NASA Technical Memorandum 85708

NASA-TM-85708 19860014094

# User's Guide for NASCRIN - A Vectorized Code for Calculating Two-Dimensional Supersonic Internal Flow Fields

Ajay Kumar

FOR EARLY DOMESTIC DISSEMINATION

Because of its significant early commercial potential, this information, which has been developed under a U.S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations.

Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.

Date for general release February 28, 1986

FEBRUARY 1984

LIBRARY COPY

1 FEB 9 1984

LANGLEY RESEARCH CENTER
LIBRARY NASA
HAMPTON, VIRGINIA

NASA

NASA Technical Memorandum 85708

# User's Guide for NASCRIN - A Vectorized Code for Calculating Two-Dimensional Supersonic Internal Flow Fields

Ajay Kumar
*Langley Research Center*
*Hampton, Virginia*

National Aeronautics
and Space Administration

**Scientific and Technical**
**Information Office**

1984

## SUMMARY

A computer code has been developed for analyzing two-dimensional flow fields in supersonic combustion ramjet (scramjet) inlets. The code is written in the CYBER 200 FORTRAN language for the CDC® CYBER 203 vector-processing computer system. It solves the two-dimensional Euler or Navier-Stokes equations in conservation form by an unsplit, explicit, two-step finite-difference method. A more recent explicit-implicit, two-step scheme has also been incorporated in the code for the viscous flow analysis. An algebraic, two-layer eddy-viscosity model is used for the turbulent flow calculations. The code can analyze both inviscid and viscous flows with no strut, one strut, or multiple struts embedded in the flow field. Although the code is primarily written for supersonic internal flow, it can be used for a variety of other flow problems provided that suitable modifications are made to the boundary conditions. This report contains a listing of the program, descriptions of the input and output parameters, and a sample flow-field solution.


## INTRODUCTION

A two-dimensional computer code NASCRIN (Numerical Analysis of SCRamjet INlet) has been developed to analyze scramjet-inlet flow fields. The code is based on the analysis developed in references 1 and 2. The analysis as such uses the two-dimensional Euler or Navier-Stokes equations in conservation form to describe the inlet flow. A two-layer eddy-viscosity model due to Baldwin and Lomax (ref. 3) is used for the turbulent flow calculation. In order to facilitate the use of a general inlet geometry with embedded bodies, an algebraic coordinate transformation is used which generates a set of boundary-fitted curvilinear coordinates (ref. 4). It transforms the physical domain into a rectangular domain with uniform mesh spacing, and embedded bodies in the flow field are transformed into slits (fig. 1). The transformation also allows for concentrating mesh lines in regions of high gradients. The transformed governing equations are solved by an unsplit, explicit, two-step finite-difference method due to MacCormack (ref. 5). This method is highly efficient on vector-processing computers, as it allows a high degree of vectorization. A more recent explicit-implicit scheme (refs. 6 and 7) has also been incorporated in the code for the viscous flow analysis and can be used upon the option of the user. As discussed in references 1 and 2, the code can also be used in a quasi-three-dimensional sense for a class of scramjet inlets under certain simplifying assumptions.

The code is written in the CYBER 200 FORTRAN language for the CDC® CYBER 203 vector-processing computer system at the Langley Research Center. In its present form, it can analyze two-dimensional inviscid and viscous (laminar and turbulent) flows with no strut, one strut, or multiple struts in the flow field. Although the code is primarily written for supersonic internal flow, it can be used for a variety of other flow problems, including subsonic inflow with supersonic outflow, supersonic inflow with subsonic outflow, or fully subsonic flow, by suitably modifying the boundary condition subroutines.

This report contains a brief description of the code structure, various input parameters, and output flow quantities. A listing of the program can be found in appendix A. A sample case is also presented to illustrate the use of the code.

## CODE DESCRIPTION

The code listing given in appendix A is primarily set up for supersonic flow, although by making suitable changes in the boundary condition subroutines, it can be used for supersonic-subsonic or fully subsonic flow. For the supersonic flow case, the conditions at the inflow boundary are held fixed at some known values, and extrapolation is used at the outflow boundary. The code assumes a solid top boundary from the second grid point to the last grid point in the axial direction. In the case of a symmetric problem, the code solves the upper half of the flow field. (See examples in appendix B.)

The code consists of a main program, NASCRIN, and 20 subroutines. It has been structured in such a way that the user has to make changes only in the first few subroutines for a particular problem. A brief description of program NASCRIN and the various subroutines is given here in the order they appear in the code.

### Program NASCRIN

Most input for parameters such as type of geometry, flow, and grid is done in NASCRIN. Each of these input parameters is discussed below. SI units are used throughout the code.

| Line no. | Parameter | Description |
|---|---|---|
| NASCRIN 52 | N1 | Number of grid points in x-coordinate direction. It is also the second dimension in various arrays. |
| NASCRIN 53 | M1 | Number of grid points in y-coordinate direction. It is also the first dimension in various arrays. |
| NASCRIN 57 | SWEEP | Sidewall sweep angle, $\Lambda$ (fig. 1). If the code is to be used in a quasi-three-dimensional sense, define SWEEP in degrees; otherwise, set it to be zero. |
| NASCRIN 59 | NSTRUT | NSTRUT = 0 for no strut embedded in the flow field;<br>NSTRUT = 1 for a strut embedded in the flow field. |
| NASCRIN 62<br>NASCRIN 63<br>NASCRIN 64 | NS1<br>NS2<br>MS | If NSTRUT = 0, NS1, NS2, and MS are not required. For NSTRUT = 1, NS1 and NS2 are grid points in the x-direction between which the strut lies. MS is the grid point in the y-direction through which an axial mesh line passes that is coincident with the top surface of the strut. (See examples in appendix B.) |
| NASCRIN 69 | LSYM | A parameter for specifying the type of lower boundary. If the flow is symmetric about the lower boundary and only the upper half of the flow is being calculated, set LSYM = 1; otherwise, LSYM = 0. |
| NASCRIN 73 | NCWL | If the lower boundary is fully or partially solid, set NCWL = 1. If it is a free boundary, set NCWL = 0. |

| Line no. | Parameter | Description |
|---|---|---|
| NASCRIN 74<br>NASCRIN 75 | NCWL1<br>NCWL2 | If NCWL = 0, NCWL1 and NCWL2 are not required. If NCWL = 1, NCWL1 and NCWL2 are the grid points in the x-direction between which there is a solid lower boundary (cowl, strut). |
| NASCRIN 117 | NFLOW | NFLOW = 0 for inviscid flow; NFLOW = 1 for laminar flow; and NFLOW = 2 for turbulent flow. If NFLOW = 2, it is necessary to modify calls to the subroutine EDDY, which calculates eddy viscosity. These calls are made in subroutine TURB. |
| NASCRIN 123 | LFI | LFI is the unit number from which the solution is read to restart the program. It is used only for ILT = 2, and its name must be defined on the program card. |
| NASCRIN 124 | LFO | LFO is the unit number on which the output of the current run is saved. It also must be defined on the program card. |
| NASCRIN 127 | ILT | ILT = 1 for the first run of the computer program;<br>ILT = 2 for second and subsequent restarts. |
| NASCRIN 132 | LMAX | LMAX is the maximum number of time-steps the current program will run. The program will stop after LMAX time-steps, print the complete solution, and save it on LFO. The following message is printed at the end of the solution: "PROGRAM TERMINATED ON LMAX." |
| NASCRIN 133 | LW | LW is the number of time-steps after which an intermediate solution is printed. LMAX should be equal to or an integer multiple of LW. |
| NASCRIN 136 | LEXIT | A parameter used to apply boundary conditions at the outflow boundary. LEXIT = 0 for zeroth-order extrapolation; LEXIT = 1 for first-order extrapolation. |
| NASCRIN 139 | CRIT | Local convergence criterion. Convergence is defined as<br>$\text{ERROR} = \frac{\text{New density - old density}}{\text{Old density}} \frac{1}{\text{FDT}} \frac{1}{\text{LSTEP}}$<br>where LSTEP = Number of time-steps after which the convergence is checked. In the present code, LSTEP = 2. That is, after every two time-steps, quantity ERROR is calculated at each grid point, and if it is less than CRIT at all the grid points, the solution is assumed converged based on the local convergence criterion. The program then stops, prints the complete solution, and saves it on LFO. The following message is printed at the end of the solution: "PROGRAM TERMINATED ON LOCAL ERROR CRITERION." FDT is defined on line NASCRIN 219. |

| Line no. | Parameter | Description |
|---|---|---|
| NASCRIN 140 | CRITAVG | Average or global convergence criterion. Average convergence is defined as $$\text{ERRORAVG} = \frac{\Sigma(\text{ERROR}^2)}{\text{Number of grid points}}$$ If ERRORAVG becomes less than CRITAVG, the program again stops based on the average error criterion. It prints the solution and saves it on LFO. The following message is printed at the end of the solution: "PROGRAM TERMINATED ON MEAN SORT ERROR CRITERION." CRITAVG should be one or two orders of magnitude smaller than CRIT. |
| NASCRIN 144<br>NASCRIN 145<br>NASCRIN 146 | FM<br>PF<br>TF | These are the Mach number, pressure, and temperature, respectively, at the inflow boundary. SI units are used. |
| NASCRIN 170 | IMET | IMET = 1 if transformation metric coefficients are to be printed at each grid point; otherwise, IMET = 0. |
| NASCRIN 171 | BETA | BETA controls grid clustering near upper and lower boundaries. The program uses a stretching function that clusters the mesh lines equally near the boundaries. The stretching function can be changed by making changes in subroutine XY. BETA should be greater than 1. The closer it gets to 1, the higher the stretching (i.e., more points are clustered near the boundaries). |
| NASCRIN 217 | CCIMPY | This parameter controls the artificial damping in the implicit stage of the explicit-implicit method (refs. 6 and 7). CCIMPY = 0 turns off the damping, and CCIMPY = 1 turns it on. |
| NASCRIN 218 | FDTL | A parameter that allows the use of local time-step marching to speed up the convergence to steady state. If the local time-step is not used, set FDTL = 1. If the local time-step is to be used, set FDTL greater than 1. For example, if FDTL = 2, the time-step at a grid point cannot exceed twice the global minimum time-step. If at a grid point, the local time-step is less than two times the global minimum, the local time-step is used; otherwise, the time-step at the grid point is set equal to two times the global minimum. |
| NASCRIN 219 | FDT | FDT is a constant multiplier used in the time-step calculations. In the present code, $$\text{Time-step} = \left(\text{FDT}\ \frac{\Delta y}{\lvert v\rvert + a}\right)_{\min}$$ over the entire grid. Here $\Delta y$ is the grid spacing in the y-coordinate direction, $v$ is the velocity in the y-direction, and $a$ is the speed of sound at a grid point. |

| Line no. | Parameter | Description |
|---|---|---|
| NASCRIN 219 (cont.) | FDT | Set FDT < 1 for the fully explicit scheme. For the case of viscous flow calculations, if FDT > 1, the explicit-implicit scheme of MacCormack is turned on automatically. |
| NASCRIN 220<br>NASCRIN 223 | CCP<br>CCT | Coefficients in the artificial damping expression in the explicit stage. The damping expression has two parts, one of which depends on the pressure gradient and the other on the temperature gradient. CCP multiplies the first part, and CCT multiplies the second part. Typical values of CCP and CCT are in the neighborhood of 0.5 and should be kept as small as possible to keep the artificial damping contribution to a minimum. |
| NASCRIN 224 | LKK | This parameter allows the user to try various values of FDT or CCP and CCT. For LKK = 2, only one set of values can be used. But, for LKK = 3, the user can specify two values of FDT or CCP and CCT. If the program detects a negative temperature in the flow field for the first value, it will restart the calculations with the second value. But, if the program proceeds normally with the first value, it will ignore the second value. (See appendix C for examples in using LKK.) |
| NASCRIN 237 | TIMEL | TIMEL is the time required by the free stream to travel 3 times the length of the flow domain. For supersonic flows, it has been observed that the flow reaches steady state if the calculations have advanced in time to TIMEL. If the program does not stop because of local or average convergence criterion and LMAX is sufficiently large, then it will stop if the physical time TIME ≥ TIMEL. Physical time TIME is obtained by summing up the global minimum time-step size from each time-step. The program again prints the solution and saves it also on LFO. The following message is printed at the end of the solution: "PROGRAM TERMINATED ON PHYSICAL TIME CONVERGENCE CRITERION." |

## Subroutine START

The x,y coordinates of the upper and lower boundaries are prescribed in subroutine START between lines START 27 and START 78. For example, if the strut EFGH is removed from figure 1(c), define the x,y coordinates of lines APQRD and BC between the above-mentioned lines in START. However, if there is a strut in the flow field as shown in figure 1(c), define the x,y coordinates of lines APQRD and UEHGV between lines START 27 and START 61, and define the x,y coordinates of lines UEFGV and BC between lines START 64 and START 78. Grid points on lines UE and GV for lower and upper portions should have the same x,y coordinates. All the distances should be defined in meters.

For a quasi-three-dimensional application of the code in scramjet-inlet calculations, geometry should be prescribed in a plane normal to the sidewall sweep.

## Subroutine TURB

Subroutine TURB is called in subroutine VISCOUS at line VISCOUS 222 for turbulent flow calculations (i.e., when NFLOW = 2). Subroutine TURB, in turn, calls subroutines VORT and EDDY to calculate eddy viscosity. It may be necessary to make several calls to subroutine EDDY in order to cover the entire flow field. (See examples in appendix B.)

There are six parameters in the call to subroutine EDDY. These parameters are described below:

| | |
|---|---|
| NNX1,NNX2<br>NY1, NY2 | Grid numbers in the x,y coordinate directions, respectively, for the flow domain being considered in a particular call to subroutine EDDY. |
| NWALL | NWALL = 1 when there is only one solid boundary (i.e., a wall) at NY2; NWALL = 2 when there are solid boundaries at both NY1 and NY2. |
| NSTR | NSTR = 1 for the region in which the lower strut surface forms the upper solid boundary of the flow domain. In this case, set NY2 = MS1; otherwise, NSTR = 0. |

## Subroutine XY

Subroutine XY performs the algebraic coordinate transformation and stretches the grid toward the upper and lower boundaries. Transformation metric coefficients are also calculated in this subroutine.

## Subroutines VISCOUS and INVICID

Subroutine VISCOUS is called at line NASCRIN 241 for viscous (laminar and turbulent) flow calculations, and subroutine INVICID is called at line NASCRIN 240 for inviscid flow calculations. Both subroutines generate a starting solution for the first run of the program (ILT = 1) and read the solution from file LFI for second and subsequent runs (ILT = 2). Subroutines for flux vectors, damping, and boundary conditions are called in these subroutines, and the finite differencing for predictor and corrector steps is also done here. After the program is terminated either on a convergence criterion or on LMAX, subroutine PRINT is called to print the solution, and the solution is saved on file LFO.

In the case of subroutine VISCOUS, if FDT > 1, subroutines IMPY and VECIM2 are called to apply the implicit stage of the explicit-implicit method. This method is applied only in the y-coordinate direction because for a viscous flow calculation, the grid is highly compressed in the y-coordinate direction near the boundaries, which makes the minimum grid spacing in the y-direction much smaller than the minimum grid spacing in the x-direction. Because of this large difference in spacings, even if an implicit stage was incorporated in the x-direction, it would not be used.

Sometimes it is necessary to use a small value of FDT or a large value of damping coefficients CCP and CCT to keep the calculations stable in the initial stages, probably because the starting solution is a poor guess. But as the calculations progress, it is possible to gradually increase the time-step by increasing FDT or to

decrease damping by decreasing CCP and CCT to a desired value over a number of time-steps. In the present program, these changes in FDT or CCP and CCT can be incorporated by inserting a few cards after line VISCOUS 51 and after line INVICID 60. Several examples of implementing these changes are given in appendix D.

### Subroutines DVISCP and DVISCC

Subroutines DVISCP and DVISCC are called only for viscous flow calculations. Both subroutines calculate the viscous dissipation terms in the Navier-Stokes equations. Subroutine DVISCP uses backward differencing, and subroutine DIVSCC uses forward differencing in the calculation of these terms.

### Subroutines VEC1 and IVEC1

Subroutines VEC1 and IVEC1 calculate the flux vectors for viscous and inviscid flow, respectively.

### Subroutines DAMPP and DAMPC

Subroutines DAMPP and DAMPC are used to add artificial damping in the flux vectors. Subroutine DAMPP uses backward differencing, and subroutine DAMPC uses forward differencing in the calculation of damping terms.

### Subroutine VEC2

Subroutine VEC2 is called after the predictor or corrector step. It calculates local flow quantities from the conserved variables.

### Subroutine BOUND

Subroutine BOUND applies the boundary conditions for the viscous flow. The subroutine is set up for supersonic inflow and outflow boundaries. It applies a no-slip boundary condition on the upper boundary from the second grid point to the last grid point. The lower boundary can be either a solid wall, a free boundary, or a symmetry line. The conditions are held fixed at some prescribed values at the inflow boundary, and zeroth- or first-order extrapolation is used at the outflow boundary.

To use the program for supersonic inflow and subsonic outflow, the user needs to prescribe a back pressure at the outflow boundary. This can be done by changing line BOUND 76 for LEXIT = 1 or line BOUND 82 for LEXIT = 0 to the line P(1,N1;M1) = PB where PB is the back pressure in pascals.

It is also possible to modify the code for a subsonic inflow boundary. The user should insert the proper boundary conditions for subsonic inflow after line BOUND 33.

## Subroutine IBOUND

Subroutine IBOUND applies the boundary conditions for the inviscid flow. This subroutine is also set up for supersonic inflow and outflow boundaries, although changes similar to those suggested in subroutine BOUND can be made to modify it to handle a subsonic inflow or subsonic outflow boundary.

Both subroutines BOUND and IBOUND check for negative temperature in the field during each predictor and corrector step. Anytime a negative temperature occurs in the field, the code prints the following message: "NEGATIVE TEMPERATURE IN THE FIELD." It then prints the number of time-steps at which the negative temperature occurred and the values of all the flow variables at each grid point. This allows the user to examine the flow variables in the region where the calculations became unstable. After this, the code returns to statement 1000 at line NASCRIN 221. It either restarts with a new value of time-step or damping parameters, or stops.

## Subroutines VORT and EDDY

Subroutines VORT and EDDY calculate the eddy viscosity from the algebraic turbulence model of reference 3. Subroutine VORT calculates the vorticity at all the grid points. The vorticity is then used in subroutine EDDY to calculate the eddy viscosity. The parameters in subroutine EDDY were discussed earlier in subroutine TURB.

## Subroutines IMPY and VECIM2

Subroutines IMPY and VECIM2 are used only for the viscous flow calculations to apply the implicit stage in the y-direction of the explicit-implicit method. Subroutine IMPY actually applies the implicit stage, and then subroutine VECIM2 calculates the flow variables at each grid point from the conserved variables, calls subroutine BOUND, and saves the flux quantities on the appropriate upper or lower boundary for the implicit stage of the next time-step. The implicit stage is used only when FDT > 1.

## Subroutine SPILL

Subroutine SPILL is called after every LW time-steps at line VISCOUS 244 for the viscous flow and at line INVICID 172 for the inviscid flow. The purpose of this subroutine is to calculate the spillage in a quasi-three-dimensional application of the code. It takes the solution in the plane normal to the sidewall sweep and projects it back to the cowl plane. The constant tangential velocity component is then properly superimposed on the projected solution to provide the solution in the cowl plane. Knowing the component of velocity normal to the plane of the cowl and the density field, flow spillage (i.e., the amount of flow leaving the inlet ahead of the cowl) can be obtained. For viscous flow calculations, an arbitrary boundary-layer profile is prescribed on the constant tangential velocity component before superimposing it on the projected solution. In subroutine SPILL, it is assumed that the tangential velocity changes from zero to the constant inviscid value over 10 to 12 grid points from the solid boundary. For the case of zero sweep, obviously the spillage in the quasi-three-dimensional application of the code will be zero. For this case, the subroutine will skip the spillage calculations.

Subroutine SPILL also calculates and prints the nondimensional pressure on the upper and lower boundaries and on the upper and lower strut surfaces if there is a strut in the flow field. The pressure is nondimensionalized with respect to the pressure at the inflow boundary.

## Subroutine PRINT

Subroutine PRINT is called after every LW time-steps and after the normal termination of the program. It prints the complete flow-field solution at all the grid points. It also prints certain input parameters and the mass flux across a given axial location.

## SAMPLE PROBLEM AND OUTPUT QUANTITIES

To illustrate the use of the code, the listing given in appendix A is set up for a quasi-three-dimensional application of the code for the inlet geometry shown in figure 1. The geometry input in subroutine START is in a plane normal to the sweep line (fig. 1(c)). Only the upper half of the problem is solved. The problem is set up to calculate turbulent flow at Mach 5. The following values of input parameters have been used:

N1 = 55

M1 = 61

SWEEP = 30

NSTRUT = 1

NS1 = 16

NS2 = 46

MS = 29

LSYM = 1

NCWL = 1

NCWL1 = 24

NCWL2 = 44

NFLOW = 2

LFI = 9
LFO = 9

(The program card equates UNIT9 to CASE20 at line NASCRIN 2. CASE20 is the name of the file on which the solution is saved.)

ILT = 1

LMAX = 25000

LW = LMAX

LEXIT = 1

CRIT = 0.0001

CRITAVG = 0.1 * CRIT

FM = 5. (FM is the Mach number at the face of the inlet in the plane normal to the sweep line.)

PF = 10000.

TF = 300.

IMET = 1

BETA = 1.005

FDTL = 1.

FDT = 1

CCP = 0.5

CCT = CCP

TIMEL = 3. * x(1,N1)/UF (UF is the velocity at the face of the inlet corresponding to Mach 5.)

To calculate eddy viscosity, five calls to subroutine EDDY are made in subroutine TURB as shown in the listing in appendix A. As mentioned earlier, for the quasi-three-dimensional application of the code, the flow is calculated in a plane normal to the sidewall sweep. The code prints the detailed solution in this plane. It then calculates the spillage from the inlet by projecting the solution to a plane parallel to the cowl plane and properly superimposing on it the component of velocity parallel to sweep line. The sample outputs are presented in appendix E, and a brief description of output quantities is given below.

Since parameter IMET is set equal to 1, the code prints the metric coefficients at all the grid points. In this portion of the output,

N grid number in x-direction

M grid number in y-direction

X distance in x-direction, m

Y distance in y-direction, m

XXI $= \frac{\partial x}{\partial \xi}$

YXI $= \frac{\partial y}{\partial \xi}$

XETA $= \frac{\partial x}{\partial \eta}$

YETA $= \frac{\partial y}{\partial \eta}$

AJ $= \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}$

where $\xi, \eta$ are the coordinate directions in the transformed plane.

The code then prints the maximum error and mean square root error after every 50 time-steps till it stops either on LMAX or on one of the convergence criteria. It then prints the total number of time-steps (or iterations), the physical time advanced in seconds, and the time-limit in seconds. (Time-limit is set by the parameter TIMEL.) It also prints the values of CRIT (local error), CRITAVG (average error), and LEXIT (order of extrapolation at outflow). The complete solution is then printed at each body station in the x-direction. It includes

| | |
|---|---|
| X,Y | coordinates of the grid point, m |
| U,V | axial and normal velocities, m/sec |
| P | pressure, Pa |
| T | temperature, K |
| RO | density, $kg/m^3$ |
| SH | static enthalpy, $m^2/sec^2$ |
| VIST | turbulent viscosity, kg/m-sec |
| ERRO | local error at a grid point |

For NSTRUT = 1, flow variables at M = MS between N = NS1 to NS2 correspond to the top surface of the strut. Flow variables at the lower surface are printed separately. At the lower surface of the strut, the code prints total enthalpy HS and laminar viscosity VISS in place of VIST and ERRO.

With each call to subroutine PRINT, after printing the detailed solution, the code prints the mass flow rate at each axial station. This allows the user to check for mass conservation inside the inlet.

After the code stops on LMAX or any of the convergence criteria, it calls subroutine SPILL, which calculates flow spillage. If the sweep angle is zero, spillage calculations are not done. A sample output for spillage calculations is shown in appendix E. In this output, XP is the projected axial distance in the cowl place (i.e., XP = X/cos $\Lambda$). Local mass spilled (in kilograms per meter-second) is the integrated value of the product of density and downward velocity component at a given XP location. Total mass spilled (in kilograms per second) is obtained by integrating the local values up to a given XP location. If, for example, the cowl closure starts at XP = 0.132791 m, corresponding to body station 40, the total mass spilled ahead of the cowl is equal to 0.01193 kg/sec. The user can then calculate the spillage in

terms of the percentage of the incoming mass flow rate at the face of the three-dimensional inlet.

Subroutine SPILL also calculates and prints the surface pressure distribution on the top and bottom (or centerline in the case of a symmetric problem) boundaries and on the upper and lower strut surfaces for NSTRUT = 1. These pressures are nondimensionalized with respect to the pressure at the face of the inlet.

Finally, the code prints a message that tells on what basis the calculations were terminated.

From the solution of the sample problem presented here, pressure contours and the velocity vector field were plotted. These plots are shown in figures 2 and 3, respectively. Although only the upper half of the inlet was solved, the results were projected to the lower half of the inlet for plotting purposes.

## COMPUTATIONAL RATE AND TIME REQUIREMENTS

The code is operational on the CDC CYBER 203 computer system and is highly vectorized to take full advantage of the vector-processing capability of the system. With the explicit method, it has a computational rate of $2 \times 10^{-5}$ second per grid point per time-step for viscous flow and $1.3 \times 10^{-5}$ second per grid point per time-step for inviscid flow when the 64-bit word arithmetic is used. The code can easily be modified to use the 32-bit word arithmetic, in which case, the computational rate is improved by a factor of about 2.5. For example, the code can perform approximately 50 time-steps per second for viscous flow and 75 time-steps per second for inviscid flow on a grid of 51 × 51 when the explicit method is used with the 32-bit word arithmetic. An inviscid problem using a 51 × 51 grid and requiring 1000 time-steps for convergence can be solved in about 15 seconds.

Finally, in the case of viscous flow calculations with very fine mesh spacing near the solid boundaries, a significant improvement in computational rate can be obtained by using the explicit-implicit method.

## CONCLUDING REMARKS

A vectorized computer program NASCRIN has been developed that solves the two-dimensional Euler or Navier-Stokes equations in conservation form. It primarily uses MacCormack's explicit method, but for viscous flow calculations, MacCormack's explicit-implicit method can also be used at the user's discretion. An algebraic, two-layer eddy-viscosity model is used for the turbulent flow calculations. The code can analyze problems with or without embedded bodies in the flow field. It is highly user oriented and is structured in such a way that for most supersonic flow problems, the user has to make changes only in the main program and first two subroutines. Although the code is primarily written for supersonic internal flow, it can be used with some modifications for a variety of other flow problems.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
December 1, 1983

