1-4-82

LA-8872

I-682

DR 184

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

VNAP2: A Computer Program for Computation of Two-Dimensional, Time-Dependent, Compressible, Turbulent Flow

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DISCLAIMER

This work was supported by the Propulsion Aerodynamics Branch of the National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia.

Edited by Patricia Metropolis
Photocomposition by IS-6 Composing Section

LA--8872

DE82 007360

LA-8872

UC-32
Issued: August 1981

VNAP2:
A Computer Program for Computation of Two-Dimensional, Time-Dependent, Compressible, Turbulent Flow

Michael C. Cline

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

PAGES ii to iii

WERE INTENTIONALLY

LEFT BLANK

CONTENTS

VNAP2: A COMPUTER PROGRAM FOR COMPUTATION OF TWO-DIMENSIONAL, TIME-DEPENDENT, COMPRESSIBLE, TURBULENT FLOW

by

Michael C. Cline

ABSTRACT

VNAP2 is a computer program for calculating turbulent (as well as laminar and inviscid), steady, and unsteady flow. VNAP2 solves the two-dimensional, time-dependent, compressible Navier-Stokes equations. The turbulence is modeled with either an algebraic mixing-length model, a one-equation model, or the Jones-Launder two-equation model. The geometry may be a single- or a dual-flowing stream. The interior grid points are computed using the unsplit MacCormack scheme. Two options to speed up the calculations for high Reynolds number flows are included. The boundary grid points are computed using a reference-plane-characteristic scheme with the viscous terms treated as source functions. An explicit artificial viscosity is included for shock computations. The fluid is assumed to be a perfect gas. The flow boundaries may be arbitrary curved solid walls, inflow/outflow boundaries, or free-jet envelopes. Typical problems that can be solved concern nozzles, inlets, jet-powered afterbodies, airfoils, and free-jet expansions. The accuracy and efficiency of the program are shown by calculations of several inviscid and turbulent flows. The program and its use are described completely, and six sample cases and a code listing are included.

I. THE BASIC METHOD

A. Introduction

VNAP2 is a computer program for calculating turbulent (as well as laminar and inviscid), steady, and unsteady flow. VNAP2 is a modified version of the VNAP code discussed in Ref. 1. Like the VNAP code, VNAP2 solves the two-dimensional (2D, axisymmetric), time-dependent, compressible Navier-Stokes equations by a second-order-accurate finite-difference method. Unlike the VNAP code, VNAP2 allows arbitrary grid spacing, has two options to speed up the calculations for high Reynolds number flows, contains three different turbulence models, and can solve either single- or dual-flowing stream geometries. This last option allows the VNAP2 code to compute internal/external flows, such as inlets, and jet-powered afterbodies as well as airfoils.

Because of the variable grid and the options to speed up the calculations for high Reynolds number flows, VNAP2 computes high Reynolds number flows much more efficiently than VNAP. However, full-scale Reynolds numbers (10^6-10^8) still require fairly long run times (see Sec. I.G). In addition, determination of a reasonable variable grid and selection of the best numerical scheme parameters for high Reynolds number flows require a certain amount of trial and error.

Although the VNAP code replaced the NAP^2 code, VNAP2 is not necessarily intended to replace the VNAP code. Although VNAP2 can handle all the flows that VNAP is capable of solving, as well as many additional flows, VNAP2 is approximately double the size of VNAP and somewhat more complex. As a result, VNAP2 is more difficult to modify as well as to run on smaller computing systems. For these reasons, many users may prefer to use both codes.

B. Discussion

The VNAP2 code follows the philosophy of the VNAP code; that is, the boundary grid points are the most important. In addition, except for purely supersonic inflow and outflow, these grid points are generally the most difficult. For these reasons, the construction of boundary grid point routines is not left to the general user, and VNAP2 contains complete and accurate routines for calculating all boundary grid points. Several different boundary conditions are included as options, and all unspecified variables are calculated using a second-order-accurate, reference-plane-characteristic scheme, with the viscous terms treated as source functions. The code also continually checks for subsonic or supersonic flow, as well as inflow or outflow, to apply the correct boundary conditions. Most of the options for inflow and outflow boundary conditions include nonreflecting conditions to accelerate the convergence to steady state.

Like VNAP, VNAP2 employs the unsplit MacCormack scheme[3] to compute the interior grid points. The governing equations are left in nonconservation form. For flows with thin boundary layers or free shear layers, the small grid spacing required for resolution greatly increases the computer time. To reduce this time, the grid points in the finer parts of the mesh are subcycled. In addition, an explicit modification to the MacCormack scheme (allowing the removal of the speed of sound from the C-F-L condition and thus increasing the time-step size) is also included. An explicit artificial viscosity model stablizes the computations for shock waves.

C. Governing Equations

The 2D time-dependent, compressible, Navier-Stokes equations for turbulent flow of a perfect gas can be written as

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + \rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\varepsilon v}{y}\right)$$

$$= \bar{\bar{\alpha}}\left[\frac{\partial}{\partial x}\left(\frac{\mu_T}{\rho}\frac{\partial \rho}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\mu_T}{\rho}\frac{\partial \rho}{\partial y}\right) + \frac{\varepsilon \mu_T}{\rho y}\frac{\partial \rho}{\partial y}\right], \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} = \frac{1}{\rho}\frac{\partial}{\partial x}\left[(\lambda + 2\mu)\frac{\partial u}{\partial x} + \lambda\frac{\partial v}{\partial y}\right] + \frac{1}{\rho}\frac{\partial}{\partial y}\left[\mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right]$$

$$+ \frac{\bar{\alpha}}{\rho}\left[u\frac{\partial}{\partial x}\left(\frac{\mu_T}{\rho}\frac{\partial \rho}{\partial x}\right) + v\frac{\partial}{\partial y}\left(\frac{\mu_T}{\rho}\frac{\partial \rho}{\partial x}\right)\right] + \frac{\varepsilon}{\rho y}\left[(\lambda + \mu)\frac{\partial v}{\partial x} + \mu\frac{\partial u}{\partial y} + \frac{\bar{\alpha}\mu_T v}{\rho}\frac{\partial \rho}{\partial x}\right]$$

$$- \frac{1}{\rho}\frac{2}{3}\frac{\partial \rho q}{\partial x}, \tag{2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} = \frac{1}{\rho}\frac{\partial}{\partial y}\left[(\lambda + 2\mu)\frac{\partial v}{\partial y} + \lambda\frac{\partial u}{\partial x}\right] + \frac{1}{\rho}\frac{\partial}{\partial x}\left[\mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right]$$

$$+ \frac{\bar{\alpha}}{\rho}\left[v\frac{\partial}{\partial y}\left(\frac{\mu_T}{\rho}\frac{\partial \rho}{\partial y}\right) + u\frac{\partial}{\partial x}\left(\frac{\mu_T}{\rho}\frac{\partial \rho}{\partial y}\right)\right] + \frac{\varepsilon}{\rho y}\left[(\lambda + 2\mu)\left(\frac{\partial v}{\partial y} - \frac{v}{y}\right) + \frac{\bar{\alpha}\mu_T v'}{\rho}\frac{\partial \rho}{\partial y}\right]$$

$$- \frac{1}{\rho}\frac{2}{3}\frac{\partial \rho q}{\partial y} , \tag{3}$$

$$\frac{\partial p}{\partial t} + u\frac{\partial p}{\partial x} + v\frac{\partial p}{\partial y} - a^2\left(\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y}\right) = (\gamma - 1)\left\{(\lambda_M + 2\mu_M)\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right]\right.$$

$$+ \mu_M\left[\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right] + 2\lambda_M\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} + 2\mu_M\frac{\partial v}{\partial x}\frac{\partial u}{\partial y} + \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)$$

$$- \bar{\alpha}RT\left[\frac{\partial}{\partial x}\left(\frac{\mu_T}{\rho}\frac{\partial \rho}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\mu_T}{\rho}\frac{\partial \rho}{\partial y}\right)\right] + \frac{\varepsilon}{y}\left[(\lambda_M + 2\mu_M)\frac{v^2}{y} + 2\lambda_M v\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right.$$

$$\left.\left. + k\frac{\partial T}{\partial y} - \frac{\bar{\alpha}RT\mu_T}{\rho}\frac{\partial \rho}{\partial y}\right] + \rho e\right\} , \tag{4}$$

and

$$p = \rho RT , \tag{5}$$

where ρ is the density; p is the pressure; T is the temperature; u and v are the velocity components; q is the turbulence energy; e is the turbulence dissipation rate; a is the speed of sound; R is the gas constant; $\mu = \mu_M + \mu_T$; $\lambda = \lambda_M + \lambda_T$; μ_M and λ_M are the first and second coefficients of molecular viscosity; μ_T and λ_T are the corresponding turbulent quantities; γ is the ratio of specific heats; $k = k_M + k_T$; k_M is the coefficient of molecular conductivity; k_T is the turbulent value; x and y are the space coordinates; t is the time; $\bar{\alpha}$ is a constant; and ε is 0 for planar flow and 1 for axisymmetric flows. Equations (2)-(4) are written for the two-equation turbulence model. For the mixing-length and one-equation models discussed below, Eqs. (2)-(4) are slightly different. The density gradient terms, premultiplied by the constant $\bar{\alpha}$, on the right-hand side of Eqs. (1)-(4) are from turbulent density fluctuations and are, therefore, zero for laminar flows. Equation (1) is the conservation of mass or continuity equation, Eqs. (2) and (3) are the x and y momentum equations, respectively, and Eq. (4) is the internal energy equation written in terms of pressure using the equation of state for a perfect gas, Eq. (5). Thus there is a system of five equations for the eight unknowns u, v, p, ρ, T, μ_T, λ_T, and k_T. (In the two-equation turbulence model, there are two additional equations for the unknowns q and e.) To close this set of equations, the turbulence quantities μ_T, λ_T and k_T need definition. VNAP2 uses the following three turbulence models to accomplish this.

1. **Mixing-Length Turbulence Model.** The first model is an algebraic mixing-length model that can be written as

$$\mu_T = \rho\ell^2\left[\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right]^{1/2} , \tag{6}$$

$$\lambda_T = \lambda\mu_T/\mu , \tag{7}$$

and

$$k_T = \gamma R \mu_T / (\gamma - 1) Pr_T \ , \tag{8}$$

where ℓ is the mixing length defined below and Pr_T is the turbulent Prandtl number. For free shear layer flows, the model follows Ref. 4. For monotonic velocity profiles, ℓ is defined as

$$\ell = C_{ML2} \cdot |y_2 - y_1| \ , \tag{9}$$

where C_{ML2} is a constant and

$$y_1 = y \quad \text{for} \quad \frac{u - u_L}{u_U - u_L} = 0.1 \ ,$$

$$y_2 = y \quad \text{for} \quad \frac{u - u_L}{u_U - u_L} = 0.9 \ ,$$

and u_L and u_U are the lower and upper velocities of a monotonically increasing or decreasing velocity profile. For free shear flows with a velocity profile that has the minimum velocity u_M in the interior, ℓ is defined as

$$\ell = C_{ML1} \cdot |y_2 - y_1| \ , \tag{10}$$

where C_{ML1} is a constant and

$$y_1 = y \quad \text{for} \quad \frac{u - u_L}{u_M - u_L} = 0.1 \text{ and } y < y_2 \ ,$$

$$y_1 = y \quad \text{for} \quad \frac{u - u_M}{u_U - u_M} = 0.9 \text{ and } y > y_2 \ ,$$

and

$$y_2 = y \quad \text{for} \quad u = u_M \ .$$

The program continually checks to determine the type of velocity profile present. If u_M is within 5% of the minimum of u_L or u_U, then the monotonic profile is assumed. This check on the size of u_M is intended to stop small velocity variations, away from the shear region, from switching the velocity profile type. The 5% value is arbitrary and can be changed in subroutine MIXLEN (see Sec. II. A). On the centerline or midplane, Eq. (6) is replaced by

$$\mu_T = \rho \ell^3 \left| \frac{\partial^2 u}{\partial y^2} \right| \ . \tag{11}$$

For boundary-layer flows, the Cebeci-Smith[5] two-layer model is used. In the inner layer, ℓ is defined as

$$\ell = 0.4y \left[1.0 - \exp\left(\frac{-y\sqrt{\rho \tau_w}}{26.0\, \mu_M} \right) \right] \ , \tag{12}$$

where y is the distance from the wall and τ_w is the shear stress at the wall. In the outer layer, Eqs. (6) and (12) are replaced by

$$\mu_T = 0.0168 \rho u_E \delta^* \left[1.0 + 5.5 \frac{y}{\delta}^6 \right]^{-1} , \quad (13)$$

where u_E is the velocity at the edge of the boundary layer, δ is the boundary-layer velocity thickness, and δ^* is the boundary-layer displacement thickness given by

$$\delta^* = \int_0^\delta \left(1 - \frac{\rho u}{\rho_E u_E} \right) dy \ .$$

The switch from the inner-layer model, given by Eqs. (6) and (12), to the outer-layer model, given by Eq. (13), occurs when the inner μ_T is greater than the outer value. This model does not employ a relaxation or lag parameter. The values for C_{ML1} and C_{ML2} are 0.125 for planar flows and 0.11 for axisymmetric flows.

For this model, the last term on the right-hand side of Eqs. (2)-(4) vanishes. In addition, the viscosity coefficients λ_M and μ_M in the first four terms on the right-hand side of Eq. (4) as well as the first two axisymmetric terms, also in Eq. (4), are replaced by λ and μ.

2. One-Equation Turbulence Model. This model was developed at Los Alamos National Laboratory by Bart J. Daly. At present, this model has not been extensively proof-tested and, therefore, should be considered experimental. The model attempts to combine the best features of the algebraic mixing-length models and the two-equation models.

This model consists of the following transport equation for the turbulence energy q,

$$\frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} + v \frac{\partial q}{\partial y} = \frac{2}{3} \frac{q}{\rho} \left(\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} \right)$$

$$+ \frac{\lambda_T + 2\mu_T}{\rho} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right]$$

$$+ \frac{\mu_T}{\rho} \left[\left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 \right] + \frac{2\lambda_T}{\rho} \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} + \frac{2\mu_T}{\rho} \frac{\partial v}{\partial x} \frac{\partial u}{\partial y}$$

$$+ \frac{1}{\rho} \frac{\partial}{\partial x} \left[\left(\mu_M + \frac{\mu_T}{\sigma_q} \right) \frac{\partial q}{\partial x} \right] + \frac{1}{\rho} \frac{\partial}{\partial y} \left[\left(\mu_M + \frac{\mu_T}{\sigma_q} \right) \frac{\partial q}{\partial y} \right] - \frac{2\mu_M q \Delta}{\rho S^2}$$

$$- \frac{2\bar{\alpha} q}{3\rho} \left[\frac{\partial}{\partial x} \left(\frac{\mu_T}{\rho} \frac{\partial \rho}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\mu_T}{\rho} \frac{\partial \rho}{\partial y} \right) \right] + \frac{\varepsilon}{y} \left[\frac{\lambda_T + 2\mu_T}{\rho} \frac{v^2}{y} + \frac{2\lambda_T v}{\rho} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right.$$

$$\left. + \frac{\mu}{\rho} \frac{\partial q}{\partial y} - \frac{2\bar{\alpha} q \mu_T}{\rho^2} \frac{\partial \rho}{\partial y} \right] , \quad (14)$$

where

$$S = C_q \ell \ , \quad (15)$$

$$\Delta = \begin{cases} 5 & \text{for } \dfrac{S\rho\sqrt{2q}}{\mu_M} \leqslant 5 \\ \\ \dfrac{S\rho\sqrt{2q}}{\mu_M} & \text{for } \dfrac{S\rho\sqrt{2q}}{\mu_M} > 5 \ , \end{cases} \tag{16}$$

ℓ is the mixing length from the first model, and c_q is a constant. The turbulent viscosity μ_T is defined as

$$\mu_T = \begin{cases} 0.1\, C_\mu \dfrac{\rho^2 S^2 q}{\mu_M} & \text{for } \dfrac{S\rho\sqrt{2q}}{\mu_M} \leqslant 5 \\ \\ 0.3534\, C_\mu \rho S\sqrt{q} & \text{for } \dfrac{S\rho\sqrt{2q}}{\mu_M} > 5 \ , \end{cases} \tag{17}$$

where C_q is 17.2 for planar flows and 12.3 for axisymmetric flows and $C_\mu = 0.09$. The quantities λ_T and k_T are determined from Eqs. (7) and (8), respectively.

For this model, the last term on the right-hand side of Eq. (4) is replaced with $2\mu_M q\Delta/S^2$.

3. Two-Equation, Jones-Launder[6-9] Turbulence Model. This model employs two transport equations, one for the turbulence energy q and the second for the turbulence dissipation rate e. These equations can be written as

$$\begin{aligned} \frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} + v\frac{\partial q}{\partial y} &= \frac{\lambda_T + 2\mu_T}{\rho}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] + \frac{\mu_T}{\rho}\left[\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right] \\ &+ \frac{2\lambda_T}{\rho}\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} + \frac{2\mu_T}{\rho}\frac{\partial v}{\partial x}\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial}{\partial x}\left[\left(\mu_M + \frac{\mu_T}{\sigma_q}\right)\frac{\partial q}{\partial x}\right] + \frac{1}{\rho}\frac{\partial}{\partial y}\left[\left(\mu_M + \frac{\mu_T}{\sigma_q}\right)\frac{\partial q}{\partial y}\right] \\ &- e - \frac{2\mu}{\rho}\left(\frac{\partial q^{1/2}}{\partial x} + \frac{\partial q^{1/2}}{\partial y}\right)^2 + \frac{\varepsilon}{y}\left[\frac{\lambda_T + 2\mu_T}{\rho}\frac{v^2}{y} + \frac{2\lambda_T v}{\rho}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \frac{1}{\rho}\left(\mu_M + \frac{\mu_T}{\sigma_q}\right)\frac{\partial q}{\partial y}\right] \end{aligned} \tag{18}$$

and

$$\begin{aligned} \frac{\partial e}{\partial t} + u\frac{\partial e}{\partial x} + v\frac{\partial e}{\partial y} &= \frac{C_1 e}{q}\left\{\frac{\lambda_T + 2\mu_T}{\rho}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] + \frac{\mu_T}{\rho}\left[\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right]\right. \\ &\left. + \frac{2\lambda_T}{\rho}\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} + \frac{2\mu_T}{\rho}\frac{\partial v}{\partial x}\frac{\partial u}{\partial y}\right\} + \frac{1}{\rho}\frac{\partial}{\partial x}\left[\left(\mu_M + \frac{\mu_T}{\sigma_e}\right)\frac{\partial e}{\partial x}\right] \\ &+ \frac{1}{\rho}\frac{\partial}{\partial y}\left[\left(\mu_M + \frac{\mu_T}{\sigma_e}\right)\frac{\partial e}{\partial y}\right] - \frac{C_2 e}{q}\left[e - \frac{2\mu}{\rho}\left(\frac{\partial q^{1/2}}{\partial x} + \frac{\partial q^{1/2}}{\partial y}\right)\right. \\ &+ \frac{2\mu_M\mu_T}{\rho^2}\left[\left(\frac{\partial^2 u}{\partial x^2}\right)^2 + \left(\frac{\partial^2 v}{\partial x^2}\right)^2 + \left(\frac{\partial^2 u}{\partial y^2}\right)^2 + \left(\frac{\partial^2 v}{\partial y^2}\right)^2\right] \end{aligned}$$

$$+\frac{\varepsilon}{y}\left\{\frac{C_1 e}{q}\left[\frac{\lambda_T+2\mu_T}{\rho}\frac{v^2}{y}+\frac{2\lambda_T v}{\rho}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right]+\frac{1}{\rho}\left(\mu_M+\frac{\mu_T}{\sigma_e}\right)\frac{\partial e}{\partial y}\right.$$

$$\left.+\frac{2y\mu_M\mu_T}{\rho^2}\left[\left(\frac{1}{y}\frac{\partial u}{\partial y}\right)^2+\left(\frac{1}{y}\frac{\partial v}{\partial y}\right)^2+\frac{2}{y}\frac{\partial u}{\partial y}\frac{\partial^2 u}{\partial y^2}+\frac{2}{y}\frac{\partial v}{\partial y}\frac{\partial^2 v}{\partial y^2}\right]\right\} , \qquad (19)$$

where

$$\left.\begin{aligned} &C_1 = 1.44,\ \sigma_q = 1.0,\ \sigma_e = 1.3\ , \\ &C_2 = \overline{C}_2\ [1.0 - 0.2222\ \exp(-0.0278\ R_T^2)], \\ &\text{and} \\ &R_T = \rho q^2/\mu_M e\ . \end{aligned}\right\} \qquad (20)$$

The turbulent viscosity is calculated from

$$\mu_T = C_\mu \exp[-3.4/(1+0.02\ R_T)^2]\rho q^2/e\ , \qquad (21)$$

where $C_\mu = 0.09$. The quantities λ_T and k_T are determined from Eqs. (7) and (8), respectively. The solid wall boundary condition on e for this version of the Jones-Launder model is $\partial e/\partial y = 0$.

For strongly separated flows, this model has two numerical problems. One problem is that the turbulence dissipation rate becomes extremely small near a reattachment point. To overcome this, a lower bound on q and e at a given y was added as an option to VNAP2 in the manner of Coakley and Viegas.[10] The second problem is associated with the treatment of the convection terms in Eqs. (18) and (19). In the far field where $q \to 0$, the variations of q and e are such in some problems that extremely large values of μ_T occur. Using the donor cell scheme in the x direction and the MacCormack scheme in the y direction removes this problem for all cases tested so far. Also included is the following fourth-order smoothing term added to Eq. (18):

$$C_Q\left(\frac{(|u|+a)\Delta x^3}{q}\left|\frac{\partial^2 q}{\partial x^2}\right|\frac{\partial^2 q}{\partial x^2}+\frac{(|v|+a)\Delta y^3}{q}\left|\frac{\partial^2 q}{\partial y^2}\right|\frac{\partial^2 q}{\partial y^2}\right) , \qquad (22)$$

where C_Q is a constant. A similar term, with e replacing q and C_E replacing C_Q, is added to Eq. (19). These smoothing terms were added as a possible alternative to the donor cell differencing. However, at this time, the donor cell differencing appears to be more satisfactory.

4. Artificial Viscosity Model. To stabilize the numerical method for shock wave calculations, an explicit artificial viscosity model is included. This model replaces the explicit fourth-order smoothing usually employed by MacCormack.[11] The procedure here is first to calculate artificial viscosity coefficients μ_A, λ_A and a thermal conductivity coefficient k_A and, second, to add these values to the molecular values. These quantities are calculated from the following equations:

$$\lambda_A = C\ C_\lambda\ \Delta x \Delta y \rho\left|\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\varepsilon\frac{v}{y}\right| , \qquad (23)$$

$$\mu_A = C_{\mu 1}\lambda_A/C_\lambda\ , \qquad (24)$$

and

$$k_A = \gamma R \mu_A/(\gamma-1)Pr_A\ , \qquad (25)$$

where C, C_λ, $C_{\mu 1}$, and Pr_A are constants, with Pr_A representing an artificial Prandtl number, and Δx and Δy are the mesh spacing. The following artificial density smoothing term also is added to the right-hand side of Eq. (1).

$$\text{Equation (1)} = \frac{C_\rho}{\rho}\left[\frac{\partial}{\partial x}\left(\mu_A \frac{\partial \rho}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_A \frac{\partial \rho}{\partial y}\right) + \frac{\varepsilon \mu_A}{y}\frac{\partial \rho}{\partial y}\right] , \tag{26}$$

where C_ρ is a constant. When the divergence of the velocity is greater than zero (expansions), these artificial quantities are set equal to zero.

D. Physical and Computational Flow Spaces

Figure 1 shows the physical flow-space geometry, with flow from left to right. The upper boundary, called the wall, can be either a solid boundary, a free-jet boundary, or an arbitrary subsonic (normal to the boundary) inflow/outflow boundary. The lower boundary, called the centerbody, can be either a solid boundary or a plane (line) of symmetry. The geometry can be either a single-flowing stream or, if the dual-flow-space walls are present, a dual-flowing stream. The dual-flow-space walls may begin in the interior and continue to the exit (inlet geometry), may begin at the inlet and terminate in the interior as shown in Fig. 1 (afterbody geometry), or may begin and end in the interior (airfoil geometry). All of the above boundaries may be arbitrary curved boundaries provided the y coordinate is a single value function of x. If the dual-flow-space walls begin or end in the interior, then they must have pointed ends. The points can be very blunt, but there cannot be vertical walls. The left boundary is a subsonic, supersonic, or mixed inflow boundary, whereas the right boundary is a subsonic, supersonic, or mixed outflow boundary or a subsonic inflow/outflow boundary.

The x, y, t physical space is mapped into a rectangular ζ, η, τ computational space as shown in Fig. 1. The mapping is carried out in two stages: the first maps the physical space to a rectangular computational space and the second maps the variable grid spacing to a uniform grid spacing. Because the single- and dual-flow-space mappings are different, they will be discussed separately.

1. **Single-Flow Space.** The x, y, t physical space, with variable grid spacing, is mapped into the $\bar{\zeta}$, $\bar{\eta}$, $\bar{\tau}$ space, which also has variable grid spacing, by the following transformation:

$$\bar{\zeta} = x ; \bar{\eta} = \frac{y - y_C}{y_W - y_C} ; \bar{\tau} = t , \tag{27}$$

where y_C is a function of x and denotes the centerbody y value and y_W is a function of x and t and denotes the wall y value. The quantity $\bar{\eta}$ varies between 0 and 1. This variable grid $\bar{\zeta}$, $\bar{\eta}$, $\bar{\tau}$ space is mapped into a uniform grid ζ, η, τ space by the following transformation:

$$\zeta = \zeta(\bar{\zeta}); \quad \eta = \eta(\bar{\eta}); \quad \tau = \bar{\tau} ; \tag{28}$$

that is, ζ is an arbitrary tabular function of $\bar{\zeta}$, etc. Using Eqs. (27) and (28), the derivatives become

$$\frac{\partial}{\partial x} = \omega \frac{\partial}{\partial \zeta} + \alpha \frac{\partial}{\partial \eta} ; \quad \frac{\partial}{\partial y} = \beta \frac{\partial}{\partial \eta} ; \quad \frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} + \delta \frac{\partial}{\partial \eta} , \tag{29}$$

where

$$\omega = \frac{d\zeta}{d\bar{\zeta}}; \quad \beta = \frac{d\eta}{d\bar{\eta}} \frac{1}{y_w - y_C}; \quad \alpha = \beta \left[(\bar{\eta} - 1) \frac{dy_C}{dx} - \bar{\eta} \frac{\partial y_w}{\partial x} \right]; \quad \delta = -\bar{\eta} \beta \frac{\partial y_w}{\partial t} . \tag{30}$$

The derivatives $d\zeta/d\bar{\zeta}$ and $d\eta/d\bar{\eta}$ are computed numerically using differences consistent with the MacCormack scheme.

This results in a physical space grid with the following properties: one set of grid lines is straight and in the y direction with arbitrary spacing in the x direction; the second set of grid lines approximately follows the wall and centerbody contours; the Δy spacing of these grid lines is arbitrary at one x location and is proportional to those values at any other x location; and the proportionality factor is based on the distance between y_W and y_C. For more details on the physical space grid, see the example shown in Fig. 2 as well as the computed results in Sec. I.G.

2. **Dual-Flow Space.** If part of the flow in the dual-flow-space example is a single-flow space, then the single-flow-space option discussed above is used in that part. In the dual-flow-space section, the procedure is to divide the dual-flow space into two single-flow spaces and then to use the single-flow-space transformations discussed above. Both the upper and lower dual-flow-space walls collapse to the same grid line in the computational space, as shown in Fig. 1. The flow variables at the grid points on the upper dual-flow-space wall are stored in the regular solution array, whereas the lower wall variables are stored in a dummy array. These flow variables are continually switched between the two arrays during the calculation. For the dual-flow-space example, Eq. (27) becomes

$$\left.\begin{aligned} &\bar{\zeta} = x; \quad \bar{\eta} = c \, \frac{y - y_C}{y_L - y_C}; \quad \bar{\tau} = t \qquad \text{for} \quad y_C \leqslant y \leqslant y_L , \\ &\bar{\zeta} = x; \quad \bar{\eta} = c + (1 - c) \, \frac{y - y_U}{y_W - y_U}; \quad \bar{\tau} = t \quad \text{for} \quad y_U \leqslant y \leqslant y_W , \end{aligned}\right\} \tag{31}$$

where y_L and y_U are functions of x and denote the lower and upper dual-flow-space walls, respectively. The parameter c is a constant and equals $(y_L - y_C)/(y_W - y_U + y_L - y_C)$ evaluated at a specified x location. For completely dual flows, c can be evaluated at any x and in practice is evaluated at the left boundary. However, for flows with both dual- and single-flow-space parts, c must be evaluated at the x location where the dual-flow-space walls either begin or end. This ensures that the single grid line that corresponds to the lower and upper dual-flow-space walls remains continuous as it extends into the single-flow-space section. If the dual-flow-space walls begin and end in the interior, as in the case of a planar airfoil, then the values of c must be equal at both ends of the dual-flow-space walls. This requirement means that if y_C and y_W are straight horizontal lines, then the airfoil must be at a zero angle of attack. If the upper boundary or wall is the arbitrary inflow/outflow option, then y_W can be adjusted to produce an angle of attack. However, if the upper boundary or wall is a fixed solid boundary, as in the case of an airfoil in a wind tunnel, then the angle of attack of the airfoil relative to the wall is fixed. For the axisymmetric case, the airfoil becomes a duct and the angle of attack discussion deals with the duct-axial area variation. For the dual-flow-space example, Eqs. (28) and (29) remain unchanged, and Eq. (30) becomes

$$
\left.\begin{aligned}
&\omega=\frac{d\zeta}{d\bar{\zeta}};\quad \beta=\frac{d\eta}{d\bar{\eta}}\frac{c}{y_L-y_C};\quad \alpha=\frac{\beta}{c}\left[(\bar{\eta}-c)\frac{dy_C}{dx}-\bar{\eta}\frac{dy_L}{dx}\right];\quad \delta=0\\
&\quad\text{for}\quad y_C\leqslant y\leqslant y_L\ ,\\
&\text{and}\\
&\omega=\frac{d\zeta}{d\bar{\zeta}};\quad \beta=\frac{d\eta}{d\bar{\eta}}\frac{1-c}{y_W-y_U};\quad \alpha=\frac{\beta}{1-c}\left[(\bar{\eta}-1)\frac{dy_U}{dx}-(\bar{\eta}-c)\frac{\partial y_W}{\partial x}\right];\\
&\quad\delta=\frac{\beta(\bar{\eta}-c)}{1-c}\frac{\partial y_W}{\partial t}\quad\text{for}\quad y_U\leqslant y\leqslant y_W\ .
\end{aligned}\right\}\qquad(32)
$$

3. Transformed Governing Equations. Using Eqs. (27) and (29), the original governing equation can be written in the ζ, η, τ variables. For example Eq. (1) becomes

$$
\frac{\partial\rho}{\partial\tau}+u\omega\frac{\partial\rho}{\partial\zeta}+\bar{v}\frac{\partial\rho}{\partial\eta}+\rho\left(\omega\frac{\partial u}{\partial\zeta}+\alpha\frac{\partial u}{\partial\eta}+\beta\frac{\partial v}{\partial\eta}+\frac{\varepsilon v}{y}\right)
$$

$$
=\frac{\bar{\alpha}}{\rho}\left\{\left(\omega\frac{\partial}{\partial\zeta}+\alpha\frac{\partial}{\partial\eta}\right)\left[\mu_T\left(\omega\frac{\partial\rho}{\partial\zeta}+\alpha\frac{\partial\rho}{\partial\eta}\right)\right]+\beta\frac{\partial}{\partial\eta}\left(\mu_T\beta\frac{\partial\rho}{\partial\eta}\right)+\varepsilon\frac{\mu_T\beta}{y}\frac{\partial\rho}{\partial\eta}\right\}\ ,\qquad(33)
$$

where

$$
\bar{v}-u\alpha+v\beta+\delta\ ,\qquad(34)
$$

$$
\left.\begin{aligned}
&y=y_C+\eta\,(y_W-y_C) && \text{for the single-flow space}\ ,\\
&y=y_C+\frac{\bar{\eta}}{c}(y_L-y_C) && \text{for the lower dual-flow space}\ ,\\
&y=y_U+\frac{\bar{\eta}-c}{1-c}(y_W-Y_U) && \text{for the upper dual-flow space}\ ,
\end{aligned}\right\}\qquad(35)
$$

and the u and v velocity components are the original values.

E. Numerical Method

The computational plane grid points are divided into interior and boundary points. The boundary grid points are further divided into left-boundary, right-boundary, wall, centerbody, and dual-flow-space wall points (see Fig. 1).

1. Interior Grid Points. The interior grid points are computed using the unsplit MacCormack scheme discussed in Ref. 3. This scheme is a second-order-accurate, noncentered, two-step, finite-difference scheme. Backward differences are used on the first step, forward differences on the second. The governing equations are left in nonconservation form. As an example of the basic scheme, the finite-difference equations for Eq. (2) for planar ($\varepsilon=0$), laminar ($\bar{\alpha}=q=0$) flow are

$$
\begin{aligned}
\bar{u}^{N+1}_{L,M} = u^{N}_{L,M} - \Delta t \Bigg[& u^{N}_{L,M}\left(\frac{u^{N}_{L,M} - u^{N}_{L-1,M}}{\Delta x}\right) + v^{N}_{L,M}\left(\frac{u^{N}_{L,M} - u^{N}_{L,M-1}}{\Delta y}\right) \\
& + \frac{1}{\rho^{N}_{L,M}}\left(\frac{p^{N}_{L,M} - p^{N}_{L-1,M}}{\Delta x}\right)\Bigg] \\
& + \frac{\Delta t}{\rho^{N}_{L,M}\Delta x}\Bigg[(\lambda + 2\mu)_{L+1/2,M}\left(\frac{u^{N}_{L+1,M} - u^{N}_{L,M}}{\Delta x}\right) \\
& + \lambda_{L+1/2,M}\left(\frac{v^{N}_{L+1,M+1} + v^{N}_{L,M+1} - v^{N}_{L+1,M-1} - v^{N}_{L,M-1}}{4\Delta y}\right) \\
& - (\lambda + 2\mu)_{L-1/2,M}\left(\frac{u^{N}_{L,M} - u^{N}_{L-1,M}}{\Delta x}\right) \\
& - \lambda_{L-1/2,M}\left(\frac{v^{N}_{L,M+1} + v^{N}_{L-1,M+1} - v^{N}_{L,M-1} - v^{N}_{L-1,M-1}}{4\Delta y}\right)\Bigg] \\
& + \frac{\Delta t}{\rho^{N}_{L,M}\Delta y}\Bigg[\mu_{L,M+1/2}\left(\frac{v^{N}_{L+1,M+1} + v^{N}_{L+1,M} - v^{N}_{L-1,M+1} - v^{N}_{L-1,M}}{4\Delta x}\right) \\
& + \mu_{L,M+1/2}\left(\frac{u^{N}_{L,M+1} - u^{N}_{L,M}}{\Delta y}\right) \\
& - \mu_{L,M-1/2}\left(\frac{v^{N}_{L+1,M} + v^{N}_{L+1,M-1} - v^{N}_{L-1,M} - v^{N}_{L-1,M-1}}{4\Delta x}\right) \\
& - \mu_{L,M-1/2}\left(\frac{u^{N}_{L,M} - u^{N}_{L,M-1}}{\Delta y}\right)\Bigg] ,
\end{aligned}
\tag{36}
$$

for the first step and

$$
\begin{aligned}
u^{N+1}_{L,M} = 0.5\Bigg\{ & u^{N}_{L,M} + \bar{u}^{N+1}_{L,M} - \Delta t\Bigg[\bar{u}^{N+1}_{L,M}\left(\frac{\bar{u}^{N+1}_{L+1,M} - \bar{u}^{N+1}_{L,M}}{\Delta x}\right) \\
& + \bar{v}^{N+1}_{L,M}\left(\frac{\bar{u}^{N+1}_{L,M+1} - \bar{u}^{N+1}_{L,M}}{\Delta y}\right) + \frac{1}{\bar{\rho}^{N+1}_{L,M}}\left(\frac{\bar{p}^{N+1}_{L+1,M} - \bar{p}^{N+1}_{L,M}}{\Delta x}\right)\Bigg] + Q\Bigg\}
\end{aligned}
\tag{37}
$$

for the second step, where the subscripts L and M denote axial and radial grid points, respectively, the superscript N denotes the time step, the bar denotes values calculated on the first step, and Q denotes the terms in the last two brackets on the right-hand side of Eq. (36), that is, the viscous terms. Equations (36) and (37) show that all viscous terms are calculated using center differences in the initial-value plane only, so that they are second-order accurate in space but first-order accurate in time. Raising them to second-order accuracy in time requires re-evaluating them using the $\bar{u}^{N+1}$ values from the first step. For most problems, this greater accuracy does not seem worth the increased effort.

To improve the computational efficiency for high Reynolds number flows, the grid points in the fine part of the grid may be subcycled. This is accomplished by first computing the grid points in the coarse part of the grid for one time step Δt. Next, the grid points in the fine grid are calculated k times (where k is an integer) with a time step $\Delta t/k$. The grid points at the edge of the fine grid require a special procedure, because one of their neighboring points is calculated as part of the coarse grid. Except for the first subcycled time step, this point is unknown. However, the values at t and t + Δt are known from the coarse grid solution, so that the values between t and t + Δt are determined by linear interpolation.

To improve the computational efficiency further, a special procedure (called the Quick Solver) is employed to increase the allowable time step in the subcycled part of the grid. This procedure allows the removal of the sound speed from the time-step C-F-L condition. Procedures that accomplish this have been proposed by Harlow and Amsden[12] and MacCormack.[13] The procedure of Harlow and Amsden removes the sound speed, in both the x and y directions, by an implicit treatment of the mass equation and the pressure gradient terms in the momentum equations. MacCormack's procedure is explicit and removes the sound speed in only one direction. (It also includes an implicit procedure to remove the viscous diffusion restriction from the time-step C-F-L condition.) Because explicit schemes are easier to program for efficient computation on vector computers and because high Reynolds number flows usually require fine grid spacing in only one direction, a procedure similar to MacCormack's was chosen.

MacCormack's procedure is based on the assumption that the velocity component, in the coordinate direction with the fine grid spacing, is negligible compared to the sound speed. This allows the governing equations to be simplified. MacCormack then applies the Method of Characteristics to these simplified equations. However, for flows over bodies with large amounts of curvature as well as many shear flows, this assumption is questionable; and because VNAP2 is intended as a general code for solving a variety of problems, MacCormack's assumption seems too restrictive. Therefore, the main differences between MacCormack's scheme and the one presented below are that this restriction is removed and that the flow in the y direction is assumed to be subsonic.

The sound speed limitation is associated with the inviscid part of the Navier-Stokes equations. In addition, because the following procedure is used only in the y direction, it can be illustrated by using the following inviscid, one-dimensional (1D) equations

$$\frac{\partial \rho}{\partial t} + v\frac{\partial \rho}{\partial y} + \rho\frac{\partial v}{\partial y} = 0 \ , \tag{38}$$

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} = 0 \ , \tag{39}$$

and

$$\frac{\partial p}{\partial t} + v\frac{\partial p}{\partial y} + \rho a^2\frac{\partial v}{\partial y} = 0 \ , \tag{40}$$

where v is the velocity, ρ is the density, p is the pressure, a is the speed of sound, y is distance, t is time, Eq. (38) is the continuity equation, Eq. (39) is the momentum equation, and Eq. (40) is the internal energy equation written in terms of pressure using the equation of state for an ideal gas. The time step for explicit methods used to solve Eqs. (38)-(40) is the C-F-L condition and can be written as $\Delta t \leqslant \Delta y/(|v| + a)$. However, to improve the computational efficiency in the boundary layers, where Δy and v are small but a is large, a procedure that allows $\Delta t \leqslant \Delta y/|v|$ is developed. Writing Eqs. (38)-(40) in characteristic form yields

$$\frac{dp}{dt} = a^2 \frac{d\rho}{dt} \quad \text{for} \quad \frac{dy}{dt} = v \ , \tag{41}$$

$$\frac{dp}{dt} + \rho a \frac{du}{dt} = 0 \quad \text{for} \quad \frac{dy}{dt} = v + a \ , \tag{42}$$

and

$$\frac{dp}{dt} - \rho a \frac{du}{dt} = 0 \quad \text{for} \quad \frac{dy}{dt} = v - a \ . \tag{43}$$

Therefore, Eq. (41) applies along the flow streamline and Eqs. (42) and (43) apply along Mach lines. Thus, if a time step $\Delta t \leqslant \Delta y/|v|$ is selected for some finite-difference method, the domain of dependence for Eq. (41) is included in the adjacent grid points, but the domain of dependence of Eqs. (42) and (43) is outside the adjacent grid points. This larger domain of dependence can be determined by solving for the intersection of the characteristics of Eqs. (42) and (43) with the initial-value surface. Using these intersection points allows differences to be calculated for the larger domain of dependence in much the same manner as for the adjacent grid points.

The final step is to determine which derivatives in Eqs. (38)-(40) depend on the streamline (the adjacent grid points) and which derivatives depend on the Mach lines (the characteristic initial-value surface intersection points). Following the procedure used by Kentzer[14] in his boundary condition scheme and replacing the total derivatives along characteristics in Eqs. (41)-(43) with partial derivatives, while denoting the space derivatives in Eq. (42) with bars and Eq. (43) with hats give

$$\frac{\partial \rho}{\partial t} + v \frac{\partial \rho}{\partial y} - \frac{1}{a^2}\left(\frac{\partial p}{\partial t} + v \frac{\partial p}{\partial y}\right) = 0 \ , \tag{44}$$

$$\frac{\partial v}{\partial t} + \frac{v+a}{2}\frac{\partial \bar{v}}{\partial y} + \frac{v-a}{2}\frac{\partial \hat{v}}{\partial y} + \frac{1}{\rho a}\left(\frac{v+a}{2}\frac{\partial \bar{p}}{\partial y} - \frac{v-a}{2}\frac{\partial \hat{p}}{\partial y}\right) = 0 \ , \tag{45}$$

and

$$\frac{\partial p}{\partial t} + \frac{v+a}{2}\frac{\partial \bar{p}}{\partial y} + \frac{v-a}{2}\frac{\partial \hat{p}}{\partial y} + \rho a\left(\frac{v+a}{2}\frac{\partial \bar{v}}{\partial y} - \frac{v-a}{2}\frac{\partial \hat{v}}{\partial y}\right) = 0 \, . \tag{46}$$

The derivatives without bars or hats are calculated by the unsplit MacCormack scheme using the adjacent grid points, with backward differences on the predictor step and forward differences on the corrector step. For the bar derivatives the following procedure is employed: first, the values of the dependent variables at the point (denoted by 1 in Fig. 3) where the v + a Mach line intersects the initial-value surface N are determined by linear interpolation; then the bar derivatives, using v as an example, are evaluated by

$$\frac{\partial \bar{v}}{\partial y} = \frac{[C_S v_M^N + (1 - C_S)(v_{M+1}^N + V_{M-1}^N)/2] - v_1}{y_M - y_1} \tag{47}$$

on the predictor step and

$$\frac{\partial \bar{v}}{\partial y} = \frac{v_M^{N+1} - v_{M-1}^{N+1}}{y_M - y_{M-1}} , \tag{48}$$

on the corrector step. The hat derivatives are calculated by

$$\frac{\partial \hat{v}}{\partial y} = \frac{v_2 - [C_S v_M^N + (1 - C_S)(v_{M+1}^N + v_{M-1}^N)/2]}{y_2 - y_M} \tag{49}$$

on the predictor step and

$$\frac{\partial \hat{v}}{\partial y} = \frac{v_{M+1}^{N+1} - v_M^{N+1}}{y_{M+1} - y_M} \tag{50}$$

on the corrector step. The coefficient C_S is usually set equal to 0.5. If the intersection points 1 and 2 in Fig. 3 lie outside the computational grid, then reflection is used to obtain flow variables at these points from points inside the grid.

The above analysis used the 1D equations to illustrate the method. The actual equations used are derived from the ζ = constant reference-plane-characteristic scheme used at the wall boundary. The Mach line compatibility equations, without the viscous and ζ direction convection source terms, are computed on the large domain as discussed above. The streamline compatibility equation, including all source terms, is computed using the standard MacCormack scheme.

The above procedure for evaluating the terms that depend on the sound speed is only first-order accurate in space. Using the ideas of the λ scheme[15] could probably produce second-order accuracy. However, this was not done because the procedure is used only in boundary and shear layers where the viscous terms dominate the sound speed terms and, in addition, the λ scheme increases the size of the domain of dependence of the difference scheme.

2. **Left-Boundary Grid Points.** The left boundary can only be an inflow boundary. For supersonic inflow, u, v, p, and ρ are specified. The temperature is determined from the equation of state. For subsonic inflow, there are three different boundary condition options. The first specifies the total pressure p_T, total temperature T_T, and flow angle θ as proposed by Serra.[16] The second and third, which are discussed by Oliger and Sundström,[17] specify either u, v, and ρ or p, v, and ρ. For a discussion of the relative merits of these boundary conditions, see Sec. I.F. Following the ideas of Moretti and Abbett,[18] all the unspecified dependent variables are computed using a second-order-accurate, reference-plane-characteristic scheme. In this scheme, the partial derivatives with respect to η in the convective terms are computed in the initial-value and solution surfaces using noncentered differencing as in the MacCormack scheme. In the viscous terms, the partial derivatives with respect to η are computed as in the interior point scheme and the derivatives with respect to ζ are calculated using reflection. The cross derivative viscous terms are set equal to zero. These convection and viscous term derivatives are then treated as source terms, and the resulting system of equations is solved in the η = constant reference planes using a two-step, two-independent-variable characteristic scheme. The characteristic relation that couples the interior flow to the boundary is derived following the procedure of Ref. 1 and can be written as

$$dp - \rho a du = (\psi_4 + a^2\psi_1 - \rho a \psi_2)d\tau \quad \text{for} \quad d\zeta = \omega(u - a)d\tau , \tag{51}$$

where the first equation is called the compatibility equation and the second is called the characteristic curve equation. The ψ terms follow the definitions in Ref. 1. Equation (51) may be written in finite-difference form by first replacing the differentials by differences along the characterisitic curve. The coefficients are either evaluated in the initial-value plane (first step) or considered as averages of the coefficients evaluated in both the initial-value and solution planes (second step). A discussion of the unit processes and details of the schemes are given in Ref. 1.

For the p_T, T_T, and θ boundary condition, the following equations that relate the stagnation or total conditions to the static conditions are required.

$$p_T/p = [1 + (\gamma - 1)M^2/2]^{\gamma/(\gamma-1)} \tag{52}$$

and

$$T_T/T = 1 + (\gamma - 1)M^2/2 \ , \tag{53}$$

where γ is the ratio of specific heats, M is the Mach number, T is the temperature, and the subscript T denotes the stagnation or total conditions. The solution procedure is as follows: M is assumed, p and T are calculated from Eqs. (52) and (53), ρ is calculated from the equation of state, u is calculated from Eq. (51), v is calculated from the specified flow angle, a new M is calculated from u, v, p, and ρ, and the process is continued until the change in M has converged to 10^{-3}.

For the u, v, and ρ boundary condition, there is only one unspecified variable, p, which can be calculated from Eq. (51). Likewise, the p, v, and ρ boundary condition has one unspecified variable, u, which also can be determined from Eq. (51). In both cases the temperature is determined from the equation of state.

Both the u, v, and ρ and the p, v, and ρ boundary conditions include a nonreflecting option based on the ideas of the outflow boundary condition of Rudy and Strikwerda.[19] Rudy and Strikwerda use the following equation

$$\frac{\partial p}{\partial t} - \rho a \frac{\partial u}{\partial t} + C_\alpha(p - p_e) = 0 \tag{54}$$

to replace the outflow boundary conditon $p = p_e$, where p_e is the exit pressure and C_α is a constant. The first two terms of Eq. (54) can be interpreted as the 1D compatibility equation on the incoming characteristic (where this characteristic is parallel to the boundary) and the last term is included to asymptotically enforce the specification of the exit pressure. Forcing the incoming characteristic to be parallel to the boundary as if the outflow were sonic removes normal reflections back into the interior. This interpretation of Rudy and Strikwerda's outflow boundary condition allows formulation of a similar procedure for inflow boundaries. Therefore, for the inflow case, using the p, v, and ρ boundary condition, Eq. (54) becomes

$$\frac{\partial p}{\partial t} + \rho a \frac{\partial u}{\partial t} + C_\alpha(p - p_I) = 0 \ , \tag{55}$$

where p_I is the specified inflow pressure. For the u, v, and ρ boundary conditions, Eq. (54) becomes

$$\frac{\partial u}{\partial t} + \frac{1}{\rho a}\frac{\partial p}{\partial t} + C_\alpha(u - u_I) = 0 \ , \tag{56}$$

where u_I is the specified inflow velocity. Equation (55) or (56) is solved with Eq. (51) to determine u and p at the inflow boundary.

For mixed supersonic/subsonic inflow, VNAP2 uses the supersonic boundary condition at grid points where the flow is supersonic and either the u, v, and ρ or the p, v, and ρ boundary conditions (but not the p_T, T_T, and θ boundary conditions) at the subsonic points. VNAP2 allows using the supersonic boundary condition everywhere as an option.

The turbulence model boundary conditions are the specification of q for the one-equation model and q and e for the Jones-Launder two-equation model. The specified values of q and e can be determined following a procedure similar to that of Ref. 4. The value of q is calculated from

$$q = \frac{\mu_T |\partial u/\partial y|}{0.3\rho} \quad ,$$

where $|\partial u/\partial y|$ and ρ can be determined from the inflow velocity profile and μ_T can be determined by the mixing-length model. The value of e for the two-equation model can be calculated from Eq. (21). For large R_T, Eq. (21) reduces to

$$\mu_T = C_\mu \, \rho \, q^2/e \quad ,$$

which can be easily solved for e, and for small R_T a trial and error solution can be used. For some flows this procedure produces values of q that are much lower than the evolved value at the first downstream grid point. However, increasing q to agree with the first downstream grid point value, while adjusting e to keep μ_T constant, produces little change in the solution. If the p_T, T_T, and θ inflow boundary condition is used then a short run can be made, using the mixing-length model, to determine an inflow velocity profile. If the inflow profile is a uniform flow profile, that is, no shearing flow is present, then the inflow values of q and e can be set to some small values so that μ_T is negligible when compared to the molecular value.

3. Right-Boundary Grid Points. The right boundary can be a supersonic outflow boundary or a subsonic inflow/outflow boundary. This subsonic inflow option is required for internal flows with flow separation at the right boundary. For supersonic outflow, the flow variables are extrapolated. For subsonic outflow, the exit pressure is specified and the remaining variables are calculated using a characteristic scheme similar to the left-boundary scheme. The characteristic relations that couple the interior flow to the boundary are derived following the procedure of Ref. 1 and can be written as

$$\left.\begin{aligned} dp - a^2 d\rho &= \psi_4 d\tau \quad (57) \\ dv &= \psi_3 d\tau \quad (58) \end{aligned}\right\} \quad \text{for} \quad d\zeta = \omega u d\tau$$

and

$$dp + \rho a du = (\psi_4 + a^2\psi_1 + \rho a \psi_2)\, dt \qquad \text{for} \quad d\zeta = \omega(u + a)d\tau \quad (59)$$

These equations are written in finite-difference form like those for the left-boundary scheme. The pressure is specified, and the u velocity component is then calculated from Eq. (59); the density from Eq. (57); the v velocity component from Eq. (58); and the temperature from the equation of state. If subsonic reverse flow (inflow) occurs at the right boundary, inflow boundary conditions must be specified. This is accomplished by leaving p equal to the specified exit pressure, setting ρ equal to the value at the boundary where separation occurred, and setting the flow angle equal to the value obtained by linear interpolation between the boundaries. The ρ and v boundary conditions used here are arbitrary and can be changed by modifying subroutine EXITT (see Sec. II.A).

The code includes the nonreflecting outflow boundary condition of Rudy and Strikwerda.[19] Here, u and p are calculated from Eqs. (54) and (59); the density from Eq. (57); v from Eq. (58); and T from the equation of state. This nonreflection option is also used when reverse flow occurs.

For mixed supersonic/subsonic outflow, VNAP2 uses the supersonic boundary condition at grid points where the flow is supersonic and the subsonic boundary condition at subsonic points. VNAP2 allows using either the supersonic or subsonic boundary conditions everywhere as an option.

The turbulence model boundary conditions are the extrapolation of q for the one-equation model and q and e for the Jones Launder two-equation model.

4. Wall Grid Points. The wall boundary can be a free-slip boundary, a free-jet boundary, a no-slip boundary, or a constant pressure inflow/outflow boundary. The constant pressure inflow/outflow boundary is required for external flows.

a. Free-Slip Boundary. For a free-slip boundary, a reference-plane-characteristic scheme is used. Partial derivatives with respect to ζ in the convective terms are computed in the initial-value and solution surfaces using noncentered differencing as in the MacCormack scheme. All derivatives in the viscous terms are computed in the initial-value surface only, using centered differencing. The η and cross derivatives in the viscous terms are calculated by either reflecting or extrapolating a row of fictitious mesh points outside the flow boundary. These convection and viscous term derivatives are then treated as source terms, and the resulting system of equations is solved in the ζ = constant reference planes using a two-step, two-independent-variable characteristic scheme.

The characteristic relations that couple the interior flow to the boundary are derived following the procedure of Ref. 1 and can be written as

$$\beta du - \alpha dv = (\beta\psi_2 - \alpha\psi_3)d\tau \quad (60)$$

$$dp - a^2 d\rho = \psi_4 d\tau \quad \text{for} \quad d\eta = \bar{v} d\tau \quad (61)$$

and

$$dp + \rho\alpha a\, du/\alpha^* + \rho\beta a dv/\alpha^* = (\psi_4 + a^2\psi_1 + \rho\alpha a\psi_2/\alpha^* + \rho\beta a\psi_3/\alpha^*)d\tau$$
$$\text{for} \quad d\eta = (\bar{v} + \alpha^* a)d\tau \ , \quad (62)$$

where

$$\alpha^* = (\alpha^2 + \beta^2)^{1/2} \ .$$

These equations are written in finite-difference form like those for the left-boundary scheme.

The boundary condition is that the flow is tangent to the boundary. This can be written as

$$v = u \tan\theta + \partial y_w/\partial t \ , \quad (63)$$

where θ is the local boundary angle. The time derivative is present because, in the free-jet option, the wall boundary coordinates are a function of time. Equation (63) is substituted into Eq. (60), and the resulting equation is solved for the velocity component u. Then the v velocity component is obtained from Eq. (63); the pressure from Eq. (62); the density from Eq. (61); and the temperature from the equation of state.

The turbulence model boundary conditions are the extrapolation of q for the one-equation model and q and e for the Jones-Launder two-equation model.

This code has an option to improve the accuracy of the calculation of one sharp expansion corner on the wall contour. The flow at this corner must be supersonic and the boundary condition option must be the free-slip boundary with no free jet. The grid point is treated by a special procedure. First, an upstream solution is computed at the corner grid point, using the upstream flow tangency condition as the boundary condition and backward ζ differences in both initial-value and solution planes. Next, a downstream solution is calculated, using the Prandtl-Meyer exact solution and the stagnation conditions from the upstream grid point. The upstream solution is used when computing wall grid points upstream of the corner grid point as well as the adjacent interior grid point; the downstream solution is used when computing downstream wall grid points.

b. Free-Jet Boundary. The free-jet boundary grid points are computed by the wall routine so that the pressure equals the specified pressure. This is accomplished by first assuming the shape of the jet boundary and then using the wall routine to calculate the pressure. Next, the jet boundary location is changed slightly and a second pressure is computed. The secant method determines a new jet boundary location. This procedure is then repeated at each grid point until the jet boundary pressure and the ambient pressure agree within some specified tolerance.

When a free-jet calculation is made, the wall exit lip grid point becomes a singularity, so it is treated by a special procedure. First, an upstream solution is computed at the exit grid point, using the flow tangency condition as the boundary condition and backward ζ differences in both the initial-value and solution planes. Next, a downstream solution is calculated, using the specified pressure as the boundary condition and the stagnation conditions calculated from the upstream grid point. The upstream solution is used in computing wall grid points upstream of the exit grid point and the downstream solution in computing downstream free-jet grid points. A third exit grid point solution for interior grid point calculation is determined as follows. When the upstream solution is subsonic, the two solution Mach numbers are averaged to be less than or equal to one. This Mach number, along with the upstream stagnation temperature and pressure, is then used to calculate the exit grid point solution for computing the interior grid points. When the upstream solution is supersonic, it is used to calculate the interior grid points.

c. No-Slip Boundary. Unlike the VNAP code, VNAP2 uses the characteristic scheme to enforce the no-slip boundary condition. The boundary condition is the vanishing of the velocity components and either the vanishing of the temperature gradient normal to the boundary (adiabatic wall) or the specification of the temperature. The pressure is calculated from Eq. (62) with the normal temperature gradient set equal to zero and the density from Eq. (61). If the vanishing of the normal temperature gradient option is desired, then the temperature can be determined from the equation of state. If the specified wall temperature option is desired, then the pressure is recomputed from the equation of state.

The boundary conditions for the turbulence models are the vanishing of q for the one-equation model and the vanishing of q and the specification of e so that $\partial e/\partial y = 0$ for the Jones-Launder two-equation model.

d. Constant Pressure Inflow/Outflow Boundary. The constant pressure inflow/outflow boundary grid points are also calculated using the characteristic scheme. The pressure is always specified. If the flow across the boundary is outflow, then u and v are calculated from Eqs. (60) and (62), and ρ is calculated from Eq. (61). For inflow, u and ρ are specified and v is calculated from Eq. (62). The actual values of u and ρ specified are the values at the grid point where the left boundary intersects the wall. The temperature is determined from the equation of state. A nonreflecting boundary condition option, similar to that used at the right boundary, is employed here.

The turbulence model boundary conditions are the extrapolation of q for the one-equation model and q and e for the Jones-Launder two-equation model.

5. **Centerbody Grid Points.** The centerbody boundary can be a free-slip boundary, a no-slip boundary, or a plane (axis) of symmetry. The free-slip and no-slip boundary calculations follow the wall procedure. The characteristic relation that couples the interior flow to the boundary is derived following the procedure of Ref. 1 and can be written as

$$dp - \rho\alpha a du/\alpha^* - \rho\beta a dv/\alpha^* = (\psi_4 + a^2\psi_1 - \rho\alpha a\psi_2/\alpha^* - \rho\beta a\psi_3/\alpha^*)d\tau \tag{64}$$

for $d\eta = (\bar{v} - \alpha^* a)d\tau$.

Equation (63) becomes

$$v = u \tan\theta \ . \tag{65}$$

The time derivative in Eq. (63) does not appear in Eq. (65) because the centerbody coordinates are not a function of time.

For flows where the centerbody is a plane (axis) of symmetry, the centerbody grid points are computed by the interior point scheme. The boundary condition is flow symmetry.

The turbulence model boundary conditions are the same as the wall boundary for the free-slip and no-slip cases. For the plane (axis) of symmetry case, q and e are specified so that $\partial q/\partial y = \partial e/\partial y = 0$.

6. **Dual-Flow-Space Wall Grid Points.** The dual-flow-space walls can be either a free-slip or a no-slip boundary. The calculations follow the wall and centerbody procedures. The centerbody equations are used for the upper dual-flow space, and the wall equations, with Eq. (65) replacing Eq. (63), are used for the lower dual-flow-space wall. The turbulence model boundary conditions are the same as the wall and centerbody boundaries.

7. **Step Size.** The step size Δt is determined by

$$\Delta t = \min(\Delta t_x, \Delta t_y) \ , \tag{66}$$

where

$$\Delta t_x = A/[(|u| + a)/\Delta x + \mu/A_1\rho\Delta x^2] \tag{67}$$

and

$$\Delta t_y = A/[(|v| + a)/\Delta y + \mu/A_1\rho\Delta y^2] \ , \tag{68}$$

where A and A_1 are constants that usually equal 0.9 and 0.25, respectively. For the Quick Solver option, Eq. (68) becomes

$$\Delta t_y = A/(|v|/\Delta y + \mu/A_1\rho\Delta y^2) \ . \tag{69}$$

These conditions are checked at each grid point in the flow field at each time step. However, these conditions are not checked on the subcycled time steps.

F. Comments on the Calculation of Steady, Subsonic Flows

Because signals propagate in all directions in subsonic flows, disturbances can reflect inside the computational grid for many time steps and can significantly prolong the convergence to steady state.

However, in supersonic flows, signals only propagate downstream and are, therefore, swept out of the grid. As a result, supersonic flows generally converge to steady state in fewer time steps than subsonic flows. As an example, consider the following two inviscid accelerating flows: planar subsonic sink flow and planar supersonic source flow. The comptational regions for the subsonic sink and supersonic source flows are enclosed by the dashed lines in Figs. 4 and 5, respectively. The top dashed line is treated as a free-slip wall, the bottom dashed line is the flow midplane, and the left and right dashed lines are inflow and outflow boundaries, respectively. The outflow midplane Mach number for the subsonic case is 0.5, and the inflow midplane Mach number for the supersonic case is 1.5. The boundary conditions for the subsonic flow are the specification of p_T, T_T, and θ at the inlet and p at the exit. For the supersonic flow, all inlet variables are specified and all outlet variables are extrapolated. The initial-data surface for both flows is the 1D solution generated by the VNAP2 code. Figure 6 shows the pressure vs number of time steps for both flows. The top curve for both flows gives the solution at an interior grid point near the inflow boundary, and the lower line is a grid point near the outflow boundary. The supersonic flow reaches steady state in around 150 time steps, whereas the subsonic case requires approximately 1200. For very complex flows, this difference is often greater. Therefore, the following discussion will be concerned with improving the convergence to steady state of subsonic flows.

Figure 7 shows the pressure vs number of time steps for the subsonic sink flow employing different techniques to accelerate the convergence to steady state. Again, the p_T, T_T, and θ inflow boundary condition is used. The grid point plotted in Fig. 7 is the one near the inlet in Fig. 6. The top curve is for a calculation that started from an initial-data surface consisting of a stationary flow at the stagnation pressure and temperature. At time equal to zero, the pressure at the outflow boundary was dropped from the stagnation value to the sink flow exact solution, thus simulating a bursting diaphragm. The other four calculations started with an initial-data surface generated by the VNAP2 code, which is the 1D solution. The third line from the top shows the solution using the Rudy and Strikwerda[19] nonreflecting outflow boundary condition. The coefficient C_α (ALE in Namelist BC) in Eq. (54) equals 0.1. (Namelists are given in Sec. II.C.) The fourth curve from the top shows the solution for which all the dependent variables were smoothed in space for the first 500 time steps. This calculation multiplies the value at a grid point by a weighting parameter and adds it to the average of the values of its nearest neighboring grid points multiplied by one minus the weighting parameter. The weighting parameter was 0.5 for the first time step and linearly increased to 1.0 (no smoothing) by the 500th time step (SMP = 0.5, SMPF = 1.0, and NST = 500 in Namelist AVL). The bottom curve used the extended-interval time-smoothing option, which stores the solution for all dependent variables on the first time step and then monitors the pressure at a specified grid point on each time step. When this pressure changes direction, the solution at the current time step is averaged with the solution at the first time step. This averaged solution replaces the current time-step solution and, in addition, is stored in place of the first time-step solution. This process is continued for the entire computation (SMPT = 0.5, SMPTF = 0.5, NTST = 0, and NST = NMAX in Namelist AVL). The diaphragm initial-data surface solution requires around 1800 time steps to reach steady state, whereas the 1D initial-data surface solution is steady in approximately 1100 times. The nonreflecting and space-smoothing options further increase the convergence to steady state. However, the largest increase is due to the time smoothing, which results in a converged solution in about 400 time steps. The increased convergence rate of the time-smoothed solution over the other options is more pronounced for more complex flows.

Figure 8 shows the pressure vs the number of time steps for the u, v, and ρ inflow boundary condition. The diaphragm initial-data surface solution produced results similar to the 1D curve and, therefore, is not shown. The top three curves correspond to the same options in Fig. 7. The bottom curve is the solution using the nonreflecting inflow option discussed in Sec. I.E.2. The coefficient C_β (ALI in Namelist BC) in Eq. (56) equals 0.1. The top curve of Fig. 8 shows that the u, v, and ρ boundary condition trapped the initial disturbances in the computational grid. The Rudy and Strikwerda[19] nonreflecting outflow boundary condition option (not shown) did not significantly improve this result. Note that the Rudy and Strikwerda boundary condition is used in conjunction with the reference-plane-characteristic scheme, which is somewhat different from the numerical procedure they used. Their procedure may produce different results. As Fig. 8 shows, the space- and time-smoothing options, as well as the nonreflecting inflow boundary condition option, all produce steady solutions.

The 1D solution, which is used as the initial-data surface, has an outflow Mach number of 0.55. The sink flow exact solution has midplane and upper wall outflow Mach numbers of 0.5 and 0.42, respectively. The high 1D solution Mach number was chosen so that the 1D solution would not approximate the 2D sink flow solution too closely. However, this high Mach number produces a 12% difference in mass flow between the 1D solution and the 2D sink flow solution. Because the u, v, and ρ inflow boundary condition specifies the 2D sink flow solution mass flow, an expansion wave is produced at the inlet. This expansion wave causes the large drop in pressure, shown in Fig. 8, during the early stages of the calculation. Adjusting the Mach number of the 1D solution so that the 1D mass flow closely approximates the mass flow specified by the u, v, and ρ boundary condition yields the results shown in Fig. 9 where, except for the top curve, the convergence to steady state is greatly improved.

From the above and other similar results, some general conclusions can be drawn. First of all, for steady, subsonic flows the p_T, T_T, and θ inflow boundary condition is preferred over the u, v, and ρ boundary condition. For subsonic computations that require long run times, the extended-interval time smoothing can significantly reduce computational time. For subsonic/supersonic nozzle flows, the p_T, T_T, and θ inflow boundary condition is also preferred, because the mass flow is usually not known in advance. If the u, v, and ρ boundary condition is used for steady, subsonic flows, then either the nonreflecting inflow option of space or extended-interval time smoothing should be used. The u, v, and ρ inflow boundary condition is useful for unsteady subsonic flows where the user wishes to specify the mass flow. VNAP2 allows only constant values of u, v, and ρ to be specified; however, the code could easily be modified to allow time-dependent functions for u, v, and ρ. The u, v, and ρ inflow boundary condition also works well for the subsonic part of the boundary layer in a supersonic flow. In many cases, this subsonic part of the boundary can be treated with supersonic boundary conditions. However, where this practice gives poor results, the u, v, and ρ boundary condition is an improvement. The test cases run to date indicate that the u, v, and ρ boundary condition produces results more consistent with the supersonic part of the flow than does the p_T, T_T, and θ boundary condition. As a result, VNAP2 allows only the u, v, and ρ option at subsonic parts of a mixed subsonic/supersonic inflow.

The p, v, and ρ boundary condition has received little use to date because, in general, it should be used with either the u specified subsonic outflow boundary condition or supersonic outflow. When p is specified as the subsonic outflow boundary condition, some flows are not uniquely defined. For example, if p, v, and ρ are specified at the inflow and p is specified at the outflow for inviscid subsonic flow in a constant area duct, the Mach number would not be uniquely specified. The specified u outflow boundary condition is not incorporated (as originally intended) because there is little use for it. The p, v, and ρ boundary condition can be used for subsonic/supersonic nozzle flows because it does not specify the mass flow; however, the p_T, T_T, and θ boundary condition is preferred.

In general, the closer the initial-data surface is to the final solution, the faster the solution converges to the steady state. This is also true for viscous flows, where using initial data that approximate all boundary and free-shear layers generally reduces the run time.

Finally, Moretti and I disagree[20-22] on the u, v, and ρ subsonic inflow boundary condition. Moretti feels that the u, v, and ρ boundary condition is incorrect for a well-posed problem, because disturbances reflected by this boundary condition may remain trapped in the finite-difference grid. Reference 22 lists several published proofs of the correctness of this boundary condition. As a result of these proofs, I feel that this boundary condition is mathematically correct for a well-posed problem and that the trapping of disturbances is a numerical problem that can be overcome. In addition to these mathematical proofs, the u, v, and ρ boundary condition satisfies the characteristic compatibility conditions, as does the p_T, T_T, and θ boundary condition. Both boundary conditions falsify the time-dependent flow by holding quantities fixed that actually vary in time (p_T and T_T are constant only for steady flow). As a result, both cause nonphysical reflections at subsonic boundaries. The u, v, and ρ boundary condition causes a reflection that has approximately the same amplitude, whereas the p_T, T_T, and θ boundary condition produces a highly damped reflection. These reflection properties differ because they model different upstream conditions—constant mass flow as opposed to constant total pressure—which makes them suitable for different problems. In Ref. 23, Moretti seems to imply that the u, v, and ρ boundary condition requires knowledge of the exact solution. Although I specified the exact solution in Ref. 20, as did Moretti in Ref. 23, the exact solution values of u, v, and ρ or p_T, T_T, and θ are generally not known in advance. (For the special case of inviscid, steady flow, p_T and T_T, but not θ, are usually known.) Therefore, one specifies his best guess boundary values. The computed solution will satisfy these boundary values as well as the governing equations, and its accuracy will depend on how well these boundary values were estimated. Therefore, I feel that both boundary conditions are correct and that the best choice is problem-dependent.

A second point that concerns this section is Moretti's claim[23] that the initial-data surface and boundary conditions must be matched so that the transient part of a steady state calculation follows the true transient solution. Although this is the most correct way to formulate problems, it is generally not the most economical. It is true that there are flows where following the true transient solution is very desirable. One such case is the startup of a supersonic wind tunnel. If, for example, the area of the throat downstream of the test section is not large enough to pass the startup shock, then the shock will stand in the test section or nozzle. Beginning a time-dependent calculation of this flow with a purely supersonic initial-data surface will produce the started, all supersonic, steady solution, even though this solution is physically impossible. Beginning this calculation with a 1D subsonic initial-data surface would yield the right solution. However, use of Moretti's recommendation[23] of the diaphragm initial-data surface, discussed above, provides the right solution without requiring any knowledge of the starting of a supersonic wind tunnel. Thus, there are flows where either hysteresis effects or lack of understanding suggest following Moretti's recommendation. However, for steady, subsonic flows this recommendation can be very expensive. In addition, I have never found a subsonic flow calculation using a time-dependent method where the steady solution depended on the initial-data surface (except for small differences from truncation errors and provided the initial-data surface is subsonic). As a result, I feel that the special procedures discussed above for accelerating the convergence of subsonic flows to their steady state may be used to reduce these lengthy computational times. I have included these last two paragraphs to warn the users of VNAP2 that some of the ideas expressed above are my own and may not be universally accepted as correct procedures.

G. Results and Discussion

Presented here are three relatively simple flows that are intended to illustrate the three general classes of flows that can be computed with VNAP2: internal, external, and internal/external flows. The data files for these three cases are included at the back of the Fortran listing of the VNAP2 code in the Appendix. The initial-data surfaces for the external and internal/external cases assume solution array sizes of 41 by 25. For the application of VNAP2 to more complex flows, see Ref. 24.

1. **Internal, Inviscid Flow.** The first case is steady, subsonic/supersonic, inviscid flow in the 45-15° conical, converging-diverging nozzle shown in Fig. 10 with the flow from left to right. This calculation is also presented in Refs. 1, 2, and 25. The upper boundary is a free-slip wall and the lower boundary is the centerline. The left boundary is a subsonic inflow boundary using the p_T, T_T, and θ boundary condition.

The right boundary is a supersonic outflow boundary and, therefore, the variables are extrapolated. The Mach number contours and wall pressure ratio are shown in Fig. 11. The experimental data are those of Cuffel et al.[26] The computed discharge coefficient is 0.983, compared with the experimental value of 0.985. The 21 by 8 uniform computational grid requires 299 time planes and a computation time of 35 s on the CDC 6600 and 6 s on the CDC 7600.

Although the Mach number, wall pressure, and throat mass flow results are in good agreement with experiment, the mass flow variation at different axial locations is fairly poor. For example, the mass flow variation between the inlet and throat is 4.5%. If the grid spacing is halved by using a 41 by 15 uniform grid, the mass flow variation between the inlet and throat is 1.4%. Halving the grid spacing again, by using an 81 by 29 uniform grid, produces a mass flow variation between the inlet and throat of 0.1%. Therefore, the mass flow variation appears to go to zero as the grid spacing goes to zero. Some of the error in the coarse grid case may be due to the trapezoidal rule used to evaluate the mass flow integral. However, much of the error is probably due to the large truncation error of the finite-difference equations, owing to the steep gradients in the nozzle throat region. The variation in throat mass flow between the 81 by 29 and 21 by 8 grid cases is 0.25%, whereas between the 81 by 29 and 41 by 15 grid cases it is 0.06%. Therefore, the throat mass flow is fairly good for coarse grid spacings even though the overall mass flow conservation is fairly poor.

This case uses the convergence tolerance option to determine when the steady state has been reached. That is, when the relative change in axial velocity in the throat and downstream regions is less than 0.003%, the flow is assumed to have reached steady state. In general, I have not found this convergence tolerance option to be very useful, because the value of the convergence tolerance depends on the grid spacing and flow conditions and as such is usually not known in advance. One exception to this is the case involving a large parametric study. Here, once the convergence tolerance has been determined by trial and error, it can be used repeatedly in the remaining runs of the parametric study. However, a procedure based on the time of flight of an average fluid particle seems to work more consistently. In this procedure, one sets the total number of time steps so that an average fluid particle will travel through the computational grid a particular number of times. The velocity of an average fluid particle can be estimated from the 1D solution or some other initial-data surface. This average velocity can also be estimated from the numerical solution itself by running the program for a fairly short time and using that solution to estimate the average fluid particle velocity. Use of the restart option allows this run to be contiued to steady state. The time step can be obtained by running the code for one time step (two for viscous flows). Once the average fluid particle velocity and time step have been determined, then the number of time steps required for one trip can be calculated. The last piece of required information is the number of trips made by the average fluid particle through the grid to reach steady state. For supersonic, inviscid flows, three trips are usually sufficient, whereas supersonic, viscous flows require around five. Converging-diverging, supersonic, inviscid nozzle flows usually require around five trips, whereas viscous nozzle flows need around seven. The numbers of trips given above are only rough estimates and should be supplemented by the user's own experiences. In addition, when in doubt as to how many time steps are necessary, always use the restart option.

Finally, for subsonic flows, neither the convergence tolerance nor the time of flight procedure is really effective. The most effective method that I have found is to monitor the static pressure at several spots in the flow (see LPP1, MPP1 in Namelist CNTRL). Provided that an average fluid particle has made at least one trip, then the flow can be assumed to be steady when the pressure is oscillating with an acceptable amplitude about a constant value. Looking at only the amplitude of the oscillation, without regard to whether it occurs about a constant value, is sometimes not sufficient.

2. External, Turbulent Flow. The second case is steady, subsonic, turbulent flow over a boattail afterbody with a solid body simulating the jet exhaust. The geometry is shown in Fig. 12, with the dashed line enclosing the computational region, and the flow is from left to right. This calculation is also presented in Ref. 24. The upper boundary is a constant pressure inflow/outflow boundary and the lower boundary is a no-slip wall. The left boundary is a subsonic inflow boundary using the p_T, T_T, and θ boundary condition. The values of p_T and T_T are determined from an inviscid/boundary-layer solution procedure for the forebody. The right boundary is a subsonic outflow boundary and, therefore, the static pressure is specified. The free-stream Mach number is 0.8 and the Reynolds number, based on the length at the inflow boundary, is 10.5×10^6. For more details on the geometry or experimental data, see Ref. 27. The turbulence is modeled using the mixing-length model. This calculation employed the subcycling, Quick Solver, and extended-interval time-smoothing options. Figure 13 shows the physical space grid, pressure, and Mach number contours. Figure 14 shows the surface pressure coefficient on the boattail and jet exhaust simulator. Figures 13 and 14 show that the boundary layer remains attached. For cases with separation and exhaust jets, see Ref. 24. This calculation employs a 40 by 25 variable grid that requires 750 time steps (15 000 subcycled time steps in the boundary layer) and a computation time of 1 h on the CDC 7600. Swanson[28] compared several different formulations of the mixing-length model for computing this case as well as separated cases.

3. **Internal/External, Turbulent Flow.** The third case is steady, subsonic, turbulent flow for a plane jet in a uniform stream. The geometry is shown in Fig. 15 with the dashed line enclosing the computational region, and the flow is from left to right. The upper boundary is a constant pressure inflow/outflow boundary and the lower boundary is the midplane. The dual-flow-space boundaries are no-slip walls. The left boundary is a subsonic inflow boundary using the u, v, and ρ boundary condition, with the nonreflecting option. The right boundary is a subsonic outflow boundary and, therefore, the static pressure is specified. The jet and external stream have initial Mach numbers of 0.14 and 0.02, respectively, while the Reynolds number, based on the jet height, is 3.0×10^4. The turbulence is modeled using the mixing-length and Jones-Launder two-equation models. This case, assuming free-slip inflow profiles and a solid free-slip upper boundary and employing the mixing-length turbulence model, was presented in Ref. 1. The physical space grid and Mach number contours for the mixing-length model are shown in Fig. 16. Figure 17 shows the midplane velocity decay for both turbulence models. The subscript JE denotes the midplane velocity just downstream of the end of the dual-flow-space walls. The increase in the velocity is due to the acceleration of the mean flow caused by the growth of the boundary layer. The experimental data are from Ref. 29. This calculation employs a 41 by 17 variable grid that requires 6000 time steps and a computation time of 24 min (mixing-length model) on the CDC 7600.

This rather lengthy run time, even though a fairly coarse grid spacing was used, is because the flow is almost incompressible. That is, the flow velocity is much smaller than the sound speed. The explicit numerical scheme is limited to time steps so that sound waves travel less than one mesh spacing. (The problem geometry did not allow the use of the Quick Solver option, although some reduction in run time could be made using the subcycle option.) Therefore, many time steps are required before a particle of fluid travels from the inflow to the outflow boundary.

II. DESCRIPTION AND USE OF THE VNAP2 PROGRAM

A. Subroutine Description

The computer program consists of 1 program, 1 function, and 18 subroutines. A complete Fortran listing of the VNAP2 program is included in the Appendix.

1. **Program VNAP2.** Program VNAP2 initiates a run by reading in the input data. Next, the program title, abstract, and input data descriptions are printed. The program then calls subroutines GEØM, GEØMCB, and GEØMLU to calculate the geometry. If requested, program VNAP2 calls

subroutine ØNEDIM to calculate the 1D, initial-value surface. Program VNAP2 then prints the initial-value surface, which includes a mass flow and momentum thrust calculation made by subroutine MASFLØ. Next, subroutine PLØT is called to plot the data on film. The final part of VNAP2 consists of the time-step loop, which calculates the next time-step size; calls subroutine VISCØUS to calculate the artificial, molecular, and turbulent viscosity-heat conduction terms; calls subroutine QSØLVE to calculate the special derivatives used by the Quick Solver package; calls subroutine INTER to compute the interior mesh points; calls subroutine WALL to compute the wall, centerbody, and dual-flow-space wall mesh points; calls subroutine INLET to compute the inlet mesh points; calls subroutine EXITT to compute the exit mesh points; calls subroutine TURBC to set the boundary conditions for the turbulence variables; if requested, calls subroutine SMØØTH to smooth the solution; calls subroutine MASFLØ to compute the mass flow and momentum thrust; prints the solution surface; calls subroutine PLØT to plot the data on film; checks the solution for its convergence to the steady-state solution; and punches (writes) the last solution plane on cards (disc or tape) for restart.

2. **Subroutine GEØM.** Subroutine GEØM calculates the wall coordinates and slopes for four different wall geometries: a constant area duct wall; a circular-arc, conical wall; and two tabular input walls. In the case of the first tabular wall, a completely general set of wall coordinates is read in. Subroutine GEØM then calls subroutine MTLUP, which interpolates for the coordinates. Next, subroutine GEØM calls function DIF, which calculates the slopes of the coordinates. For the second tabular wall, the coordinates and slopes are read in.

3. **Subroutine GEØMCB.** Subroutine GEØMCB calculates the centerbody coordinates and slopes for four different centerbody geometries and is similar to subroutine GEØM.

4. **Subroutine GEØMLU.** Subroutine GEØMLU calculates the upper and lower dual-flow-space wall coordinates and slopes for two tabular input geometries. These tabular cases are the same as those in subroutine GEØM.

5. **Subroutine MTLUP.** Subroutine MTLUP (September 12, 1969) was taken from the National Aeronautics and Space Administration (NASA) Langley program library. This subroutine is called by subroutines GEØM, GEØMCB, and GEØMLU to interpolate the wall, centerbody, and dual-flow-space wall coordinates.

6. **Function DIF.** Function DIF (August 1, 1968) was also taken from the NASA Langley program library. This function is called by subroutines GEØM, GEØMCB, and GEØMLU to calculate the slopes of the wall, centerbody, and dual-flow-space wall coordinates.

7. **Subroutine ØNEDIM.** Subroutine ØNEDIM is called by program VNAP2 to compute the 1D, isentropic initial-value surface. A Newton-Raphson scheme calculates the Mach number for the area ratios, which are determined from the geometry.

8. **Subroutine MAP.** Subroutine MAP calculates the functions that map the physical plane to a rectangular computational plane. Therefore, this subroutine is called before each mesh point is calculated.

9. **Subroutine MASFLØ.** Subroutine MASFLØ is called by program VNAP2 to calculate the mass flow and momentum thrust for the initial-value and solution surfaces. The trapezoidal rule evaluates the mass flow and momentum thrust integrals.

10. **Subroutine PLØT.** Subroutine PLØT is called by program VNAP2 to produce velocity vector plots, the physical space grid, and contour plots of density, pressure, temperature, Mach number,

turbulence energy, and dissipation rate, using the SC-4020 microfilm recorder. The SC-4020 recorder uses a 1022 by 1022 array of plotting points or coordinates on each film frame. The origin is the upper left corner of the array. The coordinates to be plotted by the SC-4020 recorder are assumed to be integer constants. The first section sets up the plot size by setting the maximum left (XXL), right (XR), top (YT), and bottom (YB) coordinates in the physical space. Then the film frame coordinates and scaling factors are determined with the plot beginning at 900, instead of 1022, to allow for labeling.

The next section generates the velocity vector plot. First, the maximum velocity is determined to scale the plot, which is done so that the maximum velocity vector is 0.9 Δx, where Δx is the average value. Subroutine ADV (Los Alamos system routine) advances the film one frame. Then the velocity vector is calculated in fixed point film frame coordinates. Subroutine DRV (Los Alamos system routine) draws a line between the points (IX1, IY1) and (IX2, IY2), after which subroutine PLT (Los Alamos system routine) plots a plus sign at the point (IX1, IY1). Subroutine LINCNT (Los Alamos system routine) skips down 58 lines. (Each printed line height equals 16 film frame points.) The routine then returns to set up the plot size for the next velocity vector plot if IVPTS > 1, or goes on to the next section if IVPTS $\leqslant 1$.

The next section resets the plot size for the contour plots in case the different scaled velocity vector plots were requested (IVPTS > 1).

The next section fills the plotting array called CQ with the following variables: density (lbm/ft^3 or kg/m^3), pressure (psia or kPa), temperature (°R or K), and Mach number.

The next section determines the plotting line quantities using the formula

$$CQ_K = CQ_{MIN} + 0.1K(CQ_{MAX} - CQ_{MIN}) \ ,$$

where K goes from one to nine. This section also labels the frames.

The next section determines the location of each contour line segment and plots it. The contour line segment defined by the film frame coordinates (IX1, IY1) and (IX2, IY2) is drawn by subroutine DRV. Subroutine PLT plots an L on the low contour (K=1) and an H on the high contour (K=9).

The last section draws the geometry boundaries for the contour plots. The upper boundary is specified by YW, the lower by YCB, the upper dual-flow-space boundary by YU, and the lower dual-flow-space boundary by YL. Next, the routine returns to the section that fills the plotting array CQ for the next contour plot.

11. **Subroutine SWITCH.** Subroutine SWITCH switches the solution values from the solution array to the dummy array when dual-flow-space boundaries are present. The dummy array is required because the two dual-flow-space walls collapse to one grid line in the computational plane.

12. **Subroutine VISCØUS.** Subroutine VISCØUS calculates the artificial viscosity terms for shock computations using a velocity gradient viscosity coefficient. It also calculates the molecular viscosity terms in the Navier-Stokes equations. In addition, this subroutine calculates the various turbulence terms in the Navier-Stokes equations, as well as the turbulence energy and dissipation rate equations.

13. **Subroutine SMØØTH.** Subroutine SMØØTH is called by program VNAP2 to add either space or time numerical smoothing to stabilize the calculations for nonuniform initial-data surfaces or to accelerate the convergence to steady state. The physically correct molecular viscous terms (with a large viscosity coefficient) could also be used; however, they are much slower and cannot be reduced or turned off during a run.

14. **Subroutine MIXLEN.** Subroutine MIXLEN is called by subroutine VISCØUS to calculate the shear layer width or the boundary layer thickness and kinematic displacement thickness for the mixing-length model (ITM = 1). These parameters also determine the length scale used by the turbulence energy model (ITM = 2).

15. Subroutine TURBC. Subroutine TURBC is called by program VNAP2 to set the boundary conditions for the turbulence energy, Q, and the dissipation rate, E.

16. Subroutine INTER. Subroutine INTER is called by program VNAP2 to calculate the interior mesh points. The conservation of mass, momenta, internal energy, turbulence energy, and dissipation rate equations are solved by the MacCormack second-order, finite-difference scheme. Subroutine INTER also contains part of the Quick Solver package. Special values of the derivatives u_η, v_η, and p_η, calculated by subroutine QSØLVE, are used in special forms of the governing equations to allow an increased time step.

17. Subroutine WALL. Subroutine WALL is called by program VNAP2 to compute the wall, centerbody, dual-flow-space walls, free-jet boundary, and sharp expansion corner mesh points. This subroutine uses a second-order, reference-plane-characteristic scheme and also controls the interpolation process for locating the free-jet boundary. Subroutine WALL also contains part of the Quick Solver package that allows an increased time step. However, this subroutine does not use the special derivatives calculated by subroutine QSØLVE.

18. Subroutine INLET. Subroutine INLET is called by program VNAP2 to compute the inlet mesh points. If the flow is subsonic, a second-order, reference-plane-characteristic scheme is employed, whereas specification of the boundary conditions is used for supersonic flow. This subroutine also checks the Mach number to determine which boundary condition should be used at each mesh point. In addition, subroutine INLET contains part of the Quick Solver package and uses the special derivatives calculated by subroutine QSØLVE.

19. Subroutine EXITT. Subroutine EXITT is called by program VNAP2 to calculate the exit mesh points. It uses a second-order, reference-plane-characteristic scheme when the flow is subsonic and extrapolation when the flow is supersonic. This subroutine also checks the Mach number to determine which boundary condition should be used at each mesh point. In addition, subroutine EXITT contains part of the Quick Solver package and uses the special derivatives calculated by subroutine QSØLVE.

20. Subroutine QSØLVE. Subroutine QSØLVE, which is part of the Quick Solver package, calculates the partial derivatives u_η, v_η, and p_η that are used in subroutines INTER, INLET, and EXITT. These special derivatives are calculated from the domain of dependence defined by the characteristics through the solution point and, therefore, allow an increased time step.

B. Computational Grid Description

The computational grid for the single-flow-space example is shown in Fig. 18. The grid is rectangular with equal spacing in the ζ and η directions, although $\Delta\zeta$ and $\Delta\eta$ are not in general equal. The grid spacing $(\Delta x, \Delta y)$ in the physical space does not have to be equal.

The dual-flow-space grid (Fig. 19) is the same as the single-flow-space grid except for an extra row of grid points (M = MDFS and L between LDFSS and LDFSF). The solution values at these extra grid points are stored in arrays UL, VL, PL, RØL, QL, and EL. During the calculation, subroutine SWITCH exchanges these values continually with the values in the solution arrays U, V, P, RØ, Q, and E for M = MDFS and L between LDFSS and LDFSF. For reading in initial-data values, the values in UL, VL, PL, RØL, QL, and EL arrays correpond to the lower dual-flow-space wall, whereas values in the U, V, P, RØ, Q, and E arrays for M = MDFS and L between LDFSS and LDFSF correspond to the upper dual-flow-space wall.

The computational grid for the subcycled grid option is shown in Fig. 20. The code advances the solution one time step in the large spacing grid points (from M = 1 to MVCB − 1 and from M = MVCT

+ 1 to MMAX) and then subcycles the small spacing grid points (from M = MVCB to MVCT). In this way, the small time step requirement of the small spacing grid points (small spacing in the physical plane) is not forced on the large spacing grid points.

The flow is assumed to enter from the left and exit on the right. In addition, flow may enter or exit the wall (see IWALL in Namelist BC).

C. Input Data Description

The program input data are entered by a title card and 10 namelists: CNTRL, IVS, GEMTRY, GCBL, BC, AVL, RVL, TURBL, DFSL, and VCL. The title card and each namelist are described below. The program will continue reading in data decks and executing them until a file mark is encountered. After each data deck is executed, the default values for the input data are restored before the next data deck is read in.

1. Title Card. The first card of each data deck is a title card consisting of 80 alphanumeric characters that identify the job. This card must always be the first card of the data deck, even if no information is specified on the card. The 10 namelists must appear in the order in which they are discussed below.

