are mathematically implemented through consideration of the eigenvalues of the flux Jacobian matrices.

The eigenvalues of the flux Jacobian matrices are the mathematical characteristics of the system represented by (8). For the present problem they turn out to be identical to the characteristic velocities with which information is propagated in the flow. (The theory of characteristics is described, at length, in Ref. 7; the operations required to find the eigenvalues of the matrices here are outlined in Appendix C). Each flux matrix is a 5 x 5 system and has, therefore, 5 eigenvalues

λ^i_ξ, λ^i_η, λ^i_ζ, $i = 1, 5$ with (ξ,η,ζ) referring to (A,B,C), respectively.

The eigenvalues for the flux Jacobian matrices here are

$$\lambda^1_k = \lambda^2_k = \lambda^3_k = k_x u + k_y v + k_z w$$

$$\lambda^4_k = \lambda^1_k + c|\nabla k| \tag{9}$$

$$\lambda^5_k = \lambda^1_k - c|\nabla k|$$

where $k = \xi,\eta,\zeta$ for F,G,H respectively, c is the speed of sound and

$|\nabla k| = (k_x^2 + k_y^2 + k_z^2)^{1/2}$.

Interpretation of the eigenvalues as measures of the velocities at which fluid properties and information are propagated through the flow field is evident in these expressions.

Flux Vector Splitting

To take advantage of the physical significance of the eigenvalues, we write the flux vectors as a linear combination of, so-called, subvectors which have as coefficients the five eigenvalues of each vector.(1,2,3)

Hence

$$F_i = \sum_{j=1}^{5} \lambda_\xi^j f_{ij} = \lambda_\xi^1 f_{i1} + \lambda_\xi^2 f_{i1} + \lambda_\xi^3 f_{i3} + \lambda_\xi^4 f_{i4} + \lambda_\xi^5 f_{i5}, \quad i = 1,5 \qquad (10a)$$

and similarily for

$$G_i = \sum_{j=1}^{5} \lambda_\eta^j g_{ij} \text{ and } H_i = \sum_{j=1}^{5} \lambda_\zeta^j h_{ij}$$

Since, $\lambda_k{}^1 = \lambda_k{}^2 = \lambda_k{}^3$, then the first three terms can be combined together

to give $F_i = \lambda_\xi^1 f_{i1} + \lambda_\xi^4 f_{i4} + \lambda_\xi^5 f_{i5}$ and similarly for G and H. (10b)

A very important sequence of steps is now undertaken. Recall that the equations (8) are spatially implicit. This implicit aspect of the equations can be removed as follows. The flux vectors F,G,H, by (10b) are sums of subvectors based on the eigenvalues. These subvectors can be grouped together according to whether their eigenvalue coefficients are positive or negative, so that

$$F = F^+ + F^-, \; G = G^+ + G^- \text{ and } H = H^+ + H^- \qquad (10c)$$

where, for example, F^+ is a subvector made up of the λf terms which have $\lambda > 0$ and F^- is composed of the remaining λf terms (with $\lambda < 0$).

A simple physical idea motivates the above decomposition of the flux vectors into components with positive and negative eigenvalues. Because the eigenvalues are the characteristic velocities of information propagation, then their sign indicates in which direction information is carried through the flow field. For example, suppose ξ increases to the right. Then, if λ_ξ is positive this means it is carrying information to the right. Conversely, a negative value of λ_ξ indicates information moving to the left. Now F represents the actual transport of information through the flow. Since λ_ξ can be positive and negative, then F can transport to a cell from both the left and the right. A similar argument applies to G and H so that, in general, information reaching a cell comes from all surrounding cells. The surrounding cells, themselves, contain unknown qualities so that the problem is spatially implicit; the solution at each cell depends on the solution in all the surrounding cells. However, if we decompose the flux vectors into components with positive and negative eigenvalues then we can isolate contributions to a cell due to the flux from any particular direction. The ability to isolate flux contributions is used to develop a numerical algorithm which, at each stage in the calculation, involves only flux terms transporting known quantities into each cell and thereby removes the spatially implicit nature of the equations.

Returning now to equation development, we observe that each of these new subvectors has a Jacobian matrix associated with it, that is

$$A^+ = \left(\frac{\partial F^+}{\partial Q}\right) \quad , A^- = \left(\frac{\partial F^-}{\partial Q}\right) \; , B^+ = \left(\frac{\partial G^+}{\partial Q}\right) \; , B^- = \left(\frac{\partial G^-}{\partial Q}\right) \; , C^+ = \left(\frac{\partial H^+}{\partial Q}\right) \; , C^- = \left(\frac{\partial H^-}{\partial Q}\right) \qquad (11)$$

These new subvectors can also be linearized exactly as the entire vector was linearized to yield equation (6b),

$(F^+)^{n+1} = (F^+)^n + \left(\frac{\partial F^+}{\partial Q}\right)^n (Q^{n+1} - Q^n)$ and so on for the other subvectors.

We now take this linearization plus equations (10) and (11) to form an equation analogous to (8),

$$\left(I + \Delta\tau\delta_i A^{+}\cdot + \Delta\tau\delta_i A^{-}\cdot + \Delta\tau\delta_j B^{+}\cdot + \Delta\tau\delta_j B^{-}\cdot + \Delta\tau\delta_k C^{+}\cdot + \Delta\tau\delta_k C^{-}\cdot\right)\Delta Q^{n} =$$
$$- \Delta\tau \left(\delta_i F^{+} + \delta_i F^{-} + \delta_j G^{+} + \delta_j G^{-} + \delta_k H^{+} + \delta_k H^{-}\right) \tag{12}$$

Approximate Factorization

Equation (12) is a block tridiagonal system and is still implicit, both spatially and with respect to the dependent variables at a given point. To eliminate the spatial implicitness, an approximation to equation (12) is formed via approximate factorization,(8)

$$\left(I + \Delta\tau\delta_i A^{+}\cdot + \Delta\tau\delta_j B^{+}\cdot + \Delta\tau\delta_k C^{+}\cdot\right)\left(I + \Delta\tau\delta_i A^{-}\cdot + \Delta\tau\delta_j B^{-}\cdot + \Delta\tau\delta_k C^{-}\cdot\right)\Delta Q^{n} = -\Delta\tau R^{n} \tag{13}$$

where $R^{n} = \text{residual} = \left(\delta_i F^{+} + \delta_i F^{-} + \delta_j G^{+} + \delta_j G^{-} + \delta_k H^{+} + \delta_k H^{-}\right)^{n}$

In this form, terms of order $(\Delta\tau)^2$ appear on the left hand side which are not in the original equation (12). Consequently this formulation is, at best, second order accurate in time. We solve equation (13) here with the two step scheme

$$\left(I + \Delta\tau\delta_i A^{+}\cdot + \Delta\tau_j B^{+}\cdot + \Delta\tau\delta_k C^{+}\cdot\right) X^{1} = -\Delta\tau R^{n}$$
$$\left(I + \Delta\tau\delta_i A^{-}\cdot + \Delta\tau\delta_j B^{-}\cdot + \Delta\tau\delta_k C^{-}\cdot\right) X^{2} = X^{1} \tag{14}$$
$$\Delta Q^{n} = X^{2}$$

Final Equations

The algorithm in equation (14) is now illustrated. Consider the first equation multiplied out,

$$X^{1} + \Delta\tau\delta_i\left(A^{+}X^{1}\right) + \Delta\tau\delta_j\left(B^{+}X^{1}\right) + \Delta\tau\delta_k\left(C^{+}X^{1}\right) = -\Delta\tau R^{n}$$

and write out the finite difference operations using a first order spatial difference.

$$X^1_{i,j,k} + \Delta\tau(A^+X^1)_{i,j,k} - \Delta\tau(A^+X^1)_{i-1,j,k} + \Delta\tau(B^+X^1)_{i,j,k} - \Delta\tau(B^+X^1)_{i,j-1,k}$$

$$+ \Delta\tau(C^+X^1)_{i,j,k} - \Delta\tau(C^+X^1)_{i,j,k-1} = -\Delta\tau R^n_{i,j,k} \qquad (15)$$

The flux matrices (A^+,B^+,C^+) are functions of the cell metrics and the dependent variables. The metrics are evaluated at the cell faces indicated by the subscripts appearing in equation (15). The dependent variables to be used in evaluating the flux matrices are determined based on the facts that (1) this is a cell centered, finite volume scheme so that the dependent variables are taken as constant within a cell, and (2) the matrices (A^+,B^+,C^+) involve only positive eigenvalues and represent transport in the positive (ξ,η,ζ) directions only. For example, considering the ξ flux through cell (i,j,k), the flux in comes from cell (i-1,j,k) while the flux out is from cell (i,j,k), which means we use

$Q^n_{i,j,k}$ to evaluate $(A^+X^1)_{i,j,k}$ and $Q^n_{i-1,j,k}$ to evaluate $(A^+X^1)_{i-1,j,k}$.

To show this we will express the flux matrices as functions of the dependent variables whose indices indicate the locations at which they are evaluated. Equation (15) now becomes

$$X^1_{i,j,k} + \Delta\tau[A^+(Q^n_{i,j,k})]_{i,j,k}X^1_{i,j,k} - \Delta\tau[A^+(Q^n_{i-1,j,k})]_{i-1,j,k}X^1_{i-1,j,k}$$

$$+ \Delta\tau[B^+(Q^n_{i,j,k})]_{i,j,k}X^1_{i,j,k} - \Delta\tau[B^+(Q^n_{i,j-1,k})]_{i,j-1,k}X^1_{i,j-1,k}$$

$$+ \Delta\tau[C^+(Q^n_{i,j,k})]_{i,j,k}X^1_{i,j,k} - \Delta\tau[C^+(Q^n_{i,j,k-1})]_{i,j,k-1}X^1_{i,j,k-1}$$

$$= -\Delta\tau R^n_{i,j,k} \qquad (16)$$

or rearranging,

$$\{I + \Delta\tau[A^{+}(Q^{n}_{i,j,k}) + B^{+}(Q^{n}_{i,j,k}) + C^{+}(Q^{n}_{i,j,k})]_{i,j,k}\}X^{1}_{i,j,k} = -\Delta\tau R^{n}_{i,j,k}$$

$$+ \Delta\tau[A^{+}(Q^{n}_{i-1,j,k})]_{i-1,j,k}X^{1}_{i-1,j,k} + \Delta\tau[B^{+}(Q^{n}_{i,j-1,k})]_{i,j-1,k}X^{1}_{i,j-1,k}$$

$$+ \Delta\tau[C^{+}(Q^{n}_{i,j,k-1})]_{i,j,k-1}X^{1}_{i,j,k-1} \qquad (17)$$

Examining equation (17) reveals that if we start at the lowest index boundaries and advance in the direction of increasing (i,j,k) consistent with the positive eigenvalues, then the right side of (17) is always known and therefore the unknowns, $X^{1}_{i,j,k}$, can be solved for directly. (The boundary conditions supply the values for the first cell.) This solution process is referred to as a forward pass, in that we march through the computational grid in the direction of increasing (i,j,k). The problem is no longer spatially implicit. Each step does, however, require the solution of a 5 x 5 system similar to equation (8a) which expresses the interrelation among the Q's at each point. Formally, equation (17) represents a lower block bidiagonal system which can be solved directly by forward substitution, in contrast to the fully implicit block tridiagonal system of equation (12).

The second equation in (14) is treated in exactly the same manner as above with the exception that negative eigenvalues are used. This means the transport is in the negative (ξ,η,ζ) directions and that the differencing proceeds in the negative (i,j,k) directions. For example considering the ξ flux through cell (i,j,k), the flux in crosses face (i,j,k) but comes from cell (i+1,j,k) while the flux out crosses face (i-1,j,k) but comes from cell (i,j,k). This gives

$$\begin{aligned} X^2_{i,j,k} &+ \Delta\tau[A^-(Q^n_{i+1,j,k})]_{i,j,k}X^2_{i+1,j,k} - \Delta\tau[A^-(Q^n_{i,j,k})]_{i-1,j,k}X^2_{i,j,k} \\ &+ \Delta\tau[B^-(Q^n_{i,j+1,k})]_{i,j,k}X^2_{i,j+1,k} - \Delta\tau[B^-(Q^n_{i,j,k})]_{i,j-1,k}X^2_{i,j,k} \\ &+ \Delta\tau[C^-(Q^n_{i,j,k+1})]_{i,j,k}X^2_{i,j,k+1} - \Delta\tau[C^-(Q^n_{i,j,k})]_{i,j,k-1}X^2_{i,j,k} \\ &= X^1_{i,j,k} \end{aligned} \tag{18}$$

or, rearranging

$$\begin{aligned} &\{I - \Delta\tau[A^-(Q^n_{i,j,k})]_{i-1,j,k} - \Delta\tau[B^-(Q^n_{i,j,k})]_{i,j-1,k} - \Delta\tau[C^-(Q^n_{i,j,k})]_{i,j,k-1}\} \\ &X^2_{i,j,k} = X^1_{i,j,k} - \Delta\tau[A^-(Q^n_{i+1,j,k})]_{i,j,k}X^2_{i+1,j,k} - \Delta\tau[B^-(Q^n_{i,j+1,k})]_{i,j,k} \\ &X^2_{i,j+1,k} - \Delta\tau[C^-(Q^n_{i,j,k+1})]_{i,j,k}X^2_{i,j,k+1} \end{aligned} \tag{19}$$

In equations (16) through (19) the indices on Q^n and X indicate the location at which these quantities are evaluated while the indices appended to the brackets denote the faces for which the metrics are to be evaluated. Equation (19) represents an upper block bidiagonal system for the unknown vector $X^2_{i,j,k}$. This can be solved directly for the unknowns in a backward pass starting at the highest index boundaries and solving in the negative (i,j,k) directions since, at each step, the i+1, j+1 or k+1 values of X^2 are known. Again, a 5x5 system similar to equation (8a) must be solved at each (i,j,k) location. Appendix B provides a more complete explanation of the hierarchy of systems embodied in equations (16) through (19).

With the solution for X^2 now available, the last of equations (14) provides the desired solution vector ΔQ^n, which, in turn, provides the dependent variable vector at the n+1 time level,

$$Q^{n+1} = Q^n + \Delta Q^n \tag{20}$$

III. Solution Procedure

Problem Statement

The solution process can now be presented. The problem to be solved is the sequence of equations in (14) which, when the finite difference operations are written out, become equations (16) and (18). As described in Appendix B, these equations are actually block bidiagonal systems of equations which might be schematically represented (using the first of equations (14)) by

$$\begin{bmatrix} 1 & 0 & 0 & 0 \cdots & 0 \\ A_{21} & 1+A_{22} & 0 & 0 & 0 \\ 0 & A_{32} & 1+A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & 1+A_{44} & 0 \\ 0 & 0 & 0 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \cdots & 1+A_{nn} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ \vdots \\ \\ X_n \end{bmatrix} = -\Delta\tau \begin{bmatrix} -X_{1/\Delta\tau} \\ R_2 \\ R_3 \\ R_4 \\ \vdots \\ \\ R_n \end{bmatrix} \qquad (21)$$

where the subscripts refer to spatial position and n is the total number of cells in the computational grid. This system is solved directly and quite simply by forward substitution marching through the grid. That is,

$$X_1 = X_1$$

$$A_{21}X_1 + (1+A_{22})X_2 = -\Delta\tau R_2 \Rightarrow (1+A_{22})X_2 = -\Delta\tau R_2 - A_{21}X_1$$

$$A_{32}X_2 + (1+A_{33})X_3 = -\Delta\tau R_3 \Rightarrow (1+A_{33})X_3 = -\Delta\tau R_3 - A_{32}X_2 \qquad (22)$$

$$\vdots$$

$$A_{nn-1}X_{n-1} + (1+A_{nn})X_n = -\Delta\tau R_n \Rightarrow (1+A_{nn})X_n = -\Delta\tau R_n - A_{nn-1}X_{n-1}$$

But this set of equations is exactly equation (17) (or (19) for the second step). Therefore we see that the rearrangement from (16) to (17) or (18) to (19) is, in fact, the solution of the block bidiagonal system. The value of X in each cell is determined by X in the preceeding cell and R.

What, then, does the program do? We must realize that each X in (17) is actually a vector, $X = (x_1, x_2, x_3, x_4, x_5)^T$, and each equation in (17) is actually a 5x5 system of equations of the form

$$\begin{bmatrix} 1+A_{11} & A_{12} & \cdots & A_{15} \\ A_{21} & 1+A_{22} & & \\ \vdots & & & \\ A_{51} & & & 1+A_{55} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_5 \end{bmatrix} = \Delta\tau \begin{bmatrix} -R^n_{1i} + (A_{11}x_1 + \cdots + A_{15}x_5)_{i-1} \\ \\ \vdots \\ -R_{5i} + (A_{51}x_1 + \cdots + A_{55}x_5)_{i-1} \end{bmatrix} \tag{23}$$

where the number subscripts now refer to the individual Q's ($\rho, \rho u, \rho v, \rho w, e$) and i and i-1 refer to spatial positions. This system must be solved for each cell in the computational grid. It is the construction of the individual terms in this 5x5 system and then the solution of this system, for every cell, which is the essential function of the computation.

Doolittle's Method

The 5x5 linear systems of equations are solved here using Doolittle's method. This method[9] is one of a family of techniques in which the linear system $A\vec{x} = \vec{b}$ is solved by first factoring the coefficient matrix into lower-triangular and upper-triangular terms, L and U, such that LU = A and $LU\vec{x} = \vec{b}$. This last expression is rearranged to give $U\vec{x} = L^{-1}\vec{b}$ or $L\vec{x}$

$= U^{-1}\vec{b}$ which can be solved directly for $\vec{x}$ by backward or forward substitution. The aspect of this scheme which is particular to Doolittle's method is that the terms on the diagonal of the L matrix are all 1. As shown in Appendix B, the 5x5 system involves unknown quantities at the cell in question and the known solution from a neighboring cell. The backward or forward substitution required to find the unknowns from the known solution is a recursive relation which prohibits vectorization (on current vector processors) in the direction of the substitution. However, a vector solution can be effected by vectorizing along a computational line normal to the backward or forward substitution directions. Such vectorization has been implemented on a variety of vector processors (Cray-1S, Cray X-MP, VPS-32) so that totally vectorized routines for solving these linear systems with Doolittle's method are available.

System Details and Procedure

In order to properly understand the computational procedure let us examine some of the details of the 5x5 system. The first equation in the system is

$$\begin{aligned}\left((1+A_{11})x_1 + A_{12}x_2 + A_{13}x_3 + A_{14}X_4 + A_{15}X_5\right)_i &= -\Delta\tau R_{1i} \\ &+ \left(A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + A_{14}x_4 + A_{15}x_5\right)_{i-1} \qquad (24)\end{aligned}$$

Writing out the A's and R's (see equations (17) and (13)) we have

$$A_{\ell m} = \Delta\tau\left(A^+_{\ell m} + B^+_{\ell m} + C^+_{\ell m}\right) = \Delta\tau\ \left(\frac{\partial F^+_\ell}{\partial Q_m} + \frac{\partial G^+_\ell}{\partial Q_m} + \frac{\partial H^+_\ell}{\partial Q_m}\right) \qquad (25)$$

and

$$R^n_\ell = \delta_i F^+_\ell + \delta_i F^-_\ell + \delta_j G^+_\ell + \delta_j G^-_\ell + \delta_k H^+_\ell + \delta_k H^-_\ell \qquad (26)$$

where $\ell = 1,5, m = 1,5$ and the flux vectors $(F^+, \cdots, H^+)$ are functions of of $(Q_1, Q_2, Q_3, Q_4, Q_5) = J(\rho, \rho u, \rho v, \rho w, e)$ (see equations (2)). (27)
Recall that the flux vectors are evaluated at the cell faces; this evaluation requires that a spatial extrapolation of the Q's from the cell centers to the faces be performed. In order to maintain the second order spatial accuracy of the overall computation it is necessary to use a two-point extrapolation for the Q's which are used to evaluate R on the right hand side of the 5x5 system. This extrapolation (see Ref. 1) is

$$\begin{aligned} Q^L &= 1.5Q_{i,j,k} - 0.5Q_{i-1,j,k} \\ Q^R &= 1.5Q_{i+1,j,k} - 0.5Q_{i+2,j,k} \end{aligned} \tag{28}$$

depending on which of equations (14) is being considered. A one point extrapolation is sufficient on the left hand side.