APPENDIX A

PROGRAM LISTING

```
      PROGRAM NASCRIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1UNIT9=CASE20)
C*********************************************************************
C NASCRIN STANDS FOR NUMERICAL ANALYSIS OF SUPERSONIC COMBUSTION
C RAMJET INLETS.
C
C THIS PROGRAM IS BASED ON AIAA PAPER NO. 81-185, NASA TP 1940,
C AND NASA TP 1934, ALL BY AJAY KUMAR OF NASA LANGLEY RESEARCH CENTER.
C
C THE PROGRAM SOLVES THE TWO-DIMENSIONAL NAVIER-STOKES EQUATIONS FOR
C LAMINAR AND TURBULENT FLOW AND TWO-DIMENSIONAL EULER EQUATIONS FOR
C INVISCID FLOW IN FULLY CONSERVATIVE FORM.
C
C IT USES MACCORMACK'S UNSPLIT, EXPLICIT FINITE-DIFFERENCE METHOD.
C
C FOR VISCOUS FLOW, MACCORMACK'S EXPLICIT-IMPLICIT SCHEME CAN ALSO BE
C USED AT USER'S OPTION.
C
C TURBULENCE IS MODELED BY BALDWIN-LOMAX MODEL.
C
C AN ALGEBRAIC COORDINATE TRANSFORMATION IS USED TO GENERATE A SET OF
C BOUNDARY FITTED CURVILINEAR COORDINATES.
C
C OPERATIONAL ON CDC-CYBER-203 VECTOR PROCESSING COMPUTER.
C
C*********************************************************************
      BIT 81
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),
     2 PS(55),TS(55),SHS(55),HS(55)
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),
     1 XETA(61,55),YETA(61,55),AJ(61,55)
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),
     1 YSETA(55),AJS(55)
      COMMON/F9/INT(55),TE1(55),TE2(55),TE3(55),TE4(55)
      COMMON/F10/DT(61,55),DTM
      COMMON/F11/ERRO(61,55),B1(61,55)
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF
      COMMON/F14/DAU(61,55,4),BDAU(55,4),BDAUS(55,4),TI(55,5),
     1CCIMPY,GM1,GM2,GM3,GM4,GM5
      N1=55
      M1=61
C     DEFINE SIDEWALL SWEEP.  IF THE FLOW IS BEING CALCULATED IN A PLANE
C     NORMAL TO SIDEWALL SWEEP, DEFINE SIDEWALL SWEEP IN DEGREES.
C     OTHERWISE EQUATE IT TO ZERO.
      SWEEP=30.
C     NSTRUT=0 FOR NO STRUT CASE. NSTRUT=1 FOR A STRUT IN THE FLOW.
      NSTRUT=1
C     STRUT IS LOCATED BETWEEN NS1 AND NS2 MESH POINTS IN XI DIRECTION
C     AND IS COINCIDENT WITH MESH MS IN ETA DIRECTION.
      NS1=16
      NS2=46
      MS=29
C     LSYM IS A PARAMETER FOR THE LOWER BOUNDARY.
C     IF THE LOWER BOUNDARY IS A SYMMETRY LINE AND
C     WE ARE SOLVING ONLY UPPER HALF OF THE FLOW DOMAIN,  LSYM=1
```

```
C     IF THE LOWER BOUNDARY IS NOT A SYMMETRY LINE,  LSYM = 0
      LSYM=1
C     NCWL1 AND NCWL2 ARE THE MESH POINTS BETWEEN WHICH COWL OR THE
C     CENTER STRUT IS LOCATED.  IF THERE IS NO COWL OR STRUT ON THE
C     CENTER LINE, SET NCWL = 0 OTHERWISE SET = 1 .
      NCWL=1
      NCWL1=24
      NCWL2=44
C
      N11=N1-1
      N12=N1-2
      N13=N1-3
      M11=M1-1
      M12=M1-2
      M13=M1-3
      NXM=N1*M1
      NXM1=NXM-1
      NXM2=NXM-2
      NXM3=NXM-2*M1
      NXM4=NXM3-2
      NXM5=NXM-4*M1-2
      NXM6=NXM-2*M1-4
      NXM7=NXM-4*M1-4
      NS1M1=NS1-1
      NS1M2=NS1-2
      NS1P1=NS1+1
      NS2M1=NS2-1
      NS2P1=NS2+1
      NNS=N1-NS1M1
      NS=NS2-NS1+1
      NSM1=NS-1
      MS1=MS-1
      MS2=MS-2
      MS3=MS-3
      MS4=MS-4
      MSP1=MS+1
      MSP2=MS+2
      MSP3=MS+3
      NCWLM=NCWL1-1
      NCWLP=NCWL2+1
C     NFLOW IS A PARAMETER FOR THE TYPE OF FLOW. NFLOW= 0, 1 OR 2
C     DEPLENDING ON WHETHER THE FLOW IS INVISCID, LAMINAR OR TURBULENT.
C*********************************************************************
C IF NFLOW=2, IT IS NECESSARY TO MODIFY THE CALLS TO SUBROUTINE EDDY
C WHICH ARE FOUND IN SUBROUTINE TURB.
C THE ENTIRE FLOW FIELD MAY HAVE TO BE DIVIDED INTO SEVERAL SUB-
C DOMAINS AND MORE THAN ONE CALL TO SUBROUTINE EDDY MAY BE REQUIRED.
C SEE EXAMPLES IN USER'S MANUAL FOR FURTHER CLARIFICATION.
C*********************************************************************
      NFLOW=2
C     LFI IS THE PERMANENT FILE NUMBER FROM WHICH THE SOLUTION IS
C     READ TO RESTART THE PROGRAM.  LFI IS REQUIRED ONLY WHEN ILT=2.
C     LFO IS THE PERMANENT FILE NUMBER ON WHICH THE OUTPUT OF THE
C     CURRENT COMPUTER RUN IS STORED.  BOTH LFI AND LFO SHOULD BE
C     DEFINED ON THE PROGRAM CARD.
      LFI=9
      LFO=9
C     SET ILT=1 FOR THE FIRST RUN OF THE COMPUTER PROGRAM.
C     TO RESTART THE PROGRAM, SET ILT=2
      ILT=1
C     LMAX IS THE MAXIMUM NUMBER OF TIME-STEPS.  THE PROGRAM WILL
C     STOP AFTER LMAX TIME-STEPS AND WILL STORE THE SOLUTION ON LFO.
C     AN INTERMEDIATE SOLUTION IS PRINTED AFTER EVERY LW TIME-STEPS.
C     LMAX SHOULD BE EQUAL TO OR AN INTEGER MULTIPLE OF LW.
      LMAX=25000
      LW=LMAX
C     LEXIT=0 FOR ZERO ORDER EXTRAPOLATION AND LEXIT=1 FOR FIRST ORDER
C     EXTRAPOLATION AT THE OUTFLOW BOUNDARY.
      LEXIT=1
```

```
C     ERROR CRITERION                                                   NASCRIN  137
C     CRITAVG IS THE AVERAGE ERROR OVER THE ENTIRE FLOW FIELD           NASCRIN  138
      CRIT=.0001                                                        NASCRIN  139
      CRITAVG=0.1*CRIT                                                  NASCRIN  140
C                                                                       NASCRIN  141
C     CONDITIONS AT THE FACE OF THE INLET.  USE SI UNITS.               NASCRIN  142
C     (P IN N/M2 AND T IN K)                                            NASCRIN  143
      FM=5.                                                             NASCRIN  144
      PF=10000.                                                         NASCRIN  145
      TF=300.                                                           NASCRIN  146
      GAMA=1.4                                                          NASCRIN  147
      RGAS=287.                                                         NASCRIN  148
      CP=GAMA*RGAS/(GAMA-1.)                                            NASCRIN  149
      GM1=GAMA-1.                                                       NASCRIN  150
      GM2=GM1/GAMA                                                      NASCRIN  151
      GM3=CP/GAMA                                                       NASCRIN  152
      PR=.72                                                            NASCRIN  153
      PRT=.9                                                            NASCRIN  154
      GM4=2.*GAMA/PR                                                    NASCRIN  155
      GM5=GAMA*RGAS                                                     NASCRIN  156
C     CVIS IS THE COEFFICIENT USED IN SUTHERLAND'S VISCOSITY FORMULA    NASCRIN  157
C     FOR AIR.                                                          NASCRIN  158
      CVIS=1.458E-6                                                     NASCRIN  159
      UF=FM*(GAMA*RGAS*TF)**.5                                          NASCRIN  160
      ROF=PF/RGAS/TF                                                    NASCRIN  161
      SHF=GAMA*RGAS*TF/(GAMA-1.)                                        NASCRIN  162
      HF=SHF+UF*UF/2.                                                   NASCRIN  163
C     ****************                                                  NASCRIN  164
C     GEOMETRY INPUT AND METRIC COEFFICIENTS CALCULATIONS.              NASCRIN  165
C                                                                       NASCRIN  166
C     DEFINE IMET=0 IF METRIC COEFFICIENTS NEED NOT BE PRINTED.         NASCRIN  167
C     IMET=1 IF METRIC COEFFICIENTS ARE TO BE PRINTED.                  NASCRIN  168
C     BETA=STRETCHING FACTOR.  IT SHOULD BE GREATER THAN 1.             NASCRIN  169
      IMET=1                                                            NASCRIN  170
      BETA=1.005                                                        NASCRIN  171
      MM=0                                                              NASCRIN  172
      IF(NSTRUT.EQ.0)MS=M1                                              NASCRIN  173
      CALL START(MM,SWEEP)                                              NASCRIN  174
      CALL XY(1,MS,BETA)                                                NASCRIN  175
      IF(NSTRUT.EQ.0)GO TO 500                                          NASCRIN  176
      MM=1                                                              NASCRIN  177
      DO 10 NN=NS1,NS2                                                  NASCRIN  178
      N=NN-NS1+1                                                        NASCRIN  179
      XS(N)=X(MS,NN)                                                    NASCRIN  180
      YS(N)=Y(MS,NN)                                                    NASCRIN  181
      XSETA(N)=XETA(MS,NN)                                              NASCRIN  182
      YSETA(N)=YETA(MS,NN)                                              NASCRIN  183
      XSXI(N)=XXI(MS,NN)                                                NASCRIN  184
 10   YSXI(N)=YXI(MS,NN)                                                NASCRIN  185
      CALL START(MM,SWEEP)                                              NASCRIN  186
      CALL XY(MS,M1,BETA)                                               NASCRIN  187
      DO 20 N=1,NS1M1                                                   NASCRIN  188
      YETA(MS,N)=(Y(MS+1,N)-Y(MS-1,N))/2.                               NASCRIN  189
 20   XETA(MS,N)=(X(MS+1,N)-X(MS-1,N))/2.                               NASCRIN  190
      DO 30 N=NS2P1,N1                                                  NASCRIN  191
      YETA(MS,N)=(Y(MS+1,N)-Y(MS-1,N))/2.                               NASCRIN  192
 30   XETA(MS,N)=(X(MS+1,N)-X(MS-1,N))/2.                               NASCRIN  193
      AJS(1;NS)=XSXI(1;NS)*YSETA(1;NS)-XSETA(1;NS)*YSXI(1;NS)           NASCRIN  194
 500  CONTINUE                                                          NASCRIN  195
      AJ(1,1;NXM)=XXI(1,1;NXM)*YETA(1,1;NXM)-XETA(1,1;NXM)*YXI(1,1;NXM) NASCRIN  196
      IF(IMET.EQ.0)GO TO 45                                             NASCRIN  197
      DO 40 N=1,N1                                                      NASCRIN  198
      WRITE(6,1100)                                                     NASCRIN  199
      DO 40 MM=1,M1                                                     NASCRIN  200
      M=M1-MM+1                                                         NASCRIN  201
 40   WRITE(6,1200)N,M,X(M,N),Y(M,N),XXI(M,N),XETA(M,N),YXI(M,N),YETA(M,NASCRIN  202
     1N),AJ(M,N)                                                        NASCRIN  203
      IF(NSTRUT.EQ.0)GO TO 45                                           NASCRIN  204
      M=MS                                                              NASCRIN  205
```

```
      WRITE(6,1100)                                                     NASCRIN    206
      WRITE(6,1200)(N,M,XS(N),YS(N),XSXI(N),XSETA(N),YSXI(N),YSETA(N),AJNASCRIN    207
     1S(N),N=1,NS)                                                      NASCRIN    208
 1100  FORMAT(/,3X,'N',3X,'M',10X,'X',10X,'Y',8X,'XXI',7X,'XETA',8X,    NASCRIN    209
     1'YXI',7X,'YETA',9X,'AJ',/)                                        NASCRIN    210
 1200  FORMAT(2X,2I3,7F11.8)                                            NASCRIN    211
 45   CONTINUE                                                          NASCRIN    212
C     ********************                                              NASCRIN    213
C     INTERVAL VECTOR.                                                  NASCRIN    214
      INT(1;N1)=Q8VINTL(1,M1;INT(1;N1))                                 NASCRIN    215
      LKK=0                                                             NASCRIN    216
      CCIMPY=1.                                                         NASCRIN    217
      FDTL=1.                                                           NASCRIN    218
      FDT=1.                                                            NASCRIN    219
      CCP=.5                                                            NASCRIN    220
 1000 CONTINUE                                                          NASCRIN    221
      LKK=LKK+1                                                         NASCRIN    222
      CCT=CCP                                                           NASCRIN    223
      IF(LKK.EQ.2)STOP                                                  NASCRIN    224
      LK=1                                                              NASCRIN    225
      DO 50 I=1,4                                                       NASCRIN    226
      BDAJS(1,I;N1)=0.                                                  NASCRIN    227
      BDAU(1,I;N1)=0.                                                   NASCRIN    228
      DAU(1,1,I;NXM)=0.                                                 NASCRIN    229
 50   CONTINUE                                                          NASCRIN    230
      VIST(1,1;NXM)=0.                                                  NASCRIN    231
      VISL(1,1;NXM)=0.                                                  NASCRIN    232
      VIS(1,1;NXM)=0.                                                   NASCRIN    233
      VISS(1;N1)=0.                                                     NASCRIN    234
C     TIMEL IS THE TIME REQUIRED BY THE FREE STREAM TO TRAVEL 3 TIMES   NASCRIN    235
C     OVER THE LENGTH OF THE FLOW DOMAIN.                               NASCRIN    236
      TIMEL=3.*X(1,N1)/UF                                               NASCRIN    237
      TIME=0.                                                           NASCRIN    238
      L=0                                                               NASCRIN    239
      IF(NFLOW.EQ.0) CALL INVICID(SWEEP)                                NASCRIN    240
      IF(NFLOW.NE.0) CALL VISCOUS(SWEEP)                                NASCRIN    241
      IF(LK.EQ.2) GO TO 1000                                            NASCRIN    242
      STOP                                                              NASCRIN    243
      END                                                               NASCRIN    244
      SUBROUTINE START(MM,SWEEP)                                        START        1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 START        2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              START        3
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   START        4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      START        5
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                START        6
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                START        7
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),              START        8
     1 YSETA(55),AJS(55)                                                START        9
C     FOR MM=0, THIS SUBROUTINE DESCRIBES THE X,Y COORDINATES OF THE    START       10
C     LOWER BOUNDARY AND THE MESH LINES PASSING THROUGH THE LOWER SURFACSTART       11
C     OF THE STRUT.  FOR MM=1, IT DESCRIBES THE COORDINATES OF THE      START       12
C     MESH LINE PASSING THROUGH THE UPPER SURFACE OF THE STRUT AND THE  START       13
C     UPPER BOUNDARY.  IF THERE IS  NO STRUT IN THE FLOWFIELD, THEN IT  START       14
C     DESCRIBES THE COORDINATES OF UPPER AND LOWER BOUNDARIES.          START       15
      PI=4.*ATAN(1.)                                                    START       16
      SW=SWEEP*PI/180.                                                  START       17
C    ****************                                                   START       18
C                                                                       START       19
C DESCRIBE X(1,N), X(M1,N), Y(1,N),Y(M1,N), FOR N=1 TO N1               START       20
C AND, IF MM=1, DESCRIBE                                                START       21
C          X(MS,N), Y(MS,N)                                             START       22
C                                                                       START       23
C  FOR QUASI-THREE-DIMENSTIONAL CALCULATIONS, PROVIDE THE GEOMETRY IN   START       24
C  A PLANE NORMAL TO THE SIDEWALL SWEEP.                                START       25
C IF IN INCHES, MULTIPLY BY .0254 TO GET METRIC UNITS                   START       26
C     ****************                                                  START       27
C     THE GEOMETRY GIVEN BELOW IS IN A PLANE NORMAL TO SWEEP LINE.      START       28
C     SWEEP ANGLE IS 33 DEGREES.                                        START       29
```

```
      TANC1=TAN(6.*PI/180.)                                             START
      TANC2=TAN(7.5*PI/180.)                                            START
      TANC3=TAN(6.271178*PI/180.)                                       START
      X(1,1)=0.                                                         START
      DO 1 N=2,6                                                        START
    1 X(1,N)=X(1,N-1)+.005                                              START
      X(1,7)=X(1,6)+.0045                                               START
      X(1,8)=X(1,7)+.004                                                START
      X(1,9)=X(1,8)+.0035                                               START
      X(1,10)=X(1,9)+.003                                               START
      DO 2 N=11,N1                                                      START
    2 X(1,N)=X(1,N-1)+.0025                                             START
      DX=.0025                                                          START
C     ******************                                                START
      IF(MM.EQ.1)GO TO 300                                              START
C     ******************                                                START
      DO 3 N=1,N1                                                       START
      X(MS,N)=X(1,N)                                                    START
      Y(1,N)=0.                                                         START
    3 Y(MS,N)=.018                                                      START
      DO 4 N=25,34                                                      START
    4 Y(1,N)=Y(1,N-1)+DX*TANC2                                          START
      DO 5 N=35,43                                                      START
    5 Y(1,N)=Y(1,N-1)-DX*TANC2                                          START
      DO 6 N=17,34                                                      START
    6 Y(MS,N)=Y(MS,N-1)-DX*TANC2                                        START
      DO 7 N=35,46                                                      START
    7 Y(MS,N)=Y(MS,N-1)+DX*TANC3                                        START
      DY=(Y(MS,46)-.013)/.0225                                          START
      DO 8 N=47,N1                                                      START
    8 Y(MS,N)=Y(MS,N-1)-DX*DY                                           START
C     ******************                                                START
      IF(MM.EQ.0)RETURN                                                 START
C     ******************                                                START
  300 CONTINUE                                                          START
      Y(M1,1)=.04                                                       START
      Y(M1,2)=.04                                                       START
      DO 10 N=1,N1                                                      START
      X(M1,N)=X(1,N)                                                    START
      X(MS,N)=X(1,N)                                                    START
   10 Y(MS,N)=.018                                                      START
      DO 11 N=3,N1                                                      START
   11 Y(M1,N)=Y(M1,N-1)-(X(M1,N)-X(M1,N-1))*TANC1                       START
      DO 12 N=37,46                                                     START
   12 Y(MS,N)=Y(MS,N-1)-DX*TANC1                                        START
      DY=(Y(MS,46)-.013)/.0225                                          START
      DO 13 N=47,N1                                                     START
   13 Y(MS,N)=Y(MS,N-1)-DX*DY                                           START
C                                                                       START
      RETURN                                                            START
      END                                                               START
      SUBROUTINE TURB                                                   TURB
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 TURB
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              TURB
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   TURB
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      TURB
      CALL VORT                                                         TURB
C                                                                       TURB
C***********************************************************************TURB
C IF NFLOW=2 WAS SPECIFIED, EDDY VISCOSITY IS CALCULATED USING CALLS TO TURB
C SUBROUTINE EDDY.                                                      TURB
C                                                                       TURB
C SUBROUTINE EDDY HAS SIX PARAMETERS:  NNX1,NNX2,NY1,NY2,NWALL,&NSTR.   TURB
C   NNX1,NNX2 AND NY1,NY2 ARE THE GRID NUMBERS IN THE X AND Y DIRECTIONSTURB
C            RESPECTIVELY FOR THE FLOW DOMAIN BEING CONSIDERED.         TURB
C   NWALL=1 WHEN THERE IS ONLY ONE WALL (I.E., A WALL AT NY2)           TURB
C   NWALL=2 WHEN THERE ARE TWO WALLS (I.E., WALLS AT BOTH NY1 AND NY2). TURB
C   NSTR=1 FOR THE REGION IN WHICH THE LOWER SURFACE OF THE STRUT FORMS TURB
```

```
C            THE UPPER WALL OF THE FLOW DOMAIN (NY2=MS1).                    TURB        18
C   NSTR=0 OTHERWISE                                                         TURB        19
C                                                                            TURB        20
C USE AS MANY CALLS TO SUBROUTINE EDDY AS NECESSARY TO DEFINE ENTIRE REGTURB 21
C SEE EXAMPLES IN THE USER'S MANUAL FOR FURTHER CLARIFICATION.               TURB        22
C*********************************************************************TURB 23
C                                                                            TURB        24
      CALL EDDY(2,15,1,M1,1,0)                                               TURB        25
      CALL EDDY(16,46,MS,M1,2,0)                                             TURB        26
      CALL EDDY(16,23,1,MS1,1,1)                                             TURB        27
      CALL EDDY(24,46,1,MS1,2,1)                                             TURB        28
      CALL EDDY(47,N1,1,M1,1,0)                                              TURB        29
      RETURN                                                                 TURB        30
      END                                                                    TURB        31
      SUBROUTINE XY(ETAMN,ETAMX,BETA)                                        XY           1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 XY         2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                                   XY           3
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3        XY           4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                           XY           5
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                     XY           6
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                     XY           7
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),                   XY           8
     1 YSETA(55),AJS(55)                                                     XY           9
      DIMENSION ETAB(61),ETAD(61)                                            XY          10
      INTEGER ETAMX,ETAMN                                                    XY          11
      B=ALOG((BETA+1.)/(BETA-1.))                                            XY          12
      NETA=ETAMX-ETAMN                                                       XY          13
      DO 10 M=ETAMN,ETAMX                                                    XY          14
      A=(2.*(M-ETAMN)/NETA-1.)*B                                             XY          15
      EXPA=EXP(A)                                                            XY          16
      ETAY=((BETA+1.)*EXPA-BETA+1.)/(1.+EXPA)                                XY          17
      ETAB(M)=ETAY*NETA/2.+ETAMN                                             XY          18
 10   ETAD(M)=2.*B*BETA*EXPA/(1.+EXPA)**2                                    XY          19
      DO 15 N=1,N1                                                           XY          20
      DO 15 M=ETAMN,ETAMX                                                    XY          21
      X(M,N)=(X(ETAMX,N)*(ETAB(M)-ETAMN)+X(ETAMN,N)*(ETAMX-ETAB(M)))/        XY          22
     1NETA                                                                   XY          23
      Y(M,N)=(Y(ETAMX,N)*(ETAB(M)-ETAMN)+Y(ETAMN,N)*(ETAMX-ETAB(M)))/        XY          24
     1NETA                                                                   XY          25
      XETA(M,N)=(X(ETAMX,N)-X(ETAMN,N))*ETAD(M)/NETA                         XY          26
 15   YETA(M,N)=(Y(ETAMX,N)-Y(ETAMN,N))*ETAD(M)/NETA                         XY          27
      DO 20 N=2,N11                                                          XY          28
      DO 20 M=ETAMN,ETAMX                                                    XY          29
      XXI(M,N)=(X(M,N+1)-X(M,N-1))/2.                                        XY          30
 20   YXI(M,N)=(Y(M,N+1)-Y(M,N-1))/2.                                        XY          31
      DO 30 M=ETAMN,ETAMX                                                    XY          32
      XXI(M,1)=X(M,2)-X(M,1)                                                 XY          33
      YXI(M,1)=Y(M,2)-Y(M,1)                                                 XY          34
      XXI(M,N1)=X(M,N1)-X(M,N11)                                             XY          35
 30   YXI(M,N1)=Y(M,N1)-Y(M,N11)                                             XY          36
      RETURN                                                                 XY          37
      END                                                                    XY          38
      SUBROUTINE VISCOUS(SWEEP)                                              VISCOUS      1
      BIT BI                                                                 VISCOUS      2
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 VISCOUS    3
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                                   VISCOUS      4
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3        VISCOUS      5
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                           VISCOUS      6
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),               VISCOUS      7
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                             VISCOUS      8
     2 PS(55),TS(55),SHS(55),HS(55)                                          VISCOUS      9
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)                  VISCOUS     10
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),              VISCOUS     11
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)              VISCOUS     12
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)             VISCOUS     13
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                     VISCOUS     14
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                     VISCOUS     15
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                     VISCOUS     16
```

```
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),                VISCOUS    17
     1 YSETA(55),AJS(55)                                                  VISCOUS    18
      COMMON/F9/INT(55),TE1(55),TE2(55),TE3(55),TE4(55)                   VISCOUS    19
      COMMON/F10/DT(61,55),DTM                                            VISCOUS    20
      COMMON/F11/ERRO(61,55),B1(61,55)                                    VISCOUS    21
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,      VISCOUS    22
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                    VISCOUS    23
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF          VISCOUS    24
      COMMON/F14/DAU(61,55,4),BDAU(55,4),BDAUS(55,4),TI(55,5),            VISCOUS    25
     1CCIMPY,GM1,GM2,GM3,GM4,GM5                                          VISCOUS    26
C                                                                         VISCOUS    27
C     INITIAL GUESS                                                       VISCOUS    28
C                                                                         VISCOUS    29
      U(1,1;NXM)=UF                                                       VISCOUS    30
      V(1,1;NXM)=0.                                                       VISCOUS    31
      T(1,1;NXM)=TF                                                       VISCOUS    32
      SH(1,1;NXM)=SHF                                                     VISCOUS    33
       H(1,1;NXM)=HF                                                      VISCOUS    34
      P(1,1;NXM)=PF                                                       VISCOUS    35
      RO(1,1;NXM)=P(1,1;NXM)/RGAS/T(1,1;NXM)                              VISCOUS    36
      IF(ILT.EQ.1)GO TO 2                                                 VISCOUS    37
      READ(LFI,561) L,TIME                                                VISCOUS    38
      DO 3 N=1,N1                                                         VISCOUS    39
       READ(LFI,560)(X(M,N),Y(M,N),U(M,N),V(M,N),P(M,N),T(M,N),RO(M,N),   VISCOUS    40
     1SH(M,N),H(M,N),VIST(M,N),M=1,M1)                                    VISCOUS    41
 3    CONTINUE                                                            VISCOUS    42
      LMAX=LMAX+L                                                         VISCOUS    43
      IF(NSTRUT.EQ.0)GO TO 2                                              VISCOUS    44
       READ(LFI,560)(XS(N),YS(N),US(N),VS(N),PS(N),TS(N),ROS(N),SHS(N),   VISCOUS    45
     1HS(N),VISS(N),N=1,NS)                                               VISCOUS    46
 2    CONTINUE                                                            VISCOUS    47
      L=L+1                                                               VISCOUS    48
      CALL BOUND                                                          VISCOUS    49
 1    CONTINUE                                                            VISCOUS    50
      LSTEP=1                                                             VISCOUS    51
      ERRO(1,1;NXM)=RO(1,1;NXM)                                           VISCOUS    52
      QX(2,2;NXM4)=GAMA*P(2,2;NXM4)/RO(2,2;NXM4)                          VISCOUS    53
      QX(2,2;NXM4)=VSQRT(QX(2,2;NXM4);QX(2,2;NXM4))                       VISCOUS    54
      DT(2,2;NXM4)=FDT*YETA(2,2;NXM4)            /(VABS(V(2,2;NXM4);      VISCOUS    55
     1QY(2,2;NXM4))+QX(2,2;NXM4))                                         VISCOUS    56
      DTM=Q8SMIN(DT(2,2;NXM4))                                            VISCOUS    57
      DTML=FDTL*DTM                                                       VISCOUS    58
      B1(2,2;NXM4)=DT(2,2;NXM4).GT.DTML                                   VISCOUS    59
      DT(2,2;NXM4)=Q8VCTRL(DTML,B1(2,2;NXM4);DT(2,2;NXM4))                VISCOUS    60
      TIME=TIME+DTM                                                       VISCOUS    61
      CALL DVISCP                                                         VISCOUS    62
      CALL VEC1                                                           VISCOUS    63
      CALL DAMPP                                                          VISCOUS    64
C                                                                         VISCOUS    65
C     PREDICTOR STEP WITH  FORWARD DIFFERENCING.                          VISCOUS    66
C                                                                         VISCOUS    67
      IF(FDT.GT.1.)GO TO 10                                               VISCOUS    68
      DO 100 I=1,4                                                        VISCOUS    69
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)-DT(2,2;NXM4)*((AM(2,3,I;NXM4)-AM(2,VISCOUS    70
     12,I;NXM4))+(AN(3,2,I;NXM4)-AN(2,2,I;NXM4)))                         VISCOUS    71
      IF(NSTRUT.EQ.0)GO TO 100                                            VISCOUS    72
      DO 105 N=1,NS                                                       VISCOUS    73
      NN=N+NS1-1                                                          VISCOUS    74
      AU1(MS1,NN,I)=AU(MS1,NN,I)-DT(MS1,NN)*(AM(MS1,NN+1,I)-AM(MS1,NN,I)VISCOUS    75
     1+ANS(N,I)-AN(MS1,NN,I))                                             VISCOUS    76
 105  CONTINUE                                                            VISCOUS    77
 100  CONTINUE                                                            VISCOUS    78
      CALL VEC2                                                           VISCOUS    79
      CALL BOUND                                                          VISCOUS    80
      GO TO 20                                                            VISCOUS    81
  10  CONTINUE                                                            VISCOUS    82
      DO 15 I=1,4                                                         VISCOUS    83
      DAU(2,2,I;NXM4)=-DT(2,2;NXM4)*(AM(2,3,I;NXM4)-AM(2,2,I;NXM4)+       VISCOUS    84
```

```
     1AN(3,2,I;NXM4)-AN(2,2,I;NXM4))/AJ(2,2;NXM4)                       VISCOUS    85
      IF(NSTRUT.EQ.0)GO TO 15                                           VISCOUS    86
      DO 16 N=1,NS                                                      VISCOUS    87
      NN=N+NS1-1                                                        VISCOUS    88
      DAU(MS1,NN,I)=-DT(MS1,NN)*(AM(MS1,NN+1,I)-AM(MS1,NN,I)+ANS(N,I)-  VISCOUS    89
     1AN(MS1,NN,I))/AJ(MS1,NN)                                          VISCOUS    90
   16 CONTINUE                                                          VISCOUS    91
   15 CONTINUE                                                          VISCOUS    92
      IADD=1                                                            VISCOUS    93
      CALL IMPY(IADD)                                                   VISCOUS    94
      DO 17 I=1,4                                                       VISCOUS    95
   17 AU1(2,2,I;NXM4)=AU(2,2,1;NXM4)+DAU(2,2,I;NXM4)*AJ(2,2;NXM4)       VISCOUS    96
C      SUBROUTINE BOUND IS CALLED IN VECIM2.                            VISCOUS    97
      CALL VECIM2(IADD)                                                 VISCOUS    98
   20 CONTINUE                                                          VISCOUS    99
      IF(LK.EQ.2)RETURN                                                 VISCOUS   100
      DO 160 I=1,4                                                      VISCOUS   101
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)+AU1(2,2,I;NXM4)                    VISCOUS   102
  160 CONTINUE                                                          VISCOUS   103
      CALL DVISCC                                                       VISCOUS   104
      CALL VEC1                                                         VISCOUS   105
      CALL DAMPC                                                        VISCOUS   106
C                                                                       VISCOUS   107
C     CORRECTOR STEP WITH BACKWORD DIFFERENCING                         VISCOUS   108
C                                                                       VISCOUS   109
      IF(FDT.GT.1.)GO TO 30                                             VISCOUS   110
      DO 210 I=1,4                                                      VISCOUS   111
      AU1(2,2,I;NXM4)=0.5*(AU1(2,2,I;NXM4)-DT(2,2;NXM4)*((AM(2,2,I;NXM4)VISCOUS   112
     1-AM(2,1,I;NXM4))+(AN(2,2,I;NXM4)-AN(1,2,I;NXM4))))                VISCOUS   113
  210 CONTINUE                                                          VISCOUS   114
      CALL VEC2                                                         VISCOUS   115
      CALL BOUND                                                        VISCOUS   116
      GO TO 40                                                          VISCOUS   117
   30 CONTINUE                                                          VISCOUS   118
      DO 35 I=1,4                                                       VISCOUS   119
      DAU(2,2,I;NXM4)=-DT(2,2;NXM4)*(AM(2,2,I;NXM4)-AM(2,1,I;NXM4)+     VISCOUS   120
     1AN(2,2,I;NXM4)-AN(1,2,I;NXM4))/AJ(2,2;NXM4)                       VISCOUS   121
   35 CONTINUE                                                          VISCOUS   122
      IADD=0                                                            VISCOUS   123
      CALL IMPY(IADD)                                                   VISCOUS   124
      DO 36 I=1,4                                                       VISCOUS   125
   36 AU1(2,2,I;NXM4)=.5*(AJ1(2,2,I;NXM4)+DAU(2,2,I;NXM4)*AJ(2,2;NXM4)) VISCOUS   126
      IADD=1                                                            VISCOUS   127
C     SUBROUTINE BOUND IS CALLED IN VECIM2.                             VISCOUS   128
      CALL VECIM2(IADD)                                                 VISCOUS   129
   40 CONTINUE                                                          VISCOUS   130
      IF(LK.EQ.2)RETURN                                                 VISCOUS   131
      L=L+1                                                             VISCOUS   132
      LSTEP=2                                                           VISCOUS   133
      QX(2,2;NXM4)=GAMA*P(2,2;NXM4)/RO(2,2;NXM4)                        VISCOUS   134
      QX(2,2;NXM4)=VSQRT(QX(2,2;NXM4);QX(2,2;NXM4))                     VISCOUS   135
      DT(2,2;NXM4)=FDT*YETA(2,2;NXM4)            /(VABS(V(2,2;NXM4);    VISCOUS   136
     1QY(2,2;NXM4))+QX(2,2;NXM4))                                       VISCOUS   137
      DTM=Q8SMIN(DT(2,2;NXM4))                                          VISCOUS   138
      DTML=FDTL*DTM                                                     VISCOUS   139
      B1(2,2;NXM4)=DT(2,2;NXM4).GT.DTML                                 VISCOUS   140
      DT(2,2;NXM4)=Q8VCTRL(DTML,B1(2,2;NXM4);DT(2,2;NXM4))              VISCOUS   141
      TIME=TIME+DTM                                                     VISCOUS   142
      CALL DVISCC                                                       VISCOUS   143
      CALL VEC1                                                         VISCOUS   144
      CALL DAMPC                                                        VISCOUS   145
C                                                                       VISCOUS   146
C     PREDICTOR STEP WITH BACKWARD DIFFERENCING.                        VISCOUS   147
C                                                                       VISCOUS   148
      IF(FDT.GT.1.)GO TO 50                                             VISCOUS   149
      DO 110 I=1,4                                                      VISCOUS   150
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)-DT(2,2;NXM4)*((AM(2,2,I;NXM4)-AM(2,VISCOUS   151
     11,I;NXM4))+(AN(2,2,I;NXM4)-AN(1,2,I;NXM4)))                       VISCOUS   152
```

```
 110  CONTINUE                                                          VISCOUS    153
      CALL VEC2                                                         VISCOUS    154
      CALL BOUND                                                        VISCOUS    155
      GO TO 60                                                          VISCOUS    156
  50  CONTINUE                                                          VISCOUS    157
      DO 55 I=1,4                                                       VISCOUS    158
      DAU(2,2,I;NXM4)=-DT(2,2;NXM4)*(AM(2,2,1;NXM4)-AM(2,1,I;NXM4)+     VISCOUS    159
     1AN(2,2,I;NXM4)-AN(1,2,I;NXM4))/AJ(2,2;NXM4)                       VISCOUS    160
  55  CONTINUE                                                          VISCOUS    161
      IADD=0                                                            VISCOUS    162
      CALL IMPY(IADD)                                                   VISCOUS    163
      DO 56 I=1,4                                                       VISCOUS    164
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)+DAU(2,2,I;NXM4)*AJ(2,2;NXM4)       VISCOUS    165
  56  CONTINUE                                                          VISCOUS    166
C      SUBROUTINE BOUND IS CALLED IN VECIM2.                            VISCOUS    167
      CALL VECIM2(IADD)                                                 VISCOUS    168
  60  CONTINUE                                                          VISCOUS    169
      IF(LK.EQ.2)RETURN                                                 VISCOUS    170
      DO 165 I=1,4                                                      VISCOUS    171
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)+AU1(2,2,I;NXM4)                    VISCOUS    172
      IF(NSTRUT.EQ.0)GO TO 165                                          VISCOUS    173
      DO 150 N=1,NS                                                     VISCOUS    174
      NN=N+NS1-1                                                        VISCOUS    175
 150  AUS1(N,I)=AU1(MS1,NN,I)                                           VISCOUS    176
 165   CONTINUE                                                         VISCOUS    177
      CALL DVISCP                                                       VISCOUS    178
      CALL VEC1                                                         VISCOUS    179
      CALL DAMPP                                                        VISCOUS    180
C                                                                       VISCOUS    181
C     CORRECTOR STEP WITH FORWARD DIFFERENCING                          VISCOUS    182
C                                                                       VISCOUS    183
      IF(FDT.GT.1.)GO TO 70                                             VISCOUS    184
      DO 215 I=1,4                                                      VISCOUS    185
      AU1(2,2,I;NXM4)=0.5*(AU1(2,2,I;NXM4)-DT(2,2;NXM4)*((AM(2,3,I;NXM4)VISCOUS    186
     1-AM(2,2,I;NXM4))+(AN(3,2,I;NXM4)-AN(2,2,I;NXM4))))                VISCOUS    187
      IF(NSTRUT.EQ.0)GO TO 215                                          VISCOUS    188
      DO 120 N=1,NS                                                     VISCOUS    189
      NN=N+NS1-1                                                        VISCOUS    190
      AU1(MS1,NN,I)=0.5*(AUS1(N,I)-DT(MS1,NN)*(AM(MS1,NN+1,I)-AM(MS1,NN,VISCOUS    191
     1I)+ANS(N,I)-AN(MS1,NN,I)))                                        VISCOUS    192
 120  CONTINUE                                                          VISCOUS    193
 215  CONTINUE                                                          VISCOUS    194
      CALL VEC2                                                         VISCOUS    195
      CALL BOUND                                                        VISCOUS    196
      GO TO 80                                                          VISCOUS    197
  70  CONTINUE                                                          VISCOUS    198
      DO 75 I=1,4                                                       VISCOUS    199
      DAU(2,2,I;NXM4)=-DT(2,2;NXM4)*(AM(2,3,1;NXM4)-AM(2,2,I;NXM4)+     VISCOUS    200
     1AN(3,2,I;NXM4)-AN(2,2,I;NXM4))/AJ(2,2;NXM4)                       VISCOUS    201
      IF(NSTRUT.EQ.0)GO TO 75                                           VISCOUS    202
      DO 76 N=1,NS                                                      VISCOUS    203
      NN=N+NS1-1                                                        VISCOUS    204
      DAU(MS1,NN,I)=-DT(MS1,NN)*(AM(MS1,NN+1,I)-AM(MS1,NN,I)+           VISCOUS    205
     1ANS(N,I)-AN(MS1,NN,I))/AJ(MS1,NN)                                 VISCOUS    206
 76   CONTINUE                                                          VISCOUS    207
 75   CONTINUE                                                          VISCOUS    208
