NASA Technical Memorandum 85899

A Compressible Navier-Stokes Code for Turbulent Flow Modeling

Thomas J. Coakley

February 1984

National Aeronautics and
Space Administration

NASA Technical Memorandum 85899

A Compressible Navier-Stokes Code for Turbulent Flow Modeling

Thomas J. Coakley, Ames Research Center, Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

A COMPRESSIBLE NAVIER-STOKES CODE FOR TURBULENT-FLOW MODELING

Thomas J. Coakley
NASA Ames Research Center, Moffett Field, California

ABSTRACT

An implicit, finite volume code for solving two-dimensional, compressible turbulent flows is described. Second-order upwind differencing of the inviscid terms of the equations is used to enhance stability and accuracy. A diagonal form of the implicit algorithm is used to improve efficiency. Several zero- and two-equation turbulence models are incorporated to study their impact on overall flow-modeling accuracy. Applications to several external and internal flows are discussed.

1. INTRODUCTION

A computer code for solving the time-dependent, Reynolds-averaged, compressible Navier-Stokes equations is described. The code solves the two-dimensional (2-D) planar and axisymmetric equations in arbitrary, curvilinear coordinates utilizing the finite volume approach. The numerical algorithm (refs. 1 and 2) is an approximately factored ADI method based on the work of Beam and Warming (ref. 3) and Briley and McDonald (ref. 4). It utilizes second-order upwind differencing techniques for the inviscid terms of the equations and a diagonal form of the implicit operators. These procedures are designed to enhance the overall accuracy, stability, and efficiency of the algorithm. Rapid convergence to steady-state solutions is achieved through the use of spatially varying time steps. A variety of inviscid and viscous boundary conditions are incorporated. These include subsonic and supersonic inflow and outflow conditions, which make use of the method of characteristics, and inviscid and viscous solid-wall boundary conditions.

Various turbulence models are integrated into the code to compare, evaluate, and ultimately improve model performance (ref. 5). The models include a zero-equation (Cebeci-Smith) model and three two-equation models. The models use the procedure of integration to the wall where no-slip conditions at a solid surface are used. Although in an early stage of development, the option of using wall functions (or law-of-the-wall boundary conditions) in place of no-slip conditions is also included.

In addition to the above features, the code uses a self-contained, algebraic, grid-generation routine, a special data-loading system designed to facilitate implicit

integration across interior mesh boundaries, a restart capability, and several types of input, output, and plotting routines.

In the following discussion, we open with a description of the theoretical basis of the numerical methods and turbulence models. We then discuss several features of the code which includes a summary of subroutines. Finally, sample results of inviscid and viscous calculations are presented.

2. THEORETICAL BACKGROUND

2.1 Governing Equations

The basic equations governing the flow of a viscous compressible fluid are the Navier-Stokes equations. These equations, expressed in terms of a stationary two-dimensional, curvilinear coordinate system and utilizing vector-dyadic notation, are written in strong conservation form as follows:

$$\left.\begin{aligned} &\upsilon \partial_t U + \partial_\alpha \vec{S}_\alpha \cdot \mathscr{F} = 0 \\ &U = \begin{bmatrix} \rho \\ \rho\vec{u} \\ \rho E \end{bmatrix}, \quad \mathscr{F}' = \begin{bmatrix} \rho\vec{u} \\ \rho\vec{u}\vec{u} + \vec{\vec{I}}p \\ (\rho E + p)\vec{u} \end{bmatrix}, \quad \mathscr{F}'' = \begin{bmatrix} 0 \\ \vec{\vec{\tau}} \\ \vec{\vec{\tau}} \cdot \vec{u} + \vec{q} \end{bmatrix} \\ &\mathscr{F} = \mathscr{F}' + \mathscr{F}'', \quad \alpha = \xi,\eta \end{aligned}\right\} \tag{1}$$

In these equations, ξ and η are the computational coordinates, t is the time, $\partial_t = \partial/\partial t$, $\partial_\alpha = \partial/\partial\alpha$ are partial derivatives, and the summation convention is used on the repeated subscript $\alpha = \xi,\eta$. The computational coordinates are related to Cartesian coordinates through the coordinate transformation

$$x = x(\xi,\eta), \quad y = y(\xi,\eta) \tag{2}$$

The geometric variables in equation (1) are the differential volume element υ (actually an area in 2-D) and the surface element vectors $\vec{S}_\alpha$ (actually line elements in 2-D), which are defined below.

$$\left.\begin{aligned} \upsilon &= x_\xi y_\eta - x_\eta y_\xi \\ \vec{S}_\xi &= \vec{i}S_{\xi x} + \vec{j}S_{\xi y} = \vec{i}y_\eta - \vec{j}x_\eta \\ \vec{S}_\eta &= \vec{i}S_{\eta x} + \vec{j}S_{\eta y} = -\vec{i}y_\xi + \vec{j}x_\xi \end{aligned}\right\} \tag{3}$$

where $\vec{i}$ and $\vec{j}$ are unit Cartesian base vectors and $x_\xi = \partial_\xi x$, $x_\eta = \partial_\eta x$, etc. (see fig. 1). An important property of the surface vectors is the metric identity

$$\partial_\alpha \vec{S}_\alpha = \partial_\xi \vec{S}_\xi + \partial_\eta \vec{S}_\eta = 0 \tag{4}$$

which permits $\vec{S}_\alpha$ to be taken either inside or outside the derivatives in operations involving the gradient operator $\vec{\nabla}$, i.e.,

$$\vec{\nabla} \circ f = \frac{1}{\upsilon} \vec{S}_\alpha \circ \partial_\alpha f = \frac{1}{\upsilon} \partial_\alpha \vec{S}_\alpha \circ f \tag{5}$$

where the symbol $\circ$ represents any vector product operator.

The physical variables appearing in equation (1) are the density ρ, the fluid velocity vector $\vec{u} = \vec{i}u + \vec{j}v$, the total specific energy $E = e + \vec{u} \cdot \vec{u}/2$, the specific internal energy e, and the pressure p. The variables U and $\mathscr{F}$ are the fluid-state vector and flux tensor (or dyadic), respectively. The flux tensor is split into inviscid and viscous components $\mathscr{F}'$ and $\mathscr{F}''$. The tensors $\vec{\vec{I}}$, $\vec{\vec{\tau}}$, and $\vec{q}$ appearing in $\mathscr{F}''$ are, respectively, the unit tensor, the viscous stress tensor, and the heat flux vector, which may be written (using Stokes hypothesis) as

$$\left.\begin{aligned} \vec{\vec{I}} &= \vec{i}\vec{i} + \vec{j}\vec{j} \\ \vec{\vec{\tau}} &= -\mu_v\left(\vec{\nabla}\vec{u} + \vec{u}\vec{\nabla} - \frac{2}{3}\vec{\vec{I}}\vec{\nabla} \cdot \vec{u}\right) \\ &= -\frac{\mu_v}{\upsilon}\left[\vec{S}_\beta(\partial_\beta \vec{u}) + (\partial_\beta \vec{u})\vec{S}_\beta - \frac{2}{3}\vec{\vec{I}}\vec{S}_\beta \cdot \partial_\beta \vec{u}\right] \\ \vec{q} &= -\mu_e \vec{\nabla} e = -\frac{\mu_e}{\upsilon}\vec{S}_\beta \partial_\beta e \end{aligned}\right\} \tag{6}$$

where $\vec{u}\vec{\nabla}$ is the conjugate of $\vec{\nabla}\vec{u}$. For laminar flows, $\mu_v = \mu$ and $\mu_e = \gamma\mu/Pr$ are the molecular viscosity and conductivity coefficients, where $Pr = C_p\mu/K$ is the Prandtl number. It is assumed that the fluid is a perfect gas governed by the equation of state

$$p = (\gamma - 1)\rho e = \rho c^2/\gamma \ , \quad e = C_v T \ , \quad \gamma = C_p/C_v \tag{7}$$

where γ is the ratio of specific heats, c is the sound speed, and T is temperature.

For turbulent flows, the basic equations are time- or ensemble-averaged using mass-weighted averaging. The resulting equations are called the Reynolds-averaged Navier-Stokes equations and can be cast in a form essentially identical with equation (1). The stress tensor and heat-flux vector in this case contain turbulent contributions involving two- and three-point correlations, in addition to molecular

contributions. The turbulent correlations are unknown and must be approximated by a mathematical turbulence model. The turbulence models used in the code are called eddy-viscosity models and are represented in equations (6) by expressing μ_v and μ_e in terms of an eddy-viscosity function μ_T, i.e.,

$$\mu_v = \mu + \mu_T \ , \quad \mu_e = \gamma\left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right) \tag{8}$$

where Pr_T is a turbulent Prandtl number.

