

LA-11009-MS

**DO NOT CIRCULATE
PERMANENT RETENTION
REQUIRED BY CONTRACT**

*N451-FOR3D: A Three-Dimensional
Computer Program for Incompressible
Flows with Free Surfaces*



Los Alamos National Laboratory
Los Alamos, New Mexico 87545

This work was supported by the U.S. Department of Energy and the National Atmospheric and Space Administration's Langley Research Center.

Proposed by: Jennifer Shaefer (Group 1, 3)

200

the first hours of the morning, the sun was still low in the sky, and the temperature was still below freezing. The ground was covered with a thin layer of snow, and the trees were bare. The air was cold and crisp, and the sound of the wind was the only noise.

LA11009-MS

UC-32

Issued: July 1987

NASA-VOF3D: A Three-Dimensional Computer Program for Incompressible Flows with Free Surfaces

Martin D. Torrey
Raymond C. Mjolsness
Leland R. Stein

SCANNED APR 17 1998



LOS ALAMOS NATIONAL LABORATORY

3 9338 00123 6289



Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

TABLE OF CONTENTS

ABSTRACT	1
I. INTRODUCTION	1
II. THE NUMERICAL METHOD	4
A. The Computational Mesh	4
B. Boundary Conditions	6
C. Initial Conditions	6
D. Surface Physics	7
1. Orientation of the Surface	7
2. Surface Tension	10
3. Wall Adhesion	18
E. Incrementing the Variables	19
1. Provisional (TILDE) Velocities	19
2. SOR Option for Pressures and Velocities	20
3. CR Option for Pressures and Velocities	21
4. Pressures and Velocities for Surface Cells	21
5. Incrementing F Values	22
6. Time Step Limitations	23
III. PROGRAM DESCRIPTION	24
A. Program Architecture	24
1. Description of the Flow Chart	25
2. Subroutines and their Functions	29
B. Communications Among Subroutines	32
1. Common Variables	33
2. Common Arrays	36
C. Problem Input	39
1. NAMELIST XPUT	39

2.	NAMELIST MSHGN	41
3.	NAMELIST GRAFIC	44
4.	NAMELIST CONTUR	46
5.	NAMELIST FLUIDGN	48
D.	User Options	49
E.	Example of Problem Setup	50
IV.	EXAMPLES OF SURFACE TENSION CALCULATIONS	66
V.	SAMPLE APPLICATIONS	68
VI.	FURTHER DEVELOPMENTS	90
ACKNOWLEDGMENT		91
REFERENCES		92
APPENDIX. PROGRAM LISTING		93

**NASA-VOF3D: A THREE-DIMENSIONAL COMPUTER PROGRAM
FOR INCOMPRESSIBLE FLOWS WITH FREE SURFACES***

by

Martin D. Torrey, Raymond C. Mjolsness, and Leland R. Stein

ABSTRACT

We present the NASA-VOF3D three-dimensional, transient, free-surface hydrodynamics program. This three-dimensional extension of NASA-VOF2D will, in principle, permit treatment in full three-dimensional generality of the wide variety of applications that could be treated by NASA-VOF2D only within the two-dimensional idealization. In particular, it, like NASA-VOF2D, is specifically designed to calculate confined flows in a low g environment. The code is presently restricted to cylindrical geometry. The code is based on the fractional volume-of-fluid method and allows multiple free surfaces with surface tension and wall adhesion. It also has a partial cell treatment that allows curved boundaries and internal obstacles. This report provides a brief discussion of the numerical method, a code listing, and some sample problems.

I. INTRODUCTION

The NASA-VOF3D program simulates three-dimensional incompressible flows with free surfaces using the volume-of-fluid (VOF) algorithm.¹⁻³ It is specifically designed to calculate flows in a low g environment, in which surface physics must be accurately treated. This technique is based on the use of donor-acceptor differencing⁴ to track the free surface across an Eulerian grid. Similar procedures were developed contemporarily by other workers⁵⁻⁹ to track material interfaces as well as free surfaces. A similar, but 2D, program has been written specifically for simulating the draining of spacecraft propellant tanks.¹⁰ Both the NASA-VOF2D and the NASA-VOF3D programs contain several improvements over these codes, and these improvements will be described along with the governing equations, the numerical algorithm, a program description and several numerical solutions. A program

*The code is available through the National Energy Software Center, 9700 South Cass Avenue, Argonne, IL 60439, telephone (312/972-7250).

listing will be presented. Because much of the numerical methodology is very similar in 2D and in 3D, we will frequently refer to the NASA-VOF2D report¹¹ for a fuller description of topics mentioned here.

In particular, Reference 11 discusses the developments leading to the use of SOLA-VOF methodology in low g flows. We have incorporated the NASA-VOF2D improvements in donor-acceptor differencing, and in the inclusion of a conjugate residual option as well as a successive-over-relaxation (SOR) option for advancing the pressures (and velocities) in time throughout the computing mesh. The conjugate residual method is vectorized for the Cray and employs a scaled coefficient matrix, as in NASA-VOF2D. We found it necessary for accuracy to radically alter the procedure for calculating 3D surface tension effects. The choice of surface orientation, greatly generalized from the 2D procedure, may require further attention. We have not included the 3D extension of our 2D wall-adhesion procedure for curved boundaries; this must be done to obtain a truly general-purpose code.

The basic algorithm is based on the Navier-Stokes equations for an incompressible fluid, which we record here in cylindrical coordinates only. We introduce:

x, θ, z for the radial, azimuthal, and axial coordinates

u, v, w for the corresponding velocity components

g_x, g_y, g_z , for the corresponding components of external acceleration

p, ρ , and Θ for the pressure, density, and partial cell parameter.¹¹

Then the continuity equation is

$$\Delta \cdot (\Theta \mathbf{u}) = \frac{1}{x} \frac{\partial(x\Theta u)}{\partial x} + \frac{1}{x} \frac{\partial(\Theta v)}{\partial \theta} + \frac{\partial(\Theta w)}{\partial z} = 0 , \quad (1)$$

and the momentum equations are

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{v}{x} \frac{\partial u}{\partial \theta} + w \frac{\partial u}{\partial z} - \frac{v^2}{x} \\ = g_x - \frac{1}{\rho} \frac{\partial p}{\partial x} + v \left| \frac{\partial^2 u}{\partial x^2} + \frac{1}{x} \frac{\partial u}{\partial x} - \frac{1}{x^2} u + \frac{1}{x^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial z^2} - \frac{2}{x^2} \frac{\partial v}{\partial \theta} \right| , \end{aligned} \quad (2)$$

$$\begin{aligned}
& \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + \frac{v}{x} \frac{\partial v}{\partial \theta} + w \frac{\partial v}{\partial z} + \frac{1}{x} u v \\
&= g_y - \frac{1}{\rho x} \frac{\partial p}{\partial \theta} + v \left[\frac{\partial^2 v}{\partial x^2} + \frac{1}{x} \frac{\partial v}{\partial x} - \frac{1}{x^2} v + \frac{1}{x^2} \frac{\partial^2 v}{\partial \theta^2} + \frac{\partial^2 v}{\partial z^2} + \frac{2}{x^2} \frac{\partial u}{\partial \theta} \right], \tag{3}
\end{aligned}$$

and

$$\begin{aligned}
& \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + \frac{v}{x} \frac{\partial w}{\partial \theta} + w \frac{\partial}{\partial z} w \\
&= g_z - \frac{1}{\rho} \frac{\partial p}{\partial z} + v \left[\frac{\partial^2 w}{\partial x^2} + \frac{1}{x} \frac{\partial w}{\partial x} + \frac{1}{x^2} \frac{\partial^2 w}{\partial \theta^2} + \frac{\partial^2 w}{\partial z^2} \right], \tag{4}
\end{aligned}$$

where v is the kinematic viscosity. These equations apply at every point inside the fluid. When implemented numerically, all equations use the arc length azimuthal coordinate $y=x_{IM1}\theta$ rather than θ , where x_{IM1} is the radius of the computational mesh; this emphasizes the similarity of our equations to Cartesian equations.

The free surfaces are treated by introducing a function $F(x, \theta, z, t)$ that is defined to be unity at any point occupied by the fluid and zero elsewhere. When averaged over a cell of the computing mesh, $\langle F \rangle_{cell}$ is the fractional volume of the cell occupied by fluid. Thus, $\langle F \rangle_{cell}=1$ implies a cell full of fluid, while $\langle F \rangle_{cell}=0$ denotes an empty cell. Finite-difference solution of the governing equations for F , (Eq. 5), gives $\langle F \rangle_{cell}$ directly. Cells with F values between zero and one are partially filled with fluid; they are either intersected by a free surface or contain voids (bubbles) smaller than cell mesh dimensions. The F function is used to determine which cells contain a boundary and where the fluid is located in the cells. Gradients of F determine the mean local surface normal, and, together with F values, permit construction of an approximate interface. Surface-tension and wall-adhesion forces may then be evaluated. The special requirements for construction of gradients of the step function F are discussed in Reference 11. The governing equation for F is

$$\frac{\partial F}{\partial t} + u \frac{\partial F}{\partial x} + \frac{v}{x} \frac{\partial F}{\partial \theta} + w \frac{\partial F}{\partial z} = 0 . \quad (5)$$

This approach allows the fluid interfaces to be represented simply via different equations. However, the flux of F through each face of an Eulerian grid will be required. Standard finite-difference techniques would lead to a smearing of F values, degrading the interface definition. Using the step function character of F , we employ a form of donor-acceptor differencing that preserves the discontinuous nature of F .