The flux vectors (F,G,H) required in this computation are in the form of the subvectors $(F^\pm, G^\pm, H^\pm)$. The discussion concerning equation (10) describes qualitatively how the flux vectors are split into terms (or subvectors) having the eigenvalues as coefficients and how they are further divided into the subvectors having positive or negative eigenvalues as coefficients. A detailed description of this splitting (and the general eigensystem for the Euler equations) is presented in Appendix C. From the development of Appendix C the flux vectors are

$$K = \lambda_k^1 K_1 + \lambda_k^4 K_2 + \lambda_k^5 K_3 \tag{C16b}$$

$$K_1 = J\frac{\gamma-1}{\gamma}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \frac{\rho\theta}{\gamma-1} \end{bmatrix} \quad K_2 = \frac{J}{2\gamma}\begin{bmatrix} \rho \\ \rho u+\rho c\tilde{k}_x \\ \rho v+\rho c\tilde{k}_y \\ \rho w+\rho c\tilde{k}_z \\ e+p+\rho c\tilde{\theta}_k \end{bmatrix} \quad K_3 = \frac{J}{2\gamma}\begin{bmatrix} \rho \\ \rho u-\rho c\tilde{k}_x \\ \rho v-\rho c\tilde{k}_y \\ \rho w-\rho c\tilde{k}_z \\ e+p-\rho c\tilde{\theta}_k \end{bmatrix} \qquad (C17)$$

with

$K = F,G,H$ when $k = \xi,\eta,\zeta$

$$\rho c^2 + \gamma((\gamma-1)e-\rho\phi),\ \phi = \frac{\gamma-1}{2}(u^2+v^2+w^2),\ \theta_k = k_x u+k_y v+k_z w \qquad (C7)$$

$$\lambda_k^1 = \theta_k,\ \lambda_k^4 = \theta_c+c|\nabla k|,\ \lambda_k^5 = \theta_c-c|\nabla|,\ \nabla k = (k_x^2+ k_y^2+ k_z^2)^{1/2} \qquad (C9)$$

$(\tilde{\ })$ => divide by$|\nabla k|$

Appendix D describes the construction of the flux Jacobian matrices (A,B,C) which are required in this computation. The elements of these matrices are formed by a straightforward differentiation of the split flux vectors, after the flux vectors have been written as explicit functions of the conserved variables, Q. One difficulty associated with forming the flux Jacobians is that the flux vectors become long, unwieldy expressions when written in terms of the the conserved variables. Another tedious aspect of the Jacobian computation is properly assigning contributions to the positive and negative components. This assignment must be done separately for each cell and each time step. Because of the length of the expressions for the flux Jacobian elements, it is not appropriate to write them out in this presentation.

The time step $\Delta\tau$ is based on the maximum allowable time step in each cell volume in order to accelerate convergence for steady state solutions. The time step is determined at each point in the grid by (see Ref. 1)

$$\Delta\tau = \frac{\Delta\tau^{\xi}\Delta\tau^{\eta}\Delta\tau^{\zeta}}{\Delta\tau^{\xi}\Delta\tau^{\eta} + \Delta\tau^{\xi}\Delta\tau^{\zeta} + \Delta\tau^{\eta}\Delta\tau^{\zeta}}$$

$$\text{with } \Delta\tau^{k} = \frac{\text{CFL } \Delta k}{\max\limits_{\ell}|\lambda_k^{\ell}|}, \; k = \xi,\eta,\zeta, \; \ell = 1,4,5 \qquad (29)$$

To express this in a more convenient form introduce short hand notation for the maximum eigenvalues $\lambda_{k_m} = \max\limits_{\ell} |\lambda_k^{\ell}|$ and substitute the expression for $\Delta\tau^k$ into $\Delta\tau$ to give

$$\Delta\tau = \text{CFL} \frac{\Delta\xi\Delta\eta\Delta\zeta}{\lambda_{\xi_m}\lambda_{\eta_m}\lambda_{\zeta_m}} \frac{1}{\frac{\Delta\xi\Delta\eta}{\lambda_{\xi_m}\lambda_{\eta_m}} + \frac{\Delta\xi\Delta\zeta}{\lambda_{\xi_m}\lambda_{\zeta_m}} + \frac{\Delta\eta\Delta\zeta}{\lambda_{\eta_m}\lambda_{\zeta_m}}}$$

Simplifying yields

$$\Delta\tau = \text{CFL} \frac{\Delta\xi\Delta\eta\Delta\zeta}{\Delta\xi\Delta\eta\lambda_{\zeta_m} + \Delta\xi\Delta\zeta\lambda_{\eta_m} + \Delta\eta\Delta\zeta\lambda_{\xi_m}}$$

However, in the computational grid $\Delta\xi = \Delta\eta = \Delta\zeta = 1$ so that we have

$$\Delta\tau = \frac{\text{CFL}}{\lambda_{\xi_m} + \lambda_{\eta_m} + \lambda_{\zeta_m}}$$

The eigenvalues, λ_k^{ℓ}, are (see equation (9))

$$\lambda_k^1 = k_x u + k_y v + k_z w = \theta_k, \; \lambda_k^{4,5} = \theta_k \pm c|\nabla k|$$

and the maximum absolute value will always be $|\theta_k| + c|\nabla k|$ so that

$$\Delta\tau = \frac{\text{CFL}}{\sum\limits_k (|\theta_k| + c|\nabla k|)} \qquad k = \xi,\eta,\zeta \qquad (30)$$

This is the expression by which the time step is evaluated in the program.

Returning to the computational procedure we see that the 5x5 system of equations at each grid point requires a determination of the flux Jacobian matrices $(A^+, A^-, \cdots, C^-)$, the time step $\Delta\tau$ and the residuals R^n. The residuals, in turn, require determination of the two-point extrapolated Q's, the flux vectors $(F^+, F^-, \cdots, H^-)$, and the sum of the difference operations $(\delta_i F^+ + \cdots + \delta_k H^-)$. With these quantities determined the system can be solved.

The calculations thus proceed as follows.

1. two point extrapolated Q's
2. eigenvalues
3. flux vectors
4. sum of flux vector differences
5. time step
6. flux Jacobian matrices
7. coefficient matrix of left hand side
8. lower/upper decomposition of left hand side coefficient matrix
9. sum of right hand side terms
10. solution of the system by forward or backward substitution
11. update Q's

Supporting Calculations

The above procedure constitutes the essential function of the computer program. Additional computations and operations are required, however, to obtain a complete solution. These additional steps include establishing the initial flow conditions, computing the metrics (that is, the dimensions) of each cell, enforcing boundary conditions, determining the viscous influence and presenting the results in a meaningful manner. Brief descriptions of some of these steps follows.

The boundary conditions at the far field boundary and the body surface are enforced using characteristic variables and phantom points as developed in Ref. 1. In a manner similar to the +/- flux vector splitting this method takes advantage of the natural signalling processes (information propagation) in the flow to more correctly determine conditions on the exterior and interior surfaces of the computational domain. The eigenvalues are used to determine whether information propagates into or out of the computational domain and also to provide expressions for the flow variables at the boundaries. Phantom points (fictitous points which are immediately exterior to far field boundaries and immediately embedded inside solid bodies) facilitate the computations near the boundaries. Through extensive numerical experimentation the combination of characteristic variable boundary conditions and phantom points appears to provide more efficient and accurate evaluation of the boundary conditions than other techniques such as extrapolation, zero pressure gradient and use of the normal momentum equation.

The only metrics required in this computational solution to the Euler equations are the area vectors of the grid cell faces. These vectors have components $J(k_x,k_y,k_z,)$ where $k = \xi,\eta,\zeta$. For example, $J\eta_z$ is the z component of area for an η = constant face of a cell. The total area of a cell face is $J|\nabla k|$; for example, $J\ |\nabla\eta|$ is the total area of an η = constant face. The area vectors are computed as the cross product of the diagonals of each face. Also computed are the direction cosines of each component of the area vector, which are $(\tilde{k}_x,\tilde{k}_y,\tilde{k}_z) = (k_x,k_y,k_z)/|\nabla k|$.

Initial conditions are prescribed as freestream conditions everywhere. The flow variables are made non-dimensional using ρ_∞ and c_∞. A perfect gas is assumed. The freestream pressure is

$$P_\infty = \rho_\infty R T_\infty$$

$$= \rho_\infty R \frac{c_\infty^2}{\gamma R} = \frac{\rho_\infty c_\infty{}^2}{\gamma}$$

which, when made non-dimensional with $\rho_\infty c_\infty{}^2$, is

$$p_\infty = \frac{1}{\gamma} \tag{31a}$$

The freestream energy (internal, $c_v T_\infty$, plus kinetic, $\frac{1}{2}\, q_\infty^2$) per unit volume

$$e_\infty = \frac{1}{\gamma-1}\, p_\infty + \frac{1}{2}\, \rho_\infty q_\infty$$

where $q_\infty{}^2 = u_\infty{}^2 + v_\infty{}^2 + w_\infty{}^2$

Using $\rho_\infty c_\infty{}^2$ to non-dimensionalize and recognizing $M_\infty{}^2 = q_\infty{}^2/c_\infty^2$ gives

$$e_\infty = \frac{1}{(\gamma-1)}\;\frac{1}{\gamma} + \frac{1}{2}\; M_\infty^2 \tag{31b}$$

Numerical Properties

The scheme as described is implicit, upwind and finite volume. First order differencing is used in the left-hand side matrix operator which implicitly operates on the dependent variable difference ΔQ^n. This yields second-order accuracy in space. For consistency, second-order differencing is used in the residuals on the right-hand side such that the overall scheme is second-order accurate in space.

Because only steady state solutions are of concern here local time stepping and first-order temporal differences are used.

The upwind character of the solution scheme precludes the necessity of adding artificial dissipation to damp oscillations as is commonly required in central difference schemes.

Analysis of a scalar equation and a system of equations[14] shows that the present scheme is conditionally stable. The practical limit for CFL number is approximately 20, although under certain flow conditions much higher CFL numbers are possible.

IV. Viscous Calculations

Viscous effects are determined through the use of an inverse integral method for computation of turbulent compressible boundary layers. The method is referred to as integral in that the fundamental basis of the calculation is integration of the momentum and kinetic energy equations. Use of a prescribed displacement thickness distribution in place of the more usual prescribed pressure distribution as part of the input to the solution gives rise to the inverse nature of this method. The influence of the boundary layers is imposed upon the external inviscid flow by a surface source model in which the velocity normal to solid surfaces induced by the viscous displacement thickness acts as the effective strength of an inviscid source distribution on the solid surface. In the present application the viscous calculations are along two-dimensional strips; no explicit spanwise calculations are performed. A full description of the procedure appears in references (10,11,12).

Central to this scheme is the representation of the turbulent boundary layer by an analytical expression for the velocity profile. This analytical expression, developed from curve fits to experimental data, is applicable to both attached and mildly separated compressible turbulent boundary layers. From this expression the various boundary layer length scales, the skin friction and the dissipation can be obtained. The velocity profile used here is actually for the equivalent incompressible viscous flow. The compressible flow properties are determined from correlations of the three compressible shape factors and skin friction with the incompressible shape factor and boundary layer edge Mach number. These correlations are also based on curve fits to experimental data.

The basic governing equations for the particular inverse integral method used here are those for momentum and mean-flow kinetic energy. With suitable algebraic manipulations, these two equations become a coupled pair of first order ordinary differential equations for $\bar{H}$ (the incompressible shape factor) and M_e (the edge Mach number). The equations contain the four compressible length scales, along with M_e, c_f (the skin friction coefficient), and D (the dissipation integral). With the analytic velocity profile, the correlations and the displacement thickness distribution (from the previous viscous solution) known the pair of differential equations can be solved for the unknowns $\bar{H}$ and M_e. In order to evaluate the dissipation integral the Cebeci-Smith two-layer eddy viscosity turbulence model is used for τ while the velocity profile provides $\partial u/\partial y$. The actual solution of the system is obtained through a fourth order, four stage Runge-Kutta routine.

The solution procedure for a given viscous calculation begins with input of $\bar{H}$, $\bar{u}_e^+$, $\bar{Re}_\theta$ (the equivalent incompressible shape factor, friction velocity and Reynolds number based on $\bar{\theta}$, the incompressible momentum thickness) to the main viscous subroutine (SOURCE). These input quantities come from the previous viscous cycle. Also input are the conserved dependent variables (ρ, ρu, ρv, ρw, e). A sequence of steps follow which lead to a new or updated velocity profile. This profile is then used in the solution of the coupled pair of differential equations for $\bar{H}$ and M_e. Also required in the solution for $\bar{H}$and M_e is δ, the displacement thickness distribution. The displacement thickness for the current viscous calculation is obtained by multiplying the thickness from the previous calculation by the ratio of the previous edge velocity to the previous inviscid surface velocity magnitude. (On the very first viscous pass, the solution is started by specifying the location and displacement thickness at the locations of boundary layer transition and computing a flat plate turbulent boundary layer.) With the δ^* distribution available updated values of $\bar{H}$ and M_e are determined. The new values of $\bar{H}$ and M_e distributed along the body can then be used with the correlations to obtain distributions of all of the actual compressible properties of the boundary layer.

Among the compressible properties determined in the above procedure is the mass flux defect, $\rho_e u_e \delta^*$, created by the boundary layer. The usefulness of this property is apparent if we integrate the differential continuity equation; the result is

$$(\rho v)_n = \frac{d}{dx} (\rho_e u_e \delta^*)$$

where $(\rho v)_n$ is the mass flux per unit area normal to the surface induced by the boundary layer and x is the streamwise coordinate. With respect to the external inviscid flow, the influence of the boundary layer is to displace the streamlines away from the body. This displacement is equivalent to superimposing a mass flux, which is equal to the mass flux defect, normal to the body surface. The quantity $(\rho v)_n$ is precisely this normal mass flux. Thus the distribution of $\rho_e u_e \delta^*$ provides the means to determine the viscous influence on the inviscid flow, or, in other words, to determine the viscous-inviscid interaction. The normal mass flux applied in the solid wall boundary conditions may be interpreted as a surface source strength. An alternate interpretation of $(\rho v)_n$ is that it represents the porosity of the surface. The boundary condition routine in this program includes $(\rho v)_n$ in the characteristic variable solid wall computations.

V. Program Details

Subroutines

The solution procedure is directed by a single calling subroutine, STEP. Some computations are performed in STEP but its main purpose is to call the proper sequence of subroutines to perform the requisite calculations. Perhaps the best way to briefly describe each subroutine is to put each into the context of its sequence and role in the computational scheme. The following table details, in chronological order (in the computer program), each subroutine and its function, relevant equation or reference and primary output variables.

subroutine	function	equation	output variables
IC	initial conditions	31	$\rho,\rho u,\rho v,\rho w,e$
METRIC	cell face area vectors	32	$k_x,k_y,k_z,\nabla k$ $(k=\xi,\eta,\zeta)$
BC	boundary conditions	42-47 in Ref. 1	$\rho,\rho u,\rho v,\rho w,e$
DELQ	2 point extrapolated Q's	28	$\rho,\rho u,\rho v,\rho w,e$
FLUX	eigenvalues	9	λ_k^ℓ $(k=\xi,\eta,\zeta,$ $\ell=1,4,5)$
FLUX	flux vectors	C16, C17	$F^\pm,G^\pm,H^\pm$, at step n
DELQ	residual=sum of flux vector differences	13	R_ℓ^n, $\ell=1,\ldots,5$
EIGENV	time step	30	$\Delta\tau$
FJMAT	eigenvalues	9	λ_k^ℓ
FJMAT	flux Jacobian matrices	Appendix D	$A^\pm,B^\pm,C^\pm$
STEP	coefficient matrix of 5x5 system,	17, 19	
AEQLU	lower/upper decomposition of coefficient matrix of left hand side	linear algebra text	

DOOP, DOOM	right hand side of 5x5 system	17, 19	$-R^n+(A_{11}x_1+A_{12}x_2+\ldots+A_{15}x_5)$
DOOP, DOOM	forward/backward substitution solution of 5x5 system	linear algebra text	ΔQ^n
STEP	update Q's	20	$\rho,\rho u,\rho v,\rho w,e$
BC	boundary conditions	42-47 in Ref. 1	$\rho,\rho u,\rho v,\rho w,e$
SOURCE	viscous calculation	Section IV	ρv_n
PVAR	print results		c_p, x/c, c_ℓ, c_d

Variables

As would be expected in a program designed to solve a complex system of equations, there are a large number of variables. The names used for the variables, in most cases, are representative of the variables' physical or mathematical meanings. Brief definitions of the more important quantities are presented here.

The following variables appear in nearly all subroutines.

I,J,K = ξ,η,ζ; indices for grid points

NI,NJ,NK = number of (I,J,K) lines

R,RU,RV,RW,E,P = $\rho,\rho u,\rho v,\rho w,e,p$ in Cartesian frame, (x,y,z)

X,Y,Z = grid point coordinates in Cartesian frame, (x,y,z)

AIX,AIY,AIZ = Cartesian components of the area vector of an I constant cell face

AJX,AJY,AJZ = Cartesian components of the area vector of a J constant cell face

AKX,AKY,AKZ = Cartesian components of the area vector of a K constant cell face

SADAI,SADAJ,SADAK = magnitude of the area vectors for I,J, or K constant cell faces

The following variables appear in only some of the subroutines:

B(L,M,N,I,J,K) = flux Jacobian matrices; L=1,6 and refers to $A^+,\ldots,C^-$; M=1,5 and refers to the flux vector component; N=1,5 and refers to the dependent conserved variable; I,J,K refer to grid location

D(L,M,N,I,J,K) = coefficient matrix on the left hand side of the 5x5 systems; L=1,2 and refers to the forward or backward passes; M=1,5 and N=1,5 refer to the location within the matrix; I,J,K refer to grid location

DR,DRU,DRV,DRW,DE = right hand side of two pass algorithm (R,X^1); ΔQ^n after two pass algorithm

DT = time step

In DELQ,

RR,RUR,...,RWL,EL = right and left, two point, extrapolated dependent variables

XR,XRU,XRV,XRW,XE = flux vectors

In DOOP and DOOM,

T = sum of transport terms on right hand side of the 5x5 systems

Z = total right hand side of the 5x5 systems, initially; solution to 5x5 system, finally

Y = intermediate variables in solution of 5x5 system

In FJMAT,

EV1,EV4,EV5 = eigenvalues

CG1,CG2,CG3 = switches to assign terms to appropriate (±) flux Jacobians

AXT,AYT,AZT = $\tilde{k}_x,\tilde{k}_y,\tilde{k}_z$

In FLUX,

EV1R,...,EV5L = eigenvalues predicted on cell faces using right and left extrapolated dependent variables

XR,XRU,XRV,XRW,XE = flux vectors

In EIGENV,

CONU,CONV,CONW = θ_k

EI,EJ,EK = $|\theta_k| + c|\nabla k|$

DT = time step

Input parameters (in order of read),

CFL = CFL number

FSMACH = freestream Mach number

ALPHA,BETA,PHI = roll, pitch (angle of attack) and yaw angles

NB = number of printouts

NT = number of computational cycles per printout

NV = number of computational cycles per call to viscous calculations

ITL,ITU,ILE = I values at lower and upper trailing edges and at leading edge

KTIP = K value at wing tip

X1 = array storing the computational grid coordinates

IFREQ = frequency of calls to EIGENV, FJMAT and AEQLU (not read but set in MAIN)

NI,NJ,NK = grid size (not read but set in PARAMETER statements)

Output, every 5 cycles

NCYC = cycle number
RTMAX = maximum DR
RTRMS = rms value of DR
ETMAX = maximum DE
ETRMS = rms value of DE
XL2 = sum of squares of DR,...,DE
TCL = lift coefficient (wing)
TCD = drag coefficient (wing)
NSUP = number of supersonic points

Output, every NT cycles

ZLOC = spanwise position
S = chordwise position
CP2 = surface pressure coefficient
CL = sectional lift coefficient
CD = sectional drag coefficient
TCL = wing lift coefficient
TCD = wing drag coefficient

Execution sequence

```
MAIN

  call IC

  read input

  call METRIC

  call BC

  call PGEOM


do 3 L=1,NB

do 2 M=1,NT


call STEP
  call DELQ

     call FLUX, (NK-1)*(NJ-1)+(NK-1)*(NI-1) + (NJ-1)*(NI-1)  times

  call EIGENV, NB*NT/IFREQ  times

  call FJMAT, 6*NB*NT/IFREQ  times

  call AEQLU, 2*NB*NT/IFREQ  times

  call DOOP

  call DOOM

  call BC

  call SOURCE NB*NT/NV  times

     call INVBL

       call CORREL

       call LININT

       call RUNGE

2  continue

   call PVAR

3  continue
```

VI. Applications

This code has been used to compute transonic flow past a variety of two-dimensional and three-dimensional bodies. To demonstrate the application of the program, results from computations of a representative wide body subsonic jet transport will be presented.