2. Namelist CNTRL. This namelist reads in the parameters that control the overall logic of the program.

LMAX	An integer specifying the number of mesh points in the x direction with a maximum value specified by a PARAMETER statement (see Sec. II.E.1). No default value is specified.
MMAX	An integer specifying the number of mesh points in the y direction with a maximum value specified by a PARAMETER statement (see Sec. II.E.1). No default value is specified.
NMAX	An integer specifying the maximum number of time steps. For NMAX = 0, only the initial-data surface is computed and printed (provided NPRINT > 0). The default value is 0.
NPRINT	An integer specifying the amount of output desired. For NPRINT = N, every Nth solution plane, plus the initial-data and final solution planes, is printed. For NPRINT = −N, every Nth solution plane, plus the final solution plane, is printed. For NPRINT = 0, only the final solution plane is printed. The default value is 0.
TCONV	Specifies the axial velocity steady-state convergence tolerance in percentage. If equal to zero, the convergence is not checked. This parameter is a function of the problem as well as of grid spacing and, therefore, should be used carefully. The default value is 0.0.
FDT	The parameter A in Eqs. (67)-(69) that premultiplies the allowable C-F-L time step. It is desirable to use as large a value of FDT as possible without causing the computation to become unstable. Values as large as 1.3 have been used successfully for shock-free flows, but smaller values are required for flows with shocks (see Sec. II.F). The defalut value is 0.9.
FDTI	The same as FDT, except it applies on the first time step only. Because the viscous contribution to the time-step limitation is not used on the first time step, FDTI may be used to get the calculation started with a small time step, without having to use this small value for the entire calculation. Some flows may require a small time step for the first few steps owing to initial gradients in the flow variables. This is often

true for viscous flows when the Quick Solver option is used. For this situation, make a short run with small enough values of FDT or FDT1 so that the code will run. Then use the restart option (see IPUNCH) to continue the run with more desirable values of FDT or FDT1. For any long running problem, it is usually worth experimenting with FDT and FDT1 (as well as VDT and VDT1) to make sure that optimum values are being used. The default value is FDT.

FDT1 — The same as FDT, except it applies only in the subcycled part of the mesh. That is, FDT1 is used from M = MVCB to M = MVCT (see Namelist VCL). The default value is 1.0.

VDT — The parameter A_1 in Eqs. (67)-(69) that premultiplies the viscous part of the time-step equation, whereas FDT premultiplies the entire time step. Increasing VDT increases the time step. The default value is 0.25.

VDT1 — The same as VDT, except it applies only in the subcycled part of the mesh. That is, VDT1 is used from M = MVCB to M = MVCT (see Namelist VCL). The default value is 0.25, although values larger than 1.0 have been used in free-shear layers.

GAMMA — Denotes the ratio of specific heats. The default value is 1.4.

RGAS — Denotes the gas constant in lbf-ft/lbm—°R if English units are used, or J/kg—K if metric units are used. The default value is 53.35.

TSTØP — Specifies the physical time, in seconds, at which the computations will be stopped. The default value is 1.0.

IUI — An integer specifying the type of units to be used for the input quantities. IF IUI = 1, English units are assumed; if IUI = 2, metric units are assumed. In using any default values, make sure the values correspond to the proper units. The default value is 1.

IUØ — The same as IUI except for output quantities. IF IUØ = 3, both English and metric units are printed. The default value is 1.

IPUNCH — An integer which, if nonzero, punches (writes) the last solution plane on cards (disc or tape) for restart. The default value is 0.

NPLOT — An integer which, if greater than or equal to zero, plots both velocity vectors and contours of density, pressure, temperature, Mach number, turbulence energy, and dissipation rate on an SC-4020 microfilm recorder. For NPLOT = N, all Nth solution planes, plus the initial-data and final solution plane, are plotted. For NPLOT = 0, only the final solution plane is plotted. The default value is −1.

LPP1,MPP1
LPP2,MPP2
LPP3,MPP3 — Three sets of integers that specify three grid points (the first point is L = LPP1, M = MPP1) for which the pressure is printed at each time step. When MPP1(MPP2 or MPP3) = MDFS $\neq$ 0 (Namelist DFSL), the upper dual-flow-space wall value is printed. This pressure history is very useful for determining when subsonic flows have reached steady state. If LPP1 < 0, the pressure at each subcycled grid point (see MVCB and MVCT in Namelist VCL) is also printed. The default values are 0 (no printing).

The remaining parameters in Namelist CNTRL are less important than the parameters given above. For most flows, these remaining parameters can be left at their default values.

NASM — An integer specifying which part of the flow field is tested for steady-state convergence. For NASM = 0, the entire flow field is tested. For NASM = 1, the transonic and supersonic (throat region to exit) regions are tested. The default value is 1.

NAME — An integer that, when nonzero, causes the 10 namelists to be printed in addition to the regular output. The default value is 0.

NCØNVI — An integer specifying how many times the convergence tolerance TCØNV must be satisfied on consecutive time steps before the solution is considered to have converged. The default value is 1.

IUNIT — An integer that, when equal to zero, causes the program to use either English or metric units (see IUI and IUØ). For IUNIT = 1, a nondimensional set of units is used. The default value is 0.

PLØW — If the pressure becomes negative during a calculation, it is set equal to PLØW in psia or kPa. The default value is 0.01.

RØLØW — If the density becomes negative during a calculation, it is set equal to RØLØW in lb/ft^3 or kg/m^3. The default value is 0.0001.

IVPTS — An integer that controls the scaling of the velocity vector plots. IVPTS = 1 produces onc plot with thc maximum vcctor cqual to 0.9 Δx, whcrc Δx is the averagc value. IVPTS = 2 produces the above plot and a second plot where the maximum vector is 1.9 Δx, and so on. The default value is 1.

3. **Namelist IVS.** This namelist specifies the flow variable for the initial-data surface.

N1D — An integer specifying the type of initial-data surface desired. For N1D = 0, a 2D initial-data surface is read in. A value of U, V, P, and RØ (discussed below) must be read in for all mesh points from L = 1 to LMAX and from M = 1 to MMAX. In addition, for dual-flow-space examples, values of UL, VL, PL, and RØL (discussed below) must be read in for all mesh points from L = LDFSS to LDFSF. For the single-equation turbulence model, a value of Q, along with QL for the dual-flow-space example, may be read in. For the two-equation model, a value of E, along with EL for the dual-flow-space example, may also be read in. If the arrays Q and QL and the arrays E and EL are not read in, they are set equal to FSQ and FSE (Namelist TURBL), respectively. Values of Q and E may be read in for either N1D = 0 or N1D ≠ 0. For N1D ≠ 0, a 1D data surface is computed internally.

The following combinations are possible:

N1D = −2 subsonic N1D = −1 supersonic	see RSTAR and RSTARS
N1D = 1 subsonic-sonic-supersonic N1D = 2 subsonic-sonic-subsonic N1D = 3 supersonic-sonic-supersonic N1D = 4 supersonic-sonic-subsonic	No additional data are needed.

The default value is 1.

U(L,M,1) — An array denoting the x-direction velocity component in ft/s or m/s. For N1D = 0, U(L,M,1) must be read in for cases from L = 1 to LMAX and from M = 1 to MMAX. For N1D ≠ 0, U(L,M,1) is not read in. No default values are specified.

V(L,M,1) — An array denoting the y-direction velocity component in ft/s or m/s. See U(L,M,1) for additional information. No default values are specified.

P(L,M,1) — An array denoting the pressure in psia or kPa. See U(L,M,1) for additional information. No default values are specified.

RØ(L,M,1) — An array denoting the density in lbm/ft^3 or kg/m^3. See U(L,M,1) for additional information. No default values are specified.

Q(L,M,1) — An array denoting the turbulence energy in ft^2/s^2 or m^2/s^2. See U(L,M,1) for additional information. The default value is FSQ(M) in Namelist TURBL.

E(L,M,1) — An array denoting the dissipation ratc in ft^2/s^3 or m^2/s^3. Scc U(L,M,1) for additional information. The default value is FSE(M) in Namelist TURBL.

UL(L,1) — An array denoting the x-direction velocity component in ft/s or m/s and corresponding to the lower dual-flow-space wall. The values for the upper dual-flow-space wall are read in by U(L,MDFS,1). For N1D = 0 and MDFS ≠ 0, UL(L,1) must be read in for cases from L = LDFSS to LDFSF. For N1D ≠ 0 or MDFS = 0, UL(L,1) is not read in. No default values are specified.

VL(L,1) — An array denoting the y-direction velocity component in ft/s or m/s. See UL(L,1) for additional information. No default values are specified.

PL(L,1) — An array denoting the pressure in psia or kPa. See UL(L,1) for additional information. No default values are specified.

RØL(L,1) — An array denoting the density in lbm/ft^3 or kg/m^3. See UL(L,1) for additional information. No default values are specified.

QL(L,1) — An array denoting the turbulence energy in ft^2/s^2 or m^2/s^2. See UL(L,1) for additional information. The default value is FSQL in Namelist TURBL.

EL(L,1) — An array denoting the dissipation rate in ft^2/s^3 or m^2/s^3. See UL(L,1) for additional information. The default value is FSEL in Namelist TURBL.

RSTAR, RSTARS — If N1D = −1 or −2, either RSTAR for planar or RSTARS for axisymmetric flow must be read in. RSTAR is the area per unit depth or height (in in. or cm) where the Mach number is unity. RSTARS is the area divided by π that is the radius squared (in $in.^2$ or cm^2) where the Mach number is unity. No default values are specified.

If the restart option is to be used, the initial run must be made with IPUNCH ≠ 0 in CNTRL, thereby causing a new IVS Namelist deck to be punched or written on disc or tape. The new IVS Namelist replaces the one used initially and includes two additional parameters, NSTART and TSTART, which denote, respectively, the time step and the physical time where the solution was restarted.

When N1D ≠ 0, the initial data are calculated using 1D isentropic theory. However, the x and y velocity components are adjusted while the magnitude is kept constant and the flow angle is satisfied. The flow angles are linearly interpolated between the slope of the wall and the centerbody. For the dual-flow-space example, the Mach number is assumed to be equal in both flow spaces at a given value of x. However, the flow angles are interpolated between the centerbody and the lower dual-flow-space boundary for the lower space and between the upper dual-flow-space boundary and the wall for the upper space.

4. **Namelist GEMTRY.** This namelist specifies the parameters that define the wall contour.

NDIM — An integer denoting the flow geometry. For NDIM = 0, 2D planar flow is assumed, and for NDIM = 1, axisymmetric flow is assumed. The default value is 1.

NGEØM — An integer specifying one of four different wall geometries. A discussion of these four cases follows the definitions of the additional parameters in this namelist. No default value is specified.

XI — The x coordinate, in in. or cm, of the wall inlet. No default value is specified.

RI — The y coordinate, in in. or cm, of the wall inlet. No default value is specified.

RT — The y coordinate, in in. or cm, of the wall throat. No default value is specified.

XE — The x coordinate, in in. or cm, of the wall or free-jet exit. No default value is specified.

RCI — The radius of curvature, in in. or cm, of the wall inlet. No default value is specified.

RCT — The radius of curvature, in in. or cm, of the wall throat. No default value is specified.

ANGI — The angle, in degrees, of the converging section. No default value is specified.

ANGE — The angle, in degrees, of the diverging section. No default value is specified.

XWI — A 1D array of nonequally spaced x coordinates in in. or cm. No default values are specified.

YWI — A 1D array of y coordinates, in in. or cm, corresponding to the x coordinates in array XWI. No default values are specified.

NWPTS — An integer specifying the number of entries in arrays XWI and YWI. The maximum value is specified by a PARAMETER statement (see Sec. II.E.1). No default value is specified.

IINT — An integer specifying the order of interpolation used. The maximum value is 2. The default value is 2.

IDIF — An integer specifying the order of differentiation used. The maximum value is 5. The default value is 2.

YW — A 1D array of y coordinates, in in. or cm, which correspond to LMAX x coordinates, given by XP in Namelist VCL. No default values are specified.

NXNY — A 1D array (floating point) of the negative of the wall slopes corresponding to the elements of YW. No default values are specified.

JFLAG — An integer that, when equal to 1, denotes that a free-jet calculation is to be carried out and, when equal to −1, denotes that a supersonic sharp expansion corner is present on the wall. These two options are allowed only for the free-slip wall boundary condition. Many free-jet flows contain shocks and will, therefore, require artificial viscosity (see Namelist AVL). The default value is 0 (no free jet and no sharp expansion corner).

LJET — An integer that, when JFLAG = 1, denotes the first mesh point of the free-jet boundary (the last wall mesh point is LJET − 1). However, when JFLAG = −1, LJET is the next mesh point downstream of the sharp expansion corner (the corner mesh point is LJET − 1). The program assumes that either the wall ends exactly at LJET − 1 (JFLAG = 1) or the sharp expansion corner is located exactly at LJET − 1) (JFLAG = −1). Also, for the sharp expansion corner case (JFLAG = −1), the slope of the wall at the corner (LJET − 1) should be the upstream value. The program does not allow both a sharp expansion corner and a free-jet calculation. In addition LJET must be > 2 and < LMAX − 1. No default value is given.

The following is a discussion of the four different wall geometries considered by this program.

a. Constant Area Duct (NGEØM = 1). The parameters XI,RI (radius of the duct) and XE must be specified.

b. Circular-Arc, Conical Wall (NGEØM = 2). The geometry for this case is shown in Fig. 21. The parameters XI, RI, RT, XE, RCI, RCT, ANGI, and ANGE are specified. The x coordinate of the throat and the radius of the exit are computed internally.

c. General Wall (NGEØM = 3). An arbitrary wall contour is specified by tabular input. NWPTS x- and y-coordinate pairs are specified by the arrays XWI and YWI, respectively. The tabular data need not be equally spaced. From the specified values of NWPTS, XWI, YWI, IINT, and IDIF, the program uses IINT-order interpolation to obtain LMAX y coordinates that correspond to the x coordinates given by XP in Namelist VCL. Next, IDIF-order differentiation is used to obtain the wall slope at these LMAX points.

d. General Wall (NGEØM = 4). An arbitrary wall contour is specified by tabular input. LMAX y coordinates and the negative of their slopes are specified by the arrays YW and NXNY, respectively. These y coordinates correspond to the LMAX x coordinates given by XP in Namelist VCL. XI and XE also must be read in.

5. Namelist GCBL. This namelist specifies the parameters that define the centerbody geometry. If no centerbody is present, this namelist is left blank but must still be present in the data deck.

NGCB — An integer that, when nonzero, specifies one of four different centerbody geometries. A discussion of these four cases will follow the definitions of the additional parameters in this namelist. The default value is 0.

RICB — The y coordinate, in in. or cm, of the centerbody inlet. No default value is specified.

RTCB — The y coordinate, in in. or cm, of the centerbody maximum radius. No default value is specified.

RCICB	The radius of curvature, in in. or cm, of the centerbody inlet. No default value is specified.
RCTCB	The radius of curvature, in in. or cm, of the centerbody maximum radius. No default value is specified.
ANGICB	The angle, in degrees, of the converging section. No default value is specified.
ANGECB	The angle, in degrees, of the diverging section. No default value is specified.
XCBI	A 1D array of nonequally spaced x coordinates in in. or cm. No default values are specified.
YCBI	A 1D array of y coordinates, in in. or cm, corresponding to the x coordinates in array XCBI. No default values are specified.
NCBPTS	An integer specifying the number of entries in arrays XCBI and YCBI. The maximum value is specified by a PARAMETER statement (see Sec. II.E.1). No default value is specified.
IINTCB	An integer specifying the order of interpolation used. The maximum value is 2. The default value is 2.
IDIFCB	An integer specifying the order of differentiation used. The maximum value is 5. The default value is 2.
YCB	A 1D array of y coordinates, in in. or cm, which correspond to LMAX x coordinates given by XP in Namelist VCL. The default values are 0.0.
NXNYCB	The 1D array (floating point) of the negative of the centerbody slopes corresponding to the elements of YCB. The default values are 0.0.

The following is a discussion of the four different centerbody geometries considered by this program.

a. Cylindrical Centerbody (NGCB = 2). The parameter RICB (radius of the centerbody) must be specified.

b. Circular-Arc, Conical Centerbody (NGCB = 2). The geometry for this case is shown in Fig. 22. The parameters RICB, RTCB, RCICB, RCTCB, ANGICB, and ANGECB are specified. The x coordinate of the maximum radius and the radius of the exit are computed internally.

c. General Centerbody (NGCB = 3). An arbitrary centerbody contour is specified by tabular input. NCBPTS x- and y-coordinate pairs are specified by the arrays XCBI and YCBI, respectively. The tabular data need not be equally spaced. From the specified values of NCBPTS, XCBI, YCBI, IINTCB, and IDIFCB, the program uses IINTCB-order interpolation to obtain LMAX y coordinates that correspond to the x coordinates given by XP in Namelist VCL. Next, IDIFCB-order differentiation is used to obtain the centerbody slope at these LMAX points.

d. General Centerbody (NGCB = 4). An arbitrary centerbody contour is specified by tabular input. LMAX y coordinates and the negatives of their slopes are specified by the arrays YCB and NXNYCB, respectively. These y coordinates correspond to the LMAX x coordinates given by XP in Namelist VCL.

6. Namelist BC. This namelist specifies the flow boundary conditions for all computational boundaries.

NSTAG	An integer that, when nonzero, denotes that variable total pressure PT, variable total temperature TT, and variable flow angle THETA (all discussed below) have been specified. If NSTAG $\neq$ 0, then a value for PT, TT, and THETA must be specified at all the points from M = 1 to MMAX, even if one or two of the variables are constant or some grid points are not used (ISUPER = 2 or 3). If NSTAG = 0, only the first value for each of the three arrays needs to be specified. The default value is 0.
PT(M)	A 1D array denoting the stagnation pressure, in psia or kPa, across the inlet (see ISUPER). This array is used to calculate the 1D initial-data surface as well as the inflow conditions for ISUPER = 0, 2, or 3. No default values are specified.

TT(M)	A 1D array denoting the stagnation temperature, in °R or K, across the inlet (see ISUPER). This array is used to calculate the 1D initial-data surface as well as the inflow conditions for ISUPER = 0, 2, or 3. No default values are specified.
THETA(M)	A 1D array denoting the flow angle, in degrees, across the inlet (see ISUPER). The default value is THETA(1) = 0.0, which is meaningful only when NSTAG = 0.
PTL	Denotes the stagnation pressure, in psia or kPa, at the point where the lower dual-flow-space wall intersects the inlet (see Namelist DFSL). The upper dual-flow-space wall value is read in by PT(MDFS). If NSTAG = 0 or MDFS = 0 or LDFSS $\neq$ 1, then PTL is not read in. No default value is specified.
TTL	The same as PTL, except denotes the stagnation temperature in °R or K.
THETAL	The same as PTL, except denotes the flow angle is degrees.
PE(M)	A 1D array denoting the pressure, in psia or kPa, to which the flow is exiting. This pressure is used to compute the flow exit conditions when the flow is subsonic, the free-jet boundary location when a free-jet calculation is requested, or the wall inflow-outflow boundary when IWALL = 1. The free-jet or wall inflow/outflow boundary pressure is assumed to be constant and equal to PE(MMAX). Subroutine WALL could be modified to allow PE to be a function of x or t. This array starts with the centerline or centerbody value and ends with the wall value. If the exit pressure is constant, only the first value of the array needs to be read in. The default value is 14.7.
PEL	Denotes the pressure, in psia or kPa, to which the flow is exiting at the point where the lower dual-flow-space wall intersects the exit (see Namelist DFSL). The upper dual-flow-space wall value is read in by PE(MDFS). If MDFS = 0 or LDFSF $\neq$ LMAX, PEL is not read in. No default value is specified.
UI(M)	A 1D array denoting the x velocity, in ft/s or m/s, across the inlet (see ISUPER). This array, as well as the arrays VI, PI, and RØI below, starts with the centerline or centerbody value and ends with the wall value. Values must be specified for points from M = 1 to MMAX even if some grid points are not used (ISUPER = 2 or 3). No default values are specified.
VI(M)	The same as UI, except y velocity.
PI(M)	The same as UI, except denotes pressure in psia or kPa.
RØI(M)	The same as UI, except denotes density in lbm/ft^3 or kg/m^3.
UIL	Denotes the x velocity in ft/s or m/s at the point where the lower dual-flow-space wall intersects the inlet (see Namelist DFSL). The upper dual-flow-space wall value is read in by UI(MDFS). For MDFS = 0 or LDFSS $\neq$ 1, UIL is not read in. See ISUPER for additional information. No default value is specified.
VIL	The same as UIL, except y velocity.
PIL	The same as UIL, except denotes pressure in psia or kPa.
RØIL	The same as UIL, except denotes density in lbm/ft^3 or kg/m^3.
TW	A 1D array denoting the wall temperature in °R or K corresponding to the x mesh points. If TW is not specified, the wall is assumed to be adiabatic.
TCB	The same as TW, except denotes centerbody temperature.
TL	The same as TW, except denotes lower dual-flow-space wall (see Namelist DFSL). If MDFS = 0, TL is not read in.
TU	The same as TW, except denotes upper dual-flow-space wall (see Namelist DFSL). If MDFS = 0, TU is not read in.
ISUPER	An integer that specifies whether the inlet flow is subsonic, supersonic, or both. ISUPER may have the following values:

ISUPER = 0 — Subsonic inflow with PT, TT, and THETA as the specified quantities.

ISUPER = 1 — Subsonic, supersonic, or mixed inflow with UI, VI, PI, and RØI as the specified quantities. For subsonic flow, PI is only an initial guess if INBC = 0, and UI is only an initial guess if INBC $\neq$ 0.

ISUPER = 2 — Subsonic, supersonic, or mixed inflow between the centerbody and lower dual-flow-space wall with UI, VI, PI, and RØI as the specified quantities. For subsonic flow, PI is only an initial guess if INBC = 0, and UI is only an initial guess if INBC $\neq$ 0. ISUPER = 2 is subsonic inflow between the upper dual-flow-space wall and the wall with PT, TT, and THETA as the specified quantities.

ISUPER = 3 — The same as ISUPER = 2, except subsonic and subsonic, supersonic or mixed sides are switched.

The default value is 0.

INBC — An integer that specifies whether u or p will be the inflow boundary condition for ISUPER $\neq$ 0. If INBC = 0, u is the boundary condition and p is calculated. If INBC $\neq$ 0, the reverse is true. The default value is 0.

IWALL — An integer that denotes whether the wall is a solid boundary (includes free-jet option) or a constant pressure inflow/outflow boundary that is fixed with respect to time.

IWALL = 0 — Specifies a solid or free-jet boundary.

IWALL = 1 — Specifies a constant pressure [PE(MMAX)] boundary. When there is inflow across this constant pressure boundary, u and ρ are set equal to the wall-inlet value. This option cannot be used with JFLAG $\neq$ 0 in Namelist GEMTRY. The default value is 0.

IWALLØ — An integer that, when not equal to 0, forces linear extrapolation of the pressure at the wall for the IWALL = 1 case. This option is useful when a shock wave exits the wall boundary or when the flow normal to the boundary is supersonic outflow. The default value is 0.

IINLET — An integer that, when not equal to 0, forces specification of all variables as the inflow boundary condition regardless of the Mach number. It applies only when ISUPER $\neq$ 0. The default value is 0.

IEXITT — An integer that, when not equal to 0, forces either extrapolation (IEXITT = 1) or specified pressure (IEXITT = 2) as the outflow boundary condition regardless of the Mach number. The default value is 0.

IEX — An integer that denotes the type of extrapolation to be used for supersonic outflow. IEX = 0 denotes zeroth-order extrapolation, and IEX = 1 denotes linear extrapolation. The default value is 1.

IVBC — An integer that specifies whether extrapolation or reflection is used to determine the viscous terms at boundaries. IVBC = 0 specifies reflection, IVBC = 1 specifies linear extrapolation, and IVBC = 2 specifies zeroth-order extrapolation. Reflection is always used at the centerline or midplane. The adiabatic wall boundary condition (that is, TW, TCB, TL, and TU not specified) requires IVBC = 0. The default value is 0.

NØSLIP — An integer that, when equal to zero, specifies free-slip walls whereas NØSLIP = 1 specifies no-slip (u = v = 0) walls for all solid boundaries. The no-slip boundary condition is not enforced at the wall when IWALL $\neq$ 0. The default value is 0.

DYW	A parameter that specifies the maximum change that is allowed on each time step in the free-jet boundary location. The default value is 0.001, that is, 0.1% maximum change per time step.
IAS	An integer that, if not equal to zero, causes the upper and lower dual-flow-space wall slopes to be set equal to the average of the two slopes. This occurs only at the point or points where the two dual-flow-space walls intersect. That is, for LDFSS $\neq$ 1, the slopes at LDFSS will be set equal to their average. Also, if LDFSF $\neq$ LMAX, the same occurs. The default value is 0.
ALI	The coefficient C_α in Eqs. (55) and (56). This coefficient controls the nonreflecting inflow boundary condition employed at the left boundary. Any nonzero value will activate the nonreflecting option; however, values of approximately 0.1 appear to work well for many problems. Specifying ALI $\neq$ 0.0 for the P_T, T_T, and θ boundary condition or supersonic inflow has no effect. The default value is 0.0.
ALE	The coefficient C_α in Eq. (54). This coefficient controls the nonreflecting inflow and outflow boundary condition at the right boundary. See ALI for further details. Specifying ALE $\neq$ 0.0 for supersonic outflow has no effect. The default value is 0.0.
ALW	The coefficient C_α in Eq. (54). This coefficient controls the nonreflecting inflow and outflow boundary condition at the wall boundary. See ALI for further details. Specifying ALW $\neq$ 0 when IWALL = 0 (Namelist BC) has no effect. The default value is 0.0.

7. **Namelist AVL.** This namelist specifies the parameters that determine the artificial viscosity used to stabilize the calculations for shocks and control the space- and time-smoothing options. For flows without shocks or where space or time smoothing is not desired, this namelist is left blank. See Sec. II.F for additional information.

CAV	Denotes the artificial viscosity premultiplier C in Eq. (23). See Sec. II.F for typical values. The default value is 0.0.
XMU	Denotes the coefficient $C_{\mu 1}$ in Eq. (24) in the artificial viscosity model. A nondimensional value is used. The default value is 0.4.
XLA	Denotes the coefficient C_λ in Eq. (23) in the aritificial viscosity model. A nondimensional value is used. The default value is 1.0.
PRA	Denotes the coefficient Pr_A in Eq. (25) in the artificial viscosity model and represents an artificial Prandtl number. The default value is 0.7.
XRØ	Denotes the coefficient C_ρ in Eq. (26) in the artificial viscosity model. The default value is 0.6.
LSS, LSF	Integers that specify the x mesh points at which the addition of the artificial viscosity will begin (LSS) and end (LSF). These parameters can significantly reduce the run time for inviscid flows where a shock occupies only a small part of the flow. The default values are LSS = 1 and LSF = 999.
MSS, MSF	The same as LSS and LSF, except that these specify the y mesh points at which the addition of the artificial viscosity begins (MSS) and ends (MSF). The default values are MSS = 1 and MSF = 999.
IDIVC	An integer that, when not equal to 0, bypasses the check on the sign of the velocity divergence in the artificial viscosity model. That is, the artificial viscosity will be nonzero for both expansions and compressions. This improves some complex multiple shock interactions, but also increases the smearing of expansions. The default value is 0.
ISS	An integer that, when not equal to 0, adds the sound speed gradient to the velocity divergence in Eq. (23). For ISS = 1, the sound speed gradient is added to the

velocity divergence only if the velocity divergence is <0. For ISS = 2, the sound speed gradient is always added. This term improves contact surface calculations (see Sec. I.F). The default value is 0.

SMACH — Denotes the Mach number below which no artificial viscosity for shock calculations is added to the solution. This option is useful for moderate-to-high Reynolds number, steady flow, where the artificial viscosity swamps the molecular and turbulent viscosities in the boundary layer. By setting SMACH equal to ~0.5, the artificial viscosity is zero for most of the subsonic part of the boundary layer. See Sec. I.F for additional details. The default value is 0.0.

NST — An integer denoting the time step at which a small amount of numerical space or time smoothing is stopped. Smoothing is employed on the regular time steps and not the subcycled steps (see Namelist VCL). This smoothing may be required to stabilize the calculations for very nonuniform or impulsively started initial-data surfaces. Some initial smoothing in space causes subsonic flows to reach steady state faster, but this is not the case for transonic and supersonic flows. Time smoothing also causes subsonic flows to converge to steady state faster. When using the restart option, make sure NST is set equal to zero unless additional smoothing is desired. If additional smoothing is desired on a restart, make sure that the values of SMP or SMPT on the restart equal the final values of the previous run (see SMP and SMPT discussion below). The default value is 0 (no smoothing).

SMP — A parameter that, along with NST and SMPF, controls the amount of space smoothing (provided NST $\neq$ 0). SMP must be between 0.0 and 1.0. The dependent variables are smoothed by the following formula: $u_{L,M} = SMP*u_{L,M} + (1.0 - SMP)*(u_{L+1,M} + u_{L,M+1} + u_{L-1,M} + u_{L,M-1})/4.0$. The value of SMP changes on each time step by the following replacement formula:
SMP = SMP + (SMPF − $\underline{SMP}$)/NST,
where the underlined SMP denotes the original input value. The inlet (L = 1) and exit (L = LMAX) columns of grid points are not smoothed. The default value is 1.0.

SMPF — A parameter that, along with NST and SMP, controls the amount of space smoothing (see SMP for details). SMPF must be between 0.0 and 1.0. The default value is 1.0.

SMPT — A parameter that, along with NST and SMPTF, controls the amount of time smoothing or relaxation (provided NST $\neq$ 0). The dependent variables are smoothed by the following formula:

$$u^{N+1}_{L,M} = SMPT*u^{N+1}_{L,M} + (1.0 - SMPT)*u^{N}_{L,M} .$$

The value of SMPT changes on each time step by the following replacement formula:
SMPT = SMPT + (SMPTF − $\underline{SMPT}$)/NST,
where the underlined SMPT denotes the original input value. Where some initial space smoothing followed by longer duration time smoothing is desired, flows can be computed using the restart option. The default value is 1.0.

SMPTF — A parameter that, along with NST and SMPT, controls the amount of time smoothing (see SMPT for details). The default value is 1.0.

NTST — An integer that specifies the interval of time steps over which the solution is time smoothed (provided NST $\neq$ 0 and SMPT $\neq$ 1.0). For example, if NTST = 10, then after every 10 time steps the solution at the current time step N is time averaged with the solution at time step N − 10. This averaged solution is then stored and used to average with the solution at N + 10. For NTST = 0, the code monitors the pressure at the L = LPP1 and M = MPP1 grid point (Namelist CNTRL) and time smooths when this pressure changes direction. If LPP1 and MPP1 are not specified and NTST = 0, there is no time smoothing. This extended-interval time smoothing usually improves the convergence to steady state of subsonic flows. To use this option with NTST = 0 or >1, the arrays US, VS, PS, RØS, QS, and ES must be dimensioned for LMAX and MMAX, while arrays ULS, VLS, PLS, RØLS, QLS, and ELS must be dimensioned for LMAX. These arrays are located in Common AV. The default value is 1.

IAV — An integer that, when not equal to 0, causes the viscous-turbulence terms, turbulence energy, and dissipation rate (or length scale) to be printed at the solution planes specified by NPRINT. IAV = 2 causes the viscous terms for each subcycled time step to be printed (provided MVCB and MVCT in Namelist VCL are nonzero). The default value is 0.

8. **Namelist RVL.** This namelist specifies the real or molecular viscosity parameters. For inviscid flows, this namelist is left blank.

CMU, EMU — These parameters specify the molecular viscosity μ by the following equation:

$$\mu = CMU \cdot T^{EMU} ,$$

where T is the temperature in °R or K. The units of μ are lbf-s/ft^2 or Pa-s. The default values are 0.0.

CLA, ELA — These parameters specify the second coefficient of viscosity λ by the following equations:

$$\lambda = CLA \cdot T^{ELA} ,$$

where T is the temperature in °R or K. The units of λ are lbf-s/ft^2 or Pa-s. The default values are 0.0.

CK, EK — These parameters specify the thermal conductivity k by the following equation:

$$k = CK \cdot T^{EK} ,$$

where T is temperature in °R or K. The units of k are lbf/s-°R or W/m-K. The default values are 0.0.

9. **Namelist TURBL.** This namelist specifies the turbulence model parameters. For laminar as well as inviscid flows, it is left blank. For turbulent flows, Namelist RVL cannot be blank.

ITM — An integer that, when nonzero, specifies one of three different turbulence models. ITM = 1 specifies a mixing-length model; ITM = 2 specifies a one-equation, turbulence energy model; and ITM = 3 specifies a two-equation, turbulence energy-dissipation-rate model. The default value is 0.

IMLM — An integer, required for ITM = 1 or 2, that specifies whether the flow is a free shear layer (IMLM = 1) or a boundary-layer flow (IMLM = 2). This information is required because the equations for the mixing length (ITM = 1) and the length scale of the one-equation model (ITM = 2) are different depending on whether the flow is a free shear or boundary layer. For single-flow spaces, the shear-layer option assumes either that the boundaries are free slip or that the lower boundary is a symmetry boundary and the wall must be a constant pressure inflow/outflow boundary. The boundary-layer option assumes one no-slip boundary, which is either a centerbody or a wall, but not both. For dual-flow spaces (see Namelist DFSL), the dual-flow-space walls are assumed to be no-slip boundaries, but the lower boundary must be a symmetry boundary and the wall must be a constant pressure inflow/outflow boundary. The program then uses the boundary-layer option between the dual-flow-space walls and the shear-layer option elsewhere, regardless of IMLM. Therefore, for dual-flow spaces IMLM does not need to be specified. The default value is 1.

CML1, CML2 — These coefficients, defined in Eqs. (9) and (10) and required for ITM = 1 or 2, are used in the shear-layer option (for IMLM = 1 or for dual-flow spaces). The mixing length, used in both ITM = 1 and 2, is calculated by multiplying the shear-layer thickness by these coefficients. CML2 is for velocity profiles where the minimum velocity is in the flow interior, and CML1 is for monotonic profiles. The default values for both coefficients are 0.125 for planar flows and 0.11 for axisymmetric flow.

CAL — Denotes the coefficient $\overline{\alpha}$ in the governing equations, Eqs. (1)-(4). This coefficient controls the effect of variable density for all three turbulence models. The recommended and default value is 1.0.

CQL — This coefficient, which is C_q in Eq. (15) and required by ITM = 2, is multiplied by the mixing length to obtain the length scale used in the one-equation model. The default value is 17.2 for planar flows and 12.3 for axisymmetric flow.

CQMU — This coefficient, which is C_u in Eqs. (17) and (21) and required by ITM = 2 or 3, premultiplies the expression for the turbulent viscosity in the one- and two-equation models. The recommended and default value is 0.09.

C1,C2, SIGQ,SIGE — Coefficients, which are C_1,$\bar{C}_2$, σ_q, and σ_e, respectively, in Eq. (20) and required by ITM = 3, for the two-equation, turbulence energy-dissipation-rate model. The recommended and default values are 1.44, 1.8, 1.0, and 1.3, respectively.

BFST — A parameter, required by ITM = 3, that sets a lower bound for q and e in the two-equation model by the following relation:

$$q_{L,M} \geqslant BFST*FSQ(M)$$
$$e_{L,M} \geqslant BFST*FSE(M)\ ,$$

where FSQ and FSE are defined below. A value between 0.0 and 1.0 is necessary for some separated flows. If MDFS $\neq$ 0 and L < LDFSS or L > LDFSF (Namelist DFSL), then BFST is set to zero. The default value is 0.0.

FSQ(M) — A 1D array that denotes the inlet or free-stream turbulence energy level (ITM = 2 or 3) in ft^2/s^2 or m^2/s^2. This array, as well as the array FSE, starts with the centerline or centerbody value and ends with the wall value. The default value is 0.0001.

FSE(M) — The same as FSQ, except that the dissipation rate level (ITM = 3) is given in ft^2/s^3 or m^2/s^3. The default value is 0.1.

FSQL — Denotes the inlet or free-stream turbulence energy level (ITM = 2 or 3) in ft^2/s^2 or m^2/s^2 at the point where the lower dual-flow-space wall intersects the inlet (see Namelist DFSL). The upper dual-flow-space wall is read in by FSQ(MDFS). For MDFS = 0 or LDFSS $\neq$ 1, FSQL is not read in. The default value is 0.0001.

FSEL — The same as FSQL, except that the dissipation rate level (ITM = 3) is given in ft^2/s^3 or m^2/s^3. The default value is 0.1.

QLØW — If during a calculation the turbulence energy (ITM = 2 or 3) becomes less than or equal to QLØW, it is set equal to QLØW. The default value is 0.0001.

ELØW The same as QLØW except for the dissipation rate (ITM = 3). The default value is 0.1.

LPRINT, MPRINT Integers that, when greater than zero, cause the convection, production, dissipation, and diffusion terms of the turbulence energy (ITM = 2 or 3) and dissipation rate (ITM = 3) to be printed for L = LPRINT, M = MPRINT at every time step. The axisymmetric terms are not included. The default value is 0.

PRT Denotes the turbulent Prandtl number in Eq. (8). The turbulent viscosity μ_T is calculated by the turbulence model, after which the turbulent conductivity k_T is calculated from PRT. The default value is 0.9.

STBQ, STBE Denote the coefficients C_Q and C_E, respectively, in Eq. (22). These coefficients control the fourth-order smoothing for the two-equation model (ITM = 3). This smoothing may improve the results for strongly separated flows. The default values are 0.0 (no smoothing).

10. **Namelist DFSL.** This namelist specifies the dual-flow-space walls. For single-flow-space examples, this namelist is left blank.

MDFS An integer that, when nonzero, specifies the M row of grid points along which the dual-flow-space walls are positioned. MDFS cannot be set equal to 2 or MMAX − 1. The default value is 0.

LDFSS, LDFSF Integers that specify the x grid points where the dual-flow-space walls start and end, respectively. LDFSS and LDFSF cannot be set equal to 2 or LMAX − 1, respectively. The default values are 0.

NDFS An integer specifying one of two different dual-flow-space wall geometries. A discussion of these two cases follows the definitions of the additional parameters in this namelist. No default value is specified.

YU, YL 1D arrays of y coordinates in in. or cm, which correspond to the LMAX x coordinates given by XP in Namelist VCL. YU denotes the upper dual-flow-space wall and YL denotes the lower. The default values are 0.0.

NXNYU, NXNYL 1D arrays (floating point) of the negative of the dual-flow-space wall slopes corresponding to the elements of YU and YL, respectively. The default values are 0.0.

XUI, XLI 1D arrays of nonequally spaced x coordinates in in. or cm. XUI corresponds to the upper dual-flow-space wall and XLI corresponds to the lower. No default values are specified.

YUI, YLI 1D arrays of y coordinates in in. or cm, corresponding to the x coordinates in arrays XUI and XLI, respectively. No default values are specified.

NUPTS, NLPTS Integers specifying the number of entries in arrays XUI-YUI and XLI-YLI, respectively. The maximum value is specified by a PARAMETER statement (see Sec. II.E.1). No default values are specified.

IINTDFS An integer specifying the order of interpolation used. The maximum value is 2. The default value is 2.

IDIFDFS An integer specifying the order of differentiation used. The maximum value is 5. The default value is 2.

The following is a discussion of the two different dual-flow-space wall geometries considered by this program. If the dual-flow-space walls begin in the interior (LDFSS ≠ 1), the values of YL and YU (or YLI and YUI) for L = LDFSS must be equal. The same is true at L = LDFSF if the dual-flow-space walls end in the interior (LDFSF ≠ LMAX). If the dual-flow-space walls begin and end in the interior, than the ratio (YL − YCB)/(YW − YCB) at L = LDFSS must equal that at L = LDFSF. The angle of attack of the dual-flow-space walls can be varied somewhat by changing the shape of the centerbody and wall. However, if the centerbody and wall shapes are fixed, then the angle of attack cannot be varied.

a. General Dual-Flow-Space Wall (NDFS = 1). An arbitrary dual-flow-space wall contour is specified by tabular input. NUPTS x and y coordinate pairs are specified by the arrays XUI and YUI, respectively. NLPTS x and y coordinate pairs are specified by the arrays XLI and YLI, respectively. The tabular data need not be equally spaced. From the specified values of NUPTS, XUI, YUI, NLPTS, XLI, YLI, IINTDFS, and IDIFDFS, the program uses IINTDFS-order interpolation to obtain (LDFSF − LDFSS + 1) upper and lower dual-flow-space wall y coordinates that correspond to the (LDFSF − LDFSS + 1) x coordinates given by XP(LDFSS) to XP(LDFSF) in Namelist VCL. Next, IDIFDFS-order differentiation is used to obtain the upper and lower dual-flow-space wall slopes at these (LDFSF − LDFSS + 1) points.

b. General Dual-Flow-Space Wall (NDFS = 2). An arbitrary wall contour is specified by tabular input. (LDFSF − LDFSS + 1) y coordinates and the negative of their slopes are specified by the arrays YU and NXNYU for the upper dual-flow-space wall and YL and NXNYL for the lower, respectively. The y coordinates correspond to the (LDFSF − LDFSS + 1) x coordinates given by XP(LDFSS) to XP(LDFSF) in Namelist VCL.

11. **Namelist VCL.** This namelist specifies the variable grid coordinates as well as the parameters that control the subcycle and Quick Solver options. For equal or uniform grid spacing, this namelist is left blank.

The subcycle option allows the part of the mesh with the small grid spacing to be computed for many time steps with the required small time step, whereas the rest of the mesh is calculated only one time step. The Quick Solver option can be used with the subcycle option to increase the time step in the small grid part of the mesh and, therefore, reduce the number of time steps or subcycles. The Quick Solver allows the increased time step by a procedure that removes the sound speed from the usual C-F-L stability condition. The Quick Solver assumes the flow in the y direction is subsonic.

IST — An integer that, when nonzero, specifies that both the x and y coordinates will have variable grid spacings. When IST = 0, the program will generate equally spaced values of XP and YI. The default value is 0.

XP — A 1D array that denotes the x coordinate grid spacing. The elements of XP begin with the inlet (L = 1) and extend to the outlet (L = LMAX). The first element XP(1) must equal XI [or XWI(1)] of Namelist GEMTRY and XP(LMAX) must equal XE [or XWI(NWPTS)]. For IST = 0, the default values of XP consist of LMAX equally spaced grid points. For IST $\neq$ 0, no default values are given.

YI — A 1D array that specifies the y coordinate grid spacing at the inlet or x = XP(1) column of grid points. The elements of YI begin with the centerline or centerbody and extend to the wall. If MDFS $\neq$ 0 and LDFSS = 1 (Namelist DFSL), then YI(MDFS) must equal YU(1) and a value of YI = YL(1) is not read in. The grid spacing for the columns corresponding to x = XP(2), XP(3), . . . , XP(LMAX) is proportional to the YI spacings. For IST = 0, the default values of YI consist of MMAX equally spaced grid points. For IST $\neq$ 0, no default values are given.

MVCB, MVCT — Integers that, when nonzero, denote which grid points will be subcycled. The subcycled grid points are M = MVCB to MVCT for all L. The restrictions are MVCB $\neq$ 2, MVCT $\neq$ MMAX − 1, and MVCT > MVCB + 1. Where dual-flow-space walls are present, MVCB $\neq$ MDFS + 1 and MVCT $\neq$ MDFS − 1. Finally, if the subcycled grid points extend on each side of the dual-flow-space walls, MVCB < MDFS − 1 and MVCT > MDFS + 1. The default values are 0.

NVCMI — An integer that, when nonzero, specifies the number of times the small spacing grid points are subcycled. If NVCMI = 0, the program determines the value internally. NVCMI must be an odd integer for indexing reasons. See NIQSS and NIQSF for additional details. The default value is 0.

IQS	An integer that, when nonzero, specifies the Quick Solver option. This option assumes that the flow in the y direction is subsonic. Also, if MVCT = MMAX, then the wall boundary must be a no-slip solid wall (IWALL = 0 and NØSLIP = 1 in Namelist BC). If MVCB = 1, then the centerbody boundary must be a no-slip solid wall (NGCB = 1 in Namelist GCBL and NØSLIP = 1). If dual-flow-space walls are present (see Namelist DFSL), the Quick Solver assumes that the subcycled grid points extend on each side of the dual-flow-space walls; that is, MVCB < MDFS < MVCT. The default value is 0.
NIQSS, NIQSF	Integers that, when nonzero, denote at which time step N the Quick Solver will start (NIQSS) and stop (NIQSF). If NIQSS > 1 and NVCMI is nonzero, then the program internally calculates the number of times to subcycle the small spacing grid points for N < NIQSS and uses NVCMI when N ⩾ NIQSS. The default values are NIQSS = 2 and NIQSF = NMAX in Namelist CNTRL.
CQS	A parameter that specifies the convergence tolerance for the iteration that locates the characteristic intersection points in the Quick Solver. The default value is 0.001.
ILLQS	An integer that specifies the maximum number of iterations allowed in locating the characteristic intersection points in the Quick Solver. The default value is 30.
SQS	The coefficient C_S, in Eqs. (47) and (49), that controls the amount of numerical smoothing necessary to stabilize the Quick Solver. The recommended and default value is 0.5.

D. Output Description

Program output consists of printed output, film plots, and punched cards (disc or tape file) for restart. The first two pages (or first three pages in the tabular input geometry case) of output include the program title, abstract, list of control parameters, fluid model, flow geometry, nozzle geometry, boundary conditions, artificial viscosity, molecular viscosity, turbulence model, and variable grid parameters.

Following the title pages is the initial-data surface. Before each initial-data surface, a page is printed that gives the mass flow, ratio of mass flow to inlet (L = 1) mass flow, exit momentum thrust, and ratio of momentum thrust to inlet momentum thrust for L = 1 to LMAX. These data are either data that have been read in or a 1D solution that has been computed by the program. All units are given. For planar flow, the mass flow units are lbm/in.-s or kg/cm-s and the momentum thrust units are lbf/in. or N/cm.

After the initial-data surface has been printed, the solution surfaces are printed. Before each solution surface, a page is printed that gives the mass flow, ratio of mass flow to inlet (L = 1) mass flow, exit momentum thrust, and ratio of momentum thrust to inlet momentum thrust for L = 1 to LMAX. After the mass flow page, the solution surfaces are printed. These surfaces have the same format as the initial-data surface. Each solution surface gives the flow field for a certain value of time. At the top of each solution surface page is the number of time steps N, the time, the time step, the number of subcycles NVCM, and the subcycled Courant number CNUMS. At the top right of each page are two pairs of numbers enclosed in parentheses. These give the grid points where the limiting time step was found. The one on the right is for the subcycled grid. As many solution planes as desired may be printed by varying the input data.

If requested (IAV ≠ 0), artificial viscosity, molecular viscosity, and turbulence parameters are printed before each solution plane. QUT denotes the x momentum equation right-hand-side terms in ft/s or m/s, QVT denotes the y momentum equation right-hand-side terms in ft/s or m/s, QPT denotes the internal energy equation right-hand-side terms in psia or kPa, and QRØT denotes the continuity equation right-hand-side terms in lbm/ft^3 or kg/m^3. AVMUR and TLMUR are the ratios of artificial and turbulent viscosities to the laminar value, respectively, Q is the turbulence energy at the N − 1 time step in ft^2/s^2 or

m^2/s^2, and E is the dissipation rate at the N − 1 time step in ft^2/s^3 or m^2/s^3. QQT is the turbulence energy equation right-hand-side terms in ft^2/s^2 or m^2/s^2, QET is the dissipation rate equation right-hand-side terms in ft^2/s^3 or m^2/s^3, and TML is the mixing-length (ITM = 1) or length scale (ITM = 2) in in. or cm. The parameters for the upper dual-flow-space wall are printed on the last page of the viscous printout. At the end of the viscosity parameters are the grid points whose viscous terms limit the time-step size in the x and y directions. Also printed is the ratio of the y terms to the x terms. The larger this ratio, the more restrictive the y direction terms become in limiting the time step size. If LPRINT and MPRINT are read in, the turbulence energy and dissipation rate convection, production, dissipation, and diffusion terms (not including axisymmetric terms) are also printed in internal units. Also, film plots with the units of the printed output are made for each requested time step. When the computation is stopped because the flow has satisfied the convergence tolerance, the physical time equals TSTØP, or the maximum number of time steps has been reached, the final solution plane is always printed and plotted.

E. Computing System Compatibility

1. **Deck Set-Up.** The deck begins with the common deck called MCC, followed by the main program called VNAP2 and the remaining function and subroutines. The common deck is preceeded by the card *CØMDECK,MCC, beginning in column 1. This common deck is separated from the main program VNAP2 by the card *DECK,VNAP2, also beginning in column 1. Any routine that uses the common deck MCC has the card *CALL,MCC, beginning in column 1, at the location where the common deck should be in that routine. The CDC routine UPDATE will place the common deck in each routine containing a *CALL,MCC card. This simplifies making changes to the COMMON statements as well as array sizes (see below). For computing systems without an UPDATE or comparable routine, remove the *CØMDECK,MCC and *DECK,VNAP2 cards and replace all *CALL,MCC cards with the common deck, MCC.

2. **Array Sizes.** This version of the program allows for a maximum of 41 x and 25 y mesh points. These values are set by use of a PARAMETER statement, which is the first card in the common deck MCC. In this PARAMETER statement, LI ⩾ LMAX, MI ⩾ MMAX, LI1 = LI + 1, and MI1 = MI + 1. MQS ⩾ MVCT sets the Quick Solver array sizes. When the Quick Solver is not being used (IQS = 0), then MQS can be set equal to one to reduce the amount of storage. LTS = LI and MTS = MI set the extended-interval time-smoothing array sizes. When the extended-interval time smoothing is not being used (NTST = 1 or NST = 0), then LTS and MTS can be set equal to one to reduce the amount of storage. By using the routine UPDATE, discussed above, the array sizes may be changed by changing the one PARAMETER statement card. For computing systems that do not allow a PARAMETER statement, remove the PARAMETER statement and replace the integers LI, MI, LI1, MI1, MQS, LTS, and MTS in the common block, as well as the two cards defining LD and MD (following the NAMELIST statements in program VNAP2) with the desired values.

3. **Film Plotting.** The subroutine PLØT discussion in Sec. II.A describes the Los Alamos National Laboratory system routines used by this code. For other computing systems, the Los Alamos routines in subroutine PLØT will have to be replaced by comparable routines. On the other hand, if velocity vector and contour plots are not needed, then subroutine PLØT can be replaced by a dummy subroutine.

4. **Single-Subscripted Arrays.** Unlike VNAP, VNAP2 contains no single subscripting of arrays that are dimensioned with multiple subscripts, because most current Fortran compilers generate nearly as efficient a code with either single or multiple subscripts. For example, the single subscript version of VNAP2 was approximately 1 to 2% faster than the multiple subscript version using the CDC FTN 4.8 compiler. This small increase in efficiency did not seem to be worth the added complexity.

F. Artificial Viscosity Discussion

The artificial viscosity model contains many parameters. However, in most cases the user needs to be concerned with only two, CAV and FDT. CAV controls the overall amount of smoothing and FDT controls the time step. If the space oscillations (that is, oscillations from point to point in the same time plane) are too large, then increase CAV. If the shock is too smeared, then decrease CAV. However, if the time oscillations (that is, oscillations at the same space point in different time planes) are too large, then decrease FDT. Increases in CAV often require decreases in FDT, whereas decreases in CAV often allow increases in FDT. For computation efficiency one uses large values of FDT and, therefore, small values of CAV. In calculations where FDT is too large, the solution usually "blows up" in less than 10 time steps. For calculations where CAV is too small, the solution usually takes longer to blow up. If FDT is smaller than necessary and CAV is larger than required, the solution will not blow up but, instead, will be inaccurate and inefficient. However, there is a lower limit of FDT below which space oscillations will appear. The code includes an artificial viscosity contribution in the time step calculation and, therefore, a given value of FDT will usually suffice for a wide range of CAV.

As an example, an oblique shock produced by supersonic flow (Mach naumber = 3.2) over a 30° wedge (pressure ratio = 6.84) required a CAV of 1.5 and an FDT of 0.8. In general, stronger shocks require larger values of CAV and smaller values of FDT. The opposite is true for weaker shocks.

The artificial viscosity discussed above is intended for shocks and is very small for contact surfaces and zero for expansions. Because of this, if contact surfaces are present, additional smoothing is usually needed. This can be accomplished by using the sound speed gradient option (ISS $\neq 0$). For ISS $\neq 0$, the sound speed gradient is added to the velocity divergence. If ISS = 1 and the divergence of the velocity is <0, then the sound speed gradient is set equal to zero, which again mainly smooths only shocks. If ISS = 2, the sound speed gradient is always nonzero, which smooths shocks, contact surfaces, and, unfortunately, expansions. Therefore, for contact surfaces or dual flows with very different densities, use the ISS = 2 option. The IDIVC $\neq 0$ (ISS = 1) option could also be used, but here both the velocity divergence and the sound speed gradient are nonzero, causing additional smearing of any expansions that may be present.

Another problem concerning the artificial viscosity is the shock wave-boundary layer interaction. Here, the artificial viscosity that is necessary for the shock may swamp the molecular and turbulent viscosities in the boundary layer. To minimize this problem, the artificial viscosity depends on the velocity divergence and not the shear gradients. In addition, λ_A and μ_A are multiplied by the Mach number squared in the subsonic part of the boundary layer. If this is not sufficient, the SMACH option can be used. There are no claims that this artificial viscosity model is the best way to treat shock wave-boundary layer interactions. It is to be hoped that additional work will produce better procedures.

G. Sample Calculations

1. **Case No. 1: Subsonic Constant Area, Supersonic Source Flow.** The geometry for this case is shown in Fig. 23 and consists of a constant area duct on top containing subsonic flow and a diverging duct on the bottom containing supersonic source flow. The data deck and printed output are presented in Figs. 24 and 25, respectively.

a. Namelist CNTRL. This case uses a 21 by 11 mesh, therefore LMAX = 21 and MMAX = 11. The maximum number of time steps NMAX is set equal to 500. After 500 time steps, the supersonic flow is steady, but the subsonic flow is still changing slightly. Film plots of the final solution plane are requested by setting NPLØT = 500. A nondimensional set of units is used, so IUNIT = 1. The gas constant for this nondimensional set of units is 0.01; therefore RGAS = 0.01. So that the calculation will not be stopped before the number of time steps reaches NMAX, TSTØP is increased to 100.0. The additional parameters are left equal to their default values.

b. Namelist IVS. An initial-data surface that is subsonic in the upper flow space and supersonic in the lower is desired. Because this is not possible using the internally generated initial data, a general initial-data surface is read in. Therefore, N1D = 0 and values for the arrays U, V, P, RØ, UL, VL, PL, and RØL must be read in. All the values are assumed to be constant in each flow space. The additional parameters are left equal to their default values.

c. Namelist GEMTRY. The flow geometry for this case is 2D planar flow; therefore NDIM = 0. The wall is a constant area duct; therefore NGEØM = 1. The inlet location XI equals 0.0, the exit location XE equals 4.0, and the radius RI equals 2.1547. No other input is required.

d. Namelist GCBL. Because this case has no centerbody, no input is required.

e. Namelist BC. Because the lower flow-space inflow is supersonic and the upper flow space is subsonic, ISUPER = 2. The stagnation pressure PT, stagnation temperature TT, and exit pressure PE for the upper flow space are 213.514, 124.2, and 180.0, respectively. Values for the arrays UI, VI, PI, and RØI, as well as the variables UIL, VIL, PIL, and RØIL, are read in for the lower flow space. No other input is required.

f. Namelist AVL. Because there are no shocks and the initial data is smooth, no input is required.

g. Namelist RVL. Because the flow is inviscid, no input is required.

h. Namelist TURBL. Because the flow is inviscid, no input is required.

i. Namelist DFSL. For this case, the upper and lower dual-flow-space walls are specified by LMAX equally spaced values of YL and YU and the corresponding negative of their slopes NXNYL and NXNYU; therefore NDFS = 2. The dual-flow-space walls begin at the inlet and end at the exit; therefore LDFSS = 1 and LDFSF = 21. The dual-flow-space walls correspond to the M = 6 row of grid points; therefore MDFS = 6. No other input is required.

j. Namelist VCL. Because a uniform grid is used, no input is required.

2. **Case No. 2: Supersonic Source, Subsonic Constant Area Flow.** This case is the same as Case No. 1, except that the lower dual-flow space is the subsonic constant area duct and the upper flow space is the supersonic source flow. The geometry is shown in Fig. 26. The data deck and printed output are presented in Figs. 27 and 28, respectively. Because the discussion for this case closely follows that of Case No. 1, it is not included here.

3. **Case No. 3: Subsonic Airfoil.** The geometry for this case is shown in Fig. 29 and consists of a 10° double wedge airfoil between two solid walls. The data deck and printed output are presented in Figs. 30 and 31, respectively.

a. Namelist CNTRL. This case uses a 21 by 11 mesh; therefore LMAX = 21 and MMAX = 11. The maximum number of time steps NMAX is set equal to 500. Film plots of the final solution plane are requested by setting NPLØT equal to 500. A nondimensional set of units is used, so IUNIT = 1. The gas constant for this nondimensional set of units is 0.01; therefore RGAS = 0.01. So that the calculation will not be stopped before the number of time steps reaches NMAX, TSTØP is increased to 100.0. The additional parameters are left equal to their default values.

b. Namelist IVS. A subsonic initial-data surface is computed by the program, so N1D = –2. The Mach number everywhere is set by specifying the height for the area where the Mach number equals 1.0; therefore RSTAR = 0.7464. No other input is required.

c. Namelist GEMTRY. The flow geometry for this case is 2D planar flow; therefore NDIM = 0. The wall is a constant area duct, therefore NGEØM = 1. The inlet location XI = 0.0, the exit location XE = 4.0, and the radius RI = 1.0. No other input is required.

d. Namelist GCBL. The centerbody is a horizontal wall, and so NGCB = 1. The radius RICB = 0.0. No other input is required.

e. Namelist BC. The stagnation pressure PT = 213.514, the stagnation temperature TT = 124.2, and the exit pressure PE = 180.0. No other input is required.

f. Namelist AVL. Because there are no shocks and the initial data is smooth, no input is required.

g. Namelist RVL. Because the flow is inviscid, no input is required.

h. Namelist TURBL. Because the flow is inviscid, no input is required.

i. Namelist DFSL. For this case, the upper and lower dual-flow-space walls are specified by 11 (LDFSF – LDFSS + 1) equally spaced values of YL and YU and the corresponding negative of their slopes NXNYL and NXNYU; therefore NDFS = 2. The dual-flow-space walls begin at L = 6 and end at L = 16, therefore LDFSS = 6 and LDFSF = 16. The dual-flow-space walls correspond to the M = 6 row of grid points; therefore MDFS = 6. No other input is required.

j. Namelist VCL. Because a uniform grid is used, no input is required.

ACKNOWLEDGMENTS

This work was performed at the Los Alamos National Laboratory, Los Alamos, New Mexico, and supported by the National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia, and the United States Department of Energy. The contract monitor was Richard G. Wilmoth of the Langley Research Center Propulsion Aerodynamics Branch. The author wishes to thank Richard G. Wilmoth, Lawrence E. Putnam, and R. Charles Swanson, Jr., of the Langley Research Center Propulsion Aerodynamics Branch, and Bart J. Daly, of the Los Alamos National Laboratory, for their many helpful discussions.

REFERENCES

1. M. C. Cline, "VNAP: A Computer Program for Computation of Two-Dimensional, Time-Dependent, Compressible, Viscous, Internal Flow," Los Alamos Scientific Laboratory report LA-7326 (November 1978).

2. M. C. Cline, "NAP: A Computer Program for the Computation of Two-Dimensional, Time-Dependent, Inviscid Nozzle Flow," Los Alamos Scientific Laboratory report LA-5984 (January 1977).

3. R. W. MacCormack, "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Hypervelocity Impact Conference, Cincinnati, Ohio, April 1969, AIAA Paper 69-354.

4. B. E. Launder, A. Morse, W. Rodi, and D. B. Spalding, "The Prediction of Free Shear Flows—A Comparison of the Performance of Six Turbulence Models," Proc. NASA Conf. Free Shear Flows, NASA Langley Research Center, Hampton, Virginia, July 1972 (NASA SP-321, 1973), Vol. I, pp. 361-426.

5. Tuncer Cebeci, A. M. O. Smith, and G. Mosinskis, "Calculation of Compressible Adiabatic Turbulent Boundary Layers," AIAA J. **8**, 1974 (1970).

6. W. P. Jones and B. E. Launder, "The Prediction of Laminarization with a Two-Equation Model of Turbulence," Int. J. Heat Mass Transfer **15**, 301 (1972).

7. W. P. Jones and B. E. Launder, "The Calculation of Low-Reynolds-Number Phenomena with a Two-Equation Model of Turbulence," Int. J. Heat Mass Transfer **16**, 1119 (1973).

8. B. E. Launder and B. I. Sharma, "Application of the Energy-Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disc," Lett. Heat Mass Transfer **1**, 131 (1974).

9. K. Hanjalic and B. E. Launder, "Contribution Towards a Reynolds-Stress Closure for Low-Reynolds-Number Turbulence," J. Fluid Mech. **74**, 593 (1976).

10. T. J. Coakley and J. R. Viegas, "Turbulence Modeling of Shock Separated Boundary-Layer Flows," Symp. Turbulent Shear Flows, University Park, Pennsylvania, April 18-20, 1977.

11. C. M. Hung and R. W. MacCormack, "Numerical Solutions of Supersonic and Hypersonic Laminar Compression Corner Flows," AIAA J. **14**, 475 (1976).

12. F. H. Harlow and A. A. Amsden, "A Numerical Fluid Dynamics Calculation Method for All Flow Speeds," J. Comput. Phys. **8**, 197 (1971).

13. R. W. MacCormack, "A Rapid Solver for Hyperbolic Systems of Equations," Lect. Notes Phys. **59**, 307-317 (1976).

14. C. P. Kentzer, "Discretization of Boundary Conditions of Moving Discontinuities," Lect. Notes Phys. **8**, 108-113 (1970).

15. G. Moretti, "The λ-Scheme," Comput. Fluids **7**, 191 (1979).

16. R. A. Serra, "Determination of Internal Gas Flows by a Transient Numerical Technique," AIAA J. **10**, 603 (1972).

17. J. Oliger and A. Sundström, "Theoretical and Practical Aspects of Some Initial Boundary Value Problems in Fluid Dynamics," SIAM J. Appl. Math. **35**, 419 (1978).

18. G. Moretti and M. Abbett, "A Time-Dependent Computational Method for Blunt Body Flows," AIAA J. 4, 2136 (1966).

19. D. H. Rudy and J. C. Strikwerda, "A Nonreflecting Outflow Boundary Condition for Subsonic Navier-Stokes Calculations," J. Comput. Phys. 36, 55 (1980).

20. M. C. Cline, "Stability Aspects of Diverging Subsonic Flow," AIAA J. 18, 534 (1980).

21. G. Moretti, "Comment on Stability Aspects of Diverging Subsonic Flow," AIAA J. 19 (May 1981).

22. M. C. Cline, "Reply by Author to G. Moretti," AIAA J. 19, (May 1981).

23. G. Moretti and M. Pandolfi, "On the Calculation of Subsonic Flows in Ducts," Polytechnic Institute of New York report M/AE 80-18 (June 1980).

24. M. C. Cline and R. G. Wilmoth, "Computation of High Reynolds Number Internal/External Flows," AIAA 14th Fluid Plasma Dyn. Conf., Palo Alto, California, June 23-25, 1981, AIAA Paper 81-1194.

25. M. C. Cline, "Computation of Steady Nozzle Flow by a Time-Dependent Method," AIAA J. 12, 419 (1974).

26. R. F. Cuffel, L. H. Back, and P. F. Massier, "Transonic Flow-Field in a Supersonic Nozzle with Small Throat Radius of Curvature," AIAA J. 7, 1364 (1969).

27. D. E. Reubush, "Experimental Study of the Effectiveness of Cylindrical Plume Simulators for Predicting Jet-On Boattail Drag of Mach Numbers up to 1.30," NASA TN D-7795, 1974.

28. R. C. Swanson, Jr., "Numerical Solutions of the Navier-Stokes Equations for Transonic Afterbody Flows," NASA Technical Paper 1784 (December 1980).

29. L. J. S. Bradbury, "The Structure of a Self-Preserving Turbulent Plane Jet," J. Fluid Mech. 23, 31 (1965).

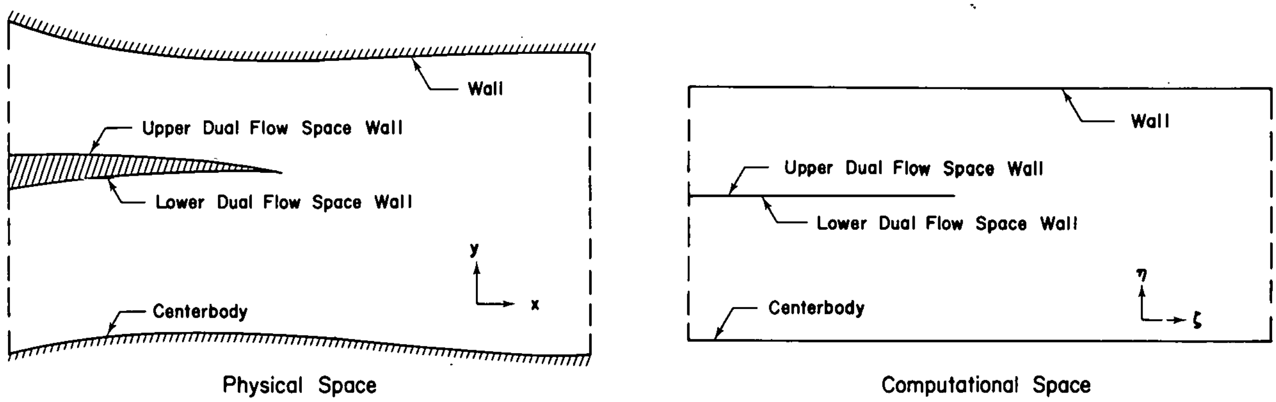

Fig. 1.
Physical and computational spaces.

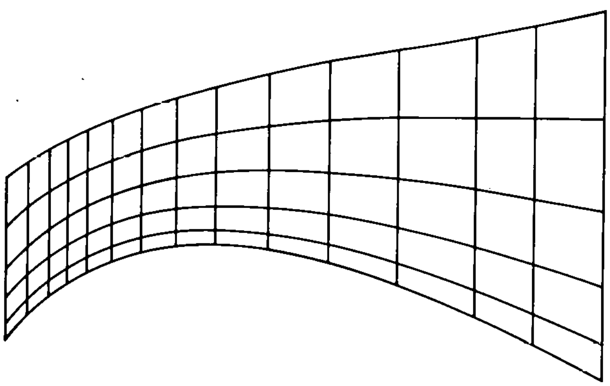

Fig. 2.
Physical space grid.

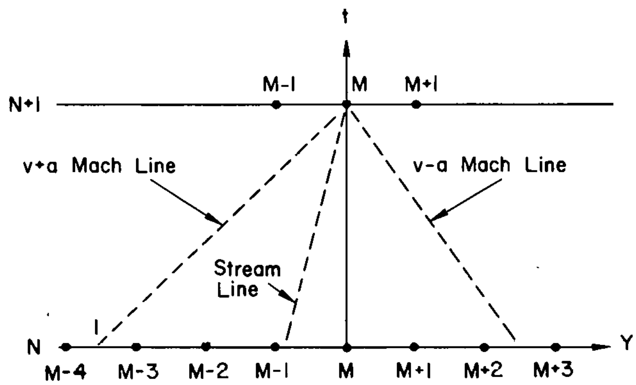

Fig. 3.
Quick Solver characteristic grid.

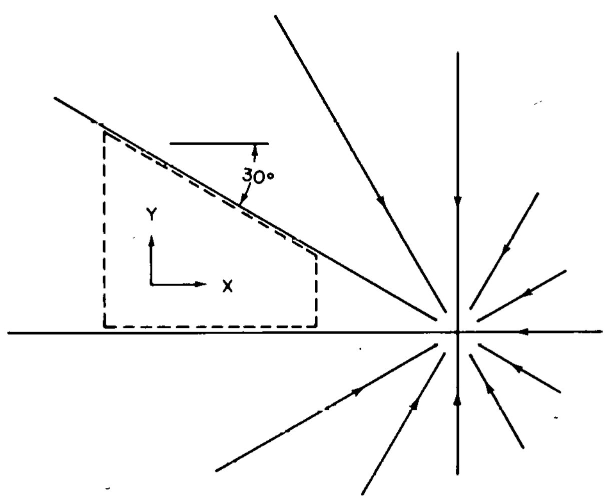

Fig. 4.
Planar subsonic sink flow.

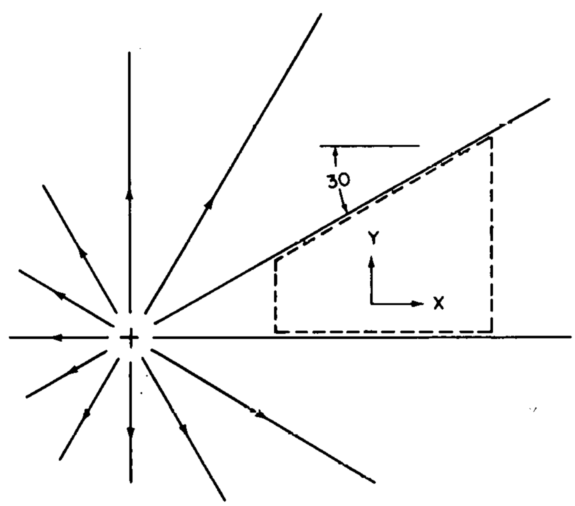

Fig. 5.
Planar supersonic source flow.

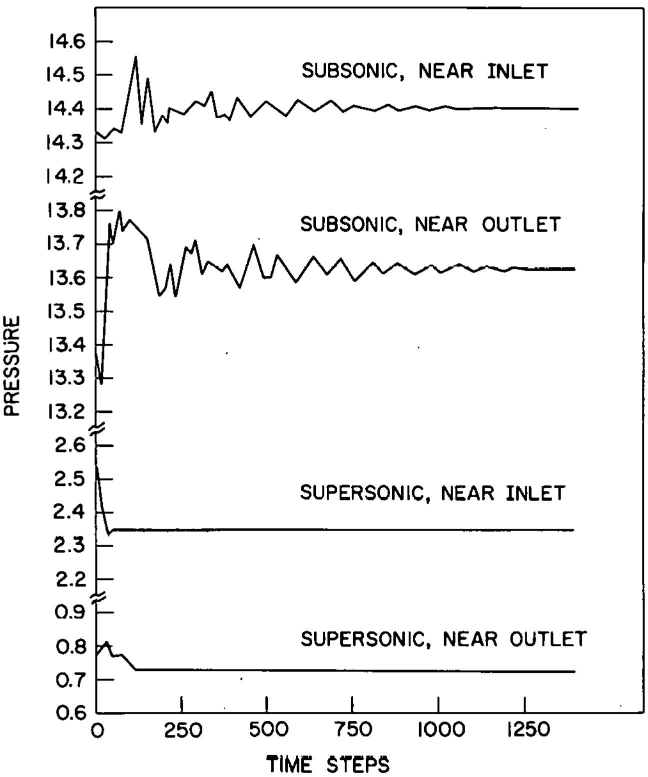

Fig. 6.
Subsonic sink and supersonic source flow solutions.

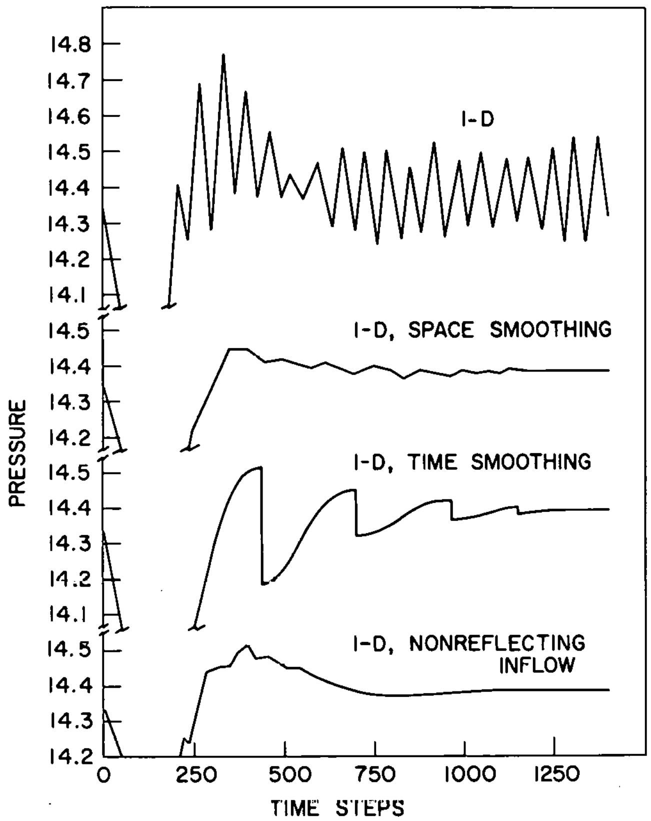

Fig. 8.
Subsonic sink flow with the u, v, and ρ inflow boundary condition.

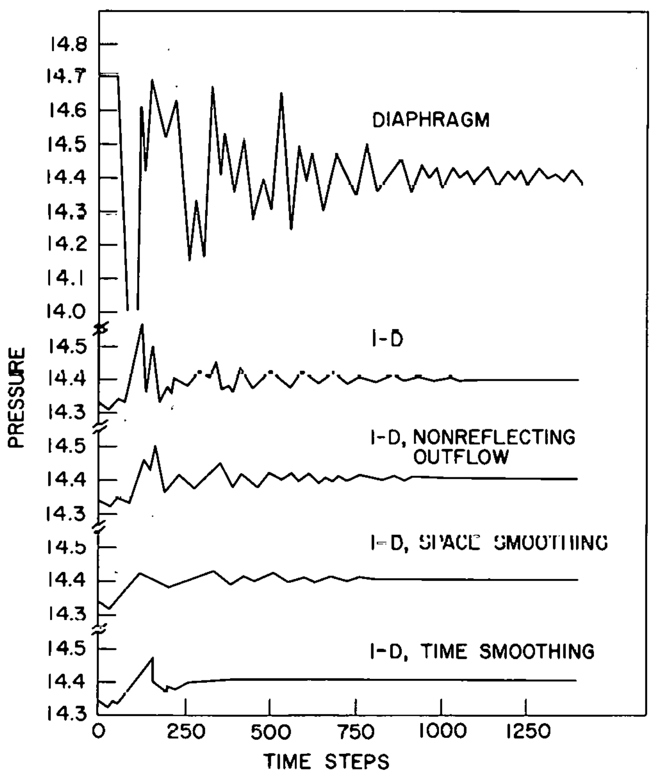

Fig. 7.
Subsonic sink flow with the p_T, T_T, and θ inflow boundary condition.

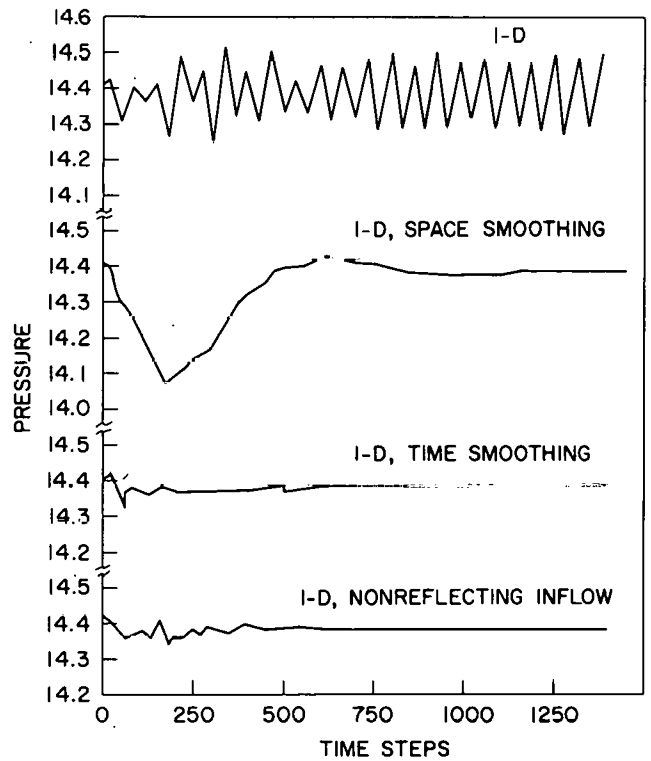

Fig. 9.
Subsonic sink flow with the u, v, and ρ inflow boundary condition and matched mass flow, 1D, initial-data surface.

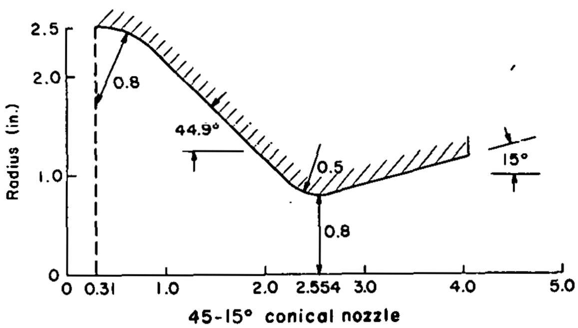

Fig. 10.
Nozzle geometry for Case No. 1.

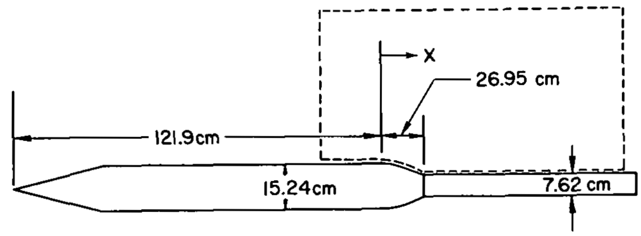

Fig. 12.
Boattail afterbody geometry for Case No. 2.

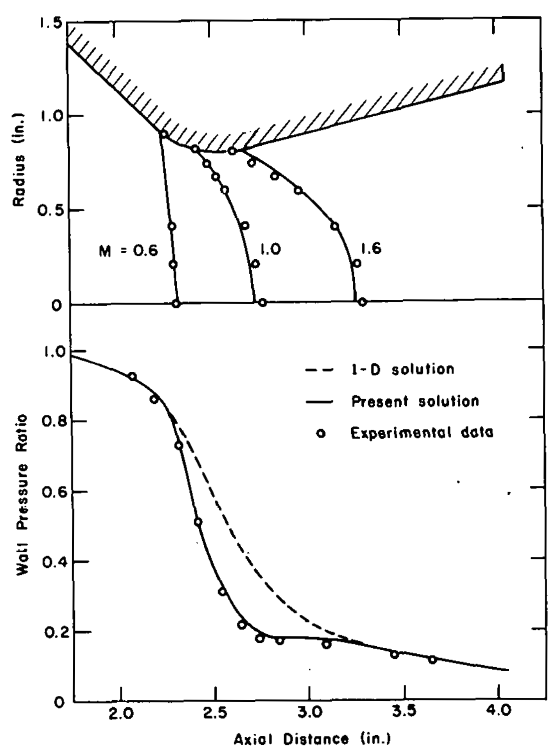

Fig. 11.
Mach number contours (top) and wall pressure ratio for Case No. 1.

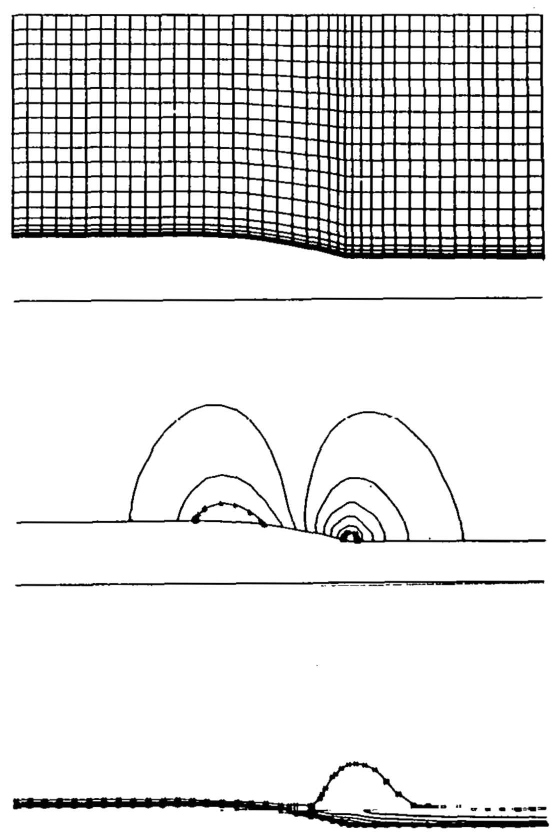

Fig. 13.
Physical space grid (top), pressure (middle), and Mach number contours for Case No. 2.

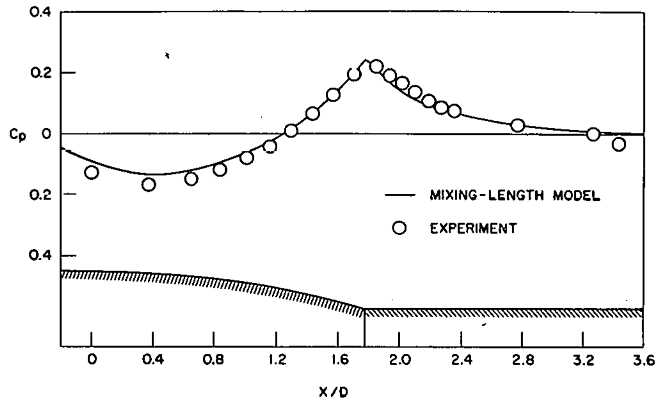

Fig. 14.
Surface pressure coefficient for Case No. 2.

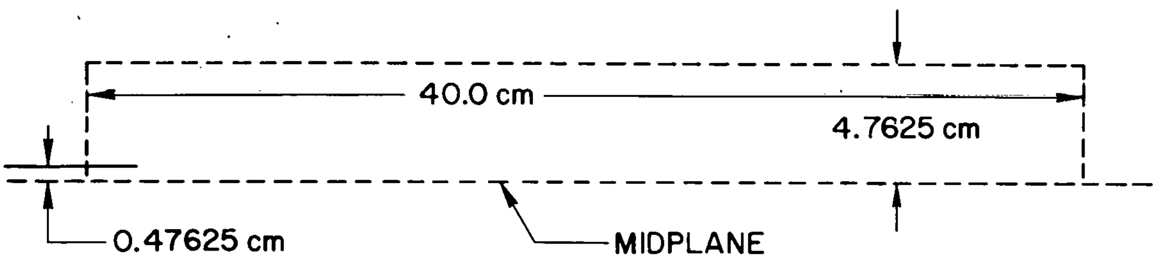

Fig. 15.
Plane jet geometry for Case No. 3.

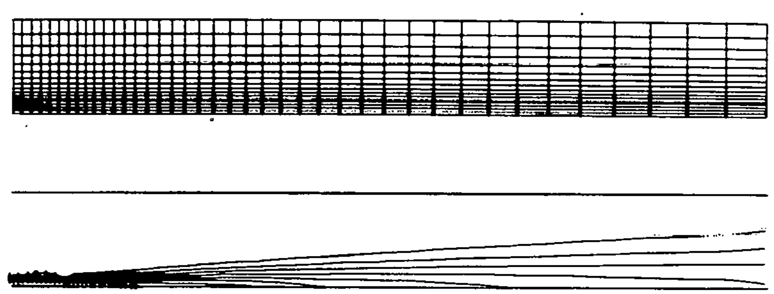

Fig. 16.
Physical space grid (top) and Mach number contours for Case No. 3.

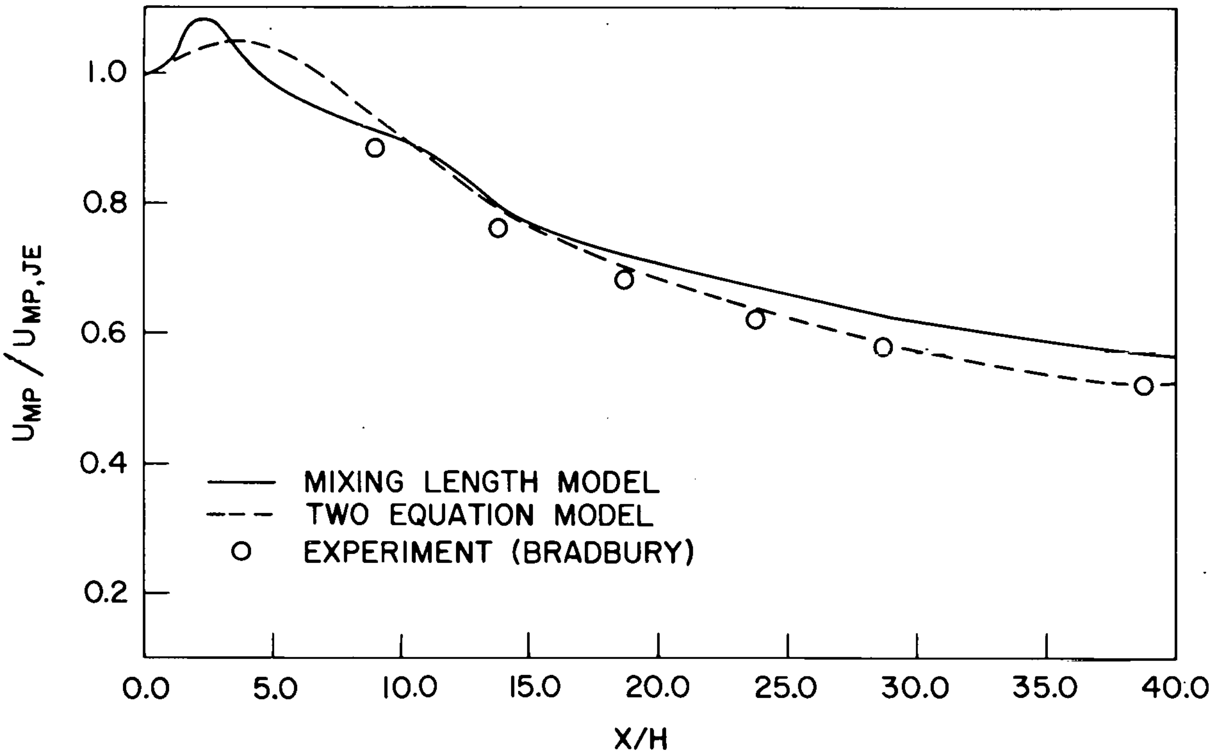

Fig. 17.
Midplane velocity decay for Case No. 3.

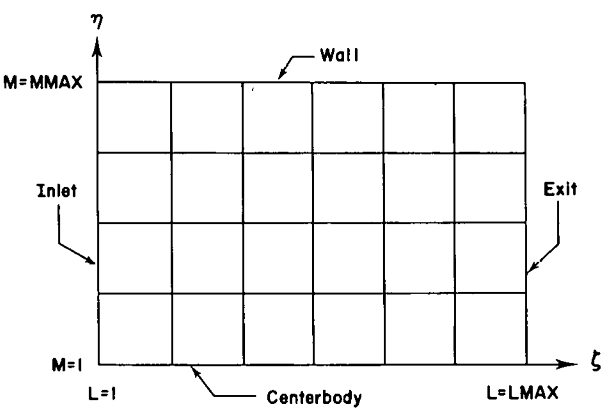

Fig. 18.
Single-flow-space computational grid.

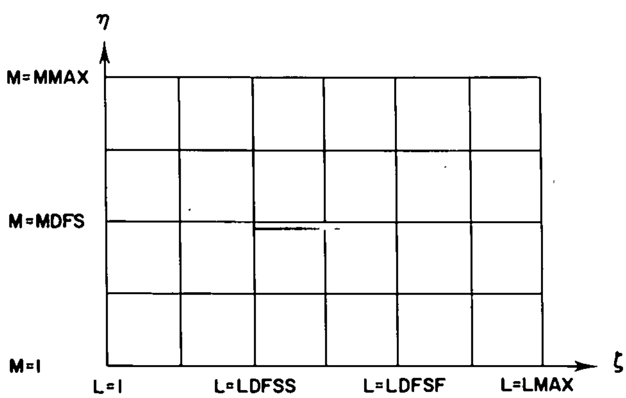

Fig. 19.
Dual-flow-space computational grid.

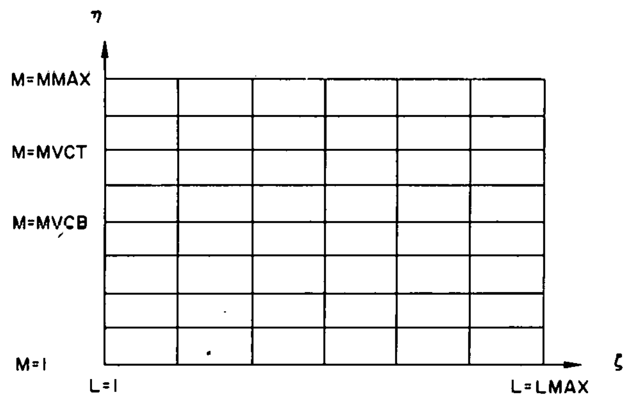

Fig. 20.
Subcycled computational grid.

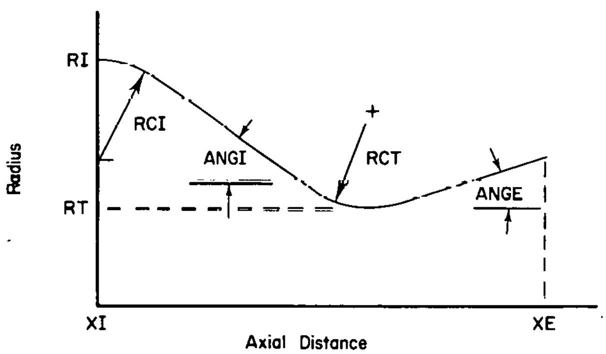

Fig. 21.
Circular-arc, conical wall geometry.

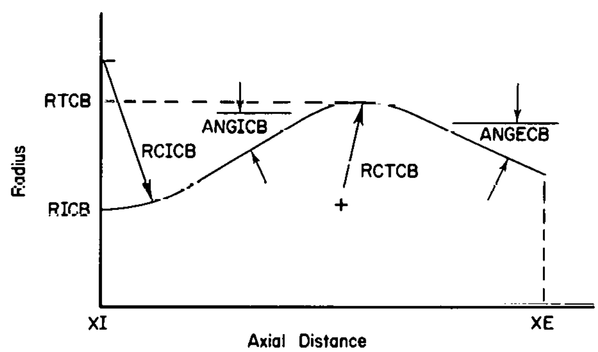

Fig. 22.
Circular-arc, conical centerbody geometry.