      IADD=1                                                            VISCOUS    209
      CALL IMPY(IADD)                                                   VISCOUS    210
      DO 77 I=1,4                                                       VISCOUS    211
  77  AU1(2,2,I;NXM4)=.5*(AU1(2,2,I;NXM4)+DAU(2,2,I;NXM4)*AJ(2,2;NXM4)) VISCOUS    212
      IADD=0                                                            VISCOUS    213
C     SUBROUTINE BOUND IS CALLED IN VECIM2.                             VISCOUS    214
      CALL VECIM2(IADD)                                                 VISCOUS    215
  80  CONTINUE                                                          VISCOUS    216
      IF(LK.EQ.2)RETURN                                                 VISCOUS    217
      IF(NFLOW.NE.2)GO TO 216                                           VISCOUS    218
      LL=L/20                                                           VISCOUS    219
      LL=LL*20                                                          VISCOUS    220
```

```
      IF(LL.NE.L)GO TO 216                                              VISCOUS    221
      CALL TURB                                                         VISCOUS    222
216    CONTINUE                                                         VISCOUS    223
      ERRO(1,1;NXM)=(RO(1,1;NXM)-ERRO(1,1;NXM))/ERRO(1,1;NXM)/FDT/LSTEP VISCOUS    224
      ERRO(1,1;NXM)=VABS(ERRO(1,1;NXM);ERRO(1,1;NXM))                   VISCOUS    225
      ERRMX=Q8SMAX(ERRO(1,1;NXM))                                       VISCOUS    226
      IF(ERRMX.LE.CRIT)GO TO 217                                        VISCOUS    227
      IF(TIME.GT.TIMEL)GO TO 217                                        VISCOUS    228
      LL=L/50                                                           VISCOUS    229
      LL=LL*50                                                          VISCOUS    230
      IF(LL.NE.L)GO TO 220                                              VISCOUS    231
      SIGX(1,1;NXM)=ERRO(1,1;NXM)*ERRO(1,1;NXM)                         VISCOUS    232
      ERRAVG=Q8SSUM(SIGX(1,1;NXM))                                      VISCOUS    233
      ERRAVG=SQRT(ERRAVG)/NXM                                           VISCOUS    234
      IF(ERRAVG.LE.CRITAVG) GO TO 217                                   VISCOUS    235
      WRITE(6,510)L,ERRMX,ERRAVG                                        VISCOUS    236
510   FORMAT(/,10X,'L=',I5,10X,'MAX.ERROR=',E12.5,10X,                  VISCOUS    237
     1'MEAN SQRT ERROR=',E12.5)                                         VISCOUS    238
      LL=L/LW                                                           VISCOUS    239
      LL=LL*LW                                                          VISCOUS    240
      IF(LL.NE.L)GO TO 220                                              VISCOUS    241
217   CONTINUE                                                          VISCOUS    242
      CALL PRINT                                                        VISCOUS    243
      CALL SPILL(SWEEP)                                                 VISCOUS    244
      IF(ERRAVG.LE.CRITAVG) WRITE(6,600)                                VISCOUS    245
      IF(ERRAVG.LE.CRITAVG) GO TO 218                                   VISCOUS    246
      IF(TIME.GT.TIMEL) WRITE(6,610)                                    VISCOUS    247
      IF(TIME.GT.TIMEL) GO TO 218                                       VISCOUS    248
      IF(ERRMX.LE.CRIT) WRITE(6,620)                                    VISCOUS    249
      IF(ERRMX.LE.CRIT)GO TO 218                                        VISCOUS    250
220    CONTINUE                                                         VISCOUS    251
      L=L+1                                                             VISCOUS    252
      IF(L.LE.LMAX)GO TO 1                                              VISCOUS    253
      L=L-1                                                             VISCOUS    254
      WRITE(6,630)                                                      VISCOUS    255
218   CONTINUE                                                          VISCOUS    256
      REWIND LFO                                                        VISCOUS    257
      WRITE(LFO,561) L,TIME                                             VISCOUS    258
      DO 300 N=1,N1                                                     VISCOUS    259
      WRITE(LFO,560)(X(M,N),Y(M,N),U(M,N),V(M,N),P(M,N),T(M,N),RO(M,N), VISCOUS    260
     1SH(M,N),H(M,N),VIST(M,N),M=1,M1)                                  VISCOUS    261
300   CONTINUE                                                          VISCOUS    262
      IF(NSTRUT.EQ.0)GO TO 310                                          VISCOUS    263
      WRITE(LFO,560)(XS(N),YS(N),US(N),VS(N),PS(N),TS(N),ROS(N),SHS(N), VISCOUS    264
     1HS(N),VISS(N),N=1,NS)                                             VISCOUS    265
310   CONTINUE                                                          VISCOUS    266
560   FORMAT(1X, 2F9.6,2F12.5,F14.5,F13.5,F11.5,F14.5,F15.6,F10.6)      VISCOUS    267
 561  FORMAT(1X,I6,2X,E15.7)                                            VISCOUS    268
600   FORMAT(//,20X,'PROGRAM TERMINATED ON MEAN SQRT ERROR CRITERION')  VISCOUS    269
 610  FORMAT(//,20X,'PROGRAM TERMINATED ON PHYSICAL TIME ',             VISCOUS    270
     1              'CONVERGENCE CRITERION')                            VISCOUS    271
 620  FORMAT(//,20X,'PROGRAM TERMINATED ON LOCAL ERROR CRITERION')      VISCOUS    272
 630  FORMAT(//,20X,'PROGRAM TERMINATED ON LMAX')                       VISCOUS    273
      RETURN                                                            VISCOUS    274
      END                                                               VISCOUS    275
      SUBROUTINE INVICID(SWEEP)                                         INVICID      1
      BIT B1                                                            INVICID      2
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 INVICID      3
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              INVICID      4
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   INVICID      5
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      INVICID      6
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          INVICID      7
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        INVICID      8
     2 PS(55),TS(55),SHS(55),HS(55)                                     INVICID      9
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)             INVICID     10
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),         INVICID     11
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)         INVICID     12
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)        INVICID     13
```

```
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                INVICID    14
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                INVICID    15
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                INVICID    16
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),              INVICID    17
     1 YSETA(55),AJS(55)                                                INVICID    18
      COMMON/F9/INT(55),TE1(55),TE2(55),TE3(55),TE4(55)                 INVICID    19
      COMMON/F10/DT(61,55),DTM                                          INVICID    20
      COMMON/F11/ERRO(61,55),B1(61,55)                                  INVICID    21
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,    INVICID    22
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                  INVICID    23
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF        INVICID    24
C                                                                       INVICID    25
C     INITIAL GUESS                                                     INVICID    26
C                                                                       INVICID    27
      V(1,1;NXM)=YXI(1,1;NXM)/XXI(1,1;NXM)*UF                           INVICID    28
      U(1,1;NXM)=UF*UF-V(1,1;NXM)*V(1,1;NXM)                            INVICID    29
      U(1,1;NXM)=VSQRT(U(1,1;NXM);U(1,1;NXM))                           INVICID    30
      T(1,1;NXM)=TF                                                     INVICID    31
      P(1,1;NXM)=PF                                                     INVICID    32
      RO(1,1;NXM)=ROF                                                   INVICID    33
      SH(1,1;NXM)=SHF                                                   INVICID    34
      H(1,1;NXM)=HF                                                     INVICID    35
      IF(NSTRUT.EQ.0) GO TO 4                                           INVICID    36
      VS(1;NS)=YSXI(1;NS)/XSXI(1;NS)*UF                                 INVICID    37
      US(1;NS)=UF*UF-VS(1;NS)*VS(1;NS)                                  INVICID    38
      US(1;NS)=VSQRT(US(1;NS);US(1;NS))                                 INVICID    39
      PS(1;NS)=PF                                                       INVICID    40
      TS(1;NS)=TF                                                       INVICID    41
      SHS(1;NS)=SHF                                                     INVICID    42
      ROS(1;NS)=ROF                                                     INVICID    43
      HS(1;NS)=HF                                                       INVICID    44
 4    CONTINUE                                                          INVICID    45
      IF(ILT.EQ.1)GO TO 2                                               INVICID    46
      READ(LFI,561) L,TIME                                              INVICID    47
      DO 3 N=1,N1                                                       INVICID    48
       READ(LFI,560)(X(M,N),Y(M,N),U(M,N),V(M,N),P(M,N),T(M,N),RO(M,N), INVICID    49
     1SH(M,N),H(M,N),VIST(M,N),M=1,M1)                                  INVICID    50
 3    CONTINUE                                                          INVICID    51
      LMAX=LMAX+L                                                       INVICID    52
      IF(NSTRUT.EQ.0)GO TO 2                                            INVICID    53
       READ(LFI,560)(XS(N),YS(N),US(N),VS(N),PS(N),TS(N),ROS(N),SHS(N), INVICID    54
     1HS(N),VISS(N),N=1,NS)                                             INVICID    55
 2    CONTINUE                                                          INVICID    56
      L=L+1                                                             INVICID    57
      CALL IBOUND                                                       INVICID    58
 1    CONTINUE                                                          INVICID    59
      LSTEP=1                                                           INVICID    60
      ERRO(1,1;NXM)=RO(1,1;NXM)                                         INVICID    61
      QX(2,2;NXM4)=GAMA*P(2,2;NXM4)/RO(2,2;NXM4)                        INVICID    62
      QX(2,2;NXM4)=VSQRT(QX(2,2;NXM4);QX(2,2;NXM4))                     INVICID    63
      DT(2,2;NXM4)=FDT*YETA(2,2;NXM4)            /(VABS(V(2,2;NXM4);    INVICID    64
     1QY(2,2;NXM4))+QX(2,2;NXM4))                                       INVICID    65
      DTM=Q8SMIN(DT(2,2;NXM4))                                          INVICID    66
      DTML=FDTL*DTM                                                     INVICID    67
      B1(2,2;NXM4)=DT(2,2;NXM4).GT.DTML                                 INVICID    68
      DT(2,2;NXM4)=Q8VCTRL(DTML,B1(2,2;NXM4);DT(2,2;NXM4))              INVICID    69
      TIME=TIME+DTM                                                     INVICID    70
      CALL IVEC1                                                        INVICID    71
      CALL DAMPP                                                        INVICID    72
C                                                                       INVICID    73
C     PREDICTOR STEP WITH  FORWARD DIFFERENCING.                        INVICID    74
C                                                                       INVICID    75
      DO 100 I=1,4                                                      INVICID    76
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)-DT(2,2;NXM4)*((AM(2,3,I;NXM4)-AM(2,INVICID    77
     12,I;NXM4))+(AN(3,2,I;NXM4)-AN(2,2,I;NXM4)))                       INVICID    78
      IF(NSTRUT.EQ.0)GO TO 100                                          INVICID    79
      DO 105 N=1,NS                                                     INVICID    80
      NN=N+NS1-1                                                        INVICID    81
```

```
      AU1(MS1,NN,I)=AU(MS1,NN,I)-DT(MS1,NN)*(AM(MS1,NN+1,I)-AM(MS1,NN,I)INVICID
     1+ANS(N,I)-AN(MS1,NN,I))                                           INVICID
 105  CONTINUE                                                           INVICID
 100  CONTINUE                                                           INVICID
      CALL VEC2                                                          INVICID
      CALL IBOUND                                                        INVICID
      IF(LK.EQ.2)RETURN                                                  INVICID
      DO 160 I=1,4                                                       INVICID
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)+AU1(2,2,I;NXM4)                     INVICID
 160  CONTINUE                                                           INVICID
      CALL IVEC1                                                         INVICID
      CALL DAMPC                                                         INVICID
C                                                                        INVICID
C     CORRECTOR STEP WITH BACKWORD DIFFERENCING                          INVICID
C                                                                        INVICID
      DO 210 I=1,4                                                       INVICID
      AU1(2,2,I;NXM4)=0.5*(AU1(2,2,I;NXM4)-DT(2,2;NXM4)*((AM(2,2,I;NXM4)INVICID
     1-AM(2,1,I;NXM4))+(AN(2,2,I;NXM4)-AN(1,2,I;NXM4))))                 INVICID
 210  CONTINUE                                                           INVICID
      CALL VEC2                                                          INVICID
      CALL IBOUND                                                        INVICID
      IF(LK.EQ.2)RETURN                                                  INVICID
      L=L+1                                                              INVICID
      LSTEP=2                                                            INVICID
      QX(2,2;NXM4)=GAMA*P(2,2;NXM4)/RO(2,2;NXM4)                         INVICID
      QX(2,2;NXM4)=VSQRT(QX(2,2;NXM4);QX(2,2;NXM4))                      INVICID
      DT(2,2;NXM4)=FDT*YETA(2,2;NXM4)            /(VABS(V(2,2;NXM4);     INVICID
     1QY(2,2;NXM4))+QX(2,2;NXM4))                                        INVICID
      DTM=Q8SMIN(DT(2,2;NXM4))                                           INVICID
      DTML=FDTL*DTM                                                      INVICID
      B1(2,2;NXM4)=DT(2,2;NXM4).GT.DTML                                  INVICID
      DT(2,2;NXM4)=Q8VCTRL(DTML,B1(2,2;NXM4);DT(2,2;NXM4))               INVICID
      TIME=TIME+DTM                                                      INVICID
      CALL IVEC1                                                         INVICID
      CALL DAMPC                                                         INVICID
C                                                                        INVICID
C     PREDICTOR STEP WITH BACKWARD DIFFERENCING.                         INVICID
C                                                                        INVICID
      DO 110 I=1,4                                                       INVICID
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)-D1(2,2;NXM4)*((AM(2,2,I;NXM4)-AM(2,INVICID
     11,I;NXM4))+(AN(2,2,I;NXM4)-AN(1,2,I;NXM4)))                        INVICID
 110  CONTINUE                                                           INVICID
      CALL VEC2                                                          INVICID
      CALL IBOUND                                                        INVICID
      IF(LK.EQ.2)RETURN                                                  INVICID
      DO 165 I=1,4                                                       INVICID
      AU1(2,2,I;NXM4)=AU(2,2,I;NXM4)+AU1(2,2,I;NXM4)                     INVICID
      IF(NSTRUT.EQ.0)GO TO 165                                           INVICID
      DO 150 N=1,NS                                                      INVICID
      NN=N+NS1-1                                                         INVICID
 150  AUS1(N,I)=AU1(MS1,NN,I)                                            INVICID
 165   CONTINUE                                                          INVICID
      CALL IVEC1                                                         INVICID
      CALL DAMPP                                                         INVICID
C                                                                        INVICID
C     CORRECTOR STEP WITH FORWARD DIFFERENCING                           INVICID
C                                                                        INVICID
      DO 215 I=1,4                                                       INVICID
      AU1(2,2,I;NXM4)=0.5*(AU1(2,2,I;NXM4)-DT(2,2;NXM4)*((AM(2,3,I;NXM4)INVICID
     1-AM(2,2,I;NXM4))+(AN(3,2,I;NXM4)-AN(2,2,I;NXM4))))                 INVICID
      IF(NSTRUT.EQ.0)GO TO 215                                           INVICID
      DO 120 N=1,NS                                                      INVICID
      NN=N+NS1-1                                                         INVICID
      AU1(MS1,NN,I)=0.5*(AUS1(N,I)-DT(MS1,NN)*(AM(MS1,NN+1,I)-AM(MS1,NN,INVICID
     1I)+ANS(N,I)-AN(MS1,NN,I)))                                         INVICID
 120  CONTINUE                                                           INVICID
 215  CONTINUE                                                           INVICID
      CALL VEC2                                                          INVICID
```

```
      CALL IBOUND                                                       INVICID    150
      IF(LK.EQ.2)RETURN                                                 INVICID    151
      ERRO(1,1;NXM)=(RO(1,1;NXM)-ERRO(1,1;NXM))/ERRO(1,1;NXM)/FDT/LSTEP INVICID    152
      ERRO(1,1;NXM)=VABS(ERRO(1,1;NXM);ERRO(1,1;NXM))                   INVICID    153
      ERRMX=Q8SMAX(ERRO(1,1;NXM))                                       INVICID    154
      IF(ERRMX.LE.CRIT)GO TO 217                                        INVICID    155
      IF(TIME.GT.TIMEL)GO TO 217                                        INVICID    156
      LL=L/50                                                           INVICID    157
      LL=LL*50                                                          INVICID    158
      IF(LL.NE.L)GO TO 220                                              INVICID    159
      SIGX(1,1;NXM)=ERRO(1,1;NXM)*ERRO(1,1;NXM)                         INVICID    160
      ERRAVG=Q8SSUM(SIGX(1,1;NXM))                                      INVICID    161
      ERRAVG=SQRT(ERRAVG)/NXM                                           INVICID    162
      IF(ERRAVG.LE.CRITAVG) GO TO 217                                   INVICID    163
      WRITE(6,510)L,ERRMX,ERRAVG                                        INVICID    164
510   FORMAT(/,10X,'L=',I5,10X,'MAX. ERROR=',E12.5,10X,                 INVICID    165
     1'MEAN SQRT ERROR=',E12.5)                                         INVICID    166
      LL=L/LW                                                           INVICID    167
      LL=LL*LW                                                          INVICID    168
      IF(LL.NE.L)GO TO 220                                              INVICID    169
217   CONTINUE                                                          INVICID    170
      CALL PRINT                                                        INVICID    171
      CALL SPILL(SWEEP)                                                 INVICID    172
      IF(ERRAVG.LE.CRITAVG) WRITE(6,600)                                INVICID    173
      IF(ERRAVG.LE.CRITAVG) GO TO 218                                   INVICID    174
      IF(TIME.GT.TIMEL) WRITE(6,610)                                    INVICID    175
      IF(TIME.GT.TIMEL) GO TO 218                                       INVICID    176
      IF(ERRMX.LE.CRIT) WRITE(6,620)                                    INVICID    177
      IF(ERRMX.LE.CRIT)GO TO 218                                        INVICID    178
220    CONTINUE                                                         INVICID    179
      L=L+1                                                             INVICID    180
      IF(L.LE.LMAX)GO TO 1                                              INVICID    181
      L=L-1                                                             INVICID    182
      WRITE(6,630)                                                      INVICID    183
218   CONTINUE                                                          INVICID    184
      REWIND LFO                                                        INVICID    185
      WRITE(LFO,561) L,TIME                                             INVICID    186
      DO 300 N=1,N1                                                     INVICID    187
      WRITE(LFO,560)(X(M,N),Y(M,N),U(M,N),V(M,N),P(M,N),T(M,N),RO(M,N), INVICID    188
     1SH(M,N),H(M,N),VIST(M,N),M=1,M1)                                  INVICID    189
300   CONTINUE                                                          INVICID    190
      IF(NSTRUT.EQ.0)GO TO 310                                          INVICID    191
      WRITE(LFO,560)(XS(N),YS(N),US(N),VS(N),PS(N),TS(N),ROS(N),SHS(N), INVICID    192
     1HS(N),VISS(N),N=1,NS)                                             INVICID    193
310   CONTINUE                                                          INVICID    194
560   FORMAT(1X, 2F9.6,2F12.5,F14.5,F13.5,F11.5,F14.5,F15.6,F10.6)      INVICID    195
 561  FORMAT(1X,I6,2X,E15.7)                                            INVICID    196
600   FORMAT(//,20X,'PROGRAM TERMINATED ON MEAN SQRT ERROR CRITERION')  INVICID    197
 610  FORMAT(//,20X,'PROGRAM TERMINATED ON PHYSICAL TIME ',             INVICID    198
     1              'CONVERGENCE CRITERION')                            INVICID    199
 620  FORMAT(//,20X,'PROGRAM TERMINATED ON LOCAL ERROR CRITERION')      INVICID    200
 630  FORMAT(//,20X,'PROGRAM TERMINATED ON LMAX')                       INVICID    201
      RETURN                                                            INVICID    202
      END                                                               INVICID    203
      SUBROUTINE DVISCP                                                 DVISCP       1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 DVISCP       2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              DVISCP       3
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   DVISCP       4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      DVISCP       5
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          DVISCP       6
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        DVISCP       7
     2 PS(55),TS(55),SHS(55),HS(55)                                     DVISCP       8
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)             DVISCP       9
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),         DVISCP      10
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)         DVISCP      11
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)        DVISCP      12
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                DVISCP      13
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                DVISCP      14
```

```
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                DVISCP      15
       COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),             DVISCP      16
     1 YSETA(55),AJS(55)                                                DVISCP      17
       COMMON/F9/INT(55),TE1(55),TE2(55),TE3(55),TE4(55)                DVISCP      18
       COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF       DVISCP      19
       VIS(1,1;NXM)=VISL(1,1;NXM)+VIST(1,1;NXM)                         DVISCP      20
       NXMM=NXM-M1                                                      DVISCP      21
       AU(1,2,1;NXMM)=U(1,2;NXMM)-U(1,1;NXMM)                           DVISCP      22
       AU(1,1,1;M1)=U(1,2;M1)-U(1,1;M1)                                 DVISCP      23
       AU(1,2,2;NXMM)=V(1,2;NXMM)-V(1,1;NXMM)                           DVISCP      24
       AU(1,1,2;M1)=V(1,2;M1)-V(1,1;M1)                                 DVISCP      25
       AM(1,2,1;NXMM)=SH(1,2;NXMM)-SH(1,1;NXMM)                         DVISCP      26
       AM(1,1,1;M1)=SH(1,2;M1)-SH(1,1;M1)                               DVISCP      27
       AU(2,1,3;NXM1)=U(2,1;NXM1)-U(1,1;NXM1)                           DVISCP      28
       TE1(1;N1)=Q8VGATHR(U(1,1;NXM),INT(1;N1);TE1(1;N1))               DVISCP      29
       TE2(1;N1)=Q8VGATHR( U(2,1;NXM),INT(1;N1);TE2(1;N1))              DVISCP      30
       TE3(1;N1)=TE2(1;N1)-TE1(1;N1)                                    DVISCP      31
       AU(1,1,3;NXM)=Q8VSCATR(TE3(1;N1),INT(1;N1);AU(1,1,3;NXM))        DVISCP      32
       AU(2,1,4;NXM1)=V(2,1;NXM1)-V(1,1;NXM1)                           DVISCP      33
       TE1(1;N1)=Q8VGATHR(V(1,1;NXM),INT(1;N1);TE1(1;N1))               DVISCP      34
       TE2(1;N1)=Q8VGATHR( V(2,1;NXM),INT(1;N1);TE2(1;N1))              DVISCP      35
       TE3(1;N1)=TE2(1;N1)-TE1(1;N1)                                    DVISCP      36
       AU(1,1,4;NXM)=Q8VSCATR(TE3(1;N1),INT(1;N1);AU(1,1,4;NXM))        DVISCP      37
       AM(2,1,2;NXM1)=SH(2,1;NXM1)-SH(1,1;NXM1)                         DVISCP      38
       TE1(1;N1)=Q8VGATHR(SH(1,1;NXM),INT(1;N1);TE1(1;N1))              DVISCP      39
       TE2(1;N1)=Q8VGATHR(SH(2,1;NXM),INT(1;N1);TE2(1;N1))              DVISCP      40
       TE3(1;N1)=TE2(1;N1)-TE1(1;N1)                                    DVISCP      41
       AM(1,1,2;NXM)=Q8VSCATR(TE3(1;N1),INT(1;N1);AM(1,1,2;NXM))        DVISCP      42
       IF(NSTRUT.EQ.0)GO TO 15                                          DVISCP      43
       DO 10 N=NS1,NS2                                                  DVISCP      44
       AU(MS,N,3)=U(MSP1,N)-U(MS,N)                                     DVISCP      45
       AU(MS,N,4)=V(MSP1,N)-V(MS,N)                                     DVISCP      46
10     AM(MS,N,2)=SH(MSP1,N)-SH(MS,N)                                   DVISCP      47
15     CONTINUE                                                         DVISCP      48
       AM(1,1,3;NXM)=(YETA(1,1;NXM)*AM(1,1,1;NXM)-YXI(1,1;NXM)*AM(1,1,2; DVISCP      49
     1NXM))/AJ(1,1;NXM)                                                 DVISCP      50
       AM(1,1,4;NXM)=(XXI(1,1;NXM)*AM(1,1,2;NXM)-XETA(1,1;NXM)*AM(1,1,1; DVISCP      51
     1NXM))/AJ(1,1;NXM)                                                 DVISCP      52
       AN(1,1,1;NXM)=(YETA(1,1;NXM)*AU(1,1,1;NXM)-YXI(1,1;NXM)*AU(1,1,3; DVISCP      53
     1NXM))/AJ(1,1;NXM)                                                 DVISCP      54
       AN(1,1,2;NXM)=(XXI(1,1;NXM)*AU(1,1,3;NXM)-XETA(1,1;NXM)*AU(1,1,1; DVISCP      55
     1NXM))/AJ(1,1;NXM)                                                 DVISCP      56
       AN(1,1,3;NXM)=(YETA(1,1;NXM)*AU(1,1,2;NXM)-YXI(1,1;NXM)*AU(1,1,4; DVISCP      57
     1NXM))/AJ(1,1;NXM)                                                 DVISCP      58
       AN(1,1,4;NXM)=(XXI(1,1;NXM)*AU(1,1,4;NXM)-XETA(1,1;NXM)*AU(1,1,2; DVISCP      59
     1NXM))/AJ(1,1;NXM)                                                 DVISCP      60
       SIGX(1,1;NXM)=P(1,1;NXM)+2.*VIS(1,1;NXM)/3.*(AN(1,1,4;NXM)-2.*   DVISCP      61
     1AN(1,1,1;NXM))                                                    DVISCP      62
       SIGY(1,1;NXM)=P(1,1;NXM)+2.*VIS(1,1;NXM)/3.*(AN(1,1,1;NXM)-2.*   DVISCP      63
     1AN(1,1,4;NXM))                                                    DVISCP      64
       TAUXY(1,1;NXM)=-VIS(1,1;NXM)*(AN(1,1,2;NXM)+AN(1,1,3;NXM))       DVISCP      65
       QX(1,1;NXM)=-(VISL(1,1;NXM)/PR+VIST(1,1;NXM)/PRT)*AM(1,1,3;NXM)  DVISCP      66
       QY(1,1;NXM)=-(VISL(1,1;NXM)/PR+VIST(1,1;NXM)/PRT)*AM(1,1,4;NXM)  DVISCP      67
       IF(NSTRUT.EQ.0)RETURN                                            DVISCP      68
       DO 20 N=1,NS                                                     DVISCP      69
       NN=N-1+NS1                                                       DVISCP      70
       AUS(N,3)=US(N)-U(MS1,NN)                                         DVISCP      71
       AUS(N,4)=VS(N)-V(MS1,NN)                                         DVISCP      72
20     AMS(N,2)=SHS(N)-SH(MS1,NN)                                       DVISCP      73
       DO 30 N=2,NS                                                     DVISCP      74
       AUS(N,1)=US(N)-US(N-1)                                           DVISCP      75
       AUS(N,2)=VS(N)-VS(N-1)                                           DVISCP      76
30     AMS(N,1)=SHS(N)-SHS(N-1)                                         DVISCP      77
       AUS(1,1)=US(1)-U(MS,NS1M1)                                       DVISCP      78
       AUS(1,2)=VS(1)-V(MS,NS1M1)                                       DVISCP      79
       AMS(1,1)=SHS(1)-SH(MS,NS1M1)                                     DVISCP      80
       AMS(1,3;NS)=(YSETA(1;NS)*AMS(1,1;NS)-YSXI(1;NS)*AMS(1,2;NS))/    DVISCP      81
     1AJS(1;NS)                                                         DVISCP      82
```

```
      AMS(1,4;NS)=(XSXI(1;NS)*AMS(1,2;NS)-XSETA(1;NS)*AMS(1,1;NS))/      DVISCP      83
     1AJS(1;NS)                                                          DVISCP      84
      ANS(1,1;NS)=(YSETA(1;NS)*AUS(1,1;NS)-YSXI(1;NS)*AUS(1,3;NS))/      DVISCP      85
     1AJS(1;NS)                                                          DVISCP      86
      ANS(1,2;NS)=(XSXI(1;NS)*AUS(1,3;NS)-XSETA(1;NS)*AUS(1,1;NS))/      DVISCP      87
     1AJS(1;NS)                                                          DVISCP      88
      ANS(1,3;NS)=(YSETA(1;NS)*AUS(1,2;NS)-YSXI(1;NS)*AUS(1,4;NS))/      DVISCP      89
     1AJS(1;NS)                                                          DVISCP      90
      ANS(1,4;NS)=(XSXI(1;NS)*AUS(1,4;NS)-XSETA(1;NS)*AUS(1,2;NS))/      DVISCP      91
     1AJS(1;NS)                                                          DVISCP      92
      SIGXS(1;NS)=PS(1;NS)+2.*VISS(1;NS)/3.*(ANS(1,4;NS)-2.*ANS(1,1;NS))DVISCP      93
      SIGYS(1;NS)=PS(1;NS)+2.*VISS(1;NS)/3.*(ANS(1,1;NS)-2.*ANS(1,4;NS))DVISCP      94
      TAUXYS(1;NS)=-VISS(1;NS)*(ANS(1,2;NS)+ANS(1,3;NS))                 DVISCP      95
      QXS(1;NS)= -VISS(1;NS)/PR*AMS(1,3;NS)                              DVISCP      96
      QYS(1;NS)= -VISS(1;NS)/PR*AMS(1,4;NS)                              DVISCP      97
      RETURN                                                             DVISCP      98
      END                                                                DVISCP      99
      SUBROUTINE DVISCC                                                  DVISCC       1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 DVISCC       2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                               DVISCC       3
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   DVISCC       4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                       DVISCC       5
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),           DVISCC       6
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                         DVISCC       7
     2 PS(55),TS(55),SHS(55),HS(55)                                      DVISCC       8
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)              DVISCC       9
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),          DVISCC      10
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)          DVISCC      11
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)         DVISCC      12
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                 DVISCC      13
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                 DVISCC      14
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                 DVISCC      15
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),               DVISCC      16
     1 YSETA(55),AJS(55)                                                 DVISCC      17
      COMMON/F9/INT(55),TE1(55),TE2(55),TE3(55),TE4(55)                  DVISCC      18
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF         DVISCC      19
      VIS(1,1;NXM)=VISL(1,1;NXM)+VIST(1,1;NXM)                           DVISCC      20
      NXMM=NXM-M1                                                        DVISCC      21
      AU(1,1,1;NXMM)=U(1,2;NXMM)-U(1,1;NXMM)                             DVISCC      22
      AU(1,N1,1;M1)=U(1,N1;M1)-U(1,N11;M1)                               DVISCC      23
      AU(1,1,2;NXMM)=V(1,2;NXMM)-V(1,1;NXMM)                             DVISCC      24
      AU(1,N1,2;M1)=V(1,N1;M1)-V(1,N11;M1)                               DVISCC      25
      AM(1,N1,1;M1)=SH(1,N1;M1)-SH(1,N11;M1)                             DVISCC      26
      AM(1,1,1;NXMM)=SH(1,2;NXMM)-SH(1,1;NXMM)                           DVISCC      27
      AU(1,1,3;NXM1)=J(2,1;NXM1)-U(1,1;NXM1)                             DVISCC      28
      TE1(1;N1)=Q8VGATHR(U(M11,1;NXM),INT(1;N1);TE1(1;N1))               DVISCC      29
      TE2(1;N1)=Q8VGATHR( U( M1,1;NXM),INT(1;N1);TE2(1;N1))              DVISCC      30
      TE3(1;N1)=TE2(1;N1)-TE1(1;N1)                                      DVISCC      31
      AU(M1,1,3;NXM)=Q8VSCATR(TE3(1;N1),INT(1;N1);AU(M1,1,3;NXM))        DVISCC      32
      AU(1,1,4;NXM1)=V(2,1;NXM1)-V(1,1;NXM1)                             DVISCC      33
      TE1(1;N1)=Q8VGATHR(V(M11,1;NXM),INT(1;N1);TE1(1;N1))               DVISCC      34
      TE2(1;N1)=Q8VGATHR( V( M1,1;NXM),INT(1;N1);TE2(1;N1))              DVISCC      35
      TE3(1;N1)=TE2(1;N1)-TE1(1;N1)                                      DVISCC      36
      AU(M1,1,4;NXM)=Q8VSCATR(TE3(1;N1),INT(1;N1);AU(M1,1,4;NXM))        DVISCC      37
      AM(1,1,2;NXM1)=SH(2,1;NXM1)-SH(1,1;NXM1)                           DVISCC      38
      TE1(1;N1)=Q8VGATHR(SH(M11,1;NXM),INT(1;N1);TE1(1;N1))              DVISCC      39
      TE2(1;N1)=Q8VGATHR(SH( M1,1;NXM),INT(1;N1);TE2(1;N1))              DVISCC      40
      TE3(1;N1)=TE2(1;N1)-TE1(1;N1)                                      DVISCC      41
      AM(M1,1,2;NXM)=Q8VSCATR(TE3(1;N1),INT(1;N1);AM(M1,1,2;NXM))        DVISCC      42
      IF(NSTRUT.EQ.0)GO TO 15                                            DVISCC      43
      DO 10 N=NS1,NS2                                                    DVISCC      44
      NN=N-NS1+1                                                         DVISCC      45
      AU(MS1,N,3)=US(NN)-U(MS1,N)                                        DVISCC      46
      AU(MS1,N,4)=VS(NN)-V(MS1,N)                                        DVISCC      47
   10 AM(MS1,N,2)=SHS(NN)-SH(MS1,N)                                      DVISCC      48
   15 CONTINUE                                                           DVISCC      49
      AM(1,1,3;NXM)=(YETA(1,1;NXM)*AM(1,1,1;NXM)-YXI(1,1;NXM)*AM(1,1,2; DVISCC      50
     1NXM))/AJ(1,1;NXM)                                                  DVISCC      51
```

```
      AM(1,1,4;NXM)=(XXI(1,1;NXM)*AM(1,1,2;NXM)-XETA(1,1;NXM)*AM(1,1,1;  DVISCC      52
     1NXM))/AJ(1,1;NXM)                                                  DVISCC      53
      AN(1,1,1;NXM)=(YETA(1,1;NXM)*AU(1,1,1;NXM)-YXI(1,1;NXM)*AU(1,1,3;  DVISCC      54
     1NXM))/AJ(1,1;NXM)                                                  DVISCC      55
      AN(1,1,2;NXM)=(XXI(1,1;NXM)*AU(1,1,3;NXM)-XETA(1,1;NXM)*AU(1,1,1;  DVISCC      56
     1NXM))/AJ(1,1;NXM)                                                  DVISCC      57
      AN(1,1,3;NXM)=(YETA(1,1;NXM)*AU(1,1,2;NXM)-YXI(1,1;NXM)*AU(1,1,4;  DVISCC      58
     1NXM))/AJ(1,1;NXM)                                                  DVISCC      59
      AN(1,1,4;NXM)=(XXI(1,1;NXM)*AU(1,1,4;NXM)-XETA(1,1;NXM)*AU(1,1,2;  DVISCC      60
     1NXM))/AJ(1,1;NXM)                                                  DVISCC      61
      SIGX(1,1;NXM)=P(1,1;NXM)+2.*VIS(1,1;NXM)/3.*(AN(1,1,4;NXM)-2.*     DVISCC      62
     1AN(1,1,1;NXM))                                                     DVISCC      63
      SIGY(1,1;NXM)=P(1,1;NXM)+2.*VIS(1,1;NXM)/3.*(AN(1,1,1;NXM)-2.*     DVISCC      64
     1AN(1,1,4;NXM))                                                     DVISCC      65
      TAUXY(1,1;NXM)=-VIS(1,1;NXM)*(AN(1,1,2;NXM)+AN(1,1,3;NXM))         DVISCC      66
      QX(1,1;NXM)=-(VISL(1,1;NXM)/PR+VIST(1,1;NXM)/PRT)*AM(1,1,3;NXM)    DVISCC      67
      QY(1,1;NXM)=-(VISL(1,1;NXM)/PR+VIST(1,1;NXM)/PRT)*AM(1,1,4;NXM)    DVISCC      68
      IF(NSTRUT.EQ.0)RETURN                                              DVISCC      69
      DO 20 N=1,NS                                                       DVISCC      70
      NN=N-1+NS1                                                         DVISCC      71
      AUS(N,3)=US(N)-U(MS1,NN)                                           DVISCC      72
      AUS(N,4)=VS(N)-V(MS1,NN)                                           DVISCC      73
20    AMS(N,2)=SHS(N)-SH(MS1,NN)                                         DVISCC      74
      DO 30 N=1,NSM1                                                     DVISCC      75
      AUS(N,1)=US(N+1)-US(N)                                             DVISCC      76
      AUS(N,2)=VS(N+1)-VS(N)                                             DVISCC      77
30    AMS(N,1)=SHS(N+1)-SHS(N)                                           DVISCC      78
      AUS(NS,1)=U(MS,NS2P1)-US(NS)                                       DVISCC      79
      AUS(NS,2)=V(MS,NS2P1)-VS(NS)                                       DVISCC      80
      AMS(NS,1)=SH(MS,NS2P1)-SHS(NS)                                     DVISCC      81
      AMS(1,3;NS)=(YSETA(1;NS)*AMS(1,1;NS)-YSXI(1;NS)*AMS(1,2;NS))/      DVISCC      82
     1AJS(1;NS)                                                          DVISCC      83
      AMS(1,4;NS)=(XSXI(1;NS)*AMS(1,2;NS)-XSETA(1;NS)*AMS(1,1;NS))/      DVISCC      84
     1AJS(1;NS)                                                          DVISCC      85
      ANS(1,1;NS)=(YSETA(1;NS)*AUS(1,1;NS)-YSXI(1;NS)*AUS(1,3;NS))/      DVISCC      86
     1AJS(1;NS)                                                          DVISCC      87
      ANS(1,2;NS)=(XSXI(1;NS)*AUS(1,3;NS)-XSETA(1;NS)*AUS(1,1;NS))/      DVISCC      88
     1AJS(1;NS)                                                          DVISCC      89
      ANS(1,3;NS)=(YSETA(1;NS)*AUS(1,2;NS)-YSXI(1;NS)*AUS(1,4;NS))/      DVISCC      90
     1AJS(1;NS)                                                          DVISCC      91
      ANS(1,4;NS)=(XSXI(1;NS)*AUS(1,4;NS)-XSETA(1;NS)*AUS(1,2;NS))/      DVISCC      92
     1AJS(1;NS)                                                          DVISCC      93
      SIGXS(1;NS)=PS(1;NS)+2.*VISS(1;NS)/3.*(ANS(1,4;NS)-2.*ANS(1,1;NS))DVISCC      94
      SIGYS(1;NS)=PS(1;NS)+2.*VISS(1;NS)/3.*(ANS(1,1;NS)-2.*ANS(1,4;NS))DVISCC      95
      TAUXYS(1;NS)=-VISS(1;NS)*(ANS(1,2;NS)+ANS(1,3;NS))                 DVISCC      96
      QXS(1;NS)= -VISS(1;NS)/PR*AMS(1,3;NS)                              DVISCC      97
      QYS(1;NS)= -VISS(1;NS)/PR*AMS(1,4;NS)                              DVISCC      98
      RETURN                                                             DVISCC      99
      END                                                                DVISCC     100
      SUBROUTINE VEC1                                                    VEC1         1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1  VEC1         2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                               VEC1         3
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),           VEC1         4
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                         VEC1         5
     2 PS(55),TS(55),SHS(55),HS(55)                                      VEC1         6
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),          VEC1         7
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)          VEC1         8
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)         VEC1         9
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                 VEC1        10
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                 VEC1        10
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                 VEC1        11
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                 VEC1        12
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),               VEC1        13
     1 YSETA(55),AJS(55)                                                 VEC1        14
      AN(1,1,4;NXM)=RO(1,1;NXM)*H(1,1;NXM)-P(1,1;NXM)                    VEC1        15
      AU(1,1,1;NXM)=RO(1,1;NXM)*U(1,1;NXM)                               VEC1        16
       AU(1,1,2;NXM)=AU(1,1,1;NXM)*U(1,1;NXM)+SIGX(1,1;NXM)              VEC1        17
       AU(1,1,3;NXM)=AU(1,1,1;NXM)*V(1,1;NXM)+TAUXY(1,1;NXM)             VEC1        18
```

```
      AU(1,1,4;NXM)=(AN(1,1,4;NXM)+SIGX(1,1;NXM))*U(1,1;NXM)+V(1,1;NXM)VEC1
     1*TAUXY(1,1;NXM)+QX(1,1;NXM)                                      VEC1
      AN(1,1,1;NXM)=RO(1,1;NXM)*V(1,1;NXM)                             VEC1
      AN(1,1,2;NXM)=AN(1,1,1;NXM)*U(1,1;NXM)+TAUXY(1,1;NXM)            VEC1
      AN(1,1,3;NXM)=AN(1,1,1;NXM)*V(1,1;NXM)+SIGY(1,1;NXM)             VEC1
      AN(1,1,4;NXM)=(AN(1,1,4;NXM)+SIGY(1,1;NXM))*V(1,1;NXM)+U(1,1;NXM) VEC1
     1*TAUXY(1,1;NXM)+QY(1,1;NXM)                                      VEC1
      DO 20 I=1,4                                                      VEC1
      AN(1,1,I;NXM)=YETA(1,1;NXM)* AU(1,1,I;NXM)-XETA(1,1;NXM)*AN(1,1,I;VEC1
     1NXM)                                                             VEC1
      AN(1,1,I;NXM)=-YXI(1,1;NXM)* AU(1,1,I;NXM)+XXI(1,1;NXM)*AN(1,1,I; VEC1
     1NXM)                                                             VEC1
20    CONTINUE                                                         VEC1
      AU(1,1,1;NXM)=RO(1,1;NXM)*AJ(1,1;NXM)                            VEC1
      AU(1,1,2;NXM)=AU(1,1,1;NXM)*U(1,1;NXM)                           VEC1
      AU(1,1,3;NXM)=AJ(1,1,1;NXM)*V(1,1;NXM)                           VEC1
      AU(1,1,4;NXM)=(RO(1,1;NXM)*H(1,1;NXM)-P(1,1;NXM))*AJ(1,1;NXM)    VEC1
      IF(NSTRUT.EQ.0)RETURN                                            VEC1
      DO 30 N=1,NS                                                     VEC1
      ANS(N,4)=ROS(N)*HS(N)-PS(N)                                      VEC1
      AUS(N,1)=ROS(N)*US(N)                                            VEC1
      AUS(N,2)=AUS(N,1)*US(N)+SIGXS(N)                                 VEC1
      AUS(N,3)=AUS(N,1)*VS(N)+TAUXYS(N)                                VEC1
      AUS(N,4)=(ANS(N,4)+SIGXS(N))*US(N)+VS(N)*TAUXYS(N)+QXS(N)        VEC1
      ANS(N,1)=ROS(N)*VS(N)                                            VEC1
      ANS(N,2)=ANS(N,1)*US(N)+TAUXYS(N)                                VEC1
      ANS(N,3)=ANS(N,1)*VS(N)+SIGYS(N)                                 VEC1
      ANS(N,4)=(ANS(N,4)+SIGYS(N))*VS(N)+US(N)*TAUXYS(N)+QYS(N)        VEC1
30    CONTINUE                                                         VEC1
      DO 40 I=1,4                                                      VEC1
      DO 40 N=1,NS                                                     VEC1
      ANS(N,I)=YSETA(N)*AUS(N,1)-XSETA(N)*ANS(N,I)                     VEC1
      ANS(N,I)=-YSXI(N)*AUS(N,I)+XSXI(N)*ANS(N,I)                      VEC1
40    CONTINUE                                                         VEC1
      DO 50 N=1,NS                                                     VEC1
