

NASA CR-169,103

NASA-CR-169103  
19820020720

# A Reproduced Copy

^ OF

NASA CR-169,103

---

Reproduced for NASA

*by the*

**NASA Scientific and Technical Information Facility**

LIBRARY COPY

OCT 4 1969

LANGLEY RESEARCH CENTER  
LIBRARY NASA  
HAMPTON, VIRGINIA

**PORCTIONS  
OF THIS  
DOCUMENT  
ARE  
ILLEGIBLE**

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

**DISCLAIMER**

This document was prepared as part of the work performed by the Los Alamos National Laboratory for the United States Government under contract number W-7400-ENG-30. The views and opinions contained herein are those of the author and do not necessarily represent those of the United States Government or any agency thereof.

*182-28596 #*

**Los Alamos** Los Alamos National Laboratory  
Los Alamos, New Mexico 87545

RESTRICTION OF THIS DOCUMENT IS UNLIMITED

## CONTENTS

ABSTRACT .....	1
<b>I. THE BASIC METHOD .....</b>	<b>1</b>
A. Introduction .....	1
B. Discussion .....	2
C. Governing Equations .....	2
D. Physical and Computational Flow Spaces .....	8
E. Numerical Method .....	10
F. Comments on the Calculation of Steady Subsonic Flows .....	19
G. Results and Discussion .....	22
<b>II. DESCRIPTION AND USE OF THE VANP2 PROGRAM .....</b>	<b>24</b>
A. Subroutine Description .....	24
B. Computational Grid Description .....	27
C. Input Data Description .....	28
D. Output Description .....	42
E. Computing System Compatibility .....	43
F. Artificial Viscosity Discussion .....	44
G. Sample Calculations .....	44
ACKNOWLEDGMENTS .....	46
REFERENCES .....	45
FIGURES .....	49
APPENDIX. FORTRAN LISTING OF THE VNAP2 PROGRAM .....	91

**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 ( $10^4$ - $10^6$ ) still require fairly long run times (see Sec. 1G). 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<sup>2</sup> 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<sup>3</sup> 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 stabilizes 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

$$\begin{aligned} \frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + \rho \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\epsilon v}{y} \right) \\ = \bar{\alpha} \left[ \frac{\partial}{\partial x} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial y} \right) + \frac{\epsilon \mu_T}{\rho y} \frac{\partial \rho}{\partial y} \right], \end{aligned} \quad (1)$$

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{1}{\rho} \frac{\partial p}{\partial x} = \frac{1}{\rho} \frac{\partial}{\partial x} \left[ (\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} \right] + \frac{1}{\rho} \frac{\partial}{\partial y} \left[ \mu \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right] \\ + \frac{\bar{\alpha}}{\rho} \left[ u \frac{\partial}{\partial x} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial x} \right) + v \frac{\partial}{\partial y} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial x} \right) \right] + \frac{\epsilon}{\rho y} \left[ (\lambda + \mu) \frac{\partial v}{\partial x} + \mu \frac{\partial u}{\partial y} + \frac{\bar{\alpha} \mu_T v}{\rho} \frac{\partial \rho}{\partial x} \right] \\ - \frac{1}{\rho} \frac{2}{3} \frac{\partial \rho q}{\partial x}, \end{aligned} \quad (2)$$

$$\begin{aligned}
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{1}{\rho} \frac{\partial p}{\partial y} = \frac{1}{\rho} \frac{\partial}{\partial y} \left[ (\lambda + 2\mu) \frac{\partial v}{\partial y} + \lambda \frac{\partial u}{\partial x} \right] + \frac{1}{\rho} \frac{\partial}{\partial x} \left[ \mu \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right] \\
+ \frac{\bar{\alpha}}{\rho} \left[ v \frac{\partial}{\partial y} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial y} \right) + u \frac{\partial}{\partial x} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial y} \right) \right] + \frac{\varepsilon}{\rho y} \left[ (\lambda + 2\mu) \left( \frac{\partial v}{\partial y} - \frac{v}{y} \right) + \frac{\bar{\alpha} \mu_T v}{\rho} \frac{\partial \rho}{\partial y} \right] \\
- \frac{1}{\rho} \frac{2}{3} \frac{\partial \rho q}{\partial y} , \tag{3}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} - a^2 \left( \frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} \right) = (\gamma - 1) \left\{ (\lambda_M + 2\mu_M) \left[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 \right] \right. \\
+ \mu_M \left[ \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 \right] + 2\lambda_M \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} + 2\mu_M \frac{\partial v}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) \\
- \bar{\alpha} R T \left[ \frac{\partial}{\partial x} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial y} \right) \right] + \frac{\varepsilon}{y} \left[ (\lambda_M + 2\mu_M) \frac{v^2}{y} + 2\lambda_M v \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right. \\
\left. \left. + k \frac{\partial T}{\partial y} - \frac{\bar{\alpha} R T \mu_T}{\rho} \frac{\partial \rho}{\partial y} \right] + \rho c \right\} , \tag{4}
\end{aligned}$$

and

$$p = \rho R T , \tag{5}$$

where  $\rho$  is the density;  $p$  is the pressure;  $T$  is the temperature;  $u$  and  $v$  are the velocity components;  $q$  is the turbulence energy;  $\varepsilon$  is the turbulence dissipation rate;  $a$  is the speed of sound;  $R$  is the gas constant;  $\mu = \mu_M + \mu_T$ ;  $\lambda = \lambda_M + \lambda_T$ ;  $\mu_M$  and  $\lambda_M$  are the first and second coefficients of molecular viscosity;  $\mu_T$  and  $\lambda_T$  are the corresponding turbulent quantities;  $\gamma$  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  $\varepsilon$  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  $u$ ,  $v$ ,  $p$ ,  $\rho$ ,  $T$ ,  $\mu_T$ ,  $\lambda_T$ , and  $k_T$ . (In the two-equation turbulence model, there are two additional equations for the unknowns  $q$  and  $\varepsilon$ .) To close this set of equations, the turbulence quantities  $\mu_T$ ,  $\lambda_T$  and  $k_T$  need definition. V NAP2 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

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

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

and

$$k_T = \gamma R \mu_T / (\gamma - 1) Pr_T , \quad (8)$$

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

$$l = C_{ML2} \cdot |y_2 - y_1| , \quad (9)$$

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

$$l = C_{ML1} \cdot |y_2 - y_1| , \quad (10)$$

where  $C_{ML1}$  is a constant and

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

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

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

$$\mu_T = \rho l^3 \left| \frac{\partial^2 u}{\partial y^2} \right| . \quad (11)$$

For boundary-layer flows, the Cebeci-Smith<sup>3</sup> two-layer model is used. In the inner layer,  $l$  is defined as

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

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



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

where  $u_E$  is the velocity at the edge of the boundary layer,  $\delta$  is the boundary-layer velocity thickness, and  $\delta^*$  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  $\mu_T$  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  $\lambda_M$  and  $\mu_M$  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  $\lambda$  and  $\mu$ .

**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  $q$ ,

$$\begin{aligned} \frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x} + v \frac{\partial q}{\partial y} &= \frac{2}{3} \frac{q}{\rho} \left( \frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} \right) \\ &+ \frac{\lambda_T + 2\mu_T}{\rho} \left[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 \right] \\ &+ \frac{\mu_T}{\rho} \left[ \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 \right] + \frac{2\lambda_T}{\rho} \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} + \frac{2\mu_T}{\rho} \frac{\partial v}{\partial x} \frac{\partial u}{\partial y} \\ &+ \frac{1}{\rho} \frac{\partial}{\partial x} \left[ \left( \mu_M + \frac{\mu_T}{\sigma_q} \right) \frac{\partial q}{\partial x} \right] + \frac{1}{\rho} \frac{\partial}{\partial y} \left[ \left( \mu_M + \frac{\mu_T}{\sigma_q} \right) \frac{\partial q}{\partial y} \right] - \frac{2\mu_M q \Delta}{\rho S^2} \\ &- \frac{2\bar{a}q}{3\rho} \left[ \frac{\partial}{\partial x} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\mu_T}{\rho} \frac{\partial \rho}{\partial y} \right) \right] + \frac{\epsilon}{y} \left[ \frac{\lambda_T + 2\mu_T}{\rho} \frac{v^2}{y} + \frac{2\lambda_T v}{\rho} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right] \\ &+ \frac{\mu}{\rho} \frac{\partial q}{\partial y} - \frac{2\bar{a}q\mu_T}{\rho^2} \frac{\partial \rho}{\partial y}, \end{aligned} \quad (14)$$

where

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

$$\Delta = \begin{cases} 5 & \text{for } \frac{S\rho\sqrt{2q}}{\mu_M} \leq 5 \\ \frac{S\rho\sqrt{2q}}{\mu_M} & \text{for } \frac{S\rho\sqrt{2q}}{\mu_M} > 5 \end{cases} \quad (16)$$

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

$$\mu_T = \begin{cases} 0.1 C_\mu \frac{\rho^2 S^2 q}{\mu_M} & \text{for } \frac{S\rho\sqrt{2q}}{\mu_M} \leq 5 \\ 0.3534 C_\mu \rho S \sqrt{q} & \text{for } \frac{S\rho\sqrt{2q}}{\mu_M} > 5 \end{cases} \quad (17)$$

where  $C_q$  is 17.2 for planar flows and 12.3 for axisymmetric flows and  $C_\mu = 0.09$ . The quantities  $\lambda_T$  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-Lauder<sup>6-9</sup> 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

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

and

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

$$\begin{aligned}
& + \frac{\varepsilon}{y} \left\{ \frac{C_1 e}{q} \left[ \frac{\lambda_T + 2\mu_T}{\rho} \frac{v^2}{y} + \frac{2\lambda_T v}{\rho} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right] + \frac{1}{\rho} \left( \mu_M + \frac{\mu_T}{\sigma_e} \right) \frac{\partial e}{\partial y} \right. \\
& \left. + \frac{2y\mu_M\mu_T}{\rho^2} \left[ \left( \frac{1}{y} \frac{\partial u}{\partial y} \right)^2 + \left( \frac{1}{y} \frac{\partial v}{\partial y} \right)^2 + \frac{2}{y} \frac{\partial u}{\partial y} \frac{\partial^2 u}{\partial y^2} + \frac{2}{y} \frac{\partial v}{\partial y} \frac{\partial^2 v}{\partial y^2} \right] \right\} , \quad (19)
\end{aligned}$$

where

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

The turbulent viscosity is calculated from

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

where  $C_\mu = 0.09$ . The quantities  $\lambda_T$  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.<sup>10</sup> The second problem is associated with the treatment of the convection terms in Eqs. (18) and (19). In the far field where  $q \rightarrow 0$ , the variations of  $q$  and  $e$  are such in some problems that extremely large values of  $\mu_T$  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):

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

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.<sup>11</sup> The procedure here is first to calculate artificial viscosity coefficients  $\mu_A$ ,  $\lambda_A$  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:

$$\lambda_A = C C_1 \Delta x \Delta y \rho \left| \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \varepsilon \frac{v}{y} \right| , \quad (23)$$

$$\mu_A = C_{\mu 1} \lambda_A / C_1 . \quad (24)$$

and

$$k_A = \gamma R \mu_A / (\gamma - 1) \text{Pr}_A , \quad (25)$$

where  $C$ ,  $C_1$ ,  $C_{\mu 1}$ , and  $Pr_A$  are constants, with  $Pr_A$  representing an artificial Prandtl number, and  $\Delta x$  and  $\Delta y$  are the mesh spacing. The following artificial density smoothing term also is added to the right-hand side of Eq. (1).

$$\text{Equation (1)} = \frac{C_p}{\rho} \left[ \frac{\partial}{\partial x} \left( \mu_A \frac{\partial \rho}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu_A \frac{\partial \rho}{\partial y} \right) + \frac{\epsilon \mu_A}{y} \frac{\partial \rho}{\partial y} \right], \quad (26)$$

where  $C_p$  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  $\zeta, \eta, \tau$  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:

$$\bar{\zeta} = x; \bar{\eta} = \frac{y - y_c}{y_w - y_c}; \bar{\tau} = t, \quad (27)$$

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  $\zeta, \eta, \tau$  space by the following transformation:

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

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

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

where

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

The derivatives  $d\bar{\zeta}/d\bar{\zeta}$  and  $d\bar{\eta}/d\bar{\eta}$  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  $\Delta y$  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

$$\left. \begin{aligned} \bar{\zeta} = x; \quad \bar{\eta} = c \frac{y - y_c}{y_L - y_c}; \quad \bar{\tau} = t \quad \text{for } y_c \leq y \leq y_L. \\ \bar{\zeta} = x; \quad \bar{\eta} = c + (1 - c) \frac{y - y_L}{y_w - y_L}; \quad \bar{\tau} = t \quad \text{for } y_L \leq y \leq y_w. \end{aligned} \right\} \quad (31)$$

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  $(y_U - y_c)/(y_w - y_U + y_L - y_c)$  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

$$\begin{aligned}
\omega &= \frac{d\zeta}{d\zeta}; \quad \beta = \frac{d\eta}{d\bar{\eta}} \frac{c}{y_L - y_c}; \quad \alpha = \frac{\beta}{c} \left[ (\bar{\eta} - c) \frac{dy_c}{dx} - \bar{\eta} \frac{dy_L}{dx} \right]; \quad \delta = 0 \\
&\text{for } y_c \leq y \leq y_L, \\
&\text{and} \\
\omega &= \frac{d\zeta}{d\zeta}; \quad \beta = \frac{d\eta}{d\bar{\eta}} \frac{1-c}{y_w - y_u}; \quad \alpha = \frac{\beta}{1-c} \left[ (\bar{\eta} - 1) \frac{dy_u}{dx} - (\bar{\eta} - c) \frac{\partial y_w}{\partial x} \right]; \\
\delta &= \frac{\beta(\bar{\eta} - c)}{1-c} \frac{\partial y_w}{\partial t} \quad \text{for } y_u \leq y \leq y_w.
\end{aligned} \tag{32}$$

3. **Transformed Governing Equations.** Using Eqs. (27) and (29), the original governing equation can be written in the  $\zeta, \eta, \tau$  variables. For example Eq. (1) becomes

$$\begin{aligned}
\frac{\partial \rho}{\partial \tau} + u\omega \frac{\partial \rho}{\partial \zeta} + \bar{v} \frac{\partial \rho}{\partial \eta} + \rho \left( \omega \frac{\partial u}{\partial \zeta} + \alpha \frac{\partial u}{\partial \eta} + \beta \frac{\partial v}{\partial \eta} + \frac{\epsilon v}{y} \right) \\
= \frac{\bar{\alpha}}{\rho} \left\{ \left( \omega \frac{\partial}{\partial \zeta} + \alpha \frac{\partial}{\partial \eta} \right) \left[ \mu_\tau \left( \omega \frac{\partial \rho}{\partial \zeta} + \alpha \frac{\partial \rho}{\partial \eta} \right) \right] + \beta \frac{\partial}{\partial \eta} \left( \mu_\tau \beta \frac{\partial \rho}{\partial \eta} \right) + \epsilon \frac{\mu_\tau \beta}{y} \frac{\partial \rho}{\partial \eta} \right\}. \tag{33}
\end{aligned}$$

where

$$\bar{v} = u\alpha + v\beta + \delta, \tag{34}$$

$$\begin{aligned}
y &= y_c + \bar{\eta} (y_w - y_c) && \text{for the single-flow space,} \\
y &= y_c + \frac{\bar{\eta}}{c} (y_L - y_c) && \text{for the lower dual-flow space,} \\
y &= y_u + \frac{\bar{\eta} - c}{1-c} (y_w - y_u) && \text{for the upper dual-flow space,}
\end{aligned} \tag{35}$$

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 ( $\epsilon = 0$ ), laminar ( $\bar{\alpha} = q = 0$ ) flow are

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

for the first step and

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

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  $\Delta t$ . 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  $t + \Delta t$  are known from the coarse grid solution, so that the values between  $t$  and  $t + \Delta t$  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<sup>12</sup> and MacCormack.<sup>13</sup> 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

$$\frac{\partial p}{\partial t} + v \frac{\partial p}{\partial y} + \rho \frac{\partial v}{\partial y} = 0, \quad (38)$$

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

and

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



where  $v$  is the velocity,  $\rho$  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  $\Delta t \leq \Delta y / (|v| + a)$ . However, to improve the computational efficiency in the boundary layers, where  $\Delta y$  and  $v$  are small but  $a$  is large, a procedure that allows  $\Delta t \leq \Delta y / v$  is developed. Writing Eqs. (38)-(40) in characteristic form yields

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

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

and

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

Therefore, Eq. (41) applies along the flow streamline and Eqs. (42) and (43) apply along Mach lines. Thus, if a time step  $\Delta t \leq \Delta y / |v|$  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<sup>14</sup> 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

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

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

and

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

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

$$\frac{\partial \bar{v}}{\partial y} = \frac{[C_3 v_M^N + (1 - C_3)(v_{M+1}^N + v_{M-1}^N)/2] - v_1}{y_M - y_1} \quad (47)$$

on the predictor step and

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

on the corrector step. The hat derivatives are calculated by

$$\frac{\partial \hat{v}}{\partial y} = \frac{v_2 - [C_3 v_M^N + (1 - C_3)(v_{M+1}^N + v_{M-1}^N)/2]}{y_2 - y_M} \quad (49)$$

on the predictor step and

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

on the corrector step. The coefficient  $C_3$  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  $\zeta = \text{constant}$  reference-plane-characteristic scheme used at the wall boundary. The Mach line compatibility equations, without the viscous and  $\zeta$  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  $\lambda$  scheme<sup>15</sup> 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  $\lambda$  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  $\rho$  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  $\theta$  as proposed by Serra.<sup>16</sup> The second and third, which are discussed by Olinger and Sundström,<sup>17</sup> specify either  $u$ ,  $v$ , and  $\rho$  or  $p$ ,  $v$ , and  $\rho$ . For a discussion of the relative merits of these boundary conditions, see Sec. I.F. Following the ideas of Moretti and Abbett,<sup>18</sup> 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  $\eta$  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  $\eta$  are computed as in the interior point scheme and the derivatives with respect to  $\zeta$  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  $\eta = \text{constant}$  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

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

where the first equation is called the compatibility equation and the second is called the characteristic curve equation. The  $\psi$  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 characteristic 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  $\theta$  boundary condition, the following equations that relate the stagnation or total conditions to the static conditions are required.

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

and

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

where  $\gamma$  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),  $\rho$  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  $\rho$ , and the process is continued until the change in  $M$  has converged to  $10^{-3}$ .

For the  $u$ ,  $v$ , and  $\rho$  boundary condition, there is only one unspecified variable,  $p$ , which can be calculated from Eq. (51). Likewise, the  $p$ ,  $v$ , and  $\rho$  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  $\rho$  and the  $p$ ,  $v$ , and  $\rho$  boundary conditions include a nonreflecting option based on the ideas of the outflow boundary condition of Rudy and Strikwerda.<sup>19</sup> Rudy and Strikwerda use the following equation

$$\frac{\partial p}{\partial t} - \rho a \frac{\partial u}{\partial t} + C_a(p - p_e) = 0 \quad (54)$$

to replace the outflow boundary condition  $p = p_e$ , where  $p_e$  is the exit pressure and  $C_a$  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  $\rho$  boundary condition, Eq. (54) becomes

$$\frac{\partial p}{\partial t} + \rho a \frac{\partial u}{\partial t} + C_a(p - p_i) = 0, \quad (55)$$

where  $p_i$  is the specified inflow pressure. For the  $u$ ,  $v$ , and  $\rho$  boundary conditions, Eq. (54) becomes

$$\frac{\partial u}{\partial t} + \frac{1}{\rho a} \frac{\partial p}{\partial t} + C_a(u - u_i) = 0, \quad (56)$$

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  $\rho$  or the  $p$ ,  $v$ , and  $\rho$  boundary conditions (but not the  $p_T$ ,  $T_T$ , and  $\theta$  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

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

where  $|\partial u / \partial y|$  and  $\rho$  can be determined from the inflow velocity profile and  $\mu_T$  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

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

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  $\mu_T$  constant, produces little change in the solution. If the  $p_T$ ,  $T_T$ , and  $\theta$  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  $\mu_T$  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

$$dp - a^2 d\rho = \psi_1 dt \quad \left. \vphantom{dp - a^2 d\rho = \psi_1 dt} \right\} \quad \text{for } d\zeta = \omega dt \quad (57)$$

$$dv = \psi_2 dt \quad \left. \vphantom{dv = \psi_2 dt} \right\} \quad \text{for } d\zeta = \omega dt \quad (58)$$

and

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

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  $\rho$  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  $\rho$  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.<sup>19</sup> 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  $\zeta$  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  $\eta$  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  $\zeta = \text{constant}$  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

$$\left. \begin{aligned} \beta du - \alpha dv &= (\beta \psi_2 - \alpha \psi_3) d\tau \\ dp - a^2 d\rho &= \psi_4 d\tau \end{aligned} \right\} \text{for } d\eta = \bar{v} d\tau \quad (60)$$

and

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

where

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

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  $\theta$  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  $\zeta$  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  $\zeta$  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  $\rho$  is calculated from Eq. (61). For inflow,  $u$  and  $\rho$  are specified and  $v$  is calculated from Eq. (62). The actual values of  $u$  and  $\rho$  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

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

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

Equation (63) becomes

$$v = u \tan \theta . \quad (65)$$

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  $\Delta t$  is determined by

$$\Delta t = \min(\Delta t_x, \Delta t_y) , \quad (66)$$

where

$$\Delta t_x = A / (|u| + a) / \Delta x + \mu / A_1 \rho \Delta x^2 \quad (67)$$

and

$$\Delta t_y = A / (|v| + a) / \Delta y + \mu / A_1 \rho \Delta y^2 , \quad (68)$$

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

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

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 computational 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  $\theta$  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  $\theta$  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<sup>19</sup> nonreflecting outflow boundary condition. The coefficient  $C_a$  (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  $\rho$  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  $C_a$  (ALI in Namelist BC) in Eq. (56) equals 0.1. The top curve of Fig. 8 shows that the  $u$ ,  $v$ , and  $\rho$  boundary condition trapped the



initial disturbances in the computational grid. The Rudy and Strikwerda<sup>19</sup> 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  $\rho$  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  $\rho$  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  $\theta$  inflow boundary condition is preferred over the  $u$ ,  $v$ , and  $\rho$  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  $\theta$  inflow boundary condition is also preferred, because the mass flow is usually not known in advance. If the  $u$ ,  $v$ , and  $\rho$  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  $\rho$  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  $\rho$  to be specified; however, the code could easily be modified to allow time-dependent functions for  $u$ ,  $v$ , and  $\rho$ . The  $u$ ,  $v$ , and  $\rho$  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  $\rho$  boundary condition is an improvement. The test cases run to date indicate that the  $u$ ,  $v$ , and  $\rho$  boundary condition produces results more consistent with the supersonic part of the flow than does the  $p_T$ ,  $T_T$ , and  $\theta$  boundary condition. As a result, VNAP2 allows only the  $u$ ,  $v$ , and  $\rho$  option at subsonic parts of a mixed subsonic/supersonic inflow.

The  $u$ ,  $v$ , and  $\rho$  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  $u$ ,  $v$ , and  $\rho$  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  $u$ ,  $v$ , and  $\rho$  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  $\theta$  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<sup>20-22</sup> on the  $u$ ,  $v$ , and  $\rho$  subsonic inflow boundary condition. Moretti feels that the  $u$ ,  $v$ , and  $\rho$  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  $\rho$  boundary condition satisfies the characteristic compatibility conditions, as does the  $p_T$ ,  $T_T$ , and  $\theta$  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 non-physical reflections at subsonic boundaries. The  $u$ ,  $v$ , and  $\rho$  boundary condition causes a reflection that has approximately the same amplitude, whereas the  $p_T$ ,  $T_T$ , and  $\theta$  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  $\rho$  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  $\rho$  or  $p_T$ ,  $T_T$ , and  $\theta$  are generally not known in advance. (For the special case of inviscid, steady flow,  $p_T$  and  $T_T$ , but not  $\theta$ , 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<sup>23</sup> 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<sup>23</sup> 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  $\theta$  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.<sup>26</sup> 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 continued 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  $\theta$  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  $10.5 \times 10^6$ . 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<sup>28</sup> 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  $\rho$  boundary condition, with the non-reflecting 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  $3.0 \times 10^4$ . 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 GEOM, GEOMCB, and GEOMLU to calculate the geometry. If requested, program VNAP2 calls

sub-routine  $\emptyset$ NEDIM to calculate the ID, initial-value surface. Program VNAP2 then prints the initial-value surface, which includes a mass flow and momentum thrust calculation made by subroutine MASFL $\emptyset$ . Next, subroutine PL $\emptyset$ T 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 VIS $\emptyset$ US to calculate the artificial, molecular, and turbulent viscosity-heat conduction terms; calls subroutine QS $\emptyset$ LVE 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 SM $\emptyset$ OTH to smooth the solution; calls subroutine MASFL $\emptyset$  to compute the mass flow and momentum thrust; prints the solution surface; calls subroutine PL $\emptyset$ T 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 GE $\emptyset$ M. Subroutine GE $\emptyset$ M 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 GE $\emptyset$ M then calls subroutine MTLUP, which interpolates for the coordinates. Next, subroutine GE $\emptyset$ M calls function DIF, which calculates the slopes of the coordinates. For the second tabular wall, the coordinates and slopes are read in.

3. Subroutine GE $\emptyset$ MCB. Subroutine GE $\emptyset$ MCB calculates the centerbody coordinates and slopes for four different centerbody geometries and is similar to subroutine GE $\emptyset$ M.

4. Subroutine GE $\emptyset$ MLU. Subroutine GE $\emptyset$ MLU 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 GE $\emptyset$ M.

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 GE $\emptyset$ M, GE $\emptyset$ MCB, and GE $\emptyset$ MLU 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 GE $\emptyset$ M, GE $\emptyset$ MCB, and GE $\emptyset$ MLU to calculate the slopes of the wall, centerbody, and dual-flow-space wall coordinates.

7. Subroutine  $\emptyset$ NEDIM. Subroutine  $\emptyset$ NEDIM is called by program VNAP2 to compute the ID, 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 MASFL $\emptyset$ . Subroutine MASFL $\emptyset$  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 PL $\emptyset$ T. Subroutine PL $\emptyset$ T 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  $0.9 \Delta x$ , where  $\Delta x$  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 ≤ 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<sup>3</sup> or kg/m<sup>3</sup>), 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 SMOOTH. Subroutine SMOOTH 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  $u_n$ ,  $v_n$ , and  $p_n$ , 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  $u_n$ ,  $v_n$ , and  $p_n$  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  $\zeta$  and  $\eta$  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 ( $M = \text{MDFS}$  and  $L$  between  $\text{LDFSS}$  and  $\text{LDFSF}$ ). 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  $M = \text{MDFS}$  and  $L$  between  $\text{LDFSS}$  and  $\text{LDFSF}$ . For reading in initial data values, the values in  $UL$ ,  $VL$ ,  $PL$ ,  $RØL$ ,  $QL$ , and  $EL$  arrays correspond to the lower dual-flow-space wall, whereas values in the  $U$ ,  $V$ ,  $P$ ,  $RØ$ ,  $Q$ , and  $E$  arrays for  $M = \text{MDFS}$  and  $L$  between  $\text{LDFSS}$  and  $\text{LDFSF}$  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  $\text{MVCB} - 1$  and from  $M = \text{MVCT}$

+ 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 default 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



	<p>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.</p>
FDT1	<p>The same as FDT, except it applies only in the subcycled part of the mesh. That is, FDT1 is used from <math>M = MVCB</math> to <math>M = MVCT</math> (see Namelist VCL). The default value is 1.0.</p>
VDT	<p>The parameter <math>A_1</math> 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.</p>
VDT1	<p>The same as VDT, except it applies only in the subcycled part of the mesh. That is, VDT1 is used from <math>M = MVCB</math> to <math>M = MVCT</math> (see Namelist VCL). The default value is 0.25, although values larger than 1.0 have been used in free-shear layers.</p>
GAMMA	<p>Denotes the ratio of specific heats. The default value is 1.4.</p>
RGAS	<p>Denotes the gas constant in <math>\text{lb-ft/lbm-}^\circ\text{R}</math> if English units are used, or <math>\text{J/kg-K}</math> if metric units are used. The default value is 53.35.</p>
TSTØP	<p>Specifies the physical time, in seconds, at which the computations will be stopped. The default value is 1.0.</p>
IUI	<p>An integer specifying the type of units to be used for the input quantities. If <math>IUI = 1</math>, English units are assumed; if <math>IUI = 2</math>, metric units are assumed. In using any default values, make sure the values correspond to the proper units. The default value is 1.</p>
IUØ	<p>The same as IUI except for output quantities. If <math>IUØ = 3</math>, both English and metric units are printed. The default value is 1.</p>
IPUNCH	<p>An integer which, if nonzero, punches (writes) the last solution plane on cards (disc or tape) for restart. The default value is 0.</p>
NPLOT	<p>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 <math>NPLOT = N</math>, all <math>N</math>th solution planes, plus the initial-data and final solution plane, are plotted. For <math>NPLOT = 0</math>, only the final solution plane is plotted. The default value is -1.</p>
LPP1,MPP1 LPP2,MPP2 LPP3,MPP3	<p>Three sets of integers that specify three grid points (the first point is <math>L = LPP1</math>, <math>M = MPP1</math>) for which the pressure is printed at each time step. When <math>MPP1(MPP2</math> or <math>MPP3) = MDFS \neq 0</math> (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 <math>LPP1 &lt; 0</math>, the pressure at each subcycled grid point (see <math>MVCB</math> and <math>MVCT</math> in Namelist VCL) is also printed. The default values are 0 (no printing).</p>

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	<p>An integer specifying which part of the flow field is tested for steady-state convergence. For <math>NASM = 0</math>, the entire flow field is tested. For <math>NASM = 1</math>, the transonic and supersonic (throat region to exit) regions are tested. The default value is 1.</p>
NAME	<p>An integer that, when nonzero, causes the 10 namelists to be printed in addition to the regular output. The default value is 0.</p>
NCØNVI	<p>An integer specifying how many times the convergence tolerance <math>TCØNV</math> must be satisfied on consecutive time steps before the solution is considered to have converged. The default value is 1.</p>

- IUNIT** An integer that, when equal to zero, causes the program to use either English or metric units (see IUI and IU $\theta$ ). For IUNIT = 1, a nondimensional set of units is used. The default value is 0.
- PL $\theta$ W** If the pressure becomes negative during a calculation, it is set equal to PL $\theta$ W in psia or kPa. The default value is 0.01.
- R $\theta$ L $\theta$ W** If the density becomes negative during a calculation, it is set equal to R $\theta$ L $\theta$ W in lb/ft<sup>3</sup> or kg/m<sup>3</sup>. The default value is 0.0001.
- IVPTS** An integer that controls the scaling of the velocity vector plots. IVPTS = 1 produces one plot with the maximum vector equal to 0.9  $\Delta x$ , where  $\Delta x$  is the average value. IVPTS = 2 produces the above plot and a second plot where the maximum vector is 1.9  $\Delta x$ , and so on. The default value is 1.

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

- NID** An integer specifying the type of initial-data surface desired. For NID = 0, a 2D initial-data surface is read in. A value of U, V, P, and R $\theta$  (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 R $\theta$ L (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 NID = 0 or NID  $\neq$  0. For NID  $\neq$  0, a 1D data surface is computed internally.

The following combinations are possible:

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

The default value is 1.

- U(L,M,1)** An array denoting the x-direction velocity component in ft/s or m/s. For NID = 0, U(L,M,1) must be read in for cases from L = 1 to LMAX and from M = 1 to MMAX. For NID  $\neq$  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.
- R $\theta$ (L,M,1)** An array denoting the density in lbm/ft<sup>3</sup> or kg/m<sup>3</sup>. 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<sup>2</sup>/s<sup>2</sup> or m<sup>2</sup>/s<sup>2</sup>. 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 rate in ft<sup>2</sup>/s<sup>3</sup> or m<sup>2</sup>/s<sup>3</sup>. See 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 NID = 0 and MDFS  $\neq$  0, UL(L,1) must be read in for cases from L = LDFSS to LDFSF. For NID  $\neq$  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 <sup>3</sup> or kg/m <sup>3</sup> . See UL(L,1) for additional information. No default values are specified.
QL(L,1)	An array denoting the turbulence energy in ft <sup>2</sup> /s <sup>2</sup> or m <sup>2</sup> /s <sup>2</sup> . See UL(L,1) for additional information. The default value is FSQI in Namelist TURBL.
EL(L,1)	An array denoting the dissipation rate in ft <sup>2</sup> /s <sup>3</sup> or m <sup>2</sup> /s <sup>3</sup> . See UL(L,1) for additional information. The default value is FSEL in Namelist TURBL.
RSTAR, RSTARS	If NID = -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 $\pi$ that is the radius squared (in in. <sup>2</sup> or cm <sup>2</sup> ) 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  $\neq$  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 NID  $\neq$  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. ILE.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 (NGEOM = 1).* The parameters XI, RI (radius of the duct) and XE must be specified.

*b. Circular-Arc, Conical Wall (NGEOM = 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 (NGEOM = 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 (NGEOM = 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.
YCB1	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 YCB1. 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 YCB1, respectively. The tabular data need not be equally spaced. From the specified values of NCBPTS, XCBI, YCB1, 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 ≠ 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 in 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 ≠ 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 Rρi 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.
Rρi(M)	The same as UI, except denotes density in lbm/ft <sup>3</sup> or kg/m <sup>3</sup> .
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 ≠ 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.
RρiL	The same as UIL, except denotes density in lbm/ft <sup>3</sup> or kg/m <sup>3</sup> .
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 R $\rho$ I 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 R $\rho$ I 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 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 p 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.
IWALL $\emptyset$		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.
N $\emptyset$ SLIP		An integer that, when equal to zero, specifies free-slip walls whereas N $\emptyset$ SLIP = 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 $C_a$ 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 $\theta$ boundary condition or supersonic inflow has no effect. The default value is 0.0.
ALE	The coefficient $C_a$ 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 $C_a$ 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 $C_{a1}$ in Eq. (24) in the artificial viscosity model. A nondimensional value is used. The default value is 0.4.
XLA	Denotes the coefficient $C_A$ in Eq. (23) in the artificial 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.
XR0	Denotes the coefficient $C_p$ 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  $\sim 0.5$ , 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.

**NS7** 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:  

$$u_{L,M}^{N+1} = SMPT * u_{L,M}^{N+1} + (1.0 - SMPT) * u_{L,M}^N.$$
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  $US$ ,  $VS$ ,  $PS$ ,  $R\theta S$ ,  $QS$ , and  $ES$  must be dimensioned for  $LMAX$  and  $MMAX$ , while arrays  $ULS$ ,  $VLS$ ,  $PLS$ ,  $R\theta LS$ ,  $QLS$ , 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  $\mu$  by the following equation:

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

where  $T$  is the temperature in  $^{\circ}R$  or  $K$ . The units of  $\mu$  are  $lbf\cdot s/ft^2$  or  $Pa\cdot s$ . The default values are 0.0.

CLA,  
ELA

These parameters specify the second coefficient of viscosity  $\lambda$  by the following equations:

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

where  $T$  is the temperature in  $^{\circ}R$  or  $K$ . The units of  $\lambda$  are  $lbf\cdot s/ft^2$  or  $Pa\cdot s$ . 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  $^{\circ}R$  or  $K$ . The units of  $k$  are  $lbf/s\cdot^{\circ}R$  or  $W/m\cdot K$ . 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  $\bar{a}$  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  $C_1, C_2, \sigma_q,$  and  $\sigma_e,$  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:

$$q_{L,M} \geq \text{BFST} * \text{FSQ}(M)$$

$$e_{L,M} \geq \text{BFST} * \text{FSE}(M),$$

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  $L < \text{LDFSS}$  or  $L > \text{LDFSF}$  (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  $\text{ft}^2/\text{s}^2$  or  $\text{m}^2/\text{s}^2$ . 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  $\text{ft}^2/\text{s}^3$  or  $\text{m}^2/\text{s}^3$ . The default value is 0.1.
- FSQL Denotes the inlet or free-stream turbulence energy level (ITM = 2 or 3) in  $\text{ft}^2/\text{s}^2$  or  $\text{m}^2/\text{s}^2$  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  $\text{ft}^2/\text{s}^3$  or  $\text{m}^2/\text{s}^3$ . 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 $\mu_T$ 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  $\neq$  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  $\neq$  LMAX). If the dual-flow-space walls begin and end in the interior, then 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(LDFSF) 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(LDFSF) 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 ≠ 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 ≠ 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 ≠ 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 ≠ 2, MVCT ≠ MMAX - 1, and MVCT > MVCB + 1. Where dual-flow-space walls are present, MVCB ≠ MDFS + 1 and MVCT ≠ 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 <sub>s</sub> , 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 lbf/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 NVCMI, 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 lbf/ft<sup>3</sup> or kg/m<sup>3</sup>. 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<sup>2</sup>/s<sup>2</sup> 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  $TST\cancel{O}P$ , 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 preceded by the card  $*C\cancel{O}MDECK,MCC$ , 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  $*C\cancel{O}MDECK,MCC$  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,  $LI \geq LMAX$ ,  $MI \geq MMAX$ ,  $LII = LI + 1$ , and  $MII = MI + 1$ .  $MQS \geq MVCT$  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  $LI$ ,  $MI$ ,  $LII$ ,  $MII$ ,  $MQS$ ,  $LTS$ , 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  $PL\cancel{O}T$  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  $PL\cancel{O}T$  will have to be replaced by comparable routines. On the other hand, if velocity vector and contour plots are not needed, then subroutine  $PL\cancel{O}T$  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 number = 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,  $\lambda_A$  and  $\mu_A$  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  $NPLBT = 500$ . 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  $NMAX$ ,  $TSTPT$  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,  $NID = 0$  and values for the arrays  $U, V, P, R\theta, UL, VL, PL,$  and  $R\theta L$  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  $NGE\theta M = 1$ . 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  $R\theta I$ , as well as the variables  $UIL, VIL, PIL,$  and  $R\theta IL$ , 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 TURRL.* 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  $10^\circ$  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  $NPL\theta T$  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  $NMAX$ ,  $TSTOP$  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  $NID = -2$ . 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  $NGEOM = 1$ . 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  $\lambda$ -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).

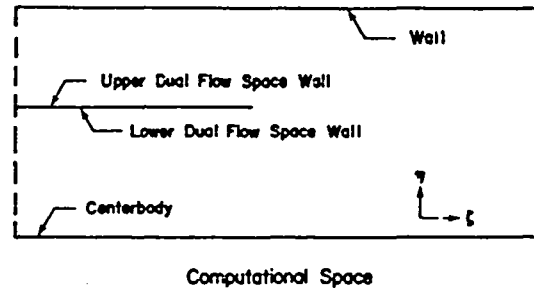
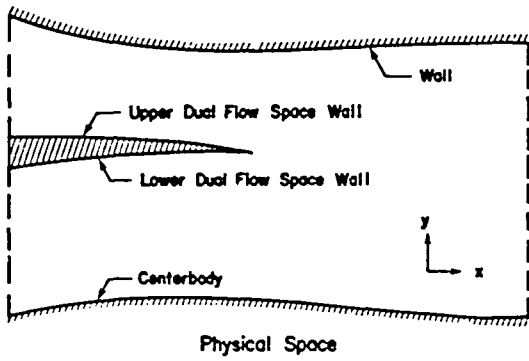


Fig. 1. Physical and computational spaces.

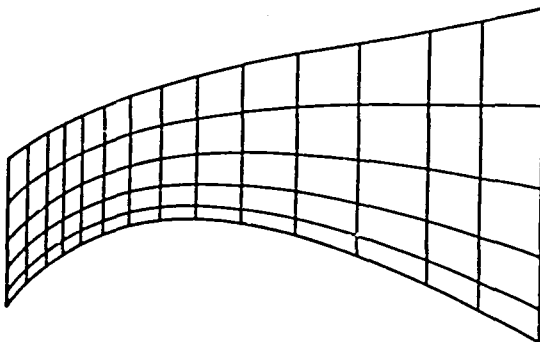


Fig. 2. Physical space grid.

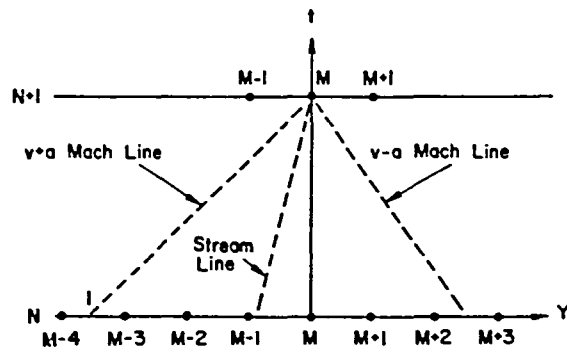


Fig. 3. Quick Solver characteristic grid.

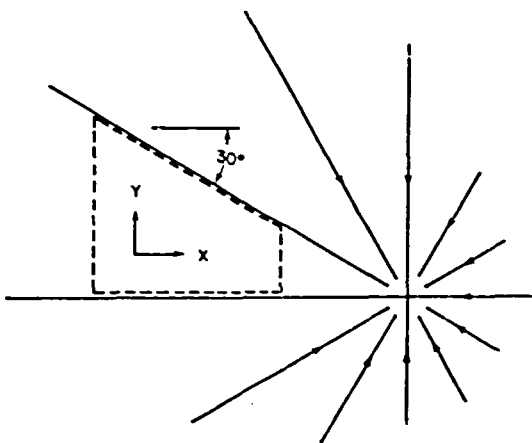


Fig. 4. Planar subsonic sink flow.

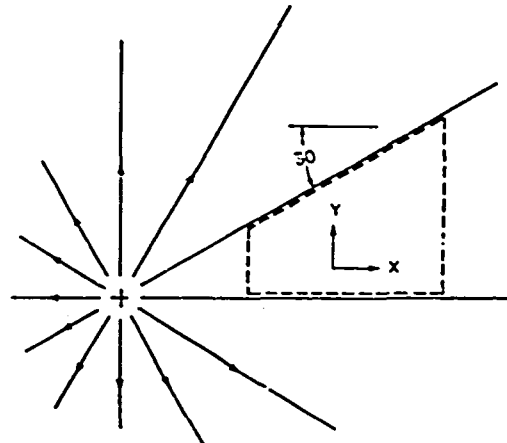


Fig. 5. Planar supersonic source flow.

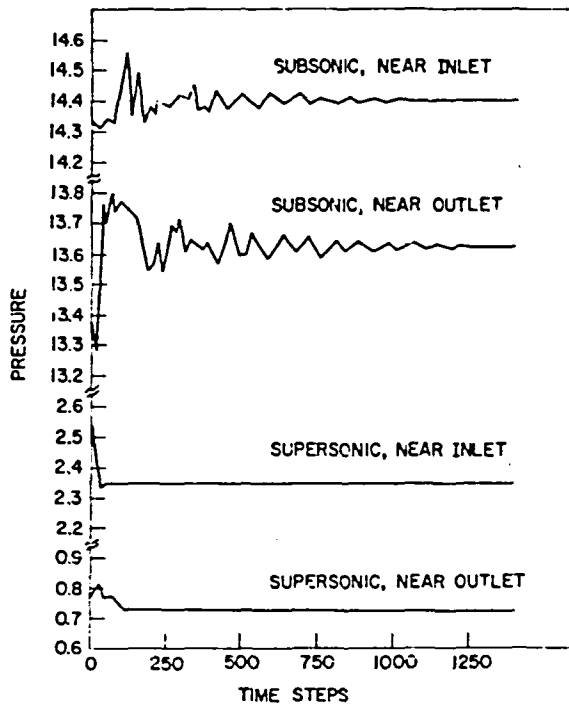


Fig. 6.  
Subsonic sink and supersonic source flow solutions.

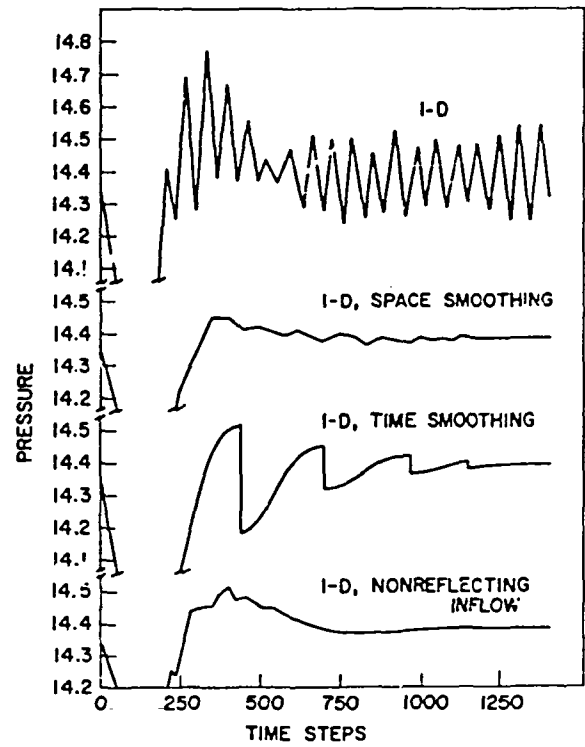


Fig. 8.  
Subsonic sink flow with the  $u$ ,  $v$ , and  $p$  inflow boundary condition.

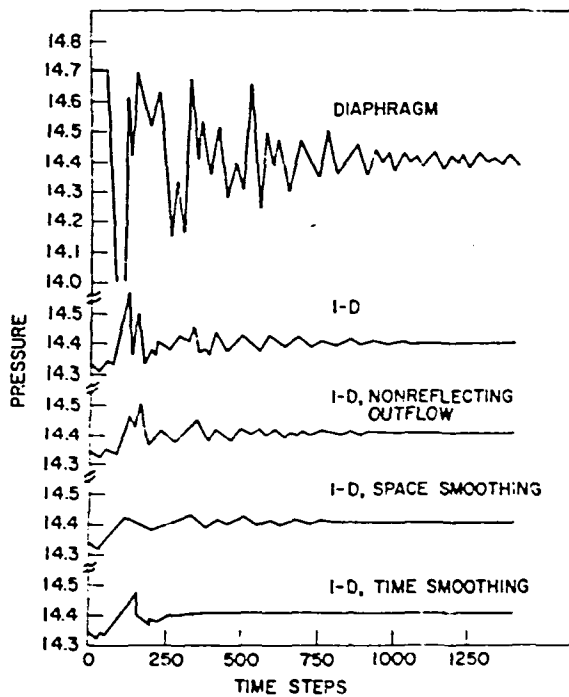


Fig. 7.  
Subsonic sink flow with the  $p_T$ ,  $T_T$ , and  $\theta$  inflow boundary condition.

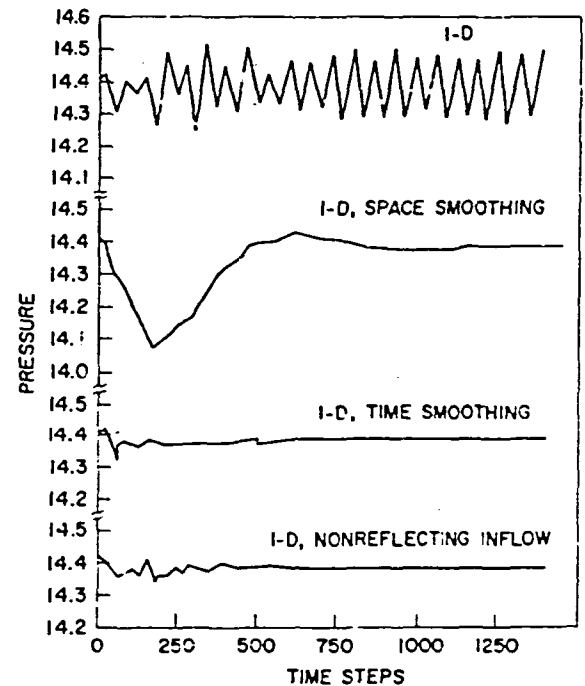


Fig. 9.  
Subsonic sink flow with the  $u$ ,  $v$ , and  $p$  inflow boundary condition and matched mass flow, 1D, initial-data surface.

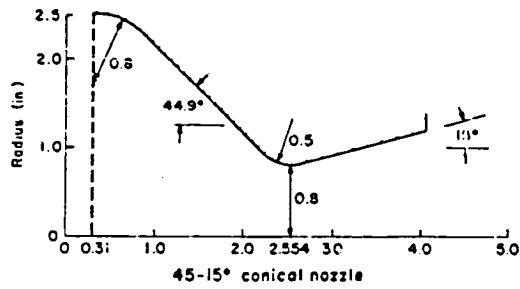


Fig. 10.  
Nozzle geometry for Case No. 1.

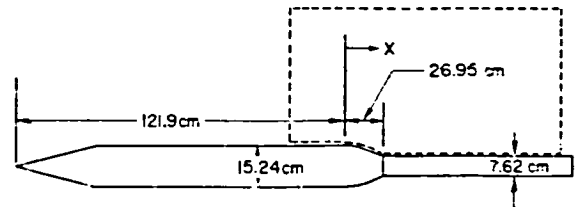


Fig. 12.  
Boattail afterbody geometry for Case No. 2.

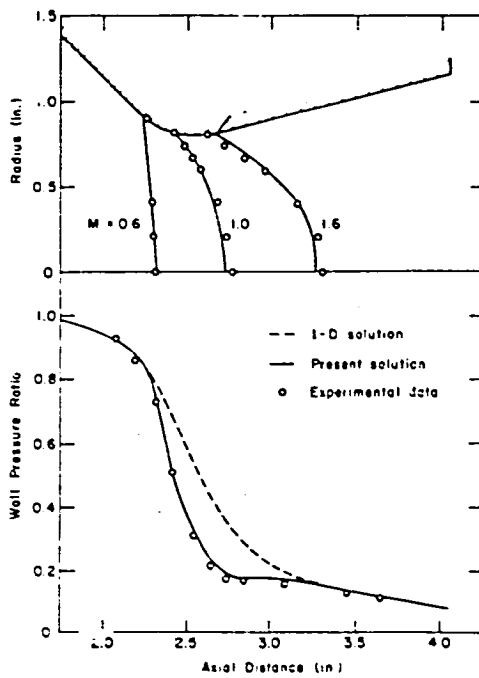


Fig. 11.  
Mach number contours (top) and wall pressure ratio for Case No. 1.

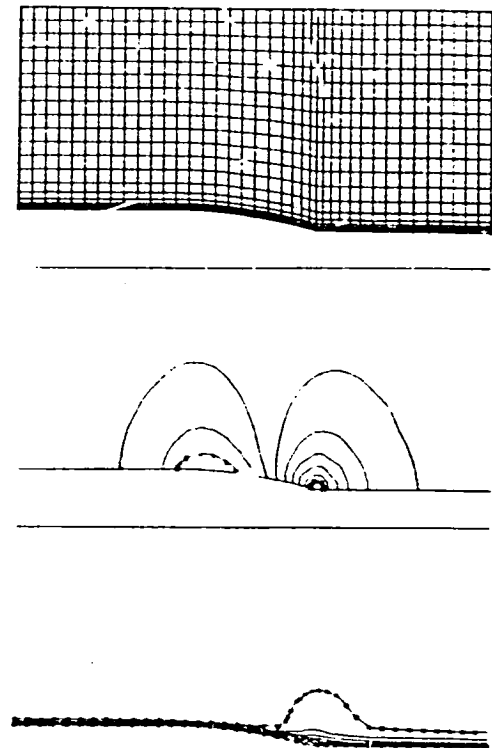


Fig. 13.  
Physical space grid (top), pressure (middle), and Mach number contours for Case No. 2.

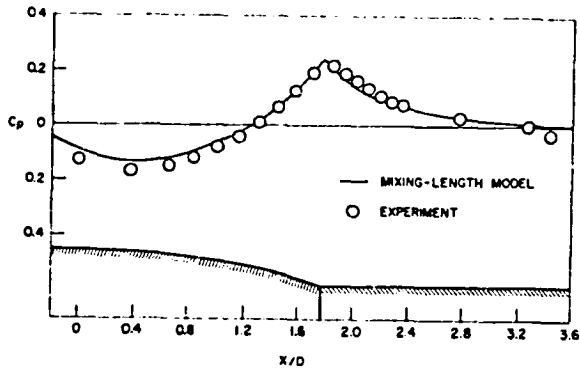


Fig. 14.  
Surface pressure coefficient for Case No. 2.

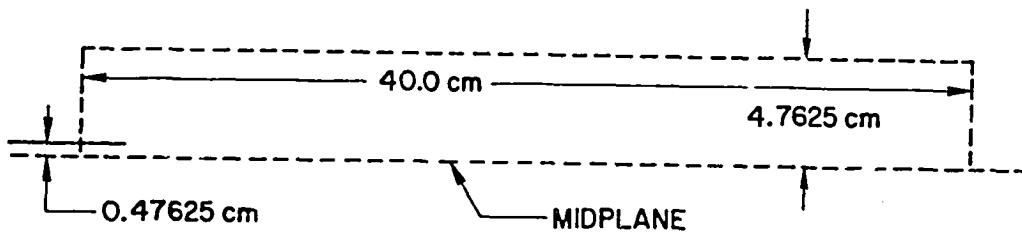


Fig. 15.  
Plane jet geometry for Case No. 3.

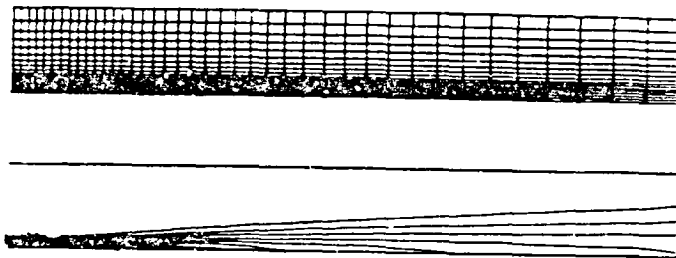


Fig. 16.  
Physical space grid (top) and Mach number contours for Case No. 3.



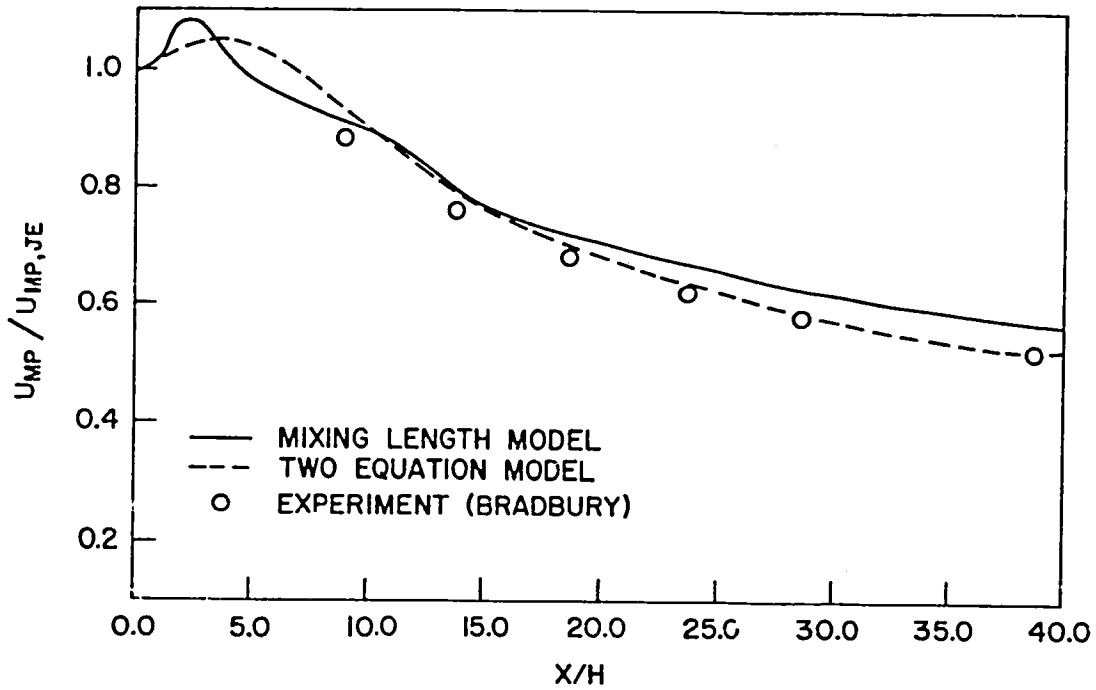


Fig. 17.  
Midplane velocity decay for Case No. 3.

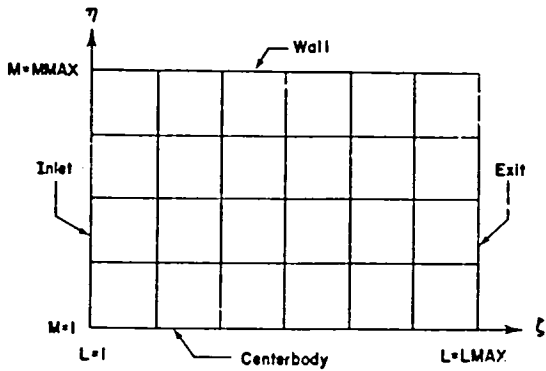


Fig. 18.  
Single-flow-space computational grid.

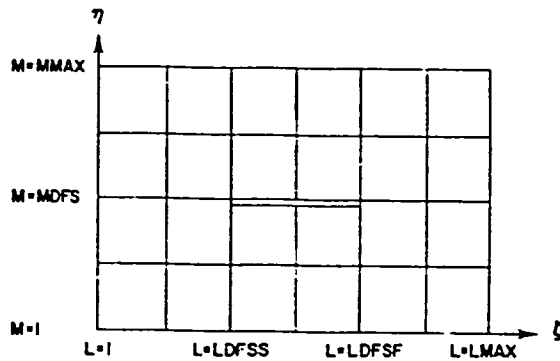


Fig. 19.  
Dual-flow-space computational grid.

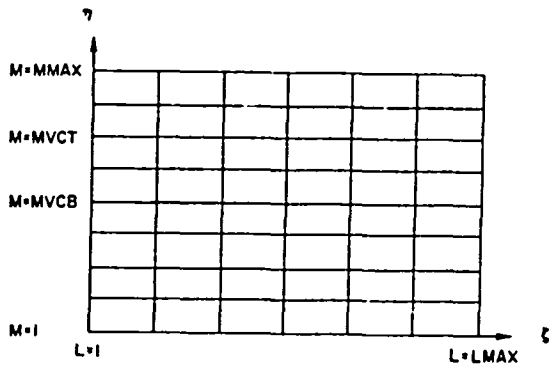


Fig. 20.  
Subcycled computational grid.

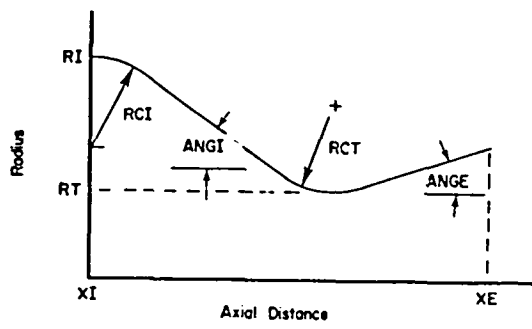


Fig. 21.  
Circular-arc, conical wall geometry.

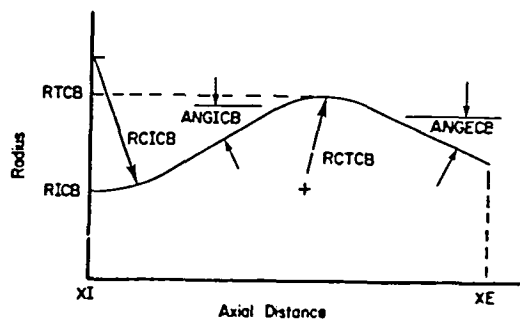


Fig. 22.  
Circular-arc, conical centerbody geometry.

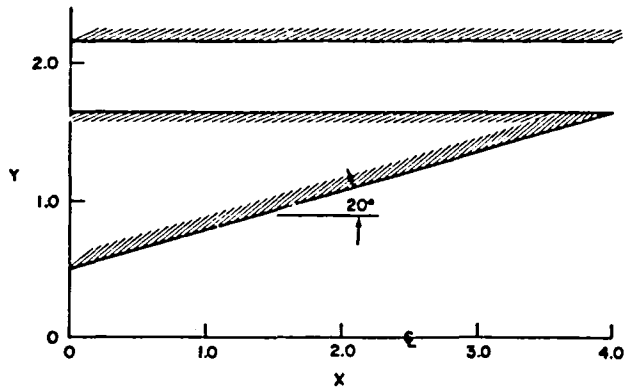


Fig. 23.  
Case No. 1 geometry.

```

VMAP2 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 NID=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 $
$GENTRY NDIM=0,NGEOM=1,XI=0.0,XE=4.0,RI=2.1547 $
$GEBL $
$SBC 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,LDFS=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,
1.0774,1.1351,1.1928,1.2506,1.3083,1.3660,1.4238,1.4815,1.5392,1.5970,
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 NPLT= 500 FDT= .90 FDT1=1.00 FDTI= .90 IPUNCH=0  
 IUI=1 IUD=1 IVPTS=1 NCONVI= 1 TSTOP= .10E+03 NID= 0 TCONV=0.000 NASM=1 IUNIT=1  
 RSTAR= 0.000000 RSTARS= 0.000000 PLOW= .0100 RLOW= .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

## ARTIFICIAL 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 SMPF=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 IOS=0 NIOSS=2 NIOSF=0 NVCM= 0 ILLOS=30 SOS= .50 COS= .001

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

N=	10.	T=	.67677118 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=	.05587706 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=	8.28066455	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. (con.)

## 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 (°SIA)	RHO (LRM/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	69.51261	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	0.0000	179.92401	152.127155	.6441	.5006	118.2721
2	8	.2000	1.8547	.6441	0.0000	179.92401	152.127155	.6441	.5006	118.2721
2	9	.2000	1.9547	.6441	0.0000	179.92401	152.127155	.6441	.5006	118.2721
2	10	.2000	2.0547	.6441	0.0000	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	1.7291	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	0.0000	179.92725	152.129108	.6441	.5006	118.2727
3	8	.4000	1.8547	.6441	0.0000	179.92725	152.129108	.6441	.5006	118.2727
3	9	.4000	1.9547	.6441	0.0000	179.92725	152.129108	.6441	.5006	118.2727
3	10	.4000	2.0547	.6441	0.0000	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	0.0000	179.93296	152.132558	.6441	.5005	118.2738
4	8	.6000	1.8547	.6441	0.0000	179.93296	152.132558	.6441	.5005	118.2738
4	9	.6000	1.9547	.6441	0.0000	179.93296	152.132558	.6441	.5005	118.2738
4	10	.6000	2.0547	.6441	0.0000	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 = 28.99644355 SECONDS (DELTA T = .05754550, NVCM = 1, CNUMS = 1.00, (20, 2), (0, 0))

62

I.	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	.3330	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, CNUMS = 1.00, (20, 2), (0, 0))

49

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	8	2.4000	1.9547	.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	13.01958	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	2.2434	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	.6266	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	.9921	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)

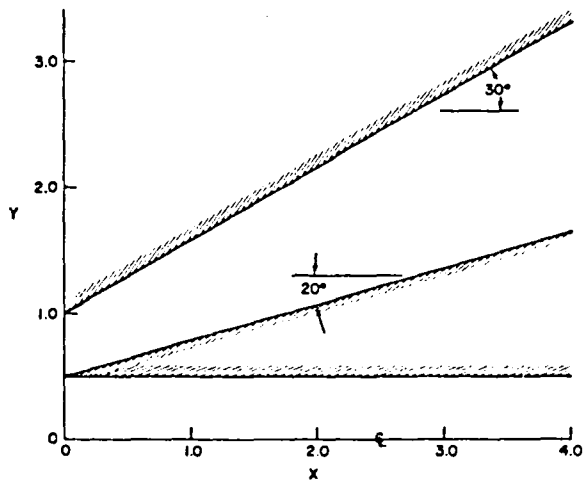


Fig. 26.  
Case No. 2 geometry.