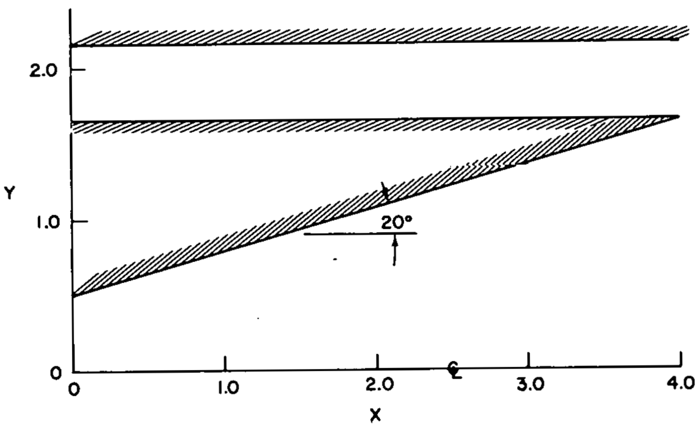

Fig. 23.
Case No. 1 geometry.

```
VNAP2 CASE 1 - SUBSONIC CONSTANT AREA-SUPERSONIC SOURCE FLOW
 $CNTRL LMAX=21,MMAX=11,NMAX=500,NPLOT=500,IUNIT=1,RGAS=0.01,
 TSTOP=100.0   $
 $IVS N1D=0,
 U(1,1,1)=21*1.39,U(1,2,1)=21*1.39,U(1,3,1)=21*1.39,U(1,4,1)=21*1.39,
 U(1,5,1)=21*1.39,UL=21*1.39,U(1,6,1)=21*0.67,U(1,7,1)=21*0.67,
 U(1,8,1)=21*0.67,U(1,9,1)=21*0.67,U(1,10,1)=21*0.67,U(1,11,1)=21*0.67,
 V(1,1,1)=21*0.4,V(1,2,1)=21*0.4,V(1,3,1)=21*0.4,V(1,4,1)=21*0.4,
 V(1,5,1)=21*0.4,VL=21*0.4,V(1,6,1)=21*0.0,V(1,7,1)=21*0.0,
 V(1,8,1)=21*0.0,V(1,9,1)=21*0.0,V(1,10,1)=21*0.0,V(1,11,1)=21*0.0,
 P(1,1,1)=21*81.7,P(1,2,1)=21*81.7,P(1,3,1)=21*81.7,P(1,4,1)=21*81.7,
 P(1,5,1)=21*81.7,PL=21*81.7,P(1,6,1)=21*180.0,P(1,7,1)=21*180.0,
 P(1,8,1)=21*180.0,P(1,9,1)=21*180.0,P(1,10,1)=21*180.0,P(1,11,1)=21*180.0,
 RO(1,1,1)=21*86.6,RO(1,2,1)=21*86.6,RO(1,3,1)=21*86.6,RO(1,4,1)=21*86.6,
 RO(1,5,1)=21*86.6,ROL=21*86.6,RO(1,6,1)=21*150.0,RO(1,7,1)=21*150.0,
 RO(1,8,1)=21*150.0,RO(1,9,1)=21*150.0,RO(1,10,1)=21*150.0,
 RO(1,11,1)=21*150.0   $
 $GEMTRY NDIM=0,NGEOM=1,XI=0.0,XE=4.0,RI=2.1547   $
 $GCBL   $
 $BC ISUPER=2,PT=213.514,TT=124.2,PE=180.0,
 UI=1.301538,1.3092,1.3276,1.3494,1.3701,UIL=1.3877,
 VI=0.0,0.07559,0.1533,0.2337,0.3164,VIL=0.4006,
 PI=100.0,98.7152,95.4717,91.2200,86.5300,PIL=81.7273,
 ROI=100.0,99.0805,96.7441,93.6450,90.1800,ROIL=86.5775   $
 $AVL   $
 $RVL   $
 $TURBL   $
 $DFSL NDFS=2,LDFSS=1,LDFSF=21,MDFS=6,
 YL=0.5,0.5577,0.6155,0.6732,0.7309,0.7887,0.8464,0.9041,0.9619,1.0196,
 1.0774,1.1361,1.1928,1.2506,1.3083,1.3860,1.4238,1.4815,1.5392,1.5970,
 1.6547,
 NXNYL=21*-0.28868,
 YU=21*1.6547,
 NXNYU=21*-0.0   $
 $VCL   $
```

Fig. 24.
Case No. 1 data deck.

VNAP2, A COMPUTER PROGRAM FOR THE COMPUTATION OF TWO-DIMENSIONAL, TIME-DEPENDENT, COMPRESSIBLE, TURBULENT FLOW

BY MICHAEL C. CLINE, T-3 - LOS ALAMOS NATIONAL LABORATORY

PROGRAM ABSTRACT -

THE NAVIER-STOKES EQUATIONS FOR TWO-DIMENSIONAL, TIME-DEPENDENT FLOW ARE SOLVED USING THE SECOND-ORDER, MACCORMACK FINITE-DIFFERENCE SCHEME. ALL BOUNDARY CONDITIONS ARE COMPUTED USING A SECOND-ORDER, REFERENCE PLANE CHARACTERISTIC SCHEME WITH THE VISCOUS TERMS TREATED AS SOURCE FUNCTIONS. THE FLUID IS ASSUMED TO BE A PERFECT GAS. THE STEADY-STATE SOLUTION IS OBTAINED AS THE ASYMPTOTIC SOLUTION FOR LARGE TIME. THE FLOW BOUNDARIES MAY BE ARBITRARY CURVED SOLID WALLS AS WELL AS JET ENVELOPES. THE GEOMETRY MAY CONSIST OF SINGLE AND DUAL FLOWING STREAMS. TURBULENCE EFFECTS ARE MODELED WITH EITHER A MIXING-LENGTH, A TURBULENCE ENERGY EQUATION, OR A TURBULENCE ENERGY-DISSIPATION RATE EQUATIONS MODEL. THIS PROGRAM ALLOWS VARIABLE GRID SPACING AND INCLUDES OPTIONS TO SPEED UP THE CALCULATION FOR HIGH REYNOLDS NUMBER FLOWS.

JOB TITLE -

VNAP2 CASE 1 - SUBSONIC CONSTANT AREA-SUPERSONIC SOURCE FLOW

CONTROL PARAMETERS -

LMAX=21	MMAX=11	NMAX= 500	NPRINT= 0	NPLOT= 500	FDT= .90	FDT1=1.00	FDTI= .90	IPUNCH=0
IUI=1	IUO=1	IVPTS=1	NCONVI= 1	TSTOP= .10E+03	N1D= 0	TCONV=0.000	NASM=1	IUNIT=1
RSTAR=	0.000000	RSTARS=	0.0000000	PLOW= .0100	ROLOW=	.000100	VDT= .25	VDT1= .25

FLUID MODEL -

THE RATIO OF SPECIFIC HEATS, GAMMA =1.4000 AND THE GAS CONSTANT, R = .0100 (FT-LBF/LBM-R)

FLOW GEOMETRY -

TWO-DIMENSIONAL, PLANAR FLOW HAS BEEN SPECIFIED

DUCT GEOMETRY -

A CONSTANT AREA DUCT HAS BEEN SPECIFIED BY XI= 0.0000 (IN), RI= 2.1547 (IN), AND XE= 4.0000 (IN)

Fig. 25.
Case No. 1 output.

DUAL FLOW SPACE BOUNDARY GEOMETRY -

GENERAL BOUNDARIES HAVE BEEN SPECIFIED BY THE FOLLOWING PARAMETERS.

L	XP(IN)	YL(IN)	SLOPEL	YU(IN)	SLOPEU
1	0.0000	.5000	.2887	1.6547	0.0000
2	.2000	.5577	.2887	1.6547	0.0000
3	.4000	.6155	.2887	1.6547	0.0000
4	.6000	.6732	.2887	1.6547	0.0000
5	.8000	.7309	.2887	1.6547	0.0000
6	1.0000	.7887	.2887	1.6547	0.0000
7	1.2000	.8464	.2887	1.6547	0.0000
8	1.4000	.9041	.2887	1.6547	0.0000
9	1.6000	.9619	.2887	1.6547	0.0000
10	1.8000	1.0196	.2887	1.6547	0.0000
11	2.0000	1.0774	.2887	1.6547	0.0000
12	2.2000	1.1351	.2887	1.6547	0.0000
13	2.4000	1.1928	.2887	1.6547	0.0000
14	2.6000	1.2506	.2887	1.6547	0.0000
15	2.8000	1.3083	.2887	1.6547	0.0000
16	3.0000	1.3660	.2887	1.6547	0.0000
17	3.2000	1.4238	.2887	1.6547	0.0000
18	3.4000	1.4815	.2887	1.6547	0.0000
19	3.6000	1.5392	.2887	1.6547	0.0000
20	3.8000	1.5970	.2887	1.6547	0.0000
21	4.0000	1.6547	.2887	1.6547	0.0000

Fig. 25. (cont)

BOUNDARY CONDITIONS -

M	PT(PSIA)	TT(R)	THETA(DEG)	PE(PSIA)	FSQ(FT2/S2)	FSE(FT2/S3)
1	213.5140	124.20	0.00	180.00000	.0001	.1
2	213.5140	124.20	0.00	180.00000	.0001	.1
3	213.5140	124.20	0.00	180.00000	.0001	.1
4	213.5140	124.20	0.00	180.00000	.0001	.1
5	213.5140	124.20	0.00	180.00000	.0001	.1
6	213.5140	124.20	0.00	180.00000	.0001	.1
6	213.5140	124.20	0.00	180.00000	.0001	.1
7	213.5140	124.20	0.00	180.00000	.0001	.1
8	213.5140	124.20	0.00	180.00000	.0001	.1
9	213.5140	124.20	0.00	180.00000	.0001	.1
10	213.5140	124.20	0.00	180.00000	.0001	.1
11	213.5140	124.20	0.00	180.00000	.0001	.1

IINLET=0 IEXITT=0 IEX=1 ISUPER=2 DYW= .0010 IVBC=0 INBC=0 IWALL=0 IWALLO=0 ALI=0.00 ALE=0.00
ALW=0.00 NSTAG=0 NPE= 0 PEI= 0.00000

FREE-SLIP WALLS ARE SPECIFIED

ADIABATIC UPPER WALL IS SPECIFIED

ADIABATIC LOWER DUAL FLOW SPACE BOUNDARY IS SPECIFIED

ADIABATIC UPPER DUAL FLOW SPACE BOUNDARY IS SPECIFIED

ARTIFICAL VISCOSITY -

CAV=0.00 XMU= .40 XLA=1.00 PRA= .70 XRD= .60 LSS= 1 LSF=999 IDIVC=0 ISS=0 SMACH=0.00
NST= 0 SMP=1.00 SMPF=1.00 SMPT=1.00 SMPTF=1.00 NTST= 1 IAV=0 MSS= 1 MSF=999

MOLECULAR VISCOSITY -

CMU=0. (LBF-S/FT2) CLA= 0. (LBF-S/FT2) CK=0. (LBF/S-R) EMU=0.00 ELA=0.00 EK=0.00

TURBULENCE MODEL -

NO MODEL IS SPECIFIED

VARIABLE GRID PARAMETERS -

IST=0 MVCB= 0 MVCT= 0 IQS=0 NIQSS=2 NIQSF=0 NVCMI= 0 ILQS=30 SQS= .50 CQS= .001

***** EXPECT FILM OUTPUT FOR N= 0 *****

N= 10, T= .67077118 SECONDS DT= .06533031 SECONDS, NVCM = 1, CNUMS = 1.00, (5, 2), (0, 0)
N= 20, T= 1.30994032 SECONDS DT= .06310698 SECONDS, NVCM = 1, CNUMS = 1.00, (4, 2), (0, 0)
N= 30, T= 1.92866943 SECONDS DT= .06108173 SECONDS, NVCM = 1, CNUMS = 1.00, (7, 2), (0, 0)
N= 40, T= 2.52996808 SECONDS DT= .05947900 SECONDS, NVCM = 1, CNUMS = 1.00, (9, 2), (0, 0)
N= 50, T= 3.11792749 SECONDS DT= .05828222 SECONDS, NVCM = 1, CNUMS = 1.00, (12, 2), (0, 0)
N= 60, T= 3.69420884 SECONDS DT= .05712885 SECONDS, NVCM = 1, CNUMS = 1.00, (16, 2), (0, 0)
N= 70, T= 4.25870480 SECONDS DT= .05537706 SECONDS, NVCM = 1, CNUMS = 1.00, (19, 2), (0, 0)
N= 80, T= 4.82666822 SECONDS DT= .05753545 SECONDS, NVCM = 1, CNUMS = 1.00, (20, 2), (0, 0)

Fig. 25. (cont)

N=	90,	T=	5.40287119	SECONDS,	DT=	.05753144	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	100,	T=	5.97823303	SECONDS,	DT=	.05755439	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	110,	T=	6.55370146	SECONDS,	DT=	.05754307	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	120,	T=	7.12915448	SECONDS,	DT=	.05754566	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	130,	T=	7.70460850	SECONDS,	DT=	.05754561	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	140,	T=	8.28006455	SECONDS,	DT=	.05754551	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	150,	T=	8.85551982	SECONDS,	DT=	.05754552	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	160,	T=	9.43097477	SECONDS,	DT=	.05754549	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	170,	T=	10.00642973	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	180,	T=	10.58188469	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	190,	T=	11.15733965	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	200,	T=	11.73279462	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	210,	T=	12.30824958	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	220,	T=	12.88370455	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	230,	T=	13.45915951	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	240,	T=	14.03461448	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	250,	T=	14.61006944	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	260,	T=	15.18552440	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	270,	T=	15.76097937	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	280,	T=	16.33643433	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	290,	T=	16.91188930	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	300,	T=	17.48734426	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	310,	T=	18.06279923	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	320,	T=	18.63825419	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	330,	T=	19.21370915	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	340,	T=	19.78916412	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	350,	T=	20.36461908	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	360,	T=	20.94007405	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	370,	T=	21.51552901	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	380,	T=	22.09098398	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	390,	T=	22.66643894	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	400,	T=	23.24189390	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	410,	T=	23.81734887	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	420,	T=	24.39280383	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	430,	T=	24.96825880	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	440,	T=	25.54371376	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	450,	T=	26.11916873	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	460,	T=	26.69462369	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	470,	T=	27.27007865	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	480,	T=	27.84553362	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	490,	T=	28.42098858	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)
N=	500,	T=	28.99644355	SECONDS,	DT=	.05754550	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 2),	(0, 0)

Fig. 25. (cont)

MASS FLOW AND THRUST CALCULATION, N= 500

L	MF(LBM/S)	MF/MFI	T(LBF)	T/TI
1	112.31686	1.0000	116.3781	1.0000
2	113.59665	1.0114	131.6404	1.1311
3	114.00741	1.0151	140.5393	1.2076
4	114.20202	1.0168	146.8395	1.2617
5	114.31913	1.0178	151.7055	1.3036
6	114.40858	1.0186	155.6649	1.3376
7	114.46970	1.0192	158.9635	1.3659
8	114.51786	1.0196	161.7840	1.3902
9	114.56452	1.0200	164.2508	1.4114
10	114.59610	1.0203	166.4094	1.4299
11	114.62785	1.0206	168.3425	1.4465
12	114.64830	1.0208	170.0656	1.4613
13	114.66477	1.0209	171.6242	1.4747
14	114.68457	1.0211	173.0564	1.4870
15	114.69615	1.0212	174.3582	1.4982
16	114.70792	1.0213	175.5579	1.5085
17	114.71788	1.0214	176.6759	1.5181
18	114.73392	1.0215	177.7096	1.5270
19	114.71499	1.0214	178.6491	1.5351
20	114.89094	1.0229	179.9148	1.5460
21	114.86489	1.0227	180.7273	1.5529

Fig. 25. (cont)

SOLUTION SURFACE NO. 500 - TIME = 28.99644355 SECONDS (DELTA T = .05754550, NVCM = 1, CNUMS = 1.00, (20, 2), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
1	1	0.0000	0.0000	1.3015	0.0000	100.00000	100.000000	1.3015	1.1000	100.0000
1	2	0.0000	.1000	1.3092	.0756	98.71520	99.080500	1.3114	1.1104	99.6313
1	3	0.0000	.2000	1.3276	.1533	95.47170	96.744100	1.3364	1.1370	98.6848
1	4	0.0000	.3000	1.3494	.2337	91.22000	93.645000	1.3695	1.1727	97.4104
1	5	0.0000	.4000	1.3701	.3164	86.53000	90.180000	1.4062	1.2132	95.9525
1	6	0.0000	.5000	1.3877	.4006	81.72730	86.577500	1.4444	1.2564	94.3979
1	6	0.0000	1.6547	.6441	0.0000	179.92863	152.129942	.6441	.5006	118.2730
1	7	0.0000	1.7547	.6441	0.0000	179.92863	152.129942	.6441	.5006	118.2730
1	8	0.0000	1.8547	.6441	0.0000	179.92863	152.129942	.6441	.5006	118.2730
1	9	0.0000	1.9547	.6441	0.0000	179.92863	152.129942	.6441	.5006	118.2730
1	10	0.0000	2.0547	.6441	0.0000	179.92863	152.129942	.6441	.5006	118.2730
1	11	0.0000	2.1547	.6441	0.0000	179.92863	152.129942	.6441	.5006	118.2730
2	1	.2000	0.0000	1.5437	0.0000	70.42400	77.366711	1.5437	1.3675	91.0262
2	2	.2000	.1115	1.5460	.0941	69.84010	76.921446	1.5488	1.3738	90.7941
2	3	.2000	.2231	1.5485	.1857	68.51261	75.925220	1.5596	1.3876	90.2370
2	4	.2000	.3346	1.5515	.2765	66.53859	74.403717	1.5759	1.4084	89.4291
2	5	.2000	.4462	1.5540	.3651	64.12068	72.501138	1.5963	1.4345	88.4409
2	6	.2000	.5577	1.5492	.4472	62.00111	70.846121	1.6125	1.4568	87.5152
2	6	.2000	1.6547	.6441	0.0000	179.92401	152.127155	.6441	.5006	118.2721
2	7	.2000	1.7547	.6441	.0000	179.92401	152.127155	.6441	.5006	118.2721
2	8	.2000	1.8547	.6441	-.0000	179.92401	152.127155	.6441	.5006	118.2721
2	9	.2000	1.9547	.6441	.0000	179.92401	152.127155	.6441	.5006	118.2721
2	10	.2000	2.0547	.6441	-.0000	179.92401	152.127155	.6441	.5006	118.2721
2	11	.2000	2.1547	.6441	0.0000	179.92401	152.127155	.6441	.5006	118.2721
3	1	.4000	0.0000	1.6850	0.0000	54.68094	64.419078	1.6850	1.5457	84.8831
3	2	.4000	.1231	1.6841	.1023	54.44593	64.240050	1.6872	1.5489	84.7539
3	3	.4000	.2462	1.6807	.2019	53.77164	63.725641	1.6928	1.5575	84.3799
3	4	.4000	.3693	1.6756	.2990	52.69850	62.868180	1.7021	1.5712	83.8238
3	5	.4000	.4924	1.6689	.3913	51.31118	61.731773	1.7142	1.5891	83.1196
3	6	.4000	.6155	1.6572	.4784	49.91357	60.590319	1.7248	1.6061	82.3788
3	6	.4000	1.6547	.6441	0.0000	179.92725	152.129108	.6441	.5006	118.2727
3	7	.4000	1.7547	.6441	-.0000	179.92725	152.129108	.6441	.5006	118.2727
3	8	.4000	1.8547	.6441	-.0000	179.92725	152.129108	.6441	.5006	118.2727
3	9	.4000	1.9547	.6441	.0000	179.92725	152.129108	.6441	.5006	118.2727
3	10	.4000	2.0547	.6441	.0000	179.92725	152.129108	.6441	.5006	118.2727
3	11	.4000	2.1547	.6441	0.0000	179.92725	152.129108	.6441	.5006	118.2727
4	1	.6000	0.0000	1.7836	0.0000	44.63367	55.651536	1.7836	1.6832	80.2020
4	2	.6000	.1346	1.7814	.1065	44.53166	55.579614	1.7845	1.6850	80.1223
4	3	.6000	.2693	1.7752	.2108	44.14704	55.288347	1.7877	1.6908	79.8487
4	4	.6000	.4039	1.7657	.3123	43.51038	54.770628	1.7931	1.7002	79.4411
4	5	.6000	.5386	1.7544	.4090	42.57997	53.982336	1.8015	1.7143	78.8776
4	6	.6000	.6732	1.7390	.5020	41.49522	53.054077	1.8100	1.7297	78.2131
4	6	.6000	1.6547	.6441	0.0000	179.93296	152.132558	.6441	.5005	118.2738
4	7	.6000	1.7547	.6441	-.0000	179.93296	152.132558	.6441	.5005	118.2738
4	8	.6000	1.8547	.6441	.0000	179.93296	152.132558	.6441	.5005	118.2738
4	9	.6000	1.9547	.6441	.0000	179.93296	152.132558	.6441	.5005	118.2738
4	10	.6000	2.0547	.6441	-.0000	179.93296	152.132558	.6441	.5005	118.2738
4	11	.6000	2.1547	.6441	0.0000	179.93296	152.132558	.6441	.5005	118.2738
5	1	.8000	0.0000	1.8584	0.0000	37.61577	49.212201	1.8584	1.7965	76.4359

Fig. 25. (cont)

SOLUTION SURFACE NO. 500 - TIME = 28.9964435E SECONDS (DELTA T = .05754550, NVCM = 1, CNUMS = 1.00, (20, 2), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
5	2	.8000	.1462	1.8556	.1090	37.56076	49.178536	1.8588	1.7976	76.3763
5	3	.8000	.2924	1.8481	.2165	37.30861	48.990486	1.8607	1.8020	76.1548
5	4	.8000	.4385	1.8364	.3215	36.87358	48.631677	1.8643	1.8095	75.8221
5	5	.8000	.5847	1.8226	.4228	36.16039	48.005901	1.8710	1.8220	75.3249
5	6	.8000	.7309	1.8047	.5210	35.25577	47.196082	1.8784	1.8368	74.7006
5	6	.8000	1.6547	.6440	0.0000	179.94062	152.137183	.6440	.5005	118.2752
5	7	.8000	1.7547	.6440	-.0000	179.94062	152.137183	.6440	.5005	118.2752
5	8	.8000	1.8547	.6440	.0000	179.94062	152.137183	.6440	.5005	118.2752
5	9	.8000	1.9547	.6440	-.0000	179.94062	152.137183	.6440	.5005	118.2752
5	10	.8000	2.0547	.6440	-.0000	179.94062	152.137183	.6440	.5005	118.2752
5	11	.8000	2.1547	.6440	0.0000	179.94062	152.137183	.6440	.5005	118.2752
6	1	1.0000	0.0000	1.9181	0.0000	32.42246	44.234596	1.9181	1.8935	73.2966
6	2	1.0000	.1577	1.9152	.1109	32.37947	44.209019	1.9184	1.8945	73.2418
6	3	1.0000	.3155	1.9070	.2209	32.18301	44.061170	1.9197	1.8984	73.0417
6	4	1.0000	.4732	1.8942	.3290	31.83592	43.767295	1.9226	1.9052	72.7391
6	5	1.0000	.6310	1.8790	.4343	31.23841	43.223165	1.9285	1.9172	72.2724
6	6	1.0000	.7887	1.8592	.5367	30.45887	42.495530	1.9351	1.9318	71.6755
6	6	1.0000	1.6547	.6440	0.0000	179.94731	152.141222	.6440	.5004	118.2765
6	7	1.0000	1.7547	.6440	.0000	179.94731	152.141222	.6440	.5004	118.2765
6	8	1.0000	1.8547	.6440	.0000	179.94731	152.141222	.6440	.5004	118.2765
6	9	1.0000	1.9547	.6440	.0000	179.94731	152.141222	.6440	.5004	118.2765
6	10	1.0000	2.0547	.6440	-.0000	179.94731	152.141222	.6440	.5004	118.2765
6	11	1.0000	2.1547	.6440	0.0000	179.94731	152.141222	.6440	.5004	118.2765
7	1	1.2000	0.0000	1.9676	0.0000	28.41421	40.241246	1.9676	1.9789	70.6097
7	2	1.2000	.1693	1.9646	.1127	28.36953	40.210997	1.9679	1.9801	70.5517
7	3	1.2000	.3386	1.9562	.2248	28.19273	40.072029	1.9691	1.9840	70.3551
7	4	1.2000	.5078	1.9429	.3356	27.88247	39.798801	1.9717	1.9908	70.0586
7	5	1.2000	.6771	1.9266	.4443	27.35629	39.301382	1.9772	2.0029	69.6064
7	6	1.2000	.8464	1.9055	.5501	26.67300	38.638731	1.9833	2.0174	69.0318
7	6	1.2000	1.6547	.6440	0.0000	179.94594	152.140394	.6440	.5004	118.2762
7	7	1.2000	1.7547	.6440	.0000	179.94594	152.140394	.6440	.5004	118.2762
7	8	1.2000	1.8547	.6440	.0000	179.94594	152.140394	.6440	.5004	118.2762
7	9	1.2000	1.9547	.6440	.0000	179.94594	152.140394	.6440	.5004	118.2762
7	10	1.2000	2.0547	.6440	.0000	179.94594	152.140394	.6440	.5004	118.2762
7	11	1.2000	2.1547	.6440	0.0000	179.94594	152.140394	.6440	.5004	118.2762
8	1	1.4000	0.0000	2.0096	0.0000	25.21985	36.945647	2.0096	2.0557	68.2620
8	2	1.4000	.1808	2.0068	.1144	25.16963	36.906817	2.0100	2.0571	68.1978
8	3	1.4000	.3616	1.9982	.2285	24.99772	36.763487	2.0112	2.0614	67.9960
8	4	1.4000	.5425	1.9847	.3418	24.70364	36.492815	2.0139	2.0687	67.6945
8	5	1.4000	.7233	1.9676	.4532	24.22841	36.027167	2.0192	2.0809	67.2504
8	6	1.4000	.9041	1.9453	.5616	23.62296	35.419115	2.0248	2.0954	66.6955
8	6	1.4000	1.6547	.6440	0.0000	179.94038	152.137043	.6440	.5004	118.2752
8	7	1.4000	1.7547	.6440	-.0000	179.94038	152.137043	.6440	.5004	118.2752
8	8	1.4000	1.8547	.6440	-.0000	179.94038	152.137043	.6440	.5004	118.2752
8	9	1.4000	1.9547	.6440	-.0000	179.94038	152.137043	.6440	.5004	118.2752
8	10	1.4000	2.0547	.6440	.0000	179.94038	152.137043	.6440	.5004	118.2752
8	11	1.4000	2.1547	.6440	0.0000	179.94038	152.137043	.6440	.5004	118.2752
9	1	1.6000	0.0000	2.0461	0.0000	22.61059	34.166358	2.0461	2.1257	66.1779
9	2	1.6000	.1924	2.0433	.1162	22.55543	34.119462	2.0466	2.1274	66.1072

Fig. 25. (cont)

SOLUTION SURFACE NO. 500 - TIME = 28.99644355 SECONDS (DELTA T = .05754550, NVCM = 1, CNUMS = 1.00, (20, 2), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
9	3	1.6000	.3848	2.0348	.2321	22.38412	33.968206	2.0480	2.1322	65.8973
9	4	1.6000	.5771	2.0210	.3476	22.09955	33.694880	2.0507	2.1401	65.5873
9	5	1.6000	.7695	2.0034	.4613	21.66517	33.254731	2.0558	2.1526	65.1491
9	6	1.6000	.9619	1.9801	.5716	21.12425	32.693800	2.0610	2.1670	64.6124
9	6	1.6000	1.6547	.6440	0.0000	179.93258	152.132332	.6440	.5005	118.2737
9	7	1.6000	1.7547	.6440	-.0000	179.93258	152.132332	.6440	.5005	118.2737
9	8	1.6000	1.8547	.6440	-.0000	179.93258	152.132332	.6440	.5005	118.2737
9	9	1.6000	1.9547	.6440	.0000	179.93258	152.132332	.6440	.5005	118.2737
9	10	1.6000	2.0547	.6440	.0000	179.93258	152.132332	.6440	.5005	118.2737
9	11	1.6000	2.1547	.6440	0.0000	179.93258	152.132332	.6440	.5005	118.2737
10	1	1.8000	0.0000	2.0783	0.0000	20.43660	31.781310	2.0783	2.1904	64.3038
10	2	1.8000	.2039	2.0756	.1181	20.37869	31.728703	2.0789	2.1924	64.2279
10	3	1.8000	.4078	2.0670	.2357	20.20832	31.570578	2.0804	2.1976	64.0100
10	4	1.8000	.6118	2.0531	.3530	19.93247	31.295166	2.0832	2.2061	63.6918
10	5	1.8000	.8157	2.0349	.4686	19.53324	30.877874	2.0881	2.2189	63.2597
10	6	1.8000	1.0196	2.0108	.5805	19.04665	30.358132	2.0929	2.2332	62.7398
10	6	1.8000	1.6547	.6440	0.0000	179.92792	152.129500	.6440	.5005	118.2729
10	7	1.8000	1.7547	.6440	-.0000	179.92792	152.129500	.6440	.5005	118.2729
10	8	1.8000	1.8547	.6440	.0000	179.92792	152.129500	.6440	.5005	118.2729
10	9	1.8000	1.9547	.6440	.0000	179.92792	152.129500	.6440	.5005	118.2729
10	10	1.8000	2.0547	.6440	.0000	179.92792	152.129500	.6440	.5005	118.2729
10	11	1.8000	2.1547	.6440	0.0000	179.92792	152.129500	.6440	.5005	118.2729
11	1	2.0000	0.0000	2.1071	0.0000	18.59719	29.707043	2.1071	2.2507	62.6019
11	2	2.0000	.2155	2.1043	.1199	18.53893	29.651490	2.1077	2.2529	62.5228
11	3	2.0000	.4310	2.0957	.2392	18.37146	29.489336	2.1093	2.2586	62.2987
11	4	2.0000	.6464	2.0816	.3582	18.10560	29.214569	2.1122	2.2676	61.9746
11	5	2.0000	.8619	2.0629	.4752	17.73778	28.818925	2.1169	2.2805	61.5491
11	6	2.0000	1.0774	2.0382	.5884	17.29735	28.335426	2.1214	2.2948	61.0450
11	6	2.0000	1.6547	.6440	0.0000	179.92841	152.129794	.6440	.5005	118.2730
11	7	2.0000	1.7547	.6440	-.0000	179.92841	152.129794	.6440	.5005	118.2730
11	8	2.0000	1.8547	.6440	-.0000	179.92841	152.129794	.6440	.5005	118.2730
11	9	2.0000	1.9547	.6440	-.0000	179.92841	152.129794	.6440	.5005	118.2730
11	10	2.0000	2.0547	.6440	-.0000	179.92841	152.129794	.6440	.5005	118.2730
11	11	2.0000	2.1547	.6440	0.0000	179.92841	152.129794	.6440	.5005	118.2730
12	1	2.2000	0.0000	2.1330	0.0000	17.02108	27.883243	2.1330	2.3073	61.0441
12	2	2.2000	.2270	2.1303	.1216	16.96446	27.827205	2.1337	2.3096	60.9636
12	3	2.2000	.4540	2.1215	.2426	16.80203	27.664177	2.1353	2.3157	60.7357
12	4	2.2000	.6811	2.1072	.3630	16.54782	27.393221	2.1382	2.3251	60.4084
12	5	2.2000	.9081	2.0881	.4813	16.20857	27.018521	2.1428	2.3382	59.9906
12	6	2.2000	1.1351	2.0628	.5955	15.80773	26.567110	2.1470	2.3524	59.5011
12	6	2.2000	1.6547	.6440	0.0000	179.93270	152.132432	.6440	.5004	118.2737
12	7	2.2000	1.7547	.6440	-.0000	179.93270	152.132433	.6440	.5004	118.2737
12	8	2.2000	1.8547	.6440	-.0000	179.93270	152.132433	.6440	.5004	118.2737
12	9	2.2000	1.9547	.6440	-.0000	179.93270	152.132433	.6440	.5004	118.2737
12	10	2.2000	2.0547	.6440	-.0000	179.93270	152.132433	.6440	.5004	118.2737
12	11	2.2000	2.1547	.6440	0.0000	179.93270	152.132432	.6440	.5004	118.2737
13	1	2.4000	0.0000	2.1567	0.0000	15.65648	26.265270	2.1567	2.3608	59.6091
13	2	2.4000	.2386	2.1538	.1233	15.60287	26.210640	2.1573	2.3631	59.5288
13	3	2.4000	.4771	2.1449	.2458	15.44724	26.049531	2.1589	2.3695	59.2995

Fig. 25. (cont)

SOLUTION SURFACE NO. 500 - TIME = 28.99644355 SECONDS (DELTA T = .05754550, NVCM = 1, CNJMS = 1.00, (20, 2), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
13	4	2.4000	.7157	2.1304	.3674	15.20602	25.785211	2.1618	2.3792	58.9719
13	5	2.4000	.9542	2.1108	.4868	14.89292	25.430803	2.1662	2.3924	58.5625
13	6	2.4000	1.1928	2.0850	.6019	14.52626	25.007882	2.1702	2.4065	58.0867
13	6	2.4000	1.6547	.6439	0.0000	179.93859	152.135984	.6439	.5004	118.2748
13	7	2.4000	1.7547	.6439	-.0000	179.93859	152.135984	.6439	.5004	118.2748
13	8	2.4000	1.8547	.6439	-.0000	179.93859	152.135984	.6439	.5004	118.2748
13	9	2.4000	1.9547	.6439	-.0000	179.93859	152.135984	.6439	.5004	118.2748
13	10	2.4000	2.0547	.6439	-.0000	179.93859	152.135984	.6439	.5004	118.2748
13	11	2.4000	2.1547	.6439	0.0000	179.93859	152.135984	.6439	.5004	118.2748
14	1	2.6000	0.0000	2.1783	0.0000	14.46522	24.819942	2.1783	2.4115	58.2807
14	2	2.6000	.2501	2.1753	.1248	14.41545	24.768017	2.1789	2.4138	58.2019
14	3	2.6000	.5002	2.1663	.2488	14.26784	24.611024	2.1805	2.4203	57.9734
14	4	2.6000	.7504	2.1515	.3716	14.04046	24.355569	2.1833	2.4303	57.6478
14	5	2.6000	1.0005	2.1315	.4919	13.75138	24.020806	2.1875	2.4435	57.2478
14	6	2.6000	1.2506	2.1052	.6077	13.41446	23.623374	2.1912	2.4576	56.7847
14	6	2.6000	1.6547	.6439	0.0000	179.94405	152.139205	.6439	.5004	118.2759
14	7	2.6000	1.7547	.6439	.0000	179.94405	152.139205	.6439	.5004	118.2759
14	8	2.6000	1.8547	.6439	.0000	179.94405	152.139205	.6439	.5004	118.2759
14	9	2.6000	1.9547	.6439	.0000	179.94405	152.139205	.6439	.5004	118.2759
14	10	2.6000	2.0547	.6439	.0000	179.94405	152.139205	.6439	.5004	118.2759
14	11	2.6000	2.1547	.6439	0.0000	179.94405	152.139205	.6439	.5004	118.2759
15	1	2.8000	0.0000	2.1982	0.0000	13.41706	23.520057	2.1982	2.4597	57.0452
15	2	2.8000	.2617	2.1952	.1263	13.37150	23.471611	2.1988	2.4621	56.9688
15	3	2.8000	.5233	2.1859	.2516	13.23257	23.320296	2.2003	2.4687	56.7427
15	4	2.8000	.7850	2.1708	.3754	13.01938	23.075264	2.2031	2.4788	56.4214
15	5	2.8000	1.0466	2.1505	.4966	12.75228	22.759286	2.2071	2.4919	56.0311
15	6	2.8000	1.3083	2.1238	.6131	12.44127	22.384540	2.2105	2.5060	55.5798
15	6	2.8000	1.6547	.6439	0.0000	179.94865	152.142000	.6439	.5003	118.2768
15	7	2.8000	1.7547	.6439	.0000	179.94865	152.142000	.6439	.5003	118.2768
15	8	2.8000	1.8547	.6439	.0000	179.94865	152.142000	.6439	.5003	118.2768
15	9	2.8000	1.9547	.6439	.0000	179.94865	152.142000	.6439	.5003	118.2768
15	10	2.8000	2.0547	.6439	.0000	179.94865	152.142000	.6439	.5003	118.2768
15	11	2.8000	2.1547	.6439	0.0000	179.94865	152.142000	.6439	.5003	118.2768
16	1	3.0000	0.0000	2.2165	0.0000	12.48999	22.346238	2.2165	2.5057	55.8930
16	2	3.0000	.2732	2.2134	.1276	12.44870	22.301650	2.2171	2.5080	55.8196
16	3	3.0000	.5464	2.2040	.2542	12.31873	22.157049	2.2186	2.5147	55.5973
16	4	3.0000	.8196	2.1886	.3790	12.11967	21.923448	2.2212	2.5248	55.2818
16	5	3.0000	1.0928	2.1679	.5009	11.87278	21.625516	2.2250	2.5379	54.9017
16	6	3.0000	1.3660	2.1408	.6180	11.58481	21.271525	2.2283	2.5519	54.4616
16	6	3.0000	1.6547	.6438	0.0000	179.95352	152.145039	.6438	.5003	118.2776
16	7	3.0000	1.7547	.6438	.0000	179.95352	152.145039	.6438	.5003	118.2776
16	8	3.0000	1.8547	.6438	.0000	179.95352	152.145039	.6438	.5003	118.2776
16	9	3.0000	1.9547	.6438	.0000	179.95352	152.145039	.6438	.5003	118.2776
16	10	3.0000	2.0547	.6438	.0000	179.95352	152.145039	.6438	.5003	118.2776
16	11	3.0000	2.1547	.6438	0.0000	179.95352	152.145039	.6438	.5003	118.2776
17	1	3.2000	0.0000	2.2337	0.0000	11.66280	21.277714	2.2337	2.5499	54.8123
17	2	3.2000	.2848	2.2305	.1288	11.62563	21.237063	2.2342	2.5521	54.7422
17	3	3.2000	.5695	2.2208	.2565	11.50454	21.099725	2.2356	2.5588	54.5246
17	4	3.2000	.8543	2.2052	.3823	11.31923	20.878028	2.2381	2.5689	54.2160

Fig. 25. (cont)

SOLUTION SURFACE NO. 500 - TIME = 28.99644355 SECONDS (DELTA T = .05754550, NVCM = 1, CNUMS = 1.00, (20, 2), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
17	5	3.2000	1.1390	2.1841	.5049	11.09066	20.596966	2.2417	2.5819	53.8461
17	6	3.2000	1.4238	2.1567	.6226	10.82254	20.260958	2.2448	2.5959	53.4158
17	6	3.2000	1.6547	.6438	0.0000	179.96004	152.148947	.6438	.5003	118.2789
17	7	3.2000	1.7547	.6438	.0000	179.96004	152.148947	.6438	.5003	118.2789
17	8	3.2000	1.8547	.6438	.0000	179.96004	152.148947	.6438	.5003	118.2789
17	9	3.2000	1.9547	.6438	.0000	179.96004	152.148947	.6438	.5003	118.2789
17	10	3.2000	2.0547	.6438	-.0000	179.96004	152.148947	.6438	.5003	118.2789
17	11	3.2000	2.1547	.6438	0.0000	179.96004	152.148947	.6438	.5003	118.2789
18	1	3.4000	0.0000	2.2494	0.0000	10.92756	20.309865	2.2494	2.5917	53.8042
18	2	3.4000	.2963	2.2461	.1300	10.89422	20.273038	2.2498	2.5939	53.7375
18	3	3.4000	.5926	2.2362	.2587	10.78171	20.143218	2.2511	2.6005	53.5253
18	4	3.4000	.8889	2.2203	.3853	10.60953	19.933551	2.2535	2.6106	53.2245
18	5	3.4000	1.1852	2.1989	.5085	10.39790	19.668722	2.2569	2.6234	52.8652
18	6	3.4000	1.4815	2.1712	.6268	10.14847	19.350539	2.2598	2.6373	52.4454
18	6	3.4000	1.6547	.6437	0.0000	179.96894	152.154211	.6437	.5003	118.2806
18	7	3.4000	1.7547	.6437	-.0000	179.96894	152.154211	.6437	.5003	118.2806
18	8	3.4000	1.8547	.6437	-.0000	179.96894	152.154211	.6437	.5003	118.2806
18	9	3.4000	1.9547	.6437	-.0000	179.96894	152.154211	.6437	.5003	118.2806
18	10	3.4000	2.0547	.6437	-.0000	179.96894	152.154211	.6437	.5003	118.2806
18	11	3.4000	2.1547	.6437	0.0000	179.96894	152.154211	.6437	.5003	118.2806
19	1	3.6000	0.0000	2.2648	0.0000	10.25387	19.407756	2.2648	2.6334	52.8339
19	2	3.6000	.3078	2.2614	.1310	10.22406	19.374567	2.2652	2.6354	52.7705
19	3	3.6000	.6157	2.2513	.2608	10.11976	19.252252	2.2664	2.6420	52.5640
19	4	3.6000	.9235	2.2351	.3881	9.96008	19.054503	2.2686	2.6519	52.2716
19	5	3.6000	1.2314	2.2134	.5120	9.76366	18.804465	2.2719	2.6646	51.9220
19	6	3.6000	1.5392	2.1853	.6309	9.52893	18.499550	2.2746	2.6785	51.5090
19	6	3.6000	1.6547	.6437	0.0000	179.97978	152.160790	.6437	.5002	118.2826
19	7	3.6000	1.7547	.6437	-.0000	179.97978	152.160790	.6437	.5002	118.2826
19	8	3.6000	1.8547	.6437	-.0000	179.97978	152.160790	.6437	.5002	118.2826
19	9	3.6000	1.9547	.6437	-.0000	179.97978	152.160790	.6437	.5002	118.2826
19	10	3.6000	2.0547	.6437	-.0000	179.97978	152.160790	.6437	.5002	118.2826
19	11	3.6000	2.1547	.6437	0.0000	179.97978	152.160790	.6437	.5002	118.2826
20	1	3.8000	0.0000	2.2785	0.0000	9.69812	18.639909	2.2785	2.6697	52.0288
20	2	3.8000	.3194	2.2750	.1319	9.67116	18.609777	2.2788	2.6716	51.9681
20	3	3.8000	.6388	2.2647	.2626	9.57404	18.494487	2.2799	2.6781	51.7670
20	4	3.8000	.9582	2.2483	.3905	9.42515	18.307660	2.2820	2.6879	51.4820
20	5	3.8000	1.2776	2.2263	.5150	9.24183	18.071130	2.2850	2.7005	51.1414
20	6	3.8000	1.5970	2.1979	.6345	9.01826	17.776265	2.2876	2.7144	50.7320
20	6	3.8000	1.6547	.6437	0.0000	179.99068	152.167457	.6437	.5002	118.2846
20	7	3.8000	1.7547	.6437	-.0000	179.99068	152.167457	.6437	.5002	118.2846
20	8	3.8000	1.8547	.6437	-.0000	179.99068	152.167457	.6437	.5002	118.2846
20	9	3.8000	1.9547	.6437	.0000	179.99068	152.167457	.6437	.5002	118.2846
20	10	3.8000	2.0547	.6437	.0000	179.99068	152.167457	.6437	.5002	118.2846
20	11	3.8000	2.1547	.6437	0.0000	179.99068	152.167457	.6437	.5002	118.2846
21	1	4.0000	0.0000	2.2921	0.0000	9.14238	17.872062	2.2921	2.7085	51.1546
21	2	4.0000	.3309	2.2886	.1327	9.11825	17.844986	2.2924	2.7104	51.0970
21	3	4.0000	.6619	2.2781	.2643	9.02832	17.736723	2.2934	2.7168	50.9018
21	4	4.0000	.9928	2.2615	.3930	8.89021	17.560817	2.2953	2.7265	50.6253
21	5	4.0000	1.3238	2.2391	.5180	8.71999	17.337795	2.2982	2.7389	50.2947

Fig. 25. (cont)

SOLUTION SURFACE NO. 500 - TIME = 28.99644355 SECONDS (DELTA T = .05754550, NVCM = 1, CNUMS = 1.00, (20, 2), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
21	6	4.0000	1.6547	2.2104	.6381	8.50758	17.052979	2.3007	2.7529	49.8891
21	6	4.0000	1.6547	.6436	0.0000	180.00000	152.173097	.6436	.5002	118.2863
21	7	4.0000	1.7547	.6436	.0000	180.00000	152.173097	.6436	.5002	118.2863
21	8	4.0000	1.8547	.6436	.0000	180.00000	152.173097	.6436	.5002	118.2863
21	9	4.0000	1.9547	.6436	.0000	180.00000	152.173097	.6436	.5002	118.2863
21	10	4.0000	2.0547	.6436	.0000	180.00000	152.173097	.6436	.5002	118.2863
21	11	4.0000	2.1547	.6436	0.0000	180.00000	152.173097	.6436	.5002	118.2863

***** EXPECT FILM OUTPUT FOR N= 500 *****

Fig. 25. (cont)

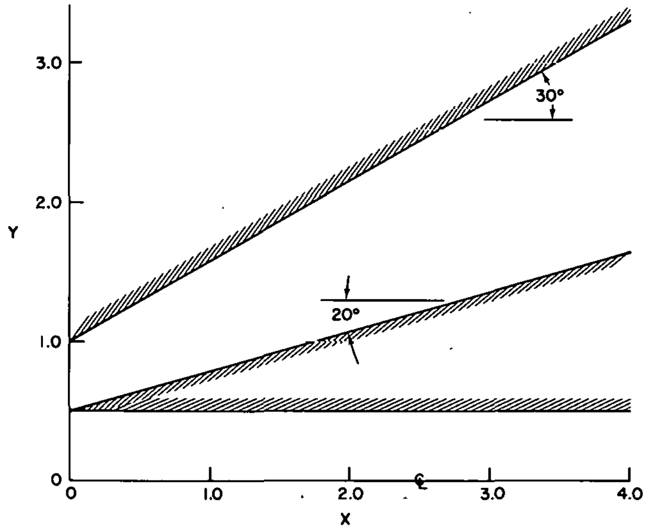

Fig. 26.
Case No. 2 geometry.

```
VNAP2 CASE 2 - SUPERSONIC SOURCE-SUBSONIC CONSTANT AREA FLOW
 $CNTRL LMAX=21,MMAX=11,NMAX=500,NPLOT=500,IUNIT=1,RGAS=0.01,
 TSTOP=100.0   $
 $IVS N1D=0,
 U(1,1,1)=21*0.67,U(1,2,1)=21*0.67,U(1,3,1)=21*0.67,U(1,4,1)=21*0.67,
 U(1,5,1)=21*0.67,UL=21*0.67,U(1,6,1)=21*1.39,U(1,7,1)=21*1.39;
 U(1,8,1)=21*1.39,U(1,9,1)=21*1.39,U(1,10,1)=21*1.39,U(1,11,1)=21*1.39,
 V(1,1,1)=21*0.0,V(1,2,1)=21*0.0,V(1,3,1)=21*0.0,V(1,4,1)=21*0.0,
 V(1,5,1)=21*0.0,VL=21*0.0,V(1,6,1)=21*0.4,V(1,7,1)=21*0.4,
 V(1,8,1)=21*0.4,V(1,9,1)=21*0.4,V(1,10,1)=21*0.4,V(1,11,1)=21*0.4,
 P(1,1,1)=21*180.0,P(1,2,1)=21*180.0,P(1,3,1)=21*180.0,P(1,4,1)=21*180.0,
 P(1,5,1)=21*180.0,PL=21*180.0,P(1,6,1)=21*81.7,P(1,7,1)=21*81.7,
 P(1,8,1)=21*81.7,P(1,9,1)=21*81.7,P(1,10,1)=21*81.7,P(1,11,1)=21*81.7,
 RO(1,1,1)=21*150.0,RO(1,2,1)=21*150.0,RO(1,3,1)=21*150.0,RO(1,4,1)=21*150.0,
 RO(1,5,1)=21*150.0,ROL=21*150.0,RO(1,6,1)=21*86.6,RO(1,7,1)=21*86.6,
 RO(1,8,1)=21*86.6,RO(1,9,1)=21*86.6,RO(1,10,1)=21*86.6,
 RO(1,11,1)=21*86.6   $
 $GEMTRY NDIM=0,NGEOM=4,XI=0.0,XE=4.0,
 YW=1.0,1.1155,1.2309,1.3464,1.4619,1.5773,1.6928,1.8083,1.9238,2.0392,
 2.1547,2.2702,2.3856,2.5011,2.6166,2.7321,2.8475,2.9630,3.0785,3.1939,
 3.3094,
 NXNY=21*-0.57735   $
 $GCBL   $
 $BC ISUPER=3,PT=213.514,TT=124.2,PE=180.0,
 UI(6)=1.3877,1.4010,1.4099,1.4142,1.4143,1.41032,
 VI(6)=0.4006,0.4853,0.5698,0.6532,0.7349,0.81425,
 PI(6)=81.7273,76.9763,72.3470,67.9110,63.7063,59.7460,
 ROI(6)=86.5775,82.9519,79.3570,75.8503,72.4653,69.2182   $
 $AVL   $
 $RVL   $
 $TURBL   $
 $DFSL NDFS=2,LDFSS=1,LDFSF=21,MDFS=6,
 YL=21*0.5,
 NXNYL=21*-0.0,
 YU=0.5,0.5577,0.6155,0.6732,0.7309,0.7887,0.8464,0.9041,0.9619,1.0196,
 1.0774,1.1351,1.1928,1.2506,1.3083,1.3660,1.4238,1.4815,1.5392,1.5970,
 1.6547,
 NXNYU=21*-0.28868   $
 $VCL   $
```

Fig. 27.
Case No. 2 data deck.

VNAP2, A COMPUTER PROGRAM FOR THE COMPUTATION OF TWO-DIMENSIONAL, TIME-DEPENDENT, COMPRESSIBLE, TURBULENT FLOW

BY MICHAEL C. CLINE, T-3 - LOS ALAMOS NATIONAL LABORATORY

PROGRAM ABSTRACT -

THE NAVIER-STOKES EQUATIONS FOR TWO-DIMENSIONAL, TIME-DEPENDENT FLOW ARE SOLVED USING THE SECOND-ORDER, MACCORMACK FINITE-DIFFERENCE SCHEME. ALL BOUNDARY CONDITIONS ARE COMPUTED USING A SECOND-ORDER, REFERENCE PLANE CHARACTERISTIC SCHEME WITH THE VISCOUS TERMS TREATED AS SOURCE FUNCTIONS. THE FLUID IS ASSUMED TO BE A PERFECT GAS. THE STEADY-STATE SOLUTION IS OBTAINED AS THE ASYMPTOTIC SOLUTION FOR LARGE TIME. THE FLOW BOUNDARIES MAY BE ARBITRARY CURVED SOLID WALLS AS WELL AS JET ENVELOPES. THE GEOMETRY MAY CONSIST OF SINGLE AND DUAL FLOWING STREAMS. TURBULENCE EFFECTS ARE MODELED WITH EITHER A MIXING-LENGTH, A TURBULENCE ENERGY EQUATION, OR A TURBULENCE ENERGY-DISSIPATION RATE EQUATIONS MODEL. THIS PROGRAM ALLOWS VARIABLE GRID SPACING AND INCLUDES OPTIONS TO SPEED UP THE CALCULATION FOR HIGH REYNOLDS NUMBER FLOWS.

JOB TITLE -

VNAP2 CASE 2 - SUPERSONIC SOURCE-SUBSONIC CONSTANT AREA FLOW

CONTROL PARAMETERS -

LMAX=21	MMAX=11	NMAX= 500	NPRINT= 0	NPLOT= 500	FDT= .90	FDT1=1.00	FDTI= .90	IPUNCH=0
IUI=1	IUO=1	IVPTS=1	NCONVI= 1	TSTOP= .10E+03	N1D= 0	TCONV=0.000	NASM=1	IUNIT=1
RSTAR=	0.000000	RSTARS=	0.000000	PLOW= .0100	ROLOW=	.000100	VDT= .25	VDT1= .25

FLUID MODEL -

THE RATIO OF SPECIFIC HEATS, GAMMA =1.4000 AND THE GAS CONSTANT, R = .0100 (FT-LBF/LBM-R)

FLOW GEOMETRY -

TWO-DIMENSIONAL, PLANAR FLOW HAS BEEN SPECIFIED

Fig. 28.
Case No. 2 output.

DUCT GEOMETRY -

A GENERAL WALL HAS BEEN SPECIFIED BY THE FOLLOWING PARAMETERS, XT= 0.0000 (IN), RT= 1.0000 (IN),

L	XP(IN)	YW(IN)	SLOPE
1	0.0000	1.0000	.5773
2	.2000	1.1155	.5773
3	.4000	1.2309	.5773
4	.6000	1.3464	.5773
5	.8000	1.4619	.5773
6	1.0000	1.5773	.5773
7	1.2000	1.6928	.5773
8	1.4000	1.8083	.5773
9	1.6000	1.9238	.5773
10	1.8000	2.0392	.5773
11	2.0000	2.1547	.5773
12	2.2000	2.2702	.5773
13	2.4000	2.3856	.5773
14	2.6000	2.5011	.5773
15	2.8000	2.6166	.5773
16	3.0000	2.7321	.5773
17	3.2000	2.8475	.5773
18	3.4000	2.9630	.5773
19	3.6000	3.0785	.5773
20	3.8000	3.1939	.5773
21	4.0000	3.3094	.5773

Fig. 28. (cont)

DUAL FLOW SPACE BOUNDARY GEOMETRY -

GENERAL BOUNDARIES HAVE BEEN SPECIFIED BY THE FOLLOWING PARAMETERS,

L	XP(IN)	YL(IN)	SLOPEL	YU(IN)	SLOPEU
1	0.0000	.5000	0.0000	.5000	.2887
2	.2000	.5000	0.0000	.5577	.2887
3	.4000	.5000	0.0000	.6155	.2887
4	.6000	.5000	0.0000	.6732	.2887
5	.8000	5000	0.0000	.7309	.2887
6	1.0000	.5000	0.0000	.7887	.2887
7	1.2000	.5000	0.0000	.8464	.2887
8	1.4000	.5000	0.0000	.9041	.2887
9	1.6000	.5000	0.0000	.9619	.2887
10	1.8000	.5000	0.0000	1.0196	.2887
11	2.0000	.5000	0.0000	1.0774	.2887
12	2.2000	.5000	0.0000	1.1351	.2887
13	2.4000	.5000	0.0000	1.1928	.2887
14	2.6000	.5000	0.0000	1.2506	.2887
15	2.8000	.5000	0.0000	1.3083	.2887
16	3.0000	.5000	0.0000	1.3660	.2887
17	3.2000	.5000	0.0000	1.4238	.2887
18	3.4000	.5000	0.0000	1.4815	.2887
19	3.6000	.5000	0.0000	1.5392	.2887
20	3.8000	.5000	0.0000	1.5970	.2887
21	4.0000	.5000	0.0000	1.6547	.2887

Fig. 28. (cont)

BOUNDARY CONDITIONS -

M	PT(PSIA)	TT(R)	THETA(DEG)	PE(PSIA)	FSQ(FT2/S2)	FSE(FT2/S3)
1	213.5140	124.20	0.00	180.00000	.0001	.1
2	213.5140	124.20	0.00	180.00000	.0001	.1
3	213.5140	124.20	0.00	180.00000	.0001	.1
4	213.5140	124.20	0.00	180.00000	.0001	.1
5	213.5140	124.20	0.00	180.00000	.0001	.1
6	213.5140	124.20	0.00	180.00000	.0001	.1
6	213.5140	124.20	0.00	180.00000	.0001	.1
7	213.5140	124.20	0.00	180.00000	.0001	.1
8	213.5140	124.20	0.00	180.00000	.0001	.1
9	213.5140	124.20	0.00	180.00000	.0001	.1
10	213.5140	124.20	0.00	180.00000	.0001	.1
11	213.5140	124.20	0.00	180.00000	.0001	.1

IINLET=0 IEXITT=0 IEX=1 ISUPER=3 DYW= .0010 IVBC=0 INBC=0 IWALL=0 IWALLO=0 ALI=0.00 ALE=0.00
ALW=0.00 NSTAG=0 NPE= 0 PEI= 0.00000

FREE-SLIP WALLS ARE SPECIFIED

ADIABATIC UPPER WALL IS SPECIFIED

ADIABATIC LOWER DUAL FLOW SPACE BOUNDARY IS SPECIFIED

ADIABATIC UPPER DUAL FLOW SPACE BOUNDARY IS SPECIFIED

ARTIFICAL VISCOSITY -

CAV=0.00 XMU= .40 XLA=1.00 PRA= .70 XRO= .60 LSS= 1 LSF=999 IDIVC=0 ISS=0 SMACH=0.00
NST= 0 SMP=1.00 SMPF=1.00 SMPT=1.00 SMPTF=1.00 NTST= 1 IAV=0 MSS= 1 MSF=999

MOLECULAR VISCOSITY -

CMU=0. (LBF-S/FT2) CLA= 0. (LBF-S/FT2) CK=0. (LBF/S-R) EMU=0.00 ELA=0.00 EK=0.00

TURBULENCE MODEL -

NO MODEL IS SPECIFIED

VARIABLE GRID PARAMETERS -

IST=0 MVCB= 0 MVCT= 0 IQS=0 NIQSS=2 NIQSF=0 NVCMI= 0 ILLQS=30 SQS= .50 CQS= .001

***** EXPECT FILM OUTPUT FOR N= 0 *****

```
N=   10,  T=   .68420880 SECONDS,  DT= .06810572 SECONDS,  NVCM =  1,  CNUMS = 1.00,  ( 2, 7),  ( 0, 0)
N=   20,  T=  1.34793476 SECONDS,  DT= .06534079 SECONDS,  NVCM =  1,  CNUMS = 1.00,  ( 4, 7),  ( 0, 0)
N=   30,  T=  1.98638646 SECONDS,  DT= .06307091 SECONDS,  NVCM =  1,  CNUMS = 1.00,  ( 6, 7),  ( 0, 0)
N=   40,  T=  2.60736599 SECONDS,  DT= .06132799 SECONDS,  NVCM =  1,  CNUMS = 1.00,  ( 9, 7),  ( 0, 0)
N=   50,  T=  3.21411644 SECONDS,  DT= .06007029 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (12, 7),  ( 0, 0)
N=   60,  T=  3.80843898 SECONDS,  DT= .05878989 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (15, 7),  ( 0, 0)
N=   70,  T=  4.39046913 SECONDS,  DT= .05748448 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (18, 7),  ( 0, 0)
N=   80,  T=  4.97345167 SECONDS,  DT= .05944419 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (20, 7),  ( 0, 0)
```

Fig. 28. (cont)

N=	90,	T=	5.57260436	SECONDS,	DT=	.05987051	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(19, 7),	(0, 0)
N=	100,	T=	6.17059463	SECONDS,	DT=	.05982064	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	110,	T=	6.76882330	SECONDS,	DT=	.05981464	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	120,	T=	7.36700424	SECONDS,	DT=	.05982018	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	130,	T=	7.96519419	SECONDS,	DT=	.05981884	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	140,	T=	8.56338521	SECONDS,	DT=	.05981901	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	150,	T=	9.16157502	SECONDS,	DT=	.05981902	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	160,	T=	9.75976499	SECONDS,	DT=	.05981899	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	170,	T=	10.35795503	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	180,	T=	10.95614503	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	190,	T=	11.55433504	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	200,	T=	12.15252505	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	210,	T=	12.75071507	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	220,	T=	13.34890508	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	230,	T=	13.94709509	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	240,	T=	14.54528510	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	250,	T=	15.14347511	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	260,	T=	15.74166512	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	270,	T=	16.33985513	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	280,	T=	16.93804515	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	290,	T=	17.53623516	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	300,	T=	18.13442517	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	310,	T=	18.73261518	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	320,	T=	19.33080519	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	330,	T=	19.92899520	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	340,	T=	20.52718522	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	350,	T=	21.12537523	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	360,	T=	21.72356524	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	370,	T=	22.32175525	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	380,	T=	22.91994526	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	390,	T=	23.51813527	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	400,	T=	24.11632528	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	410,	T=	24.71451530	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	420,	T=	25.31270531	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	430,	T=	25.91089532	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	440,	T=	26.50908533	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	450,	T=	27.10727534	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	460,	T=	27.70546535	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	470,	T=	28.30365536	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	480,	T=	28.90184538	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	490,	T=	29.50003539	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)
N=	500,	T=	30.09822540	SECONDS,	DT=	.05981900	SECONDS,	NVCM =	1,	CNUMS =	1.00,	(20, 7),	(0, 0)

Fig. 28. (cont)

MASS FLOW AND THRUST CALCULATION, N= 500

L	MF(LBM/S)	MF/MFI	T(LBF)	T/TI
1	103.66476	1.0000	108.4936	1.0000
2	104.17112	1.0049	116.7201	1.0758
3	104.38358	1.0069	122.2641	1.1269
4	104.52396	1.0083	126.4869	1.1658
5	104.62213	1.0092	129.8680	1.1970
6	104.68099	1.0098	132.6420	1.2226
7	104.73904	1.0104	135.0156	1.2445
8	104.78580	1.0108	137.0625	1.2633
9	104.81792	1.0111	138.8408	1.2797
10	104.84414	1.0114	140.4132	1.2942
11	104.86594	1.0116	141.8178	1.3072
12	104.88962	1.0118	143.0932	1.3189
13	104.90506	1.0120	144.2406	1.3295
14	104.91802	1.0121	145.2872	1.3391
15	104.93230	1.0122	146.2559	1.3481
16	104.94652	1.0124	147.1501	1.3563
17	104.94665	1.0124	147.9603	1.3638
18	104.96477	1.0125	148.7347	1.3709
19	104.95407	1.0124	149.4413	1.3774
20	105.09699	1.0138	150.3916	1.3862
21	105.08073	1.0137	151.0062	1.3918

Fig. 28. (cont)

SOLUTION SURFACE NO. 500 - TIME = 30.09822540 SECONDS (DELTA T = .05981900, NVCM = 1, CNUMS = 1.00, (20, 7), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
1	1	0.0000	0.0000	.6440	0.0000	179.93650	152.134697	.6440	.5005	118.2745
1	2	0.0000	.1000	.6440	0.0000	179.93650	152.134697	.6440	.5005	118.2745
1	3	0.0000	.2000	.6440	0.0000	179.93650	152.134697	.6440	.5005	118.2745
1	4	0.0000	.3000	.6440	0.0000	179.93650	152.134697	.6440	.5005	118.2745
1	5	0.0000	.4000	.6440	0.0000	179.93650	152.134697	.6440	.5005	118.2745
1	6	0.0000	.5000	.6440	0.0000	179.93650	152.134697	.6440	.5005	118.2745
1	6	0.0000	.5000	1.3877	.4006	81.72730	86.577500	1.4444	1.2564	94.3979
1	7	0.0000	.6000	1.4010	.4853	76.97630	82.951900	1.4827	1.3008	92.7963
1	8	0.0000	.7000	1.4099	.5698	72.34700	79.357000	1.5207	1.3460	91.1665
1	9	0.0000	.8000	1.4142	.6532	67.91100	75.850300	1.5578	1.3914	89.5329
1	10	0.0000	.9000	1.4143	.7349	63.70630	72.465300	1.5938	1.4367	87.9128
1	11	0.0000	1.0000	1.4103	.8142	59.74600	69.218200	1.6285	1.4814	86.3154
2	1	.2000	0.0000	.6440	0.0000	179.94450	152.139524	.6440	.5005	118.2760
2	2	.2000	.1000	.6440	.0000	179.94450	152.139524	.6440	.5005	118.2760
2	3	.2000	.2000	.6440	.0000	179.94450	152.139524	.6440	.5005	118.2760
2	4	.2000	.3000	.6440	.0000	179.94450	152.139524	.6440	.5005	118.2760
2	5	.2000	.4000	.6440	.0000	179.94450	152.139524	.6440	.5005	118.2760
2	6	.2000	.5000	.6440	0.0000	179.94450	152.139524	.6440	.5005	118.2760
2	6	.2000	.5577	1.5603	.4504	60.90799	69.921220	1.6240	1.4706	87.1095
2	7	.2000	.6693	1.5558	.5415	58.14545	67.677034	1.6474	1.5021	85.9161
2	8	.2000	.7808	1.5494	.6294	55.33358	65.350453	1.6723	1.5360	84.6721
2	9	.2000	.8924	1.5405	.7154	52.45968	62.925772	1.6985	1.5722	83.3676
2	10	.2000	1.0039	1.5301	.7992	49.55768	60.424814	1.7262	1.6110	82.0155
2	11	.2000	1.1155	1.5141	.8741	47.12494	58.310333	1.7483	1.6436	80.8175
3	1	.4000	0.0000	.6440	0.0000	179.95138	152.143681	.6440	.5005	118.2773
3	2	.4000	.1000	.6440	-.0000	179.95138	152.143681	.6440	.5005	118.2773
3	3	.4000	.2000	.6440	-.0000	179.95138	152.143681	.6440	.5005	118.2773
3	4	.4000	.3000	.6440	-.0000	179.95138	152.143681	.6440	.5005	118.2773
3	5	.4000	.4000	.6440	-.0000	179.95138	152.143681	.6440	.5005	118.2773
3	6	.4000	.5000	.6440	0.0000	179.95138	152.143681	.6440	.5005	118.2773
3	6	.4000	.6155	1.6730	.4830	48.48676	59.307674	1.7413	1.6276	81.7546
3	7	.4000	.7386	1.6605	.5787	46.56392	57.657919	1.7584	1.6538	80.7589
3	8	.4000	.8617	1.6462	.6700	44.59378	55.936316	1.7773	1.6823	79.7224
3	9	.4000	.9847	1.6299	.7585	42.53974	54.104554	1.7978	1.7135	78.6251
3	10	.4000	1.1078	1.6122	.8423	40.48913	52.240680	1.8190	1.7462	77.5050
3	11	.4000	1.2309	1.5902	.9181	38.68948	50.592252	1.8362	1.7746	76.4731
4	1	.6000	0.0000	.6440	0.0000	179.95091	152.143398	.6440	.5005	118.2772
4	2	.6000	.1000	.6440	.0000	179.95091	152.143398	.6440	.5005	118.2772
4	3	.6000	.2000	.6440	-.0000	179.95091	152.143398	.6440	.5005	118.2772
4	4	.6000	.3000	.6440	.0000	179.95091	152.143398	.6440	.5005	118.2772
4	5	.6000	.4000	.6440	.0000	179.95091	152.143398	.6440	.5005	118.2772
4	6	.6000	.5000	.6440	0.0000	179.95091	152.143398	.6440	.5005	118.2772
4	6	.6000	.6732	1.7563	.5070	40.07869	51.712182	1.8280	1.7549	77.5034
4	7	.6000	.8078	1.7392	.6061	38.61233	50.391387	1.8418	1.7782	76.6249
4	8	.6000	.9425	1.7201	.6999	37.13224	49.034950	1.8570	1.8036	75.7261
4	9	.6000	1.0771	1.6988	.7899	35.59385	47.596833	1.8735	1.8310	74.7820
4	10	.6000	1.2118	1.6764	.8746	34.03739	46.115423	1.8908	1.8601	73.8091
4	11	.6000	1.3464	1.6503	.9528	32.58363	44.719000	1.9056	1.8868	72.8631
5	1	.8000	0.0000	.6440	0.0000	179.94718	152.141143	.6440	.5005	118.2765

Fig. 28. (cont)

SOLUTION SURFACE NO. 500 - TIME = 30.09822540 SECONDS (DELTA T = .05981900, NVCM = 1, CNUMS = 1.00, (20, 7), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
5	2	.8000	.1000	.6440	.0000	179.94718	152.141143	.6440	.5005	118.2765
5	3	.8000	.2000	.6440	.0000	179.94718	152.141143	.6440	.5005	118.2765
5	4	.8000	.3000	.6440	.0000	179.94718	152.141143	.6440	.5005	118.2765
5	5	.8000	.4000	.6440	.0000	179.94718	152.141143	.6440	.5005	118.2765
5	6	.8000	.5000	.6440	0.0000	179.94718	152.141143	.6440	.5005	118.2765
5	6	.8000	.7309	1.8217	.5259	33.98808	45.937772	1.8961	1.8631	73.9872
5	7	.8000	.8771	1.8016	.6274	32.81371	44.833210	1.9077	1.8846	73.1906
5	8	.8000	1.0233	1.7793	.7228	31.65328	43.724326	1.9205	1.9077	72.3928
5	9	.8000	1.1695	1.7546	.8143	30.44179	42.543676	1.9344	1.9327	71.5542
5	10	.8000	1.3157	1.7289	.9006	29.18547	41.297156	1.9494	1.9598	70.6719
5	11	.8000	1.4619	1.6998	.9814	27.95671	40.064681	1.9627	1.9858	69.7789
6	1	1.0000	0.0000	.6441	0.0000	179.94134	152.137620	.6441	.5005	118.2754
6	2	1.0000	.1000	.6441	.0000	179.94134	152.137620	.6441	.5005	118.2754
6	3	1.0000	.2000	.6441	.0000	179.94134	152.137620	.6441	.5005	118.2754
6	4	1.0000	.3000	.6441	.0000	179.94134	152.137620	.6441	.5005	118.2754
6	5	1.0000	.4000	.6441	-.0000	179.94134	152.137620	.6441	.5005	118.2754
6	6	1.0000	.5000	.6441	0.0000	179.94134	152.137620	.6441	.5005	118.2754
6	6	1.0000	.7887	1.8752	.5413	29.37709	41.375081	1.9518	1.9577	71.0019
6	7	1.0000	.9464	1.8529	.6446	28.40590	40.425452	1.9619	1.9780	70.2674
6	8	1.0000	1.1041	1.8283	.7414	27.46269	39.489769	1.9729	1.9995	69.5438
6	9	1.0000	1.2619	1.8012	.8343	26.46292	38.477828	1.9851	2.0230	68.7745
6	10	1.0000	1.4196	1.7731	.9224	25.40415	37.386397	1.9987	2.0492	67.9502
6	11	1.0000	1.5773	1.7415	1.0055	24.34057	36.276694	2.0109	2.0748	67.0970
7	1	1.2000	0.0000	.6441	0.0000	179.93970	152.136628	.6441	.5005	118.2751
7	2	1.2000	.1000	.6441	-.0000	179.93970	152.136628	.6441	.5005	118.2751
7	3	1.2000	.2000	.6441	-.0000	179.93970	152.136628	.6441	.5005	118.2751
7	4	1.2000	.3000	.6441	-.0000	179.93970	152.136628	.6441	.5005	118.2751
7	5	1.2000	.4000	.6441	-.0000	179.93970	152.136628	.6441	.5005	118.2751
7	6	1.2000	.5000	.6441	0.0000	179.93970	152.136628	.6441	.5005	118.2751
7	6	1.2000	.8464	1.9201	.5543	25.77387	37.670133	1.9986	2.0420	68.4199
7	7	1.2000	1.0157	1.8962	.6590	24.95046	36.836126	2.0074	2.0615	67.7337
7	8	1.2000	1.1850	1.8698	.7569	24.15841	36.023058	2.0172	2.0818	67.0637
7	9	1.2000	1.3542	1.8409	.8514	23.30152	35.124801	2.0283	2.1047	66.3392
7	10	1.2000	1.5235	1.8109	.9413	22.38400	34.144972	2.0409	2.1304	65.5558
7	11	1.2000	1.6928	1.7774	1.0262	21.45017	33.134878	2.0523	2.1558	64.7359
8	1	1.4000	0.0000	.6441	0.0000	179.94290	152.138561	.6441	.5005	118.2757
8	2	1.4000	.1000	.6441	-.0000	179.94290	152.138561	.6441	.5005	118.2757
8	3	1.4000	.2000	.6441	-.0000	179.94290	152.138561	.6441	.5005	118.2757
8	4	1.4000	.3000	.6441	-.0000	179.94290	152.138561	.6441	.5005	118.2757
8	5	1.4000	.4000	.6441	-.0000	179.94290	152.138561	.6441	.5005	118.2757
8	6	1.4000	.5000	.6441	0.0000	179.94290	152.138561	.6441	.5005	118.2757
8	6	1.4000	.9041	1.9586	.5654	22.88582	34.595375	2.0386	2.1183	66.1528
8	7-	1.4000	1.0849	1.9333	.6713	22.17336	33.850018	2.0466	2.1371	65.5047
8	8	1.4000	1.2658	1.9056	.7704	21.48936	33.125244	2.0555	2.1568	64.8731
8	9	1.4000	1.4466	1.8753	.8664	20.73441	32.307548	2.0658	2.1794	64.1782
8	10	1.4000	1.6275	1.8437	.9579	19.92505	31.414473	2.0777	2.2048	63.4264
8	11	1.4000	1.8083	1.8086	1.0442	19.09689	30.488485	2.0883	2.2301	62.6364
9	1	1.6000	0.0000	.6440	0.0000	179.94924	152.142382	.6440	.5005	118.2769
9	2	1.6000	.1000	.6440	-.0000	179.94924	152.142382	.6440	.5005	118.2769

Fig. 28. (cont)

SOLUTION SURFACE NO. 500 - TIME = 30.09822540 SECONDS (DELTA T = .05981900, NVCM = 1, CNUMS = 1.00, (20, 7), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
9	3	1.6000	.2000	.6440	-.0000	179.94924	152.142382	.6440	.5005	118.2769
9	4	1.6000	.3000	.6440	-.0000	179.94924	152.142382	.6440	.5005	118.2769
9	5	1.6000	.4000	.6440	-.0000	179.94924	152.142382	.6440	.5005	118.2769
9	6	1.6000	.5000	.6440	0.0000	179.94924	152.142382	.6440	.5005	118.2769
9	6	1.6000	.9619	1.9921	.5751	20.52185	31.997014	2.0734	2.1881	64.1368
9	7	1.6000	1.1543	1.9657	.6821	19.89505	31.321326	2.0807	2.2065	63.5192
9	8	1.6000	1.3467	1.9370	.7824	19.29097	30.661937	2.0890	2.2259	62.9151
9	9	1.6000	1.5390	1.9055	.8798	18.61319	29.905045	2.0988	2.2484	62.2410
9	10	1.6000	1.7314	1.8725	.9727	17.89087	29.083375	2.1100	2.2737	61.5158
9	11	1.6000	1.9238	1.8361	1.0600	17.15044	28.229734	2.1201	2.2988	60.7531
10	1	1.8000	0.0000	.6440	0.0000	179.95680	152.146960	.6440	.5004	118.2783
10	2	1.8000	.1000	.6440	-.0000	179.95680	152.146960	.6440	.5004	118.2783
10	3	1.8000	.2000	.6440	-.0000	179.95680	152.146960	.6440	.5004	118.2783
10	4	1.8000	.3000	.6440	-.0000	179.95680	152.146960	.6440	.5004	118.2783
10	5	1.8000	.4000	.6440	-.0000	179.95680	152.146960	.6440	.5004	118.2783
10	6	1.8000	.5000	.6440	0.0000	179.95680	152.146960	.6440	.5004	118.2783
10	6	1.8000	1.0196	2.0216	.5836	18.55307	29.768315	2.1042	2.2526	62.3249
10	7	1.8000	1.2235	1.9944	.6917	17.99419	29.148766	2.1109	2.2707	61.7323
10	8	1.8000	1.4274	1.9647	.7931	17.45156	28.539665	2.1187	2.2899	61.1485
10	9	1.8000	1.6314	1.9322	.8919	16.83548	27.831693	2.1282	2.3126	60.4903
10	10	1.8000	1.8353	1.8981	.9859	16.18554	27.071089	2.1388	2.3378	59.7890
10	11	1.8000	2.0392	1.8605	1.0742	15.51887	26.280284	2.1483	2.3628	59.0514
11	1	2.0000	0.0000	.6439	0.0000	179.96285	152.150622	.6439	.5004	118.2794
11	2	2.0000	.1000	.6439	.0000	179.96285	152.150622	.6439	.5004	118.2794
11	3	2.0000	.2000	.6439	.0000	179.96285	152.150622	.6439	.5004	118.2794
11	4	2.0000	.3000	.6439	.0000	179.96285	152.150622	.6439	.5004	118.2794
11	5	2.0000	.4000	.6439	.0000	179.96285	152.150622	.6439	.5004	118.2794
11	6	2.0000	.5000	.6439	0.0000	179.96285	152.150622	.6439	.5004	118.2794
11	6	2.0000	1.0774	2.0480	.5912	16.88929	27.832515	2.1316	2.3127	60.6819
11	7	2.0000	1.2929	2.0200	.7003	16.38558	27.259293	2.1379	2.3305	60.1101
11	8	2.0000	1.5083	1.9895	.8029	15.89210	26.690556	2.1454	2.3498	59.5420
11	9	2.0000	1.7238	1.9562	.9029	15.32754	26.024145	2.1545	2.3726	58.8974
11	10	2.0000	1.9392	1.9210	.9978	14.73910	25.316975	2.1647	2.3977	58.2183
11	11	2.0000	2.1547	1.8824	1.0868	14.13497	24.581054	2.1737	2.4226	57.5035
12	1	2.2000	0.0000	.6439	0.0000	179.96793	152.153669	.6439	.5004	118.2804
12	2	2.2000	.1000	.6439	.0000	179.96793	152.153669	.6439	.5004	118.2804
12	3	2.2000	.2000	.6439	.0000	179.96793	152.153669	.6439	.5004	118.2804
12	4	2.2000	.3000	.6439	.0000	179.96793	152.153669	.6439	.5004	118.2804
12	5	2.2000	.4000	.6439	.0000	179.96793	152.153669	.6439	.5004	118.2804
12	6	2.2000	.5000	.6439	0.0000	179.96793	152.153669	.6439	.5004	118.2804
12	6	2.2000	1.1351	2.0718	.5981	15.46617	26.133668	2.1564	2.3690	59.1810
12	7	2.2000	1.3621	2.0430	.7082	15.00826	25.599550	2.1623	2.3867	58.6271
12	8	2.2000	1.5891	2.0118	.8119	14.55552	25.064617	2.1695	2.4061	58.0720
12	9	2.2000	1.8162	1.9777	.9129	14.03541	24.434968	2.1783	2.4291	57.4398
12	10	2.2000	2.0432	1.9416	1.0086	13.49995	23.775252	2.1880	2.4540	56.7815
12	11	2.2000	2.2702	1.9023	1.0983	12.94929	23.087544	2.1966	2.4788	56.0878
13	1	2.4000	0.0000	.6439	0.0000	179.97290	152.156657	.6439	.5004	118.2813
13	2	2.4000	.1000	.6439	.0000	179.97290	152.156657	.6439	.5004	118.2813
13	3	2.4000	.2000	.6439	.0000	179.97290	152.156657	.6439	.5004	118.2813

Fig. 28. (cont)

SOLUTION SURFACE NO. 500 - TIME = 30.09822540 SECONDS (DELTA T = .05981900, NVCM = 1, CNUMS = 1.00, (20, 7), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
13	4	2.4000	.3000	.6439	.0000	179.97290	152.156657	.6439	.5004	118.2813
13	5	2.4000	.4000	.6439	.0000	179.97290	152.156657	.6439	.5004	118.2813
13	6	2.4000	.5000	.6439	0.0000	179.97290	152.156657	.6439	.5004	118.2813
13	6	2.4000	1.1928	2.0934	.6043	14.23589	24.629139	2.1789	2.4222	57.8010
13	7	2.4000	1.4314	2.0640	.7154	13.81672	24.128653	2.1845	2.4397	57.2627
13	8	2.4000	1.6699	2.0321	.8202	13.39878	23.623069	2.1914	2.4592	56.7191
13	9	2.4000	1.9085	1.9973	.9221	12.91781	23.026793	2.1999	2.4823	56.0990
13	10	2.4000	2.1470	1.9604	1.0185	12.42841	22.409512	2.2092	2.5072	55.4604
13	11	2.4000	2.3856	1.9203	1.1087	11.92377	21.764301	2.2174	2.5319	54.7859
14	1	2.6000	0.0000	.6438	0.0000	179.97908	152.160424	.6438	.5003	118.2824
14	2	2.6000	.1000	.6438	-.0000	179.97908	152.160424	.6438	.5003	118.2824
14	3	2.6000	.2000	.6438	-.0000	179.97908	152.160424	.6438	.5003	118.2824
14	4	2.6000	.3000	.6438	-.0000	179.97908	152.160424	.6438	.5003	118.2824
14	5	2.6000	.4000	.6438	.0000	179.97908	152.160424	.6438	.5003	118.2824
14	6	2.6000	.5000	.6438	0.0000	179.97908	152.160424	.6438	.5003	118.2824
14	6	2.6000	1.2506	2.1132	.6100	13.16282	23.286586	2.1994	2.4725	56.5253
14	7	2.6000	1.5007	2.0831	.7221	12.77693	22.815539	2.2047	2.4900	56.0010
14	8	2.6000	1.7508	2.0506	.8278	12.38946	22.336234	2.2114	2.5095	55.4680
14	9	2.6000	2.0009	2.0152	.9306	11.94346	21.770772	2.2197	2.5328	54.8601
14	10	2.6000	2.2510	1.9776	1.0275	11.49447	21.191776	2.2286	2.5574	54.2403
14	11	2.6000	2.5011	1.9368	1.1182	11.02983	20.584401	2.2364	2.5821	53.5834
15	1	2.8000	0.0000	.6438	0.0000	179.98692	152.165171	.6438	.5003	118.2839
15	2	2.8000	.1000	.6438	-.0000	179.98692	152.165171	.6438	.5003	118.2839
15	3	2.8000	.2000	.6438	-.0000	179.98692	152.165171	.6438	.5003	118.2839
15	4	2.8000	.3000	.6438	-.0000	179.98692	152.165171	.6438	.5003	118.2839
15	5	2.8000	.4000	.6438	.0000	179.98692	152.165171	.6438	.5003	118.2839
15	6	2.8000	.5000	.6438	0.0000	179.98692	152.165171	.6438	.5003	118.2839
15	6	2.8000	1.3083	2.1313	.6153	12.21939	22.080390	2.2184	2.5203	55.3405
15	7	2.8000	1.5700	2.1007	.7283	11.86240	21.635351	2.2234	2.5377	54.8288
15	8	2.8000	1.8316	2.0677	.8349	11.50196	21.179933	2.2299	2.5574	54.3059
15	9	2.8000	2.0933	2.0316	.9384	11.08743	20.643063	2.2379	2.5807	53.7102
15	10	2.8000	2.3549	1.9934	1.0359	10.67390	20.098480	2.2464	2.6053	53.1080
15	11	2.8000	2.6166	1.9520	1.1270	10.24396	19.524511	2.2540	2.6299	52.4672
16	1	3.0000	0.0000	.6437	0.0000	179.99548	152.170292	.6437	.5002	118.2856
16	2	3.0000	.1000	.6437	-.0000	179.99548	152.170292	.6437	.5002	118.2856
16	3	3.0000	.2000	.6437	-.0000	179.99548	152.170292	.6437	.5002	118.2856
16	4	3.0000	.3000	.6437	-.0000	179.99548	152.170292	.6437	.5002	118.2856
16	5	3.0000	.4000	.6437	-.0000	179.99548	152.170292	.6437	.5002	118.2856
16	6	3.0000	.5000	.6437	0.0000	179.99548	152.170292	.6437	.5002	118.2856
16	6	3.0000	1.3660	2.1481	.6201	11.38500	20.991480	2.2358	2.5658	54.2363
16	7	3.0000	1.6392	2.1170	.7340	11.05352	20.569916	2.2406	2.5833	53.7364
16	8	3.0000	1.9124	2.0834	.8415	10.71752	20.136775	2.2469	2.6030	53.2236
16	9	3.0000	2.1857	2.0467	.9457	10.33154	19.626696	2.2546	2.6263	52.6402
16	10	3.0000	2.4589	2.0079	1.0436	9.94958	19.113580	2.2629	2.6507	52.0550
16	11	3.0000	2.7321	1.9659	1.1350	9.55056	18.570203	2.2701	2.6753	51.4295
17	1	3.2000	0.0000	.6437	0.0000	180.00297	152.174806	.6437	.5002	118.2870
17	2	3.2000	.1000	.6437	-.0000	180.00297	152.174806	.6437	.5002	118.2870
17	3	3.2000	.2000	.6437	-.0000	180.00297	152.174806	.6437	.5002	118.2870
17	4	3.2000	.3000	.6437	-.0000	180.00297	152.174806	.6437	.5002	118.2870

Fig. 28. (cont)

SOLUTION SURFACE NO. 500 - TIME = 30.09822540 SECONDS (DELTA T = .05981900, NVCM = 1, CNUMS = 1.00, (20, 7), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
17	5	3.2000	.4000	.6437	-.0000	180.00297	152.174806	.6437	.5002	118.2870
17	6	3.2000	.5000	.6437	0.0000	180.00297	152.174806	.6437	.5002	118.2870
17	6	3.2000	1.4238	2.1638	.6246	10.64075	20.000878	2.2521	2.6095	53.2014
17	7	3.2000	1.7085	2.1322	.7393	10.33162	19.600098	2.2567	2.6270	52.7121
17	8	3.2000	1.9933	2.0980	.8477	10.01753	19.187340	2.2628	2.6467	52.2091
17	9	3.2000	2.2780	2.0609	.9524	9.65735	18.702112	2.2703	2.6701	51.6378
17	10	3.2000	2.5628	2.0215	1.0507	9.30312	18.217098	2.2782	2.6944	51.0681
17	11	3.2000	2.8475	1.9791	1.1426	8.93058	17.699921	2.2852	2.7190	50.4555
18	1	3.4000	0.0000	.6436	0.0000	180.00750	152.177594	.6436	.5002	118.2878
18	2	3.4000	.1000	.6436	.0000	180.00750	152.177594	.6436	.5002	118.2878
18	3	3.4000	.2000	.6436	.0000	180.00750	152.177594	.6436	.5002	118.2878
18	4	3.4000	.3000	.6436	-.0000	180.00750	152.177594	.6436	.5002	118.2878
18	5	3.4000	.4000	.6436	-.0000	180.00750	152.177594	.6436	.5002	118.2878
18	6	3.4000	.5000	.6436	0.0000	180.00750	152.177594	.6436	.5002	118.2878
18	6	3.4000	1.4815	2.1781	.6288	9.97767	19.101643	2.2670	2.6510	52.2346
18	7	3.4000	1.7778	2.1460	.7443	9.68915	18.720764	2.2714	2.6684	51.7561
18	8	3.4000	2.0741	2.1114	.8534	9.39553	18.327845	2.2773	2.6882	51.2637
18	9	3.4000	2.3704	2.0737	.9586	9.05890	17.865995	2.2845	2.7115	50.7047
18	10	3.4000	2.6667	2.0338	1.0573	8.72981	17.407209	2.2922	2.7356	50.1505
18	11	3.4000	2.9630	1.9909	1.1495	8.38247	16.916345	2.2989	2.7601	49.5525
19	1	3.6000	0.0000	.6436	0.0000	180.00804	152.177928	.6436	.5001	118.2879
19	2	3.6000	.1000	.6436	.0000	180.00804	152.177928	.6436	.5001	118.2879
19	3	3.6000	.2000	.6436	-.0000	180.00804	152.177928	.6436	.5001	118.2879
19	4	3.6000	.3000	.6436	-.0000	180.00804	152.177928	.6436	.5001	118.2879
19	5	3.6000	.4000	.6436	-.0000	180.00804	152.177928	.6436	.5001	118.2879
19	6	3.6000	.5000	.6436	0.0000	180.00804	152.177928	.6436	.5001	118.2879
19	6	3.6000	1.5392	2.1922	.6328	9.37246	18.267047	2.2817	2.6922	51.3080
19	7	3.6000	1.8471	2.1597	.7490	9.10049	17.901453	2.2859	2.7096	50.8366
19	8	3.6000	2.1549	2.1246	.8588	8.82515	17.526411	2.2916	2.7294	50.3534
19	9	3.6000	2.4628	2.0864	.9645	8.51013	17.086474	2.2986	2.7527	49.8062
19	10	3.6000	2.7706	2.0461	1.0636	8.20309	16.650830	2.3060	2.7767	49.2654
19	11	3.6000	3.0785	2.0026	1.1562	7.87531	16.179224	2.3125	2.8013	48.6754
20	1	3.8000	0.0000	.6436	0.0000	180.00500	152.176053	.6436	.5002	118.2873
20	2	3.8000	.1000	.6436	-.0000	180.00500	152.176053	.6436	.5002	118.2873
20	3	3.8000	.2000	.6436	-.0000	180.00500	152.176053	.6436	.5002	118.2873
20	4	3.8000	.3000	.6436	-.0000	180.00500	152.176053	.6436	.5002	118.2873
20	5	3.8000	.4000	.6436	-.0000	180.00500	152.176053	.6436	.5002	118.2873
20	6	3.8000	.5000	.6436	0.0000	180.00500	152.176053	.6436	.5002	118.2873
20	6	3.8000	1.5970	2.2047	.6364	8.87248	17.555816	2.2947	2.7280	50.5387
20	7	3.8000	1.9164	2.1718	.7532	8.61417	17.203403	2.2987	2.7455	50.0725
20	8	3.8000	2.2358	2.1363	.8635	8.35413	16.843914	2.3043	2.7653	49.5973
20	9	3.8000	2.5551	2.0978	.9696	8.05694	16.422722	2.3110	2.7886	49.0597
20	10	3.8000	2.8745	2.0570	1.0691	7.76865	16.007691	2.3182	2.8124	48.5307
20	11	3.8000	3.1939	2.0130	1.1622	7.45564	15.550579	2.3245	2.8372	47.9445
21	1	4.0000	0.0000	.6436	0.0000	180.00000	152.173018	.6436	.5002	118.2864
21	2	4.0000	.1000	.6436	-.0000	180.00000	152.173018	.6436	.5002	118.2864
21	3	4.0000	.2000	.6436	-.0000	180.00000	152.173018	.6436	.5002	118.2864
21	4	4.0000	.3000	.6436	-.0000	180.00000	152.173018	.6436	.5002	118.2864
21	5	4.0000	.4000	.6436	-.0000	180.00000	152.173018	.6436	.5002	118.2864

Fig. 28. (cont)

SOLUTION SURFACE NO. 500 - TIME = 30.09822540 SECONDS (DELTA T = .05981900, NVCM = 1, CNUMS = 1.00, (20, 7), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
21	6	4.0000	.5000	.6436	0.0000	180.00000	152.173018	.6436	.5002	118.2864
21	6	4.0000	1.6547	2.2172	.6401	8.37249	16.844586	2.3077	2.7665	49.7044
21	7	4.0000	1.9856	2.1839	.7574	8.12785	16.505354	2.3116	2.7840	49.2437
21	8	4.0000	2.3166	2.1481	.8683	7.88311	16.161417	2.3169	2.8038	48.7774
21	9	4.0000	2.6475	2.1091	.9748	7.60375	15.758970	2.3235	2.8270	48.2503
21	10	4.0000	2.9785	2.0679	1.0746	7.33421	15.364551	2.3304	2.8507	47.7346
21	11	4.0000	3.3094	2.0234	1.1682	7.03598	14.921935	2.3365	2.8757	47.1520

***** EXPECT FILM OUTPUT FOR N= 500 *****

Fig. 28. (cont)

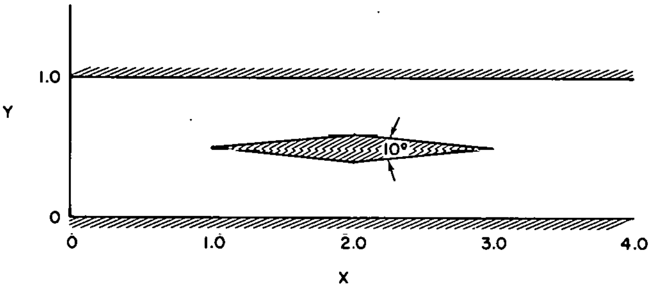

Fig. 29.
Case No. 3 geometry.

```
VNAP2 CASE 3 - SUBSONIC AIRFOIL
 $CNTRL LMAX=21,MMAX=11,NMAX=500,NPLOT=500,IUNIT=1,RGAS=0.01,
 TSTOP=100.0   $
 $IVS N1D=-2,RSTAR=0.7464   $
 $GEMTRY NDIM=0,NGEOM=1,XI=0.0,XE=4.0,RI=1.0   $
 $GCBL NGCB=1,RICB=0.0   $
 $BC PT=213.514.TT=124.2.PE=180.0   $
 $AVL   $
 $RVL   $
 $TURBL   $
 $DFSL NDFS=2,LDFSS=6,LDFSF=16,MDFS=6,
 YL(6)=0.5,0.4825,0.4650,0.4475,0.4300,0.4125,0.4300,0.4475,
 0.4650,0.4825,0.5,
 NXNYL(6)=0.04374,4*0.08749,0.0,4*-0.08749,-0.04374,
 YU(6)=0.5,0.5175,0.5350,0.5525,0.5700,0.5875,0.5700,0.5525,
 0.5350,0.5175,0.5,
 NXNYU(6)=-0.04374,4*-0.08749,0.0,4*0.08749,0.04374   $
 $VCL   $
```

Fig. 30.
Case No. 3 data deck.