Geometry and Grid

The configuration in question is a complex wing/fuselage combination designated for this work as the Pathfinder geometry. The wing is a high aspect ratio, compound sweep surface with supercritical sections, span wise washout and positive dihedral. This wing is mid-mounted on a simple axisymmetric fuselage which has a slightly blunted ogive nose and a blunt base without boattailing. For the computational geometry there is no wing root fillet or fairing. A geometry very similar to this has undergone wind tunnel testing at NASA Langley and some results from those experiments will be used for comparison.

To generate the computational grid the program FL059, authored by T. Jameson,(13) was applied. This code is a full Euler equation solver but also includes a grid generation package for wing/body/tail geometries, which has been slightly modified by Dr. J. Luckring of NASA Langley and further modified during the present work. Figures 1 through 6 show portions of the grid produced by FL059 for the Pathfinder configuration. This is a C-H grid with 97 I lines and 17 each of the J and K lines. The lower and upper wing trailing edges are at I = 19 and 79 respectively; the leading edge is at I = 49. The wing tip is at K = 11.

The K = 1 surface is the plane of symmetry (z = 0) above and below the body and lies on the surface of the fuselage, cutting through the wing at the wing/fuselage junction. Figure 1 shows the K = 1 surface (viewed along the z axis) in the immediate vicinity of the fuselage. The blunt base, wing root profile and fuselage nose shape are clearly evident. An oblique view of the K = 1 surface near the fuselage appears in Figure 2; prominent in this figure is the rapid transition at the fuselage base of lines on the surface to lines on the mid plane. On Figure 3 an enlargement of the wing root area (K = 1, view along z axis) is presented. The supercritical section is clear. Two less desirable grid properties are also apparent. There is a slight irregularity in the grid lines immediately downstream, of the trailing edge. More importantly, the cells on the surface (upper and lower) on the aft half of the airfoil are somewhat large; this may adversely influence computations on the compression (downstream facing) surfaces of the wing.

Three views of the J = 1 surface are presented in Figures 4, 5 and 6. The entire J = 1 surface appears in an oblique view in Figure 4 where the left-most line runs along the wing leading edge. The grid extends approximately one and one-half semispans beyond the wing tip in the z direction and approximately one semispan downstream of the fuselage base. The abrupt bend in the bottom most grid line is at the fuselage base. In Figure 5 the wing is viewed along the y axis. Here the slight change in leading edge sweep is apparent. Another oblique view of the wing, but somewhat enlarged, is presented in Figure 6. The geometry of the trailing edge can be more clearly discerned in this view.

The outer boundaries of the grid are relatively near the body, being roughly one semispan in each direction from the body. (The extent in the y direction and ahead of the body are not shown but are about one and one-half semispans each.) Numerical experiments have not shown any obvious adverse influence resulting from insufficient distance to the outer boundaries although this spacing and its effects may need to be further explored.

Calculations

Computations were performed over a range of values for the various parameters which can be set in the code, in particular IFREQ, CFL, NV, NB*NT, DST, XTOP and XBOT. The best combination to minimize machine time, maximize convergence rate and still obtain reasonable output involved running for 100 to 300 cycles (NB*NT) with IFREQ=9999, CFL=15 and NV between 10 and 20. This means only one call is made to FJMAT, AEQLU and EIGENV and from five to 30 calls to the viscous routines (when they are used). For this combination the residuals are reduced by at least three orders of magnitude and the lift is within five to ten percent of its asymptotic value. All results presented here are for parameters in these ranges. The examples to be presented here are for two cases with the Pathfinder geometry: case (1), $M_\infty = 0.82$, $\alpha = 2°$, $Re_{\bar{c}} = 9.9*10^6$ and case (2), $M_\infty = 0.70$, $\alpha = 2°$, $Re_{\bar{c}} = 5.3*10^6$. These cases correspond to wind tunnel experiments conducted at NASA Langley Research Center.

Figure 7a presents computed pressure distributions for case (1) at the 45% span location along with an experimental distribution at the 43.2% span location position. The two curves on this figure are for an inviscid run of 100 cycles and a viscous run of 100 cycles with five calls to the viscous routines. For the viscous calculations the turbulent boundary layer was begun at x/c values of 0.05 and 0.2 on the lower and upper surfaces, respectively, while the initial displacement thickness (δ^*/c) was 0.0010 and 0.0012, lower and upper. On Figure 7a, the lower surface agreement is quite good up to x/c = 0.8; the deviation downstream of that point is expected in view of the relatively coarse grid. On the upper surface, the leading edge suction peak is properly computed and the basic character of the recompression is represented by the calculations. The shock, however, is not accurately captured. This may be a consequence of the coarse grid.

Distributions at other spanwise locations for case (1) appear in Figure 7b-f. The agreement between experiment and calculation deteriorates as the root and tip are approached. The nature of the distributions outboard of the 45% location suggests that the wing twist is not correctly modelled. There is progressively too much upper surface expansion and too little lower surface expansion as the tip is approached which indicate that these sections are at too high an angle of attack. At the root there is insufficient upper surface expansion indicating that the angle of attack is not large enough. It is most likely that angle of attack is not the only problem at the root. The interaction between the fuselage and wing is probably not being adequately computed also.

As an example of the influence of angle of attack, Figure 8 is presented in which viscous runs (same parameter values as in Figure 7) for three angles of attack are compared. It should first be observed that the experimental angle of attack is 2° but that the results in Figure 7 are actually at 2.5° as this gives reasonably good agreement at the 45% span position. Figure 8 shows computations at 2°, 2.3° and 2.6°. At the outboard sections, the tip in particular, the calculations more closely match the experimental results as the angle of attack is decreased. The trend is reversed at the inboard sections. These results further support the idea that the wing twist is not correct.

Results for case (2) appear in Figure 9, where the computations are viscous and were run for 250 cycles with NV=10. The agreement here is good except at the most inboard section. A spanwise variation which is probably due to incorrect twist is still detectable but, overall, the calculations quite satisfactorily predict the flow.

Concluding Remarks

Development and application of a code to accurately compute three-dimensional, transonic flow with viscous effects is not a simple matter as the preceding discussions hopefully show. The emphasis during this research has been to concentrate on refinement of the numerical scheme with particular attention paid to improving the computational speed and eliminating numerical oscillations. Both of these goals have been closely approached, if not fully met. However, there is clearly much that still needs to be addressed. An improved and finer mesh is obviously necessary to obtain accurate flow predictions. The grid size chosen here was dictated primarily by economy rather than accuracy. A fully three-dimensional viscous calculation scheme should be implemented in order to accurately compute root and tip flows. Multi-grid and block schemes and adaptive grids should also be considered.

Although many improvements can be made, the code as described here is an efficient and robust Euler equation solver, relatively easy to use and with reasonable accuracy with a coarse mesh. When coupled with a well-developed computational grid this program, in its present form, should prove to be a useful tool for engineering calculations of three-dimensional, high Reynolds number, transonic flow.

References

1. Whitfield, D.L. and Janus, J.M. "Three-Dimensional Unsteady Euler Equations Solutions Using Flux Vector Splitting," AIAA Paper No. 84-1552, June 1984.

2. Janus, J.M., "The Development of a Three-Dimensional Split Flux Vector Euler Solver with Dynamic Grid Applications," M.S. Thesis, Department of Aerospace Engineering, Mississippi State University, August 1984.

3. Whitfield, D.L., "Implicit Upwind Finite Volume Scheme for the Three-Dimensional Euler Equations," Mississippi State University Engineering and Industrial Research Station report MSSU-EIRS-ASE-85-1, September 1985.

4. Beam, R.M. and Warming, R.F. "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form," Journal of Computational Physics, Vol. 22, 1976, pp. 87-110.

5. Briley, W.R. and McDonald, H. "Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," Journal of Computational Physics, Vol. 24, 1977, pp. 372-397.

6. Briley, W.R. and McDonald, H. "On the Structure and Use of Linearized Block Implicit Schemes," Journal of Computational Physics, Vol. 34, 1980, pp. 54-73.

7. Whitham, G.B., Linear and Nonlinear Waves, Wiley, New York, 1974.

8. Buning, P.G. and Steger, J.L. "Solution of the Two-Dimensional Euler Equations with Generalized Coordinate Transformation Using Flux Vector Splitting," AIAA Paper No. 82-0971, June 1982.

9. Dahlquist, G. and Bjorck, A. Numerical Methods. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974, p. 157.

10. Whitfield, D.L., Swafford, T.W. and Jacocks, J.L., "Calculation of Turbulent Boundary Layers with Separation and Viscous-Inviscid Interaction," AIAA Journal, Vol. 19, No. 10, 1981, pp. 1315-1322.

11. Whitfield, D.L., Swafford, T.W., and Donegan, T.L., "An Inverse Integral Computational Method for Compressible Turbulent Boundary Layers," in Recent Contributions to Fluid Mechanics, W. Haase, editor, Springer-Verlag, New York, 1982, pp. 294-302.

12. Whitfield, D.L. and Thomas, J.L., "Transonic Viscous-Inviscid Interaction Using Euler and Inverse Boundary-Layer Equations," in Viscous Flow Computational Methods, W.T. Habashi, editor, Pineridge Press, Swanson, U.K., 1983.

13. Caughey, D.A., and Jameson, A., "Progress in Finite-Volume Calculations for Wing-Fuselage Combinations," AIAA Journal, Vol. 18, Nov. 1980, pp. 1281-1288.

14. Anderson, W.K., "Implicit Multigrid Algorithms for the Three-Dimensional Flux Split Euler Equations," Ph.D. Dissertation, Mississippi State University, August, 1986.

15. Jacocks, J.L. and Kneile, K.R., "Computation of Three-Dimensional, Time Dependent Flow Using the Euler Equations," AEDC-TR-80-49, Arnold Engineering and Development Center, Arnold Air Force Station, TN, July 1981.

16. Koenig, K. and Barton, J.M., "Numerical Solution of the Three-Dimensional Unsteady Euler Equations," AEDC-TR-83-22, Arnold Engineering and Development Center, Arnold Air Force Station, TN, August 1983.

17. Griffin, L.W., "Explicit Vectorization and Application of a Finite Volume Euler Equation Solver on the NASA Langley VPS-32 Supercomputer for Transonic Flow Calculation," M.S. Thesis, Department of Aerospace Engineering, Mississippi State University, December 1986.

Appendix A. Coordinate Transformation

The steps required to transform the Euler equations from Cartesian to curvilinear coordinates are described here$^{(2)}$. The logic behind this procedure may be more clear if it is realized that the main purpose of this transformation is to replace all Cartesian derivatives with curvilinear derivatives.

The Euler equations in Cartesian coordinates are in the form

$$q_t + f_x + g_y + h_z = 0 \tag{A1}$$

Consider (ξ, η, ζ) = function (x, y, z, t),

(x, y, z) = function (ξ, η, ζ, τ) and $t = \tau$.

Cartesian derivatives are then

$$\begin{aligned} \frac{\partial}{\partial t} &= \tau_t \frac{\partial}{\partial \tau} + \xi_t \frac{\partial}{\partial \xi} + \eta_t \frac{\partial}{\partial \eta} + \zeta_t \frac{\partial}{\partial \zeta} \\ \frac{\partial}{\partial x} &= \tau_x \frac{\partial}{\partial \tau} + \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \zeta_x \frac{\partial}{\partial \tau} \text{ , etc.} \end{aligned} \tag{A3}$$

and curvilinear derivatives are

$$\begin{aligned} \frac{\partial}{\partial \tau} &= t_\tau \frac{\partial}{\partial t} + x_\zeta \frac{\partial}{\partial x} + y_\tau \frac{\partial}{\partial y} + z_\tau \frac{\partial}{\partial z} \\ \frac{\partial}{\partial \xi} &= t_\xi \frac{\partial}{\partial \tau} + x_\xi \frac{\partial}{\partial x} + y_\xi \frac{\partial}{\partial y} + z_\xi \frac{\partial}{\partial z} \text{ , etc.} \end{aligned} \tag{A4}$$

Substitute (A3) into (A1), realizing $t = \tau \ast f(x, y, z)$ gives

$$\begin{aligned} &q_\tau + \xi_t q_\xi + \eta_t q_\eta + \zeta_t q_\zeta + \xi_x f_\xi + \eta_x f_\eta + \zeta_x f_\zeta \\ &\quad + \xi_y g_\xi + \eta_y g_\eta + \zeta_y h_\zeta + \xi_z h_\xi + \eta_z h_\eta + \zeta_z g_\zeta = 0 \end{aligned} \tag{A5}$$

At this point we have curvilinear derivatives of the dependent variables and flux terms (q, f, g, h) but we still have Cartesian derivatives of the independent variables (ξ, η, ζ, τ). We must, therefore, find expressions for the Cartesian derivatives in terms of curvilinear derivatives. To do this first write equation (4) in matrix form

$$[\text{curv deriv}] = [J]\ [\text{cart deriv}] \tag{A6}$$

and realize that the determinant of the coefficient matrix, $|[J]|$ is the matrix Jacobian, here denoted simply as J,

$$J = x_\xi(y_\eta z_\zeta - z_\eta y_\zeta) - y_\xi(x_\eta z_\zeta - z_\eta x_\zeta) + z_\xi(x_\eta y_\zeta - y_\eta x_\zeta) \tag{A7}$$

Multiplying (A6) by $[J]^{-1}$ gives the desired relation

$$[\text{cart deriv}] = [J]^{-1}\ [\text{curv deriv}] \tag{A8}$$

Apply (A8) to ξ_t, ξ_x, etc. to give

$$\begin{aligned} \xi_t &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z \\ \xi_x &= J^{-1}\ (y_\eta z_\zeta - z_\eta y_\zeta), \text{ and so on.} \end{aligned} \tag{A9}$$

Before equations (A9) are used in equation (A5), a simplification is possible. This simplification is achieved by multiplying (A5) by J and using the product rule for derivatives in the following particular form.

$$Jk_m \frac{\partial C}{\partial k} = \frac{\partial}{\partial k}\left(Jk_m C\right) - C \frac{\partial}{\partial k}\left(Jk_m\right) \tag{A10}$$

where k = (ξ, η, ζ, τ), m = (x, y, z, t) and C = (q, f, g, h), respectively. Doing so yields an equation of the form

$$\begin{aligned}
&\frac{\partial}{\partial \tau}\ [Jq]^{1} + \frac{\partial}{\partial \xi}\,[J(\xi_t q + \xi_x f + \xi_y g + \xi_z h)]^{\ 2} \ +\ 3\ +\ 4 \\
&\qquad - q\left[\frac{\partial}{\partial \tau}\,(J) + \frac{\partial}{\partial \xi}\,(J\xi_t) + \frac{\partial}{\partial \eta}\,(J\eta_t) + \frac{\partial}{\partial \zeta}\,(J\zeta_t)\right]^{5} \\
&\qquad - f\left[\frac{\partial}{\partial \xi}\,(J\xi_x) + \frac{\partial}{\partial \eta}\,(J\eta_x) + \frac{\partial}{\partial \zeta}\,(J\zeta_x)\right]^{\ 6} \ +\ 7\ +\ 8
\end{aligned} \tag{A11}$$

Terms 3 and 4 are just like 2 but with η or ζ replacing ξ. Similarly, terms 7 and 8 are like 6 both with g and y or h and z replacing f and x.

The bracketed quantities in terms 5 through 8 of this equation are the, so-called, invariants of the transformation and are zero. This can be shown in a 'brute force' sort of way as follows. Consider the bracketed quantity in term 6:

$$\frac{\partial}{\partial \xi}\,(J\xi_x) + \frac{\partial}{J\eta}\,(J\eta_x) + \frac{\partial}{\partial \zeta}\,(J\zeta_x)$$

chain rule

$$= J\,\frac{\partial}{\partial \xi}\ \xi_x + \xi_x \frac{\partial J}{\partial \xi} + J\,\frac{\partial}{\partial \eta}\,\eta_x + \eta_x \frac{\partial J}{\partial \eta} + J\,\frac{\partial}{\partial \zeta}\,\zeta_x + \zeta_x \frac{\partial J}{\partial \zeta}$$

rearrange

$$= J\left(\frac{\partial}{\partial \xi}\,\xi_x + \frac{\partial}{\partial \eta}\,\eta_x + \frac{\partial}{\partial \zeta}\,\zeta_x\right) + \xi_x \frac{\partial J}{\partial \xi} + \eta_x \frac{\partial J}{\partial \eta} + \zeta_x \frac{\partial J}{\partial \zeta}$$

change order of differentiation and use (A3)

$$= J\,\frac{\partial}{\partial x}\left(\frac{\partial}{\partial \xi}\,\xi + \frac{\partial}{\partial \eta}\,\eta + \frac{\partial}{\partial \zeta}\,\zeta\right) + \frac{\partial}{\partial x}\,J$$

evaluate

$$= J \frac{\partial}{\partial x} (3) + \frac{\partial}{\partial x} \left(x_\xi (y_\eta z_\zeta - y_\zeta z_\eta) + \cdots \right)$$

$$= 0 \quad + \quad 0 \quad \text{(if change order of differentiation).}$$

Terms 5, 7 and 8 behave similarly.

Equation (A11) is now of the form

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0 \qquad \text{(A12)}$$

where $Q = Jq$, $F = J(\xi_t q + \xi_x f + \xi_y g + \xi_z h)$

$$G = J(\eta_t q + \eta_x f + \eta_y g + \eta_z h), \; H = J(\zeta_t q + \zeta_x f + \zeta_y g + \zeta_z h) \qquad \text{(A13)}$$

The quantities ξ_f, ξ_x, $\cdots$, ζ_y, ζ_z can be replaced by equations (A9) to finally yield an expression entirely in terms of curvilinear derivatives, and the transformation is thus complete.

Appendix B. Equation Hierarchy

Equations (16) and (18) actually represent systems of equations imbedded in a system of equations. That is what is meant, for example, by the description that (16) is a lower <u>block</u> bidiagonal system. The block is, itself a system of equations. To show this clearly, let us consider a different factorization of equation (12) which will yield the same hierarchy of systems as in (16) and (18) but with fewer terms. The structure of the systems will then not be obscured with lengthy expressions.

Equation (12) can be factored as

$$(I + \Delta\tau\delta_i A^{+}\cdot)(I + \Delta\tau\delta_i A^{-}\cdot)(I + \Delta\tau\delta_j B^{+}\cdot)(I + \Delta\delta_j B^{-}\cdot)(I + \delta_k C^{+}\cdot)(I + \delta_k C^{-}\cdot)\Delta Q^n = -\Delta\tau R^n \tag{B1}$$

and solved in a six step scheme

$$\begin{aligned}
(I + \Delta\tau\delta_i A^{+}\cdot)X^1 &= -\Delta\tau R^n \\
(I + \Delta\tau\delta_i A^{-}\cdot)X^2 &= X^1 \\
&\vdots \\
(I + \Delta\tau\delta_k C^{-}\cdot)X^6 &= X^5 \\
\Delta Q^n &= X^6
\end{aligned} \tag{B2}$$

The motivation for this factorization and scheme and the solution process are exactly the same as described for equations (13) and (14).