```

VNAP2 CASE 2 - SUPERSONIC SC 'RGE-SUBSONIC CONSTANT AREA FLGW
$CNTRL LMAX=21,MMAX=11,NMAX=500,NPLOT=500,IUNIT=1,RGAS=0.01,
TSTOP=100.0 $
$IVS NID=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,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,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,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,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=1.0,1.1155,1.2709,1.3464,1.4619,1.5773,1.6928,1.8083,1.9238,2.0392,
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 $
JBC 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 NID= 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

IINLFT=0 IEXITT=0 IEX=1 ISUPER=3 DYW=.0010 IVRC=0 INBC=0 IWALL=0 IWALLO=0 ALI=0.00 ALE=0.00  
 ALW=0.00 NSIAG=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

ARTIFICIAL 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 SMPF=1.00 SMPF=1.00 SMPT=1.00 SMPF=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 MVCR= 0 MVCT= 0 IOS=0 NIOSS=2 NIOSF=0 NVCM= 0 ILLQS=30 SOS= .50 COS= .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= 2.60736589 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#	9.75976439	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#	350.	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#	26.50900533	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/MF1	T(LBF)	T/T1
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. 5G0 - TIME = 30.09322540 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	94.5979
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	0.0000	179.94450	152.139524	.6440	.5005	118.2760
2	3	.2000	.2000	.6440	0.0000	179.94450	152.139524	.6440	.5005	118.2760
2	4	.2000	.3000	.6440	0.0000	179.94450	152.139524	.6440	.5005	118.2760
2	5	.2000	.4000	.6440	0.0000	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	60.50799	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	0.0000	179.95138	152.143681	.6440	.5005	118.2773
3	3	.4000	.2000	.6440	0.0000	179.95138	152.143681	.6440	.5005	118.2773
3	4	.4000	.3000	.6440	0.0000	179.95138	152.143681	.6440	.5005	118.2773
3	5	.4000	.4000	.6440	0.0000	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	46.56352	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	.8427	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	0.0000	179.95091	152.143398	.6440	.5005	118.2772
4	3	.6000	.2000	.6440	0.0000	179.95091	152.143398	.6440	.5005	118.2772
4	4	.6000	.3000	.6440	0.0000	179.95091	152.143398	.6440	.5005	118.2772
4	5	.6000	.4000	.6440	0.0000	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	34.03735	46.115423	1.8908	1.8601	73.8091
4	11	.6000	1.3464	1.6503	.9528	32.58263	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. 000 - 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	7	.8000	.7309	1.8217	.5259	33.98803	45.937772	1.8961	1.8631	73.9872
5	8	.8000	.8771	1.8016	.6274	32.81371	44.833210	1.9077	1.8846	73.1906
5	9	.8000	1.0233	1.7793	.7228	31.65328	43.724326	1.9205	1.9077	72.3928
5	10	.8000	1.1695	1.7546	.8143	30.44179	42.543676	1.9344	1.9327	71.5542
5	11	.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	7	1.0000	.7887	1.8752	.5413	29.37709	41.375081	1.9518	1.9577	71.0019
6	8	1.0000	.9464	1.8529	.6446	28.40590	40.425452	1.9619	1.9780	70.2574
6	9	1.0000	1.1041	1.8283	.7414	27.46269	39.489769	1.9729	1.9995	69.5438
6	10	1.0000	1.2619	1.8012	.8343	26.46292	38.477828	1.9851	2.0230	68.7745
6	11	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	7	1.2000	.8464	1.9201	.5543	25.77387	37.670133	1.9986	2.0420	69.4199
7	8	1.2000	1.0157	1.8962	.6590	24.95046	36.836126	2.0074	2.0615	67.7337
7	9	1.2000	1.1850	1.8698	.7569	24.15841	36.023058	2.0172	2.0818	67.0637
7	10	1.2000	1.3542	1.8409	.8514	23.30152	35.124801	2.0283	2.1047	66.3392
7	11	1.2000	1.5235	1.8109	.9413	22.32400	34.144972	2.0409	2.1304	65.5558
7	11	1.2000	1.6928	1.7774	1.0262	21.45017	33.134878	2.0533	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	7	1.4000	.9041	1.9586	.5654	22.88582	34.595375	2.0386	2.1183	66.1528
8	8	1.4000	1.0849	1.9333	.6713	22.17336	33.850018	2.0466	2.1371	65.5047
8	9	1.4000	1.2658	1.9056	.7704	21.48936	33.125244	2.0555	2.1568	64.8731
8	10	1.4000	1.4466	1.8753	.8664	20.73441	32.307548	2.0658	2.1794	64.1782
8	11	1.4000	1.6275	1.8437	.9579	19.92505	31.414173	2.0777	2.2048	63.4264
8	11	1.4000	1.8083	1.8086	1.0442	19.09689	30.488485	2.0893	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))

76

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	2.1821	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	.5036	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	1.0888	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	15.00820	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	1.0065	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	113.2813
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	11.94300	21.770772	2.2197	2.5328	54.8601
14	10	2.6000	2.2510	1.9776	1.0275	11.49447	21.191775	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	1.3080	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	53.1030
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	1.5192	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	2.0020	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	.6426	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	7.32421	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)

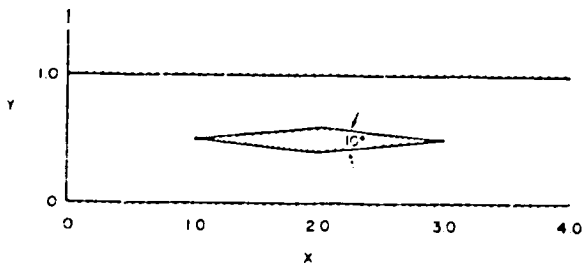


Fig. 29.  
Case No. 3 geometry.

```

V*MAP2 CASE 3 - SUBSONIC AIRFOIL
$CNTRL LMAX=21,MMAX=11,NMAX=500,NPLOT=500,IUNIT=1,RGAS=0.01,
TSTOP=100 0 $
$IVS NID=-2,RSTAR=0.7464 $
$GEMTRY NU:M=0,NGEOM=1,XI=0.0,XE=4.0,RI=1.0 $
$GCBL NGCB=1,RICB=0.0 $
$BC PT=213.514,TI=124.2,PE=180.0 $
$AVL $
$RVL $
$TURBL $
$DFSL NDFS=2,LDFS=6,LDFS=16,MDFS=6,
YL(6)=0.5,0.4825,0.4650,0.4475,0.4300,0.4125,0.4300,0.4475,
O 4650,0.4825,0.5,
N*XYL(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,
O 5350,0.5175,.5,
N*XYU(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, 1-3 - 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 FDT1= .90 IPUNCH=0  
JUI=1 IUD=1 IVPTS=1 NCONVI= 1 ISTOP= .10E+03 NID=-2 ICONV=0.000 NASM=1 IUNIT=1  
RSTAR= .746400 RSTARS= 0.000000 PLOW= .0100 RLOW= .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	.4000	.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)

BOUNDARY CONDITIONS -

M	PT(P5IA)	Y(T(R))	THETA(DEG)	PE(P5IA)	F5Q(FT2/S2)	F5E(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

ARTIFICIAL 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 NST=1 LAV=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 IOS=0 NIOSS=2 NIOSF=0 NVC41=0 ILLOS=30 SOS=.50 COS=.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=	2.33866242	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=	.05366945	SECONDS.	NVCM =	1.	CNUMS =	1.00.	(11.10).	(0.0)
N=	120.	T=	7.03092347	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=	.05867589	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=	24.63373059	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=	.05367600	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/MF1	T(LBF)	T/T1
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))

98

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	.6439	-.0000	179.96177	152.150775	.6439	.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	180.18227	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	1	.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)

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

L	N	X (IN)	Y (IN)	U (F/S)	V (F/S)	P (PSIA)	RHO (LBM/FT <sup>3</sup> )	VMAG (F/S)	WACH NO	T (R)
5	6	.9000	.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	118.1403
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	153.029413	.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	.5255	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	147.712329	.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	145.052813	.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))

88

I	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	116.0476
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	.0060	.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	1.9000	.5700	.8242	.0721	161.05797	140.605579	.8273	.6537	114.5459
10	7	1.8000	.6560	.8158	.0521	162.54924	141.513242	.8174	.6446	114.8650
10	8	1.8000	.7420	.8075	.0358	163.45810	142.057836	.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	.0558	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	.8155	.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	116.1745
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	.5375	.7444	-.0651	169.18976	145.621219	.7473	.5859	116.1848
13	7	2.4000	.6270	.7483	-.0502	169.11983	145.574651	.7500	.5881	116.1740

Fig. 31. (cont)

SOLUTION SURFACE NO. 500 - TIME = 29.22792830 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/FT <sup>3</sup> )	VMAG (F/S)	MACH NO	T (R)
12	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	.5135	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	118.4742
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	152.371759	.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, NVGM = 1, CNUMS = 100, (11,10), (0,0))

8

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	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	.6455	.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	179.59576	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	.4988	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	152.086371	.6422	.4992	118.2171
20	7	3.8000	.6000	.6440	-.0002	179.79889	152.079376	.6440	.5006	118.2204
20	8	3.8000	.7000	.6445	-.0002	179.78175	152.067717	.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	179.77673	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

```

1 *COMDECK,MCC
2   PARAMETER (LI=41, MI=25, LI1=42, MI1=26, MQS=9, LTS=41, MTS=25)
3   COMMON /ONESID/ UD(4), VD(4), PD(4), ROD(4)
4   COMMON /SOLUTN/ U(LI,MI,2), V(LI,MI,2), P(LI,MI,2), RO(LI,MI,2),
5   1 UL(LI,2), VL(LI,2), PL(LI,2), ROI(LI,2)
6   COMMON /CNTRLC/ LMAX, MMAX, NMAX, NPRINT, TCONV, FDT, GAMMA, RGAS,
7   1 GAM1, GAM2, L1, L2, L3, M1, M2, DX, DY, DT, N, N1, N3, NASM,
8   2 ICHAR, NID, LJET, JFLAG, IERR, IUI, IUO, DXR, DYS, IB, RS,AR,
9   3 RSTARS, NPLOT, G, PC, TC, LC, PLOW, ROLOW, CO(LI,MI1), NSTART,
10  4 GAM3, RG, NC, ISTOP
11  COMMON /GEMTRYC/ NGEOM, XI, RI, XT, RT, XE, RE, RCI, RCT, ANGI,
12  1 ANGE, YW(LI), XWI(LI), YWI(LI), NXNY(LI), NWPIS, IINT, IDIF, LT,
13  2 NDIM
14  COMMON /GCB/ NGCB, XICB, RICB, XTCB, RTCB, XECB, RECB, RCICB,
15  1 RCTCB, ANGICB, ANGECB, YCB(LI), XCB(LI), YCB(LI), NXNYCB(LI),
16  2 NCBPTS, IINTCB, IDIFCB
17  COMMON /BCC/ PT(MI), TT(MI), THETA(MI), MASSE, MASSI, MASST,
18  1 THRUST, NSTAG, NOSLIP, IEXITT, TW(LI), TCB(LI), ISUPER, DYW, IVBC
19  2 , IEX, IAS, PTL, TLT, THETA, UIL, VIL, PIL, ROIL, TL(LI), TU(LI)
20  3 , IWALL, UI(MI), VI(MI), PI(MI), ROI(MI), PE(MI), PEL, PEI, NPE,
21  4 INBC, IINLET, IWALLO, ALI, ALE, ALW
22  COMMON /AV/ IAV, CAV, NST, SMP, LSS, XMU, XLA, PRA, XRO, OUT(LI,MI
23  1 ), OVT(LI,MI), CPT(LI,MI), OROT(LI,MI), SMACH, OUTL(LI), OVTL(LI)
24  2 , OPTL(LI), OROTL(LI), SMPT, US(LTS,MTS), VS(LTS,MTS), PS(LTS,MTS)
25  3 ), ROS(LTS,MTS), OS(LTS,MTS), ES(LTS,MTS), ULS(LTS), VLS(LTS),
26  4 PLS(LTS), ROLS(LTS), OLS(LTS), ELS(LTS), NTST, NTC, LSF, IDIVC,
27  5 ISS, MSS, MSF
28  COMMON /RV/ CMU, CLA, CK, EMU, ELA, EK, CHECK, TMUX, TMUY, TMUIX,
29  1 TMUIY
30  COMMON /TURB/ ITM, TML, Q(LI,MI,2), E(LI,MI,2), QL(LI,2), EL(LI,2)
31  1 , CAL, COMU, C1, C2, SIGO, SIGE, QO(LI,MI), QEF(LI,MI), QOTL(LI)
32  2 , OETL(LI), FSO(MI), FSE(MI), FSOL, FSEL, COL, LPRINT, MPRINT,
33  3 QLOW, ELOW, IMLM, DEL, DELS, UBLE, YSL1, YSL2, YMIN, MMIN, IMP,
34  4 BFST, CML1, CML2, PRT, STBO, STRE
35  COMMON /DFS/ YU(LI), YL(LI), NXNYU(LI), NXNYL(LI), MDFS1, MDFS1,
36  1 MMAXD, LDFSS, LDFSF, MDFS, NDFS, IINTDFS, IDIFDFS, NLPTS, NUPTS,
37  2 XLI(LI), YLI(LI), XUI(LI), YUI(LI), MDFSC
38  COMMON /VC/ IST, MVCB, MVCT, XP(LI), YI(MI), IVC, VN(MI), RIND,
39  1 RIND1, MVCB1, MVCT1, MVC, NN1, NN3, UU1(LI), UU2(LI), VV1(LI),
40  2 VV2(LI), PP1(LI), PP2(LI), ROR01(LI), POR02(LI), QQ1(LI), QQ2(LI)
41  3 , EE1(LI), EE2(LI), DZDX(LI), X(LI), DYDVW(MI), Y(MI), IQSD,
42  4 ILLOS, DUDYQS(LI,MQS,2), DVDYQS(LI,MQS,2), DFDYQS(LI,MQS,2), SOS,
43  5 IOS, COS, NVCM
44  COMMON /MAPC/ IP, LMAP, MMAP, AL3, AL4, BE3, BE4, DE3, DE4, OM1,
45  1 OM2, VP
46  REAL M3J, NXNY, NXNYCB, MASSI, MASST, MASSE, LC, LC2, NXNYL, NXNYU
47 *DECK,VNAP2
48 PROGRAM VNAP2 (ITAPE,OTAPE1,PUN1,ITY,TAPE5=ITAPE,TAPE6=OTAPE1
49  1 ,TAPE8=PUN1,IA=559=ITY)
50 C
51 C
52 C
53 C VNAP2, A COMPUTER PROGRAM FOR THE COMPUTATION OF TWO DIMENSIONAL,
54 C TIME-DEPENDENT, COMPRESSIBLE, TURBULENT FLOW
55 C
56 C BY MICHAEL C. CLINE, I-3
57 C LOS ALAMOS NATIONAL LABORATORY
58 C
59 C
60 C
61 C
62 C

```

PROGRAM ABSTRACT

```

63 C      THE NAVIER-STOKES EQUATIONS FOR TWO-DIMENSIONAL, TIME-
64 C      DEPENDENT FLOW ARE SOLVED USING THE SECOND-ORDER, MACCORMACK
65 C      FINITE DIFFERENCE SCHEME. ALL BOUNDARY CONDITIONS ARE COMPUTED
66 C      USING A SECOND-ORDER, REFERENCE PLANE CHARACTERISTIC SCHEME
67 C      WITH THE VISCOUS TERMS TREATED AS SOURCE FUNCTIONS. THE FLUID
68 C      IS ASSUMED TO BE A PERFECT GAS. THE STEADY-STATE SOLUTION IS
69 C      OBTAINED AS THE ASYMPTOTIC SOLUTION FOR LARGE TIME. THE FLOW
70 C      BOUNDARIES MAY BE ARBITRARY CURVED SOLID WALLS AS WELL AS FREE
71 C      JET ENVELOPES. THE GEOMETRY MAY CONSIST OF SINGLE AND DUAL
72 C      FLOWING STREAMS. TURBULENCE EFFECTS ARE MODELED WITH EITHER
73 C      A MIXING-LENGTH, A TURBULENCE ENERGY EQUATION, OR A TURBULENCE
74 C      ENERGY-DISSIPATION RATE EQUATIONS MODEL. THIS PROGRAM ALLOWS
75 C      VARIABLE GRID SPACING AND INCLUDES OPTIONS TO SPEED UP THE
76 C      CALCULATION FOR HIGH REYNOLDS NUMBER FLOWS.
77 C
78       DIMENSION TITLE(10)
79 *CALL ,MCC
80       NAMelist /CNTRL/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,FDTI,FDTI,VDT
81       1 ,VDT1,GAMMA,RGAS,TSTOP,IUI,IUO,IPUNCH,NPLOT,LPP1,MPP1,LPP2,MPP2
82       2 ,LFP3,MPP3,NASM,NAME,NCONVI,IUNIT,PLOW,ROLOW,IVPTS
83       NAMelist /IVS/ NID,U,V,P,RO,Q,E,UL,VL,PL,ROL,QL,EI,RSTAR,RSTARS
84       1 ,NSTART,ISTART
85       NAMelist /GEMTRY/ NDIM,NGEOM,XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XWI,YWI
86       1 ,NWPTS,IINT,IDIF,YW,NXNY,JFLAG,LJET
87       NAMelist /GCB/ NGCB,RCB,RCB,RCIB,RCTCB,ANGICB,ANGECB,XCBI,YCBI
88       1 ,NGBPTS,IINTCB,IDIFCB,YCB,NXNYCB
89       NAMelist /BC/ NSTAG,PI,TT,THETA,PTL,TTL,THETA,PE,PEL,PEI,NPE,UI
90       1 ,VI,PI,ROI,UIL,VIL,PIL,ROIL,TW,TCB,TL,TU,ISUPER,INBC,IWALL,IWALLO
91       2 ,IINLET,IEXIT,IEX,IVBC,NOSLIP,DYW,IAS,ALI,ALE,ALW
92       NAMelist /AVL/ CAV,XMU,XLA,PRA,XRO,LSS,LSF,MSS,MSF,IDIVC,ISS,SMACH
93       1 ,NST,SMP,SMPF,SMPT,SMPTF,NTST,IAV
94       NAMelist /RVL/ CMU,EMU,CLA,ELA,CK,EK
95       NAMelist /TURBL/ ITA,IMLM,CML1,CML2,CAL,COL,COMU,C1,C2,SIGD,SIGE
96       1 ,BFST,FSQ,FSE,FSQL,FSEL,QLOW,ELOW,LPRINT,MPRINT,PRF,SIBQ,STBE
97       NAMelist /DFSL/ MDFS,LDFSS,LDFSF,NDFS,YU,YL,NXNYU,NXNYL,XUI,XII
98       1 ,YUI,YLI,NUPTS,NLPTS,IINTDFS,IDIDFS
99       NAMelist /VCL/ ISXP,VI,MVCB,MVCT,NVCM1,IOS,NIQSS,NIQSF,COS,ILLUS
100      1 ,SQS
101 C
102 C      SET THE ARRAY SIZES FOR SPECIFYING THE INPUT DEFAULT VALUES
103 C
104       LD=LI
105       MD=MI
106 C
107 C      SET DEFAULT VALUES
108 C
109       IO TCONV=0.0
110       TSTART=0.0
111       THETA(1)=0.0
112       CAV=0.0
113       XMU=0.4
114       XLA=1.0
115       PRA=0.7
116       XRO=0.6
117       LSS=1
118       LSF=999
119       MSS=1
120       MSF=999
121       SMACH=0.0
122       IDIVC=0
123       ISS=0
124       IC=0.0
125       CMU=0.0
126       CLA=0.0
127       CK=0.0
128       EMU=0.0
129       ELA=0.0
130       EK=0.0
131       RSTAR=0.0
132       RICB=0.0
133       RTCB=0.0
134       RSTARS=0.0

```



135	PT(1)=0.0
136	TT(1)=0.0
137	DEL SMP=0.0
138	DOSMP=0.0
139	CM1=0.0
140	CM2=0.0
141	PTL=0.0
142	TTL=0.0
143	THETA=0.0
144	UIU=0.0
145	VIL=0.0
146	PIL=0.0
147	ROIL=0.0
148	PE(1)=14.7
149	PE(2)=-1.0
150	PEL=0.0
151	PEI=0.0
152	NPE=0
153	TMUX=0.0
154	TMU=0.0
155	EA=ST=0.0
156	CQL=0.0
157	TMUX=0.0
158	TMU=0.0
159	ALI=0.0
160	ALE=0.0
161	ALW=0.0
162	FDTI=0.0
163	ISTOP=1.0
164	CRIMS=1.0
165	SMP=1.0
166	SMPE=1.0
167	SMPT=1.0
168	SMPE=1.0
169	FDTI=1.0
170	FDT=0.9
171	VDT=0.25
172	VDTI=0.25
173	NADM=1
174	NID=1
175	NDIM=1
176	IFX=1
177	NCONVI=1
178	IUI=1
179	IUD=1
180	IAPTS=1
181	NLCM=1
182	NLCM=1
183	IMM=1
184	NIST=1
185	NSTAC=0
186	NAME=0
187	IFUNCH=0
188	NPCR=0
189	NMA=0
190	NRSIN=0
191	NTR=0
192	NTR=0
193	IFRR=0
194	IFLAG=0
195	ISUPER=0
196	NVCM=0
197	IUNIT=0
198	NOSLIP=0
199	IINCE=0
200	IEUNIT=0
201	NSTART=0
202	ITM=0
203	IAS=0
204	IVC=0
205	MVTE=0
206	MVCT=0

```

207      IWALL=0
208      IWALLO=0
209      IB=0
210      LDFSS=0
211      LDFSF=0
212      MDFS=0
213      LPRINT=0
214      MPRINT=0
215      IVBC=0
216      INBC=0
217      LPP1=0
218      IQS=0
219      NIOSF=0
220      IAV=0
221      IST=0
222      LDUF=0
223      MDUF=0
224      NTC=0
225      IINT=2
226      IDIF=2
227      IINTCB=2
228      IDIFCB=2
229      IINTDFS=2
230      IDIFDFS=2
231      ILLOS=30
232      GAMMA=1.4
233      RGAS=53.35
234      NPILOT=-1
235      G=32.174
236      PC=144.0
237      LC=12.0
238      PLOW=0.01
239      ROLOW=0.0001
240      DYW=0.001
241      CAL=1.0
242      COMU=0.09
243      C1=1.44
244      C2=1.8
245      SIGO=1.0
246      SIGE=1.3
247      SOS=C.5
248      COS=0.001
249      NIOSS=2
250      FSUL=0.0001
251      OLOW=0.0001
252      FSEL=0.1
253      ELOW=0.1
254      PRT=0.9
255      STBO=0.0
256      STBE=0.0
257      ISTOP=0
258      DO 20 M=1,MD
259      UI(M)=0.0
260      VI(M)=0.0
261      PI(M)=0.0
262      ROI(M)=0.0
263      FSO(M)=0.0001
264      FSE(M)=0.1
265      20 CONTINUE
266      DO 30 L=1,LD
267      YCB(L)=0.0
268      YL(L)=0.0
269      YU(L)=0.0
270      NXNYCB(L)=0.0
271      NXNYL(L)=0.0
272      NXNYU(L)=0.0
273      OL(L,1)=0.0
274      EL(L,1)=0.0
275      OL(L,2)=0.0
276      EL(L,2)=0.0
277      UL(L,1)=0.0
278      VL(L,1)=0.0

```

```

279     PL(L,1)=0.0
280     ROL(L,1)=0.0
281     PL(L,2)=0.0
282     ROL(L,2)=0.0
283     TW(L)=-1.0
284     TCB(L)=-1.0
285     TL(L)=-1.0
286     TU(L)=-1.0
287     DO 30 M=1,MD
288     O(L,M,1)=0.0
289     E(L,M,1)=0.0
290     O(L,M,2)=0.0
291     E(L,M,2)=0.0
292 30 CONTINUE
293 C
294 C     READ IN INPUT DATA
295 C
296     READ (5,1370) TITLE
297     IF (EOF(5)) 40,50
298 40 CALL EXIT
299 50 READ (5,C:TRI)
300     READ (5,IVS)
301     READ (5,GFMTRY)
302     READ (5,GCBL)
303     READ (5,BC)
304     READ (5,AVL)
305     READ (5,RVL)
306     READ (5,TURBL)
307     READ (5,DFSL)
308     READ (5,VCL)
309     IF (NAME.EQ.O) GO TO 60
310     WRITE (6,CNTRL)
311     WRITE (6,IVS)
312     WRITE (6,GEMTRY)
313     WRITE (6,GCBI)
314     WRITE (6,BC)
315     WRITE (6,AVL)
316     WRITE (6,RVL)
317     WRITE (6,TURBL)
318     WRITE (6,DFSL)
319     WRITE (6,VCL)
320 C
321 C     PRINT INPUT DATA
322 C
323 60 WRITE (6,1380)
324     WRITE (6,1410)
325     WRITE (6,1400)
326     WRITE (6,1420)
327     WRITE (6,1430)
328     WRITE (6,1390)
329     WRITE (6,1440) TITLE
330     WRITE (6,1390)
331     WRITE (6,1450)
332     NPRIND=ABS(FLOAT(NPRINT))
333     IF (FDTI.EQ.O.O) FDTI=FDT
334     WRITE (6,1460) LMAX,MMAX,NMAX,NPRIND,NPLOT,FDT,FDT1,FDTI,IPUNCH
335 1 .IUI,IUD,IVPTS,NCONVI,ISTOP,N1D,TCONV,NASH,IUNIT,RSTAR,RSTARS
336 2 .PLOW,ROLOW,VDT,VDT1
337     WRITE (6,1390)
338     IF (IUI.EQ.1) WRITE (6,1470) GAMMA,RGAS
339     IF (IUI.EQ.2) WRITE (6,1480) GAMMA,RGAS
340     WRITE (6,1390)
341     WRITE (6,1490)
342     IF (NDIM.EQ.O) WRITE (6,1500)
343     IF (NDIM.EQ.1) WRITE (6,1510)
344 C
345 C     CALCULATE THE GEOMETRY RADIUS AND SLOPE
346 C
347     L1=LMAX-1
348     L2=LMAX-2
349     L3=LMAX-3
350     M1=MMAX-1

```

```

351      M2=MMA*2
352      MDFSM1=MDFS-1
353      MDFSPI=MDFS+1
354      MMA*2=MMA*2-MDFS
355      CHECK=ABS(CMU)+ABS(CLA)+ABS(CK)
356      IF (NGEOM.NE.3) GO TO 70
357      XI=XWI(1)
358      XE=XWI(NWPTS)
359      70 DX=(XE-XI)/FLOAT(L1)
360      DY=1.0/FLOAT(M1)
361      IF (ISF.NE.0) GO TO 90
362      DO 80 L=1,LMAX
363      XP(L)=XI+FLOAT(L-1)*DX
364      80 CONTINUE
365      90 XP(1)=XI
366      XP(LMAX)=XE
367      WRITE (6,1390)
368      CALL GEOM
369      IF (IERR.NE.0) GO TO 10
370      XICB=XI
371      XECB=XE
372      IF (NGCB.EQ.0.AND.MDFS.EQ.0) GO TO 140
373      IF (NGCB.NE.0) CALL GEOMCB
374      IF (IERR.NE.0) GO TO 10
375      IF (MDFS.NE.0) CALL GECMLU
376      IF (IAS.EQ.0) GO TO 110
377      IF (LDFSF.EQ.LMAX) GO TO 100
378      NXNYL(LDFSF)=0.5*(NXNYL(LDFSF)+NXNYU(LDFSF))
379      NXNYU(LDFSF)=NXNYL(LDFSF)
380      100 IF (LDFSS.EQ.1) GO TO 110
381      NXNYL(LDFSS)=0.5*(NXNYL(LDFSS)+NXNYU(LDFSS))
382      NXNYU(LDFSS)=NXNYL(LDFSS)
383      110 LT=1
384      Y0=YW(1)-YU(1)+YL(1) YCB(1)
385      DO 130 L=1,LMAX
386      IF (NDIM.EQ.0) YY=YW(L)-YU(L)+YL(L)-YCB(L)
387      IF (NDIM.EQ.1) YY=YW(L)+2*YU(L)+2*YL(L)+2*YCB(L)+2
388      IF (YY.GT.0.0) GO TO 120
389      WRITE (6,1610)
390      GO TO 10
391      120 IF (YY.LT.Y0) LT=L
392      IF (LT.EQ.L) Y0=YY
393      130 CONTINUE
394 C
395 C      CONTINUE SET UP AND PRINTING OF INPUT DATA
396 C
397      140 GAM1=GAMMA/(GAMMA-1.0)
398      GAM2=(GAMMA-1.0)/2.0
399      GAM3=(GAMMA+1.0)/(GAMMA-1.0)
400      IF (PE(2).NE.-1.0) GO TO 160
401      DO 150 M=2,MMA*2
402      PE(M)=PE(1)
403      150 CONTINUE
404      PEL=PE(1)
405      160 IF (MDFS.NE.0.AND.LDFSF.NE.LMAX) PEL=PE(MDFS)
406      IF (NSTAG.NE.0) GO TO 180
407      DO 170 M=2,MMA*2
408      PT(M)=PT(1)
409      TT(M)=TT(1)
410      THETA(M)=THETA(1)
411      170 CONTINUE
412      PTL=PT(1)
413      TTL=TT(1)
414      THETA1=THETA(1)
415      180 IF (ISUPER.NE.3) GO TO 190
416      PI(MDFS)=PTL
417      TI(MDFS)=TTL
418      THETA(MDFS)=THETA1
419      190 IF (ISUPER.NE.2) GO TO 200
420      PI(MDFS)=PIL
421      200 WRITE (6,1380)
422      IF (IUI.EQ.1) WRITE (6,1580)

```

```

423     IF (IUI.EQ.2) WRITE (6,1590)
424     DO 210 M=1,MMAX
425     IF (M.EQ.MDFS.AND.LDSS.EQ.1) WRITE (6,1600) M,PTL,TTL,THETA,PEL
426     1 ,FSQL,FSEL
427     WRITE (6,1600) M,PT(M),TT(M),THE(A(M),PE(M),FSQ(M),FSF(M)
428 210 CONTINUE
429     WRITE (6,2020) IINFT,IEYIT,IFX,ISUPER,DYW,IVBC,INBC,IWALL,IWALLO
430     1 ,ALT,AIE,ALW,NSTAG,NPF,PEI
431     IF (NOSLIP.EQ.0) WRITE (6,1840)
432     IF (NOSLIP.NE.0) WRITE (6,1850)
433     WRITE (6,1390)
434     IF (TW(1).LT.0.0.AND.IWALL.EQ.0) WRITE (6,1890)
435     IF (TW(1).GE.0.0) WRITE (6,1900)
436     WRITE (6,1390)
437     IF (TCR(1).LT.0.0.AND.NGCB.NE.0) WRITE (6,1910)
438     IF (TCR(1).GE.0.0) WRITE (6,1940)
439     IF (MDFS.EQ.0) GO TO 220
440     IF (TCR(1).GE.0.0.OR.NGCB.NE.0) WRITE (6,1390)
441     IF (TII(1).LT.0.0) WRITE (6,1920)
442     IF (TII(1).GE.0.0) WRITE (6,1950)
443     WRITE (6,1390)
444     IF (TI(1).LT.0.0) WRITE (6,1930)
445     IF (TUI(1).GE.0.0) WRITE (6,1960)
446 220 WRITE (6,1390)
447     IF (SMP.LT.0.0) SMP=0.0
448     IF (SMPF.LT.0.0) SMPF=0.0
449     IF (SMP.GT.1.0) SMP=1.0
450     IF (SMPF.GT.1.0) SMPF=1.0
451     WRITE (6,1830) CAV,XMU,XLA,PRA,XRD,LSS,LSF,IDIVC,ISS,SMACH,NST,SMP
452     1 ,SMPF,SMPT,SMPTF,NTST,IAV,MSS,MSF
453     WRITE (6,1390)
454     IF (CML1.NE.0.0.OR.CML2.NE.0.0) GO TO 230
455     IF (NDIM.EQ.0) CML1=0.125
456     IF (NDIM.EQ.0) CML2=0.125
457     IF (NDIM.NE.0) CML1=0.11
458     IF (NDIM.NE.0) CML2=0.11
459 230 IF (CQL.NE.0.0) GO TO 240
460     CQL=17.2
461     IF (NDIM.NE.0) CQL=CQL*0.625/0.875
462 240 IF (IUI.EQ.1) WRITE (6,1860) CMU,CLA,CK,EMU,ELA,EK
463     IF (IUI.EQ.2) WRITE (6,1870) CMU,CLA,CK,EMU,ELA,EK
464     WRITE (6,1390)
465     IF (ITM.EQ.0) WRITE (6,1970)
466     IF (ITM.EQ.1) WRITE (6,1980) CAL,IMLM,CML1,CML2,PR1
467     IF (ITM.EQ.2) WRITE (6,1990)
468     IF (ITM.EQ.2) WRITE (6,2000) CAL,COL,COMU,IMLM,CML1,CML2,PR1
469     IF (ITM.EQ.3) WRITE (6,2010)
470     IF (ITM.EQ.3) WRITE (6,2030) CAL,COMU,C1,C2,SIGQ,SIGE,BFST,PR1
471     1 ,STBO,STBE
472     WRITE (6,1390)
473     WRITE (6,2040) IST,MVCB,MVCT,IOS,NIOS,NIOSF,NVCM1,ILLOS,SOS,COS
474 C
475 C     CHECK THE WALL OPTIONS
476 C
477     IVCE=0
478     IF (JFLAG.EQ.0) GO TO 250
479     IF (LJET.LE.2.OR.LJET.GE.L1) IVCE=1
480     IF (NOSLIP.NE.0) IVCE=1
481     IF (IWALL.NE.0) IVCE=1
482     IF (IVCE.EQ.0) GO TO 250
483     WRITE (6,2150)
484     GO TO 10
485 250 IF (ISUPER.EQ.0) GO TO 260
486     WRITE (6,2140)
487     GO TO 10
488 C
489 C     CHECK MIXING-LENGTH TURBULENCE MODEL
490 C
491 260 IF (ITM.NE.1) GO TO 300
492     IF (MDFS.NE.0) GO TO 280
493     IF (IMLM.EQ.1) GO TO 270
494     IF (NOSLIP.EQ.0) IVCE=1
495     IF (NGCB.NE.0.AND.IWALL.EQ.0) IVCE=1
496     IF (NGCB.EQ.0.AND.IWALL.NE.0) IVCE=1

```

```

497      GO TO 290
498 270 IF (NOSLIP.NE.O) IVCE=1
499      IF (NGCB.EQ.O.AND.IWALL.NE.O) IVCE=O
500      GO TO 290
501 280 IF (NGCB.NE.O.OR.IWALL.EQ.O) IVCE=1
502 290 IF (IVCE.EQ.O) GO TO 300
503      WRITE (6,2050)
504      GO TO 10
505 C
506 C      CHECK THE DUAL FLOW SPACE AND VARIABLE GRID PARAMETERS
507 C
508 300 IF (MVCB.NE.O.AND.MVCT.NE.O) IVC=1
509      IP=-1
510      CALL MAP
511      IF (IERR.NE.O) GO TO 10
512      MDFSC=O
513      IF (MDFS.GE.MVCB.AND.MDFS.LE.MVCT) MDFSC=1
514      IF (MDFS.EQ.O) LDFSS=O
515      IF (MDFS.EQ.O) LDFSF=O
516      IF (MDFS.EQ.O) GO TO 320
517      IF (LDFSS.EQ.1) GO TO 310
518      IF (TL(LDFSS).GT.O.O) TL(1)=TL(LDFSS)
519      IF (TU(LDFSS).GT.O.O) TU(1)=TU(LDFSS)
520 310 IF (MDFS.EQ.2.OR.MDFS.EQ.MMAX-1) IVCE=1
521      IF (LDFSS.EQ.2.OR.LDFSF.EQ.LMAX-1) IVCE=1
522      IF (ISUPER.GE.2.AND.LDFSS.NE.1) IVCE=1
523      CLDFSS=ABS(YU(LDFSS)-YL(LDFSS))/YL(LDFSS)
524      CLDFSF=ABS(YU(LDFSF)-YL(LDFSF))/YL(LDFSF)
525      IF (LDFSS.NE.1.AND.CLDFFS.GT.O.OO1) IVCE=1
526      IF (LDFSF.NE.LMAX.AND.CLDFFS.GT.O.OO1) IVCE=1
527      IF (JFLAG.EQ.1.AND.LJET.LE.LDFSF) IVCE=1
528      IF (IVCE.EQ.O) GO TO 320
529      WRITE (6,2060)
530      GO TO 10
531 C
532 C      CHECK THE SUBCYCLED GRID PARAMETERS
533 C
534 320 IF (IVC.EQ.O) GO TO 350
535      MVCB1=MVCB+1
536      MVCT1=MVCT-1
537      IF (NVCMI.EQ.O) GO TO 330
538      I11=NVCMI/2
539      I12=(NVCMI+1)/2
540      IF (I11.EQ.I12) IVCE=1
541      IF (IVCE.EQ.O) GO TO 330
542      WRITE (6,2070)
543      GO TO 10
544 330 IF (MVCB.EQ.1.AND.MVCT.EQ.MMAX) IVCE=1
545      IF (MDFS.EQ.O) GO TO 340
546      IF (MVCT.LT.MDFS-1.OR.MVCB.GT.MDFS+1) GO TO 340
547      IF (MVCB.EQ.MDFS+1.OR.MVCT.EQ.MDFS-1) IVCE=1
548      IF (MVCB.GT.MDFS-2) IVCE=1
549      IF (MVCT.LT.MDFS+2) IVCE=1
550      IF (IVCE.EQ.O) GO TO 350
551      WRITE (6,2080)
552      GO TO 10
553 340 IF (MVCB.EQ.2.OR.MVCT.EQ.MMAX-1) IVCE=1
554      IF (MVCT-MVCB.LT.2) IVCE=1
555      IF (IVCE.EQ.O) GO TO 350
556      WRITE (6,2090)
557      GO TO 10
558 C
559 C      CHECK THE QUICK SOLVER PARAMETERS
560 C
561 350 IF (IVC.EQ.O) IQS=O
562      IF (IQS.EQ.O) GO TO 370
563      IF (NIQSF.EQ.O) NIOSF=NMAX
564      IF (NOSLIP.EQ.O) IVCE=1
565      IF (MVCT.EQ.MMAX.AND.IWALL.NE.O) IVCE=1
566      IF (MVCB.EQ.1.AND.NGCB.EQ.O) IVCE=1
567      IF (MDFS.EQ.O) GO TO 360
568      IF (MVCB.GT.MDFS.OR.MVCT.LT.MDFS) IVCE=1

```

```

569 360 IF (IVCE EQ.0) GO TO 370
570 WRITE (6.2130)
571 GO TO 10
572 C
573 C CHECK FOR ZERO VALUES OF Q AND E - SET THE DEFAULT VALUES
574 C
575 370 IF (ITM.LE.1) GO TO 390
576 IF (NSTART.NE.0) GO TO 390
577 DO 380 L=1,LMAX
578 IF (OL(L,1).LE.0.0) OL(L,1)=FSQL
579 IF (EL(L,1).LE.0.0) EL(L,1)=FSFL
580 DO 380 M=1,MMAX
581 IF (O(L,M,1).LE.0.0) O(L,M,1)=FSO(M)
582 IF (E(L,M,1).LE.0.0) E(L,M,1)=FSE(M)
583 380 CONTINUE
584 C
585 C CONVERT METRIC UNITS TO ENGLISH UNITS
586 C
587 390 IF (IUI.EQ.1) GO TO 540
588 RSTAR=RSTAR/2.54
589 RSTARS=RSTARS/6.4516
590 PLOW=PLow/6.8948
591 ROLOW=ROLOW/16.02
592 CMU=CMU/47.88/1.8**EMU
593 CLA=CLA/47.88/1.8**ELA
594 CK=CK*0.125/1.8**EK
595 RGAS=RGAS/5.38032
596 XT=XT/2.54
597 RT=RT/2.54
598 XI=XI/2.54
599 XE=XE/2.54
600 XICB=XICB/2.54
601 XECB=XECB/2.54
602 DX=DX/2.54
603 DO 400 L=1,LMAX
604 XP(L)=XP(L)/2.54
605 YW(L)=YW(L)/2.54
606 YCB(L)=YCB(L)/2.54
607 YL(L)=YL(L)/2.54
608 YUL(L)=YUL(L)/2.54
609 400 CONTINUE
610 DO 410 M=1,MMAX
611 PT(M)=PT(M)/6.8948
612 PE(M)=PE(M)/6.8948
613 TT(M)=TT(M)*1.8
614 410 CONTINUE
615 PTL=PTL/6.8948
616 PEL=PEL/6.8948
617 PEI=PEI/6.8948
618 TTL=TTL*1.8
619 IF (TCB(1).LT.0.0) GO TO 430
620 DO 420 L=1,LMAX
621 TCB(L)=TCB(L)*1.8
622 420 CONTINUE
623 430 IF (TW(1).LT.0.0) GO TO 450
624 DO 440 L=1,LMAX
625 TW(L)=TW(L)*1.8
626 440 CONTINUE
627 450 IF (TL(1).LT.0.0) GO TO 470
628 DO 460 L=1,LMAX
629 TL(L)=TL(L)*1.8
630 460 CONTINUE
631 470 IF (TU(1).LT.0.0) GO TO 490
632 DO 480 L=1,LMAX
633 TU(L)=TU(L)*1.8
634 480 CONTINUE
635 490 IF (ISUPER.EQ.0) GO TO 520
636 DO 500 M=1,MMAX
637 UI(M)=UI(M)/0.3048
638 VI(M)=VI(M)/0.3048
639 PI(M)=PI(M)/6.8948
640 ROI(M)=ROI(M)/16.02

```

```

641 500 CONTINUE
642 U1L=U1L/O.3048
643 V1L=V1L/O.3048
644 P1L=P1L/6.8948
645 R01L=R01L/16.02
646 QLOW=QLOW/O.0929
647 ELOW=ELOW/O.0929
648 FSQ=FSQ/O.0929
649 FSEL=FSEL/O.0929
650 DO 510 M=1,MMAX
651 FSQ(M)=FSQ(M)/O.0929
652 FSE(M)=FSE(M)/O.0929
653 510 CONTINUE
654 520 IF (N1D.NE.O) GO TO 540
655 IF (NSTART.NE.O) GO TO 540
656 DO 530 L=1,LMAX
657 UL(L,1)=UL(L,1)/O.3048
658 VL(L,1)=VL(L,1)/O.3048
659 PL(L,1)=PL(L,1)/6.8948
660 ROL(L,1)=ROL(L,1)/16.02
661 QL(L,1)=QL(L,1)/O.0929
662 EL(L,1)=EL(L,1)/O.0929
663 DO 530 M=1,MMAX
664 U(L,M,1)=U(L,M,1)/O.3048
665 V(L,M,1)=V(L,M,1)/O.3048
666 P(L,M,1)=P(L,M,1)/6.8948
667 RO(L,M,1)=RO(L,M,1)/16.02
668 Q(L,M,1)=Q(L,M,1)/O.0929
669 E(L,M,1)=E(L,M,1)/O.0929
670 530 CONTINUE
671 C
672 C CONVERT INPUT DATA UNITS TO INTERNAL UNITS - THE INTERNAL UNITS
673 C ARE P=LBF/FT2, RO=LBF-S2/FT4, X=YCG-VL=YU-YW-INCHES, Y-
674 C DIMENSIONLESS, DT=IN-S/FT, MU-LA-LRF S IN/FT3, K=LRF-IN/S R FT,
675 C U-FT2/S2, F-FT2/S3, TML=INCHES, U=V-FT/S, AND RGAS=LRF FT/LBM-R.
676 C
677 540 IF (IUNIT.EQ.O) GO TO 550
678 PC=1.0
679 LC=1.0
680 G=1.0
681 550 TCONV=TCONV/100.0
682 T=TSTART+LC
683 TSTOP=TSTOP+LC
684 CMU=CMU+LC
685 CIA=CLA+LC
686 CK=CK+LC
687 DO 560 L=1,LMAX
688 XWI(L)=0.0
689 560 CONTINUE
690 DO 570 M=1,MMAX
691 PT(M)=PT(M)+PC
692 PE(M)=PE(M)+PC
693 THETA(M)=THETA(M)+O.0174533
694 570 CONTINUE
695 PTL=PTL+PC
696 PEL=PEL+PC
697 PEI=PEI+PC
698 THETA=THETA+O.0174533
699 IF (N1D.NE.O) GO TO 590
700 DO 580 L=1,LMAX
701 PL(L,1)=PL(L,1)+PC
702 ROL(L,1)=ROL(L,1)/G
703 DO 580 M=1,MMAX
704 P(L,M,1)=P(L,M,1)+PC
705 RO(L,M,1)=RO(L,M,1)/G
706 580 CONTINUE
707 590 RG=RGAS+G
708 C
709 C FILL THE ARRAYS AT L=1 WITH THE INFLOW BOUNDARY CONDITIONS
710 C
711 IF (ISUPER.EQ.O) GO TO 620
712 DO 600 M=1,MMAX

```



```

713     IF (ISUPER.EQ.2.AND.M.GE.MDFS) GO TO 600
714     IF (ISUPER.EQ.3.AND.M.LT.MDFS) GO TO 600
715     U(1,M,1)=UI(M)
716     V(1,M,1)=VI(M)
717     IF (NSTART.EQ.0) P(1,M,1)=PI(M)*PC
718     RO(1,M,1)=ROI(M)/G
719     U(1,M,2)=U(1,M,1)
720     V(1,M,2)=V(1,M,1)
721     P(1,M,2)=P(1,M,1)
722     RO(1,M,2)=RO(1,M,1)
723 600 CONTINUE
724     IF (ISUPER.EQ.3) GO TO 610
725     UL(1,1)=UIL
726     VL(1,1)=VIL
727     IF (NSTART.EQ.0) PL(1,1)=P.L*PC
728     ROL(1,1)=ROI/L/G
729     UI(1,2)=UL(1,1)
730     VL(1,2)=VL(1,1)
731     PL(1,2)=PL(1,1)
732     ROL(1,2)=ROL(1,1)
733     GO TO 620
734 610 PT(MDFS)=PTL
735     TT(MDFS)=TTL
736     THETA(MDFS)=THETAL
737 C
738 C     ZERO VISCOUS TERM ARRAYS
739 C
740 620 DO 630 L=1,LMAX
741     QUTL(L)=0.0
742     QVTL(L)=0.0
743     QPTL(L)=0.0
744     QROTL(L)=0.0
745     QOTL(L)=0.0
746     QETL(L)=0.0
747     DU 630 M=1,MMAX
748     QUT(L,M)=0.0
749     QVT(L,M)=0.0
750     QPT(L,M)=0.0
751     QROT(L,M)=0.0
752     QOT(L,M)=0.0
753     QET(L,M)=0.0
754 630 CONTINUE
755     IF (NID.EQ.0) GO TO 640
756 C
757 C     COMPUTE THE 1-D INITIAL-DATA SURFACE
758 C
759     CALL ONEDIM
760     IF (IERR.NE.0) GO TO 10
761 C
762 C     COMPUTE THE INITIAL-DATA SURFACE MASS FLOW AND MOMENTUM THRUST
763 C
764 640 IF (NFRINT.GT.0) GO TO 650
765     NPRINT=-NPRINT
766     GO TO 730
767 650 CALL MASFLO
768 C
769 C     PRINT THE INITIAL-DATA SURFACE
770 C
771     IP=0
772     DO 720 IU=1,2
773     IF (IU.EQ.1.AND.IU.EQ.2) GO TO 720
774     IF (IU.EQ.2.AND.IU.EQ.1) GO TO 720
775     NLINE=0
776     WRITE (5,1380)
777     WRITE (6,1520) TSTART,NSTART
778     WRITE (6,1530)
779     IF (IU.EQ.1) WRITE (6,1540)
780     IF (IU.EQ.2) WRITE (6,1550)
781     WRITE (6,1390)
782     DO 710 L=1,LMAX
783     LMAP=L
784     IF (MDFS.NE.0) IB=3

```

```

785     IF (L.NE.1) WRITE (6,1880)
786     IF (L.NE.1) NLINE=NLINE+1
787     LDFS=0
788     IF (L.GE.LDFS.AND.L.LE.LDFS) LDFS=1
789     DO 710 M=1,MMAX
790     MMAP=M
791     CALL MAP
792     IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 670
793     IMDFS=0
794     VELMAG=SQRT(UL(L,1)**2+VL(L,1)**2)
795     XMACH=VELMAG/SQRT(GAMMA*PL(L,1)/ROL(L,1))
796     PRES=PL(L,1)/PC
797     RHO=ROL(L,1)*G
798     TEMP=PL(L,1)/(RHO*RGAS)
799     XPP=XPI(L)
800     UP=UL(L,1)
801     VP=VL(L,1)
802     GO TO 680
803 660  IMDFS=1
804     IB=4
805     CALL MAP
806 670  VELMAG=SQRT(U(L,M,1)**2+V(L,M,1)**2)
807     XMACH=VELMAG/SQRT(GAMMA*P(L,M,1)/RO(L,M,1))
808     PRES=P(L,M,1)/PC
809     RHO=RO(L,M,1)*G
810     TEMP=P(L,M,1)/RHO/RGAS
811     XPP=XPI(L)
812     UP=U(L,M,1)
813     VP=V(L,M,1)
814 680  IF (IU.EQ.1) GO TO 690
815     XPP=XPI(L)*2.54
816     YP=YP*2.54
817     UP=UP*0.3048
818     VP=VP*0.3048
819     PRES=PRES*6.8948
820     RHO=RHO*16.02
821     VELMAG=VELMAG*0.3048
822     TEMP=TEMP*5.0/9.0
823 690  NLINE=NLINE+1
824     IF (NLINE.LT.54) GO TO 700
825     WRITE (6,1380)
826     WRITE (6,1520) TSTART,NSTART
827     WRITE (6,1530)
828     IF (IU.EQ.1) WRITE (6,1540)
829     IF (IU.EQ.2) WRITE (6,1550)
830     WRITE (6,1390)
831     NLINE=1
832 700  WRITE (6,1560) L,M,XPP,YP,UP,VP,PRES,RHO,VELMAG,XMACH,TEMP
833     IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 710
834     IF (IMDFS.EQ.0) GO TO 660
835 710  CONTINUE
836     IF (IU.NE.3) GO TO 730
837 720  CONTINUE
838 730  IF (NPLOT.LE.0) GO TO 740
839     CALL PLOT (TITLE,TSTART,NSTART,IVPTS)
840     WRITE (6,1810) NSTART
841 740  IF (NMAX.EQ.0) GO TO 10
842 C
843 C     SAVE THE SOLUTION FOR THE EXTENDED INTERVAL TIME SMOOTHING
844 C
845     IF (NTST.EQ.1) GO TO 760
846     DO 750 L=1,LMAX
847     ULS(L)=UL(L,1)
848     VLS(L)=VL(L,1)
849     PLS(L)=PL(L,1)
850     ROLS(L)=ROL(L,1)
851     QLS(L)=QL(L,1)
852     ELS(L)=EL(L,1)
853     DO 750 M=1,MMAX
854     US(L,M)=U(L,M,1)
855     VS(L,M)=V(L,M,1)
856     PS(L,M)=P(L,M,1)

```

```

857     RQS(L,M)=RO(L,M,1)
858     QS(L,M)=Q(L,M,1)
859     ES(L,M)=E(L,M,1)
860     750 CONTINUE
861 C
862 C     INITIALIZE THE TIME STEP INTEGRATION LOOP PARAMETERS
863 C
864     760 NMAXD=NMAX
865     N1=1
866     N3=2
867     DQM=0.0
868     MCONV=0
869     NC=0
870     MPC=0
871     MPD=0
872     LDUM=1
873     DXR=1.0/DX
874     DYR=1.0/DY
875     DXPS=DXR*DXR
876     DYRS=DYR*DYR
877     VDT=1.0/VDT
878     VDTI=1.0/VDT
879     FDTD=FDT
880     FDT=FDTI
881     IF (NST.NE.0) DELSMP=(SMPF-SMP)/FLOAT(NST)
882     IF (NST.NE.0) DSMPT=(SMPTF-SMPT)/FLOAT(NST)
883     INTST=0
884     MVCTD=MVCT+1
885     IF (NASM.NE.0.AND.LT.NE.1) LDUM=LT-1
886     WRITE (6,1400)
887     LPP1D=LPP1
888     LPP1=IABS(LPP1)
889     IF (JFLAG.EQ.0) GO TO 770
890     UD(1)=U(LJET-1,MMAX,N1)
891     VD(1)=V(LJET-1,MMAX,N1)
892     PD(1)=P(LJET-1,MMAX,N1)
893     ROD(1)=RO(LJET-1,MMAX,N1)
894     UD(2)=UD(1)
895     VD(2)=VD(1)
896     PD(2)=PD(1)
897     ROD(2)=ROD(1)
898 C
899 C     ENTER THE TIME STEP INTEGRATION LOOP
900 C
901     770 DO 1250 N=1,NMAXD
902 C
903     IF (N.EQ.2) FDT=FDTD
904     SMP=SMP+DELSMP
905     SMPT=SMPT+DSMPT
906     IQSD=0
907     IF (IQS.EQ.0) GO TO 780
908     IF (N.GE.NIQSS.AND.N.LE.NIQSF) IQSD=IQS
909 C
910 C     CALCULATE DELTA T
911 C
912     780 IP=1
913     LPAM=0.0
914     DO 810 L=2,L1
915     LMAP=L
916     IF (MDFS.NE.0) IB=3
917     LDFS=0
918     IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
919     DXP1=XP(L)-XP(L-1)
920     DXP2=XP(L+1)-XP(L)
921     DXP=AMIN1(DXP1,DXP2)
922     DO 810 M=2,M1
923     IF (IVC.EQ.0) GO TO 790
924     IF (M.GE.MVCB.AND.M.LE.MVCT) GO TO 810
925     790 IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 800
926     IB=4
927     GO TO 810
928     800 MWAP=M
929     CALL MAP
930     DXP3=DX/BE3

```

```

931      DYP4=DY/BE4
932      DYP=AMINI(DYP3,DYP4)
933      ABR=AL3/BE3
934      A=SQRT(GAMMA*P(L,M,N1)/RO(L,M,N1))
935      UPA1=(ABS(U(L,M,N1))+A)/DXP+VDT1*TMU1X
936      VTOL=ABS(U(L,M,N1)+ABR+V(L,M,N1))
937      AST=SQRT(AL3*AL3+BE3*BE3)/FE3
938      UPA2=(VTOL+AST*A)/DYP+VDT1*TMU1Y
939      UPA=AMAX1(UPA1,UPA2)
940      IF (UPA.LE.UPAM) GO TO 810
941      UPAM=UPA
942      LDU=L
943      MDU=M
944      810 CONTINUE
945 C
946 C      CALCULATE DELTA T FOR THE SUBCYCLED GRID
947 C
948      IF (IVC.EQ.O) GO TO 860
949      IF (NVCM1.EQ.O) GO TO 820
950      IF (IQS.NE.O.AND.IQSD.EQ.O) GO TO 820
951      NVCM=NVCM1
952      NVCM1=NVCM+1
953      CNUMS=O.O
954      LDUF=O
955      MDUF=O.O
956      GO TO 860
957      820 UPAMF=O.O
958      DO 840 L=2,L1
959      LMAP=L
960      IF (MDFS.NE.O) IB=3
961      LDFS=O
962      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
963      DXP1=XP(L)-XP(L-1)
964      DXP2=XP(L+1)-XP(L)
965      DXP=AMINI(DXP1,DXP2)
966      DO 840 M=MVCB1,MVCT1
967      IF (M.NE.MDFS.OR.LDFS.EQ.O) GO TO 830
968      IB=4
969      GO TO 840
970      830 MMAP=M
971      CALL MAP
972      DYP3=DY/BE3
973      DYP4=DY/BE4
974      DYP=AMINI(DYP3,DYP4)
975      ABR=AL3/BE3
976      A=SQRT(GAMMA*P(L,M,N1)/RO(L,M,N1))
977      UPA1=(ABS(U(L,M,N1))+A)/DXP+VDT1*TMU1X
978      VTOL=ABS(U(L,M,N1)+ABR+V(L,M,N1))
979      AST=SQRT(AL3*AL3+BE3*BE3)/BE3
980      IF (IQSD.NE.O) AST=AST-1.O
981      UPA2=(VTOL+AST*A)/DYP+VDT1*TMU1Y
982      UPA=AMAX1(UPA1,UPA2)
983      IF (UPA.LE.UPAMF) GO TO 840
984      UPAMF=UPA
985      LDU=L
986      MDUF=M
987      840 CONTINUE
988 C
989 C      DETERMINE THE NUMBER OF SUBCYCLES
990 C
991      XNVCM=UPAMF/(UPAM*FDT1)
992      NVCM=O
993      I=-1
994      IF (XNVCM.LE.200.O) GO TO 850
995      IF (N.EQ.1) GO TO 850
996      NP=N*NSTART
997      WRITE (6,2100) NP
998      NMAX=N
999      NVCM=XNVCM
1000     DT=FDT/UPAM
1001     GO TO 1110
1002     850 I=I+2

```

```

1003      IF (XNVCM.LE.FLOAT(I)) NVCM=I
1004      IF (NVCM.EQ.0) GO TO 850
1005      NVCM1=NVCM+1
1006      CNUMS=XNVCM/FLOAT(NVCM)
1007 860 DT=FDI/UPAM
1008      T=T+DT
1009      IF (T.LT.TSTOP) GO TO 870
1010      T=T-DT
1011      DT=TSTOP-T
1012      T=TSTOP
1013      ISTOP=1
1014 C
1015 C      PRINT N,T AND DT
1016 C
1017 870 NPD=NPD+1
1018      NC=NC+1
1019      NPC=NPC+1
1020      TMUX=0.0
1021      TMUY=0.0
1022      TMU1X=0.0
1023      TMU1Y=0.0
1024      IF (NPD.NE.10) GO TO 880
1025      NP=N+NSTART
1026      TIME=T/LC
1027      DTIME=DT/LC
1028      WRITE (6,1820) NP,TIME,DTIME,NVCM,CNUMS,LDU,MDU,LDUF,MDUF
1029      NPD=0
1030 C
1031 C      BEGIN THE SUBCYCLE LOOP
1032 C
1033 880 DO 1010 NVC=1,NVCM1
1034      RIND=FLOAT(NVC-2)/FLOAT(NVCM)
1035      RIND1=FLOAT(NVC-1)/FLOAT(NVCM)
1036      IF (NVC.NE.2) GO TO 890
1037      DT=DT/FLOAT(NVCM)
1038 C
1039 C      CALCULATE THE PREDICTOR SOLUTION
1040 C
1041 890 IB=1
1042      IF ((IUSD.NE.0.AND.NVC.NE.1)) CALL QSOLVE
1043      IF (IERR.NE.0) GO TO 1100
1044      IF (CAV.NE.0.0.OR.CHECK.NE.0.0) CALL VISCOS
1045      IF (IERR.NE.0) GO TO 1100
1046      ICHAR=1
1047      IB=1
1048      CALL INTER
1049      IF (NVC.GT.1.AND.MVCT.NE.MMAX) GO TO 900
1050      IF (NVC.EQ.1.AND.MVCT.EQ.MMAX) GO TO 900
1051      CALL WALL
1052      IF (IERR.NE.0) GO TO 1090
1053 900 IF (NGCB.EQ.0) GO TO 910
1054      IF (NVC.GT.1.AND.MVCB.NE.1) GO TO 910
1055      IF (NVC.EQ.1.AND.MVCB.EQ.1) GO TO 910
1056      IB=2
1057      CALL WALL
1058      IF (IERR.NE.0) GO TO 1090
1059 910 IF (LDFSS.NE.1.OR.(NVC.EQ.1.AND.MDFSC.NE.0)) CALL INLET
1060      IF (LDFSF.NE.LMAX.OR.(NVC.EQ.1.AND.MDFSC.NE.0)) CALL EXITT
1061      IF (IERR.NE.0) GO TO 1090
1062 C
1063 C      CALCULATE THE DUAL FLOW SPACE BOUNDARY PREDICTOR SOLUTION
1064 C
1065      IF (MDFS.EQ.0) GO TO 920
1066      IF (NVC.EQ.1.AND.MDFSC.NE.0) GO TO 920
1067      IF (NVC.GT.1.AND.MVCT.LT.MDFS) GO TO 920
1068      IF (NVC.GT.1.AND.MVCB.GT.MDFS) GO TO 920
1069      IB=4
1070      CALL WALL
1071      IF (IERR.NE.0) GO TO 1090
1072      IF (LDFSS.EQ.1) CALL INLET
1073      IF (LDFSF.EQ.LMAX) CALL EXITT
1074      IF (IERR.NE.0) GO TO 1090

```

```

1075     CALL SWITCH (2)
1076     IB=3
1077     CALL WALL
1078     IF (IERR.NE.O) GO TO 1080
1079     IF (LDFSS.EQ.1) CALL INLET
1080     IF (LDFSF.EQ.LMAX) CALL EXITT
1081     IF (IERR.NE.O) GO TO 1080
1082 C
1083 C     CALCULATE THE CORRECTOR SOLUTION
1084 C
1085     920 IF (ITM.GE.2) CALL TURBC (3)
1086     ICHAR=2
1087     IB 1
1088     CALL INTER
1089     IF (NVC.GT.1.AND.MVCT.NE.MMAX) GO TO 930
1090     IF (NVC.EQ.1.AND.MVCT.EQ.MMAX) GO TO 930
1091     CALL WALL
1092     IF (IERR.NE.O) GO TO 1070
1093     930 IF (NGCB.EQ.O) GO TO 940
1094     IF (NVC.GT.1.AND.MVCB.NE.1) GO TO 940
1095     IF (NVC.EQ.1.AND.MVCB.EQ.1) GO TO 940
1096     IB=2
1097     CALL WALL
1098     IF (IERR.NE.O) GO TO 1070
1099     940 IF (LDFSS.NE.1.OR.(NVC.EQ.1.AND.MDFSC.NE.O)) CALL INLET
1100     IF (LDFSF.NE.LMAX.OR.(NVC.EQ.1.AND.MDFSC.NE.O)) CALL EXITT
1101     IF (IERR.NE.O) GO TO 1070
1102 C
1103 C     CALCULATE THE DUAL FLOW SPACE BOUNDARY CORRECTOR SOLUTION
1104 C
1105     IF (MDFS.EQ.O) GO TO 950
1106     IF (NVC.EQ.1.AND.MDFSC.NE.O) GO TO 950
1107     IF (NVC.GT.1.AND.MVCT.LT.MDFS) GO TO 950
1108     IF (NVC.GT.1.AND.MVCB.GT.MDFS) GO TO 950
1109     IB=3
1110     CALL WALL
1111     IF (IERR.NE.O) GO TO 1080
1112     IF (LDFSS.EQ.1) CALL INLET
1113     IF (LDFSF.EQ.LMAX) CALL EXITT
1114     IF (IERR.NE.O) GO TO 1080
1115     CALL SWITCH (2)
1116     IB=4
1117     CALL WALL
1118     IF (IERR.NE.O) GO TO 1090
1119     IF (LDFSS.EQ.1) CALL INLET
1120     IF (LDFSF.EQ.LMAX) CALL EXITT
1121     IF (IERR.NE.O) GO TO 1090
1122 C
1123 C     SET THE SUBCYCLED GRID END CONDITIONS
1124 C
1125     950 IF (NVCM1.EQ.1) GO TO 1010
1126     IF (NVC.EQ.1) GO TO 990
1127     IF (NVC.EQ.NVCM1) GO TO 970
1128     IF (LPP1D.GE.O) GO TO 960
1129     PCDUM=PC
1130     IF (IUO.EQ.2) PCDUM=PC/6.8948
1131     PPP1=P(LPP1,MPP1,N3)/PCDUM
1132     PPP2=P(LPP2,MPP2,N3)/PCDUM
1133     PPP3=P(LPP3,MPP3,N3)/PCDUM
1134     WRITE (6,211C) NVC,LPP1,MPP1,PPP1,LPP2,MPP2,PPP2,LPP3,MPP3,PPP3
1135     960 IF (ITM.GE.2) CALL TURBC (2)
1136     NNN=N1
1137     N1=N3
1138     N3=NNN
1139     GO TO 1010
1140     970 DT=DT*FLOAT(NVCM)
1141     IF (MVCTD.GE.MMAX) GO TO 1010
1142     DO 980 L=1,LMAX
1143     U(L,MVCTD,N3)=UU2(L)
1144     V(L,MVCTD,N3)=VV2(L)
1145     P(L,MVCTD,N3)=PP2(L)
1146     RO(L,MVCTD,N3)=RORO2(L)

```