VNAP2, A COMPUTER PROGRAM FOR THE COMPUTATION OF TWO-DIMENSIONAL, TIME-DEPENDENT, COMPRESSIBLE, TURBULENT FLOW

BY MICHAEL C. CLINE, T-3 - LOS ALAMOS NATIONAL LABORATORY

PROGRAM ABSTRACT -

THE NAVIER-STOKES EQUATIONS FOR TWO-DIMENSIONAL, TIME-DEPENDENT FLOW ARE SOLVED USING THE SECOND-ORDER, MACCORMACK FINITE-DIFFERENCE SCHEME. ALL BOUNDARY CONDITIONS ARE COMPUTED USING A SECOND-ORDER, REFERENCE PLANE CHARACTERISTIC SCHEME WITH THE VISCOUS TERMS TREATED AS SOURCE FUNCTIONS. THE FLUID IS ASSUMED TO BE A PERFECT GAS. THE STEADY-STATE SOLUTION IS OBTAINED AS THE ASYMPTOTIC SOLUTION FOR LARGE TIME. THE FLOW BOUNDARIES MAY BE ARBITRARY CURVED SOLID WALLS AS WELL AS JET ENVELOPES. THE GEOMETRY MAY CONSIST OF SINGLE AND DUAL FLOWING STREAMS. TURBULENCE EFFECTS ARE MODELED WITH EITHER A MIXING-LENGTH, A TURBULENCE ENERGY EQUATION, OR A TURBULENCE ENERGY-DISSIPATION RATE EQUATIONS MODEL. THIS PROGRAM ALLOWS VARIABLE GRID SPACING AND INCLUDES OPTIONS TO SPEED UP THE CALCULATION FOR HIGH REYNOLDS NUMBER FLOWS.

JOB TITLE -

VNAP2 CASE 3 - SUBSONIC AIRFOIL

CONTROL PARAMETERS -

```
LMAX=21   MMAX=11    NMAX= 500   NPRINT=    0   NPLOT= 500       FDT= .90   FDT1=1.00      FDTI= .90   IPUNCH=0
IUI=1     IUO=1      IVPTS=1     NCONVI= 1      TSTOP= .10E+03   N1D=-2     TCONV=0.000    NASM=1      IUNIT=1
RSTAR=     .746400   RSTARS=     0.0000000      PLOW= .0100      ROLOW=      .000100       VDT= .25    VDT1= .25
```

FLUID MODEL -

THE RATIO OF SPECIFIC HEATS, GAMMA =1.4000 AND THE GAS CONSTANT, R =   .0100 (FT-LBF/LBM-R)

FLOW GEOMETRY -

TWO-DIMENSIONAL, PLANAR FLOW HAS BEEN SPECIFIED

DUCT GEOMETRY -

A CONSTANT AREA DUCT HAS BEEN SPECIFIED BY XI= 0.0000 (IN), RI= 1.0000 (IN), AND XE= 4.0000 (IN)

A CYLINDRICAL CENTERBODY HAS BEEN SPECIFIED BY XICB= 0.0000 (IN), RICB= 0.0000 (IN), AND XECB= 4.0000 (IN)

Fig. 31.
Case No. 3 output.

DUAL FLOW SPACE BOUNDARY GEOMETRY -

GENERAL BOUNDARIES HAVE BEEN SPECIFIED BY THE FOLLOWING PARAMETERS,

L	XP(IN)	YL(IN)	SLOPEL	YU(IN)	SLOPEU
6	1.0000	.5000	-.0437	.5000	.0437
7	1.2000	.4825	-.0875	.5175	.0875
8	1.4000	.4650	-.0875	.5350	.0875
9	1.6000	.4475	-.0875	.5525	.0875
10	1.8000	.4300	-.0875	.5700	.0875
11	2.0000	.4125	0.0000	.5875	0.0000
12	2.2000	.4300	.0875	.5700	-.0875
13	2.4000	.4475	.0875	.5525	-.0875
14	2.6000	.4650	.0875	.5350	-.0875
15	2.8000	.4825	.0875	.5175	-.0875
16	3.0000	.5000	.0437	.5000	-.0437

Fig. 31. (cont)

)UNDARY CONDITIONS -

M	PT(PSIA)	TT(R)	THETA(DEG)	PE(PSIA)	FSQ(FT2/S2)	FSE(FT2/S3)
1	213.5140	124.20	0.00	180.00000	.0001	.1
2	213.5140	124.20	0.00	180.00000	.0001	.1
3	213.5140	124.20	0.00	180.00000	.0001	.1
4	213.5140	124.20	0.00	180.00000	.0001	.1
5	213.5140	124.20	0.00	180.00000	.0001	.1
6	213.5140	124.20	0.00	180.00000	.0001	.1
7	213.5140	124.20	0.00	180.00000	.0001	.1
8	213.5140	124.20	0.00	180.00000	.0001	.1
9	213.5140	124.20	0.00	180.00000	.0001	.1
10	213.5140	124.20	0.00	180.00000	.0001	.1
11	213.5140	124.20	0.00	180.00000	.0001	.1

```
IINLET=0  IEXITT=0  IEX=1     ISUPER=0  DYW= .0010  IVBC=0  INBC=0  IWALL=0  IWALLO=0  ALI=0.00  ALE=0.00
ALW=0.00  NSTAG=0   NPE=   0  PEI=   0.00000
```

FREE-SLIP WALLS ARE SPECIFIED

ADIABATIC UPPER WALL IS SPECIFIED

ADIABATIC LOWER CENTERBODY IS SPECIFIED

ADIABATIC LOWER DUAL FLOW SPACE BOUNDARY IS SPECIFIED

ADIABATIC UPPER DUAL FLOW SPACE BOUNDARY IS SPECIFIED

ARTIFICAL VISCOSITY -

```
CAV=0.00  XMU= .40   XLA=1.00   PRA= .70   XRO= .60    LSS= 1     LSF=999  IDIVC=0  ISS=0  SMACH=0.00
NST=   0  SMP=1.00   SMPF=1.00  SMPT=1.00  SMPTF=1.00 NTST=   1  IAV=0    MSS= 1   MSF=999
```

MOLECULAR VISCOSITY -

```
CMU=0.        (LBF-S/FT2)  CLA= 0.        (LBF-S/FT2)  CK=0.        (LBF/S-R)  EMU=0.00  ELA=0.00  EK=0.00
```

TURBULENCE MODEL -

NO MODEL IS SPECIFIED

VARIABLE GRID PARAMETERS -

```
IST=0  MVCB= 0  MVCT= 0  IQS=0  NIQSS=2  NIQSF=0  NVCMI=  0  ILLQS=30  SQS=  .50  CQS= .001
```

```
***** EXPECT FILM OUTPUT FOR N=      0 *****

N=    10,  T=   .58314928 SECONDS,  DT= .05852585 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11, 9),  ( 0, 0)
N=    20,  T=  1.16827635 SECONDS,  DT= .05844147 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11, 9),  ( 0, 0)
N=    30,  T=  1.75321496 SECONDS,  DT= .05855721 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=    40,  T=  2.33886242 SECONDS,  DT= .05854766 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=    50,  T=  2.92510260 SECONDS,  DT= .05864965 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=    60,  T=  3.51161404 SECONDS,  DT= .05866771 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11, 9),  ( 0, 0)
N=    70,  T=  4.09813240 SECONDS,  DT= .05864804 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
```

Fig. 31. (cont)

```
N=   80,  T=  4.68467521 SECONDS,  DT= .05864410 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=   90,  T=  5.27110101 SECONDS,  DT= .05864288 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  100,  T=  5.85752276 SECONDS,  DT= .05864519 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  110,  T=  6.44412855 SECONDS,  DT= .05866945 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  120,  T=  7.03093347 SECONDS,  DT= .05868448 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  130,  T=  7.61777889 SECONDS,  DT= .05867627 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  140,  T=  8.20451351 SECONDS,  DT= .05866424 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  150,  T=  8.79116556 SECONDS,  DT= .05866439 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  160,  T=  9.37790252 SECONDS,  DT= .05867774 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  170,  T=  9.96477314 SECONDS,  DT= .05868897 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  180,  T= 10.55166486 SECONDS,  DT= .05868457 SECONDS,  NVCM =  1,  CNUMS = 1 00   (11,10),  ( 0, 0)
N=  190,  T= 11.13845402 SECONDS,  DT= .05867157 SECONDS,  NVCM =  1,  CNUMS = 1 00   (11,10),  ( 0, 0)
N=  200,  T= 11.72513295 SECONDS,  DT= .05866462 SECONDS,  NVCM =  1,  CNUMS = 1 00   (11,10),  ( 0, 0)
N=  210,  T= 12.31179764 SECONDS,  DT= .05866799 SECONDS,  NVCM =  1,  CNUMS = 1 00   (11,10),  ( 0, 0)
N=  220,  T= 12.89852648 SECONDS,  DT= .05867558 SECONDS,  NVCM =  1,  CNUMS = 1.00   (11,10),  ( 0, 0)
N=  230,  T= 13.48531342 SECONDS,  DT= .05867941 SECONDS,  NVCM =  1,  CNUMS = 1.00   (11,10),  ( 0, 0)
N=  240,  T= 14.07210155 SECONDS,  DT= .05867689 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  250,  T= 14.65884622 SECONDS,  DT= .05867195 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  260,  T= 15.24555352 SECONDS,  DT= .05867004 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  270,  T= 15.83226876 SECONDS,  DT= .05867304 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  280,  T= 16.41902890 SECONDS,  DT= .05867807 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  290,  T= 17.00582680 SECONDS,  DT= .05868030 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  300,  T= 17.59262199 SECONDS,  DT= .05867823 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  310,  T= 18.17938815 SECONDS,  DT= .05867542 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  320,  T= 18.76614179 SECONDS,  DT= .05867578 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  330,  T= 19.35291453 SECONDS,  DT= .05867846 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  340,  T= 19.93970721 SECONDS,  DT= .05867932 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  350,  T= 20.52648847 SECONDS,  DT= .05867669 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  360,  T= 21.11323656 SECONDS,  DT= .05867345 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  370,  T= 21.69996588 SECONDS,  DT= .05867300 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  380,  T= 22.28670681 SECONDS,  DT= .05867517 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  390,  T= 22.87347132 SECONDS,  DT= .05867727 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  400,  T= 23.46024735 SECONDS,  DT= .05867758 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  410,  T= 24.04701862 SECONDS,  DT= .05867666 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  420,  T= 24.63378059 SECONDS,  DT= .05867592 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  430,  T= 25.22054014 SECONDS,  DT= .05867613 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  440,  T= 25.80730653 SECONDS,  DT= .05867710 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  450,  T= 26.39408256 SECONDS,  DT= .05867790 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  460,  T= 26.98086107 SECONDS,  DT= .05867763 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  470,  T= 27.56763170 SECONDS,  DT= .05867656 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  480,  T= 28.15439329 SECONDS,  DT= .05867600 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  490,  T= 28.74115613 SECONDS,  DT= .05867667 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
N=  500,  T= 29.32792830 SECONDS,  DT= .05867757 SECONDS,  NVCM =  1,  CNUMS = 1.00,  (11,10),  ( 0, 0)
```

Fig. 31. (cont)

MASS FLOW AND THRUST CALCULATION, N= 500

L	MF(LBM/S)	MF/MFI	T(LBF)	T/TI
1	97.97247	1.0000	63.0912	1.0000
2	97.98057	1.0001	63.1015	1.0002
3	97.95771	.9998	63.0786	.9998
4	98.04827	1.0008	63.1755	1.0013
5	97.71521	.9974	62.8246	.9958
6	98.04291	1.0007	63.3069	1.0034
7	97.76694	.9979	65.8924	1.0444
8	98.06552	1.0009	69.9632	1.1089
9	97.52895	.9955	73.2858	1.1616
10	98.74291	1.0079	79.8388	1.2655
11	97.73069	.9975	83.8832	1.3296
12	97.85483	.9988	78.7409	1.2480
13	97.56375	.9958	73.0532	1.1579
14	98.01990	1.0005	69.9273	1.1084
15	97.29910	.9931	65.2744	1.0346
16	98.11851	1.0015	63.4184	1.0052
17	97.79844	.9982	62.7905	.9952
18	98.01869	1.0005	63.2089	1.0019
19	97.87349	.9990	62.9277	.9974
20	97.97664	1.0000	63.1262	1.0006
21	97.89335	.9992	62.9648	.9980

Fig. 31. (cont)

SOLUTION SURFACE NO. 500 - TIME = 29.32792830 SECONDS (DELTA T = .05867757, NVCM = 1, CNUMS = 1.00, (11,10), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
1	1	0.0000	0.0000	.6441	0.0000	179.92963	152.130548	.6441	.5006	118.2732
1	2	0.0000	.1000	.6442	0.0000	179.92510	152.127812	.6442	.5006	118.2723
1	3	0.0000	.2000	.6440	0.0000	179.93776	152.135456	.6440	.5005	118.2747
1	4	0.0000	.3000	.6438	0.0000	179.95468	152.145675	.6438	.5003	118.2779
1	5	0.0000	.4000	.6438	0.0000	179.96019	152.149000	.6438	.5003	118.2789
1	6	0.0000	.5000	.6435	0.0000	179.98871	152.166225	.6435	.5000	118.2843
1	7	0.0000	.6000	6438	0.0000	179.95754	152.147399	.6438	.5003	118.2784
1	8	0.0000	.7000	6440	0.0000	179.93879	152.136077	.6440	.5005	118.2749
1	9	0.0000	.8000	6441	0.0000	179.93403	152.133201	.6441	.5005	118.2740
1	10	0.0000	.9000	6443	0.0000	179.91313	152.120584	.6443	.5007	118.2701
1	11	0.0000	1.0000	6443	0.0000	179.90967	152.118493	.6443	.5007	118.2694
2	1	.2000	0.0000	6442	0.0000	179.93163	152.132487	.6442	.5006	118.2730
2	2	.2000	.1000	6443	-.0001	179.93517	152.134726	.6443	.5007	118.2736
2	3	.2000	.2000	.6441	-.0001	179.94171	152.138639	.6441	.5005	118.2748
2	4	.2000	.3000	.6439	-.0001	179.95268	152.145235	.6439	.5004	118.2769
2	5	.2000	.4000	.6438	-.0000	179.96177	152.150775	.6438	.5003	118.2786
2	6	.2000	.5000	.6433	.0001	179.96011	152.149066	.6433	.4999	118.2788
2	7	.2000	.6000	.6439	.0001	179.95908	152.148965	.6439	.5004	118.2782
2	8	.2000	.7000	.6441	.0001	179.94327	152.139524	.6441	.5005	118.2752
2	9	.2000	.8000	.6441	.0001	179.92701	152.129707	.6441	.5005	118.2721
2	10	.2000	.9000	.6444	.0000	179.92106	152.126143	.6444	.5008	118.2710
2	11	.2000	1.0000	.6444	0.0000	179.91326	152.121433	.6444	.5008	118.2695
3	1	.4000	0.0000	.6444	0.0000	179.84896	152.081505	.6444	.5008	118.2583
3	2	.4000	.1000	.6442	-.0001	179.85251	152.083603	.6442	.5007	118.2590
3	3	.4000	.2000	.6439	-.0002	179.88470	152.103070	.6439	.5004	118.2650
3	4	.4000	.3000	.6436	-.0002	179.92894	152.129802	.6436	.5002	118.2733
3	5	.4000	.4000	.6432	-.0000	179.98890	152.166007	.6432	.4998	118.2846
3	6	.4000	.5000	.6435	.0001	180.04365	152.199378	.6435	.5000	118.2946
3	7	.4000	.6000	.6439	.0003	179.99410	152.169190	.6439	.5004	118.2855
3	8	.4000	.7000	.6439	.0004	179.92717	152.128717	.6439	.5004	118.2730
3	9	.4000	.8000	.6442	.0003	179.87968	152.100037	.6442	.5007	118.2641
3	10	.4000	.9000	.6445	.0001	179.84139	152.076893	.6445	.5009	118.2569
3	11	.4000	1.0000	.6445	0.0000	179.82842	152.069065	.6445	.5009	118.2544
4	1	.6000	0.0000	.6455	0.0000	179.83587	152.075105	.6455	.5017	118.2546
4	2	.6000	.1000	.6455	-.0009	179.85532	152.087029	.6455	.5017	118.2582
4	3	.6000	.2000	.6449	-.0016	179.91666	152.124012	.6449	.5012	118.2697
4	4	.6000	.3000	.6439	-.0018	180.02353	152.188501	.6439	.5003	118.2898
4	5	.6000	.4000	.6432	-.0015	180.13405	152.255283	.6432	.4998	118.3105
4	6	.6000	.5000	.6413	.0004	180.24852	152.323238	.6413	.4982	118.3329
4	7	.6000	.6000	.6429	.0017	180.18237	152.284226	.6429	.4995	118.3198
4	8	.6000	.7000	.6448	.0019	180.04020	152.198552	.6448	.5011	118.2930
4	9	.6000	.8000	.6452	.0017	179.90938	152.119568	.6452	.5015	118.2684
4	10	.6000	.9000	.6459	.0009	179.83438	152.074321	.6459	.5020	118.2543
4	11	.6000	1.0000	.6458	0.0000	179.78844	152.046557	.6458	.5020	118.2456
5	[illegible]	.8000	0.0000	.6483	0.0000	179.27175	151.732123	.6483	.5041	118.1502
5	2	.8000	.1000	.6477	-.0014	179.30708	151.753332	.6477	.5036	118.1569
5	3	.8000	.2000	.6456	-.0025	179.45714	151.844019	.6456	.5019	118.1852
5	4	.8000	.3000	.6422	-.0029	179.70017	151.990842	.6422	.4992	118.2309
5	5	.8000	.4000	.6370	-.0014	180.14007	152.256542	.6370	.4949	118.3135

Fig. 31. (cont)

SOLU   SURFACE NO.   500 - TIME = 29.32792830 SECONDS (DELTA T = .05867757, NVCM = 1, CNUMS = 1.00, (11,10), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
5	6	.8000	.5000	.6328	.0007	180.60786	152.540245	.6328	.4915	118.4001
5	7	.8000	.6000	.6379	.0022	180.16143	152.269756	.6379	.4956	118.3173
5	8	.8000	.7000	.6426	.0033	179.72030	152.003133	.6426	.4994	118.2346
5	9	.8000	.8000	.6461	.0027	179.44078	151.834197	.6461	.5023	118.1821
5	10	.8000	.9000	.6484	.0014	179.26674	151.729012	.6484	.5041	118.1493
5	11	.8000	1.0000	.6494	0.0000	179.18449	151.679309	.6494	.5049	118.1338
6	1	1.0000	0.0000	.6553	0.0000	178.69533	151.386362	.6553	.5098	118.0393
6	2	1.0000	.1000	.6540	-.0059	178.76880	151.431231	.6541	.5088	118.0528
6	3	1.0000	.2000	.6505	-.0116	179.07243	151.614839	.6506	.5060	118.1101
6	4	1.0000	.3000	.6449	-.0175	179.58106	151.922426	.6451	.5015	118.2058
6	5	1.0000	.4000	.6359	-.0224	180.45083	152.448094	.6363	.4942	118.3687
6	6	1.0000	.5000	.6225	-.0272	181.43926	153.039413	.6231	.4837	118.5572
6	6	1.0000	.5000	.6237	.0273	181.25469	152.934071	.6243	.4846	118.5182
6	7	1.0000	.6000	.6376	.0227	180.17262	152.279217	.6380	.4957	118.3173
6	8	1.0000	.7000	.6464	.0170	179.44205	151.838009	.6466	.5027	118.1799
6	9	1.0000	.8000	.6520	.0113	178.92438	151.525080	.6521	.5072	118.0824
6	10	1.0000	.9000	.6554	.0057	178.64862	151.358340	.6554	.5099	118.0302
6	11	1.0000	1.0000	.6568	0.0000	178.56632	151.308482	.6568	.5110	118.0147
7	1	1.2000	0.0000	.6804	0.0000	176.36226	149.972133	.6804	.5303	117.5967
7	2	1.2000	.0965	.6794	-.0108	176.43367	150.016652	.6795	.5295	117.6094
7	3	1.2000	.1930	.6771	-.0220	176.68532	150.170556	.6774	.5278	117.6564
7	4	1.2000	.2895	.6732	-.0339	177.08691	150.416278	.6741	.5250	117.7312
7	5	1.2000	.3860	.6671	-.0464	177.71065	150.796887	.6687	.5206	117.8477
7	6	1.2000	.4825	.6633	-.0580	178.03249	150.994017	.6658	.5182	117.9070
7	6	1.2000	.5175	.6597	.0577	178.16165	151.067225	.6622	.5154	117.9353
7	7	1.2000	.6140	.6688	.0441	177.45605	150.642222	.6702	.5219	117.7997
7	8	1.2000	.7105	.6741	.0321	176.89414	150.298820	.6749	.5257	117.6950
7	9	1.2000	.8070	.6777	.0210	176.50877	150.062813	.6781	.5284	117.6233
7	10	1.2000	.9035	.6799	.0105	176.32830	149.952214	.6800	.5300	117.5897
7	11	1.2000	1.0000	.6805	0.0000	176.28884	149.926980	.6805	.5304	117.5831
8	1	1.4000	0.0000	.7162	0.0000	172.60884	147.692533	.7162	.5599	116.8704
8	2	1.4000	.0930	.7164	-.0119	172.64040	147.712929	.7165	.5602	116.8756
8	3	1.4000	.1860	.7154	-.0238	172.68209	147.739069	.7158	.5596	116.8832
8	4	1.4000	.2790	.7137	-.0360	172.75828	147.786599	.7146	.5586	116.8971
8	5	1.4000	.3720	.7115	-.0490	172.81628	147.824383	.7132	.5575	116.9065
8	6	1.4000	.4650	.7096	-.0621	172.99911	147.929190	.7123	.5567	116.9472
8	6	1.4000	.5350	.7084	.0620	173.19136	148.059243	.7111	.5557	116.9744
8	7	1.4000	.6280	.7108	.0488	172.91918	147.884985	.7125	.5569	116.9282
8	8	1.4000	.7210	.7123	.0360	172.81147	147.816503	.7132	.5575	116.9095
8	9	1.4000	.8140	.7142	.0237	172.75060	147.779000	.7146	.5586	116.8979
8	10	1.4000	.9070	.7151	.0118	172.73111	147.766929	.7152	.5591	116.8943
8	11	1.4000	1.0000	.7153	0.0000	172.73544	147.769380	.7153	.5591	116.8953
9	1	1.6000	0.0000	.7537	0.0000	168.36449	145.080278	.7537	.5913	116.0492
9	2	1.6000	.0895	.7527	-.0114	168.33985	145.066195	.7528	.5906	116.0435
9	3	1.6000	.1790	.7523	-.0234	168.31983	145.053813	.7526	.5905	116.0396
9	4	1.6000	.2685	.7517	-.0363	168.29635	145.039824	.7525	.5904	116.0346
9	5	1.6000	.3580	.7505	-.0504	168.28215	145.028951	.7522	.5902	116.0335
9	6	1.6000	.4475	.7443	-.0651	167.83234	144.757824	.7472	.5865	115.9401
9	6	1.6000	.5525	.7440	.0651	167.92753	144.804303	.7469	.5862	115.9686

Fig. 31. (cont)

SOLUTION SURFACE NO. 500 - TIME = 29.32792830 SECONDS (DELTA T = .05867757, NVCM = 1, CNUMS = 1.00, (11,10), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
9	7	1.6000	.6420	.7513	.0506	168.05350	144.893533	.7530	.5909	115.9841
9	8	1.6000	.7315	.7527	.0362	168.23076	145.003094	.7535	.5913	116.0187
9	9	1.6000	.8210	.7526	.0235	168.35166	145.076201	.7530	.5908	116.0436
9	10	1.6000	.9105	.7527	.0116	168.44078	145.130086	.7528	.5906	116.0619
9	11	1.6000	1.0000	.7532	0.0000	168.53170	145.184299	.7532	.5908	116.0812
10	1	1.8000	0.0000	.7992	0.0000	164.43271	142.668435	.7992	.6291	115.2551
10	2	1.8000	.0860	.8008	-.0127	164.35470	142.621085	.8009	.6305	115.2387
10	3	1.8000	.1720	.8044	-.0264	164.04485	142.432924	.8048	.6338	115.1734
10	4	1.8000	.2580	.8104	-.0419	163.45307	142.072841	.8115	.6394	115.0488
10	5	1.8000	.3440	.8185	-.0601	162.52985	141.515621	.8207	.6472	114.8494
10	6	1.8000	.4300	.8401	-.0735	159.64350	139.715563	.8433	.6668	114.2632
10	6	1.8000	.5700	.8242	.0721	161.05797	140.605579	.8273	.6533	114.5459
10	7	1.8000	.6560	.8158	.0521	162.54924	141.513242	.8174	.6446	114.8650
10	8	1.8000	.7420	.8075	.0368	163.45810	142.067836	.8084	.6369	115.0564
10	9	1.8000	.8280	.8010	.0236	164.09803	142.459858	.8014	.6310	115.1890
10	10	1.8000	.9140	.7972	.0120	164.48294	142.696498	.7973	.6276	115.2677
10	11	1.8000	1.0000	.7947	0.0000	164.69942	142.830843	.7947	.6255	115.3108
11	1	2.0000	0.0000	.8394	0.0000	158.81698	139.164508	.8394	.6641	114.1218
11	2	2.0000	.0825	.8436	.0004	158.58480	139.025564	.8436	.6676	114.0688
11	3	2.0000	.1650	.8510	.0012	157.73375	138.495248	.8510	.6739	113.8911
11	4	2.0000	.2475	.8641	.0022	156.27318	137.583529	.8641	.6853	113.5842
11	5	2.0000	.3300	.8856	.0010	153.71971	135.978802	.8856	.7039	113.0468
11	6	2.0000	.4125	.9196	0.0000	151.21521	134.403945	.9196	.7327	112.5080
11	6	2.0000	.5875	.8996	0.0000	153.40795	135.757117	.8996	.7152	113.0018
11	7	2.0000	.6700	.8692	-.0003	156.11195	137.471736	.8692	.6894	113.5593
11	8	2.0000	.7525	.8508	.0001	158.01311	138.662628	.8508	.6736	113.9551
11	9	2.0000	.8350	.8390	.0003	159.30354	139.467588	.8390	.6635	114.2226
11	10	2.0000	.9175	.8323	-.0001	159.95114	139.870389	.8323	.6578	114.3567
11	11	2.0000	1.0000	.8289	0.0000	160.14805	139.988771	.8289	.6550	114.4006
12	1	2.2000	0.0000	.7964	0.0000	163.58950	142.152968	.7964	.6274	115.0799
12	2	2.2000	.0860	.7974	.0127	163.39534	142.042019	.7975	.6284	115.0331
12	3	2.2000	.1720	.8008	.0263	162.88680	141.733474	.8012	.6317	114.9247
12	4	2.2000	.2580	.8058	.0419	162.07130	141.238851	.8069	.6366	114.7498
12	5	2.2000	.3440	.8153	.0598	160.77808	140.452870	.8175	.6458	114.4712
12	6	2.2000	.4300	.8130	.0711	160.66941	140.396774	.8161	.6448	114.4395
12	6	2.2000	.5700	.8224	-.0719	159.94058	139.929516	.8255	.6526	114.3008
12	7	2.2000	.6560	.8126	-.0520	161.36048	140.801458	.8142	.6428	114.6014
12	8	2.2000	.7420	.8048	-.0365	162.49001	141.487087	.8056	.6353	114.8444
12	9	2.2000	.8280	.7990	-.0231	163.25273	141.951062	.7993	.6300	115.0063
12	10	2.2000	.9140	.7966	-.0119	163.57937	142.148126	.7967	.6277	115.0767
12	11	2.2000	1.0000	.7962	0.0000	163.64165	142.180759	.7962	.6273	115.0941
13	1	2.4000	0.0000	.7503	0.0000	169.11378	145.548993	.7503	.5883	116.1903
13	2	2.4000	.0895	.7505	.0113	169.08978	145.543276	.7505	.5885	116.1783
13	3	2.4000	.1790	.7500	.0229	169.09601	145.550863	.7504	.5884	116.1766
13	4	2.4000	.2685	.7496	.0348	169.11170	145.566970	.7504	.5884	116.1745
13	5	2.4000	.3580	.7462	.0494	169.39531	145.745691	.7478	.5862	116.2266
13	6	2.4000	.4475	.7436	.0651	169.55218	145.844506	.7464	.5851	116.2554
13	6	2.4000	.5525	.7444	-.0651	169.18976	145.621219	.7473	.5859	116.1848
13	7	2.4000	.6420	.7483	-.0502	169.11983	145.574651	.7500	.5881	116.1740

Fig. 31. (cont)

SOLUTION SURFACE NO. 500 - TIME = 29.32792830 SECONDS (DELTA T = .05867757, NVCM = 1, CNUMS = 1.00, (11,10), (0, 0))

L	M	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
13	8	2.4000	.7315	.7492	-.0363	169.08397	145.544534	.7501	.5881	116.1734
13	9	2.4000	.8210	.7496	-.0236	169.09189	145.544310	.7499	.5880	116.1790
13	10	2.4000	.9105	.7501	-.0118	169.06066	145.521759	.7502	.5882	116.1755
13	11	2.4000	1.0000	.7502	0.0000	169.05207	145.510546	.7502	.5883	116.1786
14	1	2.6000	0.0000	.7164	0.0000	172.36093	147.547241	.7164	.5602	116.8175
14	2	2.6000	.0930	.7164	.0118	172.36350	147.557870	.7165	.5603	116.8108
14	3	2.6000	.1860	.7155	.0237	172.39963	147.584491	.7159	.5598	116.8142
14	4	2.6000	.2790	.7140	.0362	172.45723	147.627584	.7150	.5591	116.8191
14	5	2.6000	.3720	.7118	.0491	172.60962	147.727886	.7135	.5578	116.8430
14	6	2.6000	.4650	.7094	.0621	173.11309	148.046439	.7121	.5566	116.9316
14	6	2.6000	.5350	.7086	-.0620	173.27541	148.124471	.7113	.5558	116.9796
14	7	2.6000	.6280	.7102	-.0481	172.98563	147.949866	.7118	.5563	116.9218
14	8	2.6000	.7210	.7125	-.0354	172.80929	147.832618	.7134	.5576	116.8952
14	9	2.6000	.8140	.7139	-.0232	172.68210	147.749196	.7142	.5584	116.8752
14	10	2.6000	.9070	.7149	-.0116	172.60397	147.697519	.7150	.5590	116.8631
14	11	2.6000	1.0000	.7151	0.0000	172.55756	147.663132	.7151	.5591	116.8589
15	1	2.8000	0.0000	.6782	0.0000	176.40221	150.004977	.6782	.5286	117.5976
15	2	2.8000	.0965	.6774	.0106	176.44090	150.038176	.6775	.5280	117.5973
15	3	2.8000	.1930	.6747	.0216	176.61634	150.149443	.6751	.5260	117.6270
15	4	2.8000	.2895	.6703	.0330	176.92917	150.347476	.6711	.5229	117.6802
15	5	2.8000	.3860	.6631	.0458	177.43889	150.664840	.6647	.5177	117.7706
15	6	2.8000	.4825	.6497	.0568	178.46103	151.286038	.6522	.5075	117.9627
15	6	2.8000	.5175	.6546	-.0573	177.97915	150.991957	.6571	.5115	117.8733
15	7	2.8000	.6140	.6651	-.0441	177.31139	150.580243	.6666	.5192	117.7521
15	8	2.8000	.7105	.6714	-.0321	176.83180	150.279437	.6721	.5237	117.6687
15	9	2.8000	.8070	.6759	-.0211	176.50801	150.076833	.6762	.5270	117.6118
15	10	2.8000	.9035	.6787	-.0106	176.31330	149.954547	.6787	.5290	117.5778
15	11	2.8000	1.0000	.6799	0.0000	176.23057	149.898024	.6799	.5300	117.5670
16	1	3.0000	0.0000	.6562	0.0000	178.57515	151.321025	.6562	.5105	118.0108
16	2	3.0000	.1000	.6547	.0058	178.65575	151.379994	.6547	.5094	118.0181
16	3	3.0000	.2000	.6513	.0114	178.97167	151.575645	.6514	.5066	118.0742
16	4	3.0000	.3000	.6454	.0166	179.48814	151.895740	.6456	.5019	118.1654
16	5	3.0000	.4000	.6363	.0222	180.44888	152.482391	.6367	.4947	118.3408
16	6	3.0000	.5000	.6256	.0274	181.29569	153.003296	.6262	.4862	118.4914
16	6	3.0000	.5000	.6270	-.0274	181.11328	152.871307	.6276	.4873	118.4743
16	7	3.0000	.6000	.6374	-.0225	180.06779	152.246798	.6378	.4957	118.2736
16	8	3.0000	.7000	.6465	-.0175	179.32977	151.791409	.6467	.5028	118.1422
16	9	3.0000	.8000	.6522	-.0120	178.81648	151.475165	.6523	.5074	118.0500
16	10	3.0000	.9000	.6557	-.0061	178.54805	151.308804	.6557	.5101	118.0024
16	11	3.0000	1.0000	.6572	0.0000	178.47106	151.255431	.6572	.5113	117.9932
17	1	3.2000	0.0000	.6477	0.0000	179.67603	151.986138	.6477	.5035	118.2187
17	2	3.2000	.1000	.6470	.0014	179.74705	152.039364	.6470	.5029	118.2240
17	3	3.2000	.2000	.6448	.0024	179.97430	152.181260	.6448	.5011	118.2631
17	4	3.2000	.3000	.6419	.0023	180.27633	152.371799	.6419	.4988	118.3134
17	5	3.2000	.4000	.6347	.0031	181.02479	152.830633	.6347	.4929	118.4480
17	6	3.2000	.5000	.6312	.0007	181.20769	152.940791	.6312	.4901	118.4822
17	7	3.2000	.6000	.6375	-.0024	180.65552	152.602871	.6375	.4952	118.3828
17	8	3.2000	.7000	.6423	-.0031	180.18298	152.307114	.6423	.4991	118.3024
17	9	3.2000	.8000	.6455	-.0028	179.85039	152.100038	.6455	.5017	118.2448

Fig. 31. (cont)

SOLUTION SURFACE NO. 500 - TIME = 29.32792830 SECONDS (DELTA T = .05867757, NVCM = 1, CNUMS = 1.00, (11,10), (0, 0))

L	M	X (IN)	Y (IN)	J (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT3)	VMAG (F/S)	MACH NO	T (R)
17	10	3.2000	.9000	.6473	-.0014	179.69516	152.002028	.6473	.5032	118.2189
17	11	3.2000	1.0000	.6478	0.0000	179.65637	151.971562	.6478	.5036	118.2171
18	1	3.4000	0.0000	.6469	0.0000	179.54338	151.906250	.6469	.5029	118.1935
18	2	3.4000	.1000	.6465	.0009	179.57099	151.933177	.6465	.5026	118.1908
18	3	3.4000	.2000	.6456	.0015	179.63098	151.974009	.6456	.5019	118.1985
18	4	3.4000	.3000	.6444	.0020	179.72757	152.040465	.6444	.5009	118.2104
18	5	3.4000	.4000	.6429	.0013	179.82734	152.107726	.6429	.4997	118.2237
18	6	3.4000	.5000	.6414	-.0004	179.82416	152.105322	.6414	.4985	118.2235
18	7	3.4000	.6000	.6437	-.0016	179.75819	152.060802	.6437	.5004	118.2147
18	8	3.4000	.7000	.6449	-.0018	179.67170	151.998224	.6449	.5013	118.2064
18	9	3.4000	.8000	.6458	-.0015	179.59676	151.946866	.6459	.5021	118.1971
18	10	3.4000	.9000	.6465	-.0008	179.56182	151.921583	.6465	.5026	118.1938
18	11	3.4000	1.0000	.6468	0.0000	179.55297	151.909237	.6468	.5028	118.1975
19	1	3.6000	0.0000	.6442	0.0000	179.98355	152.172301	.6442	.5006	118.2762
19	2	3.6000	.1000	.6440	.0002	179.99746	152.190994	.6440	.5004	118.2708
19	3	3.6000	.2000	.6435	.0003	180.02851	152.214380	.6435	.5001	118.2730
19	4	3.6000	.3000	.6427	.0004	180.08225	152.254990	.6427	.4995	118.2767
19	5	3.6000	.4000	.6417	.0003	180.12635	152.288732	.6417	.4987	118.2795
19	6	3.6000	.5000	.6403	-.0000	180.12693	152.288461	.6403	.4976	118.2801
19	7	3.6000	.6000	.6423	-.0003	180.08996	152.261252	.6423	.4991	118.2769
19	8	3.6000	.7000	.6431	-.0004	180.04633	152.224539	.6431	.4998	118.2768
19	9	3.6000	.8000	.6436	-.0003	180.00328	152.192431	.6436	.5002	118.2735
19	10	3.6000	.9000	.6441	-.0002	179.98525	152.177342	.6441	.5005	118.2734
19	11	3.6000	1.0000	.6442	0.0000	179.97940	152.166774	.6442	.5006	118.2777
20	1	3.8000	0.0000	.6451	0.0000	179.77278	152.045085	.6451	.5014	118.2365
20	2	3.8000	.1000	.6450	.0001	179.77647	152.057672	.6450	.5013	118.2291
20	3	3.8000	.2000	.6447	.0002	179.78302	152.066276	.6447	.5011	118.2268
20	4	3.8000	.3000	.6443	.0002	179.78979	152.078542	.6443	.5008	118.2217
20	5	3.8000	.4000	.6436	.0001	179.79404	152.088246	.6436	.5003	118.2169
20	6	3.8000	.5000	.6422	-.0000	179.79208	152.086341	.6422	.4992	118.2171
20	7	3.8000	.6000	.6440	-.0002	179.78889	152.079376	.6440	.5006	118.2204
20	8	3.8000	.7000	.6445	-.0002	179.78175	152.064717	.6445	.5009	118.2271
20	9	3.8000	.8000	.6447	-.0002	179.77840	152.056602	.6447	.5011	118.2312
20	10	3.8000	.9000	.6450	-.0001	179.77595	152.050939	.6450	.5013	118.2340
20	11	3.8000	1.0000	.6451	0.0000	179.77676	152.044388	.6451	.5014	118.2397
21	1	4.0000	0.0000	.6438	0.0000	180.00000	152.182221	.6438	.5003	118.2793
21	2	4.0000	.1000	.6437	.0001	180.00000	152.192626	.6437	.5002	118.2712
21	3	4.0000	.2000	.6435	.0002	180.00000	152.197272	.6435	.5001	118.2676
21	4	4.0000	.3000	.6433	.0002	180.00000	152.205457	.6433	.4999	118.2612
21	5	4.0000	.4000	.6427	.0001	180.00000	152.212584	.6427	.4995	118.2557
21	6	4.0000	.5000	.6414	-.0001	180.00000	152.211907	.6414	.4984	118.2562
21	7	4.0000	.6000	.6430	-.0002	180.00000	152.206858	.6430	.4997	118.2601
21	8	4.0000	.7000	.6434	-.0002	180.00000	152.196505	.6434	.5000	118.2682
21	9	4.0000	.8000	.6435	-.0002	180.00000	152.190416	.6435	.5001	118.2729
21	10	4.0000	.9000	.6437	-.0001	180.00000	152.186234	.6437	.5003	118.2761
21	11	4.0000	1.0000	.6438	0.0000	180.00000	152.179192	.6438	.5003	118.2816

***** EXPECT FILM OUTPUT FOR N= 500 *****

Fig. 31. (cont)