The dot product of the area vector $\vec{S}_\alpha$ and the flux tensor $\mathscr{F}$ appearing in equation (1) is the flux vector which can be expressed in terms of inviscid and viscous contributions as follows:

$$\left.\begin{aligned} F_\alpha &= \vec{S}_\alpha \cdot \mathscr{F} = F'_\alpha + F''_\alpha \\ Q_\alpha &= \vec{u} \cdot \vec{S}_\alpha = uS_{\alpha x} + vS_{\alpha y} \\ F'_\alpha &= \begin{bmatrix} \rho Q_\alpha \\ \rho\vec{u}Q_\alpha + p\vec{S}_\alpha \\ (\rho E + p)Q_\alpha \end{bmatrix}, \quad F''_\alpha = \begin{bmatrix} 0 \\ \vec{S}_\alpha \cdot \vec{\vec{\tau}} \\ \vec{S}_\alpha \cdot (\vec{\vec{\tau}} \cdot \vec{u} + \vec{q}) \end{bmatrix} \end{aligned}\right\} \tag{9}$$

The viscous flux vector F''_α may be represented in matrix form by the sum

$$F''_\alpha = -B_{\alpha\beta}\partial_\beta U = -\sum_{\beta=\xi,\eta} B_{\alpha\beta}\partial_\beta U \tag{10}$$

The above equations have been written explicitly for 2-D plane flow. For axisymmetric flow, the corresponding equations can be obtained from the 2-D forms by the replacements

$$\left.\begin{aligned} \vec{S}_\alpha &\to r\vec{S}_\alpha \ , \quad \upsilon \to r\upsilon \\ \nabla \cdot \vec{u} &\to \nabla \cdot \vec{u} + \frac{v}{r} = \frac{1}{\upsilon}\vec{S}_\alpha \cdot \partial_\alpha \vec{u} + \frac{v}{r} \\ H &= \vec{j}\,\frac{\upsilon}{r}\left[p - \mu_v\left(2\,\frac{v}{r} - \frac{2}{3}\vec{\nabla} \cdot \vec{u}\right)\right] \end{aligned}\right\} \tag{11}$$

where H is a source function to be added to the right-hand side (RHS) of the v momentum equation, $\vec{S}_\alpha \cdot \partial_\alpha \vec{u} = S_{\alpha x}\partial_\alpha u + S_{\alpha y}\partial_\alpha v$ and $r = y$ is the radius or distance from the axis of symmetry.

2.2 Finite Volume Method

The finite-volume technique is used to express the equations of motion in discrete form. In this method, a lattice of mesh points is generated by specifying the Cartesian coordinates x and y as discrete functions of the computational coordinates ξ and η. In the discrete approximation, ξ and η are given either integer or half integer values, and are used interchangeably with the subscripts i and j (see fig. 1). The discrete representation of the surface vectors and volume element is given by the following formulas:

$$\left.\begin{aligned}
(\vec{S}_\xi)_{i-(1/2),j} &= \vec{i}(y_{i,j+1} - y_{i,j}) - \vec{j}(x_{i,j+1} - x_{i,j}) \\
(\vec{S}_\eta)_{i,j-(1/2)} &= -i(y_{i+1,j} - y_{i,j}) + \vec{j}(x_{i+1,j} - x_{i,j}) \\
\upsilon_{i,j} &= (\bar{x}_{i+1} - \bar{x}_i)(\bar{y}_{j+1} - \bar{y}_j) - (\bar{x}_{j+1} - \bar{x}_j)(\bar{y}_{i+1} - \bar{y}_i) \\
\bar{x}_i &= \frac{1}{2}(x_{i,j} + x_{i,j+1}) \ , \quad \bar{x}_j = \frac{1}{2}(x_{i,j} + x_{i+1,j}) \ , \text{ etc.}
\end{aligned}\right\} \tag{12}$$

In the axisymmetric case, the radial variable $r = y$ in equations (11) is represented as an average of the four associated points in the expression for υ, and an average of the two associated points in the expressions for $\vec{S}_\alpha$. For the planar case, the surface vectors satisfy the numerical analog of the metric identity, equation (4). A related identity is satisfied in the axisymmetric case. This feature leads to the important physical property of free-stream maintenance in which the difference approximation of $\partial_\alpha \vec{S}_\alpha \cdot \mathscr{F}$ vanishes when $p, \rho, \vec{u}$ are constant regardless of the difference approximation for $\mathscr{F}$.

In the finite-volume method the state vector is assumed to be an integrated average over the cell volume (area). It is located at the centroid of a cell and is given integer subscripts, i.e., $U_{i,j}$. This notation is frequently contracted, e.g., $U_{i+1,j} = U_{i+1}$. The surface vectors are located at cell boundaries and are given mixed integer/half integer subscripts.

2.3 Diagonal Transformation

The Jacobian matrix of the inviscid flux vector F'_α is defined by $A_\alpha = \partial F'_\alpha / \partial U$. This notation will be shortened here by dropping the subscript α. Thus, we write

$$A = \frac{\partial F'}{\partial U} = \frac{\partial}{\partial U}\begin{bmatrix} \rho Q \\ \rho\vec{u}Q + p\vec{S} \\ (\rho E + p)Q \end{bmatrix} \tag{13}$$

$$Q = \vec{u} \cdot \vec{S} = uS_x + vS_y$$

In the following development, considerable use will be made of the diagonal representation of A. This is written

$$\left.\begin{aligned} A &= SR^{-1}\Lambda R \\ \Lambda &= \mathrm{diag}(u_n, u_n, u_n + c, u_n - c) \\ S &= |\vec{S}| \ , \quad \vec{n} = \vec{S}/S \ , \quad u_n = \vec{u} \cdot \vec{n} \\ R &= CNP \ , \quad R^{-1} = P^{-1}N^{-1}C^{-1} \end{aligned}\right\} \tag{14}$$

The matrix R is a similarity matrix which diagonalizes A, $S\Lambda$ is the diagonal matrix of eigenvalues of A, and u_n is the component of velocity normal to the surfaces defined by $\vec{S}$. The matrix R is factored into three submatrices which are written

$$C = \begin{bmatrix} -c^2 & & & 1 \\ & 1 & & \\ & & c & 1 \\ & & -c & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & & & & \\ & m_x & m_y & & \\ & n_x & n_y & & \\ & & & 1 & \end{bmatrix}, \quad P = \begin{bmatrix} 1 & & & \\ -u & 1 & & \\ -v & & 1 & \\ \frac{\chi}{2}\vec{u}^2 & -\chi u & -\chi v & \chi \end{bmatrix}$$

$$C^{-1} = \begin{bmatrix} \frac{-1}{c^2} & & \frac{1}{2c^2} & \frac{1}{2c^2} \\ & 1 & & \\ & & \frac{1}{2c} & \frac{-1}{2c} \\ & & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad N^{-1} = \begin{bmatrix} 1 & & & & \\ & m_x & n_x & & \\ & m_y & n_y & & \\ & & & 1 & \end{bmatrix}, \quad P^{-1} = \begin{bmatrix} 1 & & & \\ u & 1 & & \\ v & & 1 & \\ \frac{1}{2}\vec{u}^2 & u & v & \frac{1}{\chi} \end{bmatrix} \tag{15}$$

where $\chi = \gamma - 1$ and the empty spaces in the matrices represent zeros. In the orthogonal matrix N, the elements n_x and n_y are the Cartesian components of $\vec{n}$, i.e., $\vec{n} = \vec{i}n_x + \vec{j}n_y$, and $\vec{m} = \vec{i}m_x + \vec{j}m_y$ is a unit vector orthogonal to $\vec{n}$ with $m_x = -n_y$, $m_y = n_x$. The component of velocity tangent to the surface $\vec{S}$ is given by the dot product $u_m = \vec{u} \cdot \vec{m}$. Under the transformation R, the vector of conserved

variable differentials is transformed into a vector of differential Reimann variables, i.e., $dW = RdU$ or

$$dW = \begin{bmatrix} dp - c^2 d\rho \\ \rho du_m \\ dp + \rho c du_n \\ dp - \rho c du_n \end{bmatrix} = R \begin{bmatrix} d\rho \\ d\rho u \\ d\rho v \\ d\rho E \end{bmatrix} \tag{16}$$

2.4 Time Differencing

The basic algorithm used to solve equations (1) is an approximately factored ADI method based on the work of Beam and Warming (ref. 3). The implicit time-differenced equations are obtained by Taylor series expansions in t utilizing the relations

$$\left.\begin{aligned} \partial_t U &= [U(t + \Delta t) - U(t)]/\Delta t = \Delta U/\Delta t \\ U(t + \theta\Delta t) &\sim U(t) + \theta\Delta U \\ F'_\alpha(t + \theta\Delta t) &\sim F'_\alpha(t) + \theta A_\alpha(t)\Delta U \\ F''_\alpha(t + \theta\Delta t) &\sim F''_\alpha(t) + \theta B_{\alpha\beta}(t)\partial_\beta \Delta U \end{aligned}\right\} \tag{17}$$

In these expressions ΔU is the incremental or delta variable, Δt is the time step, and θ is a constant (between 0.5 and 1.0). An approximate form of the expansion of F''_α is used in which the viscous matrices $B_{\alpha\beta}$ are frozen in time. Substitution of the above expansions into equations (1) gives the unfactored form of the implicit algorithm

$$\left.\begin{aligned} &[I + h'\partial_\alpha(A_\alpha - B_{\alpha\beta}\partial_\beta)]\Delta U = -h\partial_\alpha F_\alpha \\ &h = \Delta t/\upsilon \ , \quad h' = \theta h \ , \quad \alpha,\beta = \xi,\eta \end{aligned}\right\} \tag{18}$$

where the summation is taken on both α and β, and I is the identity matrix. The value of θ is normally taken to be $\theta = 1$ corresponding to first-order, backward-time differencing.

The above equation is simplified by approximately factoring the implicit, or left-hand side (LHS) operators into ξ and η operators. A further approximation consists in discarding the off-diagonal viscous matrices, i.e., $B_{\alpha\beta} = 0$, $\beta \neq \alpha$. By redefining $B_\alpha = B_{\alpha\beta}$, $\beta = \alpha$, a chain of implicit factored equations in ADI form is obtained:

$$\left.\begin{aligned} [I + h'\partial_\xi(A_\xi - B_\xi\partial_\xi)]\Delta U^* &= -h(\partial_\xi F_\xi + \partial_\eta F_\eta) \\ [I + h'\partial_\eta(A_\eta - B_\eta\partial_\eta)]\Delta U &= \Delta U^* \\ U(t + \Delta t) &= U(t) + \Delta U \end{aligned}\right\} \tag{19}$$

By suitably defining the partial derivatives in terms of spatial difference operators, the above equations can be expressed as a system of coupled algebraic equations which may be solved by block-tridiagonal matrix inversions. This is a costly numerical process. A more efficient procedure, which is used in the code, is to simplify the left-hand sides of equations (19) utilizing the diagonal transformation so that they become uncoupled and can be solved by scalar tridiagonal inversions. This process results in the diagonal algorithm (ref. 6).