```

1147      Q(L,MVCTD,N3)=Q2(L)
1148      E(L,MVCTD,N3)=EE2(L)
1149      980 CONTINUE
1150      GO TO 1010
1151      990 NN1=N1
1152          NN3=N3
1153      IF (MVCTD.GE.MMAX) GO TO 1010
1154      DO 1000 L=1,LMAX
1155          UU1(L)=U(L,MVCTD,N1)
1156          VV1(L)=V(L,MVCTD,N1)
1157          PP1(L)=P(L,MVCTD,N1)
1158          ROR01(L)=RO(L,MVCTD,N1)
1159          QQ1(L)=Q(L,MVCTD,N1)
1160          EE1(L)=E(L,MVCTD,N1)
1161          UU2(L)=U(L,MVCTD,N3)
1162          VV2(L)=V(L,MVCTD,N3)
1163          PP2(L)=P(L,MVCTD,N3)
1164          ROR02(L)=RO(L,MVCTD,N3)
1165          QQ2(L)=Q(L,MVCTD,N3)
1166          EE2(L)=E(L,MVCTD,N3)
1167      1000 CONTINUE
1168      1010 CONTINUE
1169      C
1170      C      PRINT THE PRESSURE AT THE THREE REQUESTED POINTS
1171      C
1172          IF (LPP1.EQ.0) GO TO 1040
1173          NP=N+NSTART
1174          PCDUM=PC
1175          IF (IU0.EQ.2) PCDUM=PC/6.8948
1176          PPP1=P(LPP1,MPP1,N3)/PCDUM
1177          PPP2=P(LPP2,MPP2,N3)/PCDUM
1178          PPP3=P(LPP3,MPP3,N3)/PCDUM
1179          IF (N.GT.NST) GO TO 1030
1180          IF (NTST.GT.0) GO TO 1030
1181          IF (N.GT.2) GO TO 1020
1182          IF (N.EQ.1) PC2=PPP1
1183          IF (N.EQ.2) PC3=PPP1
1184          GO TO 1030
1185      1020 PC1=PC2
1186          PC2=PC3
1187          PC3=PPP1
1188          IF ((PC3-PC2)*(PC2-PC1).LT.0.0) NTST=-1
1189          IF (INTST.EQ.3) INTST=0
1190          IF (INTST.EQ.2) INTST=3
1191          IF (INTST.EQ.1) INTST=2
1192          IF (INTST.EQ.0.AND.NTST.NE.0) INTST=1
1193          IF (INTST.NE.1) NTST=0
1194      1030 WRITE (6,2120) NP,LPP1,MPP1,PPP1,LPP2,MPP2,PPP2,LPP3,MPP3,PPP3
1195          1,NTST
1196      1040 IF (N.LE.NST) CALL SMOOTH
1197          IF (NTST.EQ.-1) NTST=0
1198          IF (ITH.GE.2) CALL TURBC (1)
1199      C
1200      C      DETERMINE THE MAXIMUM (DELTA U)/U
1201      C
1202          IF (TCONV.LE.0.0) GO TO 1060
1203          DQM=0.0
1204          DO 1050 L=LDUM,LMAX
1205          DO 1050 M=1,MMAX
1206          IF (U(L,M,N1).EQ.0.0) GO TO 1050
1207          DO=ABS((U(L,M,N3)-U(L,M,N1))/U(L,M,N1))
1208          IF (DO.GT.DQM) DQM=DO
1209      1050 CONTINUE
1210      C
1211      C      CHECK FOR REQUESTED PRINTING OR PLOTTING
1212      C
1213      1060 IF (DQM.GE.TCONV) GO TO 1110
1214          NCONV=NCONV+1
1215          IF (NCONV.EQ.1) NCHECK=N-1
1216          IF (NCONV.GE.NCONVI) NC=NPRINT
1217          IF (NCONV.GE.NCONVI) NPC=NPLOT
1218          IF (N.GE.NCHECK+NCONVI) NCONV=0

```

```

1219      GO TO 1110
1220 C
1221 1070 IF (MDFS.EQ.O) GO TO 1090
1222 1080 CALL SWITCH (3)
1223 1090 N3=N1
1224 1100 NMAX=N
1225      IF (NVC.GE.2) DT=DT*FLOAT(NVCM)
1226 C
1227 1110 IF (N.EQ.NMAX) NC=NPRINT
1228      IF (N.EQ.NMAX) NPC=NPLOT
1229      IF (ISTOP.NE.O) NC=NPRINT
1230      IF (ISTOP.NE.O) NPC=NPLOT
1231      IF (NC.EQ.NPRINT) GO TO 1120
1232      IF (NPC.EQ.NPLOT) GO TO 1220
1233      GO TO 1240
1234 C
1235 C      COMPUTE THE SOLUTION SURFACE MASS FLOW AND MOMENTUM THRUST
1236 C
1237 1120 ICN=0
1238      IF (JFLAG.EQ.O) GO TO 1130
1239      IF (LT.NE.LJET-1) GO TO 1130
1240      UDUM=U(LT,MMAX,N3)
1241      RODUM=RO(LT,MMAX,N3)
1242      U(LT,MMAX,N3)=UD(3)
1243      RO(LT,MMAX,N3)=ROD(3)
1244      ICN=1
1245 1130 CALL MASFLO
1246      IF (ICN.EQ.O) GO TO 1140
1247      U(LT,MMAX,N3)=UDUM
1248      RO(LT,MMAX,N3)=RODUM
1249 C
1250 C      PRINT THE SOLUTION SURFACE
1251 C
1252 1140 IP=0
1253      DO 1210 IU=1,2
1254      IF (IU.EQ.1.AND.IU.EQ.2) GO TO 1210
1255      IF (IU.EQ.2.AND.IU.EQ.1) GO TO 1210
1256      NLINE=0
1257      WRITE (6,1380)
1258      TIME=T/LC
1259      DTIME=DT/LC
1260      NP=N+NSTART
1261      WRITE (6,1570) NP,TIME,DTIME,NVCM,CNUMS,LDU,MDU,LDUF,MDUF
1262      WRITE (6,1530)
1263      IF (IU.EQ.1) WRITE (6,1540)
1264      IF (IU.EQ.2) WRITE (6,1550)
1265      WRITE (6,1390)
1266      DO 1200 L=1,LMAX
1267      LMAP=L
1268      IF (MDFS.NE.O) IB=3
1269      IF (L.NE.1) WRITE (6,1880)
1270      IF (L.NE.1) NLINE=NLINE+1
1271      LDFS=0
1272      IF (L.GE.LDFS.AND.L.LE.LDFS) LDFS=1
1273      DO 1200 M=1,MMAX
1274      MMAP=M
1275      CALL MAP
1276      IF (M.NE.MDFS.OR.LDFS.EQ.O) GO TO 1160
1277      IMDFS=0
1278      VELMAG=SQRT(U(L,N3)**2+V(L,N3)**2)
1279      XMACH=VELMAG/SQRT(GAMMA*PL(L,N3)/RO(L,N3))
1280      PRES=PL(L,N3)/PC
1281      RHO=RO(L,N3)*G
1282      TEMP=PL(L,N3)/(RHO*RGAS)
1283      XPP=XP(L)
1284      UP=U(L,N3)
1285      VP=V(L,N3)
1286      GO TO 1170
1287 1150 IMDFS=1
1288      IB=4
1289      CALL MAP
1290 1160 VELMAG=SQRT(U(L,M,N3)**2+V(L,M,N3)**2)

```



```

1291      XMACH=VELMAG/SORT(GAMMA*P(L,M,N3)/RO(L,M,N3))
1292      PRES=P(L,M,N3)/PC
1293      RHO=RO(L,M,N3)*G
1294      TEMP=P(L,M,N3)/RHO/RGAS
1295      XPP=XP(L)
1296      UP=U(L,M,N3)
1297      VP=V(L,M,N3)
1298  1170 IF (IU.EQ.1) GO TO 1180
1299      XPP=XP(L)*2.54
1300      YP=YP*2.54
1301      UP=UP*0.3048
1302      VP=VP*0.3048
1303      PRES=PRES*6.8948
1304      RHO=RHO*16.02
1305      VELMAG=VELMAG*0.3048
1306      TEMP=TEMP*5.0/9.0
1307  1180 NLINF=NLIN+1
1308      IF (NLIN.LT.54) GO TO 1190
1309      WRITE (6,1380)
1310      WRITE (6,1570) NP,TIME,DTIME,NVCM,CNUMS,LDU,MDU,LDF,MDUF
1311      WRITE (6,1530)
1312      IF (IU.EQ.1) WRITE (6,1540)
1313      IF (IU.EQ.2) WRITE (6,1550)
1314      WRITE (6,1390)
1315      NLIN=1
1316  1190 WRITE (6,1560) L,M,XPP,YP,UP,VP,PRES,RHO,VELMAG,XMACH,TEMP
1317      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 1200
1318      IF (IMDFS.EQ.0) GO TO 1150
1319  1200 CONTINUE
1320      IF (IUO.NE.3) GO TO 1220
1321  1210 CONTINUE
1322 C
1323 C      GENERATE THE FILM PLOTS
1324 C
1325  1220 IF (NPLT.LT.0) GO TO 1230
1326      IF (NPC.NE.NPLT) GO TO 1230
1327      TIME=T/LC
1328      NP=N+NSTART
1329      CALL PLOT (TITLE,TIME,NP,IVPTS)
1330      WRITE (6,1810) NP
1331 C
1332 C      CHECK FOR CONVERGENCE OF THE STEADY STATE SOLUTION
1333 C
1334  1230 IF (DOM.LT.TCONV) GO TO 1260
1335      IF (ISTOP.NE.0) GO TO 1260
1336      IF (N.EQ.NMAX) GO TO 1260
1337      IF (NC.EQ.NPRINT) NC=0
1338      IF (NPC.EQ.NPLT) NPC=0
1339  1240 NNN=N1
1340      N1=N2
1341      N2=NNN
1342  1250 CONTINUE
1343 C
1344 C      PUNCH(WRITE) A $IVS NAMELIST FOR RESTART
1345 C
1346  1260 IF (NPLT.GE.0) CALL ADV (10)
1347      IF (IPUNCH.EQ.0) GO TO 10
1348      DO 1270 L=1,LMAX
1349          PL(L,N3)=PL(L,N3)/PC
1350          ROL(L,N3)=ROL(L,N3)*G
1351      DO 1270 M=1,MMAX
1352          P(L,M,N3)=P(L,M,N3)/PC
1353          RO(L,M,N3)=RO(L,M,N3)*G
1354  1270 CONTINUE
1355      WRITE (8,1620) NP,TIME
1356      DO 1280 M=1,MMAX
1357          WRITE (8,1630) M,U(1,M,N3)
1358          WRITE (8,1650) (U(L,M,N3),L=2,LMAX)
1359  1280 CONTINUE
1360      DO 1290 M=1,MMAX
1361          WRITE (8,1660) M,V(1,M,N3)
1362          WRITE (8,1650) (V(L,M,N3),L=2,LMAX)
1363  1290 CONTINUE
1364      DO 1300 M=1,MMAX

```

```

1365     WRITE (8,1680) M,P(1,M,N3)
1366     WRITE (8,1700) (P(L,M,N3),L=2,LMAX)
1367 1300 CONTINUE
1368     DO 1310 M=1,MMAX
1369     WRITE (8,1710) M,RO(1,M,N3)
1370     WRITE (8,1730) (RO(L,M,N3),L=2,LMAX)
1371 1310 CONTINUE
1372     IF (ITM.LE.1) GO TO 1340
1373     DO 1320 M=1,MMAX
1374     WRITE (8,1740) M,Q(1,M,N3)
1375     WRITE (8,1760) (Q(L,M,N3),L=2,LMAX)
1376 1320 CONTINUE
1377     IF (ITM.EQ.2) GO TO 1340
1378     DO 1330 M=1,MMAX
1379     WRITE (8,1750) M,E(1,M,N3)
1380     WRITE (8,1760) (E(L,M,N3),L=2,LMAX)
1381 1330 CONTINUE
1382 1340 IF (MDFS.EQ.0) GO TO 1350
1383     LDFSSP1=LDFSS+1
1384     WRITE (8,1640) LDFSS,UL(LDFSS,N3)
1385     WRITE (8,1650) (UL(L,N3),L=LDFSSP1,LDFSF)
1386     WRITE (8,1670) LDFSS,VL(LDFSS,N3)
1387     WRITE (8,1650) (VL(L,N3),L=LDFSSP1,LDFSF)
1388     WRITE (8,1690) LDFSS,PL(LDFSS,N3)
1389     WRITE (8,1700) (PL(L,N3),L=LDFSSP1,LDFSF)
1390     WRITE (8,1720) LDFSS,ROL(LDFSS,N3)
1391     WRITE (8,1730) (ROL(L,N3),L=LDFSSP1,LDFSF)
1392     IF (ITM.LE.1) GO TO 1350
1393     WRITE (8,1770) LDFSS,QL(LDFSS,N3)
1394     WRITE (8,1760) (QL(L,N3),L=LDFSSP1,LDFSF)
1395     IF (ITM.EQ.2) GO TO 1350
1396     WRITE (8,1780) LDFSS,EL(LDFSS,N3)
1397     WRITE (8,1760) (EL(L,N3),L=LDFSSP1,LDFSF)
1398 1350 WRITE (8,1790)
1399     NCARDS=(LMAX/7+2)*MMAX*4+2+LDFSF-LDFSS
1400     WRITE (6,1800) NCARDS
1401     GO TO 10
1402 C
1403 C     FORMAT STATEMENTS
1404 C
1405 1370 FORMAT (10A8)
1406 1380 FORMAT (1H1)
1407 1390 FORMAT (~H )
1408 1400 FORMAT (1H0)
1409 1410 FORMAT (1H0,10X,47HVNAP2, A COMPUTER PROGRAM FOR THE COMPUTATION O
1410     1 .59HF TWO-DIMENSIONAL, TIME-DEPENDENT, COMPRESSIBLE, TURBULENT,5H
1411     2 FLOW,/.37X,57HBY MICHAEL C. CLINE, T-3 - LOS ALAMOS NATIONAL LABO
1412     3RATORY)
1413 1420 FORMAT (1H0,10X,18HPROGRAM ABSTRACT -,.//26X,17HTHE NAVIER-STOKES,6
1414     1 2H EQUATIONS FOR TWO-DIMENSIONAL, TIME-DEPENDENT FLOW ARE SOLVED,
1415     2 10H USING THE,/.21X,62HSECOND-ORDER, MACCORMACK FINITE-DIFFERENCE
1416     3 SCHEME. ALL BOUNDAR,31HY CONDITIONS ARE COMPUTED USING,/.21X,13HA
1417     4 SECOND-ORDE,62HR, REFERENCE PLANE CHARACTERISTIC SCHEME WITH THE
1418     5VISCOSUS TERM,19HS TREATED AS SOURCE)
1419 1430 FORMAT (1H .20X,41HFUNCTIONS. THE FLUID IS ASSUMED TO BE A ,54HPE
1420     1RFECT GAS. THE STEADY-STATE SOLUTION IS OBTAINED AS,/.21X,62HTHE
1421     2ASYMPTOTIC SOLUTION FOR LARGE TIME. THE FLOW BOUNDARIES M,34HAY B
1422     3E ARBITRARY CURVED SOLID WALLS,/.21X,62HAS WELL AS JET ENVELOPES.
1423     4 THE GEOMETRY MAY CONSIST OF SINGLE ,36HAND DUAL FLOWING STREAMS.
1424     5TURBULFNCE,/.21X,62HEFFECTS ARE MODELED WITH EITHER A MIXING-LENGT
1425     6H, A TURBULENCE ,32HENERGY EQUATION, OR A TURBULENCE,/.21X,62HENER
1426     7GY-DISSIPATION RATE EQUATIONS MODEL. THIS PROGRAM ALLOWS ,34HVARI
1427     8ABLE GRID SPACING A'D INCLUDES,/.21X,17HOPTIONS TO SPEED ,50HUP TH
1428     9E CALCULATION FOR HIGH REYNOLDS NUMBER FLOWS.)
1429 1440 FORMAT (1H0,10X,11HJOB TITLE -//21X,10A8)
1430 1450 FORMAT (1H0,10X,20HCONTROL PARAMETERS -)
1431 1460 FORMAT (1H0,20X,5H1LMAX=,12,2X,5HMMAX=,12,3X,5H1NMAX=,14,2X,7HNPRINT
1432     1=,14,2X,6HNPL0T=,14,6X,4HFDT=,F4,2,2X,5HFDT1=,F4,2,3X,5HFDTI=,F4,2
1433     2 ,2X,7HIPUNCH=,11,/.21X,4HIUI=,11,4X,4HIUO=,11,5X,6HIVPTS=,11,4X,7
1434     3 HNCONVI=,12,4X,6HTSTOP=,E8,2,2X,4HNID=,12,4X,6HTCONV=,F5,3,1X,5HN
1435     4ASM=,11,5X,6HIUNIT=,11,/.21X,6HRSTAR=,F11,6,2X,7HRSTARS=,F13,7,4X,
1436     5 5HPLOW=,F6,4,5X,6HROLOW=,F11,6,5X,4HVDT=,F4,2,3X,5HVDT1=,F4,2)

```

```

1437 1470 FORMAT (1HO,10X,13HFLUID MODEL -./21X,36HTHE RATIO OF SPECIFIC HE
1438 1ATS, GAMMA =.F6.4,26H AND THE GAS CONSTANT, R =.F9.4,15H (FT-LBF/L
1439 2BN-R))
1440 1480 FORMAT (1HO,10X,13HFLUID MODEL -./21X,36HTHE RATIO OF SPECIFIC HE
1441 1ATS, GAMMA =.F6.4,26H AND THE GAS CONSTANT, R =.F9.4,9H (J/KG-K))
1442 1490 FORMAT (1HO,10X,15HFLOW GEGMETRY -)
1443 1500 FORMAT (1HO,20X,47HTWO-DIMENSIONAL, PLANAR FLOW HAS BEEN SPECIFIED
1444 1 )
1445 1510 FORMAT (1HO,20X,36HAXISYMMETRIC FLOW HAS BEEN SPECIFIED)
1446 1520 FORMAT (1H ,30HINITIAL-DATA SURFACE - TIME = .F12.8,8H SECONDS,4H
1447 1(N=.16,1H))
1448 1530 FORMAT (1HO,11X,1HL,4X,1HM,9X,1HX,10X,1HY,10X,1HU,11X,1HV,12X,1HP,
1449 1 11X,3HRHO,7X,4HVMA,10X,4HMACH,8X,1HT)
1450 1540 FORMAT (1H ,25X,4H(IN),7X,4H(IN),6X,5H(F/S),7X,5H(F/S),7X,6H(PA)
1451 1 ,6X,9H(LBM/FT3),4X,5H(F/S),10X,2HNO,8X,3H(R))
1452 1550 FORMAT (1H ,25X,4H(CM),7X,4H(CM),6X,5H(M/S),7X,5H(M/S),7X,6H (KPA)
1453 1 ,7X,7H(KG/M3),5X,5H(M/S),10X,2HNO,8X,3H(K))
1454 1560 FORMAT (1H ,7X,215.4F12.4,F13.5,F12.6,3F12.4)
1455 1570 FCRMAT (1H ,20HSLUTION SURFACE NO.,16,3H - .7HTIME = .F12.8,20H S
1456 1ECOND, (DELTA T = .F10.8,8H, NVCM =.13,9H, CNUMS =.F5.2,3H, (.12,1
1457 2 H,.12,4H), (.12,1H,.12,2H))
1458 1580 FORMAT (1HO,10X,21HBOUNDARY CONDITIONS -./22X,1HM,10X,8HPT(PA),
1459 1 11X,5HTT(R),10X,10HTHETA( DEG),10X,8HPE(PA),7X,11HFSO(FT2/S2),7X
1460 2 .!1HFSE(FT2/S3),/)
1461 1590 FORMAT (1HO,10X,21HBOUNDARY CONDITIONS -./22X,1HM,10X,7HPT(KPA),1
1462 1 2X,5HTT(K),10X,10HTHETA( DEG),10X,7HPE(KPA),8X,10HFSO(M2/S2),8X,10
1463 2 HFSE(M2/S3),/)
1464 1600 FORMAT (1H ,20X,12,7X,F10.4,10X,F7.2,10X,F7.2,9X,F11.5,F18.4,F18.1
1465 1 )
1466 1610 FORMAT (1HO,51H***** THE RADIUS OF THE CENTERBODY IS LARGER THAN T
1467 1 ,20HHE WALL RADIL'S *****)
1468 1620 FORMAT (1X,18H$IVS NID=O,NSTART=.16,8H,TSTART=.F14.10,1H,)
1469 1630 FORMAT (1X,4HU(1.,12.5H,1) =.F10.3,1H,)
1470 1640 FORMAT (1X,3HVL(.12.5H,1) =.F10.3,1H,)
1471 1650 FORMAT ((1X,7(F10.3,1H,)))
1472 1660 FORMAT (1X,4HV(1.,12.5H,1) =.F10.3,1H,)
1473 1670 FURHAT (1X,3HVL(.12.5H,1) =.F10.3,1H,)
1474 1680 FORMAT (1X,4HP(1.,12.5H,1) =.F10.4,1H,)
1475 1690 FORMAT (1X,3HPL(.12.5H,1) =.F10.4,1H,)
1476 1700 FORMAT ((1X,7(F10.4,1H,)))
1477 1710 FORMAT (1X,5HRO(1.,12.5H,1) =.F10.6,1H,)
1478 1720 FORMAT (1X,4HRDL(.12.5H,1) =.F10.6,1H,)
1479 1730 FORMAT ((1X,7(F10.6,1H,)))
1480 1740 FORMAT (1X,4HO(1.,12.5H,1) =.E10.4,1H,)
1481 1750 FORMAT (1X,4HE(1.,12.5H,1) =.E10.4,1H,)
1482 1760 FORMAT ((1X,7(E10.4,1H,)))
1483 1770 FORMAT (1X,3HOL(.12.5H,1) =.E10.4,1H,)
1484 1780 FCRMAT (1X,3HEL(.12.5H,1) =.E10.4,1H,)
1485 1790 FORMAT (1X,1H$)
1486 1800 FORMAT (1HO,27H***** EXPECT APPROXIMATELY .14,20H PUNCHED CARDS **
1487 1***)
1488 1810 FORMAT (1HO,31H***** EXPECT FILM OUTPUT FOR N=.16,6H *****)
1489 1820 FORMAT (1H ,10X,2HN=.16,5H, T=.F12.8,14H SECONDS, DT=.F10.8,8H S
1490 1ECOND,9H, NVCM =.13,10H, CNUMS =.F5.2,4H, (.12,1H,.12,5H), (
1491 2 .12,1H,.12,1H))
1492 1830 FORMAT (1HO,10X,21HARTIFICIAL VISCOSITY -./21X,4HCAV=.F4.2,3X,4HXM
1493 1U=.F4.2,3X,4HXL=.F4.2,3X,4HPRA=.F4.2,3X,4HXRO=.F4.2,3X,4HLSS=.12,
1494 2 5X,4HLSF=.13,3X,6HIDIVC=.11,3X,4HISS=.11,3X,6HSMACH=.F4.2,/,21X,4
1495 3 HNST=.14,3X,4HSMP=.F4.2,3X,5HSMPP=.F4.2,2X,5HSMT=.F4.2,2X,6HSMPT
1496 4F=.F4.2,1X,5HNTST=.14,2X,4HIAV=.11,5X,4HMSS=.12,4X,4HMSF=.12)
1497 1840 FORMAT (1HO,20X,29HFREE-SLIP WALLS ARE SPECIFIED)
1498 1850 FCRMAT (1HO,20X,27HNO-SLIP WALLS ARE SPECIFIED)
1499 1860 FORMAT (1HO,10X,21HMOLECULAR VISCOSITY -./21X,4HCMU=.E10.4,18H (L
1500 1BF-S/FT2) CLA=.E11.4,17H (LBF-S/FT2) CK=.E10.4,16H (LBF/S-R) EM
1501 2U=.F4.2,6H ELA=.F4.2,5H EK=.F4.2)
1502 1870 FORMAT (1HO,10X,21HMOLECULAR VISCOSITY -./21X,4HCMU=.E10.4,13H (P
1503 1A-S) CLA=.E11.4,12H (PA-S) CK=.E10.4,14H (W/M-K) EMU=.F4.2,6H
1504 2ELA=.F4.2,5H EK=.F4.2)
1505 1880 FORMAT (1H ,10X,48H-----
1506 1-.61H-----)
1507 2 .7H-----)
1508 1890 FORMAT (1H ,20X,33HADIBATIC UPPER WALL IS SPECIFIED)

```

```

1509 1900 FORMAT (1H .20X.15HTW IS SPECIFIED)
1510 1910 FORMAT (1H .20X.39HADIABATIC LOWER CENTERBODY IS SPECIFIED)
1511 1920 FORMAT (1H .20X.44HADIABATIC LOWER DUAL FLOW SPACE BOUNDARY IS .9H
1512 1SPECIFIED)
1513 1930 FORMAT (1H .20X.44HADIABATIC UPPER DUAL FLOW SPACE BOUNDARY IS .9H
1514 1SPECIFIED)
1515 1940 FORMAT (1H .20X.16HTCB IS SPECIFIED)
1516 1950 FORMAT (1H .20X.15HTL IS SPECIFIED)
1517 1960 FORMAT (1H .20X.15HTU IS SPECIFIED)
1518 1970 FORMAT (1HO.10X.18HTURBULENCE MODEL -./21X.21HNO MODEL IS SPECIFI
1519 1ED)
1520 1980 FORMAT (1HO.10X.18HTURBULENCE MODEL -./21X.38HMIXING-LENGTH MODEL
1521 1 IS SPECIFIED. CAL=.F4.2.2X.5HIMLM=.12.2X.5HCML1=.F5.3.2X.5HCML2=
1522 2 .F5.3.2X.4HPRT=.F4.2)
1523 1990 FORMAT (1HO.10X.18HTURBULENCE MODEL -./21X.45HTURBULENCE ENERGY E
1524 1QUATION MODEL IS SPECIFIED)
1525 2000 FORMAT (1HO.20X.4HCAL=.F4.2.2X.4HCOL=.F5.2.2X.5HCOMU=.F4.2.2X.5HIM
1526 1LM=.12.2X.5HCML1=.F5.3.2X.5HCML2=.F5.3.2X.4HPRT=.F4.2)
1527 2010 FORMAT (1HO.10X.18HTURBULENCE MODEL -./21X.62HTURBULENCE ENERGY -
1528 1 DISSIPATION RATE EQUATIONS MODEL IS SPECIF.3HIFD)
1529 2020 FORMAT (1HO.20X.7HIINLET=.11.2X.7HIEXIT=.11.2X.4HIEX=.11.5X.7HISU
1530 1PER=.11.2X.4HDYW=.F6.4.2X.5HIVBC=.11.2X.5HINBC=.11.2X.6HIWALL=.11.
1531 2 2X.7HIWALLO=.11.2X.4HALI=.F4.2.2X.4HALE=.F4.2./21X.4HALW=.F4.2.2
1532 3 X.6HNSTAG=.11.3X.4HNPE=.14.2X.4HPRI=.F10.5)
1533 2030 FORMAT (1HO.20X.4HCAL=.F4.2.2X.5HCOMU=.F4.2.2X.3HC1=.F4.2.2X.3HC2=
1534 1 .F4.2.2X.5HSIGO=.F4.2.2X.5HSIGE=.F4.2.2X.5HBFSI=.F4.2.2X.4HPRT=
1535 2 .F4.2.2X.5HSTBO=.F6.4.2X.5HSTBE=.F6.4)
1536 2040 FORMAT (1HO.10X.26HVARIABLE GRID PARAMETERS -./21X.4HIST=.11.3X.5
1537 1 HMVCB=.12.3X.5HMVCT=.12.3X.4HIQS=.11.3X.6HNIQSS=.11.3X.6HNIQSF=
1538 2 .11.3X.6HNVCMI=.13.3X.6HILLOS=.12.3X.4HSQS=.F5.2.3X.4HCQS=.F5.3)
1539 2050 FORMAT (1HO.53H***** INCOMPATIBLE TURBULENCE MODEL - GEOMETRY PARA
1540 1METERS *****)
1541 2060 FORMAT (1HO.51H***** INCOMPATIBLE DUAL FLOW SPACE PARAMETERS *****
1542 1 )
1543 2070 FORMAT (1HO.29H***** NVCM1 MUST BE ODD ***** )
1544 2080 FORMAT (1HO.52H***** INCOMPATIBLE DUAL FLOW SPACE - SURCYCLED GRID
1545 1 .16HPARAMETERS *****)
1546 2090 FORMAT (1HO.50H***** INCOMPATIBLE SURCYCLED GRID PARAMETERS *****)
1547 2100 FORMAT (1HO.35H***** NVCM IS GREATER THAN 200 AT N=.16.34H. CHECK
1548 1LAST SOLUTION PLANE. *****)
1549 2110 FORMAT (1H .13X.4HNVC=.13.5X.2HP(.12.1H..12.2H)=.F10.5.5X.2HP(.12.
1550 1 1H..12.2H)=.F10.5.5X.2HP(.12.1H..12.2H)=.F10.5)
1551 2120 FORMAT (1H .10X.2HN=.16.13X.2HP(.12.1H..12.2H)=.F10.5.5X.2HP(.12.1
1552 1 H..12.2H)=.F10.5.5X.2HP(.12.1H..12.2H)=.F10.5.5X.5HNFTST=.15)
1553 2130 FORMAT (1HO.48H***** INCOMPATIBLE QUICK SOLVER PARAMETERS *****)
1554 2140 FORMAT (1HO.53H***** ISUPER MUST BE GREATER THAN OR EQUAL TO 0 ***
1555 1** )
1556 2150 FORMAT (1HO.65H***** INCOMPATIBLE WALL GEOMETRY AND/OR BOUNDARY CO
1557 1NDITIONS *****)
1558 END

```

```

1559      SUBROUTINE GEOM
1560 C
1561 C      .....
1562 C
1563 C      THIS SUBROUTINE CALCULATES THE WALL RADIUS AND SLOPE
1564 C
1565 C      .....
1566 C
1567 *CALL MCC
1568      GO TO (10,30,120,170), NGEOM
1569 C
1570 C      CGNSTANT AREA WALL CASE
1571 C
1572      10 WRITE (6,230)
1573      IF (IUI.EQ.1) WRITE (6,250) XI,RI,XE
1574      IF (IUI.EQ.2) WRITE (6,260) XI,RI,XE
1575      LT=LMAX
1576      XT=XE
1577      RT=RI
1578      RE=RI
1579      DO 20 L=1,LMAX
1580      YW(L)=RI
1581      NXNY(L)=0.0
1582      20 CONTINUE
1583      GO TO 210
1584 C
1585 C      CIRCULAR-ARC, CONICAL WALL CASE
1586 C
1587      30 WRITE (6,230)
1588      IF (RCI.EQ.0.0.OR.RCT.EQ.0.0) GO TO 200
1589      ANI=ANGI*3.141593/180.0
1590      ANE=ANGE*3.141593/180.0
1591      XTAN=XI+RCI*SIN(ANI)
1592      RTAN=RI+RCI*(COS(ANI)-1.0)
1593      RT1=RT-RCT*(COS(ANI)-1.0)
1594      XT1=XTAN+(RTAN-RT1)/TAN(ANI)
1595      IF (XT1.GE.XTAN) GO TO 40
1596      XT1=XTAN
1597      RT1=RTAN
1598      40 XT=XT1+RCT*SIN(ANI)
1599      XT2=XT+RCT*SIN(ANE)
1600      RT2=RT+RCT*(1.0-COS(ANE))
1601      RE=RT2+(XE-XT2)*TAN(ANE)
1602      LT=1
1603      IF (IUI.EQ.1) WRITE (6,270) XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XT,RE
1604      IF (IUI.EQ.2) WRITE (6,280) XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XT,RE
1605      DO 110 L=1,LMAX
1606      IF (XP(L).LE.XTAN) GO TO 50
1607      IF (XP(L).GT.XTAN.AND.XP(L).LE.XT1) GO TO 60
1608      IF (XP(L).GT.XT1.AND.XP(L).LE.XT) GO TO 70
1609      IF (XP(L).GT.XT.AND.XP(L).LE.XT2) GO TO 80
1610      GO TO 90
1611 C
1612      50 YW(L)=RI+RCI*(COS(ASIN((XP(L)-XI)/RCI))-1.0)
1613      NXNY(L)=(XP(L)-XI)/(YW(L)-RI+RCI)
1614      GO TO 100
1615 C
1616      60 YW(L)=RT1+(XT1-XP(L))*TAN(ANI)
1617      NXNY(L)=TAN(ANI)
1618      GO TO 100
1619 C
1620      70 YW(L)=RT+RCT*(1.0-COS(ASIN((XT-XP(L))/RCT)))
1621      NXNY(L)=(XT-XP(L))/(RCT+RT-YW(L))
1622      GO TO 100
1623 C
1624      80 YW(L)=RT+RCT*(1.0-COS(ASIN((XP(L)-XT)/RCT)))
1625      NXNY(L)=(XT-XP(L))/(RCT+RT-YW(L))
1626      GO TO 100
1627 C
1628      90 YW(L)=RT2+(XP(L)-XT2)*TAN(ANE)
1629      NXNY(L)=-TAN(ANE)
1630 C

```

```

1631 100 IF (L.EQ.1) GO TO 110
1632     IF (YW(L).LT.YW(LT)) LT=L
1633 110 CONTINUE
1634     GO TO 210
1635 C
1636 C     GENERAL WALL CASE - INPUT WALL COORDINATES
1637 C
1638 120 WRITE (6,240)
1639     WRITE (6,230)
1640     YW(1)=YWI(1)
1641     YW(LMAX)=YWI(NWPTS)
1642     RI=YW(1)
1643     RE=YW(LMAX)
1644     LT=1
1645     DO 130 L=2,NWPTS
1646     IF (YWI(L).LE.YWI(LT)) LT=L
1647 130 CONTINUE
1648     XT=XWI(LT)
1649     RT=YWI(LT)
1650     IF (IUI.EQ.1) WRITE (6,290) XT,RT,IINT,IDIF
1651     IF (IUI.EQ.2) WRITE (6,300) XT,RT,IINT,IDIF
1652     LT=1
1653     LI=LMAX-1
1654     IPP=1
1655     DO 140 L=2,LI
1656     CALL MTLUP (XP(L),YW(L),IINT,NWPTS,NWPTS,1,IPP,XWI,YWI)
1657     IF (L.EQ.1) GO TO 140
1658     IF (YW(L).LE.YW(LT)) LT=L
1659 140 CONTINUE
1660     LDUM=NWPTS
1661     IF (LMAX.GT.NWPTS) LDUM=LMAX
1662     DO 160 L=1,LDUM
1663     IF (L.GT.LMAX) GO TO 150
1664     SLOPE=DIF(L,IDIF,LMAX,XP,YW)
1665     NXNY(L)=-SLOPE
1666 150 IF (L.LE.NWPTS.AND.L.LE.LMAX) WRITE (6,330) L,XWI(L),YWI(L),XP(L)
1667     1 ,YW(L),SLOPE
1668     IF (L.GT.NWPTS.AND.L.LE.LMAX) WRITE (6,340) L,XP(L),YW(L),SLOPE
1669     IF (L.LE.NWPTS.AND.L.GT.LMAX) WRITE (6,350) L,XWI(L),YWI(L)
1670 160 CONTINUE
1671     GO TO 210
1672 C
1673 C     GENERAL WALL CASE - INPUT WALL RADIUS AND SLOPE
1674 C
1675 170 WRITE (6,240)
1676     WRITE (6,230)
1677     RI=YW(1)
1678     RE=YW(LMAX)
1679     LT=1
1680     DO 180 L=2,LMAX
1681     IF (YW(L).LE.YW(LT)) LT=L
1682 180 CONTINUE
1683     XT=XP(LT)
1684     RT=YW(LT)
1685     IF (IUI.EQ.1) WRITE (6,310) XT,RT
1686     IF (IUI.EQ.2) WRITE (6,320) XT,RT
1687     DO 190 L=1,LMAX
1688     SLOPE=-NXNY(L)
1689     WRITE (6,360) L,XP(L),YW(L),SLOPE
1690 190 CONTINUE
1691     GO TO 210
1692 C
1693 200 WRITE (6,390)
1694     IERR=1
1695     RETURN
1696 C
1697 210 IF (JFLAG.EQ.0) RETURN
1698     XWL=XP(LJET-1)
1699     IF (JFLAG.EQ.-1) GO TO 220
1700     IF (IUI.EQ.1) WRITE (6,370) XWL,LJET,LMAX
1701     IF (IUI.EQ.2) WRITE (6,380) XWL,LJET,LMAX
1702     RETURN

```

```

1703 220 IF (IUI.EQ.1) WRITE (6.400) XWL
1704 IF (IUI.EQ.2) WRITE (6.410) XWL
1705 RETURN
1706 C
1707 C FORMAT STATEMENTS
1708 C
1709 230 FORMAT (1HO,10X,15HDUCT GEOMETRY -)
1710 240 FORMAT (1H1)
1711 250 FORMAT (1HO,20X,46HA CONSTANT AREA DUCT HAS BEEN SPECIFIED BY XI=
1712 1 .FB.4,10H (IN), RI=.FB.4,14H (IN), AND XE=.FB.4,5H (IN))
1713 260 FORMAT (1HO,20X,46HA CONSTANT AREA DUCT HAS BEEN SPECIFIED BY XI=
1714 1 .FB.4,10H (CM), RI=.FB.4,14H (CM), AND XE=.FB.4,5H (CM))
1715 270 FORMAT (1HO,20X,56HA CIRCULAR-ARC, CONICAL NOZZLE HAS BEEN SPECIFI
1716 1 ED BY XI=.FB.4,10H (IN), RI=.FB.4,6H (IN)../.21X,3HRT=.FB.4,10H (I
1717 2N), XE=.FB.4,11H (IN), RCI=.FB.4,11H (IN), RCT=.FB.4,12H (IN), ANG
1718 3I=.F6.2,7H (DEG)../.21X,9HAND ANGE=.F6.2,35H (DEG). THE COMPUTED V
1719 4ALUES ARE XT=.FB.4,13H (IN) AND RE=.FB.4,6H (IN).)
1720 280 FORMAT (1HO,20X,56HA CIRCULAR-APC, CONICAL NOZZLE HAS BEEN SPECIFI
1721 1 ED BY XI=.FB.4,10H (CM), RI=.FB.4,6H (CM)../.21X,3HRT=.FB.4,10H (C
1722 2M), XE=.FB.4,11H (CM), RCI=.FB.4,11H (CM), RCT=.FB.4,12H (CM), ANG
1723 3I=.F6.2,7H (DEG)../.21X,9HAND ANGE=.F6.2,35H (DEG). THE COMPUTED V
1724 4ALUES ARE XT=.FB.4,13H (CM) AND RE=.FB.4,6H (CM).)
1725 290 FORMAT (1HO,20X,45HA GENERAL WALL HAS BEEN SPECIFIED BY THE FOLL,2
1726 1 HOWING PARAMETERS, XT=.FB.4,10H (IN), RI=.FB.4,6H (IN)../.21X,5H
1727 2IINT=.I1,7H, IDIF=.I1,1H..//22X,1HL,10X,7HXWI (IN),10X,7HYWI (IN),11
1728 3 X,6HXP (IN),11X,6HYW (IN),12X,5HSLOPE,/)
1729 300 FORMAT (1HO,20X,45HA GENERAL WALL HAS BEEN SPECIFIED BY THE FOLL,2
1730 1 HOWING PARAMETERS, XT=.FB.4,10H (CM), RI=.FB.4,6H (CM)../.21X,5H
1731 2IINT=.I1,7H, IDIF=.I1,1H..//22X,1HL,10X,7HXWI (CM),10X,7HYWI (CM),11
1732 3 X,6HXP (CM),11X,6HYW (CM),12X,5HSLOPE,/)
1733 310 FORMAT (1HO,20X,45HA GENERAL WALL HAS BEEN SPECIFIED BY THE FOLL,2
1734 1 HOWING PARAMETERS, XT=.FB.4,10H (IN), RI=.FB.4,6H (IN)..//22X,1H
1735 2L,11X,6HXP (IN),11X,6HYW (IN),12X,5HSLOPE,/)
1736 320 FORMAT (1HO,20X,45HA GENERAL WALL HAS BEEN SPECIFIED BY THE FOLL,2
1737 1 HOWING PARAMETERS, XT=.FB.4,10H (CM), RI=.FB.4,6H (CM)..//22X,1H
1738 2L,11X,6HXP (CM),11X,6HYW (CM),12X,5HSLOPE,/)
1739 330 FORMAT (1H .20X,12.7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
1740 340 FORMAT (1H .20X,12.41X,F10.4,7X,F10.4,7X,F10.4)
1741 350 FORMAT (1H .20X,12.7X,F10.4,7X,F10.4)
1742 360 FORMAT (1H .20X,12.7X,F10.4,7X,F10.4,7X,F10.4)
1743 370 FORMAT (1HO,20X,43HA FREE-JET CALCULATION HAS BEEN REQUESTED. .20H
1744 1 THE WALL ENDS AT X=.FB.4,11H (IN). THE./.21X,14H MESH POINTS L=
1745 2 .I3,6H TO L=.I3,55H ARE AN INITIAL APPROXIMATION TO THE FREE-JET
1746 3BOUNDARY.)
1747 380 FORMAT (1HO,20X,43HA FREE-JET CALCULATION HAS BEEN REQUESTED. .20H
1748 1 THE WALL ENDS AT X=.FB.4,11H (CM). THE./.21X,14H MESH POINTS L=
1749 2 .I3,6H TO L=.I3,55H ARE AN INITIAL APPROXIMATION TO THE FREE-JET
1750 3BOUNDARY.)
1751 390 FORMAT (1HO,44H***** RCI OR RCT WAS SPECIFIED AS ZERO *****)
1752 400 FORMAT (1HO,20X,54H THE WALL CONTOUR HAS AN EXPANSION CORNER LOCATE
1753 1D AT X=.FB.4,6H (IN).)
1754 410 FORMAT (1HO,20X,54H THE WALL CONTOUR HAS AN EXPANSION CORNER LOCATE
1755 1D AT X=.FB.4,6H (CM).)
1756 END

```

```

1757      SUBROUTINE GEOMCB
1758 C
1759 C      .....
1760 C
1761 C      THIS SUBROUTINE CALCULATES THE CENTERBODY RADIUS AND SLOPE
1762 C
1763 C      .....
1764 C
1765 *CALL MCC
1766      GO TO (10,30,120,160), NGCB
1767 C
1768 C      CYLINDRICAL CENTERBODY CASE
1769 C
1770      10 IF (IUI.EQ.1) WRITE (6,210) XICB,RICB,XECB
1771          IF (IUI.EQ.2) WRITE (6,220) XICB,RICB,XECB
1772          DO 20 L=1,LMAX
1773              YCB(L)=RICB
1774              NXNYCB(L)=0.0
1775      20 CONTINUE
1776          RETURN
1777 C
1778 C      CIRCULAR-ARC, CONICAL CENTERBODY CASE
1779 C
1780      30 RICB=2.0*RTCB-RICB
1781          IF (RCICB.EQ.0.0.OR.RCTCB.EQ.0.0) GO TO 130
1782          ANI=ANGICB*3.141593/180.0
1783          ANE=ANGECB*3.141593/180.0
1784          XTAN=XICB+RCICB*SIN(ANI)
1785          RTAN=RICB+RCICB*(COS(ANI)-1.0)
1786          RT1=RTCB-RCTCB*(COS(ANI)-1.0)
1787          XT1=XTAN+(RTAN-RT1)/TAN(ANI)
1788          IF (XT1.GE.XTAN) GO TO 40
1789          XT1=XTAN
1790          RT1=RTAN
1791      40 XTCB=XT1+RCTCB*SIN(ANI)
1792          XT2=XTCB+RCTCB*SIN(ANE)
1793          RT2=RTCB+RCTCB*(1.0-COS(ANE))
1794          RECB=RT2+(XECB-XT2)*TAN(ANE)
1795          RICB=2.0*RTCB-RICB
1796          RECB=2.0*RTCB-RECB
1797          IF (IUI.EQ.1) WRITE (6,230) XICB,RICB,RTCB,XECB,RCICB,RCTCB,ANGICB
1798      1 ,ANGECEB,XTCB,RECB
1799          IF (IUI.EQ.2) WRITE (6,240) XICB,RICB,RTCB,XECB,RCICB,RCTCB,ANGICB
1800      1 ,ANGECEB,XTCB,RECB
1801          RICB=2.0*RTCB-RICB
1802          RECB=2.0*RTCB-RECB
1803          DO 110 L=1,LMAX
1804              IF (XP(L).LE.XTAN) GO TO 50
1805              IF (XP(L).GT.XTAN.AND.XP(L).LE.XT1) GO TO 60
1806              IF (XP(L).GT.XT1.AND.XP(L).LE.XTCB) GO TO 70
1807              IF (XP(L).GT.XTCB.AND.XP(L).LE.XT2) GO TO 80
1808              GO TO 90
1809 C
1810      50 YCB(L)=RICB+RCICB*(COS(ASIN((XP(L)-XICB)/RCICB))-1.0)
1811          NXNYCB(L)=(XP(L)-XICB)/(YCB(L)-RICB+RCICB)
1812          GO TO 100
1813 C
1814      60 YCB(L)=RT1+(XT1-XP(L))*TAN(ANI)
1815          NXNYCB(L)=TAN(ANI)
1816          GO TO 100
1817 C
1818      70 YCB(L)=RTCB+RCTCB*(1.0-COS(ASIN((XTCB-XP(L))/RCTCB)))
1819          NXNYCB(L)=(XTCB-XP(L))/(RCTCB+RTCB-YCB(L))
1820          GO TO 100
1821 C
1822      80 YCB(L)=RTCB+RCTCB*(1.0-COS(ASIN((XP(L)-XTCB)/RCTCB)))
1823          NXNYCB(L)=(XTCB-XP(L))/(RCTCB+RTCB-YCB(L))
1824          GO TO 100
1825 C
1826      90 YCB(L)=RT2+(XP(L)-XT2)*TAN(ANE)
1827          NXNYCB(L)=-TAN(ANE)
1828 C

```



```

1829 100 YCB(L)=2.0*RTCB-YCB(L)
1830     NXNYCB(L)=-NXNYCB(L)
1831     IF (YCB(L).GE.O.O.OR.NDIM.EQ.O) GO TO 110
1832     YCB(L)=O.O
1833     NXNYCB(L)=O.O
1834 110 CONTINUE
1835     RETURN
1836 C
1837 C     GENERAL CENTERBODY CASE - INPUT CENTERBODY COORDINATES
1838 C
1839 120 WRITE (6,200)
1840     IF (IUI.EQ.1) WRITE (6,250) IINTCB, IDIFCB
1841     IF (IUI.EQ.2) WRITE (6,260) IINICB, IOIFCB
1842     L1=LMAX-1
1843     IPP=1
1844     DO 130 L=1,LMAX
1845     CALL MTLUP (XP(L),YCB(L),IINTCB,NCBPTS,NCBPTS,1,IPP,XCBI,YCBI)
1846 130 CONTINUE
1847     LDUM=NCBPTS
1848     IF (LMAX.GT.NCBPTS) LDUM=LMAX
1849     DO 150 L=1,LDUM
1850     IF (L.GT.LMAX) GO TO 140
1851     SLOPE=DIF(L, IDIFCB, LMAX, XP, YCB)
1852     NXNYCB(L)=-SLOPE
1853     IF (YCB(L).GE.O.O.OR.NDIM.EQ.O) GO TO 140
1854     YCB(L)=O.O
1855     NXNYCB(L)=O.O
1856     SLOPE=-NXNYCB(L)
1857 140 IF (L.LE.NCBPTS.AND.L.LE.LMAX) WRITE (6,290) L, XCBI(L), YCBI(L), XP
1858     1 (L), YCB(L), SLOPE
1859     IF (L.GT.NCBPTS.AND.L.LE.LMAX) WRITE (6,300) L, XP(L), YCB(L), SLOPE
1860     IF (L.LE.NCBPTS.AND.L.GT.LMAX) WRITE (6,310) L, XCBI(L), YCBI(L)
1861 150 CONTINUE
1862     RETURN
1863 C
1864 C     GENERAL CENTERBODY CASE - INPUT CENTERBODY RADIUS AND SLOPE
1865 C
1866 160 WRITE (6,200)
1867     IF (IUI.EQ.1) WRITE (6,270)
1868     IF (IUI.EQ.2) WRITE (6,280)
1869     DO 180 L=1,LMAX
1870     IF (YCB(L).GE.O.O.OR.NDIM.EQ.O) GO TO 170
1871     YCB(L)=O.O
1872     NXNYCB(L)=O.O
1873 170 SLOPE=-NXNYCB(L)
1874     WRITE (6,320) L, XP(L), YCB(L), SLOPE
1875 180 CONTINUE
1876     RETURN
1877 C
1878 190 WRITE (6,330)
1879     IERR=1
1880     RETURN
1881 C
1882 C     FORMAT STATEMENTS
1883 C
1884 200 FORMAT (1H1)
1885 210 FORMAT (1HO,20X,52HA CYLINDRICAL CENTERBODY HAS BEEN SPECIFIED BY
1886     1XICB=.F8.4,12H (IN), RICB=.F8.4,16H (IN), AND XECB=.F8.4,5H (IN))
1887 220 FORMAT (1HO,20X,52HA CYLINDRICAL CENTERBODY HAS BEEN SPECIFIED BY
1888     1XICB=.F8.4,12H (CM), RICB=.F8.4,16H (CM), AND XECB=.F8.4,5H (CM))
1889 230 FORMAT (1HO,20X,62HA CIRCULAR-ARC, CONICAL CENTERBODY HAS BEEN SPE
1890     1CIFIED BY XICB=.F8.4,5H (IN),7H, RICB=.F8.4,6H (IN),./,21X,SHRTCB=
1891     2 ,F8.4,7H (IN), .5HXECB=.F8.4,5H (IN),8H, RCICB=.F8.4,5H (IN),8H,
1892     3RCTCB=.F8.4,5H (IN),9H, ANGICB=.F6.2,7H (DEG),./,21X,11HAND ANGECB
1893     4=.F6.2,8H (DEG), .29HTHE COMPUTED VALUES ARE XTCB=.F8.4,5H (IN),10
1894     5 H AND RECB=.F8.4,6H (IN).)
1895 240 FORMAT (1HO,20X,62HA CIRCULAR-ARC, CONICAL CENTERBODY HAS BEEN SPE
1896     1CIFIED BY XICB=.F8.4,5H (CM),7H, RICB=.F8.4,6H (CM),./,21X,SHRTCB=
1897     2 ,F8.4,7H (CM), .5HXECB=.F8.4,5H (CM),8H, RCICB=.F8.4,5H (C4),8H,
1898     3RCTCB=.F8.4,5H (CM),9H, ANGICB=.F6.2,7H (DEG),./,21X,11HAND ANGECB
1899     4=.F6.2,8H (DEG), .29HTHE COMPUTED VALUES ARE XTCB=.F8.4,5H (CM),10
1900     5 H AND RECB=.F8.4 6H (CM).)

```

```

1901 250 FORMAT (1H0,20X,47HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE
1902 1 ,25HFOLLOWING PARAMETERS. IINTCB=.11,5H, IDIFCB=.11,1H, //22X,1HL
1903 2 ,10X,8HXCB1(IN),10X,8HYCB1(IN),10X,6HXP(IN),10X,7HYCB(IN),11X,5HS
1904 3LOPE,/)
1905 260 FORMAT (1H0,20X,47HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE
1906 1 ,29HFOLLOWING PARAMETERS. IINTCB=.11,9H, IDIFCB=.11,1H, //22X,1HL
1907 2 ,10X,8HXCB1(CM),10X,8HYCB1(CM),10X,6HXP(CM),10X,7HYCB(CM),11X,5HS
1908 3LOPE,/)
1909 270 FORMAT (1H0,20X,47HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE
1910 1 ,21HFOLLOWING PARAMETERS. //22X,1HL,12X,6HXP(IN),10X,7HYCB(IN),11
1911 2 X,5HSLOPE,/)
1912 280 FORMAT (1H0,20X,47HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE
1913 1 ,21HFOLLOWING PARAMETERS. //22X,1HL,12X,6HXP(CM),10X,7HYCB(CM),11
1914 2 X,5HSLOPE,/)
1915 290 FORMAT (1H ,20X,12,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
1916 300 FORMAT (1H ,20X,12,41X,F10.4,7X,F10.4,7X,F10.4)
1917 310 FORMAT (1H ,20X,12,7X,F10.4,7X,F10.4)
1918 320 FORMAT (1H ,20X,12,7X,F10.4,7X,F10.4,7X,F10.4)
1919 330 FORMAT (1H0,48H***** RCICB OR RCTCB WAS SPECIFIED AS ZERO *****
1920 END

```

```

1921      SUBROUTINE GEOMLU
1922 C
1923 C .....
1924 C
1925 C      THIS SUBROUTINE CALCULATES THE DUAL FLOW SPACE BOUNDARY RADIUS
1926 C      AND SLOPES
1927 C
1928 C .....
1929 C
1930 *CALL MCC
1931      GO TO (10,100), NDFS
1932 C
1933 C      INPUT DUAL FLOW SPACE BOUNDARY COORDINATES
1934 C
1935      10 WRITE (6,120)
1936          WRITE (6,140)
1937          IF (IUI.EQ.1) WRITE (6,180) IINTDFS,IDIFDFS
1938          IF (IUI.EQ.2) WRITE (6,190) IINTDFS,IDIFDFS
1939          IPP=1
1940          DO 20 L=LDFSS,LDFSF
1941              CALL MTLUP (XP(L),YL(L),IINTDFS,NLPTS,NLPTS,1,IPP,XLI,YLI)
1942          20 CONTINUE
1943              LDUM=NLPTS
1944              IF (LDFSF.GT.NLPTS) LDUM=LDFSF
1945              LDF=0
1946              DO 30 L=LDFSS,LDFSF
1947                  LDF=LDF+1
1948                  XWI(LDF)=XP(L)
1949                  YWI(LDF)=YL(L)
1950          30 CONTINUE
1951              LMDF=LDFSF-LDFSS+1
1952              LDF=0
1953              DO 50 L=1,LDUM
1954                  LDFS=0
1955                  IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
1956                  IF (LDFS.EQ.0) GO TO 40
1957                  LDF=LDF+1
1958                  SLOPE=DIF(LDF,IDIFDFS,LMDF,XWI,YWI)
1959                  NXNYL(L)=-SLOPE
1960                  IF (YL(L).GE.0.O.OR.NDIM.EQ.0) GO TO 40
1961                  YL(L)=0.O
1962                  NXNYL(L)=0.O
1963                  SLOPE=-NXNYL(L)
1964          40 IF (L.LE.NLPTS.AND.LDFS.EQ.1) WRITE (6,220) L,XLI(L),YLI(L),XP(L)
1965              1 ,YL(L),SLOPE
1966              IF (L.GT.NLPTS.AND.LDFS.EQ.1) WRITE (6,230) L,XP(L),YL(L),SLOPE
1967              IF (L.LE.NLPTS.AND.LDFS.EQ.0) WRITE (6,240) L,XLI(L),YL(L)
1968          50 CONTINUE
1969 C
1970          WRITE (6,130)
1971          IF (IUI.EQ.1) WRITE (6,200)
1972          IF (IUI.EQ.2) WRITE (6,210)
1973          IPP=1
1974          DO 60 L=LDFSS,LDFSF
1975              CALL MTLUP (XP(L),YU(L),IINTDFS,NUPTS,NUPTS,1,IPP,XUI,YUI)
1976          60 CONTINUE
1977              LDUM=NUPTS
1978              IF (LDFSF.GT.NUPTS) LDUM=LDFSF
1979              LDF=0
1980              DO 70 L=LDFSS,LDFSF
1981                  LDF=LDF+1
1982                  XWI(LDF)=XP(L)
1983                  YWI(LDF)=YU(L)
1984          70 CONTINUE
1985              LMDF=LDFSF-LDFSS+1
1986              LDF=0
1987              DO 90 L=1,LDUM
1988                  LDFS=0
1989                  IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
1990                  IF (LDFS.EQ.0) GO TO 80
1991                  LDF=LDF+1
1992          90 CONTINUE
1992              SLOPE=DIF(LDF,IDIFDFS,LMDF,XWI,YWI)

```

```

1993      NXNYU(L)=-SLOPE
1994      IF (YU(L).GE.O.O.OR.NGIM.EQ.O) GO TO 80
1995      YU(L)=O.O
1996      NXNYU(L)=O.O
1997      SLOPE=-NXNYU(L)
1998      80 IF (L.LE.NUPTS.AND.LDFS.EQ.1) WRITE (6.220) L,XUI(L),YUI(L),XP(L)
1999      1 ,YU(L),SLOPE
2000      IF (L.GT.NUPTS.AND.LDFS.EQ.1) WRITE (6.230) L,XP(L),YU(L),SLOPE
2001      IF (L.LE.NUPTS.AND.LDFS.EQ.O) WRITE (6.240) L,XUI(L),YUI(L)
2002      90 CONTINUE
2003      RETURN
2004 C
2005 C      INPUT DUAL FLOW SPACE BOUNDARY RADIUS AND SLOPE
2006 C
2007      100 WRITE (6.120)
2008      WRITE (6.140)
2009      IF (IUI.EQ.1) WRITE (6.150)
2010      IF (IUI.EQ.2) WRITE (6.160)
2011      DO 110 L=LDFSS,LDFSF
2012      SLOPEL=-NXNYL(L)
2013      SLOPEU=-NXNYU(L)
2014      WRITE (6.170) L,XP(L),YL(L),SLOPEL,YU(L),SLOPEU
2015      110 CONTINUE
2016      RETURN
2017 C
2018 C      FORMAT STATEMENTS
2019 C
2020      120 FORMAT (1H1)
2021      130 FORMAT (1HO)
2022      140 FORMAT (1HO,10X,35HDUAL FLOW SPACE BOUNDARY GEOMETRY -)
2023      150 FORMAT (1HO,20X,41HGENERAL BOUNDARIES HAVE BEEN SPECIFIED BY,26H T
2024      1HE FOLLOWING PARAMETERS,./22X,1HL,11X,6HXP(IN),11X,6HYL(IN),11X,6
2025      2 HSLOPEL,11X,6HYU(IN),11X,6HSLOPEU,/)
2026      160 FORMAT (1HO,20X,41HGENERAL BOUNDARIES HAVE BEEN SPECIFIED BY,26H T
2027      1HE FOLLOWING PARAMETERS,./22X,1HL,11X,6HXP(CM),11X,6HYL(CM),11X,6
2028      2 HSLOPEL,11X,6HYU(CM),11X,6HSLOPEU,/)
2029      170 FORMAT (1H ,20X,12,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
2030      180 FORMAT (1HO,20X,46HGENERAL BOUNDARIES HAVE BEEN SPECIFIED BY THE
2031      1 30HFOLLOWING PARAMETERS, IINTDFS=.11,10H, IDIFDFS=.11,1H,./22X,1
2032      2 HL,10X,7HXL(IN),10X,7HYL(IN),11X,6HXP(IN),11X,6HYL(IN),11X,6HSL
2033      3OPEL,/)
2034      190 FORMAT (1HO,20X,46HGENERAL BOUNDARIES HAVE BEEN SPECIFIED BY THE
2035      1 30HFOLLOWING PARAMETERS, IINTDFS=.11,10H, IDIFDFS=.11,1H,./22X,1
2036      2 HL,10X,7HXL(CM),10X,7HYL(CM),11X,6HXP(CM),11X,6HYL(CM),11X,6HSL
2037      3OPEL,/)
2038      200 FORMAT (1HO,21X,1HL,10X,7HXUI(IN),10X,7HYUI(IN),11X,6HXP(IN),11X,6
2039      1 HYU(IN),11X,6HSLOPEU,/)
2040      210 FORMAT (1HO,21X,1HL,10X,7HXUI(CM),10X,7HYUI(CM),11X,6HXP(CM),11X,6
2041      1 HYU(CM),11X,6HSLOPEU,/)
2042      220 FORMAT (1H ,20X,12,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
2043      230 FORMAT (1H ,20X,12,41X,F10.4,7X,F10.4,7X,F10.4)
2044      240 FORMAT (1H ,20X,12,7X,F10.4,7X,F10.4)
2045      END

```

```

2046      SUBROUTINE MTLUP (X,Y,M,N,MAX,NTAB,I,VARI,VARD)
2047 C
2048 C      .....
2049 C
2050 C      THIS SUBROUTINE IS CALLED BY SUBROUTINES GEOM, GEOMCB, AND GEOMLU
2051 C      TO INTERPOLATE FOR WALL COORDINATES FOR THE TABULAR INPUT CASE.
2052 C      SUBROUTINE MTLUP WAS TAKEN FROM THE NASA-LANGLEY PROGRAM
2053 C      LIBRARY. THE DATE OF THIS VERSION IS 09-12-69.
2054 C
2055 C      .....
2056 C
2057 C      MODIFICATION OF LIBRARY INTERPOLATION SUBROUTINE FTLUP
2058 C      MULTIPLE TABLE LOOK-UP ON ONE INDEPENDENT VARIABLE TABLE
2059 C      USES AN EXTERNAL INTERVAL POINTER (I) TO START SEARCH
2060 C      I LESS THAN 0 WILL CHECK MONOTONICITY
2061 C
2062      DIMENSION VARI(1), VARD(MAX,1), Y(1), V(3), YY(2)
2063      LOGICAL EX
2064 C
2065      IF (M.EQ.0) GO TO 170
2066      IF (N.LE.1) GO TO 170
2067      EX=.FALSE.
2068      IF (I.GE.0) GO TO 60
2069      IF (N.LT.2) GO TO 60
2070 C
2071 C      MONOTONICITY CHECK
2072 C
2073      IF (VARI(2)-VARI(1)) 20,20,40
2074 C
2075 C      ERROR IN MONOTONICITY
2076 C
2077      10 K=LOC(F(VARI(1)))
2078      WRITE (6,190) J,K,(VARI(J),J=1,N)
2079      CALL EXIT
2080 C
2081 C      MONOTONIC DECREASING
2082 C
2083      20 DO 30 J=2,N
2084      IF (VARI(J)-VARI(J-1)) 30,10,10
2085      30 CONTINUE
2086      GO TO 60
2087 C
2088 C      MONOTONIC INCREASING
2089 C
2090      40 DO 50 J=2,N
2091      IF (VARI(J)-VARI(J-1)) 10,10,50
2092      50 CONTINUE
2093 C
2094 C      INTERPOLATION
2095 C
2096      60 IF (I.LE.0) I=1
2097      IF (I.GE.N) I=N-1
2098 C
2099 C      LOCATE I INTERVAL (X(I).LE.X.LT.X(I+1))
2100 C
2101      IF ((VARI(I)-X)*(VARI(I+1)-X)) 100,100,70
2102 C
2103 C      IN GIVES DIRECTION FOR SEARCH OF INTERVALS
2104 C
2105      70 IN=SIGN(1.0,(VARI(I+1)-VARI(I))*(X-VARI(I)))
2106 C
2107 C      IF X OUTSIDE ENDPOINTS, EXTRAPOLATE FROM END INTERVAL
2108 C
2109      80 IF ((I+IN).LE.0) GO TO 90
2110      IF ((I+IN).GE.N) GO TO 90
2111      I=I+IN
2112      IF ((VARI(I)-X)*(VARI(I+1)-X)) 100,100,80
2113 C
2114 C      EXTRAPOLATION
2115 C
2116      90 EX=.TRUE.
2117      100 IF (M.EQ.2) GO TO 120

```