      AUS(N,1)=ROS(N)*AJS(N)                                           VEC1
      AUS(N,2)=AUS(N,1)*US(N)                                          VEC1
      AUS(N,3)=AUS(N,1)*VS(N)                                          VEC1
      AUS(N,4)=(POS(N)*HS(N)-PS(N))*AJS(N)                             VEC1
50    CONTINUE                                                         VEC1
      RETURN                                                           VEC1
      END                                                              VEC1
      SUBROUTINE IVEC1                                                 IVEC1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 IVEC1
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                             IVEC1
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),         IVEC1
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                       IVEC1
     2 PS(55),TS(55),SHS(55),HS(55)                                    IVEC1
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)       IVEC1
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)               IVEC1
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),               IVEC1
     1 XETA(61,55),YETA(61,55),AJ(61,55)                               IVEC1
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),             IVEC1
     1 YSETA(55),AJS(55)                                               IVEC1
      AU(1,1,1;NXM)=RO(1,1;NXM)*U(1,1;NXM)                             IVEC1
       AU(1,1,2;NXM)=AU(1,1,1;NXM)*U(1,1;NXM)+P(1,1;NXM)               IVEC1
       AU(1,1,3;NXM)=AU(1,1,1;NXM)*V(1,1;NXM)                          IVEC1
       AU(1,1,4;NXM)=AU(1,1,1;NXM)*H(1,1;NXM)                          IVEC1
      AN(1,1,1;NXM)=RO(1,1;NXM)*V(1,1;NXM)                             IVEC1
      AN(1,1,2;NXM)=AN(1,1,1;NXM)*U(1,1;NXM)                           IVEC1
      AN(1,1,3;NXM)=AN(1,1,1;NXM)*V(1,1;NXM)+P(1,1;NXM)                IVEC1
      AN(1,1,4;NXM)=AN(1,1,1;NXM)*H(1,1;NXM)                           IVEC1
      DO 20 I=1,4                                                      IVEC1
      AN(1,1,I;NXM)=YETA(1,1;NXM)* AU(1,1,I;NXM)-XETA(1,1;NXM)*AN(1,1,I;IVEC1
     1NXM)                                                             IVEC1
      AN(1,1,I;NXM)=-YXI(1,1;NXM)* AU(1,1,I;NXM)+XXI(1,1;NXM)*AN(1,1,I; IVEC1
     1NXM)                                                             IVEC1
20    CONTINUE                                                         IVEC1
```

```
      AU(1,1,1;NXM)=RO(1,1;NXM)*AJ(1,1;NXM)                              IVEC1     27
      AU(1,1,2;NXM)=AU(1,1,1;NXM)*U(1,1;NXM)                             IVEC1     28
      AU(1,1,3;NXM)=AU(1,1,1;NXM)*V(1,1;NXM)                             IVEC1     29
      AU(1,1,4;NXM)=(RO(1,1;NXM)*H(1,1;NXM)-P(1,1;NXM))*AJ(1,1;NXM)      IVEC1     30
      IF(NSTRUT.EQ.0)RETURN                                              IVEC1     31
      DO 30 N=1,NS                                                       IVEC1     32
      AUS(N,1)=ROS(N)*US(N)                                              IVEC1     33
      AUS(N,2)=AUS(N,1)*US(N)+PS(N)                                      IVEC1     34
      AUS(N,3)=AUS(N,1)*VS(N)                                            IVEC1     35
      AUS(N,4)=AUS(N,1)*HS(N)                                            IVEC1     36
      ANS(N,1)=ROS(N)*VS(N)                                              IVEC1     37
      ANS(N,2)=ANS(N,1)*US(N)                                            IVEC1     38
      ANS(N,3)=ANS(N,1)*VS(N)+PS(N)                                      IVEC1     39
      ANS(N,4)=ANS(N,1)*HS(N)                                            IVEC1     40
30    CONTINUE                                                           IVEC1     41
      DO 40 I=1,4                                                        IVEC1     42
      DO 40 N=1,NS                                                       IVEC1     43
      AMS(N,I)=YSETA(N)*AUS(N,I)-XSETA(N)*ANS(N,I)                       IVEC1     44
      ANS(N,I)=-YSXI(N)*AUS(N,I)+XSXI(N)*ANS(N,I)                        IVEC1     45
40    CONTINUE                                                           IVEC1     46
      DO 50 N=1,NS                                                       IVEC1     47
      AUS(N,1)=ROS(N)*AJS(N)                                             IVEC1     48
      AUS(N,2)=AUS(N,1)*US(N)                                            IVEC1     49
      AUS(N,3)=AUS(N,1)*VS(N)                                            IVEC1     50
      AUS(N,4)=(ROS(N)*HS(N)-PS(N))*AJS(N)                               IVEC1     51
50    CONTINUE                                                           IVEC1     52
      RETURN                                                             IVEC1     53
      END                                                                IVEC1     54
      SUBROUTINE DAMPP                                                   DAMPP      1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 DAMPP      2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                               DAMPP      3
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   DAMPP      4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                       DAMPP      5
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),           DAMPP      6
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                         DAMPP      7
     2 PS(55),TS(55),SHS(55),HS(55)                                      DAMPP      8
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),          DAMPP      9
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)          DAMPP     10
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)         DAMPP     11
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                 DAMPP     12
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                 DAMPP     13
      COMMON/F10/DT(61,55),DTM                                           DAMPP     14
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,     DAMPP     15
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                   DAMPP     16
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF         DAMPP     17
      QX(1,1;NXM)=GAMA*RGAS*T(1,1;NXM)                                   DAMPP     18
      QX(1,1;NXM)=VSQRT(QX(1,1;NXM);QX(1,1;NXM))                         DAMPP     19
      SIGX(1,1;NXM)=(VABS(U(1,1;NXM);SIGY(1,1;NXM))+QX(1,1;NXM))         DAMPP     20
      SIGY(2,2;NXM4)=P(2,3;NXM4)-2.*P(2,2;NXM4)+P(2,1;NXM4)              DAMPP     21
      SIGY(2,2;NXM4)=VABS(SIGY(2,2;NXM4);SIGY(2,2;NXM4))                 DAMPP     22
        QY(2,2;NXM4)=T(2,3;NXM4)-2.*T(2,2;NXM4)+T(2,1;NXM4)              DAMPP     23
      QY(2,2;NXM4)=VABS(QY(2,2;NXM4);QY(2,2;NXM4))                       DAMPP     24
      SIGX(2,2;NXM4)=SIGX(2,2;NXM4)*(SIGY(2,2;NXM4)*CCP/                 DAMPP     25
     1(P(2,3;NXM4)+2.*P(2,2;NXM4)+P(2,1;NXM4))+QY(2,2;NXM4)*CCT/         DAMPP     26
     1(T(2,3;NXM4)+2.*T(2,2;NXM4)+T(2,1;NXM4)))                          DAMPP     27
      QX(1,1;NXM)=(VABS(V(1,1;NXM);SIGY(1,1;NXM))+QX(1,1;NXM))           DAMPP     28
      SIGY(2,2;NXM4)=P(3,2;NXM4)-2.*P(2,2;NXM4)+P(1,2;NXM4)              DAMPP     29
      SIGY(2,2;NXM4)=VABS(SIGY(2,2;NXM4);SIGY(2,2;NXM4))                 DAMPP     30
        QY(2,2;NXM4)=T(3,2;NXM4)-2.*T(2,2;NXM4)+T(1,2;NXM4)              DAMPP     31
      QY(2,2;NXM4)=VABS(QY(2,2;NXM4);QY(2,2;NXM4))                       DAMPP     32
      SIGY(2,2;NXM4)=              QX(2,2;NXM4)*(SIGY(2,2;NXM4)*CCP/     DAMPP     33
     1(P(3,2;NXM4)+2.*P(2,2;NXM4)+P(1,2;NXM4))+QY(2,2;NXM4)*CCT/(        DAMPP     34
     2 T(3,2;NXM4)+2.*T(2,2;NXM4)+T(1,2;NXM4)))                          DAMPP     35
      IF(NSTRUT.EQ.0)GO TO 35                                            DAMPP     36
      DO 30 N=NS1,NS2                                                    DAMPP     37
      NN=N-NS1+1                                                         DAMPP     38
      SIGY(MS1,N)=QX(MS1,N)*(ABS(PS(NN)-2.*P(MS1,N)+P(MS2,N))*CCP/(PS(NNDAMPP     39
     1)+2.*P(MS1,N)+P(MS2,N))+ABS(TS(NN)-2.*T(MS1,N)+T(MS2,N))*CCT/(TS( DAMPP     40
```

```
     2NN)+2.*T(MS1,N)+T(MS2,N)))                                        DAMPP
30    CONTINUE                                                          DAMPP
35    CONTINUE                                                          DAMPP
      DO 100 I=1,4                                                      DAMPP
      QX(2,2;NXM4)=SIGX(2,2;NXM4)*(AU(2,2,I;NXM4)-AU(2,1,I;NXM4))       DAMPP
      QY(2,2;NXM4)=SIGY(2,2;NXM4)*(AU(2,2,I;NXM4)-AU(1,2,I;NXM4))       DAMPP
      DO 10 N=2,N11                                                     DAMPP
      QX(1,N)=0.                                                        DAMPP
      QX(M1,N)=0.                                                       DAMPP
      QY(1,N)=0.                                                        DAMPP
10    QY(M1,N)=0.                                                       DAMPP
      IF(NSTRUT.EQ.0)GO TO 25                                           DAMPP
      DO 20 N=NS1,NS2                                                   DAMPP
      QX(MS,N)=0.                                                       DAMPP
20    QY(MS,N)=0.                                                       DAMPP
25    CONTINUE                                                          DAMPP
      AM(2,2,I;NXM4)=AM(2,2,I;NXM4)-QX(2,2;NXM4)/XXI(2,2;NXM4)          DAMPP
      AN(2,2,I;NXM4)=AN(2,2,I;NXM4)-QY(2,2;NXM4)/YETA(2,2;NXM4)         DAMPP
100   CONTINUE                                                          DAMPP
      RETURN                                                            DAMPP
      END                                                               DAMPP
      SUBROUTINE DAMPC                                                  DAMPC
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 DAMPC
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              DAMPC
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   DAMPC
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      DAMPC
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          DAMPC
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        DAMPC
     2 PS(55),TS(55),SHS(55),HS(55)                                     DAMPC
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),         DAMPC
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)         DAMPC
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)        DAMPC
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                DAMPC
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                DAMPC
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                DAMPC
      COMMON/F10/DT(61,55),DTM                                          DAMPC
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,    DAMPC
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                  DAMPC
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF        DAMPC
      QX(1,1;NXM)=GAMA*RGAS*T(1,1;NXM)                                  DAMPC
      QX(1,1;NXM)=VSQRT(QX(1,1;NXM);QX(1,1;NXM))                        DAMPC
      SIGX(1,1;NXM)=(VABS(U(1,1;NXM);SIGY(1,1;NXM))+QX(1,1;NXM))        DAMPC
      SIGY(2,2;NXM4)=P(2,3;NXM4)-2.*P(2,2;NXM4)+P(2,1;NXM4)             DAMPC
      SIGY(2,2;NXM4)=VABS(SIGY(2,2;NXM4);SIGY(2,2;NXM4))                DAMPC
        QY(2,2;NXM4)=T(2,3;NXM4)-2.*T(2,2;NXM4)+T(2,1;NXM4)             DAMPC
      QY(2,2;NXM4)=VABS(QY(2,2;NXM4);QY(2,2;NXM4))                      DAMPC
      SIGX(2,2;NXM4)=SIGX(2,2;NXM4)*(SIGY(2,2;NXM4)*CCP/                DAMPC
     1(P(2,3;NXM4)+2.*P(2,2;NXM4)+P(2,1;NXM4))+QY(2,2;NXM4)*CCT/        DAMPC
     1(T(2,3;NXM4)+2.*T(2,2;NXM4)+T(2,1;NXM4)))                         DAMPC
      QX(1,1;NXM)=(VABS(V(1,1;NXM);SIGY(1,1;NXM))+QX(1,1;NXM))          DAMPC
      SIGY(2,2;NXM4)=P(3,2;NXM4)-2.*P(2,2;NXM4)+P(1,2;NXM4)             DAMPC
      SIGY(2,2;NXM4)=VABS(SIGY(2,2;NXM4);SIGY(2,2;NXM4))                DAMPC
        QY(2,2;NXM4)=T(3,2;NXM4)-2.*T(2,2;NXM4)+T(1,2;NXM4)             DAMPC
      QY(2,2;NXM4)=VABS(QY(2,2;NXM4);QY(2,2;NXM4))                      DAMPC
      SIGY(2,2;NXM4)=              QX(2,2;NXM4)*(SIGY(2,2;NXM4)*CCP/    DAMPC
     1(P(3,2;NXM4)+2.*P(2,2;NXM4)+P(1,2;NXM4))+QY(2,2;NXM4)*CCT/(       DAMPC
     2 T(3,2;NXM4)+2.*T(2,2;NXM4)+T(1,2;NXM4)))                         DAMPC
      IF(NSTRUT.EQ.0)GO TO 35                                           DAMPC
      DO 30 N=NS1,NS2                                                   DAMPC
      NN=N-NS1+1                                                        DAMPC
      SIGY(MS1,N)=QX(MS1,N)*(ABS(PS(NN)-2.*P(MS1,N)+P(MS2,N))*CCP/(PS(NNDAMPC
     1)+2.*P(MS1,N)+P(MS2,N))+ABS(TS(NN)-2.*T(MS1,N)+T(MS2,N))*CCT/(TS( DAMPC
     2NN)+2.*T(MS1,N)+T(MS2,N)))                                        DAMPC
30    CONTINUE                                                          DAMPC
35    CONTINUE                                                          DAMPC
      DO 100 I=1,4                                                      DAMPC
      QX(2,2;NXM4)=SIGX(2,2;NXM4)*(AU(2,3,I;NXM4)-AU(2,2,I;NXM4))       DAMPC
      QY(2,2;NXM4)=SIGY(2,2;NXM4)*(AU(3,2,I;NXM4)-AU(2,2,I;NXM4))       DAMPC
```

```
      DO 10 N=2,N11                                                     DAMPC      48
      QX(1,N)=0.                                                        DAMPC      49
      QX(M1,N)=0.                                                       DAMPC      50
      QY(1,N)=0.                                                        DAMPC      51
10    QY(M1,N)=0.                                                       DAMPC      52
      IF(NSTRUT.EQ.0)GO TO 25                                           DAMPC      53
      DO 20 N=NS1,NS2                                                   DAMPC      54
      NN=N-NS1+1                                                        DAMPC      55
      QY(MS1,N)=SIGY(MS1,N)*(AUS(NN,I)-AU(MS1,N,I))                     DAMPC      56
      QX(MS,N)=0.                                                       DAMPC      57
20    QY(MS,N)=0.                                                       DAMPC      58
25    CONTINUE                                                          DAMPC      59
      AM(2,2,I;NXM4)=AM(2,2,I;NXM4)-QX(2,2;NXM4)/XXI(2,2;NXM4)          DAMPC      60
      AN(2,2,I;NXM4)=AN(2,2,I;NXM4)-QY(2,2;NXM4)/YETA(2,2;NXM4)         DAMPC      61
100   CONTINUE                                                          DAMPC      62
      RETURN                                                            DAMPC      63
      END                                                               DAMPC      64
      SUBROUTINE VEC2                                                   VEC2        1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 VEC2        2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              VEC2        3
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          VEC2        4
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        VEC2        5
     2 PS(55),TS(55),SHS(55),HS(55)                                     VEC2        6
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),         VEC2        7
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)         VEC2        8
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)        VEC2        9
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                VEC2       10
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                VEC2       11
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF        VEC2       12
      COMMON/F14/DAU(61,55,4),BDAU(55,4),BDAUS(55,4),TI(55,5),          VEC2       13
     1CCIMPY,GM1,GM2,GM3,GM4,GM5                                        VEC2       14
      RO(2,2;NXM4)=AU1(2,2,1;NXM4)/AJ(2,2;NXM4)                         VEC2       15
       U(2,2;NXM4)=AU1(2,2,2;NXM4)/AU1(2,2,1;NXM4)                      VEC2       16
       V(2,2;NXM4)=AU1(2,2,3;NXM4)/AU1(2,2,1;NXM4)                      VEC2       17
      QX(2,2;NXM4)=(U(2,2;NXM4)*U(2,2;NXM4)+V(2,2;NXM4)*V(2,2;NXM4))/2. VEC2       18
      SH(2,2;NXM4)=(AU1(2,2,4;NXM4)/AU1(2,2,1;NXM4)-QX(2,2;NXM4))*GAMA  VEC2       19
      H(2,2;NXM4)=SH(2,2;NXM4)+QX(2,2;NXM4)                             VEC2       20
      P(2,2;NXM4)=GM2*RO(2,2;NXM4)*SH(2,2;NXM4)                         VEC2       21
      T(2,2;NXM4)=SH(2,2;NXM4)/CP                                       VEC2       22
      RETURN                                                            VEC2       23
      END                                                               VEC2       24
      SUBROUTINE BOUND                                                  BOUND       1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 BOUND       2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              BOUND       3
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   BOUND       4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      BOUND       5
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          BOUND       6
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        BOUND       7
     2 PS(55),TS(55),SHS(55),HS(55)                                     BOUND       8
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)             BOUND       9
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                BOUND      10
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                BOUND      11
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),              BOUND      12
     1 YSETA(55),AJS(55)                                                BOUND      13
      COMMON/F9/INT(55),TE1(55),TE2(55),TE3(55),TE4(55)                 BOUND      14
      COMMON/F10/DT(61,55),DTM                                          BOUND      15
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,    BOUND      16
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                  BOUND      17
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF        BOUND      18
      TE1(2;N12)=Q8VGATHR(T(M11,2;NXM3),INT(1;N12);TE1(2;N12))          BOUND      19
      TE2(2;N12)=Q8VGATHR(P(M11,2;NXM3),INT(1;N12);TE2(2;N12))          BOUND      20
      TE3(2;N12)=TE2(2;N12)/RGAS/TE1(2;N12)                             BOUND      21
      TE4(2;N12)=CP*TE1(2;N12)                                          BOUND      22
       T(M1,2;NXM3)=Q8VSCATR(TE1(2;N12),INT(1;N12); T(M1,2;NXM3))       BOUND      23
      P(M1,2;NXM3)=Q8VSCATR(TE2(2,N12),INT(1;N12);P(M1,2;NXM3))         BOUND      24
      RO(M1,2;NXM3)=Q8VSCATR(TE3(2;N12),INT(1;N12);RO(M1,2;NXM3))       BOUND      25
      SH(M1,2;NXM3)=Q8VSCATR(TE4(2;N12),INT(1;N12);SH(M1,2;NXM3))       BOUND      26
      TE1(2;N12)=0.                                                     BOUND      27
```

```
      TE2(2;N12)=0.                                                     BOUND      28
      U(M1,2;NXM3)=Q8VSCATR(TE1(2;N12),INT(1;N12);U(M1,2;NXM3))         BOUND      29
      V(M1,2;NXM3)=Q8VSCATR(TE2(2;N12),INT(1;N12);V(M1,2;NXM3))         BOUND      30
      TE3(2;N12)=TE4(2;N12)+(TE1(2;N12)*TE1(2;N12)+TE2(2;N12)*TE2(2;N12)BOUND      31
     1)/2.                                                              BOUND      32
      H(M1,2;NXM3)=Q8VSCATR(TE3(2;N12),INT(1;N12);H(M1,2;NXM3))         BOUND      33
      IF(NSTRUT.EQ.0)GO TO 15                                           BOUND      34
      DO 10 N=NS1,NS2                                                   BOUND      35
      U(MS,N)=0.                                                        BOUND      36
      V(MS,N)=0.                                                        BOUND      37
      P(MS,N)=P(MSP1,N)                                                 BOUND      38
      T(MS,N)=T(MSP1,N)                                                 BOUND      39
      SH(MS,N)=CP*T(MS,N)                                               BOUND      40
      RO(MS,N)=P(MS,N)/RGAS/T(MS,N)                                     BOUND      41
 10   H(MS,N)=SH(MSP1,N)                                                BOUND      42
      DO 20 N=1,NS                                                      BOUND      43
      NN=N+NS1M1                                                        BOUND      44
      US(N)=0.                                                          BOUND      45
      VS(N)=0.                                                          BOUND      46
      PS(N)=P(MS1,NN)                                                   BOUND      47
      TS(N)=T(MS1,NN)                                                   BOUND      48
      SHS(N)=CP*TS(N)                                                   BOUND      49
      ROS(N)=PS(N)/RGAS/TS(N)                                           BOUND      50
 20   HS(N)=SHS(N)                                                      BOUND      51
 15   CONTINUE                                                          BOUND      52
      DO 25 N=2,N11                                                     BOUND      53
      T(1,N)=T(2,N)                                                     BOUND      54
      P(1,N)=P(2,N)                                                     BOUND      55
      SH(1,N)=CP*T(1,N)                                                 BOUND      56
      RO(1,N)=RO(2,N)                                                   BOUND      57
      U(1,N)=U(2,N)                                                     BOUND      58
      V(1,N)=V(2,N)                                                     BOUND      59
 25   H(1,N)=H(2,N)                                                     BOUND      60
      IF(LSYM.EQ.0)GO TO 55                                             BOUND      61
      DO 60 N=2,N11                                                     BOUND      62
      V(1,N)=0.                                                         BOUND      63
 60   H(1,N)=SH(1,N)+U(1,N)*U(1,N)/2.                                   BOUND      64
 55   CONTINUE                                                          BOUND      65
C     COWL PLATE OR CENTER STRUT STARTS FROM N=NCWL1 AND ENDS AT NCWL2. BOUND      66
      IF(NCWL.EQ.0)GO TO 35                                             BOUND      67
      DO 30 N=NCWL1,NCWL2                                               BOUND      68
      H(1,N)=SH(1,N)                                                    BOUND      69
      U(1,N)=0.                                                         BOUND      70
 30   V(1,N)=0.                                                         BOUND      71
 35   CONTINUE                                                          BOUND      72
      IF(LEXIT.EQ.0)GO TO 36                                            BOUND      73
      U(1,N1;M1)=2.*U(1,N11;M1)-U(1,N12;M1)                             BOUND      74
      V(1,N1;M1)=2.*V(1,N11;M1)-V(1,N12;M1)                             BOUND      75
       P(1,N1;M1)=2.* P(1,N11;M1)- P(1,N12;M1)                          BOUND      76
      SH(1,N1;M1)=2.*SH(1,N11;M1)-SH(1,N12;M1)                          BOUND      77
      GO TO 37                                                          BOUND      78
 36   CONTINUE                                                          BOUND      79
      U(1,N1;M1)=U(1,N11;M1)                                            BOUND      80
      V(1,N1;M1)=V(1,N11;M1)                                            BOUND      81
      P(1,N1;M1)=P(1,N11;M1)                                            BOUND      82
      SH(1,N1;M1)=SH(1,N11;M1)                                          BOUND      83
 37   CONTINUE                                                          BOUND      84
      T(1,N1;M1)=SH(1,N1;M1)/CP                                         BOUND      85
      RO(1,N1;M1)=P(1,N1;M1)/(RGAS*T(1,N1;M1))                          BOUND      86
      H(1,N1;M1)=SH(1,N1;M1)+(U(1,N1;M1)*U(1,N1;M1)+V(1,N1;M1)*         BOUND      87
     1V(1,N1;M1))/2.                                                    BOUND      88
      II=Q8SLT(T(1,1;NXM),0.5)                                          BOUND      89
      IF(II.EQ.NXM)GO TO 50                                             BOUND      90
      WRITE(6,500)L                                                     BOUND      91
 500  FORMAT(/,2X,'NEGATIVE TEMPERATURE IN THE FIELD',15X,              BOUND      92
     1        'NO. OF ITERATIONS=',I5)                                  BOUND      93
      DO 40 N=1,N1                                                      BOUND      94
      WRITE(6,550)N,X(1,N)                                              BOUND      95
```

```
      DO 40 MM=1,M1                                                     BOUND      96
      M=M1-MM+1                                                         BOUND      97
      WRITE(6,560)Y(M,N),U(M,N),V(M,N),P(M,N),T(M,N),H(M,N),VIST(M,N)   BOUND      98
40    CONTINUE                                                          BOUND      99
      IF(NSTRUT.EQ.0)GO TO 45                                           BOUND     100
      WRITE(6,570)                                                      BOUND     101
      WRITE(6,560)(YS(N),US(N),VS(N),PS(N),TS(N),HS(N),VISS(N),N=1,NS)  BOUND     102
45    CONTINUE                                                          BOUND     103
      LK=2                                                              BOUND     104
      RETURN                                                            BOUND     105
50    CONTINUE                                                          BOUND     106
550   FORMAT(/,8X,'Y',10X,'U',10X,'V',11X,'P',11X,'T',12X,' H',10X,     BOUND     107
     1'VIST',10X,'BODY STATION=',I3,5X,'X=',F10.5)                      BOUND     108
560   FORMAT(2X,F12.5,F12.5,F12.5,F13.4,F12.5,2F15.5)                   BOUND     109
570   FORMAT(/,8X,'YS',9X,'US',9X,'VS',10X,'PS',10X,'TS',12X,'HS',      BOUND     110
     110X,'VISS',/)                                                     BOUND     111
      VISL(1,1;NXM)=VSQRT(T(1,1;NXM);VISL(1,1;NXM))                     BOUND     112
      VISL(1,1;NXM)=CVIS*VISL(1,1;NXM)*T(1,1;NXM)/(T(1,1;NXM)+110.)     BOUND     113
      IF(NSTRUT.EQ.0)RETURN                                             BOUND     114
      VISS(1;NS)=VSQRT(TS(1;NS);VISS(1;NS))                             BOUND     115
      VISS(1;NS)=CVIS*VISS(1;NS)*TS(1;NS)/(TS(1;NS)+110.)               BOUND     116
      RETURN                                                            BOUND     117
      END                                                               BOUND     118
      SUBROUTINE IBOUND                                                 IBOUND      1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 IBOUND      2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              IBOUND      3
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   IBOUND      4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      IBOUND      5
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          IBOUND      6
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        IBOUND      7
     2 PS(55),TS(55),SHS(55),HS(55)                                     IBOUND      8
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),         IBOUND      9
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)         IBOUND     10
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)        IBOUND     11
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                IBOUND     12
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                IBOUND     13
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                IBOUND     14
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),              IBOUND     15
     1 YSETA(55),AJS(55)                                                IBOUND     16
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,    IBOUND     17
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                  IBOUND     18
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF        IBOUND     19
      DO 5 N=2,N11                                                      IBOUND     20
      RO(1,N)=RO(2,N)                                                   IBOUND     21
       P(1,N)= P(2,N)                                                   IBOUND     22
       T(1,N)= T(2,N)                                                   IBOUND     23
       H(1,N)= H(2,N)                                                   IBOUND     24
      SH(1,N)=SH(2,N)                                                   IBOUND     25
      CON1=YXI(1,N)/XXI(1,N)                                            IBOUND     26
      CON2=1.+CON1*CON1                                                 IBOUND     27
      U(1,N)=(H(1,N)-SH(1,N))*2./CON2                                   IBOUND     28
      U(1,N)=SQRT(U(1,N))                                               IBOUND     29
 5    V(1,N)=CON1*U(1,N)                                                IBOUND     30
      DO 20 N=2,N11                                                     IBOUND     31
      RO(M1,N)=RO(M11,N)                                                IBOUND     32
      SH(M1,N)=SH(M11,N)                                                IBOUND     33
       P(M1,N)= P(M11,N)                                                IBOUND     34
       T(M1,N)= T(M11,N)                                                IBOUND     35
       H(M1,N)= H(M11,N)                                                IBOUND     36
      CON1=YXI(M1,N)/XXI(M1,N)                                          IBOUND     37
      CON2=1.+CON1*CON1                                                 IBOUND     38
      U(M1,N)=(H(M1,N)-SH(M1,N))*2./CON2                                IBOUND     39
      U(M1,N)=SQRT(U(M1,N))                                             IBOUND     40
20    V(M1,N)=CON1*U(M1,N)                                              IBOUND     41
      IF(LSYM.EQ.1)GO TO 60                                             IBOUND     42
      IF(NCWL.EQ.0)GO TO 62                                             IBOUND     43
      IF(NCWL1.LE.2.AND.NCWL2.GE.N11)GO TO 60                           IBOUND     44
      DO 50 N=1,NCWLM                                                   IBOUND     45
```

```
      U(1,N)=U(2,N)                                                     IBOUND      46
50    V(1,N)=V(2,N)                                                     IBOUND      47
      IF(NCWL2.GE.N11)GO TO 60                                          IBOUND      48
      DO 55 N=NCWLP,N11                                                 IBOUND      49
      U(1,N)=U(2,N)                                                     IBOUND      50
55    V(1,N)=V(2,N)                                                     IBOUND      51
      GO TO 60                                                          IBOUND      52
62    CONTINUE                                                          IBOUND      53
      DO 56 N=2,N11                                                     IBOUND      54
      U(1,N)=U(2,N)                                                     IBOUND      55
 56   V(1,N)=V(2,N)                                                     IBOUND      56
60    CONTINUE                                                          IBOUND      57
      IF(NSTRUT.EQ.0)GO TO 6                                            IBOUND      58
      DO 10 N=1,NS                                                      IBOUND      59
      NN=N+NS1-1                                                        IBOUND      60
      RO(MS,NN)=RO(MSP1,NN)                                             IBOUND      61
      SH(MS,NN)=SH(MSP1,NN)                                             IBOUND      62
       P(MS,NN)= P(MSP1,NN)                                             IBOUND      63
       T(MS,NN)= T(MSP1,NN)                                             IBOUND      64
       H(MS,NN)= H(MSP1,NN)                                             IBOUND      65
      CON1=YXI(MS,NN)/XXI(MS,NN)                                        IBOUND      66
      CON2=1.+CON1*CON1                                                 IBOUND      67
      U(MS,NN)=(H(MS,NN)-SH(MS,NN))*2./CON2                             IBOUND      68
      U(MS,NN)=SQRT(U(MS,NN))                                           IBOUND      69
      V(MS,NN)=CON1*U(MS,NN)                                            IBOUND      70
      ROS(N)=RO(MS1,NN)                                                 IBOUND      71
       PS(N)= P(MS1,NN)                                                 IBOUND      72
       TS(N)= T(MS1,NN)                                                 IBOUND      73
       HS(N)= H(MS1,NN)                                                 IBOUND      74
      SHS(N)=SH(MS1,NN)                                                 IBOUND      75
      CON1=YSXI(N)/XSXI(N)                                              IBOUND      76
      CON2=1.+CON1*CON1                                                 IBOUND      77
      US(N)=(HS(N)-SHS(N))*2./CON2                                      IBOUND      78
      US(N)=SQRT(US(N))                                                 IBOUND      79
 10   VS(N)=CON1*US(N)                                                  IBOUND      80
6     CONTINUE                                                          IBOUND      81
      IF(LEXIT.EQ.0) GO TO 36                                           IBOUND      82
      U(1,N1;M1)=2.*U(1,N11;M1)-U(1,N12;M1)                             IBOUND      83
      V(1,N1;M1)=2.*V(1,N11;M1)-V(1,N12;M1)                             IBOUND      84
      SH(1,N1;M1)=2.*SH(1,N11;M1)-SH(1,N12;M1)                          IBOUND      85
       P(1,N1;M1)=2.* P(1,N11;M1)- P(1,N12;M1)                          IBOUND      86
      GO TO 37                                                          IBOUND      87
 36   CONTINUE                                                          IBOUND      88
      U(1,N1;M1)=U(1,N11;M1)                                            IBOUND      89
      V(1,N1;M1)=V(1,N11;M1)                                            IBOUND      90
      SH(1,N1;M1)=SH(1,N11;M1)                                          IBOUND      91
      P(1,N1;M1)=P(1,N11;M1)                                            IBOUND      92
 37   CONTINUE                                                          IBOUND      93
      T(1,N1;M1)=SH(1,N1;M1)/CP                                         IBOUND      94
      RO(1,N1;M1)=P(1,N1;M1)/(RGAS*T(1,N1;M1))                          IBOUND      95
      H(1,N1;M1)=SH(1,N1;M1)+(U(1,N1;M1)*U(1,N1;M1)+V(1,N1;M1)*         IBOUND      96
     1V(1,N1;M1))/2.                                                    IBOUND      97
      I1=Q8SLT(T(1,1;NXM),0.5)                                          IBOUND      98
      IF(II.EQ.NXM)RETURN                                               IBOUND      99
      WRITE(6,500)L                                                     IBOUND     100
500   FORMAT(/,2X,'NEGATIVE TEMPERATURE IN THE FIELD',15X,              IBOUND     101
     1            'NO. OF ITERATIONS =',I5)                             IBOUND     102
      DO 40 N=1,N1                                                      IBOUND     103
      WRITE(6,550)N,X(1,N)                                              IBOUND     104
      DO 40 MM=1,M1                                                     IBOUND     105
      M=M1-MM+1                                                         IBOUND     106
      WRITE(6,560)Y(M,N),U(M,N),V(M,N),P(M,N),T(M,N),H(M,N)             IBOUND     107
40    CONTINUE                                                          IBOUND     108
      IF(NSTRUT.EQ.0)GO TO 45                                           IBOUND     109
      WRITE(6,570)                                                      IBOUND     110
      WRITE(6,560)(YS(N),US(N),VS(N),PS(N),TS(N),HS(N),N=1,NS)          IBOUND     111
45    CONTINUE                                                          IBOUND     112
550   FORMAT(/,8X,'Y',10X,'U',10X,'V',11X,'P',11X,'T',12X,' H',10X,     IBOUND     113
```

```
     1'BODY STATION=',I3,5X,'X=',F10.5)                                  IBOUND     114
 560  FORMAT(2X,F12.5,F12.5,F12.5,F13.4,F12.5,F15.5)                     IBOUND     115
 570  FORMAT(/,8X,'YS',9X,'US',9X,'VS',10X,'PS',10X,'TS',12X,'HS',/)     IBOUND     116
      LK=2                                                               IBOUND     117
      RETURN                                                             IBOUND     118
      END                                                                IBOUND     119
      SUBROUTINE VORT                                                    VORT         1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 VORT         2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                               VORT         3
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3    VORT         4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                       VORT         5
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),           VORT         6
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                         VORT         7
     2 PS(55),TS(55),SHS(55),HS(55)                                      VORT         8
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),          VORT         9
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)          VORT        10
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                 VORT        11
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                 VORT        12
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),               VORT        13
     1 YSETA(55),AJS(55)                                                 VORT        14
      SIGX(2,1;NXM2)=(U(3,1;NXM2)-U(1,1;NXM2))/2.                        VORT        15
      SIGY(2,1;NXM2)=(V(3,1;NXM2)-V(1,1;NXM2))/2.                        VORT        16
      DO 10 N=1,N1                                                       VORT        17
      SIGX(1,N)=U(2,N)-U(1,N)                                            VORT        18
      SIGY(1,N)=V(2,N)-V(1,N)                                            VORT        19
      SIGY(1,N)=V(2,N)-V(1,N)                                            VORT        19
      SIGX(M1,N)=U(M1,N)-U(M11,N)                                        VORT        20
 10   SIGY(M1,N)=V(M1,N)-V(M11,N)                                        VORT        21
      QX(1,2;NXM3)=(U(1,3;NXM3)-U(1,1;NXM3))/2.                          VORT        22
      QY(1,2;NXM3)=(V(1,3;NXM3)-V(1,1;NXM3))/2.                          VORT        23
      DO 20 M=1,M1                                                       VORT        24
      QX(M,1)=U(M,2)-U(M,1)                                              VORT        25
      QY(M,1)=V(M,2)-V(M,1)                                              VORT        26
      QX(M,N1)=U(M,N1)-U(M,N11)                                          VORT        27
 20   QY(M,N1)=V(M,N1)-V(M,N11)                                          VORT        28
      IF(NSTRUT.EQ.0)GO TO 50                                            VORT        29
      DO 30 N=NS1,NS2                                                    VORT        30
      NN=N-NS1+1                                                         VORT        31
      SIGX(MS,N)=U(MSP1,N)-J(MS,N)                                       VORT        32
      SIGY(MS,N)=V(MSP1,N)-V(MS,N)                                       VORT        33
      SIGX(MS1,N)=(US(NN)-U(MS2,N))/2.                                   VORT        34
 30   SIGY(MS1,N)=(VS(NN)-V(MS2,N))/2.                                   VORT        35
      DO 40 N=2,NSM1                                                     VORT        36
 40     QXS(N)=(US(N+1)-US(N-1))/2.                                     VORT        37
        QXS(1)=(US(2)-U(MS,NS1M1))/2.                                   VORT        38
      QXS(NS)=(U(MS,NS2P1)-US(NSM1))/2.                                  VORT        39
      DO 60 N=1,NS                                                       VORT        40
      NN=N-1+NS1                                                         VORT        41
 60   SIGXS(N)=US(N)-U(MS1,NN)                                           VORT        42
      QXS(1;NS)=(XSXI(1;NS)*SIGXS(1;NS)-XSETA(1;NS)*  QXS(1;NS))/AJS(1; VORT        43
     1NS)                                                                VORT        44
 50   CONTINUE                                                           VORT        45
      QX(1,1;NXM)=(XXI(1,1;NXM)*SIGX(1,1;NXM)-XETA(1,1;NXM)*  QX(1,1;    VORT        46
     1NXM))/AJ(1,1;NXM)                                                  VORT        47
      QY(1,1;NXM)=(YETA(1,1;NXM)*  QY(1,1;NXM)-YXI(1,1;NXM)*SIGY(1,1;    VORT        48
     1NXM))/AJ(1,1;NXM)                                                  VORT        49
      TAUXY(1,1;NXM)=QX(1,1;NXM)-QY(1,1;NXM)                             VORT        50
      TAUXY(1,1;NXM)=VABS(TAUXY(1,1;NXM);TAUXY(1,1;NXM))                 VORT        51
      RETURN                                                             VORT        52
      END                                                                VORT        53
      SUBROUTINE EDDY(NNX1,NNX2,NY1,NY2,NWALL,NSTR)                      EDDY         1
C     EDDY VISCOSITY MODEL TAKEN FROM AIAA PAPER NO. 78-257 BY BALDWIN   EDDY         2
C     AND LOMAX.  NWALL = 1 FOR ONE WALL AND = 2 FOR TWO WALLS.          EDDY         3
      DIMENSION YD(61),VISTI(61),VISTO(61),FY(61),UDIF1(61)              EDDY         4
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 EDDY         5
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                               EDDY         6
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),           EDDY         7
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                         EDDY         8
```

```
     2 PS(55),TS(55),SHS(55),HS(55)                                    EDDY      9
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)            EDDY     10
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),        EDDY     11
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)        EDDY     12
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),               EDDY     13
     1 XETA(61,55),YETA(61,55),AJ(61,55)                               EDDY     14
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),             EDDY     15
     1 YSETA(55),AJS(55)                                               EDDY     16
      REAL K,KK                                                        EDDY     17
      DATA AP,CCP,CKLEB,CWK,K,KK/26.,1.6,0.3,0.25,0.4,.0168/           EDDY     18
C     NSTR EQUAL TO 1  FOR THE REGION WHERE LOWER SURFACE OF THE STRUT EDDY     19
C     FORMS THE UPPER WALL OF THE DOMAIN IN WHICH EDDY VISCOSITY IS    EDDY     20
C     BEING CALCULATED.   IT IS ZERO OTHERWISE.                        EDDY     21
      NNY=NY2-NY1+1                                                    EDDY     22
      NNY1=NY1                                                         EDDY     23
      NNY2=NY2                                                         EDDY     24
      IF(NWALL.EQ.1)GO TO 5                                            EDDY     25
      NNY2=NY2                                                         EDDY     26
      NNY1=NY1+NNY/2                                                   EDDY     27
 5    CONTINUE                                                         EDDY     28
      NNY2M1=NNY2-1                                                    EDDY     29
      NNY1P1=NNY1+1                                                    EDDY     30
      NNY=NNY2-NNY1+1                                                  EDDY     31