To develop the diagonal algorithm we write a typical member of equations (19) by dropping the subscripts on the matrices A and B, e.g.,

$$[I + h'\partial_\xi(A - B\partial_\xi)]\Delta U = \Delta U_o \tag{20}$$

The viscous matrix B is approximated by the identity matrix multiplied by the largest eigenvalue of B, i.e.,

$$B = \nu I , \quad \nu = \frac{S}{\rho U} \max\left(\frac{4}{3}\mu_v, \mu_e\right) \tag{21}$$

By introducing the diagonal decomposition of A, i.e., $A = SR^{-1}\Lambda R$, and making the further approximation of taking A, R, S, and ν to be locally constant (so that they can be moved outside the partial derivatives) we obtain a diagonal, uncoupled system of equations or the diagonal algorithm

$$\left.\begin{aligned} [I + h''(\Lambda\partial_\xi - \nu I\partial_\xi^2)]R\Delta U &= R\Delta U_o \\ h'' &= \theta\Delta t S/U \end{aligned}\right\} \tag{22}$$

After making a suitable specification of the spatial derivatives in terms of finite differences, the above equations are solved recursively by scalar tridiagonal inversions.

The approximations involved in the reduction to diagonal form diminish to some extent the time accuracy of the algorithm (ref. 6). For example, the nonconservative nature of the implicit operator causes some error in the propagation speeds of strong shock waves. However, the approximations do not affect the accuracy of a steady-state solution, which is governed by the spatial flux differencing of the right-hand side of equations (19) and which is normally taken to be second order.

2.5 Spatial Differencing

To complete the specification of the numerical algorithm, the spatial derivatives in equations (19) and (22) must be replaced by finite differences. This is done by using central differences for the viscous terms and upwind differencing for the inviscid terms.

Consider first the RHS explicit flux differencing of equations (19). To maintain strict conservation regardless of the type of differencing used to approximate the fluxes, the difference approximation for $\partial_\xi F_\xi$ is written as

$$\partial_\xi F_\xi = (F_\xi)_{i+(1/2),j} - (F_\xi)_{i-(1/2),j} \tag{23}$$

The $(F_\xi)_{i+(1/2),j}$ are fluxes evaluated at the cell boundaries that must be expressed in terms of the state vector evaluated at the cell centers, e.g., $U_{i-1,j}$, $U_{i,j}$, $U_{i+1,j}$, etc. To simplify the notation we will drop the subscripts ξ and j from F_ξ in equation (23), i.e.,

$$\left.\begin{aligned} \partial_\xi F_\xi = \partial_\xi F &= F_{i+(1/2)} - F_{i-(1/2)} \\ F_{i+(1/2)} &= \vec{S}_{i+(1/2)} \cdot \mathscr{F}_{i+(1/2)} = F'_{i+(1/2)} + F''_{i+(1/2)} \end{aligned}\right\} \tag{24}$$

(a) Inviscid flux differencing- The inviscid flux vector $F'_{i+(1/2)}$ is represented in terms of a central average involving $\mathscr{F}'_i = \mathscr{F}'(U_i)$ and $\mathscr{F}'_{i+1} = \mathscr{F}(U_{i+1})$ and a dissipation function $D_{i+(1/2)}$:

$$F'_{i+(1/2)} = \frac{1}{2}\left[\vec{S}_{i+(1/2)} \cdot (\mathscr{F}'_i + \mathscr{F}'_{i+1}) - |\vec{S}_{i+(1/2)}| D_{i+(1/2)}\right] \tag{25}$$

If $D_{i+(1/2)} = 0$, the differencing for $\partial_\xi F$ reduces to pure second-order central differencing.

The dissipation function is expressed in terms of special operators $\mathscr{M}$, $\mathscr{N}$, and δ that are defined as follows:

$$\left.\begin{aligned} D_{i+(1/2)} &= [R^{-1}(\Lambda \mathscr{M} U + |\Lambda| \mathscr{N} U)]_{i+(1/2)} \\ (\mathscr{M}U)_{i+(1/2)} &= \alpha(\delta W_{i+(3/2)} - \delta W_{i-(1/2)}) \\ (\mathscr{N}U)_{i+(1/2)} &= \beta \delta W_{i+(1/2)} - \gamma(\delta W_{i+(3/2)} + \delta W_{i-(1/2)}) \\ (\delta W)_{i+(1/2)} &= \delta W_{i+(1/2)} = R_{i+(1/2)}(U_{i+1} - U_i) \end{aligned}\right\} \tag{26}$$

The constants α, β, γ control the accuracy of the spatial differencing for $\partial_\xi F'$. Several cases are listed below.

Spatial flux differencing options

$$\left.\begin{array}{ll} \text{First-order upwind} & \beta = 1\ , \quad \alpha = \gamma = 0 \\ \text{Second-order upwind} & \beta = 1\ , \quad \alpha = \gamma = 1/2 \\ \text{Third-order upwind} & \beta = 1/3\ , \quad \alpha = \gamma = 1/6 \\ \text{Second-order central} & \beta = \text{free}\ , \quad \alpha = 0\ , \quad \gamma = \beta/2 \end{array}\right\} \tag{27}$$

Most of the cases reported in this paper were done with the second-order upwind option. Assuming A is constant, spatial accuracy may be checked by Taylor series expansion of equation (24) using equations (25), (26), and (27).

To completely specify the inviscid flux differencing, the method of averaging, i.e., computation of $R_{i+(1/2)}$ in terms of U_i and U_{i+1}, must be given. The type of averaging used here is the nonlinear averaging of Roe (refs. 7 and 8). The specific formulas used in computing $R_{i+(1/2)}$ and $\Lambda_{i+(1/2)}$ are given below.

$$\left.\begin{aligned} a &= (\rho_{i+1}/\rho_i)^{1/2}\ , \quad H = E + p/\rho \\ u_{i+(1/2)} &= (au_{i+1} + u_i)/(1 + a) \\ v_{i+(1/2)} &= (av_{i+1} + v_i)/(1 + a) \\ c^2_{i+(1/2)} &= (\gamma - 1)[(aH_{i+1} + H_i)/(1 + a) - (u^2_{i+(1/2)} + v^2_{i+(1/2)})/2] \end{aligned}\right\} \tag{28}$$

These formulas are used with equations (15) where it is noted that the unit-vector components n_x, n_y, m_x, m_y, being obtained from $\vec{S}_{i+(1/2)}$, require no averaging.

(b) Viscous flux differencing- As stated above, the viscous flux $F'' = \vec{S} \cdot \mathscr{F}''$ is centrally differenced using equation (24). Explicit formulas for evaluating the heat-flux vector $\vec{q}$ at the cell boundary $i + (1/2)$ are described below.

$$\vec{q}_{i+(1/2)} = -\left[\frac{\mu_v}{v}\left(\vec{S}_\xi \partial_\xi \dot{e} + \bar{S}_\eta \partial_\eta e\right)\right]_{i+(1/2)} \tag{29}$$

The viscosity and volume element in this expression are evaluated by central averaging, i.e.,

$$[v]_{i+(1/2)} = \frac{1}{2}(v_i + v_{i+1}) \tag{30}$$

The surface vector $(\vec{S}_\xi)_{i+(1/2)}$ is given directly by the finite-volume construction. The normal derivative term $\partial_\xi e$ is evaluated by

$$(\partial_\xi e)_{i+(1/2)} = e_{i+1} - e_i \tag{31}$$

The two remaining terms are cross-derivative expressions in which a special averaging procedure is used to center known quantities at the cell face. These cross-derivative terms are given below.

$$\left.\begin{aligned}(\vec{\vec{S}}_\eta)_{i+(1/2)} &= \frac{1}{2}\left[(\vec{\vec{S}}_\eta)_{j-(1/2)} + (\vec{\vec{S}}_\eta)_{j+(1/2)}\right]_{i+(1/2)} \\ (\partial_\eta e)_{i+(1/2)} &= \frac{1}{2}\left[e_{j+1} - e_{j-1}\right]_{i+(1/2)}\end{aligned}\right\} \tag{32}$$

The additional elements of $\mathscr{F}''$, i.e., $\vec{\vec{\tau}}$ and $\vec{\vec{\tau}} \cdot \vec{u}$ are determined by a similar construction.

An option to use a simplified form of the Navier-Stokes equations called the thin-layer Navier-Stokes equations is included in the code. Under this approximation the derivatives tangential to a thin shear layer (say ∂_ξ) are discarded resulting in a simplified form of the equations. Use of this option gives a reduction of about 15% of the computing time required for the full Navier-Stokes equations.

(c) Implicit differencing- The differencing of the LHS or implicit terms of equations (22) follows a pattern similar to that used for the RHS or explicit terms. That is, upwind differencing is used for inviscid first derivatives and central differencing is used for viscous second derivatives. These differencing expressions are given as follows:

$$\left.\begin{aligned}\lambda\partial_\xi f &= \left(\frac{\lambda_i + |\lambda_i|}{2}\right)(f_i - f_{i-1}) + \left(\frac{\lambda_i - |\lambda_i|}{2}\right)(f_{i+1} - f_i) \\ &= \frac{1}{2}\left[\lambda_i(f_{i+1} - f_{i-1}) - |\lambda_i|(f_{i+1} + f_{i-1} - 2f_i)\right] \\ \nu\partial_\xi^2 f &= \nu_i(f_{i+1} + f_{i-1} - 2f_i)\end{aligned}\right\} \tag{33}$$

where λ is one of the eigenvalues $u_n, u_n, u_n + c, u_n - c$, and f is a corresponding element of the vector of differential Reimann variables, i.e., $\Delta W = R\Delta U$ (see equation (16)).