Let us consider the first equation in (B21),

$$(I + \Delta\tau\delta_i A^{+}\cdot)X^1 = X^1 + \Delta\tau\delta_i(A^{+}X_1) = -\Delta\tau R^n \tag{B3}$$

and write out the difference operator with indices as described for equations (16) - (19).

$$X^1_{i,j,k} + \Delta\tau[A^+(Q^n_{i,j,k})]_{i,j,k}X^1_{i,j,k} - \Delta\tau[A^+(Q^n_{i-1,j,k})]_{i-1,j,k}X^1_{i-1,j,k} = -\Delta\tau R^n_{i,j,k} \qquad \text{(B4)}$$

We must realize several facts concerning equation (B4) and its notation. The (i,j,k) subscripts refer to spatial location, the 1 superscript to the solution pass and the n superscript to the time level. In addition, at each (i,j,k) location there are 5 distinct Q^n (representing $\rho,\rho u,\rho v,\rho w,e$) and consequently 5 distinct X^1. Finally, A^+ denotes $\frac{\partial F\ell^+}{\partial Q_m}$ so that there are 25 A^+ terms since there are 5 flux vectors F^+_ℓ.

To display the structure clearly, suppose there are only 6 cells in the i direction. Also suppose we are evaluating (B4) along a line of constant j and k so that the (j,k) indices can be dropped. Further, let us drop the superscripts for this example and finally let A stand for $\Delta\tau A^+$. Then (B4) becomes

$$X_i + A_iX_i - A_{i-1}X_{i-1} = -\Delta\tau R_i \quad \text{with } i = 2,6 \qquad \text{(B5)}$$

Writing (B5) out gives

$$\begin{aligned} X_1 \qquad\qquad\qquad &= X_1 \\ -A_1X_1 + (1+A_2)X_2 &= -\Delta\tau R_2 \\ -A_2X_2 + (1+A_3)X_3 &= -\Delta\tau R_3 \\ -A_3X_3 + (1+A_4)X_4 &= -\Delta\tau R_4 \\ -A_4X_4 + (1+A_5)X_5 &= -\Delta\tau R_5 \end{aligned}$$

$$- A_5 X_5 + (1 + A_6)X_6 = -\Delta\tau R_6$$

or, in matrix form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ A_1 & 1+A_2 & 0 & 0 & 0 & 0 \\ 0 & A_2 & 1+A_3 & 0 & 0 & 0 \\ 0 & 0 & A_3 & 1+A_4 & 0 & 0 \\ 0 & 0 & 0 & A_4 & 1+A_5 & 0 \\ 0 & 0 & 0 & 0 & A_5 & 1+A_6 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} = -\Delta\tau \begin{bmatrix} \frac{X_1}{-\Delta\tau} \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \end{bmatrix} \qquad \text{(B6b)}$$

The lower bidiagonal form of (B6b) is now clear. Remember that the subscripts 1 through 6 here refer to <u>spatial</u> locations.

Now let us consider the details of any one of the equations in (B6a) or (B6b). That is, we now have a fixed value of i and therefore are considering a particular location in space. Since (B5) represents a typical equation in (B6), we can use (B5) realizing that i is fixed and rearrange to give

$$(1 + A_i)X_i = -\Delta\tau R_i + A_{i-1}X_{i-1} \qquad \text{(B7)}$$

But, at each i there are 5 distinct components of X (say, x_ℓ, ℓ = 1,5) and 25 distinct A terms. Thus (B7) actually expresses the following matrix equation

$$\begin{bmatrix} 1+A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & 1+A_{22} & \cdots & & \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ A_{51} & \cdots & & & 1+A_{55} \end{bmatrix}_i \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}_i = \Delta\tau \begin{bmatrix} -R_{1_i} + (A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + A_{14}x_4 + A_{15}x_5)_{i-1} \\ \cdot \\ \cdot \\ \cdot \\ -R_{5_i} + (A_{51}x_1 + A_{52}x_2 + A_{53}x_3 + A_{54}x_4 + A_{55}x_5)_{i-1} \end{bmatrix} \qquad \text{(B8)}$$

Here the number subscripts refer to the dependent variables $(Q_1, Q_2, Q_3, Q_4, Q_5) = (\rho, \rho u, \rho v, \rho w, e)$ while the i subscripts refer to the location, which is now fixed. So, each equation in the factored scheme (B2) represents a system of equations (B6b) over space. Each equation in system (B6b), in turn, represents a system of equations (B8) over the dependent variables. This last system (B8) is the "block" in the block bidiagonal descriptor. Notice that this system (B8) is similar to system (8) in the main text.

The procedure described above would be carried out for each of the six steps in system (B2), with appropriate care in evaluating the difference operators as described for equations (17) and (19). As a note, the six step scheme of (B2) satisfactorily provides solutions to the Euler equations. It requires more computation than the two step scheme in the present program but requires less memory and thus is advantageous when memory limitations are significant.

Relating these ideas back to the present scheme, we see that equations (14) correspond to (B2), equation (16) to (B4), and (17) to (B7). Thus, the first equation in the factored scheme (14) represents a lower block bidiagonal system of equations, (16). Each equation in system (16), in turn, represents a system of equations (17) over the dependent variables for fixed (i,j,k). In a similar fashion, the second equation in (14) represents an upper block bidiagonal system which is expressed in equation (18). For fixed (i,j,k) each equation in system (18) represents a system of equations, represented by equation (19), over the dependent variables.

Appendix C. Eigensystems of the Transformed Euler Equations

The development of the eigenvalues, eigenvectors and split flux vectors of the transformed Euler equations is briefly presented here. This presentation will be in condensed notation to emphasize the essential steps. Details of the development, particularly in regard to the actual elements in the various matrices are available in References 1 and 2. In addition, several theorems and operations of matrices and linear algebra are required and these are documented in any quality text on linear algebra or numerical analysis.

To aid in the description it is helpful to first define some notation. Certain terms represent any one of three terms depending on the direction (in computational space) of interest. These are

$k = \xi,\eta,\zeta$ = curvilinear directions

$K = F,G,H$ = flux vectors (see equation (2) in main text)

$\bar{K} = \dfrac{\partial K}{\partial Q} = A,B,C$ = flux Jacobian matrices, where the Q are the dependent variables in the conservative form of the Euler equations (eqn (2))

$\kappa = a,b,c$ = coefficient matrices for the nonconservative Euler equations

Typically these are used such that when $k = \xi$ then $K=F$, $\bar{K}=A$ and $\kappa=a$, and so on for $k=\eta$ and $k=\zeta$. Another item of notation here is that when (τ,ξ,η,ζ) appear as subscripts they refer to derivatives $(\partial/\partial\tau,\ \partial/\partial\xi,\ \partial/\partial\eta,\ \partial/\partial\zeta)$. similarly, (x,y,z) subscripts mean $(\partial/\partial x,\ \partial/\partial y,\ \partial/\partial z)$. All other subscripts will not indicate differentiation.

We may now proceed with the development. The starting place is the Euler equations in conservative form and in curvilinear coordinates,

$$Q_\tau + F_\xi + G_\eta + H_\zeta = 0 \tag{C1}$$

This can be written in the quasilinear form

$$Q_\tau + AQ_\xi + BQ_\eta + CQ_\zeta = 0 \qquad \text{(C2)}$$

where, recall, $A = \partial F/\partial Q$, $B = \partial G/\partial Q$, $C = \partial H/\partial Q$ are terms of $\bar{K}$.

With the governing equation written as (C2) the eigenvalues of the governing equation are, in fact, the eigenvalues of the flux Jacobian matrices $\bar{K}$. However, when $\bar{K}$ is actually written out (see Appendix D) it is seen that there are very few zero elements in $\bar{K}$ and consequently that it would be a difficult task to extract the eigenvalues of $\bar{K}$. To find the eigenvalues, a similarity transformation (that is, one which preserves the eigenvalues) of $\bar{K}$ is sought which yields a matrix with many more zero elements and hence makes extraction of the eigenvalues of $\bar{K}$more tractable. This transformation is achieved by considering the nonconservative form of the Euler equations

$$q_\tau + aq_\xi + bq_\eta + cq_\zeta = 0 \qquad \text{(C3)}$$

where $q = J[\rho,u,v,w,p]^T$.

Now (C2) can be written as

$$Mq_\tau + AMq_\xi + BMq_\eta + CMq_\zeta = 0 \qquad \text{(C4)}$$

where $M = \partial Q/\partial q$.

Multiply (C4) by M^{-1} to give (with I = identity matrix)

$$Iq_\tau + M^{-1}AMq_\xi + M^{-1}BMq_\eta + M^{-1}CMq_\zeta = 0 \qquad \text{(C5)}$$

Comparing (C3) and (C5) we see that

$$a = M^{-1}AM,\ b = M^{-1}BM,\ c = M^{-1}CM \qquad \text{(C6a)}$$

and, therefore, $\kappa = M^{-1}\bar{K}M$, $\bar{K} = M\kappa M^{-1}$ (C6b)

Equation (C6b) is the required similarity transformation. The matrices $\bar{K}$ and κ are similar and therefore have the same eigenvalues. Writing out κ we have

$$\kappa = \begin{bmatrix} \theta_k & \rho k_x & \rho k_y & \rho k_z & 0 \\ 0 & \theta_k & 0 & 0 & k_{x/\rho} \\ 0 & 0 & \theta_k & 0 & k_{y/\rho} \\ 0 & 0 & 0 & \theta_k & k_{z/\rho} \\ 0 & k_x\rho c^2 & k_y\rho c^2 & k_z\rho c^2 & \theta_k \end{bmatrix} \qquad \begin{aligned} &\text{where } \theta_k = k_x u + k_y v + k_z w \\ &\rho c^2 = \gamma((\gamma-1)e - \rho\phi) \\ &\phi = \frac{\gamma-1}{2}(u^2+v^2+w^2) \end{aligned} \qquad (C7)$$

The eigenvalues, λ_k, are found from the solution of

$$|\kappa - I\lambda_k| = 0 \qquad (C8)$$

and are

$$\lambda_k^1 = \lambda_k^2 = \lambda_k^3 = \theta_k, \quad \lambda_k^4 = \theta_k + c|\nabla k|, \quad \lambda_k^5 = \theta_k - c|\nabla k| \qquad (C9)$$

with $|\nabla k| = (k_x^2 + k_y^2 + k_z^2)^{1/2}$.

The eigenvalues have been found and now the task is to find the eigenvectors of $\bar{K}$. The eigenvectors, T_k, corresponding to the eigenvalues are such that

$$\bar{K} = T_k \Lambda_k T_k^{-1} \qquad (C10)$$

where Λ_k is the diagonal matrix whose diagonal elements are the eigenvalues, (C9). To determine T_k, consider a similar expression for κ

$$\kappa = P_k \Lambda_k P_k^{-1} \qquad (C11)$$

The matrix P_k is also composed of eigenvectors corresponding to the eigenvalues (C9). Here the columns of P_k are the right eigenvectors of κ, each column being a linearly independent set of eigenvectors. These eigenvectors are normalized such that multiplying the matrix of the right eigenvectors times the matrix of the left eigenvectors gives the identity matrix. (As explanation, given a matrix [A] and $[A]x_r = \lambda x_r$ then λ is an eigenvalue and x_r is the corresponding right eigenvector. Similarly given [A] and $x_\ell[A]=x_\ell\lambda$ then λ, again, is an eigenvalue and x_ℓ is the corresponding left eigenvector.) It is much easier to find P_k (refer to a good linear algebra text), than T_k. Once P_k is found then T_k can be readily obtained by:

from (C6b) $\bar{K} = M\kappa M^{-1}$ and using (C11) then, $\bar{K} = MP_k\Lambda_k P_k^{-1}M^{-1}$

so that $T_k = MP_k$, $T_k^{-1} = M^{-1}P_k^{-1}$ (C12)

The eigenvalues and eigenvectors of the flux Jacobian matrices are now available in (C9) and (C12). It remains to obtain the flux vectors as a linear expansion in the eigenvalues (eqn (10)). Because the flux vectors are homogeneous of degree one in the Q's we can write, by invoking Euler's Theorem

$$K = \bar{K}Q \tag{C13}$$

(Essentially, a homogeneous function f(x), say, of degree n is one such that $f(\alpha x) = \alpha^n f(x)$ where α is a constant. Euler's Theorem essentially says that for f(x) continuously differentiable and homogeneous of degree n, then x df/dx = nf. These ideas can be extended to functions of several variables.)

Using (C10) in (C13) gives

$$K = T_k\Lambda_k T_k^{-1}Q \tag{C14}$$

We can write the diagonal matrix Λ_k as the sum

$$\Lambda_k = \Lambda_k^1 I_{1,2,3} + \lambda_k^4 I_4 + \lambda_k^5 I_5 \tag{C15}$$

where $I_{1,2,3}$ is a matrix with 1 as the first 3 diagonal elements and 0 elsewhere, I_4 has 1 as the fourth diagonal element with 0 elsewhere and I_5 has 1 as the fifth diagonal element with 0 elsewhere. Then

$$K = \lambda_k^1 T_k I_{1,2,3} T_k^{-1} Q + \lambda_k^4 T_k I_4 T_k^{-1} Q + \lambda_k^5 T_k I_5 T_k^{-1} Q \tag{C16a}$$

$$= \lambda_k^1 K_1 + \lambda_k^4 K_2 + \lambda_k^5 K_3 \tag{C16b}$$

Equation (C16) is the expansion of the flux vectors in the eigenvectors and is the desired split form. When the details of K_1,K_2 and K_3 are determined we finally have

$$K_1 = J\,\frac{\gamma-1}{\gamma}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \dfrac{\rho\phi}{(\gamma-1)}\end{bmatrix}\quad K_2 = \frac{J}{2\gamma}\begin{bmatrix}\rho\\ \rho u+\rho c\tilde{k}_x\\ \rho v+\rho c\tilde{k}_y\\ \rho w+\rho c\tilde{k}_z\\ e+p+\rho c\tilde{\theta}_k\end{bmatrix}\quad K_3 = \frac{J}{2\gamma}\begin{bmatrix}\rho\\ \rho u-\rho c\tilde{k}_x\\ \rho v-\rho c\tilde{k}_y\\ \rho w-\rho c\tilde{k}_z\\ e+p-\rho c\tilde{\theta}_k\end{bmatrix} \qquad (C17)$$

where $(\tilde{k}_x,\tilde{k}_y,\tilde{k}_z,\tilde{\theta}_k) = (k_x,k_y,k_z,\theta_k)/|\nabla k|$ and ϕ and θ_k are defined with equation (C7).

Appendix D. Evaluation of the Flux Jacobian Matrices

The dependent variables of the conservative, transformed Euler equations are

$$Q = Q_m, \ m = 1,5 = J[\rho,\rho u,\rho v,\rho w,e]^T = [J\rho,J\rho u,J\rho v,J\rho w,Je]^T \qquad (2)$$

The flux vectors are K = F,G,H and are given by

$$K = \lambda_k^1 K_1 + \lambda_k^4 K_2 + \lambda_k^5 K_3 \qquad (C16b)$$

where the subvectors $K_{1,2,3}$ are given in equation (C17) and $\lambda_k^{1,4,5}$ are given by equation (C8). The flux Jacobian matrices are formed by differentiating K with respect to Q,

$$A_{\ell m} = \partial F_\ell/\partial Q_m, \quad B_{\ell m} = \partial G_\ell/\partial Q_m, \quad C = \partial H_\ell/\partial Q_m \qquad (6b)$$

In order to carry out the differentiations, it is necessary to write the flux vectors explicitly in terms of the conserved variables, Q. For example, the second element in the K_2 subvector, K_{22}, is

$$K_{22} = \frac{J}{2\gamma}(\rho u + \rho c\tilde{k}_x) = \frac{1}{2\gamma}(J\rho u + J\rho c\tilde{k}_x) \qquad (C17)$$

Now, from (C7)

$$\rho c^2 = \gamma((\gamma-1)e - \rho\phi) = \gamma(\gamma-1)(e - \frac{1}{2}\rho(u^2 + v^2 + w^2))$$

and we can write

$$J\rho c = (J\rho * J\rho c^2)^{1/2} = (\gamma(\gamma-1))^{1/2}(J\rho * J(e - \frac{1}{2}\rho(u^2+v^2+w^2))^{1/2} \qquad (D1a)$$

or

$$J\rho c = (\gamma(\gamma-1))^{1/2}(J\rho * Je - \frac{1}{2}\ ((J\rho u)^2 + (J\rho v)^2 + (J\rho w)^2))^{1/2} \qquad \text{(D1b)}$$

$$= (\gamma(\gamma-1))^{1/2}(Q_1Q_5 - \frac{1}{2}(Q_2^2 + Q_3^2 + Q_4^2))^{1/2} \quad \text{(using (2))} \qquad \text{(D1c)}$$

So, K_{22} becomes

$$K_{22} = \frac{1}{2\gamma}(Q_2 + (\gamma(\gamma-1))^{1/2}(Q_1Q_5 - \frac{1}{2}(Q_2^2 + Q_3^2 + Q_4^2))^{1/2}\ \tilde{k}_x) \qquad \text{(D2)}$$

We now turn our attention to λ_k^4,

$$\lambda_k^4 = \theta_k + c|\nabla k| = k_xu + k_yv + k_zw + |\nabla k|\ \rho\ \frac{c}{\rho} \qquad \text{(D3a)}$$

$$= \frac{1}{J\rho}\ (k_xJ\rho u + k_yJ\rho v + k_zJ\rho w + |\nabla k|J\rho c) \qquad \text{(D3b)}$$

Using (2) and (D1c) in (D3b) gives

$$\lambda_k^4 = \frac{1}{Q_1}(k_xQ_2+k_yQ_3+k_zQ_4+|\nabla k|(\gamma(\gamma-1))^{1/2}(Q_1Q_5 - \frac{1}{2}\ (Q_2^2+Q_3^2+Q_4^2))^{1/2}) \qquad \text{(D4)}$$

Finally, to construct the contribution of K_{22} to K we must multiply (D2) and (D4) together. Clearly, the contribution is a lengthy expression. Moreover, this product is only one of the fifteen terms which compose K from (C16b). Suffice to say, the total expression for K is quite involved.

Once all fifteen terms of K are constructed, in the manner described above, the derivatives with respect to the five Q's can be obtained. Now, we must remember that the solution algorithm is based not only on splitting the flux vectors into terms with eigenvalues as coefficients but the fluxes are further split according to the sign of the eigenvalue, as expressed in (10c). The flux Jacobians are correspondingly split, as in (11). Consequently the fifteen terms of K must be separated into terms with positive and negative eigenvectors; differentiation then yields the appropriate flux Jacobian matrices $A^{\pm}$, $B^{\pm}$ and $C^{\pm}$.

In the program the derivatives of all fifteen terms for each of A, B and C are computed. Coefficients of either 1 or 0 are then assigned to each derivative according to whether the corresponding eigenvalue is positive or negative and which flux Jacobian, the plus or minus, is being evaluated. This procedure is organized in an efficient and orderly, although quite lengthy, scheme.