Section II describes the numerical method briefly, while Sec. III describes the construction of the program. In Sec. IV we present examples of the performance of the surface tension algorithm, while in Sec. V we present runs of an (initially) spherical bubble moving in an ellipsoidal tank. In Sec. VI we summarize what may be done to obtain from NASA-VOF3D a program of more general applicability for low g flows, but emphasize that the present form of the program has a very significant range of applicability. Finally, in Appendix A, we present a program listing.

II. THE NUMERICAL METHOD

This section provides brief descriptions of many features of the numerical method. Those features that are very similar to the 2D method are discussed at greater length in Reference 11. The program is Eulerian and allows for an arbitrary number of segments of free surface with any reasonable shape. Variable mesh spacing is permitted in all three coordinate directions, as is necessary for efficient numerical solution of many problems. The complete Navier-Stokes equations for an incompressible fluid are solved by finite differences, with surface-tension effects included. Wall adhesion may be included or neglected via a user option.

A. The Computational Mesh

The mesh is very similar to the 2D mesh of Reference 11. But note that the y coordinate of Reference 11 is our z coordinate. Thus $(\delta y_j)_{2D} \leftrightarrow (\delta z_k)_{3D}$, and δy_j for us denotes an increment in the azimuthal coordinate. There is a plane of fictitious cells on each boundary of the computational mesh and δx_i , δy_j , and δz_k denote the coordinate increments in the (i, j, k) cell. Thus

$$i = 1, i = IMAX$$

refers to non-physical cells and

$$i = 2, i = IM1 (\equiv IMAX - 1)$$

refers to the first and last physical cells in the x direction. Of course, because of the code's capability to treat curved boundaries, we mean only that $i = IM1$ is a physical cell for some value of j and k . Then

$$x_{i+1/2} = \sum_{i'=2}^i \Delta x_{i'}, \quad 2 \leq i \leq IM1$$

$$y_{j+1/2} = \sum_{j'=2}^j \Delta y_{j'}, \quad 2 \leq j \leq JM1$$

$$z_{k+1/2} = \sum_{k'=2}^k \Delta z_{k'}, \quad 2 \leq k \leq KM1$$

denote coordinate values on the right, back, and top faces (faces of maximal coordinate value) of the (i, j, k) cell and x_i, y_j , and z_k denote coordinates at cell center, e.g.,

$$x_i = \frac{1}{2} (x_{i+1/2} + x_{i-1/2}).$$

We emphasize the program will readily accept variable mesh spacing, i.e., Δx_i may depend on i , Δy_j on j and Δz_k on k , subject only to reasonable requirements on avoiding neighboring cells of grossly different size and avoiding cells of large aspect ratio.¹¹

The pressure $p_{i,j,k}$, fractional volume of fluid $F_{i,j,k}$, and fraction of the cell volume open to flow $AC_{i,j,k}$ are cell-centered quantities, while the velocities $u_{i+1/2,j,k}, v_{i,j+1/2,k}$, and $w_{i,j,k+1/2}$ and the fractional areas open to flow $AR_{i+1/2,j,k}, ABK_{i,j+1/2,k}$, and $AT_{i,j,k+1/2}$ are face-centered quantities, e.g., $v_{i,j+1/2,k}$, and $ABK_{i,j+1/2,k}$, are defined at $(x_i, y_{j+1/2}, z_k)$. The fractional areas and volumes open to

flow are related to Θ of the continuity equation as described in Reference 11 (with the obvious extension to 3D geometry), e.g.,

$$ABK_{i,j+1/2,k} = \Theta_{i,j+1/2,k}$$

Interior obstacles are defined as any nonflow region within the computational mesh; they are specified by a series of 2D conic sections (in x, z plane)

$$0(x,z) = a_2 x^2 + a_1 x + b_2 z^2 + b_1 z + c_2 x z + c_1 = 0 ,$$

such that some portion of each locus forms a portion of the obstacle, and by a parameter that flags a cell overlapping an obstacle as closed to flow, partially closed to flow, or open to flow. We emphasize that only cylindrical obstacles can be treated by the present version of the code. The operation of the obstacle setting procedure is described in Reference 11. With the obstacles specified, the fractional areas and volumes open to flow, e.g., ABK and AC, are specified as in 2D.¹¹

B. Boundary Conditions

The boundary conditions are imposed in a standard way on the six boundaries (left, right, forward, back, bottom, and top) of the computational mesh by six flags WL, WR, WF, WBK, WB, and WT. Integer numerical values for the flags are as follows: 1 denotes rigid free-slip wall, 2 denotes rigid no-slip wall, 3 denotes continuative boundary, 4 denotes periodic boundary (this value must appear on both flags for two opposite boundaries), and 5 denotes specified pressure outflow boundary. Details on implementation are discussed in the SOLA-VOF report.³ Other types of boundary conditions, if needed, must be supplied by the user; and they will typically be placed at the end of SUBROUTINE BC. An example of this (a tank-drain hole) will be displayed in Sec. III.

C. Initial Conditions

The program specifies the location of obstacles, and the initial locations of fluid, void, and fluid-void interfaces. When appropriate, a fluid-void interface may be calculated as an equilibrium surface; otherwise, its shape must be specified by the user in advance, and must conform to the geometric capabilities of the code,

discussed for obstacle generation in Sec. II.A. The machinery for accomplishing all this is a direct 3D extension of our 2D capabilities.¹¹

The initial fluid velocities are taken to be constant or zero. It would, in principle, be possible to take the fluid to be initially in nonuniform motion; the appropriate place for the user to insert such changes would be in SUBROUTINE SETUP. However, it is very rare for a dynamic velocity field to be known with sufficient precision to be suitable as an initial condition, except when it is obtained from some other numerical simulation. Thus, it is likely that the user would wish to start an initially dynamic configuration from a tape dump rather than to build the initial conditions into the program's subroutines.

While the pressure field is initially zeroed, this command is overridden to produce the hydrostatic head as the system's initial pressure field. It is important that this more accurate initial pressure field be employed; problems with zero initial pressure frequently have difficulty. As with the initial velocity fields, it will frequently be more useful to pick up the more intricate pressure field of an initially dynamic configuration from a tape dump, rather than building into the program's subroutines.

D. Surface Physics

All the surface physics algorithms must be improved and generalized considerably from their 2D analogs and from the initial algorithms used in our 3D program; it is more difficult to calculate surface physics in 3D than in 2D. In 2D, one surface curvature may be calculated algebraically, and only one curvature need be calculated by finite-difference approximations. Moreover, the algebraic curvature is typically the larger one in the crucial near-axis cells. In 3D, both surface curvatures must be calculated by less accurate finite-difference approximations. We describe the 3D surface physics algorithms. While the radial coordinate is denoted by x elsewhere in the text and throughout the computer program, it is denoted by r in the following Secs. 1 and 2.

1. *Orientation of the Surface.* The basic orientation of a surface cell is described by the parameter (array) NF through its numerical values 1, 2, 3, 4, 5, or 6. The definition of numerical values for the array is specified in Table I. The differential geometry describing the surface curvatures depends strongly on the numerical value selected for NF. Inspection of Table I indicates that NF=1 or 2 corresponds to a surface defined by $r=R(\theta, z)$, and NF=3 or 4 to one defined by $\theta=\Phi(r, z)$, and NF=5 or 6 to one defined by $z=H(r, \theta)$; each surface description leads to a distinct

surface differential geometry. Moreover, with only imprecise numerical data available to the program, the value obtained for the local surface pressure, p_s , depends strongly on the numerical value selected for NF; it is important to make an appropriate choice for NF.

Nevertheless, it has not been possible to devise a completely satisfactory algorithm of general applicability for the choice of NF values. We summarize here the algorithms that are currently implemented in the code; they suffice to provide reasonable simulations for the problems to which the code has been applied. The algorithms are activated by the choice of numerical value for the input parameter NFCAL.

TABLE I
DEFINITION OF VALUES FOR NF ARRAY

NF = 0	fluid cell, i.e., contains fluid and has no void cell adjacent to any of its faces
NF = 1	surface cell, fluid surface most nearly parallel to cylinder $r=\text{constant}$, defined by $r=R(\theta, z)$, interpolation neighbor fluid cell is on the left
NF = 2	surface cell, fluid surface most nearly parallel to cylinder $r=\text{constant}$, defined by $r=R(\theta, z)$, interpolation neighbor fluid cell is on the right
NF = 3	surface cell, fluid surface most nearly parallel to plane $\theta=\text{constant}$, defined by $\theta=\Phi(r, z)$, interpolation neighbor fluid cell is to the front
NF = 4	surface cell, fluid surface is most nearly parallel to plane $\theta=\text{constant}$, defined by $\theta=\Phi(r, z)$, interpolation neighbor fluid cell is to the back
NF = 5	surface cell, fluid surface most nearly parallel to plane $z=\text{constant}$, defined by $z=H(r, \theta)$, interpolation neighbor fluid cell is to the bottom
NF = 6	surface cell, fluid surface most nearly parallel to plane $z=\text{constant}$, defined by $z=H(r, \theta)$, interpolation neighbor fluid cell is to the top
NF = 7	isolated cell, contains fluid but all cells adjacent to one of its faces are void
NF = 8	void cell, contains no fluid.

We calculate a preliminary value of NF, which is needed for the earliest uses of NF in the computational cycle. This provisional NF value is determined from the "wettest surface" of the computational cycle by the same numerical ordering as described for the "interpolation neighbor fluid cell" in **Table I**. For example, if the "wettest surface" is to the left of the computational cell the provisional algorithm chooses NF = 1, if to the right it chooses NF = 2, etc. The "wettest surface" of the computational cell is obtained by adding the fractional fluid occupancies, i.e., by summing the F values, in the 9 adjacent cells (in a 3×3 array) centered on each one of the computational cells' surfaces. For example, if the computational cell had the cell index (i,j,k) , the 9 adjacent cells "to the left" would be labeled by $(i-1,j-1,k-1)$, $(i-1,j,k-1)$, $(i-1,j+1,k-1)$, ..., $(i-1,j+1,k+1)$, corresponding to the interface between (i,j,k) and $(i-1,j,k)$. The "wettest surface" is chosen to be that surface with the largest total fluid occupancy. The provisional set of NF values is updated each cycle, after a new array of F values becomes available. By specifying NFCAL = 1, one may require that this provisional set of NF values also be adopted as the final set of NF values to be used throughout the calculational cycle.