2.6 Boundary Conditions

Boundary conditions must be specified for the flux vector F_α and delta variable $\Delta W = R\Delta U$ at the physical boundaries of the mesh. Unfortunately, space does not permit a detailed exposition of boundary techniques here; thus, only a brief summary will be given. A more detailed description is given in reference 1.

We will discuss first boundary conditions for the inviscid Euler equations.

(a) Inviscid boundary conditions- The basic principle in specifying boundary conditions for the inviscid equations comes from the method of characteristics (MOC). This principle states that the number of conditions or variables that may be arbitrarily prescribed at a boundary is equal to the number of eigenvalues of A whose signs indicate signal propagation from the boundary to the interior domain. The remaining conditions or variables must be determined by some form of numerical extrapolation from the interior to the boundary. This can be done in many ways. The method used in the code uses a set of compatibility relations associated with each eigenvalue and derived from a locally one-dimensional approximation. These compatibility relations are listed below.

Eigenvalue	Compatibility relations
u_n	$\partial_\xi s' = \partial_\xi u_m = 0$
$u_n + c$	$\partial_\xi p + \rho c \partial_\xi u_n = 0$
$u_n - c$	$\partial_\xi p - \rho c \partial_\xi u_n = 0$

(34)

where $s' = c/p^{(\gamma-1)/2\gamma}$ is an entropy variable and u_n and u_m are the normal and tangential components of velocity at the boundary. The compatibility relations are or are not used at boundary depending on the signs of the eigenvalues. The procedure is summarized below for a left boundary.

Inviscid boundary conditions

	Flow state		Boundary condition	
1.	Inflow,	$u_n > 0$	Prescribe	s', u_m
2.	Outflow,	$u_n < 0$	Extrapolate	$\partial_\xi(s', u_m) = 0$
3.	Supersonic, inflow	$u_n > c$	Prescribe	p, u_n
4.	Supersonic, outflow	$u_n < -c$	Extrapolate	$\partial_\xi(p, u_n) = 0$
5.	Subsonic, flow	$\lvert u_n \rvert < c$	Prescribe	$f(p, u_n) = 0$
			Extrapolate	$\partial_\xi p - \rho c \partial_\xi u_n = 0$

(a) Prescribe normal velocity u_n

(b) Prescribe pressure p

(c) Prescribe total temperature $c^2 + \frac{\gamma - 1}{2}(u_n^2 + u_m^2)$

The extrapolation conditions for $\partial_\xi w$ are numerically approximated by either a first- or second-order form, i.e.,

$$\partial_\xi w = 2(w_I - w_B) \quad \text{or} \quad 3w_I - w_{I+1} - 2w_B \tag{35}$$

where the indices I, I + 1, and B are indicated in figure 2.

The above conditions result in the determination of a nonconservative state vector $V = (s', u_m, u_n, p)^T$ at the boundary. The flux vector F'_B, which is required at the boundary is constructed from V, i.e., $F'_B = F'_B(V)$.

Inviscid boundary conditions on the delta variables $\Delta W = R\Delta U$ are determined by a procedure similar to that described above. Details may be found in reference 1.

Boundary conditions are needed on the differences $\delta W_{i+(1/2)}$ of equations (26). The procedure presently used in the code for these terms is simple extrapolation, i.e., $\delta W_{I-(1/2)} = \delta W_B = \delta W_{I+(1/2)}$ (see fig. 2). This procedure may lead to some error, especially for inviscid flows. More rigorous conditions are described in reference 9 but these have not yet been incorporated into the code.

(b) Viscous boundary conditions- For flows in which viscous effects are important at boundaries, the inviscid conditions described above must be modified. The basic principle in these cases is that the variables associated with the normal-velocity component (i.e., the tangential velocity u_m and the entropy s, or temperature T) may be prescribed independently of the sign of u_n at the boundary (since the equations are no longer hyperbolic but parabolic). For viscous flows that are subsonic at the boundary, the pressure and normal velocity are determined by the same procedure used in the inviscid case.

Additional boundary conditions may also be imposed. In the code, the option of using periodic boundary conditions (associated with airfoil "O" grids) is incorporated. In addition, a centrifugal correction to the wall pressure (determined from the normal momentum equation) which is important for inviscid flows over highly curved surfaces, can also be used.

2.7 Turbulence Models

At the present state of development, there is no single model that holds a decisive advantage over all other turbulence models. For this reason several zero- and two-equation turbulence models are integrated into the code to investigate their impact on overall modeling accuracy and to facilitate further development. These models are described in detail in reference 5 and only a brief summary of them will be given here. The zero-equation model is a version of the Cebeci-Smith model (ref. 10) and is summarized below.

(a) Zero-equation model (Cebeci-Smith)-

$$\left.\begin{aligned} \mu_T &= \min(\mu_{TI},\mu_{To}) \ , \quad Pr_T = 0.9 \\ \mu_{TI} &= \rho \ell^2 |\vec{\nabla} \times \vec{u}| \ , \quad \mu_{To} = 0.0168 \rho u_e \delta^* \\ \ell &= 0.4yD \ , \quad D = 1 - \exp(-\sqrt{\rho_w \tau_w}\, y/26\mu_w) \end{aligned}\right\} \tag{36}$$

where y is the distance normal to the surface, u_e is the boundary-layer edge velocity, δ^* is the kinematic displacement thickness, and the subscript w in the expression for the damper, D, represents wall values. The above model is applicable to smooth-wall boundary layers, and a modified form must be used for trailing edge and wake flows (ref. 5).

(b) Two-equation models- Two-equation models use two additional PDEs for variables that are used to define the eddy-viscosity function, μ_T. An advantage of these models over zero-equation models is that they can be applied over a wide range of flow geometries without the need for a special treatment or tailoring of an algebraic length scale. The field equations for two-equation models can be expressed in the form

$$\left.\begin{aligned} \upsilon \partial_t \rho s + \partial_\alpha \vec{S}_\alpha \cdot (\rho s \vec{u} + \vec{q}_s) &= H_s \\ s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \ , \quad \vec{q}_s &= -\frac{\mu_s}{\upsilon} \vec{S}_\beta \partial_\beta s \\ \mu_T = C_\mu D \frac{k}{\omega} \ , \quad \mu_s &= \mu + \frac{\mu_T}{Pr_s} \end{aligned}\right\} \tag{37}$$

where C_μ and Pr_s are modeling constants, k is the turbulent kinetic energy, $\omega = \varepsilon/k$ is the specific dissipation rate of turbulence, and ε is the absolute dissipation rate. The vector H_s represents a pair of turbulence source functions and D is a damping function the details of which are given in reference 5 along with C_μ and Pr_s. With these models, a turbulent-pressure term, $(2/3)\rho k \vec{\vec{I}}$ is normally included with the viscous stress tensor $\vec{\vec{\tau}}$ of equation (6).

The three two-equation models incorporated into the code are identified below.

Two-equation models

1.	$k - \varepsilon$	Chien model (ref. 11):	$s_1 = k$,	$s_2 = \varepsilon$
2.	$k - \omega^2$	Wilcox-Rubesin (ref. 12):	$s_1 = k$,	$s_2 = \omega^2$
3.	$q - \omega$	Coakley model (ref. 5):	$s_1 = \sqrt{k}$,	$s_2 = \omega$

The upwind-flux differencing procedure described earlier for the Euler equations is extended in a straightforward manner to the convective flux of the model equations, i.e., $G_\alpha = \rho s \vec{u} \cdot \vec{S}_\alpha$. The implicit factored equation for s (corresponding to equations (22)) takes the (nonconservative) form

$$[1 + bh''\nu + h''(\lambda\partial_\xi - \nu\partial_\xi^2)]\rho\Delta s = (\rho\Delta s)_o \tag{38}$$

where $\lambda = u_n$, and h'' and ν are given as before. The term involving the parameter b is a critically important implicit source term used to balance the explicit RHS source term H_s. The importance and use of this term are discussed more fully in reference 5.

The viscous no-slip boundary conditions used with the two-equation models are $k = q = 0$ and $\varepsilon = 0$ for the $k - \varepsilon$ model, $\partial_\eta\omega = 0$ for the $q - \omega$ model, and an analytic BC, $\omega = 20\mu/(0.15\rho y^2)$, for the $k - \omega^2$ model. At inviscid inflow and outflow boundaries these variables are treated in the same manner as s and u_m (see eqs. (34) and (35)).

Optional boundary conditions using wall functions (ref. 13), or matching to the law-of-the-wall, are also provided in the code. Compared with the conventional procedure, this procedure is applicable only for high-Reynolds-number flows (i.e., not transitional flows) but has the advantages of requiring fewer points to resolve the boundary layer and can lead to substantial reductions in computing time. Our experience with this procedure is limited, however, and we will show no results of its use.

2.8 Time Step Selection

For time-accurate calculations, a spatially constant time step is used. For airfoils, the value of Δt chosen is $\Delta t \sim L/20c_\infty$, where L is the chord length of the airfoil and c_∞ is the free-stream sound speed.

For achieving rapid convergence to a steady-state solution, a spatially varying time step is used. We define the local explicit time step in the α direction to be Δt_α or

$$\Delta t_\alpha = \upsilon/(\vec{u} \cdot \vec{S}_\alpha + c|\vec{S}_\alpha|) \tag{39}$$

which is the characteristic time required for an inviscid disturbance to propagate across a cell at the velocity $|u_n| + c$. Utilizing this characteristic time, two options for a spatially varying or local time step are incorporated in the code. They are

Local time-step options

$$\text{(a)}\quad \Delta t(\xi,\eta) = \text{CFL} \cdot \min(\Delta t_\xi, \Delta t_\eta)$$
$$\text{CFL} = O\ (5\text{-}20)$$

$$\text{(b)}\quad \Delta t(\xi,\eta) = \text{CFL} \cdot \max(\Delta t_\xi, \Delta t_\eta)$$
$$\text{CFL} = O\ (1\text{-}10)$$

In most of the computations reported in this paper, option (a) was used with a Courant number of CFL = 6-10.