APPENDIX

FORTRAN LISTING OF THE VNAP2 PROGRAM

Los Alamos Identification No. LP-833

```
*COMDECK,MCC
      PARAMETER (LI=41, MI=25, LI1=42, MI1=26, MQS=9, LTS=41, MTS=25)
      COMMON /ONESID/ UD(4), VD(4), PD(4), ROD(4)
      COMMON /SOLUTN/ U(LI,MI,2), V(LI,MI,2), P(LI,MI,2), RO(LI,MI,2),
     1 UL(LI,2), VL(LI,2), PL(LI,2), ROL(LI,2)
      COMMON /CNTRLC/ LMAX, MMAX, NMAX, NPRINT, TCONV, FDT, GAMMA, RGAS,
     1 GAM1, GAM2, L1, L2, L3, M1, M2, DX, DY, DT, N, N1, N3, NASM,
     2 ICHAR, N1D, LJET, JFLAG, IERR, IUI, IUO, DXR, DYR, IB, RSTAR,
     3 RSTARS, NPLOT, G, PC, TC, LC, PLOW, ROLOW, CQ(LI,MI1), NSTART,
     4 GAM3, RG, NC, ISTOP
      COMMON /GEMTRYC/ NGEOM, XI, RI, XT, RT, XE, RE, RCI, RCT, ANGI,
     1 ANGE, YW(LI), XWI(LI), YWI(LI), NXNY(LI), NWPTS, IINT, IDIF, LT,
     2 NDIM
      COMMON /GCB/ NGCB, XICB, RICB, XTCB, RTCB, XECB, RECB, RCICB,
     1 RCTCB, ANGICB, ANGECB, YCB(LI), XCBI(LI), YCBI(LI), NXNYCB(LI),
     2 NCBPTS, IINTCB, IDIFCB
      COMMON /BCC/ PT(MI), TT(MI), THETA(MI), MASSE, MASSI, MASST,
     1 THRUST, NSTAG, NOSLIP, IEXITT, TW(LI), TCB(LI), ISUPER, DYW, IVBC
     2 , IEX, IAS, PTL, TTL, THETAL, UIL, VIL, PIL, ROIL, TL(LI), TU(LI)
     3 , IWALL, UI(MI), VI(MI), PI(MI), ROI(MI), PE(MI), PEL, PEI, NPE,
     4 INBC, IINLET, IWALLO, ALI, ALE, ALW
      COMMON /AV/ IAV, CAV, NST, SMP, LSS, XMU, XLA, PRA, XRO, QUT(LI,MI
     1 ), QVT(LI,MI), QPT(LI,MI), QROT(LI,MI), SMACH, QUTL(LI), QVTL(LI)
     2 , QPTL(LI), QROTL(LI), SMPT, US(LTS,MTS), VS(LTS,MTS), PS(LTS,MTS
     3 ), ROS(LTS,MTS), QS(LTS,MTS), ES(LTS,MTS), UILS(LTS), VLS(LTS),
     4 PLS(LTS), ROLS(LTS), QLS(LTS), ELS(LTS), NTST, NTC, LSF, IDIVC,
     5 ISS, MSS, MSF
      COMMON /RV/ CMU, CLA, CK, EMU, ELA, EK, CHECK, TMUX, TMUY, TMU1X,
     1 TMU1Y
      COMMON /TURB/ ITM, TML, Q(LI,MI,2), E(LI,MI,2), QL(LI,2), EL(LI,2)
     1 , CAL, CQMU, C1, C2, SIGQ, SIGE, QQT(LI,MI), QET(LI,MI), QQTL(LI)
     2 , QETL(LI), FSQ(MI), FSE(MI), FSQL, FSEL, CQL, LPRINT, MPRINT,
     3 QLOW, ELOW, IMLM, DEL, DELS, UBLE, YSL1, YSL2, YMIN, MMIN, IMP,
     4 BFST, CML1, CML2, PRT, STBQ, STBE
      COMMON /DFS/ YU(LI), YL(LI), NXNYU(LI), NXNYL(LI), MDFSM1, MDFSP1,
     1 MMAXD, LDFSS, LDFSF, MDFS, NDFS, IINTDFS, IDIFDFS, NLPTS, NUPTS,
     2 XLI(LI), YLI(LI), XUI(LI), YUI(LI), MDFSC
      COMMON /VC/ IST, MVCB, MVCT, XP(LI), YI(MI), IVC, VN(MI), RIND,
     1 RIND1, MVCB1, MVCT1, NVC, NN1, NN3, UU1(LI), UU2(LI), VV1(LI),
     2 VV2(LI), PP1(LI), PP2(LI), RORO1(LI), RORO2(LI), QQ1(LI), QQ2(LI)
     3 , EE1(LI), EE2(LI), DZDX(LI1), X(LI), DYDVN(MI1), Y(MI), IQSD,
     4 ILLQS, DUDYQS(LI,MQS,2), DVDYQS(LI,MQS,2), DPDYQS(LI,MQS,2), SQS,
     5 IQS, CQS, NVCM
      COMMON /MAPC/ IP, LMAP, MMAP, AL3, AL4, BE3, BE4, DE3, DE4, OM1,
     1 OM2, YP
      REAL MN3, NXNY, NXNYCB, MASSI, MASST, MASSE, LC, LC2, NXNYL, NXNYU
*DECK,VNAP2
      PROGRAM VNAP2 (ITAPE,OTAPE1,PUN1,TTY,TAPE5=ITAPE,TAPE6=OTAPE1
     1 ,TAPE8=PUN1,TAPE59=TTY)
C
C     ****************************************************************
C
C     VNAP2, A COMPUTER PROGRAM FOR THE COMPUTATION OF TWO-DIMENSIONAL,
C              TIME-DEPENDENT, COMPRESSIBLE, TURBULENT FLOW
C
C                        BY MICHAEL C. CLINE, T-3
C                     LOS ALAMOS NATIONAL LABORATORY
C
C     ****************************************************************
C
C                            PROGRAM ABSTRACT
C
```

```
C           THE NAVIER-STOKES EQUATIONS FOR TWO-DIMENSIONAL, TIME-
C      DEPENDENT FLOW ARE SOLVED USING THE SECOND-ORDER, MACCORMACK
C      FINITE-DIFFERENCE SCHEME.  ALL BOUNDARY CONDITIONS ARE COMPUTED
C      USING A SECOND-ORDER, REFERENCE PLANE CHARACTERISTIC SCHEME
C      WITH THE VISCOUS TERMS TREATED AS SOURCE FUNCTIONS.  THE FLUID
C      IS ASSUMED TO BE A PERFECT GAS.  THE STEADY-STATE SOLUTION IS
C      OBTAINED AS THE ASYMPTOTIC SOLUTION FOR LARGE TIME.  THE FLOW
C      BOUNDARIES MAY BE ARBITRARY CURVED SOLID WALLS AS WELL AS FREE
C      JET ENVELOPES.  THE GEOMETRY MAY CONSIST OF SINGLE AND DUAL
C      FLOWING STREAMS.  TURBULENCE EFFECTS ARE MODELED WITH EITHER
C      A MIXING-LENGTH, A TURBULENCE ENERGY EQUATION, OR A TURBULENCE
C      ENERGY-DISSIPATION RATE EQUATIONS MODEL.  THIS PROGRAM ALLOWS
C      VARIABLE GRID SPACING AND INCLUDES OPTIONS TO SPEED UP THE
C      CALCULATION FOR HIGH REYNOLDS NUMBER FLOWS.
C
       DIMENSION TITLE(10)
*CALL,MCC
       NAMELIST /CNTRL/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,FDTI,FDT1,VDT
      1 ,VDT1,GAMMA,RGAS,TSTOP,IUI,IUO,IPUNCH,NPLOT,LPP1,MPP1,LPP2,MPP2
      2 ,LPP3,MPP3,NASM,NAME,NCONVI,IUNIT,PLOW,ROLOW,IVPT3
       NAMELIST /IVS/ N1D,U,V,P,RO,Q,E,UL,VL,PL,ROL,QL,EL,RSTAR,RSTARS
      1 ,NSTART,TSTART
       NAMELIST /GEMTRY/ NDIM,NGEOM,XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XWI,YWI
      1 ,NWPTS,IINT,IDIF,YW,NXNY,JFLAG,LJET
       NAMELIST /GCBL/ NGCB,RICB,RTCB,RCICB,RCTCB,ANGICB,ANGECB,XCBI,YCBI
      1 ,NCBPTS,IINTCB,IDIFCB,YCB,NXNYCB
       NAMELIST /BC/ NSTAG,PT,TT,THETA,PTL,TTL,THETAL,PE,PEL,PEI,NPE,UI
      1 ,VI,PI,ROI,UIL,VIL,PIL,ROIL,TW,TCB,TL,TU,ISUPER,INBC,IWALL,IWALLO
      2 ,IINLET,IEXITT,IEX,IVBC,NOSLIP,DYW,IAS,ALI,ALE,ALW
       NAMELIST /AVL/ CAV,XMU,XLA,PRA,XRO,LSS,LSF,MSS,MSF,IDIVC,ISS,SMACH
      1 ,NST,SMP,SMPF,SMPT,SMPTF,NTST,IAV
       NAMELIST /RVL/ CMU,EMU,CLA,ELA,CK,EK
       NAMELIST /TURBL/ ITM,IMLM,CML1,CML2,CAL,COL,COMU,C1,C2,SIGQ,SIGE
      1 ,BFST,FSQ,FSE,FSQL,FSEL,QLOW,ELOW,LPRINT,MPRINT,PRT,STBQ,STBE
       NAMELIST /DFSL/ MDFS,LDFSS,LDFSF,NDFS,YU,YL,NXNYU,NXNYL,XUI,XLI
      1 ,YUI,YLI,NUPTS,NLPTS,IINTDFS,IDIFDFS
       NAMELIST /VCL/ IST,XP,YI,MVCB,MVCT,NVCMI,IQS,NIQSS,NIQSF,CQS,ILLQS
      1 ,SQS
C
C      SET THE ARRAY SIZES FOR SPECIFYING THE INPUT DEFAULT VALUES
C
       LD=LI
       MD=MI
C
C      SET DEFAULT VALUES
C
    10 TCONV=0.0
       TSTART=0.0
       THETA(1)=0.0
       CAV=0.0
       XMU=0.4
       XLA=1.0
       PRA=0.7
       XRO=0.6
       LSS=1
       LSF=999
       MSS=1
       MSF=999
       SMACH=0.0
       IDIVC=0
       ISS=0
       TC=0.0
       CMU=0.0
       CLA=0.0
       CK=0.0
       EMU=0.0
       ELA=0.0
       EK=0.0
       RSTAR=0.0
       RICB=0.0
       RTCB=0.0
       RSTARS=0.0
```

```
      PT(1)=0.0
      TT(1)=0.0
      DELSMP=0.0
      DSMPT=0.0
      CML1=0.0
      CML2=0.0
      PTL=0.0
      TTL=0.0
      THETAL=0.0
      UIL=0.0
      VIL=0.0
      PIL=0.0
      ROIL=0.0
      PE(1)=14.7
      PE(2)=-1.0
      PEL=0.0
      PEI=0.0
      NPE=0
      TMUX=0.0
      TMUY=0.0
      BFST=0.0
      COL=0.0
      TMU1X=0.0
      TMU1Y=0.0
      ALI=0.0
      ALE=0.0
      ALW=0.0
      FDTI=0.0
      TSTOP=1.0
      CNUMS=1.0
      SMP=1.0
      SMPF=1.0
      SMPT=1.0
      SMPTF=1.0
      FDT1=1.0
      FDT=0.9
      VDT=0.25
      VDT1=0.25
      NASM=1
      N1D=1
      NDIM=1
      IEX=1
      NCONVI=1
      IUI=1
      IUO=1
      IVPTS=1
      NVCM=1
      NVCM1=1
      IMLM=1
      NTST=1
      NSTAG=0
      NAME=0
      IPUNCH=0
      NGCB=0
      NMAX=0
      NPRINT=0
      NST=0
      N-0
      IERR=0
      JFLAG=0
      ISUPER-0
      NVCMI=0
      IUNIT=0
      NOSLIP=0
      IINLET=0
      IEXITT=0
      NSTART=0
      ITM=0
      IAS=0
      IVC=0
      MVCB=0
      MVCT=0
```

```
      IWALL=0
      IWALLO=0
      IB=0
      LDFSS=0
      LDFSF=0
      MDFS=0
      LPRINT=0
      MPRINT=0
      IVBC=0
      INBC=0
      LPP1=0
      IQS=0
      NIQSF=0
      IAV=0
      IST=0
      LDUF=0
      MDUF=0
      NTC=0
      IINT=2
      IDIF=2
      IINTCB=2
      IDIFCB=2
      IINTDFS=2
      IDIFDFS=2
      ILLQS=30
      GAMMA=1.4
      RGAS=53.35
      NPLOT=-1
      G=32.174
      PC=144.0
      LC=12.0
      PLOW=0.01
      ROLOW=0.0001
      DYW=0.001
      CAL=1.0
      CQMU=0.09
      C1=1.44
      C2=1.8
      SIGQ=1.0
      SIGE=1.3
      SQS=0.5
      CQS=0.001
      NIQSS=2
      FSQL=0.0001
      QLOW=0.0001
      FSEL=0.1
      ELOW=0.1
      PRT=0.9
      STBQ=0.0
      STBE=0.0
      ISTOP=0
      DO 20 M=1,MD
      UI(M)=0.0
      VI(M)=0.0
      PI(M)=0.0
      ROI(M)=0.0
      FSQ(M)=0.0001
      FSE(M)=0.1
   20 CONTINUE
      DO 30 L=1,LD
      YCB(L)=0.0
      YL(L)=0.0
      YU(L)=0.0
      NXNYCB(L)=0.0
      NXNYL(L)=0.0
      NXNYU(L)=0.0
      QL(L,1)=0.0
      EL(L,1)=0.0
      QL(L,2)=0.0
      EL(L,2)=0.0
      UL(L,1)=0.0
      VL(L,1)=0.0
```

```
      PL(L,1)=0.0
      ROL(L,1)=0.0
      PL(L,2)=0.0
      ROL(L,2)=0.0
      TW(L)=-1.0
      TCB(L)=-1.0
      TL(L)=-1.0
      TU(L)=-1.0
      DO 30 M=1,MD
      Q(L,M,1)=0.0
      E(L,M,1)=0.0
      Q(L,M,2)=0.0
      E(L,M,2)=0.0
   30 CONTINUE
C
C     READ IN INPUT DATA
C
      READ (5,1370) TITLE
      IF (EOF(5)) 40,50
   40 CALL EXIT
   50 READ (5,CNTRL)
      READ (5,IVS)
      READ (5,GEMTRY)
      READ (5,GCBL)
      READ (5,BC)
      READ (5,AVL)
      READ (5,RVL)
      READ (5,TURBL)
      READ (5,DFSL)
      READ (5,VCL)
      IF (NAME.EQ.0) GO TO 60
      WRITE (6,CNTRL)
      WRITE (6,IVS)
      WRITE (6,GEMTRY)
      WRITE (6,GCBL)
      WRITE (6,BC)
      WRITE (6,AVL)
      WRITE (6,RVL)
      WRITE (6,TURBL)
      WRITE (6,DFSL)
      WRITE (6,VCL)
C
C     PRINT INPUT DATA
C
   60 WRITE (6,1380)
      WRITE (6,1410)
      WRITE (6,1400)
      WRITE (6,1420)
      WRITE (6,1430)
      WRITE (6,1390)
      WRITE (6,1440) TITLE
      WRITE (6,1390)
      WRITE (6,1450)
      NPRIND=ABS(FLOAT(NPRINT))
      IF (FDTI.EQ.0.0) FDTI=FDT
      WRITE (6,1460) LMAX,MMAX,NMAX,NPRIND,NPLOT,FDT,FDT1,FDTI,IPUNCH
     1 ,IUI,IUO,IVPTS,NCONVI,TSTOP,N1D,TCONV,NASM,IUNIT,RSTAR,RSTARS
     2 ,PLOW,ROLOW,VDT,VDT1
      WRITE (6,1390)
      IF (IUI.EQ.1) WRITE (6,1470) GAMMA,RGAS
      IF (IUI.EQ.2) WRITE (6,1480) GAMMA,RGAS
      WRITE (6,1390)
      WRITE (6,1490)
      IF (NDIM.EQ.0) WRITE (6,1500)
      IF (NDIM.EQ.1) WRITE (6,1510)
C
C     CALCULATE THE GEOMETRY RADIUS AND SLOPE
C
      L1=LMAX-1
      L2=LMAX-2
      L3=LMAX-3
      M1=MMAX-1
```

```
      M2=MMAX-2
      MDFSM1=MDFS-1
      MDFSP1=MDFS+1
      MMAXD=MMAX-MDFS
      CHECK=ABS(CMU)+ABS(CLA)+ABS(CK)
      IF (NGEOM.NE.3) GO TO 70
      XI=XWI(1)
      XE=XWI(NWPTS)
   70 DX=(XE-XI)/FLOAT(L1)
      DY=1.0/FLOAT(M1)
      IF (IST.NE.0) GO TO 90
      DO 80 L=1,LMAX
      XP(L)=XI+FLOAT(L-1)*DX
   80 CONTINUE
   90 XP(1)=XI
      XP(LMAX)=XE
      WRITE (6,1390)
      CALL GEOM
      IF (IERR.NE.0) GO TO 10
      XICB=XI
      XECB=XE
      IF (NGCB.EQ.0.AND.MDFS.EQ.0) GO TO 140
      IF (NGCB.NE.0) CALL GEOMCB
      IF (IERR.NE.0) GO TO 10
      IF (MDFS.NE.0) CALL GEOMLU
      IF (IAS.EQ.0) GO TO 110
      IF (LDFSF.EQ.LMAX) GO TO 100
      NXNYL(LDFSF)=0.5*(NXNYL(LDFSF)+NXNYU(LDFSF))
      NXNYU(LDFSF)=NXNYL(LDFSF)
  100 IF (LDFSS.EQ.1) GO TO 110
      NXNYL(LDFSS)=0.5*(NXNYL(LDFSS)+NXNYU(LDFSS))
      NXNYU(LDFSS)=NXNYL(LDFSS)
  110 LT=1
      YO=YW(1)-YU(1)+YL(1)-YCB(1)
      DO 130 L=1,LMAX
      IF (NDIM.EQ.0) YY=YW(L)-YU(L)+YL(L)-YCB(L)
      IF (NDIM.EQ.1) YY=YW(L)**2-YU(L)**2+YL(L)**2-YCB(L)**2
      IF (YY.GT.0.0) GO TO 120
      WRITE (6,1610)
      GO TO 10
  120 IF (YY.LT.YO) LT=L
      IF (LT.EQ.L) YO=YY
  130 CONTINUE
C
C     CONTINUE SET UP AND PRINTING OF INPUT DATA
C
  140 GAM1=GAMMA/(GAMMA-1.0)
      GAM2=(GAMMA-1.0)/2.0
      GAM3=(GAMMA+1.0)/(GAMMA-1.0)
      IF (PE(2).NE.-1.0) GO TO 160
      DO 150 M=2,MMAX
      PE(M)=PE(1)
  150 CONTINUE
      PEL=PE(1)
  160 IF (MDFS.NE.0.AND.LDFSF.NE.LMAX) PEL=PE(MDFS)
      IF (NSTAG.NE.0) GO TO 180
      DO 170 M=2,MMAX
      PT(M)=PT(1)
      TT(M)=TT(1)
      THETA(M)=THETA(1)
  170 CONTINUE
      PTL=PT(1)
      TTL=TT(1)
      THETAL=THETA(1)
  180 IF (ISUPER.NE.3) GO TO 190
      PT(MDFS)=PTL
      TT(MDFS)=TTL
      THETA(MDFS)=THETAL
  190 IF (ISUPER.NE.2) GO TO 200
      PI(MDFS)=PIL
  200 WRITE (6,1380)
      IF (IUI.EQ.1) WRITE (6,1580)
```

```
      IF (IUI.EQ.2) WRITE (6,1590)
      DO 210 M=1,MMAX
      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) WRITE (6,1600) M,PTL,TTL,THETAL,PEL
     1 ,FSQL,FSEL
      WRITE (6,1600) M,PT(M),TT(M),THETA(M),PE(M),FSQ(M),FSE(M)
  210 CONTINUE
      WRITE (6,2020) IINLET,IEXITT,IEX,ISUPER,DYW,IVBC,INBC,IWALL,IWALLO
     1 ,ALI,ALE,ALW,NSTAG,NPE,PEI
      IF (NOSLIP.EQ.0) WRITE (6,1840)
      IF (NOSLIP.NE.0) WRITE (6,1850)
      WRITE (6,1390)
      IF (TW(1).LT.0.0.AND.IWALL.EQ.0) WRITE (6,1890)
      IF (TW(1).GE.0.0) WRITE (6,1900)
      WRITE (6,1390)
      IF (TCB(1).LT.0.0.AND.NGCB.NE.0) WRITE (6,1910)
      IF (TCB(1).GE.0.0) WRITE (6,1940)
      IF (MDFS.EQ.0) GO TO 220
      IF (TCB(1).GE.0.0.OR.NGCB.NE.0) WRITE (6,1390)
      IF (TL(1).LT.0.0) WRITE (6,1920)
      IF (TL(1).GE.0.0) WRITE (6,1950)
      WRITE (6,1390)
      IF (TL(1).LT.0.0) WRITE (6,1930)
      IF (TU(1).GE.0.0) WRITE (6,1960)
  220 WRITE (6,1390)
      IF (SMP.LT.0.0) SMP=0.0
      IF (SMPF.LT.0.0) SMPF=0.0
      IF (SMP.GT.1.0) SMP=1.0
      IF (SMPF.GT.1.0) SMPF=1.0
      WRITE (6,1830) CAV,XMU,XLA,PRA,XRO,LSS,LSF,IDIVC,ISS,SMACH,NST,SMP
     1 ,SMPF,SMPT,SMPTF,NTST,IAV,MSS,MSF
      WRITE (6,1390)
      IF (CML1.NE.0.0.OR.CML2.NE.0.0) GO TO 230
      IF (NDIM.EQ.0) CML1=0.125
      IF (NDIM.EQ.0) CML2=0.125
      IF (NDIM.NE.0) CML1=0.11
      IF (NDIM.NE.0) CML2=0.11
  230 IF (CQL.NE.0.0) GO TO 240
      CQL=17.2
      IF (NDIM.NE.0) CQL=CQL*0.625/0.875
  240 IF (IUI.EQ.1) WRITE (6,1860) CMU,CLA,CK,EMU,ELA,EK
      IF (IUI.EQ.2) WRITE (6,1870) CMU,CLA,CK,EMU,ELA,EK
      WRITE (6,1390)
      IF (ITM.EQ.0) WRITE (6,1970)
      IF (ITM.EQ.1) WRITE (6,1980) CAL,IMLM,CML1,CML2,PRT
      IF (ITM.EQ.2) WRITE (6,1990)
      IF (ITM.EQ.2) WRITE (6,2000) CAL,CQL,CQMU,IMLM,CML1,CML2,PRT
      IF (ITM.EQ.3) WRITE (6,2010)
      IF (ITM.EQ.3) WRITE (6,2030) CAL,CQMU,C1,C2,SIGQ,SIGE,BEST,PRT
     1 ,STBQ,STBE
      WRITE (6,1390)
      WRITE (6,2040) IST,MVCB,MVCT,IQS,NIQSS,NIQSF,NVCMI,ILLQS,SQS,CQS
C
C     CHECK THE WALL OPTIONS
C
      IVCE=0
      IF (JFLAG.EQ.0) GO TO 250
      IF (LJET.LE.2.OR.LJET.GE.L1) IVCE=1
      IF (NOSLIP.NE.0) IVCE=1
      IF (IWALL.NE.0) IVCE=1
      IF (IVCE.EQ.0) GO TO 250
      WRITE (6,2150)
      GO TO 10
  250 IF (ISUPER.GE.0) GO TO 260
      WRITE (6,2140)
      GO TO 10
C
C     CHECK MIXING-LENGTH TURBULENCE MODEL
C
  260 IF (ITM.NE.1) GO TO 300
      IF (MDFS.NE.0) GO TO 280
      IF (IMLM.EQ.1) GO TO 270
      IF (NOSLIP.EQ.0) IVCE=1
      IF (NGCB.NE.0.AND.IWALL.EQ.0) IVCE=1
      IF (NGCB.EQ.0.AND.IWALL.NE.0) IVCE=1
```

```
      GO TO 290
  270 IF (NOSLIP.NE.0) IVCE=1
      IF (NGCB.EQ.0.AND.IWALL.NE.0) IVCE=0
      GO TO 290
  280 IF (NGCB.NE.0.OR.IWALL.EQ.0) IVCE=1
  290 IF (IVCE.EQ.0) GO TO 300
      WRITE (6,2050)
      GO TO 10
C
C     CHECK THE DUAL FLOW SPACE AND VARIABLE GRID PARAMETERS
C
  300 IF (MVCB.NE.0.AND.MVCT.NE.0) IVC=1
      IP=-1
      CALL MAP
      IF (IERR.NE.0) GO TO 10
      MDFSC=0
      IF (MDFS.GE.MVCB.AND.MDFS.LE.MVCT) MDFSC=1
      IF (MDFS.EQ.0) LDFSS=0
      IF (MDFS.EQ.0) LDFSF=0
      IF (MDFS.EQ.0) GO TO 320
      IF (LDFSS.EQ.1) GO TO 310
      IF (TL(LDFSS).GT.0.0) TL(1)=TL(LDFSS)
      IF (TU(LDFSS).GT.0.0) TU(1)=TU(LDFSS)
  310 IF (MDFS.EQ.2.OR.MDFS.EQ.MMAX-1) IVCE=1
      IF (LDFSS.EQ.2.OR.LDFSF.EQ.LMAX-1) IVCE=1
      IF (ISUPER.GE.2.AND.LDFSS.NE.1) IVCE=1
      CLDFSS=ABS(YU(LDFSS)-YL(LDFSS))/YL(LDFSS)
      CLDFSF=ABS(YU(LDFSF)-YL(LDFSF))/YL(LDFSF)
      IF (LDFSS.NE.1.AND.CLDFSS.GT.0.001) IVCE=1
      IF (LDFSF.NE.LMAX.AND.CLDFSF.GT.0.001) IVCE=1
      IF (JFLAG.EQ.1.AND.LJET.LE.LDFSF) IVCE=1
      IF (IVCE.EQ.0) GO TO 320
      WRITE (6,2060)
      GO TO 10
C
C     CHECK THE SUBCYCLED GRID PARAMETERS
C
  320 IF (IVC.EQ.0) GO TO 350
      MVCB1=MVCB+1
      MVCT1=MVCT-1
      IF (NVCMI.EQ.0) GO TO 330
      II1=NVCMI/2
      II2=(NVCMI+1)/2
      IF (II1.EQ.II2) IVCE=1
      IF (IVCE.EQ.0) GO TO 330
      WRITE (6,2070)
      GO TO 10
  330 IF (MVCB.EQ.1.AND.MVCT.EQ.MMAX) IVCE=1
      IF (MDFS.EQ.0) GO TO 340
      IF (MVCT.LT.MDFS-1.OR.MVCB.GT.MDFS+1) GO TO 340
      IF (MVCB.EQ.MDFS+1.OR.MVCT.EQ.MDFS-1) IVCE=1
      IF (MVCB.GT.MDFS-2) IVCE=1
      IF (MVCT.LT.MDFS+2) IVCE=1
      IF (IVCE.EQ.0) GO TO 350
      WRITE (6,2080)
      GO TO 10
  340 IF (MVCB.EQ.2.OR.MVCT.EQ.MMAX-1) IVCE=1
      IF (MVCT-MVCB.LT.2) IVCE=1
      IF (IVCE.EQ.0) GO TO 350
      WRITE (6,2090)
      GO TO 10
C
C     CHECK THE QUICK SOLVER PARAMETERS
C
  350 IF (IVC.EQ.0) IQS=0
      IF (IQS.EQ.0) GO TO 370
      IF (NIQSF.EQ.0) NIQSF=NMAX
      IF (NOSLIP.EQ.0) IVCE=1
      IF (MVCT.EQ.MMAX.AND.IWALL.NE.0) IVCE=1
      IF (MVCB.EQ.1.AND.NGCB.EQ.0) IVCE=1
      IF (MDFS.EQ.0) GO TO 360
      IF (MVCB.GT.MDFS.OR.MVCT.LT.MDFS) IVCE=1
```

```
  360 IF (IVCE.EQ.0) GO TO 370
      WRITE (6,2130)
      GO TO 10
C
C     CHECK FOR ZERO VALUES OF Q AND E - SET THE DEFAULT VALUES
C
  370 IF (ITM.LE.1) GO TO 390
      IF (NSTART.NE.0) GO TO 390
      DO 380 L=1,LMAX
      IF (QL(L,1).LE.0.0) QL(L,1)=FSQL
      IF (EL(L,1).LE.0.0) EL(L,1)=FSEL
      DO 380 M=1,MMAX
      IF (Q(L,M,1).LE.0.0) Q(L,M,1)=FSQ(M)
      IF (E(L,M,1).LE.0.0) E(L,M,1)=FSE(M)
  380 CONTINUE
C
C     CONVERT METRIC UNITS TO ENGLISH UNITS
C
  390 IF (IUI.EQ.1) GO TO 540
      RSTAR=RSTAR/2.54
      RSTARS=RSTARS/6.4516
      PLOW=PLOW/6.8948
      ROLOW=ROLOW/16.02
      CMU=CMU/47.88/1.8**EMU
      CLA=CLA/47.88/1.8**ELA
      CK=CK*0.125/1.8**EK
      RGAS=RGAS/5.38032
      XT=XT/2.54
      RT=RT/2.54
      XI=XI/2.54
      XE=XE/2.54
      XICB=XICB/2.54
      XECB=XECB/2.54
      DX=DX/2.54
      DO 400 L=1,LMAX
      XP(L)=XP(L)/2.54
      YW(L)=YW(L)/2.54
      YCB(L)=YCB(L)/2.54
      YL(L)=YL(L)/2.54
      YU(L)=YU(L)/2.54
  400 CONTINUE
      DO 410 M=1,MMAX
      PT(M)=PT(M)/6.8948
      PE(M)=PE(M)/6.8948
      TT(M)=TT(M)*1.8
  410 CONTINUE
      PTL=PTL/6.8948
      PEL=PEL/6.8948
      PEI=PEI/6.8948
      TTL=TTL*1.8
      IF (TCB(1).LT.0.0) GO TO 430
      DO 420 L=1,LMAX
      TCB(L)=TCB(L)*1.8
  420 CONTINUE
  430 IF (TW(1).LT.0.0) GO TO 450
      DO 440 L=1,LMAX
      TW(L)=TW(L)*1.8
  440 CONTINUE
  450 IF (TL(1).LT.0.0) GO TO 470
      DO 460 L=1,LMAX
      TL(L)=TL(L)*1.8
  460 CONTINUE
  470 IF (TU(1).LT.0.0) GO TO 490
      DO 480 L=1,LMAX
      TU(L)=TU(L)*1.8
  480 CONTINUE
  490 IF (ISUPER.EQ.0) GO TO 520
      DO 500 M=1,MMAX
      UI(M)=UI(M)/0.3048
      VI(M)=VI(M)/0.3048
      PI(M)=PI(M)/6.8948
      ROI(M)=ROI(M)/16.02
```

```
  500 CONTINUE
      UIL=UIL/0.3048
      VIL=VIL/0.3048
      PIL=PIL/6.8948
      ROIL=ROIL/16.02
      QLOW=QLOW/0.0929
      ELOW=ELOW/0.0929
      FSQL=FSQL/0.0929
      FSEL=FSEL/0.0929
      DO 510 M=1,MMAX
      FSQ(M)=FSQ(M)/0.0929
      FSE(M)=FSE(M)/0.0929
  510 CONTINUE
  520 IF (N1D.NE.0) GO TO 540
      IF (NSTART.NE.0) GO TO 540
      DO 530 L=1,LMAX
      UL(L,1)=UL(L,1)/0.3048
      VL(L,1)=VL(L,1)/0.3048
      PL(L,1)=PL(L,1)/6.8948
      ROL(L,1)=ROL(L,1)/16.02
      QL(L,1)=QL(L,1)/0.0929
      EL(L,1)=EL(L,1)/0.0929
      DO 530 M=1,MMAX
      U(L,M,1)=U(L,M,1)/0.3048
      V(L,M,1)=V(L,M,1)/0.3048
      P(L,M,1)=P(L,M,1)/6.8948
      RO(L,M,1)=RO(L,M,1)/16.02
      Q(L,M,1)=Q(L,M,1)/0.0929
      E(L,M,1)=E(L,M,1)/0.0929
  530 CONTINUE
C
C     CONVERT INPUT DATA UNITS TO INTERNAL UNITS - THE INTERNAL UNITS
C     ARE P=LBF/FT2, RO=LBF-S2/FT4, X=YCB=YL=YU=YW=INCHES, Y=
C     DIMENSIONLESS, DT=IN-S/FT, MU=LA=LBF-S-IN/FT3, K=LBF-IN/S-R-FT,
C     Q=FT2/S2, E=FT2/S3, TML=INCHES, U=V=FT/S, AND RGAS=LBF-FT/LBM-R.
C
  540 IF (IUNIT.EQ.0) GO TO 550
      PC=1.0
      LC=1.0
      G=1.0
  550 TCONV=TCONV/100.0
      T=TSTART*LC
      TSTOP=TSTOP*LC
      CMU=CMU*LC
      CLA=CLA*LC
      CK=CK*LC
      DO 560 L=1,LMAX
      XWI(L)=0.0
  560 CONTINUE
      DO 570 M=1,MMAX
      PT(M)=PT(M)*PC
      PE(M)=PE(M)*PC
      THETA(M)=THETA(M)*0.0174533
  570 CONTINUE
      PTL=PTL*PC
      PEL=PEL*PC
      PEI=PEI*PC
      THETAL=THETAL*0.0174533
      IF (N1D.NE.0) GO TO 590
      DO 580 L=1,LMAX
      PL(L,1)=PL(L,1)*PC
      ROL(L,1)=ROL(L,1)/G
      DO 580 M=1,MMAX
      P(L,M,1)=P(L,M,1)*PC
      RO(L,M,1)=RO(L,M,1)/G
  580 CONTINUE
  590 RG=RGAS*G
C
C     FILL THE ARRAYS AT L=1 WITH THE INFLOW BOUNDARY CONDITIONS
C
      IF (ISUPER.EQ.0) GO TO 620
      DO 600 M=1,MMAX
```

```
      IF (ISUPER.EQ.2.AND.M.GE.MDFS) GO TO 600
      IF (ISUPER.EQ.3.AND.M.LT.MDFS) GO TO 600
      U(1,M,1)=UI(M)
      V(1,M,1)=VI(M)
      IF (NSTART.EQ.0) P(1,M,1)=PI(M)*PC
      RO(1,M,1)=ROI(M)/G
      U(1,M,2)=U(1,M,1)
      V(1,M,2)=V(1,M,1)
      P(1,M,2)=P(1,M,1)
      RO(1,M,2)=RO(1,M,1)
  600 CONTINUE
      IF (ISUPER.EQ.3) GO TO 610
      UL(1,1)=UIL
      VL(1,1)=VIL
      IF (NSTART.EQ.0) PL(1,1)=PIL*PC
      ROL(1,1)=ROIL/G
      UL(1,2)=UL(1,1)
      VL(1,2)=VL(1,1)
      PL(1,2)=PL(1,1)
      ROL(1,2)=ROL(1,1)
      GO TO 620
  610 PT(MDFS)=PTL
      TT(MDFS)=TTL
      THETA(MDFS)=THETAL
C
C     ZERO VISCOUS TERM ARRAYS
C
  620 DO 630 L=1,LMAX
      QUTL(L)=0.0
      QVTL(L)=0.0
      QPTL(L)=0.0
      QROTL(L)=0.0
      QQTL(L)=0.0
      QETL(L)=0.0
      DO 630 M=1,MMAX
      QUT(L,M)=0.0
      QVT(L,M)=0.0
      QPT(L,M)=0.0
      QROT(L,M)=0.0
      QQT(L,M)=0.0
      QET(L,M)=0.0
  630 CONTINUE
      IF (N1D.EQ.0) GO TO 640
C
C     COMPUTE THE 1-D INITIAL-DATA SURFACE
C
      CALL ONEDIM
      IF (IERR.NE.0) GO TO 10
C
C     COMPUTE THE INITIAL-DATA SURFACE MASS FLOW AND MOMENTUM THRUST
C
  640 IF (NPRINT.GT.0) GO TO 650
      NPRINT=-NPRINT
      GO TO 730
  650 CALL MASFLO
C
C     PRINT THE INITIAL-DATA SURFACE
C
      IP=0
      DO 720 IU=1,2
      IF (IUO.EQ.1.AND.IU.EQ.2) GO TO 720
      IF (IUO.EQ.2.AND.IU.EQ.1) GO TO 720
      NLINE=0
      WRITE (6,1380)
      WRITE (6,1520) TSTART,NSTART
      WRITE (6,1530)
      IF (IU.EQ.1) WRITE (6,1540)
      IF (IU.EQ.2) WRITE (6,1550)
      WRITE (6,1390)
      DO 710 L=1,LMAX
      LMAP=L
      IF (MDFS.NE.0) IB=3
```

```
      IF (L.NE.1) WRITE (6,1880)
      IF (L.NE.1) NLINE=NLINE+1
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DO 710 M=1,MMAX
      MMAP=M
      CALL MAP
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 670
      IMDFS=0
      VELMAG=SQRT(UL(L,1)**2+VL(L,1)**2)
      XMACH=VELMAG/SQRT(GAMMA*PL(L,1)/ROL(L,1))
      PRES=PL(L,1)/PC
      RHO=ROL(L,1)*G
      TEMP=PL(L,1)/(RHO*RGAS)
      XPP=XP(L)
      UP=UL(L,1)
      VP=VL(L,1)
      GO TO 680
  660 IMDFS=1
      IB=4
      CALL MAP
  670 VELMAG=SQRT(U(L,M,1)**2+V(L,M,1)**2)
      XMACH=VELMAG/SQRT(GAMMA*P(L,M,1)/RO(L,M,1))
      PRES=P(L,M,1)/PC
      RHO=RO(L,M,1)*G
      TEMP=P(L,M,1)/RHO/RGAS
      XPP=XP(L)
      UP=U(L,M,1)
      VP=V(L,M,1)
  680 IF (IU.EQ.1) GO TO 690
      XPP=XP(L)*2.54
      YP=YP*2.54
      UP=UP*0.3048
      VP=VP*0.3048
      PRES=PRES*6.8948
      RHO=RHO*16.02
      VELMAG=VELMAG*0.3048
      TEMP=TEMP*5.0/9.0
  690 NLINE=NLINE+1
      IF (NLINE.LT.54) GO TO 700
      WRITE (6,1380)
      WRITE (6,1520) TSTART,NSTART
      WRITE (6,1530)
      IF (IU.EQ.1) WRITE (6,1540)
      IF (IU.EQ.2) WRITE (6,1550)
      WRITE (6,1390)
      NLINE=1
  700 WRITE (6,1560) L,M,XPP,YP,UP,VP,PRES,RHO,VELMAG,XMACH,TEMP
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 710
      IF (IMDFS.EQ.0) GO TO 660
  710 CONTINUE
      IF (IUO.NE.3) GO TO 730
  720 CONTINUE
  730 IF (NPLOT.LE.0) GO TO 740
      CALL PLOT (TITLE,TSTART,NSTART,IVPTS)
      WRITE (6,1810) NSTART
  740 IF (NMAX.EQ.0) GO TO 10
C
C     SAVE THE SOLUTION FOR THE EXTENDED INTERVAL TIME SMOOTHING
C
      IF (NTST.EQ.1) GO TO 760
      DO 750 L=1,LMAX
      ULS(L)=UL(L,1)
      VLS(L)=VL(L,1)
      PLS(L)=PL(L,1)
      ROLS(L)=ROL(L,1)
      QLS(L)=QL(L,1)
      ELS(L)=EL(L,1)
      DO 750 M=1,MMAX
      US(L,M)=U(L,M,1)
      VS(L,M)=V(L,M,1)
      PS(L,M)=P(L,M,1)
```

```
      ROS(L,M)=RO(L,M,1)
      QS(L,M)=Q(L,M,1)
      ES(L,M)=E(L,M,1)
  750 CONTINUE
C
C     INITIALIZE THE TIME STEP INTEGRATION LOOP PARAMETERS
C
  760 NMAXD=NMAX
      N1=1
      N3=2
      DQM=0.0
      NCONV=0
      NC=0
      NPC=0
      NPD=0
      LDUM=1
      DXR=1.0/DX
      DYR=1.0/DY
      DXRS=DXR*DXR
      DYRS=DYR*DYR
      VDT=1.0/VDT
      VDT1=1.0/VDT1
      FDTD=FDT
      FDT=FDTI
      IF (NST.NE.0) DELSMP=(SMPF-SMP)/FLOAT(NST)
      IF (NST.NE.0) DSMPT=(SMPTF-SMPT)/FLOAT(NST)
      INTST=0
      MVCTD=MVCT+1
      IF (NASM.NE.0.AND.LT.NE.1) LDUM=LT-1
      WRITE (6,1400)
      LPP1D=LPP1
      LPP1=IABS(LPP1)
      IF (JFLAG.EQ.0) GO TO 770
      UD(1)=U(LJET-1,MMAX,N1)
      VD(1)=V(LJET-1,MMAX,N1)
      PD(1)=P(LJET-1,MMAX,N1)
      ROD(1)=RO(LJET-1,MMAX,N1)
      UD(2)=UD(1)
      VD(2)=VD(1)
      PD(2)=PD(1)
      ROD(2)=ROD(1)
C
C     ENTER THE TIME STEP INTEGRATION LOOP
C
  770 DO 1250 N=1,NMAXD
C
      IF (N.EQ.2) FDT=FDTD
      SMP=SMP+DELSMP
      SMPT=SMPT+DSMPT
      IQSD=0
      IF (IQS.EQ.0) GO TO 780
      IF (N.GE.NIQSS.AND.N.LE.NIQSF) IQSD=IQS
C
C     CALCULATE DELTA T
C
  780 IP=1
      UPAM=0.0
      DO 810 L=2,L1
      LMAP=L
      IF (MDFS.NE.0) IB=3
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DXP1=XP(L)-XP(L-1)
      DXP2=XP(L+1)-XP(L)
      DXP=AMIN1(DXP1,DXP2)
      DO 810 M=2,M1
      IF (IVC.EQ.0) GO TO 790
      IF (M.GE.MVCB.AND.M.LE.MVCT) GO TO 810
  790 IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 800
      IB=4
      GO TO 810
  800 MMAP=M
      CALL MAP
      DYP3=DY/BE3
```

```
      DYP4=DY/BE4
      DYP=AMIN1(DYP3,DYP4)
      ABR=AL3/BE3
      A=SQRT(GAMMA*P(L,M,N1)/RO(L,M,N1))
      UPA1=(ABS(U(L,M,N1))+A)/DXP+VDT*TMUX
      VTOL=ABS(U(L,M,N1)*ABR+V(L,M,N1))
      AST=SQRT(AL3*AL3+BE3*BE3)/BE3
      UPA2=(VTOL+AST*A)/DYP+VDT*TMUY
      UPA=AMAX1(UPA1,UPA2)
      IF (UPA.LE.UPAM) GO TO 810
      UPAM=UPA
      LDU=L
      MDU=M
  810 CONTINUE
C
C     CALCULATE DELTA T FOR THE SUBCYCLED GRID
C
      IF (IVC.EQ.0) GO TO 860
      IF (NVCMI.EQ.0) GO TO 820
      IF (IQS.NE.0.AND.IQSD.EQ.0) GO TO 820
      NVCM=NVCMI
      NVCM1=NVCM+1
      CNUMS=0.0
      LDUF=0
      MDUF=0.0
      GO TO 860
  820 UPAMF=0.0
      DO 840 L=2,L1
      LMAP=L
      IF (MDFS.NE.0) IB=3
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DXP1=XP(L)-XP(L-1)
      DXP2=XP(L+1)-XP(L)
      DXP=AMIN1(DXP1,DXP2)
      DO 840 M=MVCB1,MVCT1
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 830
      IB=4
      GO TO 840
  830 MMAP=M
      CALL MAP
      DYP3=DY/BE3
      DYP4=DY/BE4
      DYP=AMIN1(DYP3,DYP4)
      ABR=AL3/BE3
      A=SQRT(GAMMA*P(L,M,N1)/RO(L,M,N1))
      UPA1=(ABS(U(L,M,N1))+A)/DXP+VDT1*TMU1X
      VTOL=ABS(U(L,M,N1)*ABR+V(L,M,N1))
      AST=SQRT(AL3*AL3+BE3*BE3)/BE3
      IF (IQSD.NE.0) AST=AST-1.0
      UPA2=(VTOL+AST*A)/DYP+VDT1*TMU1Y
      UPA=AMAX1(UPA1,UPA2)
      IF (UPA.LE.UPAMF) GO TO 840
      UPAMF=UPA
      LDUF=L
      MDUF=M
  840 CONTINUE
C
C     DETERMINE THE NUMBER OF SUBCYCLES
C
      XNVCM=UPAMF/(UPAM*FDT1)
      NVCM=0
      I=-1
      IF (XNVCM.LE.200.0) GO TO 850
      IF (N.EQ.1) GO TO 850
      NP=N+NSTART
      WRITE (6,2100) NP
      NMAX=N
      NVCM=XNVCM
      DT=FDT/UPAM
      GO TO 1110
  850 I=I+2
```

```
      IF (XNVCM.LE.FLOAT(I)) NVCM=I
      IF (NVCM.EQ.0) GO TO 850
      NVCM1=NVCM+1
      CNUMS=XNVCM/FLOAT(NVCM)
  860 DT=FDT/UPAM
      T=T+DT
      IF (T.LT.TSTOP) GO TO 870
      T=T-DT
      DT=TSTOP-T
      T=TSTOP
      ISTOP=1
C
C     PRINT N,T AND DT
C
  870 NPD-NPD+1
      NC=NC+1
      NPC=NPC+1
      TMUX=0.0
      TMUY=0.0
      TMU1X=0.0
      TMU1Y=0.0
      IF (NPD.NE.10) GO TO 880
      NP=N+NSTART
      TIME=T/LC
      DTIME=DT/LC
      WRITE (6,1820) NP,TIME,DTIME,NVCM,CNUMS,LDU,MDU,LDUF,MDUF
      NPD=0
C
C     BEGIN THE SUBCYCLE LOOP
C
  880 DO 1010 NVC=1,NVCM1
      RIND=FLOAT(NVC-2)/FLOAT(NVCM)
      RIND1=FLOAT(NVC-1)/FLOAT(NVCM)
      IF (NVC.NE.2) GO TO 890
      DT=DT/FLOAT(NVCM)
C
C     CALCULATE THE PREDICTOR SOLUTION
C
  890 IB=1
      IF (IQSD.NE.0.AND.NVC.NE.1) CALL QSOLVE
      IF (IERR.NE.0) GO TO 1100
      IF (CAV.NE.0.0.OR.CHECK.NE.0.0) CALL VISCOUS
      IF (IERR.NE.0) GO TO 1100
      ICHAR=1
      IB=1
      CALL INTER
      IF (NVC.GT.1.AND.MVCT.NE.MMAX) GO TO 900
      IF (NVC.EQ.1.AND.MVCT.EQ.MMAX) GO TO 900
      CALL WALL
      IF (IERR.NE.0) GO TO 1090
  900 IF (NGCB.EQ.0) GO TO 910
      IF (NVC.GT.1.AND.MVCB.NE.1) GO TO 910
      IF (NVC.EQ.1.AND.MVCB.EQ.1) GO TO 910
      IB=2
      CALL WALL
      IF (IERR.NE.0) GO TO 1090
  910 IF (LDFSS.NE.1.OR.(NVC.EQ.1.AND.MDFSC.NE.0)) CALL INLET
      IF (LDFSF.NE.LMAX.OR.(NVC.EQ.1.AND.MDFSC.NE.0)) CALL EXITT
      IF (IERR.NE.0) GO TO 1090
C
C     CALCULATE THE DUAL FLOW SPACE BOUNDARY PREDICTOR SOLUTION
C
      IF (MDFS.EQ.0) GO TO 920
      IF (NVC.EQ.1.AND.MDFSC.NE.0) GO TO 920
      IF (NVC.GT.1.AND.MVCT.LT.MDFS) GO TO 920
      IF (NVC.GT.1.AND.MVCB.GT.MDFS) GO TO 920
      IB=4
      CALL WALL
      IF (IERR.NE.0) GO TO 1090
      IF (LDFSS.EQ.1) CALL INLET
      IF (LDFSF.EQ.LMAX) CALL EXITT
      IF (IERR.NE.0) GO TO 1090
```

```
      CALL SWITCH (2)
      IB=3
      CALL WALL
      IF (IERR.NE.O) GO TO 1080
      IF (LDFSS.EQ.1) CALL INLET
      IF (LDFSF.EQ.LMAX) CALL EXITT
      IF (IERR.NE.O) GO TO 1080
C
C     CALCULATE THE CORRECTOR SOLUTION
C
  920 IF (ITM.GE.2) CALL TURBC (3)
      ICHAR=2
      IB=1
      CALL INTER
      IF (NVC.GT.1.AND.MVCT.NE.MMAX) GO TO 930
      IF (NVC.EQ.1.AND.MVCT.EQ.MMAX) GO TO 930
      CALL WALL
      IF (IERR.NE.O) GO TO 1070
  930 IF (NGCB.EQ.O) GO TO 940
      IF (NVC.GT.1.AND.MVCB.NE.1) GO TO 940
      IF (NVC.EQ.1.AND.MVCB.EQ.1) GO TO 940
      IB=2
      CALL WALL
      IF (IERR.NE.O) GO TO 1070
  940 IF (LDFSS.NE.1.OR.(NVC.EQ.1.AND.MDFSC.NE.O)) CALL INLET
      IF (LDFSF.NE.LMAX.OR.(NVC.EQ.1.AND.MDFSC.NE.O)) CALL EXITT
      IF (IERR.NE.O) GO TO 1070
C
C     CALCULATE THE DUAL FLOW SPACE BOUNDARY CORRECTOR SOLUTION
C
      IF (MDFS.EQ.O) GO TO 950
      IF (NVC.EQ.1.AND.MDFSC.NE.O) GO TO 950
      IF (NVC.GT.1.AND.MVCT.LT.MDFS) GO TO 950
      IF (NVC.GT.1.AND.MVCB.GT.MDFS) GO TO 950
      IB=3
      CALL WALL
      IF (IERR.NE.O) GO TO 1080
      IF (LDFSS.EQ.1) CALL INLET
      IF (LDFSF.EQ.LMAX) CALL EXITT
      IF (IERR.NE.O) GO TO 1080
      CALL SWITCH (2)
      IB=4
      CALL WALL
      IF (IERR.NE.O) GO TO 1090
      IF (LDFSS.EQ.1) CALL INLET
      IF (LDFSF.EQ.LMAX) CALL EXITT
      IF (IERR.NE.O) GO TO 1090
C
C     SET THE SUBCYCLED GRID END CONDITIONS
C
  950 IF (NVCM1.EQ.1) GO TO 1010
      IF (NVC.EQ.1) GO TO 990
      IF (NVC.EQ.NVCM1) GO TO 970
      IF (LPP1D.GE.O) GO TO 960
      PCDUM=PC
      IF (IUO.EQ.2) PCDUM=PC/6.8948
      PPP1=P(LPP1,MPP1,N3)/PCDUM
      PPP2=P(LPP2,MPP2,N3)/PCDUM
      PPP3=P(LPP3,MPP3,N3)/PCDUM
      WRITE (6,2110) NVC,LPP1,MPP1,PPP1,LPP2,MPP2,PPP2,LPP3,MPP3,PPP3
  960 IF (ITM.GE.2) CALL TURBC (2)
      NNN=N1
      N1=N3
      N3=NNN
      GO TO 1010
  970 DT=DT*FLOAT(NVCM)
      IF (MVCTD.GE.MMAX) GO TO 1010
      DO 980 L=1,LMAX
      U(L,MVCTD,N3)=UU2(L)
      V(L,MVCTD,N3)=VV2(L)
      P(L,MVCTD,N3)=PP2(L)
      RO(L,MVCTD,N3)=RORO2(L)
```

```
      Q(L,MVCTD,N3)=QQ2(L)
      E(L,MVCTD,N3)=EE2(L)
  980 CONTINUE
      GO TO 1010
  990 NN1=N1
      NN3=N3
      IF (MVCTD.GE.MMAX) GO TO 1010
      DO 1000 L=1,LMAX
      UU1(L)=U(L,MVCTD,N1)
      VV1(L)=V(L,MVCTD,N1)
      PP1(L)=P(L,MVCTD,N1)
      RORO1(L)=RO(L,MVCTD,N1)
      QQ1(L)=Q(L,MVCTD,N1)
      EE1(L)=E(L,MVCTD,N1)
      UU2(L)=U(L,MVCTD,N3)
      VV2(L)=V(L,MVCTD,N3)
      PP2(L)=P(L,MVCTD,N3)
      RORO2(L)=RO(L,MVCTD,N3)
      QQ2(L)=Q(L,MVCTD,N3)
      EE2(L)=E(L,MVCTD,N3)
 1000 CONTINUE
 1010 CONTINUE
C
C     PRINT THE PRESSURE AT THE THREE REQUESTED POINTS
C
      IF (LPP1.EQ.0) GO TO 1040
      NP=N+NSTART
      PCDUM=PC
      IF (IUO.EQ.2) PCDUM=PC/6.8948
      PPP1=P(LPP1,MPP1,N3)/PCDUM
      PPP2=P(LPP2,MPP2,N3)/PCDUM
      PPP3=P(LPP3,MPP3,N3)/PCDUM
      IF (N.GT.NST) GO TO 1030
      IF (NTST.GT.0) GO TO 1030
      IF (N.GT.2) GO TO 1020
      IF (N.EQ.1) PC2=PPP1
      IF (N.EQ.2) PC3=PPP1
      GO TO 1030
 1020 PC1=PC2
      PC2=PC3
      PC3=PPP1
      IF ((PC3-PC2)*(PC2-PC1).LT.0.0) NTST=-1
      IF (INTST.EQ.3) INTST=0
      IF (INTST.EQ.2) INTST=3
      IF (INTST.EQ.1) INTST=2
      IF (INTST.EQ.0.AND.NTST.NE.0) INTST=1
      IF (INTST.NE.1) NTST=0
 1030 WRITE (6,2120) NP,LPP1,MPP1,PPP1,LPP2,MPP2,PPP2,LPP3,MPP3,PPP3
     1 ,NTST
 1040 IF (N.LE.NST) CALL SMOOTH
      IF (NTST.EQ.-1) NTST=0
      IF (ITM.GE.2) CALL TURBC (1)
C
C     DETERMINE THE MAXIMUM (DELTA U)/U
C
      IF (TCONV.LE.0.0) GO TO 1060
      DQM=0.0
      DO 1050 L=LDUM,LMAX
      DO 1050 M=1,MMAX
      IF (U(L,M,N1).EQ.0.0) GO TO 1050
      DQ=ABS((U(L,M,N3)-U(L,M,N1))/U(L,M,N1))
      IF (DQ.GT.DQM) DQM=DQ
 1050 CONTINUE
C
C     CHECK FOR REQUESTED PRINTING OR PLOTTING
C
 1060 IF (DQM.GE.TCONV) GO TO 1110
      NCONV=NCONV+1
      IF (NCONV.EQ.1) NCHECK-N-1
      IF (NCONV.GE.NCONVI) NC=NPRINT
      IF (NCONV.GE.NCONVI) NPC=NPLOT
      IF (N.GE.NCHECK+NCONVI) NCONV=0
```

```
      GO TO 1110
C
 1070 IF (MDFS.EQ.0) GO TO 1090
 1080 CALL SWITCH (3)
 1090 N3=N1
 1100 NMAX=N
      IF (NVC.GE.2) DT=DT*FLOAT(NVCM)
C
 1110 IF (N.EQ.NMAX) NC=NPRINT
      IF (N.EQ.NMAX) NPC=NPLOT
      IF (ISTOP.NE.0) NC=NPRINT
      IF (ISTOP.NE.0) NPC=NPLOT
      IF (NC.EQ.NPRINT) GO TO 1120
      IF (NPC.EQ.NPLOT) GO TO 1220
      GO TO 1240
C
C     COMPUTE THE SOLUTION SURFACE MASS FLOW AND MOMENTUM THRUST
C
 1120 ICN=0
      IF (JFLAG.EQ.0) GO TO 1130
      IF (LT.NE.LJET-1) GO TO 1130
      UDUM=U(LT,MMAX,N3)
      RODUM=RO(LT,MMAX,N3)
      U(LT,MMAX,N3)=UD(3)
      RO(LT,MMAX,N3)=ROD(3)
      ICN=1
 1130 CALL MASFLO
      IF (ICN.EQ.0) GO TO 1140
      U(LT,MMAX,N3)=UDUM
      RO(LT,MMAX,N3)=RODUM
C
C     PRINT THE SOLUTION SURFACE
C
 1140 IP=0
      DO 1210 IU=1,2
      IF (IUO.EQ.1.AND.IU.EQ.2) GO TO 1210
      IF (IUO.EQ.2.AND.IU.EQ.1) GO TO 1210
      NLINE=0
      WRITE (6,1380)
      TIME=T/LC
      DTIME=DT/LC
      NP=N+NSTART
      WRITE (6,1570) NP,TIME,DTIME,NVCM,CNUMS,LDU,MDU,LDUF,MDUF
      WRITE (6,1530)
      IF (IU.EQ.1) WRITE (6,1540)
      IF (IU.EQ.2) WRITE (6,1550)
      WRITE (6,1390)
      DO 1200 L=1,LMAX
      LMAP=L
      IF (MDFS.NE.0) IB=3
      IF (L.NE.1) WRITE (6,1880)
      IF (L.NE.1) NLINE=NLINE+1
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DO 1200 M=1,MMAX
      MMAP=M
      CALL MAP
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 1160
      IMDFS=0
      VELMAG=SQRT(UL(L,N3)**2+VL(L,N3)**2)
      XMACH=VELMAG/SQRT(GAMMA*PL(L,N3)/ROL(L,N3))
      PRES=PL(L,N3)/PC
      RHO=ROL(L,N3)*G
      TEMP=PL(L,N3)/(RHO*RGAS)
      XPP=XP(L)
      UP=UL(L,N3)
      VP=VL(L,N3)
      GO TO 1170
 1150 IMDFS=1
      IB=4
      CALL MAP
 1160 VELMAG=SQRT(U(L,M,N3)**2+V(L,M,N3)**2)
```

```
      XMACH=VELMAG/SQRT(GAMMA*P(L,M,N3)/RO(L,M,N3))
      PRES=P(L,M,N3)/PC
      RHO=RO(L,M,N3)*G
      TEMP=P(L,M,N3)/RHO/RGAS
      XPP=XP(L)
      UP=U(L,M,N3)
      VP=V(L,M,N3)
 1170 IF (IU.EQ.1) GO TO 1180
      XPP=XP(L)*2.54
      YP=YP*2.54
      UP=UP*0.3048
      VP=VP*0.3048
      PRES=PRES*6.8948
      RHO=RHO*16.02
      VELMAG=VELMAG*0.3048
      TEMP=TEMP*5.0/9.0
 1180 NLINE=NLINE+1
      IF (NLINE.LT.54) GO TO 1190
      WRITE (6,1380)
      WRITE (6,1570) NP,TIME,DTIME,NVCM,CNUMS,LDU,MDU,LDUF,MDUF
      WRITE (6,1530)
      IF (IU.EQ.1) WRITE (6,1540)
      IF (IU.EQ.2) WRITE (6,1550)
      WRITE (6,1390)
      NLINE=1
 1190 WRITE (6,1560) L,M,XPP,YP,UP,VP,PRES,RHO,VELMAG,XMACH,TEMP
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 1200
      IF (IMDFS.EQ.0) GO TO 1150
 1200 CONTINUE
      IF (IUO.NE.3) GO TO 1220
 1210 CONTINUE
C
C     GENERATE THE FILM PLOTS
C
 1220 IF (NPLOT.LT.0) GO TO 1230
      IF (NPC.NE.NPLOT) GO TO 1230
      TIME=T/LC
      NP=N+NSTART
      CALL PLOT (TITLE,TIME,NP,IVPTS)
      WRITE (6,1810) NP
C
C     CHECK FOR CONVERGENCE OF THE STEADY STATE SOLUTION
C
 1230 IF (DQM.LT.TCONV) GO TO 1260
      IF (ISTOP.NE.0) GO TO 1260
      IF (N.EQ.NMAX) GO TO 1260
      IF (NC.EQ.NPRINT) NC=0
      IF (NPC.EQ.NPLOT) NPC=0
 1240 NNN=N1
      N1=N3
      N3=NNN
 1250 CONTINUE
C
C     PUNCH(WRITE) A $IVS NAMELIST FOR RESTART
C
 1260 IF (NPLOT.GE.0) CALL ADV (10)
      IF (IPUNCH.EQ.0) GO TO 10
      DO 1270 L=1,LMAX
      PL(L,N3)=PL(L,N3)/PC
      ROL(L,N3)=ROL(L,N3)*G
      DO 1270 M=1,MMAX
      P(L,M,N3)=P(L,M,N3)/PC
      RO(L,M,N3)=RO(L,M,N3)*G
 1270 CONTINUE
      WRITE (8,1620) NP,TIME
      DO 1280 M=1,MMAX
      WRITE (8,1630) M,U(1,M,N3)
      WRITE (8,1650) (U(L,M,N3),L=2,LMAX)
 1280 CONTINUE
      DO 1290 M=1,MMAX
      WRITE (8,1660) M,V(1,M,N3)
      WRITE (8,1650) (V(L,M,N3),L=2,LMAX)
 1290 CONTINUE
      DO 1300 M=1,MMAX
```

```
      WRITE (8,1680) M,P(1,M,N3)
      WRITE (8,1700) (P(L,M,N3),L=2,LMAX)
 1300 CONTINUE
      DO 1310 M=1,MMAX
      WRITE (8,1710) M,RO(1,M,N3)
      WRITE (8,1730) (RO(L,M,N3),L=2,LMAX)
 1310 CONTINUE
      IF (ITM.LE.1) GO TO 1340
      DO 1320 M=1,MMAX
      WRITE (8,1740) M,Q(1,M,N3)
      WRITE (8,1760) (Q(L,M,N3),L=2,LMAX)
 1320 CONTINUE
      IF (ITM.EQ.2) GO TO 1340
      DO 1330 M=1,MMAX
      WRITE (8,1750) M,E(1,M,N3)
      WRITE (8,1760) (E(L,M,N3),L=2,LMAX)
 1330 CONTINUE
 1340 IF (MDFS.EQ.0) GO TO 1350
      LDFSSP1=LDFSS+1
      WRITE (8,1640) LDFSS,UL(LDFSS,N3)
      WRITE (8,1650) (UL(L,N3),L=LDFSSP1,LDFSF)
      WRITE (8,1670) LDFSS,VL(LDFSS,N3)
      WRITE (8,1650) (VL(L,N3),L=LDFSSP1,LDFSF)
      WRITE (8,1690) LDFSS,PL(LDFSS,N3)
      WRITE (8,1700) (PL(L,N3),L=LDFSSP1,LDFSF)
      WRITE (8,1720) LDFSS,ROL(LDFSS,N3)
      WRITE (8,1730) (ROL(L,N3),L=LDFSSP1,LDFSF)
      IF (ITM.LE.1) GO TO 1350
      WRITE (8,1770) LDFSS,QL(LDFSS,N3)
      WRITE (8,1760) (QL(L,N3),L=LDFSSP1,LDFSF)
      IF (ITM.EQ.2) GO TO 1350
      WRITE (8,1780) LDFSS,EL(LDFSS,N3)
      WRITE (8,1760) (EL(L,N3),L=LDFSSP1,LDFSF)
 1350 WRITE (8,1790)
      NCARDS=(LMAX/7+2)*MMAX*4+2+LDFSF-LDFSS
      WRITE (6,1800) NCARDS
      GO TO 10
C
C     FORMAT STATEMENTS
C
 1370 FORMAT (10A8)
 1380 FORMAT (1H1)
 1390 FORMAT (1H )
 1400 FORMAT (1H0)
 1410 FORMAT (1H0,10X,47HVNAP2, A COMPUTER PROGRAM FOR THE COMPUTATION O
     1 ,58HF TWO-DIMENSIONAL, TIME-DEPENDENT, COMPRESSIBLE, TURBULENT,5H
     2 FLOW,//37X,57HBY MICHAEL C. CLINE, T-3 - LOS ALAMOS NATIONAL LABO
     3RATORY)
 1420 FORMAT (1H0,10X,18HPROGRAM ABSTRACT -,//26X,17HTHE NAVIER-STOKES,6
     1 2H EQUATIONS FOR TWO-DIMENSIONAL, TIME DEPENDENT FLOW ARE SOLVED,
     2 10H USING THE,/,21X,62HSECOND-ORDER, MACCORMACK FINITE-DIFFERENCE
     3 SCHEME. ALL BOUNDAR,31HY CONDITIONS ARE COMPUTED USING,/,21X,13HA
     4 SECOND-ORDE,62HR, REFERENCE PLANE CHARACTERISTIC SCHEME WITH THE
     5VISCOUS TERM,19HS TREATED AS SOURCE)
 1430 FORMAT (1H ,20X,41HFUNCTIONS.  THE FLUID IS ASSUMED TO BE A ,54HPE
     1RFECT GAS.  THE STEADY-STATE SOLUTION IS OBTAINED AS,/,21X,62HTHE
     2ASYMPTOTIC SOLUTION FOR LARGE TIME.  THE FLOW BOUNDARIES M,34HAY B
     3E ARBITRARY CURVED SOLID WALLS,/,21X,62HAS WELL AS JET ENVELOPES.
     4 THE GEOMETRY MAY CONSIST OF SINGLE ,36HAND DUAL FLOWING STREAMS.
     5TURBULENCE,/,21X,62HEFFECTS ARE MODELED WITH EITHER A MIXING-LENGT
     6H, A TURBULENCE ,32HENERGY EQUATION, OR A TURBULENCE,/,21X,62HENER
     7GY-DISSIPATION RATE EQUATIONS MODEL.  THIS PROGRAM ALLOWS ,34HVARI
     8ABLE GRID SPACING AND INCLUDES,/,21X,17HOPTIONS TO SPEED ,50HUP TH
     9E CALCULATION FOR HIGH REYNOLDS NUMBER FLOWS.)
 1440 FORMAT (1H0,10X,11HJOB TITLE -//21X,10A8)
 1450 FORMAT (1H0,10X,20HCONTROL PARAMETERS -)
 1460 FORMAT (1H0,20X,5HLMAX=,I2,2X,5HMMAX=,I2,3X,5HNMAX=,I4,2X,7HNPRINT
     1=,I4,2X,6HNPLOT=,I4,6X,4HFDT=,F4.2,2X,5HFDT1=,F4.2,3X,5HFDTI=,F4.2
     2 ,2X,7HIPUNCH=,I1,/,21X,4HIUI=,I1,4X,4HIUO=,I1,5X,6HIVPTS=,I1,4X,7
     3 HNCONVI=,I2,4X,6HTSTOP=,E8.2,2X,4HN1D=,I2,4X,6HTCONV=,F5.3,1X,5HN
     4ASM=,I1,5X,6HIUNIT=,I1,/,21X,6HRSTAR=,F11.6,2X,7HRSTARS=,F13.7,4X,
     5 5HPLOW=,F6.4,5X,6HROLOW=,F11.6,5X,4HVDT=,F4.2,3X,5HVDT1=,F4.2)
```

```
1470 FORMAT (1H0,10X,13HFLUID MODEL -,//21X,36HTHE RATIO OF SPECIFIC HE
    1ATS, GAMMA =,F6.4,26H AND THE GAS CONSTANT, R =,F9.4,15H (FT-LBF/L
    2BM-R))
1480 FORMAT (1H0,10X,13HFLUID MODEL -,//21X,36HTHE RATIO OF SPECIFIC HE
    1ATS, GAMMA =,F6.4,26H AND THE GAS CONSTANT, R =,F9.4,9H (J/KG-K))
1490 FORMAT (1H0,10X,15HFLOW GEOMETRY -)
1500 FORMAT (1H0,20X,47HTWO-DIMENSIONAL, PLANAR FLOW HAS BEEN SPECIFIED
    1 )
1510 FORMAT (1H0,20X,36HAXISYMMETRIC FLOW HAS BEEN SPECIFIED)
1520 FORMAT (1H ,30HINITIAL-DATA SURFACE - TIME = ,F12.8,8H SECONDS,4H
    1(N=,I6,1H))
1530 FORMAT (1H0,11X,1HL,4X,1HM,9X,1HX,10X,1HY,10X,1HU,11X,1HV,12X,1HP,
    1 11X,3HRHO,7X,4HVMAG,10X,4HMACH,8X,1HT)
1540 FORMAT (1H ,25X,4H(IN),7X,4H(IN),6X,5H(F/S),7X,5H(F/S),7X,6H(PSIA)
    1 ,6X,9H(LBM/FT3),4X,5H(F/S),10X,2HNO,8X,3H(R))
1550 FORMAT (1H ,25X,4H(CM),7X,4H(CM),6X,5H(M/S),7X,5H(M/S),7X,6H (KPA)
    1 ,7X,7H(KG/M3),5X,5H(M/S),10X,2HNO,8X,3H(K))
1560 FORMAT (1H ,7X,2I5,4F12.4,F13.5,F12.6,3F12.4)
1570 FORMAT (1H ,20HSOLUTION SURFACE NO.,I6,3H - ,7HTIME = ,F12.8,20H S
    1ECONDS (DELTA T = ,F10.8,8H, NVCM =,I3,9H, CNUMS =,F5.2,3H, (,I2,1
    2 H,,I2,4H), (,I2,1H,,I2,2H)))
1580 FORMAT (1H0,10X,21HBOUNDARY CONDITIONS -,//22X,1HM,10X,8HPT(PSIA),
    1 11X,5HTT(R),10X,10HTHETA(DEG),10X,8HPE(PSIA),7X,11HFSQ(FT2/S2),7X
    2 ,11HFSE(FT2/S3),/)
1590 FORMAT (1H0,10X,21HBOUNDARY CONDITIONS -,//22X,1HM,10X,7HPT(KPA),1
    1 2X,5HTT(K),10X,10HTHETA(DEG),10X,7HPE(KPA),8X,10HFSQ(M2/S2),8X,10
    2 HFSE(M2/S3),/)
1600 FORMAT (1H ,20X,I2,7X,F10.4,10X,F7.2,10X,F7.2,9X,F11.5,F18.4,F18.1
    1 )
1610 FORMAT (1H0,51H***** THE RADIUS OF THE CENTERBODY IS LARGER THAN T
    1 ,20HHE WALL RADIUS *****)
1620 FORMAT (1X,18H$IVS N1D=0,NSTART=,I6,8H,TSTART=,F14.10,1H,)
1630 FORMAT (1X,4HU(1,,I2,5H,1) =,F10.3,1H,)
1640 FORMAT (1X,3HUL(,I2,5H,1) =,F10.3,1H,)
1650 FORMAT ((1X,7(F10.3,1H,)))
1660 FORMAT (1X,4HV(1,,I2,5H,1) =,F10.3,1H,)
1670 FORMAT (1X,3HVL(,I2,5H,1) =,F10.3,1H,)
1680 FORMAT (1X,4HP(1,,I2,5H,1) =,F10.4,1H,)
1690 FORMAT (1X,3HPL(,I2,5H,1) =,F10.4,1H,)
1700 FORMAT ((1X,7(F10.4,1H,)))
1710 FORMAT (1X,5HRO(1,,I2,5H,1) =,F10.6,1H,)
1720 FORMAT (1X,4HROL(,I2,5H,1) =,F10.6,1H,)
1730 FORMAT ((1X,7(F10.6,1H,)))
1740 FORMAT (1X,4HQ(1,,I2,5H,1) =,E10.4,1H,)
1750 FORMAT (1X,4HE(1,,I2,5H,1) =,E10.4,1H,)
1760 FORMAT ((1X,7(E10.4,1H,)))
1770 FORMAT (1X,3HQL(,I2,5H,1) =,E10.4,1H,)
1780 FORMAT (1X,3HEL(,I2,5H,1) =,E10.4,1H,)
1790 FORMAT (1X,1H$)
1800 FORMAT (1H0,27H***** EXPECT APPROXIMATELY ,I4,20H PUNCHED CARDS **
    1***)
1810 FORMAT (1H0,31H***** EXPECT FILM OUTPUT FOR N=,I6,6H *****)
1820 FORMAT (1H ,10X,2HN=,I6,5H,  T=,F12.8,14H SECONDS,  DT=,F10.8,8H S
    1ECONDS,9H,  NVCM =,I3,10H,  CNUMS =,F5.2,4H,  (,I2,1H,,I2,5H),  (
    2 ,I2,1H,,I2,1H))
1830 FORMAT (1H0,10X,21HARTIFICAL VISCOSITY -,//21X,4HCAV=,F4.2,3X,4HXM
    1U=,F4.2,3X,4HXLA=,F4.2,3X,4HPRA=,F4.2,3X,4HXRO=,F4.2,3X,4HLSS=,I2,
    2 5X,4HLSF=,I3,3X,6HIDIVC=,I1,3X,4HISS=,I1,3X,6HSMACH=,F4.2,/,21X,4
    3 HNST=,I4,3X,4HSMP=,F4.2,3X,5HSMPF=,F4.2,2X,5HSMPT=,F4.2,2X,6HSMPT
    4F=,F4.2,1X,5HNTST=,I4,2X,4HIAV=,I1,5X,4HMSS=,I2,4X,4HMSF=,I3)
1840 FORMAT (1H0,20X,29HFREE-SLIP WALLS ARE SPECIFIED)
1850 FORMAT (1H0,20X,27HNO-SLIP WALLS ARE SPECIFIED)
1860 FORMAT (1H0,10X,21HMOLECULAR VISCOSITY -,//21X,4HCMU=,E10.4,18H (L
    1BF-S/FT2)  CLA=,E11.4,17H (LBF-S/FT2)  CK=,E10.4,16H (LBF/S-R)  EM
    2U=,F4.2,6H  ELA=,F4.2,5H  EK=,F4.2)
1870 FORMAT (1H0,10X,21HMOLECULAR VISCOSITY -,//21X,4HCMU=,E10.4,13H (P
    1A-S)  CLA=,E11.4,12H (PA-S)  CK=,E10.4,14H (W/M-K)  EMU=,F4.2,6H
    2ELA=,F4.2,5H  EK=,F4.2)
1880 FORMAT (1H ,10X,48H------------------------------------------------
    1-,61H-------------------------------------------------------------
    2 ,7H-------)
1890 FORMAT (1H ,20X,33HADIABATIC UPPER WALL IS SPECIFIED)
```

```
 1900 FORMAT (1H ,20X,15HIW IS SPECIFIED)
 1910 FORMAT (1H ,20X,39HADIABATIC LOWER CENTERBODY IS SPECIFIED)
 1920 FORMAT (1H ,20X,44HADIABATIC LOWER DUAL FLOW SPACE BOUNDARY IS ,9H
     1SPECIFIED)
 1930 FORMAT (1H ,20X,44HADIABATIC UPPER DUAL FLOW SPACE BOUNDARY IS ,9H
     1SPECIFIED)
 1940 FORMAT (1H ,20X,16HTCB IS SPECIFIED)
 1950 FORMAT (1H ,20X,15HTL IS SPECIFIED)
 1960 FORMAT (1H ,20X,15HTU IS SPECIFIED)
 1970 FORMAT (1HO,10X,18HTURBULENCE MODEL -,//21X,21HNO MODEL IS SPECIFI
     1ED)
 1980 FORMAT (1HO,10X,18HTURBULENCE MODEL -,//21X,38HMIXING-LENGTH MODEL
     1 IS SPECIFIED. CAL=,F4.2,2X,5HIMLM=,I2,2X,5HCML1=,F5.3,2X,5HCML2=
     2 ,F5.3,2X,4HPRT=,F4.2)
 1990 FORMAT (1HO,10X,18HTURBULENCE MODEL -,//21X,45HTURBULENCE ENERGY E
     1QUATION MODEL IS SPECIFIED)
 2000 FORMAT (1HO,20X,4HCAL=,F4.2,2X,4HCQL=,F5.2,2X,5HCQMU=,F4.2,2X,5HIM
     1LM=,I2,2X,5HCML1=,F5.3,2X,5HCML2=,F5.3,2X,4HPRT=,F4.2)
 2010 FORMAT (1HO,10X,18HTURBULENCE MODEL -,//21X,62HTURBULENCE ENERGY -
     1 DISSIPATION RATE EQUATIONS MODEL IS SPECIF,3HIED)
 2020 FORMAT (1HO,20X,7HIINLET=,I1,2X,7HIEXITT=,I1,2X,4HIEX=,I1,5X,7HISU
     1PER=,I1,2X,4HDYW=,F6.4,2X,5HIVBC=,I1,2X,5HINBC=,I1,2X,6HIWALL=,I1,
     2 2X,7HIWALLO=,I1,2X,4HALI=,F4.2,2X,4HALE=,F4.2,/,21X,4HALW=,F4.2,2
     3 X,6HNSTAG=,I1,3X,4HNPE=,I4,2X,4HPEI=,F10.5)
 2030 FORMAT (1HO,20X,4HCAL=,F4.2,2X,5HCQMU=,F4.2,2X,3HC1=,F4.2,2X,3HC2=
     1 ,F4.2,2X,5HSIGO=,F4.2,2X,5HSIGE=,F4.2,2X,5HBFST=,F4.2,2X,4HPRT=
     2 ,F4.2,2X,5HSTBQ=,F6.4,2X,5HSTBE=,F6.4)
 2040 FORMAT (1HO,10X,26HVARIABLE GRID PARAMETERS -,//21X,4HIST=,I1,3X,5
     1 HMVCB=,I2,3X,5HMVCT=,I2,3X,4HIQS=,I1,3X,6HNIQSS=,I1,3X,6HNIQSF=
     2 ,I1,3X,6HNVCMI=,I3,3X,6HILLQS=,I2,3X,4HSQS=,F5.2,3X,4HCQS=,F5.3)
 2050 FORMAT (1HO,63H***** INCOMPATIBLE TURBULENCE MODEL - GEOMETRY PARA
     1METERS *****)
 2060 FORMAT (1HO,51H***** INCOMPATIBLE DUAL FLOW SPACE PARAMETERS *****
     1 )
 2070 FORMAT (1HO,29H***** NVCMI MUST BE ODD *****)
 2080 FORMAT (1HO,52H***** INCOMPATIBLE DUAL FLOW SPACE - SUBCYCLED GRID
     1 ,16HPARAMETERS *****)
 2090 FORMAT (1HO,50H***** INCOMPATIBLE SUBCYCLED GRID PARAMETERS *****)
 2100 FORMAT (1HO,36H***** NVCM IS GREATER THAN 200 AT N=,I6,34H. CHECK
     1LAST SOLUTION PLANE. *****)
 2110 FORMAT (1H ,18X,4HNVC=,I3,5X,2HP(,I2,1H,,I2,2H)=,F10.5,5X,2HP(,I2,
     1 1H,,I2,2H)=,F10.5,5X,2HP(,I2,1H,,I2,2H)=,F10.5)
 2120 FORMAT (1H ,10X,2HN=,I6,13X,2HP(,I2,1H,,I2,2H)=,F10.5,5X,2HP(,I2,1
     1 H,,I2,2H)=,F10.5,5X,2HP(,I2,1H,,I2,2H)=,F10.5,5X,5HNTST=,I5)
 2130 FORMAT (1HO,48H***** INCOMPATIBLE QUICK SOLVER PARAMETERS *****)
 2140 FORMAT (1HO,53H***** ISUPER MUST BE GREATER THAN OR EQUAL TO 0 ***
     1**)
 2150 FORMAT (1HO,65H***** INCOMPATIBLE WALL GEOMETRY AND/OR BOUNDARY CO
     1NDITIONS *****)
      END
```

```
      SUBROUTINE GEOM
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE WALL RADIUS AND SLOPE
C
C     ****************************************************************
C
*CALL,MCC
      GO TO (10,30,120,170), NGEOM
C
C     CONSTANT AREA WALL CASE
C
   10 WRITE (6,230)
      IF (IUI.EQ.1) WRITE (6,250) XI,RI,XE
      IF (IUI.EQ.2) WRITE (6,260) XI,RI,XE
      LT=LMAX
      XT=XE
      RT=RI
      RE=RI
      DO 20 L=1,LMAX
      YW(L)=RI
      NXNY(L)=0.0
   20 CONTINUE
      GO TO 210
C
C     CIRCULAR-ARC, CONICAL WALL CASE
C
   30 WRITE (6,230)
      IF (RCI.EQ.0.0.OR.RCT.EQ.0.0) GO TO 200
      ANI=ANGI*3.141593/180.0
      ANE=ANGE*3.141593/180.0
      XTAN=XI+RCI*SIN(ANI)
      RTAN=RI+RCI*(COS(ANI)-1.0)
      RT1=RT-RCT*(COS(ANI)-1.0)
      XT1=XTAN+(RTAN-RT1)/TAN(ANI)
      IF (XT1.GE.XTAN) GO TO 40
      XT1=XTAN
      RT1=RTAN
   40 XT=XT1+RCT*SIN(ANI)
      XT2=XT+RCT*SIN(ANE)
      RT2=RT+RCT*(1.0-COS(ANE))
      RE=RT2+(XE-XT2)*TAN(ANE)
      LT=1
      IF (IUI.EQ.1) WRITE (6,270) XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XT,RE
      IF (IUI.EQ.2) WRITE (6,280) XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XT,RE
      DO 110 L=1,LMAX
      IF (XP(L).LE.XTAN) GO TO 50
      IF (XP(L).GT.XTAN.AND.XP(L).LE.XT1) GO TO 60
      IF (XP(L).GT.XT1.AND.XP(L).LE.XT) GO TO 70
      IF (XP(L).GT.XT.AND.XP(L).LE.XT2) GO TO 80
      GO TO 90
C
   50 YW(L)=RI+RCI*(COS(ASIN((XP(L)-XI)/RCI))-1.0)
      NXNY(L)=(XP(L)-XI)/(YW(L)-RI+RCI)
      GO TO 100
C
   60 YW(L)=RT1+(XT1-XP(L))*TAN(ANI)
      NXNY(L)=TAN(ANI)
      GO TO 100
C
   70 YW(L)=RT+RCT*(1.0-COS(ASIN((XT-XP(L))/RCT)))
      NXNY(L)=(XT-XP(L))/(RCT+RT-YW(L))
      GO TO 100
C
   80 YW(L)=RT+RCT*(1.0-COS(ASIN((XP(L)-XT)/RCT)))
      NXNY(L)=(XT-XP(L))/(RCT+RT-YW(L))
      GO TO 100
C
   90 YW(L)=RT2+(XP(L)-XT2)*TAN(ANE)
      NXNY(L)=-TAN(ANE)
C
```

```
  100 IF (L.EQ.1) GO TO 110
      IF (YW(L).LT.YW(LT)) LT=L
  110 CONTINUE
      GO TO 210
C
C     GENERAL WALL CASE - INPUT WALL COORDINATES
C
  120 WRITE (6,240)
      WRITE (6,230)
      YW(1)=YWI(1)
      YW(LMAX)=YWI(NWPTS)
      RI=YW(1)
      RE=YW(LMAX)
      LT=1
      DO 130 L=2,NWPTS
      IF (YWI(L).LE.YWI(LT)) LT=L
  130 CONTINUE
      XT=XWI(LT)
      RT=YWI(LT)
      IF (IUI.EQ.1) WRITE (6,290) XT,RT,IINT,IDIF
      IF (IUI.EQ.2) WRITE (6,300) XT,RT,IINT,IDIF
      LT=1
      L1=LMAX-1
      IPP=1
      DO 140 L=2,L1
      CALL MTLUP (XP(L),YW(L),IINT,NWPTS,NWPTS,1,IPP,XWI,YWI)
      IF (L.EQ.1) GO TO 140
      IF (YW(L).LE.YW(LT)) LT=L
  140 CONTINUE
      LDUM=NWPTS
      IF (LMAX.GT.NWPTS) LDUM=LMAX
      DO 160 L=1,LDUM
      IF (L.GT.LMAX) GO TO 150
      SLOPE=DIF(L,IDIF,LMAX,XP,YW)
      NXNY(L)=-SLOPE
  150 IF (L.LE.NWPTS.AND.L.LE.LMAX) WRITE (6,330) L,XWI(L),YWI(L),XP(L)
     1 ,YW(L),SLOPE
      IF (L.GT.NWPTS.AND.L.LE.LMAX) WRITE (6,340) L,XP(L),YW(L),SLOPE
      IF (L.LE.NWPTS.AND.L.GT.LMAX) WRITE (6,350) L,XWI(L),YWI(L)
  160 CONTINUE
      GO TO 210
C
C     GENERAL WALL CASE - INPUT WALL RADIUS AND SLOPE
C
  170 WRITE (6,240)
      WRITE (6,230)
      RI=YW(1)
      RE=YW(LMAX)
      LT=1
      DO 180 L=2,LMAX
      IF (YW(L).LE.YW(LT)) LT=L
  180 CONTINUE
      XT=XP(LT)
      RT=YW(LT)
      IF (IUI.EQ.1) WRITE (6,310) XT,RT
      IF (IUI.EQ.2) WRITE (6,320) XT,RT
      DO 190 L=1,LMAX
      SLOPE=-NXNY(L)
      WRITE (6,360) L,XP(L),YW(L),SLOPE
  190 CONTINUE
      GO TO 210
C
  200 WRITE (6,390)
      IERR=1
      RETURN
C
  210 IF (JFLAG.EQ.0) RETURN
      XWL=XP(LJET-1)
      IF (JFLAG.EQ.-1) GO TO 220
      IF (IUI.EQ.1) WRITE (6,370) XWL,LJET,LMAX
      IF (IUI.EQ.2) WRITE (6,380) XWL,LJET,LMAX
      RETURN
```

```
  220 IF (IUI.EQ.1) WRITE (6,400) XWL
      IF (IUI.EQ.2) WRITE (6,410) XWL
      RETURN
C
C     FORMAT STATEMENTS
C
  230 FORMAT (1HO,10X,15HDUCT GEOMETRY -)
  240 FORMAT (1H1)
  250 FORMAT (1HO,20X,46HA CONSTANT AREA DUCT HAS BEEN SPECIFIED BY XI=
     1 ,F8.4,10H (IN), RI=,F8.4,14H (IN), AND XE=,F8.4,5H (IN))
  260 FORMAT (1HO,20X,46HA CONSTANT AREA DUCT HAS BEEN SPECIFIED BY XI=
     1 ,F8.4,10H (CM), RI=,F8.4,14H (CM), AND XE=,F8.4,5H (CM))
  270 FORMAT (1HO,20X,56HA CIRCULAR-ARC, CONICAL NOZZLE HAS BEEN SPECIFI
     1ED BY XI=,F8.4,10H (IN), RI=,F8.4,6H (IN),,/,21X,3HRT=,F8.4,10H (I
     2N), XE=,F8.4,11H (IN), RCI=,F8.4,11H (IN), RCT=,F8.4,12H (IN), ANG
     3I=,F6.2,7H (DEG),,/,21X,9HAND ANGE=,F6.2,35H (DEG). THE COMPUTED V
     4ALUES ARE XT=,F8.4,13H (IN) AND RE=,F8.4,6H (IN).)
  280 FORMAT (1HO,20X,56HA CIRCULAR-ARC, CONICAL NOZZLE HAS BEEN SPECIFI
     1ED BY XI=,F8.4,10H (CM), RI=,F8.4,6H (CM),,/,21X,3HRT=,F8.4,10H (C
     2M), XE=,F8.4,11H (CM), RCI=,F8.4,11H (CM), RCT=,F8.4,12H (CM), ANG
     3I=,F6.2,7H (DEG),,/,21X,9HAND ANGE=,F6.2,35H (DEG). THE COMPUTED V
     4ALUES ARE XT=,F8.4,13H (CM) AND RE=,F8.4,6H (CM).)
  290 FORMAT (1HO,20X,45HA GENERAL WALL HAS BEEN SPECIFIED BY THE FOLL,2
     1 1HOWING PARAMETERS, XT=,F8.4,10H (IN), RT=,F8.4,6H (IN),,/,21X,5H
     2IINT=,I1,7H, IDIF=,I1,1H,,//22X,1HL,10X,7HXWI(IN),10X,7HYWI(IN),11
     3 X,6HXP(IN),11X,6HYW(IN),12X,5HSLOPE,/)
  300 FORMAT (1HO,20X,45HA GENERAL WALL HAS BEEN SPECIFIED BY THE FOLL,2
     1 1HOWING PARAMETERS, XT=,F8.4,10H (CM), RT=,F8.4,6H (CM),,/,21X,5H
     2IINT=,I1,7H, IDIF=,I1,1H,,//22X,1HL,10X,7HXWI(CM),10X,7HYWI(CM),11
     3 X,6HXP(CM),11X,6HYW(CM),12X,5HSLOPE,/)
  310 FORMAT (1HO,20X,45HA GENERAL WALL HAS BEEN SPECIFIED BY THE FOLL,2
     1 1HOWING PARAMETERS, XT=,F8.4,10H (IN), RT=,F8.4,6H (IN),,//22X,1H
     2L,11X,6HXP(IN),11X,6HYW(IN),12X,5HSLOPE,/)
  320 FORMAT (1HO,20X,45HA GENERAL WALL HAS BEEN SPECIFIED BY THE FOLL,2
     1 1HOWING PARAMETERS, XT=,F8.4,10H (CM), RT=,F8.4,6H (CM),,//22X,1H
     2L,11X,6HXP(CM),11X,6HYW(CM),12X,5HSLOPE,/)
  330 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
  340 FORMAT (1H ,20X,I2,41X,F10.4,7X,F10.4,7X,F10.4)
  350 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4)
  360 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4,7X,F10.4)
  370 FORMAT (1HO,20X,43HA FREE-JET CALCULATION HAS BEEN REQUESTED. ,20H
     1 THE WALL ENDS AT X=,F8.4,11H (IN).  THE,/,21X,14HMESH POINTS L=
     2 ,I3,6H TO L=,I3,55H ARE AN INITIAL APPROXIMATION TO THE FREE-JET
     3BOUNDARY.)
  380 FORMAT (1HO,20X,43HA FREE-JET CALCULATION HAS BEEN REQUESTED. ,20H
     1 THE WALL ENDS AT X=,F8.4,11H (CM).  THE,/,21X,14HMESH POINTS L=
     2 ,I3,6H TO L=,I3,55H ARE AN INITIAL APPROXIMATION TO THE FREE-JET
     3BOUNDARY.)
  390 FORMAT (1HO,44H***** RCI OR RCT WAS SPECIFIED AS ZERO *****)
  400 FORMAT (1HO,20X,54HTHE WALL CONTOUR HAS AN EXPANSION CORNER LOCATE
     1D AT X=,F8.4,6H (IN).)
  410 FORMAT (1HO,20X,54HTHE WALL CONTOUR HAS AN EXPANSION CORNER LOCATE
     1D AT X=,F8.4,6H (CM).)
      END
```

```
      SUBROUTINE GEOMCB
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE CENTERBODY RADIUS AND SLOPE
C
C     ****************************************************************
C
*CALL,MCC
      GO TO (10,30,120,160), NGCB
C
C     CYLINDRICAL CENTERBODY CASE
C
   10 IF (IUI.EQ.1) WRITE (6,210) XICB,RICB,XECB
      IF (IUI.EQ.2) WRITE (6,220) XICB,RICB,XECB
      DO 20 L=1,LMAX
      YCB(L)=RICB
      NXNYCB(L)=0.0
   20 CONTINUE
      RETURN
C
C     CIRCULAR-ARC, CONICAL CENTERBODY CASE
C
   30 RICB=2.0*RTCB-RICB
      IF (RCICB.EQ.0.0.OR.RCTCB.EQ.0.0) GO TO 190
      ANI=ANGICB*3.141593/180.0
      ANE=ANGECB*3.141593/180.0
      XTAN=XICB+RCICB*SIN(ANI)
      RTAN=RICB+RCICB*(COS(ANI)-1.0)
      RT1=RTCB-RCTCB*(COS(ANI)-1.0)
      XT1=XTAN+(RTAN-RT1)/TAN(ANI)
      IF (XT1.GE.XTAN) GO TO 40
      XT1=XTAN
      RT1=RTAN
   40 XTCB=XT1+RCTCB*SIN(ANI)
      XT2=XTCB+RCTCB*SIN(ANE)
      RT2=RTCB+RCTCB*(1.0-COS(ANE))
      RECB=RT2+(XECB-XT2)*TAN(ANE)
      RICB=2.0*RTCB-RICB
      RECB=2.0*RTCB-RECB
      IF (IUI.EQ.1) WRITE (6,230) XICB,RICB,RTCB,XECB,RCICB,RCTCB,ANGICB
     1 ,ANGECB,XTCB,RECB
      IF (IUI.EQ.2) WRITE (6,240) XICB,RICB,RTCB,XECB,RCICB,RCTCB,ANGICB
     1 ,ANGECB,XTCB,RECB
      RICB=2.0*RTCB-RICB
      RECB=2.0*RTCB-RECB
      DO 110 L=1,LMAX
      IF (XP(L).LE.XTAN) GO TO 50
      IF (XP(L).GT.XTAN.AND.XP(L).LE.XT1) GO TO 60
      IF (XP(L).GT.XT1.AND.XP(L).LE.XTCB) GO TO 70
      IF (XP(L).GT.XTCB.AND.XP(L).LE.XT2) GO TO 80
      GO TO 90
C
   50 YCB(L)=RICB+RCICB*(COS(ASIN((XP(L)-XICB)/RCICB))-1.0)
      NXNYCB(L)=(XP(L)-XICB)/(YCB(L)-RICB+RCICB)
      GO TO 100
C
   60 YCB(L)=RT1+(XT1-XP(L))*TAN(ANI)
      NXNYCB(L)=TAN(ANI)
      GO TO 100
C
   70 YCB(L)=RTCB+RCTCB*(1.0-COS(ASIN((XTCB-XP(L))/RCTCB)))
      NXNYCB(L)=(XTCB-XP(L))/(RCTCB+RTCB-YCB(L))
      GO TO 100
C
   80 YCB(L)=RTCB+RCTCB*(1.0-COS(ASIN((XP(L)-XTCB)/RCTCB)))
      NXNYCB(L)=(XTCB-XP(L))/(RCTCB+RTCB-YCB(L))
      GO TO 100
C
   90 YCB(L)=RT2+(XP(L)-XT2)*TAN(ANE)
      NXNYCB(L)=-TAN(ANE)
C
```

```
  100 YCB(L)=2.0*RTCB-YCB(L)
      NXNYCB(L)=-NXNYCB(L)
      IF (YCB(L).GE.0.0.OR.NDIM.EQ.0) GO TO 110
      YCB(L)=0.0
      NXNYCB(L)=0.0
  110 CONTINUE
      RETURN
C
C     GENERAL CENTERBODY CASE - INPUT CENTERBODY COORDINATES
C
  120 WRITE (6,200)
      IF (IUI.EQ.1) WRITE (6,250) IINTCB,IDIFCB
      IF (IUI.EQ.2) WRITE (6,260) IINTCB,IDIFCB
      L1=LMAX-1
      IPP=1
      DO 130 L=1,LMAX
      CALL MTLUP (XP(L),YCB(L),IINTCB,NCBPTS,NCBPTS,1,IPP,XCBI,YCBI)
  130 CONTINUE
      LDUM=NCBPTS
      IF (LMAX.GT.NCBPTS) LDUM=LMAX
      DO 150 L=1,LDUM
      IF (L.GT.LMAX) GO TO 140
      SLOPE=DIF(L,IDIFCB,LMAX,XP,YCB)
      NXNYCB(L)=-SLOPE
      IF (YCB(L).GE.0.0.OR.NDIM.EQ.0) GO TO 140
      YCB(L)=0.0
      NXNYCB(L)=0.0
      SLOPE=-NXNYCB(L)
  140 IF (L.LE.NCBPTS.AND.L.LE.LMAX) WRITE (6,290) L,XCBI(L),YCBI(L),XP
     1 (L),YCB(L),SLOPE
      IF (L.GT.NCBPTS.AND.L.LE.LMAX) WRITE (6,300) L,XP(L),YCB(L),SLOPE
      IF (L.LE.NCBPTS.AND.L.GT.LMAX) WRITE (6,310) L,XCBI(L),YCBI(L)
  150 CONTINUE
      RETURN
C
C     GENERAL CENTERBODY CASE - INPUT CENTERBODY RADIUS AND SLOPE
C
  160 WRITE (6,200)
      IF (IUI.EQ.1) WRITE (6,270)
      IF (IUI.EQ.2) WRITE (6,280)
      DO 180 L=1,LMAX
      IF (YCB(L).GE.0.0.OR.NDIM.EQ.0) GO TO 170
      YCB(L)=0.0
      NXNYCB(L)=0.0
  170 SLOPE=-NXNYCB(L)
      WRITE (6,320) L,XP(L),YCB(L),SLOPE
  180 CONTINUE
      RETURN
C
  190 WRITE (6,330)
      IERR=1
      RETURN
C
C     FORMAT STATEMENTS
C
  200 FORMAT (1H1)
  210 FORMAT (1H0,20X,52HA CYLINDRICAL CENTERBODY HAS BEEN SPECIFIED BY
     1XICB=,F8.4,12H (IN), RICB=,F8.4,16H (IN), AND XECB=,F8.4,5H (IN))
  220 FORMAT (1H0,20X,52HA CYLINDRICAL CENTERBODY HAS BEEN SPECIFIED BY
     1XICB=,F8.4,12H (CM), RICB=,F8.4,16H (CM), AND XECB=,F8.4,5H (CM))
  230 FORMAT (1H0,20X,62HA CIRCULAR-ARC, CONICAL CENTERBODY HAS BEEN SPE
     1CIFIED BY XICB=,F8.4,5H (IN),7H, RICB=,F8.4,6H (IN),,/,21X,5HRTCB=
     2 ,F8.4,7H (IN), ,5HXECB=,F8.4,5H (IN),8H, RCICB=,F8.4,5H (IN),8H,
     3RCTCB=,F8.4,5H (IN),9H, ANGICB=,F6.2,7H (DEG),,/,21X,11HAND ANGECB
     4=,F6.2,8H (DEG). ,29HTHE COMPUTED VALUES ARE XTCB=,F8.4,5H (IN),10
     5 H AND RECB=,F8.4,6H (IN).)
  240 FORMAT (1H0,20X,62HA CIRCULAR-ARC, CONICAL CENTERBODY HAS BEEN SPE
     1CIFIED BY XICB=,F8.4,5H (CM),7H, RICB=,F8.4,6H (CM),,/,21X,5HRTCB=
     2 ,F8.4,7H (CM). .5HXECB=,F8.4,5H (CM),8H, RCICB=,F8.4,5H (CM),8H,
     3RCTCB=,F8.4,5H (CM),9H, ANGICB=,F6.2,7H (DEG),,/,21X,11HAND ANGECB
     4=,F6.2,8H (DEG). ,29HTHE COMPUTED VALUES ARE XTCB=,F8.4,5H (CM),10
     5 H AND RECB=,F8.4,6H (CM).)
```

```
      250 FORMAT (1H0,20X,47HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE
     1 ,29HFOLLOWING PARAMETERS, IINTCB=,I1,9H, IDIFCB=,I1,1H,,//22X,1HL
     2 ,10X,8HXCBI(IN),10X,8HYCBI(IN),10X,6HXP(IN),10X,7HYCB(IN),11X,5HS
     3LOPE,/)