A second set of "slope" NF values is also calculated. The "slope" NF values choose the basic surface mathematical description, and thus the basic surface differential geometry, by picking the surface description that yields a surface normal with minimum angle with respect to the normal of the corresponding coordinate surface, i.e., the surface is "most nearly parallel to" that coordinate surface. This limits the NF values to one of two possible numerical values. The choice of NF is made definite by choosing from the two adjacent cells sharing the preferred coordinate surface, that cell with the largest F value, i.e., containing the most fluid, to be the interpolation neighbor fluid cell. For example, if the "slope" value picks NF = 1 for the computational cell (i,j,k) , then (a) defining the surface by $r = R(\theta,z)$ yields a surface normal \hat{n} more clearly parallel to the normal \hat{r} of the coordinate surface $r = \text{constant}$ than would follow from one of the other modes of surface description, and (b) $F_{i-1jk} > F_{i+1jk}$, so it is reasonable to use cell $(i-1,j,k)$ as the "base" of the surface, i.e., as the interpolation neighbor fluid cell. By specifying NFCAL = 2, one may require that the "slope" NF values are adopted as the final set of NF values.

Finally, by specifying NFCAL = 3, one may adopt a mixed strategy for choosing between the provisional NF values and the "slope" NF values that has proven to be adequate for the problems to which the program has been applied. With NFCAL = 3, one uses the provisional NF values, except a) if there is an obstacle cell

adjacent to the computational cell (i,j,k) in the direction indicated by the "slope" value for NF, i.e., if when the "slope" $NF = 1$ or 2 either of the cells $(i-1,j,k)$ or $(i+1,j,k)$ is an obstacle cell the "slope" value of NF is used, or (b) if all the previously computed part of the array of NF has a value of NF in those of the 16 ($= 3 \times 3 + 3 \times 2 + 1$) cells "adjoining" the computational cell that are surface cells that agrees with the "slope" NF value for the computational cell, then the "slope" NF value is used. To partially offset the occasional failure of even the $NFCAL=3$ algorithm, and for another reason, we introduce a postprocessing algorithm that "clips" anomalous values of p_s , as explained in detail later.

2. Surface-Tension. Considerable refinement of our original 3D surface tension algorithms was necessary in order to produce algorithms that give accurate values for p_s . We first record here the differential geometric expressions for p_s arising for the three basic types of surface orientation mentioned earlier. We consider the (i,j,k) computational cell to be bounded by

$$r_I \leq r \leq r_O$$

$$\theta_W \leq \theta \leq \theta_E$$

$$z_S \leq z \leq z_N ,$$

with mean coordinate values

$$r_m = 1/2(r_I + r_O)$$

$$\theta_m = 1/2(\theta_W + \theta_E)$$

$$z_m = 1/2(z_S + z_N) ,$$

and coordinate differences

$$\Delta r = (r_O - r_I)$$

$$\Delta\theta = (\theta_E - \theta_W)$$

$$\Delta z = (z_N - z_S)$$

The surface-tension of the fluid-void interface is denoted by σ . It should be pointed out that the present program actually works with the reduced pressure $\phi = pp^{-1}$, even though the reduced pressure is still denoted p . Thus, the σ , as implemented in the program, actually denotes σp^{-1} . Similarly, densities do not appear explicitly in places, e.g., the calculation of provisional, or TILDE, velocities where one would expect to find the factor pp^{-1} .

When the fluid-void interface is described locally by a surface of the form $r=R(\theta, z)$ (so that $NF=1$ or 2) we obtain a particular geometric form for p_s in a surface computational cell which we display for the case $NF=1$, namely

$$\begin{aligned}
 p_s(i,j,k) = & -\sigma \left\{ \frac{1}{\Delta z} \left[\left(\frac{R_z}{D} \right)_{r=R(\theta_m, z_N)} - \left(\frac{R_z}{D} \right)_{r=R(\theta_m, z_S)} \right] \right. \\
 & - \frac{1}{2} \left[\left(\frac{1+R_z^2}{rD} \right)_{r=R(\theta_E, z_m)} + \left(\frac{1+R_z^2}{rD} \right)_{r=R(\theta_W, z_m)} \right] \\
 & \left. + \frac{1}{2R(\theta_m, z_m)} \operatorname{ctn} \left(\frac{\Delta\theta}{2} \right) \left[\left(\frac{R_\theta}{rD} \right)_{r=R(\theta_E, z_m)} - \left(\frac{R_\theta}{rD} \right)_{r=R(\theta_W, z_m)} \right] \right\}, \tag{6}
 \end{aligned}$$

where

$$D = \left(1 + \frac{1}{r^2} R_\theta^2 + R_z^2 \right)^{\frac{1}{2}},$$

and R_θ, R_z are the gradients of the radial coordinate describing the surface. Similarly, we display the expression for p_s in the NF=3 case of the surface definition $\theta = \Phi(r, z)$ as

$$\begin{aligned}
p_s(i,j,k) = & -\sigma \left\{ \frac{1}{\Delta z} \left| \left(\frac{r\Phi_z}{D} \right)_{r=r_m} \cos \delta_N - \left(\frac{r^2\Phi_r\Phi_z}{D} \right)_{r=r_m} \sin \delta_N \right. \right. \\
& \quad \left. \left. \begin{array}{ll} z=z_N & z=z_N \end{array} \right. \right. \\
& - \left(\frac{r\Phi_z}{D} \right)_{r=r_m} \cos \delta_S + \left(\frac{r^2\Phi_r\Phi_z}{D} \right)_{r=r_m} \sin \delta_S \Big| \\
& \quad \left. \begin{array}{ll} z=z_S & z=z_S \end{array} \right. \\
& + \frac{1}{\Delta r} \left[\left(\frac{r\Phi_r}{D} \right)_{r=r_O} \cos \delta_E + \left(\frac{1+r^2\Phi_z^2}{D} \right)_{r=r_O} \sin \delta_E \right. \\
& \quad \left. \begin{array}{ll} z=z_m & z=z_m \end{array} \right. \\
& - \left. \left. \left(\frac{r\Phi_r}{D} \right)_{r=r_I} \cos \delta_W - \left(\frac{1+r^2\Phi_z^2}{D} \right)_{r=r_I} \sin \delta_W \right] \right\} , \tag{7}
\end{aligned}$$

where

$$D = \left(1 + r^2 [\Phi_r^2 + \Phi_z^2] \right)^{\frac{1}{2}},$$

and, if $\hat{r}(\theta)$, $\hat{\theta}(\theta)$ denote the radial and azimuthal unit vectors evaluated at angle θ , then at the north face of the computational cell, i.e., at $z = z_N$,

$$\hat{\theta}(\theta_N) \cdot \hat{\theta}(\theta_m) = \cos \delta_N,$$

$$\hat{r}(\theta_N) \cdot \hat{\theta}(\theta_m) = \sin \delta_N,$$

with

$$\theta_N = \Phi(r_m, z_N) ,$$

and similar expressions hold for the other angles δ evaluated at the S, E, and W faces of the computational cell. Finally, we display p_s in the NF = 5 case of the surface definition $z = H(r, \theta)$ as

$$p_s(i,j,k) = - \frac{\sigma}{r_m} \left\{ \frac{1}{\Delta r} \left[\left(\frac{rH_r}{D} \right)_{r=r_O} - \left(\frac{rH_r}{D} \right)_{r=r_I} \right] + \frac{1}{\Delta \theta} \left[\left(\frac{H_\theta}{rD} \right)_{r=r_m} - \left(\frac{H_\theta}{rD} \right)_{r=r_E} \right] \right\} ,$$

$$\theta=\theta_m \quad \theta=\theta_m \quad \theta=\theta_E \quad \theta=\theta_W$$
(8)

where

$$D = \left(1 + H_r^2 + \frac{1}{r^2} H_\theta^2 \right)^{\frac{1}{2}} .$$

In Eqs. (7) and (8), Φ_r , Φ_z , H_r , and H_θ denote the gradients of the coordinate defining the surface.

In NASA-VOF2D and in earlier versions of our 3D program, the height functions defining the surface locally are computed from three cell summation formulas which convert the information available to the program, namely values of F in an array of cells near the computational cell, to a reconstruction of the local fluid-void interface. When this is done using reasonable zoning for the computational mesh and Eqs. (6)-(8) for p_s it is fairly common to obtain rather poor values for p_s . The problem is that regions in which the surface is nearly parallel to a particular height function are very poorly represented by the three-cell formula. Thus, in the interior of the computational mesh, we have defined the height functions by a five-cell summation formula which evolves smoothly (via a four-point summation formula) to the old three-cell formula still used at the edge of the mesh. For example, in the mesh interior we compute H for cell (i,j,k) from

$$H(i,j,k) = \sum_{k'=k-2}^{k+2} \left(1.0 + AC_{i,j,k'} * (F(i,j,k') - 1.0) \right) * \Delta Z_{k'} * FOF_{k'} \quad (9)$$

where $FOF_{k'}$ is a cut-off factor which excludes a cell from the sum when appropriate. While Eq. (9) and its analogs substantially solve the problem of inaccurate height functions due to geometric effects, it is still necessary to be sure that the zoning is appropriate to the problem at hand; even Eq. (9) can yield inaccurate p_s values locally if the zoning is too coarse.