For viscous-flow problems at high Reynolds numbers where a finely spaced mesh is used adjacent to solid surfaces, the above formula (option (a)) leads to very small values of Δt in the neighborhood of the surface and results in slow convergence. In these cases, the formula is modified by placing a lower bound on Δt, i.e.,

$$\Delta t > \Delta t_{min} \sim L/20c_\infty \tag{40}$$

This procedure results in a much faster convergence to steady state.

In starting a calculation from a uniform initial state, a local time step using option (a) is normally used with CFL = 1. As the iterations progress, the CFL number is increased by a factor of 1.1 per step to its maximum value. For high-Reynolds-number viscous calculations, the time-step limit of equation (39) is imposed gradually over a period of about 50 cycles.

3. CODE ORGANIZATION AND DESCRIPTION

3.1 Summary of TURF Subroutines

The code is named TURF (turbulent upwind resolution of flow). A summaary of TURF subroutines is listed in table 1 along with a brief description of the function of each subroutine.

Input data (in the form of data statements) must be specified in the routines, MAIN, MSHA, MSHB, INCA, OUTA, CPLOT, AND SPLOT. These data are extensive and will not be discussed here. A manual explaining the code and how to set up test cases is currently under preparation and will be available in the near future.

3.2 Grid Generation

Algebraic grid generation routines are self-contained within the code (i.e., MSHA and MSHB). Examples of the types of grids that can be generated are shown in figure 3. They are basically of two types: (1) sheared parabolic and elliptic conformal mappings and (2) generalized H-grids where the points on two arbitrarily

defined curves are joined by straight lines. Spacing functions using uniform, exponential, and cubic-spline spacing are used to space grid points in both coordinate directions (SPAC). Combinations of different gridding procedures can be used to build up composite grids such as the airfoil grid shown in figure 3(a). A routine to compute the coordinates of NACA four-digit airfoils is also provided. For other airfoils and shapes a general curve drawing routine (CURV) is used to generate numerical coordinates from a set of specific input coordinates using linear or cubic-spline interpolation. Two types of airfoil grids can be generated: C-grids and O-grids. To date most numerical experience has been with airfoil C-type grids.

3.3 Data Loading System and Vectorization

A special data loading and sweep system is used (in MAIN) that is designed to permit implicit integration across interior mesh boundaries (see fig. 4). This integration is achieved through the use of two data-array systems (or commons) called BIG and SMALL. Each system contains the same variables (e.g., ρ, ρu . . .) but while the array BIG contains the data for the whole grid, the array SMALL contains the data for a single ξ or η grid line. Both array systems are one dimensional, as indicated below.

(a) BIG array: RO(LV), ROU(LV), . . .
LV = I + IDM * (J - 1), ξ = I, η = J

(b) SMALL array: R(L), RU(L), . . .
L = I or J

where IDM is the maximum dimension in the I direction. The variables shown here are density RO and R, and u-momentum ROU and RU.

The general procedure for performing difference operations (either explicit in RHS or implicit in LHS) along a particular ξ or η grid line is to first load the corresponding data from the BIG array into the SMALL array. The data in the SMALL array run sequentially while the data in the BIG array may not. Next, finite difference operations are performed on the SMALL array variables. Finally data from the SMALL array are off-loaded back into the BIG array. In this manner a single subroutine is used for differencing in either direction, which simplifies and reduces the amount of coding required. This procedure also permits implicit integration across internal mesh boundaries (fig. 4), thus allowing larger time steps than explicit procedures.

Except for the scalar tridiagonal solver routine (TRX), essentially all important DO loops in the code are vectorized on the Ames CRAY XMP computer. Work is

currently under way to vectorize the TRX routine (by interchanging DO loops) and it is anticipated that when this is accomplished computing times can be reduced by a factor of 2 to 3.

4. RESULTS

Representative results of inviscid and viscous flow calculations are shown in figures 5-9. These were steady-state calculations run with a spatially varying time step that used, except for one case, the second-order upwind method. Computing times on the Ames CRAY XMP computer for the viscous airfoil calculations using the full Navier-Stokes equations take 128 and 160 sec for the zero- and two-equation models, respectively, for 500 time steps using a 160 × 50 C-mesh. Steady-state lift and drag convergence is achieved between 200 and 400 steps. Computing times for airfoils using the Euler equations, on a 120 × 30 mesh take 48 sec for 500 steps and converge to a steady state in 100-300 steps.

4.1 Inviscid Flows

(a) Lifting transonic airfoil- Results of inviscid flow calculations are shown in figures 5 and 6. Computed surface pressures for the inviscid transonic flow about an NACA 0012 airfoil in free air are shown in figure 5. This figure compares results obtained using the present method with those obtained from reference 14 which used the central-differencing and flux-splitting (second-order upwind) methods. The three methods are in basic agreement except in the neighborhood of the shock wave where the present method shows a sharper shock capture.

(b) Oblique shock reflection- Figure 6 shows a calculation of a shock-reflection problem investigated by Yee et al. (refs. 8 and 15). Figures 6(a) and 6(b) show contour plots and pressure distributions using the second-order upwind method. A uniformly spaced 60 × 20 cell H-mesh was used in the calculations. Supersonic inflow and outflow boundary conditions were imposed at the upstream and downstream boundaries, respectively, and free-slip ($u_n = 0$) conditions were given at the lower reflecting surface. The Rankine-Hugoniot oblique-shock relations were used to impose conditions along the top boundary of the mesh.

Although the oblique shock wave is captured with reasonable accuracy, there is a distinct undershoot in the pressure distribution ahead of the shock wave and a smaller overshoot behind it. These may be essentially removed by a high resolution option (ref. 2) in which a switch to first-order spatial differencing is done at

locations of relative maxima and minima in the pressure distribution. Results using this option (labeled UW2II-HR) are shown in figures 6(c) and 6(d).

4.2 Viscous Flows

(a) Lifting transonic airfoil- Results of viscous turbulent flow calculations are shown in figures 7-9. Transonic calculations of an RAE 2822 airfoil are compared with experimental measurements (ref. 16) in figure 7. The calculations were done at the experimental geometric angle of attack and nominal Mach number and do not account for wind-tunnel-wall interference effects.

The meshes used in the transonic airfoil calculations were 120 × 50 (coarse) and 160 × 50 (fine) C-grids (see fig. 3(a)). The mesh points were exponentially spaced away from the airfoil surface and the cell spacing adjacent to the surface corresponded to a y^+ of 0.8. About 23 mesh points were contained in the boundary layer, and the outer boundary was placed at 10 chord lengths from the airfoil. The cell spacings in the streamwise direction over the central part of the airfoil were $\Delta x/c = 0.036$ and 0.024 for the coarse and fine meshes, respectively, with finer spacings used at the leading and trailing edges.

Boundary conditions for the airfoil calculations were as follows: no-slip velocity conditions at the airfoil surface, constant total enthalpy, entropy and tangential velocity component (u_m) around the outer C part of the mesh, and constant (free-stream) static-pressure conditions along the vertical back (outflow) boundary.

Computed Mach contours for the RAE airfoil, which illustrate the flow geometry and indicate overall computational quality, are shown in figure 7(a). Computed pressure distributions are compared with experimental measurements in figure 7(b). The computations were done with the $q - \omega$ two-equation model and include results using both the coarse- and fine-mesh systems. Pressures obtained using the various other turbulence models showed very little difference from the pressures obtained with the $q - \omega$ model. Almost identical results were achieved using the coarse- and fine-mesh systems, except in the region of the shock wave where small differences can be observed.

Skin-friction results are compared in figure 7(c). Since the computed results using the coarse mesh were very similar to those obtained with the fine mesh, only results obtained with the fine mesh are shown. For the zero-equation model, the boundary layer was assumed to be turbulent starting at the leading edge. The $q - \omega$ two-equation model predicts (natural) transition at an $x/c = 0.05$. The other two models indicate fully turbulent boundary layers starting at the leading edge. Of the two types of models tested, the zero-equation model gives the best overall prediction

of skin friction downstream of the shock wave. This trend in model performance is usually reversed, however, for flows with extensive separation. An example of this is given in the oblique shock boundary-layer interaction flow to be discussed shortly.

(b) Symmetric airfoil in a channel- Calculations of the transonic flow over an 18%-thick circular arc airfoil in a channel are compared with measurements (ref. 17) in figure 8. The shadowgraph and experimental pressure distributions of figures 8(a) and 8(c) show the basic features of the flow — an oblique shock-wave pattern, a thick zone of separated flow, and nearly critical values of the pressure coefficient downstream of the shock wave.

The numerical grid used in these calculations was an 80 × 60 H-mesh, the middle segment of which is shown in figure 3(b). The top mesh boundary is fitted to the channel wall and a plane of symmetry is used so that only the flow in the upper half of the channel need be computed. Boundary conditions at the upstream boundary were taken as constant total enthalpy and entropy conditions. Constant pressure conditions were taken at the downstream boundary. Along the top wall of the channel inviscid free-slip conditions were used, and along the airfoil surface viscous no-slip conditions were used. Inviscid free-slip conditions were also used along the symmetry plane upstream and downstream of the airfoil. The turbulence models utilized in the calculations included standard and modified forms of the Wilcox-Rubesin $k - \omega^2$ model described in references 1 and 5. The numerical scheme for computing the inviscid fluxes was an earlier version of the second-order central scheme reported in reference 1.

Numerical calculations of the pressure distribution along the airfoil surface are compared with measurements in figure 8(c). A plot of computed Mach contours is shown in figure 8(b) which may be compared with the shadowgraph of figure 8(a).