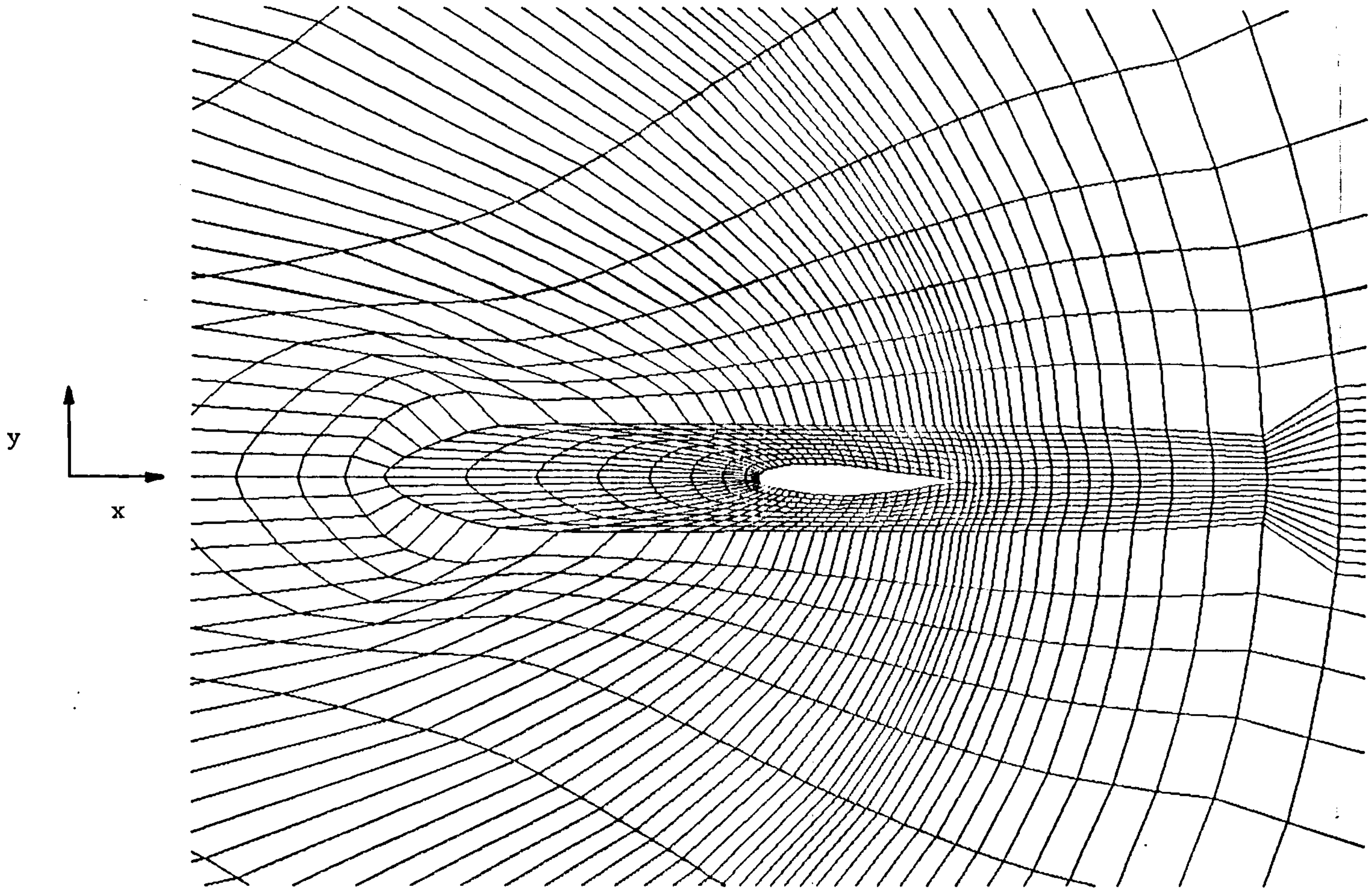

Figure 1. Portion of the K=1 surface for the Pathfinder geometry.

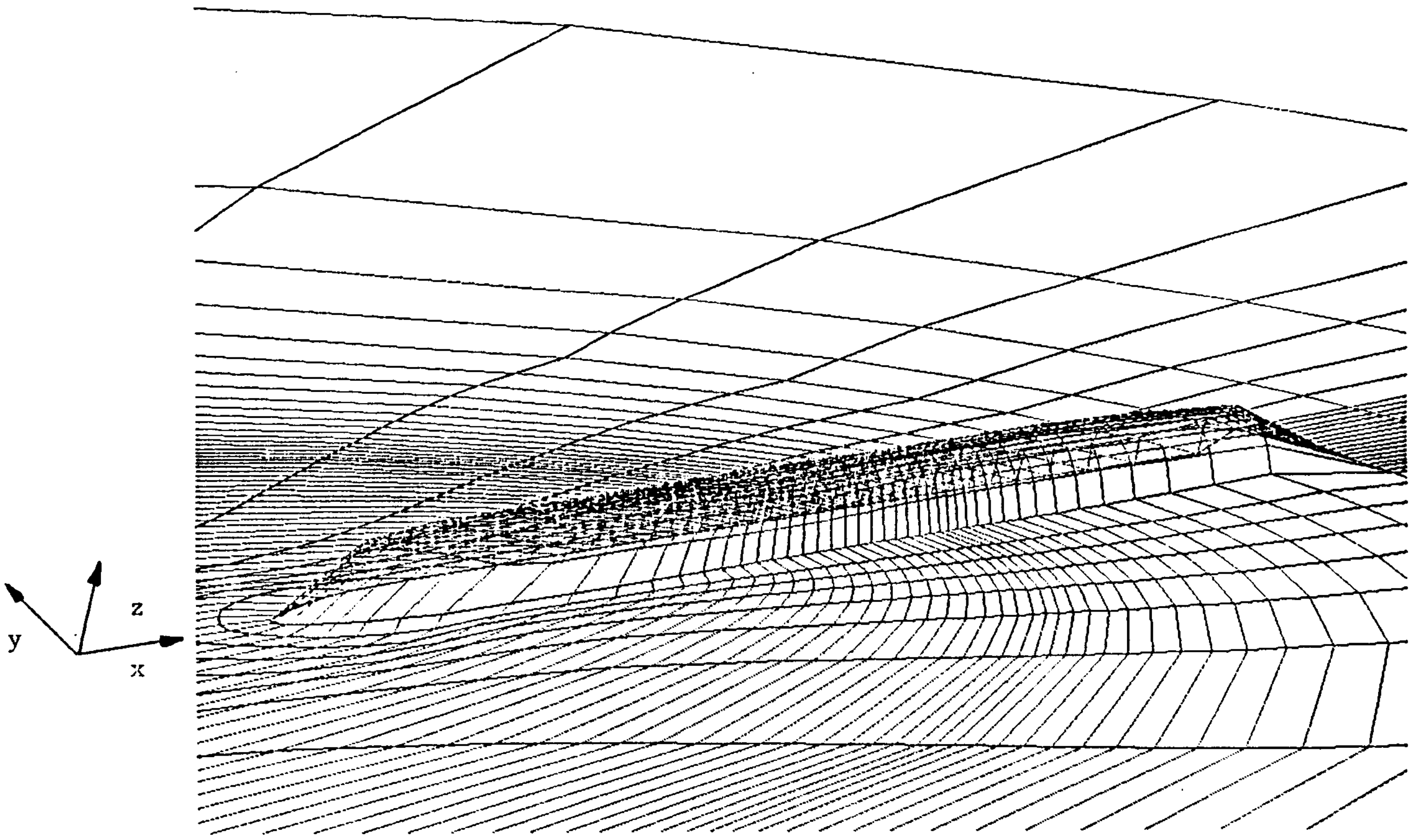

Figure 2. Oblique view of the K=1 surface.

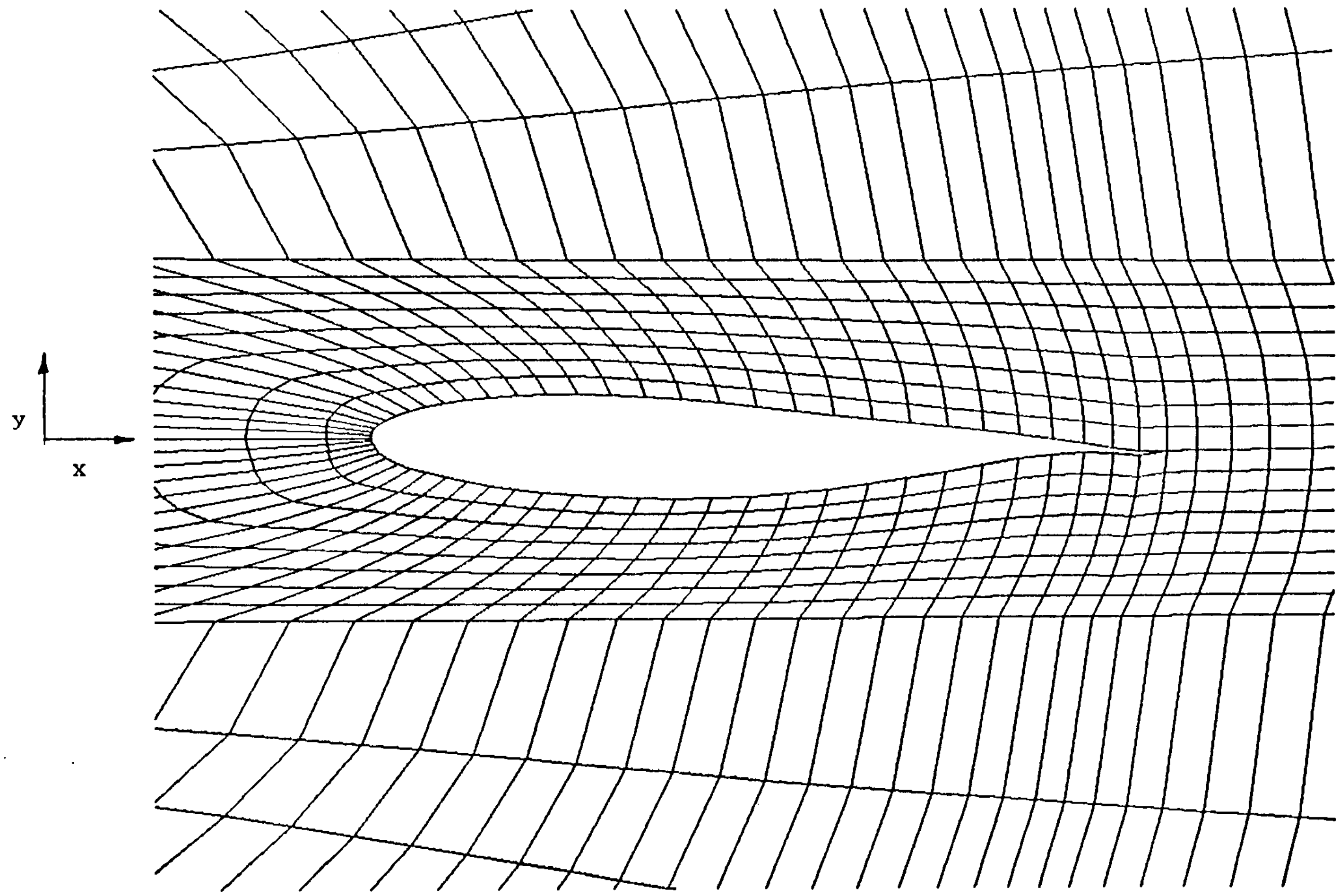

Figure 3. Closeup of the root section of the K=1 surface.

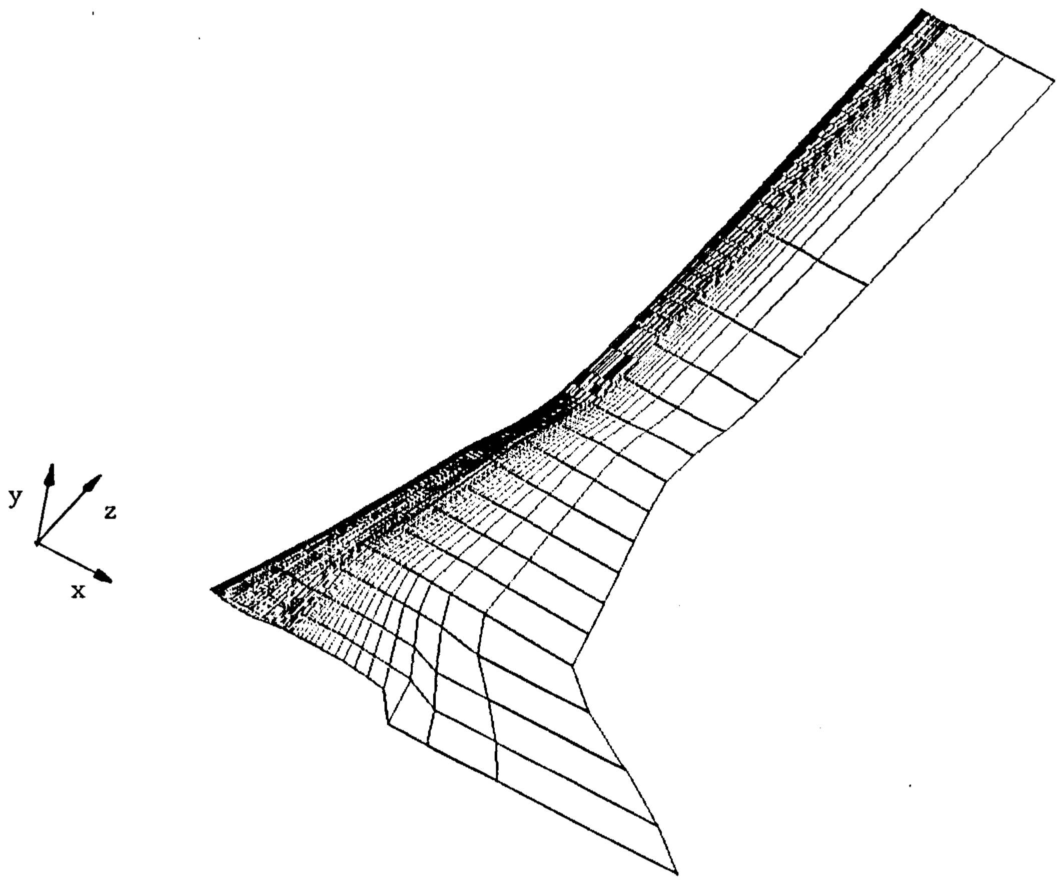

Figure 4. Oblique view of the J=1 surface for the Pathfinder geometry.

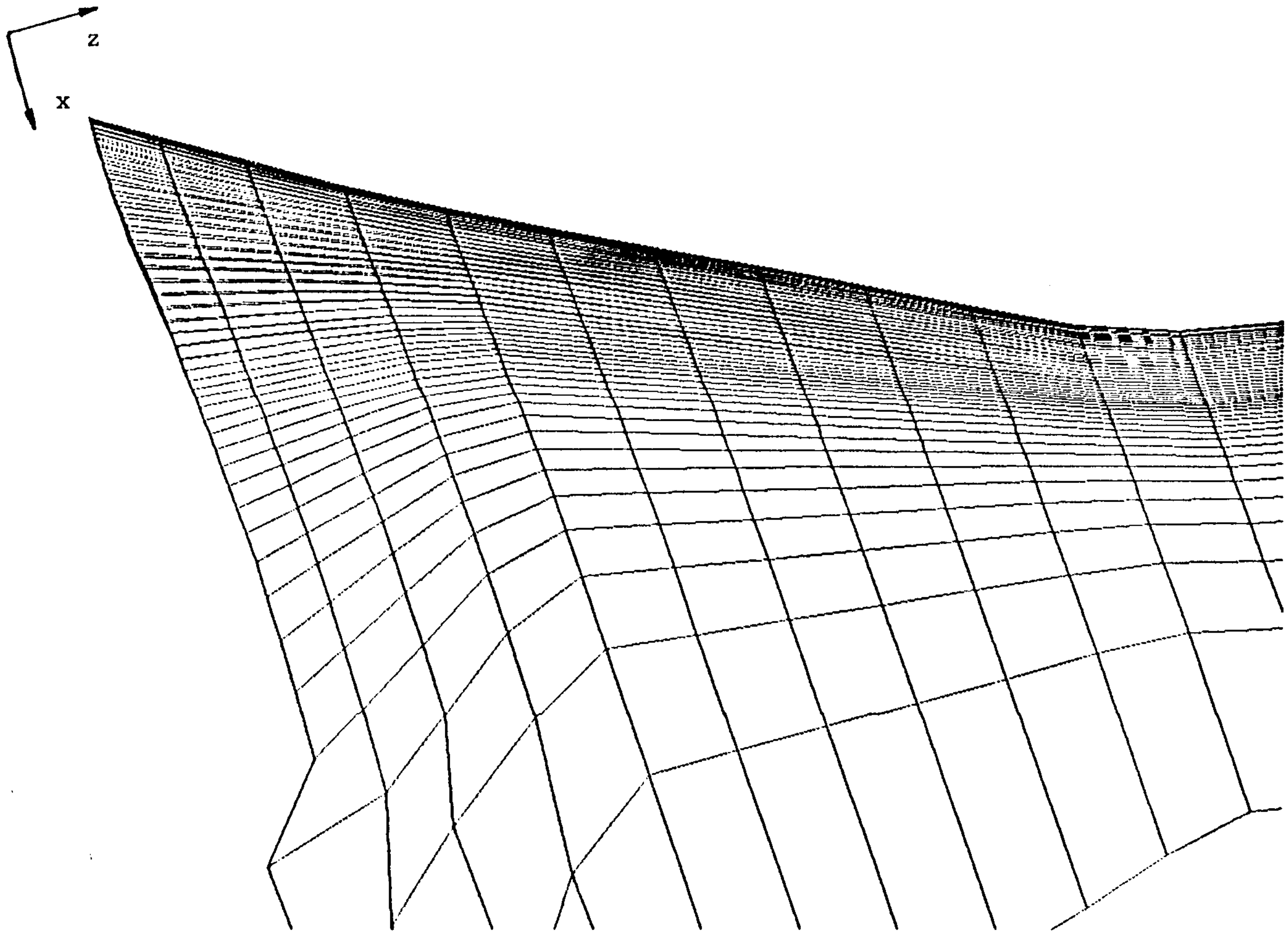

Figure 5. Normal view of the J=1 surface near the wing.

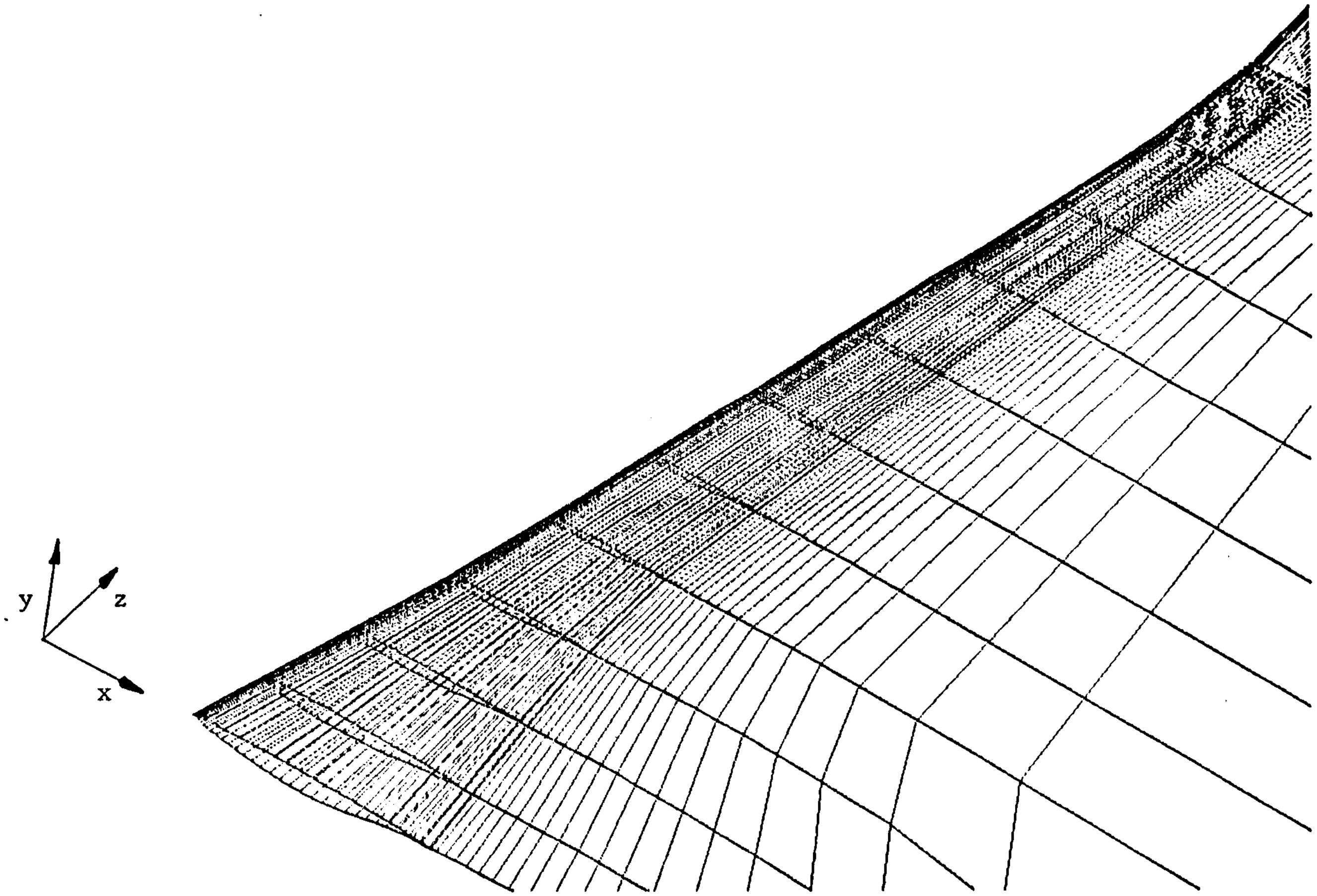

Figure 6. Oblique view of the J=1 surface near the wing.