In 2D and in the earlier 3D algorithms, gradients of height functions were evaluated from interpolation formulae which do not maintain accuracy when variable zoning is used. It is important to overcome this defect in 3D because: a) the 3D formulas are more sensitive to errors in gradients, b) variable zoning is essential for many 3D problems to reduce problem running time, and c) we have seen cases where the older 3D formulation led to very large errors in p_s even for a uniform mesh. For our present algorithms we consider 9 height functions defined in a 3×3 array of cells centered on the computational cell, and define gradients of the height function evaluated at the center of a face of the computational cell by either 6 or 7 point formulas which use the information contained in the 9 height functions to:

- (1) eliminate all constant, linear, and quadratic coefficients of the Taylor series expansions of the functions, except for the coefficient of the gradient being evaluated (which is set equal to unity)
- (2) balance the two principal cubic errors of the Taylor series expansions.

The result of these operations is either a 6-point formula (if the gradient is parallel to the face where the gradient is being evaluated) or a 7-point formula (if the gradient is perpendicular to that face). The old 3D gradient formulas were all 6-point formulas, but they were different, and less satisfactory, formulas. The Taylor series being used in the preceding operations are all centered on the point where the gradient is being evaluated. We illustrate these processes by displaying two gradients of the height function $z = H(r,\theta)$, H_{rE} and H_{rN} , evaluated at the center of the east and north faces of the computational cell (i,j,k) respectively. The first gradient is evaluated by a 7-point sum

$$H_{rE} = \sum_{ii=1}^3 \sum_{jj=1}^3 c_{jj ii} H(i', j') , \quad (10)$$

where

$$i' = i - 2 + ii$$

and

$$j' = j - 2 + jj$$

As stated, the computational cell is in the center of the 3×3 array of cells used in the sum. Using the definitions

$$\delta r_O = 1/2 \Delta r_{i+1}$$

$$\delta r = 1/2 \Delta r_i$$

$$\delta r_I = 1/2 \Delta r_{i-1}$$

$$\delta \theta_E = 1/2 \Delta \theta_{j+1}$$

$$\delta \theta = 1/2 \Delta \theta_j$$

$$\delta \theta_W = 1/2 \Delta \theta_{j-1}$$

$$\Delta r_+ = \delta r_O + \delta r$$

$$\Delta r_- = \delta r_I + \delta r$$

$$\Delta \theta_+ = \delta \theta_E + \delta \theta$$

$$\Delta \theta_- = \delta \theta_W + \delta \theta , \quad (10a)$$

we express the 7 nonzero c_{jjii} coefficients as

$$c_{33} = \frac{1}{3} \frac{1}{\Delta r_+^2 \Delta r_- (\Delta r_+ + \Delta r_-)} \frac{\Delta \theta_-}{(\Delta \theta_+ + \Delta \theta_-)} \left((\delta r_O - \delta r) (\Delta r_- + \delta r)^2 + (\Delta r_- + 2\delta r) \delta r_O^2 \right)$$

$$c_{23} = - \left(1.0 + \frac{\Delta \theta_+}{\Delta \theta_-} \right) c_{33} + \frac{(\Delta r_- + 2\delta r)}{\Delta r_+ (\Delta r_+ + \Delta r_-)}$$

$$c_{13} = \frac{\Delta \theta_+}{\Delta \theta_-} c_{33}$$

$$c_{32} = -c_{33}$$

$$c_{22} = \left(1.0 + \frac{\Delta \theta_+}{\Delta \theta_-} \right) c_{33} + \frac{(\delta r_O - \delta r - \Delta r_-)}{\Delta r_+ \Delta r_-}$$

$$c_{12} = - \frac{\Delta \theta_+}{\Delta \theta_-} c_{33}$$

$$c_{21} = - \frac{(\delta r_O - \delta r)}{\Delta r_- (\Delta r_+ + \Delta r_-)} \quad (10b)$$

The second gradient is evaluated by the 6-point sum

$$H_{rN} = \sum_{i=1}^3 \sum_{j=1}^3 a_{jjii} H(i^{'}, j^{'}) . \quad (11)$$

In this case, the 6 nonzero a_{jjii} coefficients are given by

$$\begin{aligned}
 a_{33} &= \frac{1}{3\Delta r_+} \left(1.0 + 3.0 \frac{80}{\Delta\theta_+} \frac{\Delta r_-}{(\Delta r_+ + \Delta r_-)} \right) \\
 a_{32} &= - \left(1.0 + \frac{\Delta r_+}{\Delta r_-} \right) a_{33} + \frac{1}{\Delta r_-} \frac{80}{\Delta\theta_+} \\
 a_{31} &= \frac{\Delta r_+}{\Delta r_-} a_{33} - \frac{1}{\Delta r_-} \frac{80}{\Delta\theta_+} \\
 a_{23} &= - a_{33} + \frac{\Delta r_-}{\Delta r_+(\Delta r_+ + \Delta r_-)} \\
 a_{22} &= \left(1.0 + \frac{\Delta r_+}{\Delta r_-} \right) a_{33} - \frac{1}{\Delta r_-} \frac{80}{\Delta\theta_+} + \frac{(\Delta r_+ - \Delta r_-)}{\Delta r_+ \Delta r_-} \\
 a_{21} &= - \frac{\Delta r_+}{\Delta r_-} a_{33} + \frac{1}{\Delta r_-} \frac{80}{\Delta\theta_+} - \frac{\Delta r_+}{\Delta r_-(\Delta r_+ + \Delta r_-)} . \tag{11a}
 \end{aligned}$$

Corresponding 6 and 7 point formulas hold for the other 22 gradients of the height functions. The various gradient formulas must be altered somewhat for the axial cylinder of cells ($i = 2$ computational cells). In this case, the singularity of the coordinate system at $r = 0$ must be respected and the 3×3 array of cells must be offset from the computational cell by one integer of the i index (to include $i = 2, 3$, and 4 cells) for the cases of $NF = 3, 4, 5$, and 6. The offset 3×3 array of cells still contains enough information to determine accurate values of the gradient, but the 6 and 7 point formulas must be altered somewhat to allow for the fact that the computational cell is now at the edge, rather than at the center, of the 3×3 array.

The expressions in which the radial coordinate of the surface must be evaluated with high precision to obtain accurate values for p_s are indicated in Eq. (6). When necessary, e.g., in the evaluation of $R(\theta_E, z_m)$, 3 point interpolation formulas

are used to obtain sufficiently accurate r values from the 3×3 array of R values available locally at the computational cell.

The p_s algorithm concludes by a set of postprocessing procedures which 'clip' anomalous values of p_s . Specifically, the procedures compute an average value of the surface pressure, p_{sav} , in the surface cells adjacent to, i.e., within a 3×3 array of cells centered on, the computational cell and a) insist that the algebraic sign of p_s agree with the algebraic sign of p_{sav} (this eliminates any local "pimples" in the surface), and b) insist that $|p_s|$ differ by no more than 50% from $|p_{sav}|$. For simplicity, the calculation proceeds from $i = 3, 4, \dots$ and then comes back with a slightly different algorithm for $i = 2$, reflecting the topological difference of the "inner edge" of the computing mesh. This procedure allows for: (1) the occasional malfunctioning of the decision procedure leading to the choice of NF values, and (2) the possibility that an inappropriate zoning of the problem produces locally some inaccurate values of p_s . These procedures limit the pressure gradient acceleration of fluid in the surface cells, and have proven to be an essential part of an algorithm which prevents an unphysical growth of such velocities, which, if unchecked, will eventually terminate the calculation in the F convection routine.

3. Wall-Adhesion. The present version of the program treats wall-adhesion accurately only for straight-walled containers. Since curved-container walls may give rise to substantial errors in wall-adhesion forces, we introduce an input parameter NOWALL, which can force the surface-tension option to be chosen. When $NOWALL = 0$ the program calculates surface-tension or wall-adhesion as directed by the general surface-physics algorithm; but when $NOWALL = 1$ the surface-physics algorithm will always select the surface-tension option.

Considerable effort was spent in obtaining a realistic wall-adhesion algorithm for curved boundaries in 2D problems. (It is a standard feature of NASA-VOF2D). This turned out to be of crucial importance to calculating thin liquid film features, such as occur on the tank wall during the initial phases of a fuel reorientation problem or the final phases of a tank-draining problem. In 2D the task was accomplished by a) defining a cell index array NW, redefined each computational cycle, which determines whether the wall-adhesion option or the surface-tension option is to be selected, and b) defining some angles, defined once and for all by the container geometry, which determine the direction in which any wall-adhesion force is to point. For 2D the wall-adhesion calculation simplifies because any wall-adhesion force points either up or down the container wall. (This neglects the,

usually very small, contact angle of the fluid.) In contrast, a successful 3D wall-adhesion algorithm must introduce a further angle, redefined each computational cycle, which determines from the local surface geometry what direction along the container surface any wall-adhesion force is to point.

We emphasize that the present program, without a universally accurate calculation of wall-adhesion, still has a very significant range of applicability which permits a wide range of design questions to be meaningfully addressed. For example, questions concerning the bulk movement of fluid in a given geometry seldom involve thin liquid films and thus may be addressed without having the wall interaction of the fluid and its containers specified fully correctly.

E. Incrementing the Variables

Several sets of quantities must be incremented to advance the calculation through one time step. Some procedures are applied to all cells open to flow, others to fluid cells only, with alternative procedures for surface cells. We discuss the main stages of this process below. Velocities and pressures, having been determined at cycle n , i.e., at $t = n\delta t$, are to be determined at cycle $n + 1$.