```

2118 C
2119 C   FIRST ORDER
2120 C
2121   DO 110 NT=1,NTAB
2122 110 Y(NT)=(VARD(1,NT)*(VARI(I+1)-X)-VARD(I+1,NT)*(VARI(I)-X))/(VARI(I+
2123 1 1)-VARI(I))
2124   IF (EX) I=I+IN
2125   RETURN
2126 C
2127 C   SECOND ORDER
2128 C
2129 120 IF (N.EQ.2) GO TO 10
2130   IF (I.EQ.(N-1)) GO TO 140
2131   IF (I.EQ.1) GO TO 130
2132 C
2133 C   PICK THIRD POINT
2134 C
2135   SK=VARI(I+1)-VARI(I)
2136   IF ((SK*(X-VARI(I-1)))..I.T.(SK*(VARI(I+2)-X))) GO TO 140
2137 130 L=I
2138   GO TO 150
2139 140 L=I-1
2140 150 V(1)=VARI(L)-X
2141   V(2)=VARI(L+1)-X
2142   V(3)=VARI(L+2)-X
2143   DO 160 NT=1,NTAB
2144   YY(1)=(VARD(L,NT)*V(2)-VARD(L+1,NT)*V(1))/(VARI(L+1)-VARI(L))
2145   YY(2)=(VARD(L+1,NT)*V(3)-VARD(L+2,NT)*V(2))/(VARI(L+2)-VARI(L+1))
2146 160 Y(NT)=(YY(1)*V(3)-YY(2)*V(1))/(VARI(L+2)-VARI(L))
2147   IF (EX) I=I+IN
2148   RETURN
2149 C
2150 C   ZERO ORDER
2151 C
2152 170 DO 180 NT=1,NTAB
2153 180 Y(NT)=VARD(1,NT)
2154   RETURN
2155 C
2156 C   FORMAT STATEMENTS
2157 C
2158 190 FORMAT (1H1,49H TABLE BELOW OUT OF ORDER FOR MTLUP AT POSITION
2159 1 ,15,/31H X TABLE IS STORED IN LOCATION ,06,/(8G15.8))
2160   END

```



```

2217      SUBROUTINE ONEDIM
2218 C
2219 C .....
2220 C
2221 C      THIS SUBROUTINE CALCULATES THE 1-D INITIAL-DATA SURFACE
2222 C
2223 C .....
2224 C
2225 *CALL,MCC
2226      IF (PT(1).NE.O.O.AND.TT(1).NE.O.O) GO TO 10
2227      IERR=1
2228      WRITE (6,200)
2229      RETURN
2230      10 MN3=0.01
2231      IF (N1D.EQ.-1.OR.N1D.GT.2) MN3=2.0
2232      NXCK=0
2233      ACDEF=2.O/(GAMMA+1.O)
2234      BCDEF=(GAMMA-1.O)/(GAMMA+1.O)
2235      CCDEF=(GAMMA+1.O)/2.O/(GAMMA-1.O)
2236      IF (N1D.LT.O) GO TO 30
2237 C
2238 C      OVERALL LOOP
2239 C
2240      IF (NGCB.NE.O.OR.MDFS.NE.O) GO TO 20
2241      RSTAR=RT
2242      RSTARS=RT*RT
2243      GO TO 30
2244      20 RSTAR=YW(LT)-YU(LT)+YL(LT)-YCB(LT)
2245      RSTARS=YW(LT)**2-YU(LT)**2+YL(LT)**2-YCB(LT)**2
2246      30 DO 180 L=1,LMAX
2247      IF (L.EQ.1.AND.ISUPER.EQ.1) GO TO 180
2248      IF (N1D.LT.O) GO TO 60
2249      IF (NGCB.NE.O) GO TO 40
2250      IF (MDFS.NE.O) GO TO 40
2251      IF (XP(L).LT.XT) GO TO 60
2252      IF (XP(L).GT.XT) GO TO 50
2253      MN3=1.0
2254      GO TO 110
2255      40 IF (L.LT.L1) GO TO 60
2256      IF (L.GT.LT) GO TO 50
2257      MN3=1.0
2258      GO TO 110
2259      50 IF (NXCK.EQ.1) GO TO 60
2260      IF (N1D.EQ.1.OR.N1D.EQ.3) MN3=1.1
2261      IF (N1D.EQ.2.OR.N1D.EQ.4) MN3=0.9
2262      NXCK=1
2263      60 IF (NDIM.EQ.1) GO TO 70
2264      RAD=YW(L)-YU(L)+YL(L)-YCB(L)
2265      ARATIO=RAD/RSTAR
2266      GO TO 80
2267      70 RADS=YW(L)**2-YU(L)**2+YL(L)**2-YCB(L)**2
2268      ARATIO=RADS/RSTARS
2269 C
2270 C      NEWTON-RAPHSON ITERATION LOOP
2271 C
2272      80 DO 100 ITER=1,100
2273      ABM=ACDEF*BCDEF*MN3*MN3
2274      ABMC=ABM**CCDEF
2275      FM=ABMC/MN3-ARATIO
2276      FPM=ABMC*(2.O*BCDEF*CCDEF/ABM-1.O/(MN3*MN3))
2277      OMN3=MN3
2278      MN3=OMN3-FM/FPM
2279      IF (OMN3.GT.0.99.AND.OMN3.LT.1.01) MN3=0.5*(OMN3+MN3)
2280      IF (MN3.GT.1.0.AND.OMN3.LT.1.0) MN3=0.99
2281      IF (MN3.LT.1.0.AND.OMN3.GT.1.0) MN3=1.01
2282      IF (N1D.EQ.-1.AND.MN3.LE.1.0) MN3=1.01
2283      IF (N1D.EQ.-2.AND.MN3.GE.1.0) MN3=0.99
2284      IF (MN3.GT.50.0) MN3=50.0
2285      IF (MN3.GE.0.0) GO TO 90
2286      MN3=-MN3
2287      GO TO 100
2288      90 IF (ABS(MN3-OMN3)/OMN3.LE.0.0005) GO TO 110

```



```

2289 100 CONTINUE
2290 WRITE (6,190) L
2291 C
2292 C FILL IN 2-D ARRAYS LGOP
2293 C
2294 110 LDFS=0
2295 IF (L.GE.LDFSS.AND.L.LE.LDFS) LDFS=1
2296 DEM=1.0+GAM2*MN3*MN3
2297 DEMP=DEM+GAM1
2298 DNXNY=(NXNY(L)-NXNYCB(L))/FLOAT(M1)
2299 IF (MDFS.EQ.0.OR.LDFS.EQ.0) GO TO 120
2300 DNXNY1=(NXNY(L)-NXNYCB(L))/FLOAT(MDFS-1)
2301 DNXNY2=(NXNY(L)-NXNYU(L))/FLOAT(MMAX-MDFS)
2302 120 DO 170 M=1,MMAX
2303 IF (MDFS.EQ.0.OR.LDFS.EQ.0) GO TO 150
2304 IF (L.NE.1) GO TO 130
2305 IF (ISUPER.EQ.2.AND.M.LT.MDFS) GO TO 170
2306 IF (ISUPER.EQ.3.AND.M.GT.MDFS) GO TO 170
2307 IF (ISUPER.EQ.2.AND.M.EQ.MDFS) GO TO 150
2308 130 IF (M.LT.MDFS) DNXNY=DNXNY1
2309 IF (M.GT.MDFS) DNXNY=DNXNY2
2310 IF (M.NE.MDFS) GO TO 150
2311 PL(L,1)=PTL/DEMP
2312 TEMP=TTL/DEM
2313 ROL(L,1)=PL(L,1)/(RG*TEMP)
2314 JQ=MN3*SQRT(GAMMA*PL(L,1)/ROL(L,1))
2315 IF (NXNY(L).EQ.0.0) GO TO 140
2316 UL(L,1)=JQ/SQRT(1.0+NXNY(L)*NXNY(L))
2317 VL(L,1)=-UL(L,1)*NXNY(L)
2318 GO TO 150
2319 140 UL(L,1)=JQ
2320 VL(L,1)=0.0
2321 150 IF (ISUPER.EQ.3.AND.(M.EQ.MDFS.AND.L.EQ.1)) GO TO 170
2322 P(L,M,1)=PT(M)/DEMP
2323 TEMP=TT(M)/DEM
2324 RO(L,M,1)=P(L,M,1)/(RG*TEMP)
2325 QO=MN3*SQRT(GAMMA*P(L,M,1)/RO(L,M,1))
2326 DN=NXNYCB(L)+DNXNY*FLOAT(M-1)
2327 IF (LDFS.NE.0.AND.M.GE.MDFS) DN=NXNYU(L)+DNXNY*FLOAT(M-MDFS)
2328 DNS=DN*DN
2329 IF (DNS.EQ.0.0) GO TO 160
2330 SIGN=1.0
2331 IF (DN.GT.0.0) SIGN=-1.0
2332 U(L,M,1)=QO/SQRT(1.0+DNS)
2333 V(L,M,1)=SIGN*QO/SQRT(1.0+1.0/DNS)
2334 GO TO 170
2335 160 U(L,M,1)=QO
2336 V(L,M,1)=0.0
2337 170 CONTINUE
2338 180 CONTINUE
2339 RETURN
2340 C
2341 C FORMAT STATEMENTS
2342 C
2343 190 FORMAT (1H0,10X,47H***** THE 1-D SOLUTION FOR THE INITIAL-DATA SUR
2344 1 .47HFACE FAILED TO CONVERGE IN 100 ITERATIONS AT L=,12.6H *****)
2345 200 FORMAT (1H0,10X,48H***** THE STAGNATION CONDITIONS FOR THE 1-D INI
2346 1T,41HIAL-DATA SURFACE WERE NOT SPECIFIED *****)
2347 END

```

```

2348      SUBROUTINE MAP
2349 C
2350 C      .....
2351 C
2352 C      THIS SUBROUTINE CALCULATES THE MAPPING FUNCTIONS
2353 C
2354 C      .....
2355 C
2356 *CALL,MCC
2357 C
2358 C      SINGLE FLOW SPACE
2359 C
2360      IF (IP.EQ.-1) GO TO 40
2361      IF (LMAP.GE.LDFS.S.AND.LMAP.LL.LDFS.F) GO TO 10
2362      YP=YCB(LMAP)+VN(MMAP)*(YW(LMAP)-YCB(LMAP))
2363      IF (IP.EQ.0) RETURN
2364      OM1=DZDX(LMAP)
2365      OM2=DZDX(LMAP+1)
2366      BE=1.0/(YW(LMAP)-YCB(LMAP))
2367      BE3=DYDVN(MMAP)*BE
2368      BE4=DYDVN(MMAP+1)*BE
2369      AL=NXNYCB(LMAP)+VN(MMAP)*(NXNY(LMAP)-NXNYCB(LMAP))
2370      AL3=BE3*AL
2371      AL4=BE4*AL
2372      DE=-VN(MMAP)*XWI(LMAP)
2373      DE3=BE3*DE
2374      DE4=BE4*DE
2375      RETURN
2376 C
2377 C      DUAL FLOW SPACE
2378 C
2379      10 IF (MMAP.LT.MDFS) GO TO 20
2380      IF (MMAP.GT.MDFS) GO TO 30
2381      IF (IB.EQ.4) GO TO 30
2382 C
2383      20 YP=YCB(LMAP)+VN(MMAP)*(YL(LMAP)-YCB(LMAP))/CC
2384      IF (IP.EQ.0) RETURN
2385      OM1=DZDX(LMAP)
2386      OM2=DZDX(LMAP+1)
2387      BE=CC/(YL(LMAP)-YCB(LMAP))
2388      BE3=DYDVN(MMAP)*BE
2389      BE4=DYDVN(MMAP+1)*BE
2390      AL=(VN(MMAP)*NXNYL(LMAP)-(VN(MMAP)-CC)*NXNYCB(LMAP))/CC
2391      AL3=BE3*AL
2392      AL4=BE4*AL
2393      DE3=0.0
2394      DE4=0.0
2395      IF (MMAP.NE.MDFS) RETURN
2396      AL4=AL3
2397      BE4=BE3
2398      RETURN
2399 C
2400      30 YP=YU(LMAP)+(VN(MMAP)-CC)*(YW(LMAP)-YU(LMAP))/(1.0-CC)
2401      IF (IP.EQ.0) RETURN
2402      OM1=DZDX(LMAP)
2403      OM2=DZDX(LMAP+1)
2404      BE=(1.0-CC)/(YW(LMAP)-YU(LMAP))
2405      BE3=DYDVN(MMAP)*BE
2406      BE4=DYDVN(MMAP+1)*BE
2407      AL=((VN(MMAP)-CC)*NXNY(LMAP)-(VN(MMAP)-1.0)*NXNYU(LMAP))/(1.0-CC)
2408      AL3=BE3*AL
2409      AL4=BE4*AL

```

```

2410      DE=(VN(MMAP)-CC)*XWI(LMAP)/(1.0-CC)
2411      DE3=BE3+DE
2412      DE4=BE4+DE
2413      IF (MMAP.NE.MDFS) RETURN
2414      AL3=AL4
2415      BE3=BE4
2416      DE3=DE4
2417      RETURN
2418 C
2419 C      CALCULATE THE MAPPING FUNCTIONS FOR THE INITIAL SET-UP
2420 C
2421      40 DO 50 L=1,LMAX
2422          X(L)=XP(1)+FLOAT(L-1)*DX
2423      50 CONTINUE
2424          DO 60 L=1,L1
2425              DZDX(L+1)=(X(L+1)-X(L))/(XP(L+1)-XP(L))
2426      60 CONTINUE
2427          DZDX(1)=DZDX(2)
2428          DZDX(LMAX+1)=DZDX(LMAX)
2429          IF (MDFS.EQ.0) GO TO 70
2430          LVN=LDFSS
2431          IF (LDFSS.EQ.1.AND.LDFSF.NE.LMAX) LVN=LDFSF
2432          CC=(YL(LVN)-YCB(LVN))/(YW(LVN)+YL(LVN)-YCB(LVN))
2433          IF (LDFSS.EQ.1.OR.LDFSF.EQ.LMAX) GO TO 70
2434          CCD=(YL(LDFSF)-YCB(LDFSF))/(YW(LDFSF)+YL(LDFSF)-YCB
2435              1 (LDFSF))
2436          IF (ABS(CCD-CC)/CC.LE.0.01) GO TO 70
2437          WRITE (6,140)
2438          IERR=1
2439          RETURN
2440      70 DO 80 M=1,MMAX
2441          Y(M)=FLOAT(M-1)*DY
2442      80 CONTINUE
2443          IF (IST.NE.0) GO TO 100
2444          DO 90 M=1,MMAX
2445              VN(M)=Y(M)
2446              DYDVN(M)=1.0
2447              YI(M)=Y(M)
2448      90 CONTINUE
2449              DYDVN(MMAX+1)=1.0
2450              RETURN
2451      100 DO 120 M=1,MMAX
2452          VN(M)=(YI(M)-YCB(1))/(YW(1)-YCB(1))
2453          IF (MDFS.EQ.0.OR.LDFSS.NE.1) GO TO 120
2454          IF (M.GE.MDFS) GO TO 110
2455          VN(M)=CC*(YI(M)-YCB(1))/(YL(1)-YCB(1))
2456          GO TO 120
2457      110 VN(M)=CC+(1.0-CC)*(YI(M)-YU(1))/(YW(1)-YU(1))
2458      120 CONTINUE
2459          DO 130 M=1,M1
2460              DYDVN(M+1)=(Y(M+1)-Y(M))/(VN(M+1)-VN(M))
2461      130 CONTINUE
2462              DYDVN(1)=DYDVN(2)
2463              DYDVN(MMAX+1)=DYDVN(MMAX)
2464              RETURN
2465 C
2466      140 FORMAT (1H0,100H***** DUAL FLOW SPACE WALLS DO NOT BEGIN AND END A
2467          IT APPROXIMATELY THE SAME PROPORTIONAL HEIGHT *****)
2468          END

```

```

2469      SUBROUTINE MASFLO
2470 C
2471 C      .....
2472 C
2473 C      THIS SUBROUTINE CALCULATES THE INITIAL-DATA OR SOLUTION SURFACE
2474 C      MASS FLOW AND MOMENTUM THRUST
2475 C
2476 C      .....
2477 C
2478 *CALL MCC
2479      LC2=LC*LC
2480 C
2481 C      CALCULATE AND PRINT THE MASS FLOW AT EACH L LOCATION
2482 C
2483      IP=0
2484      ND=N3
2485      IF (N.EQ.0) ND=1
2486      NP=N+NSTART
2487      IF (IUD.NE.2) WRITE (6,80) NP
2488      IF (IUD.EQ.2) WRITE (6,90) NP
2489      DO 70 L=1,LMAX
2490      LMAP=L
2491      XMASS=0.0
2492      THRUST=0.0
2493      IF (MDFS.NE.0) IB=3
2494      LDFS=0
2495      IF (L.GE.LDFSS.AND.L.LE.LDFS) LDFS=1
2496      DO 50 M=2,MMAX
2497      MMAP=M
2498      CALL MAP
2499      MMAP=M-1
2500      YP1=YP
2501      CALL MAP
2502      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 10
2503      ROU=(RO(L,ND)*UL(L,ND)+RO(L,M-1,ND)*U(L,M-1,ND))*0.5
2504      ROU2=(RO(L,ND)*UL(L,ND)**2+RO(L,M-1,ND)*U(L,M-1,ND)**2)*0.5
2505      IB=4
2506      GO TO 20
2507      10 ROU=(RO(L,M,ND)*U(L,M,ND)+RO(L,M-1,ND)*U(L,M-1,ND))*0.5
2508      ROU2=(RO(L,M,ND)*U(L,M,ND)**2+RO(L,M-1,ND)*U(L,M-1,ND)**2)*0.5
2509      20 IF (NDIM.EQ.1) GO TO 30
2510      AREA=(YP1-YP)/LC2
2511      GO TO 40
2512      30 AREA=3.141593*(YP1**2-YP**2)/LC2
2513      40 XMASS=XMASS+ROU*AREA*G
2514      THRUST=THRUST+ROU2*AREA
2515      50 CONTINUE
2516      IF (L.EQ.1) XMASS1=XMASS
2517      XMFR=0.0
2518      IF (XMASS1.NE.0.0) XMFR=XMASS/XMASS1
2519      IF (L.EQ.1) THRUST1=THRUST
2520      TR=0.0
2521      IF (THRUST1.NE.0.0) TR=THRUST/THRUST1
2522      IF (IUD.NE.2) GO TO 60
2523      XMASS=XMASS*0.4536
2524      THRUST=THRUST*4.4477
2525      IF (NDIM.NE.0) GO TO 60
2526      XMASS=XMASS/2.54
2527      THRUST=THRUST/2.54
2528      60 WRITE (6,100) L,XMASS,XMFR,THRUST,TR
2529      70 CONTINUE
2530      RETURN
2531 C
2532 C      FORMAT STATEMENTS
2533 C
2534      80 FORMAT (1H1,20X,36HMASS FLOW AND THRUST CALCULATION, N=.16, //30X,1
2535      1 HL,7X,9HMF (LEM/S),8X,6HMF/MF1,8X,6HT (LEF),11X,4HT/TI,/)
2536      90 FORMAT (1H1,20X,36HMASS FLOW AND THRUST CALCULATION, N=.16, //30X,1
2537      1 HL,8X,8HMF (KG/S),8X,6HMF/MF1,10X,4HT (N),11X,4HT/TI,/)
2538      100 FORMAT (1H1,20X,11O,F16.5,F14.4,2F15.4)
2539      END

```

```

2540      SUBROUTINE PLOT (TITLE,T,NP,IVPTS)
2541 C
2542 C      .....
2543 C
2544 C      THIS SUBROUTINE PLOTS THE VELOCITY VECTORS AND DEPENDENT VARIABLE
2545 C      CONTOUR PLOTS
2546 C
2547 C      .....
2548 C
2549      DIMENSION CON(9), XCO(4), YCO(4), TITLE(10)
2550 *CALL MCC
2551 C
2552 C      SET UP THE PLOT SIZE
2553 C
2554      IP=0
2555      ND=N3
2556      IF (N.EQ.0) ND=1
2557      XXL=XI
2558      XR=XE
2559      YT=YW(1)
2560      YB=YCB(1)
2561      DO 10 L=2,LMAX
2562      YT=AMAX1(YT,YW(L))
2563      YB=AMIN1(YB,YCB(L))
2564 10 CONTINUE
2565      VV=-0.1*DX
2566      DO 70 IDUM=1,IVPTS
2567      VV=VV+DX
2568      FIYB=900.0
2569      XD=(XR-XXL)/(YT-YB)
2570      FIR=(1022.0-1022.0/FLOAT(L1)-FLOAT(IDUM)*1022.0/FLOAT(L1))/884.0
2571      IF (XD.LE.FIR) GO TO 20
2572      FIXL=1022.0/FLOAT(L1)
2573      FIXR=1022.0-FIXL-FLOAT(IDUM)*1022.0/FLOAT(L1)
2574      FIYT=900.0-(FIXR-FIXL)/XD
2575      GO TO 30
2576 20 FIXL=511.0-450.0*XD
2577      FIXR=511.0+450.0*XD
2578      FIYT=16.0
2579 30 XCONV=(FIXR-FIXL)/(XR-XXL)
2580      YCONV=(FIYT-FIYB)/(YT-YB)
2581 C
2582 C      GENERATE THE VELOCITY VECTOR PLOT
2583 C
2584      VMAX=0.0
2585      DO 40 L=1,LMAX
2586      DO 40 M=1,MMAX
2587      VMAX=AMAX1(VMAX,ABS(U(L,M,ND)),ABS(V(L,M,ND)))
2588 40 CONTINUE
2589      IF (VMAX.LT.1.0E-10) GO TO 80
2590      DROU=VV/VMAX
2591      CALL ADV (1)
2592      DO 60 L=1,LMAX
2593      LMAP=L
2594      IF (MDFS.NE.0) IB=3
2595      LDFS=0
2596      IF (L.GE.LDFSS.AND.L.LE.LDFS) LDFS=1
2597      IX1=FIXL+(XP(L)-XI)*XCONV
2598      DO 60 M=1,MMAX
2599      MMAP=M
2600      CALL MAP
2601      IF (M.NE.MDFS.OR.LDFS.EQ.0) GO TO 50
2602      IY1=FIYB+(YP-YB)*YCONV
2603      IX2=IX1+UL(L,ND)*DROU*XCONV
2604      IY2=IY1+VL(L,ND)*DROU*YCONV
2605      CALL DRV (IX1,IY1,IX2,IY2)
2606      CALL PLT (IX1,IY1,16)
2607      IR=4
2608      CALL MAP
2609 50 IY1=FIYB+(YP-YB)*YCONV
2610      IX2=IX1+U(L,M,ND)*DROU*XCONV
2611      IY2=IY1+V(L,M,ND)*DROU*YCONV

```

```

2612      CALL DRV (IX1,IY1,IX2,IY2)
2613      CALL PLT (IX1,IY1,16)
2614      60 CONTINUE
2615      CALL LINCNT (58)
2616      WRITE (7,580) IDUM,NP,T
2617      WRITE (7,500) TITLE
2618      70 CONTINUE
2619 C
2620 C      RESET PLOT SIZE FOR CONTOUR PLOTS
2621 C
2622      80 IF (XD.LE.FIR) GO TO 90
2623      FIXR=1022.0-FIXL-1022.0/FLCAT(L1)
2624      FIYT=500.0-(FIXR-FIXL)/XD
2625      XCONV=(FIXR-FIXL)/(XR-XXL)
2626      YCONV=(FIYT-FIYB)/(YT-YB)
2627 C
2628 C      GENERATE THE PHYSICAL SPACE GRID
2629 C
2630      90 CALL ADV (1)
2631      DO 110 L=2,LMAX
2632      IF (MDFS.NE.O) IB=3
2633      IX1=FIXL+(XP(L-1)-XI)*XCONV
2634      IX2=FIXL+(XP(L)-XI)*XCONV
2635      LDFS=0
2636      IF (L.GE.LDFS.S.AND.L.LE.LDFS.F) LDFS=1
2637      DO 110 M=1,MMAX
2638      LMAP=L-1
2639      MMAP=M
2640      CALL MAP
2641      LMAP=L
2642      YP1=YP
2643      CALL MAP
2644      IF (M.NE.MDFS.OR.LDFS.EQ.O) GO TO 100
2645      IY1=FIYB+(YP1-YB)*YCONV
2646      IY2=FIYB+(YP-YB)*YCONV
2647      CALL DRV (IX1,IY1,IX2,IY2)
2648      IB=4
2649      LMAP=L-1
2650      CALL MAP
2651      LMAP=L
2652      YP1=YP
2653      CALL MAP
2654      100 IY1=FIYB+(YP1-YB)*YCONV
2655      IY2=FIYB+(YP-YB)*YCONV
2656      CALL DRV (IX1,IY1,IX2,IY2)
2657      110 CONTINUE
2658 C
2659      DO 130 L=1,LMAX
2660      IX1=FIXL+(XP(L)-XI)*XCONV
2661      IY1=FIYB+(YCB(L)-YB)*YCONV
2662      IF (MDFS.EQ.O) GO TO 120
2663      IF (L.LT.LDFS.S.OR.L.GT.LDFS.F) GO TO 120
2664      IY2=FIYB+(YL(L)-YB)*YCONV
2665      CALL DRV (IX1,IY1,IX1,IY2)
2666      IY1=FIYB+(YU(L)-YB)*YCONV
2667      120 IY2=FIYB+(YW(L)-YB)*YCONV
2668      CALL DRV (IX1,IY1,IX1,IY2)
2669      130 CONTINUE
2670      CALL LINCNT (58)
2671      WRITE (7,590) NP,T
2672      WRITE (7,500) TITLE
2673 C
2674 C      FILL THE PLOTTING ARRAY CO FOR THE CONTOUR PLOTS
2675 C
2676      MDUM=MMAX
2677      IF (MDFS.NE.O) MDUM=MMAX+1
2678      IUC=1.0
2679      IF (IUD.EQ.2) IUC=0.0
2680      IDUM=4
2681      IF (ITM.EQ.2) IDUM=5
2682      IF (ITM.EQ.3) IDUM=6
2683      DO 490 I=1,IDUM

```

```

2684 C
2685 DO 270 L=1,LMAX
2686 LDFS=0
2687 IF (L.GE.LDFSS.AND.L.LE.LDFS) LDFS=1
2688 DO 270 M=1,MDUM
2689 IF (LDFS.EQ.O.AND.M.EQ.MMAX+1) GO TO 270
2690 MD1=M
2691 IF (LDFS.NE.O.AND.M.GT.MDFS) MD1=M-1
2692 IF (M.NE.MDFS.OR.LDFS.EQ.O) GO TO 200
2693 GO TO (140,150,160,170,180,190), I
2694 140 CQ(L,M)=ROL(L,ND)*G*(16.02-IUC*15.02)
2695 GO TO 270
2696 150 CU(L,M)=PL(L,ND)/PC*(6.8948-IUC*5.8948)
2697 GO TO 270
2698 160 CO(L,M)=PL(L,ND)/(ROL(L,ND)*RG)*(0.555556+IUC*0.444444)
2699 GO TO 270
2700 170 CQ(L,M)=SQRT((UL(L,ND)**2+VL(L,ND)**2)/(GAMMA*PL(L,ND)/ROL(L,ND)))
2701 GO TO 270
2702 180 CQ(L,M)=QL(L,ND)*(0.0929+IUC*0.9071)
2703 GO TO 270
2704 190 CQ(L,M)=EL(L,ND)*(0.0929+IUC*0.9071)
2705 GO TO 270
2706 200 GO TO (210,220,230,240,250,260), I
2707 210 CQ(L,M)=RO(L,MD1,ND)*G*(16.02-IUC*15.02)
2708 GO TO 270
2709 220 CQ(L,M)=P(L,MD1,ND)/PC*(6.8948-IUC*5.8948)
2710 GO TO 270
2711 230 CQ(L,M)=P(L,MD1,ND)/(RO(L,MD1,ND)*RG)*(0.555556+IUC*0.444444)
2712 GO TO 270
2713 240 CQ(L,M)=SQRT((U(L,MD1,ND)**2+V(L,MD1,ND)**2)/(GAMMA*P(L,MD1,ND)/RO
2714 L(L,MD1,ND)))
2715 GO TO 270
2716 250 CQ(L,M)=Q(L,MD1,ND)*(0.0929+IUC*0.9071)
2717 GO TO 270
2718 260 CQ(L,M)=E(L,MD1,ND)*(0.0929+IUC*0.9071)
2719 270 CONTINUE
2720 C
2721 C DETERMINE THE PLOTTING LINE QUANTITIES AND LABEL THE FRAMES
2722 C
2723 QMN=1.0E06
2724 QMX=-QMN
2725 DO 280 L=1,LMAX
2726 LDFS=0
2727 IF (L.GE.LDFSS.AND.L.LE.LDFS) LDFS=1
2728 DO 280 M=1,MDUM
2729 IF (LDFS.EQ.O.AND.M.EQ.MMAX+1) GO TO 280
2730 QMN=AMIN1(CQ(L,M),QMN)
2731 QMX=AMAX1(CQ(L,M),QMX)
2732 280 CONTINUE
2733 XX=QMX-QMN
2734 DJ=0.1*XX
2735 DO 290 K=1,9
2736 CON(K)=QMN+(FLOAT(K))*DJ
2737 290 CONTINUE
2738 K=9
2739 CALL ADV (1)
2740 CALL LINCNT (58)
2741 GO TO (300,310,320,330,340,350), I
2742 300 WRITE (7.510) NP,T
2743 GO TO 360
2744 310 WRITE (7.520) NP,T
2745 GO TO 360
2746 320 WRITE (7.530) NP,T
2747 GO TO 360
2748 330 WRITE (7.540) NP,T
2749 GO TO 360
2750 340 WRITE (7.550) NP,T
2751 GO TO 360
2752 350 WRITE (7.560) NP,T
2753 360 WRITE (7.570) QMN,QMX,CON(1),CON(K),DJ
2754 WRITE (7.500) TITLE

```

```

2755 C
2756 C   DETERMINE THE LOCATION OF EACH CONTOUR LINE SEGMENT AND PLOT IT
2757 C
2758   DO 470 L=2,LMAX
2759   IF (MOFS.NE.O) IB=3
2760   XCO(1)=XP(L-1)
2761   XCO(2)=XP(L)
2762   XCO(3)=XCO(1)
2763   XCO(4)=XCO(2)
2764   LDFS=0
2765   IF (L.GE.LDFS.S.AND.L.LE.LDFS.F) LDFS=1
2766   DO 470 M=2,MMAX
2767   MD2=M
2768   MD3=M
2769   IF (MOFS.EQ.O.OR.M.LE.MOFS) GO TO 370
2770   IF (L.GE.LDFS.S.AND.L.LE.LDFS.F) MD2=M+1
2771   IF (L.GE.LDFS.S+1.AND.L.LE.LDFS.F+1) MD3=M+1
2772 370 LMAP=L-1
2773   MMAP=M-1
2774   CALL MAP
2775   LMAP=L
2776   YCO(1)=YP
2777   CALL MAP
2778   LMAP=L-1
2779   MMAP=M
2780   YCO(2)=YP
2781   CALL MAP
2782   LMAP=L
2783   YCO(3)=YP
2784   CALL MAP
2785   YCO(4)=YP
2786   IF (M.NE.MOFS.OR.LDFS.EQ.O) GO TO 380
2787   IB=4
2788 380 DO 460 KK=1,K
2789   K1=0
2790   K2=0
2791   K3=0
2792   K4=0
2793   IF (CO(L-1,MD3-1).LE.CON(KK)) K1=1
2794   IF (CO(L,MD2-1).LE.CON(KK)) K2=1
2795   IF (CO(L-1,MD3).LE.CON(KK)) K3=1
2796   IF (CO(L,MD2).LE.CON(KK)) K4=1
2797   IF (K1+K2+K3+K4.NE.O) GO TO 460
2798   IF (K1+K2+K3+K4.EQ.O) GO TO 460
2799   LL=0
2800   IF (K1+K3.NE.1) GO TO 390
2801   IC1=1
2802   IC2=3
2803   LP1=L-1
2804   MP1=MD3-1
2805   LP2=L-1
2806   MP2=MD3
2807   ASSIGN 390 TO KR1
2808   GO TO 420
2809 390 IF (K1+K2.NE.1) GO TO 400
2810   IC1=1
2811   IC2=2
2812   LP1=L-1
2813   MP1=MD3-1
2814   LP2=L
2815   MP2=MD2-1
2816   ASSIGN 400 TO KR1
2817   GO TO 420
2818 400 IF (K2+K4.NE.1) GO TO 410
2819   IC1=2
2820   IC2=4
2821   LP1=L
2822   MP1=MD2-1
2823   LP2=L
2824   MP2=MD2
2825   ASSIGN 410 TO KR1
2826   GO TO 420

```



```

2827 410 IF (K3+K4.NE.1) GO TO 460
2828     IC1=3
2829     IC2=4
2830     LP1=L-1
2831     MP1=MD3
2832     LP2=L
2833     MP2=MD2
2834     ASSIGN 460 TO KR1
2835 420 LL=LL+1
2836     X=(CON(KK)-CO(LP1,MP1))/(CO(LP2,MP2)-CO(LP1,MP1))
2837     IF (LL.EQ.2) GO TO 430
2838     IX1=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XXL)*XCONV
2839     IY1=FIYB+(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YB)*YCONV
2840     GO TO KR1. (390,400,410,460)
2841 430 IX2=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XXL)*YCONV
2842     IY2=FIYB+(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YB)*YCONV
2843     CALL DRV (IX1,IY1,IX2,IY2)
2844     IF (KK.NE.1) GO TO 440
2845     CALL PLT (IX1,IY1,35)
2846 440 IF (KK.NE.K) GO TO 450
2847     CALL PLT (IX1,IY1,24)
2848 450 LL=0
2849     IF (LP2.NE.L) GO TO 460
2850     IF (MP2.NE.MD2-1) GO TO 460
2851     GO TO 400
2852 460 CONTINUE
2853 470 CONTINUE
2854 C
2855 C     DRAW THE GEOMETRY BOUNDARIES FOR THE CONTOUR PLOTS
2856 C
2857     DO 480 L=2,LMAX
2858     IX1=FIXL+(XP(L-1)-X1)*XCONV
2859     IX2=FIXL+(XP(L)-X1)*XCONV
2860     IY1=FIYB+(YCB(L-1)-YB)*YCONV
2861     IY2=FIYB+(YCB(L)-YB)*YCONV
2862     IY3=FIYB+(YW(L-1)-YB)*YCONV
2863     IY4=FIYB+(YW(L)-YB)*YCONV
2864     IY5=FIYB+(YL(L-1)-YB)*YCONV
2865     IY6=FIYB+(YL(L)-YB)*YCONV
2866     IY7=FIYB+(YU(L-1)-YB)*YCONV
2867     IY8=FIYB+(YU(L)-YB)*YCONV
2868     CALL DRV (IX1,IY1,IX2,IY2)
2869     CALL DRV (IX1,IY3,IX2,IY4)
2870     IF (MDFS.EQ.0) GO TO 480
2871     IF (L.LE.LDFSS.OR.L.GT.LDFS) GO TO 480
2872     CALL DRV (IX1,IY5,IX2,IY6)
2873     CALL DRV (IX1,IY7,IX2,IY8)
2874 480 CONTINUE
2875 490 CONTINUE
2876     CALL ADV (1)
2877     RETURN
2878 C
2879 C     FORMAT STATEMENTS
2880 C
2881 500 FORMAT (1H ,10A8)
2882 510 FORMAT (1H ,7HDENSITY,20X,2HN=,16,2X,2HT=,1PE10.4,4H SEC)
2883 520 FORMAT (1H ,8HPRESSURE,20X,2HN=,16,2X,2HT=,1PE10.4,4H SEC)
2884 530 FORMAT (1H ,11HTEMPERATURE,20X,2HN=,16,2X,2HT=,1PE10.4,4H SEC)
2885 540 FORMAT (1H ,11HMACH NUMBER,20X,2HN=,16,2X,2HT=,1PE10.4,4H SEC)
2886 550 FORMAT (1H ,17HTURBULENCE ENERGY,20X,2HN=,16,2X,2HT=,1PE10.4,4H SE
2887     1C)
2888 560 FORMAT (1H ,16HDISSIPATION RATE,20X,2HN=,16,2X,2HT=,1PE10.4,4H SEC
2889     1 )
2890 570 FORMAT (1H ,10HLCW VALUE=,1PE11.4,2X,11HHIGH VALUE=,E11.4,2X,12HLD
2891     1W CONTOUR=,E11.4,/,1X,13HHIGH CONTOUR=,E11.4,2X,14HDELTA CONTOUR=
2892     2 ,E11.4)
2893 580 FORMAT (1H ,18HVELOCITY VECTORS (.,1,2HX),10X,2HN=,16,2X,2HT=,1PE1
2894     1 0.4,4H SEC)
2895 590 FORMAT (1H ,19HPHYSICAL SPACE GRID,10X,2HN=,16,2X,2HT=,1PE10.4,4H
2896     1SEC)
2897     END

```

```

2898      SUBROUTINE SWITCH (ISWITCH)
2899 C
2900 C      .....
2901 C
2902 C      THIS SUBROUTINE SWITCHES THE DUAL FLOW SPACE BOUNDARY SOLUTIONS
2903 C      BETWEEN THE DUMMY ARRAYS AND THE SOLUTION ARRAYS
2904 C
2905 C      .....
2906 C
2907 C      ISWITCH=2 SWITCHES THE FLOW VARIABLES, BOUNDARY CONDITIONS, AND
2908 C      VISCOUS TERMS AT N AND N+1.  ISWITCH=3 SWITCHES THE FLOW VARIABLES
2909 C      AT N.  ISWITCH=5 SWITCHES THE FLOW VARIABLES AT N AND STORES THE
2910 C      VISCOUS TERMS.
2911 C
2912 *CALL,MCC
2913 C
2914 C      SWITCH THE FLOW VARIABLES AT N
2915 C
2916      DO 10 L=LDFSS,LDFSF
2917      UDFS=UL(L,N1)
2918      VDFS=VL(L,N1)
2919      PDFS=PL(L,N1)
2920      RODFS=ROL(L,N1)
2921      UL(L,N1)=U(L,MDFS,N1)
2922      VL(L,N1)=V(L,MDFS,N1)
2923      PL(L,N1)=P(L,MDFS,N1)
2924      ROL(L,N1)=RO(L,MDFS,N1)
2925      U(L,MDFS,N1)=UDFS
2926      V(L,MDFS,N1)=VDFS
2927      P(L,MDFS,N1)=PDFS
2928      RO(L,MDFS,N1)=RODFS
2929      IF (ITM.LE.1) GO TO 10
2930      ODFS=OL(L,N1)
2931      EDFS=EL(L,N1)
2932      OL(L,N1)=O(L,MDFS,N1)
2933      EL(L,N1)=E(L,MDFS,N1)
2934      O(L,MDFS,N1)=ODFS
2935      E(L,MDFS,N1)=EDFS
2936 10 CONTINUE
2937      IF (ISWITCH.EQ.3) RETURN
2938      IF (ISWITCH.EQ.5) GO TO 70
2939 C
2940 C      SWITCH THE FLOW VARIABLES AT N+1
2941 C
2942      DO 20 L=LDFSS,LDFSF
2943      UDFS=UL(L,N3)
2944      VDFS=VL(L,N3)
2945      PDFS=PL(L,N3)
2946      RODFS=ROL(L,N3)
2947      UL(L,N3)=U(L,MDFS,N3)
2948      VL(L,N3)=V(L,MDFS,N3)
2949      PL(L,N3)=P(L,MDFS,N3)
2950      ROL(L,N3)=RO(L,MDFS,N3)
2951      U(L,MDFS,N3)=UDFS
2952      V(L,MDFS,N3)=VDFS
2953      P(L,MDFS,N3)=PDFS
2954      RO(L,MDFS,N3)=RODFS
2955      IF (ITM.LE.1) GO TO 20
2956      ODFS=OL(L,N3)
2957      EDFS=EL(L,N3)
2958      OL(L,N3)=O(L,MDFS,N3)
2959      EL(L,N3)=E(L,MDFS,N3)
2960      O(L,MDFS,N3)=ODFS
2961      E(L,MDFS,N3)=EDFS
2962 20 CONTINUE
2963 C
2964 C      SWITCH THE BOUNDARY CONDITIONS
2965 C
2966      IF (LDFSS.NE.1) GO TO 40
2967      IF (ISUPER.GE.2) GO TO 40
2968      IF (ISUPER.EQ.1) GO TO 30
2969      PTDFS=PTL

```

```

2970      TTDFS=TTL
2971      THETDFS=THETAL
2972      PTL=PT(MDFS)
2973      TTL=TT(MDFS)
2974      THETAL=THETA(MDFS)
2975      PT(MDFS)=PTDFS
2976      TT(MDFS)=TTDFS
2977      THETA(MDFS)=THETDFS
2978      GO TO 40
2979      30 PIDFS=PIL
2980      PIL=PI(MDFS)
2981      PI(MDFS)=PIDFS
2982      40 IF (LDFSF.NE.LMAX) GO TO 50
2983      PEDFS=PEL
2984      PEL=PE(MDFS)
2985      PE(MDFS)=PEDFS
2986 C
2987 C      SWITCH THE VISCOUS TERMS
2988 C
2989      50 IF (CAV.EQ.O.O.AND.CHECK.EQ.O.O) RETURN
2990      DO 60 L=LDFSS,LDFSF
2991      QUDFS=QUTL(L)
2992      QVDFS=QVTL(L)
2993      QPDFS=QPTL(L)
2994      QRDFS=QRRTL(L)
2995      QUTL(L)=QUT(L,MDFS)
2996      QVTL(L)=QVT(L,MDFS)
2997      QPTL(L)=QPT(L,MDFS)
2998      QRRTL(L)=QROT(L,MDFS)
2999      QUT(L,MDFS)=QUDFS
3000      QVT(L,MDFS)=QVDFS
3001      QPT(L,MDFS)=QPDFS
3002      QROT(L,MDFS)=QRDFS
3003      IF (ITM.LE.1) GO TO 60
3004      QODFS=QOTL(L)
3005      QEDFS=QETL(L)
3006      QOTL(L)=QOT(L,MDFS)
3007      QETL(L)=QET(L,MDFS)
3008      QOT(L,MDFS)=QODFS
3009      QET(L,MDFS)=QEDFS
3010      60 CONTINUE
3011      RETURN
3012 C
3013 C      STORE THE VISCOUS TERMS
3014 C
3015      70 DO 80 L=LDFSS,LDFSF
3016      QUTL(L)=QUT(L,MDFS)
3017      QVTL(L)=QVT(L,MDFS)
3018      QPTL(L)=QPT(L,MDFS)
3019      QRRTL(L)=QROT(L,MDFS)
3020      IF (ITM.LE.1) GO TO 80
3021      QOTL(L)=QOT(L,MDFS)
3022      QETL(L)=QET(L,MDFS)
3023      80 CONTINUE
3024      RETURN
3025      END

```

```

3026      SUBROUTINE VISCOUS
3027 C
3028 C      .....
3029 C
3030 C      THIS SUBROUTINE CALCULATES THE LOCAL ARTIFICIAL VISCOSITY,
3031 C      MOLECULAR VISCOSITY, AND TURBULENCE TERMS
3032 C      .....
3033 C
3034 C
3035 *CALL MCC
3036      REAL MU, LA, LP2M, LPM, K, MUT, LAT, LP2MT, LPMT, KT, MUT1, MUT2,
3037      1 MUT3, MUT4, LAT1, LAT2, LAT3, LAT4, KT1, KT2, KT3, KT4, LP2MT1,
3038      2 LP2MT2, LP2MT3, LP2MT4, MU1, MU2, MU3, MU4, LA1, LA2, LA3, LA4,
3039      3 K1, K2, K3, K4, LP2M1, LP2M2, LP2M3, LP2M4, MUTD, MUTT
3040 C
3041      IP=1
3042      IF (N.NE.1) GO TO 10
3043      IF (NVC.NE.1) GO TO 10
3044      SIGOR=1.0/SIGO
3045      SIGER=1.0/SIGE
3046      F2I=FLOAT(2-IVBC)
3047      GAM=GAMMA-1.0
3048      DRK=GAM1*RG/PRA
3049      TRK=GAM1*RC/PRT
3050      GRG=GAMMA*RG
3051      XITM=0.0
3052      IF (ITM.EQ.2) XITM=0.67
3053      MU=0.0
3054      LA=0.0
3055      K=0.0
3056      MU1=0.0
3057      MU2=0.0
3058      MU3=0.0
3059      MU4=0.0
3060      LA1=0.0
3061      LA2=0.0
3062      LA3=0.0
3063      LA4=0.0
3064      K1=0.0
3065      K2=0.0
3066      K3=0.0
3067      K4=0.0
3068      LP2M=0.0
3069      LP2M1=0.0
3070      LP2M2=0.0
3071      LP2M3=0.0
3072      LP2M4=0.0
3073      LPM=0.0
3074      MUTD=0.0
3075      DLP2MT=0.0
3076      DMUT=0.0
3077      DLAT=0.0
3078      MUT=0.0
3079      LAT=0.0
3080      KT=0.0
3081      MUT1=0.0
3082      MUT2=0.0
3083      MUT3=0.0
3084      MUT4=0.0
3085      LAT1=0.0
3086      LAT2=0.0
3087      LAT3=0.0
3088      LAT4=0.0
3089      KT1=0.0
3090      KT2=0.0
3091      KT3=0.0
3092      KT4=0.0
3093      LP2MT=0.0
3094      SMU1=0.0
3095      SMU2=0.0
3096      SMU3=0.0
3097      SMU4=0.0

```

3098 LP2MT1=0.0  
3099 LP2MT2=0.0  
3100 LP2MT3=0.0  
3101 LP2MT4=0.0  
3102 LPMT=0.0  
3103 TML=0.0  
3104 RMU=0.0  
3105 RMU1=0.0  
3106 RMU2=0.0  
3107 RMU3=0.0  
3108 RMU4=0.0  
3109 RLA=0.0  
3110 RLA1=0.0  
3111 RLA2=0.0  
3112 RLA3=0.0  
3113 RLA4=0.0  
3114 RK=0.0  
3115 RK1=0.0  
3116 RK2=0.0  
3117 RK3=0.0  
3118 RK4=0.0  
3119 RLP2M=0.0  
3120 RLP2M1=0.0  
3121 RLP2M2=0.0  
3122 RLP2M3=0.0  
3123 RLP2M4=0.0  
3124 RLPM=0.0  
3125 RRO=0.0  
3126 RRO1=0.0  
3127 RRO2=0.0  
3128 RRO3=0.0  
3129 RRO4=0.0  
3130 RODIFF=0.0  
3131 EROT=0.0  
3132 TLMUR=0.0  
3133 AVMUR=0.0  
3134 DEL=0.0  
3135 QSMO=0.0  
3136 ESMO=0.0  
3137 ROXY1=0.0  
3138 ROXY2=0.0  
3139 ROXY3=0.0  
3140 ROXY4=0.0  
3141 ROXY12=0.0  
3142 BROY1=0.0  
3143 BROY2=0.0  
3144 BROY3=0.0  
3145 BROY4=0.0  
3146 BROY34=0.0  
3147 UROT=0.0  
3148 VROT=0.0  
3149 PROT=0.0  
3150 QDISS=0.0  
3151 QPROD=0.0  
3152 QDIFF=0.0  
3153 QROT1=0.0  
3154 EPROD=0.0  
3155 EDIFF=0.0  
3156 EDISS=0.0  
3157 ELOWR=0.0  
3158 ROOX=0.0  
3159 ROOY=0.0  
3160 ATERM=0.0  
3161 ATERM1=0.0  
3162 ATERM2=0.0  
3163 ATERM3=0.0  
3164 ATERM4=0.0  
3165 UVTA=0.0  
3166 VVTA=0.0  
3167 PVTA=0.0  
3168 PCTA=0.0  
3169 RODIFFA=0.0

```

3170      UROTA=0.0
3171      VROTA=0.0
3172      PROTA=0.0
3173      QPRODA=0.0
3174      QDIFFA=0.0
3175      EPRODA=0.0
3176      EDIFFA=0.0
3177      QROTTA=0.0
3178      EROTA=0.0
3179      ELOWRA=0.0
3180      SMT=1.0
3181      RDUM=0.0
3182      ECHECK=ABS(EMU)+ABS(ELA)+ABS(EK)
3183      IF (ECHECK.EQ.0.0) GO TO 10
3184      IF (ABS(EMU).EQ.ABS(ELA).AND.ABS(EMU).EQ.ABS(EK)) ECHECK=-1.0
3185 C
3186      10 NLINE=0
3187      IF (IAV.EQ.0) GO TO 30
3188      IF (NC.NE.NPRINT.AND.(N.NE.NMAX.AND.ISTOP.EQ.0)) GO TO 30
3189      IF (IAV.EQ.2) GO TO 20
3190      IF (NVC.GT.2.AND.NVC.NE.NVCM+1) GO TO 30
3191      20 WRITE (6,1460)
3192      NP=N+NSTART
3193      WRITE (6,1450) NP,NVC
3194 C
3195 C      DO LOOP SET-UP
3196 C
3197      30 MIS=1
3198      MIF=MMAX
3199      IF (IVC.EQ.0) GO TO 40
3200      IF (NVC.EQ.1) GO TO 40
3201      MIS=MVCB
3202      MIF=MVCT
3203      40 IDFS=0
3204 C
3205      IF (MDFS.EQ.0) GO TO 70
3206      IF (NVC.EQ.1.AND.MDFSC.NE.0) GO TO 70
3207      CALL SWITCH (3)
3208      MIS=1
3209      IF (NVC.NE.1) MIS=MVCB
3210      MIF=MDFS
3211      IB=3
3212      GO TO 70
3213      50 CALL SWITCH (5)
3214      MIS=MDFS+1
3215      MIF=MMAX
3216      IF (NVC.NE.1) MIF=MVCT
3217      IB=4
3218      GO TO 70
3219      60 IDFS=1
3220      MIS=MDFS
3221      MIF=MDFS
3222 C
3223 C      BEGIN THE L OR X DO LOOP
3224 C
3225      70 DO 1410 L=1,LMAX
3226      LMAP=L
3227      LDFS=0
3228      IF (L.GE.LDFS.AND.L.LE.LDFS) LDFS=1
3229      IF (L.NE.1.AND.L.NE.LMAX) DXF=0.5*(XP(L+1)-XP(L-1))
3230      IF (L.EQ.1) DXF=XP(2)-XP(1)
3231      IF (L.EQ.LMAX) DXF=XP(LMAX)-XP(L1)
3232      IF (IDFS.EQ.1.AND.LDFS.EQ.0) GO TO 1410
3233 C
3234 C      CALCULATE THE WALL SHEAR STRESS FOR THE MIXING LENGTH TURBULENCE
3235 C      MODELS
3236 C
3237      IF (ITM.NE.1.AND.ITM.NE.2) GO TO 160
3238      IF (MDFS.EQ.0) GO TO 80
3239      IMLM=1
3240      IF (LDFS.NE.0) IMLM=2
3241      80 IF (IMLM.EQ.1) GO TO 160

```

```

3242     IF (MDFS.NE.O) GO TO 100
3243     IF (NGCB.NE.O) GO TO 90
3244     MT=M1
3245     MMAP=MMAX
3246     GO TO 110
3247 90  MT=2
3248     MMAP=1
3249     GO TO 110
3250 100 MMAP=MDFS
3251     IBD=IB
3252     IB=4
3253     MT=MDFS+1
3254 110 CALL MAP
3255     IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 120
3256     UAVG=0.25*(U(L-1,MT,N1)+U(L+1,MT,N1)+2.0*U(L,MT,N1))
3257     VAVG=0.25*(V(L-1,MT,N1)+V(L+1,MT,N1)+2.0*V(L,MT,N1))
3258     GO TO 130
3259 120 UAVG=U(L,MT,N1)
3260     VAVG=V(L,MT,N1)
3261 130 TAUW=ABS(BE3*UAVG+AL3*VAVG)*DYP
3262     IF (MDFS.EQ.O) GO TO 160
3263     TAUWP=TAUW
3264     IB=3
3265     CALL MAP
3266     MT=MDFS-1
3267     IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 140
3268     UAVG=0.25*(U(L-1,MT,N1)+U(L+1,MT,N1)+2.0*U(L,MT,N1))
3269     VAVG=0.25*(V(L-1,MT,N1)+V(L+1,MT,N1)+2.0*V(L,MT,N1))
3270     GO TO 150
3271 140 UAVG=U(L,MT,N1)
3272     VAVG=V(L,MT,N1)
3273 150 TAUWM=ABS(BE3*UAVG+AI3*VAVG)*DYP
3274     IB=IBD
3275 C
3276 C     BEGIN THE M OR Y DO LOOP
3277 C
3278 160 DO 1400 M=MIS,MIF
3279     IF (IVC.EQ.O) GO TO 190
3280     IF (NVC.NE.1) GO TO 190
3281     IF (MVCB.NE.1) GO TO 170
3282     IF (M.EQ.1) GO TO 1400
3283     GO TO 180
3284 170 IF (MVCT.NE.MMAX) GO TO 180
3285     IF (M.EQ.MMAX) GO TO 1400
3286 180 IF (M.LT.MVCB.OR.M.GT.MVCT) GO TO 190
3287     GO TO 1400
3288 190 IES=0
3289     IF (M.EQ.MMAX) IES=1
3290     IF (M.EQ.1.AND.NGCB.NE.O) IES=1
3291     IF (M.EQ.MDFS.AND.LDFS.NE.O) IES=1
3292 C
3293 C     CALCULATE THE TURBULENT MIXING LENGTH
3294 C
3295     IF (ITM.EQ.O.OR.ITM.EQ.3) GO TO 210
3296     IF (NVC.NF.1) GO TO 200
3297     IF (M.EQ.MIS) CALL MIXLEN (L,M)
3298     IF (M.EQ.MVCT+1.AND.MVCB.EQ.1) CALL MIXLEN (L,M)
3299     IF (M.EQ.MVCT+1.AND.(MDFSC.NE.O.AND.LDFS.NE.O)) CALL MIXLEN (L,M)
3300     GO TO 210
3301 200 IF (M.EQ.MVCB) CALL MIXLEN (L,M)
3302     IF (M.EQ.MDFS.AND.(LDFS.NE.O.AND.IDFS.NE.O)) CALL MIXLEN (L,M)
3303     IF (M.EQ.MDFS+1.AND.MDFS.NE.O) CALL MIXLEN (L,M)
3304 C
3305 C     SET SPECIAL CONDITIONS FOR L=1 OR LMAX
3306 C
3307 210 IF (L.NE.LMAX.AND.L.NE.1) GO TO 230
3308     TML=0.0
3309     MUT=0.0
3310     TLMUR=0.0
3311     AVMUR=0.0
3312     IF (M.EQ.1.OR.M.EQ.MMAX) GO TO 1340
3313     IF (L.EQ.1) GO TO 220

```

```

3314      IF (LDFSF.EQ.LMAX.AND.M.EQ.MDFS) GO TO 1340
3315      GO TO 230
3316 220 IF (LDFSS.EQ.1.AND.M.EQ.MDFS) GO TO 1340
3317 C
3318 230 RORR=1.0/RO(L,M,N1)
3319      MMAP=M
3320      CALL MAP
3321      OM=2.0*OM1*OM2/(OM1+OM2)
3322      BE=2.0*BE3+BE4/(RE3+BE4)
3323      AL34=AL3+AL4
3324      DE34=DE3+DE4
3325      IF (AL34.EQ.O.O) AL34=1.0
3326      IF (DE34.EQ.O.O) DE34=1.0
3327      AL=2.0*AL3*AL4/AL34
3328      DE=2.0*DE3*DE4/DE34
3329      IF (YP.NE.O.O) RYP=1.0/YP
3330      YP3=YP-O.5*DY/RE3
3331      YP4=YP+O.5*DY/BE4
3332 C
3333 C      CHECK FOR ARTIFICIAL VISCOSITY
3334 C
3335      IF (CAV.EQ.O.O) GO TO 250
3336      IF (L.LT.LSS.AND.CHECK.EQ.O.O) GO TO 1340
3337      IF (L.GT.LSF.AND.CHECK.EQ.O.O) GO TO 1340
3338      IF (M.LT.MSS.AND.CHECK.EQ.O.O) GO TO 1340
3339      IF (M.GT.MSF.AND.CHECK.EQ.O.O) GO TO 1340
3340      XV=U(L,M,N1)*U(L,M,N1)+V(L,M,N1)*V(L,M,N1)
3341      XA=GAMMA*P(L,M,N1)*RORR
3342      XM=XV/XA
3343      SMT=1.0
3344      IF (NOSLIP.NE.O.O.AND.XM.LT.1.0) SMT=XM
3345      IF (SMACH.EQ.O.O) GO TO 250
3346      IF (XM.LT.SMACH*SMACH.AND.CHECK.EQ.O.O) GO TO 240
3347      GO TO 250
3348 240 CUT(L,M)=O.O
3349      QVT(L,M)=O.O
3350      OPT(L,M)=O.O
3351      QROT(L,M)=O.O
3352      GO TO 1340
3353 C
3354 C      CALCULATE THE X DERIVATIVES
3355 C
3356 250 T=P(L,M,N1)/(RO(L,M,N1)*RG)
3357      A=SQRT(GRG*T)
3358      IF (L.EQ.1) GO TO 260
3359      ULM=U(L-1,M,N1)
3360      VLM=V(L-1,M,N1)
3361      PLM=P(L-1,M,N1)
3362      ROLM=RO(L-1,M,N1)
3363      QLM=Q(L-1,M,N1)
3364      ELM=E(L-1,M,N1)
3365      IF (L.EQ.LMAX) GO TO 280
3366 260 ULP=U(L+1,M,N1)
3367      VLP=V(L+1,M,N1)
3368      PLP=P(L+1,M,N1)
3369      ROLP=RO(L+1,M,N1)
3370      QLP=Q(L+1,M,N1)
3371      ELP=E(L+1,M,N1)
3372      IF (L.EQ.1) GO TO 290
3373      IF (M.NE.MDFS) GO TO 280
3374      IF (L.NE.LDFSS-1) GO TO 270
3375      ULP=O.5*(ULP+UL(L+1,N1))
3376      VLP=O.5*(VLP+VL(L+1,N1))
3377      PLP=O.5*(PLP+PL(L+1,N1))
3378      ROLP=O.5*(ROLP+ROL(L+1,N1))
3379      IF (1/M.LE.1) GO TO 280
3380      QLP=O.5*(QLP+QL(L+1,N1))
3381      ELP=O.5*(ELP+EL(L+1,N1))
3382      GO TO 280
3383 270 IF (L.NE.LDFSF+1) GO TO 280
3384      ULM=O.5*(ULM+UL(L-1,N1))
3385      VLM=O.5*(VLM+VL(L-1,N1))

```



```

3386     PLM=0.5*(PLM+PL(L-1,N1))
3387     ROLM=0.5*(ROLM+ROL(L-1,N1))
3388     IF (ITM.LE.1) GO TO 280
3389     QLM=0.5*(QLM+QL(L-1,N1))
3390     ELM=0.5*(ELM+EL(L-1,N1))
3391 280  UX1=(U(L,M,N1)-ULM)*DXR
3392     VX1=(V(L,M,N1)-VLM)*DXR
3393     TLM=PLM/(ROLM*RG)
3394     TX1=(T-TLM)*DXR
3395     ROX1=(RO(L,M,N1)-ROLM)*DXR
3396     IF (ITM.GE.2) ROOX=(RO(L,M,N1)*Q(L,M,N1)-ROLM*QLM)*DXR
3397     IF (L.EQ.LMAX) GO TO 340
3398 290  UX2=(ULP-U(L,M,N1))*DXR
3399     VX2=(VLP-V(L,M,N1))*DXR
3400     TLP=PLP/(R.LP*RG)
3401     TX2=(T-TLP)*DXR
3402     ROX2=(ROLP-RO(L,M,N1))*DXR
3403     IF (ITM.GE.2) ROOX=(ROLP*QLP-RO(L,M,N1)*Q(L,M,N1))*DXR
3404     IF (L.EQ.1) GO TO 340
3405     IF (CAV.EQ.O.O) GO TO 300
3406     IF (ISS.EQ.O.O) GO TO 300
3407     ALP=SQRT(GRG*TLP)
3408     ALM=SQRT(GRG*TLM)
3409     AX1=(A-ALM)*DXR
3410     AX2=(ALP-A)*DXR
3411 300  IF (ITM.LE.1) GO TO 320
3412     ROOX=(ROLP*QLP-ROLM*QLM)*DXR*0.5
3413     OX1=(O(L,M,N1)-OLM)*DXR
3414     OX2=(QLP-O(L,M,N1))*DXR
3415     QX2=0.5*(SQRT(OLP)-SQRT(OLM))*DXR
3416     IF (ITM.EQ.3) GO TO 310
3417     ROSQ=RO(L,M,N1)*SQRT(Q(L,M,N1))
3418     ROSQ1=RO(L-1,M,N1)*SQRT(Q(L-1,M,N1))
3419     ROSQ2=RO(L+1,M,N1)*SQRT(Q(L+1,M,N1))
3420     GO TO 320
3421 310  EX1=(E(L,M,N1)-ELM)*DXR
3422     EX2=(ELP-E(L,M,N1))*DXR
3423     MUT=COMU*RO(L,M,N1)*Q(L,M,N1)*Q(L,M,N1)*LC/E(L,M,N1)
3424     MUT1=COMU*ROLM*QLM*QLM*LC/ELM
3425     MUT2=COMU*ROLP*QLP*QLP*LC/ELP
3426 320  IF (M.NE.MDFS.OR.LDFS.EQ.O) GO TO 330
3427     IF (IB.EQ.3) GO TO 680
3428     GO TO 490
3429 330  IF (M.EQ.1) GO TO 490
3430     IF (M.EQ.MMAX) GO TO 680
3431 C
3432 C     BEGIN THE INTERIOR POINT Y DERIVATIVE CALCULATION
3433 C
3434 340  DYP=DY/BE
3435     UMP=U(L,M+1,N1)
3436     UMM=U(L,M-1,N1)
3437     VMP=V(L,M+1,N1)
3438     VMM=V(L,M-1,N1)
3439     PMP=P(L,M+1,N1)
3440     PMM=P(L,M-1,N1)
3441     RCMP=RO(L,M+1,N1)
3442     ROMM=RO(L,M-1,N1)
3443     QMP=Q(L,M+1,N1)
3444     QMM=Q(L,M-1,N1)
3445     EMP=E(L,M+1,N1)
3446     EMM=E(L,M-1,N1)
3447     IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 350
3448     ULFMP=U(L+1,M+1,N1)
3449     ULMMP=U(L-1,M+1,N1)
3450     ULPMM=U(L+1,M-1,N1)
3451     ULMMM=U(L-1,M-1,N1)
3452     VLPMP=V(L+1,M+1,N1)
3453     VLMMP=V(L-1,M+1,N1)
3454     VLPMM=V(L+1,M-1,N1)
3455     VLMMM=V(L-1,M-1,N1)
3456     PLPMP=P(L+1,M+1,N1)
3457     PLMMP=P(L-1,M+1,N1)
3458     PLPMM=P(L+1,M-1,N1)
3459     PLMMM=P(L-1,M-1,N1)

```