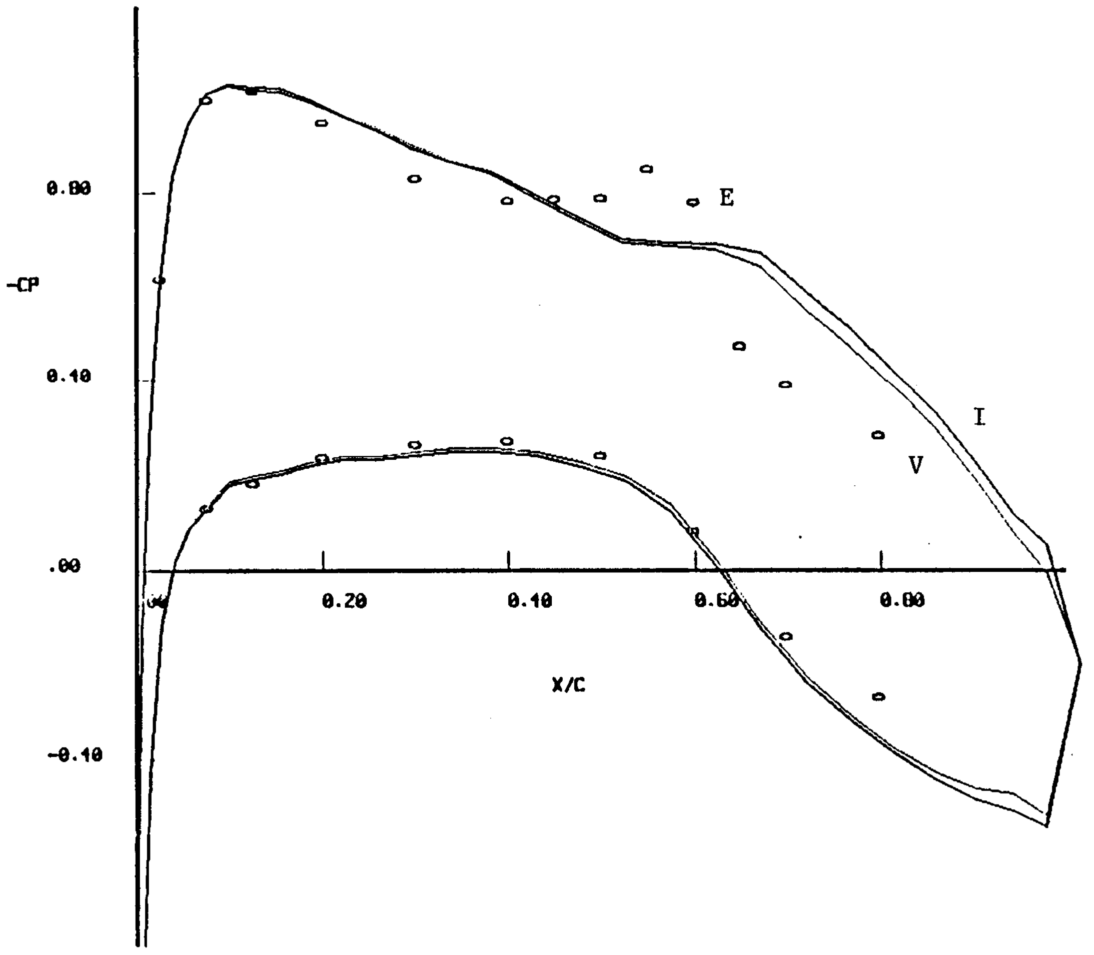

Figure 7a. Section pressure distribution, M_∞=0.82, α=2°, Re=9.9*10^6; E = experiment, 43.2% span; I = inviscid, 45% span; V = viscous, 45% span.

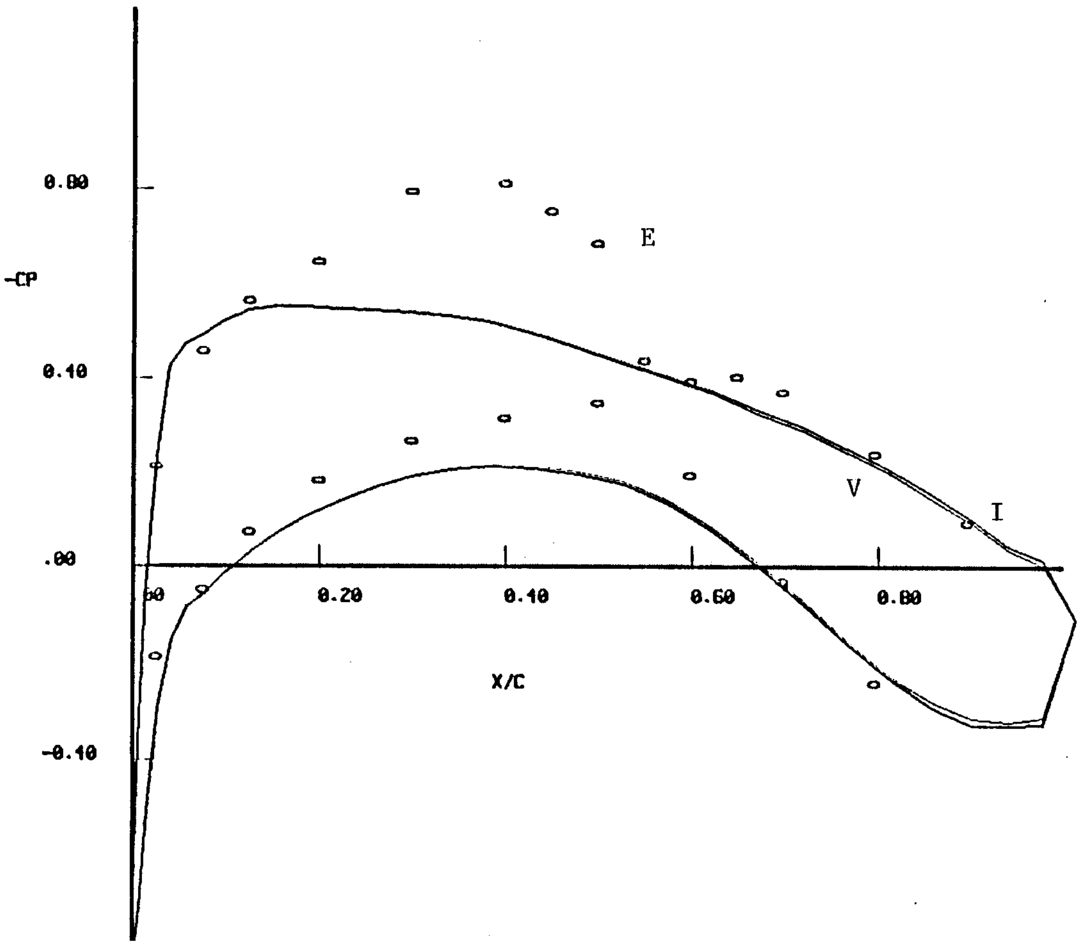

Figure 7b. E, 13.1%; I and V, 15%.

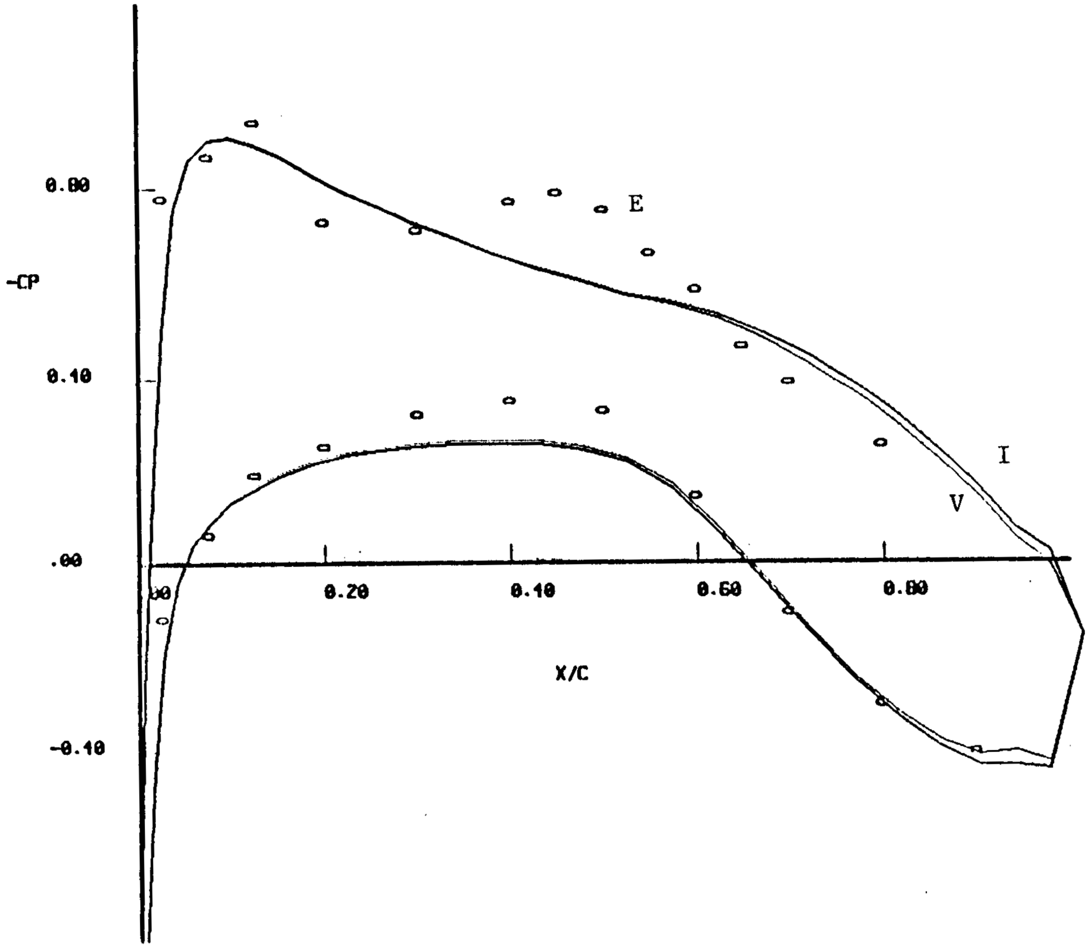

Figure 7c. E, 29.2%; I and V, 30%.

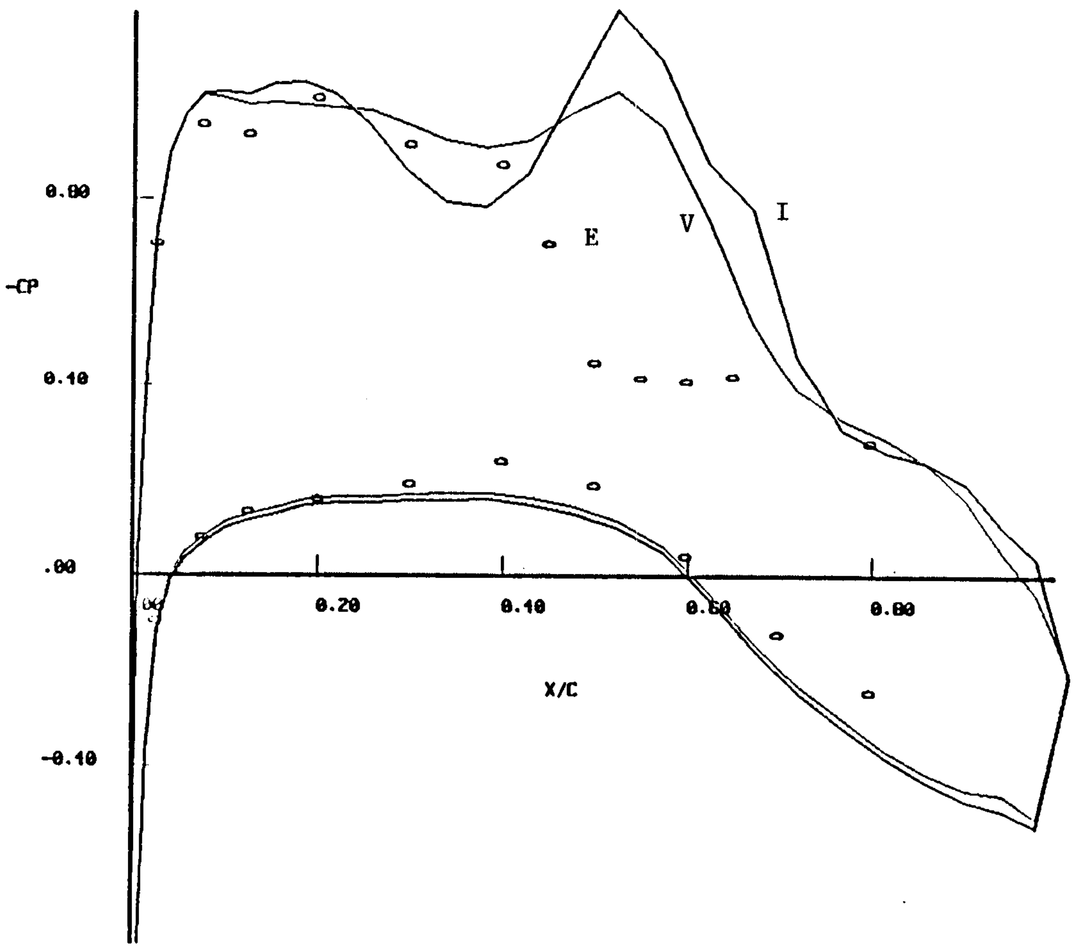

Figure 7d. E, 64%; I and V, 65%.

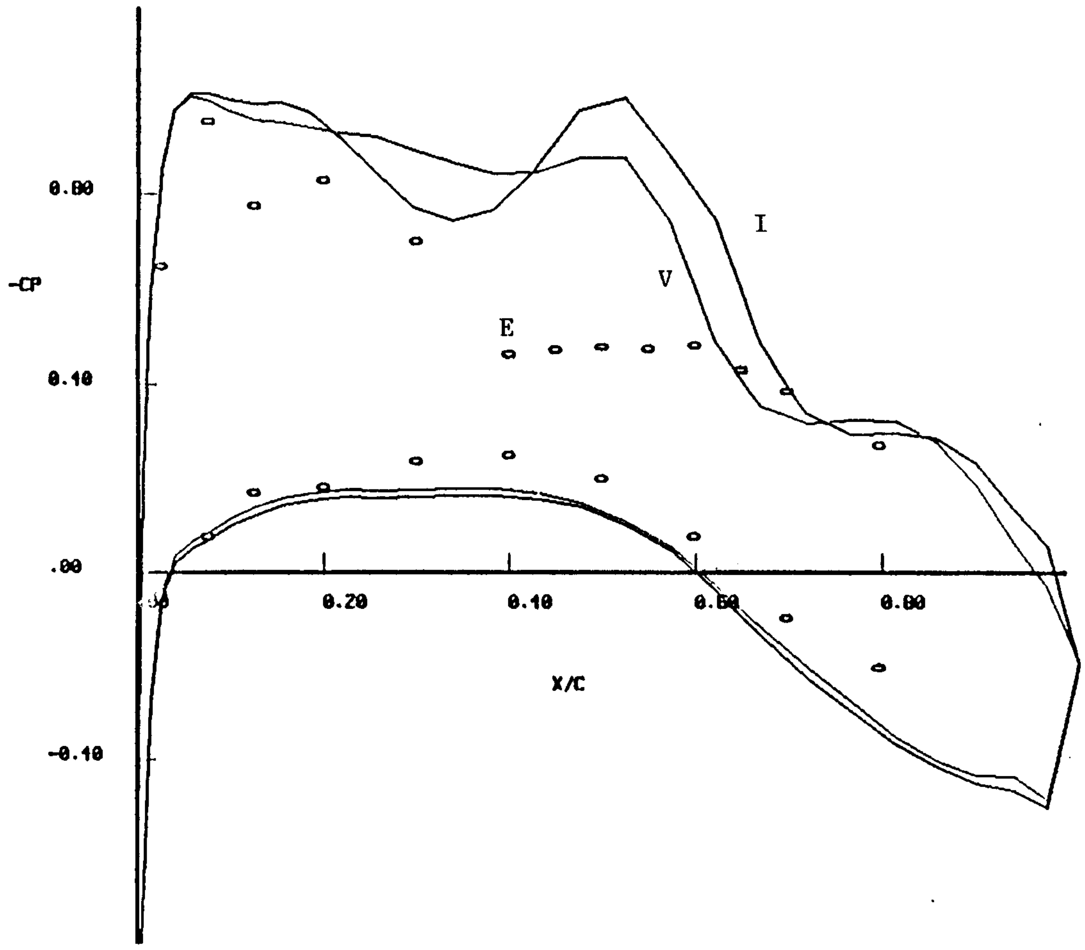

Figure 7e. E, 84%; I and V, 85%.

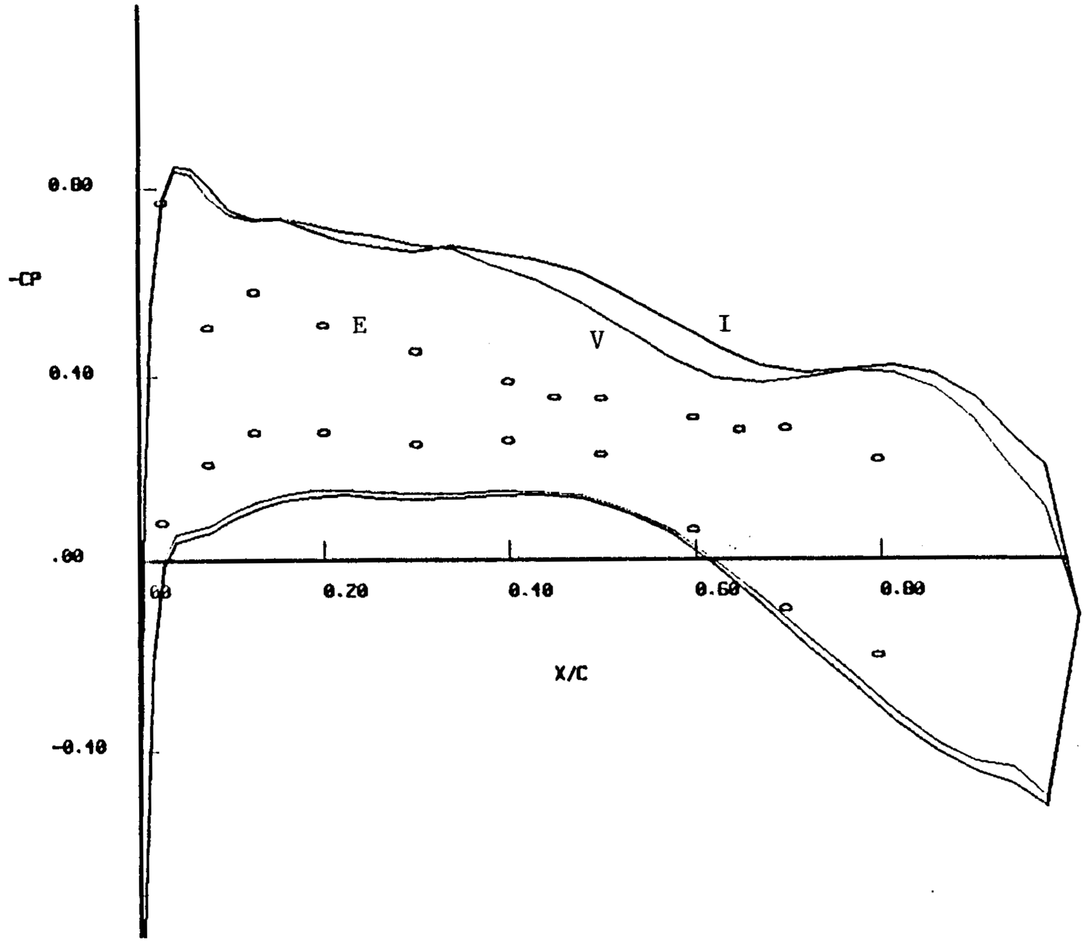

Figure 7f. E, 96.1%; I and V, 95%.