1. *Provisional (TILDE) Velocities.* As an aid to incrementing the pressure and velocity variables, we calculate provisional, or TILDE, velocities, i.e., $\tilde{u}_{i+\frac{1}{2},j,k}$, etc., including the gravitational acceleration, and we use the cycle n variables to provide provisional values for the specific pressure acceleration $SPGX$, the advection $ADVX$, and the viscous acceleration $VISX$. Thus, the TILDE velocities are determined by equations of the form

$$u_{i+\frac{1}{2},j,k} = u_{i+\frac{1}{2},j,k} + \delta t \left[g_x + SPGX_{i+\frac{1}{2},j,k} + ADVX_{i+\frac{1}{2},j,k} + VISX_{i+\frac{1}{2},j,k} \right], \quad (12)$$

with similar expressions for the y and z velocity components. The expressions used for the x , y , and z components of SPG , ADV , and VIS are the direct 3D analogs of the 2D algorithms adopted in the NASA-VOF2D program.¹¹ In particular, the components of the advection ADV are evaluated as a linear combination of centered differencing and donor-cell differencing, with the input parameter α (ALPHA) specifying the donor-cell fraction. The operation of the algorithms and useful rules of thumb for specifying α are discussed in Reference 11. The present program actually works with the reduced pressure $\phi = p\rho^{-1}$, although it is still denoted p . Thus, den-

sities do not appear explicitly in places, i.e., the TILDE calculation, where $p_{\text{pp}^{-1}}$ occurs in the 2D program. Similarly, in this program σ denotes σp^{-1} . TILDE velocities are obtained for all computational cells.

2. SOR Option for Pressures and Velocities. TILDE velocities must be further corrected in order to obtain cycle $n + 1$ velocities that satisfy the continuity equation. This is done by using one of two options to solve equations for the pressure increments δp^{n+1} in all fluid cells (excluding surface cells, isolated cells, and void cells) which relate the pressures at successive cycles, i.e.,

$$p^{n+1} = p^n + \delta p^{n+1} .$$

In the successive over-relaxation (SOR) option, a straightforward generalization of the 2D SOR option,¹¹ δp is calculated from

$$\delta p = -\beta S , \quad (13)$$

where

$$\beta = \frac{1}{\frac{\partial S}{\partial p}} ,$$

and S is the continuity equation written in finite-difference form. An algorithm is used to obtain updated velocities from the δp 's, and then the most updated values for velocity and pressure are used in re-evaluating S . The process is continued iteratively until finally

$$|S| < EPSI , \quad (14)$$

where EPSI is an input parameter to the program. Usually convergence is accelerated by multiplying the δp of Eq. (13) by an over-relaxation parameter ω , i.e., by an input parameter OMG . The 2D version of the process, with guidance for the choice of input parameters, is discussed in some detail in Reference 11.

3. CR Option for Pressures and Velocities. In the conjugate residual (CR) option for solving for δp , which also applies to fluid cells only, the pressure increments are assembled into a vector P , the cycle $n + 1$ velocities are expressed in terms of the TILDE velocities and of P . The continuity equation at cycle $n + 1$ is then written as the vector equation

$$AP = h , \quad (15)$$

where the matrix A is symmetric and the vector h contains the TILDE velocities and boundary conditions. The conjugate residual algorithm is then used to iteratively solve this equation for the pressure increments. The CR algorithm is terminated when the convergence criterion based on EPSI is satisfied. When P has been obtained, the cycle $n + 1$ pressure and velocity fields are constructed from their defining relations in terms of P . The 3D CR algorithm is the same as the 2D CR algorithm, which is described in more detail in Reference 11. At present we prefer the CR option to the SOR option; it saves computer time in running a problem and, in general, it is a more computationally robust algorithm.

4. Pressures and Velocities for Surface Cells. Surface cell pressures are included indirectly in both pressure iteration options. For the CR option, this must be done carefully to maintain the symmetry of the matrix A . The pressure of a surface cell is determined by an interpolation procedure that involves the surface pressure, p_s , and the pressure, p_n , of the interpolation neighbor fluid cell, which, in turn, is determined by the orientation index, NF, of the surface cell. For example, if the (i,j,k) surface cell has $NF = 1$, then the interpolation neighbor fluid cell is $(i-1,j,k)$. The program uses surface cell geometry and F value to evaluate an interpolation parameter η (implemented in the program through the PETA array), and the pressure in the surface cell is then evaluated from

$$p = (1 - \eta)p_n + \eta p_s . \quad (16)$$

The evaluation of the PETA array is the direct extension of the 2D procedure described in Reference 11.

The velocities of surface cells are incremented beyond the TILDE values by essentially the 2D procedure;¹¹ namely, the gradients of the new pressure increments are used to increment the velocities, with due attention to limiting gradients not

likely to be reliable (because they would, without modification, involve pressures not computed from the pressure iteration mesh), and the normal velocity on the cell face directly opposite to the interpolation neighbor fluid cell is reset afterward by imposing the continuity equation on surface cell velocities. However, this procedure still permits the slow growth of velocities by some nonphysical pressure gradients, typically resulting eventually in large nonphysical azimuthal velocities in cells near the coordinate axis. Thus the velocities in surface cells and void cells are further limited to be not greater than corresponding velocities in neighboring fluid cells.

5. Incrementing the F Values. The algorithms used to advance the F values throughout the mesh from cycle n to cycle $n + 1$ are the direct 3D extension of the 2D algorithms discussed in detail in Reference 11 and in the earlier literature. Here we merely reiterate the principal considerations involved in carrying out the calculation.

The convection algorithms use a form of donor-acceptor differencing which is designed to (1) preserve the sharp definition of free boundaries, which we denote here as fluid interfaces; (2) avoid negative diffusion truncation errors; and (3) not flux more fluid, or more void, across a computing cell interface than the cell losing the flux contains. The algorithms also contain features designed to suppress the appearance of spurious small wisps of fluid.

At each cell boundary the two adjacent cells are distinguished, one as the donor cell and the other as the acceptor cell, according to the algebraic sign of the normal velocity at the cell boundary. The continuity equation for F is written in finite difference form, for which an F value is specified at each boundary of the computational cell. The primary choice of F for each cell boundary is given by a fairly subtle algorithm, simple to state but based on a great deal of fluid dynamics experience, which picks either the donor or the acceptor value of F, F_D or F_A . This primary value of F is then limited to require that neither too much flux nor too much void is fluxed from the donor to the acceptor cell. This limiting feature also contains a provision which helps suppress the nonphysical convection of small wisps of fluid throughout the computing mesh.

When the "defoamer" control parameter IDIV is on, the value of F is further corrected by a term proportional to the divergence of the actual velocity field (the S value referred to previously) which is not rigorously zero. However, when the "defoamer" control parameter IDEFM is on (correcting the S value used in the

pressure iteration subroutine), the IDIV option is skipped. Both “defoamer” options work much as they do in 2D.¹¹ However, the NPACK option is not included in NASA-VOF3D.

As in the 2D program,¹¹ large and small values of F^{n+1} are rounded so that F may be used as a boundary cell flag. F is reset to zero (an empty cell) when its computed value is less than ε_f (EMF in the program) and neighboring full cells become surface cells by having their F value reduced by $1.1 \varepsilon_f$. F is reset to unity (a full cell) when its computed value is greater than $1 - \varepsilon_f$. A typical value for ε_f is 10^{-6} . The rounding error in F values is included in the accumulated volume error. The accumulated volume error is typically very small during the course of a calculation.

As described in detail in Reference 11, the indexing of the F values is not strictly accurate, though it becomes accurate for a problem with constant time step, δt , which is started from rest. The program ignores the change in F occurring in the first half time step and the graphics display F values at the n th time step t_n instead of at the more accurate $t_n + \frac{1}{2} \delta t_n$.

6. Time Step Limitations. Limitations on the time step δt are in very close parallel to the limitations in 2D.¹¹ Specifically, there is a convective limit (the Courant condition), a diffusive limit, and a capillary limit.

The absolute convective limit may be stated

$$\delta t < \min \left\{ \frac{\delta x_i}{|u_{i,j,k}|}, \frac{\delta y_j}{|v_{i,j,k}|}, \frac{\delta z_k}{|w_{i,j,k}|} \right\}, \quad (17)$$

where the minimum is taken over all cells of the computing mesh, but it is usual to require that δt be no more than a small fraction, e.g., 0.25, of the minimum. The parameter α describing the proportion of donor cell differencing must be at least as large as, roughly, 1.2 to 1.5 times the value adopted for the fraction described above, to preserve the stability of the convective differencing.

The avoidance of diffusive instabilities imposes a further limit

$$v\delta t < 0.5 \left(\delta x_i^{-2} + \delta y_j^{-2} + \delta z_k^{-2} \right)^{-1}. \quad (18)$$

Unlike the convective limit of Eq. (17), no harm results from imposing a δt limit very close to the critical value specified by Eq. (18), and this is usually done.

The limitation on the time step from capillary forces stems from the requirement that a capillary wave not travel more than one cell width in one time step. This is enforced in the program by the rough requirement

$$8\sigma \delta t^2 < \rho \delta x_m^3, \quad (19)$$

where δx_m is the minimum average cell width in any direction anywhere in the mesh. In practice, Eq. (19) is usually dominated by the average azimuthal dimension of the cell nearest the axis ($i = 2$ cells).

All the above time step limitations are imposed automatically by setting the input parameter AUTOT = 1.0. The choice AUTOT = 0.0 permits a constant δt to be used. The input parameter ALPHA (needed, as described above, for numerical stability) must be set separately. The operation of the time step limitation procedure is described more fully in Reference 11.