      DO 100 N=NNX1,NNX2                                               EDDY     32
      IF(NSTR.EQ.1)GO TO 10                                            EDDY     33
      TAUW=ABS(VIS(NNY2,N)*QX(NNY2,N))*RO(NNY2,N)                      EDDY     34
      YPI=SQRT(TAUW)/VIS(NNY2,N)                                       EDDY     35
      DO 20 M=NNY1,NNY2                                                EDDY     36
 20   YD(M)=Y(NNY2,N)-Y(M,N)                                           EDDY     37
      GO TO 15                                                         EDDY     38
 10   NN=N-NS1+1                                                       EDDY     39
      TAUW=ABS(VISS(NN)*QXS(NN))*ROS(NN)                               EDDY     40
      YPI=SQRT(TAUW)/VISS(NN)                                          EDDY     41
      DO 30 M=NNY1,NNY2                                                EDDY     42
 30   YD(M)=YS(NN)-Y(M,N)                                              EDDY     43
 15   CONTINUE                                                         EDDY     44
      DO 40 M=NNY1,NNY2                                                EDDY     45
      YP=YPI*YD(M)                                                     EDDY     46
      RL=-YP/AP                                                        EDDY     47
      RL=EXP(RL)                                                       EDDY     48
      RL=YD(M)*(1.-RL)                                                 EDDY     49
      VISTI(M)=RO(M,N)*K*K*RL*RL*TAUXY(M,N)                            EDDY     50
      UDIF1(M)=U(M,N)*U(M,N)+V(M,N)*V(M,N)                             EDDY     51
      UDIF1(M)=SQRT(UDIF1(M))                                          EDDY     52
 40   FY(M)=RL*TAUXY(M,N)                                              EDDY     53
C                                                                      EDDY     54
C     SOMETIMES CALCULATIONS OF FYMAX MAY REQUIRE SOME ATTENTION DUE   EDDY     55
C     TO MORE THAN ONE MAXIMA OCCURING IN FUNCTION FY.                 EDDY     56
C                                                                      EDDY     57
      FYMAX=FY(NNY2M1)                                                 EDDY     58
       YMAX=YD(NNY2M1)                                                 EDDY     59
      DO 50 MM=NNY1P1,NNY2M1                                           EDDY     60
      M=NNY2-MM+NNY1                                                   EDDY     61
      IF(FY(M-1).GT.FY(M))GO TO 50                                     EDDY     62
      FYMAX=FY(M)                                                      EDDY     63
      YMAX=YD(M)                                                       EDDY     64
      GO TO 51                                                         EDDY     65
 50   CONTINUE                                                         EDDY     66
      FYMAX=FY(NNY1P1)                                                 EDDY     67
       YMAX=YD(NNY1P1)                                                 EDDY     68
 51   CONTINUE                                                         EDDY     69
      UDIF=QBSMAX(UDIF1(NNY1;NNY))                                     EDDY     70
      FWAKE1=YMAX*FYMAX                                                EDDY     71
      FWAKE2=CWK*YMAX*UDIF*UDIF/FYMAX                                  EDDY     72
      FWAKE=AMIN1(FWAKE1,FWAKE2)                                       EDDY     73
      DO 60 M=NNY1,NNY2                                                EDDY     74
      FKLEB=1./(1.+5.5*(CKLEB*YD(M)/YMAX)**6)                          EDDY     75
 60   VISTO(M)=KK*CCP*RO(M,N)*FWAKE*FKLEB                              EDDY     76
```

```
      IOUT=0                                                            EDDY
      DO 70 MM=NNY1,NNY2                                                EDDY
      M=NNY2-MM+NNY1                                                    EDDY
      DVIST=VISTO(M)-VISTI(M)                                           EDDY
      IF(DVIST.LE.0.)IOUT=1                                             EDDY
      VIST(M,N)=VISTI(M)                                                EDDY
      IF(IOUT.EQ.1)VIST(M,N)=VISTO(M)                                   EDDY
 70   CONTINUE                                                          EDDY
 100   CONTINUE                                                         EDDY
      IF(NWALL.EQ.1)GO TO 500                                           EDDY
      NNY2=NNY1-1                                                       EDDY
      NNY1=NY1                                                          EDDY
      NNY=NNY2-NNY1+1                                                   EDDY
      NNY1P1=NNY1+1                                                     EDDY
      NNY1P2=NNY1+2                                                     EDDY
      DO 200 N=NNX1,NNX2                                                EDDY
      TAUW=ABS(VIS(NNY1,N)*QX(NNY1,N))*RO(NNY1,N)                       EDDY
      YPI=SQRT(TAUW)/VIS(NNY1,N)                                        EDDY
      DO 210 M=NNY1,NNY2                                                EDDY
      YD(M)=Y(M,N)-Y(NNY1,N)                                            EDDY
      YP=YPI*YD(M)                                                      EDDY
      RL=-YP/AP                                                         EDDY
      RL=EXP(PL)                                                        EDDY
      RL=YD(M)*(1.-RL)                                                  EDDY
      VISTI(M)=RO(M,N)*K*K*RL*RL*TAUXY(M,N)                             EDDY
      UDIF1(M)=U(M,N)*U(M,N)+V(M,N)*V(M,N)                              EDDY
      UDIF1(M)=SQRT(UDIF1(M))                                           EDDY
 210  FY(M)=RL*TAUXY(M,N)                                               EDDY
C                                                                       EDDY
C     SOMETIMES CALCULATIONS OF FYMAX MAY REQUIRE SOME ATTENTION DUE    EDDY
C     TO MORE THAN ONE MAXIMA OCCURING IN FUNCTION FY.                  EDDY
C                                                                       EDDY
      FYMAX=FY(NNY1P1)                                                  EDDY
       YMAX=YD(NNY1P1)                                                  EDDY
      DO 220 M=NNY1P2,NNY2                                              EDDY
      IF(FY(M).GT.FY(M-1))GO TO 220                                     EDDY
      FYMAX=FY(M-1)                                                     EDDY
      YMAX=YD(M-1)                                                      EDDY
      GO TO 221                                                         EDDY
 220  CONTINUE                                                          EDDY
      FYMAX=FY(NNY2)                                                    EDDY
       YMAX=YD(NNY2)                                                    EDDY
 221  CONTINUE                                                          EDDY
      UDIF=QBSMAX(UDIF1(NNY1;NNY))                                      EDDY
      FWAKE1=YMAX*FYMAX                                                 EDDY
      FWAKE2=CWK*YMAX*UDIF*UDIF/FYMAX                                   EDDY
      FWAKE=AMIN1(FWAKE1,FWAKE2)                                        EDDY
      DO 230 M=NNY1,NNY2                                                EDDY
      FKLEB=1./(1.+5.5*(CKLEB*YD(M)/YMAX)**6)                           EDDY
 230  VISTO(M)=KK*CCP*RO(M,N)*FWAKE*FKLEB                               EDDY
      IOUT=0                                                            EDDY
      DO 240 M=NNY1,NNY2                                                EDDY
      DVIST=VISTO(M)-VISTI(M)                                           EDDY
      IF(DVIST.LE.0.)IOUT=1                                             EDDY
      VIST(M,N)=VISTI(M)                                                EDDY
      IF(IOUT.EQ.1)VIST(M,N)=VISTO(M)                                   EDDY
 240  CONTINUE                                                          EDDY
 200  CONTINUE                                                          EDDY
 500  CONTINUE                                                          EDDY
      RETURN                                                            EDDY
      END                                                               EDDY
      SUBROUTINE IMPY(IADD)                                             IMPY
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 IMPY
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              IMPY
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   IMPY
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      IMPY
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          IMPY
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        IMPY
```

```
     2 PS(55),TS(55),SHS(55),HS(55)                                     IMPY      8
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)             IMPY      9
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),         IMPY     10
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)         IMPY     11
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)        IMPY     12
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                IMPY     13
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                IMPY     14
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                IMPY     15
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),              IMPY     16
     1 YSETA(55),AJS(55)                                                IMPY     17
      COMMON/F9/INT(55),TE1(55),TE2(55),TE3(55),TE4(55)                 IMPY     18
      COMMON/F10/DT(61,55),DTM                                          IMPY     19
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,    IMPY     20
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                  IMPY     21
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF        IMPY     22
      COMMON/F14/DAU(61,55,4),BDAU(55,4),BDAUS(55,4),TI(55,5),          IMPY     23
     1CCIMPY,GM1,GM2,GM3,GM4,GM5                                        IMPY     24
      QX(1,1;NXM)=GM5*T(1,1;NXM)                                        IMPY     25
      QX(1,1;NXM)=VSQRT(QX(1,1;NXM);QX(1,1;NXM))                        IMPY     26
      SIGX(1,1;NXM)=.5*(U(1,1;NXM)*U(1,1;NXM)+V(1,1;NXM)*V(1,1;NXM))    IMPY     27
      QY(1,1;NXM)=DTM/YETA(1,1;NXM)                                     IMPY     28
      TAUXY(1,1;NXM)=GM4*VIS(1,1;NXM)*QY(1,1;NXM)/(RO(1,1;NXM)*         IMPY     29
     1YETA(1,1;NXM))                                                    IMPY     30
      SIGY(1,1;NXM)=VABS(V(1,1;NXM);SIGY(1,1;NXM))                      IMPY     31
      AM(1,1,1;NXM)=SIGY(1,1;NXM)*QY(1,1;NXM)-.5                        IMPY     32
      SIGY(1,1;NXM)=(SIGY(1,1;NXM)+QX(1,1;NXM))*QY(1,1;NXM)-.5          IMPY     33
      AM(1,1,2;NXM)=V(1,1;NXM)+QX(1,1;NXM)                              IMPY     34
      AM(1,1,2;NXM)=(VABS(AM(1,1,2;NXM);AM(1,1,2;NXM)))*QY(1,1;NXM)-.5  IMPY     35
      AM(1,1,3;NXM)=V(1,1;NXM)-QX(1,1;NXM)                              IMPY     36
      AM(1,1,3;NXM)= VABS(AM(1,1,3;NXM);AM(1,1,3;NXM))*QY(1,1;NXM)-.5   IMPY     37
      NIM=0                                                             IMPY     38
      NN1=2                                                             IMPY     39
      NN2=NS1M1                                                         IMPY     40
      IF(NSTRUT.EQ.0)NN2=N11                                            IMPY     41
15    CONTINUE                                                          IMPY     42
      NIM=NIM+1                                                         IMPY     43
      DO 10 N=NN1,NN2                                                   IMPY     44
      DO 10 M=2,M11                                                     IMPY     45
      MM=(1-IADD)*M+IADD*(M11+2-M)                                      IMPY     46
      IF(SIGY(MM,N).LT.0.)GO TO 10                                      IMPY     47
      VV=V(MM,N)                                                        IMPY     48
      CC=QX(MM,N)                                                       IMPY     49
      ALP=SIGX(MM,N)                                                    IMPY     50
      RCCSQ=1./(CC*CC)                                                  IMPY     51
      UU=U(MM,N)                                                        IMPY     52
      RR=RO(MM,N)                                                       IMPY     53
      IF(IADD.EQ.0)GO TO 30                                             IMPY     54
      IF(MM.NE.M11)GO TO 20                                             IMPY     55
      DO 110 I=1,4                                                      IMPY     56
110   DAU(M11,N,I)=DAU(M11,N,I)+BDAU(N,I)*QY(M1,N)                      IMPY     57
      GO TO 20                                                          IMPY     58
30    CONTINUE                                                          IMPY     59
      IF(MM.NE.2)GO TO 20                                               IMPY     60
      DO 230 I=1,4                                                      IMPY     61
230   DAU(2,N,I)=DAU(2,N,I)+BDAU(N,I)*QY(1,N)                           IMPY     62
20    CONTINUE                                                          IMPY     63
      DP=(ALP*DAU(MM,N,1)-UU*DAU(MM,N,2)-VV*DAU(MM,N,3)+DAU(MM,N,4))*GM1IMPY     64
      XX1=DAU(MM,N,1)-DP*RCCSQ                                          IMPY     65
      XX2=(DAU(MM,N,2)-UU*DAU(MM,N,1))/RR                               IMPY     66
      XX3=-VV*CC*DAU(MM,N,1)+CC*DAU(MM,N,3)+DP                          IMPY     67
      XX4=-XX3+2.*DP                                                    IMPY     68
      VIS1=TAUXY(MM,N)+ABS(XX1)/(GM1*RR/GAMA)*CCIMPY                    IMPY     69
      DD1=AMAX1((VIS1+AM(MM,N,1)),0.)                                   IMPY     70
      DD3=AMAX1((VIS1+AM(MM,N,2)),0.)                                   IMPY     71
      DD4=AMAX1((VIS1+AM(MM,N,3)),0.)                                   IMPY     72
      YY1=XX1/(1.+DD1)                                                  IMPY     73
      YY2=XX2/(1.+DD1)                                                  IMPY     74
      YY3=XX3/(1.+DD3)                                                  IMPY     75
```

```
      YY4=XX4/(1.+DD4)                                                  IMPY      76
      DR=YY1+.5*(YY3+YY4)*RCCSQ                                         IMPY      77
      DAU(MM,N,1)=DR                                                    IMPY      78
      DAU(MM,N,2)=UU*DR+RR*YY2                                          IMPY      79
      DAU(MM,N,3)=VV*DR+.5*(YY3-YY4)/CC                                 IMPY      80
      DAU(MM,N,4)=ALP*DR+RR*UU*YY2+.5*(VV*(YY3-YY4)/CC+(YY3+YY4)/GM1)   IMPY      81
      IF(M.EQ.M11)GO TO 10                                              IMPY      82
      MJ=MM-2*IADD+1                                                    IMPY      83
      DD=YETA(MM,N)/YETA(MJ,N)                                          IMPY      84
      ZZ1=DD*DD1*YY1                                                    IMPY      85
      ZZ2=DD*DD1*YY2                                                    IMPY      86
      ZZ3=DD*DD3*YY3                                                    IMPY      87
      ZZ4=DD*DD4*YY4                                                    IMPY      88
      DR=ZZ1+.5*(ZZ3+ZZ4)*RCCSQ                                         IMPY      89
      DAU(MJ,N,1)=DAU(MJ,N,1)+DR                                        IMPY      90
      DAU(MJ,N,2)=DAU(MJ,N,2)+UU*DR+ZZ2*RR                              IMPY      91
      DAU(MJ,N,3)=DAU(MJ,N,3)+VV*DR+.5*(ZZ3-ZZ4)/CC                     IMPY      92
      DAU(MJ,N,4)=DAU(MJ,N,4)+ALP*DR+RR*UU*ZZ2+.5*(VV*(ZZ3-ZZ4)/CC+    IMPY      93
     1(ZZ3+ZZ4)/GM1)                                                    IMPY      94
10    CONTINUE                                                          IMPY      95
      IF(NSTRUT.EQ.0)RETURN                                             IMPY      96
      NN1=NS2P1                                                         IMPY      97
      NN2=N11                                                           IMPY      98
      IF(N1M.EQ.1)GO TO 15                                              IMPY      99
      DO 5 N=NS1,NS2                                                    IMPY     100
      NN=N-NS1+1                                                        IMPY     101
 5    QYS(N)=DTM/YSETA(NN)                                              IMPY     102
      DO 40 N=NS1,NS2                                                   IMPY     103
      DO 45 M=2,MS1                                                     IMPY     104
      MM=(1-IADD)*M+IADD*(MS1+2-M)                                      IMPY     105
      IF(SIGY(MM,N).LT.0.)GO TO 45                                      IMPY     106
      VV=V(MM,N)                                                        IMPY     107
      CC=QX(MM,N)                                                       IMPY     108
      ALP=SIGX(MM,N)                                                    IMPY     109
      RCCSQ=1./(CC*CC)                                                  IMPY     110
      UU=U(MM,N)                                                        IMPY     111
      RR=RO(MM,N)                                                       IMPY     112
      IF(IADD.EQ.0)GO TO 65                                             IMPY     113
      IF(MM.NE.MS1)GO TO 60                                             IMPY     114
      DO 70 I=1,4                                                       IMPY     115
 70   DAU(MS1,N,I)=DAU(MS1,N,I)+BDAUS(N,I)*QYS(N)                       IMPY     116
      GO TO 60                                                          IMPY     117
 65   CONTINUE                                                          IMPY     118
      IF(MM.NE.2)GO TO 60                                               IMPY     119
      DO 75 I=1,4                                                       IMPY     120
 75   DAU(2,N,I)=DAU(2,N,I)+BDAU(N,I)*QY(1,N)                           IMPY     121
 60   CONTINUE                                                          IMPY     122
      DP=(ALP*DAU(MM,N,1)-UU*DAU(MM,N,2)-VV*DAU(MM,N,3)+DAU(MM,N,4))*GM1IMPY     123
      XX1=DAU(MM,N,1)-DP*RCCSQ                                          IMPY     124
      XX2=(DAU(MM,N,2)-UU*DAU(MM,N,1))/RR                               IMPY     125
      XX3=-VV*CC*DAU(MM,N,1)+CC*DAU(MM,N,3)+DP                          IMPY     126
      XX4=-XX3+2.*DP                                                    IMPY     127
      VIS1=TAUXY(MM,N)+ABS(XX1)/(GM1*RR/GAMA)*CCIMPY                    IMPY     128
      DD1=AMAX1((VIS1+AM(MM,N,1)),0.)                                   IMPY     129
      DD3=AMAX1((VIS1+AM(MM,N,2)),0.)                                   IMPY     130
      DD4=AMAX1((VIS1+AM(MM,N,3)),0.)                                   IMPY     131
      YY1=XX1/(1.+DD1)                                                  IMPY     132
      YY2=XX2/(1.+DD1)                                                  IMPY     133
      YY3=XX3/(1.+DD3)                                                  IMPY     134
      YY4=XX4/(1.+DD4)                                                  IMPY     135
      DR=YY1+.5*(YY3+YY4)*RCCSQ                                         IMPY     136
      DAU(MM,N,1)=DR                                                    IMPY     137
      DAU(MM,N,2)=UU*DR+RR*YY2                                          IMPY     138
      DAU(MM,N,3)=VV*DR+.5*(YY3-YY4)/CC                                 IMPY     139
      DAU(MM,N,4)=ALP*DR+RR*UU*YY2+.5*(VV*(YY3-YY4)/CC+(YY3+YY4)/GM1)   IMPY     140
      IF(M.EQ.MS1)GO TO 45                                              IMPY     141
      MJ=MM-2*IADD+1                                                    IMPY     142
      DD=YETA(MM,N)/YETA(MJ,N)                                          IMPY     143
```

```
      ZZ1=DD*DD1*YY1                                                    IMPY      144
      ZZ2=DD*DD1*YY2                                                    IMPY      145
      ZZ3=DD*DD3*YY3                                                    IMPY      146
      ZZ4=DD*DD4*YY4                                                    IMPY      147
      DR=ZZ1+.5*(ZZ3+ZZ4)*RCCSQ                                         IMPY      148
      DAU(MJ,N,1)=DAU(MJ,N,1)+DR                                        IMPY      149
      DAU(MJ,N,2)=DAU(MJ,N,2)+UU*DR+ZZ2*RR                              IMPY      150
      DAU(MJ,N,3)=DAU(MJ,N,3)+VV*DR+.5*(ZZ3-ZZ4)/CC                     IMPY      151
      DAU(MJ,N,4)=DAU(MJ,N,4)+ALP*DR+RR*UU*ZZ2+.5*(VV*(ZZ3-ZZ4)/CC+    IMPY      152
     1(ZZ3+ZZ4)/GM1)                                                    IMPY      153
45    CONTINUE                                                          IMPY      154
      DO 50 M=MSP1,M11                                                  IMPY      155
      MM=(1-IADD)*M+IADD*(M11+MSP1-M)                                   IMPY      156
      IF(SIGY(MM,N).LT.0.)GO TO 50                                      IMPY      157
      VV=V(MM,N)                                                        IMPY      158
      CC=QX(MM,N)                                                       IMPY      159
      ALP=SIGX(MM,N)                                                    IMPY      160
      RCCSQ=1./(CC*CC)                                                  IMPY      161
      UU=U(MM,N)                                                        IMPY      162
      RR=RO(MM,N)                                                       IMPY      163
      IF(IADD.EQ.0)GO TO 85                                             IMPY      164
      IF(MM.NE.M11)GO TO 80                                             IMPY      165
      DO 90 I=1,4                                                       IMPY      166
90    DAU(M11,N,I)=DAU(M11,N,I)+BDAU(N,I)*QY(M1,N)                      IMPY      167
      GO TO 80                                                          IMPY      168
85    CONTINUE                                                          IMPY      169
      IF(MM.NE.MSP1)GO TO 80                                            IMPY      170
      DO 95 I=1,4                                                       IMPY      171
95    DAU(MSP1,N,I)=DAU(MSP1,N,I)+BDAUS(N,I)*QY(MS,N)                   IMPY      172
80    CONTINUE                                                          IMPY      173
      DP=(ALP*DAU(MM,N,1)-UU*DAU(MM,N,2)-VV*DAU(MM,N,3)+DAU(MM,N,4))*GM1IMPY      174
      XX1=DAU(MM,N,1)-DP*RCCSQ                                          IMPY      175
      XX2=(DAU(MM,N,2)-UU*DAU(MM,N,1))/RR                               IMPY      176
      XX3=-VV*CC*DAU(MM,N,1)+CC*DAU(MM,N,3)+DP                          IMPY      177
      XX4=-XX3+2.*DP                                                    IMPY      178
      VIS1=TAUXY(MM,N)+ABS(XX1)/(GM1*RR/GAMA)*CCIMPY                    IMPY      179
      DD1=AMAX1((VIS1+AM(MM,N,1)),0.)                                   IMPY      180
      DD3=AMAX1((VIS1+AM(MM,N,2)),0.)                                   IMPY      181
      DD4=AMAX1((VIS1+AM(MM,N,3)),0.)                                   IMPY      182
      YY1=XX1/(1.+DD1)                                                  IMPY      183
      YY2=XX2/(1.+DD1)                                                  IMPY      184
      YY3=XX3/(1.+DD3)                                                  IMPY      185
      YY4=XX4/(1.+DD4)                                                  IMPY      186
      DR=YY1+.5*(YY3+YY4)*RCCSQ                                         IMPY      187
      DAU(MM,N,1)=DR                                                    IMPY      188
      DAU(MM,N,2)=UU*DR+RR*YY2                                          IMPY      189
      DAU(MM,N,3)=VV*DR+.5*(YY3-YY4)/CC                                 IMPY      190
      DAU(MM,N,4)=ALP*DR+RR*UU*YY2+.5*(VV*(YY3-YY4)/CC+(YY3+YY4)/GM1)   IMPY      191
      IF(M.EQ.M11)GO TO 50                                              IMPY      192
      MJ=MM-2*IADD+1                                                    IMPY      193
      DD=YETA(MM,N)/YETA(MJ,N)                                          IMPY      194
      ZZ1=DD*DD1*YY1                                                    IMPY      195
      ZZ2=DD*DD1*YY2                                                    IMPY      196
      ZZ3=DD*DD3*YY3                                                    IMPY      197
      ZZ4=DD*DD4*YY4                                                    IMPY      198
      DR=ZZ1+.5*(ZZ3+ZZ4)*RCCSQ                                         IMPY      199
      DAU(MJ,N,1)=DAU(MJ,N,1)+DR                                        IMPY      200
      DAU(MJ,N,2)=DAU(MJ,N,2)+UU*DR+ZZ2*RR                              IMPY      201
      DAU(MJ,N,3)=DAU(MJ,N,3)+VV*DR+.5*(ZZ3-ZZ4)/CC                     IMPY      202
      DAU(MJ,N,4)=DAU(MJ,N,4)+ALP*DR+RR*UU*ZZ2+.5*(VV*(ZZ3-ZZ4)/CC+    IMPY      203
     1(ZZ3+ZZ4)/GM1)                                                    IMPY      204
50    CONTINUE                                                          IMPY      205
40    CONTINUE                                                          IMPY      206
      RETURN                                                            IMPY      207
      END                                                               IMPY      208
      SUBROUTINE VECIM2(IADD)                                           VECIM2      1
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 VECIM2      2
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              VECIM2      3
```

```
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   VECIM2      4
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      VECIM2      5
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          VECIM2      6
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        VECIM2      7
     2 PS(55),TS(55),SHS(55),HS(55)                                     VECIM2      8
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)             VECIM2      9
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),         VECIM2     10
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)         VECIM2     11
      COMMON/F5/AU(61,55,4),AM(61,55,4),AN(61,55,4),AU1(61,55,4)        VECIM2     12
      COMMON/F6/AUS(55,4),AMS(55,4),ANS(55,4),AUS1(55,4)                VECIM2     13
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                VECIM2     14
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                VECIM2     15
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),              VECIM2     16
     1 YSETA(55),AJS(55)                                                VECIM2     17
      COMMON/F9/INT(55),TE1(55),TE2(55),TE3(55),TE4(55)                 VECIM2     18
      COMMON/F10/DT(61,55),DTM                                          VECIM2     19
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,    VECIM2     20
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                  VECIM2     21
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF        VECIM2     22
      COMMON/F14/DAU(61,55,4),BDAU(55,4),BDAUS(55,4),TI(55,5),          VECIM2     23
     1CCIMPY,GM1,GM2,GM3,GM4,GM5                                        VECIM2     24
C     *************************                                        VECIM2     25
      IF(IADD.EQ.1)GO TO 10                                             VECIM2     26
      K=M1                                                              VECIM2     27
      GO TO 20                                                          VECIM2     28
   10 K=1                                                               VECIM2     29
   20 CONTINUE                                                          VECIM2     30
      TI(1,1;N1)=Q8VGATHP(RO(K,1;NXM),M1,N1;TI(1,1;N1))                 VECIM2     31
      TI(1,2;N1)=Q8VGATHP(U(K,1;NXM),M1,N1;TI(1,2;N11))                 VECIM2     32
      TI(1,3;N1)=Q8VGATHP(V(K,1;NXM),M1,N1;TI(1,3;N1))                  VECIM2     33
      TI(1,4;N1)=Q8VGATHP(T(K,1;NXM),M1,N1;TI(1,4;N1))                  VECIM2     34
      TI(1,5;N1)=Q8VGATHP(VIS(K,1;NXM),M1,N1;TI(1,5;N1))                VECIM2     35
      IF(NSTRUT.EQ.0)GO TO 30                                           VECIM2     36
C                                                                       VECIM2     37
C     THE SUBROUTINE ASSUMES U=V=0 CONDITIONS ON THE                    VECIM2     38
C     STRUT SURFACE.                                                    VECIM2     39
C                                                                       VECIM2     40
      IF(IADD.EQ.1)GO TO 15                                             VECIM2     41
      DO 16 N=NS1,NS2                                                   VECIM2     42
      NN=N-NS1+1                                                        VECIM2     43
      ANS(N,1)=ROS(NN)                                                  VECIM2     44
      ANS(N,2)=TS(NN)                                                   VECIM2     45
   16 ANS(N,3)=VISS(NN)                                                 VECIM2     46
      GO TO 30                                                          VECIM2     47
   15 CONTINUE                                                          VECIM2     48
      DO 17 N=NS1,NS2                                                   VECIM2     49
      ANS(N,1)=RO(MS,N)                                                 VECIM2     50
      ANS(N,2)=T(MS,N)                                                  VECIM2     51
   17 ANS(N,3)=VIS(MS,N)                                                VECIM2     52
   30 CONTINUE                                                          VECIM2     53
C     *************************                                        VECIM2     54
      RO(2,2;NXM4)=AU1(2,2,1;NXM4)/AJ(2,2;NXM4)                         VECIM2     55
       U(2,2;NXM4)=AU1(2,2,2;NXM4)/AU1(2,2,1;NXM4)                      VECIM2     56
       V(2,2;NXM4)=AU1(2,2,3;NXM4)/AU1(2,2,1;NXM4)                      VECIM2     57
      QX(2,2;NXM4)=(U(2,2;NXM4)*U(2,2;NXM4)+V(2,2;NXM4)*V(2,2;NXM4))/2. VECIM2     58
      SH(2,2;NXM4)=(AU1(2,2,4;NXM4)/AU1(2,2,1;NXM4)-QX(2,2;NXM4))*GAMA  VECIM2     59
      H(2,2;NXM4)=SH(2,2;NXM4)+QX(2,2;NXM4)                             VECIM2     60
      P(2,2;NXM4)=GM2*RO(2,2;NXM4)*SH(2,2;NXM4)                         VECIM2     61
      T(2,2;NXM4)=SH(2,2;NXM4)/CP                                       VECIM2     62
C     *************************                                        VECIM2     63
      CALL BOUND                                                        VECIM2     64
C     *************************                                        VECIM2     65
      DO 40 N=2,N11                                                     VECIM2     66
      ALP=.5*(TI(N,2)*TI(N,2)+TI(N,3)*TI(N,3))                          VECIM2     67
      CC=SQRT(GM5*TI(N,4))                                              VECIM2     68
      AMS(N,1)=RO(K,N)-TI(N,1)                                          VECIM2     69
      AMS(N,2)=RO(K,N)*U(K,N)-TI(N,1)*TI(N,2)                           VECIM2     70
      AMS(N,3)=RO(K,N)*V(K,N)-TI(N,1)*TI(N,3)                           VECIM2     71
```

```
      AMS(N,4)=RO(K,N)*(GM3*T(K,N)+.5*(U(K,N)*U(K,N)+V(K,N)*V(K,N)))-
     1TI(N,1)*(GM3*TI(N,4)+ALP)
      DP=GM1*(ALP*AMS(N,1)-TI(N,2)*AMS(N,2)-TI(N,3)*AMS(N,3)+AMS(N,4))
      RCCSQ=1./(CC*CC)
      XX1=AMS(N,1)-DP*RCCSQ
      XX2=(AMS(N,2)-TI(N,2)*AMS(N,1))/TI(N,1)
      XX3=CC*(AMS(N,3)-TI(N,3)*AMS(N,1))+DP
      XX4=-XX3+2.*DP
      VIS1=GM4*TI(N,5)/(TI(N,1)*YETA(K,N))-.5*YETA(K,N)/DTM
      VABS1=ABS(TI(N,3))+VIS1
      VPCABS=ABS(TI(N,3)+CC)+VIS1
      VMCABS=ABS(TI(N,3)-CC)+VIS1
      DD1=AMAX1(VABS1,0.)
      DD3=AMAX1(VPCABS,0.)
      DD4=AMAX1(VMCABS,0.)
      YY1=DD1*XX1
      YY2=DD1*XX2
      YY3=DD3*XX3
      YY4=DD4*XX4
      AY1=(YY3+YY4)/2.
      AY2=(YY3-YY4)/(2.*CC)
      BDAU(N,1)=YY1+AY1*RCCSQ
      BDAU(N,2)=TI(N,2)*BDAU(N,1)+TI(N,1)*YY2
      BDAU(N,3)=TI(N,3)*BDAU(N,1)+AY2
      BDAU(N,4)=ALP*BDAU(N,1)+TI(N,1)*TI(N,2)*YY2+TI(N,3)*AY2+AY1/GM1
 40   CONTINUE
      IF(NSTRUT.EQ.0)RETURN
      IF(IADD.EQ.1)GO TO 55
      DO 50 N=NS1,NS2
      NN=N-NS1+1
      TE1(N)=ROS(NN)
      TE2(N)=TS(NN)
 50   TE3(N)=YSETA(NN)
      GO TO 60
 55   CONTINUE
      DO 56 N=NS1,NS2
      TE1(N)=RO(MS,N)
      TE2(N)=T(MS,N)
 56   TE3(N)=YETA(MS,N)
 60   CONTINUE
      DO 65 N=NS1,NS2
      BDAUS(N,1)=TE1(N)-ANS(N,1)
      BDAUS(N,4)=GM3*(TE1(N)*TE2(N)-ANS(N,1)*ANS(N,2))
      VIS1=GM4*ANS(N,3)/(ANS(N,1)*TE3(N))-.5*TE3(N)/DTM
      TE4(N)=SQRT(GM5*ANS(N,2))
      TI(N,1)=AMAX1(VIS1,0.)
      TI(N,2)=AMAX1(VIS1+TE4(N),0.)
      BDAUS(N,1)=TI(N,1)*BDAUS(N,1)-GM1/(TE4(N)*TE4(N))*(TI(N,1)-TI(N,2)
     1)*BDAUS(N,4)
      BDAUS(N,4)=TI(N,2)*BDAUS(N,4)
 65   CONTINUE
C     ************************
      RETURN
      END
      SUBROUTINE SPILL(SWEEP)
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),
     2 PS(55),TS(55),SHS(55),HS(55)
      COMMON/F4/SIGX(61,55),SIGY(61,55),TAUXY(61,55),QX(61,55),
     1 QY(61,55),SIGXS(55),SIGYS(55),TAUXYS(55),QXS(55),QYS(55)
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),
     1 XETA(61,55),YETA(61,55),AJ(61,55)
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),
     1 YSETA(55),AJS(55)
```

```
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF
      DIMENSION XP(61,55),USP(61,55),UST1(55),USP1(55)
      DIMENSION FMASS(55),UST(61,55),FMAST(55)
      PI=4.*ATAN(1.)
      THESW=SWEEP*PI/180.
      CTHESW=COS(THESW)
      STHESW=SIN(THESW)
      FMS= FM/CTHESW
      XP(1,1;NXM)=X(1,1;NXM)/CTHESW
      VT=FMS*SQRT(GAMA*RGAS*TF)*STHESW*CTHESW
      UST(1,1;NXM)=VT
      UST1(1;NS)=VT
      IF(NFLOW.EQ.0)GO TO 35
      QX(1,1;NXM)=VABS(U(1,1;NXM);QX(1,1;NXM))
C
C     IN M112=M1-12, 12 IS AN ARBITRARY NUMBER. WHAT IT MEANS IS THAT
C     FIRST 12 POINTS LIE IN THE B.L.PROFILE.  THIS IS REQUIRED TO PUT
C     VISCOUS EFFECTS ON THE INVISCID COMPONENT VT.  NUMBER 12 MAY
C     NEED TO BE CHANGED FOR CERTAIN CALCULATIONS.
C
      M112=M1-12
      DO 30 N=1,N1
      DO 30 M=M112,M1
 30   UST(M,N)=VT*QX(M,N)/QX(M112,N)
      IF(NSTRUT.EQ.0)GO TO 45
      QXS(1;NS)=VABS(US(1;NS);QXS(1;NS))
      MS11=MS+11
      DO 40 N=NS1,NS2
      DO 40 M=MS,MS11
 40   UST(M,N)=VT*QX(M,N)/QX(MS11,N)
      MS10=MS-10
      DO 50 N=NS1,NS2
      NN=N-NS1+1
      DO 50 M=MS10,MS1
      QYS(NN)=QX(MS10,N)
 50   UST(M,N)=VT*QX(M,N)/QX(MS10,N)
      UST1(1;NS)=VT*QXS(1;NS)/QYS(1;NS)
      USP1(1;NS)=UST1(1;NS)-US(1;NS)*STHESW
 45   CONTINUE
      IF(NCWL.EQ.0)GO TO 35
      DO 55 N=NCWL1,NCWL2
      DO 55 M=1,12
 55   UST(M,N)=VT*QX(M,N)/QX(12,N)
 35   CONTINUE
      USP(1,1;NXM)=UST(1,1;NXM)-U(1,1;NXM)*STHESW
      NSM=NS1M1
      IF(NSTRUT.EQ.0)NSM=N1
      FMASS(1;N1)=0.
      DO 275 N=1,NSM
      DO 275 M=2,M1
      FMASS(N)=FMASS(N)+(RO(M-1,N)*USP(M-1,N)+RO(M,N)*USP(M,N))*(Y(M,N)-
     1Y(M-1,N))/2.
 275  CONTINUE
      IF(NSTRUT.EQ.0)GO TO 290
      DO 280 N=NS2P1,N1
      DO 280 M=2,M1
      FMASS(N)=FMASS(N)+(RO(M-1,N)*USP(M-1,N)+RO(M,N)*USP(M,N))*(Y(M,N)-
     1Y(M-1,N))/2.
 280  CONTINUE
      DO 250 N=NS1,NS2
      DO 255 M=2,MS1
      FMASS(N)=FMASS(N)+(RO(M-1,N)*USP(M-1,N)+RO(M,N)*USP(M,N))*(Y(M,N)-
     1Y(M-1,N))/2.
 255  CONTINUE
      DO 260 M=MSP1,M1
      FMASS(N)=FMASS(N)+(RO(M-1,N)*USP(M-1,N)+RO(M,N)*USP(M,N))*(Y(M,N)-
```

```
     1Y(M-1,N))/2.                                                      SPILL      83
260   CONTINUE                                                          SPILL      84
      NN=N-NS1+1                                                        SPILL      85
      FMASS(N)=FMASS(N)+(ROS(NN)*USP1(NN)+                              SPILL      86
     1RO(MS1,N)*USP(MS1,N))*(YS(NN)-Y(MS1,N))/2.                        SPILL      87
250    CONTINUE                                                         SPILL      88
290   CONTINUE                                                          SPILL      89
      FMAST(1)=FMASS(1)                                                 SPILL      90
      DO 285 N=2,N1                                                     SPILL      91
      FMAST(N)=FMAST(N-1)+(FMASS(N)+FMASS(N-1))*(XP(1,N)-XP(1,N-1))/2.  SPILL      92
285   CONTINUE                                                          SPILL      93
      WRITE(6,300)(N,XP(1,N),FMASS(N),FMAST(N),N=1,N1)                  SPILL      94
300   FORMAT(10X,'BODY STATION NO.=',I4,3X,'XP=',F10.6,3X,'LOCAL MASS SPSPILL      95
     1ILLED=',F10.5,3X,'TOTAL MASS SPILLED UPTO XP=',F10.5)             SPILL      96
      QX(1,1;NXM)=P(1,1;NXM)/P(1,1)                                     SPILL      97
      WRITE(6,805)                                                      SPILL      98
      WRITE(6,810)(N,XP(1,N),QX(1,N),QX(M1,N),N=1,N1)                   SPILL      99
805   FORMAT(///,5X,'N',16X,'XP',21X,'CENTER LINE PR. RATIO',           SPILL     100
     120X,'SIDEWALL PR. RATIO',/)                                       SPILL     101
810   FORMAT(4X,I3,7X,F13.5,20X,F15.4,20X,F15.4)                        SPILL     102
      IF(NSTRUT.EQ.0)GO TO 20                                           SPILL     103
      QXS(NS1;NS)=PS(1;NS)/P(1,1)                                       SPILL     104
      WRITE(6,806)                                                      SPILL     105
      WRITE(6,810)(N,XP(MS,N),QX(MS,N),QXS(N),N=NS1,NS2)                SPILL     106
20    CONTINUE                                                          SPILL     107
806   FORMAT(///,5X,'N',16X,'XP',16X,'STRUT UPPER SURFACE PR. RATIO',   SPILL     108
     116X,'STRUT LOWER SURFACE PR. RATIO',/)                            SPILL     109
      RETURN                                                            SPILL     110
      END                                                               SPILL     111
      SUBROUTINE PRINT                                                  PRINT       1
      BIT B1                                                            PRINT       2
      COMMON/F1/NXM,NXM1,NXM2,NXM3,NXM4,N1,M1,N11,M11,NS1,NS2,NS,MS,MS1 PRINT       3
     1,NCWL,NCWL1,NCWL2,NCWLM,NCWLP,NSTRUT                              PRINT       4
      COMMON/FA1/M12,M13,N12,N13,NXM5,NXM6,MS2,MS3,MS4,MSP1,MSP2,MSP3   PRINT       5
     1,NSM1,NS1P1,NS2M1,NS1M1,NS2P1,NNS,NS1M2,NXM7                      PRINT       6
      COMMON/F2/RO(61,55),U(61,55),V(61,55),P(61,55),T(61,55),          PRINT       7
     1 SH(61,55),H(61,55),ROS(55),US(55),VS(55),                        PRINT       8
     2 PS(55),TS(55),SHS(55),HS(55)                                     PRINT       9
      COMMON/F3/VISL(61,55),VIST(61,55),VIS(61,55),VISS(55)             PRINT      10
      COMMON/F7/X(61,55),Y(61,55),XXI(61,55),YXI(61,55),                PRINT      11
     1 XETA(61,55),YETA(61,55),AJ(61,55)                                PRINT      12
      COMMON/F8/XS(55),YS(55),XSXI(55),YSXI(55),XSETA(55),              PRINT      13
     1 YSETA(55),AJS(55)                                                PRINT      14
      COMMON/F11/ERRO(61,55),B1(61,55)                                  PRINT      15
      COMMON/F12/LSYM,NFLOW,LFI,LFO,ILT,LMAX,LW,LEXIT,CRIT,FDT,FDTL,    PRINT      16
     1 TIME,CCP,CCT,L,LK,TIMEL,CRITAVG                                  PRINT      17
      COMMON/F13/GAMA,RGAS,CP,PR,PRT,CVIS,FM,PF,TF,UF,ROF,SHF,HF        PRINT      18
      WRITE(6,500)L,TIME,TIMEL                                          PRINT      19
      WRITE(6,505) CCP,CCT,FDT,FDTL                                     PRINT      20
      WRITE(6,510) CRIT,CRITAVG,LEXIT                                   PRINT      21
      DO 250 N=1,N1                                                     PRINT      22
      WRITE(6,545)N                                                     PRINT      23
      WRITE(6,550)                                                      PRINT      24
      DO 250 MM=1,M1                                                    PRINT      25
      M=M1-MM+1                                                         PRINT      26
      WRITE(6,560)X(M,N),Y(M,N),U(M,N),V(M,N),P(M,N),T(M,N),RO(M,N),    PRINT      27
     1SH(M,N),VIST(M,N),ERRO(M,N)                                       PRINT      28
250   CONTINUE                                                          PRINT      29
      IF(NSTRUT.EQ.0)GO TO 255                                          PRINT      30
      WRITE(6,570)                                                      PRINT      31
      WRITE(6,560)(XS(N),YS(N),US(N),VS(N),PS(N),TS(N),ROS(N),SHS(N),   PRINT      32
     1HS(N),VISS(N),N=1,NS)                                             PRINT      33
255   CONTINUE                                                          PRINT      34
      ERRO(1,1;NXM)=RO(1,1;NXM)*U(1,1;NXM)                              PRINT      35
      DO 265 N=1,N1                                                     PRINT      36
      FMASS=0.                                                          PRINT      37
      IF(NSTRUT.EQ.0)GO TO 261                                          PRINT      38
      IF(N.LT.NS1.OR.N.GT.NS2)GO TO 261                                 PRINT      39
```

```
      NN=N-NS1+1                                                        PRINT     40
      DO 263 M=2,MS1                                                    PRINT     41
263    FMASS=FMASS+(ERRO(M,N)+ERRO(M-1,N))/2.*(Y(M,N)-Y(M-1,N))         PRINT     42
      FMASS=FMASS+(ERRO(MS1,N)+ROS(NN)*US(NN))/2.*(YS(NN)-Y(MS1,N))     PRINT     43
      DO 262 M=MSP1,M1                                                  PRINT     44
262    FMASS=FMASS+(ERRO(M,N)+ERRO(M-1,N))/2.*(Y(M,N)-Y(M-1,N))         PRINT     45
      GO TO 264                                                         PRINT     46
261   CONTINUE                                                          PRINT     47
      DO 260 M=2,M1                                                     PRINT     48
260    FMASS=FMASS+(ERRO(M,N)+ERRO(M-1,N))/2.*(Y(M,N)-Y(M-1,N))         PRINT     49
264   CONTINUE                                                          PRINT     50
      WRITE(6,700)N,X(1,N),FMASS                                        PRINT     51
265    CONTINUE                                                         PRINT     52
500   FORMAT(/,2X,'NO. OF ITERATIONS =',I5,2X,'TIME =',E15.7,           PRINT     53
     1       2X,'TIME LIMIT=',E15.7)                                    PRINT     54
 505 FORMAT(/,10X,'CCP=',F8.4,10X,'CCT=',F8.4,10X,'FDT=',F8.4,10X,      PRINT     55
     1       'FDTL=',F8.4,/)                                            PRINT     56
 510 FORMAT(/,1X,'LOCAL ERROR CRITERION=',F10.5,5X,                     PRINT     57
     1             'AVERAGE ERROR CRITERION=',F11.7,5X,                 PRINT     58
     1             'ORDER OF EXTRAPOLATION AT OUTFLOW =',I2)            PRINT     59
545   FORMAT(/,10X,'BODY STATION NO.=',I3,/)                            PRINT     60
550   FORMAT(/,6X,'X',9X,'Y',11X,'U',12X,'V',12X,'P',12X,'T',10X,'RO',  PRINT     61
     113X,'SH',11X,'VIST',5X,'ERRO')                                    PRINT     62
560   FORMAT(1X, 2F9.6,2F12.5,F14.5,F13.5,F11.5,F14.5,F15.6,F10.6)      PRINT     63
570   FORMAT(/,5X,'XS',8X,'YS',10X,'US',11X,'VS',11X,'PS',11X,'TS',10X, PRINT     64
     1'ROS',11X,'SHS',11X,'HS',9X,'VISS')                               PRINT     65
700   FORMAT(/,10X,'BODY STATION NO.=',I3,10X,'X(METERS)=',F9.5,10X,    PRINT     66
     1'MASS FLOW RATE(KG/M/SEC)=',F10.5)                                PRINT     67
      RETURN                                                            PRINT     68
      END                                                               PRINT     69
```