      260 FORMAT (1H0,20X,47HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE
     1 ,29HFOLLOWING PARAMETERS, IINTCB=,I1,9H, IDIFCB=,I1,1H,,//22X,1HL
     2 ,10X,8HXCBI(CM),10X,8HYCBI(CM),10X,6HXP(CM),10X,7HYCB(CM),11X,5HS
     3LOPE,/)
      270 FORMAT (1H0,20X,47HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE
     1 ,21HFOLLOWING PARAMETERS,,//22X,1HL,12X,6HXP(IN),10X,7HYCB(IN),11
     2 X,5HSLOPE,/)
      280 FORMAT (1H0,20X,47HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE
     1 ,21HFOLLOWING PARAMETERS,,//22X,1HL,12X,6HXP(CM),10X,7HYCB(CM),11
     2 X,5HSLOPE,/)
      290 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
      300 FORMAT (1H ,20X,I2,41X,F10.4,7X,F10.4,7X,F10.4)
      310 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4)
      320 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4,7X,F10.4)
      330 FORMAT (1H0,48H***** RCICB OR RCTCB WAS SPECIFIED AS ZERO *****)
          END
```

```
      SUBROUTINE GEOMLU
C
C     **************************************************************
C
C     THIS SUBROUTINE CALCULATES THE DUAL FLOW SPACE BOUNDARY RADIUS
C     AND SLOPES
C
C     **************************************************************
C
*CALL,MCC
      GO TO (10,100), NDFS
C
C     INPUT DUAL FLOW SPACE BOUNDARY COORDINATES
C
   10 WRITE (6,120)
      WRITE (6,140)
      IF (IUI.EQ.1) WRITE (6,180) IINTDFS,IDIFDFS
      IF (IUI.EQ.2) WRITE (6,190) IINTDFS,IDIFDFS
      IPP=1
      DO 20 L=LDFSS,LDFSF
      CALL MTLUP (XP(L),YL(L),IINTDFS,NLPTS,NLPTS,1,IPP,XLI,YLI)
   20 CONTINUE
      LDUM=NLPTS
      IF (LDFSF.GT.NLPTS) LDUM=LDFSF
      LDF=0
      DO 30 L=LDFSS,LDFSF
      LDF=LDF+1
      XWI(LDF)=XP(L)
      YWI(LDF)=YL(L)
   30 CONTINUE
      LMDF=LDFSF-LDFSS+1
      LDF=0
      DO 50 L=1,LDUM
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      IF (LDFS.EQ.0) GO TO 40
      LDF=LDF+1
      SLOPE=DIF(LDF,IDIFDFS,LMDF,XWI,YWI)
      NXNYL(L)=-SLOPE
      IF (YL(L).GE.0.0.OR.NDIM.EQ.0) GO TO 40
      YL(L)=0.0
      NXNYL(L)=0.0
      SLOPE=-NXNYL(L)
   40 IF (L.LE.NLPTS.AND.LDFS.EQ.1) WRITE (6,220) L,XLI(L),YLI(L),XP(L)
     1 ,YL(L),SLOPE
      IF (L.GT.NLPTS.AND.LDFS.EQ.1) WRITE (6,230) L,XP(L),YL(L),SLOPE
      IF (L.LE.NLPTS.AND.LDFS.EQ.0) WRITE (6,240) L,XLI(L),YLI(L)
   50 CONTINUE
C
      WRITE (6,130)
      IF (IUI.EQ.1) WRITE (6,200)
      IF (IUI.EQ.2) WRITE (6,210)
      IPP=1
      DO 60 L=LDFSS,LDFSF
      CALL MTLUP (XP(L),YU(L),IINTDFS,NUPTS,NUPTS,1,IPP,XUI,YUI)
   60 CONTINUE
      LDUM=NUPTS
      IF (LDFSF.GT.NUPTS) LDUM=LDFSF
      LDF=0
      DO 70 L=LDFSS,LDFSF
      LDF=LDF+1
      XWI(LDF)=XP(L)
      YWI(LDF)=YU(L)
   70 CONTINUE
      LMDF=LDFSF-LDFSS+1
      LDF=0
      DO 90 L=1,LDUM
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      IF (LDFS.EQ.0) GO TO 80
      LDF=LDF+1
      SLOPE=DIF(LDF,IDIFDFS,LMDF,XWI,YWI)
```

```
      NXNYU(L)=-SLOPE
      IF (YU(L).GE.0.0.OR.NDIM.EQ.0) GO TO 80
      YU(L)=0.0
      NXNYU(L)=0.0
      SLOPE=-NXNYU(L)
   80 IF (L.LE.NUPTS.AND.LDFS.EQ.1) WRITE (6,220) L,XUI(L),YUI(L),XP(L)
     1 ,YU(L),SLOPE
      IF (L.GT.NUPTS.AND.LDFS.EQ.1) WRITE (6,230) L,XP(L),YU(L),SLOPE
      IF (L.LE.NUPTS.AND.LDFS.EQ.0) WRITE (6,240) L,XUI(L),YUI(L)
   90 CONTINUE
      RETURN
C
C     INPUT DUAL FLOW SPACE BOUNDARY RADIUS AND SLOPE
C
  100 WRITE (6,120)
      WRITE (6,140)
      IF (IUI.EQ.1) WRITE (6,150)
      IF (IUI.EQ.2) WRITE (6,160)
      DO 110 L=LDFSS,LDFSF
      SLOPEL=-NXNYL(L)
      SLOPEU=-NXNYU(L)
      WRITE (6,170) L,XP(L),YL(L),SLOPEL,YU(L),SLOPEU
  110 CONTINUE
      RETURN
C
C     FORMAT STATEMENTS
C
  120 FORMAT (1H1)
  130 FORMAT (1H0)
  140 FORMAT (1H0,10X,35HDUAL FLOW SPACE BOUNDARY GEOMETRY -)
  150 FORMAT (1H0,20X,41HGENERAL BOUNDARIES HAVE BEEN SPECIFIED BY,26H T
     1HE FOLLOWING PARAMETERS,,//22X,1HL,11X,6HXP(IN),11X,6HYL(IN),11X,6
     2 HSLOPEL,11X,6HYU(IN),11X,6HSLOPEU,/)
  160 FORMAT (1H0,20X,41HGENERAL BOUNDARIES HAVE BEEN SPECIFIED BY,26H T
     1HE FOLLOWING PARAMETERS,,//22X,1HL,11X,6HXP(CM),11X,6HYL(CM),11X,6
     2 HSLOPEL,11X,6HYU(CM),11X,6HSLOPEU,/)
  170 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
  180 FORMAT (1H0,20X,46HGENERAL BOUNDARIES HAVE BEEN SPECIFIED BY THE ,
     1 30HFOLLOWING PARAMETERS, IINTDFS=,I1,10H, IDIFDFS=,I1,1H,,//22X,1
     2 HL,10X,7HXLI(IN),10X,7HYLI(IN),11X,6HXP(IN),11X,6HYL(IN),11X,6HSL
     3OPEL,/)
  190 FORMAT (1H0,20X,46HGENERAL BOUNDARIES HAVE BEEN SPECIFIED BY THE ,
     1 30HFOLLOWING PARAMETERS, IINTDFS=,I1,10H, IDIFDFS=,I1,1H,,//22X,1
     2 HL,10X,7HXLI(CM),10X,7HYLI(CM),11X,6HXP(CM),11X,6HYL(CM),11X,6HSL
     3OPEL,/)
  200 FORMAT (1H0,21X,1HL,10X,7HXUI(IN),10X,7HYUI(IN),11X,6HXP(IN),11X,6
     1 HYU(IN),11X,6HSLOPEU,/)
  210 FORMAT (1H0,21X,1HL,10X,7HXUI(CM),10X,7HYUI(CM),11X,6HXP(CM),11X,6
     1 HYU(CM),11X,6HSLOPEU,/)
  220 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
  230 FORMAT (1H ,20X,I2,41X,F10.4,7X,F10.4,7X,F10.4)
  240 FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4)
      END
```

```
      SUBROUTINE MTLUP (X,Y,M,N,MAX,NTAB,I,VARI,VARD)
C
C     ************************************************************
C
C     THIS SUBROUTINE IS CALLED BY SUBROUTINES GEOM, GEOMCB, AND GEOMLU
C     TO INTERPOLATE FOR WALL COORDINATES FOR THE TABULAR INPUT CASE.
C     SUBROUTINE MTLUP WAS TAKEN FROM THE NASA-LANGLEY PROGRAM
C     LIBRARY. THE DATE OF THIS VERSION IS 09-12-69.
C
C     ************************************************************
C
C        MODIFICATION OF LIBRARY INTERPOLATION SUBROUTINE  FTLUP
C       MULTIPLE TABLE LOOK-UP ON ONE INDEPENDENT VARIABLE TABLE
C      USES AN EXTERNAL INTERVAL POINTER (I) TO START SEARCH
C      I LESS THAN O WILL CHECK MONOTONICITY
C
      DIMENSION VARI(1), VARD(MAX,1), Y(1), V(3), YY(2)
      LOGICAL EX
C
      IF (M.EQ.O) GO TO 170
      IF (N.LE.1) GO TO 170
      EX=.FALSE.
      IF (I.GE.O) GO TO 60
      IF (N.LT.2) GO TO 60
C
C     MONOTONICITY CHECK
C
      IF (VARI(2)-VARI(1)) 20,20,40
C
C     ERROR IN MONOTONICITY
C
   10 K=LOCF(VARI(1))
      WRITE (6,190) J,K,(VARI(J),J=1,N)
      CALL EXIT
C
C     MONOTONIC DECREASING
C
   20 DO 30 J=2,N
      IF (VARI(J)-VARI(J-1)) 30,10,10
   30 CONTINUE
      GO TO 60
C
C     MONOTONIC INCREASING
C
   40 DO 50 J=2,N
      IF (VARI(J)-VARI(J-1)) 10,10,50
   50 CONTINUE
C
C     INTERPOLATION
C
   60 IF (I.LE.O) I=1
      IF (I.GE.N) I=N-1
C
C     LOCATE I INTERVAL (X(I).LE.X.LT.X(I+1))
C
      IF ((VARI(I)-X)*(VARI(I+1)-X)) 100,100,70
C
C     IN GIVES DIRECTION FOR SEARCH OF INTERVALS
C
   70 IN=SIGN(1.O,(VARI(I+1)-VARI(I))*(X-VARI(I)))
C
C     IF X OUTSIDE ENDPOINTS, EXTRAPOLATE FROM END INTERVAL
C
   80 IF ((I+IN).LE.O) GO TO 90
      IF ((I+IN).GE.N) GO TO 90
      I=I+IN
      IF ((VARI(I)-X)*(VARI(I+1)-X)) 100,100,80
C
C     EXTRAPOLATION
C
   90 EX=.TRUE.
  100 IF (M.EQ.2) GO TO 120
```

```
C
C     FIRST ORDER
C
      DO 110 NT=1,NTAB
  110 Y(NT)=(VARD(I,NT)*(VARI(I+1)-X)-VARD(I+1,NT)*(VARI(I)-X))/(VARI(I+
     1 1)-VARI(I))
      IF (EX) I=I+IN
      RETURN
C
C     SECOND ORDER
C
  120 IF (N.EQ.2) GO TO 10
      IF (I.EQ.(N-1)) GO TO 140
      IF (I.EQ.1) GO TO 130
C
C     PICK THIRD POINT
C
      SK=VARI(I+1)-VARI(I)
      IF ((SK*(X-VARI(I-1))).LT.(SK*(VARI(I+2)-X))) GO TO 140
  130 L=I
      GO TO 150
  140 L=I-1
  150 V(1)=VARI(L)-X
      V(2)=VARI(L+1)-X
      V(3)=VARI(L+2)-X
      DO 160 NT=1,NTAB
      YY(1)=(VARD(L,NT)*V(2)-VARD(L+1,NT)*V(1))/(VARI(L+1)-VARI(L))
      YY(2)=(VARD(L+1,NT)*V(3)-VARD(L+2,NT)*V(2))/(VARI(L+2)-VARI(L+1))
  160 Y(NT)=(YY(1)*V(3)-YY(2)*V(1))/(VARI(L+2)-VARI(L))
      IF (EX) I=I+IN
      RETURN
C
C     ZERO ORDER
C
  170 DO 180 NT=1,NTAB
  180 Y(NT)=VARD(1,NT)
      RETURN
C
C     FORMAT STATEMENTS
C
  190 FORMAT (1H1,49H TABLE BELOW OUT OF ORDER FOR MTLUP AT POSITION
     1 ,I5,/31H X TABLE IS STORED IN LOCATION ,O6,//(8G15.8))
      END
```

```
      FUNCTION DIF (L,M,NP,VARI,VARD)
C
C     ***********************************************************
C
C     THIS FUNCTION IS CALLED BY SUBROUTINES GEOM, GEOMCB, AND GEOMLU TO
C     CALCULATE THE WALL SLOPE FOR THE TABULAR INPUT CASE.  FUNCTION DIF
C     WAS TAKEN FROM THE NASA-LANGLEY PROGRAM LIBRARY.  THE DATE OF
C     THIS VERSION IS 8-1-68.
C
C     ***********************************************************
C
C        THIS FUNCTION SUBPROGRAM FINDS THE DERIVATIVE AT A GIVEN POINT,
C        L, FOR THE DESIRED X AND Y IN A GIVEN TABLE.  THE N-POINT
C        LAGRANGIAN FORMULA IS USED WHERE N IS ODD.
C
C        L = INTEGER, THE POINT OF X AND Y AT WHICH DERIVATIVE IS FOUND
C        M = INTEGER, 1-5, TO DETERMINE THE POINT FORMULA, N.  N=2*M+1
C        NP= INTEGER, THE NUMBER OF POINTS IN TABLE OF VARIABLES
C        VARI = ARRAY OF INDEPENDENT VARIABLE, X.  VARI(NP)
C        VARD = ARRAY OF DEPENDENT VARIABLE, Y.    VARD(NP)
C
      DIMENSION VARI(NP), VARD(NP), X(11), Y(11)
      DIF=17770000000000000000B
      IF (M.LT.1) RETURN
      N=2*M+1
      IF (M.GT.5.OR.N.GT.NP) RETURN
      M1=M+1
      M2=NP-M+1
      K=L
      IF (L.LE.M1.OR.N.EQ.NP) GO TO 10
      K=M1
      IF (L.LT.M2) GO TO 10
      K=L-(NP-N)
   10 MX=L-K
      DO 20 J=1,N
      MJ=MX+J
      X(J)=VARI(MJ)
   20 Y(J)=VARD(MJ)
      A=1.
      B=0.
      C=0.
      DO 40 J=1,N
      IF (J.EQ.K) GO TO 40
      P=1.
      DO 30 I=1,N
      IF (I.EQ.J) GO TO 30
      P=P*(X(J)-X(I))
   30 CONTINUE
      T=X(K)-X(J)
      B=B+Y(J)/(P*T)
      A=A*T
      C=C+1./T
   40 CONTINUE
      DIF=A*B+Y(K)*C
      RETURN
      END
```

```
      SUBROUTINE ONEDIM
C
C     ******************************************************************
C
C     THIS SUBROUTINE CALCULATES THE 1-D INITIAL-DATA SURFACE
C
C     ******************************************************************
C
*CALL,MCC
      IF (PT(1).NE.0.0.AND.TT(1).NE.0.0) GO TO 10
      IERR=1
      WRITE (6,200)
      RETURN
   10 MN3=0.01
      IF (N1D.EQ.-1.OR.N1D.GT.2) MN3=2.0
      NXCK=0
      ACOEF=2.0/(GAMMA+1.0)
      BCOEF=(GAMMA-1.0)/(GAMMA+1.0)
      CCOEF=(GAMMA+1.0)/2.0/(GAMMA-1.0)
      IF (N1D.LT.0) GO TO 30
C
C     OVERALL LOOP
C
      IF (NGCB.NE.0.OR.MDFS.NE.0) GO TO 20
      RSTAR=RT
      RSTARS=RT*RT
      GO TO 30
   20 RSTAR=YW(LT)-YU(LT)+YL(LT)-YCB(LT)
      RSTARS=YW(LT)**2-YU(LT)**2+YL(LT)**2-YCB(LT)**2
   30 DO 180 L=1,LMAX
      IF (L.EQ.1.AND.ISUPER.EQ.1) GO TO 180
      IF (N1D.LT.0) GO TO 60
      IF (NGCB.NE.0) GO TO 40
      IF (MDFS.NE.0) GO TO 40
      IF (XP(L).LT.XT) GO TO 60
      IF (XP(L).GT.XT) GO TO 50
      MN3=1.0
      GO TO 110
   40 IF (L.LT.LT) GO TO 60
      IF (L.GT.LT) GO TO 50
      MN3=1.0
      GO TO 110
   50 IF (NXCK.EQ.1) GO TO 60
      IF (N1D.EQ.1.OR.N1D.EQ.3) MN3=1.1
      IF (N1D.EQ.2.OR.N1D.EQ.4) MN3=0.9
      NXCK=1
   60 IF (NDIM.EQ.1) GO TO 70
      RAD=YW(L)-YU(L)+YL(L)-YCB(L)
      ARATIO=RAD/RSTAR
      GO TO 80
   70 RADS=YW(L)**2-YU(L)**2+YL(L)**2-YCB(L)**2
      ARATIO=RADS/RSTARS
C
C     NEWTON-RAPHSON ITERATION LOOP
C
   80 DO 100 ITER=1,100
      ABM=ACOEF+BCOEF*MN3*MN3
      ABMC=ABM**CCOEF
      FM=ABMC/MN3-ARATIO
      FPM=ABMC*(2.0*BCOEF*CCOEF/ABM-1.0/(MN3*MN3))
      OMN3=MN3
      MN3=OMN3-FM/FPM
      IF (OMN3.GT.0.99.AND.OMN3.LT.1.01) MN3=0.5*(OMN3+MN3)
      IF (MN3.GT.1.0.AND.OMN3.LT.1.0) MN3=0.99
      IF (MN3.LT.1.0.AND.OMN3.GT.1.0) MN3=1.01
      IF (N1D.EQ.-1.AND.MN3.LE.1.0) MN3=1.01
      IF (N1D.EQ.-2.AND.MN3.GE.1.0) MN3=0.99
      IF (MN3.GT.50.0) MN3=50.0
      IF (MN3.GE.0.0) GO TO 90
      MN3=-MN3
      GO TO 100
   90 IF (ABS(MN3-OMN3)/OMN3.LE.0.0005) GO TO 110
```

```
  100 CONTINUE
      WRITE (6,190) L
C
C     FILL IN 2-D ARRAYS LOOP
C
  110 LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DEM=1.0+GAM2*MN3*MN3
      DEMP=DEM**GAM1
      DNXNY=(NXNY(L)-NXNYCB(L))/FLOAT(M1)
      IF (MDFS.EQ.0.OR.LDFS.EQ.0) GO TO 120
      DNXNY1=(NXNYL(L)-NXNYCB(L))/FLOAT(MDFS-1)
      DNXNY2=(NXNY(L)-NXNYU(L))/FLOAT(MMAX-MDFS)
  120 DO 170 M=1,MMAX
      IF (MDFS.EQ.0.OR.LDFS.EQ.0) GO TO 150
      IF (L.NE.1) GO TO 130
      IF (ISUPER.EQ.2.AND.M.LT.MDFS) GO TO 170
      IF (ISUPER.EQ.3.AND.M.GT.MDFS) GO TO 170
      IF (ISUPER.EQ.2.AND.M.EQ.MDFS) GO TO 150
  130 IF (M.LT.MDFS) DNXNY=DNXNY1
      IF (M.GT.MDFS) DNXNY=DNXNY2
      IF (M.NE.MDFS) GO TO 150
      PL(L,1)=PTL/DEMP
      TEMP=TTL/DEM
      ROL(L,1)=PL(L,1)/(RG*TEMP)
      QQ=MN3*SQRT(GAMMA*PL(L,1)/ROL(L,1))
      IF (NXNYL(L).EQ.0.0) GO TO 140
      UL(L,1)=QQ/SQRT(1.0+NXNYL(L)*NXNYL(L))
      VL(L,1)=-UL(L,1)*NXNYL(L)
      GO TO 150
  140 UL(L,1)=QQ
      VL(L,1)=0.0
  150 IF (ISUPER.EQ.3.AND.(M.EQ.MDFS.AND.L.EQ.1)) GO TO 170
      P(L,M,1)=PT(M)/DEMP
      TEMP=TT(M)/DEM
      RO(L,M,1)=P(L,M,1)/(RG*TEMP)
      QQ=MN3*SQRT(GAMMA*P(L,M,1)/RO(L,M,1))
      DN=NXNYCB(L)+DNXNY*FLOAT(M-1)
      IF (LDFS.NE.0.AND.M.GE.MDFS) DN=NXNYU(L)+DNXNY*FLOAT(M-MDFS)
      DNS=DN*DN
      IF (DNS.EQ.0.0) GO TO 160
      SIGN=1.0
      IF (DN.GT.0.0) SIGN=-1.0
      U(L,M,1)=QQ/SQRT(1.0+DNS)
      V(L,M,1)=SIGN*QQ/SQRT(1.0+1.0/DNS)
      GO TO 170
  160 U(L,M,1)=QQ
      V(L,M,1)=0.0
  170 CONTINUE
  180 CONTINUE
      RETURN
C
C     FORMAT STATEMENTS
C
  190 FORMAT (1H0,10X,47H***** THE 1-D SOLUTION FOR THE INITIAL-DATA SUR
     1 ,47HFACE FAILED TO CONVERGE IN 100 ITERATIONS AT L=,I2,6H *****)
  200 FORMAT (1H0,10X,48H***** THE STAGNATION CONDITIONS FOR THE 1-D INI
     1T,41HIAL-DATA SURFACE WERE NOT SPECIFIED *****)
      END
```

```
      SUBROUTINE MAP
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE MAPPING FUNCTIONS
C
C     ****************************************************************
C
*CALL,MCC
C
C     SINGLE FLOW SPACE
C
      IF (IP.EQ.-1) GO TO 40
      IF (LMAP.GE.LDFSS.AND.LMAP.LE.LDFSF) GO TO 10
      YP=YCB(LMAP)+VN(MMAP)*(YW(LMAP)-YCB(LMAP))
      IF (IP.EQ.0) RETURN
      OM1=DZDX(LMAP)
      OM2=DZDX(LMAP+1)
      BE=1.0/(YW(LMAP)-YCB(LMAP))
      BE3=DYDVN(MMAP)*BE
      BE4=DYDVN(MMAP+1)*BE
      AL=NXNYCB(LMAP)+VN(MMAP)*(NXNY(LMAP)-NXNYCB(LMAP))
      AL3=BE3*AL
      AL4=BE4*AL
      DE=-VN(MMAP)*XWI(LMAP)
      DE3=BE3*DE
      DE4=BE4*DE
      RETURN
C
C     DUAL FLOW SPACE
C
   10 IF (MMAP.LT.MDFS) GO TO 20
      IF (MMAP.GT.MDFS) GO TO 30
      IF (IB.EQ.4) GO TO 30
C
   20 YP=YCB(LMAP)+VN(MMAP)*(YL(LMAP)-YCB(LMAP))/CC
      IF (IP.EQ.0) RETURN
      OM1=DZDX(LMAP)
      OM2=DZDX(LMAP+1)
      BE=CC/(YL(LMAP)-YCB(LMAP))
      BE3=DYDVN(MMAP)*BE
      BE4=DYDVN(MMAP+1)*BE
      AL=(VN(MMAP)*NXNYL(LMAP)-(VN(MMAP)-CC)*NXNYCB(LMAP))/CC
      AL3=BE3*AL
      AL4=BE4*AL
      DE3=0.0
      DE4=0.0
      IF (MMAP.NE.MDFS) RETURN
      AL4=AL3
      BE4=BE3
      RETURN
C
   30 YP=YU(LMAP)+(VN(MMAP)-CC)*(YW(LMAP)-YU(LMAP))/(1.0-CC)
      IF (IP.EQ.0) RETURN
      OM1=DZDX(LMAP)
      OM2=DZDX(LMAP+1)
      BE=(1.0-CC)/(YW(LMAP)-YU(LMAP))
      BE3=DYDVN(MMAP)*BE
      BE4=DYDVN(MMAP+1)*BE
      AL=((VN(MMAP)-CC)*NXNY(LMAP)-(VN(MMAP)-1.0)*NXNYU(LMAP))/(1.0-CC)
      AL3=BE3*AL
      AL4=BE4*AL
```

```
      DE=(VN(MMAP)-CC)*XWI(LMAP)/(1.0-CC)
      DE3=BE3*DE
      DE4=BE4*DE
      IF (MMAP.NE.MDFS) RETURN
      AL3=AL4
      BE3=BE4
      DE3=DE4
      RETURN
C
C     CALCULATE THE MAPPING FUNCTIONS FOR THE INITIAL SET-UP
C
   40 DO 50 L=1,LMAX
      X(L)=XP(1)+FLOAT(L-1)*DX
   50 CONTINUE
      DO 60 L=1,L1
      DZDX(L+1)=(X(L+1)-X(L))/(XP(L+1)-XP(L))
   60 CONTINUE
      DZDX(1)=DZDX(2)
      DZDX(LMAX+1)=DZDX(LMAX)
      IF (MDFS.EQ.0) GO TO 70
      LVN=LDFSS
      IF (LDFSS.EQ.1.AND.LDFSF.NE.LMAX) LVN=LDFSF
      CC=(YL(LVN)-YCB(LVN))/(YW(LVN)-YU(LVN)+YL(LVN)-YCB(LVN))
      IF (LDFSS.EQ.1.OR.LDFSF.EQ.LMAX) GO TO 70
      CCD=(YL(LDFSF)-YCB(LDFSF))/(YW(LDFSF)-YU(LDFSF)+YL(LDFSF)-YCB
     1 (LDFSF))
      IF (ABS(CCD-CC)/CC.LE.0.01) GO TO 70
      WRITE (6,140)
      IERR=1
      RETURN
   70 DO 80 M=1,MMAX
      Y(M)=FLOAT(M-1)*DY
   80 CONTINUE
      IF (IST.NE.0) GO TO 100
      DO 90 M=1,MMAX
      VN(M)=Y(M)
      DYDVN(M)=1.0
      YI(M)=Y(M)
   90 CONTINUE
      DYDVN(MMAX+1)=1.0
      RETURN
  100 DO 120 M=1,MMAX
      VN(M)=(YI(M)-YCB(1))/(YW(1)-YCB(1))
      IF (MDFS.EQ.0.OR.LDFSS.NE.1) GO TO 120
      IF (M.GE.MDFS) GO TO 110
      VN(M)=CC*(YI(M)-YCB(1))/(YL(1)-YCB(1))
      GO TO 120
  110 VN(M)=CC+(1.0-CC)*(YI(M)-YU(1))/(YW(1)-YU(1))
  120 CONTINUE
      DO 130 M=1,M1
      DYDVN(M+1)=(Y(M+1)-Y(M))/(VN(M+1)-VN(M))
  130 CONTINUE
      DYDVN(1)=DYDVN(2)
      DYDVN(MMAX+1)=DYDVN(MMAX)
      RETURN
C
  140 FORMAT (1H0,100H***** DUAL FLOW SPACE WALLS DO NOT BEGIN AND END A
     1T APPROXIMATELY THE SAME PROPORTIONAL HEIGHT *****)
      END
```

```
      SUBROUTINE MASFLO
C
C     ******************************************************************
C
C     THIS SUBROUTINE CALCULATES THE INITIAL-DATA OR SOLUTION SURFACE
C     MASS FLOW AND MOMENTUM THRUST
C
C     ******************************************************************
C
*CALL.MCC
      LC2=LC+LC
C
C     CALCULATE AND PRINT THE MASS FLOW AT EACH L LOCATION
C
      IP=0
      ND=N3
      IF (N.EQ.0) ND=1
      NP=N+NSTART
      IF (IUO.NE.2) WRITE (6,80) NP
      IF (IUO.EQ.2) WRITE (6,90) NP
      DO 70 L=1,LMAX
      LMAP=L
      XMASS=0.0
      THRUST=0.0
      IF (MDFS.NE.0) IB=3
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DO 50 M=2,MMAX
      MMAP=M
      CALL MAP
      MMAP=M-1
      YP1=YP
      CALL MAP
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 10
      ROU=(ROL(L,ND)*UL(L,ND)+RO(L,M-1,ND)*U(L,M-1,ND))*0.5
      ROU2=(ROL(L,ND)*UL(L,ND)**2+RO(L,M-1,ND)*U(L,M-1,ND)**2)*0.5
      IB=4
      GO TO 20
   10 ROU=(RO(L,M,ND)*U(L,M,ND)+RO(L,M-1,ND)*U(L,M-1,ND))*0.5
      ROU2=(RO(L,M,ND)*U(L,M,ND)**2+RO(L,M-1,ND)*U(L,M-1,ND)**2)*0.5
   20 IF (NDIM.EQ.1) GO TO 30
      AREA=(YP1-YP)/LC2
      GO TO 40
   30 AREA=3.141593*(YP1**2-YP**2)/LC2
   40 XMASS=XMASS+ROU*AREA*G
      THRUST=THRUST+ROU2*AREA
   50 CONTINUE
      IF (L.EQ.1) XMASSI=XMASS
      XMFR=0.0
      IF (XMASSI.NE.0.0) XMFR=XMASS/XMASSI
      IF (L.EQ.1) THRUSI=THRUST
      TR=0.0
      IF (THRUSI.NE.0.0) TR=THRUST/THRUSI
      IF (IUO.NE.2) GO TO 60
      XMASS=XMASS*0.4536
      THRUST=THRUST*4.4477
      IF (NDIM.NE.0) GO TO 60
      XMASS=XMASS/2.54
      THRUST=THRUST/2.54
   60 WRITE (6,100) L,XMASS,XMFR,THRUST,TR
   70 CONTINUE
      RETURN
C
C     FORMAT STATEMENTS
C
   80 FORMAT (1H1,20X,36HMASS FLOW AND THRUST CALCULATION, N=,I6,//30X,1
     1 HL,7X,9HMF(LBM/S),8X,6HMF/MFI,8X,6HT(LBF),11X,4HT/TI,/)
   90 FORMAT (1H1,20X,36HMASS FLOW AND THRUST CALCULATION, N=,I6,//30X,1
     1 HL,8X,8HMF(KG/S),8X,6HMF/MFI,10X,4HT(N),11X,4HT/TI,/)
  100 FORMAT (1H ,20X,I10,F16.5,F14.4,2F15.4)
      END
```

```
      SUBROUTINE PLOT (TITLE,T,NP,IVPTS)
C
C     ************************************************************
C
C     THIS SUBROUTINE PLOTS THE VELOCITY VECTORS AND DEPENDENT VARIABLE
C     CONTOUR PLOTS
C
C     ************************************************************
C
      DIMENSION CON(9), XCO(4), YCO(4), TITLE(10)
*CALL,MCC
C
C     SET UP THE PLOT SIZE
C
      IP=0
      ND=N3
      IF (N.EQ.0) ND=1
      XXL=XI
      XR=XE
      YT=YW(1)
      YB=YCB(1)
      DO 10 L=2,LMAX
      YT=AMAX1(YT,YW(L))
      YB=AMIN1(YB,YCB(L))
   10 CONTINUE
      VV=-0.1*DX
      DO 70 IDUM=1,IVPTS
      VV=VV+DX
      FIYB=900.0
      XD=(XR-XXL)/(YT-YB)
      FIR=(1022.0-1022.0/FLOAT(L1)-FLOAT(IDUM)*1022.0/FLOAT(L1))/884.0
      IF (XD.LE.FIR) GO TO 20
      FIXL=1022.0/FLOAT(L1)
      FIXR=1022.0-FIXL-FLOAT(IDUM)*1022.0/FLOAT(L1)
      FIYT=900.0-(FIXR-FIXL)/XD
      GO TO 30
   20 FIXL=511.0-450.0*XD
      FIXR=511.0+450.0*XD
      FIYT=16.0
   30 XCONV=(FIXR-FIXL)/(XR-XXL)
      YCONV=(FIYT-FIYB)/(YT-YB)
C
C     GENERATE THE VELOCITY VECTOR PLOT
C
      VMAX=0.0
      DO 40 L=1,LMAX
      DO 40 M=1,MMAX
      VMAX=AMAX1(VMAX,ABS(U(L,M,ND)),ABS(V(L,M,ND)))
   40 CONTINUE
      IF (VMAX.LT.1.0E-10) GO TO 80
      DROU=VV/VMAX
      CALL ADV (1)
      DO 60 L=1,LMAX
      LMAP=L
      IF (MDFS.NE.0) IB=3
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      IX1=FIXL+(XP(L)-XI)*XCONV
      DO 60 M=1,MMAX
      MMAP=M
      CALL MAP
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 50
      IY1=FIYB+(YP-YB)*YCONV
      IX2=IX1+UL(L,ND)*DROU*XCONV
      IY2=IY1+VL(L,ND)*DROU*YCONV
      CALL DRV (IX1,IY1,IX2,IY2)
      CALL PLT (IX1,IY1,16)
      IB=4
      CALL MAP
   50 IY1=FIYB+(YP-YB)*YCONV
      IX2=IX1+U(L,M,ND)*DROU*XCONV
      IY2=IY1+V(L,M,ND)*DROU*YCONV
```

```
      CALL DRV (IX1,IY1,IX2,IY2)
      CALL PLT (IX1,IY1,16)
   60 CONTINUE
      CALL LINCNT (58)
      WRITE (7,580) IDUM,NP,T
      WRITE (7,500) TITLE
   70 CONTINUE
C
C     RESET PLOT SIZE FOR CONTOUR PLOTS
C
   80 IF (XD.LE.FIR) GO TO 90
      FIXR=1022.0-FIXL-1022.0/FLOAT(L1)
      FIYT=900.0-(FIXR-FIXL)/XD
      XCONV=(FIXR-FIXL)/(XR-XXL)
      YCONV=(FIYT-FIYB)/(YT-YB)
C
C     GENERATE THE PHYSICAL SPACE GRID
C
   90 CALL ADV (1)
      DO 110 L=2,LMAX
      IF (MDFS.NE.0) IB=3
      IX1=FIXL+(XP(L-1)-XI)*XCONV
      IX2=FIXL+(XP(L)-XI)*XCONV
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DO 110 M=1,MMAX
      LMAP=L-1
      MMAP=M
      CALL MAP
      LMAP=L
      YP1=YP
      CALL MAP
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 100
      IY1=FIYB+(YP1-YB)*YCONV
      IY2=FIYB+(YP-YB)*YCONV
      CALL DRV (IX1,IY1,IX2,IY2)
      IB=4
      LMAP=L-1
      CALL MAP
      LMAP=L
      YP1=YP
      CALL MAP
  100 IY1=FIYB+(YP1-YB)*YCONV
      IY2=FIYB+(YP-YB)*YCONV
      CALL DRV (IX1,IY1,IX2,IY2)
  110 CONTINUE
C
      DO 130 L=1,LMAX
      IX1=FIXL+(XP(L)-XI)*XCONV
      IY1=FIYB+(YCB(L)-YB)*YCONV
      IF (MDFS.EQ.0) GO TO 120
      IF (L.LT.LDFSS.OR.L.GT.LDFSF) GO TO 120
      IY2=FIYB+(YL(L)-YB)*YCONV
      CALL DRV (IX1,IY1,IX1,IY2)
      IY1=FIYB+(YU(L)-YB)*YCONV
  120 IY2=FIYB+(YW(L)-YB)*YCONV
      CALL DRV (IX1,IY1,IX1,IY2)
  130 CONTINUE
      CALL LINCNT (58)
      WRITE (7,590) NP,T
      WRITE (7,500) TITLE
C
C     FILL THE PLOTTING ARRAY CO FOR THE CONTOUR PLOTS
C
      MDUM=MMAX
      IF (MDFS.NE.0) MDUM=MMAX+1
      IUC=1.0
      IF (IUO.EQ.2) IUC=0.0
      IDUM=4
      IF (ITM.EQ.2) IDUM=5
      IF (ITM.EQ.3) IDUM=6
      DO 490 I=1,IDUM
```

```
C
      DO 270 L=1,LMAX
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DO 270 M=1,MDUM
      IF (LDFS.EQ.0.AND.M.EQ.MMAX+1) GO TO 270
      MD1=M
      IF (LDFS.NE.0.AND.M.GT.MDFS) MD1=M-1
      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 200
      GO TO (140,150,160,170,180,190), I
  140 CQ(L,M)=ROL(L,ND)*G*(16.02-IUC*15.02)
      GO TO 270
  150 CQ(L,M)=PL(L,ND)/PC*(6.8948-IUC*5.8948)
      GO TO 270
  160 CQ(L,M)=PL(L,ND)/(ROL(L,ND)*RG)*(0.555556+IUC*0.444444)
      GO TO 270
  170 CQ(L,M)=SQRT((UL(L,ND)**2+VL(L,ND)**2)/(GAMMA*PL(L,ND)/ROL(L,ND)))
      GO TO 270
  180 CQ(L,M)=QL(L,ND)*(0.0929+IUC*0.9071)
      GO TO 270
  190 CQ(L,M)=EL(L,ND)*(0.0929+IUC*0.9071)
      GO TO 270
  200 GO TO (210,220,230,240,250,260), I
  210 CQ(L,M)=RO(L,MD1,ND)*G*(16.02-IUC*15.02)
      GO TO 270
  220 CQ(L,M)=P(L,MD1,ND)/PC*(6.8948-IUC*5.8948)
      GO TO 270
  230 CQ(L,M)=P(L,MD1,ND)/(RO(L,MD1,ND)*RG)*(0.555556+IUC*0.444444)
      GO TO 270
  240 CQ(L,M)=SQRT((U(L,MD1,ND)**2+V(L,MD1,ND)**2)/(GAMMA*P(L,MD1,ND)/RO
     1 (L,MD1,ND)))
      GO TO 270
  250 CQ(L,M)=Q(L,MD1,ND)*(0.0929+IUC*0.9071)
      GO TO 270
  260 CQ(L,M)=E(L,MD1,ND)*(0.0929+IUC*0.9071)
  270 CONTINUE
C
C     DETERMINE THE PLOTTING LINE QUANTITIES AND LABEL THE FRAMES
C
      QMN=1.0E06
      QMX=-QMN
      DO 280 L=1,LMAX
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DO 280 M=1,MDUM
      IF (LDFS.EQ.0.AND.M.EQ.MMAX+1) GO TO 280
      QMN=AMIN1(CQ(L,M),QMN)
      QMX=AMAX1(CQ(L,M),QMX)
  280 CONTINUE
      XX=QMX-QMN
      DQ=0.1*XX
      DO 290 K=1,9
      CON(K)=QMN+(FLOAT(K))*DQ
  290 CONTINUE
      K=9
      CALL ADV (1)
      CALL LINCNT (58)
      GO TO (300,310,320,330,340,350), I
  300 WRITE (7,510) NP,T
      GO TO 360
  310 WRITE (7,520) NP,T
      GO TO 360
  320 WRITE (7,530) NP,T
      GO TO 360
  330 WRITE (7,540) NP,T
      GO TO 360
  340 WRITE (7,550) NP,T
      GO TO 360
  350 WRITE (7,560) NP,T
  360 WRITE (7,570) QMN,QMX,CON(1),CON(K),DQ
      WRITE (7,500) TITLE
```

```
C
C     DETERMINE THE LOCATION OF EACH CONTOUR LINE SEGIMENT AND PLOT IT
C
      DO 470 L=2,LMAX
      IF (MDFS.NE.O) IB=3
      XCO(1)=XP(L-1)
      XCO(2)=XP(L)
      XCO(3)=XCO(1)
      XCO(4)=XCO(2)
      LDFS=O
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      DO 470 M=2,MMAX
      MD2=M
      MD3=M
      IF (MDFS.EQ.O.OR.M.LE.MDFS) GO TO 370
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) MD2=M+1
      IF (L.GE.LDFSS+1.AND.L.LE.LDFSF+1) MD3=M+1
  370 LMAP=L-1
      MMAP=M-1
      CALL MAP
      LMAP=L
      YCO(1)=YP
      CALL MAP
      LMAP=L-1
      MMAP=M
      YCO(2)=YP
      CALL MAP
      LMAP=L
      YCO(3)=YP
      CALL MAP
      YCO(4)=YP
      IF (M.NE.MDFS.OR.LDFS.EQ.O) GO TO 380
      IB=4
  380 DO 460 KK=1,K
      K1=O
      K2=O
      K3=O
      K4=O
      IF (CQ(L-1,MD3-1).LE.CON(KK)) K1=1
      IF (CQ(L,MD2-1).LE.CON(KK)) K2=1
      IF (CQ(L-1,MD3).LE.CON(KK)) K3=1
      IF (CQ(L,MD2).LE.CON(KK)) K4=1
      IF (K1*K2*K3*K4.NE.O) GO TO 460
      IF (K1+K2+K3+K4.EQ.O) GO TO 460
      LL=O
      IF (K1+K3.NE.1) GO TO 390
      IC1=1
      IC2=3
      LP1=L-1
      MP1=MD3-1
      LP2=L-1
      MP2=MD3
      ASSIGN 390 TO KR1
      GO TO 420
  390 IF (K1+K2.NE.1) GO TO 400
      IC1=1
      IC2=2
      LP1=L-1
      MP1=MD3-1
      LP2=L
      MP2=MD2-1
      ASSIGN 400 TO KR1
      GO TO 420
  400 IF (K2+K4.NE.1) GO TO 410
      IC1=2
      IC2=4
      LP1=L
      MP1=MD2-1
      LP2=L
      MP2=MD2
      ASSIGN 410 TO KR1
      GO TO 420
```

```
  410 IF (K3+K4.NE.1) GO TO 460
      IC1=3
      IC2=4
      LP1=L-1
      MP1=MD3
      LP2=L
      MP2=MD2
      ASSIGN 460 TO KR1
  420 LL=LL+1
      XX=(CON(KK)-CQ(LP1,MP1))/(CQ(LP2,MP2)-CQ(LP1,MP1))
      IF (LL.EQ.2) GO TO 430
      IX1=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XXL)*XCONV
      IY1=FIYB+(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YB)*YCONV
      GO TO KR1, (390,400,410,460)
  430 IX2=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XXL)*XCONV
      IY2=FIYB+(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YB)*YCONV
      CALL DRV (IX1,IY1,IX2,IY2)
      IF (KK.NE.1) GO TO 440
      CALL PLT (IX1,IY1,35)
  440 IF (KK.NE.K) GO TO 450
      CALL PLT (IX1,IY1,24)
  450 LL=0
      IF (LP2.NE.L) GO TO 460
      IF (MP2.NE.MD2-1) GO TO 460
      GO TO 400
  460 CONTINUE
  470 CONTINUE
C
C     DRAW THE GEOMETRY BOUNDARIES FOR THE CONTOUR PLOTS
C
      DO 480 L=2,LMAX
      IX1=FIXL+(XP(L-1)-XI)*XCONV
      IX2=FIXL+(XP(L)-XI)*XCONV
      IY1=FIYB+(YCB(L-1)-YB)*YCONV
      IY2=FIYB+(YCB(L)-YB)*YCONV
      IY3=FIYB+(YW(L-1)-YB)*YCONV
      IY4=FIYB+(YW(L)-YB)*YCONV
      IY5=FIYB+(YL(L-1)-YB)*YCONV
      IY6=FIYB+(YL(L)-YB)*YCONV
      IY7=FIYB+(YU(L-1)-YB)*YCONV
      IY8=FIYB+(YU(L)-YB)*YCONV
      CALL DRV (IX1,IY1,IX2,IY2)
      CALL DRV (IX1,IY3,IX2,IY4)
      IF (MDFS.EQ.0) GO TO 480
      IF (L.LE.LDFSS.OR.L.GT.LDFSF) GO TO 480
      CALL DRV (IX1,IY5,IX2,IY6)
      CALL DRV (IX1,IY7,IX2,IY8)
  480 CONTINUE
  490 CONTINUE
      CALL ADV (1)
      RETURN
C
C     FORMAT STATEMENTS
C
  500 FORMAT (1H ,10A8)
  510 FORMAT (1H ,7HDENSITY,24X,2HN=,I6,2X,2HT=,1PE10.4,4H SEC)
  520 FORMAT (1H ,8HPRESSURE,23X,2HN=,I6,2X,2HT=,1PE10.4,4H SEC)
  530 FORMAT (1H ,11HTEMPERATURE,20X,2HN=,I6,2X,2HT=,1PE10.4,4H SEC)
  540 FORMAT (1H ,11HMACH NUMBER,20X,2HN=,I6,2X,2HT=,1PE10.4,4H SEC)
  550 FORMAT (1H ,17HTURBULENCE ENERGY,20X,2HN=,I6,2X,2HT=,1PE10.4,4H SE
     1C)
  560 FORMAT (1H ,16HDISSIPATION RATE,20X,2HN=,I6,2X,2HT=,1PE10.4,4H SEC
     1 )
  570 FORMAT (1H ,10HLOW VALUE=,1PE11.4,2X,11HHIGH VALUE=,E11.4,2X,12HLO
     1W CONTOUR=,E11.4,/,1X,13HHIGH CONTOUR=,E11.4,2X,14HDELTA CONTOUR=
     2 ,E11.4)
  580 FORMAT (1H ,18HVELOCITY VECTORS (,I1,2HX),10X,2HN=,I6,2X,2HT=,1PE1
     1 0.4,4H SEC)
  590 FORMAT (1H ,19HPHYSICAL SPACE GRID,10X,2HN=,I6,2X,2HT=,1PE10.4,4H
     1SEC)
      END
```

```
      SUBROUTINE SWITCH (ISWITCH)
C
C     ****************************************************************
C
C     THIS SUBROUTINE SWITCHES THE DUAL FLOW SPACE BOUNDARY SOLUTIONS
C     BETWEEN THE DUMMY ARRAYS AND THE SOLUTION ARRAYS
C
C     ****************************************************************
C
C     ISWITCH=2 SWITCHES THE FLOW VARIABLES, BOUNDARY CONDITIONS, AND
C     VISCOUS TERMS AT N AND N+1.  ISWITCH=3 SWITCHES THE FLOW VARIABLES
C     AT N.  ISWITCH=5 SWITCHES THE FLOW VARIABLES AT N AND STORES THE
C     VISCOUS TERMS.
C
*CALL,MCC
C
C     SWITCH THE FLOW VARIABLES AT N
C
      DO 10 L=LDFSS,LDFSF
      UDFS=UL(L,N1)
      VDFS=VL(L,N1)
      PDFS=PL(L,N1)
      RODFS=ROL(L,N1)
      UL(L,N1)=U(L,MDFS,N1)
      VL(L,N1)=V(L,MDFS,N1)
      PL(L,N1)=P(L,MDFS,N1)
      ROL(L,N1)=RO(L,MDFS,N1)
      U(L,MDFS,N1)=UDFS
      V(L,MDFS,N1)=VDFS
      P(L,MDFS,N1)=PDFS
      RO(L,MDFS,N1)=RODFS
      IF (ITM.LE.1) GO TO 10
      QDFS=QL(L,N1)
      EDFS=EL(L,N1)
      QL(L,N1)=Q(L,MDFS,N1)
      EL(L,N1)=E(L,MDFS,N1)
      Q(L,MDFS,N1)=QDFS
      E(L,MDFS,N1)=EDFS
   10 CONTINUE
      IF (ISWITCH.EQ.3) RETURN
      IF (ISWITCH.EQ.5) GO TO 70
C
C     SWITCH THE FLOW VARIABLES AT N+1
C
      DO 20 L=LDFSS,LDFSF
      UDFS=UL(L,N3)
      VDFS=VL(L,N3)
      PDFS=PL(L,N3)
      RODFS=ROL(L,N3)
      UL(L,N3)=U(L,MDFS,N3)
      VL(L,N3)=V(L,MDFS,N3)
      PL(L,N3)=P(L,MDFS,N3)
      ROL(L,N3)=RO(L,MDFS,N3)
      U(L,MDFS,N3)=UDFS
      V(L,MDFS,N3)=VDFS
      P(L,MDFS,N3)=PDFS
      RO(L,MDFS,N3)=RODFS
      IF (ITM.LE.1) GO TO 20
      QDFS=QL(L,N3)
      EDFS=EL(L,N3)
      QL(L,N3)=Q(L,MDFS,N3)
      EL(L,N3)=E(L,MDFS,N3)
      Q(L,MDFS,N3)=QDFS
      E(L,MDFS,N3)=EDFS
   20 CONTINUE
C
C     SWITCH THE BOUNDARY CONDITIONS
C
      IF (LDFSS.NE.1) GO TO 40
      IF (ISUPER.GE.2) GO TO 40
      IF (ISUPER.EQ.1) GO TO 30
      PTDFS=PTL
```

```
      TTDFS=TTL
      THETDFS=THETAL
      PTL=PT(MDFS)
      TTL=TT(MDFS)
      THETAL=THETA(MDFS)
      PT(MDFS)=PTDFS
      TT(MDFS)=TTDFS
      THETA(MDFS)=THETDFS
      GO TO 40
   30 PIDFS=PIL
      PIL=PI(MDFS)
      PI(MDFS)=PIDFS
   40 IF (LDFSF.NE.LMAX) GO TO 50
      PEDFS=PEL
      PEL=PE(MDFS)
      PE(MDFS)=PEDFS
C
C     SWITCH THE VISCOUS TERMS
C
   50 IF (CAV.EQ.0.0.AND.CHECK.EQ.0.0) RETURN
      DO 60 L=LDFSS,LDFSF
      QUDFS=QUTL(L)
      QVDFS=QVTL(L)
      QPDFS=QPTL(L)
      QRODFS=QROTL(L)
      QUTL(L)=QUT(L,MDFS)
      QVTL(L)=QVT(L,MDFS)
      QPTL(L)=QPT(L,MDFS)
      QROTL(L)=QROT(L,MDFS)
      QUT(L,MDFS)=QUDFS
      QVT(L,MDFS)=QVDFS
      QPT(L,MDFS)=QPDFS
      QROT(L,MDFS)=QRODFS
      IF (ITM.LE.1) GO TO 60
      QQDFS=QQTL(L)
      QEDFS=QETL(L)
      QQTL(L)=QQT(L,MDFS)
      QETL(L)=QET(L,MDFS)
      QQT(L,MDFS)=QQDFS
      QET(L,MDFS)=QEDFS
   60 CONTINUE
      RETURN
C
C     STORE THE VISCOUS TERMS
C
   70 DO 80 L=LDFSS,LDFSF
      QUTL(L)=QUT(L,MDFS)
      QVTL(L)=QVT(L,MDFS)
      QPTL(L)=QPT(L,MDFS)
      QROTL(L)=QROT(L,MDFS)
      IF (ITM.LE.1) GO TO 80
      QQTL(L)=QQT(L,MDFS)
      QETL(L)=QET(L,MDFS)
   80 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VISCOUS
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE LOCAL ARTIFICAL VISCOSITY,
C     MOLECULAR VISCOSITY, AND TURBULENCE TERMS
C
C     ****************************************************************
C
*CALL,MCC
      REAL MU, LA, LP2M, LPM, K, MUT, LAT, LP2MT, LPMT, KT, MUT1, MUT2,
     1 MUT3, MUT4, LAT1, LAT2, LAT3, LAT4, KT1, KT2, KT3, KT4, LP2MT1,
     2 LP2MT2, LP2MT3, LP2MT4, MU1, MU2, MU3, MU4, LA1, LA2, LA3, LA4,
     3 K1, K2, K3, K4, LP2M1, LP2M2, LP2M3, LP2M4, MUTD, MUTT
C
      IP=1
      IF (N.NE.1) GO TO 10
      IF (NVC.NE.1) GO TO 10
      SIGQR=1.0/SIGQ
      SIGER=1.0/SIGE
      F2I=FLOAT(2-IVBC)
      GAM=GAMMA-1.0
      DRK=GAM1*RG/PRA
      TRK=GAM1*RG/PRT
      GRG=GAMMA*RG
      XITM=0.0
      IF (ITM.EQ.2) XITM=0.67
      MU=0.0
      LA=0.0
      K=0.0
      MU1=0.0
      MU2=0.0
      MU3=0.0
      MU4=0.0
      LA1=0.0
      LA2=0.0
      LA3=0.0
      LA4=0.0
      K1=0.0
      K2=0.0
      K3=0.0
      K4=0.0
      LP2M=0.0
      LP2M1=0.0
      LP2M2=0.0
      LP2M3=0.0
      LP2M4=0.0
      LPM=0.0
      MUTD=0.0
      DLP2MT=0.0
      DMUT=0.0
      DLAT=0.0
      MUT=0.0
      LAT=0.0
      KT=0.0
      MUT1=0.0
      MUT2=0.0
      MUT3=0.0
      MUT4=0.0
      LAT1=0.0
      LAT2=0.0
      LAT3=0.0
      LAT4=0.0
      KT1=0.0
      KT2=0.0
      KT3=0.0
      KT4=0.0
      LP2MT=0.0
      SMU1=0.0
      SMU2=0.0
      SMU3=0.0
      SMU4=0.0
```

```
      LP2MT1=0.0
      LP2MT2=0.0
      LP2MT3=0.0
      LP2MT4=0.0
      LPMT=0.0
      TML=0.0
      RMU=0.0
      RMU1=0.0
      RMU2=0.0
      RMU3=0.0
      RMU4=0.0
      RLA=0.0
      RLA1=0.0
      RLA2=0.0
      RLA3=0.0
      RLA4=0.0
      RK=0.0
      RK1=0.0
      RK2=0.0
      RK3=0.0
      RK4=0.0
      RLP2M=0.0
      RLP2M1=0.0
      RLP2M2=0.0
      RLP2M3=0.0
      RLP2M4=0.0
      RLPM=0.0
      RRO=0.0
      RRO1=0.0
      RRO2=0.0
      RRO3=0.0
      RRO4=0.0
      RODIFF=0.0
      EROT=0.0
      TLMUR=0.0
      AVMUR=0.0
      DEL=0.0
      QSMO=0.0
      ESMO=0.0
      ROXY1=0.0
      ROXY2=0.0
      ROXY3=0.0
      ROXY4=0.0
      ROXY12=0.0
      BROY1=0.0
      BROY2=0.0
      BROY3=0.0
      BROY4=0.0
      BROY34=0.0
      UROT=0.0
      VROT=0.0
      PROT=0.0
      QDISS=0.0
      QPROD=0.0
      QDIFF=0.0
      QROT1=0.0
      EPROD=0.0
      EDIFF=0.0
      EDISS=0.0
      ELOWR=0.0
      ROQX=0.0
      ROQY=0.0
      ATERM=0.0
      ATERM1=0.0
      ATERM2=0.0
      ATERM3=0.0
      ATERM4=0.0
      UVTA=0.0
      VVTA=0.0
      PVTA=0.0
      PCTA=0.0
      RODIFFA=0.0
```

```
      UROTA=0.0
      VROTA=0.0
      PROTA=0.0
      QPRODA=0.0
      QDIFFA=0.0
      EPRODA=0.0
      EDIFFA=0.0
      QROTTA=0.0
      EROTA=0.0
      ELOWRA=0.0
      SMT=1.0
      RDUM=0.0
      ECHECK=ABS(EMU)+ABS(ELA)+ABS(EK)
      IF (ECHECK.EQ.0.0) GO TO 10
      IF (ABS(EMU).EQ.ABS(ELA).AND.ABS(EMU).EQ.ABS(EK)) ECHECK=-1.0
C
   10 NLINE=0
      IF (IAV.EQ.0) GO TO 30
      IF (NC.NE.NPRINT.AND.(N.NE.NMAX.AND.ISTOP.EQ.0)) GO TO 30
      IF (IAV.EQ.2) GO TO 20
      IF (NVC.GT.2.AND.NVC.NE.NVCM+1) GO TO 30
   20 WRITE (6,1460)
      NP=N+NSTART
      WRITE (6,1450) NP,NVC
C
C     DO LOOP SET-UP
C
   30 MIS=1
      MIF=MMAX
      IF (IVC.EQ.0) GO TO 40
      IF (NVC.EQ.1) GO TO 40
      MIS=MVCB
      MIF=MVCT
   40 IDFS=0
C
      IF (MDFS.EQ.0) GO TO 70
      IF (NVC.EQ.1.AND.MDFSC.NE.0) GO TO 70
      CALL SWITCH (3)
      MIS=1
      IF (NVC.NE.1) MIS=MVCB
      MIF=MDFS
      IB=3
      GO TO 70
   50 CALL SWITCH (5)
      MIS=MDFS+1
      MIF=MMAX
      IF (NVC.NE.1) MIF=MVCT
      ID=4
      GO TO 70
   60 IDFS=1
      MIS=MDFS
      MIF=MDFS
C
C     BEGIN THE L OR X DO LOOP
C
   70 DO 1410 L=1,LMAX
      LMAP=L
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      IF (L.NE.1.AND.L.NE.LMAX) DXP=0.5*(XP(L+1)-XP(L-1))
      IF (L.EQ.1) DXP=XP(2)-XP(1)
      IF (L.EQ.LMAX) DXP=XP(LMAX)-XP(L1)
      IF (IDFS.EQ.1.AND.LDFS.EQ.0) GO TO 1410
C
C     CALCULATE THE WALL SHEAR STRESS FOR THE MIXING LENGTH TURBULENCE
C     MODELS
C
      IF (ITM.NE.1.AND.ITM.NE.2) GO TO 160
      IF (MDFS.EQ.0) GO TO 80
      IMLM=1
      IF (LDFS.NE.0) IMLM=2
   80 IF (IMLM.EQ.1) GO TO 160
```

```
      IF (MDFS.NE.O) GO TO 100
      IF (NGCB.NE.O) GO TO 90
      MT=M1
      MMAP=MMAX
      GO TO 110
   90 MT=2
      MMAP=1
      GO TO 110
  100 MMAP=MDFS
      IBD=IB
      IB=4
      MT=MDFS+1
  110 CALL MAP
      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 120
      UAVG=0.25*(U(L-1,MT,N1)+U(L+1,MT,N1)+2.0*U(L,MT,N1))
      VAVG=0.25*(V(L-1,MT,N1)+V(L+1,MT,N1)+2.0*V(L,MT,N1))
      GO TO 130
  120 UAVG=U(L,MT,N1)
      VAVG=V(L,MT,N1)
  130 TAUW=ABS(BE3*UAVG+AL3*VAVG)*DYR
      IF (MDFS.EQ.O) GO TO 160
      TAUWP=TAUW
      IB=3
      CALL MAP
      MT=MDFS-1
      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 140
      UAVG=0.25*(U(L-1,MT,N1)+U(L+1,MT,N1)+2.0*U(L,MT,N1))
      VAVG=0.25*(V(L-1,MT,N1)+V(L+1,MT,N1)+2.0*V(L,MT,N1))
      GO TO 150
  140 UAVG=U(L,MT,N1)
      VAVG=V(L,MT,N1)
  150 TAUWM=ABS(BE3*UAVG+AL3*VAVG)*DYR
      IB=IBD
C
C     BEGIN THE M OR Y DO LOOP
C
  160 DO 1400 M=MIS,MIF
      IF (IVC.EQ.O) GO TO 190
      IF (NVC.NE.1) GO TO 190
      IF (MVCB.NE.1) GO TO 170
      IF (M.EQ.1) GO TO 1400
      GO TO 180
  170 IF (MVCT.NE.MMAX) GO TO 180
      IF (M.EQ.MMAX) GO TO 1400
  180 IF (M.LT.MVCB.OR.M.GT.MVCT) GO TO 190
      GO TO 1400
  190 IES=0
      IF (M.EQ.MMAX) IES=1
      IF (M.EQ.1.AND.NGCB.NE.O) IES=1
      IF (M.EQ.MDFS.AND.LDFS.NE.O) IES=1
C
C     CALCULATE THE TURBULENT MIXING LENGTH
C
      IF (ITM.EQ.O.OR.ITM.EQ.3) GO TO 210
      IF (NVC.NE.1) GO TO 200
      IF (M.EQ.MIS) CALL MIXLEN (L,M)
      IF (M.EQ.MVCT+1.AND.MVCB.EQ.1) CALL MIXLEN (L,M)
      IF (M.EQ.MVCT+1.AND.(MDFSC.NE.O.AND.LDFS.NE.O)) CALL MIXLEN (L,M)
      GO TO 210
  200 IF (M.EQ.MVCB) CALL MIXLEN (L,M)
      IF (M.EQ.MDFS.AND.(LDFS.NE.O.AND.IDFS.NE.O)) CALL MIXLEN (L,M)
      IF (M.EQ.MDFS+1.AND.MDFS.NE.O) CALL MIXLEN (L,M)
C
C     SET SPECIAL CONDITIONS FOR L=1 OR LMAX
C
  210 IF (L.NE.LMAX.AND.L.NE.1) GO TO 230
      TML=0.0
      MUT=0.0
      TLMUR=0.0
      AVMUR=0.0
      IF (M.EQ.1.OR.M.EQ.MMAX) GO TO 1340
      IF (L.EQ.1) GO TO 220
```

```
      IF (LDFSF.EQ.LMAX.AND.M.EQ.MDFS) GO TO 1340
      GO TO 230
  220 IF (LDFSS.EQ.1.AND.M.EQ.MDFS) GO TO 1340
C
  230 RORR=1.0/RO(L,M,N1)
      MMAP=M
      CALL MAP
      OM=2.0*OM1*OM2/(OM1+OM2)
      BE=2.0*BE3*BE4/(BE3+BE4)
      AL34=AL3+AL4
      DE34=DE3+DE4
      IF (AL34.EQ.0.0) AL34=1.0
      IF (DE34.EQ.0.0) DE34=1.0
      AL=2.0*AL3*AL4/AL34
      DE=2.0*DE3*DE4/DE34
      IF (YP.NE.0.0) RYP=1.0/YP
      YP3=YP-0.5*DY/BE3
      YP4=YP+0.5*DY/BE4
C
C     CHECK FOR ARTIFICAL VISCOSITY
C
      IF (CAV.EQ.0.0) GO TO 250
      IF (L.LT.LSS.AND.CHECK.EQ.0.0) GO TO 1340
      IF (L.GT.LSF.AND.CHECK.EQ.0.0) GO TO 1340
      IF (M.LT.MSS.AND.CHECK.EQ.0.0) GO TO 1340
      IF (M.GT.MSF.AND.CHECK.EQ.0.0) GO TO 1340
      XV=U(L,M,N1)*U(L,M,N1)+V(L,M,N1)*V(L,M,N1)
      XA=GAMMA*P(L,M,N1)*RORR
      XM=XV/XA
      SMT=1.0
      IF (NOSLIP.NE.0.AND.XM.LT.1.0) SMT=XM
      IF (SMACH.EQ.0.0) GO TO 250
      IF (XM.LT.SMACH*SMACH.AND.CHECK.EQ.0.0) GO TO 240
      GO TO 250
  240 QUT(L,M)=0.0
      QVT(L,M)=0.0
      QPT(L,M)=0.0
      QROT(L,M)=0.0
      GO TO 1340
C
C     CALCULATE THE X DERIVATIVES
C
  250 T=P(L,M,N1)/(RO(L,M,N1)*RG)
      A=SQRT(GRG*T)
      IF (L.EQ.1) GO TO 260
      ULM=U(L-1,M,N1)
      VLM=V(L-1,M,N1)
      PLM=P(L-1,M,N1)
      ROLM=RO(L-1,M,N1)
      QLM=Q(L-1,M,N1)
      ELM=E(L-1,M,N1)
      IF (L.EQ.LMAX) GO TO 280
  260 ULP=U(L+1,M,N1)
      VLP=V(L+1,M,N1)
      PLP=P(L+1,M,N1)
      ROLP=RO(L+1,M,N1)
      QLP=Q(L+1,M,N1)
      ELP=E(L+1,M,N1)
      IF (L.EQ.1) GO TO 290
      IF (M.NE.MDFS) GO TO 280
      IF (L.NE.LDFSS-1) GO TO 270
      ULP=0.5*(ULP+UL(L+1,N1))
      VLP=0.5*(VLP+VL(L+1,N1))
      PLP=0.5*(PLP+PL(L+1,N1))
      ROLP=0.5*(ROLP+ROL(L+1,N1))
      IF (ITM.LE.1) GO TO 280
      QLP=0.5*(QLP+QL(L+1,N1))
      ELP=0.5*(ELP+EL(L+1,N1))
      GO TO 280
  270 IF (L.NE.LDFSF+1) GO TO 280
      ULM=0.5*(ULM+UL(L-1,N1))
      VLM=0.5*(VLM+VL(L-1,N1))
```

```
      PLM=0.5*(PLM+PL(L-1,N1))
      ROLM=0.5*(ROLM+ROL(L-1,N1))
      IF (ITM.LE.1) GO TO 280
      QLM=0.5*(QLM+QL(L-1,N1))
      ELM=0.5*(ELM+EL(L-1,N1))
  280 UX1=(U(L,M,N1)-ULM)*DXR
      VX1=(V(L,M,N1)-VLM)*DXR
      TLM=PLM/(ROLM*RG)
      TX1=(T-TLM)*DXR
      ROX1=(RO(L,M,N1)-ROLM)*DXR
      IF (ITM.GE.2) ROQX=(RO(L,M,N1)*Q(L,M,N1)-ROLM*QLM)*DXR
      IF (L.EQ.LMAX) GO TO 340
  290 UX2=(ULP-U(L,M,N1))*DXR
      VX2=(VLP-V(L,M,N1))*DXR
      TLP=PLP/(ROLP*RG)
      TX2=(TLP-T)*DXR
      ROX2=(ROLP-RO(L,M,N1))*DXR
      IF (ITM.GE.2) ROQX=(ROLP*QLP-RO(L,M,N1)*Q(L,M,N1))*DXR
      IF (L.EQ.1) GO TO 340
      IF (CAV.EQ.0.0) GO TO 300
      IF (ISS.EQ.0) GO TO 300
      ALP=SQRT(GRG*TLP)
      ALM=SQRT(GRG*TLM)
      AX1=(A-ALM)*DXR
      AX2=(ALP-A)*DXR
  300 IF (ITM.LE.1) GO TO 320
      ROQX=(ROLP*QLP-ROLM*QLM)*DXR*0.5
      QX1=(Q(L,M,N1)-QLM)*DXR
      QX2=(QLP-Q(L,M,N1))*DXR
      Q2X=0.5*(SQRT(QLP)-SQRT(QLM))*DXR
      IF (ITM.EQ.3) GO TO 310
      ROSQ=RO(L,M,N1)*SQRT(Q(L,M,N1))
      ROSQ1=RO(L-1,M,N1)*SQRT(Q(L-1,M,N1))
      ROSQ2=RO(L+1,M,N1)*SQRT(Q(L+1,M,N1))
      GO TO 320
  310 EX1=(E(L,M,N1)-ELM)*DXR
      EX2=(ELP-E(L,M,N1))*DXR
      MUT=CQMU*RO(L,M,N1)*Q(L,M,N1)*Q(L,M,N1)*LC/E(L,M,N1)
      MUT1=CQMU*ROLM*QLM*QLM*LC/ELM
      MUT2=CQMU*ROLP*QLP*QLP*LC/ELP
  320 IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 330
      IF (IB.EQ.3) GO TO 680
      GO TO 490
  330 IF (M.EQ.1) GO TO 490
      IF (M.EQ.MMAX) GO TO 680
C
C     BEGIN THE INTERIOR POINT Y DERIVATIVE CALCULATION
C
  340 DYP=DY/BE
      UMP=U(L,M+1,N1)
      UMM=U(L,M-1,N1)
      VMP=V(L,M+1,N1)
      VMM=V(L,M-1,N1)
      PMP=P(L,M+1,N1)
      PMM=P(L,M-1,N1)
      ROMP=RO(L,M+1,N1)
      ROMM=RO(L,M-1,N1)
      QMP=Q(L,M+1,N1)
      QMM=Q(L,M-1,N1)
      EMP=E(L,M+1,N1)
      EMM=E(L,M-1,N1)
      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 350
      ULPMP=U(L+1,M+1,N1)
      ULMMP=U(L-1,M+1,N1)
      ULPMM=U(L+1,M-1,N1)
      ULMMM=U(L-1,M-1,N1)
      VLPMP=V(L+1,M+1,N1)
      VLMMP=V(L-1,M+1,N1)
      VLPMM=V(L+1,M-1,N1)
      VLMMM=V(L-1,M-1,N1)
      PLPMP=P(L+1,M+1,N1)
      PLMMP=P(L-1,M+1,N1)
      PLPMM=P(L+1,M-1,N1)
      PLMMM=P(L-1,M-1,N1)
```

```
      ROLPMP=RO(L+1,M+1,N1)
      ROLMMP=RO(L-1,M+1,N1)
      ROLPMM=RO(L+1,M-1,N1)
      ROLMMM=RO(L-1,M-1,N1)
      QLPMP=Q(L+1,M+1,N1)
      QLMMP=Q(L-1,M+1,N1)
      QLPMM=Q(L+1,M-1,N1)
      QLMMM=Q(L-1,M-1,N1)
      ELPMP=E(L+1,M+1,N1)
      ELMMP=E(L-1,M+1,N1)
      ELPMM=E(L+1,M-1,N1)
      ELMMM=E(L-1,M-1,N1)
  350 IF (IVC.EQ.0) GO TO 380
      IF (NVC.EQ.1) GO TO 380
      IF (M.EQ.MVCB) GO TO 360
      IF (M.EQ.MVCT) GO TO 370
      GO TO 380
C
C     LINEAR INTERPOLATION IN TIME FOR M=MVCB
C
  360 UMM=U(L,M-1,NN1)+RIND*(U(L,M-1,NN3)-U(L,M-1,NN1))
      VMM=V(L,M-1,NN1)+RIND*(V(L,M-1,NN3)-V(L,M-1,NN1))
      PMM=P(L,M-1,NN1)+RIND*(P(L,M-1,NN3)-P(L,M-1,NN1))
      ROMM=RO(L,M-1,NN1)+RIND*(RO(L,M-1,NN3)-RO(L,M-1,NN1))
      QMM=Q(L,M-1,NN1)+RIND*(Q(L,M-1,NN3)-Q(L,M-1,NN1))
      EMM=E(L,M-1,NN1)+RIND*(E(L,M-1,NN3)-E(L,M-1,NN1))
      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 390
      ULPMM=U(L+1,M-1,NN1)+RIND*(U(L+1,M-1,NN3)-U(L+1,M-1,NN1))
      ULMMM=U(L-1,M-1,NN1)+RIND*(U(L-1,M-1,NN3)-U(L-1,M-1,NN1))
      VLPMM=V(L+1,M-1,NN1)+RIND*(V(L+1,M-1,NN3)-V(L+1,M-1,NN1))
      VLMMM=V(L-1,M-1,NN1)+RIND*(V(L-1,M-1,NN3)-V(L-1,M-1,NN1))
      PLPMM=P(L+1,M-1,NN1)+RIND*(P(L+1,M-1,NN3)-P(L+1,M-1,NN1))
      PLMMM=P(L-1,M-1,NN1)+RIND*(P(L-1,M-1,NN3)-P(L-1,M-1,NN1))
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 380
      ROLPMM=RO(L+1,M-1,NN1)+RIND*(RO(L+1,M-1,NN3)-RO(L+1,M-1,NN1))
      ROLMMM=RO(L-1,M-1,NN1)+RIND*(RO(L-1,M-1,NN3)-RO(L-1,M-1,NN1))
      IF (ITM.LE.1) GO TO 380
      QLPMM=Q(L+1,M-1,NN1)+RIND*(Q(L+1,M-1,NN3)-Q(L+1,M-1,NN1))
      QLMMM=Q(L-1,M-1,NN1)+RIND*(Q(L-1,M-1,NN3)-Q(L-1,M-1,NN1))
      ELPMM=E(L+1,M-1,NN1)+RIND*(E(L+1,M-1,NN3)-E(L+1,M-1,NN1))
      ELMMM=E(L-1,M-1,NN1)+RIND*(E(L-1,M-1,NN3)-E(L-1,M-1,NN1))
      GO TO 380
C
C     LINEAR INTERPOLATION IN TIME FOR M=MVCT
C
  370 UMP=UU1(L)+RIND*(UU2(L)-UU1(L))
      VMP=VV1(L)+RIND*(VV2(L)-VV1(L))
      PMP=PP1(L)+RIND*(PP2(L)-PP1(L))
      ROMP=RORO1(L)+RIND*(RORO2(L)-RORO1(L))
      QMP=QQ1(L)+RIND*(QQ2(L)-QQ1(L))
      EMP=EE1(L)+RIND*(EE2(L)-EE1(L))
      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 390
      ULPMP=UU1(L+1)+RIND*(UU2(L+1)-UU1(L+1))
      ULMMP=UU1(L-1)+RIND*(UU2(L-1)-UU1(L-1))
      VLPMP=VV1(L+1)+RIND*(VV2(L+1)-VV1(L+1))
      VLMMP=VV1(L-1)+RIND*(VV2(L-1)-VV1(L-1))
      PLPMP=PP1(L+1)+RIND*(PP2(L+1)-PP1(L+1))
      PLMMP=PP1(L-1)+RIND*(PP2(L-1)-PP1(L-1))
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 380
      ROLPMP=RORO1(L+1)+RIND*(RORO2(L+1)-RORO1(L+1))
      ROLMMP=RORO1(L-1)+RIND*(RORO2(L-1)-RORO1(L-1))
      IF (ITM.LE.1) GO TO 380
      QLPMP=QQ1(L+1)+RIND*(QQ2(L+1)-QQ1(L+1))
      QLMMP=QQ1(L-1)+RIND*(QQ2(L-1)-QQ1(L-1))
      ELPMP=EE1(L+1)+RIND*(EE2(L+1)-EE1(L+1))
      ELMMP=EE1(L-1)+RIND*(EE2(L-1)-EE1(L-1))
C
C     CALCULATE THE INTERIOR POINT Y DERIVATIVES
C
  380 IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 390
      UY1=0.25*(UMP+ULMMP-UMM-ULMMM)*DYR
      UY2=0.25*(UMP+ULPMP-UMM-ULPMM)*DYR
```

```
      VY1=0.25*(VMP+VLMMP-VMM-VLMMM)*DYR
      VY2=0.25*(VMP+VLPMP-VMM-VLPMM)*DYR
      UX3=0.25*(ULP+ULPMM-ULM-ULMMM)*DXR
      UX4=0.25*(ULP+ULPMP-ULM-ULMMP)*DXR
      VX3=0.25*(VLP+VLPMM-VLM-VLMMM)*DXR
      VX4=0.25*(VLP+VLPMP-VLM-VLMMP)*DXR
  390 VY3=(V(L,M,N1)-VMM)*DYR
      VY4=(VMP-V(L,M,N1))*DYR
      UY3=(U(L,M,N1)-UMM)*DYR
      UY4=(UMP-U(L,M,N1))*DYR
      TMM=PMM/(ROMM*RG)
      TMP=PMP/(ROMP*RG)
      TY3=(T-TMM)*DYR
      TY4=(TMP-T)*DYR
      ROY3=(RO(L,M,N1)-ROMM)*DYR
      ROY4=(ROMP-RO(L,M,N1))*DYR
      IF (ITM.LT.2) GO TO 400
      ROQY=(ROMP*QMP-ROMM*QMM)*DYR*0.5
      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 400
      IF (M.EQ.MVCB.OR.M.EQ.MVCT) GO TO 400
      ROQY=QQT(L,M)
  400 IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 410
      TLMMM=PLMMM/(ROLMMM*RG)
      TLMMP=PLMMP/(ROLMMP*RG)
      TLPMM=PLPMM/(ROLPMM*RG)
      TLPMP=PLPMP/(ROLPMP*RG)
      TY1=0.25*(TMP+TLMMP-TMM-TLMMM)*DYR
      TY2=0.25*(TLPMP+TMP-TLPMM-TMM)*DYR
      TX3=0.25*(TLP+TLPMM-TLM-TLMMM)*DXR
      TX4=0.25*(TLPMP+TLP-TLMMP-TLM)*DXR
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 450
      ROY1=0.25*(ROMP+ROLMMP-ROMM-ROLMMM)*DYR
      ROY2=0.25*(ROMP+ROLPMP-ROMM-ROLPMM)*DYR
      ROX3=0.25*(ROLP+ROLPMM-ROLM-ROLMMM)*DXR
      ROX4=0.25*(ROLP+ROLPMP-ROLM-ROLMMP)*DXR
  410 IF (CAV.EQ.0.0) GO TO 430
      IF (NDIM.EQ.0) GO TO 420
      ATERM=V(L,M,N1)*RYP
      ATERM3=0.5*(V(L,M,N1)+VMM)*RYP
      ATERM4=0.5*(V(L,M,N1)+VMP)*RYP
      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 420
      ATERM1=0.5*(V(L,M,N1)+V(L-1,M,N1))*RYP
      ATERM2=0.5*(V(L,M,N1)+V(L+1,M,N1))*RYP
  420 IF (ISS.EQ.0) GO TO 430
      AMP=SQRT(GRG*TMP)
      AMM=SQRT(GRG*TMM)
      AY3=(A-AMM)*DYR
      AY4=(AMP-A)*DYR
  430 IF (ITM.LE.1) GO TO 450
      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 450
      QY1=0.25*(QMP+QLMMP-QMM-QLMMM)*DYR
      QY2=0.25*(QMP+QLPMP-QMM-QLPMM)*DYR
      QX3=0.25*(QLP+QLPMM-QLM-QLMMM)*DXR
      QX4=0.25*(QLP+QLPMP-QLM-QLMMP)*DXR
      QY3=(Q(L,M,N1)-QMM)*DYR
      QY4=(QMP-Q(L,M,N1))*DYR
      Q2Y=0.5*(SQRT(QMP)-SQRT(QMM))*DYR
      IF (ITM.EQ.3) GO TO 440
      ROSQ3=ROMM*SQRT(QMM)
      ROSQ4=ROMP*SQRT(QMP)
      GO TO 450
  440 EY1=0.25*(EMP+ELMMP-EMM-ELMMM)*DYR
      EY2=0.25*(EMP+ELPMP-EMM-ELPMM)*DYR
      EX3=0.25*(ELP+ELPMM-ELM-ELMMM)*DXR
      EX4=0.25*(ELP+ELPMP-ELM-ELMMP)*DXR
      EY3=(E(L,M,N1)-EMM)*DYR
      EY4=(EMP-E(L,M,N1))*DYR
      MUT3=CQMU*ROMM*QMM*QMM*LC/EMM
      MUT4=CQMU*ROMP*QMP*QMP*LC/EMP
C
C     SET THE BOUNDARY CONDITIONS FOR L=1 OR LMAX
C
```

```
  450 IF (L.NE.LMAX.AND.L.NE.1) GO TO 850
      IF (L.EQ.1) GO TO 460
      UX2=-UX1
      VX2=-VX1
      TX2=-TX1
      ROX2=-ROX1
      ROLP=ROLM
      TLP=TLM
      QLP=QLM
      ELP=ELM
      GO TO 470
  460 UX1=-UX2
      VX1=-VX2
      TX1=-TX2
      ROX1=-ROX2
      ROLM=ROLP
      TLM=TLP
      QLM=QLP
      ELM=ELP
  470 YP1=YP
      YP2=YP
      UY1=0.0
      UY2=0.0
      VY1=0.0
      VY2-0.0
      UX3=0.0
      UX4=0.0
      VX3=0.0
      VX4=0.0
      TY1=0.0
      TY2=0.0
      TX3=0.0
      TX4=0.0
      ROY1=0.0
      ROY2=0.0
      ROX3=0.0
      ROX4-0.0
      ATERM1=ATERM
      ATERM2=ATERM
      AX1=0.0
      AX2=0.0
      IF (ITM.LE.1) GO TO 850
      QX1=0.0
      QX2=0.0
      QY1=0.0
      QY2=0.0
      QX3=0.0
      QX4=0.0
      QY3=0.0
      QY4=0.0
      EX1=0.0
      EX2=0.0
      EY1=0.0
      EY2=0.0
      EX3=0.0
      EX4=Q.Q
      EY3=0.0
      EY4=0.0
      IF (ITM.EQ.3) GO TO 480
      ROSQ=RO(L,M,N1)*SQRT(Q(L,M,N1))
      ROSQ1=ROSQ
      ROSQ2=ROSQ
      ROSQ3=ROMM*SQRT(QMM)
      ROSQ4=ROMP*SQRT(QMP)
      GO TO 850
  480 MUT=CQMU*RO(L,M,N1)*Q(L,M,N1)*Q(L,M,N1)*LC/E(L,M,N1)
      MUT1=MUT
      MUT2=MUT
      MUT3=CQMU*ROMM*QMM*QMM*LC/EMM
      MUT4=CQMU*ROMP*QMP*QMP*LC/EMP
      GO TO 850
C
```

```
C       BEGIN THE CENTERBODY OR UPPER DUAL FLOW SPACE BOUNDARY POINT
C       Y DERIVATIVE CALCULATION
C
  490 DYP=DY/BE4
      UMP=U(L,M+1,N1)
      VMP=V(L,M+1,N1)
      PMP=P(L,M+1,N1)
      ROMP=RO(L,M+1,N1)
      QMP=Q(L,M+1,N1)
      EMP=E(L,M+1,N1)
      UX4=0.25*(U(L+1,M,N1)+U(L+1,M+1,N1)-U(L-1,M,N1)-U(L-1,M+1,N1))*DXR
      VX4=0.25*(V(L+1,M,N1)+V(L+1,M+1,N1)-V(L-1,M,N1)-V(L-1,M+1,N1))*DXR
      UY4=(UMP-U(L,M,N1))*DYR
      VY4=(VMP-V(L,M,N1))*DYR
      TMP=PMP/(ROMP*RG)
      TLMMP=P(L-1,M+1,N1)/(RO(L-1,M+1,N1)*RG)
      TLPMP=P(L+1,M+1,N1)/(RO(L+1,M+1,N1)*RG)
      TX4=0.25*(TLPMP+TLP-TLMMP-TLM)*DXR
      TY4=(TMP-T)*DYR
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 500
      ROX4=0.25*(RO(L+1,M,N1)+RO(L+1,M+1,N1)-RO(L-1,M,N1)-RO(L-1,M+1,N1)
     1 )*DXR
      ROY4=(ROMP-RO(L,M,N1))*DYR
      IF (ITM.LE.1) GO TO 500
      QX4=0.25*(Q(L+1,M,N1)+Q(L+1,M+1,N1)-Q(L-1,M,N1)-Q(L-1,M+1,N1))*DXR
      QY4=(QMP-Q(L,M,N1))*DYR
      IF (ITM.EQ.2) GO TO 500
      EX4=0.25*(E(L+1,M,N1)+E(L+1,M+1,N1)-E(L-1,M,N1)-E(L-1,M+1,N1))*DXR
      EY4=(EMP-E(L,M,N1))*DYR
C
C       REFLECT THE CENTERBODY OR UPPER DUAL FLOW SPACE BOUNDARY
C       CONDITIONS
C
  500 IF (M.EQ.1.AND.NGCB.EQ.0) GO TO 590
      IF (IVBC.NE.0) GO TO 600
      IF (M.EQ.MDFS) GO TO 510
      DNXNY=NXNYCB(L)
      DNXNYP=NXNYCB(L+1)
      DNXNYM=NXNYCB(L-1)
      GO TO 520
  510 DNXNY=NXNYU(L)
      DNXNYP=NXNYU(L+1)
      DNXNYM=NXNYU(L-1)
  520 THEW=ATAN(-DNXNY)
      IF (UMP.EQ.0.0) GO TO 530
      THE=ATAN(VMP/UMP)
      GO TO 540
  530 THE=0.0
  540 IF (UMP.LT.0.0) THE=THE+3.14159
      VMAG=SQRT(UMP*UMP+VMP*VMP)
      RTHE=2.0*THEW-THE
      IF (NOSLIP.EQ.1.AND.NGCB.NE.0) RTHE=3.14159+THE
      IF (NOSLIP.EQ.1.AND.M.EQ.MDFS) RTHE=3.14159+THE
      UMM=VMAG*COS(RTHE)
      VMM=VMAG*SIN(RTHE)
      THEW=ATAN(-DNXNYP)
      IF (U(L+1,M+1,N1).EQ.0.0) GO TO 550
      THE=ATAN(V(L+1,M+1,N1)/U(L+1,M+1,N1))
      GO TO 560
  550 THE=0.0
  560 IF (U(L+1,M+1,N1).LT.0.0) THE=THE+3.14159
      VMAG=SQRT(U(L+1,M+1,N1)*U(L+1,M+1,N1)+V(L+1,M+1,N1)*V(L+1,M+1,N1))
      RTHE=2.0*THEW-THE
      IF (NOSLIP.EQ.1.AND.NGCB.NE.0) RTHE=3.14159+THE
      IF (NOSLIP.EQ.1.AND.M.EQ.MDFS) RTHE=3.14159+THE
      ULPMM=VMAG*COS(RTHE)
      VLPMM=VMAG*SIN(RTHE)
      THEW=ATAN(-DNXNYM)
      IF (U(L-1,M+1,N1).EQ.0.0) GO TO 570
      THE=ATAN(V(L-1,M+1,N1)/U(L-1,M+1,N1))
      GO TO 580
  570 THE=0.0
```

```
  580 IF (U(L-1,M+1,N1).LT.0.0) THE=THE+3.14159
      VMAG=SQRT(U(L-1,M+1,N1)*U(L-1,M+1,N1)+V(L-1,M+1,N1)*V(L-1,M+1,N1))
      RTHE=2.0*THEW-THE
      IF (NOSLIP.EQ.1.AND.NGCB.NE.0) RTHE=3.14159+THE
      IF (NOSLIP.EQ.1.AND.M.EQ.MDFS) RTHE=3.14159+THE
      ULMMM=VMAG*COS(RTHE)
      VLMMM=VMAG*SIN(RTHE)
C
      RFL=2.0*DNXNY*DYP/(1.0+DNXNY*DNXNY)
      RFLP=2.0*DNXNYP*DYP/(1.0+DNXNYP*DNXNYP)
      RFLM=2.0*DNXNYM*DYP/(1.0+DNXNYM*DNXNYM)
      TTERM=0.5*(OM1*TX1+OM2*TX2)
      TMM=TMP+TTERM*RFL
      TLPMM=TLPMP+TTERM*RFLP
      TLMMM=TLMMP+TTERM*RFLM
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 610
      ROTERM=0.5*(OM1*ROX1+OM2*ROX2)
      ROMM=ROMP+ROTERM*RFL
      ROLPMM=RO(L+1,M+1,N1)+ROTERM*RFLP
      ROLMMM=RO(L-1,M+1,N1)+ROTERM*RFLM
      IF (ITM.LE.1) GO TO 610
      QTERM=0.5*(OM1*QX1+OM2*QX2)
      QMM=QMP+QTERM*RFL
      QLPMM=Q(L+1,M+1,N1)+QTERM*RFLP
      QLMMM=Q(L-1,M+1,N1)+QTERM*RFLM
      IF (ITM.EQ.2) GO TO 610
      ETERM=0.5*(OM1*EX1+OM2*EX2)
      EMM=EMP+ETERM*RFL
      ELPMM=E(L+1,M+1,N1)+ETERM*RFLP
      ELMMM=E(L-1,M+1,N1)+ETERM*RFLM
      GO TO 610
C
C     REFLECT THE CENTERLINE OR MIDPLANE BOUNDARY CONDITIONS
C
  590 UMM=UMP
      VMM=-VMP
      ULPMM=U(L+1,M+1,N1)
      VLPMM=-V(L+1,M+1,N1)
      ULMMM=U(L-1,M+1,N1)
      VLMMM=-V(L-1,M+1,N1)
      TMM=TMP
      TLPMM=TLPMP
      TLMMM=TLMMP
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 610
      ROMM=ROMP
      ROLPMM=RO(L+1,M+1,N1)
      ROLMMM=RO(L-1,M+1,N1)
      IF (ITM.LE.1) GO TO 610
      QMM=QMP
      QLPMM=Q(L+1,M+1,N1)
      QLMMM=Q(L-1,M+1,N1)
      IF (ITM.EQ.2) GO TO 610
      EMM=EMP
      ELPMM=E(L+1,M+1,N1)
      ELMMM=E(L-1,M+1,N1)
      GO TO 610
C
C     EXTRAPOLATE THE CENTERBODY OR UPPER DUAL FLOW SPACE BOUNDARY
C     CONDITIONS
C
  600 UMM=U(L,M,N1)+F2I*(U(L,M,N1)-UMP)
      VMM=V(L,M,N1)+F2I*(V(L,M,N1)-VMP)
      ULPMM=U(L+1,M,N1)+F2I*(U(L+1,M,N1)-U(L+1,M+1,N1))
      VLPMM=V(L+1,M,N1)+F2I*(V(L+1,M,N1)-V(L+1,M+1,N1))
      ULMMM=U(L-1,M,N1)+F2I*(U(L-1,M,N1)-U(L-1,M+1,N1))
      VLMMM=V(L-1,M,N1)+F2I*(V(L-1,M,N1)-V(L-1,M+1,N1))
      TMM=T+F2I*(T-TMP)
      TLPMM=TLP+F2I*(TLP-TLPMP)
      TLMMM=TLM+F2I*(TLM-TLMMP)
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 610
      ROMM=RO(L,M,N1)+F2I*(RO(L,M,N1)-ROMP)
      ROLPMM=RO(L+1,M,N1)+F2I*(RO(L+1,M,N1)-RO(L+1,M+1,N1))
```

```
      ROLMMM=RO(L-1,M,N1)+F2I*(RO(L-1,M,N1)-RO(L-1,M+1,N1))
      IF (ITM.LE.1) GO TO 610
      QMM=Q(L,M,N1)+F2I*(Q(L,M,N1)-QMP)
      QLPMM=Q(L+1,M,N1)+F2I*(Q(L+1,M,N1)-Q(L+1,M+1,N1))
      QLMMM=Q(L-1,M,N1)+F2I*(Q(L-1,M,N1)-Q(L-1,M+1,N1))
      IF (ITM.EQ.2) GO TO 610
      EMM=E(L,M,N1)+F2I*(E(L,M,N1)-EMP)
      ELPMM=E(L+1,M,N1)+F2I*(E(L+1,M,N1)-E(L+1,M+1,N1))
      ELMMM=E(L-1,M,N1)+F2I*(E(L-1,M,N1)-E(L-1,M+1,N1))
C
C     CALCULATE THE CENTERBODY OR UPPER DUAL FLOW SPACE BOUNDARY
C     DERIVATIVES
C
  610 IF (M.NE.MDFS) GO TO 630
      IF (L.NE.LDFSF) GO TO 620
      ULPMM=U(L+1,M-1,N1)
      VLPMM=V(L+1,M-1,N1)
      TLPMM=P(L+1,M-1,N1)/(RO(L+1,M-1,N1)*RG)
      ROLPMM=RO(L+1,M-1,N1)
      IF (ITM.LE.1) GO TO 630
      QLPMM=Q(L+1,M-1,N1)
      ELPMM=E(L+1,M-1,N1)
      GO TO 630
  620 IF (L.NE.LDFSS) GO TO 630
      ULMMM=U(L-1,M-1,N1)
      VLMMM=V(L-1,M-1,N1)
      TLMMM=P(L-1,M-1,N1)/(RO(L-1,M-1,N1)*RG)
      ROLMMM=RO(L-1,M-1,N1)
      IF (ITM.LE.1) GO TO 630
      QLMMM=Q(L-1,M-1,N1)
      ELMMM=E(L-1,M-1,N1)
  630 UY1=0.25*(UMP+U(L-1,M+1,N1)-UMM-ULMMM)*DYR
      VY1=0.25*(VMP+V(L-1,M+1,N1)-VMM-VLMMM)*DYR
      UY2=0.25*(UMP+U(L+1,M+1,N1)-UMM-ULPMM)*DYR
      VY2=0.25*(VMP+V(L+1,M+1,N1)-VMM-VLPMM)*DYR
      UY3=(U(L,M,N1)-UMM)*DYR
      VY3=(V(L,M,N1)-VMM)*DYR
      UX3=0.25*(U(L+1,M,N1)+ULPMM-U(L-1,M,N1)-ULMMM)*DXR
      VX3=0.25*(V(L+1,M,N1)+VLPMM-V(L-1,M,N1)-VLMMM)*DXR
      TY1=0.25*(TMP+TLMMP-TMM-TLMMM)*DYR
      TY2=0.25*(TMP+TLPMP-TMM-TLPMM)*DYR
      TX3=0.25*(TLP+TLPMM-TLM-TLMMM)*DXR
      TY3=(T-TMM)*DYR
      TMM=TMM
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 850
      ROY1=0.25*(ROMP+RO(L-1,M+1,N1)-ROMM-ROLMMM)*DYR
      ROY2=0.25*(ROMP+RO(L+1,M+1,N1)-ROMM-ROLPMM)*DYR
      ROY3=(RO(L,M,N1)-ROMM)*DYR
      ROX3=0.25*(RO(L+1,M,N1)+ROLPMM-RO(L-1,M,N1)-ROLMMM)*DXR
      IF (CAV.EQ.0.0) GO TO 660
      IF (NDIM.EQ.0) GO TO 650
      IF (M.EQ.1.AND.YCB(L).EQ.0.0) GO TO 640
      ATERM=V(L,M,N1)*RYP
      ATERM1=0.5*(V(L,M,N1)+V(L-1,M,N1))*RYP
      ATERM2=0.5*(V(L,M,N1)+V(L+1,M,N1))*RYP
      ATERM3=0.5*(V(L,M,N1)+VMM)*RYP
      ATERM4=0.5*(V(L,M,N1)+VMP)*RYP
      IF (M.EQ.MDFS) GO TO 650
      IF (YCB(L-1).EQ.0.0) ATERM1=0.5*(BE4*V(L-1,M+1,N1)*DYR+V(L,M,N1)
     1 /YCB(L))
      IF (YCB(L+1).EQ.0.0) ATERM2=0.5*(BE4*V(L+1,M+1,N1)*DYR+V(L,M,N1)
     1 /YCB(L))
      IF (YCB(L-1).EQ.0.0.OR.YCB(L+1).EQ.0.0) ATERM=0.5*(ATERM1+ATERM2)
      GO TO 650
  640 ATERM=BE*VMP*DYR
      ATERM1=BE4*0.5*(VMP+V(L-1,M+1,N1))*DYR
      ATERM2=BE4*0.5*(VMP+V(L+1,M+1,N1))*DYR
      ATERM4=BE4*VMP*DYR
      ATERM3=ATERM4
      IF (YCB(L-1).NE.0.0) ATERM1=0.5*(V(L-1,M,N1)/YCB(L-1)+BE4*VMP*DYR)
      IF (YCB(L+1).NE.0.0) ATERM2=0.5*(V(L+1,M,N1)/YCB(L+1)+BE4*VMP*DYR)
      IF (YCB(L-1).NE.0.0.OR.YCB(L+1).NE.0.0) ATERM=0.5*(ATERM1+ATERM2)
  650 IF (ISS.EQ.0) GO TO 660
      AMP=SQRT(GRG*TMP)
```

```
      AMM=SQRT(GRG*TMM)
      AY3=(A-AMM)*DYR
      AY4=(AMP-A)*DYR
  660 IF (ITM.LE.1) GO TO 850
      ROQY=0.5*(ROMP*QMP-ROMM*QMM)*DYR
      QY1=0.25*(QMP+Q(L-1,M+1,N1)-QMM-QLMMM)*DYR
      QY2=0.25*(QMP+Q(L+1,M+1,N1)-QMM-QLPMM)*DYR
      QY3=(Q(L,M,N1)-QMM)*DYR
      QX3=0.25*(Q(L+1,M,N1)+QLPMM-Q(L-1,M,N1)-QLMMM)*DXR
      Q2Y=0.5*(SQRT(ABS(QMP))-SQRT(ABS(QMM)))*DYR
      IF (ITM.EQ.3) GO TO 670
      ROSQ3=ROMM*SQRT(ABS(QMM))
      ROSQ4=ROMP*SQRT(ABS(QMP))
      GO TO 850
  670 EY1=0.25*(EMP+E(L-1,M+1,N1)-EMM-ELMMM)*DYR
      EY2=0.25*(EMP+E(L+1,M+1,N1)-EMM-ELPMM)*DYR
      EY3=(E(L,M,N1)-EMM)*DYR
      EX3=0.25*(E(L+1,M,N1)+ELPMM-E(L-1,M,N1)-ELMMM)*DXR
      MUT3=CQMU*ROMM*QMM*QMM*LC/ABS(EMM)
      MUT4=CQMU*ROMP*QMP*QMP*LC/ABS(EMP)
      IF (M.EQ.1.AND.NGCB.EQ.0) MUT=0.5*(MUT3+MUT4)
      GO TO 850
C
C     BEGIN THE WALL OR LOWER DUAL FLOW SPACE BOUNDARY POINT
C     Y DERIVATIVE CALCULATION
C
  680 DYP=DY/BE3
      UMM=U(L,M-1,N1)
      VMM=V(L,M-1,N1)
      PMM=P(L,M-1,N1)
      ROMM=RO(L,M-1,N1)
      QMM=Q(L,M-1,N1)
      EMM=E(L,M-1,N1)
      UX3=0.25*(U(L+1,M,N1)+U(L+1,M-1,N1)-U(L-1,M,N1)-U(L-1,M-1,N1))*DXR
      VX3=0.25*(V(L+1,M,N1)+V(L+1,M-1,N1)-V(L-1,M,N1)-V(L-1,M-1,N1))*DXR
      UY3=(U(L,M,N1)-UMM)*DYR
      VY3=(V(L,M,N1)-VMM)*DYR
      TLPMM=P(L+1,M-1,N1)/(RO(L+1,M-1,N1)*RG)
      TMM=PMM/(ROMM*RG)
      TLMMM=P(L-1,M-1,N1)/(RO(L-1,M-1,N1)*RG)
      TX3=0.25*(TLP+TLPMM-TLM-TLMMM)*DXR
      TY3=(T-TMM)*DYR
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 690
      ROX3=0.25*(RO(L+1,M,N1)+RO(L+1,M-1,N1)-RO(L-1,M,N1)-RO(L-1,M-1,N1)
     1 )*DXR
      ROY3=(RO(L,M,N1)-ROMM)*DYR
      IF (ITM.LE.1) GO TO 690
      QX3=0.25*(Q(L+1,M,N1)+Q(L+1,M-1,N1)-Q(L-1,M,N1)-Q(L-1,M-1,N1))*DXR
      QY3=(Q(L,M,N1)-QMM)*DYR
      IF (ITM.EQ.2) GO TO 690
      EX3=0.25*(E(L+1,M,N1)+E(L+1,M-1,N1)-E(L-1,M,N1)-E(L-1,M-1,N1))*DXR
      EY3=(E(L,M,N1)-EMM)*DYR
C
C     REFLECT THE WALL OR LOWER DUAL FLOW SPACE BOUNDARY CONDITIONS
C
  690 IF (IVBC.NE.0) GO TO 780