(c) Oblique shock boundary-layer interaction- As a final case, calculations of an oblique shock-wave boundary-layer interaction flow are compared with experimental measurements (ref. 18) in figure 9. The flow consists of an oblique shock wave impinging on a turbulent boundary layer at a free-stream Mach number of 2.9 and a unit Reynolds number of 5.7×10^7. The shock wave is sufficiently strong to cause substantial separation with a corresponding plateau in the measured pressure distributions (fig. 9(c)).

The numerical grid used in the calculations was a 100 × 50 H-mesh. Supersonic inflow and outflow conditions were used at the upstream and downstream boundaries, respectively, and a boundary-layer program was used to provide profiles of the state variables at the upstream boundary. Along the top surface, boundary conditions were

determined from the Rankine-Hugoniot conditions for oblique shock waves (in subroutine SHOCK). Results using both the zero-equation Cebeci-Smith model and the two-equation $q - \omega$ model are shown. The second-order upwind scheme was used for the inviscid flux differencing.

Computed Mach contours using the two models are shown in figures 9(a) and 9(b), and experimental pressure measurements are compared with computations in figure 9(c). It may be seen that the zero-equation model significantly underpredicts the amount of separation and the extent of upstream influence (indicated by the plateau in the experimental pressure distribution). The use of the two-equation model substantially improves predictions of surface pressure.

5. CONCLUDING REMARKS

In the preceding sections we have described a code for solving the compressible 2-D and axisymmetric Navier-Stokes equations in arbitrary curvilinear coordinates using the finite volume approach. Important features of the code, the numerical differencing procedures, boundary conditions, turbulent models, and time-step selection, were discussed. In addition, the general layout and operation of the code, including self-contained grid generation and plotting routines, were described. Finally, applications to a representative set of inviscid and viscous flow problems were presented.

Although much of the code is in a final state there are several areas in which improvements are needed. These are: (a) vectorization of the implicit operator, (b) time-step selection and convergence acceleration, (c) improved boundary conditions on the dissipation variables, $\delta W_{i+(1/2)}$, of equation (26), and (d) fully checked out law-of-the-wall boundary conditions for the turbulence models. Work in these and other areas is proceeding.

REFERENCES

1. Coakley, T. J.: Numerical Method for Gas Dynamics Combining Characteristic and Conservation Concepts. AIAA Paper 81-1257, June 1981.

2. Coakley, T. J.: Implicit Upwind Methods for the Compressible Navier-Stokes Equations. AIAA Paper 83-1958, July 1983.

3. Beam, R.; and Warming, R.: An Implicit Factored Scheme for the Compressible Navier-Stokes Equations. AIAA J., vol. 16, no. 4, 1978, pp. 393-402.

4. Briley, W. F.; and McDonald, H.: Solutions of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method. J. Comp. Phys., vol. 24, no. 4, Aug. 1977, p. 372.

5. Coakley, T. J.: Turbulence Modeling Methods for the Compressible Navier-Stokes Equations. AIAA Paper 83-1693, July 1983.

6. Pulliam, T. H.; and Chaussee, D. S.: A Diagonal Form of an Implicit Approximate Factorization Algorithm. J. Comp. Phys., vol. 39, no. 2, Feb. 1981.

7. Roe, P. L.: Approximate Reimann Solvers, Parameter Vector, and Difference Schemes. J. Comp. Phys., vol. 43, 1981, pp. 357-372.

8. Yee, H. C.; Warming, R. F.; and Harten, A.: On the Application and Extension of Harten's High Resolution Scheme. NASA TM-84256, June 1982.

9. Salas, M. D.; Jameson, A.; and Melnik, R. E.: A Comparative Study of the Non-uniqueness Problem of the Potential Equation. AIAA Paper 83-1888, July 1983.

10. Cebeci, T.; and Smith, A. M. O.: Analysis of Turbulent Boundary Layers, Academic Press, New York, 1974.

11. Chien, J. Y.: Predictions of Channel Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model. AIAA J., vol. 20, Jan 1982, pp. 33-38.

12. Wilcox, D. C.; and Rubesin, M. W.: Progress in Turbulence Modeling for Complex Flow Fields Including the Effects of Compressibility. NASA TP-151, 1980.

13. Viegas, J. R.; and Rubesin, M. W.: Wall Function Boundary Conditions in the Solution of the Navier-Stokes Equations for Complex Compressible Flows. AIAA Paper 83-1694, July 1983.

14. Buning, P. G.; and Steger, J. L.: Solution of the Two-Dimensional Euler Equations with Generalized Coordinate Transformating Using Flux Vector Splitting. AIAA Paper 82-0971, June 1982.

15. Yee, H. C.; Warming, R. F.; and Harten, A.: Implicit Total Variation Diminishing (TVD) Schemes for Steady-State Calculations. AIAA Paper 83-1902, July 1983.

16. Cook, P. H.; McDonald, M. A.; and Firmin, M. C. P.: AEROFOIL RAE 2822 — Pressure Distributions, and Boundary Layer and Wake Measurements. AGARD Advisory Report No. 138, 1979.

17. McDevitt, J. B.; Levy, L. L., Jr.; and Deiwert, G. S.: Transonic Flow About a Thick Circular Arc Airfoil. AIAA J., vol. 14, May 1976, pp. 606-613.

18. Reda, D. C.; and Murphy, J. D.: Sidewall Boundary-Layer Influence on Shock Wave/Turbulent Boundary-Layer Interaction. AIAA J., vol. 11, Oct. 1973, pp. 1367-1368.

TABLE 1.- SUMMARY OF TURF SUBROUTINES

Subroutine	Function
MAIN	Input data, loading and time-stepping loops
MSHA	General-purpose grid generation
MSHB	Airfoil grid generation
CURV	Space curves
SPAC	Spacing functions
FOIL	NACA 4 digit coordinates
MTRC	Surface vectors and volume elements
CURT	Curvature of boundary surface
MPLT	Mesh plots
INCA	Initial conditions (uniform ICs)
SHOCK	State behind oblique shock wave
PROF	Initial starting profiles for boundary layer
RHS	Right-hand side control and flux differencing
VISC	Molecular and turbulent viscosities
FLX	Flux vectors at interior cell boundaries
RBC	Flux vectors at external mesh boundaries
SORC	Turbulent source functions and time step
WALL	Law-of-the-wall boundary conditions
DELT	Boundary-layer thickness functions
LHS	Left-hand side control and update
TRX	Scalar tridiagonal solver
LBC	Implicit boundary conditions
OUTA	Output printer plots, field data, surface data
MAPO	Routine used for printer plots
CPLOT	Contour plots M, p, ψ
SPLOT	Surface data plots C_p, C_f, δ^*, θ
HEAD	Plot heading routine

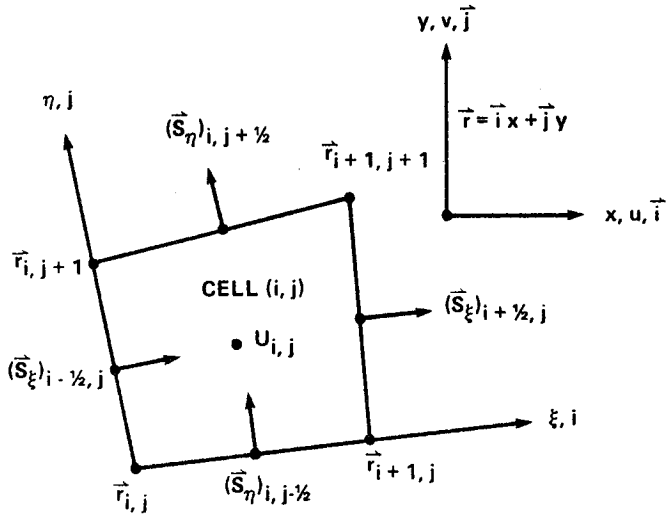


Figure 1.- Finite volume representation.

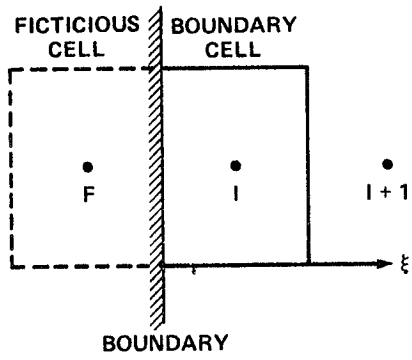


Figure 2.- Boundary cell indexing.

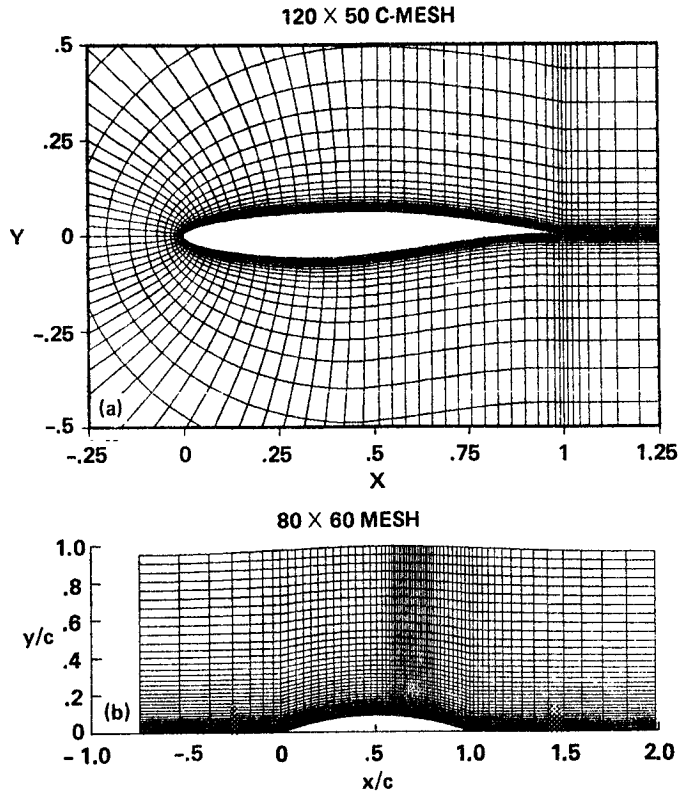


Figure 3.- Representative mesh systems. (a) C-mesh system for lifting airfoils; (b) H-mesh system for circular-arc airfoil in a channel.