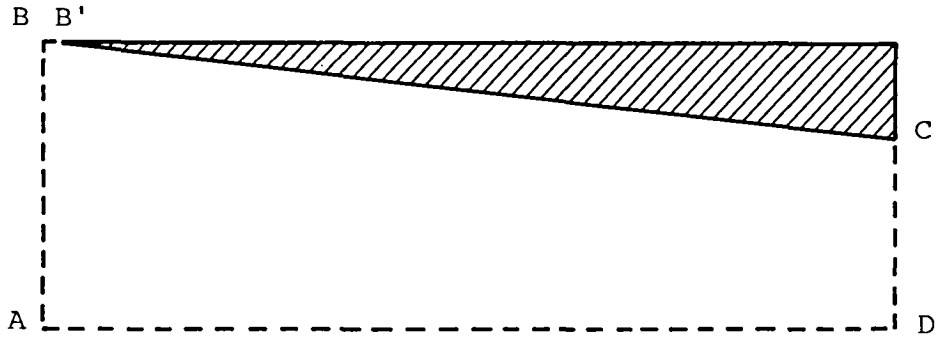
APPENDIX B

## EXAMPLES OF GEOMETRIES AND ASSOCIATED PARAMETERS

### Nonsymmetric Problems

Examples of geometries and associated parameters for nonsymmetric problems are as follows:

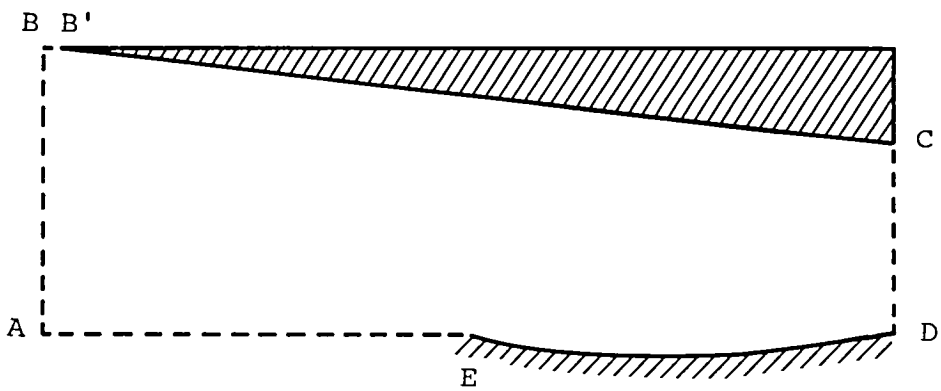
Example 1




| | | | |
|---|---|---|---|
| A: | N = 1, | M = 1 | N1 = 61 |
| B: | N = 1, | M = M1 | M1 = 51 |
| B': | N = 2, | M = M1 | SWEEP = 0 |
| C: | N = N1, | M = M1 | NSTRUT = 0 |
| D: | N = N1, | M = 1 | LSYM = 0 |
| | | | NCWL = 0 |

FOR NFLOW = 2: CALL EDDY (2,N1,1,M1,1,0)