```

3460      ROLPMP=RO(L+1,M+1,N1)
3461      ROLMMP=RO(L-1,M+1,N1)
3462      ROLPMN=RO(L+1,M-1,N1)
3463      ROLMMN=RO(L-1,M-1,N1)
3464      QLPMP=Q(L+1,M+1,N1)
3465      QLMMP=Q(L-1,M+1,N1)
3466      QLPMM=Q(L+1,M-1,N1)
3467      QLMMM=Q(L-1,M-1,N1)
3468      ELPMP=E(L+1,M+1,N1)
3469      ELMMP=E(L-1,M+1,N1)
3470      ELPMM=E(L+1,M-1,N1)
3471      ELMMM=E(L-1,M-1,N1)
3472      350 IF (IVC.EQ.0) GO TO 380
3473      IF (NVC.EQ.1) GO TO 380
3474      IF (M.EQ.MVCB) GO TO 360
3475      IF (M.EQ.MVCT) GO TO 370
3476      GO TO 380
3477      C
3478      C      LINEAR INTERPOLATION IN TIME FOR M=MVCB
3479      C
3480      360 UMM=U(L,M-1,NN1)+RIND*(U(L,M-1,NN3)-U(L,M-1,NN1))
3481      VMM=V(L,M-1,NN1)+RIND*(V(L,M-1,NN3)-V(L,M-1,NN1))
3482      PMM=P(L,M-1,NN1)+RIND*(P(L,M-1,NN3)-P(L,M-1,NN1))
3483      ROMM=RO(L,M-1,NN1)+RIND*(RO(L,M-1,NN3)-RO(L,M-1,NN1))
3484      OMM=O(L,M-1,NN1)+RIND*(O(L,M-1,NN3)-O(L,M-1,NN1))
3485      EMM=E(L,M-1,NN1)+RIND*(E(L,M-1,NN3)-E(L,M-1,NN1))
3486      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 390
3487      ULPMM=U(L+1,M-1,NN1)+RIND*(U(L+1,M-1,NN3)-U(L+1,M-1,NN1))
3488      ULMMM=U(L-1,M-1,NN1)+RIND*(U(L-1,M-1,NN3)-U(L-1,M-1,NN1))
3489      VLPMM=V(L+1,M-1,NN1)+RIND*(V(L+1,M-1,NN3)-V(L+1,M-1,NN1))
3490      VLMMM=V(L-1,M-1,NN1)+RIND*(V(L-1,M-1,NN3)-V(L-1,M-1,NN1))
3491      PLPMM=P(L+1,M-1,NN1)+RIND*(P(L+1,M-1,NN3)-P(L+1,M-1,NN1))
3492      PLMMM=P(L-1,M-1,NN1)+RIND*(P(L-1,M-1,NN3)-P(L-1,M-1,NN1))
3493      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 380
3494      ROLPMM=RO(L+1,M-1,NN1)+RIND*(RO(L+1,M-1,NN3)-RO(L+1,M-1,NN1))
3495      ROLMMM=RO(L-1,M-1,NN1)+RIND*(RO(L-1,M-1,NN3)-RO(L-1,M-1,NN1))
3496      IF (ITM.LE.1) GO TO 380
3497      QLPMM=Q(L+1,M-1,NN1)+RIND*(Q(L+1,M-1,NN3)-Q(L+1,M-1,NN1))
3498      QLMMM=Q(L-1,M-1,NN1)+RIND*(Q(L-1,M-1,NN3)-Q(L-1,M-1,NN1))
3499      ELPMM=E(L+1,M-1,NN1)+RIND*(E(L+1,M-1,NN3)-E(L+1,M-1,NN1))
3500      ELMMM=E(L-1,M-1,NN1)+RIND*(E(L-1,M-1,NN3)-E(L-1,M-1,NN1))
3501      GJ TO 380
3502      C
3503      C      LINEAR INTERPOLATION IN TIME FOR M=MVCT
3504      C
3505      370 UMP=UU1(L)+RIND*(UU2(L)-UU1(L))
3506      VMP=VV1(L)+RIND*(VV2(L)-VV1(L))
3507      PMP=PP1(L)+RIND*(PP2(L)-PP1(L))
3508      ROMP=ROR01(L)+RIND*(ROR02(L)-ROR01(L))
3509      QMP=QQ1(L)+RIND*(QQ2(L)-QQ1(L))
3510      EMP=EE1(L)+RIND*(EE2(L)-EE1(L))
3511      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 390
3512      ULPMP=UU1(L+1)+RIND*(UU2(L+1)-UU1(L+1))
3513      ULMMP=UU1(L-1)+RIND*(UU2(L-1)-UU1(L-1))
3514      VLPMP=VV1(L+1)+RIND*(VV2(L+1)-VV1(L+1))
3515      VLMMP=VV1(L-1)+RIND*(VV2(L-1)-VV1(L-1))
3516      PLPMP=PP1(L+1)+RIND*(PP2(L+1)-PP1(L+1))
3517      PLMMP=PP1(L-1)+RIND*(PP2(L-1)-PP1(L-1))
3518      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 380
3519      ROLPMP=ROR01(L+1)+RIND*(ROR02(L+1)-ROR01(L+1))
3520      ROLMMP=ROR01(L-1)+RIND*(ROR02(L-1)-ROR01(L-1))
3521      IF (ITM.LE.1) GO TO 380
3522      QLPMP=QQ1(L+1)+RIND*(QQ2(L+1)-QQ1(L+1))
3523      QLMMP=QQ1(L-1)+RIND*(QQ2(L-1)-QQ1(L-1))
3524      ELPMP=EE1(L+1)+RIND*(EE2(L+1)-EE1(L+1))
3525      ELMMP=EE1(L-1)+RIND*(EE2(L-1)-EE1(L-1))
3526      C
3527      C      CALCULATE THE INTERIOR POINT Y DERIVATIVES
3528      C
3529      380 IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 390
3530      UY1=0.25*(UMP+ULMMP-UMM-ULMMM)*DYR
3531      JY2=0.25*(UMP+ULPMP-UMM-ULPMM)*DYR

```

```

3532      VY1=O.25*(VMP+VLMMP-VMM-VLMMM)*DYR
3533      VY2=O.25*(VMP+VLPMP-VMM-VLPM)*DYR
3534      UX3=O.25*(ULP+ULPMM-ULM-ULMMM)*DXR
3535      UX4=O.25*(ULP+ULPMP-ULM-ULMMP)*DXR
3536      VX3=O.25*(VLP+VLPMM-VLM-VLMMM)*DXR
3537      VX4=O.25*(VLP+VLPMP-VLM-VLMMP)*DXR
3538  390 VY3=(V(L,M,N1)-VMM)*DYR
3539      VY4=(VMP-V(L,M,N1))*DYR
3540      UY3=(U(L,M,N1)-UMM)*DYR
3541      UY4=(UMP-U(L,M,N1))*DYR
3542      TMM=PMM/(ROMM*RG)
3543      TMP=PMP/(ROMP*RG)
3544      TY3=(T-TMM)*DYR
3545      TY4=(TMP-T)*DYR
3546      ROY3=(RO(L,M,N1)-ROMM)*DYR
3547      ROY4=(ROMP-RO(L,M,N1))*DYR
3548      IF (ITM.LT.2) GO TO 400
3549      ROOY=(ROMP*OMP-ROMM*OMM)*DYR*O.5
3550      IF (IOSD.EQ.O.OR.NVC.EQ.1) GO TO 400
3551      IF (M.EQ.MVCB.OR.M.EQ.MVCT) GO TO 400
3552      ROOY=OQT(L,M)
3553  400 IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 410
3554      TLM=PLMM/(ROLMM*RG)
3555      TLMMP=PLMMP/(ROLMMP*RG)
3556      TLPMM=PLPMM/(ROLPMM*RG)
3557      TLPMP=PLPMP/(ROLPMP*RG)
3558      TY1=O.25*(TMP+TLMMP-TMM-TLMM)*DYR
3559      TY2=O.25*(TLPMP+TMP-TLPMM-TMM)*DYR
3560      TX3=O.25*(TLP+TLPMM-TLM-TLMM)*DXR
3561      TX4=O.25*(TLMMP+TLP-TLMMP-TLM)*DXR
3562      IF (ITM.EQ.O.AND.CAV.EQ.O.O) GO TO 450
3563      ROY1=O.25*(ROMP+ROLMMP-ROMM-ROLMM)*DYR
3564      ROY2=O.25*(ROMP+ROLPMP-ROMM-ROLPM)*DYR
3565      ROX3=O.25*(ROLP+ROLPMM-ROLM-ROLMM)*DXR
3566      ROX4=C.25*(ROLP+ROLPMP-ROLM-ROLMMP)*DXR
3567  410 IF (CAV.EQ.O.O) GO TO 430
3568      IF (NDIM.EQ.O) GO TO 420
3569      ATERM=V(L,M,N1)*RYP
3570      ATERM3=O.5*(V(L,M,N1)+VMM)*RYP
3571      ATERM4=O.5*(V(L,M,N1)+VMP)*RYP
3572      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 420
3573      ATERM1=O.5*(V(L,M,N1)+V(L-1,M,N1))*RYP
3574      ATERM2=O.5*(V(L,M,N1)+V(L+1,M,N1))*RYP
3575  420 IF (ISS.EQ.O) GO TO 430
3576      AMP=SQRT(GRG*TMP)
3577      AMM=SQRT(GRG*TMM)
3578      AY3=(A-AMM)*DYR
3579      AY4=(AMP-A)*DYR
3580  430 IF (ITM.LE.1) GO TO 450
3581      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 450
3582      QY1=O.25*(OMP+OLMMP-OMM-OLMM)*DYR
3583      QY2=O.25*(OMP+OLPMP-OMM-OLPM)*DYR
3584      QX3=O.25*(OLP+OLPMM-OLM-OLMM)*DXR
3585      QX4=O.25*(OLP+OLPMP-OLM-OLMMP)*DXR
3586      QY3=(Q(L,M,N1)-OMM)*DYR
3587      QY4=(OMP-Q(L,M,N1))*DYR
3588      QY=O.5*(SQRT(OMP)-SQRT(OMM))*DYR
3589      IF (ITM.EQ.3) GO TO 440
3590      ROSQ3=ROMM*SQRT(OMM)
3591      ROSQ4=ROMP*SQRT(OMP)
3592      GO TO 450
3593  440 EY1=O.25*(EMP+ELMMP-EMM-ELMM)*DYR
3594      EY2=O.25*(EMP+ELPMP-EMM-ELPM)*DYR
3595      EX3=O.25*(ELP+ELPMM-ELM-ELMM)*DXR
3596      EX4=O.25*(ELP+ELPMP-ELM-ELMMP)*DXR
3597      EY3=(E(L,M,N1)-EMM)*DYR
3598      EY4=(EMP-E(L,M,N1))*DYR
3599      MUT3=COMU*ROMM*OMM*OMM*LC/EMM
3600      MUT4=COMU*ROMP*OMP*OMP*LC/EMP
3601 C
3602 C      SET THE BOUNDARY CONDITIONS FOR L=1 OR LMAX
3603 C

```

```

3604 450 IF (L.NE.LMAX.AND.L.NE.1) GO TO 850
3605     IF (L.EQ.1) GO TO 460
3606     UX2=-UX1
3607     VX2=-VX1
3608     TX2=-TX1
3609     ROX2=-ROX1
3610     ROLP=ROLM
3611     TLP=TLM
3612     OLP=OLM
3613     ELP=ELM
3614     GO TO 470
3615 460 UX1=-UX2
3616     VX1=-VX2
3617     TX1=-TX2
3618     ROX1=-ROX2
3619     ROLM=ROLP
3620     TLM=TLP
3621     OLM=OLP
3622     ELM=ELP
3623 470 YP1=YP
3624     YP2=YP
3625     UY1=0.0
3626     UY2=0.0
3627     VY1=0.0
3628     VY2=0.0
3629     UX3=C.0
3630     UX4=0.0
3631     VX3=0.0
3632     VX4=0.0
3633     TY1=0.0
3634     TY2=0.0
3635     TX3=C.0
3636     TX4=0.0
3637     ROY1=0.0
3638     ROY2=0.0
3639     ROX3=0.0
3640     ROX4=0.0
3641     ATERM1=ATERM
3642     ATERM2=ATERM
3643     AX1=0.0
3644     AX2=0.0
3645     IF (ITM.LE.1) GO TO 850
3646     OX1=0.0
3647     OX2=0.0
3648     OY1=0.0
3649     OY2=0.0
3650     OX3=0.0
3651     OX4=0.0
3652     OY3=0.0
3653     OY4=0.0
3654     EX1=0.0
3655     EX2=0.0
3656     EY1=0.0
3657     EY2=0.0
3658     EX3=0.0
3659     EX4=0.0
3660     EY3=0.0
3661     EY4=0.0
3662     IF (ITM.EQ.3) GO TO 480
3663     ROSQ=RO(L,A.N1)*SQRT(O(L,M,N1))
3664     ROSQ1=ROSQ
3665     ROSQ2=ROSQ
3666     ROSQ3=ROMM*SQRT(OMM)
3667     ROSQ4=ROMP*SQRT(OMP)
3668     GO TO 850
3669 480 MUT=COMU*RO(L,M,N1)*O(L,M,N1)*O(L,M,N1)*LC/E(L,M,N1)
3670     MUT1=MUT
3671     MUT2=MUT-
3672     MUT3=COMU*ROMM*OMM*QMM*LC/EMM
3673     MUT4=COMU*ROMP*OMP*QMP*LC/EMP
3674     GO TO 850
3675 C

```

```

3676 C      BEGIN THE CENTERBODY OR UPPER DUAL FLOW SPACE BOUNDARY POINT
3677 C      Y DERIVATIVE CALCULATION
3678 C
3679     490 DYP=DY/BE4
3680     UMP=U(L,M+1,N1)
3681     VMP=V(L,M+1,N1)
3682     PMP=P(L,M+1,N1)
3683     ROMP=RO(L,M+1,N1)
3684     QMP=Q(L,M+1,N1)
3685     EMP=E(L,M+1,N1)
3686     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
3687     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
3688     UY4=(UMP-U(L,M,N1))*DYR
3689     VY4=(VMP-V(L,M,N1))*DYR
3690     TMP=PMP/(ROMP*RG)
3691     TLMMP=P(L-1,M+1,N1)/(RO(L-1,M+1,N1)*RG)
3692     TLPMP=P(L+1,M+1,N1)/(RO(L+1,M+1,N1)*RG)
3693     TX4=0.25*(TLPMP+TLP-TLMMP-TLM)*DXR
3694     TY4=(TMP-T)*DYR
3695     IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 500
3696     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)
3697     1)*DXR
3698     ROY4=(ROMP-RO(L,M,N1))*DYR
3699     IF (ITM.LE.1) GO TO 500
3700     OX4=0.25*(O(L+1,M,N1)+O(L+1,M+1,N1)-O(L-1,M,N1)-O(L-1,M+1,N1))*DXR
3701     OY4=(OMP-O(L,M,N1))*DYR
3702     IF (ITM.EQ.2) GO TO 500
3703     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
3704     EY4=(EMP-E(L,M,N1))*DYR
3705 C
3706 C      REFLECT THE CENTREBODY OR UPPER DUAL FLOW SPACE BOUNDARY
3707 C      CONDITIONS
3708 C
3709     500 IF (M.EQ.1.AND.NGCB.EQ.0) GO TO 590
3710     IF (1VBC.NE.0) GO TO 800
3711     IF (M.EQ.MOFS) GO TO 310
3712     DNXY=NXYCB(L)
3713     DNXP=NXYCB(L+1)
3714     DNXYM=NXYCB(L-1)
3715     GO TO 520
3716     510 DNXY=NXYU(L)
3717     DNXP=NXYU(L+1)
3718     DNXYM=NXYU(L-1)
3719     520 THEW=ATAN(-DNXY)
3720     IF (UMP.EQ.0.0) GO TO 530
3721     THE=ATAN(VMP/UMP)
3722     GO TO 540
3723     530 THE=0.0
3724     540 IF (UMP.LT.0.0) THE=THE+3.14159
3725     VMAG=SQRT(UMP*UMP+VMP*VMP)
3726     RTHE=2.0*THEW-THE
3727     IF (NOSLIP.EQ.1.AND.NGCB.NE.0) RTHE=3.14159+THE
3728     IF (NOSLIP.EQ.1.AND.M.EQ.MOFS) RTHE=3.14159+THE
3729     UMM=VMAG*COS(RTHE)
3730     VMM=VMAG*SIN(RTHE)
3731     THEW=ATAN(-DNXP)
3732     IF (U(L+1,M+1,N1).EQ.0.0) GO TO 550
3733     THE=ATAN(V(L+1,M+1,N1)/U(L+1,M+1,N1))
3734     GO TO 560
3735     550 THE=0.0
3736     560 IF (U(L+1,M+1,N1).LT.0.0) THE=THE+3.14159
3737     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))
3738     RTHE=2.0*THEW-THE
3739     IF (NOSLIP.EQ.1.AND.NGCB.NE.0) RTHE=3.14159+THE
3740     IF (NOSLIP.EQ.1.AND.M.EQ.MOFS) RTHE=3.14159+THE
3741     ULPMM=VMAG*COS(RTHE)
3742     VLPMM=VMAG*SIN(RTHE)
3743     THEW=ATAN(-DNXYM)
3744     IF (U(L-1,M+1,N1).EQ.0.0) GO TO 570
3745     THE=ATAN(V(L-1,M+1,N1)/U(L-1,M+1,N1))
3746     GO TO 580
3747     570 THE=0.0

```

```

3748 580 IF (U(L-1,M+1,N1).LT.0.0) THE=THE+3.14159
3749 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))
3750 RTHE=2.0*THEW-THE
3751 IF (NOSLIP.EQ.1.AND.NGCB.NE.0) RTHE=3.14159*THE
3752 IF (NOSLIP.EQ.1.AND.M.EQ.MDFS) RTHE=3.14159*THE
3753 ULMM=VMAG*COS(RTHE)
3754 VLMM=VMAG*SIN(RTHE)
3755 C
3756 RFL=2.0*DNXNY*DYP/(1.0+DNXNY*DNXNY)
3757 RFLP=2.0*DNXNYP*DYP/(1.0+DNXNYP*DNXNYP)
3758 RFLM=2.0*DNXNYM*DYP/(1.0+DNXNYM*DNXNYM)
3759 TTERM=0.5*(OM1*TX1+OM2*TX2)
3760 TMM=TMP+TTERM*RFL
3761 TLPMM=TLPMP+TTERM*RFLP
3762 TLMM=TLMP+TTERM*RFLM
3763 IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 610
3764 ROTERM=0.5*(OM1*ROX1+OM2*ROX2)
3765 ROMM=ROMP+ROTERM*RFL
3766 ROLPMM=RO(L+1,M+1,N1)+ROTERM*RFLP
3767 ROLMM=RO(L-1,M+1,N1)+ROTERM*RFLM
3768 IF (ITM.LE.1) GO TO 610
3769 QTERM=0.5*(OM1*QX1+OM2*QX2)
3770 QMM=QMP+QTERM*RFL
3771 QLPMM=Q(L+1,M+1,N1)+QTERM*RFLP
3772 QLMM=Q(L-1,M+1,N1)+QTERM*RFLM
3773 IF (ITM.EQ.2) GO TO 610
3774 ETERM=0.5*(OM1*EX1+OM2*EX2)
3775 EMM=EMP+ETERM*RFL
3776 ELPMM=E(L+1,M+1,N1)+ETERM*RFLP
3777 ELMM=E(L-1,M+1,N1)+ETERM*RFLM
3778 GO TO 610
3779 C
3780 C REFLECT THE CENTERLINE OR MIDPLANE BOUNDARY CONDITIONS
3781 C
3782 590 UMM=UMP
3783 VMM=-VMP
3784 ULPMM=U(L+1,M+1,N1)
3785 VLPMM=-V(L+1,M+1,N1)
3786 ULMM=U(L-1,M+1,N1)
3787 VLMM=-V(L-1,M+1,N1)
3788 TMM=TMP
3789 TLPMM=TLPMP
3790 TLMM=TLMP
3791 IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 610
3792 ROMM=ROMP
3793 ROLPMM=RO(L+1,M+1,N1)
3794 ROLMM=RO(L-1,M+1,N1)
3795 IF (ITM.LE.1) GO TO 610
3796 QMM=QMP
3797 QLPMM=Q(L+1,M+1,N1)
3798 QLMM=Q(L-1,M+1,N1)
3799 IF (ITM.EQ.2) GO TO 610
3800 EMM=EMP
3801 ELPMM=E(L+1,M+1,N1)
3802 ELMM=E(L-1,M+1,N1)
3803 GO TO 610
3804 C
3805 C EXTRAPOLATE THE CENTERBODY OR UPPER DUAL FLOW SPACE BOUNDARY
3806 C CONDITIONS
3807 C
3808 600 UMM=U(L,M,N1)+F2I*(U(L,M,N1)-UMP)
3809 VMM=V(L,M,N1)+F2I*(V(L,M,N1)-VMP)
3810 ULPMM=U(L+1,M,N1)+F2I*(U(L+1,M,N1)-U(L+1,M+1,N1))
3811 VLPMM=V(L+1,M,N1)+F2I*(V(L+1,M,N1)-V(L+1,M+1,N1))
3812 ULMM=U(L-1,M,N1)+F2I*(U(L-1,M,N1)-U(L-1,M+1,N1))
3813 VLMM=V(L-1,M,N1)+F2I*(V(L-1,M,N1)-V(L-1,M+1,N1))
3814 TMM=T+F2I*(T-TMP)
3815 TLPMM=TLP+F2I*(TLP-TLPMP)
3816 TLMM=TLM+F2I*(TLM-TLMP)
3817 IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 610
3818 ROMM=RO(L,M,N1)+F2I*(RO(L,M,N1)-ROMP)
3819 ROLPMM=RO(L+1,M,N1)+F2I*(RO(L+1,M,N1)-RO(L+1,M+1,N1))

```

```

3820      ROLMMM=RO(L-1,M,N1)+F2I*(RO(L-1,M,N1)-RO(L-1,M+1,N1))
3821      IF (ITM.LE.1) GO TO 610
3822      OMM=Q(L,M,N1)+F2I*(Q(L,M,N1)-OMP)
3823      QLPMM=Q(L+1,M,N1)+F2I*(Q(L+1,M,N1)-Q(L+1,M+1,N1))
3824      QLMMM=Q(L-1,M,N1)+F2I*(Q(L-1,M,N1)-Q(L-1,M+1,N1))
3825      IF (ITM.EQ.2) GO TO 610
3826      EMM=E(L,M,N1)+F2I*(E(L,M,N1)-EMP)
3827      ELPMM=E(L+1,M,N1)+F2I*(E(L+1,M,N1)-E(L+1,M+1,N1))
3828      ELMMM=E(L-1,M,N1)+F2I*(E(L-1,M,N1)-E(L-1,M+1,N1))
3829 C
3830 C      CALCULATE THE CENTERBODY OR UPPER DUAL FLOW SPACE BOUNDARY
3831 C      DERIVATIVES
3832 C
3833      610 IF (M.NE.MDFS) GO TO 630
3834      IF (L.NE.LDFS) GO TO 620
3835      ULPMM=U(L+1,M-1,N1)
3836      VLPMM=V(L+1,M-1,N1)
3837      TLPMM=P(L+1,M-1,N1)/(RO(L+1,M-1,N1)*RG)
3838      ROLPMM=RO(L+1,M-1,N1)
3839      IF (ITM.LE.1) GO TO 630
3840      QLPMM=Q(L+1,M-1,N1)
3841      ELPMM=E(L+1,M-1,N1)
3842      GO TO 630
3843      620 IF (L.NE.LDFS) GO TO 630
3844      ULMMM=U(L-1,M-1,N1)
3845      VLMMM=V(L-1,M-1,N1)
3846      TLMMM=P(L-1,M-1,N1)/(RO(L-1,M-1,N1)*RG)
3847      ROLMMM=RO(L-1,M-1,N1)
3848      IF (ITM.LE.1) GO TO 630
3849      QLMMM=Q(L-1,M-1,N1)
3850      ELMMM=E(L-1,M-1,N1)
3851      630 UY1=0.25*(UMP+U(L-1,M+1,N1)-UMM-ULMMM)*DYS
3852      VY1=0.25*(VMP+V(L-1,M+1,N1)-VMM-VLMMM)*DYS
3853      UY2=0.25*(UMP+U(L+1,M+1,N1)-UMM-ULPMM)*DYS
3854      VY2=0.25*(VMP+V(L+1,M+1,N1)-VMM-VLPMM)*DYS
3855      UY3=(U(L,M,N1)-UMM)*DYS
3856      VY3=(V(L,M,N1)-VMM)*DYS
3857      UX3=0.25*(U(L+1,M,N1)+ULPMM-U(L-1,M,N1)-ULMMM)*DYS
3858      VX3=0.25*(V(L+1,M,N1)+VLPMM-V(L-1,M,N1)-VLMMM)*DYS
3859      TY1=0.25*(TMP+TLMMP-TMM-TLMMM)*DYS
3860      TY2=0.25*(TMP+TLPMP-TMM-TLPMM)*DYS
3861      TX3=0.25*(TLP+TLPMM-TLM-TLMMM)*DYS
3862      TY3=(T-TMM)*DYS
3863      TMM=TMM
3864      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 850
3865      ROY1=0.25*(ROMP+RO(L-1,M+1,N1)-ROMM-ROLMMM)*DYS
3866      ROY2=0.25*(ROMP+RO(L+1,M+1,N1)-ROMM-ROLPMM)*DYS
3867      ROY3=(RO(L,M,N1)-ROMM)*DYS
3868      ROX3=0.25*(RO(L+1,M,N1)+ROLPMM-RO(L-1,M,N1)-ROLMMM)*DYS
3869      IF (CAV.EQ.0.0) GO TO 660
3870      IF (NDIM.EQ.0) GO TO 650
3871      IF (M.EQ.1.AND.YCB(L).EQ.0.0) GO TO 640
3872      ATERM=V(L,M,N1)*RYP
3873      ATERM1=0.5*(V(L,M,N1)+V(L-1,M,N1))*RYP
3874      ATERM2=0.5*(V(L,M,N1)+V(L+1,M,N1))*RYP
3875      ATERM3=0.5*(V(L,M,N1)+VMM)*RYP
3876      ATERM4=0.5*(V(L,M,N1)+VMP)*RYP
3877      IF (M.EQ.MDFS) GO TO 650
3878      IF (YCB(L-1).EQ.0.0) ATERM1=0.5*(BE4+V(L-1,M+1,N1)*DYS+V(L,M,N1)
3879      1/YCB(L))
3880      IF (YCB(L+1).EQ.0.0) ATERM2=0.5*(BE4+V(L+1,M+1,N1)*DYS+V(L,M,N1)
3881      1/YCB(L))
3882      IF (YCB(L-1).EQ.0.0.OR.YCB(L+1).EQ.0.0) ATERM=0.5*(ATERM1+ATERM2)
3883      GO TO 650
3884      640 ATERM=BE4*VMP*DYS
3885      ATERM1=BE4*0.5*(VMP+V(L-1,M+1,N1))*DYS
3886      ATERM2=BE4*0.5*(VMP+V(L+1,M+1,N1))*DYS
3887      ATERM3=BE4*VMP*DYS
3888      ATERM4=ATERM4
3889      IF (YCB(L-1).NE.0.0) ATERM1=0.5*(V(L-1,M,N1)*YCB(L-1)+BE4*VMP*DYS)
3890      IF (YCB(L+1).NE.0.0) ATERM2=0.5*(V(L+1,M,N1)*YCB(L+1)+BE4*VMP*DYS)
3891      IF (YCB(L-1).NE.0.0.OR.YCB(L+1).NE.0.0) ATERM=0.5*(ATERM1+ATERM2)
3892      650 IF (ISS.EQ.0) GO TO 660
3893      AMP=SQRT(GRG*TMP)

```

```

3894      AMM=SQRT(GRG*TMM)
3895      AY3=(A-AMM)*DYR
3896      AY4=(AMP-A)*DYR
3897      660 IF (ITM.LE.1) GO TO 850
3898      ROQY=0.5*(ROMP*QMP-ROMM*QMM)*DYR
3899      QY1=0.25*(QMP+Q(L-1,M+1,N1)-QMM-QLMMM)*DYR
3900      QY2=0.25*(QMP+Q(L+1,M+1,N1)-QMM-QLPMM)*DYR
3901      QY3=(Q(L,M,N1)-QMM)*DYR
3902      QX3=0.25*(Q(L+1,M,N1)+OLPMM-Q(L-1,M,N1)-QLMMM)*DXR
3903      Q2Y=0.5*(SQRT(ABS(QMP))-SQRT(ABS(QMM)))*DYR
3904      IF (ITM.EQ.3) GO TO 670
3905      ROSQ3=ROMM*SQRT(ABS(QMM))
3906      ROSQ4=ROMP*SQRT(ABS(QMP))
3907      GO TO 850
3908      670 EY1=0.25*(EMP+E(L-1,M+1,N1)-EMM-ELMMM)*DYR
3909      EY2=0.25*(EMP+E(L+1,M+1,N1)-EMM-ELPMM)*DYR
3910      EY3=(E(L,M,N1)-EMM)*DYR
3911      EX3=0.25*(E(L+1,M,N1)+ELPMM-E(L-1,M,N1)-ELMMM)*DXR
3912      MUT3=COMU*ROMM*QMM*QMM*LC/ABS(EMM)
3913      MUT4=COMU*ROMP*QMP*QMP*LC/ABS(EMP)
3914      IF (M.EQ.1.AND.NGCB.EQ.O) MUT=0.5*(MUT3+MUT4)
3915      GO TO 850
3916 C
3917 C      BEGIN THE WALL OR LOWER DUAL FLOW SPACE BOUNDARY POINT
3918 C      Y DERIVATIVE CALCULATION
3919 C
3920      680 DYP=DY/BE3
3921      UMM=U(L,M-1,N1)
3922      VMM=V(L,M-1,N1)
3923      PMM=P(L,M-1,N1)
3924      ROMM=RO(L,M-1,N1)
3925      QMM=Q(L,M-1,N1)
3926      EMM=E(L,M-1,N1)
3927      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
3928      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
3929      UY3=(U(L,M,N1)-UMM)*DYR
3930      VY3=(V(L,M,N1)-VMM)*DYR
3931      TLPMM=P(L+1,M-1,N1)/(RO(L+1,M-1,N1)*RG)
3932      TMM=PMM/(ROMM*RG)
3933      TLMMM=P(L-1,M-1,N1)/(RO(L-1,M-1,N1)*RG)
3934      TX3=0.25*(TLP+TLPMM-TLM-TLMMM)*DXR
3935      TY3=(T-TMM)*DYR
3936      IF (ITM.EQ.O.AND.CAV.EQ.O.O) GO TO 690
3937      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))
3938      1)*DXR
3939      ROY3=(RO(L,M,N1)-ROMM)*DYR
3940      IF (ITM.LE.1) GO TO 690
3941      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
3942      QY3=(Q(L,M,N1)-QMM)*DYR
3943      IF (ITM.EQ.2) GO TO 690
3944      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
3945      EY3=(E(L,M,N1)-EMM)*DYR
3946 C
3947 C      REFLECT THE WALL OR LOWER DUAL FLOW SPACE BOUNDARY CONDITIONS
3948 C
3949      690 IF (IVBC.NE.O) GO TO 780
3950      IF (IWALL.EQ.1.AND.M.EQ.MMAX) GO TO 780
3951      IF (M.EQ.MDFS) GO TO 700
3952      DNXY=NXY(L)
3953      DNXYP=NXY(L+1)
3954      DNXYM=NXY(L-1)
3955      GO TO 710
3956      700 DNXY=NXY(L)
3957      DNXYP=NXY(L+1)
3958      DNXYM=NXY(L-1)
3959      710 THEW=ATAN(-DNXY)
3960      IF (UMM.EQ.O.O) GO TO 720
3961      THE=ATAN(VMM/UMM)
3962      GO TO 730
3963      720 THE=O.O
3964      730 IF (UMM.LT.O.O) THE=THE+3.14159
3965      VMAG=SQRT(UMM*UMM+VMM*VMM)
3966      RTHE=2.O*THEW-THE

```



```

3967     IF (NOSLIP.EQ.1) RTHE=3.14159+THE
3968     UMP=VMAG*COS(RTHE)
3969     VMP=VMAG*SIN(RTHE)
3970     THEW=ATAN(-DNXNYP)
3971     IF (U(L+1,M-1,N1).EQ.0.0) GO TO 740
3972     THE=ATAN(V(L+1,M-1,N1)/U(L+1,M-1,N1))
3973     GO TO 750
3974 740 THE=0.0
3975 750 IF (U(L+1,M-1,N1).LT.0.0) THE=THE+3.14159
3976     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))
3977     RTHE=2.0*THEW-THE
3978     IF (NOSLIP.EQ.1) RTHE=3.14159+THE
3979     ULPMP=VMAG*COS(RTHE)
3980     VLPMP=VMAG*SIN(RTHE)
3981     THEW=ATAN(-DNXNYM)
3982     IF (U(L-1,M-1,N1).EQ.0.0) GO TO 760
3983     THE=ATAN(V(L-1,M-1,N1)/U(L-1,M-1,N1))
3984     GO TO 770
3985 760 THE=0.0
3986 770 IF (U(L-1,M-1,N1).LT.0.0) THE=THE+3.14159
3987     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))
3988     RTHE=2.0*THEW-THE
3989     IF (NOSLIP.EQ.1) RTHE=3.14159+THE
3990     ULMMP=VMAG*COS(RTHE)
3991     VLMMP=VMAG*SIN(RTHE)
3992 C
3993     RFL=2.0*DNXNY*DYP/(1.0+DNXNY*DNXNY)
3994     RFLP=2.0*DNXNYP*DYP/(1.0+DNXNYP*DNXNYP)
3995     RFLM=2.0*DNXNYM*DYP/(1.0+DNXNYM*DNXNYM)
3996     TTERM=0.5*(OM1*TX1+OM2*TX2)
3997     TMP=TMM-TTERM*RFL
3998     TLPMP=TLPMM-TTERM*RFLP
3999     TLMMP=TLMMM-TTERM*RFLM
4000     IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 790
4001     ROTERM=0.5*(OM1*ROX1+OM2*ROX2)
4002     ROMP=ROMM-ROTERM*RFL
4003     ROLPMP=RO(L+1,M-1,N1)-ROTERM*RFLP
4004     ROLMMP=RO(L-1,M-1,N1)-ROTERM*RFLM
4005     IF (ITM.LE.1) GO TO 790
4006     QTERM=0.5*(QM1*QX1+QM2*QX2)
4007     QMP=QMM-QTERM*RFL
4008     QLPMP=Q(L+1,M-1,N1)-QTERM*RFLP
4009     QLMMP=Q(L-1,M-1,N1)-QTERM*RFLM
4010     IF (ITM.EQ.2) GO TO 790
4011     ETERM=0.5*(EM1*EX1+EM2*EX2)
4012     EMP=E(MM)-ETERM*RFL
4013     ELPMP=E(L+1,M-1,N1)-ETERM*RFLP
4014     ELMMP=E(L-1,M-1,N1)-ETERM*RFLM
4015     GO TO 790
4016 C
4017 C     EXTRAPOLATE THE WALL OR LOWER DUAL FLOW SPACE BOUNDARY COONITIONS
4018 C
4019 780 UMP=U(L,M,N1)+F2I*(U(L,M,N1)-UMM)
4020     VMP=V(L,M,N1)+F2I*(V(L,M,N1)-VMM)
4021     ULPMP=U(L+1,M,N1)+F2I*(U(L+1,M,N1)-U(L+1,M-1,N1))
4022     VLPMP=V(L+1,M,N1)+F2I*(V(L+1,M,N1)-V(L+1,M-1,N1))
4023     ULMMP=U(L-1,M,N1)+F2I*(U(L-1,M,N1)-U(L-1,M-1,N1))
4024     VLMMP=V(L-1,M,N1)+F2I*(V(L-1,M,N1)-V(L-1,M-1,N1))
4025     TMP=T+F2I*(T-TMM)
4026     TLPMP=TLP+F2I*(TLP-TLPMM)
4027     TLMMP=TLM+F2I*(TLM-TLMMM)
4028     IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 790
4029     ROMP=RO(L,M,N1)+F2I*(RO(L,M,N1)-ROMM)
4030     ROLPMP=RO(L+1,M,N1)+F2I*(RO(L+1,M,N1)-RO(L+1,M-1,N1))
4031     ROLMMP=RO(L-1,M,N1)+F2I*(RO(L-1,M,N1)-RO(L-1,M-1,N1))
4032     IF (ITM.LE.1) GO TO 790
4033     QMP=Q(L,M,N1)+F2I*(Q(L,M,N1)-QMM)
4034     QLPMP=Q(L+1,M,N1)+F2I*(Q(L+1,M,N1)-Q(L+1,M-1,N1))
4035     QLMMP=Q(L-1,M,N1)+F2I*(Q(L-1,M,N1)-Q(L-1,M-1,N1))
4036     IF (ITM.EQ.2) GO TO 790
4037     EMP=E(L,M,N1)+F2I*(E(L,M,N1)-EMM)
4038     ELPMP=E(L+1,M,N1)+F2I*(E(L+1,M,N1)-E(L+1,M-1,N1))

```

```

4039      ELMMP=E(L-1,M,N1)+F2I*(E(L-1,M,N1)-E(L-1,M-1,N1))
4040 C
4041 C      CALCULATE THE WALL AND LOWER DUAL FLOW SPACE BOUNDARY DERIVATIVES
4042 C
4043      790 IF (M.NE.MDFS) GO TO 810
4044      IF (L.NE.LDFS) GO TO 800
4045      ULPMP=U(L+1,M+1,N1)
4046      VLPMP=V(L+1,M+1,N1)
4047      TLPMP=P(L+1,M+1,N1)/(RO(L+1,M+1,N1)*RG)
4048      ROLPMP=RO(L+1,M+1,N1)
4049      IF (ITM.LE.1) GO TO 810
4050      QLPMP=Q(L+1,M+1,N1)
4051      ELPMP=E(L+1,M+1,N1)
4052      GO TO 810
4053      800 IF (L.NE.LDFS) GO TO 810
4054      ULMMP=U(L-1,M+1,N1)
4055      VLMMP=V(L-1,M+1,N1)
4056      TLMMP=P(L-1,M+1,N1)/(RO(L-1,M+1,N1)*RG)
4057      ROLMMP=RO(L-1,M+1,N1)
4058      IF (ITM.LE.1) GO TO 810
4059      QLMMP=Q(L-1,M+1,N1)
4060      ELMMP=E(L-1,M+1,N1)
4061      810 UY1=0.25*(UMP+ULMMP-UMM-U(L-1,M-1,N1))*DYZ
4062      VY1=0.25*(VMP+VLMMP-VMM-V(L-1,M-1,N1))*DYZ
4063      UY2=0.25*(UMP+ULPMP-UMM-U(L+1,M-1,N1))*DYZ
4064      VY2=0.25*(VMP+VLPMP-VMM-V(L+1,M-1,N1))*DYZ
4065      UY4=(UMP-U(L,M,N1))*DYZ
4066      VY4=(VMP-V(L,M,N1))*DYZ
4067      UX4=0.25*(U(L+1,M,N1)+ULPMP-U(L-1,M,N1)-ULMMP)*DXR
4068      VX4=0.25*(V(L+1,M,N1)+VLPMP-V(L-1,M,N1)-VLMMP)*DXR
4069      TY1=0.25*(TMP+TLMMP-TMM-TLMM)*DYZ
4070      TY2=0.25*(TMP+TLPMP-TMM-TLPM)*DYZ
4071      TX4=0.25*(TLP+TLPMP-TLM-TLMMP)*DXR
4072      TY4=(TMP-T)*DYZ
4073      TMP=TMP
4074      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 850
4075      ROY1=0.25*(ROMP+ROLMMP-ROMM-RO(L-1,M-1,N1))*DYZ
4076      ROY2=0.25*(ROMP+ROLPMP-ROMM-RO(L+1,M-1,N1))*DYZ
4077      RCX4=0.25*(RO(L+1,M,N1)+ROLPMP-RO(L-1,M,N1)-ROLMMP)*DXR
4078      ROY4=(ROMP-RO(L,M,N1))*DYZ
4079      IF (CAV.EQ.0.0) GO TO 830
4080      IF (NDIM.EQ.0) GO TO 820
4081      ATERM=V(L,M,N1)*RYP
4082      ATERM1=0.5*(V(L,M,N1)+V(L-1,M,N1))*RYP
4083      ATERM2=0.5*(V(L,M,N1)+V(L+1,M,N1))*RYP
4084      ATERM3=0.5*(V(L,M,N1)+VMM)*RYP
4085      ATERM4=0.5*(V(L,M,N1)+VMP)*RYP
4086      820 IF (ISS.EQ.0) GO TO 830
4087      AMP=SQRT(GRG*TMP)
4088      AMM=SQRT(GRG*TMM)
4089      AY3=(A-AMM)*DYZ
4090      AY4=(AMP-A)*DYZ
4091      830 IF (ITM.LE.1) GO TO 850
4092      ROQY=0.5*(ROMP+QMP-ROMM+QMM)*DYZ
4093      QY1=0.25*(QMP+QLMMP-QMM-Q(L-1,M-1,N1))*DYZ
4094      QY2=0.25*(QMP+QLPMP-QMM-Q(L+1,M-1,N1))*DYZ
4095      QX4=0.25*(Q(L+1,M,N1)+QLPMP-Q(L-1,M,N1)-QLMMP)*DXR
4096      QY4=(QMP-Q(L,M,N1))*DYZ
4097      QY2=0.5*(SQRT(ABS(QMP))-SQRT(ABS(QMM)))*DYZ
4098      IF (ITM.EQ.3) GO TO 840
4099      ROSQ3=ROMM*SQRT(ABS(QMM))
4100      ROSQ4=ROMP*SQRT(ABS(QMP))
4101      GO TO 850
4102      840 EY1=0.25*(EMP+ELMMP-EMM-E(L-1,M-1,N1))*DYZ
4103      EY2=0.25*(EMP+ELPMP-EMM-E(L+1,M-1,N1))*DYZ
4104      EX4=0.25*(E(L+1,M,N1)+ELPMP-E(L-1,M,N1)-ELMMP)*DXR
4105      EY4=(EMP-E(L,M,N1))*DYZ
4106      MUT3=COMU+ROMM+QMM+QMM*LC/ABS(EMM)
4107      MUT4=COMU+ROMP+QMP+QMP*LC/ABS(EMP)
4108 C
4109 C      COMBINE TERMS
4110 C

```

4111 850 UXY1=OM1\*UX1+AL\*UY1  
 4112 UXY2=OM2\*UX2+AL\*UY2  
 4113 UXY3=OM\*UX3+AL3\*UY3  
 4114 UXY4=OM\*UX4+AL4\*UY4  
 4115 UXY12=O.5\*(OM1\*UX1+OM2\*UX2+AL3\*UY3+AL4\*UY4)  
 4116 VXY1=OM1\*VX1+AL\*VY1  
 4117 VXY2=OM2\*VX2+AL\*VY2  
 4118 VXY3=OM\*VX3+AL3\*VY3  
 4119 VXY4=OM\*VX4+AL4\*VY4  
 4120 VXY12=O.5\*(OM1\*VX1+OM2\*VX2+AL3\*VY3+AL4\*VY4)  
 4121 BUY1=BE\*UY1  
 4122 BUY2=BE\*UY2  
 4123 BUY3=BE3\*UY3  
 4124 BUY4=BE4\*UY4  
 4125 BUY34=O.5\*(BE3\*UY3+BE4\*UY4)  
 4126 BVY1=BE\*VY1  
 4127 BVY2=BE\*VY2  
 4128 BVY3=BE3\*VY3  
 4129 BVY4=BE4\*VY4  
 4130 BVY34=O.5\*(BE3\*VY3+BE4\*VY4)  
 4131 TXY1=OM1\*TX1+AL\*TY1  
 4132 TXY2=OM2\*TX2+AL\*TY2  
 4133 TXY3=OM\*TX3+AL3\*TY3  
 4134 TXY4=OM\*TX4+AL4\*TY4  
 4135 BTY3=BE3\*TY3  
 4136 BTY4=BE4\*TY4  
 4137 IF (ITM.EQ.O.AND.CAV.EQ.O.O) GO TO 940  
 4138 ROXY1=OM1\*ROX1+AL\*ROY1  
 4139 ROXY2=OM2\*ROX2+AL\*ROY2  
 4140 ROXY3=OM\*ROX3+AL3\*ROY3  
 4141 ROXY4=OM\*ROX4+AL4\*ROY4  
 4142 ROXY12=O.5\*(OM1\*ROX1+OM2\*ROX2+AL3\*ROY3+AL4\*ROY4)  
 4143 BROY1=BE\*ROY1  
 4144 BROY2=BE\*ROY2  
 4145 BROY3=BE3\*ROY3  
 4146 BROY4=BE4\*ROY4  
 4147 BROY34=O.5\*(BE3\*ROY3+BE4\*ROY4)  
 4148 IF (ISS.EQ.O) GO TO 860  
 4149 AXY1=OM1\*AX1+O.5\*AL\*(AY3+AY4)  
 4150 AXY2=OM2\*AX2+O.5\*AL\*(AY3+AY4)  
 4151 AXY12=O.5\*(AXY1+AXY2)  
 4152 BAY3=BE3\*AY3  
 4153 BAY4=BE4\*AY4  
 4154 BAY34=O.5\*(BAY3+BAY4)  
 4155 860 IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 870  
 4156 IF (ITM.LE.1) GO TO 870  
 4157 QXY1=OM1\*QX1+AL\*QY1  
 4158 QXY2=OM2\*QX2+AL\*QY2  
 4159 QXY3=OM\*QX3+AL3\*QY3  
 4160 QXY4=OM\*QX4+AL4\*QY4  
 4161 BQY3=BE3\*QY3  
 4162 BQY4=BE4\*QY4  
 4163 BQY34=O.5\*(BE3\*QY3+BE4\*QY4)  
 4164 Q2XY=OM\*Q2X+AL\*Q2Y  
 4165 BQ2Y=BE\*Q2Y  
 4166 IF (ITM.EQ.2) GO TO 870  
 4167 EXY1=OM1\*EX1+AL\*EY1  
 4168 EXY2=OM2\*EX2+AL\*EY2  
 4169 EXY3=OM\*EX3+AL3\*EY3  
 4170 EXY4=OM\*EX4+AL4\*EY4  
 4171 BEY3=BE3\*EY3  
 4172 BEY4=BE4\*EY4  
 4173 BEY34=O.5\*(BE3\*EY3+BE4\*EY4)  
 4174 C  
 4175 C CALCULATE THE ARTIFICIAL VISCOSITY COEFFICIENTS  
 4176 C  
 4177 870 IF (CAV.EQ.O.O) GO TO 940  
 4178 IF (L.LT.LSS) GO TO 880  
 4179 IF (L.GT.LSF) GO TO 880  
 4180 IF (M.LT.MSS) GO TO 880  
 4181 IF (M.GT.MSF) GO TO 880  
 4182 IF (SMACH.EQ.O.O) GO TO 890

```

4183     IF (XM.LT.SMACH*SMACH) GO TO 680
4184     GO TO 890
4185 880  DIV1=0.0
4186     DIV2=0.0
4187     DIV3=0.0
4188     DIV4=0.0
4189     GO TO 910
4190 890  DIV1=UXY1+BVY1+ATERM1
4191     DIV2=UXY2+BVY2+ATERM2
4192     DIV3=UXY3+BVY3+ATERM3
4193     DIV4=UXY4+BVY4+ATERM4
4194     IF (IDIVC.NE 0) GO TO 910
4195     IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 900
4196     IF (DIV1.GT.0.0) DIV1=0.0
4197     IF (DIV2.GT.0.0) DIV2=0.0
4198 900  IF (DIV3.GT.0.0) DIV3=0.0
4199     IF (DIV4.GT.0.0) DIV4=0.0
4200 910  IF (ISS.EQ.0) GO TO 930
4201     IF (ISS.EQ.1) GO TO 920
4202     DIV1=ABS(DIV1)+ABS(AXY1+BAY34)
4203     DIV2=ABS(DIV2)+ABS(AXY2+BAY34)
4204     DIV3=ABS(DIV3)+ABS(AXY12+BAY3)
4205     DIV4=ABS(DIV4)+ABS(AXY12+BAY4)
4206     GO TO 930
4207 920  IF (DIV1.NE.0.0) DIV1=ABS(DIV1)+ABS(AXY1+BAY34)
4208     IF (DIV2.NE.0.0) DIV2=ABS(DIV2)+ABS(AXY2+BAY34)
4209     IF (DIV3.NE.0.0) DIV3=ABS(DIV3)+ABS(AXY12+BAY3)
4210     IF (DIV4.NE.0.0) DIV4=ABS(DIV4)+ABS(AXY12+BAY4)
4211 930  DRLA=XLA*CAV*2.0*RO(L,M,N1)*DXP*DYF
4212     RLA1=DRLA*ABS(DIV1)
4213     RLA2=DRLA*ABS(DIV2)
4214     RLA3=DRLA*ABS(DIV3)*SMT
4215     RLA4=DRLA*ABS(DIV4)*SMT
4216     RLA=0.25*(RLA1+RLA2+RLA3+RLA4)
4217     XMULA=XMU/XLA
4218     RMU1=XMULA*RLA1
4219     RMU2=XMULA*RLA2
4220     RMU3=XMULA*RLA3
4221     RMU4=XMULA*RLA4
4222     RMU=0.25*(RMU1+RMU2+RMU3+RMU4)
4223     RK1=DRK*RMU1
4224     RK2=DRK*RMU2
4225     RK3=DRK*RMU3
4226     RK4=DRK*RMU4
4227     RK=0.25*(RK1+RK2+RK3+RK4)
4228     RRO1=XRO*RMU1
4229     RRO2=XRO*RMU2
4230     RRO3=XRO*RMU3
4231     RRO4=XRO*RMU4
4232     RRO=0.25*(RRO1+RRO2+RRO3+RRO4)
4233     RLP2M=RLA+2.0*RMU
4234     RLP2M1=RLA1+2.0*RMU1
4235     RLP2M2=RLA2+2.0*RMU2
4236     RLP2M3=RLA3+2.0*RMU3
4237     RLP2M4=RLA4+2.0*RMU4
4238     RLPM=RLA+RMU
4239 C
4240 C     CALCULATE THE MOLECULAR VISCOSITY COEFFICIENTS
4241 C
4242 940  IF (CHECK.EQ.0.0) GO TO 1190
4243     TCHECK=T+TLP+TLM+TMF+TMW
4244     IF (TCHECK.GT.0.0) GO TO 950
4245     NP=N+NSTART
4246     WRITE (6,1510) NP,L,M,NVC
4247     IERR=1
4248     RETURN
4249 950  IF (ECHECK.EQ.0.0) GO TO 960
4250     IF (ECHECK.LT.0.0) GO TO 970
4251     MU=CMU*T**EMU
4252     LA=CLA*T**ELA
4253     K=CK*T**EK
4254     MU1=(CMU*TLM**EMU+MU)*0.5

```

```

4255      MU2=(CMU*TLP**EMU+MU)*0.5
4256      MU3=(CMU*TMM**EMU+MU)*0.5
4257      MU4=(CMU*TMP**EMU+MU)*0.5
4258      LA1=(CLA*TLM**ELA+LA)*0.5
4259      LA2=(CLA*TLP**ELA+LA)*0.5
4260      LA3=(CLA*TMM**ELA+LA)*0.5
4261      LA4=(CLA*TMP**ELA+LA)*0.5
4262      K1=(CK*TLM**EK+K)*0.5
4263      K2=(CK*TLP**EK+K)*0.5
4264      K3=(CK*TMM**EK+K)*0.5
4265      K4=(CK*TMP**EK+K)*0.5
4266      GO TO 980
4267 C
4268      960 MU=CMU
4269      MU1=CMU
4270      MU2=CMU
4271      MU3=CMU
4272      MU4=CMU
4273      LA=CLA
4274      LA1=CLA
4275      LA2=CLA
4276      LA3=CLA
4277      LA4=CLA
4278      K=CK
4279      K1=CK
4280      K2=CK
4281      K3=CK
4282      K4=CK
4283      GO TO 980
4284 C
4285      970 SQT=T**EMU
4286      MU=CMU*SQT
4287      LA=CLA*SQT
4288      K=CK*SQT
4289      SQTLM=(TLM**EMU+SQT)*0.5
4290      SQTLP=(TLP**EMU+SQT)*0.5
4291      SQTMM=(TMM**EMU+SQT)*0.5
4292      SQTMP=(TMP**EMU+SQT)*0.5
4293      MU1=CMU*SQTLM
4294      MU2=CMU*SQTLP
4295      MU3=CMU*SQTMM
4296      MU4=CMU*SQTMP
4297      LA1=CLA*SQTLM
4298      LA2=CLA*SQTLP
4299      LA3=CLA*SQTMM
4300      LA4=CLA*SQTMP
4301      K1=CK*SQTLM
4302      K2=CK*SQTLP
4303      K3=CK*SQTMM
4304      K4=CK*SQTMP
4305      980 LP2M=LA+2.0*MU
4306      LP2M1=LA1+2.0*MU1
4307      LP2M2=LA2+2.0*MU2
4308      LP2M3=LA3+2.0*MU3
4309      LP2M4=LA4+2.0*MU4
4310      LPN=LA+MU
4311      AVMUR=RMU/MU
4312      IF (RLA.GT.0.0) AVMUR=RLA/MU
4313 C
4314 C      CALCULATE THE TURBULENT VISCOSITY COEFFICIENTS
4315 C
4316      IF (ITM.EQ.0) GO TO 1190
4317      IF (ITM.EQ.3) GO TO 1160
4318      IF (IMLM.EQ.2) GO TO 1C10
4319 C
4320      DELTAY=YSL2-YSL1
4321      IF (IWP.NE.0) GO TO 990
4322      IF (M.LT.MMIN) DELTAY=YMIN-YSL1
4323      IF (M.GT.MMIN) DELTAY=YSL2-YMIN
4324      IF (M.NE.MMIN) GO TO 990
4325      DELTAY=0.5*(YSL2-YSL1)
4326      DELTAYC=YMIN-YSL1
4327      DELTAY4=YSL2-YMIN

```

```

4328 990 TML=CML1+ABS(DELTAY)
4329 IF (IMP.EQ.O) TML=CML2+ABS(DELTAY)
4330 IF (ITM.EQ.2) GO TO 1140
4331 TML3=TML
4332 TML4=TML
4333 IF (IMP.NE.O) GO TO 1080
4334 IF (M.EQ.MMIN-1.OR.M.EQ.MMIN+1) GO TO 1000
4335 IF (M.NE.MMIN) GO TO 1080
4336 TML3=CML2*DELTAY3
4337 TML4=CML2*DELTAY4
4338 1000 IF (L.NE.LDFSS-1.AND.L.NE.LDFSS+1) GO TO 1080
4339 TML=O.5*TML
4340 TML3=O.5*TML3
4341 TML4=O.5*TML4
4342 GO TO 1080
4343 C
4344 1010 YWB=YCB(L)
4345 YWT=YW(L)
4346 IF (MDFS.EQ.O) GO TO 1030
4347 IF (IB.EQ.4.OR.M.GT.MDFS) GO TO 1020
4348 YWT=YL(L)
4349 TAUW=TAUWM
4350 GO TO 1050
4351 1020 YWB=YU(L)
4352 TAUW=TAUWP
4353 GO TO 1040
4354 1030 IF (NGCB.EQ.O) GO TO 1050
4355 1040 YPD=YP-YWB
4356 YPD3=YP3-YWB
4357 YPD4=YP4-YWB
4358 GO TO 1060
4359 1050 YPD=YWT-YP
4360 YPD3=YWT-YP3
4361 YPD4=YWT-YP4
4362 1060 IF (YPD3.LT.O.O) YPD3=YPD4
4363 IF (YPD4.LT.O.O) YPD4=YPD3
4364 YDUM=SQRT(RO(L,M,N1)*MU*TAUW)/(26.O*MU)
4365 YPLUS=YPD+YDUM
4366 YPLUS3=YPD3+YDUM
4367 YPLUS4=YPD4+YDUM
4368 TML=O.4*YPD*(1.O-EXP(-YPLUS))
4369 TML3=O.4*YPD3*(1.C-EXP(-YPLUS3))
4370 TML4=O.4*YPD4*(1.O-EXP(-YPLUS4))
4371 IF (DEL.EQ.O.O) GO TO 1070
4372 YTERMD=O.O168*ABS(UBLE)*DELS*RO(L,M,N1)
4373 RDEL=1.O/DEL
4374 MUTD=YTERMD/(1.O+5.5*(YPD*RDEL))*6)
4375 TMLD=C.O
4376 IF (BUY34.EQ.O.O.AND.VXY12.EQ.O.O) GO TO 1120
4377 TMLD=SQRT(MUTD/(RO(L,M,N1)*SQRT(BUY34*BUY34+VXY12*VXY12)))
4378 GO TO 1080
4379 1070 TMLD=O.O
4380 MUTD=O.O
4381 GO TO 1120
4382 C
4383 1080 MUT=TML*TML*RO(L,M,N1)*SQRT(BUY34*BUY34+VXY12*VXY12)
4384 IF (IMLM.EQ.2.AND.MUTD.LT.MUT) GO TO 1120
4385 IF (ITM.EQ.2) GO TO 1140
4386 MUT1=TML*TML*RO(L,M,N1)*SQRT(BUY1*BUY1+VXY1*VXY1)
4387 MUT2=TML*TML*RO(L,M,N1)*SQRT(BUY2*BUY2+VXY2*VXY2)
4388 IF (MDFS.EQ.O) GO TO 1090
4389 IF (L.EQ.LDFSS) MUT1=MUT
4390 IF (L.EQ.LDFSS) MUT2=MUT
4391 IF (M.GE.MMIN-1.AND.M.LE.MMIN+1) GO TO 1090
4392 IF (L.EQ.LDFSS-1) MUT2=MUT
4393 IF (L.EQ.LDFSS+1) MUT1=MUT
4394 1090 IF (NOSLIP.EQ.O) GO TO 1110
4395 IF (M.EQ.1.AND.NGCB.NE.O) GO TO 1100
4396 IF (M.EQ.MMAX.AND.IWALL.EQ.O) GO TO 1100
4397 IF (M.EQ.MDFS.AND.LDFS.NE.O) GO TO 1100
4398 GO TO 1110
4399 1100 MUT=O.O

```

```

4400      MUT1=0.0
4401      MUT2=0.0
4402  1110 MUT3=TML3*TML3*RO(L,M,N1)*SQRT(BUY3*BUY3+VXY3*VXY3)
4403      MUT4=TML4*TML4*RO(L,M,N1)*SQRT(BUY4*BUY4+VXY4*VXY4)
4404      IF (IMLM.EC.1) GO TO 1130
4405      GO TO 1170
4406  1120 TML=TMLD
4407      IF (ITM.EQ.2) GO TO 1140
4408      MUT=MUTD
4409      MUT1=MUTD
4410      MUT2=MUTD
4411      MUT3=MUTD
4412      MUT4=MUTD
4413      GO TO 1170
4414  1130 IF (M.NE.1.OR.NGCB.NE.0) GO TO 1170
4415      MUT=TML*TML*TML*BE*ABS(BUY4-BUY3)*DYR*RO(L,M,N1)
4416      MUT1=MUT
4417      MUT2=MUT
4418      GO TO 1170
4419  C
4420  1140 TML=COL*TML
4421      TINT=TML*RO(L,M,N1)*SQRT(2.0*Q(L,M,N1))/MU
4422      DELTA=5.0
4423      IF (TINT.GT.5.0) GO TO 1150
4424      DCOMU=COMU*0.1*TML*TML/MU
4425      MUT=DCOMU*ROSO*ROSO
4426      MUT1=DCOMU*ROSO1*ROSO1
4427      MUT2=DCOMU*ROSO2*ROSO2
4428      MUT3=DCOMU*ROSO3*ROSO3
4429      MUT4=DCOMU*ROSO4*ROSO4
4430      GO TO 1160
4431  1150 DELTA=TINT
4432      DCOMU=COMU*0.3534*TML
4433      MUT=DCOMU*ROSO
4434      MUT1=DCOMU*ROSO1
4435      MUT2=DCOMU*ROSO2
4436      MUT3=DCOMU*ROSO3
4437      MUT4=DCOMU*ROSO4
4438  C
4439  1160 MUT1=0.5*(MUT+MUT1)
4440      MUT2=0.5*(MUT+MUT2)
4441      MUT3=0.5*(MUT+MUT3)
4442      MUT4=0.5*(MUT+MUT4)
4443      IF (ITM.EQ.2) GO TO 1170
4444  C
4445      RET=RO(L,M,N1)*Q(L,M,N1)*Q(L,M,N1)*LC/(MU*F(L,M,N1))
4446      RET1=0.5*(RET+ROLM*QLM*QLM*LC/(MU1*ELM))
4447      RET2=0.5*(RET+ROLP*QLP*QLP*LC/(MU2*ELP))
4448      RET3=0.5*(RET+ROMM*QMM*QMM*LC/(MU3*ABS(EMM)))
4449      RET4=0.5*(RET+ROMP*QMP*QMP*LC/(MU4*ABS(EMP)))
4450      FU=EXP(-3.4/(1.0+0.02*RET)**2)
4451      FU1=EXP(-3.4/(1.0+0.02*RET1)**2)
4452      FU2=EXP(-3.4/(1.0+0.02*RET2)**2)
4453      FU3=EXP(-3.4/(1.0+0.02*RET3)**2)
4454      FU4=EXP(-3.4/(1.0+0.02*RET4)**2)
4455      MUT=FU*MUT
4456      MUT1=FU1*MUT1
4457      MUT2=FU2*MUT2
4458      MUT3=FU3*MUT3
4459      MUT4=FU4*MUT4
4460      C2=C2*(1.0-0.2222*EXP(-0.0278*RET*RET))
4461  C
4462  1170 MUT=0.25*(MUT1+MUT2+MUT3+MUT4)
4463      IF (MUT1.EQ.0.0.AND.MUT2.EQ.0.0) MUT=0.0
4464      TLMJR=MUT/MU
4465      LAT1=LA1*MUT1/MU1
4466      LAT2=LA2*MUT2/MU2
4467      LAT3=LA3*MUT3/MU3
4468      LAT4=LA4*MUT4/MU4
4469      LAT=0.25*(LAT1+LAT2+LAT3+LAT4)
4470      IF (MUT.EQ.0.0) LAT=0.0
4471      KTI=TRK*MUT1

```