III. PROGRAM DESCRIPTION

The NASA-VOF3D program, like its NASA-VOF2D predecessor, is highly structured so that individual components may be easily modified to fit specific problems or to accept subsequent code upgrades. This approach allows greater efficiency in operation, simplifies problem setup, and facilitates code modification. While NASA-VOF3D can run many problems without alteration, a specific application will in general require program alterations for special inflow or outflow ports, complicated geometries, or unusual initial conditions. Seldom does anything have to be deleted; rather the modifications are usually additions to the existing structure. An example would be the modification of the wall adhesion model, which would be most naturally included primarily in SUBROUTINE SURF10N, with some indices and geometric factors specified elsewhere, to treat cases in which the wall adhesion force significantly affects important features of a simulation.

A. Program Architecture

We present here a discussion of the execution of a calculation by tracing the operation of the program flow chart and listing the subroutines, with a short specification of their functions, for NASA-VOF3D. Note that while NASA-VOF2D is the direct 2D predecessor of the present 3D program NASA-VOF3D, the historical origins of their respective program architectures are somewhat different; NASA-VOF2D

evolved most directly from SOLA-VOF, whereas the present NASA-VOF3D is most directly related to SOLA-3D, which, however, is undocumented.

1. *Description of the Flow Chart.* Figure 1 displays the program's logical sequence. On the left is illustrated the flow through the main program, ROUTINE CONTROL; on the right is an expanded version of the flow through the main subprogram, where problem execution takes place, SUBROUTINE SOLA. In the SOLA flow chart it has been assumed that the CR option has been selected for the pressure iteration.

In SUBROUTINE RINPUT the problem input data are read, and default values are specified for data that might not be read in. In addition, the film package is prepared for execution by calling SUBROUTINE GRPHCFT and the associated ROUTINES GPLOT, GRPHLUN, and SETFLSH and the input data read in from NAMELIST XPUT are written on tapes 9 and 12. The preparation of the film package is specific to the Cray systems at Los Alamos National Laboratory; for other systems the user must substitute his own comparable routines and/or eliminate these calling statements. When TD>0 all subsequent set-up routines are skipped, the problem's initial conditions are read from the problem restart tape and control is transferred to SUBROUTINE SOLA, where problem execution begins. This problem restart capability is the natural way to continue a problem run previously; one needs only to have arranged a convenient tape dump from the previous run. It will also usually represent the simplest way to initiate a problem with "complex" initial conditions, i.e., with non-trivial initial velocity and/or pressure fields. (Only occasionally will such "complex" initial conditions be known with sufficient precision to be reasonably coded into the problem setup.)

When the problem is not to be restarted from a tape dump, CONTROL begins to read problem set-up data and then execute the problem set-up. First, SUBROUTINE MESHSET reads data from NAMELIST MESHGN, sets up the computational mesh, calculates areas of partial flow cells, prints obstacle data and calculates data for use in SUBROUTINE BC. The SUBROUTINE RGRAPHIC reads data from NAMELIST GRAFIC and prepares data, formats and captions for the graphs. Next, SUBROUTINE RCONTUR reads in data from NAMELIST CONTUR, although default values are given in RCONTUR, and sets up formats and titles for contour plots of cell-centered quantities. The graphic systems are again specific to the Cray systems and at Los Alamos National Laboratory, so the user must make appropriate adjustments for other systems.

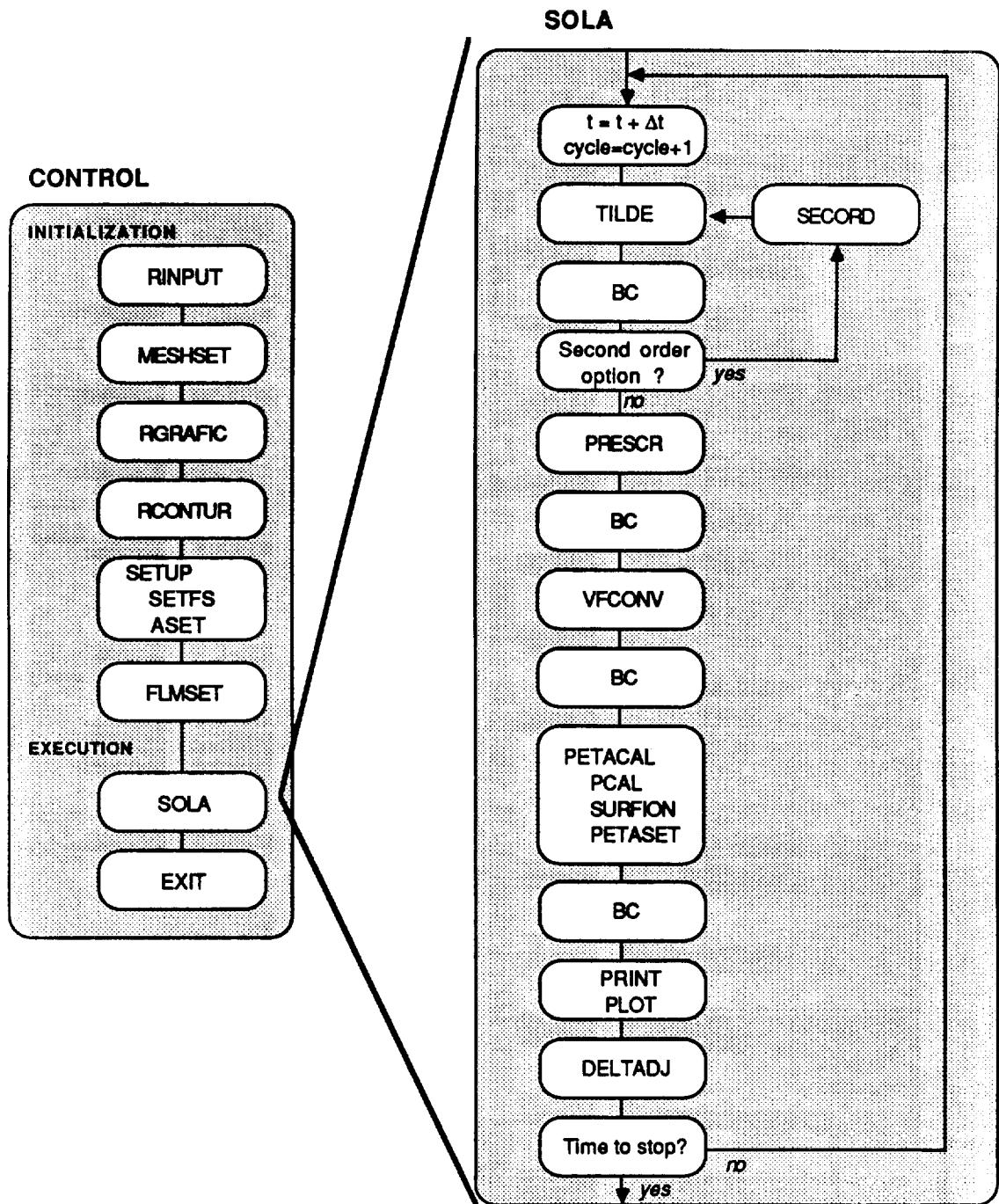


Fig. 1. Flow chart for NASA-VOF3D

In SUBROUTINE SETUP many of the problem initial values are specified; it is here that altered or more general initial conditions would be most appropriately inserted by the user. First, a number of constant parameters are specified and a number of arrays are initialized to zero. Then the initial fluid volume fraction F is calculated for each cell. When ICSURF = 0 (axisymmetric void volume) and IEQUIB > 0 the fluid surface is calculated by SUBROUTINE EQUIB; otherwise it is specified by NAMELIST FLUIDGN and calculated in SUBROUTINE SETFS, which is called to complete the specification of initial values of F, unless the initial surface is horizontal or is an equilibrium surface. In SETFS the fluid surface is specified by means of general quadratic functions. The cases of axisymmetric and nonaxisymmetric volumes are treated separately. Cell F values are specified by calculating the fraction of the cell volume that is "inside" the surface.

SUBROUTINE SETUP continues by calculating the hydrostatic pressure as an initial value for the pressure field, calculating partial flow areas for partial flow cells (this is done by calling SUBROUTINE ASET, which makes use of the obstacle information read by NAMELIST MESHGN, which has already been called by MESHSET and thus is available information), calculating the BETA array for the mesh (done by calling SUBROUTINE BETACAL), calculating the viscous limit to the time step and specifying the initial values of the velocity fields to be zero. The obstacle information (used in ASET) and the fluid surface information (used in SETFS) are set by quadratic functions that are completely analogous to the quadratic functions used in the 2D program and discussed in detail in Reference 11.

Finally, the set-up is completed by calling SUBROUTINE FLMSET, which calculates the mesh boundaries for use in various graphics routines. Then control is passed to SUBROUTINE SOLA where the problem is executed and results are printed, plotted and dumped on tape.

SUBROUTINE SOLA begins by setting some timing information. It then calls SUBROUTINE BC to set the initial boundary conditions for the problem. Then SOLA begins its principal business of advancing the problem through one time step.

First the time variable t is incremented to $t + \delta t$ and the cycle indicator CYCLE is incremented to CYCLE + 1. Then the provisional (TILDE) velocities are obtained by calling SUBROUTINE TILDE. The continuing boundary conditions are obtained by calling SUBROUTINE BC, which, in turn, calls SUBROUTINE BCFS to set velocities on boundaries of surface cells and their adjacent void cells. If the second-order accurate option is specified, SUBROUTINE SECORD is called and the calculation recycles through TILDE and BC. The calculation then proceeds to advance the

pressures, and further advance the velocities, in the fluid cells by calling SUBROUTINE PRESCR if the CR option is specified or calling SUBROUTINE PRESSIT if the SOR option is specified. Boundary conditions are reimposed, but principally the velocities on the boundaries of the surface cells and their adjacent void cells are readjusted by recalling SUBROUTINE BC.