      IF (IWALL.EQ.1.AND.M.EQ.MMAX) GO TO 780
      IF (M.EQ.MDFS) GO TO 700
      DNXNY=NXNY(L)
      DNXNYP=NXNY(L+1)
      DNXNYM=NXNY(L-1)
      GO TO 710
  700 DNXNY=NXNYL(L)
      DNXNYP=NXNYL(L+1)
      DNXNYM=NXNYL(L-1)
  710 THEW=ATAN(-DNXNY)
      IF (UMM.EQ.0.0) GO TO 720
      THE=ATAN(VMM/UMM)
      GO TO 730
  720 THE=0.0
  730 IF (UMM.LT.0.0) THE=THE+3.14159
      VMAG=SQRT(UMM*UMM+VMM*VMM)
      RTHE=2.0*THEW-THE
```

```
      IF (NOSLIP.EQ.1) RTHE=3.14159+THE
      UMP=VMAG*COS(RTHE)
      VMP=VMAG*SIN(RTHE)
      THEW=ATAN(-DNXNYP)
      IF (U(L+1,M-1,N1).EQ.0.0) GO TO 740
      THE=ATAN(V(L+1,M-1,N1)/U(L+1,M-1,N1))
      GO TO 750
  740 THE=0.0
  750 IF (U(L+1,M-1,N1).LT.0.0) THE=THE+3.14159
      VMAG=SQRT(U(L+1,M-1,N1)*U(L+1,M-1,N1)+V(L+1,M-1,N1)*V(L+1,M-1,N1))
      RTHE=2.0*THEW-THE
      IF (NOSLIP.EQ.1) RTHE=3.14159+THE
      ULPMP=VMAG*COS(RTHE)
      VLPMP=VMAG*SIN(RTHE)
      THEW=ATAN(-DNXNYM)
      IF (U(L-1,M-1,N1).EQ.0.0) GO TO 760
      THE=ATAN(V(L-1,M-1,N1)/U(L-1,M-1,N1))
      GO TO 770
  760 THE=0.0
  770 IF (U(L-1,M-1,N1).LT.0.0) THE=THE+3.14159
      VMAG=SQRT(U(L-1,M-1,N1)*U(L-1,M-1,N1)+V(L-1,M-1,N1)*V(L-1,M-1,N1))
      RTHE=2.0*THEW-THE
      IF (NOSLIP.EQ.1) RTHE=3.14159+THE
      ULMMP=VMAG*COS(RTHE)
      VLMMP=VMAG*SIN(RTHE)
C
      RFL=2.0*DNXNY*DYP/(1.0+DNXNY*DNXNY)
      RFLP=2.0*DNXNYP*DYP/(1.0+DNXNYP*DNXNYP)
      RFLM=2.0*DNXNYM*DYP/(1.0+DNXNYM*DNXNYM)
      TTERM=0.5*(OM1*TX1+OM2*TX2)
      TMP=TMM-TTERM*RFL
      TLPMP=TLPMM-TTERM*RFLP
      TLMMP=TLMMM-TTERM*RFLM
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 790
      ROTERM=0.5*(OM1*ROX1+OM2*ROX2)
      ROMP=ROMM-ROTERM*RFL
      ROLPMP=RO(L+1,M-1,N1)-ROTERM*RFLP
      ROLMMP=RO(L-1,M-1,N1)-ROTERM*RFLM
      IF (ITM.LE.1) GO TO 790
      QTERM=0.5*(OM1*QX1+OM2*QX2)
      QMP=QMM-QTERM*RFL
      QLPMP=Q(L+1,M-1,N1)-QTERM*RFLP
      QLMMP=Q(L-1,M-1,N1)-QTERM*RFLM
      IF (ITM.EQ.2) GO TO 790
      ETERM=0.5*(OM1*EX1+OM2*EX2)
      EMP=EMM-ETERM*RFL
      ELPMP=E(L+1,M-1,N1)-ETERM*RFLP
      ELMMP=E(L-1,M-1,N1)-ETERM*RFLM
      GO TO 790
C
C     EXTRAPOLATE THE WALL OR LOWER DUAL FLOW SPACE BOUNDARY CONDITIONS
C
  780 UMP=U(L,M,N1)+F2I*(U(L,M,N1)-UMM)
      VMP=V(L,M,N1)+F2I*(V(L,M,N1)-VMM)
      ULPMP=U(L+1,M,N1)+F2I*(U(L+1,M,N1)-U(L+1,M-1,N1))
      VLPMP=V(L+1,M,N1)+F2I*(V(L+1,M,N1)-V(L+1,M-1,N1))
      ULMMP=U(L-1,M,N1)+F2I*(U(L-1,M,N1)-U(L-1,M-1,N1))
      VLMMP=V(L-1,M,N1)+F2I*(V(L-1,M,N1)-V(L-1,M-1,N1))
      TMP=T+F2I*(T-TMM)
      TLPMP=TLP+F2I*(TLP-TLPMM)
      TLMMP=TLM+F2I*(TLM-TLMMM)
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 790
      ROMP=RO(L,M,N1)+F2I*(RO(L,M,N1)-ROMM)
      ROLPMP=RO(L+1,M,N1)+F2I*(RO(L+1,M,N1)-RO(L+1,M-1,N1))
      ROLMMP=RO(L-1,M,N1)+F2I*(RO(L-1,M,N1)-RO(L-1,M-1,N1))
      IF (ITM.LE.1) GO TO 790
      QMP=Q(L,M,N1)+F2I*(Q(L,M,N1)-QMM)
      QLPMP=Q(L+1,M,N1)+F2I*(Q(L+1,M,N1)-Q(L+1,M-1,N1))
      QLMMP=Q(L-1,M,N1)+F2I*(Q(L-1,M,N1)-Q(L-1,M-1,N1))
      IF (ITM.EQ.2) GO TO 790
      EMP=E(L,M,N1)+F2I*(E(L,M,N1)-EMM)
      ELPMP=E(L+1,M,N1)+F2I*(E(L+1,M,N1)-E(L+1,M-1,N1))
```

```
      ELMMP=E(L-1,M,N1)+F2I*(E(L-1,M,N1)-E(L-1,M-1,N1))
C
C     CALCULATE THE WALL AND LOWER DUAL FLOW SPACE BOUNDARY DERIVATIVES
C
  790 IF (M.NE.MDFS) GO TO 810
      IF (L.NE.LDFSF) GO TO 800
      ULPMP=U(L+1,M+1,N1)
      VLPMP=V(L+1,M+1,N1)
      TLPMP=P(L+1,M+1,N1)/(RO(L+1,M+1,N1)*RG)
      ROLPMP=RO(L+1,M+1,N1)
      IF (ITM.LE.1) GO TO 810
      QLPMP=Q(L+1,M+1,N1)
      ELPMP=E(L+1,M+1,N1)
      GO TO 810
  800 IF (L.NE.LDFSS) GO TO 810
      ULMMP=U(L-1,M+1,N1)
      VLMMP=V(L-1,M+1,N1)
      TLMMP=P(L-1,M+1,N1)/(RO(L-1,M+1,N1)*RG)
      ROLMMP=RO(L-1,M+1,N1)
      IF (ITM.LE.1) GO TO 810
      QLMMP=Q(L-1,M+1,N1)
      ELMMP=E(L-1,M+1,N1)
  810 UY1=0.25*(UMP+ULMMP-UMM-U(L-1,M-1,N1))*DYR
      VY1=0.25*(VMP+VLMMP-VMM-V(L-1,M-1,N1))*DYR
      UY2=0.25*(UMP+ULPMP-UMM-U(L+1,M-1,N1))*DYR
      VY2=0.25*(VMP+VLPMP-VMM-V(L+1,M-1,N1))*DYR
      UY4=(UMP-U(L,M,N1))*DYR
      VY4=(VMP-V(L,M,N1))*DYR
      UX4=0.25*(U(L+1,M,N1)+ULPMP-U(L-1,M,N1)-ULMMP)*DXR
      VX4=0.25*(V(L+1,M,N1)+VLPMP-V(L-1,M,N1)-VLMMP)*DXR
      TY1=0.25*(TMP+TLMMP-TMM-TLMMM)*DYR
      TY2=0.25*(TMP+TLPMP-TMM-TLPMM)*DYR
      TX4=0.25*(TLP+TLPMP-TLM-TLMMP)*DXR
      TY4=(TMP-T)*DYR
      TMP=TMP
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 850
      ROY1=0.25*(ROMP+ROLMMP-ROMM-RO(L-1,M-1,N1))*DYR
      ROY2=0.25*(ROMP+ROLPMP-ROMM-RO(L+1,M-1,N1))*DYR
      ROX4=0.25*(RO(L+1,M,N1)+ROLPMP-RO(L-1,M,N1)-ROLMMP)*DXR
      ROY4=(ROMP-RO(L,M,N1))*DYR
      IF (CAV.EQ.0.0) GO TO 830
      IF (NDIM.EQ.0) GO TO 820
      ATERM=V(L,M,N1)*RYP
      ATERM1=0.5*(V(L,M,N1)+V(L-1,M,N1))*RYP
      ATERM2=0.5*(V(L,M,N1)+V(L+1,M,N1))*RYP
      ATERM3=0.5*(V(L,M,N1)+VMM)*RYP
      ATERM4=0.5*(V(L,M,N1)+VMP)*RYP
  820 IF (ISS.EQ.0) GO TO 830
      AMP=SQRT(GRG*TMP)
      AMM=SQRT(GRG*TMM)
      AY3=(A-AMM)*DYR
      AY4=(AMP-A)*DYR
  830 IF (ITM.LE.1) GO TO 850
      ROQY=0.5*(ROMP*QMP-ROMM*QMM)*DYR
      QY1=0.25*(QMP+QLMMP-QMM-Q(L-1,M-1,N1))*DYR
      QY2=0.25*(QMP+QLPMP-QMM-Q(L+1,M-1,N1))*DYR
      QX4=0.25*(Q(L+1,M,N1)+QLPMP-Q(L-1,M,N1)-QLMMP)*DXR
      QY4=(QMP-Q(L,M,N1))*DYR
      Q2Y=0.5*(SQRT(ABS(QMP))-SQRT(ABS(QMM)))*DYR
      IF (ITM.EQ.3) GO TO 840
      ROSQ3=ROMM*SQRT(ABS(QMM))
      ROSQ4=ROMP*SQRT(ABS(QMP))
      GO TO 850
  840 EY1=0.25*(EMP+ELMMP-EMM-E(L-1,M-1,N1))*DYR
      EY2=0.25*(EMP+ELPMP-EMM-E(L+1,M-1,N1))*DYR
      EX4=0.25*(E(L+1,M,N1)+ELPMP-E(L-1,M,N1)-ELMMP)*DXR
      EY4=(EMP-E(L,M,N1))*DYR
      MUT3=COMU*ROMM*QMM*QMM*LC/ABS(EMM)
      MUT4=COMU*ROMP*QMP*QMP*LC/ABS(EMP)
C
C     COMBINE TERMS
C
```

```
  850 UXY1=OM1*UX1+AL*UY1
      UXY2=OM2*UX2+AL*UY2
      UXY3=OM*UX3+AL3*UY3
      UXY4=OM*UX4+AL4*UY4
      UXY12=0.5*(OM1*UX1+OM2*UX2+AL3*UY3+AL4*UY4)
      VXY1=OM1*VX1+AL*VY1
      VXY2=OM2*VX2+AL*VY2
      VXY3=OM*VX3+AL3*VY3
      VXY4=OM*VX4+AL4*VY4
      VXY12=0.5*(OM1*VX1+OM2*VX2+AL3*VY3+AL4*VY4)
      BUY1=BE*UY1
      BUY2=BE*UY2
      BUY3=BE3*UY3
      BUY4=BE4*UY4
      BUY34=0.5*(BE3*UY3+BE4*UY4)
      BVY1=BE*VY1
      BVY2=BE*VY2
      BVY3=BE3*VY3
      BVY4=BE4*VY4
      BVY34=0.5*(BE3*VY3+BE4*VY4)
      TXY1=OM1*TX1+AL*TY1
      TXY2=OM2*TX2+AL*TY2
      TXY3=OM*TX3+AL3*TY3
      TXY4=OM*TX4+AL4*TY4
      BTY3=BE3*TY3
      BTY4=BE4*TY4
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 940
      ROXY1=OM1*ROX1+AL*ROY1
      ROXY2=OM2*ROX2+AL*ROY2
      ROXY3=OM*ROX3+AL3*ROY3
      ROXY4=OM*ROX4+AL4*ROY4
      ROXY12=0.5*(OM1*ROX1+OM2*ROX2+AL3*ROY3+AL4*ROY4)
      BROY1=BE*ROY1
      BROY2=BE*ROY2
      BROY3=BE3*ROY3
      BROY4=BE4*ROY4
      BROY34=0.5*(BE3*ROY3+BE4*ROY4)
      IF (ISS.EQ.0) GO TO 860
      AXY1=OM1*AX1+0.5*AL*(AY3+AY4)
      AXY2=OM2*AX2+0.5*AL*(AY3+AY4)
      AXY12=0.5*(AXY1+AXY2)
      BAY3=BE3*AY3
      BAY4=BE4*AY4
      BAY34=0.5*(BAY3+BAY4)
  860 IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 870
      IF (ITM.LE.1) GO TO 870
      QXY1=OM1*QX1+AL*QY1
      QXY2=OM2*QX2+AL*QY2
      QXY3=OM*QX3+AL3*QY3
      QXY4=OM*QX4+AL4*QY4
      BQY3=BE3*QY3
      BQY4=BE4*QY4
      BQY34=0.5*(BE3*QY3+BE4*QY4)
      Q2XY=OM*Q2X+AL*Q2Y
      BQ2Y=BE*Q2Y
      IF (ITM.EQ.2) GO TO 870
      EXY1=OM1*EX1+AL*EY1
      EXY2=OM2*EX2+AL*EY2
      EXY3=OM*EX3+AL3*EY3
      EXY4=OM*EX4+AL4*EY4
      BEY3=BE3*EY3
      BEY4=BE4*EY4
      BEY34=0.5*(BE3*EY3+BE4*EY4)
C
C     CALCULATE THE ARTIFICAL VISCOSITY COEFFICIENTS
C
  870 IF (CAV.EQ.0.0) GO TO 940
      IF (L.LT.LSS) GO TO 880
      IF (L.GT.LSF) GO TO 880
      IF (M.LT.MSS) GO TO 880
      IF (M.GT.MSF) GO TO 880
      IF (SMACH.EQ.0.0) GO TO 890
```

```
      IF (XM.LT.SMACH*SMACH) GO TO 880
      GO TO 890
  880 DIV1=0.0
      DIV2=0.0
      DIV3=0.0
      DIV4=0.0
      GO TO 910
  890 DIV1=UXY1+BVY1+ATERM1
      DIV2=UXY2+BVY2+ATERM2
      DIV3=UXY3+BVY3+ATERM3
      DIV4=UXY4+BVY4+ATERM4
      IF (IDIVC.NE.0) GO TO 910
      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 900
      IF (DIV1.GT.0.0) DIV1=0.0
      IF (DIV2.GT.0.0) DIV2=0.0
  900 IF (DIV3.GT.0.0) DIV3=0.0
      IF (DIV4.GT.0.0) DIV4=0.0
  910 IF (ISS.EQ.0) GO TO 930
      IF (ISS.EQ.1) GO TO 920
      DIV1=ABS(DIV1)+ABS(AXY1+BAY34)
      DIV2=ABS(DIV2)+ABS(AXY2+BAY34)
      DIV3=ABS(DIV3)+ABS(AXY12+BAY3)
      DIV4=ABS(DIV4)+ABS(AXY12+BAY4)
      GO TO 930
  920 IF (DIV1.NE.0.0) DIV1=ABS(DIV1)+ABS(AXY1+BAY34)
      IF (DIV2.NE.0.0) DIV2=ABS(DIV2)+ABS(AXY2+BAY34)
      IF (DIV3.NE.0.0) DIV3=ABS(DIV3)+ABS(AXY12+BAY3)
      IF (DIV4.NE.0.0) DIV4=ABS(DIV4)+ABS(AXY12+BAY4)
  930 DRLA=XLA*CAV*2.0*RO(L,M,N1)*DXP*DYP
      RLA1=DRLA*ABS(DIV1)
      RLA2=DRLA*ABS(DIV2)
      RLA3=DRLA*ABS(DIV3)*SMT
      RLA4=DRLA*ABS(DIV4)*SMT
      RLA=0.25*(RLA1+RLA2+RLA3+RLA4)
      XMULA=XMU/XLA
      RMU1=XMULA*RLA1
      RMU2=XMULA*RLA2
      RMU3=XMULA*RLA3
      RMU4=XMULA*RLA4
      RMU=0.25*(RMU1+RMU2+RMU3+RMU4)
      RK1=DRK*RMU1
      RK2=DRK*RMU2
      RK3=DRK*RMU3
      RK4=DRK*RMU4
      RK=0.25*(RK1+RK2+RK3+RK4)
      RRO1=XRO*RMU1
      RRO2=XRO*RMU2
      RRO3=XRO*RMU3
      RRO4=XRO*RMU4
      RRO=0.25*(RRO1+RRO2+RRO3+RRO4)
      RLP2M=RLA+2.0*RMU
      RLP2M1=RLA1+2.0*RMU1
      RLP2M2=RLA2+2.0*RMU2
      RLP2M3=RLA3+2.0*RMU3
      RLP2M4=RLA4+2.0*RMU4
      RLPM=RLA+RMU
C
C     CALCULATE THE MOLECULAR VISCOSITY COEFFICIENTS
C
  940 IF (CHECK.EQ.0.0) GO TO 1190
      TCHECK=T*TLP*TLM*TMP*TMM
      IF (TCHECK.GT.0.0) GO TO 950
      NP=N+NSTART
      WRITE (6,1510) NP,L,M,NVC
      IERR=1
      RETURN
  950 IF (ECHECK.EQ.0.0) GO TO 960
      IF (ECHECK.LT.0.0) GO TO 970
      MU=CMU*T**EMU
      LA=CLA*T**ELA
      K=CK*T**EK
      MU1=(CMU*TLM**EMU+MU)*0.5
```

```
      MU2=(CMU*TLP**EMU+MU)*0.5
      MU3=(CMU*TMM**EMU+MU)*0.5
      MU4=(CMU*TMP**EMU+MU)*0.5
      LA1=(CLA*TLM**ELA+LA)*0.5
      LA2=(CLA*TLP**ELA+LA)*0.5
      LA3=(CLA*TMM**ELA+LA)*0.5
      LA4=(CLA*TMP**ELA+LA)*0.5
      K1=(CK*TLM**EK+K)*0.5
      K2=(CK*TLP**EK+K)*0.5
      K3=(CK*TMM**EK+K)*0.5
      K4=(CK*TMP**EK+K)*0.5
      GO TO 980
C
  960 MU=CMU
      MU1=CMU
      MU2=CMU
      MU3=CMU
      MU4=CMU
      LA=CLA
      LA1=CLA
      LA2=CLA
      LA3=CLA
      LA4=CLA
      K=CK
      K1=CK
      K2=CK
      K3=CK
      K4=CK
      GO TO 980
C
  970 SQT=T**EMU
      MU=CMU*SQT
      LA=CLA*SQT
      K=CK*SQT
      SQTLM=(TLM**EMU+SQT)*0.5
      SQTLP=(TLP**EMU+SQT)*0.5
      SQTMM=(TMM**EMU+SQT)*0.5
      SQTMP=(TMP**EMU+SQT)*0.5
      MU1=CMU*SQTLM
      MU2=CMU*SQTLP
      MU3=CMU*SQTMM
      MU4=CMU*SQTMP
      LA1=CLA*SQTLM
      LA2=CLA*SQTLP
      LA3=CLA*SQTMM
      LA4=CLA*SQTMP
      K1=CK*SQTLM
      K2=CK*SQTLP
      K3=CK*SQTMM
      K4=CK*SQTMP
  980 LP2M=LA+2.0*MU
      LP2M1=LA1+2.0*MU1
      LP2M2=LA2+2.0*MU2
      LP2M3=LA3+2.0*MU3
      LP2M4=LA4+2.0*MU4
      LPM=LA+MU
      AVMUR=RMU/MU
      IF (RLA.GT.0.0) AVMUR=RLA/MU
C
C     CALCULATE THE TURBULENT VISCOSITY COEFFICIENTS
C
      IF (ITM.EQ.0) GO TO 1190
      IF (ITM.EQ.3) GO TO 1160
      IF (IMLM.EQ.2) GO TO 1010
C
      DELTAY=YSL2-YSL1
      IF (IMP.NE.0) GO TO 990
      IF (M.LT.MMIN) DELTAY=YMIN-YSL1
      IF (M.GT.MMIN) DELTAY=YSL2-YMIN
      IF (M.NE.MMIN) GO TO 990
      DELTAY=0.5*(YSL2-YSL1)
      DELTAY3=YMIN-YSL1
      DELTAY4=YSL2-YMIN
```

```
  990 TML=CML1*ABS(DELTAY)
      IF (IMP.EQ.O) TML=CML2*ABS(DELTAY)
      IF (ITM.EQ.2) GO TO 1140
      TML3=TML
      TML4=TML
      IF (IMP.NE.O) GO TO 1080
      IF (M.EQ.MMIN-1.OR.M.EQ.MMIN+1) GO TO 1000
      IF (M.NE.MMIN) GO TO 1080
      TML3=CML2*DELTAY3
      TML4=CML2*DELTAY4
 1000 IF (L.NE.LDFSS-1.AND.L.NE.LDFSF+1) GO TO 1080
      TML=0.5*TML
      TML3=0.5*TML3
      TML4=0.5*TML4
      GO TO 1080
C
 1010 YWB=YCB(L)
      YWT=YW(L)
      IF (MDFS.EQ.O) GO TO 1030
      IF (IB.EQ.1.OR.M.GT.MDFS) GO TO 1020
      YWT=YL(L)
      TAUW=TAUWM
      GO TO 1050
 1020 YWB=YU(L)
      TAUW=TAUWP
      GO TO 1040
 1030 IF (NGCB.EQ.O) GO TO 1050
 1040 YPD=YP-YWB
      YPD3=YP3-YWB
      YPD4=YP4-YWB
      GO TO 1060
 1050 YPD=YWT-YP
      YPD3=YWT-YP3
      YPD4=YWT-YP4
 1060 IF (YPD3.LT.O.O) YPD3=YPD4
      IF (YPD4.LT.O.O) YPD4=YPD3
      YDUM=SQRT(RO(L,M,N1)*MU*TAUW)/(26.0*MU)
      YPLUS=YPD*YDUM
      YPLUS3=YPD3*YDUM
      YPLUS4=YPD4*YDUM
      TML=0.4*YPD*(1.0-EXP(-YPLUS))
      TML3=0.4*YPD3*(1.0-EXP(-YPLUS3))
      TML4=0.4*YPD4*(1.0-EXP(-YPLUS4))
      IF (DEL.EQ.O.O) GO TO 1070
      YTERMD=0.0168*ABS(UBLE)*DELS*RO(L,M,N1)
      RDEL=1.0/DEL
      MUTD=YTERMD/(1.0+5.5*(YPD*RDEL)**6)
      TMLD=0.0
      IF (BUY34.EQ.O.O.AND.VXY12.EQ.O.O) GO TO 1120
      TMLD=SQRT(MUTD/(RO(L,M,N1)*SQRT(BUY34*BUY34+VXY12*VXY12)))
      GO TO 1080
 1070 TMLD=0.0
      MUTD=0.0
      GO TO 1120
C
 1080 MUT=TML*TML*RO(L,M,N1)*SQRT(BUY34*BUY34+VXY12*VXY12)
      IF (IMLM.EQ.2.AND.MUTD.LT.MUT) GO TO 1120
      IF (ITM.EQ.2) GO TO 1140
      MUT1=TML*TML*RO(L,M,N1)*SQRT(BUY1*BUY1+VXY1*VXY1)
      MUT2=TML*TML*RO(L,M,N1)*SQRT(BUY2*BUY2+VXY2*VXY2)
      IF (MDFS.EQ.O) GO TO 1090
      IF (L.EQ.LDFSS) MUT1=MUT
      IF (L.EQ.LDFSF) MUT2=MUT
      IF (M.GE.MMIN-1.AND.M.LE.MMIN+1) GO TO 1090
      IF (L.EQ.LDFSS-1) MUT2=MUT
      IF (L.EQ.LDFSF+1) MUT1=MUT
 1090 IF (NOSLIP.EQ.O) GO TO 1110
      IF (M.EQ.1.AND.NGCB.NE.O) GO TO 1100
      IF (M.EQ.MMAX.AND.IWALL.EQ.O) GO TO 1100
      IF (M.EQ.MDFS.AND.LDFS.NE.O) GO TO 1100
      GO TO 1110
 1100 MUT=0.0
```

```
      MUT1=0.0
      MUT2=0.0
 1110 MUT3=TML3*TML3*RO(L,M,N1)*SQRT(BUY3*BUY3+VXY3*VXY3)
      MUT4=TML4*TML4*RO(L,M,N1)*SQRT(BUY4*BUY4+VXY4*VXY4)
      IF (IMLM.EQ.1) GO TO 1130
      GO TO 1170
 1120 TML=TMLD
      IF (ITM.EQ.2) GO TO 1140
      MUT=MUTD
      MUT1=MUTD
      MUT2=MUTD
      MUT3=MUTD
      MUT4=MUTD
      GO TO 1170
 1130 IF (M.NE.1.OR.NGCB.NE.0) GO TO 1170
      MUT=TML*TML*TML*BE*ABS(BUY4-BUY3)*DYR*RO(L,M,N1)
      MUT1=MUT
      MUT2=MUT
      GO TO 1170
C
 1140 TML=CQL*TML
      TINT=TML*RO(L,M,N1)*SQRT(2.0*Q(L,M,N1))/MU
      DELTA=5.0
      IF (TINT.GT.5.0) GO TO 1150
      DCQMU=CQMU*0.1*TML*TML/MU
      MUT=DCQMU*ROSQ*ROSQ
      MUT1=DCQMU*ROSQ1*ROSQ1
      MUT2=DCQMU*ROSQ2*ROSQ2
      MUT3=DCQMU*ROSQ3*ROSQ3
      MUT4=DCQMU*ROSQ4*ROSQ4
      GO TO 1160
 1150 DELTA=TINT
      DCQMU=CQMU*0.3534*TML
      MUT=DCQMU*ROSQ
      MUT1=DCQMU*ROSQ1
      MUT2=DCQMU*ROSQ2
      MUT3=DCQMU*ROSQ3
      MUT4=DCQMU*ROSQ4
C
 1160 MUT1=0.5*(MUT+MUT1)
      MUT2=0.5*(MUT+MUT2)
      MUT3=0.5*(MUT+MUT3)
      MUT4=0.5*(MUT+MUT4)
      IF (ITM.EQ.2) GO TO 1170
C
      RET=RO(L,M,N1)*Q(L,M,N1)*Q(L,M,N1)*LC/(MU*E(L,M,N1))
      RET1=0.5*(RET+ROLM*QLM*QLM*LC/(MU1*ELM))
      RET2=0.5*(RET+ROLP*QLP*QLP*LC/(MU2*ELP))
      RET3=0.5*(RET+ROMM*QMM*QMM*LC/(MU3*ABS(EMM)))
      RET4=0.5*(RET+ROMP*QMP*QMP*LC/(MU4*ABS(EMP)))
      FU=EXP(-3.4/(1.0+0.02*RET)**2)
      FU1=EXP(-3.4/(1.0+0.02*RET1)**2)
      FU2=EXP(-3.4/(1.0+0.02*RET2)**2)
      FU3=EXP(-3.4/(1.0+0.02*RET3)**2)
      FU4=EXP(-3.4/(1.0+0.02*RET4)**2)
      MUT=FU*MUT
      MUT1=FU1*MUT1
      MUT2=FU2*MUT2
      MUT3=FU3*MUT3
      MUT4=FU4*MUT4
      C2T=C2*(1.0-0.2222*EXP(-0.0278*RET*RET))
C
 1170 MUT=0.25*(MUT1+MUT2+MUT3+MUT4)
      IF (MUT1.EQ.0.0.AND.MUT2.EQ.0.0) MUT=0.0
      TLMUR=MUT/MU
      LAT1=LA1*MUT1/MU1
      LAT2=LA2*MUT2/MU2
      LAT3=LA3*MUT3/MU3
      LAT4=LA4*MUT4/MU4
      LAT=0.25*(LAT1+LAT2+LAT3+LAT4)
      IF (MUT.EQ.0.0) LAT=0.0
      KT1=TRK*MUT1
```

```
      KT2=TRK*MUT2
      KT3=TRK*MUT3
      KT4=TRK*MUT4
      KT=0.25*(KT1+KT2+KT3+KT4)
      IF (MUT.EQ.0.0) KT=0.0
      LP2MT=LAT+2.0*MUT
      LP2MT1=LAT1+2.0*MUT1
      LP2MT2=LAT2+2.0*MUT2
      LP2MT3=LAT3+2.0*MUT3
      LP2MT4=LAT4+2.0*MUT4
      LPMT=LAT+MUT
      IF (ITM.NE.1) GO TO 1180
      DLP2MT=LP2MT
      DMUT=MUT
      DLAT=LAT
      SMU1=MUT1*RORR
      SMU2=MUT2*RORR
      SMU3=MUT3*RORR
      SMU4=MUT4*RORR
      GO TO 1190
 1180 SMU1=MUT1*2.0/(RO(L,M,N1)+ROLM)
      SMU2=MUT2*2.0/(RO(L,M,N1)+ROLP)
      SMU3=MUT3*2.0/(RO(L,M,N1)+ROMM)
      SMU4=MUT4*2.0/(RO(L,M,N1)+ROMP)
C
C     DETERMINE THE VISCOUS CONTRIBUTION TO THE TIME STEP CALCULATION
C
 1190 IF (NVC.NE.1.AND.NVC.NE.NVCM+1) GO TO 1250
      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 1250
      IF (M.EQ.1.OR.M.EQ.MMAX) GO TO 1250
      IF (M.EQ.MDFS.AND.LDFS.NE.0) GO TO 1250
      DXP1=XP(L)-XP(L-1)
      DXP2=XP(L+1)-XP(L)
      DYP3=DY/BE3
      DYP4=DY/BE4
      IF (RLA.LE.0.0) GO TO 1200
      RMUD1=RLA1
      RMUD2=RLA2
      RMUD3=RLA3
      RMUD4=RLA4
      GO TO 1210
 1200 RMUD1=RMU1
      RMUD2=RMU2
      RMUD3=RMU3
      RMUD4=RMU4
 1210 TMUTX1=(MU1+RMUD1+MUT1)/(RO(L,M,N1)*DXP1*DXP1)
      TMUTX2=(MU2+RMUD2+MUT2)/(RO(L,M,N1)*DXP2*DXP2)
      TMUTY3=(MU3+RMUD3+MUT3)/(RO(L,M,N1)*DYP3*DYP3)
      TMUTY4=(MU4+RMUD4+MUT4)/(RO(L,M,N1)*DYP4*DYP4)
      TMUTX=AMAX1(TMUTX1,TMUTX2)
      TMUTY=AMAX1(TMUTY3,TMUTY4)
      IF (NVC.NE.1) GO TO 1230
      IF (TMUTX.LE.TMUX) GO TO 1220
      LDUX=L
      MDUX=M
      TMUX=TMUTX
 1220 IF (TMUTY.LE.TMUY) GO TO 1250
      LDUY=L
      MDUY=M
      TMUY=TMU1Y
      GO TO 1250
 1230 IF (TMUTX.LE.TMU1X) GO TO 1240
      LDUX=L
      MDUX=M
      TMU1X=TMUTX
 1240 IF (TMUTY.LE.TMU1Y) GO TO 1250
      LDUY=L
      MDUY=M
      TMU1Y=TMUTY
C
C     CALCULATE THE VISCOSITY AND HEAT CONDUCTION TERMS
C
 1250 UVT=OM*((LP2M2+RLP2M2+LP2MT2)*UXY2-(LP2M1+RLP2M1+LP2MT1)*UXY1+(LA2
```

```
     1 +RLA2+LAT2)*BVY2-(LA1+RLA1+LAT1)*BVY1)*DXR+AL*((LP2M4+RLP2M4
     2 +LP2MT4)*UXY4-(LP2M3+RLP2M3+LP2MT3)*UXY3+(LA4+RLA4+LAT4)*BVY4-
     3 (LA3+RLA3+LAT3)*BVY3)*DYR+BE*((MU4+RMU4+MUT4)*VXY4-(MU3+RMU3+MUT3
     4 )*VXY3+(MU4+RMU4+MUT4)*BUY4-(MU3+RMU3+MUT3)*BUY3)*DYR
      VVT=OM*((MU2+RMU2+MUT2)*(VXY2+BUY2)-(MU1+RMU1+MUT1)*(VXY1+BUY1))
     1 *DXR+AL*((MU4+RMU4+MUT4)*VXY4-(MU3+RMU3+MUT3)*VXY3+(MU4+RMU4+MUT4
     2 )*BUY4-(MU3+RMU3+MUT3)*BUY3)*DYR+BE*((LA4+RLA4+LAT4)*UXY4-(LA3
     3 +RLA3+LAT3)*UXY3+(LP2M4+RLP2M4+LP2MT4)*BVY4-(LP2M3+RLP2M3+LP2MT3)
     4 *BVY3)*DYR
      PVT=(LP2M+RLP2M+DLP2MT)*(UXY12*UXY12+BVY34*BVY34)+(MU+RMU+DMUT)*
     1 (VXY12*VXY12+BUY34*BUY34)+2.0*(LA+RLA+DLAT)*UXY12*BVY34+2.0*(MU
     2 +RMU+DMUT)*BUY34*VXY12
      PCT=OM*((K2+RK2+KT2)*TXY2-(K1+RK1+KT1)*TXY1)*DXR+AL*((K4+RK4+KT4)
     1 *TXY4-(K3+RK3+KT3)*TXY3)*DYR+BE*((K4+RK4+KT4)*BTY4-(K3+RK3+KT3)
     2 *BTY3)*DYR
      IF (ITM.EQ.O.AND.CAV.EQ.O.O) GO TO 1280
      RODIFF=OM*((CAL*SMU2+RORR*RRO2)*ROXY2-(CAL*SMU1+RORR*RRO1)*ROXY1)
     1 *DXR+AL*((CAL*SMU4+RORR*RRO4)*ROXY4-(CAL*SMU3+RORR*RRO3)*ROXY3)
     2 *DYR+BE*((CAL*SMU4+RORR*RRO4)*BROY4-(CAL*SMU3+RORR*RRO3)*BROY3)
     3 *DYR
      IF (ITM.EQ.O) GO TO 1280
      UROT=-0.67*(OM*ROQX+AL*ROQY)+CAL*(U(L,M,N1)*(OM*(SMU2*ROXY2-SMU1
     1 *ROXY1)*DXR+AL*(SMU4*ROXY4-SMU3*ROXY3)*DYR)+BE*V(L,M,N1)*(SMU4
     2 *ROXY4-SMU3*ROXY3)*DYR)
      VROT=-0.67*BE*ROQY+CAL*(V(L,M,N1)*BE*(SMU4*BROY4-SMU3*BROY3)*DYR+U
     1 (L,M,N1)*(OM*(SMU2*BROY2-SMU1*BROY1)*DXR+AL*(SMU4*BROY4-SMU3
     2 *BROY3)*DYR))
      RODUMT=OM*(SMU2*ROXY2-SMU1*ROXY1)*DXR+AL*(SMU4*ROXY4-SMU3*ROXY3)
     1 *DYR+BE*(SMU4*BROY4-SMU3*BROY3)*DYR
      PROT=-CAL*RG*T*RODUMT
      IF (IES.NE.O) GO TO 1280
      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 1290
      IF (ITM.EQ.1) GO TO 1280
      QPROD=LP2MT*(UXY12*UXY12+BVY34*BVY34)+MUT*(VXY12*VXY12+BUY34*BUY34
     1 )+2.0*LAT*UXY12*BVY34+2.0*MUT*BUY34*VXY12
      QDIFF=OM*((MU2+MUT2*SIGQR)*QXY2-(MU1+MUT1*SIGQR)*QXY1)*DXR+AL*(
     1 (MU4+MUT4*SIGQR)*QXY4-(MU3+MUT3*SIGQR)*QXY3)*DYR+BE*((MU4+MUT4
     2 *SIGQR)*BQY4-(MU3+MUT3*SIGQR)*BQY3)*DYR
      QROTT=-XITM*Q(L,M,N1)*RO(L,M,N1)*(UXY12+BVY34)
      IF (ITM.EQ.3) GO TO 1260
      QDISS=0.0
      IF (TML.NE.O.O) QDISS=2.0*MU*DELTA*Q(L,M,N1)/(TML*TML)
      GO TO 1280
 1260 EPROD=0.0
      EDISS=0.0
      IF (Q(L,M,N1).EQ.O.O) GO TO 1270
      EPROD=C1*E(L,M,N1)/Q(L,M,N1)*(LP2MT*(UXY12*UXY12+BVY34*BVY34)+MUT*
     1 (VXY12*VXY12+BUY34*BUY34)+2.0*LAT*UXY12*BVY34+2.0*MUT*BUY34*VXY12
     2 )
      EDISS=C2T*RO(L,M,N1)*E(L,M,N1)*(E(L,M,N1)-2.0*MU*RORR*LC*(Q2XY
     1 +BQ2Y)**2)/(Q(L,M,N1)*LC)
      IF (EDISS.LT.O.O) EDISS=0.0
 1270 EDIFF=OM*((MU2+MUT2*SIGER)*EXY2-(MU1+MUT1*SIGER)*EXY1)*DXR+AL*(
     1 (MU4+MUT4*SIGER)*EXY4-(MU3+MUT3*SIGER)*EXY3)*DYR+BE*((MU4+MUT4
     2 *SIGER)*BEY4-(MU3+MUT3*SIGER)*BEY3)*DYR
      QDISS=RO(L,M,N1)*(E(L,M,N1)+2.0*MU*RORR*LC*(Q2XY+BQ2Y)**2)/LC
      ELOWR=2.0*RORR*MU*MUT*LC*((OM*(UXY2-UXY1)*DXR+AL*(UXY4-UXY3)*DYR)*
     1 *2+(OM*(VXY2-VXY1)*DXR+AL*(VXY4-VXY3)*DYR)**2+(BE*(BUY4-BUY3)*DYR
     2 )**2+(BE*(BVY4-BVY3)*DYR)**2)
C
C     Q AND E FOURTH ORDER SMOOTHING
C
      IF (STBQ.LE.O.O.AND.STBE.LE.O.O) GO TO 1280
      DQX=QLP-2.0*Q(L,M,N1)+QLM
      DQY=QMP-2.0*Q(L,M,N1)+QMM
      DEX=ELP-2.0*E(L,M,N1)+ELM
      DEY=EMP-2.0*E(L,M,N1)+EMM
      QAVGX=0.25*(QLP+2.0*Q(L,M,N1)+QLM)
      QAVGY=0.25*(QMP+2.0*Q(L,M,N1)+QMM)
      IF (QAVGX.LE.O.O) QAVGX=1.0E+10
      IF (QAVGY.LE.O.O) QAVGY=1.0E+10
      EAVGX=0.25*(ELP+2.0*E(L,M,N1)+ELM)
```

```
      EAVGY=0.25*(EMP+2.0*E(L,M,N1)+EMM)
      AST=SQRT(AL*AL+BE*BE)
      QSMO=STBQ*RO(L,M,N1)*((ABS(U(L,M,N1))+A)*ABS(DQX)*OM*DXR*DQX/QAVGX
     1 +(ABS(U(L,M,N1)*AL+V(L,M,N1)*BE)+AST*A)*ABS(DQY)*DYR*DQY/QAVGY)
      ESMO=STBE*RO(L,M,N1)*((ABS(U(L,M,N1))+A)*ABS(DEX)*OM*DXR*DEX/EAVGX
     1 +(ABS(U(L,M,N1)*AL+V(L,M,N1)*BE)+AST*A)*ABS(DEY)*DYR*DEY/EAVGY)
C
C     PRINT THE TURBULENCE MODEL CONV, PROD, DISS, AND
C     DIFF TERMS FOR THE REQUESTED GRID POINT
C
 1280 IF (ITM.LE.1) GO TO 1290
      IF (L.NE.LPRINT.OR.M.NE.MPRINT) GO TO 1290
      IF (NVC.GT.2) GO TO 1290
      IF (M.EQ.1.OR.M.EQ.MMAX) GO TO 1290
      IF (M.EQ.MDFS.AND.LDFS.NE.0) GO TO 1290
      IF (N.EQ.1) WRITE (6,1470)
      UVB=U(L,M,N1)*AL+V(L,M,N1)*BE
      QCON=-(U(L,M,N1)*OM*(Q(L+1,M,N1)-Q(L-1,M,N1))*DXR+UVB*(Q(L,M+1,N1)
     1 -Q(L,M-1,N1))*DYR)*0.5*DT
      ECON=-(U(L,M,N1)*OM*(E(L+1,M,N1)-E(L-1,M,N1))*DXR+UVB*(E(L,M+1,N1)
     1 -E(L,M-1,N1))*DYR)*0.5*DT
      QPRO=QPROD*DT*RORR
      QDIS=-QDISS*DT*RORR
      QDIF=QDIFF*DT*RORR
      EPRO=EPROD*DT*RORR
      EDIS=-EDISS*DT*RORR
      EDIF=EDIFF*DT*RORR
      ELOR=ELOWR*DT*RORR
      NP=N+NSTART
      WRITE (6,1480) NP,L,M,Q(L,M,N1),QCON,QPRO,QDIS,QDIF,E(L,M,N1),ECON
     1 ,EPRO,EDIS,EDIF,ELOR
 1290 IF (NDIM.EQ.0) GO TO 1330
C
C     CALCULATE THE AXISYMMETRIC TERMS
C
      IF (M.EQ.1.AND.YCB(L).EQ.0.0) GO TO 1310
      VB=V(L,M,N1)
      UVTA=((LPM+RLPM+LPMT)*VXY12+(MU+RMU+MUT)*BUY34)/YP
      VVTA=(LP2M+RLP2M+LP2MT)*(BVY34-VB/YP)/YP
      PVTA=((LP2M+RLP2M+DLP2MT)*VB*VB/YP+2.0*(LA+RLA+DLAT)*VB*(BVY34
     1 +UXY12))/YP
      PCTA=(K+RK+KT)*0.5*(BTY4+BTY3)/YP
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 1330
      RODIFFA=(CAL*MUT+RRO)*RORR*BROY34/YP
      IF (ITM.EQ.0) GO TO 1330
      UROTA=CAL*MUT*RORR*V(L,M,N1)*ROXY12/YP
      VROTA=CAL*MUT*RORR*V(L,M,N1)*BROY34/YP
      PROTA=-CAL*RG*T*MUT*RORR*BROY34/YP
      IF (IES.NE.0) GO TO 1330
      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 1330
      IF (ITM.EQ.1) GO TO 1330
      QPRODA=(LP2MT*VB*VB/YP+2.0*LAT*VB*(BVY34+UXY12))/YP
      QDIFFA=(MU+MUT*SIGQR)*BQY34/YP
      QROTTA=-XITM*Q(L,M,N1)*RO(L,M,N1)*VB/YP
      IF (ITM.EQ.2) GO TO 1330
      IF (Q(L,M,N1).EQ.0.0) GO TO 1300
      EPRODA=C1*E(L,M,N1)*(LP2MT*VB*VB/YP+2.0*LAT*VB*(BVY34+UXY12))/(Q(L
     1 ,M,N1)*YP)
 1300 EDIFFA=(MU+MUT*SIGER)*BEY34/YP
      ELOWRA=2.0*RORR*MU*MUT*LC*((BUY34/YP)**2+(BVY34/YP)**2+2.0*BUY34
     1 *BE*(BUY4-BUY3)*DYR/YP+2.0*BVY34*BE*(BVY4-BVY3)*DYR/YP)
      GO TO 1330
C
C     CALCULATE THE AXISYMMETRIC TERMS ON THE AXIS
C
 1310 UVTA=(LPM+RLPM+LPMT)*BE*(VXY4-VXY3)*DYR+(MU+RMU+MUT)*BE*(BUY4-BUY3
     1 )*DYR
      VVTA=(LP2M+RLP2M+LP2MT)*0.5*BE*(BVY4-BVY3)*DYR
      PVTA=(LP2M+RLP2M+DLP2MT+2.0*(LA+RLA+DLAT))*BVY34*BVY34+2.0*(LA+RLA
     1 +DLAT)*BVY34*UXY12
      PCTA=(K+RK+KT)*BE*(BTY4-BTY3)*DYR
      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 1330
```

```
      RODIFFA=(CAL*MUT+RRO)*RORR*BE*(BROY4-BROY3)*DYR
      IF (ITM.EQ.O) GO TO 1330
      UROTA=CAL*MUT*RORR*BVY34*ROXY12
      VROTA=0.0
      PROTA=-CAL*RG*T*MUT*RORR*BE*(BROY4-BROY3)*DYR
      IF (IES.NE.O) GO TO 1330
      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 1330
      IF (ITM.EQ.1) GO TO 1330
      QPRODA=(LP2MT+2.0*LAT)*BVY34*BVY34+2.0*LAT*BVY34*UXY12
      QDIFFA=(MU+MUT*SIGQR)*BE*(BQY4-BQY3)*DYR
      QROTTA=-XITM*Q(L,M,N1)*RO(L,M,N1)*BVY34
      IF (ITM.EQ.2) GO TO 1330
      IF (Q(L,M,N1).EQ.0.0) GO TO 1320
      EPRODA=C1*E(L,M,N1)*((LP2MT+2.0*LAT)*BVY34*BVY34+2.0*LAT*BVY34
     1 *UXY12)/Q(L,M,N1)
 1320 EDIFFA=(MU+MUT*SIGER)*BE*(BEY4-BEY3)*DYR
      ELOWRA=6.0*RORR*MU*MUT*LC*((BE*(BUY4-BUY3)*DYR)**2+(BE*(BVY4-BVY3)
     1 *DYR)**2)
C
C     FILL THE VISCOUS TERM ARRAYS
C
 1330 QUT(L,M)=(UVT+UVTA+UROT+UROTA)*RORR
      QVT(L,M)=(VVT+VVTA+VROT+VROTA)*RORR
      QPT(L,M)=GAM*(PVT+PVTA+PCT+PCTA+PROT+PROTA+QDISS)
      IF (ITM.EQ.O.AND.CAV.EQ.0.0) GO TO 1340
      QROT(L,M)=RODIFF+RODIFFA
      IF (IES.NE.O) GO TO 1340
      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 1340
      IF (ITM.LE.1) GO TO 1340
      QQT(L,M)=(QPROD+QPRODA+QDIFF+QDIFFA+QROTT+QROTTA-QDISS+QSMO)*RORR
      QET(L,M)=(EPROD+EPRODA+EDIFF+EDIFFA-EDISS+ELOWR+ELOWRA+ESMO)*RORR
C
C     PRINT THE VISCOUS TERMS
C
 1340 IF (IAV.EQ.O) GO TO 1400
      IF (NC.NE.NPRINT.AND.(N.NE.NMAX.AND.ISTOP.EQ.O)) GO TO 1400
      IF (IAV.EQ.2) GO TO 1350
      IF (NVC.GT.2.AND.NVC.NE.NVCM+1) GO TO 1400
 1350 IF (L.EQ.1.AND.(NVC.EQ.1.AND.IB.NE.4)) GO TO 1370
      IF (L.EQ.1.AND.MDFS.EQ.O) GO TO 1370
      IF (L.EQ.1.AND.IB.EQ.3) GO TO 1370
      IF (M.EQ.MIS) GO TO 1360
      IF (M.EQ.MVCT+1.AND.(MDFS.NE.O.AND.MDFSC.EQ.O)) GO TO 1360
      IF (M.EQ.MVCT+1.AND.MVCB.EQ.1) GO TO 1360
      GO TO 1370
 1360 WRITE (6,1490)
      NLINE=NLINE+1
 1370 NLINE=NLINE+1
      IF (NLINE.LT.54) GO TO 1380
      WRITE (6,1460)
      NP=N+NSTART
      WRITE (6,1450) NP,NVC
      NLINE=1
 1380 DQPT=QPT(L,M)/PC*DT
      DQUT=QUT(L,M)*DT
      DQVT=QVT(L,M)*DT
      DQROT=QROT(L,M)*G*DT
      DQ=Q(L,M,N1)
      DDE=E(L,M,N1)
      DQQT=QQT(L,M)*DT
      DQET=QET(L,M)*DT
      DTML=TML
      IF (IUO.NE.2) GO TO 1390
      DQUT=DQUT*0.3048
      DQVT=DQVT*0.3048
      DQPT=DQPT*6.8948
      DQROT=DQROT*16.02
      DQ=DQ*0.0929
      DDE=DDE*0.0929
      DQQT=DQQT*0.0929
      DQET=DQET*0.0929
      DTML=DTML*2.54
 1390 WRITE (6,1440) L,M,DQUT,DQVT,DQPT,DQROT,AVMUR,TLMUR,DQ,DDE,DQQT
```

```
     1 ,DQET,DTML
 1400 CONTINUE
 1410 CONTINUE
      IF (MDFS.EQ.0) GO TO 1420
      IF (NVC.EQ.1.AND.MDFSC.NE.0) GO TO 1420
      IF (MIS.EQ.1.AND.MIF.EQ.MDFS) GO TO 50
      IF (MIS.EQ.MVCB.AND.MIF.EQ.MDFS) GO TO 50
      IF (MIS.EQ.MDFS+1.AND.IVC.EQ.0) GO TO 60
      IF (MIS.EQ.MDFS+1.AND.NVC.NE.1) GO TO 60
 1420 IF (IAV.EQ.0) RETURN
      IF (NC.NE.NPRINT.AND.N.NE.NMAX) RETURN
      IF (IAV.EQ.2) GO TO 1430
      IF (NVC.NE.1.AND.NVC.NE.NVCM+1) RETURN
 1430 IF (TMUX.NE.0.0) RDUM=TMUY/TMUX
      IF (NVC.NE.1.AND.TMU1X.NE.0.0) RDUM=TMU1Y/TMU1X
      WRITE (6,1500) LDUX,MDUX,LDUY,MDUY,RDUM,NVC
      RETURN
C
C     FORMAT STATEMENTS
C
 1440 FORMAT (1H ,2I5,2F11.4,F11.5,F11.6,2F11.3,F12.4,E10.3,F11.4,E10.3
     1 ,F11.6)
 1450 FORMAT (1H ,51HLOCAL VISCOSITY (ARTIFICAL-MOLECULAR-TURBULENT) AND
     1 ,26H HEAT CONDUCTION TERMS, N=,I6,6H, NVC=,I3//5X,1HL,4X,1HM,7X,3
     2 HQUT,8X,3HQVT,8X,3HQPT,7X,4HQROT,7X,5HAVMUR,6X,5HTLMUR,8X,1HQ,9X,
     3 1HE,10X,3HQQT,6X,3HQET,8X,3HTML,/)
 1460 FORMAT (1H1)
 1470 FORMAT (1H1,3X,1HN,3X,1HL,3X,1HM,5X,1HQ,8X,4HQCON,6X,4HQPRO,6X,4HQ
     1DIS,6X,4HQDIF,7X,1HE,8X,4HECON,6X,4HEPRO,6X,4HEDIS,6X,4HEDIF,6X,4H
     2ELOR,/)
 1480 FORMAT (1H ,3I4,11E10.3)
 1490 FORMAT (1H ,3X,48H------------------------------------------------
     1 ,61H-------------------------------------------------------------
     2 ,18H------------------)
 1500 FORMAT (1H0,10X,20HX TERMS GRID POINT=(,I2,1H,,I2,25H),    Y TERMS
     1GRID POINT=(,I2,1H,,I2,22H),    RATIO OF Y TO X=(,E9.3,9H),    NVC=
     2 ,I3,/)
 1510 FORMAT (1H0,109H***** THE TEMPERATURE USED IN THE MOLECULAR VISCOS
     1ITY CALCULATION IN SUBROUTINE VISCOUS BECAME NEGATIVE AT N=,I6,1H,
     2 ,/,7X,2HL=,I2,4H, M=,I2,6H, NVC=,I3,6H *****)
      END
```

```
      SUBROUTINE SMOOTH
C
C     ******************************************************************
C
C     THIS SUBROUTINE SMOOTHS THE FLOW VARIABLES IF REQUESTED
C
C     ******************************************************************
C
*CALL,MCC
C
C     SPACE SMOOTHING
C
      IF (SMP.EQ.1.0) GO TO 100
      SMP4=0.25*(1.0-SMP)
      IF (MDFS.EQ.0) GO TO 20
      IF (LDFSS.EQ.1.AND.LDFSF.EQ.LMAX) GO TO 20
      IF (LDFSS.EQ.1) GO TO 10
      UL(LDFSS-1,N3)=U(LDFSS-1,MDFS,N3)
      VL(LDFSS-1,N3)=V(LDFSS-1,MDFS,N3)
      PL(LDFSS-1,N3)=P(LDFSS-1,MDFS,N3)
      ROL(LDFSS-1,N3)=RO(LDFSS-1,MDFS,N3)
      QL(LDFSS-1,N3)=Q(LDFSS-1,MDFS,N3)
      EL(LDFSS-1,N3)=E(LDFSS-1,MDFS,N3)
   10 IF (LDFSF.EQ.LMAX) GO TO 20
      UL(LDFSF+1,N3)=U(LDFSF+1,MDFS,N3)
      VL(LDFSF+1,N3)=V(LDFSF+1,MDFS,N3)
      PL(LDFSF+1,N3)=P(LDFSF+1,MDFS,N3)
      ROL(LDFSF+1,N3)=RO(LDFSF+1,MDFS,N3)
      QL(LDFSF+1,N3)=Q(LDFSF+1,MDFS,N3)
      EL(LDFSF+1,N3)=E(LDFSF+1,MDFS,N3)
C
   20 DO 90 L=2,L1
      IF (IWALL.NE.0.AND.V(L,MMAX,N1).LT.0.0) GO TO 40
      U(L,MMAX,N3)=SMP4*(U(L-1,MMAX,N3)+U(L+1,MMAX,N3)+2.0*U(L,MMAX,N3))
     1 +SMP*U(L,MMAX,N3)
      IF (NOSLIP.NE.0.AND.IWALL.EQ.0) U(L,MMAX,N3)=0.0
      IF (IWALL.EQ.0) V(L,MMAX,N3)=-U(L,MMAX,N3)*NXNY(L)+XWI(L)
      IF (IWALL.NE.0) GO TO 30
      IF (JFLAG.EQ.1.AND.L.GE.LJET) GO TO 30
      P(L,MMAX,N3)=SMP4*(P(L-1,MMAX,N3)+P(L+1,MMAX,N3)+2.0*P(L,MMAX,N3))
     1 +SMP*P(L,MMAX,N3)
   30 RO(L,MMAX,N3)=SMP4*(RO(L-1,MMAX,N3)+RO(L+1,MMAX,N3)+2.0*RO(L,MMAX
     1 ,N3))+SMP*RO(L,MMAX,N3)
      IF (TW(1).GE.0.0) P(L,MMAX,N3)=RO(L,MMAX,N3)*RG*TW(L)
   40 U(L,1,N3)=SMP4*(U(L-1,1,N3)+U(L+1,1,N3)+2.0*U(L,1,N3))+SMP*U(L,1
     1 ,N3)
      IF (NOSLIP.NE.0.AND.NGCB.NE.0) U(L,1,N3)=0.0
      V(L,1,N3)=-U(L,1,N3)*NXNYCB(L)
      P(L,1,N3)=SMP4*(P(L-1,1,N3)+P(L+1,1,N3)+2.0*P(L,1,N3))+SMP*P(L,1
     1 ,N3)
      RO(L,1,N3)=SMP4*(RO(L-1,1,N3)+RO(L+1,1,N3)+2.0*RO(L,1,N3))+SMP*RO
     1 (L,1,N3)
      IF (TCB(1).GE.0.0.AND.NGCB.NE.0) P(L,1,N3)=RO(L,1,N3)*RG*TCB(L)
      IF (ITM.LE.1) GO TO 50
      Q(L,MMAX,N3)=SMP4*(Q(L-1,MMAX,N3)+Q(L+1,MMAX,N3)+2.0*Q(L,MMAX,N3))
     1 +SMP*Q(L,MMAX,N3)
      E(L,MMAX,N3)=SMP4*(E(L-1,MMAX,N3)+E(L+1,MMAX,N3)+2.0*E(L,MMAX,N3))
     1 +SMP*E(L,MMAX,N3)
      Q(L,1,N3)=SMP4*(Q(L-1,1,N3)+Q(L+1,1,N3)+2.0*Q(L,1,N3))+SMP*Q(L,1
     1 ,N3)
      E(L,1,N3)=SMP4*(E(L-1,1,N3)+E(L+1,1,N3)+2.0*E(L,1,N3))+SMP*E(L,1
     1 ,N3)
   50 LDFS=0
      IF (MDFS.EQ.0) GO TO 60
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      IF (LDFS.EQ.0) GO TO 60
      UL(L,N3)=SMP4*(UL(L-1,N3)+UL(L+1,N3)+2.0*U(L,MDFS-1,N3))+SMP*UL(L
     1 ,N3)
      IF (NOSLIP.NE.0) UL(L,N3)=0.0
      VL(L,N3)=-UL(L,N3)*NXNYL(L)
      PL(L,N3)=SMP4*(PL(L-1,N3)+PL(L+1,N3)+2.0*P(L,MDFS-1,N3))+SMP*PL(L
     1 ,N3)
```

```
      ROL(L,N3)=SMP4*(ROL(L-1,N3)+ROL(L+1,N3)+2.0*RO(L,MDFS-1,N3))+SMP
     1 *ROL(L,N3)
      IF (TL(1).GE.0.0) PL(L,N3)=ROL(L,N3)*RG*TL(L)
      U(L,MDFS,N3)=SMP4*(U(L-1,MDFS,N3)+U(L+1,MDFS,N3)+2.0*U(L,MDFS+1,N3
     1 ))+SMP*U(L,MDFS,N3)
      IF (NOSLIP.NE.0) U(L,MDFS,N3)=0.0
      V(L,MDFS,N3)=-U(L,MDFS,N3)*NXNYU(L)
      P(L,MDFS,N3)=SMP4*(P(L-1,MDFS,N3)+P(L+1,MDFS,N3)+2.0*P(L,MDFS+1,N3
     1 ))+SMP*P(L,MDFS,N3)
      RO(L,MDFS,N3)=SMP4*(RO(L-1,MDFS,N3)+RO(L+1,MDFS,N3)+2.0*RO(L,MDFS+
     1 1,N3))+SMP*RO(L,MDFS,N3)
      IF (TU(1).GE.0.0) P(L,MDFS,N3)=RO(L,MDFS,N3)*RG*TU(L)
      IF (ITM.LE.1) GO TO 60
      QL(L,N3)=SMP4*(QL(L-1,N3)+QL(L+1,N3)+2.0*Q(L,MDFS-1,N3))+SMP*QL(L
     1 ,N3)
      EL(L,N3)=SMP4*(EL(L-1,N3)+EL(L+1,N3)+2.0*E(L,MDFS-1,N3))+SMP*EL(L
     1 ,N3)
      Q(L,MDFS,N3)=SMP4*(Q(L-1,MDFS,N3)+Q(L+1,MDFS,N3)+2.0*Q(L,MDFS+1,N3
     1 ))+SMP*Q(L,MDFS,N3)
      E(L,MDFS,N3)=SMP4*(E(L-1,MDFS,N3)+E(L+1,MDFS,N3)+2.0*E(L,MDFS+1,N3
     1 ))+SMP*E(L,MDFS,N3)
C
   60 DO 90 M=2,M1
      IF (M.EQ.MDFS.AND.LDFS.EQ.1) GO TO 90
      IF (M.NE.MDFS) GO TO 80
      IF (L.NE.LDFSS-1.AND.L.NE.LDFSF+1) GO TO 80
      IF (L.NE.LDFSS-1) GO TO 70
      U(L,M,N3)=SMP4*(U(L-1,M,N3)+U(L,M-1,N3)+U(L,M+1,N3)+0.5*(U(L+1,M
     1 ,N3)+UL(L+1,N3)))+SMP*U(L,M,N3)
      V(L,M,N3)=SMP4*(V(L-1,M,N3)+V(L,M-1,N3)+V(L,M+1,N3)+0.5*(V(L+1,M
     1 ,N3)+VL(L+1,N3)))+SMP*V(L,M,N3)
      P(L,M,N3)=SMP4*(P(L-1,M,N3)+P(L,M-1,N3)+P(L,M+1,N3)+0.5*(P(L+1,M
     1 ,N3)+PL(L+1,N3)))+SMP*P(L,M,N3)
      RO(L,M,N3)=SMP4*(RO(L-1,M,N3)+RO(L,M-1,N3)+RO(L,M+1,N3)+0.5*(RO(L+
     1 1,M,N3)+ROL(L+1,N3)))+SMP*RO(L,M,N3)
      IF (ITM.LE.1) GO TO 90
      Q(L,M,N3)=SMP4*(Q(L-1,M,N3)+Q(L,M-1,N3)+Q(L,M+1,N3)+0.5*(Q(L+1,M
     1 ,N3)+QL(L+1,N3)))+SMP*Q(L,M,N3)
      E(L,M,N3)=SMP4*(E(L-1,M,N3)+E(L,M-1,N3)+E(L,M+1,N3)+0.5*(E(L+1,M
     1 ,N3)+EL(L+1,N3)))+SMP*E(L,M,N3)
      GO TO 90
   70 U(L,M,N3)=SMP4*(U(L+1,M,N3)+U(L,M-1,N3)+U(L,M+1,N3)+0.5*(U(L-1,M
     1 ,N3)+UL(L-1,N3)))+SMP*U(L,M,N3)
      V(L,M,N3)=SMP4*(V(L+1,M,N3)+V(L,M-1,N3)+V(L,M+1,N3)+0.5*(V(L-1,M
     1 ,N3)+VL(L-1,N3)))+SMP*V(L,M,N3)
      P(L,M,N3)=SMP4*(P(L+1,M,N3)+P(L,M-1,N3)+P(L,M+1,N3)+0.5*(P(L-1,M
     1 ,N3)+PL(L-1,N3)))+SMP*P(L,M,N3)
      RO(L,M,N3)=SMP4*(RO(L+1,M,N3)+RO(L,M-1,N3)+RO(L,M+1,N3)+0.5*(RO(L-
     1 1,M,N3)+ROL(L-1,N3)))+SMP*RO(L,M,N3)
      IF (ITM.LE.1) GO TO 90
      Q(L,M,N3)=SMP4*(Q(L+1,M,N3)+Q(L,M-1,N3)+Q(L,M+1,N3)+0.5*(Q(L-1,M
     1 ,N3)+QL(L-1,N3)))+SMP*Q(L,M,N3)
      E(L,M,N3)=SMP4*(E(L+1,M,N3)+E(L,M-1,N3)+E(L,M+1,N3)+0.5*(E(L-1,M
     1 ,N3)+EL(L-1,N3)))+SMP*E(L,M,N3)
      GO TO 90
   80 U(L,M,N3)=SMP4*(U(L-1,M,N3)+U(L+1,M,N3)+U(L,M-1,N3)+U(L,M+1,N3))
     1 +SMP*U(L,M,N3)
      V(L,M,N3)=SMP4*(V(L-1,M,N3)+V(L+1,M,N3)+V(L,M-1,N3)+V(L,M+1,N3))
     1 +SMP*V(L,M,N3)
      P(L,M,N3)=SMP4*(P(L-1,M,N3)+P(L+1,M,N3)+P(L,M-1,N3)+P(L,M+1,N3))
     1 +SMP*P(L,M,N3)
      RO(L,M,N3)=SMP4*(RO(L-1,M,N3)+RO(L+1,M,N3)+RO(L,M-1,N3)+RO(L,M+1
     1 ,N3))+SMP*RO(L,M,N3)
      IF (ITM.LE.1) GO TO 90
      Q(L,M,N3)=SMP4*(Q(L-1,M,N3)+Q(L+1,M,N3)+Q(L,M-1,N3)+Q(L,M+1,N3))
     1 +SMP*Q(L,M,N3)
      E(L,M,N3)=SMP4*(E(L-1,M,N3)+E(L+1,M,N3)+E(L,M-1,N3)+E(L,M+1,N3))
     1 +SMP*E(L,M,N3)
   90 CONTINUE
C
C     TIME SMOOTHING (NTST.EQ.1)
```

```
C
  100 IF (SMPT.EQ.1.0) RETURN
      IF (NTST.EQ.-1) GO TO 130
      NTC=NTC+1
      IF (NTC.NE.NTST) RETURN
      NTC=0
      IF (NTST.NE.1) GO TO 130
C
      DO 120 L=1,LMAX
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
C
      DO 120 M=1,MMAX
      U(L,M,N3)=SMPT*U(L,M,N3)+(1.0-SMPT)*U(L,M,N1)
      V(L,M,N3)=SMPT*V(L,M,N3)+(1.0-SMPT)*V(L,M,N1)
      P(L,M,N3)=SMPT*P(L,M,N3)+(1.0-SMPT)*P(L,M,N1)
      RO(L,M,N3)=SMPT*RO(L,M,N3)+(1.0-SMPT)*RO(L,M,N1)
      IF (ITM.LE.1) GO TO 110
      Q(L,M,N3)=SMPT*Q(L,M,N3)+(1.0-SMPT)*Q(L,M,N1)
      E(L,M,N3)=SMPT*E(L,M,N3)+(1.0-SMPT)*E(L,M,N1)
  110 IF (MDFS.EQ.0.OR.LDFS.EQ.0) GO TO 120
      UL(L,N3)=SMPT*UL(L,N3)+(1.0-SMPT)*UL(L,N1)
      VL(L,N3)=SMPT*VL(L,N3)+(1.0-SMPT)*VL(L,N1)
      PL(L,N3)=SMPT*PL(L,N3)+(1.0-SMPT)*PL(L,N1)
      ROL(L,N3)=SMPT*ROL(L,N3)+(1.0-SMPT)*ROL(L,N1)
      IF (ITM.LE.1) GO TO 120
      QL(L,N3)=SMPT*QL(L,N3)+(1.0-SMPT)*QL(L,N1)
      EL(L,N3)=SMPT*EL(L,N3)+(1.0-SMPT)*EL(L,N1)
  120 CONTINUE
      RETURN
C
C     TIME SMOOTHING (NTST.GT.1)
C
  130 DO 150 L=1,LMAX
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
C
      DO 150 M=1,MMAX
      U(L,M,N3)=SMPT*U(L,M,N3)+(1.0-SMPT)*US(L,M)
      V(L,M,N3)=SMPT*V(L,M,N3)+(1.0-SMPT)*VS(L,M)
      P(L,M,N3)=SMPT*P(L,M,N3)+(1.0-SMPT)*PS(L,M)
      RO(L,M,N3)=SMPT*RO(L,M,N3)+(1.0-SMPT)*ROS(L,M)
      US(L,M)=U(L,M,N3)
      VS(L,M)=V(L,M,N3)
      PS(L,M)=P(L,M,N3)
      ROS(L,M)=RO(L,M,N3)
      IF (ITM.LE.1) GO TO 140
      Q(L,M,N3)=SMPT*Q(L,M,N3)+(1.0-SMPT)*QS(L,M)
      E(L,M,N3)=SMPT*E(L,M,N3)+(1.0-SMPT)*ES(L,M)
      QS(L,M)=Q(L,M,N3)
      ES(L,M)=E(L,M,N3)
  140 IF (MDFS.EQ.0.OR.LDFS.EQ.0) GO TO 150
      UL(L,N3)=SMPT*UL(L,N3)+(1.0-SMPT)*ULS(L)
      VL(L,N3)=SMPT*VL(L,N3)+(1.0-SMPT)*VLS(L)
      PL(L,N3)=SMPT*PL(L,N3)+(1.0-SMPT)*PLS(L)
      ROL(L,N3)=SMPT*ROL(L,N3)+(1.0-SMPT)*ROLS(L)
      ULS(L)=UL(L,N3)
      VLS(L)=VL(L,N3)
      PLS(L)=PL(L,N3)
      ROLS(L)=ROL(L,N3)
      IF (ITM.LE.1) GO TO 150
      QL(L,N3)=SMPT*QL(L,N3)+(1.0-SMPT)*QLS(L)
      EL(L,N3)=SMPT*EL(L,N3)+(1.0-SMPT)*ELS(L)
      QLS(L)=QL(L,N3)
      ELS(L)=EL(L,N3)
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MIXLEN (L,MV)
C
C     ************************************************************
C
C     THIS SUBROUTINE CALCULATES THE SHEAR LAYER WIDTH , BOUNDARY
C     LAYER THICKNESS AND DISPLACEMENT THICKNESS FOR THE MIXING-LENGTH
C     MODEL (ITM=1) AND ONE EQUATION MODEL (ITM=2)
C
C     ************************************************************
C
*CALL,MCC
C
C     CALCULATE THE SHEAR LAYER WIDTH (YSL2-YSL1)
C
      IP=0
      LMAP=L
      IMP=0
      IF (IMLM.EQ.2) GO TO 120
      UMIN=U(L,1,N1)
      DO 10 M=1,MMAX
      IF (U(L,M,N1).GT.UMIN) GO TO 10
      UMIN=U(L,M,N1)
      MMIN=M
   10 CONTINUE
      IF (MMIN.EQ.1.OR.MMIN.EQ.MMAX) IMP=1
      IF (U(L,1,N1).EQ.U(L,MMAX,N1)) GO TO 20
      IF (U(L,MMAX,N1).GT.U(L,1,N1)) UCHECK=(U(L,1,N1)-UMIN)/(U(L,MMAX
     1 ,N1)-U(L,1,N1))
      IF (U(L,MMAX,N1).LT.U(L,1,N1)) UCHECK=(U(L,MMAX,N1)-UMIN)/(U(L,1
     1 ,N1)-U(L,MMAX,N1))
      IF (UCHECK.LT.0.05) IMP=1
   20 IF (IMP.NE.0) GO TO 30
      UDUM=UMIN
      RDUL=1.0/(U(L,1,N1)-UDUM)
      RDUU=1.0/(U(L,MMAX,N1)-UDUM)
      GO TO 40
C
   30 IF (U(L,1,N1).EQ.U(L,MMAX,N1)) GO TO 110
      UDUM=U(L,MMAX,N1)
      RDU=1.0/(U(L,1,N1)-UDUM)
C
   40 DO 90 M=1,M1
      MMAP=M
      CALL MAP
      IF (M.EQ.MMIN) YMIN=YP
      MMAP=M+1
      YP1=YP
      CALL MAP
      DYP=YP-YP1
      IF (IMP.NE.0) GO TO 50
      RDU=RDUL
      IF (M.GE.MMIN) RDU=RDUU
   50 UD1=(U(L,M,N1)-UDUM)*RDU
      UD2=(U(L,M+1,N1)-UDUM)*RDU
      IF (UD1.GE.0.9.AND.UD2.LE.0.9) GO TO 60
      IF (UD1.LE.0.9.AND.UD2.GE.0.9) GO TO 60
      IF (IMP.EQ.0) GO TO 90
      IF (UD1.GE.0.1.AND.UD2.LE.0.1) GO TO 80
      GO TO 90
   60 YSL2=YP1+(0.9-UD1)*DYP/(UD2-UD1)
      IF (IMP.NE.0) GO TO 70
      IF (M.GE.MMIN) GO TO 100
      IF (M.LT.MMIN) YSL1=YSL2
      GO TO 90
   70 IF (UD1.GE.0.1.AND.UD2.LE.0.1) GO TO 80
      GO TO 90
   80 YSL1=YP1+(0.1-UD1)*DYP/(UD2-UD1)
      GO TO 100
   90 CONTINUE
      YSL1=YW(L)
  100 IP=1
      RETURN
```

```
C
  110 YSL1=0.0
      YSL2=0.0
      YMIN=0.0
      IP=1
      RETURN
C
C     CALCULATE THE BOUNDARY LAYER THICKNESS (DEL)
C
  120 MM3=MMAX
      MM4=0
      IF (MDFS.EQ.0) GO TO 150
      IF (NVC.NE.1) GO TO 130
      IBD=IB
      IF (MV.LT.MDFS) IB=3
      IF (MV.GT.MDFS) IB=4
  130 IF (IB.EQ.4) GO TO 140
      MM3=MDFS
      M=MM3+1
      MDEL=-1
      UMAX=U(L,1,N1)*RO(L,1,N1)
      GO TO 170
  140 MM4=MDFS-1
      M=MM4
      MDEL=1
      UMAX=U(L,MMAX,N1)*RO(L,MMAX,N1)
      GO TO 170
C
  150 IF (IWALL.EQ.0) GO TO 160
      M=MM4
      MDEL=1
      UMAX=U(L,MMAX,N1)*RO(L,MMAX,N1)
      GO TO 170
  160 M=MM3+1
      MDEL=-1
      UMAX=U(L,1,N1)*RO(L,1,N1)
C
  170 DO 180 MM=1,M1
      M=M+MDEL
      IF (M+MDEL.EQ.0) GO TO 190
      IF (M+MDEL.EQ.MMAX+1) GO TO 190
      UD1=U(L,M,N1)*RO(L,M,N1)/UMAX
      UD2=U(L,M+MDEL,N1)*RO(L,M+MDEL,N1)/UMAX
      IF (UD1.LE.0.98.AND.UD2.GE.0.98) GO TO 200
      IF (UD1.GE.0.98.AND.UD2.LE.0.98) GO TO 200
  180 CONTINUE
  190 DEL=0.0
      RETURN
  200 MMAP=M
      CALL MAP
      MMAP=M+MDEL
      YP1=YP
      CALL MAP
      DYP=YP-YP1
      Y2=YP1+(0.98-UD1)*DYP/(UD2-UD1)
      IF (MDFS.EQ.0) GO TO 210
      IF (IB.EQ.3) DEL=YL(L)-Y2
      IF (IB.EQ.4) DEL=Y2-YU(L)
      GO TO 220
  210 IF (IWALL.EQ.0) DEL=YW(L)-Y2
      IF (IWALL.NE.0) DEL=Y2-YCB(L)
C
C     CALCULATE THE DISPLACEMENT THICKNESS (DELS)
C
  220 DELS=0.0
      IF (IWALL.EQ.0) GO TO 230
      IF (MDFS.NE.0.AND.IB.EQ.3) GO TO 230
      MBLE=M+1-MM4
      UBLE=U(L,M+1,N1)
      ROUBLE=UBLE*RO(L,M+1,N1)
      M=MM4
      MDEL=1
```

```
      GO TO 240
C
  230 MBLE=MM3-M+2
      UBLE=U(L,M-1,N1)
      ROUBLE=UBLE*RO(L,M-1,N1)
      M=MM3+1
      MDEL=-1
C
  240 MBLE1=MBLE-1
      DO 250 MM=1,MBLE1
      M=M+MDEL
      MMAP=M
      CALL MAP
      MMAP=M+MDEL
      YP1=YP
      CALL MAP
      DYP=ABS(YP-YP1)
      DELS=DELS+(1.0-0.5*(U(L,M,N1)*RO(L,M,N1)+U(L,M+MDEL,N1)*RO(L,M
     1 +MDEL,N1))/ROUBLE)*DYP
  250 CONTINUE
      IF (MDFS.NE.O.AND.NVC.EQ.1) IB=IBD
      IP=1
      RETURN
      END
```

```
      SUBROUTINE TURBC (II)
C
C     ****************************************************************
C
C     THIS SUBROUTINE SETS THE BOUNDARY CONDITIONS FOR THE TURBULENCE
C     QUANTITIES Q AND E
C
C     ****************************************************************
C
*CALL,MCC
      YI1=(YI(2)-YI(1))/(YI(3)-YI(2))
      YIM=(YI(MMAX)-YI(M1))/(YI(M1)-YI(M2))
      IF (MDFS.EQ.0) GO TO 10
      YIU=(YI(MDFS+1)-YI(MDFS))/(YI(MDFS+2)-YI(MDFS+1))
      IF (LDFSS.EQ.1) YIL=(YL(1)-YI(MDFS-1))/(YI(MDFS-1)-YI(MDFS-2))
      IF (LDFSS.NE.1) YIL=(YI(MDFS)-YI(MDFS-1))/(YI(MDFS-1)-YI(MDFS-2))
   10 GO TO (20,70,150), II
C
C     SET QUANTITIES AFTER EACH TIME STEP
C
   20 DO 30 M=1,MMAX
      Q(1,M,N3)=FSQ(M)
      E(1,M,N3)=FSE(M)
   30 CONTINUE
      DO 40 L=2,L1
      Q(L,MMAX,N3)=Q(L,M1,N3)+YIM*(Q(L,M1,N3)-Q(L,M2,N3))
      E(L,MMAX,N3)=E(L,M1,N3)+YIM*(E(L,M1,N3)-E(L,M2,N3))
      IF (NOSLIP.NE.0.AND.IWALL.EQ.0) Q(L,MMAX,N3)=0.0
      IF (NGCB.EQ.0) GO TO 40
      Q(L,1,N3)=Q(L,2,N3)+YI1*(Q(L,2,N3)-Q(L,3,N3))
      E(L,1,N3)=E(L,2,N3)+YI1*(E(L,2,N3)-E(L,3,N3))
      IF (NOSLIP.NE.0) Q(L,1,N3)=0.0
   40 CONTINUE
      DO 50 M=1,MMAX
      Q(LMAX,M,N3)=Q(L1,M,N3)
      E(LMAX,M,N3)=E(L1,M,N3)
   50 CONTINUE
      IF (MDFS.EQ.0) GO TO 280
      QL(1,N3)=FSQL
      EL(1,N3)=FSEL
      DO 60 L=LDFSS,LDFSF
      Q(L,MDFS,N3)=Q(L,MDFS+1,N3)+YIU*(Q(L,MDFS+1,N3)-Q(L,MDFS+2,N3))
      E(L,MDFS,N3)=E(L,MDFS+1,N3)+YIU*(E(L,MDFS+1,N3)-E(L,MDFS+2,N3))
      QL(L,N3)=Q(L,MDFS-1,N3)+YIL*(Q(L,MDFS-1,N3)-Q(L,MDFS-2,N3))
      EL(L,N3)=E(L,MDFS-1,N3)+YIL*(E(L,MDFS-1,N3)-E(L,MDFS-2,N3))
      IF (NOSLIP.NE.0) Q(L,MDFS,N3)=0.0
      IF (NOSLIP.NE.0) QL(L,N3)=0.0
   60 CONTINUE
      GO TO 280
C
C     SET QUANTITIES AFTER EACH SUBCYCLE TIME STEP
C
   70 DO 80 M=MVCB,MVCT
      Q(1,M,N3)=FSQ(M)
      E(1,M,N3)=FSE(M)
   80 CONTINUE
      IF (MVCT.NE.MMAX) GO TO 100
      DO 90 L=2,L1
      Q(L,MMAX,N3)=Q(L,M1,N3)+YIM*(Q(L,M1,N3)-Q(L,M2,N3))
      E(L,MMAX,N3)=E(L,M1,N3)+YIM*(E(L,M1,N3)-E(L,M2,N3))
      IF (NOSLIP.NE.0.AND.IWALL.EQ.0) Q(L,MMAX,N3)=0.0
   90 CONTINUE
  100 IF (MVCB.NE.1.OR.NGCB.EQ.0) GO TO 120
      DO 110 L=2,L1
      Q(L,1,N3)=Q(L,2,N3)+YI1*(Q(L,2,N3)-Q(L,3,N3))
      E(L,1,N3)=E(L,2,N3)+YI1*(E(L,2,N3)-E(L,3,N3))
      IF (NOSLIP.NE.0) Q(L,1,N3)=0.0
  110 CONTINUE
  120 DO 130 M=MVCB,MVCT
      Q(LMAX,M,N3)=Q(L1,M,N3)
      E(LMAX,M,N3)=E(L1,M,N3)
  130 CONTINUE
```

```
      IF (MDFS.EQ.O) GO TO 280
      QL(1,N3)=FSQL
      EL(1,N3)=FSEL
      IF (MVCB.GT.MDFS.OR.MVCT.LT.MDFS) GO TO 280
      DO 140 L=LDFSS,LDFSF
      Q(L,MDFS,N3)=Q(L,MDFS+1,N3)+YIU*(Q(L,MDFS+1,N3)-Q(L,MDFS+2,N3))
      E(L,MDFS,N3)=E(L,MDFS+1,N3)+YIU*(E(L,MDFS+1,N3)-E(L,MDFS+2,N3))
      QL(L,N3)=Q(L,MDFS-1,N3)+YIL*(Q(L,MDFS-1,N3)-Q(L,MDFS-2,N3))
      EL(L,N3)=E(L,MDFS-1,N3)+YIL*(E(L,MDFS-1,N3)-E(L,MDFS-2,N3))
      IF (NOSLIP.NE.O) Q(L,MDFS,N3)=0.0
      IF (NOSLIP.NE.O) QL(L,N3)=0.0
  140 CONTINUE
      GO TO 280
C
C     SET QUANTITIES AFTER ALL PREDICTOR STEPS
C
  150 IF (NVC.NE.1) GO TO 190
      IF (MVCT.EQ.MMAX) GO TO 170
      DO 160 L=2,L1
      Q(L,MMAX,N3)=Q(L,M1,N3)+YIM*(Q(L,M1,N3)-Q(L,M2,N3))
      E(L,MMAX,N3)=E(L,M1,N3)+YIM*(E(L,M1,N3)-E(L,M2,N3))
      IF (NOSLIP.NE.O.AND.IWALL.EQ.O) Q(L,MMAX,N3)=0.0
  160 CONTINUE
  170 DO 180 M=1,MMAX
      IF (M.GE.MVCB.AND.M.LE.MVCT) GO TO 180
      Q(LMAX,M,N3)=Q(L1,M,N3)
      E(LMAX,M,N3)=E(L1,M,N3)
  180 CONTINUE
      GO TO 230
  190 IF (MVCT.NE.MMAX) GO TO 210
      DO 200 L=2,L1
      Q(L,MMAX,N3)=Q(L,M1,N3)+YIM*(Q(L,M1,N3)-Q(L,M2,N3))
      E(L,MMAX,N3)=E(L,M1,N3)+YIM*(E(L,M1,N3)-E(L,M2,N3))
      IF (NOSLIP.NE.O.AND.IWALL.EQ.O) Q(L,MMAX,N3)=0.0
  200 CONTINUE
  210 DO 220 M=MVCB,MVCT
      Q(LMAX,M,N3)=Q(L1,M,N3)
      E(LMAX,M,N3)=E(L1,M,N3)
  220 CONTINUE
  230 IF (MDFS.EQ.O) GO TO 280
      IF (NVC.NE.1) GO TO 240
      IF (MDFS.GT.MVCB.AND.MDFS.LT.MVCT) GO TO 270
      GO TO 250
  240 IF (MDFS.LT.MVCB.OR.MDFS.GT.MVCT) GO TO 270
  250 DO 260 L=LDFSS,LDFSF
      QL(L,N3)=Q(L,MDFS-1,N3)+YIL*(Q(L,MDFS-1,N3)-Q(L,MDFS-2,N3))
      EL(L,N3)=E(L,MDFS-1,N3)+YIL*(E(L,MDFS-1,N3)-E(L,MDFS-2,N3))
      IF (NOSLIP.NE.O) QL(L,N3)=0.0
  260 CONTINUE
  270 IF (LDFSF.NE.LMAX) GO TO 280
      QL(LMAX,N3)=QL(L1,N3)
      EL(LMAX,N3)=EL(L1,N3)
C
  280 DO 290 L=1,LMAX
      IF (Q(L,1,N3).LT.0.0) Q(L,1,N3)=QLOW
      IF (E(L,1,N3).LT.0.0) E(L,1,N3)=ELOW
      IF (Q(L,MMAX,N3).LT.0.0) Q(L,MMAX,N3)=QLOW
      IF (E(L,MMAX,N3).LT.0.0) E(L,MMAX,N3)=ELOW
      IF (MDFS.EQ.O) GO TO 290
      IF (Q(L,MDFS,N3).LT.0.0) Q(L,MDFS,N3)=QLOW
      IF (E(L,MDFS,N3).LT.0.0) E(L,MDFS,N3)=ELOW
      IF (QL(L,N3).LT.0.0) QL(L,N3)=QLOW
      IF (EL(L,N3).LT.0.0) EL(L,N3)=ELOW
  290 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INTER
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE INTERIOR MESH POINTS
C
C     ****************************************************************
C
*CALL,MCC
      IP=1
      ATERM=0.0
      MIS=1
      IF (NGCB.NE.0) MIS=2
      MIF=M1
      IF (ICHAR.NE.1) GO TO 200
C
C     COMPUTE THE TENTATIVE SOLUTION AT T+DT
C
      IF (IVC.EQ.0) GO TO 10
      IF (NVC.EQ.1) GO TO 10
      MIS=MVCB
      MIF=MVCT+1
      IF (MVCB.EQ.1.AND.NGCB.NE.0) MIS=2
      IF (MIF.GE.MMAX) MIF=M1
C
C     BEGIN THE L OR X DO LOOP
C
   10 DO 190 L=2,L1
      LMAP=L
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
C
C     BEGIN THE M OR Y DO LOOP
C
      DO 180 M=MIS,MIF
      IF (IVC.EQ.0) GO TO 20
      IF (NVC.NE.1) GO TO 20
      IF (M.LE.MVCB.AND.MVCB.NE.1) GO TO 20
      IF (M.GT.MVCT) GO TO 20
      GO TO 180
   20 IF (M.EQ.MDFS.AND.LDFS.EQ.1) GO TO 180
      MMAP=M
      CALL MAP
      OM=OM1
      AL=AL3
      BE=BE3
      DE=DE3
      UB=U(L,M,N1)
      VB=V(L,M,N1)
      PB=P(L,M,N1)
      ROB=RO(L,M,N1)
      ROR=1.0/ROB
      ASB=GAMMA*PB*ROR
      QB=Q(L,M,N1)
      EB=E(L,M,N1)
      IF (M.NE.1) GO TO 60
C
C     CALCULATE THE QUANTITIES FOR M=1
C
      DUDX=(UB-U(L-1,M,N1))*DXR
      DPDX=(PB-P(L-1,M,N1))*DXR
      DRODX=(ROB-RO(L-1,M,N1))*DXR
      DVDY=(4.0*V(L,2,N1)-V(L,3,N1))*0.5*DYR
      IF (ITM.LE.1) GO TO 30
      DQDX=(QB-Q(L-1,M,N1))*DXR
      DEDX=(EB-E(L-1,M,N1))*DXR
   30 V(L,M,N3)=0.0
      URHS=-UB*OM*DUDX-OM*DPDX*ROR+QUT(L,M)
      RORHS=-UB*OM*DRODX-ROB*OM*DUDX-FLOAT(1+NDIM)*ROB*BE*DVDY+QROT(L,M)
      PRHS=-UB*OM*DPDX+ASB*(RORHS+UB*OM*DRODX)+QPT(L,M)
      IF (ITM.LE.1) GO TO 170
      IF (UB.GE.0.0) GO TO 40
```

```
      DQDX=(Q(L+1,M,N1)-QB)*DXR
      DEDX=(E(L+1,M,N1)-EB)*DXR
      OM=OM2
   40 QRHS=-UB*OM*DQDX+QQT(L,M)
      Q(L,M,N3)=QB+QRHS*DT
      IF (Q(L,M,N3).LT.QLOW) Q(L,M,N3)=QLOW
      IF (ITM.EQ.2) GO TO 170
      ERHS=-UB*OM*DEDX+QET(L,M)
      E(L,M,N3)=EB+ERHS*DT
      IF (MDFS.NE.O.AND.LDFS.EQ.O) GO TO 50
      IF (Q(L,M,N3).LT.BFST*FSQ(M)) Q(L,M,N3)=BFST*FSQ(M)
      IF (E(L,M,N3).LT.BFST*FSE(M)) E(L,M,N3)=BFST*FSE(M)
   50 IF (E(L,M,N3).GT.ELOW) GO TO 170
      Q(L,M,N3)=QLOW
      E(L,M,N3)=ELOW
      GO TO 170
C
C     CALCULATE THE QUANTITIES FOR M NOT EQUAL TO 1
C
   60 IF (IVC.EQ.O) GO TO 70
      IF (NVC.EQ.1.OR.M.NE.MVCT+1) GO TO 70
C
C     LINEAR INTERPOLATION IN TIME FOR M=MVCT+1
C
      UB=UU1(L)+RIND*(UU2(L)-UU1(L))
      VB=VV1(L)+RIND*(VV2(L)-VV1(L))
      PB=PP1(L)+RIND*(PP2(L)-PP1(L))
      ROB=RORO1(L)+RIND*(RORO2(L)-RORO1(L))
      ROR=1.0/ROB
      ASB=GAMMA*PB*ROR
      ULM=UU1(L-1)+RIND*(UU2(L-1)-UU1(L-1))
      VLM=VV1(L-1)+RIND*(VV2(L-1)-VV1(L-1))
      PLM=PP1(L-1)+RIND*(PP2(L-1)-PP1(L-1))
      ROLM=RORO1(L-1)+RIND*(RORO2(L-1)-RORO1(L-1))
      IF (ITM.LE.1) GO TO 80
      QB=QQ1(L)+RIND*(QQ2(L)-QQ1(L))
      EB=EE1(L)+RIND*(EE2(L)-EE1(L))
      QLM=QQ1(L-1)+RIND*(QQ2(L-1)-QQ1(L-1))
      ELM=EE1(L-1)+RIND*(EE2(L-1)-EE1(L-1))
      GO TO 80
C
   70 ULM=U(L-1,M,N1)
      VLM=V(L-1,M,N1)
      PLM=P(L-1,M,N1)
      ROLM=RO(L-1,M,N1)
      QLM=Q(L-1,M,N1)
      ELM=E(L-1,M,N1)
      IF (M.NE.MDFS.OR.L.NE.LDFSF+1) GO TO 80
      ULM=0.5*(ULM+UL(L-1,N1))
      VLM=0.5*(VLM+VL(L-1,N1))
      PLM=0.5*(PLM+PL(L-1,N1))
      ROLM=0.5*(ROLM+ROL(L-1,N1))
      IF (ITM.LE.1) GO TO 80
      QLM=0.5*(QLM+QL(L-1,N1))
      ELM=0.5*(ELM+EL(L-1,N1))
   80 UVB=UB*AL+VB*BE+DE
      IF (NDIM.NE.0) ATERM=ROB*VB/YP
      DUDX=(UB-ULM)*DXR
      DVDX=(VB-VLM)*DXR
      DPDX=(PB-PLM)*DXR
      DRODX=(ROB-ROLM)*DXR
      IF (ITM.LE.1) GO TO 90
      DQDX=(QB-QLM)*DXR
      DEDX=(EB-ELM)*DXR
   90 IF (IVC.EQ.O) GO TO 110
      IF (NVC.EQ.1.OR.M.NE.MVCB) GO TO 110
C
C     LINEAR INTERPOLATION IN TIME FOR M=MVCB
C
      UMM=U(L,M-1,NN1)+RIND*(U(L,M-1,NN3)-U(L,M-1,NN1))
      VMM=V(L,M-1,NN1)+RIND*(V(L,M-1,NN3)-V(L,M-1,NN1))
      PMM=P(L,M-1,NN1)+RIND*(P(L,M-1,NN3)-P(L,M-1,NN1))
```

```
      ROMM=RO(L,M-1,NN1)+RIND*(RO(L,M-1,NN3)-RO(L,M-1,NN1))
      IF (ITM.LE.1) GO TO 100
      QMM=Q(L,M-1,NN1)+RIND*(Q(L,M-1,NN3)-Q(L,M-1,NN1))
      EMM=E(L,M-1,NN1)+RIND*(E(L,M-1,NN3)-E(L,M-1,NN1))
C
  100 DUDY=(UB-UMM)*DYR
      DVDY=(VB-VMM)*DYR
      DPDY=(PB-PMM)*DYR
      DRODY=(ROB-ROMM)*DYR
      IF (ITM.LE.1) GO TO 120
      DQDY=(QB-QMM)*DYR
      DEDY=(EB-EMM)*DYR
      GO TO 120
  110 DUDY=(UB-U(L,M-1,N1))*DYR
      DVDY=(VB-V(L,M-1,N1))*DYR
      DPDY=(PB-P(L,M-1,N1))*DYR
      DRODY=(ROB-RO(L,M-1,N1))*DYR
      IF (ITM.LE.1) GO TO 120
      DQDY=(QB-Q(L,M-1,N1))*DYR
      DEDY=(EB-E(L,M-1,N1))*DYR
C
C     SPECIAL FORM OF THE EQUATIONS USED BY THE QUICK SOLVER
C
  120 IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 130
      IF (M.EQ.MVCB.OR.M.GE.MVCT) GO TO 130
      ALS=SQRT(AL*AL+BE*BE)
      RALS=1.0/ALS
      AB=SQRT(ASB)
      ABR=AL/BE
      UVBP=UVB+ALS*AB
      UVBM=UVB-ALS*AB
      USL=-UVB*DUDY+ABR*UVB*DVDY-UB*OM*(DUDX-ABR*DVDX)-OM*DPDX*ROR+QUT(L
     1 ,M)-ABR*QVT(L,M)
      PMLP=-UB*OM*DPDX-ROB*ASB*OM*DUDX-ASB*ATERM-ROB*AB*OM*RALS*(AL*(UB
     1 *DUDX+DPDX*ROR)+BE*UB*DVDX)+QPT(L,M)+ASB*QROT(L,M)+ROB*AB*RALS*
     2 (AL*QUT(L,M)+BE*QVT(L,M))
      PMLM=-UB*OM*DPDX-ROB*ASB*OM*DUDX-ASB*ATERM+ROB*AB*OM*RALS*(AL*(UB
     1 *DUDX+DPDX*ROR)+BE*UB*DVDX)+QPT(L,M)+ASB*QROT(L,M)-ROB*AB*RALS*
     2 (AL*QUT(L,M)+BE*QVT(L,M))
      PMLP1=-UVBP*DPDYQS(L,M,1)-ROB*AB*RALS*UVBP*(AL*DUDYQS(L,M,1)+BE
     1 *DVDYQS(L,M,1))+PMLP
      PMLM1=-UVBM*DPDYQS(L,M,2)+ROB*AB*RALS*UVBM*(AL*DUDYQS(L,M,2)+BE
     1 *DVDYQS(L,M,2))+PMLM
      VRHS=-(2.0*ROB*AB*AL*RALS*USL+PMLM1-PMLP1)/(2.0*ROB*AB*ALS/BE)
      PRHS=0.5*(PMLP1+PMLM1)
      URHS=ABR*VRHS+USL
      RORHS=-UB*OM*DRODX-UVB*DRODY+(PRHS+UB*OM*DPDX+UVB*DPDY-QPT(L,M))
     1 /ASB
      GO TO 140
C
C     REGULAR FORM OF THE EQUATIONS
C
  130 URHS=-UB*OM*DUDX-UVB*DUDY-(OM*DPDX+AL*DPDY)*ROR+QUT(L,M)
      VRHS=-UB*OM*DVDX-UVB*DVDY-BE*DPDY*ROR+QVT(L,M)
      RORHS=-UB*OM*DRODX-UVB*DRODY-ROB*(OM*DUDX+AL*DUDY+BE*DVDY)-ATERM
     1 +QROT(L,M)
      PRHS=-UB*OM*DPDX-UVB*DPDY+ASB*(RORHS+UB*OM*DRODX+UVB*DRODY)+QPT(L
     1 ,M)
C
  140 V(L,M,N3)=VB+VRHS*DT
      IF (ITM.LE.1) GO TO 170
      IF (UB.GE.0.0) GO TO 150
      DQDX=(Q(L+1,M,N1)-QB)*DXR
      DEDX=(E(L+1,M,N1)-EB)*DXR
      OM=OM2
  150 QRHS=-UB*OM*DQDX-UVB*DQDY+QQT(L,M)
      Q(L,M,N3)=QB+QRHS*DT
      IF (Q(L,M,N3).LT.QLOW) Q(L,M,N3)=QLOW
      IF (ITM.EQ.2) GO TO 170
      ERHS=-UB*OM*DEDX-UVB*DEDY+QET(L,M)
      E(L,M,N3)=EB+ERHS*DT
      IF (MDFS.NE.0.AND.LDFS.EQ.0) GO TO 160
```

```
      IF (Q(L,M,N3).LT.BFST*FSQ(M)) Q(L,M,N3)=BFST*FSQ(M)
      IF (E(L,M,N3).LT.BFST*FSE(M)) E(L,M,N3)=BFST*FSE(M)
  160 IF (E(L,M,N3).GT.ELOW) GO TO 170
      Q(L,M,N3)=QLOW
      E(L,M,N3)=ELOW
  170 U(L,M,N3)=UB+URHS*DT
      P(L,M,N3)=PB+PRHS*DT
      RO(L,M,N3)=ROB+RORHS*DT
      IF (P(L,M,N3).LE.0.0) P(L,M,N3)=PLOW*PC
      IF (RO(L,M,N3).LE.0.0) RO(L,M,N3)=ROLOW/G
  180 CONTINUE
  190 CONTINUE
      RETURN
C
C     COMPUTE THE FINAL SOLUTION AT T+DT
C
  200 IF (IVC.EQ.0) GO TO 210
      IF (NVC.EQ.1) GO TO 210
      MIS=MVCB
      MIF=MVCT
      IF (MVCB.EQ.1.AND.NGCB.NE.0) MIS=2
      IF (MIF.EQ.MMAX) MIF=M1
C
C     BEGIN THE L OR X DO LOOP
C
  210 DO 390 L=2,L1
      LMAP=L
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      UOLD=U(L,1,N3)
      VOLD=V(L,1,N3)
      POLD=P(L,1,N3)
C
C     BEGIN THE M OR Y DO LOOP
C
      DO 380 M=MIS,MIF
      IF (IVC.EQ.0) GO TO 220
      IF (NVC.NE.1) GO TO 220
      IF (M.LT.MVCB) GO TO 220
      IF (M.GT.MVCT) GO TO 220
      GO TO 380
  220 IF (M.EQ.MDFS.AND.LDFS.EQ.1) GO TO 380
      MMAP=M
      CALL MAP
      OM=OM2
      AL=AL4
      BE=BE4
      DE=DE4
      BED=BE3
      UB=U(L,M,N3)
      VB=V(L,M,N3)
      PB=P(L,M,N3)
      ROB=RO(L,M,N3)
      ROR=1.0/ROB
      ASB=GAMMA*PB*ROR
      QB=Q(L,M,N3)
      EB=E(L,M,N3)
      IF (M.NE.1) GO TO 260
C
C     CALCULATE THE QUANTITIES FOR M=1
C
      DUDX=(U(L+1,M,N3)-UB)*DXR
      DPDX=(P(L+1,M,N3)-PB)*DXR
      DRODX=(RO(L+1,M,N3)-ROB)*DXR
      DVDY=(4.0*V(L,2,N3)-V(L,3,N3))*0.5*DYR
      IF (ITM.LE.1) GO TO 230
      DQDX=(Q(L+1,M,N3)-QB)*DXR
      DEDX=(E(L+1,M,N3)-EB)*DXR
  230 V(L,M,N3)=0.0
      URHS=-UB*OM*DUDX-OM*DPDX*ROR+QUT(L,M)
      RORHS=-UB*OM*DRODX-ROB*OM*DUDX-FLOAT(1+NDIM)*ROB*BE*DVDY+QROT(L,M)
      PRHS=-UB*OM*DPDX+ASB*(RORHS+UB*OM*DRODX)+QPT(L,M)
```

```
      IF (ITM.LE.1) GO TO 370
      IF (U(L,M,N1).LT.0.0) GO TO 240
      DQDX=(QB-Q(L-1,M,N3))*DXR
      DEDX=(EB-E(L-1,M,N3))*DXR
      OM=OM1
  240 QRHS=-UB*OM*DQDX+QQT(L,M)
      Q(L,M,N3)=0.5*(Q(L,M,N1)+Q(L,M,N3)+QRHS*DT)
      IF (Q(L,M,N3).LT.QLOW) Q(L,M,N3)=QLOW
      IF (ITM.EQ.2) GO TO 370
      ERHS=-UB*OM*DEDX+QET(L,M)
      E(L,M,N3)=0.5*(E(L,M,N1)+E(L,M,N3)+ERHS*DT)
      IF (MDFS.NE.0.AND.LDFS.EQ.0) GO TO 250
      IF (Q(L,M,N3).LT.BFST*FSQ(M)) Q(L,M,N3)=BFST*FSQ(M)
      IF (E(L,M,N3).LT.BFST*FSE(M)) E(L,M,N3)=BFST*FSE(M)
  250 IF (E(L,M,N3).GT.ELOW) GO TO 370
      Q(L,M,N3)=QLOW
      E(L,M,N3)=ELOW
      GO TO 370
C
C     CALCULATE THE QUANTITIES FOR M NOT EQUAL TO 1
C
  260 IF (NDIM.NE.0) ATERM=ROB*VB/YP
      ULP=U(L+1,M,N3)
      VLP=V(L+1,M,N3)
      PLP=P(L+1,M,N3)
      ROLP=RO(L+1,M,N3)
      QLM=Q(L-1,M,N3)
      ELM=E(L-1,M,N3)
      IF (M.NE.MDFS.OR.L.NE.LDFSS-1) GO TO 270
      ULP=0.5*(ULP+UL(L+1,N3))
      VLP=0.5*(VLP+VL(L+1,N3))
      PLP=0.5*(PLP+PL(L+1,N3))
      ROLP=0.5*(ROLP+ROL(L+1,N3))
  270 IF (M.NE.MDFS.OR.L.NE.LDFSF+1) GO TO 280
      IF (ITM.LE.1) GO TO 280
      QLM=0.5*(QLM+QL(L-1,N3))
      ELM=0.5*(ELM+EL(L-1,N3))
  280 UVB=UB*AL+VB*BE+DE
      DUDX=(ULP-UB)*DXR
      DVDX=(VLP-VB)*DXR
      DPDX=(PLP-PB)*DXR
      DRODX=(ROLP-ROB)*DXR
      IF (ITM.LE.1) GO TO 290
      DQDX=(QB-QLM)*DXR
      DEDX=(EB-ELM)*DXR
  290 DUDY=(U(L,M+1,N3)-UB)*DYR
      DVDY=(V(L,M+1,N3)-VB)*DYR
      DPDY=(P(L,M+1,N3)-PB)*DYR
      DRODY=(RO(L,M+1,N3)-ROB)*DYR
      IF (ITM.LE.1) GO TO 300
      DQDY=(Q(L,M+1,N3)-QB)*DYR
      DEDY=(E(L,M+1,N3)-EB)*DYR
C
C     SPECIAL FORM OF THE EQUATIONS USED BY THE QUICK SOLVER
C
  300 IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 320
      IF (M.EQ.MVCB.OR.M.EQ.MVCT) GO TO 320
      ALS=SQRT(AL*AL+BE*BE)
      RALS=1.0/ALS
      AB=SQRT(ASB)
      ABR=AL/BE
      UVBP=UVB+ALS*AB
      UVBM=UVB-ALS*AB
      BER=BED/BE
      DUDY1=(UB-UOLD)*DYR*BER
      DVDY1=(VB-VOLD)*DYR*BER
      DPDY1=(PB-POLD)*DYR*BER
      IF (MDFS.EQ.0) GO TO 310
      IF (M.NE.MDFS+1.OR.LDFS.EQ.0) GO TO 310
      DUDY1=(UB-UL(L,N3))*DYR*BER
      DVDY1=(VB-VL(L,N3))*DYR*BER
      DPDY1=(PB-PL(L,N3))*DYR*BER
```

```
  310 USL=-UVB*DUDY+ABR*UVB*DVDY-UB*OM*(DUDX-ABR*DVDX)-OM*DPDX*ROR+QUT(L
     1 ,M)-ABR*QVT(L,M)
      PMLP=-UB*OM*DPDX-ROB*ASB*OM*DUDX-ASB*ATERM-ROB*AB*OM*RALS*(AL*(UB
     1 *DUDX+DPDX*ROR)+BE*UB*DVDX)+QPT(L,M)+ASB*QROT(L,M)+ROB*AB*RALS*
     2 (AL*QUT(L,M)+BE*QVT(L,M))
      PMLM=-UB*OM*DPDX-ROB*ASB*OM*DUDX-ASB*ATERM+ROB*AB*OM*RALS*(AL*(UB
     1 *DUDX+DPDX*ROR)+BE*UB*DVDX)+QPT(L,M)+ASB*QROT(L,M)-ROB*AB*RALS*
     2 (AL*QUT(L,M)+BE*QVT(L,M))
      PMLP1=-UVBP*DPDY1-ROB*AB*RALS*UVBP*(AL*DUDY1+BE*DVDY1)+PMLP
      PMLM1=-UVBM*DPDY+ROB*AB*RALS*UVBM*(AL*DUDY+BE*DVDY)+PMLM
      VRHS=-(2.0*ROB*AB*AL*RALS*USL+PMLM1-PMLP1)/(2.0*ROB*AB*ALS/BE)
      PRHS=0.5*(PMLP1+PMLM1)
      URHS=ABR*VRHS+USL
      RORHS=-UB*OM*DRODX-UVB*DRODY+(PRHS+UB*OM*DPDX+UVB*DPDY-QPT(L,M))
     1 /ASB
      GO TO 330
C
C     REGULAR FORM OF THE EQUATIONS
C
  320 URHS=-UB*OM*DUDX-UVB*DUDY-(OM*DPDX+AL*DPDY)*ROR+QUT(L,M)
      VRHS=-UB*OM*DVDX-UVB*DVDY-BE*DPDY*ROR+QVT(L,M)
      RORHS=-UB*OM*DRODX-UVB*DRODY-ROB*(OM*DUDX+AL*DUDY+BE*DVDY)-ATERM
     1 +QROT(L,M)
      PRHS=-UB*OM*DPDX-UVB*DPDY+ASB*(RORHS+UB*OM*DRODX+UVB*DRODY)+QPT(L
     1 ,M)
C
  330 IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 340
      UOLD=U(L,M,N3)
      VOLD=V(L,M,N3)
      POLD=P(L,M,N3)
  340 V(L,M,N3)=0.5*(V(L,M,N1)+V(L,M,N3)+VRHS*DT)
      IF (ITM.LE.1) GO TO 370
      IF (U(L,M,N1).GE.0.0) GO TO 350
      DQDX=(Q(L+1,M,N3)-QB)*DXR
      DEDX=(E(L+1,M,N3)-EB)*DXR
      OM=OM1
  350 QRHS=-UB*OM*DQDX-UVB*DQDY+QQT(L,M)
      Q(L,M,N3)=0.5*(Q(L,M,N1)+Q(L,M,N3)+QRHS*DT)
      IF (Q(L,M,N3).LT.QLOW) Q(L,M,N3)=QLOW
      IF (ITM.EQ.2) GO TO 370
      ERHS=-UB*OM*DEDX-UVB*DEDY+QET(L,M)
      E(L,M,N3)=0.5*(E(L,M,N1)+E(L,M,N3)+ERHS*DT)
      IF (MDFS.NE.0.AND.LDFS.EQ.0) GO TO 360
      IF (Q(L,M,N3).LT.BFST*FSQ(M)) Q(L,M,N3)=BFST*FSQ(M)
      IF (E(L,M,N3).LT.BFST*FSE(M)) E(L,M,N3)=BFST*FSE(M)
  360 IF (E(L,M,N3).GT.ELOW) GO TO 370
      Q(L,M,N3)=QLOW
      E(L,M,N3)=ELOW
  370 U(L,M,N3)=0.5*(U(L,M,N1)+U(L,M,N3)+URHS*DT)
      P(L,M,N3)=0.5*(P(L,M,N1)+P(L,M,N3)+PRHS*DT)
      RO(L,M,N3)=0.5*(RO(L,M,N1)+RO(L,M,N3)+RORHS*DT)
      IF (P(L,M,N3).LE.0.0) P(L,M,N3)=PLOW*PC
      IF (RO(L,M,N3).LE.0.0) RO(L,M,N3)=ROLOW/G
  380 CONTINUE
  390 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WALL
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE
C     WALL, FREE-JET BOUNDARY, CENTERBODY AND DUAL FLOW SPACE WALLS
C
C     ****************************************************************
C
*CALL,MCC
      IP=1
      Y2=0.0
      Y20=0.0
      NOSL=NOSLIP