```

4472      KT2=TRK*MUT2
4473      KT3=TRK*MUT3
4474      KT4=TRK*MUT4
4475      KT=0.25*(KT1+KT2+KT3+KT4)
4476      IF (MUT.EQ.O.O) KT=0.0
4477      LP2MT=LAT+2.O*MUT
4478      LP2MT1=LAT1+2.O*MUT1
4479      LP2MT2=LAT2+2.O*MUT2
4480      LP2MT3=LAT3+2.O*MUT3
4481      LP2MT4=LAT4+2.O*MUT4
4482      LPMT=LAT*MUT
4483      IF (ITM.NE.1) GO TO 1180
4484      QLP2MT=LP2MT
4485      DMUT=MUT
4486      DLAT=LAT
4487      SMU1=MUT1*RORR
4488      SMU2=MUT2*RORR
4489      SMU3=MUT3*RORR
4490      SMU4=MUT4*RORR
4491      GO TO 1190
4492  1180 SMU1=MUT1*2.O/(RO(L.M.N1)+ROLM)
4493      SMU2=MUT2*2.O/(RO(L.M.N1)+ROLP)
4494      SMU3=MUT3*2.O/(RO(L.M.N1)+ROMM)
4495      SMU4=MUT4*2.O/(RO(L.M.N1)+ROMP)
4496  C
4497  C      DETERMINE THE VISCOUS CONTRIBUTION TO THE TIME STEP CALCULATION
4498  C
4499  1190 IF (NVC.NE.1.AND.NVC.NE.NVCM+1) GO TO 1250
4500      IF (L.EQ.1.OR.L.EQ.LMAX) GO TO 1250
4501      IF (M.EQ.1.OR.M.EQ.MMAX) GO TO 1250
4502      IF (M.EQ.MDFS.AND.LDFS.NE.O) GO TO 1250
4503      DXP1=XP(L)-XP(L-1)
4504      DXP2=XP(L+1)-XP(L)
4505      DYP3=DY/BE3
4506      DYP4=DY/BE4
4507      IF (RLA.LE.O.O) GO TO 1200
4508      RMUD1=RLA1
4509      RMUD2=RLA2
4510      RMUD3=RLA3
4511      RMUD4=RLA4
4512      GO TO 1210
4513  1200 RMUD1=RMU1
4514      RMUD2=RMU2
4515      RMUD3=RMU3
4516      RMUD4=RMU4
4517  1210 TMUTX1=(MU1+RMUD1*MUT1)/(RO(L.M.N1)*DXP1+DXP1)
4518      TMUTX2=(MU2+RMUD2*MUT2)/(RO(L.M.N1)*DXP2+DXP2)
4519      TMUTY3=(MU3+RMUD3*MUT3)/(RO(L.M.N1)*DYP3+DYP3)
4520      TMUTY4=(MU4+RMUD4*MUT4)/(RO(L.M.N1)*DYP4+DYP4)
4521      TMUTX=AMAX1(TMUTX1, TMUTX2)
4522      TMUTY=AMAX1(TMUTY3, TMUTY4)
4523      IF (NVC.NE.1) GO TO 1230
4524      IF (TMUTX.LE.TMUY) GO TO 1220
4525      LDUX=L
4526      MDUX=M
4527      TMUX=TMUTX
4528  1220 IF (TMUTY.LE.TMUY) GO TO 1250
4529      LDUY=L
4530      MDUY=M
4531      TMUY=TMUTY
4532      GO TO 1250
4533  1230 IF (TMUTX.LE.TMU1X) GO TO 1240
4534      LDUX=L
4535      MDUX=M
4536      TMU1X=TMUTX
4537  1240 IF (TMUTY.LE.TMU1Y) GO TO 1250
4538      LDUY=L
4539      MDUY=M
4540      TMU1Y=TMUTY
4541  C
4542  C      CALCULATE THE VISCOSITY AND HEAT CONDUCTION TERMS
4543  C
4544  1250 UVT=OM*((LP2M2+RLP2M2+LP2MT2)*UXY2-(LP2M1+RLP2M1+LP2MT1)*UXY1+(LA2

```



```

4545 1 *RLA2+LAT2)*BVY2-((LA1+RLA1+LAT1)*BVY1)*DXR+AL*((LP2M4+RPL2M4
4546 2 +LP2MT4)*UXY4-(LP2M3+RPL2M3+LP2MT3)*UXY3+((LA4+RLA4+LAT4)*BVY4-
4547 3 (LA3+RLA3+LAT3)*BVY3)*DYR+BE*((MU4+RMU4+MUT4)*VXY4-(MU3+RMU3+MUT3
4548 4 )*VXY3+(MU4+RMU4+MUT4)*BUY4-(MU3+RMU3+MUT3)*BUY3)*DYR
4549 VVT=OM*((MU2+RMU2+MUT2)*(VXY2+BUY2)-(MU1+RMU1+MUT1)*(VXY1+BUY1))
4550 1 *DXR+AL*((MU4+RMU4+MUT4)*VXY4-(MU3+RMU3+MUT3)*VXY3+(MU4+RMU4+MUT4
4551 2 )*BUY4-(MU3+RMU3+MUT3)*BUY3)*DYR+BE*((LA4+RLA4+LAT4)*UXY4-(LA3
4552 3 +RLA3+LAT3)*UXY3+((LP2M4+RPL2M4+LP2MT4)*BVY4-(LP2M3+RPL2M3+LP2MT3)
4553 4 *BVY3)*DYR
4554 PVT=((LP2M+RPL2M+DLP2MT)*(UXY12*UXY12+BVY34+BVY34)+(MU+RMU+DMUT)*
4555 1 (VXY12+VXY12+BUY34+BUY34)+2.0*(LA+RLA+DLAT)*UXY12+BVY34+2.0*(MU
4556 2 +RMU+DMUT)*BUY34+VXY12
4557 PCT=OM*((K2+RK2+KT2)*TXY2-(K1+RK1+KT1)*TXY1)*DXR+AL*((K4+RK4+KT4)
4558 1 *TXY4-(K3+RK3+KT3)*TXY3)*DYR+BE*((K4+RK4+KT4)*BTY4-(K3+RK3+KT3)
4559 2 *BTY3)*DYR
4560 IF (ITM.EQ.O.AND.CAV.EQ.O.O) GO TO 1280
4561 RODIFF=OM*((CAL+SMU2+RORR+RRO2)*ROXY2-(CAL+SMU1+RORR+RRO1)*ROXY1)
4562 1 *DXR+AL*((CAL+SMU4+RORR+RRO4)*ROXY4-(CAL+SMU3+RORR+RRO3)*ROXY3)
4563 2 *DYR+BE*((CAL+SMU4+RORR+RRO4)*BROY4-(CAL+SMU3+RORR+RRO3)*BROY3)
4564 3 *DYR
4565 IF (ITM.EQ.O) GO TO 1280
4566 UROT=-0.67*(OM+ROQX+AL*ROQY)+CAL*(U(L,M,N1))*(OM*(SMU2+ROXY2-SMU1
4567 1 *ROXY1)+DXR+AL*(SMU4+ROXY4-SMU3+ROXY3)*DYR)+BE*(U(L,M,N1))*(SMU4
4568 2 *ROXY4-SMU3+ROXY3)*DYR)
4569 VROT=-0.67*BE*ROQY+CAL*(V(L,M,N1)*BE*(SMU4+BROY4-SMU3+BROY3)*DYR+U
4570 1 (L,M,N1))*(OM*(SMU2+BROY2-SMU1+BROY1)+DXR+AL*(SMU4+BROY4-SMU3
4571 2 *BROY3)*DYR)
4572 RODUMT=OM*(SMU2+ROXY2-SMU1+ROXY1)*DXR+AL*(SMU4+ROXY4-SMU3+ROXY3)
4573 1 *DYR+BE*(SMU4+BROY4-SMU3+BROY3)*DYR
4574 PROT=-CAL*RG*T*RODUMT
4575 IF (IES.NE.O) GO TO 1280
4576 IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 1290
4577 IF (ITM.EQ.1) GO TO 1280
4578 QPROD=LP2MT*(UXY12+UXY12+BVY34+BVY34)+MUT*(VXY12+VXY12+BUY34+BUY34
4579 1 )+2.0*LAT*UXY12+BVY34+2.0*MUT*BUY34+VXY12
4580 QDIFF=OM*((MU2+MUT2*SIGOR)*QXY2-(MU1+MUT1*SIGOR)*QXY1)*DXR+AL*(
4581 1 (MU4+MUT4*SIGOR)*QXY4-(MU3+MUT3*SIGOR)*QXY3)*DYR+BE*((MU4+MUT4
4582 2 *SIGOR)*BOY4-(MU3+MUT3*SIGOR)*BOY3)*DYR
4583 QROTT=-XITM*O(L,M,N1)*RO(L,M,N1)*(UXY12+BVY34)
4584 IF (ITM.EQ.3) GO TO 1260
4585 QDISS=O.O
4586 IF (TML.NE.O.O) QDISS=2.0*MU*DELTA*O(L,M,N1)/(TML+TML)
4587 GO TO 1280
4588 1260 EPROD=O.O
4589 EDISS=O.O
4590 IF (O(L,M,N1).EQ.O.O) GO TO 1270
4591 EPROD=C1*E(L,M,N1)/O(L,M,N1)*((LP2MT*(UXY12+UXY12+BVY34+BVY34)+MUT*
4592 1 (VXY12+VXY12+BUY34+BUY34)+2.0*LAT*UXY12+BVY34+2.0*MUT*BUY34+VXY12
4593 2 )
4594 EDISS=C2T*RO(L,M,N1)*E(L,M,N1)*(E(L,M,N1)-2.0*MU+RORR+LC*(Q2XY
4595 1 +BQ2Y)*2)/(O(L,M,N1)+LC)
4596 IF (EDISS.LT.O.O) EDISS=O.O
4597 1270 EDIFF=OM*((MU2+MUT2*SIGER)*EXY2-(MU1+MUT1*SIGER)*EXY1)*DXR+AL*(
4598 1 (MU4+MUT4*SIGER)*EXY4-(MU3+MUT3*SIGER)*EXY3)*DYR+BE*((MU4+MUT4
4599 2 *SIGER)*BEY4-(MU3+MUT3*SIGER)*BEY3)*DYR
4600 QDISS=RO(L,M,N1)*(E(L,M,N1)+2.0*MU+RORR+LC*(Q2XY+BQ2Y)*2)/LC
4601 ELOWR=2.0+RORR+MU*MUT+LC*(OM*(UXY2-UXY1)*DXR+AL*(UXY4-UXY3)*DYR)+
4602 1 *2+(OM*(VXY2-VXY1)*DXR+AL*(VXY4-VXY3)*DYR)*2+(BE*(BUY4-BUY3)*DYR
4603 2 )*2+(BE*(BVY4-BVY3)*DYR)*2)
4604 C
4605 C Q AND E FOURTH ORDER SMOOTHING
4606 C
4607 IF (STBO.LE.O.O.AND.STBE.LE.O.O) GO TO 1280
4608 QGX=QLP-2.0*O(L,M,N1)+QLM
4609 DOY=OMP-2.0*O(L,M,N1)+OMM
4610 DEX=ELP-2.0*E(L,M,N1)+ELM
4611 DEY=EMP-2.0*E(L,M,N1)+EMM
4612 QAVGX=O.25*(QLP+2.0*O(L,M,N1)+QLM)
4613 QAVGY=O.25*(OMP+2.0*O(L,M,N1)+OMM)
4614 IF (QAVGX.LE.O.O) QAVGX=1.OE+10
4615 IF (QAVGY.LE.O.O) QAVGY=1.OE+10
4616 EAVGX=O.25*(ELP+2.0*E(L,M,N1)+ELM)

```

```

4617      EAVGY=0.25*(EMP+2.0*E(L,M,N1)+EMM)
4618      AST=SQRT(AL*AL+BE*BE)
4619      OSMO=STBO*RO(L,M,N1)*((ABS(U(L,M,N1))+A)*ABS(DQX)+OM*DXR*DQX/QAVGX
4620 1 +(ABS(U(L,M,N1)*AL+V(L,M,N1)*BE)+AST*A)*ABS(DQY)*DYR*DQY/QAVGY)
4621      ESMO=STBE*RO(L,M,N1)*((ABS(U(L,M,N1))+A)*ABS(DEX)+OM*DXR*DEX/EAVGX
4622 1 +(ABS(U(L,M,N1)*AL+V(L,M,N1)*BE)+AST*A)*ABS(DEY)*DYR*DEY/EAVGY)
4623 C
4624 C      PRINT THE TURBULENCE MODEL CONV. PROD. DISS. AND
4625 C      DIFF TERMS FOR THE REQUESTED GRID POINT
4626 C
4627 1280 IF (ITM.LE.1) GO TO 1290
4628      IF (L.NE.LPRINT.OR.M.NE.MPRINT) GO TO 1290
4629      IF (NVC.GT.2) GO TO 1290
4630      IF (M.EQ.1.OR.M.EQ.MMAX) GO TO 1290
4631      IF (M.EQ.MDFS.AND.LDFS.NE.O) GO TO 1290
4632      IF (N.EQ.1) WRITE (6,1470)
4633      UVB=U(L,M,N1)*AL+V(L,M,N1)*BE
4634      QCON=-U(L,M,N1)*OM*(Q(L+1,M,N1)-Q(L-1,M,N1))*DXR+UVB*(Q(L,M+1,N1)
4635 1 -Q(L,M-1,N1))*DYR)*0.5*DT
4636      ECON=-U(L,M,N1)*OM*(E(L+1,M,N1)-E(L-1,M,N1))*DXR+UVB*(E(L,M+1,N1)
4637 1 -E(L,M-1,N1))*DYR)*0.5*DT
4638      OPRG=OPROD*DT*RORR
4639      QDIS=-QDISS*DT*RORR
4640      QDIF=QDIFF*DT*RORR
4641      EPRO=EPROD*DT*RORR
4642      EDIS=-EDISS*DT*RORR
4643      EDIF=EDIFF*DT*RORR
4644      ELOR=ELOWR*DT*RORR
4645      NP=N+NSTART
4646      WRITE (6,1480) NP,L,M,Q(L,M,N1),QCON,OPRO,QDIS,QDIF,E(L,M,N1),ECON
4647 1 ,EPRO,EDIS,EDIF,ELOR
4648 1290 IF (NDIM.EQ.O) GO TO 1330
4649 C
4650 C      CALCULATE THE AXISYMMETRIC TERMS
4651 C
4652      IF (M.EQ.1.AND.YCB(L).EQ.O.O) GO TO 1310
4653      VB=V(L,M,N1)
4654      UVTA=((LPM+RLPM+LPMT)*VXY12+(MU+RMU+MUT)*BUY34)/YP
4655      VVTA=(LP2M+RLP2M+LP2MT)*(BVY34-VB/YP)/YP
4656      PVTA=((LP2M+RLP2M+DLP2MT)*VB+VB/YP+2.0*(LA+RLA+DLAT)*VB*(BVY34
4657 1 +UXY12))/YP
4658      PCTA=(K+RK+KT)*0.5*(BTY4+BTY3)/YP
4659      IF (ITM.EQ.O.AND.CAV.EQ.O.O) GO TO 1330
4660      ROIFFA=(CAL*MUT+RRO)*RORR*BROY34/YP
4661      IF (ITM.EQ.O) GO TO 1330
4662      UROTA=CAL*MUT*RORR*V(L,M,N1)*ROXY12/YP
4663      VROTA=CAL*MUT*RORR*V(L,M,N1)*BROY34/YP
4664      PROTA=-CAL*RG*T*MUT*RORR*BROY34/YP
4665      IF (IES.NE.O) GO TO 1330
4666      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 1330
4667      IT (ITM.EQ.1) GO TO 1330
4668      QPRODA=(LP2MT*VB+VB/YP+2.0*LAT*VB*(BVY34+UXY12))/YP
4669      QDIFFA=(MU+NJT*SIGOR)*BQY34/YP
4670      QROTTA=-XITM*Q(L,M,N1)*RO(L,M,N1)*VB/YP
4671      IF (ITM.EQ.2) GO TO 1330
4672      IF (Q(L,M,N1).EQ.O.O) GO TO 1300
4673      EPRCDA=C1*E(L,M,N1)*(LP2MT*VB+VB/YP+2.0*LAT*VB*(BVY34+UXY12))/(O(L
4674 1 ,M,N1)*YP)
4675 1300 EDIFFA=(MU+MUT*SIGER)*BEY34/YP
4676      ELOWRA=2.0*RORR*MU*MUT*LC*((BUY34/YP)**2+(BVY34/YP)**2+2.0*BUY34
4677 1 *BE*(BUY4-BUY3)*DYR/YP+2.0*BVY34*BE*(BVY4-BVY3)*DYR,/P)
4678      GO TO 1330
4679 C
4680 C      CALCULATE THE AXISYMMETRIC TERMS ON THE AXIS
4681 C
4682 1310 UVTA=(LPM+RLPM+LPMT)*BE*(VXY4-VXY3)*DYR+(MU+RMU+MUT)*BE*(BUY4-CUY3
4683 1 )*DYR
4684      VVTA=(LP2M+RLP2M+LP2MT)*0.5*BE*(BVY4-BVY3)*DYR
4685      PVTA=(LP2M+RLP2M+DLP2MT+2.0*(LA+RLA+DLAT))*BVY34+BVY34+2.0*(LA+RLA
4686 1 +DLAT)*BVY34*UXY12
4687      PCTA=(K+RK+KT)*BE*(BTY4-BTY3)*DYR
4688      IF (ITM.EQ.O.AND.CAV.EQ.O.O) GO TO 1330

```

```

4689      RODIFFA=(CAL*MUT*RRR)*RORR*BE*(BROY4-BROY3)*DYR
4690      IF (ITM.EQ.0) GO TO 1330
4691      UROTA=CAL*MUT*RORR*BVIY34*RDXY12
4692      VROTA=0.0
4693      PROTA=-CAL*RG*T*MUT*RORR*BE*(BROY4-BROY3)*DYR
4694      IF (IES.NE.0) GO TO 1330
4695      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 1330
4696      IF (ITM.EQ.1) GO TO 1330
4697      QPRODA=(LP2MT+2.0*LAT)*BVIY34*BVIY34+2.0*LAT*BVIY34*UXY12
4698      QDIFFA=(MU*MUT*SIGQR)*BE*(BQY4-BQY3)*DYR
4699      QROTTA=-XITM*Q(L,M,N1)*RO(L,M,N1)*BVIY34
4700      IF (ITM.EQ.2) GO TO 1330
4701      IF (Q(L,M,N1).EQ.0.0) GO TO 1320
4702      EPRODA=C1+E(L,M,N1)*((LP2MT+2.0*LAT)*BVIY34*BVIY34+2.0*LAT*BVIY34
4703      1 *UXY12)/Q(L,M,N1)
4704      1320 EDIFFA=(MU*MUT*SIGER)*BE*(BEY4-BEY3)*DYR
4705      ELOWRA=6.0*RORR*MU*MUT*LC*((BL*(BUY4-BUY3)*DYR)*2*(BE*(BVIY4-BVIY3)
4706      1 *DYR)*2)
4707 C
4708 C      FILL THE VISCOUS TERM ARRAYS
4709 C
4710      1330 QUT(L,M)=(UVT+UVTA+UROT+UROTA)*RORR
4711      QVT(L,M)=(VVT+VUTA+VROT+VROTA)*RORR
4712      OPT(L,M)=GAM*(PVT+PVTA+PCT+PCTA+PROT+PROTA*ODISS)
4713      IF (ITM.EQ.0.AND.CAV.EQ.0.0) GO TO 1340
4714      QROT(L,M)=RODIFF+RODIFFA
4715      IF (IES.NE.0) GO TO 1340
4716      IF (L.EQ.LMAX.OR.L.EQ.1) GO TO 1340
4717      IF (ITM.LE.1) GO TO 1340
4718      QOT(L,M)=(OPROD+QPRODA+QDIFF+QDIFFA+QROTT+QROTTA-ODISS+QSMO)*RORR
4719      QOET(L,M)=(EPROD+EPRODA+EDIFF+EDIFFA-EDISS+ELOWR+ELOWRA+ESMO)*RORR
4720 C
4721 C      PRINT THE VISCOUS TERMS
4722 C
4723      1340 IF (IAV.EQ.0) GO TO 1400
4724      IF (NC.NE.NPRINT.AND.(N.NE.NMAX.AND.ISTOP.EQ.0)) GO TO 1400
4725      IF (IAV.EQ.2) GO TO 1350
4726      IF (NVC.GT.2.AND.NVC.NE.NVCM+1) GO TO 1400
4727      1350 IF (L.EQ.1.AND.(NVC.EQ.1.AND.IB.NE.4)) GO TO 1370
4728      IF (L.EQ.1.AND.MDFS.EQ.0) GO TO 1370
4729      IF (L.EQ.1.AND.IB.EQ.3) GO TO 1370
4730      IF (M.EQ.MIS) GO TO 1360
4731      IF (M.EQ.MVCT+1.AND.(MDFS.NE.0.AND.MDFSC.EQ.0)) GO TO 1360
4732      IF (M.EQ.MVCT+1.AND.MVCB.EQ.1) GO TO 1360
4733      GO TO 1370
4734      1360 WRITE (6,1490)
4735      NLINE=NLINE+1
4736      1370 NLINE=NLINE+1
4737      IF (NLINE.LT.54) GO TO 1380
4738      WRITE (6,1460)
4739      NP=N+NSTART
4740      WRITE (6,1450) NP,NVC
4741      NLINE=1
4742      1380 DQPT=OPT(L,M)/PC*DT
4743      DQUT=QUT(L,M)*DT
4744      DQVT=QVT(L,M)*DT
4745      DQROT=QROT(L,M)*G*DT
4746      DO=O(L,M,N1)
4747      DDE=E(L,M,N1)
4748      DQOT=QOT(L,M)*DT
4749      DQOET=QOET(L,M)*DT
4750      DTML=TML
4751      IF (IUO.NE.2) GO TO 1390
4752      DQUT=DQUT*0.3048
4753      DQVT=DQVT*0.3048
4754      DQPT=DQPT*6.8948
4755      DQROT=DQROT 5.02
4756      DQ=DO*0.0929
4757      DDE=DDE*0.0929
4758      DQOT=DQOT*0.0929
4759      DQOET=DQOET*0.0929
4760      DTML=DTML*2.54
4761      1390 WRITE (6,1440) L,M,DQUT,DQVT,DQPT,DQROT,AVMUR,TLMUR,DO,DDE,DQOT

```

```

4762      1 ,DOET,DTML
4763 1400 CONTINUE
4764 1410 CONTINUE
4765      IF (MDFS.EQ.O) GO TO 1420
4766      IF (NVC.EQ.1.AND.MDFSC.NE.O) GO TO 1420
4767      IF (MIS.EQ.1.AND.MIF.EQ.MDFS) GO TO 50
4768      IF (MIS.EQ.MVCS.AND.MIF.EQ.MDFS) GO TO 50
4769      IF (MIS.EQ.MDFS+1.AND.IVC.EQ.O) GO TO 60
4770      IF (MIS.EQ.MDFS+1.AND.NVC.NE.1) GO TO 60
4771 1420 IF (IAV.EQ.O) RETURN
4772      IF (NC.NE.NPRINT.AND.N.NE.NMAX) RETURN
4773      IF (IAV.EQ.2) GO TO 1430
4774      IF (NVC.NE.1.AND.NVC.NE.NVCM+1) RETURN
4775 1430 IF (TMUX.NE.O.O) RDUM=TMUY/TMUX
4776      IF (NVC.NE.1.AND.TMUX.NE.O.O) RDUM=TMUY/TMUX
4777      WRITE (6,1500) LDUX,MUUX,LDUY,MDUY,RDUM,NVC
4778      RETURN
4779 C
4780 C      FORMAT STATEMENTS
4781 C
4782 1440 FORMAT (1H ,2I5,2F11.4,F11.5,F11.6,2F11.3,F12.4,E10.3,F11.4,E10.3
4783      1 ,F11.6)
4784 1450 FORMAT (1H ,5ILOCAL VISCOSITY (ARTIFICIAL-MOLECULAR-TURBULENT) AND
4785      1 ,26H HEAT CONDUCTION TERMS. N=,16.6H, NVC=,13//5X,1HL,4X,1HM,7X,3
4786      2 HQUT,8X,3HQVT,8X,3HQPT,7X,4HQROT,7X,5HQV/MUR,6X,5HTLMUR,8X,1HO,9X,
4787      3 1HE,10X,3HQOT,6X,3HQET,8X,3HTML./)
4788 1460 FORMAT (1H1)
4789 1470 FORMAT (1H1,3X,1HM,3X,1HL,3X,1HM,5X,1HO,8X,4HQCON,6X,4HQPRO,6X,4HQ
4790      1DIS,6X,4HQDIF,7X,1HE,8X,4HECON,6X,4HEPRO,6X,4HEDIS,6X,4HEDIF,6X,4H
4791      2ELOR./)
4792 1480 FORMAT (1H ,3I4,11E10.3)
4793 1490 FORMAT (1H ,3X,48H-----)
4794      1 ,61H-----)
4795      2 ,18H-----)
4796 1500 FORMAT (1HO,10X,20HX TERMS GRID POINT=(.12,1H,.12,25H), Y TERMS
4797      1GRID POINT=(.12,1H,.12,22H), RATIO OF Y TO X=(.E9.3.9H), NVC=
4798      2 ,13./)
4799 1510 FORMAT (1HO,109H***** THE TEMPERATURE USED IN THE MOLECULAR VISCOS
4800      1ITY CALCULATION IN SUBROUTINE VISCOUS BECAME NEGATIVE AT N=.16,1H,
4801      2 ./,7X,2HL=.12,4H, M=.12,6H, NVC=.13,6H ***** )
4802      END

```

```

4803      SUBROUTINE SMOOTH
4804 C
4805 C      .....
4806 C
4807 C      THIS SUBROUTINE SMOOTHS THE FLOW VARIABLES IF REQUESTED
4808 C
4809 C      .....
4810 C
4811 C *CALL,MCC
4812 C
4813 C      SPACE SMOOTHING
4814 C
4815      IF (SMP.EQ.1.0) GO TO 100
4816      SMP4=0.25*(1.0-SMP)
4817      IF (MDFS.EQ.0) GO TO 20
4818      IF (LDFSS.EQ.1.AND.LDFSF.EQ.LMAX) GO TO 20
4819      IF (LDFSS.EQ.1) GO TO 10
4820      UL(LDFSS-1,N3)=U(LDFSS-1,MDFS,N3)
4821      VL(LDFSS-1,N3)=V(LDFSS-1,MDFS,N3)
4822      PL(LDFSS-1,N3)=P(LDFSS-1,MDFS,N3)
4823      ROL(LDFSS-1,N3)=R(LDFSS-1,MDFS,N3)
4824      OL(LDFSS-1,N3)=O(LDFSS-1,MDFS,N3)
4825      EL(LDFSS-1,N3)=E(LDFSS-1,MDFS,N3)
4826      10 IF (LDFSF.EQ.LMAX) GO TO 20
4827      UL(LDFSF+1,N3)=U(LDFSF+1,MDFS,N3)
4828      VL(LDFSF+1,N3)=V(LDFSF+1,MDFS,N3)
4829      PL(LDFSF+1,N3)=P(LDFSF+1,MDFS,N3)
4830      ROL(LDFSF+1,N3)=R(LDFSF+1,MDFS,N3)
4831      OL(LDFSF+1,N3)=O(LDFSF+1,MDFS,N3)
4832      EL(LDFSF+1,N3)=E(LDFSF+1,MDFS,N3)
4833 C
4834      20 DO 90 L=2,L1
4835      IF (IWALL.NE.0.AND.V(L,MMAX,N1).LT.0.0) GO TO 40
4836      U(L,MMAX,N3)=SMP4*(U(L-1,MMAX,N3)+U(L+1,MMAX,N3))+2.0*U(L,MMAX,N3)
4837      1 +SMP*U(L,MMAX,N3)
4838      IF (NOSLIP.NE.0.AND.IWALL.EQ.0) U(L,MMAX,N3)=0.0
4839      IF (IWALL.EQ.0) V(L,MMAX,N3)=-U(L,MMAX,N3)*NXNY(L)+XWI(L)
4840      IF (IWALL.NE.0) GO TO 30
4841      IF (JFLAG.EQ.1.AND.L.GE.LJET) GO TO 30
4842      P(L,MMAX,N3)=SMP4*(P(L-1,MMAX,N3)+P(L+1,MMAX,N3))+2.0*P(L,MMAX,N3)
4843      1 +SMP*P(L,MMAX,N3)
4844      30 RO(L,MMAX,N3)=SMP4*(RO(L-1,MMAX,N3)+RO(L+1,MMAX,N3))+2.0*RO(L,MMAX
4845      1 ,N3)+SMP*RO(L,MMAX,N3)
4846      IF (TW(1).GE.0.0) P(L,MMAX,N3)=RO(L,MMAX,N3)+RG*TW(L)
4847      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
4848      1 ,N3)
4849      IF (NOSLIP.NE.0.AND.NGCB.NE.0) U(L,1,N3)=0.0
4850      V(L,1,N3)=-U(L,1,N3)*NXNYCB(L)
4851      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
4852      1 ,N3)
4853      RO(L,1,N3)=SMP4*(RO(L-1,1,N3)+RO(L+1,1,N3))+2.0*RO(L,1,N3)+SMP*RO
4854      1 (L,1,N3)
4855      IF (TCB(1).GE.0.0.AND.NGCB.NE.0) P(L,1,N3)=RO(L,1,N3)+RG*TCB(L)
4856      IF (ITM.LE.1) GO TO 50
4857      Q(L,MMAX,N3)=SMP4*(Q(L-1,MMAX,N3)+Q(L+1,MMAX,N3))+2.0*Q(L,MMAX,N3)
4858      1 +SMP*Q(L,MMAX,N3)
4859      E(L,MMAX,N3)=SMP4*(E(L-1,MMAX,N3)+E(L+1,MMAX,N3))+2.0*E(L,MMAX,N3)
4860      1 +SMP*E(L,MMAX,N3)
4861      O(L,1,N3)=SMP4*(O(L-1,1,N3)+O(L+1,1,N3))+2.0*O(L,1,N3)+SMP*O(L,1
4862      1 ,N3)
4863      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
4864      1 ,N3)
4865      50 LDFS=0
4866      IF (MDFS.EQ.0) GO TO 60
4867      IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
4868      IF (LDFS.EQ.0) GO TO 60
4869      UL(L,N3)=SMP4*(UL(L-1,N3)+UL(L+1,N3))+2.0*U(L,MDFS-1,N3)+SMP*UL(L
4870      1 ,N3)
4871      IF (NOSLIP.NE.0) UL(L,N3)=0.0
4872      VL(L,N3)=-UL(L,N3)*NXNY(L)
4873      PL(L,N3)=SMP4*(PL(L-1,N3)+PL(L+1,N3))+2.0*P(L,MDFS-1,N3)+SMP*PL(L
4874      1 ,N3)

```

```

4875     ROL(L,N3)=SMP4*(ROL(L-1,N3)+ROL(L+1,N3)+2.0*RO(L,MDFS-1,N3))+SMP
4876     1*ROL(L,N3)
4877     IF (TL(1).GE.O.O) PL(L,N3)=ROL(L,N3)*RG*TL(L)
4878     U(L,MDFS,N3)=SMP4*(U(L-1,MDFS,N3)+U(L+1,MDFS,N3)+2.0*U(L,MDFS+1,N3
4879     1))+SMP*U(L,MDFS,N3)
4880     IF (WOSLIP.NE.O) U(L,MDFS,N3)=O.O
4881     V(L,MDFS,N3)=-U(L,MDFS,N3)*HXNYU(L)
4882     P(L,MDFS,N3)=SMP4*(P(L-1,MDFS,N3)+P(L+1,MDFS,N3)+2.0*P(L,MDFS+1,N3
4883     1))+SMP*P(L,MDFS,N3)
4884     RO(L,MDFS,N3)=SMP4*(RO(L-1,MDFS,N3)+RO(L+1,MDFS,N3)+2.0*RO(L,MDFS+
4885     1,N3))+SMP*RO(L,MDFS,N3)
4886     IF (TU(1).GE.O.O) P(L,MDFS,N3)=RO(L,MDFS,N3)*RG*TU(L)
4887     IF (ITM.LE.1) GO TO 60
4888     QL(L,N3)=SMP4*(QL(L-1,N3)+QL(L+1,N3)+2.0*Q(L,MDFS-1,N3))+SMP*QL(L
4889     1,N3)
4890     EL(L,N3)=SMP4*(EL(L-1,N3)+EL(L+1,N3)+2.0*E(L,MDFS-1,N3))+SMP*EL(L
4891     1,N3)
4892     O(L,MDFS,N3)=SMP4*(O(L-1,MDFS,N3)+O(L+1,MDFS,N3)+2.0*O(L,MDFS+1,N3
4893     1))+SMP*O(L,MDFS,N3)
4894     E(L,MDFS,N3)=SMP4*(E(L-1,MDFS,N3)+E(L+1,MDFS,N3)+2.0*E(L,MDFS+1,N3
4895     1))+SMP*E(L,MDFS,N3)
4896 C
4897     60 DO 90 M=2,M1
4898     IF (M.EQ.MDFS.AND.LDFS.EQ.1) GO TO 90
4899     IF (M.NE.MDFS) GO TO 80
4900     IF (L.NE.LDFS-1.AND.L.NE.LDFS+1) GO TO 80
4901     IF (L.NE.LDFS-1) GO TO 70
4902     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
4903     1,N3)+U(L+1,N3)))+SMP*U(L,M,N3)
4904     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
4905     1,N3)+V(L+1,N3)))+SMP*V(L,M,N3)
4906     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
4907     1,N3)+P(L+1,N3)))+SMP*P(L,M,N3)
4908     RO(L,M,N3)=SMP4*(RO(L-1,M,N3)+RO(L,M-1,N3)+RO(L,M+1,N3)+0.5*(RO(L+
4909     1,M,N3)+RO(L+1,N3)))+SMP*RO(L,M,N3)
4910     IF (ITM.LE.1) GO TO 90
4911     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
4912     1,N3)+Q(L+1,N3)))+SMP*Q(L,M,N3)
4913     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
4914     1,N3)+E(L+1,N3)))+SMP*E(L,M,N3)
4915     GO TO 90
4916     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
4917     1,N3)+U(L-1,N3)))+SMP*U(L,M,N3)
4918     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
4919     1,N3)+V(L-1,N3)))+SMP*V(L,M,N3)
4920     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
4921     1,N3)+P(L-1,N3)))+SMP*P(L,M,N3)
4922     RO(L,M,N3)=SMP4*(RO(L+1,M,N3)+RO(L,M-1,N3)+RO(L,M+1,N3)+0.5*(RO(L-
4923     1,M,N3)+RO(L-1,N3)))+SMP*RO(L,M,N3)
4924     IF (ITM.LE.1) GO TO 90
4925     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
4926     1,N3)+Q(L-1,N3)))+SMP*Q(L,M,N3)
4927     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
4928     1,N3)+E(L-1,N3)))+SMP*E(L,M,N3)
4929     GO TO 90
4930     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))
4931     1+SMP*U(L,M,N3)
4932     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))
4933     1+SMP*V(L,M,N3)
4934     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))
4935     1+SMP*P(L,M,N3)
4936     RO(L,M,N3)=SMP4*(RO(L-1,M,N3)+RO(L+1,M,N3)+RO(L,M-1,N3)+RO(L,M+1
4937     1,N3))+SMP*RO(L,M,N3)
4938     IF (ITM.LE.1) GO TO 90
4939     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))
4940     1+SMP*Q(L,M,N3)
4941     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))
4942     1+SMP*E(L,M,N3)
4943     90 CONTINUE
4944 C
4945 C     TIME SMOOTHING (NTST.EQ.1)

```

```

4946 C
4947 100 IF (SMPT.EQ.1.0) RETURN
4948 IF (NTST.EQ.-1) GO TO 130
4949 NTC=NTC+1
4950 IF (NTC.NE.NTST) RETURN
4951 NTC=0
4952 IF (NTST.NE.1) GO TO 130
4953 C
4954 DO 120 L=1,LMAX
4955 LDFS=0
4956 IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
4957 C
4958 DO 120 M=1,MMAX
4959 U(L,M,N3)=SMPT*U(L,M,N3)+(1.0-SMPT)*U(L,M,N1)
4960 V(L,M,N3)=SMPT*V(L,M,N3)+(1.0-SMPT)*V(L,M,N1)
4961 P(L,M,N3)=SMPT*P(L,M,N3)+(1.0-SMPT)*P(L,M,N1)
4962 RO(L,M,N3)=SMPT*RO(L,M,N3)+(1.0-SMPT)*RO(L,M,N1)
4963 IF (ITM.LE.1) GO TO 110
4964 Q(L,M,N3)=SMPT*Q(L,M,N3)+(1.0-SMPT)*Q(L,M,N1)
4965 E(L,M,N3)=SMPT*E(L,M,N3)+(1.0-SMPT)*E(L,M,N1)
4966 110 IF (MDFS.EQ.0.OR.LDFS.EQ.0) GO TO 120
4967 UL(L,N3)=SMPT*UL(L,N3)+(1.0-SMPT)*UL(L,N1)
4968 VL(L,N3)=SMPT*VL(L,N3)+(1.0-SMPT)*VL(L,N1)
4969 PL(L,N3)=SMPT*PL(L,N3)+(1.0-SMPT)*PL(L,N1)
4970 ROL(L,N3)=SMPT*ROL(L,N3)+(1.0-SMPT)*ROL(L,N1)
4971 IF (ITM.LE.1) GO TO 120
4972 QL(L,N3)=SMPT*QL(L,N3)+(1.0-SMPT)*QL(L,N1)
4973 EL(L,N3)=SMPT*EL(L,N3)+(1.0-SMPT)*EL(L,N1)
4974 120 CONTINUE
4975 RETURN
4976 C
4977 C TIME SMOOTHING (NTST.GT.1)
4978 C
4979 130 DO 150 L=1,LMAX
4980 LDFS=0
4981 IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
4982 C
4983 DO 150 M=1,MMAX
4984 U(L,M,N3)=SMPT*U(L,M,N3)+(1.0-SMPT)*US(L,M)
4985 V(L,M,N3)=SMPT*V(L,M,N3)+(1.0-SMPT)*VS(L,M)
4986 P(L,M,N3)=SMPT*P(L,M,N3)+(1.0-SMPT)*PS(L,M)
4987 RO(L,M,N3)=SMPT*RO(L,M,N3)+(1.0-SMPT)*ROS(L,M)
4988 US(L,M)=U(L,M,N3)
4989 VS(L,M)=V(L,M,N3)
4990 PS(L,M)=P(L,M,N3)
4991 ROS(L,M)=RO(L,M,N3)
4992 IF (ITM.LE.1) GO TO 140
4993 QL(L,M,N3)=SMPT*Q(L,M,N3)+(1.0-SMPT)*QS(L,M)
4994 EL(L,M,N3)=SMPT*E(L,M,N3)+(1.0-SMPT)*ES(L,M)
4995 QS(L,M)=Q(L,M,N3)
4996 ES(L,M)=E(L,M,N3)
4997 140 IF (MDFS.EQ.0.OR.LDFS.EQ.0) GO TO 150
4998 UL(L,N3)=SMPT*UL(L,N3)+(1.0-SMPT)*ULS(L)
4999 VL(L,N3)=SMPT*VL(L,N3)+(1.0-SMPT)*VLS(L)
5000 PL(L,N3)=SMPT*PL(L,N3)+(1.0-SMPT)*PLS(L)
5001 ROL(L,N3)=SMPT*ROL(L,N3)+(1.0-SMPT)*ROLS(L)
5002 ULS(L)=UL(L,N3)
5003 VLS(L)=VL(L,N3)
5004 PLS(L)=PL(L,N3)
5005 ROLS(L)=ROL(L,N3)
5006 IF (ITM.LE.1) GO TO 150
5007 QL(L,N3)=SMPT*QL(L,N3)+(1.0-SMPT)*QLS(L)
5008 EL(L,N3)=SMPT*EL(L,N3)+(1.0-SMPT)*ELS(L)
5009 QLS(L)=QL(L,N3)
5010 ELS(L)=EL(L,N3)
5011 150 CONTINUE
5012 RETURN
5013 END

```

```

5014      SUBROUTINE MIXLEN (L,MV)
5015 C
5016 C      .....
5017 C
5018 C      THIS SUBROUTINE CALCULATES THE SHEAR LAYER WIDTH , BOUNDARY
5019 C      LAYER THICKNESS AND DISPLACEMENT THICKNESS FOR THE MIXING-LENGTH
5020 C      MODEL (ITM=1) AND ONE EQUATION MODEL (ITM=2)
5021 C
5022 C      .....
5023 C
5024 *CALL ,MCC
5025 C
5026 C      CALCULATE THE SHEAR LAYER WIDTH (YSL2-YSL1)
5027 C
5028      IP=0
5029      LMAP=L
5030      IMP=0
5031      IF (ITM.EQ.2) GO TO 120
5032      UMIN=U(L,1,N1)
5033      DO 10 M=1,MMAX
5034      IF (U(L,M,N1).GT.UMIN) GO TO 10
5035      UMIN=U(L,M,N1)
5036      MMIN=M
5037 10 CONTINUE
5038      IF (MMIN.EQ.1.OR.MMIN.EQ.MMAX) IMP=1
5039      IF (U(L,1,N1).EQ.U(L,MMAX,N1)) GO TO 20
5040      IF (U(L,MMAX,N1).GT.U(L,1,N1)) UCHECK=(U(L,1,N1)-UMIN)/(U(L,MMAX
5041      1,N1)-U(L,1,N1))
5042      IF (U(L,MMAX,N1).LT.U(L,1,N1)) UCHECK=(U(L,MMAX,N1)-UMIN)/(U(L,1
5043      1,N1)-U(L,MMAX,N1))
5044      IF (UCHECK.LT.0.05) IMP=1
5045 20 IF (IMP.NE.0) GO TO 30
5046      UDUM=UMIN
5047      RDUL=1.0/(U(L,1,N1)-UDUM)
5048      RDUU=1.0/(U(L,MMAX,N1)-UDUM)
5049      GO TO 40
5050 C
5051 30 IF (U(L,1,N1).EQ.U(L,MMAX,N1)) GO TO 110
5052      UDUM=U(L,MMAX,N1)
5053      RDU=1.0/(U(L,1,N1)-UDUM)
5054 C
5055 40 DO 90 M=1,M1
5056      MMAP=M
5057      CALL MAP
5058      IF (M.EQ.MMIN) YMIN=YF
5059      MMAP=M+1
5060      YP1=YF
5061      CALL MAP
5062      DYP=YF-YP1
5063      IF (IMP.NE.0) GO TO 50
5064      RDU=RDUU
5065      IF (M.GE.MMIN) RDU=RDUU
5066 50 UD1=(U(L,M,N1)-UDUM)*RDU
5067      UD2=(U(L,M+1,N1)-UDUM)*RDU
5068      IF (UD1.GE.0.9.AND.UD2.LE.0.9) GO TO 60
5069      IF (UD1.LE.0.9.AND.UD2.GE.0.9) GO TO 60
5070      IF (IMP.EQ.0) GO TO 30
5071      IF (UD1.GE.0.1.AND.UD2.LE.0.1) GO TO 80
5072      GO TO 90
5073 60 YSL2=YP1+(0.9-UD1)*DYP/(UD2-UD1)
5074      IF (IMP.NE.0) GO TO 70
5075      IF (M.GE.MMIN) GO TO 100
5076      IF (M.LT.MMIN) YSL1=YSL2
5077      GO TO 90
5078 70 IF (UD1.GE.0.1.AND.UD2.LE.0.1) GO TO 80
5079      GO TO 90
5080 80 YSL1=YP1+(0.1-UD1)*DYP/(UD2-UD1)
5081      GO TO 100
5082 90 CONTINUE
5083      YSL1=Y(L)
5084 100 IP=1
5085      RETURN

```



```

5086 C
5087 110 YSL1=0.0
5088 YSL2=0.0
5089 YMIN=0.0
5090 IP=1
5091 RETURN
5092 C
5093 C CALCULATE THE BOUNDARY LAYER THICKNESS (DEL)
5094 C
5095 120 MM3=MMAX
5096 MM4=0
5097 IF (MDFS.EQ.0) GO TO 150
5098 IF (NVC.NE.1) GO TO 130
5099 IBD=IB
5100 IF (MV.LT.MDFS) IB=3
5101 IF (MV.GT.MDFS) IB=4
5102 130 IF (IB.EQ.4) GO TO 140
5103 MM3=MDFS
5104 M=MM3+1
5105 MDEL=-1
5106 UMAX=U(L,1,N1)*RO(L,1,N1)
5107 GO TO 170
5108 140 MM4=MDFS-1
5109 M=MM4
5110 MDEL=1
5111 UMAX=U(L,MMAX,N1)*RO(L,MMAX,N1)
5112 GO TO 170
5113 C
5114 150 IF (IWALL.EQ.0) GO TO 160
5115 M=MM4
5116 MDEL=1
5117 UMAX=U(L,MMAX,N1)*RO(L,MMAX,N1)
5118 GO TO 170
5119 160 M=MM3+1
5120 MDEL=-1
5121 UMAX=U(L,1,N1)*RO(L,1,N1)
5122 C
5123 170 DO 180 MM=1,M1
5124 M=M+MDEL
5125 IF (M+MDEL.EQ.0) GO TO 190
5126 IF (M+MDEL.EQ.MMAX+1) GO TO 190
5127 UD1=U(L,M,N1)*RO(L,M,N1)/UMAX
5128 UD2=U(L,M+MDEL,N1)*RO(L,M+MDEL,N1)/UMAX
5129 IF (UD1.LE.0.98.AND.UD2.GE.0.98) GO TO 200
5130 IF (UD1.GE.0.98.AND.UD2.LE.0.98) GO TO 200
5131 180 CONTINUE
5132 190 DEL=0.0
5133 RETURN
5134 200 MMAP=M
5135 CALL MAP
5136 MMAP=M+MDEL
5137 YP1=YP
5138 CALL MAP
5139 DYP=YP-YP1
5140 Y2=YP1+(0.98-UD1)*DYP/(UD2-UD1)
5141 IF (MDFS.EQ.0) GO TO 210
5142 IF (IB.EQ.3) DEL=Y(L)-Y2
5143 IF (IB.EQ.4) DEL=Y2-YU(L)
5144 GO TO 220
5145 210 IF (IWALL.EQ.0) DEL=YW(L)-Y2
5146 IF (IWALL.NE.0) DEL=Y2-YCB(L)
5147 C
5148 C CALCULATE THE DISPLACEMENT THICKNESS (DELS)
5149 C
5150 220 DELS=0.0
5151 IF (IWALL.EQ.0) GO TO 230
5152 IF (MDFS.NE.0.AND.IB.EQ.3) GO TO 230
5153 MBLE=M+1-MM4
5154 UBLE=U(L,M+1,N1)
5155 ROUBLE=UBLE*RO(L,M+1,N1)
5156 M=MM4
5157 MDEL=1

```

```

5158      GO TO 240
5159 C
5160 230 MBLE=MM3-M+2
5161      UBLE=U(L,M-1,N1)
5162      ROUBLE=UBLE*RO(L,M-1,N1)
5163      M=MM3+1
5164      MDEL=-1
5165 C
5166 240 MBLE1=MBLE-1
5167      DG 250 MM=1,MBLE1
5168      M=M+MDEL
5169      MMAP=M
5170      CALL MAP
5171      MMAP=M+MDEL
5172      YP1=YP
5173      CALL MAP
5174      DYP=ABS(YP-YP1)
5175      DELS=DELS+(1.0-0.5*(U(L,M,N1)*RO(L,M,N1)+U(L,M+MDEL,N1)*RO(L,M
5176 1 +MDEL,N1))/ROUBLE)*DYP
5177 250 CONTINUE
5178      IF (MODS.NE.O.AND.NVC.EQ.1) IB=IBD
5179      IP=1
5180      RETURN
5181      END

```

```

5182      SUBROUTINE TURBC (II)
5183 C
5184 C      .....
5185 C
5186 C      THIS SUBROUTINE SETS THE BOUNDARY CONDITIONS FOR THE TURBULENCE
5187 C      QUANTITIES Q AND E
5188 C
5189 C      .....
5190 C
5191 C CALL MCC
5192      YI1=(YI(2)-YI(1))/(YI(3)-YI(2))
5193      YIM=(YI(MMAX)+1(YI(M1)))/(YI(M1)-YI(M2))
5194      IF (MDFS.EQ.0) GO TO 10
5195      YIU=(YI(MDFS+1)-YI(MDFS))/(YI(MDFS+2)-YI(MDFS+1))
5196      IF (LDFSS.EQ.1) YIL=(YL(1)-YI(MDFS-1))/(YI(MDFS-1)-YI(MDFS-2))
5197      IF (LDFSS.NE.1) YIL=(YI(MDFS)-YI(MDFS-1))/(YI(MDFS-1)-YI(MDFS-2))
5198      10 GO TO (20,70,150), II
5199 C
5200 C      SET QUANTITIES AFTER EACH TIME STEP
5201 C
5202      20 DO 30 M=1,MMAX
5203          Q(1,M,N3)=FSQ(M)
5204          E(1,M,N3)=FSE(M)
5205      30 CONTINUE
5206          DO 40 L=2,L1
5207              Q(L,MMAX,N3)=Q(L,M1,N3)+YIM*(Q(L,M1,N3)-Q(L,M2,N3))
5208              E(L,MMAX,N3)=E(L,M1,N3)+YIM*(E(L,M1,N3)-E(L,M2,N3))
5209              IF (NOSLIP.NE.0.AND.IWALL.EQ.0) Q(L,MMAX,N3)=0.0
5210              IF (NGCB.EQ.0) GO TO 40
5211              Q(L,1,N3)=Q(L,2,N3)+YI1*(Q(L,2,N3)-Q(L,3,N3))
5212              E(L,1,N3)=E(L,2,N3)+YI1*(E(L,2,N3)-E(L,3,N3))
5213              IF (NOSLIP.NE.0) Q(L,1,N3)=0.0
5214      40 CONTINUE
5215          DO 50 M=1,MMAX
5216              Q(LMAX,M,N3)=Q(L1,M,N3)
5217              E(LMAX,M,N3)=E(L1,M,N3)
5218      50 CONTINUE
5219          IF (MDFS.EQ.0) GO TO 280
5220          QL(1,N3)=FSQL
5221          EL(1,N3)=FSEL
5222          DO 60 L=LDFSS,LDFSF
5223              Q(L,MDFS,N3)=Q(L,MDFS+1,N3)+YIU*(Q(L,MDFS+1,N3)-Q(L,MDFS+2,N3))
5224              E(L,MDFS,N3)=E(L,MDFS+1,N3)+YIU*(E(L,MDFS+1,N3)-E(L,MDFS+2,N3))
5225              QL(L,N3)=Q(L,MDFS-1,N3)+YIL*(Q(L,MDFS-1,N3)-Q(L,MDFS-2,N3))
5226              EL(L,N3)=E(L,MDFS-1,N3)+YIL*(E(L,MDFS-1,N3)-E(L,MDFS-2,N3))
5227              IF (NOSLIP.NE.0) Q(L,MDFS,N3)=0.0
5228              IF (NOSLIP.NE.0) QL(L,N3)=0.0
5229      60 CONTINUE
5230          GO TO 280
5231 C
5232 C      SET QUANTITIES AFTER EACH SUBCYCLE TIME STEP
5233 C
5234      70 DO 80 M=MVCB,MVCT
5235          Q(1,M,N3)=FSQ(M)
5236          E(1,M,N3)=FSE(M)
5237      80 CONTINUE
5238          IF (MVCT.NE.MMAX) GO TO 100
5239          DO 90 L=2,L1
5240              Q(L,MMAX,N3)=Q(L,M1,N3)+YIM*(Q(L,M1,N3)-Q(L,M2,N3))
5241              E(L,MMAX,N3)=E(L,M1,N3)+YIM*(E(L,M1,N3)-E(L,M2,N3))
5242              IF (NOSLIP.NE.0.AND.IWALL.EQ.0) Q(L,MMAX,N3)=0.0
5243      90 CONTINUE
5244          100 IF (MVCB.NE.1.OR.NGCB.EQ.0) GO TO 120
5245              DO 110 L=2,L1
5246                  Q(L,1,N3)=Q(L,2,N3)+YI1*(Q(L,2,N3)-Q(L,3,N3))
5247                  E(L,1,N3)=E(L,2,N3)+YI1*(E(L,2,N3)-E(L,3,N3))
5248                  IF (NOSLIP.NE.0) Q(L,1,N3)=0.0
5249      110 CONTINUE
5250          120 DO 130 M=MVCB,MVCT
5251              Q(LMAX,M,N3)=Q(L1,M,N3)
5252              E(LMAX,M,N3)=E(L1,M,N3)
5253      130 CONTINUE

```

```

5254      IF (MDFS.EQ.O) GO TO 280
5255      QL(1,N3)=FSQL
5256      EL(1,N3)=FSEL
5257      IF (MVCB.GT.MDFS.OR.MVCT.LT.MDFS) GO TO 280
5258      DO 140 L=LDFSS,LDFSF
5259      Q(L,MDFS,N3)=Q(L,MDFS+1,N3)+YIU*(Q(L,MDFS+1,N3)-Q(L,MDFS+2,N3))
5260      E(L,MDFS,N3)=E(L,MDFS+1,N3)+YIU*(E(L,MDFS+1,N3)-E(L,MDFS+2,N3))
5261      QL(L,N3)=Q(L,MDFS-1,N3)+YIL*(Q(L,MDFS-1,N3)-Q(L,MDFS-2,N3))
5262      EL(L,N3)=E(L,MDFS-1,N3)+YIL*(E(L,MDFS-1,N3)-E(L,MDFS-2,N3))
5263      IF (NOSLIP.NE.O) Q(L,MDFS,N3)=O.O
5264      IF (NOSLIP.NE.O) QL(L,N3)=O.O
5265      140 CONTINUE
5266      GO TO 280
5267 C
5268 C      SE1 QUANTITIES AFTER ALL PREDICTOR STEPS
5269 C
5270      150 IF (NVC.NE.1) GO TO 190
5271      IF (MVCT.FO.MMAX) GO TO 170
5272      DO 160 L=2,L1
5273      Q(L,MMAX,N3)=Q(L,M1,N3)+YIM*(Q(L,M1,N3)-Q(L,M2,N3))
5274      E(L,MMAX,N3)=E(L,M1,N3)+YIM*(E(L,M1,N3)-E(L,M2,N3))
5275      IF (NOSLIP.NE.O.AND.IWALL.EQ.O) Q(L,MMAX,N3)=O.O
5276      160 CONTINUE
5277      170 DO 180 M=1,MMAX
5278      IF (M.GE.MVCB.AND.M.LE.MVCT) GO TO 180
5279      Q(LMAX,M,N3)=Q(L1,M,N3)
5280      E(LMAX,M,N3)=E(L1,M,N3)
5281      180 CONTINUE
5282      GO TO 230
5283      190 IF (MVCT.NE.MMAX) GO TO 210
5284      DO 200 L=2,L1
5285      Q(L,MMAX,N3)=Q(L,M1,N3)+YIM*(Q(L,M1,N3)-Q(L,M2,N3))
5286      E(L,MMAX,N3)=E(L,M1,N3)+YIM*(E(L,M1,N3)-E(L,M2,N3))
5287      IF (NOSLIP.NE.O.AND.IWALL.EQ.O) Q(L,MMAX,N3)=O.O
5288      200 CONTINUE
5289      210 DO 220 M=MVCB,MVCT
5290      Q(LMAX,M,N3)=Q(L1,M,N3)
5291      E(LMAX,M,N3)=E(L1,M,N3)
5292      220 CONTINUE
5293      230 IF (MDFS.EQ.O) GO TO 280
5294      IF (NVC.NE.1) GO TO 240
5295      IF (MDFS.GT.MVCB.AND.MDFS.LT.MVCT) GO TO 270
5296      GO TO 250
5297      240 IF (MDFS.LT.MVCB.OR.MDFS.GT.MVCT) GO TO 270
5298      250 DO 260 L=LDFSS,LDFSF
5299      QL(L,N3)=Q(L,MDFS-1,N3)+YIL*(Q(L,MDFS-1,N3)-Q(L,MDFS-2,N3))
5300      EL(L,N3)=E(L,MDFS-1,N3)+YIL*(E(L,MDFS-1,N3)-E(L,MDFS-2,N3))
5301      IF (NOSLIP.NE.O) QL(L,N3)=O.O
5302      260 CONTINUE
5303      270 IF (LDFSF.NE.LMAX) GO TO 280
5304      QL(LMAX,N3)=QL(L1,N3)
5305      EL(LMAX,N3)=EL(L1,N3)
5306 C
5307      280 DO 290 L=1,LMAX
5308      IF (Q(L,1,N3).LT.O.O) Q(L,1,N3)=QLOW
5309      IF (E(L,1,N3).LT.O.O) E(L,1,N3)=ELOW
5310      IF (Q(L,MMAX,N3).LT.O.O) Q(L,MMAX,N3)=QLOW
5311      IF (E(L,MMAX,N3).LT.O.O) E(L,MMAX,N3)=ELOW
5312      IF (MDFS.EQ.O) GO TO 290
5313      IF (Q(L,MDFS,N3).LT.O.O) Q(L,MDFS,N3)=QLOW
5314      IF (E(L,MDFS,N3).LT.O.O) E(L,MDFS,N3)=ELOW
5315      IF (QL(L,N3).LT.O.O) QL(L,N3)=QLOW
5316      IF (EL(L,N3).LT.O.O) EL(L,N3)=ELOW
5317      290 CONTINUE
5318      RETURN
5319      END

```

```

5320      SUBROUTINE INTER
5321 C
5322 C      .....
5323 C
5324 C      THIS SUBROUTINE CALCULATES THE INTERIOR MESH POINTS
5325 C
5326 C      .....
5327 C
5328 *CALL MCC
5329      IP=1
5330      ATERM=0.0
5331      MIS=1
5332      IF (NGCB.NE.0) MIS=2
5333      MIF=M1
5334      IF (ICHAR.NE.1) GO TO 200
5335 C
5336 C      COMPUTE THE TENTATIVE SOLUTION AT T+DT
5337 C
5338      IF (IVC.EQ.0) GO TO 10
5339      IF (NVC.EQ.1) GO TO 10
5340      MIS=MVCB
5341      MIF=MVCT+1
5342      IF (MVCB.EQ.1.AND.NGCB.NE.0) MIS=2
5343      IF (MIF.GE.MMAX) MIF=M1
5344 C
5345 C      BEGIN THE L OR X DO LOOP
5346 C
5347      10 DO 190 L=2,L1
5348          LMAP=L
5349          LDFS=0
5350          IF (L.GE.LDFSS.AND.L.LE.LDFSF) LDFS=1
5351 C
5352 C      BEGIN THE M OR Y DO LOOP
5353 C
5354          DO 180 M=MIS,MIF
5355          IF (IVC.EQ.0) GO TO 20
5356          IF (NVC.NE.1) GO TO 20
5357          IF (M.LE.MVCB.AND.MVCB.NE.1) GO TO 20
5358          IF (M.GT.MVCT) GO TO 20
5359          GO TO 180
5360      20 IF (M.EQ.MDFS.AND.LDFS.EQ.1) GO TO 180
5361          MMAP=M
5362          CALL MAP
5363          OM=OM1
5364          AL=AL3
5365          BE=BE3
5366          DE=DE3
5367          UB=U(L,M,N1)
5368          VB=V(L,M,N1)
5369          PB=P(L,M,N1)
5370          ROB=RO(L,M,N1)
5371          ROR=1.0/ROB
5372          ASB=GAM*A*PB*ROR
5373          QB=C(L,M,N1)
5374          EB=E(L,M,N1)
5375          IF (M.NE.1) GO TO 60
5376 C
5377 C      CALCULATE THE QUANTITIES FOR M=1
5378 C
5379          DUDX=(UB-U(L-1,M,N1))*DXR
5380          DPDX=(PB-P(L-1,M,N1))*DXR
5381          DRDXX=(ROB-RO(L-1,M,N1))*DXR
5382          DVDY=(4.0*V(L,2,N1)-V(L,3,N1))*0.5*DVR
5383          IF (ITM.LE.1) GO TO 30
5384          DQDX=(QB-Q(L-1,M,N1))*DXR
5385          DEDX=(EB-E(L-1,M,N1))*DXR
5386      30 V(L,M,N3)=0.0
5387          URHS=-UB*OM*EJDX-OM*DPDX*ROR+QUT(L,M)
5388          RORHS=-UB*OM*DRDXX-RQB*OM*DUDX-FLOAT(1+NOIM)*ROB*BE+DVDY+CROT(L,M)
5389          PRHS=-UB*OM*DPDX+ASS*(RORHS+UB*OM*DRDXX)+QPT(L,M)
5390          IF (ITM.LE.1) GO TO 170
5391          IF (JB.GE.0.0) GO TO 40

```

```

5392      DQDX=(Q(L+1,M,N1)-QB)*DXR
5393      DEDX=(E(L+1,M,N1)-EB)*DXR
5394      OM=OM2
5395      40 GRHS=-UB*OM+DQDX+QQT(L,M)
5396      Q(L,M,N3)=QB+GRHS*DT
5397      IF (Q(L,M,N3).LT.QLOW) Q(L,M,N3)=QLOW
5398      IF (ITM.EQ.2) GO TO 170
5399      ERHS=-UB*OM+DEDX+QQT(L,M)
5400      E(L,M,N3)=EB+ERHS*DT
5401      IF (MDFS.NE.O.AND.LDFS.EQ.O) GO TO 50
5402      IF (Q(L,M,N3).LT.BFST+FSQ(M)) Q(L,M,N3)=BFST+FSQ(M)
5403      IF (E(L,M,N3).LT.BFST+FSE(M)) E(L,M,N3)=BFST+FSE(M)
5404      50 IF (E(L,M,N3).GT.ELGW) GO TO 170
5405      Q(L,M,N3)=QLCW
5406      E(L,M,N3)=ELCW
5407      GO TO 170
5408 C
5409 C      CALCULATE THE QUANTITIES FOR M NOT EQUAL TO 1
5410 C
5411      60 IF (IVC.EQ.O) GO TO 70
5412      IF (NVC.EQ.1.OR.M.NE.MVCT+1) GO TO 70
5413 C
5414 C      LINEAR INTERPOLATION IN TIME FOR M=MVCT+1
5415 C
5416      UB=UU1(L)+RIND*(UU2(L)-UU1(L))
5417      VB=VV1(L)+RIND*(VV2(L)-VV1(L))
5418      PB=PP1(L)+RIND*(PP2(L)-PP1(L))
5419      ROB=RORO1(L)+RIND*(RORO2(L)-RORO1(L))
5420      ROR=1.O/ROB
5421      ASB=GAMMA*PB*ROR
5422      ULM=UU1(L-1)+RIND*(UU2(L-1)-UU1(L-1))
5423      VLM=VV1(L-1)+RIND*(VV2(L-1)-VV1(L-1))
5424      PLM=PP1(L-1)+RIND*(PP2(L-1)-PP1(L-1))
5425      ROLM=RORO1(L-1)+RIND*(RORO2(L-1)-RORO1(L-1))
5426      IF (ITM.LE.1) GO TO 80
5427      QB=QU1(L)+RIND*(QU2(L)-QU1(L))
5428      EB=EE1(L)+RIND*(EE2(L)-EE1(L))
5429      QLM=QQ1(L-1)+RIND*(QQ2(L-1)-QQ1(L-1))
5430      ELM=EE1(L-1)+RIND*(EE2(L-1)-EE1(L-1))
5431      GO TO 80
5432 C
5433      70 ULM=U(L-1,M,N1)
5434      VLM=V(L-1,M,N1)
5435      PLM=P(L-1,M,N1)
5436      ROLM=RO(L-1,M,N1)
5437      QLM=Q(L-1,M,N1)
5438      ELM=E(L-1,M,N1)
5439      IF (M.NE.MDFS.OR.L.NE.LDFS+1) GO TO 80
5440      ULM=O.5*(ULM+UL(L-1,N1))
5441      VLM=O.5*(VLM+VL(L-1,N1))
5442      PLM=O.5*(PLM+PL(L-1,N1))
5443      ROLM=O.5*(ROLM+ROL(L-1,N1))
5444      IF (ITM.LE.1) GO TO 80
5445      QLM=O.5*(QLM+QL(L-1,N1))
5446      ELM=O.5*(ELM+EL(L-1,N1))
5447      80 UVB=UB*AL+VB*BE+DE
5448      IF (NDIM.NE.O) ATERM=ROB*VB/YD
5449      DUOX=(UB-ULM)*DXR
5450      DVDX=(VB-VLM)*DXR
5451      DPDX=(PB-PLM)*DXR
5452      DRDX=(ROB-ROLM)*DXR
5453      IF (ITM.LE.1) GO TO 90
5454      DQDX=(QB-QLM)*DXR
5455      DEDX=(EB-ELM)*DXR
5456      90 IF (IVC.EQ.O) GO TO 110
5457      IF (NVC.EQ.1.OR.M.NE.MVCB) GO TO 110
5458 C
5459 C      LINEAR INTERPOLATION IN TIME FOR M=MVCB
5460 C
5461      UMM=U(L,M-1,NN1)+RIND*(U(L,M-1,NN3)-U(L,M-1,NN1))
5462      VMM=V(L,M-1,NN1)+RIND*(V(L,M-1,NN3)-V(L,M-1,NN1))
5463      PMM=P(L,M-1,NN1)+RIND*(P(L,M-1,NN3)-P(L,M-1,NN1))

```