SOLA is then ready to advance the F values by one time step, which it does by calling SUBROUTINE VFCONV, which also computes the total change in fluid volume (by calling SUBROUTINE VCHGCAL). Next, the relevant velocities are required to accommodate the new information on F values by recalling BC. SOLA then treats the surface physics by calling SUBROUTINE PETACAL, which initializes some arrays, calls SUBROUTINE PCAL to determine provisional values of the NF array, and determines final NF values and values of the surface pressure, p_s , in the surface cells by calling SUBROUTINE SURF10N. Continuing, it calculates values of the η (PETA) array for surface cells, sets η in the neighboring interpolation cells (for reasons explained in Reference 11) by calling SUBROUTINE PETASET, and resets pressures in obstacle cells, surface cells, void cells, and isolated fluid cells by calling SUBROUTINE PRESCK. The major surface physics, described in Sec. II.D., is carried out in SUBROUTINE SURF10N, which is the most appropriate place to add the bulk of a wall adhesion algorithm when the program is upgraded in that fashion.

The surface physics now having been carried forward for an additional time cycle, a new set of NF values, and hence a new geometry of surface cells, is now available. SOLA thus calls BC to update the velocities to this new geometric information. SOLA is then ready to print and/or plot the field variables (by calling various graphics routines), set the advanced time field arrays into the time n arrays, adjust the time step δt (by calling SUBROUTINE DELTADJ), and test whether the problem has been completed. If the problem has been completed, SOLA will print, plot, and dump information on tape. CONTROL will then call EXIT and END. If the problem has not been completed, control will remain in SOLA which then begins to advance the problem through another time step δt .

The preceding abbreviated description omits numerous calls to subroutines which accomplish very limited tasks in the calculation; their full inclusion would not only make the calculational path seem much more intricate, and thus harder to understand, but also raise numerous distracting questions (often not really of interest to the user) as to why the particular calculational path, rather than some other path, was chosen. For example, the calculation frequently needs the ordinal index IJK and a number of "neighboring" indexes formed from the cell indices

(i,j,k); thus there are numerous calls to SUBROUTINE CALCIJK which provides this information. But sometimes only IJK is needed, which is provided by calls to IJKONLY, an entry point into CALCIJK. Moreover, while it is frequently advantageous to operate with problem vectors and matrices having a "loose" structure with numerous zeros, which is provided by IJK, for certain matrix operations one wants a "tight" structure in which no vectors contain elements which are identically zero. This is provided by calls to IJKAJCT, another entry point into CALCIJK, and use of the ordinal index LIJK.

We have attempted to provide a reasonable explanatory path for the program by enumerating the main features of the flow chart, identifying the main players (the subroutines, and the variables and arrays in common) of the program, specifying the problem input, and discussing the user options to influence the way in which the calculation is carried out. But there may well be no mode of explanation that successfully avoids the potential pitfalls of either explaining to the user many details which are of no interest to him or not providing a clear explanation of what the program is doing and, for the few that are interested, a rationale for why the program does what it does. We conclude this section by noting that the user would be well advised to be alert for systems calls that are specific to the Cray systems at Los Alamos National Laboratory; these he may need to do something about.

2. *SUBROUTINES and Their Functions.* We specify here the SUBROUTINES of NASA-VOF3D, and define their functions briefly.

ASET	(interior obstacle generator) Generates the problem variables, parameters, and flags to implement the partial flow volumes. Sets BETA to -1.0 in fictitious cells and in obstacle cells.
BC	(boundary conditions) Sets the appropriate fluid variables at the mesh boundaries and interior obstacles to produce specified boundary conditions; calls BCFS to set velocities on free surface and adjacent void cell boundaries in the absence of real pressure gradients. Proper place to insert special boundary conditions not built into code.
BCFS	(boundary conditions) Sets velocities on free surfaces and adjacent void cell boundaries in the absence of real pressure gradients (called by BC).
BETACAL	(β for mesh interior) Calculates BETA for nonobstacle cells. Includes effects of over-relaxation parameter ω .

CALCIJK	(cell indices) Evaluates IJK and other cell indices for cell i,j,k.
CNTR	(graphics) Plots contours of cell-centered quantities. Set-up in SUBROUTINE RCONTUR. User must input array to be plotted if other than pressure.
CONTRJB	(graphics) Sets scales of quantities to be displayed.
CONTROL	(controlling main program) Exercises logical control over the sequence of calculations. Calls SUBROUTINE SOLA for main calculation.
DELTADJ	(time step adjustment) Computes maximum allowable δt for stability, adjusts δt considering stability limit and number of iterations required for pressure convergence, recomputes relaxation factors (BETA) used with SOR method.
DGAP	(graphics) Defines graph area for perspective plots.
DRAW	(graphics) Plots velocity vectors, free surfaces, contours of F and P, and perspectives of surface on film.
DRAWQ	(graphics) Displays composite graphs of two different J-plane results in one graph, converting 3D results to a 2D display. This is the first of four supplementary subroutines needed for this purpose.
DRF	(graphics) Draws frame around computing mesh in plane view. Calls on SUBROUTINE DRFCYL for cylindrical coordinates.
DRFCYL	(graphics) Provides k = constant frame for cylindrical coordinates to SUBROUTINE DRF.
DRFP	(graphics) Draws frame around computing mesh in perspective.
DROBSP	(graphics) Draws interior obstacles in perspective.
DRVEC	(graphics) Provides a system-dependent call opportunity to draw a line segment in any coordinate plane.
DRWOBS	(graphics) Draws interior obstacles in plane view.
DRWOBSQ	(graphics) Draws the obstacles used in the computational mesh for the composite displays of DRAWQ.
DRWVEC	(graphics) Draws a vector for the composite graphics displays of DRAWQ.
EQUIB	(equilibrium surface) Solves two-point boundary value problem to obtain (cylindrically symmetric) equilibrium shape of initial free surface and gives it to SUBROUTINE SETUP.
FLMSET	(graphics) Calculates mesh boundaries for plotting.

FRAMEQ	(graphics) Draws a frame around the composite graphics displays of DRAWQ.
LPRT	(graphics) Prints data on film.
MESHSET	(mesh generator) Generates computing mesh, the fractional volumes and areas open to flow, and geometric information related to the mesh. Calls on SUBROUTINES ASET, MESHX, MESHY, and MESHZ for specific parts of the calculation.
MESHX	(x-coordinate generator) Computes x-coordinate values and their reciprocals. Called by SUBROUTINE MESHSET.
MESHY	(y-coordinate generator) Computes y-coordinate values and their reciprocals. Called by SUBROUTINE MESHSET.
MESHZ	(z-coordinate generator) Computes z-coordinate values and their reciprocals. Called by SUBROUTINE MESHSET.
PCAL	(surface physics) Calculates preliminary values of NF array and supplies them to SUBROUTINE PETACAL.
PCNV	(graphics) Produces coordinates suitable for perspective views.
PDFCALC	(partial derivatives) Calculates partial derivatives for the F function.
PETACAL	(surface physics) Determines preliminary values of surface orientation index NF and of interpolation parameter η (PETA). Calls on SUBROUTINE PCAL for preliminary NF and PETASET for PETA in neighboring interpolation cells.
PETASET	(surface physics) Calculates PETA in neighboring interpolation cells and supplies them to SUBROUTINE PETACAL.
PLTPT	(graphics) Plots a point. It is called by DRAWQ.
PRESCK	(surface physics) Resets pressure in surface cells, obstacle cells, void cells and isolated fluid cells. It is called by PETACAL.
PRESCR	(conjugate residual) Increments pressures and velocities by conjugate residual technique, as in 2D version of code.
PRESSIT	(successive-over-relaxation) Increments pressures and velocities by successive-over-relaxation technique.
RCONTUR	(graphics) Prepares formats for contour plots and supplies them to SUBROUTINE CNTR.
RDTAPE	(input) Reads input tape.
RGRAFIC	(graphics) Reads and writes data for formats of computer-drawn graphics.

RINPUT	(input) Reads problem input data.
SECORD	(velocity adjustment) Adjusts TILDE velocities to second-order accuracy for SUBROUTINE SOLA.
SETACC	(gravitational acceleration) Generates r and θ components (from x and y components) of a general gravitational field.
SETFS	(initial conditions) Generates initial configuration of fluid and gives it to SUBROUTINE SETUP.
SETUP	(initial conditions) Calculates problem parameters, problem geometry, initial arrays, fluid configuration and velocities. Calls SUBROUTINES ASET, BETACAL, EQUIB, and SETFS for specific phases of the calculation.
SOLA	(main calculational program) Increments pressures, velocities, and F values by one time step δt . Calls numerous SUBROUTINES for specific phases of calculation.
SURCNTR	(graphics) Draws the free surface on particular plane plots of velocity vectors.
SURFPLT	(graphics) Draws free surface in perspective.
SURF10N	(surface physics) Gives final NF values and calculates surface tension effects. Appropriate place to insert most of the needed wall adhesion modifications; present program uses very early version of wall adhesion algorithm which is known to be inadequate.
TILDE	(velocity increment) Calculates explicitly a set of approximate velocity increments.
VELV	(graphics) Draws velocity vectors in plane or in perspective plots.
VCHGCAL	(void volume) Computes volume of disjoint void ($F = 0.0$) regions.
VFCNV	(F increment) Computes the advective fluxes of F from the newly determined velocity field and updates the F array.
WRTAPE	(output) Writes tapes of output.
XINDF	(initialization) Sets arrays to indefinite values.

B. Communication Among Subroutines

In communication, in the sense of one subroutine calling another one, there is a sharing of information among subroutines by means of variables and arrays stored in COMMON. We enumerate these elements of COMMON here, defining them briefly.