      IF (IB.EQ.1.AND.IWALL.NE.0) NOSL=0
      IF (N.EQ.1.AND.JFLAG.NE.0) DELY=0.0001*YW(LJET-1)
      XWID=0.0
      ATERM2=0.0
      ATERM3=0.0
      IF (IB.EQ.1) GO TO 10
      IF (IB.GT.2) GO TO 20
      Y3=0.0
      MDUM=1
      MDUM1=2
      SIGN=-1.0
      GO TO 40
   10 Y3=1.0
      MDUM=MMAX
      MDUM1=M1
      SIGN=1.0
      GO TO 40
   20 Y3=Y(MDFS)
      MDUM=MDFS
      IF (IB.EQ.4) GO TO 30
      MDUM1=MDFS-1
      SIGN=1.0
      GO TO 40
   30 MDUM1=MDFS+1
      SIGN=-1.0
   40 DYS=SIGN*DYR
      MMAP=MDUM
C
C     BEGIN THE L OR X DO LOOP
C
      DO 700 L=2,L1
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      IF (IB.GE.3.AND.LDFS.EQ.0) GO TO 700
      LMAP=L
      CALL MAP
      AL=AL3
      BE=BE3
      DE=DE3
C
      IF (JFLAG.EQ.0) GO TO 70
      IF (IB.NE.1) GO TO 70
      XWID=XWI(L)
      IF (ICHAR.EQ.1) GO TO 50
C
C     USE THE DUMMY ARRAYS TO MANIPULATE THE ONE-SIDED SOLUTIONS
C     FOR THE FREE-JET OR SHARP EXPANSION CORNER CASES
C
      IF (L.NE.LJET-2) GO TO 50
      U(L+1,MDUM,N3)=UD(3)
      V(L+1,MDUM,N3)=VD(3)
      P(L+1,MDUM,N3)=PD(3)
      RO(L+1,MDUM,N3)=ROD(3)
      GO TO 70
   50 IF (L.NE.LJET-1) GO TO 60
      IF (ICHAR.EQ.1) UOLD=U(L,MDUM,N1)
      U(L,MDUM,N1)=UD(1)
      V(L,MDUM,N1)=VD(1)
```

```
      P(L,MDUM,N1)=PD(1)
      RO(L,MDUM,N1)=ROD(1)
      GO TO 70
   60 IF (L.NE.LJET) GO TO 70
      U(L-1,MDUM,N1)=UD(2)
      V(L-1,MDUM,N1)=VD(2)
      P(L-1,MDUM,N1)=PD(2)
      RO(L-1,MDUM,N1)=ROD(2)
C
   70 U1=U(L,MDUM,N1)
      V1=V(L,MDUM,N1)
      P1=P(L,MDUM,N1)
      RO1=RO(L,MDUM,N1)
      U2=U1
      V2=V1
      A1=SQRT(GAMMA*P1/RO1)
      A2=A1
      IF (ICHAR.NE.1) GO TO 80
      U3=U1
      V3=V1
      P3=P1
      RO3=RO1
      A3=A1
      GO TO 90
   80 U3=U(L,MDUM,N3)
      V3=V(L,MDUM,N3)
      P3=P(L,MDUM,N3)
      RO3=RO(L,MDUM,N3)
      A3=SQRT(GAMMA*P3/RO3)
C
C     CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
C
   90 BU=(U1-U(L,MDUM1,N1))*DYS
      BV=(V1-V(L,MDUM1,N1))*DYS
      BP=(P1-P(L,MDUM1,N1))*DYS
      BRO=(RO1-RO(L,MDUM1,N1))*DYS
      BQUT=(QUT(L,MDUM)-QUT(L,MDUM1))*DYS
      BQVT=(QVT(L,MDUM)-QVT(L,MDUM1))*DYS
      BQPT=(QPT(L,MDUM)-QPT(L,MDUM1))*DYS
      BQROT=(QROT(L,MDUM)-QROT(L,MDUM1))*DYS
      CU=U1-BU*Y3
      CV=V1-BV*Y3
      CP=P1-BP*Y3
      CRO=RO1-BRO*Y3
      CQUT=QUT(L,MDUM)-BQUT*Y3
      CQVT=QVT(L,MDUM)-BQVT*Y3
      CQPT=QPT(L,MDUM)-BQPT*Y3
      CQROT=QROT(L,MDUM)-BQROT*Y3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
      DU=(U1-U(L-1,MDUM,N1))*DXR
      DV=(V1-V(L-1,MDUM,N1))*DXR
      DP=(P1-P(L-1,MDUM,N1))*DXR
      DRO=(RO1-RO(L-1,MDUM,N1))*DXR
      DU1=(U(L,MDUM1,N1)-U(L-1,MDUM1,N1))*DXR
      DV1=(V(L,MDUM1,N1)-V(L-1,MDUM1,N1))*DXR
      DP1=(P(L,MDUM1,N1)-P(L-1,MDUM1,N1))*DXR
      DRO1=(RO(L,MDUM1,N1)-RO(L-1,MDUM1,N1))*DXR
      BDU=(DU-DU1)*DYS
      BDV=(DV-DV1)*DYS
      BDP=(DP-DP1)*DYS
      BDRO=(DRO-DRO1)*DYS
      CDU=DU-BDU*Y3
      CDV=DV-BDV*Y3
      CDP=DP-BDP*Y3
      CDRO=DRO-BDRO*Y3
C
C     CALCULATE Y2
C
      ALS=SQRT(AL*AL+BE*BE)
```

```
      UV3=U3*AL+V3*BE+DE
      DO 130 ILL=1,3
      UV2=U2*AL+V2*BE+DE
      Y2=Y3-(UV2+SIGN*ALS*A2+UV3+SIGN*ALS*A3)*DT*0.5
C
      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 100
      IF (IB.EQ.1.AND.Y2.LT.Y(M1)) Y2=Y(M1)
      IF (IB.EQ.2.AND.Y2.GT.Y(2)) Y2=Y(2)
      IF (MDFS.EQ.0) GO TO 100
      IF (IB.EQ.3.AND.Y2.LT.Y(MDFS-1)) Y2=Y(MDFS-1)
      IF (IB.EQ.4.AND.Y2.GT.Y(MDFS+1)) Y2=Y(MDFS+1)
C
  100 IF (IWALL.EQ.0.OR.IB.NE.1) GO TO 110
      UV1=U1*AL+V1*BE
      Y1=Y3-(UV1+UV3)*DT*0.5
C
C     INTERPOLATE FOR THE PROPERTIES
C
      U1=BU*Y1+CU
      V1=BV*Y1+CV
      P1=BP*Y1+CP
      RO1=BRO*Y1+CRO
  110 U2=BU*Y2+CU
      V2=BV*Y2+CV
      P2=BP*Y2+CP
      RO2=BRO*Y2+CRO
      AD=GAMMA*P2/RO2
      IF (AD.GT.0.0) GO TO 120
      NP=N+NSTART
      WRITE (6,710) NP,L,MDUM,NVC
      IERR=1
      RETURN
  120 A2=SQRT(AD)
  130 CONTINUE
      QUT2=BQUT*Y2+CQUT
      QVT2=BQVT*Y2+CQVT
      QPT2=BQPT*Y2+CQPT
      QROT2=BQROT*Y2+CQROT
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
C
      IF (IWALL.EQ.0.OR.IB.NE.1) GO TO 140
      DU1=BDU*Y1+CDU
      DV1=BDV*Y1+CDV
      DP1=BDP*Y1+CDP
      DRO1=BDRO*Y1+CDRO
      GO TO 150
  140 DU1=DU
      DV1=DV
      DP1=DP
      DRO1=DRO
  150 DU2=BDU*Y2+CDU
      DV2=BDV*Y2+CDV
      DP2=BDP*Y2+CDP
      DRO2=BDRO*Y2+CDRO
C
C     CALCULATE THE PSI TERMS
C
      IF (NDIM.EQ.0) GO TO 180
      IF (IB.EQ.2) GO TO 160
      ATERM2=RO2*V2/(YP-(Y3-Y2)/BE)
      GO TO 180
  160 IF (YCB(L).EQ.0.0) GO TO 170
      ATERM2=RO2*V2/(YCB(L)+Y2/BE)
      GO TO 180
  170 ATERM2=RO2*BE*V(L,2,N1)*DYR
  180 PSI21=-U1*OM1*DU1-OM1*DP1/RO1
      PSI31=-U1*OM1*DV1
      PSI41=-U1*OM1*DP1+A1*A1*U1*OM1*DRO1
      PSI12=-U2*OM1*DRO2-RO2*OM1*DU2-ATERM2
      PSI22=-U2*OM1*DU2-OM1*DP2/RO2
      PSI32=-U2*OM1*DV2
```

```
      PSI42=-U2*OM1*DP2+A2*A2*U2*OM1*DRO2
C
C     CALCULATE THE QUANTITIES FOR THE QUICK SOLVER
C
      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 320
      ALD=AL
      BED=BE
      DED=DE
      OMD=OM2
      YPD=YP
      ILLI=0
      MMO=0
      DO 300 ILL=1,ILLQS
      IF (ILLI.NE.0) GO TO 210
      IF (ILL.NE.1) GO TO 190
      UVAO=(UV3+SIGN*ALS*A2)*DT
      Y200=Y3
      FY3=-UVAO
      Y2=Y(MDUM1)
      GO TO 250
  190 UVAVG=0.5*((U2+U3)*ALAVG+(V2+V3)*BEAVG)+DEAVG
      UVA=(UVAVG+SIGN*ALSA2)*DT
      FY2=Y3-UVA-Y2
      IF (FY2*FY3.LT.0.0) GO TO 200
      UVAO=UVA
      Y200=Y2
      FY3=FY2
      IF (SIGN.LT.0.0) Y2=Y(MDUM+ILL)
      IF (SIGN.GT.0.0) Y2=Y(MDUM-ILL)
      IF (Y2.EQ.Y(MVCB).OR.Y2.EQ.Y(MVCT)) GO TO 200
      GO TO 250
  200 ILLI=1
      Y20=Y2
      GO TO 240
  210 UVAVG=0.5*((U2+U3)*ALAVG+(V2+V3)*BEAVG)+DEAVG
      UVAT=(UVAVG+SIGN*ALSA2)*DT
      FY2=Y3-UVAT-Y2
      FY20=Y3-UVA-Y20
      IF (FY2*FY20.LT.0.0) GO TO 220
      GO TO 230
  220 UVAO=UVA
      Y200=Y20
  230 UVA=UVAT
      Y20=Y2
  240 Y2=Y20+(Y20-Y200)*(Y3-UVA-Y20)/(UVA-UVAO+Y20-Y200)
      IF (Y2.LT.Y(MVCB)) Y2=Y(MVCB)
      IF (Y2.GT.Y(MVCT)) Y2=Y(MVCT)
      IF (Y20.EQ.0.0) GO TO 250
      IF (ABS((Y2-Y20)/Y20).LE.CQS) GO TO 310
C
  250 DO 260 MM=MVCB,MVCT1
      IF (Y2.GE.Y(MM).AND.Y2.LE.Y(MM+1)) GO TO 270
  260 CONTINUE
  270 RDY=(Y2-Y(MM))*DYR
      U2=U(L,MM,N1)+(U(L,MM+1,N1)-U(L,MM,N1))*RDY
      V2=V(L,MM,N1)+(V(L,MM+1,N1)-V(L,MM,N1))*RDY
      P2=P(L,MM,N1)+(P(L,MM+1,N1)-P(L,MM,N1))*RDY
      RO2=RO(L,MM,N1)+(RO(L,MM+1,N1)-RO(L,MM,N1))*RDY
      IF (MM.EQ.MMO) GO TO 280
      MMO=MM
      MMAP=MM
      IP=0
      CALL MAP
      YPMM=YP
      MMAP=MM+1
      IP=1
      CALL MAP
      YPMM1=YP
  280 YP2=YPMM+(YPMM1-YPMM)*RDY
      BEAVG=(Y2-Y3)/(YP2-YPD)
      ALAVG=AL3*BEAVG/BE3
      DEAVG=DE3*BEAVG/BE3
```

```
      A2D=GAMMA*P2/RO2
      IF (A2D.GT.0.0) GO TO 290
      NP=N+NSTART
      WRITE (6,710) NP,L,MDUM,NVC
      IERR=1
      RETURN
  290 ALSA2=SQRT(0.5*(A2D+A3*A3)*(ALAVG*ALAVG+BEAVG*BEAVG))
  300 CONTINUE
      NP=N+NSTART
      WRITE (6,720) ILLQS,NP,L,MDUM,NVC,ICHAR
      IERR=1
      RETURN
  310 AL=ALD
      BE=BED
      DE=DED
      OM2=OMD
      YP=YPD
      MMAP=MDUM
      A2=SQRT(A2D)
  320 IF (ICHAR.EQ.1) GO TO 350
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF (JFLAG.EQ.0) GO TO 330
      IF (IB.NE.1) GO TO 330
      IF (L.EQ.2) GO TO 330
      IF (L.NE.LJET-1) GO TO 330
      GO TO 340
  330 DU3=(U(L+1,MDUM,N3)-U3)*DXR
      DV3=(V(L+1,MDUM,N3)-V3)*DXR
      DP3=(P(L+1,MDUM,N3)-P3)*DXR
      DRO3=(RO(L+1,MDUM,N3)-RO3)*DXR
      GO TO 350
  340 DU3=(U3-U(L-1,MDUM,N3))*DXR
      DV3=(V3-V(L-1,MDUM,N3))*DXR
      DP3=(P3-P(L-1,MDUM,N3))*DXR
      DRO3=(RO3-RO(L-1,MDUM,N3))*DXR
C
C     ENTER THE FREE-JET BOUNDARY ITERATION LOOP
C
  350 YWI(L)=YW(L)
      DO 580 NJ=1,10
      IF (ICHAR.EQ.1) GO TO 450
      IF (JFLAG.LE.0) GO TO 410
      IF (IB.NE.1) GO TO 410
      IF (L.LT.LJET) GO TO 410
      IF (NJ.EQ.1) GO TO 400
      IF (NJ.GT.2) GO TO 380
  360 YWOLD=YW(L)
      POLD=P(L,MDUM,N3)
      IF (P(L,MDUM,N3).LT.PE(MMAX)) GO TO 370
      YW(L)=YW(L)+DELY
      GO TO 390
  370 YW(L)=YW(L)-DELY
      GO TO 390
  380 IF (P(L,MDUM,N3).EQ.POLD) GO TO 360
      DYDP=(YW(L)-YWOLD)/(P(L,MDUM,N3)-POLD)
      YWNEW=YW(L)+DYDP*(PE(MMAX)-P(L,MDUM,N3))
      YWOLD=YW(L)
      POLD=P(L,MDUM,N3)
      YW(L)=YWNEW
  390 IF (YW(L).LT.(1.0-DYW)*YWOLD) YW(L)=(1.0-DYW)*YWOLD
      IF (YW(L).GT.(1.0+DYW)*YWOLD) YW(L)=(1.0+DYW)*YWOLD
  400 NXNY(L)=-(YW(L)-YW(L-1))*DXR
      XWI(L)=(YW(L)-YWI(L))/DT
      XWID=XWI(L)
      CALL MAP
      AL=AL3
      BE=BE3
      DE=DE3
      ALS=SQRT(AL*AL+BE*BE)
C
```

```
C      CALCULATE THE PSI TERMS AT THE SOLUTION POINT
C
  410 IF (NDIM.EQ.0) GO TO 440
      IF (IB.EQ.2) GO TO 420
      ATERM3=RO3*V3/YP
      GO TO 440
  420 IF (YCB(L).EQ.0.0) GO TO 430
      ATERM3=RO3*V3/YCB(L)
      GO TO 440
  430 ATERM3=RO3*BE*V(L,2,N3)*DYR
  440 PSI13=-U3*OM2*DRO3-RO3*OM2*DU3-ATERM3
      PSI23=-U3*OM2*DU3-OM2*DP3/RO3
      PSI33=-U3*OM2*DV3
      PSI43=-U3*OM2*DP3+A3*A3*U3*OM2*DRO3
  450 ABR=NXNY(L)
      IF (IB.EQ.2) ABR=NXNYCB(L)
      IF (IB.EQ.3) ABR=NXNYL(L)
      IF (IB.EQ.4) ABR=NXNYU(L)
      ALB=AL/ALS
      BEB=BE/ALS
      A1B=(A1+A3)*0.5
      A2B=(A2+A3)*0.5
      RO2B=(RO2+RO3)*0.5
      IF (ICHAR.EQ.1) GO TO 460
      PSI21B=(PSI21+PSI23)*0.5+QUT(L,MDUM)
      PSI31B=(PSI31+PSI33)*0.5+QVT(L,MDUM)
      PSI41B=(PSI41+PSI43)*0.5+QPT(L,MDUM)
      PSI12B=(PSI12+PSI13+QROT(L,MDUM)+QROT2)*0.5
      PSI22B=(PSI22+PSI23+QUT(L,MDUM)+QUT2)*0.5
      PSI32B=(PSI32+PSI33+QVT(L,MDUM)+QVT2)*0.5
      PSI42B=(PSI42+PSI43+QPT(L,MDUM)+QPT2)*0.5
      GO TO 470
  460 PSI21B=PSI21+QUT(L,MDUM)
      PSI31B=PSI31+QVT(L,MDUM)
      PSI41B=PSI41+QPT(L,MDUM)
      PSI12B=PSI12+QROT2
      PSI22B=PSI22+QUT2
      PSI32B=PSI32+QVT2
      PSI42B=PSI42+QPT2
  470 IF (IWALL.EQ.0.OR.IB.NE.1) GO TO 520
C
C      SOLVE THE COMPATIBILITY EQUATIONS FOR A CONSTANT PRESSURE
C      INFLOW - OUTFLOW BOUNDARY
C
      ROAA2=SIGN*RO2B*A2B*ALB
      ROAB2=SIGN*RO2B*A2B*BEB
      PSI2T=(PSI42B-QPT2+A2B*A2B*(PSI12B-QROT2)+ROAA2*PSI22B+ROAB2
     1 *PSI32B)*DT
      P(L,MDUM,N3)=PE(MMAX)
      IF (IWALLO.NE.0) P(L,MDUM,N3)=2.0*P(L,MDUM1,N3)-P(L,MDUM1-1,N3)
      IF (ALW.EQ.0.0) GO TO 480
      P(L,MDUM,N3)=(ALW*PE(MMAX)+P2+P(L,MDUM,N1)+ROAB2*(V2-V(L,MDUM,N1))
     1 +ROAA2*(U2-U(L,MDUM,N1))+PSI2T)/(2.0+ALW)
  480 IF (P(L,MDUM,N3).LE.0.0) P(L,MDUM,N3)=PLOW*PC
      IF (Y1.GE.Y3.AND.IWALLO.EQ.0) GO TO 510
      RO(L,MDUM,N3)=RO1+(P(L,MDUM,N3)-P1-(PSI41B-QPT(L,MDUM))*DT)/(A1B
     1 *A1B)+QROT(L,MDUM)*DT
      IF (RO(L,MDUM,N3).LE.0.0) RO(L,MDUM,N3)=ROLOW/G
      PSI1T=(PSI21B-ABR*PSI31B)*DT
      IF (ABR.EQ.0.0) GO TO 490
      ABRT=ABR+1.0/ABR
      V(L,MDUM,N3)=(ABR*V1+V2/ABR+U2-U1-PSI1T+(P2-P(L,MDUM,N3)+PSI2T)
     1 /ROAA2)/ABRT
      GO TO 500
  490 V(L,MDUM,N3)=V2+(P2-P(L,MDUM,N3)+PSI2T)/(RO2B*A2B)
  500 U(L,MDUM,N3)=U1+ABR*(V(L,MDUM,N3)-V1)+PSI1T
      GO TO 700
  510 ND=N1
      IF (ICHAR.EQ.2) ND=N3
      RO(L,MDUM,N3)=0.1*RO(1,MDUM,ND)+0.9*RO(L,MDUM,N1)
      U(L,MDUM,N3)=0.1*U(1,MDUM,ND)+0.9*U(L,MDUM,N1)
      V(L,MDUM,N3)=V2+(-P(L,MDUM,N3)+P2-ROAA2*(U(L,MDUM,N3)-U2)+PSI2T)
     1 /ROAB2
      GO TO 700
```

```
C
C      SOLVE THE COMPATIBILITY EQUATIONS FOR A SOLID BOUNDARY
C
   520 U(L,MDUM,N3)=(U1-ABR*(V1-XWID)+(PSI21B-ABR*PSI31B)*DT)/(1.0+ABR
      1 *ABR)
       V(L,MDUM,N3)=-U(L,MDUM,N3)*ABR+XWID
       IF (NOSL.EQ.0) GO TO 530
       U(L,MDUM,N3)=0.0
       V(L,MDUM,N3)=0.0
       PSI22B=PSI22B-QUT2
       PSI32B=PSI32B-QVT2
   530 P(L,MDUM,N3)=P2-SIGN*RO2B*A2B*(ALB*(U(L,MDUM,N3)-U2)+BEB*(V(L,MDUM
      1 ,N3)-V2))+(PSI42B+A2B*A2B*PSI12B+SIGN*RO2B*A2B*(ALB*PSI22B+BEB
      2 *PSI32B))*DT
       IF (P(L,MDUM,N3).LE.0.0) P(L,MDUM,N3)=PLOW*PC
       RO(L,MDUM,N3)=RO1+(P(L,MDUM,N3)-P1-PSI41B*DT)/(A1B*A1B)
       IF (RO(L,MDUM,N3).LE.0.0) RO(L,MDUM,N3)=ROLOW/G
       IF (IB.EQ.2) GO TO 540
       IF (IB.EQ.3) GO TO 550
       IF (IB.EQ.4) GO TO 560
       IF (TW(1).LT.0.0) GO TO 570
       IF (JFLAG.EQ.1.AND.L.GE.LJET) GO TO 570
       P(L,MDUM,N3)=RM(L,MDUM,N3)*RG*TW(L)
       GO TO 570
   540 IF (TCB(1).LT.0.0) GO TO 570
       P(L,MDUM,N3)=RO(L,MDUM,N3)*RG*TCB(L)
       GO TO 570
   550 IF (TL(1).LT.0.0) GO TO 570
       P(L,MDUM,N3)=RO(L,MDUM,N3)*RG*TL(L)
       GO TO 570
   560 IF (TU(1).LT.0.0) GO TO 570
       P(L,MDUM,N3)=RO(L,MDUM,N3)*RG*TU(L)
C
C      TEST FOR CONVERGENCE OF THE FREE-JET BOUNDARY
C
   570 IF (JFLAG.EQ.0) GO TO 700
       IF (IB.NE.1) GO TO 700
       IF (L.LT.LJET-1) GO TO 700
       IF (L.EQ.LJET-1) GO TO 590
       IF (ICHAR.EQ.1) GO TO 700
       IF (JFLAG.EQ.-1.AND.L.NE.LJET) GO TO 700
       IF (JFLAG.EQ.-1.AND.L.EQ.LJET) GO TO 690
       DELP=ABS((P(L,MDUM,N3)-PE(MMAX))/PE(MMAX))
       IF (DELP.LE.0.001.AND.L.NE.LJET) GO TO 700
       IF (DELP.LE.0.001.AND.L.EQ.LJET) GO TO 690
   580 CONTINUE
       IF (L.EQ.LJET) GO TO 690
       GO TO 700
C
C      SOLVE FOR THE DOWNSTREAM SIDE OF THE WALL EXIT POINT FOR
C      EITHER THE SHARP EXPANSION CORNER CASE, UNDER-EXPANDED
C      FREE-JET CASE OR OVER-EXPANDED FREE-JET CASE
C
   590 UD(3)=U(L,MDUM,N3)
       VD(3)=V(L,MDUM,N3)
       PD(3)=P(L,MDUM,N3)
       ROD(3)=RO(L,MDUM,N3)
       PD(4)=PE(MMAX)
       XM1=SQRT((UD(3)*UD(3)+VD(3)*VD(3))/(GAMMA*PD(3)/ROD(3)))
       DUMD=1.0+GAM2*XM1*XM1
       TD=PD(3)/(ROD(3)*RG)
       TTD=TD*DUMD
       PTD=PD(3)*DUMD**GAM1
C
C      SHARP EXPANSION CORNER CASE
C
       IF (JFLAG.NE.-1) GO TO 630
       B=SQRT(GAM3)
       CC1=XM1*XM1-1.0
       IF (CC1.LT.0.0) CC1=0.0
       PMA1=B*ATAN(SQRT(CC1/(B*B)))-ATAN(SQRT(CC1))
       PMA=ATAN(-NXNY(LJET))-ATAN(-NXNY(LJET-1))
```

```
      PMAD=PMA+PMA1
      XM2=2.0*XM1
      DO 610 I=1,10
      CI=XM2*XM2-1.0
      PMAI=B*ATAN(SQRT(CI/(B*B)))-ATAN(SQRT(CI))
      IF (ABS((PMAI-PMAD)/PMAD).LE.0.0001) GO TO 620
      IF (I.NE.1) GO TO 600
      XMO=XM2
      XM2=0.9*XM2
      PMAO=PMAI
      GO TO 610
  600 DMDA=(XM2-XMO)/(PMAI-PMAO)
      XMO=XM2
      XM2=XM2+DMDA*(PMAD-PMAI)
      PMAO=PMAI
  610 CONTINUE
  620 DUMD=1.0+GAM2*XM2*XM2
      TD=TTD/DUMD
      PD(4)=PTD/DUMD**GAM1
      ROD(4)=PD(4)/(RG*TD)
      GO TO 660
C
C     UNDER-EXPANDED FREE-JET CASE
C
  630 IF (PE(MMAX).GT.PD(3).AND.XM1.GE.1.0) GO TO 640
      ROD(4)=ROD(3)*(PE(MMAX)/PD(3))**(1.0/GAMMA)
      GO TO 650
C
C     OVER-EXPANDED FREE-JET CASE
C
  640 PRD=PE(MMAX)/PD(3)
      ROD(4)=ROD(3)*(GAM3*PRD+1.0)/(PRD+GAM3)
  650 TE=PE(MMAX)/(ROD(4)*RG)
      XM2=SQRT((TTD/TE-1.0)/GAM2)
  660 SS=SQRT(GAMMA*PD(4)/ROD(4))
      VMAG=XM2*SS
      UD(4)=VMAG/SQRT(1.0+NXNY(LJET)*NXNY(LJET))
      VD(4)=-UD(4)*NXNY(LJET)
      IF (JFLAG.EQ.-1) GO TO 700
      IF (XM1.GE.1.0) GO TO 700
C
C     AVERAGE THE 1-SIDED MACH NOS FOR THE INTERIOR POINT CALCULATIONS
C     IF THE UPSTREAM FLOW IS SUBSONIC - FREE-JET CASE
C
      XMB=(XM1+XM2)/2.0
      IF (XMB.GE.1.0) GO TO 670
      DPL=1.0
      DPR=1.0
      GO TO 680
  670 DPL=XM2-1.0
      DPR=1.0-XM1
      XMB=1.0
  680 DPLR=DPR+DPL
      DUM=1.0+GAM2*XMB*XMB
      TEMP=TTD/DUM
      P(L,MDUM,N3)=PTD/DUM**GAM1
      RO(L,MDUM,N3)=P(L,MDUM,N3)/(RG*TEMP)
      AS=GAMMA*P(L,MDUM,N3)/RO(L,MDUM,N3)
      QA=XMB*SQRT(AS)
      DNXNY=(DPR*NXNY(LJET)+DPL*NXNY(L))/DPLR
      U(L,MDUM,N3)=QA/SQRT(1.0+DNXNY*DNXNY)
      V(L,MDUM,N3)=-U(L,MDUM,N3)*DNXNY
      GO TO 700
  690 UD(1)=UD(3)
      VD(1)=VD(3)
      PD(1)=PD(3)
      ROD(1)=ROD(3)
      UD(2)=UD(4)
      VD(2)=VD(4)
      PD(2)=PD(4)
      ROD(2)=ROD(4)
  700 CONTINUE
```

```
      IF (JFLAG.EQ.0) RETURN
      IF (IB.GE.2) RETURN
      IF (ICHAR.EQ.1) RETURN
      U(LJET-1,MMAX,N1)=UOLD
      IF (JFLAG.EQ.-1) RETURN
      YWI(LMAX)=YW(LMAX)
      YW(LMAX)=2.0*YW(L1)-YW(L2)
      NXNY(LMAX)=-(YW(LMAX)-YW(L1))*DXR
      XWI(LMAX)=(YW(LMAX)-YWI(LMAX))/DT
      RETURN
C
C     FORMAT STATEMENTS
C
  710 FORMAT (1H0,61H***** A NEGATIVE SQUARE ROOT OCCURED IN SUBROUTINE
     1WALL AT N=,I6,4H, L=,I2,4H, M=,I2,10H, AND NVC=,I3,6H *****)
  720 FORMAT (1H0,64H***** THE CHARACTERISTIC SOLUTION IN WALL FAILED TO
     1 CONVERGE IN ,I2,17H ITERATIONS AT N=,I6,4H, L=,I2,4H, M=,I2,6H, N
     2VC=,I3,1H,,/7X,10HAND ICHAR=,I1,6H *****)
      END
```

```
      SUBROUTINE INLET
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE INLET
C
C     ****************************************************************
C
*CALL,MCC
      IP=1
      LMAP=1
      LD1=2
      X3=XI
      DXP=XP(2)-X3
      ATERM2=0.0
      ATERM3=0.0
      MIS=1
      MIF=MMAX
      IF (IB.EQ.3) MIF=MDFS
      IF (IB.EQ.4) MIS=MDFS
      IF (IVC.EQ.0) GO TO 10
      IF (NVC.EQ.1) GO TO 10
      IF (MIS.EQ.1) MIS=MVCB
      IF (MIF.EQ.MMAX) MIF=MVCT
      IF (ICHAR.EQ.1.AND.MIF.NE.MMAX) MIF=MIF+1
C
C     BEGIN THE M OR Y DO LOOP
C
   10 DO 400 M=MIS,MIF
      IF (IVC.EQ.0) GO TO 20
      IF (NVC.NE.1) GO TO 20
      IF (M.LT.MVCB) GO TO 20
      IF (M.GT.MVCT) GO TO 20
      IF (ICHAR.NE.1) GO TO 400
      IF (M.EQ.MVCB.AND.MVCB.NE.1) GO TO 20
      GO TO 400
   20 IF (ISUPER.EQ.0) GO TO 70
      IF (ISUPER.EQ.2.AND.IB.EQ.4) GO TO 70
      IF (ISUPER.EQ.3.AND.IB.EQ.3) GO TO 70
      SM=U(1,M,N1)*U(1,M,N1)/(GAMMA*P(1,M,N1)/RO(1,M,N1))
      IF (SM.LT.1.0.AND.IINLET.EQ.0) GO TO 30
      IF (INBC.EQ.0) P(1,M,N3)=PI(M)*PC
      IF (INBC.NE.0) U(1,M,N3)=UI(M)
      UOLD=U(1,M,N3)
      VOLD=V(1,M,N3)
      POLD=P(1,M,N3)
      GO TO 400
C
   30 IF (INBC.NE.0) GO TO 70
      IF (M.EQ.MMAX) GO TO 40
      IF (M.EQ.MDFS.AND.IB.EQ.4) GO TO 50
      IF (M.EQ.MDFS.AND.IB.EQ.3) GO TO 60
      IF (M.NE.1) GO TO 70
      IF (NGCB.EQ.0) GO TO 70
      IF (TCB(1).LT.0.0) GO TO 70
      P(1,M,N3)=TCB(1)*RU(1,M,N3)*RG
      GO TO 400
   40 IF (TW(1).LT.0.0) GO TO 70
      P(1,M,N3)=TW(1)*RO(1,M,N3)*RG
      GO TO 400
   50 IF (TU(1).LT.0.0) GO TO 70
      P(1,M,N3)=TU(1)*RO(1,M,N3)*RG
      GO TO 400
   60 IF (TL(1).LT.0.0) GO TO 70
      P(1,M,N3)=TL(1)*RO(1,M,N3)*RG
      GO TO 400
C
   70 MMAP=M
      CALL MAP
      BED=2.0*BE3*BE4/(BE3+BE4)
      AL34=AL3+AL4
      IF (AL34.EQ.0.0) AL34=1.0
```

```
      ALD=2.0*AL3*AL4/AL34
      U2=U(1,M,N1)
      A2=SQRT(GAMMA*P(1,M,N1)/RO(1,M,N1))
      IF (ICHAR.NE.1) GO TO 90
      IF (ISUPER.EQ.1) GO TO 80
      U(1,M,N3)=U2
      V(1,M,N3)=V(1,M,N1)
   80 A3=A2
C
C     CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
C
   90 QUTB=QUT(1,M)
      QPTB=QPT(1,M)
      QROTB=QROT(1,M)
      IF (IVC.EQ.0) GO TO 100
      IF (M.EQ.MMAX) GO TO 100
      IF (NVC.EQ.1.OR.M.NE.MVCT+1) GO TO 100
C
C     LINEAR INTERPOLATION IN TIME FOR M=MVCT+1
C
      UB=UU1(1)+RIND*(UU2(1)-UU1(1))
      VB=VV1(1)+RIND*(VV2(1)-VV1(1))
      PB=PP1(1)+RIND*(PP2(1)-PP1(1))
      ROB=RORO1(1)+RIND*(RORO2(1)-RORO1(1))
      ULP=UU1(2)+RIND*(UU2(2)-UU1(2))
      VLP=VV1(2)+RIND*(VV2(2)-VV1(2))
      PLP=PP1(2)+RIND*(PP2(2)-PP1(2))
      ROLP=RORO1(2)+RIND*(RORO2(2)-RORO1(2))
      GO TO 110
C
  100 UB=U(1,M,N1)
      VB=V(1,M,N1)
      PB=P(1,M,N1)
      ROB=RO(1,M,N1)
      ULP=U(2,M,N1)
      VLP=V(2,M,N1)
      PLP=P(2,M,N1)
      ROLP=RO(2,M,N1)
  110 BU=(ULP-UB)/DXP
      BV=(VLP-VB)/DXP
      BP=(PLP-PB)/DXP
      BRO=(ROLP-ROB)/DXP
      CU=UB-BU*X3
      CV=VB-BV*X3
      CP=PB-BP*X3
      CRO=ROB-BRO*X3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
      IF (M.EQ.1) GO TO 130
      IF (M.EQ.MDFS.AND.IB.EQ.4) GO TO 140
      IF (IVC.EQ.0) GO TO 120
      IF (NVC.EQ.1.OR.M.NE.MVCB) GO TO 120
C
C     LINEAR INTERPOLATION IN TIME FOR M=MVCB
C
      ULPMM=U(2,M-1,NN1)+RIND*(U(2,M-1,NN3)-U(2,M-1,NN1))
      VLPMM=V(2,M-1,NN1)+RIND*(V(2,M-1,NN3)-V(2,M-1,NN1))
      PLPMM=P(2,M-1,NN1)+RIND*(P(2,M-1,NN3)-P(2,M-1,NN1))
      ROLPMM=RO(2,M-1,NN1)+RIND*(RO(2,M-1,NN3)-RO(2,M-1,NN1))
      UMM=U(1,M-1,NN1)+RIND*(U(1,M-1,NN3)-U(1,M-1,NN1))
      VMM=V(1,M-1,NN1)+RIND*(V(1,M-1,NN3)-V(1,M-1,NN1))
      PMM=P(1,M-1,NN1)+RIND*(P(1,M-1,NN3)-P(1,M-1,NN1))
      ROMM=RO(1,M-1,NN1)+RIND*(RO(1,M-1,NN3)-RO(1,M-1,NN1))
C
      DU=(ULP-ULPMM)(DYR
      DV=(VLP-VLPMM)*DYR
      DP=(PLP-PLPMM)*DYR
      DRO=(ROLP-ROLPMM)*DYR
      DU1=(UB-UMM)*DYR
      DV1=(VB-VMM)*DYR
```

```
      DP1=(PB-PMM)*DYR
      DRO1=(ROB-ROMM)*DYR
      GO TO 150
  120 DU=(ULP-U(2,M-1,N1))*DYR
      DV=(VLP-V(2,M-1,N1))*DYR
      DP=(PLP-P(2,M-1,N1))*DYR
      DRO=(ROLP-RO(2,M-1,N1))*DYR
      DU1=(UB-U(1,M-1,N1))*DYR
      DV1=(VB-V(1,M-1,N1))*DYR
      DP1=(PB-P(1,M-1,N1))*DYR
      DRO1=(ROB-RO(1,M-1,N1))*DYR
      GO TO 150
  130 IF (NGCB.NE.0) GO TO 140
      DU=0.0
      DV=(4.0*V(2,2,N1)-V(2,3,N1))*0.5*DYR
      DP=0.0
      DRO=0.0
      DU1=0.0
      DV1=(4.0*V(1,2,N1)-V(1,3,N1))*0.5*DYR
      DP1=0.0
      DRO1=0.0
      GO TO 150
  140 DU=(U(2,M+1,N1)-ULP)*DYR
      DV=(V(2,M+1,N1)-VLP)*DYR
      DP=(P(2,M+1,N1)-PLP)*DYR
      DRO=(RO(2,M+1,N1)-ROLP)*DYR
      DU1=(U(1,M+1,N1)-UB)*DYR
      DV1=(V(1,M+1,N1)-VB)*DYR
      DP1=(P(1,M+1,N1)-PB)*DYR
      DRO1=(RO(1,M+1,N1)-ROB)*DYR
  150 BDU=(DU-DU1)/DXP
      BDV=(DV-DV1)/DXP
      BDP=(DP-DP1)/DXP
      BDRO=(DRO-DRO1)/DXP
      CDU=DU1-BDU*X3
      CDV=DV1-BDV*X3
      CDP=DP1-BDP*X3
      CDRO=DRO1-BDRO*X3
C
C     CALCULATE THE COEFFICIENTS FOR THE QUICK SOLVER
C
      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 160
      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 160
      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) GO TO 160
      DUDYQ2=0.5*(DUDYQS(2,M,1)+DUDYQS(2,M,2))
      DVDYQ2=0.5*(DVDYQS(2,M,1)+DVDYQS(2,M,2))
      DPDYQ2=0.5*(DPDYQS(2,M,1)+DPDYQS(2,M,2))
      DUDYQ1=0.5*(DUDYQS(1,M,1)+DUDYQS(1,M,2))
      DVDYQ1=0.5*(DVDYQS(1,M,1)+DVDYQS(1,M,2))
      DPDYQ1=0.5*(DPDYQS(1,M,1)+DPDYQS(1,M,2))
      BDUQS=(DUDYQ2-DUDYQ1)/DXP
      BDVQS=(DVDYQ2-DVDYQ1)/DXP
      BDPQS=(DPDYQ2-DPDYQ1)/DXP
      CDUQS=DUDYQ1-BDUQS*X3
      CDVQS=DVDYQ1-BDVQS*X3
      CDPQS=DPDYQ1-BDPQS*X3
C
C     CALCULATE X2
C
  160 IF (ICHAR.NE.1) A3=SQRT(GAMMA*P(1,M,N3)/RO(1,M,N3))
      DO 170 IL=1,2
      X2=X3-((U(1,M,N3)-A3)*OM2+(U2-A2)*OM2)*0.5*DT
      IF (X2-X3.LE.0.05*DXP) X2=X3+0.05*DXP
C
C     INTERPOLATE FOR THE PROPERTIES
C
      U2=BU*X2+CU
      P2=BP*X2+CP
      RO2=BRO*X2+CRO
      A2=SQRT(GAMMA*P2/RO2)
  170 CONTINUE
      V2=BV*X2+CV
```

```
      UV2=U2*AL3+V2*BE3
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
C
      DU2=BDU*X2+CDU
      DV2=BDV*X2+CDV
      DP2=BDP*X2+CDP
      DRO2=BDRO*X2+CDRO
C
      IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 180
      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 180
      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) GO TO 180
      DU2QS=BDUQS*X2+CDUQS
      DV2QS=BDVQS*X2+CDVQS
      DP2QS=BDPQS*X2+CDPQS
C
C     CALCULATE THE PSI TERMS
C
  180 IF (NDIM.EQ.O) GO TO 200
      IF (M.EQ.1.AND.YCB(1).EQ.O.O) GO TO 190
      ATERM2=RO2*V2/YP
      GO TO 200
  190 ATERM2=RO2*BE3*DV2
  200 PSI12=-UV2*DRO2-RO2*AL3*DU2-RO2*BE3*DV2-ATERM2
      PSI22=-UV2*DU2-AL3*DP2/RO2
      PSI42=-UV2*DP2+A2*A2*UV2*DRO2
C
      IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 210
      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 210
      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) GO TO 210
      PSI12=-UV2*DRO2-RO2*ALD*DU2QS-RO2*BED*DV2QS-ATERM2
      UV2=U2*ALD+V2*BED
      PSI22=-UV2*DU2QS-ALD*DP2QS/RO2
  210 IF (ICHAR.EQ.1) GO TO 280
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF (M.EQ.1.AND.NGCB.EQ.O) GO TO 220
      IF (M.EQ.MDFS.AND.IB.EQ.3) GO TO 230
      IF (M.EQ.MMAX) GO TO 230
      DU3=(U(1,M+1,N3)-U(1,M,N3))*DYR
      DV3=(V(1,M+1,N3)-V(1,M,N3))*DYR
      DP3=(P(1,M+1,N3)-P(1,M,N3))*DYR
      DRO3=(RO(1,M+1,N3)-RO(1,M,N3))*DYR
      GO TO 240
  220 DU3=O.O
      DV3=(4.O*V(1,2,N3)-V(1,3,N3))*O.5*DYR
      DP3=O.O
      DRO3=O.O
      GO TO 240
  230 DU3=(U(1,M,N3)-U(1,M-1,N3))*DYR
      DV3=(V(1,M,N3)-V(1,M-1,N3))*DYR
      DP3=(P(1,M,N3)-P(1,M-1,N3))*DYR
      DRO3=(RO(1,M,N3)-RO(1,M-1,N3))*DYR
C
C     CALCULATE THE PSI TERMS AT THE SOLUTION POINT
C
  240 IF (NDIM.EQ.O) GO TO 260
      IF (M.EQ.1.AND.YCB(1).EQ.O.O) GO TO 250
      ATERM3=RO(1,M,N3)*V(1,M,N3)/YP
      GO TO 260
  250 ATERM3=RO(1,M,N3)*BE4*DV3
  260 UV3=U(1,M,N3)*AL4+V(1,M,N3)*BE4
      PSI13=-UV3*DRO3-RO(1,M,N3)*AL4*DU3-RO(1,M,N3)*BE4*DV3-ATERM3
      PSI23=-UV3*DU3-AL4*DP3/RO(1,M,N3)
      PSI43=-UV3*DP3+A3*A3*UV3*DRO3
C
      IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 290
      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 290
      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) GO TO 290
      DUDY1=O.5*(U(1,M+1,N3)-UOLD)*DYR
      DVDY1=O.5*(V(1,M+1,N3)-VOLD)*DYR
```

```
      DPDY1=0.5*(P(1,M+1,N3)-POLD)*DYR
      IF (MDFS.EQ.0) GO TO 270
      IF (M.NE.MDFS+1.OR.LDFSS.NE.1) GO TO 270
      DUDY1=0.5*(U(1,M+1,N3)-UL(1,N3))*DYR
      DVDY1=0.5*(V(1,M+1,N3)-VL(1,N3))*DYR
      DPDY1=0.5*(P(1,M+1,N3)-PL(1,N3))*DYR
  270 PSI13=-UV3*DRO3-RO(1,M,N3)*ALD*DUDY1-RO(1,M,N3)*BED*DVDY1-ATERM3
      UV3=U(1,M,N3)*ALD+V(1,M,N3)*BED
      PSI23=-UV3*DUDY1-ALD*DPDY1/RO(1,M,N3)
      GO TO 290
  280 PSI23=PSI22
      PSI43=PSI42
      PSI13=PSI12
  290 IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 300
      UOLD=U(1,M,N3)
      VOLD=V(1,M,N3)
      POLD=P(1,M,N3)
  300 PSI1B=0.5*(PSI12+PSI13)+QROTB
      PSI2B=0.5*(PSI22+PSI23)+QUTB
      PSI4B=0.5*(PSI42+PSI43)+QPTB
C
C     SOLVE THE COMPATIBILITY EQUATION FOR P OR U
C
      IF (ISUPER.EQ.0) GO TO 340
      IF (ISUPER.EQ.2.AND.IB.EQ.4) GO TO 340
      IF (ISUPER.EQ.3.AND.IB.EQ.3) GO TO 340
      ROAB=0.5*(RO2*A2+RO(1,M,N3)*A3)
      AB=0.5*(A2+A3)
      IF (INBC.NE.0) GO TO 320
      PSIT=(PSI4B-ROAB*(PSI2B-QUTB)+AB*AB*(PSI1B-QROTB))*DT
      IF (ALI.EQ.0.0) GO TO 310
      U(1,M,N3)=(ROAB*ALI*UI(M)+ROAB*(U2+U(1,M,N1))+P(1,M,N1)-P2-PSIT)/
     1 (ROAB*(2.0+ALI))
  310 P(1,M,N3)=P2+ROAB*(U(1,M,N3)-U2)+PSIT
      IF (P(1,M,N3).LE.0.0) P(1,M,N3)=PLOW*PC
      GO TO 400
  320 IF (M.EQ.MMAX.AND.IWALL.NE.0) GO TO 400
      PSIT=(PSI4B-QPTB-ROAB*PSI2B+AB*AB*(PSI1B-QROTB))*DT
      IF (ALI.EQ.0.0) GO TO 330
      P(1,M,N3)=(ALI*PI(M)*PC+ROAB*(U(1,M,N1)-U2)+P2+P(1,M,N1)+PSIT)/(2.
     1 0+ALI)
      IF (P(1,M,N3).LE.0.0) P(1,M,N3)=PLOW*PC
  330 U(1,M,N3)=U2+(P(1,M,N3)-P2-PSIT)/ROAB
      GO TO 400
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR U, V, P, AND RO
C
  340 MN3=SQRT(U(1,M,N3)*U(1,M,N3)+V(1,M,N3)*V(1,M,N3))/A3
      T2=P2/(RO2*RG)
      TTHETA=TAN(THETA(M))
      UCORR=1.0
      IF (NOSLIP.EQ.0) GO TO 350
      IF (M.EQ.MMAX.AND.IWALL.EQ.0) UCORR=0.0
      IF (M.EQ.1.AND.NGCB.NE.0) UCORR=0.0
      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) UCORR=0.0
C
  350 DO 380 ITER=1,20
      DEM=(1.0+GAM2*MN3*MN3)
      P(1,M,N3)=PT(M)/(DEM**GAM1)
      T3=TT(M)/DEM
      IF (M.EQ.MMAX.AND.TW(1).GT.0.0) T3=TW(1)
      IF (M.EQ.1.AND.TCB(1).GT.0.0) T3=TCB(1)
      IF (M.NE.MDFS.OR.LDFSS.NE.1) GO TO 360
      IF (IB.EQ.3.AND.TL(1).GT.0.0) T3=TL(1)
      IF (IB.EQ.4.AND.TU(1).GT.0.0) T3=TU(1)
  360 PAVG=(P2+P(1,M,N3))*0.5
      TAVG=(T2+T3)*0.5
      ROAVG=PAVG/(TAVG*RG)
      AS=GAMMA*PAVG/ROAVG
      U(1,M,N3)=U2+DT*PSI2B+(P(1,M,N3)-P2-(PSI4B+AS*PSI1B)*DT)/(ROAVG
     1 *SQRT(AS))
      U(1,M,N3)=U(1,M,N3)*UCORR
```

```
      V(1,M,N3)=U(1,M,N3)*TTHETA
      OMN3=MN3
      AS=GAMMA*RG*T3
      MN3=SQRT((U(1,M,N3)*U(1,M,N3)+V(1,M,N3)*V(1,M,N3))/AS)
      IF (OMN3.NE.0.0) GO TO 370
      IF (ABS(MN3-OMN3).LE.0.0001) GO TO 390
      GO TO 380
  370 IF (ABS((MN3-OMN3)/OMN3).LE.0.001) GO TO 390
  380 CONTINUE
C
      NP=N+NSTART
      WRITE (6,430) M,NP
  390 RO(1,M,N3)=P(1,M,N3)/(RG*T3)
  400 CONTINUE
      IF (IWALL.NE.0) P(1,MMAX,N3)=PE(MMAX)
C
C     ZERO THE CORNER U FOR THE P,V,RO - NO SLIP BOUNDARY CONDITION CASE
C
      IF (NOSLIP.EQ.0.OR.INBC.EQ.0) RETURN
      IF (ISUPER.EQ.0) RETURN
      IF (ISUPER.EQ.2.AND.IB.EQ.4) RETURN
      IF (ISUPER.EQ.3.AND.IB.EQ.3) RETURN
      IF (NVC.EQ.1.AND.MVCB.EQ.1) GO TO 410
      IF (NGCB.NE.0) U(1,1,N3)=0.0
  410 IF (NVC.EQ.1.AND.MVCT.EQ.MMAX) GO TO 420
      IF (IWALL.EQ.0) U(1,MMAX,N3)=0.0
  420 IF (MDFS.EQ.0) RETURN
      IF (NVC.EQ.1.AND.(MDFS.GT.MVCB.AND.MDFS.LT.MVCT)) RETURN
      U(1,MDFS,N3)=0.0
      RETURN
C
C     FORMAT STATEMENTS
C
  430 FORMAT (1H0,55H***** THE SOLUTION FOR THE ENTRANCE BOUNDARY POINT
     1( 1,,I2,1H,,I6,43H) FAILED TO CONVERGE IN 20 ITERATIONS *****)
      END
```

```
      SUBROUTINE EXITT
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE EXIT
C
C     ****************************************************************
C
*CALL,MCC
      IP=1
      LMAP=LMAX
      LD1=L1
      X3=XE
      DXP=X3-XP(L1)
      ATERM2=0.0
      ATERM3=0.0
      MIS=1
      MIF=MMAX
      SM=0.0
      RNNPE=1.0
      IF (NPE.NE.0) RNNPE=FLOAT(N)/FLOAT(NPE)
      IF (RNNPE.LE.0.0.OR.RNNPE.GT.1.0) RNNPE=1.0
      IF (IB.EQ.3) MIF=MDFS
      IF (IB.EQ.4) MIS=MDFS
      IF (IVC.EQ.0) GO TO 10
      IF (NVC.EQ.1) GO TO 10
      IF (MIS.EQ.1) MIS=MVCB
      IF (MIF.EQ.MMAX) MIF=MVCT
      IF (ICHAR.EQ.1.AND.MIF.NE.MMAX) MIF=MIF+1
C
C     BEGIN THE M OR Y DO LOOP
C
   10 DO 330 M=MIS,MIF
      IF (IVC.EQ.0) GO TO 20
      IF (NVC.NE.1) GO TO 20
      IF (M.LT.MVCB) GO TO 20
      IF (M.GT.MVCT) GO TO 20
      IF (ICHAR.NE.1) GO TO 330
      IF (M.EQ.MVCB.AND.MVCB.NE.1) GO TO 20
      GO TO 330
   20 IF (IEXITT.EQ.1) GO TO 40
      A1D=GAMMA*P(LMAX,M,N1)/RO(LMAX,M,N1)
      IF (A1D.GT.0.0) GO TO 30
      NP=N+NSTART
      WRITE (6,380) NP,M,NVC,ICHAR
      IERR=1
      RETURN
   30 A1=SQRT(A1D)
      IF (IEXITT.EQ.2) GO TO 50
      SM=U(LMAX,M,N1)*U(LMAX,M,N1)/(A1*A1)
      IF (SM.LT.1.0) GO TO 50
   40 U(LMAX,M,N3)=U(L1,M,N3)+FLOAT(IEX)*(U(L1,M,N3)-U(L2,M,N3))
      V(LMAX,M,N3)=V(L1,M,N3)+FLOAT(IEX)*(V(L1,M,N3)-V(L2,M,N3))
      P(LMAX,M,N3)=P(L1,M,N3)+FLOAT(IEX)*(P(L1,M,N3)-P(L2,M,N3))
      RO(LMAX,M,N3)=RO(L1,M,N3)+FLOAT(IEX)*(RO(L1,M,N3)-RO(L2,M,N3))
      UOLD=U(LMAX,M,N3)
      VOLD=V(LMAX,M,N3)
      POLD=P(LMAX,M,N3)
      IF (U(LMAX,M,N3).GE.0.0) GO TO 320
      P(LMAX,M,N3)=RNNPE*PE(M)+(1.0-RNNPE)*PEI
      GO TO 300
C
   50 MMAP=M
      CALL MAP
      BED=2.0*BE3*BE4/(BE3+BE4)
      AL34=AL3+AL4
      DE34=DE3+DE4
      IF (AL34.EQ.0.0) AL34=1.0
      IF (DE34.EQ.0.0) DE34=1.0
      ALD=2.0*AL3*AL4/AL34
      DED=2.0*DE3*DE4/DE34
      U1=U(LMAX,M,N1)
```

```
      U2=U1
      A2=A1
      IF (ICHAR.NE.1) GO TO 60
      U(LMAX,M,N3)=U1
      P(LMAX,M,N3)=P(LMAX,M,N1)
      RO(LMAX,M,N3)=RO(LMAX,M,N1)
      A3=A1
C
C     CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
C
   60 QUTB=QUT(LMAX,M)
      QVTB=QVT(LMAX,M)
      QPTB=QPT(LMAX,M)
      QROTB=QROT(LMAX,M)
      IF (IVC.EQ.0) GO TO 70
      IF (M.EQ.MMAX) GO TO 70
      IF (NVC.EQ.1.OR.M.NE.MVCT+1) GO TO 70
C
C     LINEAR INTERPOLATION IN TIME FOR M=MVCT+1
C
      UB=UU1(LMAX)+RIND*(UU2(LMAX)-UU1(LMAX))
      VB=VV1(LMAX)+RIND*(VV2(LMAX)-VV1(LMAX))
      PB=PP1(LMAX)+RIND*(PP2(LMAX)-PP1(LMAX))
      ROB=RORO1(LMAX)+RIND*(RORO2(LMAX)-RORO1(LMAX))
      ULM=UU1(L1)+RIND*(UU2(L1)-UU1(L1))
      VLM=VV1(L1)+RIND*(VV2(L1)-VV1(L1))
      PLM=PP1(L1)+RIND*(PP2(L1)-PP1(L1))
      ROLM=RORO1(L1)+RIND*(RORO2(L1)-RORO1(L1))
      GO TO 80
C
   70 UB=U(LMAX,M,N1)
      VB=V(LMAX,M,N1)
      PB=P(LMAX,M,N1)
      ROB=RO(LMAX,M,N1)
      ULM=U(L1,M,N1)
      VLM=V(L1,M,N1)
      PLM=P(L1,M,N1)
      ROLM=RO(L1,M,N1)
   80 BU=(UB-ULM)/DXP
      BV=(VB-VLM)/DXP
      BP=(PB-PLM)/DXP
      BRO=(ROB-ROLM)/DXP
      CU=UB-BU*X3
      CV=VB-BV*X3
      CP=PB-BP*X3
      CRO=ROB-BRO*X3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
      IF (M.EQ.1) GO TO 100
      IF (M.EQ.MDFS.AND.IB.EQ.4) GO TO 110
      IF (IVC.EQ.0) GO TO 90
      IF (NVC.EQ.1.OR.M.NE.MVCB) GO TO 90
C
C     LINEAR INTERPOLATION IN TIME FOR M=MVCB
C
      UMM-U(LMAX,M-1,NN1)+RIND*(U(LMAX,M-1,NN3)-U(LMAX,M-1,NN1))
      VMM=V(LMAX,M-1,NN1)+RIND*(V(LMAX,M-1,NN3)-V(LMAX,M-1,NN1))
      PMM=P(LMAX,M-1,NN1)+RIND*(P(LMAX,M-1,NN3)-P(LMAX,M-1,NN1))
      ROMM=RO(LMAX,M-1,NN1)+RIND*(RO(LMAX,M-1,NN3)-RO(LMAX,M-1,NN1))
      ULMMM=U(L1,M-1,NN1)+RIND*(U(L1,M-1,NN3)-U(L1,M-1,NN1))
      VLMMM=V(L1,M-1,NN1)+RIND*(V(L1,M-1,NN3)-V(L1,M-1,NN1))
      PLMMM=P(L1,M-1,NN1)+RIND*(P(L1,M-1,NN3)-P(L1,M-1,NN1))
      ROLMMM=RO(L1,M-1,NN1)+RIND*(RO(L1,M-1,NN3)-RO(L1,M-1,NN1))
C
      DU=(UB-UMM)*DYR
      DV=(VB-VMM)*DYR
      DP=(PB-PMM)*DYR
      DRO=(ROB-ROMM)*DYR
      DU1=(ULM-ULMMM)*DYR
      DV1=(VLM-VLMMM)*DYR
```

```
      DP1=(PLM-PLMMM)*DYR
      DRO1=(ROLM-ROLMMM)*DYR
      GO TO 120
   90 DU=(UB-U(LMAX,M-1,N1))*DYR
      DV=(VB-V(LMAX,M-1,N1))*DYR
      DP=(PB-P(LMAX,M-1,N1))*DYR
      DRO=(ROB-RO(LMAX,M-1,N1))*DYR
      DU1=(ULM-U(L1,M-1,N1))*DYR
      DV1=(VLM-V(L1,M-1,N1))*DYR
      DP1=(PLM-P(L1,M-1,N1))*DYR
      DRO1=(ROLM-RO(L1,M-1,N1))*DYR
      GO TO 120
  100 IF (NGCB.NE.0) GO TO 110
      DU=0.0
      DV=(4.0*V(LMAX,2,N1)-V(LMAX,3,N1))*0.5*DYR
      DP=0.0
      DRO=0.0
      DU1=0.0
      DV1=(4.0*V(L1,2,N1)-V(L1,3,N1))*0.5*DYR
      DP1=0.0
      DRO1=0.0
      GO TO 120
  110 DU=(U(LMAX,M+1,N1)-UB)*DYR
      DV=(V(LMAX,M+1,N1)-VB)*DYR
      DP=(P(LMAX,M+1,N1)-PB)*DYR
      DRO=(RO(LMAX,M+1,N1)-ROB)*DYR
      DU1=(U(L1,M+1,N1)-ULM)*DYR
      DV1=(V(L1,M+1,N1)-VLM)*DYR
      DP1=(P(L1,M+1,N1)-PLM)*DYR
      DRO1=(RO(L1,M+1,N1)-ROLM)*DYR
  120 BDU=(DU-DU1)/DXP
      BDV=(DV-DV1)/DXP
      BDP=(DP-DP1)/DXP
      BDRO=(DRO-DRO1)/DXP
      CDU=DU-BDU*X3
      CDV=DV-BDV*X3
      CDP=DP-BDP*X3
      CDRO=DRO-BDRO*X3
C
C     CALCULATE THE COEFFICIENTS FOR THE QUICK SOLVER
C
      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 130
      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 130
      IF (M.EQ.MDFS.AND.LDFSF.EQ.LMAX) GO TO 130
      DUDYQLX=0.5*(DUDYQS(LMAX,M,1)+DUDYQS(LMAX,M,2))
      DVDYQLX=0.5*(DVDYQS(LMAX,M,1)+DVDYQS(LMAX,M,2))
      DPDYQLX=0.5*(DPDYQS(LMAX,M,1)+DPDYQS(LMAX,M,2))
      DUDYQL1=0.5*(DUDYQS(L1,M,1)+DUDYQS(L1,M,2))
      DVDYQL1=0.5*(DVDYQS(L1,M,1)+DVDYQS(L1,M,2))
      DPDYQL1=0.5*(DPDYQS(L1,M,1)+DPDYQS(L1,M,2))
      BDUQS=(DUDYQLX-DUDYQL1)/DXP
      BDVQS=(DVDYQLX-DVDYQL1)/DXP
      BDPQS=(DPDYQLX-DPDYQL1)/DXP
      CDUQS=DUDYQLX-BDUQS*X3
      CDVQS=DVDYQLX-BDVQS*X3
      CDPQS=DPDYQLX-BDPQS*X3
C
C     CALCULATE X1 AND X2
C
  130 IF (ICHAR.NE.1) A3=SQRT(GAMMA*P(LMAX,M,N3)/RO(LMAX,M,N3))
      DO 140 IL=1,2
      X1=X3-(U(LMAX,M,N3)*OM1+U1*OM1)*0.5*DT
      X2=X3-((U(LMAX,M,N3)+A3)*OM1+(U2+A2)*OM1)*0.5*DT
      IF (X3-X1.LT.0.05*DXP) X1=X3-0.05*DXP
      IF (X3-X2.LT.0.05*DXP) X2=X3-0.05*DXP
C
C     INTERPOLATE FOR THE PROPERTIES
C
      U1=BU*X1+CU
      U2=BU*X2+CU
      P2=BP*X2+CP
      RO2=BRO*X2+CRO
      A2=SQRT(GAMMA*P2/RO2)
```

```
  140 CONTINUE
      V1=BV*X1+CV
      P1=BP*X1+CP
      RO1=BRO*X1+CRO
      UV1=U1*AL3+V1*BE3+DE3
      A1=SQRT(GAMMA*P1/RO1)
      V2=BV*X2+CV
      UV2=U2*AL3+V2*BE3+DE3
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
C
      DV1=BDV*X1+CDV
      DP1=BDP*X1+CDP
      DRO1=BDRO*X1+CDRO
      DU2=BDU*X2+CDU
      DV2=BDV*X2+CDV
      DP2=BDP*X2+CDP
      DRO2=BDRO*X2+CDRO
C
      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 150
      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 150
      IF (M.EQ.MDFS.AND.LDFSF.EQ.LMAX) GO TO 150
      DV1QS=BDVQS*X1+CDVQS
      DP1QS=BDPQS*X1+CDPQS
      DU2QS=BDUQS*X2+CDUQS
      DV2QS=BDVQS*X2+CDVQS
      DP2QS=BDPQS*X2+CDPQS
C
C     CALCULATE THE PSI TERMS
C
  150 IF (NDIM.EQ.0) GO TO 170
      IF (M.EQ.1.AND.YCB(LMAX).EQ.0.0) GO TO 160
      ATERM2=RO2*V2/YP
      GO TO 170
  160 ATERM2=RO2*BE3*DV2
  170 PSI31=-UV1*DV1-BE3*DP1/RO1
      PSI41=-UV1*DP1+A1*A1*UV1*DRO1
      PSI12=-UV2*DRO2-RO2*AL3*DU2-RO2*BE3*DV2-ATERM2
      PSI22=-UV2*DU2-AL3*DP2/RO2
      PSI42=-UV2*DP2+A2*A2*UV2*DRO2
C
      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 180
      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 180
      IF (M.EQ.MDFS.AND.LDFSF.EQ.LMAX) GO TO 180
      UV1=U1*ALD+V1*BED+DED
      PSI31=-UV1*DV1QS-BED*DP1QS/RO1
      PSI12=-UV2*DRO2-RO2*ALD*DU2QS-RO2*BED*DV2QS-ATERM2
      UV2=U2*ALD+V2*BED+DED
      PSI22=-UV2*DU2QS-ALD*DP2QS/RO2
  180 IF (ICHAR.EQ.1) GO TO 270
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF (M.EQ.1.AND.NGCB.EQ.0) GO TO 190
      IF (M.EQ.MDFS.AND.IB.EQ.3) GO TO 200
      IF (M.EQ.MMAX) GO TO 200
      DU3=(U(LMAX,M+1,N3)-U(LMAX,M,N3))*DYR
      DV3=(V(LMAX,M+1,N3)-V(LMAX,M,N3))*DYR
      DP3=(P(LMAX,M+1,N3)-P(LMAX,M,N3))*DYR
      DRO3=(RO(LMAX,M+1,N3)-RO(LMAX,M,N3))*DYR
      GO TO 210
  190 DU3=0.0
      DV3=(4.0*V(LMAX,2,N3)-V(LMAX,3,N3))*0.5*DYR
      DP3=0.0
      DRO3=0.0
      GO TO 210
  200 DU3=(U(LMAX,M,N3)-U(LMAX,M-1,N3))*DYR
      DV3=(V(LMAX,M,N3)-V(LMAX,M-1,N3))*DYR
      DP3=(P(LMAX,M,N3)-P(LMAX,M-1,N3))*DYR
      DRO3=(RO(LMAX,M,N3)-RO(LMAX,M-1,N3))*DYR
C
C     CALCULATE THE PSI TERMS AT THE SOLUTION POINT
```

```
C
  210 IF (NDIM.EQ.O) GO TO 230
      IF (M.EQ.1.AND.YCB(LMAX).EQ.O.O) GO TO 220
      ATERM3=RO(LMAX,M,N3)*V(LMAX,M,N3)/YP
      GO TO 230
  220 ATERM3=RO(LMAX,1,N3)*BE4*DV3
  230 UV3=U(LMAX,M,N3)*AL4+V(LMAX,M,N3)*BE4+DE4
      PSI13=-UV3*DRO3-RO(LMAX,M,N3)*(AL4*DU3+BE4*DV3)-ATERM3
      PSI23=-UV3*DU3-AL4*DP3/RO(LMAX,M,N3)
      PSI33=-UV3*DV3-BE4*DP3/RO(LMAX,M,N3)
      PSI43=-UV3*DP3+A3*A3*UV3*DRO3
C
      IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 250
      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 250
      IF (M.EQ.MDFS.AND.LDFSF.EQ.LMAX) GO TO 250
      DUDY1=O.5*(U(LMAX,M+1,N3)-UOLD)*DYR
      DVDY1=O.5*(V(LMAX,M+1,N3)-VOLD)*DYR
      DPDY1=O.5*(P(LMAX,M+1,N3)-POLD)*DYR
      IF (MDFS.EQ.O) GO TO 240
      IF (M.NE.MDFS+1.OR.LDFSF.NE.LMAX) GO TO 240
      DUDY1=O.5*(U(LMAX,M+1,N3)-UL(LMAX,N3))*DYR
      DVDY1=O.5*(V(LMAX,M+1,N3)-VL(LMAX,N3))*DYR
      DPDY1=O.5*(P(LMAX,M+1,N3)-PL(LMAX,N3))*DYR
  240 PSI13=-UV3*DRO3-RO(LMAX,M,N3)*(ALD*DUDY1+BED*DVDY1)-ATERM3
      UV3=U(LMAX,M,N3)*ALD+V(LMAX,M,N3)*BED+DED
      PSI23=-UV3*DUDY1-ALD*DPDY1/RO(LMAX,M,N3)
      PSI33=-UV3*DVDY1-BED*DPDY1/RO(LMAX,M,N3)
  250 IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 260
      UOLD=U(LMAX,M,N3)
      VOLD=V(LMAX,M,N3)
      POLD=P(LMAX,M,N3)
  260 PSI31B=(PSI31+PSI33)*O.5+QVTB
      PSI41B=(PSI41+PSI43)*O.5
      PSI12B=(PSI12+PSI13)*O.5
      PSI22B=(PSI22+PSI23)*O.5+QUTB
      PSI42B=(PSI42+PSI43)*O.5
      GO TO 280
  270 PSI31B=PSI31+QVTB
      PSI41B=PSI41
      PSI12B=PSI12
      PSI22B=PSI22+QUTB
      PSI42B=PSI42
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR U,V AND RO
C
  280 P(LMAX,M,N3)=RNNPE*PE(M)+(1.O-RNNPE)*PEI
      AB=O.5*(A2+A3)
      ROAVG=O.5*(RO2+RO(LMAX,M,N3))
      PSIT=(PSI42B+ROAVG*AB*PSI22B+AB*AB*PSI12B)*DT
      IF (ALE.EQ.O.O) GO TO 290
      PSIT=PSIT+QPTB*DT
      PSI41B=PSI41B+QPTB
      P(LMAX,M,N3)=(ALE*PE(M)+ROAVG*AB*(U2-U(LMAX,M,N1))+P2+P(LMAX,M,N1)
     1 +PSIT)/(2.O+ALE)
  290 RO(LMAX,M,N3)=RO1+2.O*(P(LMAX,M,N3)-P1-DT*PSI41B)/(A3*A3+A1*A1)
     1 +QROTB*DT
      IF (RO(LMAX,M,N3).LE.O.O) RO(LMAX,M,N3)=ROLOW/G
      U(LMAX,M,N3)=U2+(PSIT-P(LMAX,M,N3)+P2)/(ROAVG*AB)
      V(LMAX,M,N3)=V1+DT*PSI31B
      IF (NOSLIP.EQ.O) GO TO 300
      IF (M.EQ.1.AND.NGCB.NE.O) U(LMAX,M,N3)=O.O
      IF (M.EQ.MMAX.AND.IWALL.EQ.O) U(LMAX,M,N3)=O.O
      IF (M.EQ.MDFS.AND.LDFSF.EQ.LMAX) U(LMAX,M,N3)=O.O
C
C     CHECK FOR INFLOW AND IF SO, SET THE CORRECT BOUNDARY CONDITIONS
C
  300 IF (U(LMAX,M,N3).GE.O.O) GO TO 320
      RO(LMAX,M,N3)=O.5*(RO(LMAX,1,N1)+RO(LMAX,MMAX,N1))
      IF (U(LMAX,2,N1).GT.O.O.AND.U(LMAX,M1,N1).LT.O.O) RO(LMAX,M,N3)=RO
     1 (LMAX,MMAX,N1)
      IF (U(LMAX,2,N1).LT.O.O.AND.U(LMAX,M1,N1).GT.O.O) RO(LMAX,M,N3)=RO
     1 (LMAX,1,N1)
```

```
      V(LMAX,M,N3)=-U(LMAX,M,N3)*(NXNYCB(LMAX)+(YP-YCB(LMAX))/(YW(LMAX)
     1 -YCB(LMAX))*(NXNY(LMAX)-NXNYCB(LMAX)))
      IF (MDFS.EQ.0.OR.LDFSF.NE.LMAX) GO TO 320
      IF (IB.EQ.4) GO TO 310
      RO(LMAX,M,N3)=0.5*(RO(LMAX,1,N1)+RO(LMAX,MDFS,N1))
      IF (U(LMAX,2,N1).GT.0.0.AND.U(LMAX,MDFS-1,N1).LT.0.0) RO(LMAX,M,N3
     1 )=RO(LMAX,MDFS,N1)
      IF (U(LMAX,2,N1).LT.0.0.AND.U(LMAX,MDFS-1,N1).GT.0.0) RO(LMAX,M,N3
     1 )=RO(LMAX,1,N1)
      V(LMAX;M,N3)=-U(LMAX,M,N3)*(NXNYCB(LMAX)+(YP-YCB(LMAX))/(YL(LMAX)
     1 -YCB(LMAX))*(NXNYL(LMAX)-NXNYCB(LMAX)))
      GO TO 320
  310 RO(LMAX,M,N3)=0.5*(RO(LMAX,MDFS,N1)+RO(LMAX,MMAX,N1))
      IF (U(LMAX,MDFS+1,N1).GT.0.0.AND.U(LMAX,M1,N1).LT.0.0) RO(LMAX,M
     1 ,N3)=RO(LMAX,MMAX,N1)
      IF (U(LMAX,MDFS+1,N1).LT.0.0.AND.U(LMAX,M1,N1).GT.0.0) RO(LMAX,M
     1 ,N3)=RO(LMAX,MDFS,N1)
      V(LMAX,M,N3)=-U(LMAX,M,N3)*(NXNYU(LMAX)+(YP-YU(LMAX))/(YW(LMAX)-YU
     1 (LMAX))*(NXNY(LMAX)-NXNYU(LMAX)))
C
C     AVERAGE THE SOLUTION IF THE MACH NUMBER IS ALTERNATING
C     ABOVE AND BELOW 1.0
C
  320 IF (ICHAR.EQ.1.OR.IEXITT.NE.0) GO TO 330
      SM3=U(LMAX,M,N3)**2/(GAMMA*P(LMAX,M,N3)/RO(LMAX,M,N3))
      IF (SM3.LT.1.0.AND.SM.LT.1.0) GO TO 330
      IF (SM3.GT.1.0.AND.SM.GT.1.0) GO TO 330
      P(LMAX,M,N3)=RNNPE*PE(M)+(1.0-RNNPE)*PEI
  330 CONTINUE
C
C     SET BOUNDARY CONDITIONS AT THE CORNER MESH POINTS
C
      IF (IWALL.EQ.0) GO TO 340
      IF (V(LMAX,MMAX,N1).GT.0.0) GO TO 340
      ND=N1
      IF (ICHAR.EQ.2) ND=N3
      U(LMAX,MMAX,N3)=0.1*U(1,MMAX,ND)+0.9*U(LMAX,MMAX,N1)
      RO(LMAX,MMAX,N3)=0.1*RO(1,MMAX,ND)+0.9*RO(LMAX,MMAX,N1)
  340 IF (NVC.EQ.1.AND.MVCT.EQ.MMAX) GO TO 350
      IF (MDFS.NE.0.AND.IB.EQ.3) GO TO 350
      IF (IWALL.EQ.0) V(LMAX,MMAX,N3)=-U(LMAX,MMAX,N3)*NXNY(LMAX)+XWI
     1 (LMAX)
      IF (TW(1).GT.0.0.AND.P(LMAX,MMAX,N3).EQ.PE(MMAX)) RO(LMAX,MMAX,N3)
     1 =P(LMAX,MMAX,N3)/(RG*TW(LMAX))
      IF (TW(1).GT.0.0.AND.P(LMAX,MMAX,N3).NE.PE(MMAX)) P(LMAX,MMAX,N3)
     1 =RO(LMAX,MMAX,N3)*RG*TW(LMAX)
  350 IF (NVC.EQ.1.AND.MVCB.EQ.1) GO TO 360
      IF (MDFS.NE.0.AND.IB.EQ.4) GO TO 360
      V(LMAX,1,N3)= U(LMAX,1,N3)*NXNYCB(LMAX)
      IF (TCB(1).GT 0.0.AND.P(LMAX,1,N3).EQ.PE(1)) RO(LMAX,1,N3)=P(LMAX,
     1 1,N3)/(RG*TCB(LMAX))
      IF (TCB(1).GT.0.0.AND.P(LMAX,1,N3).NE.PE(1)) P(LMAX,1,N3)=RO(LMAX,
     1 1,N3)*RG*TCB(LMAX)
C
C     SET BOUNDARY CONDITIONS FOR THE DUAL FLOW SPACE
C
  360 IF (MDFS.EQ.0.OR.LDFSF.NE.LMAX) RETURN
      IF (NVC.EQ.1.AND.(MDFS.GT.MVCB.AND.MDFS.LT.MVCT)) RETURN
      IF (IB.EQ.4) GO TO 370
      V(LMAX,MDFS,N3)=-U(LMAX,MDFS,N3)*NXNYL(LMAX)
      IF (TL(1).GT.0.0.AND.P(LMAX,MDFS,N3).EQ.PE(MDFS)) RO(LMAX,MDFS,N3)
     1 =P(LMAX,MDFS,N3)/(RG*TL(LMAX))
      IF (TL(1).GT.0.0.AND.P(LMAX,MDFS,N3).NE.PE(MDFS)) P(LMAX,MDFS,N3)
     1 =RO(LMAX,MDFS,N3)*RG*TL(LMAX)
      RETURN
  370 V(LMAX,MDFS,N3)=-U(LMAX,MDFS,N3)*NXNYU(LMAX)
      IF (TU(1).GT.0.0.AND.P(LMAX,MDFS,N3).EQ.PE(MDFS)) RO(LMAX,MDFS,N3)
     1 =P(LMAX,MDFS,N3)/(RG*TU(LMAX))
```

```
      IF (TU(1).GT.0.0.AND.P(LMAX,MDFS,N3).NE.PE(MDFS)) P(LMAX,MDFS,N3)
     1 =RO(LMAX,MDFS,N3)+RG*TU(LMAX)
      RETURN
C
  380 FORMAT (1H0,57H***** A NEG SOUND SPEED OCCURED IN SUBROUTINE EXITT
     1 AT N=,I6,4H, M=,I2,6H, NVC=,I3,11H AND ICHAR=,I1,6H *****)
      END
```

```
      SUBROUTINE QSOLVE
C
C     ***************************************************************
C
C     THIS SUBROUTINE CALCULATES THE VELOCITY AND PRESSURE DERIVATIVES
C     IN THE SUBCYCLED MESH AS PART OF THE QUICK SOLVER PACKAGE
C
C     ***************************************************************
C
*CALL,MCC
      IP=1
      YWB=0.0
      YWT=1.0
      Y1=0.0
      Y2=0.0
      Y10=0.0
      Y20=0.0
      MIS=MVCB1
      MIF=MVCT1
      IF (MDFS.EQ.0) GO TO 20
C
      IB=3
      CALL SWITCH (3)
      GO TO 20
   10 MIS=MDFS+1
      MIF=MVCT1
      IB=4
      YWB=Y(MDFS)
      YWT=1.0
      CALL SWITCH (3)
C
C     BEGIN THE L OR X DO LOOP
C
   20 DO 510 L=1,LMAX
      LMAP=L
      LDFS=0
      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
      YPB=YCB(L)
      YPT=YW(L)
      IF (MDFS.EQ.0) GO TO 50
      IF (LDFS.NE.0) GO TO 30
      IF (IB.EQ.4) GO TO 510
      MIF=MVCT1
      YWT=1.0
      GO TO 50
   30 IF (IB.EQ.4) GO TO 40
      MIF=MDFS-1
      YWT=Y(MDFS)
      YPT=YL(L)
      GO TO 50
   40 YPB=YU(L)
   50 IF (MVCB.NE.1) GO TO 60
      MMAP=1
      MM=1
      RFLD=-2.0*NXNYCB(L)/(1.0+NXNYCB(L)**2)
      GO TO 80
   60 IF (MVCT.NE.MMAX) GO TO 70
      MMAP=MMAX
      MM=MMAX
      RFLD=2.0*NXNY(L)/(1.0+NXNY(L)**2)
      GO TO 80
   70 IF (MDFS.EQ.0) GO TO 110
      IF (LDFS.EQ.0) GO TO 110
      MMAP=MDFS
      MM=MDFS
      IF (IB.EQ.3) RFLD=2.0*NXNYL(L)/(1.0+NXNYL(L)**2)
      IF (IB.EQ.4) RFLD=-2.0*NXNYU(L)/(1.0+NXNYU(L)**2)
   80 CALL MAP
      OM11=2.0*OM1*OM2/(OM1+OM2)
      AL11=AL3
      BE11=BE3
      DE11=DE3
```

```
      ALS11=SQRT(AL11*AL11+BE11*BE11)
      UV11=DE11
      RFLD=RFLD/BE11
      IF (L.EQ.1) GO TO 90
      IF (L.EQ.LMAX) GO TO 100
      PTERM=0.5*OM11*(P(L+1,MM,N1)-P(L-1,MM,N1))*DXR
      ROTERM=0.5*OM11*(RO(L+1,MM,N1)-RO(L-1,MM,N1))*DXR
      QTERM=0.5*OM11*(Q(L+1,MM,N1)-Q(L-1,MM,N1))*DXR
      GO TO 110
   90 PTERM=OM11*(P(2,MM,N1)-P(1,MM,N1))*DXR
      ROTERM=OM11*(RO(2,MM,N1)-RO(1,MM,N1))*DXR
      QTERM=OM11*(Q(2,MM,N1)-Q(1,MM,N1))*DXR
      GO TO 110
  100 PTERM=OM11*(P(LMAX,MM,N1)-P(L1,MM,N1))*DXR
      ROTERM=OM11*(RO(LMAX,MM,N1)-RO(L1,MM,N1))*DXR
      QTERM=OM11*(Q(LMAX,MM,N1)-Q(L1,MM,N1))*DXR
C
C     BEGIN THE M OR Y DO LOOP
C
  110 DO 500 M=MIS,MIF
      MMAP=M
      CALL MAP
      BE=2.0*BE3*BE4/(BE3+BE4)
      BED=BE3
      YPD=YP
      Y3=Y(M)
      YPP=YP+DY/BE4
      YPM=YP-DY/BE3
C
      U3=U(L,M,N1)
      V3=V(L,M,N1)
      P3=P(L,M,N1)
      RO3=RO(L,M,N1)
      Q3=Q(L,M,N1)
      A3=SQRT(GAMMA*P3/RO3)
      UV3=U3*AL3+V3*BE3+DE3
      ALS=SQRT(AL3*AL3+BE3*BE3)
      UV3D=U3*AL4+V3*BE4+DE4
      ALSD=SQRT(AL4*AL4+BE4*BE4)
C
C     CALCULATE Y1 (SECANT - FALSE POSITION METHOD)
C
      ILLI=0
      MMO=0
      DO 270 ILL=1,ILLQS
      IF (ILLI.NE.0) GO TO 150
      IF (ILL.NE.1) GO TO 120
      UVAO=(UV3+ALS*A3)*DT
      Y100=Y3
      FY3=-UVAO
      Y1=Y(M-1)
      GO TO 190
  120 UVAVG=0.5*((U1+U3)*ALAVG+(V1+V3)*BEAVG)+DEAVG
      UVA=(UVAVG+ALSA1)*DT
      FY1=Y3-UVA-Y1
      IF (FY1*FY3.LT.0.0) GO TO 140
      UVAO=UVA
      Y100=Y1
      FY3=FY1
      IF (ILL.LT.M) Y1=Y(M-ILL)
      IF (2+ILL-M.EQ.MMAX+1) GO TO 130
      IF (ILL.GE.M) Y1=2.0*YWB-Y(2+ILL-M)
      GO TO 190
  130 NP=N+NSTART
      WRITE (6,560) NP,L,M,NVC
      IERR=1
      RETURN
  140 ILLI=1
      Y10=Y1
      GO TO 180
  150 UVAVG=0.5*((U1+U3)*ALAVG+(V1+V3)*BEAVG)+DEAVG
      UVAT=(UVAVG+ALSA1)*DT
```

```
      FY1=Y3-UVAT-Y1
      FY10=Y3-UVA-Y10
      IF (FY1*FY10.LT.0.0) GO TO 160
      GO TO 170
  160 UVA0=UVA
      Y100=Y10
  170 UVA=UVAT
      Y10=Y1
  180 Y1=Y10+(Y10-Y100)*(Y3-UVA-Y10)/(UVA-UVA0+Y10-Y100)
      IF (Y1.LT.2.0*YWB-Y(MVCT)) Y1=2.0*YWB-Y(MVCT)
      IF (MVCB.NE.1.AND.Y1.LT.Y(MVCB)) Y1=Y(MVCB)
      IF (Y1.GT.Y(M1)) Y1=Y(M1)
      IF (Y1*Y10.EQ.0.0) GO TO 290
      IF (Y10.EQ.0.0) GO TO 190
      IF (ABS((Y1-Y10)/Y10).LE.CQS) GO TO 290
C
C     INTERPOLATE FOR THE PROPERTIES AT Y=Y1
C
  190 IY1=0
      IF (Y1.GE.YWB) GO TO 200
      Y1=2.0*YWB-Y1
      IY1=1
  200 DO 210 MM=1,M1
      IF (Y1.GE.Y(MM).AND.Y1.LE.Y(MM+1)) GO TO 220
  210 CONTINUE
  220 RDY=(Y1-Y(MM))*DYR
      U1=U(L,MM,N1)+(U(L,MM+1,N1)-U(L,MM,N1))*RDY
      V1=V(L,MM,N1)+(V(L,MM+1,N1)-V(L,MM,N1))*RDY
      P1=P(L,MM,N1)+(P(L,MM+1,N1)-P(L,MM,N1))*RDY
      RO1=RO(L,MM,N1)+(RO(L,MM+1,N1)-RO(L,MM,N1))*RDY
      Q1=Q(L,MM,N1)+(Q(L,MM+1,N1)-Q(L,MM,N1))*RDY
      IF (IY1.EQ.0) GO TO 230
      U1=-U1
      V1=-V1
      RFL=RFLD*(Y1-YWB)
      P1=P1-PTERM*RFL
      RO1=RO1-ROTERM*RFL
      Q1=Q1-QTERM*RFL
  230 IF (MM.EQ.MM0) GO TO 240
      MM0=MM
      MMAP=MM
      IP=0
      CALL MAP
      YPMM=YP
      MMAP=MM+1
      IP=1
      CALL MAP
      YPMM1=YP
  240 YP1=YPMM+(YPMM1-YPMM)*RDY
      IF (IY1.EQ.0) GO TO 250
      Y1=2.0*YWB-Y1
      YP1=2.0*YPB-YP1
  250 IF (YPD.EQ.YP1) GO TO 280
      BEAVG=(Y3-Y1)/(YPD-YP1)
      ALAVG=AL3*BEAVG/BE3
      DEAVG=DE3*BEAVG/BE3
      A1D=GAMMA*P1/RO1
      IF (A1D.GT.0.0) GO TO 260
      NP=N+NSTART
      WRITE (6,520) NP,L,M,NVC
      IERR=1
      RETURN
  260 ALSA1=SQRT(0.5*(A1D+A3*A3)*(ALAVG*ALAVG+BEAVG*BEAVG))
  270 CONTINUE
  280 NP=N+NSTART
      WRITE (6,540) ILLQS,NP,L,M,NVC
      IERR=1
      RETURN
C
C     CALCULATE DUDYQS, DVDYQS AND DPDYQS AT Y=Y1
C
  290 U3D=U3
```

```
      V3D=V3
      P3D=P3
      RO3D=RO3
      Q3D=Q3
      IF (Y1.GE.Y(M-1)) GO TO 300
      U3D=SQS*U3+(1.0-SQS)*(U(L,M-1,N1)+(U(L,M+1,N1)-U(L,M-1,N1))*(YPD
     1 -YPM)/(YPP-YPM))
      V3D=SQS*V3+(1.0-SQS)*(V(L,M-1,N1)+(V(L,M+1,N1)-V(L,M-1,N1))*(YPD
     1 -YPM)/(YPP-YPM))
      P3D=SQS*P3+(1.0-SQS)*(P(L,M-1,N1)+(P(L,M+1,N1)-P(L,M-1,N1))*(YPD
     1 -YPM)/(YPP-YPM))
      RO3D=SQS*RO3+(1.0-SQS)*(RO(L,M-1,N1)+(RO(L,M+1,N1)-RO(L,M-1,N1))*
     1 (YPD-YPM)/(YPP-YPM))
      Q3D=SQS*Q3+(1.0-SQS)*(Q(L,M-1,N1)+(Q(L,M+1,N1)-Q(L,M-1,N1))*(YPD
     1 -YPM)/(YPP-YPM))
  300 RDYD=1.0/((YPD-YP1)*BED)
      DUDYQS(L,M,1)=(U3D-U1)*RDYD
      DVDYQS(L,M,1)=(V3D-V1)*RDYD
      DPDYQS(L,M,1)=(P3D-P1)*RDYD
      DROQDY1=(RO3D*Q3D-RO1*Q1)/((YPD-YP1)*BE)
C
C     CALCULATE Y2 (SECANT - FALSE POSITION METHOD)
C
      ILLI=0
      MMO=0
      DO 460 ILL=1,ILLQS
      IF (ILLI.NE.0) GO TO 340
      IF (ILL.NE.1) GO TO 310
      UVA0=(UV3D-ALSD*A3)*DT
      Y200=Y3
      FY3=-UVA0
      Y2=Y(M+1)
      GO TO 380
  310 UVAVG=0.5*((U2+U3)*ALAVG+(V2+V3)*BEAVG)+DEAVG
      UVA=(UVAVG-ALSA2)*DT
      FY2=Y3-UVA-Y2
      IF (FY2*FY3.LT.0.0) GO TO 330
      UVA0=UVA
      Y200=Y2
      FY3=FY2
      IF (M+ILL.LE.MMAX) Y2=Y(M+ILL)
      IF (MMAX+MMAX-M-ILL.EQ.0) GO TO 320
      IF (M+ILL.GT.MMAX) Y2=2.0*YWT-Y(MMAX+MMAX-M-ILL)
      GO TO 380
  320 NP=N+NSTART
      WRITE (6,570) NP,L,M,NVC
      IERR=1
      RETURN
  330 ILLI=1
      Y20=Y2
      GO TO 370
  340 UVAVG=0.5*((U2+U3)*ALAVG+(V2+V3)*BEAVG)+DEAVG
      UVAT=(UVAVG-ALSA2)*DT
      FY2=Y3-UVAT-Y2
      FY20=Y3-UVA-Y20
      IF (FY2*FY20.LT.0.0) GO TO 350
      GO TO 360
  350 UVA0=UVA
      Y200=Y20
  360 UVA=UVAT
      Y20=Y2
  370 Y2=Y20+(Y20-Y200)*(Y3-UVA-Y20)/(UVA-UVA0+Y20-Y200)
      IF (Y2.GT.2.0*YWT-Y(MVCB)) Y2=2.0*YWT-Y(MVCB)
      IF (MVCT.NE.MMAX.AND.Y2.GT.Y(MVCT)) Y2=Y(MVCT)
      IF (Y2.LT.Y(2)) Y2=Y(2)
      IF (ABS((Y2-Y20)/Y20).LE.CQS) GO TO 480
C
C     INTERPOLATE FOR THE PROPERTIES AT Y=Y2
C
  380 IY2=0
      IF (Y2.LE.YWT) GO TO 390
      Y2=2.0*YWT-Y2
```

```
      IY2=1
  390 DO 400 MM=1,M1
      IF (Y2.GE.Y(MM).AND.Y2.LE.Y(MM+1)) GO TO 410
  400 CONTINUE
  410 RDY=(Y2-Y(MM))*DYR
      U2=U(L,MM,N1)+(U(L,MM+1,N1)-U(L,MM,N1))*RDY
      V2=V(L,MM,N1)+(V(L,MM+1,N1)-V(L,MM,N1))*RDY
      P2=P(L,MM,N1)+(P(L,MM+1,N1)-P(L,MM,N1))*RDY
      RO2=RO(L,MM,N1)+(RO(L,MM+1,N1)-RO(L,MM,N1))*RDY
      Q2=Q(L,MM,N1)+(Q(L,MM+1,N1)-Q(L,MM,N1))*RDY
      IF (IY2.EQ.O) GO TO 420
      U2=-U2
      V2=-V2
      RFL=RFLD*(YWT-Y2)
      P2=P2-PTERM*RFL
      RO2=RO2-ROTERM*RFL
      Q2=Q2-QTERM*RFL
  420 IF (MM.EQ.MMO) GO TO 430
      MMO=MM
      MMAP=MM
      IP=O
      CALL MAP
      YPMM=YP
      MMAP=MM+1
      IP=1
      CALL MAP
      YPMM1=YP
  430 YP2=YPMM+(YPMM1-YPMM)*RDY
      IF (IY2.EQ.O) GO TO 440
      Y2=2.O*YWT-Y2
      YP2=2.O*YPT-YP2
  440 IF (YP2.EQ.YPD) GO TO 470
      BEAVG=(Y2-Y3)/(YP2-YPD)
      ALAVG=AL3*BEAVG/BE3
      DEAVG=DE3*BEAVG/BE3
      A2D=GAMMA*P2/RO2
      IF (A2D.GT.O.O) GO TO 450
      NP=N+NSTART
      WRITE (6,530) NP,L,M,NVC
      IERR=1
      RETURN
  450 ALSA2=SQRT(O.5*(A2D+A3*A3)*(ALAVG*ALAVG+BEAVG*BEAVG))
  460 CONTINUE
  470 NP=N+NSTART
      WRITE (6,550) ILLQS,NP,L,M,NVC
      IERR=1
      RETURN
C
C     CALCULATE DUDYQS, DVDYQS, AND DPDYQS AT Y=Y2
C
  480 U3D=U3
      V3D=V3
      P3D=P3
      RO3D=RO3
      Q3D=Q3
      IF (Y2.LE.Y(M+1)) GO TO 490
      U3D=SQS*U3+(1.O-SQS)*(U(L,M-1,N1)+(U(L,M+1,N1)-U(L,M-1,N1))*(YPD
     1 -YPM)/(YPP-YPM))
      V3D=SQS*V3+(1.O-SQS)*(V(L,M-1,N1)+(V(L,M+1,N1)-V(L,M-1,N1))*(YPD
     1 -YPM)/(YPP-YPM))
      P3D=SQS*P3+(1.O-SQS)*(P(L,M-1,N1)+(P(L,M+1,N1)-P(L,M-1,N1))*(YPD
     1 -YPM)/(YPP-YPM))
      RO3D=SQS*RO3+(1.O-SQS)*(RO(L,M-1,N1)+(RO(L,M+1,N1)-RO(L,M-1,N1))*
     1 (YPD-YPM)/(YPP-YPM))
      Q3D=SQS*Q3+(1.O-SQS)*(Q(L,M-1,N1)+(Q(L,M+1,N1)-Q(L,M-1,N1))*(YPD
     1 -YPM)/(YPP-YPM))
  490 RDYD=1.O/((YP2-YPD)*BED)
      DUDYQS(L,M,2)=(U2-U3D)*RDYD
      DVDYQS(L,M,2)=(V2-V3D)*RDYD
      DPDYQS(L,M,2)=(P2-P3D)*RDYD
      DROQDY2=(RO2*Q2-RO3D*Q3D)/((YP2-YPD)*BE)
      QQT(L,M)=O.5*(DROQDY1+DROQDY2)
```

```
C
  500 CONTINUE
  510 CONTINUE
      IF (MDFS.NE.0.AND.MIS.EQ.MVCB1) GO TO 10
      RETURN
C
C     FORMAT STATEMENTS
C
  520 FORMAT (1H0,63H***** A NEG SOUND SPEED (A1) OCCURED IN SUBROUTINE
     1QSOLVE AT N=,I6,4H, L=,I2,4H, M=,I2,9H AND NVC=,I3,6H *****)
  530 FORMAT (1H0,63H***** A NEG SOUND SPEED (A2) OCCURED IN SUBROUTINE
     1QSOLVE AT N=,I6,4H, L=,I2,4H, M=,I2,9H AND NVC=,I3,6H *****)
  540 FORMAT (1H0,84H***** THE CHARACTERISTIC SOLUTION FOR Y1 IN SUBROUT
     1INE QSOLVE FAILED TO CONVERGE IN ,I2,17H ITERATIONS AT N=,I6,4H, L
     2=,I2,4H, M=,I2,/,7X,6H, NVC=,I3,6H *****)
  550 FORMAT (1H0,84H***** THE CHARACTERISTIC SOLUTION FOR Y2 IN SUBROUT
     1INE QSOLVE FAILED TO CONVERGE IN ,I2,17H ITERATIONS AT N=,I6,4H, L
     2=,I2,4H, M=,I2,/,7X,6H, NVC=,I3,6H *****)
  560 FORMAT (1H0,59H***** THE SOLUTION FOR Y1 FAILED IN SUBROUTINE QSOL
     1VE AT N=,I6,4H, L=,I2,4H, M=,I2,6H, NVC=,I3,6H *****)
  570 FORMAT (1H0,59H***** THE SOLUTION FOR Y2 FAILED IN SUBROUTINE QSOL
     1VE AT N=,I6,4H, L=,I2,4H, M=,I2,6H, NVC=,I3,6H *****)
      END
```

```
CASE NO. 1 - CONVERGING-DIVERGING NOZZLE (45 DEG INLET, 15 DEG EXIT)
 $CNTRL LMAX=21,MMAX=8,NMAX=400,TCONV=0.003    $
 $IVS   $
 $GEMTRY NGEOM=2,XI=0.31,RI=2.5,RT=0.8,XE=4.05,RCI=0.8,RCT=0.5,ANGI=44.88,
 ANGE=15.0   $
 $GCBL   $
 $BC PT=70.0,TT=540.0    $
 $AVL   $
 $RVL   $
 $TURBL   $
 $DFSL   $
 $VCL   $
NASA CASE 1 - MIXING LENGTH MODEL (REUBUSH 3, SOLID SIMULATOR, MACH=0.8)
 $CNTRL LMAX=40,MMAX=25,NMAX=750,NPRINT=-750,NPLOT=250,IPUNCH=1,
 LPP1=15,MPP1=1,LPP2=1,MPP2=2,LPP3=25,MPP3=1,FDT1=0.7    $
 $IVS N1D=0,V=1025*0.0,P=1025*9.45,
 U(1,1,1)=41*0.0,U(1,2,1)=41*396.0,U(1,3,1)=41*509.0,
 U(1,4,1)=41*579.0,U(1,5,1)=41*640.0,U(1,6,1)=41*700.0,
 U(1,7,1)=41*780.0,U(1,8,1)=41*885.0,U(1,9,1)=697*917.0,
 RO(1,1,1)=41*0.04223,RO(1,2,1)=41*0.04300,RO(1,3,1)=41*0.04380,
 RO(1,4,1)=41*0.04421,RO(1,5,1)=41*0.04462,RO(1,6,1)=41*0.04505,
 RO(1,7,1)=41*0.04548,RO(1,8,1)=41*0.04683,RO(1,9,1)=697*0.04730    $
 $GEMTRY NGEOM=1,XI=36.0,XE=72.0,RI=18.0    $
 $GCBL NGCB=4,
 YCB=13*3.0,2.9872,2.9487,2.8844,2.7943,2.6782,2.5357,2.3667,2.1707,
 1.9942,1.8253,1.6695,16*1.53,
 NXNYCB=12*-0.0,0.0064,0.02565,0.0514,0.0772,0.1031,0.1293,0.15575,
 0.1825,0.2096,0.2316,0.2512,0.2672,0.1395,15*-0.0    $
 $BC ISUPER=0,NSTAG=1,PE=9.531,IWALL=1,NOSLIP=1,THETA=25*0.0,
 PT=9.45,10.21,10.74,11.14,11.55,11.96,12.56,13.56,14.38,16*14.5,
 TT=588.95,592.2,593.1,593.7,594.3,594.9,595.8,18*596.1    $
 $AVL NST=1000,SMPT=0.5,SMPTF=0.5,NTST=0,IAV=1    $
 $RVL CMU=0.165E-07,EMU=0.5,CLA=-0.11E-07,ELA=0.5,CK=0.143E-03,EK=0.5    $
 $TURBL ITM=1,IMLM=2    $
 $DFSL   $
 $VCL IST=1,MVCB=1,MVCT=9,IOS=1,
 XP=36.0,37.0,38.0,39.0,40.0,41.0,42.0,43.0,44.0,45.0,46.0,47.0,48.0,
 49.0,50.0,51.0,52.0,53.0,54.0,55.0,56.0,56.8,57.5,58.1,58.61,59.1,
 59.7,60.4,61.2,62.0,63.0,64.0,65.0,66.0,67.0,68.0,69.0,70.0,71.0,72.0,
 YI=3.0,3.0025,3.0075,3.0173,3.0358,3.0700,3.1317,3.2397,3.4232,
 3.7260,4.2105,4.9615,5.98,7.0,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0,
 16.0,17.0,18.0    $
```

```
CASE NO. 6 - TURBULENT PLANE JET IN A PARALLEL STREAM - TWO EQUATION
 $CNTRL LMAX=41,MMAX=17,NMAX=6000,RGAS=287.0,IUI=2,IUO=2,NPLOT=500,
 NPRINT=-6000,FDT=1.0,IPUNCH=1    $
 $IVS N1D=0,U(1,7,1)=779*7.5895,V=1025*0.0,P=1025*101.35,RO=1025*1.2047,
 U(1,1,1)=47.366,47.0,46.5,46.0,45.5,45.0,44.5,44.0,43.5,43.0,42.5,
 42.0,41.5,41.0,40.5,40.0,39.5,39.0,38.5,38.0,37.5,37.0,36.5,36.0,
 35.5,35.0,34.5,34.0,33.5,33.0,32.5,32.0,31.5,31.0,30.5,30.0,29.5,
 29.0,28.5,28.0,27.5,
 U(1,2,1)=47.366,46.5,45.5,44.5,43.5,43.0,42.5,42.0,41.5,41.0,40.5,
 40.0,39.5,39.0,38.5,38.0,37.5,37.0,36.5,36.0,35.5,35.0,34.5,34.0,
 33.5,33.0,32.5,32.0,31.5,31.0,30.5,30.0,29.5,29.0,28.5,28.0,27.5,
 27.0,26.5,26.0,25.5,
 U(1,3,1)=47.366,45.5,43.5,41.5,39.5,39.0,38.5,38.0,37.5,37.0,36.5,
 36.0,35.5,35.0,34.5,34.0,33.5,33.0,32.5,32.0,31.5,31.0,30.5,30.0,
 29.5,29.0,28.5,28.0,27.5,27.0,26.5,26.0,25.5,25.0,24.5,24.0,23.5,
 23.0,22.5,22.0,21.5,
 U(1,4,1)=5*0.0,36*18.0, UL=5*0.0, VL=5*0.0, PL=5*101.35, ROL=5*1.2047,
 U(1,5,1)=5*7.5859,36*15.0,
 U(1,6,1)=5*7.5859,36*11.0    $
 $GEMTRY NDIM=0,NGEOM=1,RI=5.0,XI=-1.9050,XE=38.1    $
 $GCBL    $
 $BC ISUPER=1,PE=101.35,UIL=0.0,VIL=0.0,PIL=101.35,ROIL=1.2047,
 UI=3*47.366,0.0,13*7.5895,VI=17*0.0,PI=17*101.35,ROI=17*1.2047,
 ALI=0.1,ALE=0.1,ALW=0.1,IWALL=1,NOSLIP=1    $
 $AVL IAV=1    $
 $RVL CMU=1.813E-05,CLA=-1.208E-05    $
 $TURBL ITM=3,FSQL=0.0,FSEL=10200.0,
 FSQ=0.0,0.0,4.4,0.0,0.11,12*0.0,
 FSE=0.1,0.1,10200.0,18.4,18.4,12*0.1    $
 $DFSL MDFS=4,LDFSS=1,LDFSF=5,NDFS=2,
 YL=5*0.47625,NXNYL=5*0.0,YU=5*0.47625,NXNYU=5*0.0    $
 $VCL IST=1,
 XP=-1.9050,-1.4288,-0.9525,-0.47625,0.0,0.47625,0.9525,1.4288,1.9050,
 2.3813,2.8816,3.4072,3.9594,4.5395,5.1489,5.7891,6.4617,7.1683,7.9107,
 8.6905,9.5098,10.3704,11.2746,12.2245,13.2224,14.2708,15.3722,16.5292,
 17.7447,19.0217,20.3632,21.7725,23.2531,24.8085,26.4426,28.1592,29.9627,
 31.8573,33.8476,35.9386,38.1,
 YI=0.0,0.15875,0.3175,0.47625,0.635,0.79375,0.9525,1.1375,1.3531,
 1.6042,1.8970,2.2380,2.6355,3.0987,3.6384,4.2673,5.0    $
```


☆U.S. GOVERNMENT PRINTING OFFICE: 1981-0-576-020/153

Printed in the United States of America
Available from
National Technical Information Service
US Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche $3.50 (A01)

Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code	Page Range	Domestic Price	NTIS Price Code
001-025	$ 5.00	A02	151-175	$11.00	A08	301-325	$17.00	A14	451-475	$23.00	A20
026-050	6.00	A03	176-200	12.00	A09	326-350	18.00	A15	476-500	24.00	A21
051-075	7.00	A04	201-225	13.00	A10	351-375	19.00	A16	501-525	25.00	A22
076-100	8.00	A05	226-250	14.00	A11	376-400	20.00	A17	526-550	26.00	A23
101-125	9.00	A06	251-275	15.00	A12	401-425	21.00	A18	551-575	27.00	A24
126-150	10.00	A07	276-300	16.00	A13	426-450	22.00	A19	576-600	28.00	A25
									601-up	†	A99

†Add $1.00 for each additional 25-page increment or portion thereof from 601 pages up.

Los Alamos