```

5464 ROMM=RO(L,M-1,NN1)+RIND*(RO(L,M-1,NN3)-RO(L,M-1,NN1))
5465 IF (ITM.LE.1) GO TO 100
5466 QMM=Q(L,M-1,NN1)+RIND*(C(L,M-1,NN3)-Q(L,M-1,NN1))
5467 EMM=E(L,M-1,NN1)+RIND*(E(L,M-1,NN3)-E(L,M-1,NN1))
5468 C
5469 100 DUDY=(UB-UMM)*DYR
5470 DVDY=(VB-VMM)*DYR
5471 DPDY=(PB-PMM)*DYR
5472 DRODY=(ROB-ROMM)*DYR
5473 IF (ITM.LE.1) GO TO 120
5474 DODY=(OB-QMM)*DYR
5475 DEDY=(EB-EMM)*DYR
5476 GO TO 120
5477 110 DUDY=(UB-U(L,M-1,N1))*DYR
5478 DVDY=(VB-V(L,M-1,N1))*DYR
5479 DPDY=(PB-P(L,M-1,N1))*DYR
5480 DRODY=(ROB-RO(L,M-1,N1))*DYR
5481 IF (ITM.LE.1) GO TO 120
5482 DODY=(OB-Q(L,M-1,N1))*DYR
5483 DEDY=(EB-E(L,M-1,N1))*DYR
5484 C
5485 C SPECIAL FORM OF THE EQUATIONS USED BY THE QUICK SOLVER
5486 C
5487 120 IF (IOSD.EQ.O.OR.NVC.EQ.1) GO TO 130
5488 IF (M.EQ.MVCB.OR.M.GE.MVCT) GO TO 130
5489 ALS=SQRT(AL*AL+BE*BE)
5490 RALS=1.O/ALS
5491 AB=SQRT(ASB)
5492 ABR=AL/BE
5493 UVBP=UVB+ALS*AB
5494 UVBM=UVB-ALS*AB
5495 USL=-UVB*DUDY+ABR*UVB*DVDY-UB*OM*(DUDX-ABR*DVDX)-OM*DPDX+ROR*QUT(L
5496 1,M)-ABR*QVT(L,M)
5497 PMLP=-UB*OM*DPDX-ROB*ASB*OM*DUDX-ASB*ATERM-ROB*AB*OM*RALS*(AL*(UB
5498 1*DUDX+DPDX+ROR)+BE*UB*DVDX)+OPT(L,M)+ASB*OROT(L,M)+ROB*AB*RALS*
5499 2*(AL*QUT(L,M)+BE*QVT(L,M))
5500 PMLM=-UB*OM*DPDX-ROB*ASB*OM*DUDX-ASB*ATERM+ROB*AB*OM*RALS*(AL*(UB
5501 1*DUDX+DPDX+ROR)+BE*UB*DVDX)+OPT(L,M)+ASB*OROT(L,M)-ROB*AB*RALS*
5502 2*(AL*QUT(L,M)+BE*QVT(L,M))
5503 PMLP1=-UVBP*DPDYQS(L,M,1)-ROB*AB*RALS*UVBP*(AL*DUDYQS(L,M,1)+BE
5504 1*DVDYQS(L,M,1))+PMLP
5505 PMLM1=-UVBM*DPDYQS(L,M,2)+ROB*AB*RALS*UVBM*(AL*DUDYQS(L,M,2)+BE
5506 1*DVDYQS(L,M,2))+PMLM
5507 VRHS=(2.O*ROB*AB*AL*RALS+USL+PMLM1-PMLP1)/(2.O*ROB*AB*ALS/BE)
5508 PRHS=O.5*(PMLP1+PMLM1)
5509 URHS=ABR*VRHS+USL
5510 RORHS=-UB*OM*DRODX-UVB*DRODY+(PRHS+UB*OM*DPDX+UVB*DPDY-QPT(L,M))
5511 1/ASB
5512 GO TO 140
5513 C
5514 C REGULAR FORM OF THE EQUATIONS
5515 C
5516 130 URHS=-UB*OM*DUDX-UVB*DUDY-(OM*DPDX+AL*DPDY)*ROR*QUT(L,M)
5517 VRHS=-UB*OM*DVDX-UVB*DVDY-BE*DPDY+ROR*QVT(L,M)
5518 RORHS=-UB*OM*DPDX-UVB*DRODY-ROB*(OM*DUDX+AL*DUDY+BE*DVDY)-ATERM
5519 1+OROT(L,M)
5520 PRHS=-UB*OM*DPDX-UVB*DPDY+ASB*(RORHS+UB*OM*DRODX+UVB*DRODY)+OPT(L
5521 1,M)
5522 C
5523 140 V(L,M,N3)=VB+VRHS*DT
5524 IF (ITM.LE.1) GO TO 170
5525 IF (UB.GE.O.O) GO TO 150
5526 DQDX=(Q(L+1,M,N1)-QB)*DXR
5527 DEDX=(E(L+1,M,N1)-EB)*DXR
5528 OM=OM2
5529 150 QRHS=-UB*OM*DQDX-UVB*DQDY+QOT(L,M)
5530 Q(L,M,N3)=QB+QRHS*DT
5531 IF (Q(L,M,N3).LT.QLOW) Q(L,M,N3)=QLOW
5532 IF (ITM.EQ.2) GO TO 170
5533 ERHS=-UB*OM*DEDX-UVB*DEDY+QET(L,M)
5534 E(L,M,N3)=EB+ERHS*DT
5535 IF (MDFS.NE.O.AND.LDFS.EQ.O) GO TO 160

```

```

5536      IF (Q(L,M,N3).LT.BFST*FSO(M)) Q(L,M,N3)=BFST*FSO(M)
5537      IF (E(L,M,N3).LT.3FST*FSE(M)) E(L,M,N3)=BFST*FSE(M)
5538 160 IF (E(L,M,N3).GT.ELW) GO TO 170
5539      Q(L,M,N3)=QLOW
5540      E(L,M,N3)=ELW
5541 170 U(L,M,N3)=UB*URHS*DT
5542      P(L,M,N3)=PB*PRHS*DT
5543      RO(L,M,N3)=ROB*RORHS*DT
5544      IF (P(L,M,N3).LE.O.O) P(L,M,N3)=PLOW*PC
5545      IF (RO(L,M,N3).LE.O.O) RO(L,M,N3)=ROLOW/G
5546 180 CONTINUE
5547 190 CONTINUE
5548      RETURN
5549 C
5550 C      COMPUTE THE FINAL SOLUTION AT T+DT
5551 C
5552 200 IF (IVC.EQ.O) GO TO 210
5553      IF (NVC.EQ.1) GO TO 210
5554      MIS=MVCB
5555      MIF=MVCT
5556      IF (NVC.EQ.1.AND.NGCB.NE.O) MIS=2
5557      IF (MIF.EQ.MMAX) MIF=M1
5558 C
5559 C      BEGIN THE L OR X DO LOOP
5560 C
5561 210 DO 39C L=2,L1
5562      LMAP=L
5563      LDFS=O
5564      IF (L.GE.LDFS.AND.L.LE.LDFS) LDFS=1
5565      UOLD=U(L,1,N3)
5566      VOLD=V(L,1,N3)
5567      POLD=P(L,1,N3)
5568 C
5569 C      BEGIN THE M OR Y DO LOOP
5570 C
5571      DO 380 M=MIS,MIF
5572      IF (IVC.EQ.O) GO TO 220
5573      IF (NVC.NE.1) GO TO 220
5574      IF (M.LT.MVCB) GO TO 220
5575      IF (M.GT.MVCT) GO TO 220
5576      GO TO 380
5577 220 IF (M.EQ.MDFS.AND.LDFS.EQ.1) GO TO 380
5578      MMAP=M
5579      CALL MAP
5580      OM=OM2
5581      AL=AL4
5582      BE=BE4
5583      DE=DE4
5584      BED=BE3
5585      UB=U(L,M,N3)
5586      VB=V(L,M,N3)
5587      PB=P(L,M,N3)
5588      ROB=RO(L,M,N3)
5589      ROR=1.O/ROB
5590      ASB=GAMMA*PB*ROR
5591      QB=Q(L,M,N3)
5592      EB=E(L,M,N3)
5593      IF (M.NE.1) GO TO 260
5594 C
5595 C      CALCULATE THE QUANTITIES FOR M=1
5596 C
5597      DUDX=(U(L+1,M,N3)-UB)*DXR
5598      DPDX=(P(L+1,M,N3)-PB)*DXR
5599      DRODX=(RO(L+1,M,N3)-ROB)*DXR
5600      DVDY=(4.O*V(L,2,N3)-V(L,3,N3))*O.5*DYR
5601      IF (ITM.LE.1) GO TO 230
5602      DQDX=(Q(L+1,M,N3)-QB)*DXR
5603      DEDX=(E(L+1,M,N3)-EB)*DXR
5604 230 V(L,M,N3)=O.O
5605      URHS=-UB*OM*DUCX-CM*DPDX-ROR*OUT(L,M)
5606      RORHS=-UB*OM*DRCDX-ROB*CM*DUDX-FLOAT(*+NDIM)*ROB*BE*DVDY+QRCT(L,M)
5607      PRHS=-UB*OM*DPDX+ASB*(RORHS+UB*CM*DRCDX)+QPT(L,M)

```



```

5608     IF (ITM.LE.1) GO TO 370
5609     IF (U(L,M,N1).LT.0.0) GO TO 240
5610     DODX=(QB-Q(L-1,M,N3))*DXR
5611     DEDX=(EB-E(L-1,M,N3))*DXR
5612     OM=OM1
5613 240 QRHS=-UB*OM*DODX+OOT(L,M)
5614     Q(L,M,N3)=0.5*(Q(L,M,N1)+Q(L,M,N3)+QRHS*DT)
5615     IF (Q(L,M,N3).LT.QLOW) Q(L,M,N3)=QLOW
5616     IF (ITM.EQ.2) GO TO 370
5617     ERHS=-UB*OM*DEDX+OET(L,M)
5618     E(L,M,N3)=0.5*(E(L,M,N1)+E(L,M,N3)+ERHS*DT)
5619     IF (MDFS.NE.0.AND.LDFS.EQ.0) GO TO 250
5620     IF (Q(L,M,N3).LT.BFST+FSQ(M)) Q(L,M,N3)=BFST+FSQ(M)
5621     IF (E(L,M,N3).LT.BFST+FSE(M)) E(L,M,N3)=BFST+FSE(M)
5622 250 IF (E(L,M,N3).GT.ELOW) GO TO 370
5623     Q(L,M,N3)=QLOW
5624     E(L,M,N3)=ELOW
5625     GO TO 370
5626 C
5627 C     CALCULATE THE QUANTITIES FOR M NOT EQUAL TO 1
5628 C
5629 260 IF (ND!M.NE.0) ATERM=ROB*VB/YP
5630     ULP=U(L+1,M,N3)
5631     VLP=V(L+1,M,N3)
5632     PLP=P(L+1,M,N3)
5633     ROLP=RO(L+1,M,N3)
5634     QLM=Q(L-1,M,N3)
5635     ELM=E(L-1,M,N3)
5636     IF (M.NE.MDFS.OR.L.NE.LDFS-1) GO TO 270
5637     ULP=0.5*(ULP+UL(L+1,N3))
5638     VLP=0.5*(VLP+VL(L+1,N3))
5639     PLP=0.5*(PLP+PL(L+1,N3))
5640     ROLP=0.5*(ROLP+ROL(L+1,N3))
5641 270 IF (M.NE.MDFS.OR.L.NE.LDFS+1) GO TO 280
5642     IF (ITM.LE.1) GO TO 280
5643     QLM=0.5*(QLM+QL(L-1,N3))
5644     ELM=0.5*(ELM+EL(L-1,N3))
5645 280 UVB=UB*AL+V3*BE+DE
5646     DUDX=(ULP-UB)*DXR
5647     DVDX=(VLP-VB)*DXR
5648     DPDY=(PLP-PB)*DXR
5649     DRODX=(ROLP-ROB)*DXR
5650     IF (ITM.LE.1) GO TO 290
5651     DQDX=(QB-QLM)*DXR
5652     DEDX=(EB-ELM)*DXR
5653 290 DUDY=(U(L,M+1,N3)-UB)*DYR
5654     DVDY=(V(L,M+1,N3)-VB)*DYR
5655     DPDY=(P(L,M+1,N3)-PB)*DYR
5656     DRODY=(RO(L,M+1,N3)-ROB)*DYR
5657     IF (ITM.LE.1) GO TO 300
5658     DODY=(Q(L,M+1,N3)-QB)*DYR
5659     DEDY=(E(L,M+1,N3)-EB)*DYR
5660 C
5661 C     SPECIAL FORM OF THE EQUATIONS USED BY THE QUICK SOLVER
5662 C
5663 300 IF (IOSD.EQ.0.OR.NVC.EQ.1) GO TO 320
5664     I= (M.EQ.MVCB.OR.M.EQ.MVCT) GO TO 320
5665     ALS=SQRT(AL*AL+BE*BE)
5666     RALS=1./ALS
5667     AB=SQRT(ASB)
5668     ABR=AL/BE
5669     UVBP=UVS+ALS*AB
5670     UVBM=UVB-ALS*AB
5671     BER=BED/BE
5672     DUDY1=(UB-UOLD)*DYR+BER
5673     DVDY1=(VB-VOLD)*DYR+BER
5674     DPDY1=(PB-POLD)*DYR+BER
5675     IF (MDFS.EQ.0) GO TO 310
5676     IF (M.NE.MDFS:1.OR.LDFS.EQ.0) GO TO 310
5677     DUDY1=(US-UL(L,N3))*DYR+BER
5678     DVDY1=(VB-VL(L,N3))*DYR+BER
5679     DPDY1=(PB-PL(L,N3))*DYR+BER

```

```

5680 310 USL=-UVB*DUDY+ABR*UVB*DVOY-UB*OM*(DUDX-A2R*DVDX)-OM*DPDX+ROR*QUT(L
5681 1,M)-ABR*QVT(L,M)
5682 PMLP=-UB*OM*DPDX-ROB*ASB*OM*DUDX-ASB*ATERM-ROB*AB*OM*RALS*(AL*(UB
5683 1*DUDX+DPOX+ROR)+BE*UB*DVDX)+OPT(L,M)+ASB*QROT(L,M)+ROB*AB*RALS*
5684 2*(AL*QUT(L,M)+BE*QVT(L,M))
5685 PMLM=-UB*OM*DPDX-ROB*ASB*OM*DUDX-ASB*ATERM+ROB*AB*OM*RALS*(AL*(UB
5686 1*DUDX+DPOX+ROR)+BE*UB*DVDX)+OPT(L,M)+ASB*QROT(L,M)-ROB*AB*RALS*
5687 2*(AL*QUT(L,M)+BE*QVT(L,M))
5688 PMLP1=-UVBP*DPDY1-ROB*AB*RALS*UVBP*(AL*DUDY1+BE*DVOY1)+PMLP
5689 PMLM1=-UVBM*DPDY+ROB*AB*RALS*UVBM*(AL*DUDY+BE*DVOY)+PMLM
5690 VRHS=-(2.0*ROB*AB*AL*RALS+USL+PMLM1-PMLP1)/(2.0*ROB*AB*ALS/BE)
5691 PRHS=0.5*(PMLP1+PMLM1)
5692 URHS=ABR*VRHS+USL
5693 RORHS=-UB*OM*DRODX-UVB*DRODY+(PRHS+UB*OM*DPDX+UVB*DPDY-OPT(L,M))
5694 1/ASB
5695 GO TO 330
5696 C
5697 C REGULAR FORM OF THE EQUATIONS
5698 C
5699 320 URHS=-UB*OM*DUDX-UVB*DUDY-(OM*DPDX+AL*DPDY)+ROR*QUT(L,M)
5700 VRHS=-UB*OM*DVDX-UVB*DVDY-BE*DPDY+ROR*QVT(L,M)
5701 RORHS=-UB*OM*DRODX-UVB*DRODY-ROB*(OM*DUDX+AL*DUDY+BE*DVOY)-ATERM
5702 1+QROT(L,M)
5703 PRHS=-UB*OM*DPDX-UVB*DPDY+ASB*(RORHS+UB*OM*DRODX+UVB*DRODY)+OPT(L
5704 1,M)
5705 C
5706 330 IF (I2SD.EQ.O.OR.NVC.EQ.1) GO TO 340
5707 UOLD=U(L,M,N3)
5708 VOLD=V(L,M,N3)
5709 POLO=P(L,M,N3)
5710 340 V(L,M,N3)=0.5*(V(L,M,N1)+V(L,M,N3)+VRHS*DT)
5711 IF (ITM.LE.1) GO TO 370
5712 IF (U(L,M,N1).GE.O.O) GO TO 350
5713 DQDX=(Q(L+1,M,N3)-QB)*DXR
5714 DEDX=(E(L+1,M,N3)-EB)*DXR
5715 OM=OM1
5716 350 QRHS=-UB*OM*DQDX-UVB*DQDY+QQT(L,M)
5717 Q(L,M,N3)=0.5*(Q(L,M,N1)+Q(L,M,N3)+QRHS*DT)
5718 IF (Q(L,M,N3).LT.QLOW) Q(L,M,N3)=QLOW
5719 IF (ITM.EQ.2) GO TO 370
5720 ERHS=-UB*OM*DEDX-UVB*DEDY+OET(L,M)
5721 E(L,M,N3)=0.5*(E(L,M,N1)+E(L,M,N3)+ERHS*DT)
5722 IF (MDFS.NE.O.AND.LDFS.EQ.O) GO TO 360
5723 IF (Q(L,M,N3).LT.BFST+F5Q(M)) Q(L,M,N3)=BFST+F5Q(M)
5724 IF (E(L,M,N3).LT.BFST+F5E(M)) E(L,M,N3)=BFST+F5E(M)
5725 360 IF (E(L,M,N3).GT.ELOW) GO TO 370
5726 Q(L,M,N3)=QLOW
5727 E(L,M,N3)=ELOW
5728 370 U(L,M,N3)=0.5*(U(L,M,N1)+U(L,M,N3)+URHS*DT)
5729 P(L,M,N3)=0.5*(P(L,M,N1)+P(L,M,N3)+PRHS*DT)
5730 RO(L,M,N3)=0.5*(RO(L,M,N1)+RO(L,M,N3)+RORHS*DT)
5731 IF (P(L,M,N3).LE.O.O) P(L,M,N3)=PLOW*PC
5732 IF (RO(L,M,N3).LE.O.O) RO(L,M,N3)=ROLOW/G
5733 380 CONTINUE
5734 390 CONTINUE
5735 RETURN
5736 END

```

```

5737      SUBROUTINE WALL
5738 C
5739 C      .....
5740 C
5741 C      THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE
5742 C      WALL, FREE-JET BOUNDARY, CENTERBODY AND DUAL FLOW SPACE WALLS
5743 C
5744 C      .....
5745 C
5746 C      *CALL MCC
5747      IP=1
5748      Y2=0.0
5749      Y20=0.0
5750      NJSL=NOSLIP
5751      IF (IB.EQ.1.AND.(WALL.NE.0) NOSLIP)
5752      IF (N.EQ.1.AND.(JFLAG.NE.0) DELY=0.0001*YW(LJET-1)
5753      XWID=0.0
5754      ATERM2=0.0
5755      ATERM3=0.0
5756      IF (IB.EQ.1) GO TO 10
5757      IF (IB.GT.2) GO TO 20
5758      Y3=0.0
5759      MDUM=1
5760      MDUM1=2
5761      SIGN=-1.0
5762      GO TO 40
5763 10 Y3=1.0
5764      MDUM=MMAX
5765      MDUM1=M1
5766      SIGN=1.0
5767      GO TO 40
5768 20 Y3=Y(MDFS)
5769      MDUM=MDFS
5770      IF (IB.EQ.4) GO TO 30
5771      MDUM1=MDFS-1
5772      SIGN=1.0
5773      GO TO 40
5774 30 MDUM1=MDFS+1
5775      SIGN=-1.0
5776 40 DYS=SIGN*DYS
5777      MMAP=MDUM
5778 C
5779 C      BEGIN THE L OR X CJ LOOP
5780 C
5781      CJ 700 L=2,L1
5782      LDFS=0
5783      IF (L.GE.LDFFS.AND.L.LE.LDFFS) LDFS=1
5784      IF (IB.GE.3.AND.LDFS.EQ.0) GO TO 700
5785      LMAP=L
5786      CALL MAP
5787      AL=AL3
5788      BE=BE3
5789      DE=DE3
5790 C
5791      IF (JFLAG.EQ.0) GO TO 70
5792      IF (IB.NE.1) GO TO 70
5793      XWID=XWI(L)
5794      IF (ICAR.EQ.1) GO TO 50
5795 C
5796 C      USE THE DUMMY ARRAYS TO MANIPULATE THE ONE-SIDED SOLUTIONS
5797 C      FOR THE FREE-JET OR SHARP EXPANSION CORNER CASES
5798 C
5799      IF (L.NE.LJET-2) GO TO 50
5800      U(L+1,MDUM,N3)=UD(3)
5801      V(L+1,MDUM,N3)=VD(3)
5802      P(L+1,MDUM,N3)=PD(3)
5803      RO(L+1,MDUM,N3)=ROD(3)
5804      GO TO 70
5805 50 IF (L.NE.LJET-1) GO TO 60
5806      IF (ICAR.EQ.1) UOLD=U(L,MDUM,N1)
5807      U(L,MDUM,N1)=UD(1)
5808      V(L,MDUM,N1)=VD(1)

```

F

```

5809      P(L,MOUM,N1)=PD(1)
5810      RO(L,MOUM,N1)=ROD(1)
5811      GO TO 70
5812      EO IF (L.NE.LJET) GO TO 70
5813      U(L-1,MOUM,N1)=UD(2)
5814      V(L-1,MOUM,N1)=VD(2)
5815      P(L-1,MOUM,N1)=PD(2)
5816      RO(L-1,MOUM,N1)=ROD(2)
5817      C
5818      70 U1=U(L,MOUM,N1)
5819      V1=V(L,MOUM,N1)
5820      P1=P(L,MOUM,N1)
5821      RO1=RO(L,MOUM,N1)
5822      U2=U1
5823      V2=V1
5824      A1=SQRT(GAMMA*P1/RO1)
5825      A2=A1
5826      IF (ICHR.NE.1) GO TO 80
5827      U3=U1
5828      V3=V1
5829      P3=P1
5830      RO3=RO1
5831      A3=A1
5832      GO TO 90
5833      80 U3=U(L,MOUM,N3)
5834      V3=V(L,MOUM,N3)
5835      P3=P(L,MOUM,N3)
5836      RO3=RO(L,MOUM,N3)
5837      A3=SQRT(GAMMA*P3/RO3)
5838      C
5839      C      CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
5840      C
5841      90 EU=(U1-U(L,MOUM,N1))*DYS
5842      BV=(V1-V(L,MOUM,N1))*DYS
5843      EP=(P1-P(L,MOUM,N1))*DYS
5844      BRO=(RO1-RO(L,MOUM,N1))*DYS
5845      BOUT=(OUT(L,MOUM)-OUT(L,MOUM1))*DYS
5846      BQVT=(QVT(L,MOUM)-QVT(L,MOUM1))*DYS
5847      BOPT=(OPT(L,MOUM)-OPT(L,MOUM1))*DYS
5848      BOROT=(OROT(L,MOUM)-OROT(L,MOUM1))*DYS
5849      CU=U1-EU*Y3
5850      CV=V1-BV*Y3
5851      CP=P1-EP*Y3
5852      CRO=RO1-BRO*Y3
5853      COUT=OUT(L,MOUM)-BOUT*Y3
5854      CQVT=QVT(L,MOUM)-BQVT*Y3
5855      COPT=OPT(L,MOUM)-BOPT*Y3
5856      COROT=OROT(L,MOUM)-BOROT*Y3
5857      C
5858      C      CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
5859      C      COEFFICIENTS
5860      C
5861      DU=(U1-U(L-1,MOUM,N1))*DXR
5862      DV=(V1-V(L-1,MOUM,N1))*DXR
5863      DP=(P1-P(L-1,MOUM,N1))*DXR
5864      DRO=(RO1-RO(L-1,MOUM,N1))*DXR
5865      DU1=(U(L,MOUM,N1)-U(L-1,MOUM,N1))*DXR
5866      DV1=(V(L,MOUM,N1)-V(L-1,MOUM,N1))*DXR
5867      DP1=(P(L,MOUM,N1)-P(L-1,MOUM,N1))*DXR
5868      DRO1=(RO(L,MOUM,N1)-RO(L-1,MOUM,N1))*DXR
5869      DUO=(DU+DU1)*DYS
5870      DVO=(DV+DV1)*DYS
5871      DPO=(DP+DP1)*DYS
5872      DROO=(DRO+DRO1)*DYS
5873      COU=DU+DUO*Y3
5874      COV=DV+DVO*Y3
5875      COP=DP+DPO*Y3
5876      CROO=DRO+DROO*Y3
5877      C
5878      C      CALCULATE Y2
5879      C
5880      ALS=SOPTAL*AL+BE*BE)

```

```

5881      UV3=U3*AL+V3*BE+DE
5882      DO 130 ILL=1,3
5883      UV2=U2*AL+V2*BE+DE
5884      Y2=Y3-(UV2+SIGN*ALS*A2+UV3+SIGN*ALS*A3)*DT*0.5
5885 C
5886      IF (IQSD.EQ.0 OR.NVC.EQ.1) GO TO 100
5887      IF (IB.EQ.1.AND.Y2.LT.Y(M1)) Y2=Y(M1)
5888      IF (IB.EQ.2.AND.Y2.GT.Y(2)) Y2=Y(2)
5889      IF (MDFS.EQ.0) GO TO 100
5890      IF (IB.EQ.3.AND.Y2.LT.Y(MDFS-1)) Y2=Y(MDFS-1)
5891      IF (IB.EQ.4.AND.Y2.GT.Y(MDFS+1)) Y2=Y(MDFS+1)
5892 C
5893 100 IF (IWALL.EQ.0 OR.IB.NE.1) GO TO 110
5894      UV1=U1*AL+V1*BE
5895      Y1=Y3-(UV1+UV3)*DT*0.5
5896 C
5897 C      INTERPOLATE FOR THE PROPERTIES
5898 C
5899      U1=BU*Y1+CU
5900      V1=BV*Y1+CV
5901      P1=BP*Y1+CP
5902      R01=BRO*Y1+CRO
5903 110 U2=BU*Y2+CU
5904      V2=BV*Y2+CV
5905      P2=BP*Y2+CP
5906      R02=BRO*Y2+CRO
5907      AD=GAMMA*P2/R02
5908      IF (AD.GT.0.0) GO TO 120
5909      NP=N+NSTART
5910      WRITE (6,710) NP,L,MDUM,NVC
5911      IERR=1
5912      RETURN
5913 120 A2=SQRT(AD)
5914 130 CONTINUE
5915      QUT2=BQUT*Y2+CQUT
5916      QVT2=BQVT*Y2+CQVT
5917      QPT2=BOPT*Y2+CQPT
5918      QROT2=BQROT*Y2+CQROT
5919 C
5920 C      INTERPLATE FOR THE CROSS DERIVATIVES
5921 C
5922      IF (IWALL.EQ.0 OR.IB.NE.1) GO TO 140
5923      DU1=BDU*Y1+CDU
5924      DV1=BDV*Y1+CDV
5925      DP1=BDP*Y1+CDP
5926      DR01=BDRO*Y1+CDRO
5927      GO TO 150
5928 140 DU1=DU
5929      DV1=DV
5930      DP1=DP
5931      DR01=DR0
5932 150 DU2=BDU*Y2+CDU
5933      DV2=BDV*Y2+CDV
5934      DP2=BDP*Y2+CDP
5935      DR02=BDRO*Y2+CDRO
5936 C
5937 C      CALCULATE THE PSI TERMS
5938 C
5939      IF (INDIM.EQ.0) GO TO 100
5940      IF (IB.EQ.2) GO TO 160
5941      ATERM2=R02*V2/(1-P*(Y3-Y2)/BE)
5942      GO TO 180
5943 160 IF (YCB(L).EQ.0.0) GO TO 170
5944      ATERM2=R02*V2/(YCB(L)+Y2/BE)
5945      GO TO 150
5946 170 ATERM2=R02*BE*(L,2,N1)*DYS
5947 180 PSI21=-U1*DM1*DU1-DM1*DP1/R01
5948      PSI21=-U1*DM1*DV1
5949      PSI41=-U1*DM1*DP1+A1*A1*U1*DM1*DR01
5950      PSI12=-U2*DM1*DR02-R02*DM1*DU2-ATERM2
5951      PSI22=-U2*DM1*DM2-DM1*DP2/R02
5952      PSI32=-U2*DM1*DV2

```

```

5153      PSI42=-U2*OM1+DP2+A2*A2+U2*OM1+DRO2
5154 C
5155 C      CALCULATE THE QUANTITIES FOR THE QUICK SOLVER
5156 C
5157      IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 320
5158      ALD=AL
5159      BED=BE
5160      DED=DE
5161      OMQ=OM2
5162      YPD=YP
5163      ILLI=O
5164      MMQ=O
5165      DO 300 ILL=1,ILLQ5
5166      IF (ILLI.NE.O) GO TO 210
5167      IF (ILL.NE.1) GO TO 190
5168      UVAO=(UV3*SIGN*AL5*A2)*DT
5169      Y200=Y3
5170      FY3=-UVAO
5171      Y2=Y(MOUM1)
5172      GO TO 250
5173 190 UVAVG=O.5*{(U2+U3)*ALAVG+(V2+V3)*BEAVG)+DEAVG
5174      UVA=(UVAVG*SIGN*AL5A2)*DT
5175      FY2=Y3-UVA-Y2
5176      IF (FY2*FY3.LT.O.O) GO TO 200
5177      UVAO=UVA
5178      Y200=Y2
5179      FY3=FY2
5180      IF (SIGN.LT.O.O) Y2=Y(MOUM+ILL)
5181      IF (SIGN.GT.O.O) Y2=Y(MOUM-ILL)
5182      IF (Y2.EQ.Y(MVCB).OR.Y2.EQ.Y(MVCT)) GO TO 200
5183      GO TO 250
5184 200 ILLI=1
5185      Y20=Y2
5186      GO TO 240
5187 210 UVAVG=O.5*{(U2+U3)*ALAVG+(V2+V3)*BEAVG)+DEAVG
5188      UVAT=(UVAVG*SIGN*AL5A2)*DT
5189      FY2=Y3-UVAT-Y2
5190      FY20=Y3-UVA-Y20
5191      IF (FY2*FY20.LT.O.C) GO TO 220
5192      GO TO 230
5193 220 UVAO=UVA
5194      Y200=Y20
5195 230 UVA=UVAT
5196      Y20=Y2
5197 240 Y2=Y20+(Y20-Y200)*(Y3-UVA-Y20)/(UVA-UVAO+Y20-Y200)
5198      IF (Y2.LT.Y(MVCB)) Y2=Y(MVCB)
5199      IF (Y2.GT.Y(MVCT)) Y2=Y(MVCT)
6000      IF (Y20.EQ.O.O) GO TO 250
6001      IF (ABS((Y2-Y20)/Y20).LE.CUS) GO TO 310
6002 C
6003 250 DO 260 MM=MVCB,MVCT1
6004      IF (Y2.GE.Y(MM).AND.Y2.LE.Y(MM+1)) GO TO 270
6005 260 CONTINUE
6006 270 RDY=(Y2-Y(MM))*DYR
6007      U2=U(L,MM,N1)+(U(L,MM+1,N1)-U(L,MM,N1))*RDY
6008      V2=V(L,MM,N1)+(V(L,MM+1,N1)-V(L,MM,N1))*RDY
6009      P2=P(L,MM,N1)+(P(L,MM+1,N1)-P(L,MM,N1))*RDY
6010      R2=R(L,MM,N1)+(R(L,MM+1,N1)-R(L,MM,N1))*RDY
6011      IF (MM.EQ.MMQ) GO TO 280
6012      MMQ=MM
6013      MMAP=MM
6014      IP=O
6015      CALL MAP
6016      YPMM=YP
6017      MMAP=MM+1
6018      IP=1
6019      CALL MAP
6020      YPMM1=YP
6021 280 YP2=YPMM+(YPMM1-YPMM)*RDY
6022      BEAVG=(Y2-Y3)/(YP2-YPD)
6023      ALAVG=AL3*BEAVG/BE3
6024      DEAVG=DE3*BEAVG/BE3

```

```

6025      A2D=GAMMA*P2/RO2
6026      IF (A2D.GT.O.O) GO TO 290
6027      NP=N+NSTART
6028      WRITE (6,710) NP,L,MDUM,NVC
6029      IERR=1
6030      RETURN
6031      290 ALSA2=SQRT(O.5*(A2D+A3*A3)*(ALAVG*ALAVG+BEAVG*BEAVG))
6032      300 CONTINUE
6033      NP=N+NSTART
6034      WRITE (6,720) ILLQS, NP,L,MDUM,NVC, ICHAR
6035      IERR=1
6036      RETURN
6037      310 AL=ALD
6038      BE=BED
6039      DE=DED
6040      OM2=OMD
6041      YP=YPD
6042      MMAP=MDUM
6043      A2=SQRT(A2D)
6044      320 IF (ICHR.EQ.1) GO TO 350
6045 C
6046 C      CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
6047 C
6048      IF (JFLAG.EQ.O) GO TO 330
6049      IF (IB.NE.1) GO TO 330
6050      IF (L.EQ.2) GO TO 330
6051      IF (L.NE.LJET-1) GO TO 330
6052      GO TO 340
6053      330 DU3=(U(L+1,MDUM,N3)-U3)*DXR
6054      DV3=(V(L+1,MDUM,N3)-V3)*DXR
6055      DP3=(P(L+1,MDUM,N3)-P3)*DXR
6056      DRO3=(RO(L+1,MDUM,N3)-RO3)*DXR
6057      GO TO 350
6058      340 DUJ=(U3-U(L-1,MDUM,N3))*DXR
6059      DVJ=(V3-V(L-1,MDUM,N3))*DXR
6060      DPJ=(P3-P(L-1,MDUM,N3))*DXR
6061      DROJ=(RO3-RO(L-1,MDUM,N3))*DXR
6062 C
6063 C      ENTER THE FREE-JET BOUNDARY ITERATION LOOP
6064 C
6065      350 YW(L)=YW(L)
6066      DO 580 N=1,10
6067      IF (ICHR.EQ.1) GO TO 450
6068      IF (JFLAG.LE.O) GO TO 410
6069      IF (IB.NE.1) GO TO 410
6070      IF (L.LT.LJET) GO TO 410
6071      IF (NJ.EQ.1) GO TO 400
6072      IF (NJ.GT.2) GO TO 380
6073      360 YWOLD=YW(L)
6074      POLD=P(L,MDUM,N3)
6075      IF (P(L,MDUM,N3).LT.PE(MMAX)) GO TO 370
6076      YW(L)=YW(L)+DELY
6077      GO TO 390
6078      370 YW(L)=YW(L)-DELY
6079      GO TO 390
6080      380 IF (P(L,MDUM,N3).EQ.POLD) GO TO 360
6081      DYDP=(YW(L)-YWOLD)/(P(L,MDUM,N3)-POLD)
6082      YWNEW=YW(L)+DYDP*(PE(MMAX)-P(L,MDUM,N3))
6083      YWOLD=YW(L)
6084      POLD=P(L,MDUM,N3)
6085      YW(L)=YWNEW
6086      390 IF (YW(L).LT.(1.0-DYW)*YWOLD) YW(L)=(1.0-DYW)*YWOLD
6087      IF (YW(L).GT.(1.0+DYW)*YWOLD) YW(L)=(1.0+DYW)*YWOLD
6088      400 NXNY(L)=-((YW(L)-YW(L-1))*DXR
6089      XWI(L)=(YW(L)-YWI(L))/DT
6090      XWID=XWI(L)
6091      CALL MAP
6092      AL=AL3
6093      BE=BE3
6094      DE=DE3
6095      ALS=SQRT(AL*AL+BE*BE)
6096 C

```

```

6097 C      CALCULATE THE PSI TERMS AT THE SOLUTION POINT
6098 C
6099      410 IF (NDIM.EQ.0) GO TO 440
6100      IF (IB.EQ.2) GO TO 420
6101      ATERM3=RO3*V3/YP
6102      GO TO 440
6103      420 IF (YCB(L).EQ.0.0) GO TO 430
6104      ATERM3=RO3*V3/YCB(L)
6105      GO TO 440
6106      430 ATERM3=RO3*BE*V(L,2,N3)*DYP
6107      440 PSI13=-U3*OM2*DRG3-RO3*OM2*DU3-ATERM3
6108      PSI23=-U3*OM2*DU3-OM2*DU3/RO3
6109      PSI33=-U3*OM2*DV3
6110      PSI43=-U3*OM2*DP3+A3*A3+U3*OM2*DRG3
6111      450 ABR=N*NY(L)
6112      IF (IB.EQ.2) ABR=N/YCB(L)
6113      IF (IB.EQ.3) ABR=N*NY(L)
6114      IF (IB.EQ.4) ABR=N/YU(L)
6115      ALB=A1/ALS
6116      BEB=B1/ALS
6117      A1C=(A1+A3)*0.5
6118      A2B=(A2+A3)*0.5
6119      RO2B=(RO2+RO3)*0.5
6120      IF (ICHR.EQ.1) GO TO .60
6121      PSI21B=(PSI21+PSI23)*0.5+QUT(L,MDUM)
6122      PSI31B=(PSI31+PSI33)*0.5+QVT(L,MDUM)
6123      PSI41B=(PSI41+PSI43)*0.5+OPT(L,MDUM)
6124      PSI12B=(PSI12+PSI13+QROT(L,MDUM)+QROT2)*0.5
6125      PSI22B=(PSI22+PSI23+QUT(L,MDUM)+QUT2)*0.5
6126      PSI32B=(PSI32+PSI33+QVT(L,MDUM)+QVT2)*0.5
6127      PSI42B=(PSI42+PSI43+OPT(L,MDUM)+OPT2)*0.5
6128      GO TO 470
6129      460 PSI21B=PSI21+QUT(L,MDUM)
6130      PSI31B=PSI31+QVT(L,MDUM)
6131      PSI41B=PSI41+OPT(L,MDUM)
6132      PSI12B=PSI12+QROT2
6133      PSI22B=PSI22+QUT2
6134      PSI32B=PSI32+QVT2
6135      PSI42B=PSI42+OPT2
6136      470 IF (IWALL.EQ.0.OR.IB.NE.1) GO TO 520
6137 C
6138 C      SOLVE THE COMPATIBILITY EQUATIONS FOR A CONSTANT PRESSURE
6139 C      INFLOW - OUTFLOW BOUNDARY
6140 C
6141      ROAA2=SIGN*RO2B*A2B*ALB
6142      ROAB2=SIGN*RO2B*A2B*BEB
6143      PSI2T=(PSI42B-OPT2+A2B*A2B*(PSI12B-QROT2)+ROAA2+PSI22B+ROAB2
6144      1*PSI32B)*DT
6145      P(L,MDUM,N3)=PE(MMAX)
6146      IF (IWALLO.NE.0) P(L,MDUM,N3)=2.0*P(L,MDUM1,N3)-P(L,MDUM1-1,N3)
6147      IF (ALW.EQ.0.0) GO TO 480
6148      G(L,MDUM,N3)=(ALW*PE(MMAX)+P2+P(L,MDUM,N1)+ROAB2*(V2-V(L,MDUM,N1))
6149      1+RJAA2*(U2-U(L,MDUM,N1))+PSI2T)/(2.0*ALW)
6150      480 IF (P(L,MDUM,N3).LE.0.0) P(L,MDUM,N3)=PLOW*PC
6151      IF (Y1.GE.Y3.AND.IWALLO.FQ.0) GO TO 510
6152      RO(L,MDUM,N3)=RO1*(P(L,MDUM,N3)-P1-(PSI41B-OPT(L,MDUM))*DT)/(A1B
6153      1*A1B)+QROT(L,MDUM)*DT
6154      IF (RO(L,MDUM,N3).LE.0.0) RO(L,MDUM,N3)=ROLOW/G
6155      PSI1T=(PSI21B-ABR*PSI31B)*DT
6156      IF (ABR.EQ.0.0) GO TO 490
6157      ABRT=ABR*1.0/ABR
6158      V(L,MDUM,N3)=(ABR*V1+V2/ABR+U2-U1-PSI1T+(P2-P(L,MDUM,N3)+PSI2T)
6159      1/ROAA2)/ABRT
6160      GO TO 500
6161      490 V(L,MDUM,N3)=V2+(P2-P(L,MDUM,N3)+PSI2T)/(RO2B*A2B)
6162      500 U(L,MDUM,N3)=U1+ABR*(V(L,MDUM,N3)-V1)+PSI1T
6163      GO TO 700
6164      510 ND=N1
6165      IF (ICHR.EQ.2) ND=N3
6166      RO(L,MDUM,N3)=0.1*RO(1,MDUM,ND)+0.9*RO(L,MDUM,N1)
6167      U(L,MDUM,N3)=0.1*U(1,MDUM,ND)+0.9*U(L,MDUM,N1)
6168      V(L,MDUM,N3)=V2+(-P(L,MDUM,N3)+P2-ROAA2*(U(L,MDUM,N3)-U2)+PSI2T)
6169      1/ROAB2
6170      GO TO 700

```



```

6171 C
6172 C      SOLVE THE COMPATIBILITY EQUATIONS FOR A SOLID BOUNDARY
6173 C
6174      520 U(L,MDUM,N3)=(U1-ABR*(VI-XWID)+(PSI2*B-ABR*PSI2B)*DT)/(1.0+ABR
6175          1 *ARR)
6176          V(L,MDUM,N3)=-U(L,MDUM,N3)*ABR*XWID
6177          IF (NOSL.EQ.0) GO TO 530
6178          U(L,MDUM,N3)=0.0
6179          V(L,MDUM,N3)=0.0
6180          PSI22B=PSI22B-OUT2
6181          PSI32B=PSI32B-QVT2
6182      570 P(L,MDUM,N3)=P2 SIGN*P02B+A2B*(AIR*(U(L,MDUM,N3)-U2)+BEB*(V(L,MDUM
6183          1 ,N3)-V2))+(PSI42B+A2B*A2B*PSI12B+SIGN*P02B*A2B*(AIR*PSI22B+BEB
6184          2 *PSI32B))*DT
6185          IF (P(L,MDUM,N3).LE.0.0) P(L,MDUM,N3)=PLOW-PC
6186          RO(L,MDUM,N3)=RO1+(P(L,MDUM,N3)-P1-PSI4*B*CT)/(AIR*AIR)
6187          IF (RO(L,MDUM,N3).LE.0.0) RO(L,MDUM,N3)=ROLOW/G
6188          IF (IB.EQ.2) GO TO 540
6189          IF (IB.EQ.3) GO TO 550
6190          IF (IB.EQ.4) GO TO 560
6191          IF (TW(1).LT.0.0) GO TO 570
6192          IF (JFLAG.EQ.1 AND L.GE.LJET) GO TO 570
6193          P(L,MDUM,N3)=RM(L,MDUM,N3)*RG+TW(L)
6194          GO TO 570
6195      540 IF (TCB(1).LT.0.0) GO TO 570
6196          P(L,MDUM,N3)=RO(L,MDUM,N3)*RG+TCB(L)
6197          GO TO 570
6198      550 IF (TL(1).LT.0.0) GO TO 570
6199          P(L,MDUM,N3)=RO(L,MDUM,N3)*RG+TL(L)
6200          GO TO 570
6201      560 IF (TU(1).LT.0.0) GO TO 570
6202          P(L,MDUM,N3)=RO(L,MDUM,N3)*RG+TU(L)
6203 C
6204 C      TEST FOR CONVERGENCE OF THE FREE-JET BOUNDARY
6205 C
6206      570 IF (JFLAG.EQ.0) GO TO 700
6207          IF (IB.NE.1) GO TO 700
6208          IF (L.LT.LJET-1) GO TO 700
6209          IF (L.EQ.LJET-1) GO TO 590
6210          IF (ICAR.EQ.1) GO TO 700
6211          IF (JFLAG.EQ.-1 AND L.NE.LJET) GO TO 700
6212          IF (JFLAG.EQ.-1 AND L.EQ.LJET) GO TO 690
6213          DELP=ABS((P(L,MDUM,N3)-PE(MMAX))/PE(MMAX))
6214          IF (DELP.LE.0.001 AND L.NE.LJET) GO TO 700
6215          IF (DELP.LE.0.001 AND L.EQ.LJET) GO TO 690
6216      580 CONTINUE
6217          IF (L.EQ.LJET) GO TO 690
6218          GO TO 700
6219 C
6220 C      SOLVE FOR THE DOWNSTREAM SIDE OF THE WALL EXIT POINT FOR
6221 C      EITHER THE SHARP EXPANSION CORNER CASE, UNDER-EXPANDED
6222 C      FREE-JET CASE OR OVER-EXPANDED FREE-JET CASE
6223 C
6224      590 UD(3)=U(L,MDUM,N3)
6225          VD(3)=V(L,MDUM,N3)
6226          PD(3)=P(L,MDUM,N3)
6227          ROD(3)=RO(L,MDUM,N3)
6228          PD(4)=PE(MMAX)
6229          XM1=SQRT((UD(3)*UD(3)+VD(3)*VD(3))/(GAMMA*PD(3)/ROD(3)))
6230          DUMD=1.0+GAM2*XM1*XM1
6231          TD=PD(3)/(ROD(3)*RG)
6232          TID=TD*DUMD
6233          PTD=PD(3)*DUMD**GAM1
6234 C
6235 C      SHARP EXPANSION CORNER CASE
6236 C
6237          IF (JFLAG.NE.-1) GO TO 630
6238          B=SQRT(GAM3)
6239          CC1=XM1*XM1-1.0
6240          IF (CC1.LT.0.0) CC1=0.0
6241          PMA1=B*ATAN(SQRT(CC1/(B*B)))-ATAN(SQRT(CC1))
6242          PMA=ATAN(-NXNY(LJET))-ATAN(-NXNY(LJET-1))

```

```

6243      PMAD=PMA+PMA1
6244      XM2=2.0*XM1
6245      DO 610 I=1,10
6246      CI=XM2*XM2-1.0
6247      PMAI=B*ATAN(SQR1(CI/(B*B)))-ATAN(SQRT(C:))
6248      IF (ABS((PMAI-PMAD)/PMAD).LE.O.OOO1) GO TO 620
6249      IF (I.NE.1) GO TO 600
6250      XMO=XM2
6251      XM2=O.9*XM2
6252      PMAO=PMA:
6253      GO TO 610
6254      600 DMDA=(XM2-XMJ)/(PMAI-PMAD)
6255      XMO=XM2
6256      XM2=XM2+DMDA*(PMAD-PMAI)
6257      PMAO=PMAI
6258      610 CONTINUE
6259      620 DUMD=1.0+GAM2*XM2*XM2
6260      TD=TTD/DUMD
6261      PD(4)=PTD/DUMD**GAM1
6262      ROD(4)=PD(4)/(RG*TD)
6263      GO TO 660
6264 C
6265 C      UNDER-EXPANDED FREE-JET CASE
6266 C
6267      650 IF (PE(MMAX).GT.PD(3).AND.XM1.GE.1.0) GO TO 640
6268      ROD(4)=ROD(3)*(PE(MMAX)/PD(3))**(1.0/GAMMA)
6269      GO TO 650
6270 C
6271 C      OVER-EXPANDED FREE-JET CASE
6272 C
6273      640 PRD=PE(MMAX)/PD(3)
6274      ROD(4)=ROD(3)*(GAM3*PRD+1.0)/(PRD+GAM3)
6275      650 TE=PE(MMAX)/(ROD(4)*RG)
6276      XM2=SQRT((TTD/TE-1.0)/GAM2)
6277      660 SS=SQRT(GAMMA*PD(4)/ROD(4))
6278      VMAG=XM2*SS
6279      UD(4)=VMAG/SQRT(1.0+NXNY(LJET)*NXNY(LJET))
6280      VD(4)=-UD(4)*NXNY(LJET)
6281      IF (JFLAG.EQ.-1) GO TO 700
6282      IF (XM1.GE.1.0) GO TO 700
6283 C
6284 C      AVERAGE THE 1-SIDED MACH NOS FOR THE INTERIOR POINT CALCULATIONS
6285 C      IF THE UPSTREAM FLOW IS SUBSONIC - FREE-JET CASE
6286 C
6287      XMB=(XM1+XM2)/2.0
6288      IF (XMB.GE.1.0) GO TO 670
6289      DPL=1.0
6290      DPR=1.0
6291      GO TO 680
6292      670 DPL=XM2-1.0
6293      DPR=1.0-XM1
6294      XMB=1.0
6295      680 DPLR=DPR+DPL
6296      DUM=1.0+GAM2*XM3*XM2
6297      TEMP=TTD/DUM
6298      F(L,MDUM,N3)=PTD/DUM**GAM1
6299      RO(L,MDUM,N3)=P(L,MDUM,N3)/(RG*TEMP)
6300      AS=GAMMA*P(L,MDUM,N3)/RO(L,MDUM,N3)
6301      QA=XMB*SQRT(AS)
6302      DNXY=(DPR+NXNY(LJET)+DPL+NXNY(L))/DPLR
6303      U(L,MDUM,N3)=QA/SQRT(1.0+DNXY*DNXY)
6304      V(L,MDUM,N3)=-U(L,MDUM,N3)*DNXY
6305      GO TO 700
6306      690 UD(1)=UD(3)
6307      VD(1)=VD(3)
6308      PD(1)=PD(3)
6309      ROD(1)=ROD(3)
6310      UD(2)=UD(4)
6311      VD(2)=VD(4)
6312      PD(2)=PD(4)
6313      ROD(2)=ROD(4)
6314      700 CONTINUE

```

```

6315      IF (JFLAG.EQ.0) RETURN
6316      IF (IS.GE.2) RETURN
6317      IF (ICAR.EQ.1) RETURN
6318      U(LJET-1,MMAX,N1)=UOLD
6319      IF (JFLAG.EQ.-1) RETURN
6320      YW(LMAX)=YW(LMAX)
6321      YW(LMAX)=2.0*YW(L1)-YW(L2)
6322      NXNY(LMAX)=- (YW(LMAX)-YW(L1))*DXR
6323      XWI(LMAX)=(YW(LMAX)-YWI(LMAX))/DT
6324      RETURN
6325 C
6326 C      FORMAT STATEMENTS
6327 C
6328      710 FORMAT (1H0,61H***** A NEGATIVE SQUARE ROOT OCCURED IN SUBROUTINE
6329      1WALL AT N=,16,4H, L=,12,4H, M=,12,10H, AND NVC=,13,6H ***** )
6330      720 FORMAT (1H0,64H***** THE CHARACTERISTIC SOLUTION IN WALL FAILED TO
6331      1 CONVERGE IN ,12,17H ITERATIONS AT N=,16,4H, L=,12,4H, M=,12,6H, N
6332      2VC=,13,1H,./7X,10HAND ICAR=,11,6H ***** )
6333      END

```

```

6334      SUBROUTINE INLET
6335 C
6336 C .....
6337 C
6338 C      THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE INLET
6339 C
6340 C .....
6341 C
6342 *CALL ,MCC
6343      IP=1
6344      IMAP=1
6345      LD1=2
6346      X3=XI
6347      DXP=XP(2)-X3
6348      ATERM2=0.0
6349      ATERM3=0.0
6350      MIS=1
6351      MIF=MMAX
6352      IF (IB.EQ.3) MIF=MDFS
6353      IF (IB.EQ.4) MIS=MDFS
6354      IF (IVC.EQ.0) GO TO 10
6355      IF (NVC.EQ.1) GO TO 10
6356      IF (MIS.EQ.1) MIS=MVCB
6357      IF (MIF.EQ.MMAX) MIF=MVCT
6358      IF (ICHAR.EQ.1.AND.MIF.NE.MMAX) MIF=MIF+1
6359 C
6360 C      BEGIN THE M OR Y DO LOOP
6361 C
6362      10 DO 400 M=MIS,MIF
6363          IF (IVC.EQ.0) GO TO 20
6364          IF (NVC.NE.1) GO TO 20
6365          IF (M.LT.MVCB) GO TO 20
6366          IF (M.GT.MVCT) GO TO 20
6367          IF (ICHAR.NE.1) GO TO 400
6368          IF (M.EQ.MVCB.AND.MVCB.NE.1) GO TO 20
6369          GO TO 400
6370      20 IF (ISUPER.EQ.0) GO TO 70
6371          IF (ISUPER.EQ.2.AND.IB.EQ.4) GO TO 70
6372          IF (ISUPER.EQ.3.AND.IS.EQ.3) GO TO 70
6373          SM=U(1,M,N1)*U(1,M,N1)/(GAMMA*P(1,M,N1)/RO(1,M,N1))
6374          IF (SM.LT.1.0.AND.IINLET.EQ.0) GO TO 30
6375          IF (INBC.EQ.0) P(1,M,N3)=PI(M)*PC
6376          IF (INBC.NE.0) U(1,M,N3)=UI(M)
6377          UOLD=U(1,M,N3)
6378          VOLD=V(1,M,N3)
6379          POLD=P(1,M,N3)
6380          GO TO 400
6381 C
6382      30 IF (INBC.NE.0) GO TO 70
6383          IF (M.EQ.MMAX) GO TO 40
6384          IF (M.EQ.MDFS.AND.IB.EQ.4) GO TO 50
6385          IF (M.EQ.MDFS.AND.IB.EQ.3) GO TO 60
6386          IF (M.NE.1) GO TO 70
6387          IF (NGCB.EQ.0) GO TO 70
6388          IF (TCB(1).LT.0.0) GO TO 70
6389          P(1,M,N3)=TCB(1)*RO(1,M,N3)*RG
6390          GO TO 400
6391      40 IF (TW(1).LT.0.0) GO TO 70
6392          P(1,M,N3)=TW(1)*RO(1,M,N3)*RG
6393          GO TO 400
6394      50 IF (TU(1).LT.0.0) GO TO 70
6395          P(1,M,N3)=TU(1)*RO(1,M,N3)*RG
6396          GO TO 400
6397      60 IF (TL(1).LT.0.0) GO TO 70
6398          P(1,M,N3)=TL(1)*RO(1,M,N3)*RG
6399          GO TO 400
6400 C
6401      70 MMAP=M
6402          CALL MAP
6403          BED=2.0*BE3+BE4/(BE3+BE4)
6404          AL34=AL3+AL4
6405          IF (AL34.EQ.0.0) AL34=1.0

```

```

6406      ALD=2.0*AL3*AL4/AL34
6407      U2=U(1,M,N1)
6408      A2=SQRT(GAMMA*P(1,M,N1)/RO(1,M,N1))
6409      IF (1CHAR.NE.1) GO TO 90
6410      IF (1SUPER.EQ.1) GO TO 80
6411      U(1,M,N3)=U2
6412      V(1,M,N3)=V(1,M,N1)
6413      80 A3=A2
6414 C
6415 C      CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
6416 C
6417      90 OUTB=OUT(1,M)
6418      OPTB=OPT(1,M)
6419      GROTB=GROT(1,M)
6420      IF (1VC.EQ.0) GO TO 100
6421      IF (M.EQ.MMAX) GO TO 100
6422      IF (11VC.EQ.1.OR.M.NE.MVCT+1) GO TO 100
6423 C
6424 C      LINEAR INTERPOLATION IN TIME FOR M=MVCT+1
6425 C
6426      UB=UU1(1)+RIND*(UU2(1)-UU1(1))
6427      VB=VV1(1)+RIND*(VV2(1)-VV1(1))
6428      PB=PP1(1)+RIND*(PP2(1)-PP1(1))
6429      ROB=RORO1(1)+RIND*(RORO2(1)-RORO1(1))
6430      ULP=UU1(2)+RIND*(UU2(2)-UU1(2))
6431      VLP=VV1(2)+RIND*(VV2(2)-VV1(2))
6432      PLP=PP1(2)+RIND*(PP2(2)-PP1(2))
6433      ROLP=RORO1(2)+RIND*(RORO2(2)-RORO1(2))
6434      GO TO 110
6435 C
6436      100 UB=U(1,M,N1)
6437      VB=V(1,M,N1)
6438      PB=P(1,M,N1)
6439      ROB=RO(1,M,N1)
6440      ULP=U(2,M,N1)
6441      VLP=V(2,M,N1)
6442      PLP=P(2,M,N1)
6443      ROLP=RO(2,M,N1)
6444      110 BU=(ULP-UB)/DXP
6445      BV=(VLP-VB)/DXP
6446      BP=(PLP-PB)/DXP
6447      BRO=(ROLP-ROB)/DXP
6448      CU=UB-BU*X3
6449      CV=VB-BV*X3
6450      CP=PB-BP*X3
6451      CRO=ROB-BRO*X3
6452 C
6453 C      CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
6454 C      COEFFICIENTS
6455 C
6456      IF (M.EQ.1) GO TO 130
6457      IF (M.EQ.MDFS.AND.1R.EQ.4) GO TO 140
6458      IF (1VC.EQ.0) GO TO 120
6459      IF (11VC.EQ.1.OR.M.NE.MVCB) GO TO 120
6460 C
6461 C      LINEAR INTERPOLATION IN TIME FOR M=MVCB
6462 C
6463      ULPMM=U(2,M-1,NN1)+RIND*(U(2,M-1,NN3)-U(2,M-1,NN1))
6464      VLPMM=V(2,M-1,NN1)+RIND*(V(2,M-1,NN3)-V(2,M-1,NN1))
6465      PLPMM=P(2,M-1,NN1)+RIND*(P(2,M-1,NN3)-P(2,M-1,NN1))
6466      ROLPMM=RO(2,M-1,NN1)+RIND*(RO(2,M-1,NN3)-RO(2,M-1,NN1))
6467      UMM=U(1,M-1,NN1)+RIND*(U(1,M-1,NN3)-U(1,M-1,NN1))
6468      VMM=V(1,M-1,NN1)+RIND*(V(1,M-1,NN3)-V(1,M-1,NN1))
6469      PMM=P(1,M-1,NN1)+RIND*(P(1,M-1,NN3)-P(1,M-1,NN1))
6470      ROMM=RO(1,M-1,NN1)+RIND*(RO(1,M-1,NN3)-RO(1,M-1,NN1))
6471 C
6472      DU=(ULP-ULPMM)*DYR
6473      DV=(VLP-VLPMM)*DYR
6474      DP=(PLP-PLPMM)*DYR
6475      DRO=(ROLP-ROLPMM)*DYR
6476      DU1=(UB-UMM)*DYR
6477      DV1=(VB-VMM)*DYR

```

```

6478      DP1=(PB-PMW)*DYP
6479      DR01=(ROB-ROMM)*DYP
6480      GO TO 150
6481 120  DU=(ULP-U(2,M-1,N1))*DYP
6482      DV=(VLP-V(2,M-1,N1))*DYP
6483      DP=(PLP-P(2,M-1,N1))*DYP
6484      DR0=(ROLP-RO(2,M-1,N1))*DYP
6485      DU1=(UB-U(1,M-1,N1))*DYP
6486      DV1=(VB-V(1,M-1,N1))*C/R
6487      DP1=(PB-P(1,M-1,N1))*DYP
6488      DR01=(ROB-RO(1,M-1,N1))*DYP
6489      GO TO 150
6490 130  IF (NGCB.NE.O) GO TO 140
6491      DU=0.0
6492      DV=(4.0*V(2,2,N1)-V(2,3,N1))*0.5*DYP
6493      DP=0.0
6494      DR0=0.0
6495      DU1=0.0
6496      DV1=(4.0*V(1,2,N1)-V(1,3,N1))*0.5*DYP
6497      DP1=0.0
6498      DR01=0.0
6499      GO TO 150
6500 140  DU=(U(2,M+1,N1)-ULP)*DYP
6501      DV=(V(2,M+1,N1)-VLP)*DYP
6502      DP=(P(2,M+1,N1)-PLP)*DYP
6503      DR0=(RO(2,M+1,N1)-ROLP)*DYP
6504      DU1=(U(1,M+1,N1)-UB)*DYP
6505      DV1=(V(1,M+1,N1)-VB)*DYP
6506      DP1=(P(1,M+1,N1)-PB)*DYP
6507      DR01=(RO(1,M+1,N1)-ROB)*DYP
6508 150  BDU=(DU-DU1)/DXP
6509      BDV=(DV-DV1)/DXP
6510      BDP=(DP-DP1)/DXP
6511      BDR0=(DR0-DR01)/DXP
6512      CDU=DU1-BDU*X3
6513      CDV=DV1-BDV*X3
6514      CDP=DP1-BDP*X3
6515      CDRO=DR01-BDR0*X3
6516 C
6517 C      CALCULATE THE COEFFICIENTS FOR THE QUICK SOLVER
6518 C
6519      IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 160
6520      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 160
6521      IF (M.EQ.MGFS.AND.LDFSS.EQ.1) GO TO 160
6522      DUDYQ2=0.5*(DUDYQS(2,M,1)+DUDYQS(2,M,2))
6523      DV DYQ2=0.5*(DVDYQS(2,M,1)+DVDYQS(2,M,2))
6524      DDPYQ2=0.5*(DPDYQS(2,M,1)+DPDYQS(2,M,2))
6525      DUDYQ1=0.5*(DUDYQS(1,M,1)+DUDYQS(1,M,2))
6526      DV DYQ1=0.5*(DVDYQS(1,M,1)+DVDYQS(1,M,2))
6527      DDPYQ1=0.5*(DPDYQS(1,M,1)+DPDYQS(1,M,2))
6528      BDUQS=(DUDYQ2-DUDYQ1)/DXP
6529      BDVQS=(DVDYQ2-DVDYQ1)/DXP
6530      BDPQS=(DPDYQ2-DPDYQ1)/DXP
6531      CDUQS=DUDYQ1-BDUQS*X3
6532      CDVQS=DVDYQ1-BDVQS*X3
6533      CDPQS=DPDYQ1-BDPQS*X3
6534 C
6535 C      CALCULATE X2
6536 C
6537 160  IF (ICHAR.NE.1) A3=SQRT(GAMMA*P(1,M,N3)/RO(1,M,N3))
6538      DO 170 IL=1,2
6539      X2=X3-((U(1,M,N3)-A3)*OM2+(U2-A2)*OM2)*0.5*DT
6540      IF (X2-X3.LE.O.05*DXP) X2=X3+0.05*DXP
6541 C
6542 C      INTERPOLATE FOR THE PROPERTIES
6543 C
6544      U2=BU*X2+CU
6545      P2=B2*X2+CP
6546      RO2=BRO*X2+CRO
6547      A2=SQRT(GAMMA*P2/RO2)
6548 170  CONTINUE
6549      V2=BV*X2+CV

```