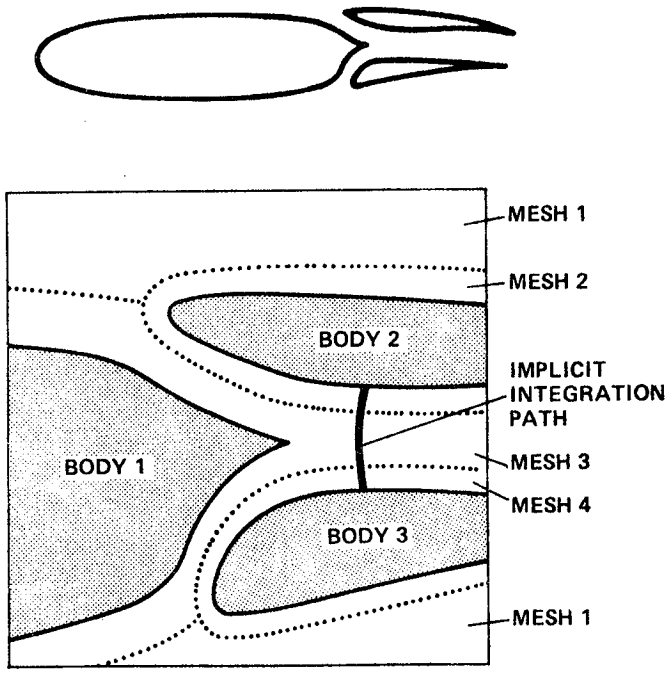


Figure 4.- Augmentor wing mesh system and implicit integration path (——).

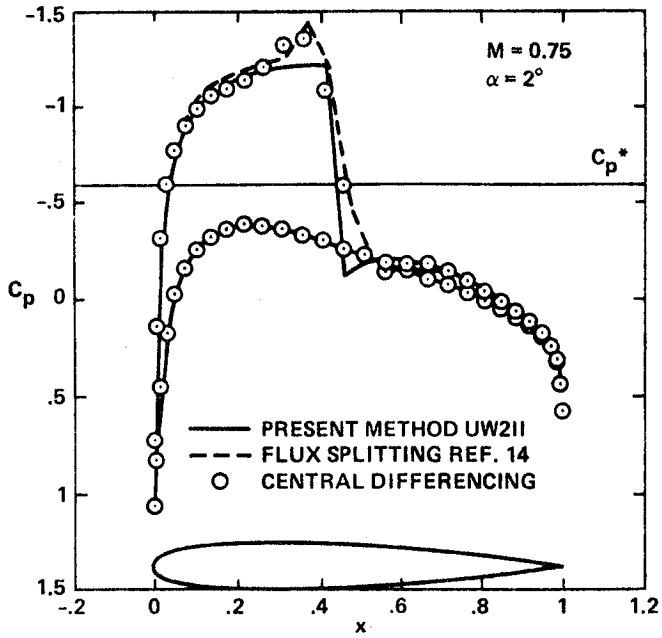


Figure 5.- Comparison of computed surface pressures for inviscid transonic flow about an NACA 0012 airfoil: M = 0.75, α = 2°, 80 × 20 cell C-mesh (79 × 31 mesh for ref. 14).

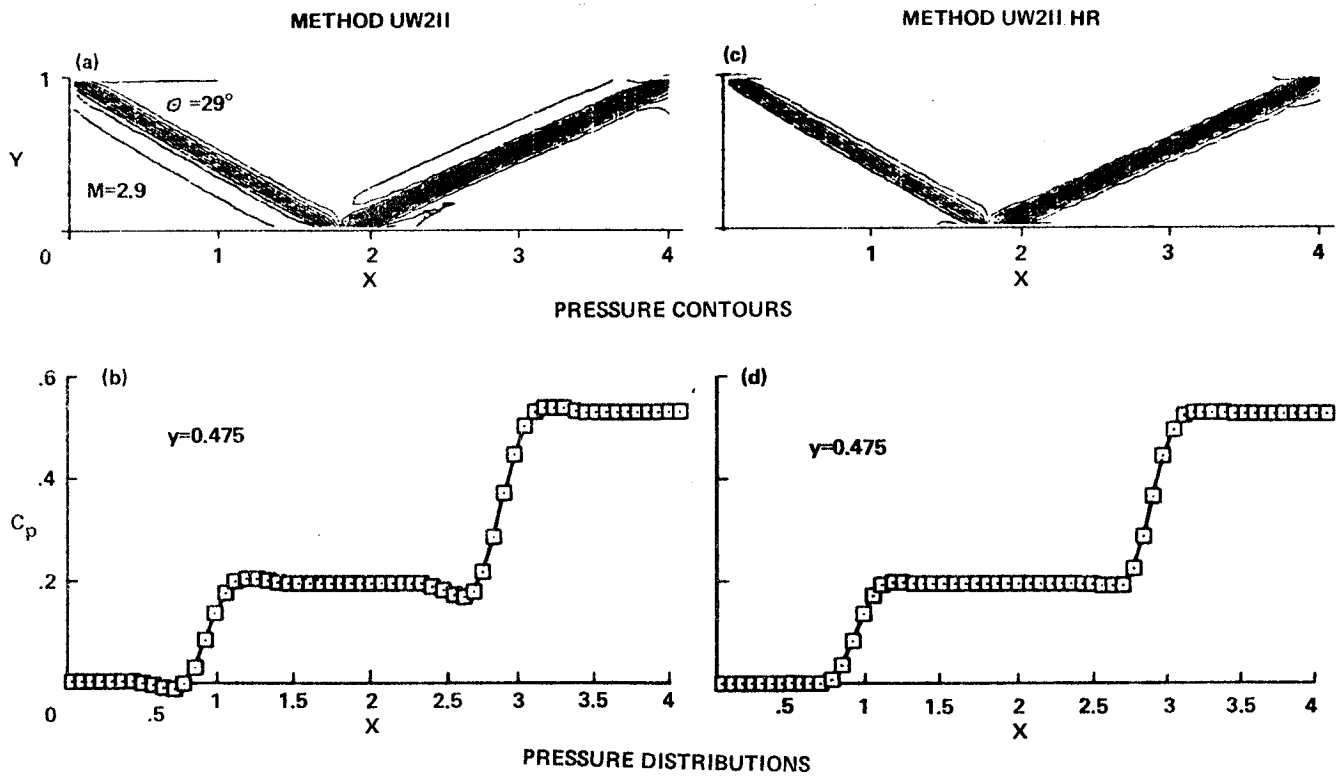
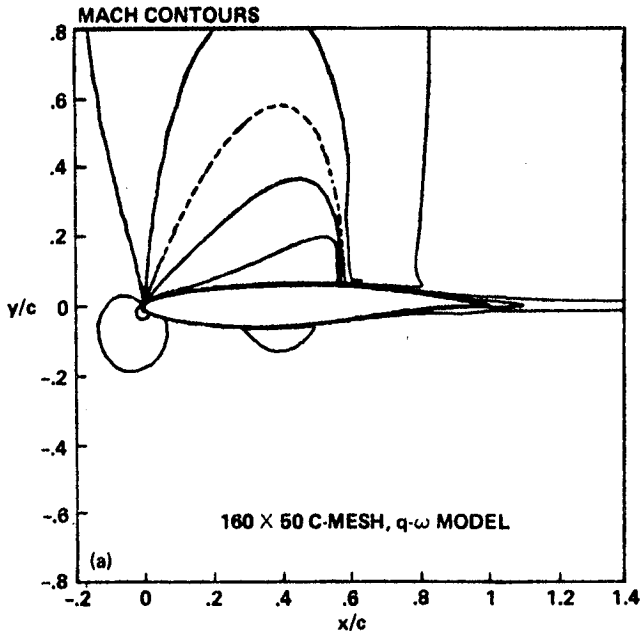
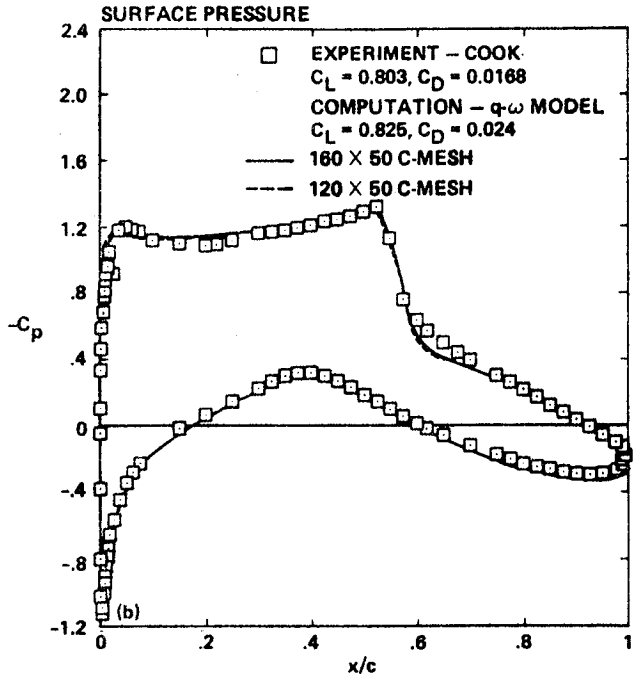


Figure 6.- Comparison of computed pressure contours and distributions for inviscid shock reflection problem: M = 2.9, θ = 29°, 60 × 20 cell H-mesh. (a) Contours, method UW2II; (b) pressure distributions, method UW2II; (c) contours, method UW2II-HR; (d) pressure distributions, method UW2II-HR.