Example 2

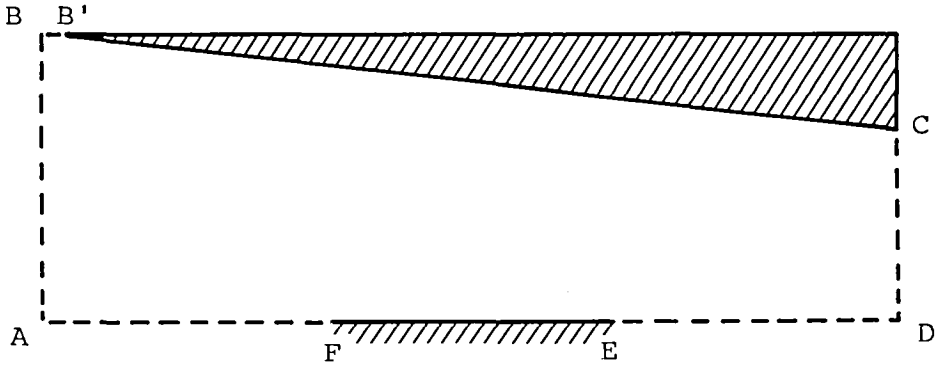



| | | |
|---|---|---|
| A: | N = 1, | M = 1 |
| B: | N = 1, | M = M1 |
| B': | N = 2, | M = M1 |
| C: | N = N1, | M = M1 |
| D: | N = N1, | M = 1 |
| E: | N = 31, | M = 1 |

N1 = 61
M1 = 51
SWEEP = 0
NSTRUT = 0
LSYM = 0
NCWL = 1
NCWL1 = 31
NCWL2 = N1

```
FOR NFLOW = 2:  CALL EDDY (2,30,1,M1,1,0)
                CALL EDDY (31,N1,1,M1,2,0)
```

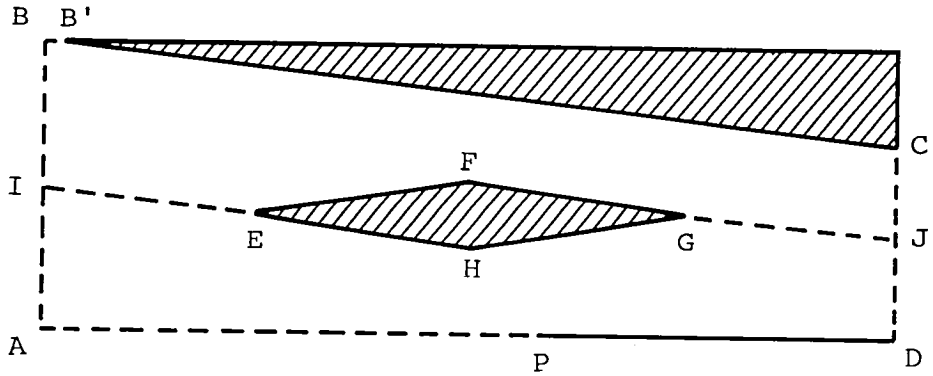
Example 3




| | | | |
|---|---|---|---|
| A: | N = 1, | M = 1 | N1 = 61 |
| B: | N = 1, | M = M1 | M1 = 51 |
| B': | N = 2, | M = M1 | SWEEP = 0 |
| C: | N = N1, | M = M1 | NSTRUT = 0 |
| D: | N = N1, | M = 1 | LSYM = 0 |
| E: | N = 41, | M = 1 | NCWL = 1 |
| F: | N = 21, | M = 1 | NCWL1 = 21 |
| | | | NCWL2 = 41 |

```
FOR NFLOW = 2:  CALL EDDY (2,20,1,M1,1,0)
                CALL EDDY (21,41,1,M1,2,0)
                CALL EDDY (42,N1,1,M1,1,0)
```

Example 4




| | | |
|---|---|---|
| A: | N = 1, | M = 1 |
| B: | N = 1, | M = M1 |
| B': | N = 2, | M = M1 |
| C: | N = N1, | M = M1 |
| D: | N = N1, | M = 1 |
| P: | N = 36, | M = 1 |
| I: | N = 1, | M = 26 |
| E: | N = 16, | M = 26 |
| F: | N = 31, | M = 26 |
| G: | N = 46, | M = 26 |
| H: | N = 46, | M = 26 |
| J: | N = N1, | M = 26 |

N1 = 61
M1 = 51
SWEEP = 0
NSTRUT = 1
NS1 = 16
NS2 = 46
MS = 26
LSYM = 0
NCWL = 1
NCWL1 = 36
NCWL2 = N1

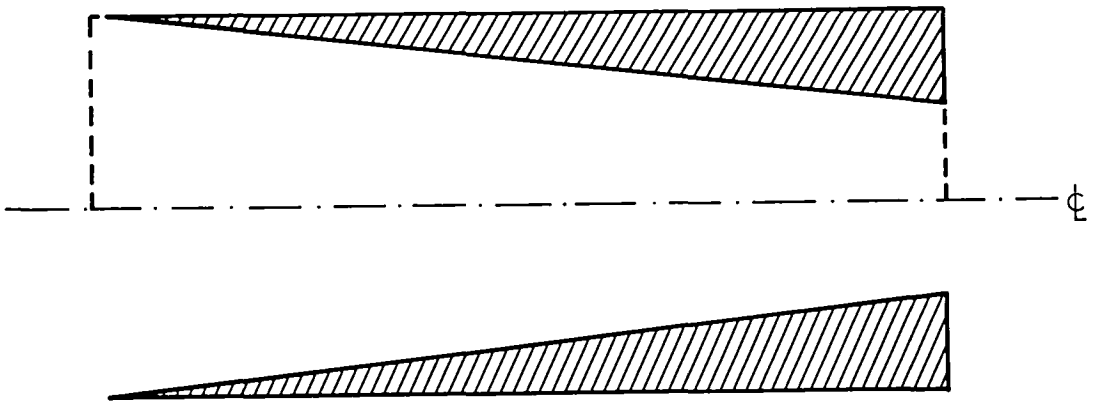
```
FOR NFLOW = 2:  CALL EDDY (2,15,1,M1,1,0)
                CALL EDDY (16,46,MS,M1,2,0)
                CALL EDDY (47,N1,1,M1,2,0)
                CALL EDDY (16,35,1,MS1,1,1)
                CALL EDDY (36,46,1,MS1,2,1)
```

APPENDIX B

## Symmetric Problems

The code solves only the upper half of the flow for symmetric problems. Examples of geometries and associated parameters for symmetric problems are as follows:

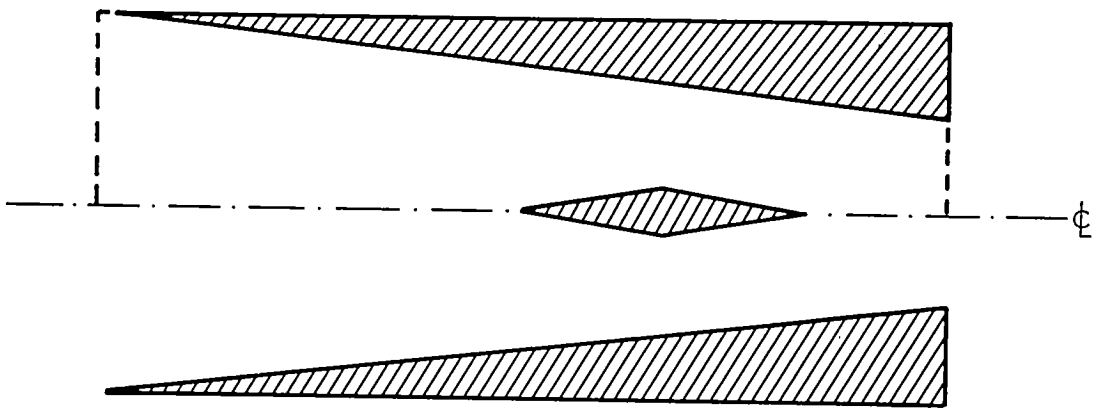
Example 1




```
    N1 = 61
    M1 = 51
 SWEEP = 0
NSTRUT = 0
  LSYM = 1
  NCWL = 0

FOR NFLOW = 2:  CALL EDDY (2,N1,1,M1,1,0)
```

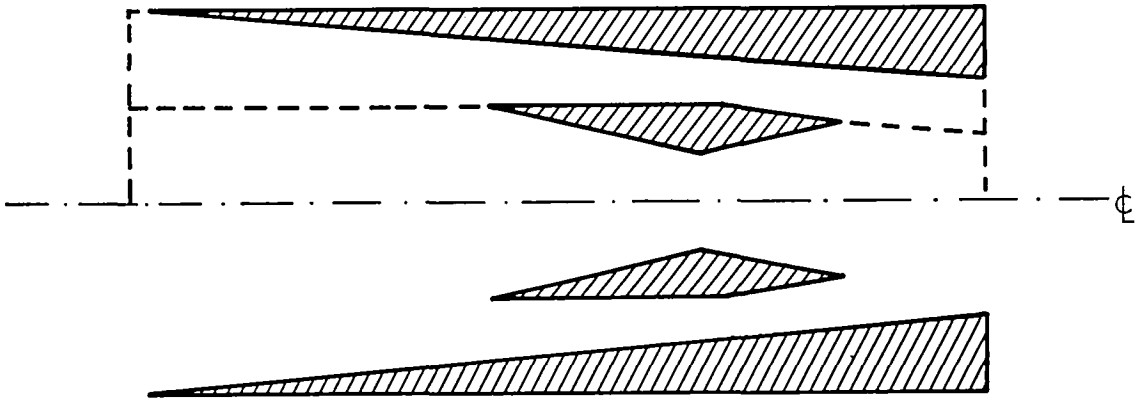
Example 2




```
    N1 = 61
    M1 = 51
 SWEEP = 0
NSTRUT = 0
  LSYM = 1
  NCWL = 1
 NCWL1 = 31
 NCWL2 = 51

FOR NFLOW = 2:  CALL EDDY (2,30,1,M1,1,0)
                CALL EDDY (31,51,1,M1,2,0)
                CALL EDDY (52,N1,1,M1,1,0)
```

Example 3



```
    N1 = 61
    M1 = 51
 SWEEP = 0
NSTRUT = 1
   NS1 = 26
   NS2 = 51
    MS = 27
  LSYM = 1
  NCWL = 0

FOR NFLOW = 2:  CALL EDDY (2,25,1,M1,1,0)
                CALL EDDY (26,51,MS,M1,2,0)
                CALL EDDY (26,51,1,MS1,1,1)
                CALL EDDY (52,N1,1,M1,1,0)
```

The sample problem discussed in this report provides another example of the type of geometries that can be analyzed with the present code.

## APPENDIX C

## USE OF PARAMETER LKK

In this appendix, several examples are given for the use of parameter LKK in trying more than one value of time-step parameter FDT or damping coefficients CCP and CCT during one computer run. For a given problem, it may be that the calculations become unstable (i.e., a negative temperature occurs in the flow field) for one particular value of, say, FDT, and the user may have to try several values before he finds the right value of FDT for which the calculations remain stable. The same thing may have to be done with CCP and CCT. One way to find the right values of FDT and CCP and CCT would be to submit several runs with a particular set of values of these parameters, but to avoid this, the code has a parameter LKK at line NASCRIN 224 that allows the user to try several values of FDT or CCP and CCT in one run of the code. To do so, the user has to change lines NASCRIN 219 through NASCRIN 224 as illustrated in the following examples:

### Example 1

```
      FDT = 1.
      CCP = 0.
1000  CONTINUE
      LKK = LKK + 1
      CCP = CCP + 0.1
      CCT = CCP
      IF(LKK.EQ.6)STOP
```

The code begins calculations with FDT = 1 and CCP = CCT = 0.1. If the calculations develop a negative temperature in the field, the code stops, prints out the flow-field variables at each grid point, and then returns to statement 1000. It is restarted with FDT = 1 and CCP = CCT = 0.2. If the calculations remain stable, the code will terminate in a normal way either on a convergence criterion or on LMAX without trying any more values of CCP and CCT. But, if a negative temperature develops again in the flow field, the calculations will restart with FDT = 1 and CCP = CCT = 0.3. For this example, the code will try five values of the damping coefficients in increments of 0.1 before stopping. The number of attempts is determined from the statement

```
      IF(LKK.EQ.6)STOP
```

and is always one less than the number in parentheses (in this case, it is 6 - 1 = 5).

### Example 2

```
      FDT = 1.
      CCP = 0.5
1000  CONTINUE
      LKK = LKK + 1
      CCT = CCP
      FDT = FDT - 0.1
      IF(LKK.EQ.4)STOP
```

This example allows the user to try three values of FDT, namely 0.9, 0.8, and 0.7, with a constant value of CCP = CCT = 0.5.

Example 3

```
        FDT = 0.9
        CCP = 0.5
1000    CONTINUE
        LKK = LKK + 1
        CCT = CCP
        IF(LKK.EQ.2)STOP
```

This example uses only one set of values of FDT, CCP, and CCT, namely FDT = 0.9 and CCP = CCT = 0.5.

If the code develops a negative temperature very early in the calculations, the chances are that the value of FDT needs to be reduced. But if the negative temperature occurs after several hundred time-steps, it may be caused by insufficient damping. This may not always be true but provides the user with a guideline in changing FDT, CCP, and CCT. Also, if the user wants to try several values of FDT, he should start from a higher value and decrease it gradually so that the allowable time-step is as large as possible. On the other hand, to keep the damping contribution as small as possible, the user should start with smaller values of CCP and CCT and gradually increase them.

## APPENDIX D

## ADJUSTMENT OF TIME-STEP AND DAMPING COEFFICIENTS

In this appendix, examples are presented to show how to gradually increase FDT or decrease CCP and CCT over a number of time-steps from their initial values.

### Example 1

Suppose the calculations are started with FDT = 0.5, and the user wants to increase it to 0.9 over 1000 time-steps. This can be done by inserting the following lines after line VISCOUS 51 for viscous flow calculations or after line INVICID 60 for inviscid flow calculations:

```
IF(L.GT.1000)GO TO 400
FDT = FDT + ((0.9 - 0.5)/1000)*2.
400 CONTINUE
```

The factor 2 in the expression for FDT occurs because the change in FDT is made after every two time-steps. By inserting the above three lines, FDT will change from 0.5 to 0.9 in 1000 time-steps. FDT will then remain constant at 0.9 for subsequent time-steps.

### Example 2

In the second example, suppose the user wants to use a damping coefficient of 0.7 for the first 1000 time-steps and then wants to reduce it to 0.5 over the next 2000 time-steps. This can be done by inserting the following lines in the code at the locations discussed in the previous example:

```
IF(L.GT.3000)GO TO 400
IF(L.LT.1000)GO TO 400
CCP = CCP - ((0.7 - 0.5)/2000)*2
CCT = CCP
400 CONTINUE
```

These statements change the damping coefficients from 0.7 to 0.5 between time-steps 1000 to 3000. The damping coefficients remain constant at 0.5 thereafter.

# APPENDIX E

## SAMPLE OUTPUTS

### Geometry Output

| N | M | X | Y | XXI | XETA | YXI | YETA | AJ |
|---|---|---|---|---|---|---|---|---|
| 1 | 61 | .00000000 | .04000000 | .00500000 | .00000000 | .00000000 | .00002055 | .00000010 |
| 1 | 60 | .00000000 | .03997510 | .00500000 | .00000000 | .00000000 | .00002983 | .00000015 |
| 1 | 59 | .00000000 | .03993897 | .00500000 | .00000000 | .00000000 | .00004324 | .00000022 |
| 1 | 58 | .00000000 | .03988665 | .00500000 | .00000000 | .00000000 | .00006259 | .00000031 |
| 1 | 57 | .00000000 | .03981099 | .00500000 | .00000000 | .00000000 | .00009040 | .00000045 |
| 1 | 56 | .00000000 | .03970187 | .00500000 | .00000000 | .00000000 | .00013018 | .00000065 |
| 1 | 55 | .00000000 | .03954509 | .00500000 | .00000000 | .00000000 | .00018662 | .00000093 |
| 1 | 54 | .00000000 | .03932106 | .00500000 | .00000000 | .00000000 | .00026582 | .00000133 |
| 1 | 53 | .00000000 | .03900339 | .00500000 | .00000000 | .00000000 | .00037522 | .00000188 |
| 1 | 52 | .00000000 | .03855786 | .00500000 | .00000000 | .00000000 | .00052288 | .00000261 |
| 1 | 51 | .00000000 | .03794250 | .00500000 | .00000000 | .00000000 | .00071578 | .00000358 |
| 1 | 50 | .00000000 | .03711035 | .00500000 | .00000000 | .00000000 | .00095622 | .00000478 |
| 1 | 49 | .00000000 | .03601659 | .00500000 | .00000000 | .00000000 | .00123655 | .00000618 |
| 1 | 48 | .00000000 | .03463154 | .00500000 | .00000000 | .00000000 | .00153337 | .00000767 |
| 1 | 47 | .00000000 | .03295800 | .00500000 | .00000000 | .00000000 | .00180529 | .00000903 |
| 1 | 46 | .00000000 | .03104684 | .00500000 | .00000000 | .00000000 | .00199974 | .00001000 |
| 1 | 45 | .00000000 | .02900000 | .00500000 | .00000000 | .00000000 | .00207073 | .00001035 |
| 1 | 44 | .00000000 | .02695316 | .00500000 | .00000000 | .00000000 | .00199974 | .00001000 |
| 1 | 43 | .00000000 | .02504200 | .00500000 | .00000000 | .00000000 | .00180529 | .00000903 |
| 1 | 42 | .00000000 | .02336846 | .00500000 | .00000000 | .00000000 | .00153337 | .00000767 |
| 1 | 41 | .00000000 | .02198341 | .00500000 | .00000000 | .00000000 | .00123655 | .00000618 |
| 1 | 40 | .00000000 | .02088965 | .00500000 | .00000000 | .00000000 | .00095622 | .00000478 |
| 1 | 39 | .00000000 | .02005750 | .00500000 | .00000000 | .00000000 | .00071578 | .00000358 |
| 1 | 38 | .00000000 | .01944214 | .00500000 | .00000000 | .00000000 | .00052288 | .00000261 |
| 1 | 37 | .00000000 | .01899661 | .00500000 | .00000000 | .00000000 | .00037522 | .00000188 |
| 1 | 36 | .00000000 | .01867894 | .00500000 | .00000000 | .00000000 | .00026582 | .00000133 |
| 1 | 35 | .00000000 | .01845491 | .00500000 | .00000000 | .00000000 | .00018662 | .00000093 |
| 1 | 34 | .00000000 | .01829813 | .00500000 | .00000000 | .00000000 | .00013018 | .00000065 |
| 1 | 33 | .00000000 | .01818901 | .00500000 | .00000000 | .00000000 | .00009040 | .00000045 |
| 1 | 32 | .00000000 | .01811335 | .00500000 | .00000000 | .00000000 | .00006259 | .00000031 |
| 1 | 31 | .00000000 | .01806103 | .00500000 | .00000000 | .00000000 | .00004324 | .00000022 |
| 1 | 30 | .00000000 | .01802490 | .00500000 | .00000000 | .00000000 | .00002983 | .00000015 |
| 1 | 29 | .00000000 | .01800000 | .00500000 | .00000000 | .00000000 | .00002443 | .00000012 |
| 1 | 28 | .00000000 | .01797604 | .00500000 | .00000000 | .00000000 | .00002941 | .00000015 |
| 1 | 27 | .00000000 | .01793941 | .00500000 | .00000000 | .00000000 | .00004494 | .00000022 |
| 1 | 26 | .00000000 | .01788348 | .00500000 | .00000000 | .00000000 | .00006853 | .00000034 |
| 1 | 25 | .00000000 | .01779835 | .00500000 | .00000000 | .00000000 | .00010416 | .00000052 |
| 1 | 24 | .00000000 | .01766927 | .00500000 | .00000000 | .00000000 | .00015752 | .00000079 |
| 1 | 23 | .00000000 | .01747481 | .00500000 | .00000000 | .00000000 | .00023643 | .00000118 |
| 1 | 22 | .00000000 | .01718460 | .00500000 | .00000000 | .00000000 | .00035085 | .00000175 |
| 1 | 21 | .00000000 | .01675752 | .00500000 | .00000000 | .00000000 | .00051199 | .00000256 |
| 1 | 20 | .00000000 | .01614187 | .00500000 | .00000000 | .00000000 | .00072909 | .00000365 |
| 1 | 19 | .00000000 | .01528027 | .00500000 | .00000000 | .00000000 | .00100279 | .00000501 |
| 1 | 18 | .00000000 | .01412311 | .00500000 | .00000000 | .00000000 | .00131509 | .00000658 |
| 1 | 17 | .00000000 | .01265206 | .00500000 | .00000000 | .00000000 | .00162060 | .00000810 |
| 1 | 16 | .00000000 | .01090722 | .00500000 | .00000000 | .00000000 | .00185017 | .00000925 |
| 1 | 15 | .00000000 | .00900000 | .00500000 | .00000000 | .00000000 | .00193626 | .00000968 |
| 1 | 14 | .00000000 | .00709278 | .00500000 | .00000000 | .00000000 | .00185017 | .00000925 |
| 1 | 13 | .00000000 | .00534794 | .00500000 | .00000000 | .00000000 | .00162060 | .00000810 |
| 1 | 12 | .00000000 | .00387689 | .00500000 | .00000000 | .00000000 | .00131509 | .00000658 |
| 1 | 11 | .00000000 | .00271973 | .00500000 | .00000000 | .00000000 | .00100279 | .00000501 |
| 1 | 10 | .00000000 | .00185813 | .00500000 | .00000000 | .00000000 | .00072909 | .00000365 |
| 1 | 9 | .00000000 | .00124248 | .00500000 | .00000000 | .00000000 | .00051199 | .00000256 |
| 1 | 8 | .00000000 | .00081540 | .00500000 | .00000000 | .00000000 | .00035085 | .00000175 |
| 1 | 7 | .00000000 | .00052519 | .00500000 | .00000000 | .00000000 | .00023643 | .00000118 |
| 1 | 6 | .00000000 | .00033073 | .00500000 | .00000000 | .00000000 | .00015752 | .00000079 |
| 1 | 5 | .00000000 | .00020165 | .00500000 | .00000000 | .00000000 | .00010416 | .00000052 |
| 1 | 4 | .00000000 | .00011652 | .00500000 | .00000000 | .00000000 | .00006853 | .00000034 |
| 1 | 3 | .00000000 | .00006059 | .00500000 | .00000000 | .00000000 | .00004494 | .00000022 |
| 1 | 2 | .00000000 | .00002396 | .00500000 | .00000000 | .00000000 | .00002941 | .00000015 |
| 1 | 1 | .00000000 | .00000000 | .00500000 | .00000000 | .00000000 | .00001922 | .00000010 |

APPENDIX E

## Geometry Output for Lower Surface of Strut

The geometry output for the lower surface of the strut is printed only for NSTRUT = 1.

| N | M | X | Y | XXI | XETA | YXI | YETA | AJ |
|---|---|---|---|---|---|---|---|---|
| 1 | 29 | .05500000 | .01800000 | .00250000 | .00000000 | -.00016457 | .00001922 | .00000005 |
| 2 | 29 | .05750000 | .01767087 | .00250000 | -.00000000 | -.00032913 | .00001887 | .00000005 |
| 3 | 29 | .06000000 | .01734174 | .00250000 | -.00000000 | -.00032913 | .00001852 | .00000005 |
| 4 | 29 | .06250000 | .01701261 | .00250000 | -.00000000 | -.00032913 | .00001816 | .00000005 |
| 5 | 29 | .06500000 | .01668348 | .00250000 | .00000000 | -.00032913 | .00001781 | .00000004 |
| 6 | 29 | .06750000 | .01635434 | .00250000 | .00000000 | -.00032913 | .00001746 | .00000004 |
| 7 | 29 | .07000000 | .01602521 | .00250000 | .00000000 | -.00032913 | .00001711 | .00000004 |
| 8 | 29 | .07250000 | .01569608 | .00250000 | -.00000000 | -.00032913 | .00001676 | .00000004 |
| 9 | 29 | .07500000 | .01536695 | .00250000 | .00000000 | -.00032913 | .00001641 | .00000004 |
| 10 | 29 | .07750000 | .01503782 | .00250000 | .00000000 | -.00032913 | .00001570 | .00000004 |
| 11 | 29 | .08000000 | .01470869 | .00250000 | .00000000 | -.00032913 | .00001500 | .00000004 |
| 12 | 29 | .08250000 | .01437956 | .00250000 | -.00000000 | -.00032913 | .00001430 | .00000004 |
| 13 | 29 | .08500000 | .01405043 | .00250000 | .00000000 | -.00032913 | .00001360 | .00000003 |
| 14 | 29 | .08750000 | .01372129 | .00250000 | .00000000 | -.00032913 | .00001289 | .00000003 |
| 15 | 29 | .09000000 | .01339216 | .00250000 | -.00000000 | -.00032913 | .00001219 | .00000003 |
| 16 | 29 | .09250000 | .01306303 | .00250000 | -.00000000 | -.00032913 | .00001149 | .00000003 |
| 17 | 29 | .09500000 | .01273390 | .00250000 | .00000000 | -.00032913 | .00001078 | .00000003 |
| 18 | 29 | .09750000 | .01240477 | .00250000 | .00000000 | -.00032913 | .00001008 | .00000003 |
| 19 | 29 | .10000000 | .01207564 | .00250000 | -.00000000 | -.00002720 | .00000938 | .00000002 |
| 20 | 29 | .10250000 | .01235037 | .00250000 | -.00000000 | .00027473 | .00001002 | .00000003 |
| 21 | 29 | .10500000 | .01262510 | .00250000 | .00000000 | .00027473 | .00001067 | .00000003 |
| 22 | 29 | .10750000 | .01289983 | .00250000 | -.00000000 | .00027473 | .00001131 | .00000003 |
| 23 | 29 | .11000000 | .01317456 | .00250000 | -.00000000 | .00027473 | .00001196 | .00000003 |
| 24 | 29 | .11250000 | .01344929 | .00250000 | -.00000000 | .00027473 | .00001260 | .00000003 |
| 25 | 29 | .11500000 | .01372402 | .00250000 | .00000000 | .00027473 | .00001325 | .00000003 |
| 26 | 29 | .11750000 | .01399875 | .00250000 | -.00000000 | .00027473 | .00001389 | .00000003 |
| 27 | 29 | .12000000 | .01427348 | .00250000 | -.00000000 | .00027473 | .00001454 | .00000004 |
| 28 | 29 | .12250000 | .01454821 | .00250000 | -.00000000 | .00027473 | .00001518 | .00000004 |
| 29 | 29 | .12500000 | .01482293 | .00250000 | -.00000000 | .00027473 | .00001583 | .00000004 |
| 30 | 29 | .12750000 | .01509766 | .00250000 | .00000000 | .00027473 | .00001612 | .00000004 |
| 31 | 29 | .13000000 | .01537239 | .00250000 | .00000000 | .00000557 | .00001641 | .00000004 |

# APPENDIX E

## Flow-Field Output

```
L=20700        MAX.ERROR=  .13830E-03        MEAN SQRT ERROR=  .16545E-06

L=20750        MAX.ERROR=  .30886E-03        MEAN SQRT ERROR=  .26362E-06

L=20800        MAX.ERROR=  .15369E-03        MEAN SQRT ERROR=  .20901E-06

L=20850        MAX.ERROR=  .29912E-03        MEAN SQRT ERROR=  .31551E-06

L=20900        MAX.ERROR=  .25054E-03        MEAN SQRT ERROR=  .25941E-06

L=20950        MAX.ERROR=  .29888E-03        MEAN SQRT ERROR=  .26156E-06

NO. OF ITERATIONS =20966  TIME =  .2635594E-03  TIME LIMIT=  .2635454E-03

CCP=  .5000        CCI=  .5000        EDT=  1.0000        EDTI=  1.0000

LOCAL ERROR CRITERION=  .00010   AVERAGE ERROR CRITERION=  .0000001   ORDER OF EXTRAPOLATION AT OUTFLOW = 1

BODY STATION NO.= 27
```

| X | Y | U | V | P | T | RO | SH | VIST | ERRO |
|---|---|---|---|---|---|---|---|---|---|
| .082500 | .031854 | .00000 | .00000 | 21117.65277 | 1439.40576 | .05112 | 1445883.08563 | .000000 | .000000 |
| .082500 | .031839 | 141.43285 | -16.06448 | 21117.65277 | 1439.40576 | .05112 | 1445883.08563 | .000000 | .000000 |
| .082500 | .031816 | 336.07852 | -36.45959 | 21090.51808 | 1420.07701 | .05175 | 1426467.36132 | .000002 | .000000 |
| .082500 | .031783 | 578.87778 | -64.58265 | 21170.87359 | 1358.09297 | .05432 | 1364204.38656 | .000017 | .000000 |
| .082500 | .031735 | 805.29764 | -85.39118 | 21015.02646 | 1250.47902 | .05856 | 1256106.17466 | .000069 | .000000 |
| .082500 | .031667 | 986.43322 | -109.12013 | 21268.45406 | 1127.02993 | .06575 | 1132101.56738 | .000198 | .000000 |
| .082500 | .031568 | 1113.72752 | -118.74615 | 20912.22281 | 1022.32666 | .07127 | 1026927.13113 | .000385 | .000000 |
| .082500 | .031427 | 1240.94851 | -138.72676 | 21389.75892 | 905.14341 | .08234 | 909216.55292 | .000445 | .000000 |
| .082500 | .031227 | 1365.88071 | -149.85078 | 20854.48174 | 774.06804 | .09387 | 777551.34409 | .000505 | .000000 |
| .082500 | .030946 | 1500.03517 | -169.13754 | 21424.67558 | 621.31771 | .12015 | 624113.64232 | .000626 | .000000 |
| .082500 | .030559 | 1587.67324 | -182.92320 | 21123.90707 | 509.05603 | .14459 | 511346.76013 | .000596 | .000000 |
| .082500 | .030035 | 1653.48728 | -184.57761 | 21248.61534 | 420.69120 | .17599 | 422584.30916 | .000281 | .000000 |
| .082500 | .029346 | 1670.36848 | -203.03762 | 22328.21261 | 391.57050 | .19868 | 393332.56396 | .000062 | .000000 |
| .082500 | .028474 | 1683.74256 | -183.51539 | 20268.58505 | 375.99293 | .18783 | 377684.90010 | .000010 | .000000 |
| .082500 | .027420 | 1680.99560 | -192.14204 | 21004.27238 | 379.87616 | .19266 | 381585.60087 | .000002 | .000000 |
| .082500 | .026216 | 1674.48531 | -206.16828 | 22323.90846 | 386.84891 | .20107 | 388589.72723 | .000001 | .000000 |
| .082500 | .024927 | 1679.96388 | -184.14661 | 21400.67316 | 381.43901 | .19549 | 383155.48809 | .000000 | .000001 |
| .082500 | .023638 | 1684.27924 | -149.13599 | 21234.56059 | 378.12336 | .19567 | 379824.91351 | .000000 | .000001 |
| .082500 | .022435 | 1676.61644 | -137.98379 | 23869.82837 | 391.32011 | .21254 | 393081.04936 | .000000 | .000000 |
| .082500 | .021381 | 1669.09743 | -150.37669 | 26157.35794 | 404.32357 | .22542 | 406143.02259 | .000000 | .000001 |
| .082500 | .020509 | 1669.07806 | -183.67230 | 25083.17521 | 397.31669 | .21997 | 399104.61804 | .000000 | .000002 |
| .082500 | .019820 | 1669.92529 | -215.63091 | 23381.30214 | 390.87465 | .20842 | 392633.58388 | .000002 | .000003 |
| .082500 | .019296 | 1663.74045 | -175.47543 | 27344.37458 | 407.72313 | .23368 | 409557.88697 | .000017 | .000001 |
| .082500 | .018908 | 1646.28555 | -70.29511 | 37976.16622 | 448.05501 | .29532 | 450071.25703 | .000149 | .000002 |
| .082500 | .018628 | 1626.93619 | -36.11208 | 41798.91318 | 480.62841 | .30302 | 482791.24141 | .000536 | .000003 |
| .082500 | .018428 | 1558.12728 | -26.30672 | 41791.84687 | 577.99980 | .25193 | 580600.79916 | .000616 | .000002 |
| .082500 | .018286 | 1437.55773 | -13.07465 | 43290.19475 | 739.04716 | .20410 | 742372.86875 | .000519 | .000000 |
| .082500 | .018188 | 1250.59124 | -13.84925 | 42510.20064 | 948.35927 | .15618 | 952626.86225 | .000399 | .000001 |
| .082500 | .018119 | 1071.71643 | -6.03502 | 43165.64131 | 1117.57226 | .13458 | 1122601.33088 | .000344 | .000002 |
| .082500 | .018071 | 865.60208 | -4.25916 | 42792.49836 | 1279.77870 | .11651 | 1285537.70088 | .000104 | .000003 |
| .082500 | .018038 | 586.90389 | 1.66831 | 43023.90209 | 1423.10408 | .10534 | 1429508.04375 | .000018 | .000003 |
| .082500 | .018016 | 267.54040 | .53586 | 42971.19190 | 1488.77445 | .10057 | 1495473.93608 | .000001 | .000003 |
| .082500 | .018000 | .00000 | .00000 | 42971.19190 | 1488.77445 | .10057 | 1495473.93608 | .000000 | .000003 |
| .082500 | .014362 | 257.94202 | -31.70449 | 24473.19644 | 1335.15858 | .06387 | 1341166.78876 | .000000 | .000000 |
| .082500 | .014334 | 584.71437 | -82.24051 | 24617.20662 | 1273.04066 | .06738 | 1278769.33934 | .000011 | .000000 |
| .082500 | .014293 | 850.75048 | -111.00615 | 24222.25756 | 1149.71597 | .07341 | 1154889.69343 | .000070 | .000000 |
| .082500 | .014230 | 1083.60635 | -151.65614 | 24878.68764 | 988.04947 | .08773 | 992495.69624 | .000182 | .000000 |
| .082500 | .014133 | 1266.48235 | -167.50022 | 23969.94329 | 825.60987 | .10116 | 829325.11520 | .000209 | .000000 |
| .082500 | .013989 | 1465.73033 | -212.01746 | 25149.41894 | 632.69228 | .13850 | 635539.39513 | .000270 | .000000 |
| .082500 | .013773 | 1561.26638 | -216.45890 | 24005.41005 | 522.66908 | .16003 | 525021.06968 | .000174 | .000000 |
| .082500 | .013455 | 1639.05709 | -230.53868 | 24985.58220 | 434.34516 | .20043 | 436299.71807 | .000034 | .000000 |
| .082500 | .012997 | 1650.56900 | -238.77675 | 26146.25372 | 417.71018 | .21810 | 419589.87790 | .000004 | .000000 |
| .082500 | .012356 | 1668.62196 | -215.22299 | 23322.23705 | 396.98005 | .20470 | 398766.45883 | .000000 | .000000 |
| .082500 | .011495 | 1647.68176 | -273.37145 | 28518.42230 | 420.50218 | .23631 | 422394.44367 | .000000 | .000000 |
| .082500 | .010401 | 1650.36467 | -254.77333 | 26639.86689 | 414.20683 | .22410 | 416070.75641 | .000000 | .000000 |
| .082500 | .009102 | 1692.36726 | -137.26285 | 17027.00245 | 363.72526 | .16311 | 365362.02588 | .000000 | .000000 |
| .082500 | .007683 | 1724.73146 | -40.02671 | 11498.40179 | 317.85142 | .12605 | 319281.75610 | .000000 | .000000 |
| .082500 | .006264 | 1732.47659 | -10.26960 | 10469.69631 | 303.73396 | .12010 | 305100.75842 | .000000 | .000000 |
| .082500 | .004966 | 1733.70155 | 2.88621 | 10311.41879 | 301.61352 | .11912 | 302970.77684 | .000000 | .000000 |
| .082500 | .003872 | 1736.49659 | -3.00226 | 9588.62752 | 301.95676 | .11064 | 303315.56324 | .000000 | .000000 |
| .082500 | .003011 | 1713.79186 | 67.65812 | 12654.71740 | 336.48697 | .13104 | 338001.16153 | .000000 | .000000 |
| .082500 | .002370 | 1662.18952 | 225.94421 | 23239.03912 | 403.92572 | .20046 | 405743.38488 | .000000 | .000000 |
| .082500 | .001912 | 1653.50732 | 269.26610 | 27138.49898 | 411.83389 | .22961 | 413687.14712 | .000001 | .000000 |
| .082500 | .001594 | 1659.49103 | 241.89582 | 25120.18274 | 397.07668 | .22043 | 398863.52915 | .000016 | .000000 |
| .082500 | .001378 | 1621.71685 | 251.95802 | 26364.45483 | 447.17010 | .20543 | 449182.36270 | .000120 | .000000 |
| .082500 | .001233 | 1524.67339 | 219.38323 | 25586.08915 | 566.95673 | .15724 | 569508.03064 | .000191 | .000000 |
| .082500 | .001137 | 1298.83159 | 181.60771 | 25970.60636 | 812.80505 | .11133 | 816462.67068 | .000146 | .000000 |
| .082500 | .001074 | 1013.01014 | 136.99949 | 25622.24331 | 1051.57282 | .08490 | 1056304.89438 | .000111 | .000000 |
| .082500 | .001032 | 711.13758 | 100.05549 | 25827.66485 | 1229.58098 | .07319 | 1235114.09398 | .000018 | .000000 |
| .082500 | .001005 | 329.17494 | 44.80071 | 25761.44902 | 1325.11109 | .06774 | 1331074.08549 | .000001 | .000000 |
| .082500 | .000987 | .00000 | .00000 | 25761.44902 | 1325.11109 | .06774 | 1331074.08549 | .000000 | .000000 |