```

6550      UV2=U2*AL3+V2*BE3
6551 C
6552 C      INTERPOLATE FOR THE CROSS DERIVATIVES
6553 C
6554      DU2=BDU*X2+CDU
6555      DV2=BDV*X2+CDV
6556      DP2=BOP*X2+CDP
6557      DRO2=BDRO*X2+CDRO
6558 C
6559      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 180
6560      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 180
6561      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) GO TO 180
6562      DU2QS=BDUQS*X2+CDUQS
6563      DV2QS=BDVQS*X2+CDVQS
6564      DP2QS=BDPOS*X2+CDPOS
6565 C
6566 C      CALCULATE THE PSI TERMS
6567 C
6568      180 IF (NDIM.EQ.0) GO TO 200
6569      IF (M.EQ.1.AND.YCB(1).EQ.0.0) GO TO 190
6570      ATERM2=R02+V2/YP
6571      GO TO 200
6572      190 ATERM2=R02+BE3+DV2
6573      200 PSI12=-UV2+DRO2-R02*AL3+DU2-R02*BE3+DV2*ATERM2
6574      PSI22=-UV2+DU2-AL3*DP2/R02
6575      PSI42=-UV2+DP2+A2*A2+UV2+DRO2
6576 C
6577      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 210
6578      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 210
6579      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) GO TO 210
6580      PSI12=-UV2+DRO2-R02*ALD+DU2QS-R02*BED+DV2QS*ATERM2
6581      UV2=U2*ALD+V2*BED
6582      PSI22=-UV2+DU2QS-ALD*DP2QS/R02
6583      210 IF (ICHR.EQ.1) GO TO 280
6584 C
6585 C      CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
6586 C
6587      IF (M.EQ.1.AND.HGCB.EQ.0) GO TO 220
6588      IF (M.EQ.MDFS.AND.IB.EQ.3) GO TO 230
6589      IF (M.EQ.MMAX) GO TO 230
6590      DU3=(U(1,M+1,N3)-U(1,M,N3))*DYR
6591      DV3=(V(1,M+1,N3)-V(1,M,N3))*DYR
6592      DP3=(P(1,M+1,N3)-P(1,M,N3))*DYR
6593      DRO3=(RO(1,M+1,N3)-RO(1,M,N3))*DYR
6594      GO TO 240
6595      220 DU3=0.0
6596      DV3=(4.0*V(1,2,N3)-V(1,3,N3))*0.5*DyR
6597      UP3=0.0
6598      DRO3=0.0
6599      GO TO 240
6600      230 DU3=(U(1,M,N3)-U(1,M-1,N3))*DyR
6601      DV3=(V(1,M,N3)-V(1,M-1,N3))*DyR
6602      DP3=(P(1,M,N3)-P(1,M-1,N3))*DyR
6603      DRO3=(RO(1,M,N3)-RO(1,M-1,N3))*DyR
6604 C
6605 C      CALCULATE THE PSI TERMS AT THE SOLUTION POINT
6606 C
6607      240 IF (NDIM.EQ.0) GO TO 260
6608      IF (M.EQ.1.AND.YCB(1).EQ.0.0) GO TO 250
6609      ATERM3=RO(1,M,N3)+V(1,M,N3)/YP
6610      GO TO 260
6611      250 ATERM3=RO(1,M,N3)+BE4+DV3
6612      260 UV3=U(1,M,N3)+AL4+V(1,M,N3)+BE4
6613      PSI13=-UV3+DRO3-RO(1,M,N3)+AL4+DU3-RO(1,M,N3)+BE4+DV3*ATERM3
6614      PSI23=-UV3+DU3-AL4*DP3/RO(1,M,N3)
6615      PSI43=-UV3+DP3+A3*A3+UV3+DRO3
6616 C
6617      IF (IQSD.EQ.0.OR.NVC.EQ.1) GO TO 290
6618      IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 290
6619      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) GO TO 290
6620      DU DY1=0.5*(U(1,M+1,N3)-UOLD)*DYR
6621      DV DY1=0.5*(V(1,M+1,N3)-VOLD)*DYR

```

```

6622      DPDY1=0.5*(P(1,M+1,N3)-POLD)*DYR
6623      IF (MDFS.EQ.0) GO TO 270
6624      IF (M.NE.MDFS+1.OR.LDFSS.NE.1) GO TO 270
6625      DUDY1=0.5*(U(1,M+1,N3)-UL(1,N3))*DYR
6626      DVDY1=0.5*(V(1,M+1,N3)-VL(1,N3))*DYR
6627      OPDY1=0.5*(P(1,M+1,N3)-PL(1,N3))*DYR
6628  270  PSI13=-UV3*DR03-RO(1,M,N3)*ALD*DUDY1-RO(1,M,N3)*BED+DVDY1-ATERM3
6629      UV3=U(1,M,N3)*ALD+V(1,M,N3)*BED
6630      PSI23=-UV3*DUDY1-ALD*OPDY1/RO(1,M,N3)
6631      GJ TO 290
6632  280  PSI23=PSI22
6633      PSI43=PSI42
6634      PSI13=PSI12
6635  290  IF (IQSC.EQ.0.OR.NVC.EQ.1) GO TO 300
6636      UOLD=U(1,M,N3)
6637      VOLD=V(1,M,N3)
6638      POLD=P(1,M,N3)
6639  300  PSI1B=0.5*(PSI12+PSI13)+QROTB
6640      PSI2B=0.5*(PSI22+PSI23)+QUTB
6641      PSI4B=0.5*(PSI42+PSI43)+QPTB
6642 C
6643 C      SOLVE THE COMPATIBILITY EQUATION FOR P OR U
6644 C
6645      IF (ISUPER.EQ.0) GO TO 340
6646      IF (ISUPER.EQ.2.AND.IB.EQ.4) GO TO 340
6647      IF (ISUPER.EQ.3.AND.IB.EQ.3) GO TO 340
6648      ROAB=0.5*(RO2+A2+RO(1,M,N3)+A3)
6649      AB=0.5*(A2+A3)
6650      IF (INBC.NE.C) GO TO 320
6651      PSIT=(PSI4B-ROAB*(PSI2B-QUTB)+AB*AB*(PSI1B-QROTB))*DT
6652      IF (ALI.EQ.0.0) GO TO 310
6653      U(1,M,N3)=(ROAB*ALI*UI(M)+ROAB*(U2+U(1,M,N1))+P(1,M,N1)-P2-PSIT)/
6654      1 (ROAB*(2.0+ALI))
6655  310  P(1,M,N3)=P2+ROAB*(U(1,M,N3)-U2)+PSIT
6656      IF (P(1,M,N3).LE.O.C) P(1,M,N3)=PLOW*PC
6657      GO TO 400
6658  320  IF (M.EQ.MMAX.AND.IWALL.NE.O) GO TO 400
6659      PSIT=(PSI4B-ROAB*(PSI2B-QUTB)+AB*AB*(PSI1B-QROTB))*DT
6660      IF (ALI.EQ.0.0) GO TO 330
6661      P(1,M,N3)=(ALI*PI(M)+PC+ROAB*(U(1,M,N1)-U2)+P2+P(1,M,N1)+PSIT)/(2.
6662      1 O+ALI)
6663      IF (P(1,M,N3).LE.O.C) P(1,M,N3)=PLOW*PC
6664  330  U(1,M,N3)=U2+(P(1,M,N3)-P2-PSIT)/ROAB
6665      GO TO 400
6666 C
6667 C      SOLVE THE COMPATIBILITY EQUATIONS FOR U, V, P, AND RC
6668 C
6669  340  MN3=SQRT(U(1,M,N3)*U(1,M,N3)+V(1,M,N3)*V(1,M,N3))/A3
6670      T2=P2/(RO2+RG)
6671      TTHETA=TAN(THETA(M))
6672      UCORR=1.0
6673      IF (NOSLIP.EQ.0) GO TO 350
6674      IF (M.EQ.MMAX.AND.IWALL.EQ.0) UCORR=0.0
6675      IF (M.EQ.1.AND.NGCB.NE.0) UCORR=0.0
6676      IF (M.EQ.MDFS.AND.LDFSS.EQ.1) UCORR=0.0
6677 C
6678  350  DO 380 ITER=1,20
6679      DEM=(1.0+GAM2*MN3*MN3)
6680      P(1,M,N3)=PT(M)/(DEM+GAM1)
6681      T3=TT(M)/DEM
6682      IF (M.EQ.MMAX.AND.TW(1).GT.O.O) T3=TW(1)
6683      IF (M.EQ.1.AND.TCB(1).GT.O.O) T3=TCB(1)
6684      IF (M.NE.MDFS.OR.LDFSS.NE.1) GO TO 360
6685      IF (IB.EQ.3.AND.TL(1).GT.O.O) T3=TL(1)
6686      IF (IB.EQ.4.AND.TU(1).GT.O.O) T3=TU(1)
6687  360  PAVG=(P2+P(1,M,N3))*0.5
6688      TAVG=(T2+T3)*0.5
6689      ROAVG=PAVG/(TAVG*RG)
6690      AS=GAMMA*PAVG/ROAVG
6691      U(1,M,N3)=U2+DT*PSI2B+(P(1,M,N3)-P2-(PSI4B+AS*PSI1B)*DT)/(ROAVG
6692      1 *SQRT(AS))
6693      U(1,M,N3)=U(1,M,N3)+UCORR

```



```

6694      V(1,M,N3)=U(1,M,N3)*THETA
6695      OMN3=MN3
6696      AS=GAMMA*RG+T3
6697      MN3=SQRT((U(1,M,N3)+U(1,M,N3)+V(1,M,N3)+V(1,M,N3))/AS)
6698      IF (OMN3.NE.0.0) GO TO 370
6699      I. (ABS(MN3-OMN3) LE.0.0001) GO TO 390
6700      GO TO 390
6701  370 IF (ABS((MN3-OMN3)/GMN3) LE.0.001) GO TO 390
6702      380 CONTINUE
6703 C
6704      NP=N+NSTART
6705      WRITE (6,430) M,NP
6706      390 RO(1,M,N3)=P(1,M,N3)/(RG+T3)
6707      400 CONTINUE
6708      IF (IWALL.NE.0) P(1,M,N3)=P(1,M,N3)
6709 C
6710 C      ZERO THE CORNER 0 FOR THE P.V.RO NO SLIP BOUNDARY CONDITION CASE
6711 C
6712      IF (INOC.EQ.0 OR INBC.EQ.0) RETURN
6713      IF (ISUPER.EQ.0) RETURN
6714      IF (ISUPER.EQ.2.AND.IB.EQ.4) RETURN
6715      IF (ISUPER.EQ.3.AND.IB.EQ.3) RETURN
6716      IF (NVC.EQ.1.AND.MVCB.EQ.1) GO TO 410
6717      IF (NVCB.NE.0) U(1,1,N3)=0.0
6718      410 IF (NVC.EQ.1.AND.MVCT.EQ.MMAX) GO TO 420
6719      IF (IWALL.EQ.0) U(1,MMAX,N3)=0.0
6720      420 IF (MDFS.EQ.0) RETURN
6721      IF (NVC.EQ.1.AND (MDFS GT MVCB.AND.MDFS LT.MVCT)) RETURN
6722      U(1,MDFS,N3)=0.0
6723      RETURN
6724 C
6725 C      FORMAT STATEMENTS
6726 C
6727      430 FORMAT (1H0,55H***** THE SOLUTION FOR THE ENTRANCE BOUNDARY POINT
6728      1( 1.,12,1H.,16,43H) FAILED TO CONVERGE IN 20 ITERATIONS *****
6729      END

```

```

6730      SUBROUTINE EXITT
6731 C
6732 C      .....
6733 C
6734 C      THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE EXIT
6735 C
6736 C      .....
6737 C
6738 *CALL MCC
6739      IP=1
6740      LMAP=LMAX
6741      LD1=L1
6742      X3=XE
6743      DXP=X3-XP(L1)
6744      ATERM2=0.0
6745      ATERM3=0.0
6746      MIS=1
6747      MIF=MMAX
6748      SM=0.0
6749      R2=DF=1.0
6750      IF (NPE.NE.0) RNNPE=FLOAT(N)/FLOAT(NPE)
6751      IF (RNNPE.LE.0.0.OR.RNNPE.GT.1.0) RNNPE=1.0
6752      IF (IB.EQ.3) MIF=MDFS
6753      IF (IB.EQ.4) MIS=MDFS
6754      IF (IVC.EQ.0) GO TO 10
6755      IF (NVC.EQ.1) GO TO 10
6756      IF (MIS.EQ.1) MIF=MVCB
6757      IF (MIF.EQ.MMAX) MIF=MVCT
6758      IF ((ICHR.EQ.1.AND.MIF.NE.MMAX) MIF=MIF+1
6759 C
6760 C      BEGIN THE M OR Y DO LOOP
6761 C
6762      10 DO 330 M=MIS,MIF
6763          IF (IVC.EQ.0) GO TO 20
6764          IF (NVC.NE.1) GO TO 20
6765          IF (M.LT.MVCB) GO TO 20
6766          IF (M.GT.MVCT) GO TO 20
6767          IF ((ICHR.NE.1) GO TO 330
6768          IF (M.EQ.MVCB.AND.MVCB.NE.1) GO TO 20
6769          GO TO 330
6770      20 IF (IEXIT.EQ.1) GO TO 40
6771          A1D=GAMMA*P(LMAX,M,N1)/RO(LMAX,M,N1)
6772          IF (A1D.GT.0.0) GO TO 30
6773          NP=N*NS*START
6774          WRITE (6,3RG) NP,M,NVC,ICHR
6775          IERR=1
6776          RETURN
6777      30 A1=SQRT(A1D)
6778          IF (IEXIT.EQ.2) GO TO 50
6779          SM=U(LMAX,M,N1)+U(LMAX,M,N1)/(A1*A1)
6780          IF (SM.LT.1.0) GO TO 50
6781      40 U(LMAX,M,N3)=U(L1,M,N3)+FLOAT(IFX)*(U(L1,M,N3)-U(L2,M,N3))
6782          V(LMAX,M,N3)=V(L1,M,N3)+FLOAT(IFX)*(V(L1,M,N3)-V(L2,M,N3))
6783          P(LMAX,M,N3)=P(L1,M,N3)+FLOAT(IFX)*(P(L1,M,N3)-P(L2,M,N3))
6784          RO(LMAX,M,N3)=RO(L1,M,N3)+FLOAT(IFX)*(RO(L1,M,N3)-RO(L2,M,N3))
6785          UOLD=U(LMAX,M,N3)
6786          VOLD=V(LMAX,M,N3)
6787          POLD=P(LMAX,M,N3)
6788          IF (U(LMAX,M,N3).GE.0.0) GO TO 320
6789          P(LMAX,M,N3)=RNNPE*PE(M)+(1.0-RNNPE)*PEI
6790          GO TO 300
6791 C
6792 C      50 MMAP=M
6793          CALL MAP
6794          BED=2.0*BE3+BE4/(BE3+BE4)
6795          AL34=AL3+AL4
6796          DE34=DE3+DE4
6797          IF (AL34.EQ.0.0) AL34=1.0
6798          IF (DE34.EQ.0.0) DE34=1.0
6799          ALD=2.0*AL3+AL4/AL34
6800          DED=2.0*DE3+DE4/DE34
6801          U1=U(LMAX,M,N1)

```

```

6802      U2=U1
6803      A2=A1
6804      IF (ICHAR.NE.1) GO TO 60
6805      U(LMAX,M,N3)=U1
6806      P(LMAX,M,N3)=P(LMAX,M,N1)
6807      RO(LMAX,M,N3)=RO(LMAX,M,N1)
6808      A3=A1
6809 C
6810 C      CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
6811 C
6812      60 QUTB=QUT(LMAX,M)
6813      QVTB=QVT(LMAX,M)
6814      OPTB=OPT(LMAX,M)
6815      QOTB=QOT(LMAX,M)
6816      IF (IVC.EQ.0) GO TO 70
6817      IF (M.EQ.MMAX) GO TO 70
6818      IF (NVC.EQ.1.OR.M.NE.MVCT+1) GO TO 70
6819 C
6820 C      LINEAR INTERPOLATION IN TIME FOR M=MVCT+1
6821 C
6822      UB=U1(LMAX)+RIND*(UU2(LMAX)-UU1(LMAX))
6823      VB=V1(LMAX)+RIND*(VV2(LMAX)-VV1(LMAX))
6824      PB=PP1(LMAX)+RIND*(PP2(LMAX)-PP1(LMAX))
6825      ROB=RORO1(LMAX)+RIND*(RORO2(LMAX)-RORO1(LMAX))
6826      ULM=U1(L1)+RIND*(UU2(L1)-UU1(L1))
6827      VLM=V1(L1)+RIND*(VV2(L1)-VV1(L1))
6828      PLM=PP1(L1)+RIND*(PP2(L1)-PP1(L1))
6829      ROLM=RORO1(L1)+RIND*(RORO2(L1)-RORO1(L1))
6830      GO TO 80
6831 C
6832      70 UB=U(LMAX,M,N1)
6833      VB=V(LMAX,M,N1)
6834      PB=P(LMAX,M,N1)
6835      ROB=RO(LMAX,M,N1)
6836      ULM=U(L1,M,N1)
6837      VLM=V(L1,M,N1)
6838      PLM=P(L1,M,N1)
6839      ROLM=RO(L1,M,N1)
6840      80 BU=(UB-ULM)/DXP
6841      BV=(VB-VLM)/DXP
6842      BP=(PB-PLM)/DXP
6843      BRO=(ROB-ROLM)/DXP
6844      CU=UB-BU*X3
6845      CV=VB-BV*X3
6846      CP=PB-BP*X3
6847      CRO=ROB-BRO*X3
6848 C
6849 C      CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
6850 C      COEFFICIENTS
6851 C
6852      IF (M.EQ.1) GO TO 100
6853      IF (M.EQ.MOFS.AND.IB.EQ.4) GO TO 110
6854      IF (IVC.EQ.0) GO TO 90
6855      IF (NVC.EQ.1.OR.M.NE.MVCB) GO TO 90
6856 C
6857 C      LINEAR INTERPOLATION IN TIME FOR M=MVCB
6858 C
6859      UMM=U(LMAX,M-1,NN1)+RIND*(U(LMAX,M-1,NN3)-U(LMAX,M-1,NN1))
6860      VMM=V(LMAX,M-1,NN1)+RIND*(V(LMAX,M-1,NN3)-V(LMAX,M-1,NN1))
6861      PMM=P(LMAX,M-1,NN1)+RIND*(P(LMAX,M-1,NN3)-P(LMAX,M-1,NN1))
6862      ROMM=RO(LMAX,M-1,NN1)+RIND*(RO(LMAX,M-1,NN3)-RO(LMAX,M-1,NN1))
6863      ULMMM=U(L1,M-1,NN1)+RIND*(U(L1,M-1,NN3)-U(L1,M-1,NN1))
6864      VLMMM=V(L1,M-1,NN1)+RIND*(V(L1,M-1,NN3)-V(L1,M-1,NN1))
6865      PLMMM=P(L1,M-1,NN1)+RIND*(P(L1,M-1,NN3)-P(L1,M-1,NN1))
6866      ROLMMM=RO(L1,M-1,NN1)+RIND*(RO(L1,M-1,NN3)-RO(L1,M-1,NN1))
6867 C
6868      DU=(UB-UMM)*DYR
6869      DV=(VB-VMM)*DYR
6870      DP=(PB-PMM)*DYR
6871      DRO=(ROB-ROMM)*DYR
6872      DU1=(ULM-ULMMM)*DYR
6873      DV1=(VLM-VLMMM)*DYR

```

```

6874     DP1=(PLM-PLMMM)*DYP
6875     DRO1=(ROLM-ROLMMM)*DYP
6876     GO TO 120
6877 90  DU=(IB-U(LMAX,M-1,N1))*DYP
6878     DV=(VB-V(LMAX,M-1,N1))*DYP
6879     DP=(PB-P(LMAX,M-1,N1))*DYP
6880     DRO=(ROB-RO(LMAX,M-1,N1))*DYP
6881     DU1=(ULM-U(L1,M-1,N1))*DYP
6882     DV1=(VLM-V(L1,M-1,N1))*DYP
6883     DP1=(PLM-P(L1,M-1,N1))*DYP
6884     DRO1=(ROLM-RO(L1,M-1,N1))*DYP
6885     GO TO 120
6886 100 IF (NGCB.NE.O) GO TO 110
6887     DU=O.O
6888     DV=(4.O*V(LMAX,2,N1)-V(LMAX,3,N1))*O.5*DYP
6889     DP=O.O
6890     DRO=O.O
6891     DU1=O.O
6892     DV1=(4.O*V(L1,2,N1)-V(L1,3,N1))*O.5*DYP
6893     DP1=O.O
6894     DRO1=O.O
6895     GO TO 120
6896 110 DU=(U(LMAX,M+1,N1)-UB)*DYP
6897     DV=(V(LMAX,M+1,N1)-VB)*DYP
6898     DP=(P(LMAX,M+1,N1)-PB)*DYP
6899     DRO=(RO(LMAX,M+1,N1)-ROB)*DYP
6900     DU1=(U(L1,M+1,N1)-ULM)*DYP
6901     DV1=(V(L1,M+1,N1)-VLM)*DYP
6902     DP1=(P(L1,M+1,N1)-PLM)*DYP
6903     DRO1=(RO(L1,M+1,N1)-ROLM)*DYP
6904 120 BDU=(DU-DU1)/DXP
6905     BDV=(DV-DV1)/DXP
6906     BDP=(DP-DP1)/DXP
6907     BDRO=(DRO-DRO1)/DXP
6908     CDU=DU*BDU*X3
6909     CDV=DV*BDV*X3
6910     CDP=DP*BDP*X3
6911     CDRO=DRO*BDRO*X3
6912 C
6913 C     CALCULATE THE COEFFICIENTS FOR THE QUICK SOLVER
6914 C
6915     IF (IOSD.EQ.O.OR.NVC.EQ.1) GO TO 130
6916     IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 130
6917     IF (M.EQ.MOFS.AND.LDFSF.EQ.LMAX) GO TO 130
6918     DUDYQLX=O.5*(DUDYQS(LMAX,M,1)+DUDYQS(LMAX,M,2))
6919     DVDYQLX=O.5*(DVDYQS(LMAX,M,1)+DVDYQS(LMAX,M,2))
6920     DDPYQLX=O.5*(DPDYQS(LMAX,M,1)+DPDYQS(LMAX,M,2))
6921     DUDYQL1=O.5*(DUDYQS(L1,M,1)+DUDYQS(L1,M,2))
6922     DVDYQL1=O.5*(DVDYQS(L1,M,1)+DVDYQS(L1,M,2))
6923     DDPYQL1=O.5*(DPDYQS(L1,M,1)+DPDYQS(L1,M,2))
6924     BDUQS=(DUDYQLX-DUDYQL1)/DXP
6925     BDVQS=(DVDYQLX-DVDYQL1)/DXP
6926     BDPQS=(DPDYQLX-PPDYQL1)/DXP
6927     CDUQS=DUDYQLX-BDUQS*X3
6928     CDVQS=DVDYQLX-BDVQS*X3
6929     CDPQS=DPDYQLX-BDPQS*X3
6930 C
6931 C     CALCULATE X1 AND X2
6932 C
6933 130 IF (ICHAR.NE.1) A3=SQRT(GAMMA*P(LMAX,M,N3)/RO(LMAX,M,N3))
6934     DO 140 IL=1,2
6935     X1=X3-(U(LMAX,M,N3)*OM1+U1*OM1)*O.5*DT
6936     X2=X3-((U(LMAX,M,N3)+A3)*OM1+(U2+A2)*OM1)*O.5*DT
6937     IF (X3-X1.LT.O.O5*DXP) X1=X3-O.05*DXP
6938     IF (X3-X2.LT.O.O5*DXP) X2=X3-O.05*DXP
6939 C
6940 C     INTERPOLATE FOR THE PROPERTIES
6941 C
6942     U1=BU*X1+CU
6943     U2=BU*X2+CU
6944     P2=BP*X2+CP
6945     RQ2=BRQ*X2+CRQ
6946     A2=SQRT(GAMMA*P2/RQ2)

```

```

6947 140 CONTINUE
6948 V1=BV*X1+CV
6949 P1=BP*X1+CP
6950 R01=BR0*X1+CRO
6951 UV1=U1*AL3+V1*BE3+DE3
6952 A1=SQRT(GAMMA*P1/R01)
6953 V2=BV*X2+CV
6954 UV2=U2*AL3+V2*BE3+DE3
6955 C
6956 C INTERPOLATE FOR THE CROSS DERIVATIVES
6957 C
6958 DV1=BDV*X1+CDV
6959 DP1=BDP*X1+CDP
6960 DRO1=BR0*X1+CRO
6961 DU2=BDU*X2+CDU
6962 DV2=BDV*X2+CDV
6963 DP2=BDP*X2+CDP
6964 DRO2=BR0*X2+CRO
6965 C
6966 IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 150
6967 IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 150
6968 IF (M.EQ.MOFS.AND.LDFS.FEQ.LMAX) GO TO 150
6969 DV1QS=BDVQS*X1+CDVQS
6970 DP1QS=BDPQS*X1+CDPQS
6971 DU2QS=BDUQS*X2+CDUQS
6972 DV2QS=BDVQS*X2+CDVQS
6973 DP2QS=BDPQS*X2+CDPQS
6974 C
6975 C CALCULATE THE PSI TERMS
6976 C
6977 150 IF (NDIM.EQ.O) GO TO 170
6978 IF (M.EQ.1.AND.YCB(LMAX).EQ.O.O) GO TO 160
6979 ATERM2=R02*V2/YP
6980 GO TO 170
6981 160 ATERM2=R02*BE3*DV2
6982 170 PSI31=-UV1*DV1-BE3*DP1/R01
6983 PSI41=-UV1*DP1+A1*A1*UV1*DRO1
6984 PSI12=-UV2*DRO2-R02*AL3*DU2-R02*BE3*DV2-ATERM2
6985 PSI22=-UV2*DU2-AL3*DP2/R02
6986 PSI42=-UV2*DP2+A2*A2*UV2*DRO2
6987 C
6988 IF (IQSD.EQ.O.OR.NVC.EQ.1) GO TO 180
6989 IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 180
6990 IF (M.EQ.MOFS.AND.LDFS.FEQ.LMAX) GO TO 180
6991 UV1=U1*ALD+V1*BED+DED
6992 PSI31=-UV1*DV1QS-BE3*DP1QS/R01
6993 PSI12=-UV2*DRO2-R02*ALD*DU2QS-R02*BE3*DV2QS-ATERM2
6994 UV2=U2*ALD+V2*BED+DED
6995 PSI22=-UV2*DU2QS-ALD*DP2QS/R02
6996 180 IF (ICHAR.EQ.1) GO TO 270
6997 C
6998 C CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
6999 C
7000 IF (M.EQ.1.AND.NGCB.EQ.O) GO TO 190
7001 IF (M.EQ.MOFS.AND.IB.EQ.3) GO TO 200
7002 IF (M.EQ.MMAX) GO TO 200
7003 DU3=(U(LMAX,M+1,N3)-U(LMAX,M,N3))*DYS
7004 DV3=(V(LMAX,M+1,N3)-V(LMAX,M,N3))*DYS
7005 DP3=(P(LMAX,M+1,N3)-P(LMAX,M,N3))*DYS
7006 DRO3=(RO(LMAX,M+1,N3)-RO(LMAX,M,N3))*DYS
7007 GO TO 210
7008 190 DU3=0.0
7009 DV3=(4.0*V(LMAX,2,N3)-V(LMAX,3,N3))*0.5*DYS
7010 DP3=0.0
7011 DRO3=0.0
7012 GO TO 210
7013 200 DU3=(U(LMAX,M,N3)-U(LMAX,M-1,N3))*DYS
7014 DV3=(V(LMAX,M,N3)-V(LMAX,M-1,N3))*DYS
7015 DP3=(P(LMAX,M,N3)-P(LMAX,M-1,N3))*DYS
7016 DRO3=(RO(LMAX,M,N3)-RO(LMAX,M-1,N3))*DYS
7017 C
7018 C CALCULATE THE PSI TERMS AT THE SOLUTION POINT

```

```

7019 C
7020 210 IF (NDIM.FO.O) GO TO 230
7021 IF (M.EQ.1.AND.YCB(LMAX).EQ.O.O) GO TO 220
7022 ATERM3=RO(LMAX,M,N3)*V(LMAX,M,N3)/YP
7023 GO TO 230
7024 220 ATERM3=RO(LMAX,1,N3)*BE4*DV3
7025 230 UV3=U(LMAX,M,N3)*AL4+V(LMAX,M,N3)*BE4*DE4
7026 PSI13=-UV3*DRO3-RO(LMAX,M,N3)*(AL4*DU3+BF4*DV3)-ATERM3
7027 PSI23=-UV3*DU3-AL4*DP3/RO(LMAX,M,N3)
7028 PSI33=-UV3*DV3-BE4*DP3/RO(LMAX,M,N3)
7029 PSI43=-UV3*DP3+A3*A3*UV3*DRO3
7030 C
7031 IF (IOSD.EQ.O.OR.NVC.EQ.1) GO TO 250
7032 IF (M.LE.MVCB.OR.M.GE.MVCT) GO TO 250
7033 IF (M.EQ.MDFS.AND.LDFS.EQ.LMAX) GO TO 250
7034 DUDY1=O.5*(U(LMAX,M+1,N3)-UOLD)*DYR
7035 DVDY1=O.5*(V(LMAX,M+1,N3)-VOLD)*DYR
7036 DPDY1=O.5*(P(LMAX,M+1,N3)-POLD)*DYR
7037 IF (MDFS.EQ.O) GO TO 240
7038 IF (M.NE.MDFS+1.OR.LDFS.NE.LMAX) GO TO 240
7039 DUDY1=O.5*(U(LMAX,M+1,N3)-UL(LMAX,N3))*DYR
7040 DVDY1=O.5*(V(LMAX,M+1,N3)-VL(LMAX,N3))*DYR
7041 DPDY1=O.5*(P(LMAX,M+1,N3)-PL(LMAX,N3))*DYR
7042 240 PSI13=-UV3*DRO3-RO(LMAX,M,N3)*(ALD*DUDY1+BED*DVDY1)-ATERM3
7043 UV3=U(LMAX,M,N3)*ALD+V(LMAX,M,N3)*BED*DED
7044 PSI23=-UV3*DUDY1-ALD*DPDY1/RO(LMAX,M,N3)
7045 PSI33=-UV3*DVDY1-BED*DPDY1/RO(LMAX,M,N3)
7046 250 IF (IOSD.EQ.O.OR.NVC.EQ.1) GO TO 260
7047 UOLD=U(LMAX,M,N3)
7048 VOLD=V(LMAX,M,N3)
7049 POLD=P(LMAX,M,N3)
7050 260 PSI31B=(PSI31+PSI33)*O.5+QVTB
7051 PSI41B=(PSI41+PSI43)*O.5
7052 PSI12B=(PSI12+PSI13)*O.5
7053 PSI22B=(PSI22+PSI23)*O.5+QVTB
7054 PSI42B=(PSI42+PSI43)*O.5
7055 GO TO 280
7056 270 PSI31B=PSI31+QVTB
7057 PSI41B=PSI41
7058 PSI12B=PSI12
7059 PSI22B=PSI22+QVTB
7060 PSI42B=PSI42
7061 C
7062 C SOLVE THE COMPATIBILITY EQUATIONS FOR U,V AND RO
7063 C
7064 280 P(LMAX,M,N3)=RNNPE*PE(M)+(1.0-RNNPE)*PEI
7065 AB=O.5*(A2+A3)
7066 ROAVG=O.5*(RO2+RO(LMAX,M,N3))
7067 PSIT=(PSI42B*ROAVG*AB+PSI22B*AB*AB+PSI12B)*DT
7068 IF (ALE.EQ.O.O) GO TO 290
7069 PSIT=PSIT+OPTB*DT
7070 PSI41B=PSI41B+OPTB
7071 P(LMAX,M,N3)=(ALE*PE(M)+ROAVG*AB*(U2-U(LMAX,M,N1))+P2+P(LMAX,M,N1)
7072 1 +PSIT)/(2.0+ALE)
7073 290 RO(LMAX,M,N3)=RO1+2.0*(P(LMAX,M,N3)-P1-DT*PSI41B)/(A3*A3+A1*A1)
7074 1 +OROTB*DT
7075 IF (RO(LMAX,M,N3).LE.O.O) RO(LMAX,M,N3)=ROLOW/G
7076 U(LMAX,M,N3)=U2+(PSIT-P(LMAX,M,N3)+P2)/(ROAVG*AB)
7077 V(LMAX,M,N3)=V1+DT*PSI31B
7078 IF (NOSLIP.EQ.O) GO TO 300
7079 IF (M.EQ.1.AND.NGCB.NE.O) U(LMAX,M,N3)=O.O
7080 IF (M.EQ.MMAX.AND.IWALL.EQ.O) U(LMAX,M,N3)=O.O
7081 IF (M.EQ.MDFS.AND.LDFS.EQ.LMAX) U(LMAX,M,N3)=O.O
7082 C
7083 C CHECK FOR INFLOW AND IF SO, SET THE CORRECT BOUNDARY CONDITIONS
7084 C
7085 300 IF (U(LMAX,M,N3).GE.O.O) GO TO 320
7086 RO(LMAX,M,N3)=O.5*(RO(LMAX,1,N1)+RO(LMAX,MMAX,N1))
7087 IF (U(LMAX,2,N1).GT.O.O.AND.U(LMAX,M1,N1).LT.O.O) RO(LMAX,M,N3)=RO
7088 1 (LMAX,MMAX,N1)
7089 IF (U(LMAX,2,N1).LT.O.O.AND.U(LMAX,M1,N1).GT.O.O) RO(LMAX,M,N3)=RO
7090 1 (LMAX,1,N1)

```

```

7091      V(LMAX,M,N3)=-U(LMAX,M,N3)+(NXNYCB(LMAX)+(YP-YCB(LMAX)))/(YW(LMAX)
7092      1 -YCB(LMAX))+(NXNY(LMAX)-NXNYCB(LMAX)))
7093      IF (MOFS.EQ.O.OR.LDFSF.NE.LMAX) GO TO 320
7094      IF (IB.EQ.4) GO TO 310
7095      RO(LMAX,M,N3)=O.5*(RO(LMAX,1,N1)+RO(LMAX,MOFS,N1))
7096      IF (U(LMAX,2,N1).GT.O.O.AND.U(LMAX,MOFS-1,N1).LT.O.O) RO(LMAX,M,N3
7097      1 )=RO(LMAX,MOFS,N1)
7098      IF (U(LMAX,2,N1).LT.O.O.AND.U(LMAX,MOFS-1,N1).GT.O.O) RO(LMAX,M,N3
7099      1 )=RO(LMAX,1,N1)
7100      V(LMAX,M,N3)=-U(LMAX,M,N3)+(NXNYCB(LMAX)+(YP-YCB(LMAX)))/(YL(LMAX)
7101      1 -YCB(LMAX))+(NXNY(LMAX)-NXNYCB(LMAX)))
7102      GO TO 320
7103      310 RO(LMAX,M,N3)=O.5*(RO(LMAX,MOFS,N1)+RO(LMAX,MMAX,N1))
7104      IF (U(LMAX,MOFS+1,N1).GT.O.O.AND.U(LMAX,M1,N1).LT.O.O) RO(LMAX,M
7105      1 ,N3)=RO(LMAX,MMAX,N1)
7106      IF (U(LMAX,MOFS+1,N1).LT.O.O.AND.U(LMAX,M1,N1).GT.O.O) RO(LMAX,M
7107      1 ,N3)=RO(LMAX,MOFS,N1)
7108      V(LMAX,M,N3)=U(LMAX,M,N3)+(N*Y(U(LMAX)+(Y*Y(U(LMAX)))/(Y*Y(U(LMAX))
7109      1 (LMAX)))+(NXNY(LMAX)-NXNY(U(LMAX)))
7110 C
7111 C      AVERAGE THE SOLUTION IF THE MACH NUMBER IS ALTERNATING
7112 C      ABOVE AND BELOW 1.0
7113 C
7114      320 IF (ICHP EQ 1 OR IFXIT NE 0) GO TO 330
7115      SM3=U(LMAX,M,N3)+.2/(GAMMA-P(LMAX,M,N3)/RO(LMAX,M,N3))
7116      IF (SM3 LT 1.0.AND.SM LT 1.0) GO TO 330
7117      IF (-SM3 GT 1.0.AND.SM GT 1.0) GO TO 330
7118      P(LMAX,M,N3)=RNNDP*PE(M)+(1.0-RNNDP)*PE
7119      330 CONTINUE
7120 C
7121 C      SET BOUNDARY CONDITIONS AT THE CORNER MESH POINTS
7122 C
7123      IF (IWALL EQ 0) GO TO 340
7124      IF (V(LMAX,MMAX,N1).GT.O.O) GO TO 340
7125      ND=111
7126      IF (ICHP EQ 2) ND=N3
7127      U(LMAX,MMAX,N3)=O.1*U(1,MMAX,ND)+O.9*U(LMAX,MMAX,N1)
7128      RO(LMAX,MMAX,N3)=O.1*RO(1,MMAX,ND)+O.9*RO(LMAX,MMAX,N1)
7129      340 IF (INVC EQ 1.AND.MVCT EQ MMAX) GO TO 350
7130      IF (MOFS NE 0.AND.IB EQ 3) GO TO 350
7131      IF (IWALL EQ 0) V(LMAX,MMAX,N3)=U(LMAX,MMAX,N3)+NXNY(LMAX)*XW1
7132      1 (LMAX)
7133      IF (TW(1).GT.O.O.AND.P(LMAX,MMAX,N3).EQ.PE(MMAX)) RO(LMAX,MMAX,N3)
7134      1 =P(LMAX,MMAX,N3)/(RG*TW(LMAX))
7135      IF (TW(1).GT.O.O.AND.P(LMAX,MMAX,N3).NE.PE(MMAX)) P(LMAX,MMAX,N3)
7136      1 =RO(LMAX,MMAX,N3)+RG*TW(LMAX)
7137      350 IF (INVC EQ 1.AND.MVCR EQ 1) GO TO 360
7138      IF (MOFS NE 0.AND.IB EQ 4) GO TO 360
7139      V(LMAX,1,N3)=U(LMAX,1,N3)+NXNYCB(LMAX)
7140      IF (ICR(1).GT.O.O.AND.P(LMAX,1,N3).EQ.PE(1)) RO(LMAX,1,N3)=P(LMAX,
7141      1 ,N3)/(RG+ICR(LMAX))
7142      IF (ICR(1).GT.O.O.AND.P(LMAX,1,N3).NE.PE(1)) P(LMAX,1,N3)=RO(LMAX,
7143      1 ,N3)+RG+ICR(LMAX)
7144 C
7145 C      SET BOUNDARY CONDITIONS FOR THE DUAL FLOW SPACE
7146 C
7147      360 IF (MOFS.EQ.O.OR.LDFSF.NE.LMAX) RETURN
7148      IF (INVC.EQ.1.AND.(MOFS.GT.MVCR.AND.MOFS.LT.MVCT)) RETURN
7149      IF (IB.EQ.4) GO TO 370
7150      V(LMAX,MOFS,N3)=U(LMAX,MOFS,N3)+NXNY(LMAX)
7151      IF (TL(1).GT.O.O.AND.P(LMAX,MOFS,N3).EQ.PE(MOFS)) RO(LMAX,MOFS,N3)
7152      1 =P(LMAX,MOFS,N3)/(RG+TL(LMAX))
7153      IF (TL(1).GT.O.O.AND.P(LMAX,MOFS,N3).NE.PE(MOFS)) P(LMAX,MOFS,N3)
7154      1 =RO(LMAX,MOFS,N3)+RG+TL(LMAX)
7155      RETURN
7156      370 V(LMAX,MOFS,N3)=U(LMAX,MOFS,N3)+NXNY(U(LMAX))
7157      IF (IU(1).GT.O.O.AND.P(LMAX,MOFS,N3).EQ.PE(MOFS)) RO(LMAX,MOFS,N3)
7158      1 =P(LMAX,MOFS,N3)/(RG+IU(LMAX))

```

```
7159     IF (TU(1).GT.0.0.AND.P(LMAX,MDFS,N3).NE.PE(MDFS)) P(LMAX,MDFS,N3)
7160     1 =RO(LMAX,MDFS,N3)*RG*TU(LMAX)
7161     RETURN
7162 C
7163 380 FORMAT (1H0,57H***** A NEG SOUND SPEED OCCURED IN SUBROUTINE FXITT
7164 1 AT N=,I6.4H, M=,I2.6H, NVC=,I3.11H AND ICHAR=,I1.6H ***** )
7165     END
```



```

7166      SUBROUTINE QSOLVE
7167 C
7168 C      .....
7169 C
7170 C      THIS SUBROUTINE CALCULATES THE VELOCITY AND PRESSURE DERIVATIVES
7171 C      IN THE SUBCYCLED MESH AS PART OF THE QUICK SOLVER PACKAGE
7172 C
7173 C      .....
7174 C
7175 *CALL MCC
7176      IP=1
7177      YWB=0.0
7178      YWT=1.0
7179      Y1=0.0
7180      Y2=0.0
7181      Y10=0.0
7182      Y20=0.0
7183      MIS=MVCB1
7184      MIF=MVCT1
7185      IF (MDFS.EQ.0) GO TO 20
7186 C
7187      IB=3
7188      CALL SWITCH (3)
7189      GO TO 20
7190 10 MIS=MDFS+1
7191      MIF=MVCT1
7192      IB=4
7193      YWB=Y(MDFS)
7194      YWT=1.0
7195      CALL SWITCH (3)
7196 C
7197 C      BEGIN THE L OR X DO LOOP
7198 C
7199 20 DO 510 L=1,LMAX
7200      LMAP=L
7201      LDFS=0
7202      IF (L.GE.LDFS.AND.L.LE.LDFS) LDFS=1
7203      YPB=YCB(L)
7204      YPT=YW(L)
7205      IF (MDFS.EQ.0) GO TO 50
7206      IF (LDFS.NE.0) GO TO 30
7207      IF (IB.EQ.4) GO TO 510
7208      MIF=MVCT1
7209      YWT=1.0
7210      GO TO 50
7211 30 IF (IB.EQ.4) GO TO 40
7212      MIF=MDFS-1
7213      YWT=Y(MDFS)
7214      YPT=YL(L)
7215      GO TO 50
7216 40 YPB=YU(L)
7217 50 IF (MVCB.NE.1) GO TO 60
7218      MMAP=1
7219      MM=1
7220      RFLD=-2.0*NXNYCB(L)/(1.0+NXYCB(L)**2)
7221      GO TO 80
7222 50 IF (MVCT.NE.MMAX) GO TO 70
7223      MMAP=MMAX
7224      MM=MMAX
7225      RFLD=2.0*NXNY(L)/(1.0+NXY(L)**2)
7226      GO TO 80
7227 70 IF (MDFS.EQ.0) GO TO 110
7228      IF (LDFS.EQ.0) GO TO 110
7229      MMAP=MDFS
7230      MM=MDFS
7231      IF (IB.EQ.3) RFLD=2.0*NXNYL(L)/(1.0+NXYL(L)**2)
7232      IF (IB.EQ.4) RFLD=-2.0*NXNYU(L)/(1.0+NXYU(L)**2)
7233 80 CALL MAP
7234      OM11=2.0*OM1*OM2/(OM1+OM2)
7235      AL11=AL3
7236      BE11=BE3
7237      DE11=DE3

```

```

7238     ALS11=SQRT(AL11*AL11+BE11*BE11)
7239     UV11=DE11
7240     RFLD=RFLD/BE11
7241     IF (L.EQ.1) GO TO 90
7242     IF (L.EQ.LMAX) GO TO 100
7243     PTERM=0.5*OM11*(P(L+1,MM,N1)-P(L-1,MM,N1))*DXR
7244     ROTERM=0.5*OM11*(RO(L+1,MM,N1)-RO(L-1,MM,N1))*DXR
7245     QTERM=0.5*OM11*(Q(L+1,MM,N1)-Q(L-1,MM,N1))*DXR
7246     GO TO 110
7247     90 PTERM=OM11*(P(2,MM,N1)-P(1,MM,N1))*DXR
7248     ROTERM=OM11*(RO(2,MM,N1)-RO(1,MM,N1))*DXR
7249     QTERM=OM11*(Q(2,MM,N1)-Q(1,MM,N1))*DXR
7250     GO TO 110
7251     100 PTERM=OM11*(P(LMAX,MM,N1)-P(L1,MM,N1))*DXR
7252     ROTERM=OM11*(RO(LMAX,MM,N1)-RO(L1,MM,N1))*DXR
7253     QTERM=OM11*(Q(LMAX,MM,N1)-Q(L1,MM,N1))*DXR
7254 C
7255 C     BEGIN THE M OR Y DO LOOP
7256 C
7257     110 DO 500 M=MIS,MIF
7258     MMAP=M
7259     CALL MAP
7260     BE=2.0*BE3*BE4/(BE3+BE4)
7261     BFD=BE3
7262     YPD=YP
7263     Y3=Y(M)
7264     YPP=YP+DY/BE4
7265     YPM=YP-DY/BE3
7266 C
7267     U3=U(L,M,N1)
7268     V3=V(L,M,N1)
7269     P3=P(L,M,N1)
7270     R03=RO(L,M,N1)
7271     Q3=Q(L,M,N1)
7272     A3=SQRT(GAMMA*P3/R03)
7273     UV3=U3*AL3+V3*BE3+DE3
7274     ALS=SQRT(AL3*AL3+BE3*BE3)
7275     UV3D=U3*AL4+V3*BE4+DE4
7276     ALSD=SQRT(AL4*AL4+BE4*BE4)
7277 C
7278 C     CALCULATE Y1 (SECANT - FALSE POSITION METHOD)
7279 C
7280     ILLI=0
7281     MMO=0
7282     DO 270 ILL=1,ILLOS
7283     IF (ILLI.NE.0) GO TO 150
7284     IF (ILL.NE.1) GO TO 120
7285     UVAO=(UV3+ALS*A3)*DT
7286     Y100=Y3
7287     FY3=-UVAO
7288     Y1=Y(M-1)
7289     GO TO 190
7290     120 UVAVG=0.5*((U1+U3)*ALAVG+(V1+V3)*BEAVG)+DEAVG
7291     UVA=(UVAVG+ALSA1)*DT
7292     FY1=Y3-UVA-Y1
7293     IF (FY1*FY3.LT.0.0) GO TO 140
7294     UVAO=UVA
7295     Y100=Y1
7296     FY3=FY1
7297     IF (ILL.LT.M) Y1=Y(M-ILL)
7298     IF (2+ILL-M.EQ.MMAX+1) GO TO 130
7299     IF (ILL.GE.M) Y1=2.0*YWB-Y(2+ILL-M)
7300     GO TO 190
7301     130 NP=N+NSTART
7302     WRITE (6,560) NP,L,M,NVC
7303     IERR=1
7304     RETURN
7305     140 ILLI=1
7306     Y10=Y1
7307     GO TO 180
7308     150 UVAVG=0.5*((U1+U3)*ALAVG+(V1+V3)*BEAVG)+DEAVG
7309     UVAT=(UVAVG+ALSA1)*DT

```

```

7310      FY1=Y3-UVAT-Y1
7311      FY10=Y3-UVA-Y10
7312      IF (FY1-FY10.LT.O.O) GO TO 160
7313      GO TO 170
7314      160 UVAO=UVA
7315      Y100=Y10
7316      170 UVA=UVAT
7317      Y10=Y1
7318      180 Y1=Y10+(Y10-Y100)*(Y3-UVA-Y10)/(UVA-UVAO+Y10-Y100)
7319      IF (Y1.LT.2.O*YWB-Y(MVCT)) Y1=2.O*YWB-Y(MVCT)
7320      IF (MVCB.NE.1.AND.Y1.LT.Y(MVCB)) Y1=Y(MVCB)
7321      IF (Y1.GT.Y(M1)) Y1=Y(M1)
7322      IF (Y1+Y10.EQ.O.O) GO TO 290
7323      IF (Y10.EQ.O.O) GO TO 190
7324      IF (ABS((Y1-Y10)/Y10).LE.COS) GO TO 290
7325 C
7326 C      INTERPOLATE FOR THE PROPERTIES AT Y=Y1
7327 C
7328      190 IY1=0
7329      IF (Y1.GE.YWB) GO TO 200
7330      Y1=2.O*YWB-Y1
7331      IY1=1
7332      200 DO 210 MM=1,M1
7333      IF (Y1.GE.Y(MM).AND.Y1.LE.Y(MM+1)) GO TO 220
7334      210 CONTINUE
7335      220 RDY=(Y1-Y(MM))*DVR
7336      U1=U(L,MM,N1)+(U(L,MM+1,N1)-U(L,MM,N1))*RDY
7337      V1=V(L,MM,N1)+(V(L,MM+1,N1)-V(L,MM,N1))*RDY
7338      P1=P(L,MM,N1)+(P(L,MM+1,N1)-P(L,MM,N1))*RDY
7339      RO1=RO(L,MM,N1)+(RO(L,MM+1,N1)-RO(L,MM,N1))*RDY
7340      O1=O(L,MM,N1)+(O(L,MM+1,N1)-O(L,MM,N1))*RDY
7341      IF (IY1.EQ.O) GO TO 230
7342      U1=-U1
7343      V1=-V1
7344      RFL=RFLD*(Y1-YWB)
7345      P1=P1-PTERM*RFL
7346      RO1=RO1-ROTERM*RFL
7347      O1=O1-OTERM*RFL
7348      230 IF (MM.EQ.MM0) GO TO 240
7349      MM0=MM
7350      MMAP=MM
7351      IP=0
7352      CALL MAP
7353      YPMM=YP
7354      MMAP=MM+1
7355      IP=1
7356      CALL MAP
7357      YPMM1=YP
7358      240 YP1=YPMM+(YPMM1-YPMM)*RDY
7359      IF (IY1.EQ.C) GO TO 250
7360      Y1=2.O*YWB-Y1
7361      YP1=2.O*YPB-YP1
7362      250 IF (YPD.EQ.YP1) GO TO 280
7363      BEAVG=(Y3-Y1)/(YPD-YP1)
7364      ALAVG=AL3+BEAVG/BE3
7365      DEAVG=DE3+BEAVG/BE3
7366      A1D=GAMMA*P1/RO1
7367      IF (A1D.GT.O.O) GO TO 260
7368      NP=N+NSTART
7369      WRITE (6,520) NP,L,M,NVC
7370      IERR=1
7371      RETURN
7372      260 ALSA1=SQRT(O.5*(A1D+A3*A3)*(ALAVG*ALAVG+BEAVG*BEAVG))
7373      270 CONTINUE
7374      280 NP=N+NSTART
7375      WRITE (6,540) ILLOS.NP,L,M,NVC
7376      IERR=1
7377      RETURN
7378 C
7379 C      CALCULATE DUDYQS, DVDYQS AND DPOYQS AT Y=Y1
7380 C
7381      290 U3D=U3

```

```

7382      V3D=V3
7383      P3D=P3
7384      R03D=R03
7385      Q3D=Q3
7386      IF (Y1.GE.Y(M-1)) GO TO 300
7387      U3D=SQS*U3+(1.0-SQS)*(U(L,M-1,N1)+(U(L,M+1,N1)-U(L,M-1,N1))*(YPP
7388      1 -YPM)/(YPP-YPM))
7389      V3D=SQS*V3+(1.0-SQS)*(V(L,M-1,N1)+(V(L,M+1,N1)-V(L,M-1,N1))*(YPP
7390      1 -YPM)/(YPP-YPM))
7391      P3D=SQS*P3+(1.0-SQS)*(P(L,M-1,N1)+(P(L,M+1,N1)-P(L,M-1,N1))*(YPP
7392      1 -YPM)/(YPP-YPM))
7393      R03D=SQS*R03+(1.0-SQS)*(R0(L,M-1,N1)+(R0(L,M+1,N1)-R0(L,M-1,N1))*
7394      1 (YPP-YPM)/(YPP-YPM))
7395      Q3D=SQS*Q3+(1.0-SQS)*(Q(L,M-1,N1)+(Q(L,M+1,N1)-Q(L,M-1,N1))*(YPP
7396      1 -YPM)/(YPP-YPM))
7397      300 R0YD=1.0/((YPP-YPM)*BED)
7398      DUDYQS(L,M,1)=(U3D-U1)*R0YD
7399      DVYQS(L,M,1)=(V3D-V1)*R0YD
7400      DPDYQS(L,M,1)=(P3D-P1)*R0YD
7401      DROODY1=(R03D*Q3D-R01*Q1)/((YPP-YPM)*BE)
7402 C
7403 C      CALCULATE Y2 (SECANT - FALSE POSITION METHOD)
7404 C
7405      ILLI=0
7406      MMO=0
7407      DO 460 ILL=1,ILLQS
7408      IF (ILLI.NE.0) GO TO 340
7409      IF (ILL.NE.1) GO TO 310
7410      UVA0=(UV3D-ALSD*A3)*DT
7411      Y200=Y3
7412      FY3=-UVA0
7413      Y2=Y(M+1)
7414      GO TO 380
7415      310 UVAVG=0.5*((U2+U3)*ALAVG+(V2+V3)*BEAVG)+DEAVG
7416      UVA=(UVAVG-ALSA2)*DT
7417      FY2=Y3-UVA-Y2
7418      IF (FY2*FY3.LT.0.0) GO TO 330
7419      UVA0=UVA
7420      Y200=Y2
7421      FY3=FY2
7422      IF (M+ILL.LE.MMAX) Y2=Y(M+ILL)
7423      IF (MMAX+MMAX-M-ILL.EQ.0) GO TO 320
7424      IF (M+ILL.GT.MMAX) Y2=2.0*YWT-Y(MMAX+MMAX-M-ILL)
7425      GO TO 380
7426      320 NP=N+NSTART
7427      WRITE (6,570) NP,L,M,NVC
7428      IERR=1
7429      RETURN
7430      330 ILLI=1
7431      Y20=Y2
7432      GO TO 370
7433      340 UVAVG=0.5*((U2+U3)*ALAVG+(V2+V3)*BEAVG)+DEAVG
7434      UVAT=(UVAVG-ALSA2)*DT
7435      FY2=Y3-UVAT-Y2
7436      FY20=Y3-UVA-Y20
7437      IF (FY2*FY20.LT.0.0) GO TO 350
7438      GO TO 360
7439      350 UVA0=UVA
7440      Y200=Y20
7441      360 UVA=UVAT
7442      Y20=Y2
7443      370 Y2=Y20+(Y20-Y200)*(Y3-UVA-Y20)/(UVA-UVA0+Y20-Y200)
7444      IF (Y2.GT.2.0*YWT-Y(MVCB)) Y2=2.0*YWT-Y(MVCB)
7445      IF (MVCT.NE.MMAX.AND.Y2.GT.Y(MVCT)) Y2=Y(MVCT)
7446      IF (Y2.LT.Y(2)) Y2=Y(2)
7447      IF (ABS((Y2-Y20)/Y20).LE.COS) GO TO 480
7448 C
7449 C      INTERPOLATE FOR THE PROPERTIES AT Y=Y2
7450 C
7451      380 IY2=0
7452      IF (Y2.LE.YWT) GO TO 390
7453      Y2=2.0*YWT-Y2

```

```

7454      IY2=1
7455 390 DO 400 MM=1,M1
7456      IF (Y2.GE.Y(MM).AND.Y2.LE.Y(MM+1)) GO TO 410
7457 400 CONTINUE
7458 410 RDY=(Y2-Y(MM))*DYR
7459      U2=U(L,MM,N1)+(U(L,MM+1,N1)-U(L,MM,N1))*RDY
7460      V2=V(L,MM,N1)+(V(L,MM+1,N1)-V(L,MM,N1))*RDY
7461      P2=P(L,MM,N1)+(P(L,MM+1,N1)-P(L,MM,N1))*RDY
7462      RO2=RO(L,MM,N1)+(RO(L,MM+1,N1)-RO(L,MM,N1))*RDY
7463      O2=O(L,MM,N1)+(O(L,MM+1,N1)-O(L,MM,N1))*RDY
7464      IF (IY2.EQ.O) GO TO 420
7465      U2=-U2
7466      V2=-V2
7467      RFL=RFLD*(YWT-Y2)
7468      P2=P2-PTERM*RFL
7469      RO2=RO2-ROTERM*RFL
7470      O2=O2-OTERM*RFL
7471 420 IF (MM.EQ.MMO) GO TO 430
7472      MMO=MM
7473      MMAP=MM
7474      IP=O
7475      CALL MAP
7476      YPMM=YP
7477      MMAP=MM+1
7478      IP=1
7479      CALL MAP
7480      YPMM1=YP
7481 430 YP2=YPMM+(YPMM1-YPMM)*RDY
7482      IF (IY2.EQ.O) GO TO 440
7483      Y2=2.O*YWT-Y2
7484      YP2=2.O*YPT-YP2
7485 440 IF (YP2.EQ.YPD) GO TO 470
7486      BEAVG=(Y2-Y3)/(YP2-YPD)
7487      ALAVG=AL3*BEAVG/BE3
7488      DEAVG=DE3*BEAVG/BE3
7489      A2D=GAMMA*P2/RO2
7490      IF (A2D.GT.O.O) GO TO 450
7491      NP=N+NSTART
7492      WRITE (6,530) NP,L,M,NVC
7493      IERR=1
7494      RETURN
7495 450 ALSA2=SQRT(O.5*(A2D+A3*A3)*(ALAVG*ALAVG+BEAVG*BEAVG))
7496 460 CONTINUE
7497 470 NP=N+NSTART
7498      WRITE (6,550) ILLOS,NP,L,M,NVC
7499      IERR=1
7500      RETURN
7501 C
7502 C      CALCULATE DUDYQS, DVDYQS, AND DDPYQS AT Y=Y2
7503 C
7504 480 U3D=U3
7505      V3D=V3
7506      P3D=P3
7507      RO3D=RO3
7508      Q3D=Q3
7509      IF (Y2.LE.Y(M+1)) GO TO 490
7510      U3D=SQS*U3+(1.O-SQS)*(U(L,M-1,N1)+(U(L,M+1,N1)-U(L,M-1,N1))*(YPD
7511 1 -YPM)/(YPP-YPM))
7512      V3D=SQS*V3+(1.O-SQS)*(V(L,M-1,N1)+(V(L,M+1,N1)-V(L,M-1,N1))*(YPD
7513 1 -YPM)/(YPP-YPM))
7514      P3D=SQS*P3+(1.O-SQS)*(P(L,M-1,N1)+(P(L,M+1,N1)-P(L,M-1,N1))*(YPD
7515 1 -YPM)/(YPP-YPM))
7516      RO3D=SQS*RO3+(1.O-SQS)*(RO(L,M-1,N1)+(RO(L,M+1,N1)-RO(L,M-1,N1))*
7517 1 (YPD-YPM)/(YPP-YPM))
7518      Q3D=SQS*Q3+(1.O-SQS)*(O(L,M-1,N1)+(O(L,M+1,N1)-O(L,M-1,N1))*(YPD
7519 1 -YPM)/(YPP-YPM))
7520 490 RDYD=1.O/((YP2-YPD)*BED)
7521      DUDYQS(L,M,2)=(U2-U3D)*RDYD
7522      DVDYQS(L,M,2)=(V2-V3D)*RDYD
7523      DDPYQS(L,M,2)=(P2-P3D)*RDYD
7524      DROQDY2=(RO2-Q2-RO3D-Q3D)/((YP2-YPD)*BE)
7525      OQT(L,M)=O.5*(DROQDY1+DROQDY2)

```

```

7526 C
7527 500 CONTINUE
7528 510 CONTINUE
7529 IF (MOFS.NE.O.AND.MIS.EQ.MVCB1) GO TO 10
7530 RETURN
7531 C
7532 C FORMAT STATEMENTS
7533 C
7534 520 FORMAT (1H0,63H***** A NEG SOUND SPEED (A1) OCCURED IN SUBROUTINE
7535 1QSOLVE AT N=.16,4H, L=.12,4H, M=.12,9H AND NVC=.13,6H *****)
7536 530 FORMAT (1H0,63H***** A NEG SOUND SPEED (A2) OCCURED IN SUBROUTINE
7537 1QSOLVE AT N=.16,4H, L=.12,4H, M=.12,9H AND NVC=.13,6H *****)
7538 540 FORMAT (1H0,84H***** THE CHARACTERISTIC SOLUTION FOR Y1 IN SUBROUT
7539 1INE QSOLVE FAILED TO CONVERGE IN .12,17H ITERATIONS AT N=.16,4H, L
7540 2=.12,4H, M=.12,/.7X,6H, NVC=.13,6H *****)
7541 550 FORMAT (1H0,84H***** THE CHARACTERISTIC SOLUTION FOR Y2 IN SUBROUT
7542 1INE QSOLVE FAILED TO CONVERGE IN .12,17H ITERATIONS AT N=.16,4H, L
7543 2=.12,4H, M=.12,/.7X,6H, NVC=.13,6H *****)
7544 560 FORMAT (1H0,59H***** THE SOLUTION FOR Y1 FAILED IN SUBROUTINE QSOL
7545 1VE AT N=.16,4H, L=.12,4H, M=.12,6H, NVC=.13,6H *****)
7546 570 FORMAT (1H0,59H***** THE SOLUTION FOR Y2 FAILED IN SUBROUTINE QSOL
7547 1VE AT N=.16,4H, L=.12,4H, M=.12,6H, NVC=.13,6H *****)
7548 END

```

CASE NO. 1 - CONVERGING-DIVERGING NOZZLE (45 DEG INLET, 15 DEG EXIT)  
 \$CNTRL LMAX=21,MMAX=6,NMAX=40C,TCONV=0.003 \$  
 \$IVS \$  
 \$GENTRY 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,OT=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,PTD1=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 \$  
 \$GENTRY NGEOM=1,XI=36.0,XE=72.0,RI=18.0 \$  
 \$GCBL NGB=4,  
 YCB=13\*3.0,2.9872,2.5487,2.8844,2.7943,2.6782,2.5357,2.3667,2.1707,  
 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=568.95,592.2,593.1,593.7,594.3,594.9,595.8,18\*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 ITN=1,MLM=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 MID=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,RDL=5\*1.2047.  
 U(1,5,1)=5\*7.5859,36\*15.0.  
 U(1,6,1)=5\*7.5859,36\*11.0 \$  
 \$GENTRY 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,FSOL=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,MDFS=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,  
 2.3813,2.8816,3.4072,3.9594,4.5395,5.1489,5.7991,6.4617,7.1683,7.9107,  
 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 \$



**END**

**DATE FILMED**

**02 / 17 / 82**



3 1176 01342 5849

**DO NOT REMOVE SLIP FROM MATERIAL**

Delete your name from this slip when returning material to the library.

NAME	MS
<del>K. Song</del>	<del>493</del>