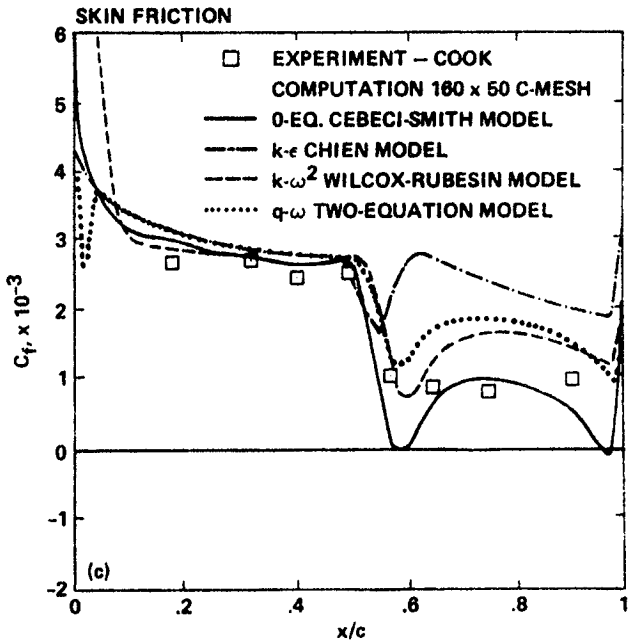


Figure 7.- Transonic flow about RAE 2822 airfoil, case 9: M = 0.73, Re = 6.5 × 10^6, α = 3.19. (a) Computed Mach contours; (b) surface pressures; (c) skin friction, upper surface.

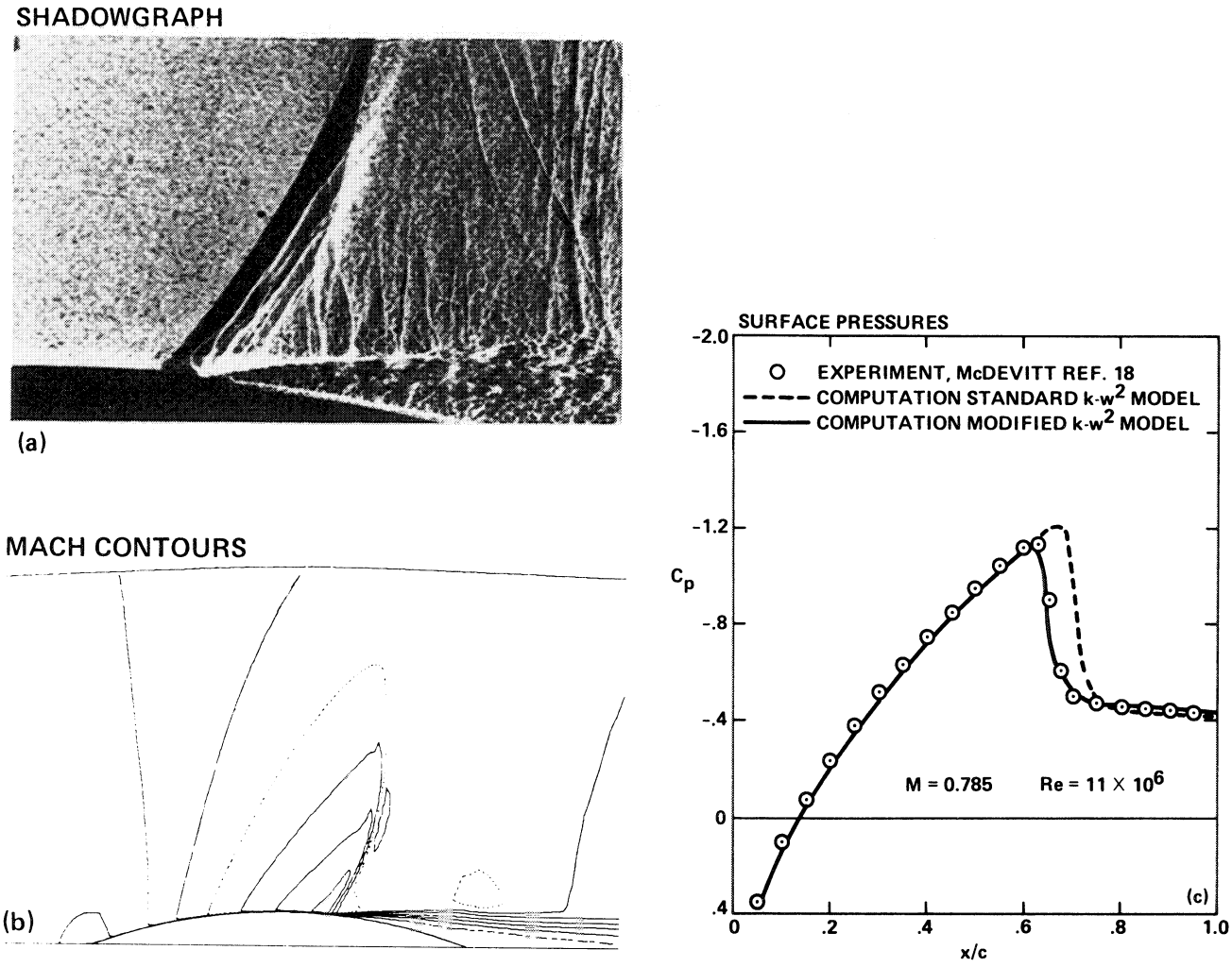


Figure 8.- Transonic flow over an 18% thick airfoil in a channel: M = 0.785, Re = 11 × 10^6. (a) Shadowgraph; (b) computed Mach contours; (c) experimental and computed surface pressures.

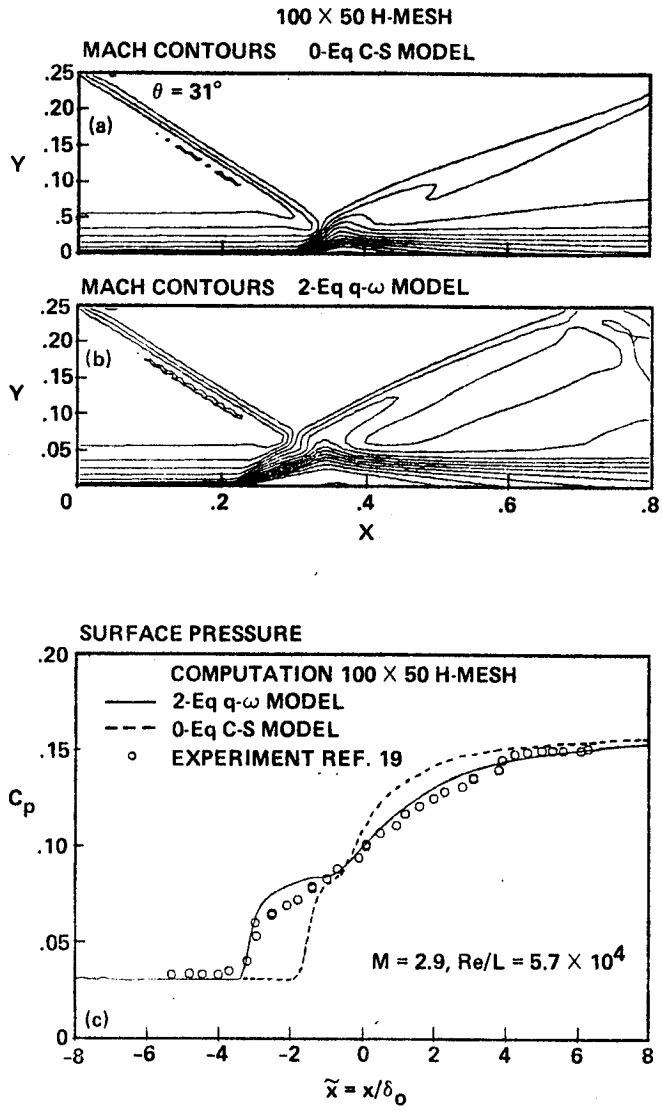


Figure 9.- Oblique shock-wave boundary-layer interaction: M = 2.9, Re/L = 5.7 × 10^7.
(a) Mach contours, zero-equation model; (b) Mach contours, q - ω model;
(c) surface pressures.

1. Report No. NASA TM 85926	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A COMPRESSIBLE NAVIER-STOKES CODE FOR TURBULENT-FLOW MODELING		5. Report Date February 1984 6. Performing Organization Code ATP
7. Author(s) Thomas J. Coakley		8. Performing Organization Report No. A-9669 10. Work Unit No. T 6384
9. Performing Organization Name and Address Ames Research Center Moffett Field, California 94035		11. Contract or Grant No. 13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code 505-51-11

15. Supplementary Notes

Point of contact: Thomas J. Coakley,
Ames Research Center, M.S. 239A-1 FTS 448-6451
Moffett Field, California 94035 (415) 965-6451

16. Abstract

An implicit, finite volume code for solving two-dimensional, compressible turbulent flows is described. Second-order upwind differencing of the inviscid terms of the equations is used to enhance stability and accuracy. A diagonal form of the implicit algorithm is used to improve efficiency. Several zero- and two-equation turbulence models are incorporated to study their impact on overall flow-modeling accuracy. Applications to several external and internal flows are discussed.

17. Key Words (Suggested by Author(s))	18. Distribution Statement		
Fluid mechanics Navier-Stokes equations Turbulence modeling Computer program Numerical methods	Unlimited Subject category: 34		
19. Security Classif. (of this report) Uncl.	20. Security Classif. (of this page) Uncl.	21. No. of Pages 33	22. Price* A03

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