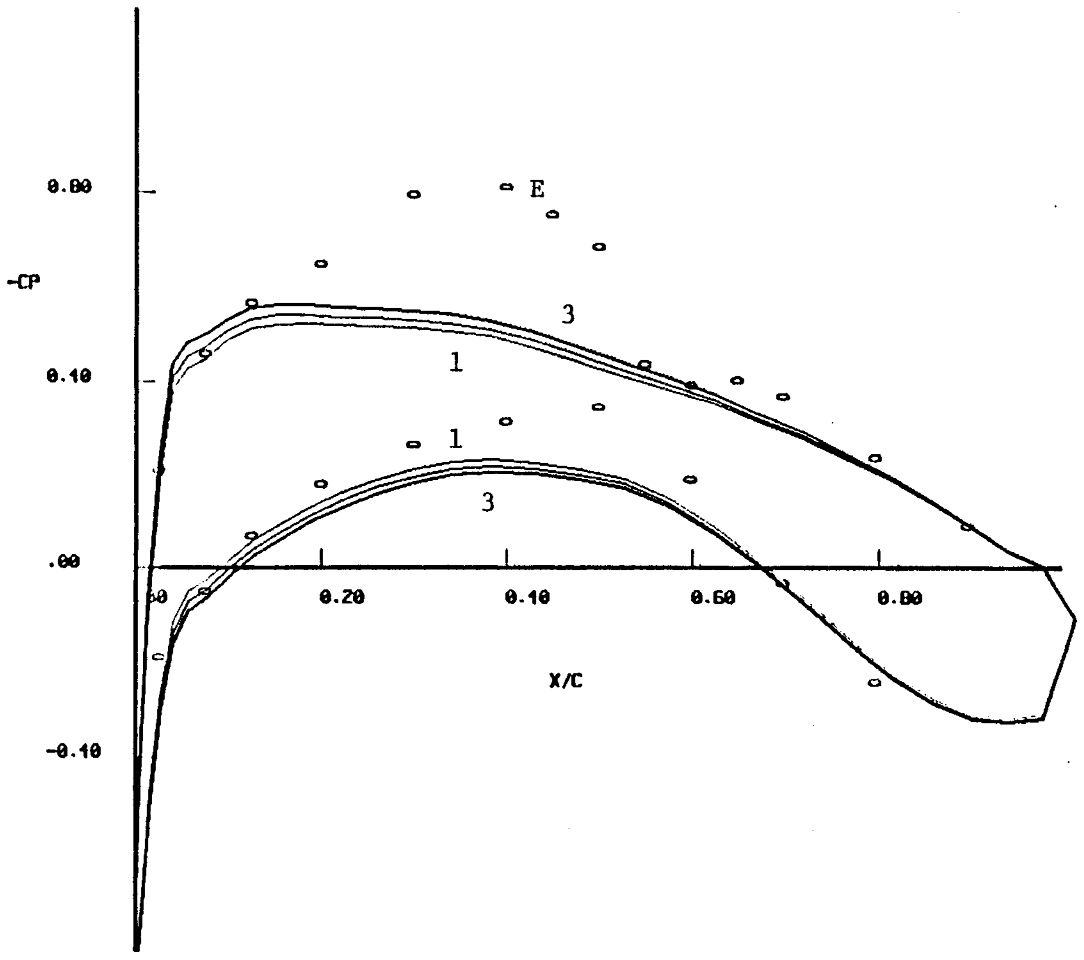

Figure 8a. Section pressure distribution, $M_\infty=0.82$, $\alpha=2°$, $Re=9.9*10^6$; E = experiment, 13.1% span; (1)=viscous, 2°, (2)=viscous, 2.3°, (3)=viscous, 2.6°; all at 15% span.

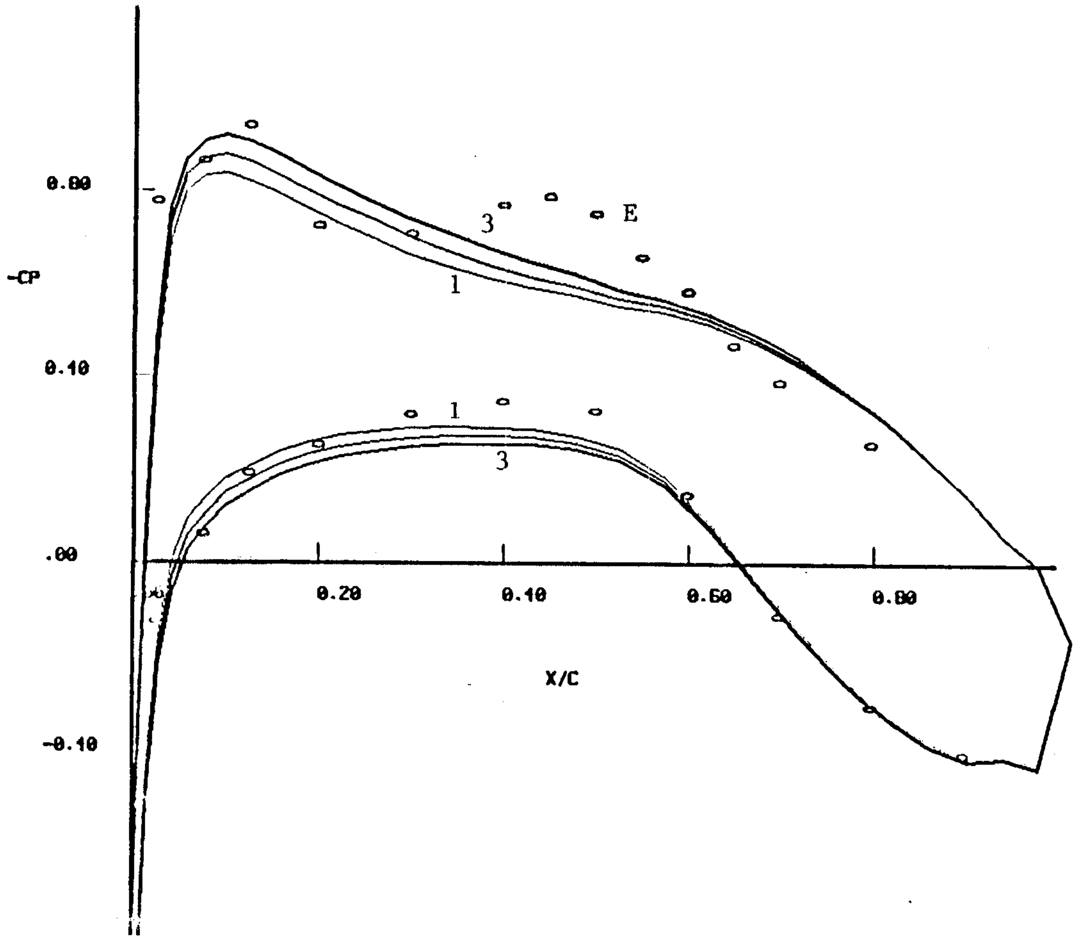

Figure 8b. E, 29.2%; (1), (2), and (3), 30%.

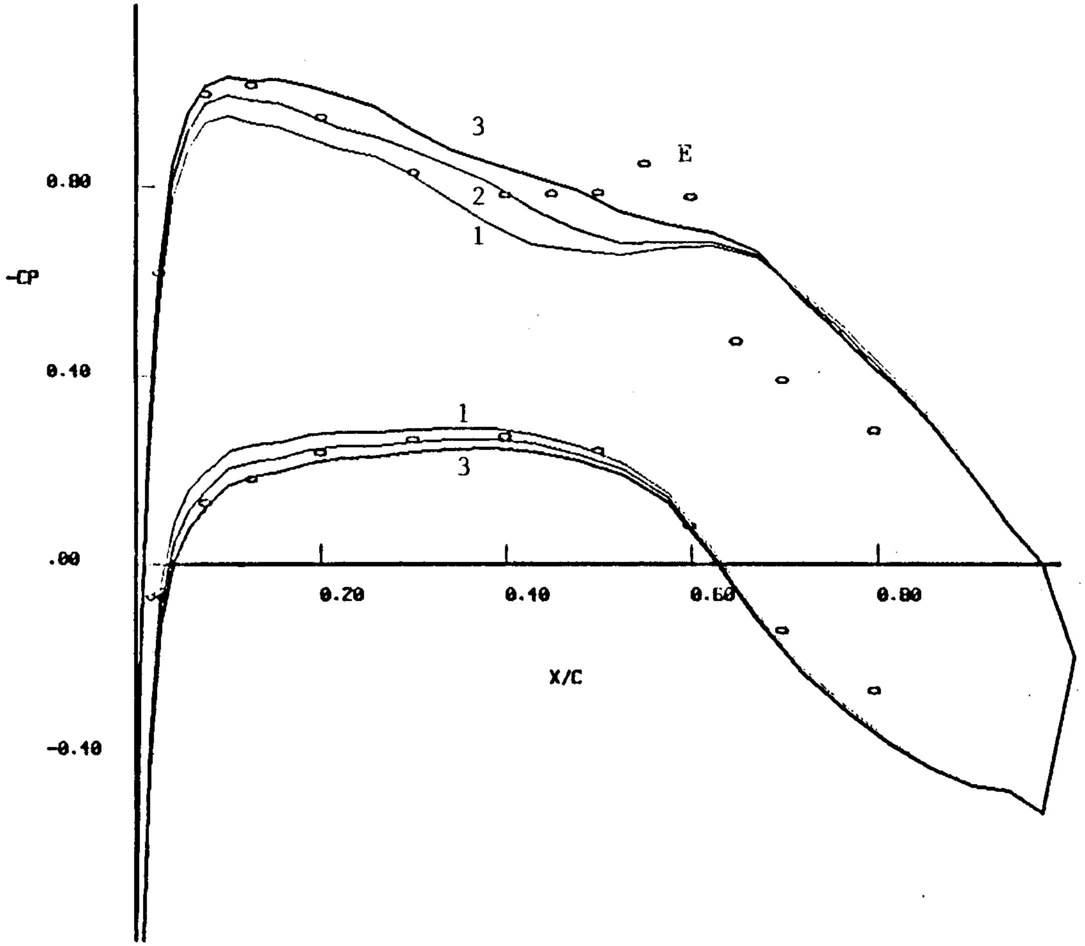

Figure 8c. E, 43.2%, (1), (2), and (3), 45%.

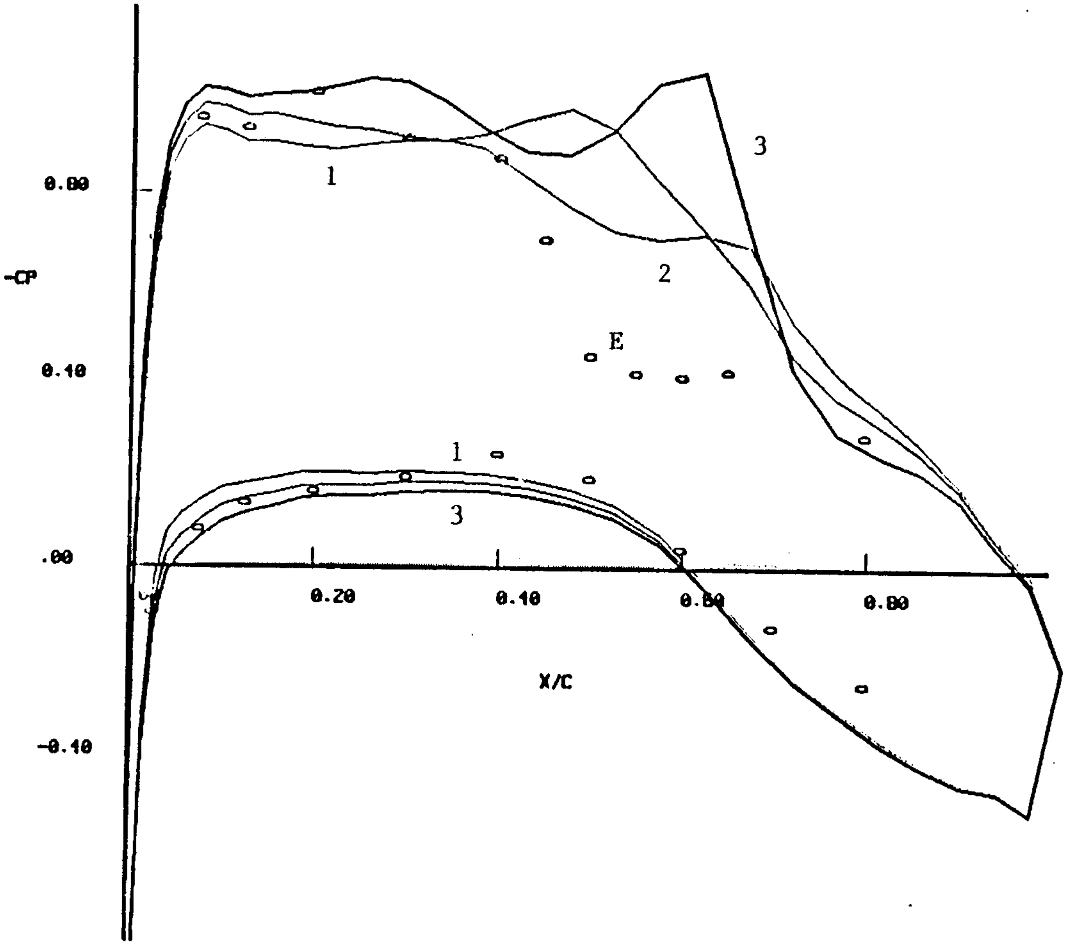

Figure 8d. E, 64%; (1), (2), and (3), 65%.

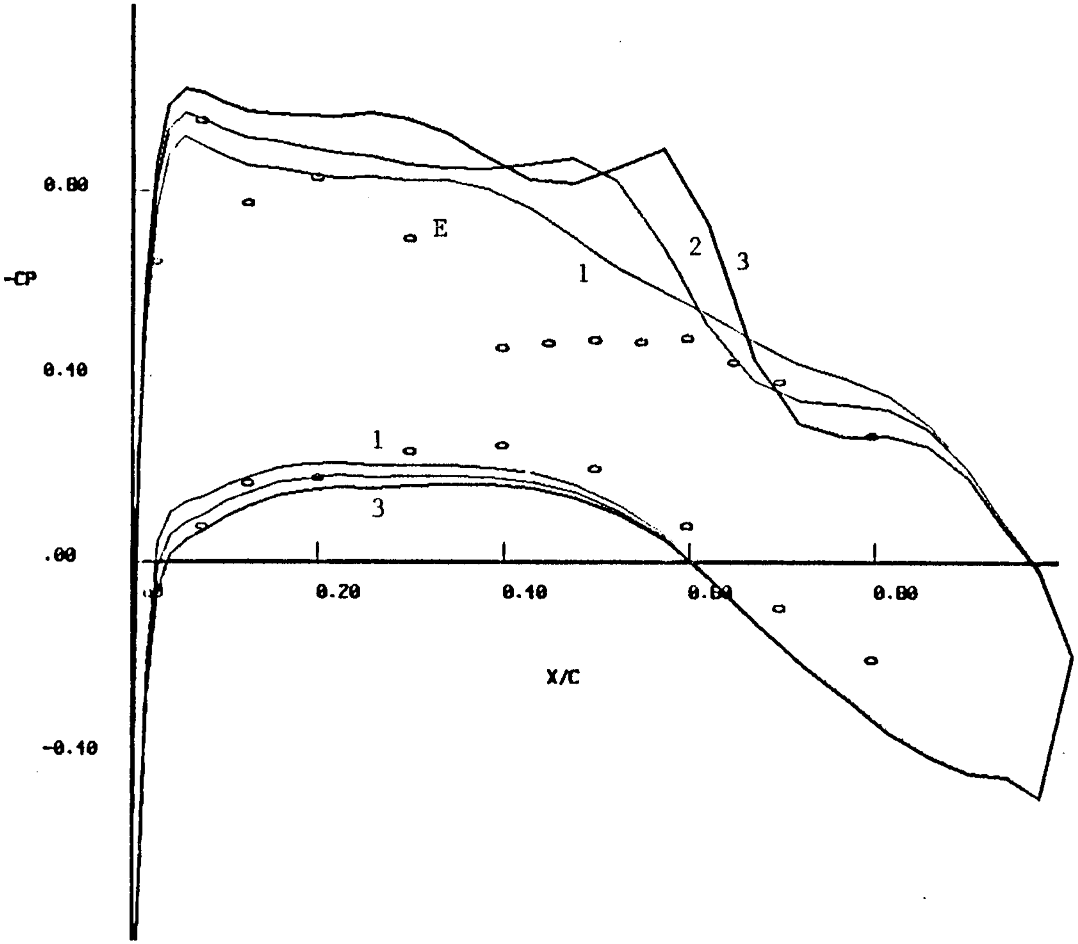

Figure 8e. E, 84%; (1), (2), and (3), 85%.

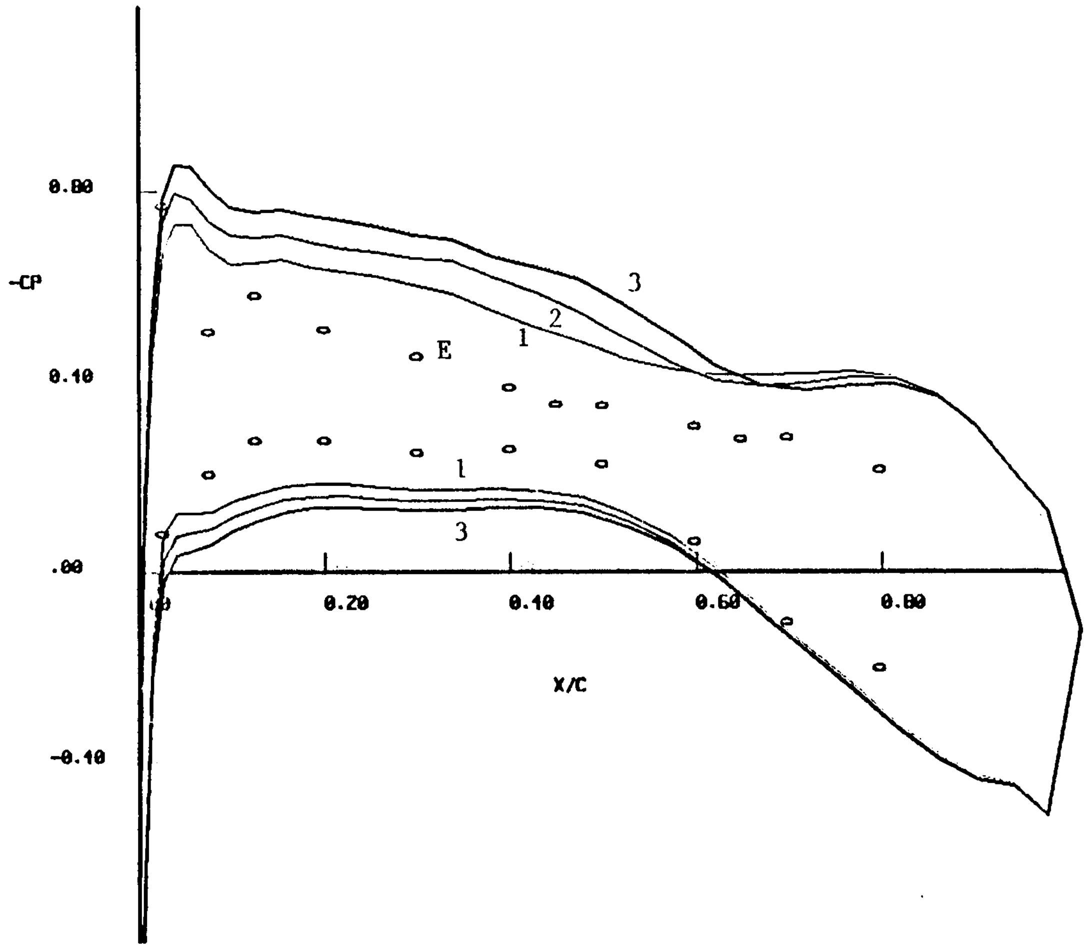

Figure 8f. E, 96.1%; (1), (2), and (3), 95%.