# APPENDIX E

## Flow-Field Output on Lower Surface of Strut

The flow-field output on the lower surface of the strut is printed only for NSTRUT = 1.

| XS | YS | US | VS | PS | TS | ROS | SHS | HS | VISS |
|---|---|---|---|---|---|---|---|---|---|
| .055000 | .018000 | .00000 | .00000 | 21150.10588 | 1022.61893 | .07206 | 1027220.71965 | 1027220.719653 | .000042 |
| .057500 | .017671 | .00000 | .00000 | 26808.30404 | 1204.73457 | .07753 | 1210155.88051 | 1210155.880506 | .000046 |
| .060000 | .017342 | .00000 | .00000 | 26062.07177 | 1258.61680 | .07215 | 1264280.57540 | 1264280.575397 | .000048 |
| .062500 | .017013 | .00000 | .00000 | 26132.18556 | 1268.71339 | .07177 | 1274422.60142 | 1274422.601416 | .000048 |
| .065000 | .016683 | .00000 | .00000 | 25678.79887 | 1306.53963 | .06848 | 1312419.05387 | 1312419.053874 | .000049 |
| .067500 | .016354 | .00000 | .00000 | 24707.57060 | 1289.19170 | .06678 | 1294993.05786 | 1294993.057856 | .000048 |
| .070000 | .016025 | .00000 | .00000 | 25546.14480 | 1330.53078 | .06690 | 1336518.17092 | 1336518.170923 | .000049 |
| .072500 | .015696 | .00000 | .00000 | 25998.06986 | 1330.99596 | .06806 | 1336985.44612 | 1336985.446119 | .000049 |
| .075000 | .015367 | .00000 | .00000 | 25403.51061 | 1315.06317 | .06731 | 1320980.95070 | 1320980.950701 | .000049 |
| .077500 | .015038 | .00000 | .00000 | 25671.50292 | 1347.91856 | .06636 | 1353984.19774 | 1353984.197735 | .000049 |
| .080000 | .014709 | .00000 | .00000 | 25094.74709 | 1342.85261 | .06511 | 1348895.44688 | 1348895.446879 | .000049 |
| .082500 | .014380 | .00000 | .00000 | 24473.19644 | 1335.15858 | .06387 | 1341166.78876 | 1341166.788756 | .000049 |
| .085000 | .014050 | .00000 | .00000 | 25658.71111 | 1373.16073 | .06511 | 1379339.95690 | 1379339.956904 | .000050 |
| .087500 | .013721 | .00000 | .00000 | 25440.28089 | 1361.34970 | .06511 | 1367475.77132 | 1367475.771321 | .000050 |
| .090000 | .013392 | .00000 | .00000 | 25335.41168 | 1357.54659 | .06503 | 1363655.55452 | 1363655.554519 | .000050 |
| .092500 | .013063 | .00000 | .00000 | 25011.34130 | 1383.89177 | .06297 | 1390119.28628 | 1390119.286284 | .000050 |
| .095000 | .012734 | .00000 | .00000 | 24701.53615 | 1368.00297 | .06292 | 1374158.98585 | 1374158.985851 | .000050 |
| .097500 | .012405 | .00000 | .00000 | 25526.21038 | 1401.97981 | .06344 | 1408288.71958 | 1408288.719577 | .000051 |
| .100000 | .012076 | .00000 | .00000 | 16999.82893 | 1348.00272 | .04394 | 1354068.73128 | 1354068.731283 | .000049 |
| .102500 | .012350 | .00000 | .00000 | 9323.11786 | 1315.50394 | .02469 | 1321423.70550 | 1321423.705498 | .000049 |
| .105000 | .012625 | .00000 | .00000 | 7954.12589 | 1334.66212 | .02077 | 1340668.10302 | 1340668.103019 | .000049 |
| .107500 | .012900 | .00000 | .00000 | 7233.42488 | 1309.19230 | .01925 | 1315083.66731 | 1315083.667314 | .000049 |
| .110000 | .013175 | .00000 | .00000 | 6951.06819 | 1353.51634 | .01789 | 1359607.16406 | 1359607.164064 | .000050 |
| .112500 | .013449 | .00000 | .00000 | 6340.89227 | 1324.26413 | .01668 | 1330223.31745 | 1330223.317449 | .000049 |
| .115000 | .013724 | .00000 | .00000 | 6184.38137 | 1353.23405 | .02107 | 1359323.60707 | 1359323.607069 | .000050 |
| .117500 | .013999 | .00000 | .00000 | 10494.36405 | 1074.83447 | .03402 | 1079671.22156 | 1079671.221560 | .000043 |
| .120000 | .014273 | .00000 | .00000 | 14690.87377 | 1445.70312 | .03541 | 1452208.78134 | 1452208.781338 | .000052 |
| .122500 | .014548 | .00000 | .00000 | 20691.04277 | 1391.46116 | .05181 | 1397722.73432 | 1397722.734321 | .000050 |
| .125000 | .014823 | .00000 | .00000 | 24926.21714 | 1450.00748 | .05990 | 1456532.51759 | 1456532.517593 | .000052 |

## Mass Flow Rate Output

| | | |
|---|---|---|
| BODY STATION NO.= 33 | X(METERS)= .09750 | MASS FLOW RATE(KG/M/SEC)= 8.07562 |
| BODY STATION NO.= 34 | X(METERS)= .10000 | MASS FLOW RATE(KG/M/SEC)= 8.05485 |
| BODY STATION NO.= 35 | X(METERS)= .10250 | MASS FLOW RATE(KG/M/SEC)= 8.05039 |
| BODY STATION NO.= 36 | X(METERS)= .10500 | MASS FLOW RATE(KG/M/SEC)= 8.05796 |
| BODY STATION NO.= 37 | X(METERS)= .10750 | MASS FLOW RATE(KG/M/SEC)= 8.03889 |
| BODY STATION NO.= 38 | X(METERS)= .11000 | MASS FLOW RATE(KG/M/SEC)= 8.05932 |
| BODY STATION NO.= 39 | X(METERS)= .11250 | MASS FLOW RATE(KG/M/SEC)= 8.04369 |
| BODY STATION NO.= 40 | X(METERS)= .11500 | MASS FLOW RATE(KG/M/SEC)= 8.06730 |
| BODY STATION NO.= 41 | X(METERS)= .11750 | MASS FLOW RATE(KG/M/SEC)= 8.05163 |
| BODY STATION NO.= 42 | X(METERS)= .12000 | MASS FLOW RATE(KG/M/SEC)= 8.07953 |
| BODY STATION NO.= 43 | X(METERS)= .12250 | MASS FLOW RATE(KG/M/SEC)= 8.06094 |
| BODY STATION NO.= 44 | X(METERS)= .12500 | MASS FLOW RATE(KG/M/SEC)= 8.09378 |

# APPENDIX E

## Flow Spillage Output

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BODY | STATION | NO.= | 26 | XP= | .092376 | LOCAL MASS SPILLED= | .12079 | TOTAL MASS SPILLED UPTO XP= | .00386 |
| BODY | STATION | NO.= | 27 | XP= | .095263 | LOCAL MASS SPILLED= | .12302 | TOTAL MASS SPILLED UPTO XP= | .00421 |
| BODY | STATION | NO.= | 28 | XP= | .098150 | LOCAL MASS SPILLED= | .12871 | TOTAL MASS SPILLED UPTO XP= | .00458 |
| BODY | STATION | NO.= | 29 | XP= | .101036 | LOCAL MASS SPILLED= | .15362 | TOTAL MASS SPILLED UPTO XP= | .00498 |
| BODY | STATION | NO.= | 30 | XP= | .103923 | LOCAL MASS SPILLED= | .17421 | TOTAL MASS SPILLED UPTO XP= | .00546 |
| BODY | STATION | NO.= | 31 | XP= | .106810 | LOCAL MASS SPILLED= | .19476 | TOTAL MASS SPILLED UPTO XP= | .00599 |
| BODY | STATION | NO.= | 32 | XP= | .109697 | LOCAL MASS SPILLED= | .20270 | TOTAL MASS SPILLED UPTO XP= | .00656 |
| BODY | STATION | NO.= | 33 | XP= | .112583 | LOCAL MASS SPILLED= | .22472 | TOTAL MASS SPILLED UPTO XP= | .00718 |
| BODY | STATION | NO.= | 34 | XP= | .115470 | LOCAL MASS SPILLED= | .24052 | TOTAL MASS SPILLED UPTO XP= | .00785 |
| BODY | STATION | NO.= | 35 | XP= | .118357 | LOCAL MASS SPILLED= | .24475 | TOTAL MASS SPILLED UPTO XP= | .00855 |
| BODY | STATION | NO.= | 36 | XP= | .121244 | LOCAL MASS SPILLED= | .24369 | TOTAL MASS SPILLED UPTO XP= | .00926 |
| BODY | STATION | NO.= | 37 | XP= | .124130 | LOCAL MASS SPILLED= | .23774 | TOTAL MASS SPILLED UPTO XP= | .00995 |
| BODY | STATION | NO.= | 38 | XP= | .127017 | LOCAL MASS SPILLED= | .23179 | TOTAL MASS SPILLED UPTO XP= | .01063 |
| BODY | STATION | NO.= | 39 | XP= | .129904 | LOCAL MASS SPILLED= | .22547 | TOTAL MASS SPILLED UPTO XP= | .01129 |
| BODY | STATION | NO.= | 40 | XP= | .132791 | LOCAL MASS SPILLED= | .22048 | TOTAL MASS SPILLED UPTO XP= | .01193 |
| BODY | STATION | NO.= | 41 | XP= | .135677 | LOCAL MASS SPILLED= | .21061 | TOTAL MASS SPILLED UPTO XP= | .01256 |
| BODY | STATION | NO.= | 42 | XP= | .138564 | LOCAL MASS SPILLED= | .19843 | TOTAL MASS SPILLED UPTO XP= | .01315 |
| BODY | STATION | NO.= | 43 | XP= | .141451 | LOCAL MASS SPILLED= | .18332 | TOTAL MASS SPILLED UPTO XP= | .01370 |

## Pressure Ratio Output on Lower and Upper Boundaries

| N | XP | CENTER LINE PR. RATIO (M = 1) | SIDEWALL PR. RATIO (M = M1) |
|---|---|---|---|
| 1 | .00000 | 1.0000 | 1.0000 |
| 2 | .00577 | 1.0070 | 1.7647 |
| 3 | .01155 | 1.0061 | 2.2301 |
| 4 | .01732 | 1.0032 | 2.2139 |
| 5 | .02309 | 1.0109 | 2.1625 |
| 6 | .02887 | 1.0011 | 2.1627 |
| 7 | .03406 | 1.0125 | 2.1768 |
| 8 | .03868 | 1.0005 | 2.1529 |
| 9 | .04272 | 1.0155 | 2.1663 |
| 10 | .04619 | .9919 | 2.1424 |
| 11 | .04907 | 1.0223 | 2.1016 |
| 12 | .05196 | .9938 | 2.0947 |
| 13 | .05485 | 1.0200 | 2.0974 |
| 14 | .05774 | .9956 | 2.1517 |
| 15 | .06062 | 1.0079 | 2.1632 |
| 16 | .06351 | 1.0029 | 2.1563 |
| 17 | .06640 | .9973 | 2.1223 |
| 18 | .06928 | .9949 | 2.1324 |
| 19 | .07217 | 1.0030 | 2.0776 |
| 20 | .07506 | .9906 | 2.1131 |
| 21 | .07794 | .9900 | 2.0951 |
| 22 | .08083 | .9968 | 2.1114 |
| 23 | .08372 | 1.0530 | 2.1145 |
| 24 | .08660 | 2.1081 | 2.1314 |
| 25 | .08949 | 2.6353 | 2.1408 |

## APPENDIX E

### Pressure Ratio Output on Lower and Upper Surfaces of Strut

The pressure ratio output on the lower and upper surfaces of the strut is printed only for NSTRUT = 1.

| N | YP | STRUT UPPER SURFACE PR. RATIO | STRUT LOWER SURFACE PR. RATIO |
|---|---|---|---|
| 16 | .06351 | 1.4932 | 2.1150 |
| 17 | .06640 | 1.2633 | 2.6808 |
| 18 | .06928 | 1.1928 | 2.6062 |
| 19 | .07217 | 1.1509 | 2.6132 |
| 20 | .07506 | 1.1512 | 2.5679 |
| 21 | .07794 | 1.1044 | 2.4708 |
| 22 | .08083 | 1.2087 | 2.5546 |
| 23 | .08372 | 1.2125 | 2.5998 |
| 24 | .08660 | 1.2956 | 2.5404 |
| 25 | .08949 | 1.9811 | 2.5672 |
| 26 | .09238 | 3.1793 | 2.5095 |
| 27 | .09526 | 4.2971 | 2.4473 |
| 28 | .09815 | 4.9100 | 2.5659 |
| 29 | .10104 | 5.0094 | 2.5440 |
| 30 | .10392 | 4.9075 | 2.5335 |
| 31 | .10681 | 4.6010 | 2.5011 |
| 32 | .10970 | 4.3770 | 2.4702 |
| 33 | .11258 | 4.1791 | 2.5526 |
| 34 | .11547 | 3.9674 | 1.7000 |
| 35 | .11836 | 4.1153 | .9323 |
| 36 | .12124 | 3.5351 | .7954 |

### Program Termination Message

PROGRAM TERMINATED ON PHYSICAL TIME CONVERGENCE CRITERION

## REFERENCES


1. Kumar, Ajay: Numerical Analysis of the Scramjet Inlet Flow Field Using Two-Dimensional Navier-Stokes Equations. AIAA-81-0185, Jan. 1981.

2. Kumar, Ajay: Numerical Analysis of the Scramjet-Inlet Flow Field by Using Two-Dimensional Navier-Stokes Equations. NASA TP-1940, 1981.

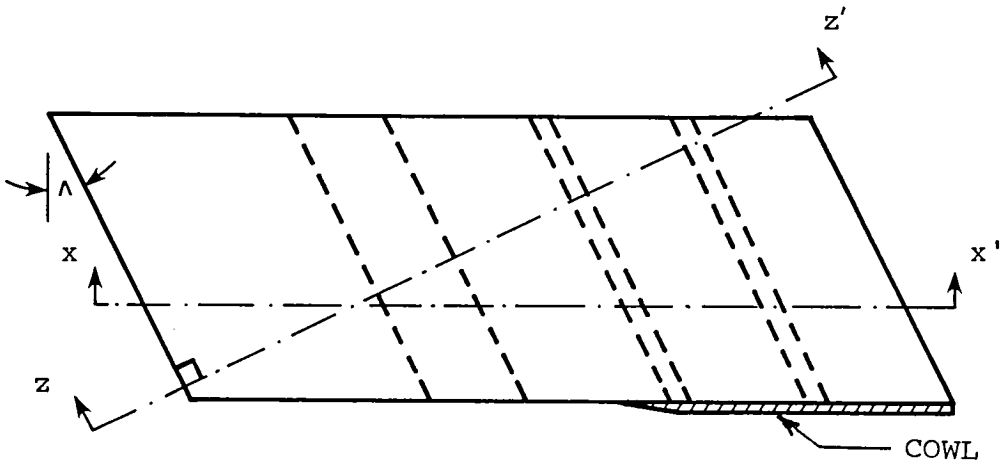
3. Baldwin, Barrett; and Lomax, Harvard: Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows: AIAA Paper 78-257, Jan. 1978.

4. Smith, Robert E.; and Weigel, Barbara L.: Analytic and Approximate Boundary-Fitted Coordinate Systems for Fluid Flow Simulation. AIAA-80-0192, Jan. 1980.

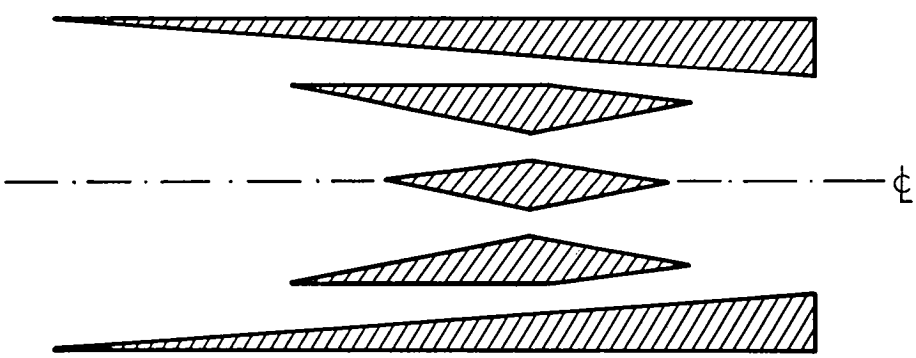
5. MacCormack, Robert W.: The Effect of Viscosity in Hypervelocity Impact Cratering. AIAA Paper No. 69-354, Apr.-May 1969.

6. MacCormack, R. W.: A Numerical Method for Solving the Equations of Compressible Viscous Flow. AIAA-81-0110, Jan. 1981.

7. Kumar, Ajay: Some Observations on a New Numerical Method for Solving the Navier-Stokes Equations. NASA TP-1934, 1981.

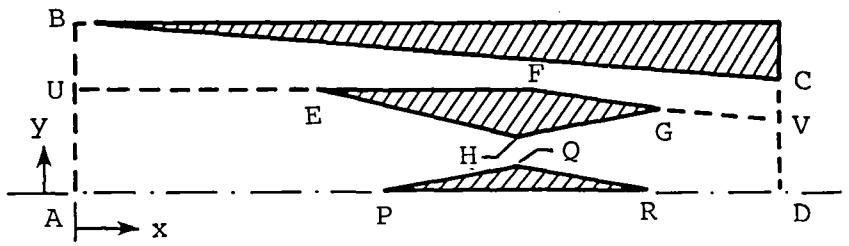



(a) Side view of a scramjet-inlet module.

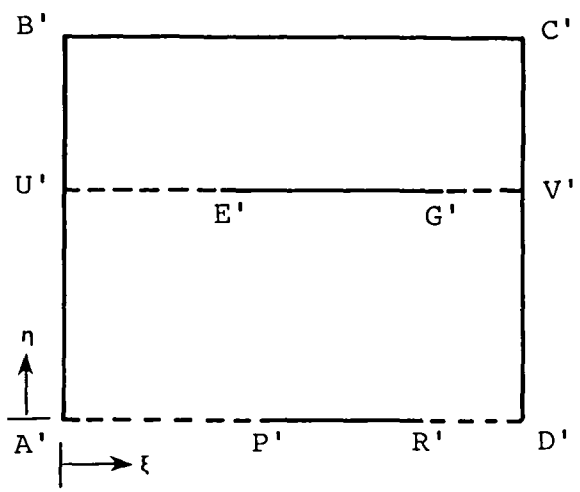



(b) Geometry in plane xx'.

Figure 1.- Geometry of sample problem.




(c) Upper half of geometry in plane zz' (physical plane).




(d) Upper half of geometry in plane zz' (transformed plane).

Figure 1.- Concluded.

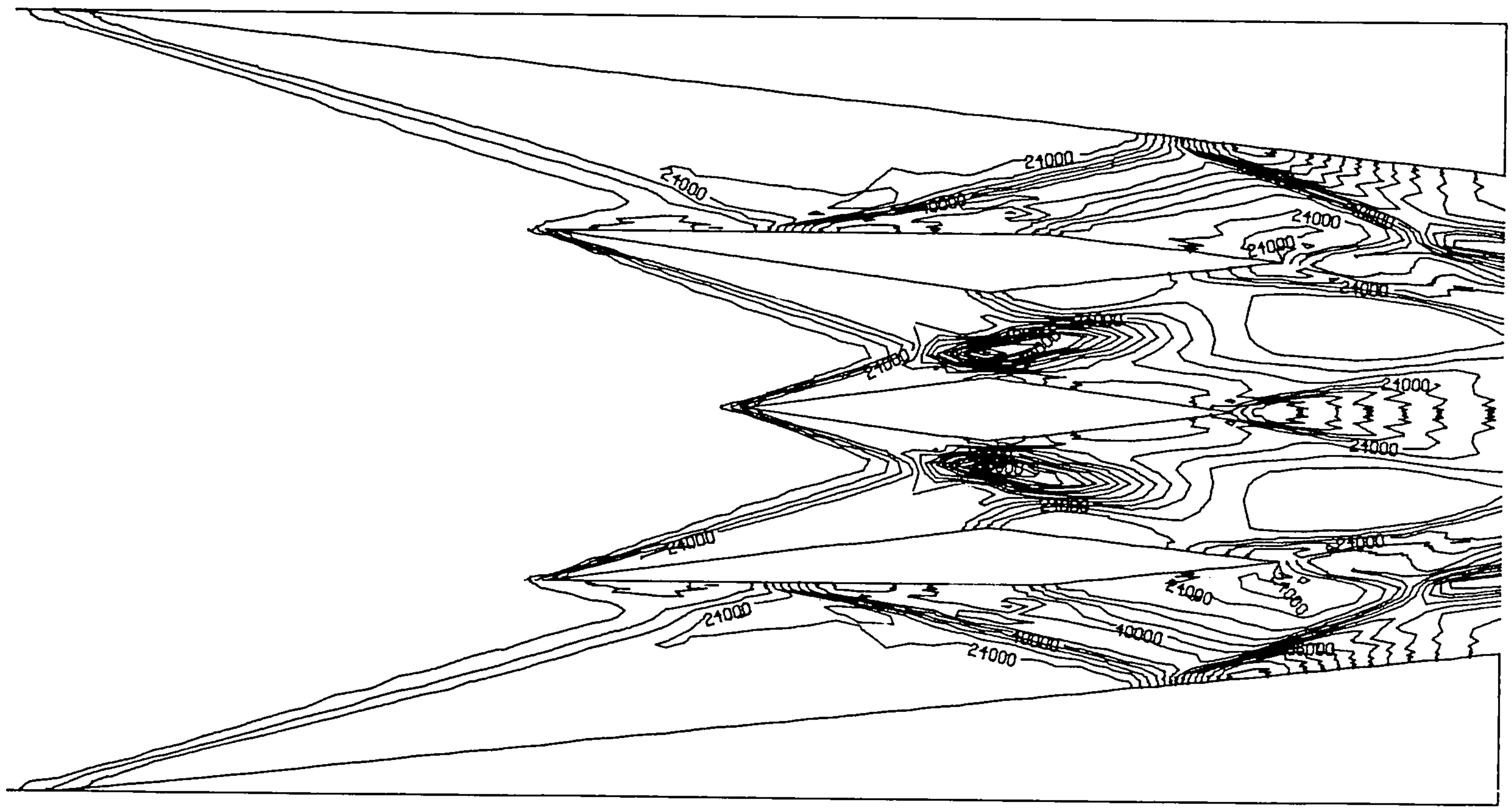



Figure 2.- Pressure contour plot for sample problem.

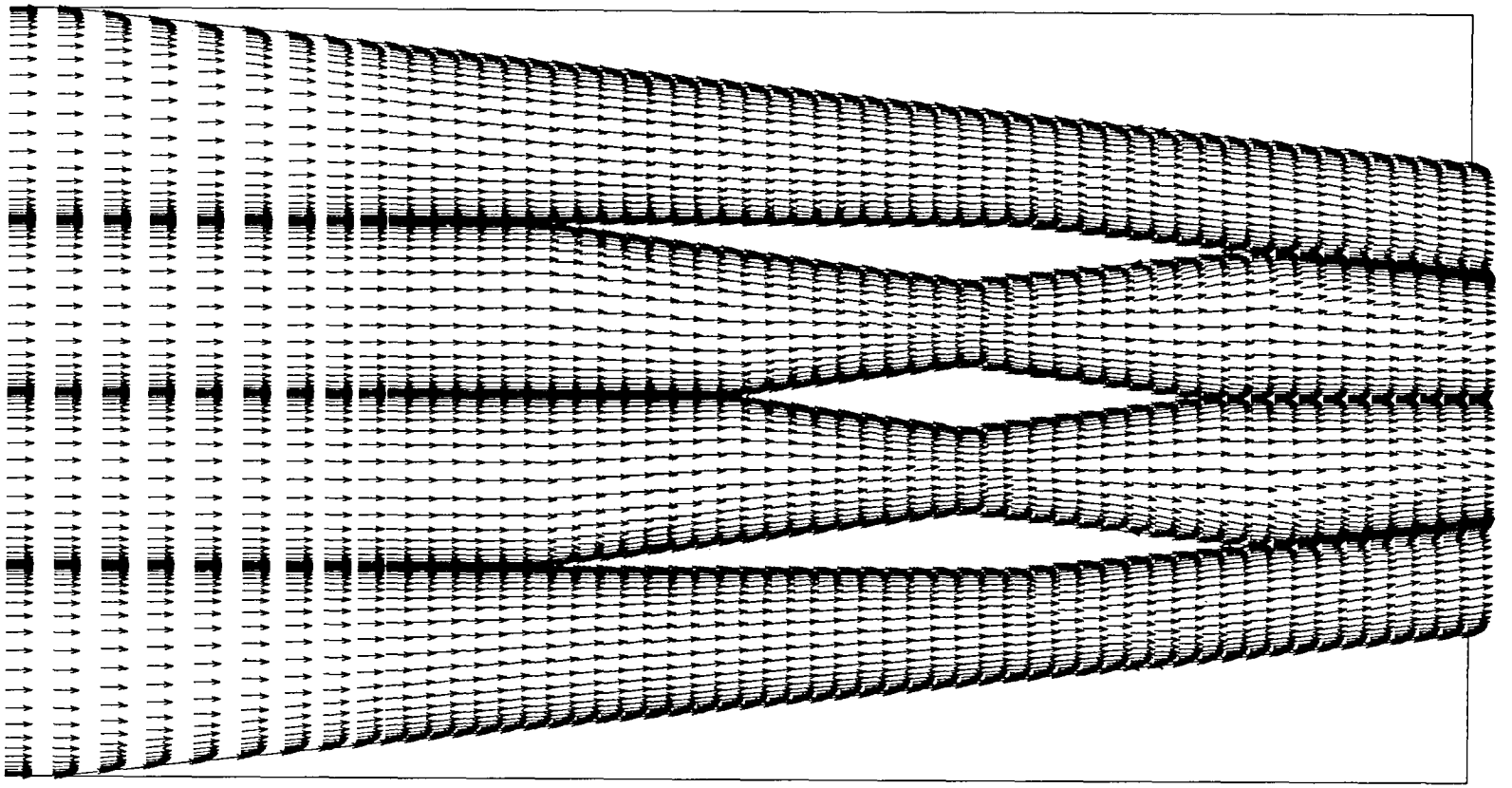


Figure 3.- Velocity vector field for sample problem.

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| NASA TM-85708 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| USER'S GUIDE FOR NASCRIN - A VECTORIZED CODE FOR CALCULATING TWO-DIMENSIONAL SUPERSONIC INTERNAL FLOW FIELDS | February 1984 |
| | 6. Performing Organization Code<br>505-43-83-06 |
| 7. Author(s)<br>Ajay Kumar | 8. Performing Organization Report No.<br>L-15678 |
| 9. Performing Organization Name and Address<br>NASA Langley Research Center<br>Hampton, VA 23665 | 10. Work Unit No. |
| | 11. Contract or Grant No. |
| 12. Sponsoring Agency Name and Address<br>National Aeronautics and Space Administration<br>Washington, DC 20546 | 13. Type of Report and Period Covered<br>Technical Memorandum |
| | 14. Sponsoring Agency Code |

15. Supplementary Notes

16. Abstract

A computer program NASCRIN has been developed for analyzing two-dimensional flow fields in high-speed inlets. It solves the two-dimensional Euler or Navier-Stokes equations in conservation form by an explicit, two-step finite-difference method. An explicit-implicit method can also be used at the user's discretion for viscous flow calculations. For turbulent flow, an algebraic, two-layer eddy-viscosity model is used. The code is operational on the CDC® CYBER 203 computer system and is highly vectorized to take full advantage of the vector-processing capability of the system. It is highly user oriented and is structured in such a way that for most supersonic flow problems, the user has to make only a few changes. Although the code is primarily written for supersonic internal flow, it can be used with suitable changes in the boundary conditions for a variety of other problems.

| 17. Key Words (Suggested by Author(s)) | 18. Distribution Statement |
|---|---|
| Scramjet inlet<br>Explicit-implicit method<br>Navier-Stokes equations<br>Euler equations | [redacted]<br>Subject Category 02 |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 68 | |

Available: NASA's Industrial Applications Centers

NASA-Langley, 1984

LANGLEY RESEARCH CENTER
3 1176 00512 0739