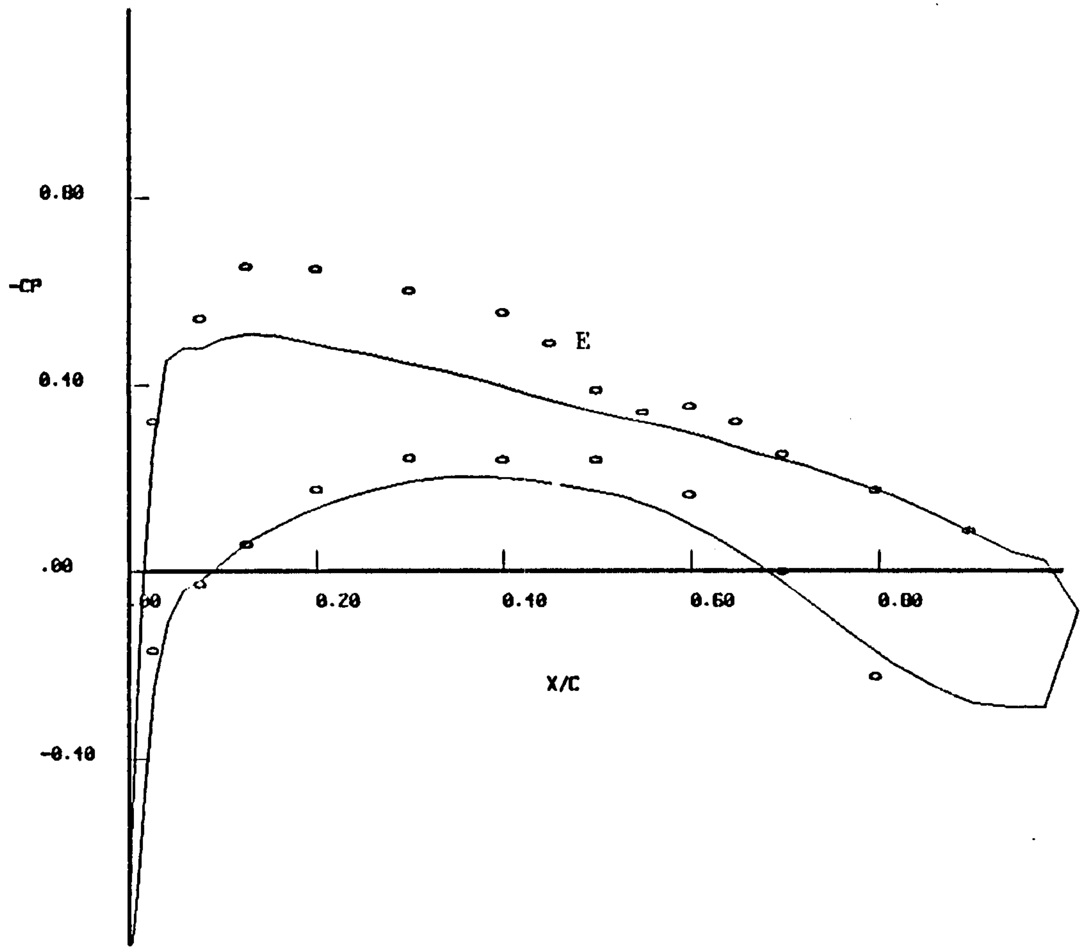

Figure 9a. Section pressure distribution, M∞=0.7, α=2°, Re=5.3*10^6;
E = experiment, 13.1% span; , viscous, 15% span.

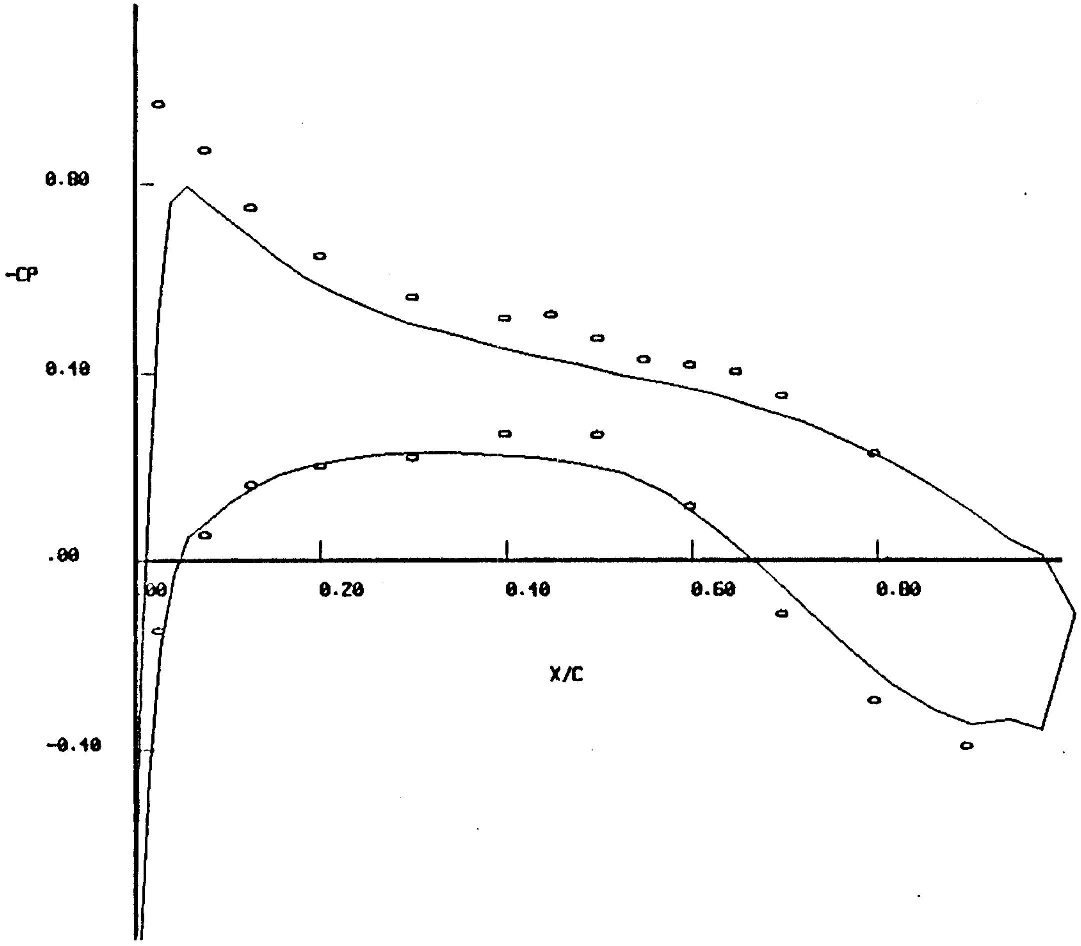

Figure 9b. E, 29.2%; calculation, 25%.

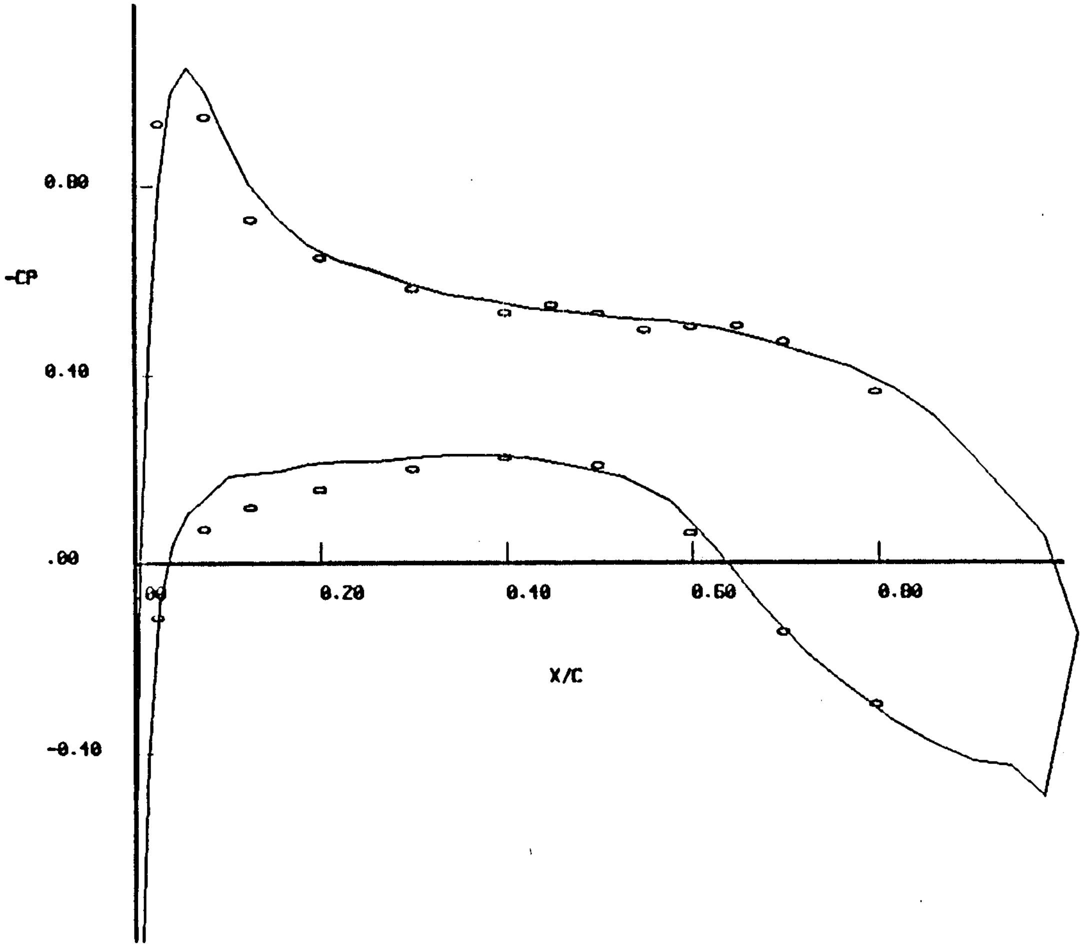

Figure 9c. E, 43.2%; calculation, 45%.

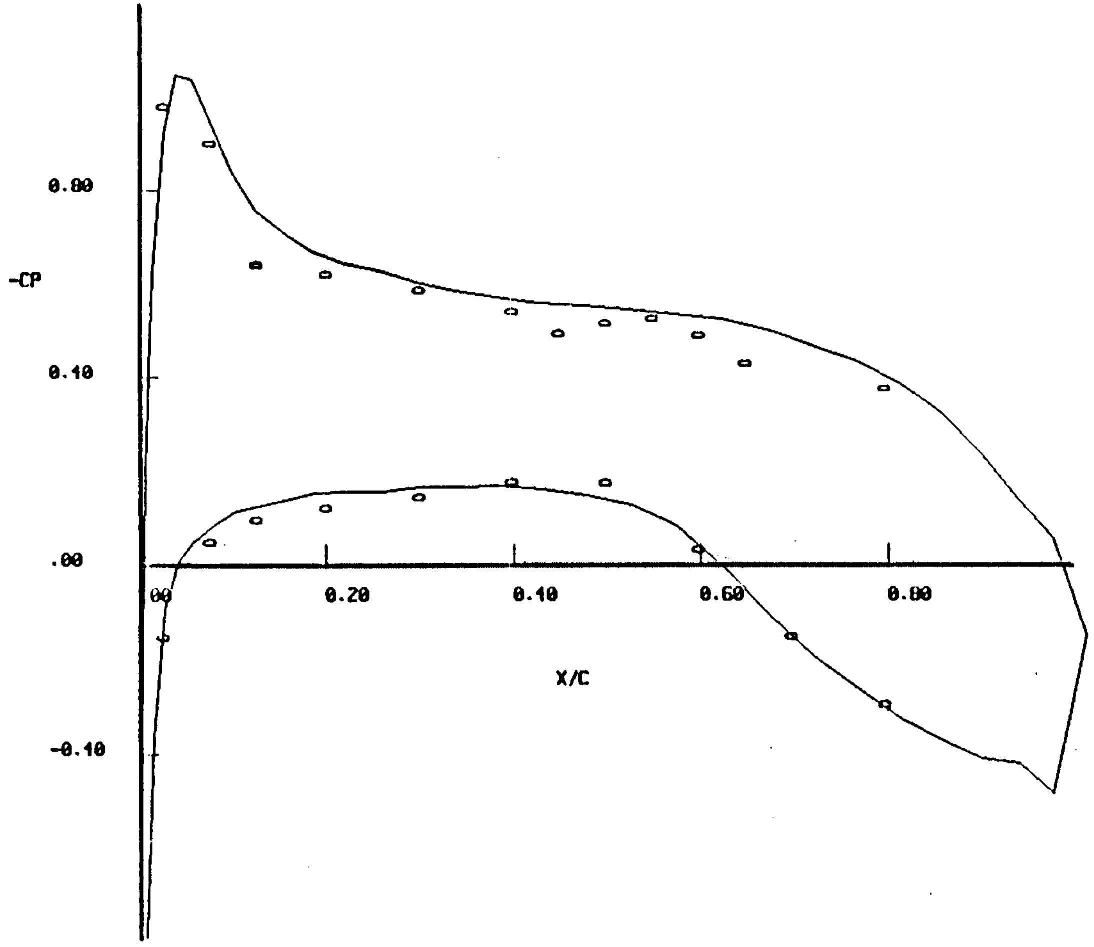

Figure 9d. E, 64%, calculation, 65%.

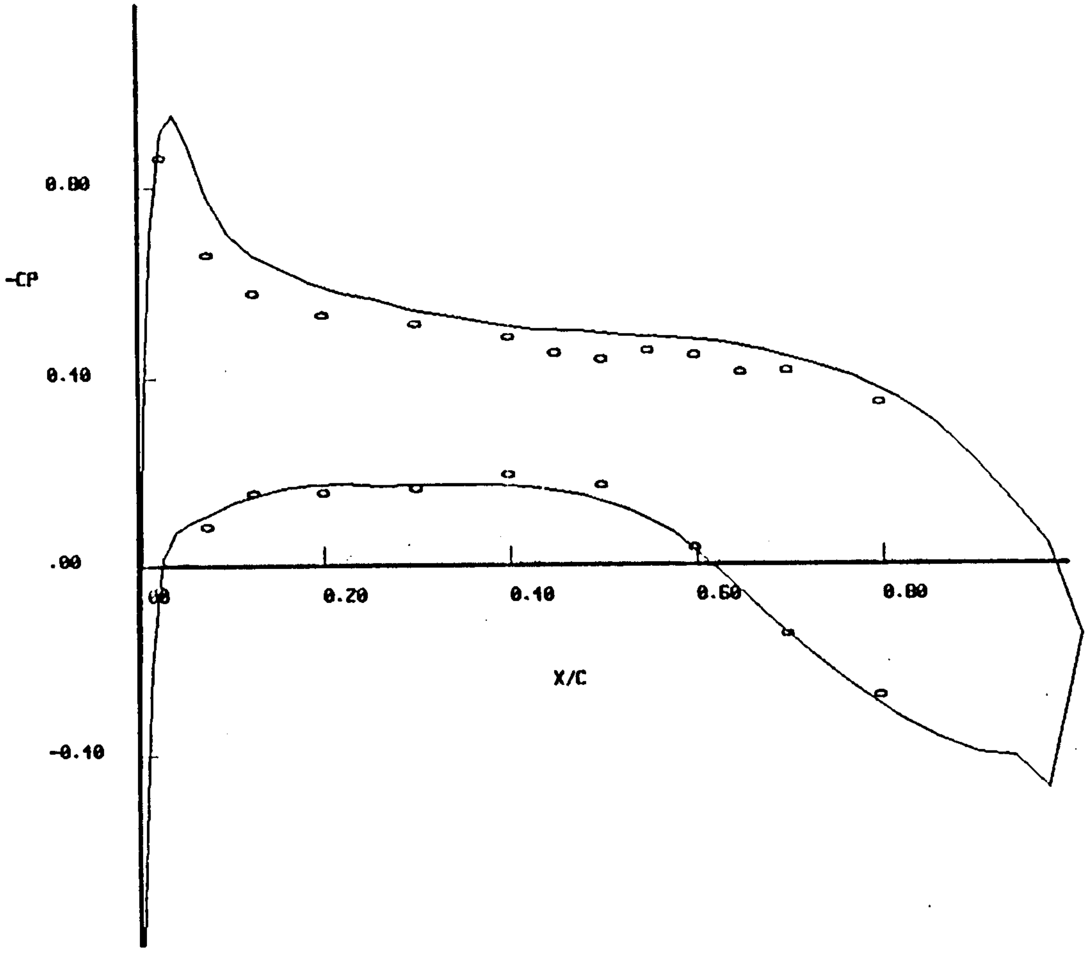

Figure 9e. E, 84%; calculation, 84%.

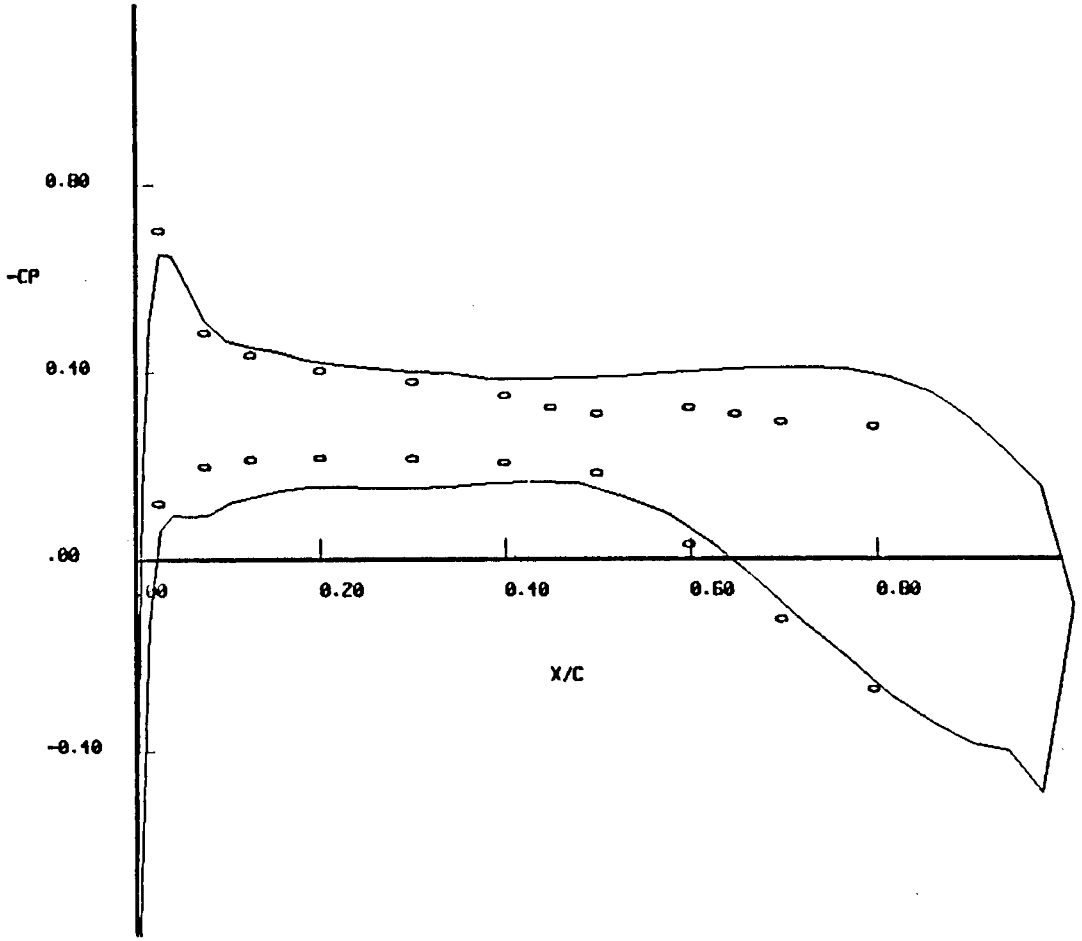

Figure 9f. E, 96.1%; calculation, 95%.