1. COMMON Variables. We first supply definition of the variables listed in COMMON.

VARIABLES IN COMMON (EXCLUDING ARRAYS AND NAMELIST INPUT PARAMETERS)

AVE	Second-order accurate option parameter
CLK	System furnished time of day for run identification
CSANG	Cosine of contact angle
CYCLE	Calculational time cycle number
DAT	Date furnished by system for run identification
DELMN	Smallest cell dimension in problem
DELXRL	Ratio of cell spacing on left side of mesh (= DELX(1)/DELX(2))
DELXRR	Ratio of cell spacing on right side of mesh (= DELX(IMAX)/DELX(IMI))
DELYRBK	Ratio of cell spacing on back side of mesh (= DELY(JMAX)/DELY(JMI))
DELYRF	Ratio of cell spacing on front side of mesh (= DELY(1)/DELY(2))
DELZRB	Ratio of cell spacing at bottom of mesh (= DELZ(1)/DELZ(2))
DELZRT	Ratio of cell spacing at top of mesh (= DELZ(KMAX)/DELZ(KM1))
DLZ	Contour plot parameter (= 0.0 program determines interval, = value specifies interval)
DPMX	Contour plot parameter (= length of x-axis)
DPMY	Contour plot parameter (= length of y-axis)
DPMZ	Contour plot parameter (= length of z-axis)
DTVIS	Maximum DELT value allowed by the viscous forces stability criterion
EMF	Small value used to negate round-off error effects in F convection (= 10^{-6})
EMFI	$1.0 - EMF$
EM6	10^{-6}
EPSI	Input parameter
FIXL(m)	Plot parameter (= left plot boundary in plotting coordinates for each plot plane)
FIXR	Plot parameter (= right plot boundary)
FIYB	Plot parameter (= bottom plot boundary)

FIYT	Plot parameter (= top plot boundary)
FLG	Pressure iteration convergence indicator (= 0.0 converged)
FLGC	Volume of fluid convection limit indicator
FNOC	Pressure convergence failure indication
I	INDEX value (normally refers to R(x) coordinate cell index)
IBAR	Number of real cells in x direction (excludes fictitious cells)
IGRD	Plotting parameter (= 0. Plot routines determine number of grid lines, = value gives IGRD + 1 grid lines)
II0	Data storage parameter (= NQ*IBAR)
III	Data storage parameter (= NQ*IMAX)
II2	Data storage parameter (= NQ*IMAX*JMAX)
II3	Data storage parameter (= NQ*IMAX*JBAR)
II4	Data storage parameter (= NQ*IMAX*JMAX*KBAR)
II5	Data storage parameter (= IMAX*JMAX)
II6	Data storage parameter (= IMAX*JBAR)
II7	Data storage parameter (= IMAX*JMI)
IJK	Single index value for cell i,j,k
IJKM	Single index value for cell i,j,k-1
IJKP	Single index value for cell i,j,k + 1
IJKM	Single index value for cell i,j - 1,k
IJKP	Single index value for cell i,j - 1,k + 1
IJKM	Single index value for cell i,j + 1,k
IJKP	Single index value for cell i,j + 1,k - 1
IM1	Value of the index I at the last real cell in the x-direction (= IMAX-1)
IM2	Value of the index I at the next to last real cell in the x-direction (= IMAX-2)
IMAX	Total number of cells in x-direction (= IBAR + 2)
IMJK	Single index value for cell i - 1,j,k
IMJKM	Single index value for cell i - 1,j,k-1
IMJKP	Single index value for cell i - 1,j,k + 1
IMJKP	Single index value for cell i - 1,j + 1,k
IPJK	Single index value for cell i + 1,j,k
IPJKM	Single index value for cell i + 1,j,k - 1
IPJKP	Single index value for cell i + 1,j,k + 1

IPJMK	Single index value for cell i + 1,j - 1,k
IPJPK	Single index value for cell i + 1,j + 1,k
IMJMK	Single index value for cell i - 1,j - 1,k
IJMKM	Single index value for cell i,j - 1,k-1
IMJPKP	Single index value for cell i - 1,j + 1,k + 1
IPJMKP	Single index value for cell i + 1,j - 1,k + 1
IPJPKM	Single index value for cell i + 1,j + 1,k - 1
IPJPKP	Single index value for cell i + 1,j + 1,k + 1
ITER	Pressure iteration counter
J	Azimuthal (y) cell index
JBAR	Number of real cells in y-direction (= JMAX - 2)
JC2PI	Indicator of 360 degree geometry (CYL = 1.0)
JM1	Value of the index J at the last real cell in the y-direction (= JMAX - 1)
JM2	Value of the index J at the next to last real cell in the y-direction (= JMAX - 2)
JMAX	Total number of mesh cells in the y-direction (= JBAR + 2)
K	Axial (z) cell index
KBAR	Number of real cells in z-direction (= KMAX - 2)
KM1	Value of the index K at the last real cell in the z-direction (= KMAX - 1)
KM2	Value of the index K at the next to last cell in the z-direction (= KMAX - 2)
KMAX	Total number of mesh cells in the z-direction (= KBAR + 2)
NCR1	Combined length of common blocks SSCM1 through SSCM6
NCR2	Individual length of common blocks SLCM1 through SLCM4
NFLGC	Accumulated F convection limit excesses
NOCON	Accumulated pressure convergence failures
NUMTD	Restart tape dump counter
PI	= 3.141592654
RIJK	Reciprocal of the number of real nonobstacle cells on computational mesh
SANG	Sine of the contact angle
STIM	System clock time when problem commences
TANCA	Tangent of contact angle

TLM	Problem control parameter (=system request time less orderly ending increment)
TWPLT	Problem time for next plot and/or data print to be sent to film
TWPRT	Problem time for next paper data print to be sent to output
TWTD	Problem time for next restart tape dump
UDUM	Temporary storage for u component of velocity
VCHGT	Accumulated fluid volume change in mesh
VDUM	Temporary storage velocity
VOFTOT	Total fluid volume on mesh at cycle n
WDUM	Temporary storage w velocity
XBL	Plot parameter--left boundary of plotting mesh in problem units (=x(1))
XBLC	Left boundary of plotting mesh in problem units for R, θ velocity vector plot (=x(1) normally)
XBR	Right boundary of plotting mesh in problem units (=x(IM1))
XLABLE	Contour plot x-axis label
YBBK	Back boundary of plotting mesh in problem units (=y(JM1))
YBF	Front boundary of plotting mesh in problem units (=y(1))
YBFC	Front boundary of plotting mesh in problem units for R,θ velocity vector plots (=y(1) normally)
YBABLE	Contour plot y-axis label
ZBB	Bottom boundary of plotting mesh in problem units (=z(1))
ZBT	Top boundary of plotting mesh in problem units (=z(KM1))
ZLABLE	Contour plot z-axis label
ZMN	Contour plot parameter
ZMX	Contour plot parameter

2. **COMMON Arrays.** Finally, we enumerate and supply definitions for the arrays listed in COMMON.

ARRAYS IN COMMON (EXCLUDING NAMELIST ARRAYS)

ABK(i,j,k)	Fractional area open to flow on back wall of cell (i,j,k)
AC(i,j,k)	Fractional volume open to flow in cell (i,j,k)
AR(i,j,k)	Fractional area open to flow on right wall of cell (i,j,k)
AT(i,j,k)	Fractional area open to flow on top wall of cell (i,j,k)
BETA(i,j,k)	Pressure iteration relaxation factor in cell (i,j,k)

CTHJ(j)	Cosine of θ (at cell center) = $\cos(YJ(j)/X(IM1))$
CTHJBK(j)	Cosine of θ (at cell back) = $\cos(Y(j)/X(IM1))$
D(i,j,k)	The residual ($\nabla \cdot u$) for cell (i,j,k) after convergence of the pressure iteration
DELX(i)	Mesh spacing of the i-th cell along the radial (x) coordinate
DELY(j)	Mesh spacing of the j-th cell along the azimuthal (y) coordinate at the maximum R(x) mesh value
DELZ(k)	Mesh spacing of the k-th cell along the axial (z) coordinate
FIXL(n)	Left boundary of mesh in plot coordinates for each plotting plane
F(i,j,k)	Volume of fluid per unit volume of cell (i,j,k) at time level n + 1
FN(i,j,k)	Volume of fluid per unit volume of cell (i,j,k) at time level n
GRDBN(n)	Array of values of plot boundaries in problem coordinates for use in DGAP plot routine
GXA(i,j)	Radial component of acceleration due to applied body force for cell (i,j,k)
GYA(i,j)	Azimuthal component of acceleration for cell (i,j,k) due to applied body force
JOP(j)	Index value of plane opposite jth J plane (appropriate for CYL = 1.0 only)
NF(i,j,k)	Flag indicating cell type at time level n + 1
NFO(i,j,k)	Flag indicating cell type at time level n
NFP(i,j,k)	Flag indicating provisional cell type at time level n + 1 (from fluid surface height function)
NFS(i,j,k)	Flag indicating provisional cell type at time level n + 1 (from fluid surface slope function)
P(i,j,k)	Pressure in cell (i,j,k) at time level N + 1
PETA(i,j,k)	Pressure interpolation factor for cell (i,j,k)
PN(i,j,k)	Pressure in cell (i,j,k) at time level N
PR(l)	Pressure in void region l (nominally = 0)
PS(i,j,k)	Surface pressure in cell (i,j,k) computed from surface tension forces
Q(m,n)	Temporary storage for plot routine
RDX(i)	Reciprocal of DELX(i)
RDXP(i)	2.0/(DELX(i) + DELX(i + 1))

RDY(j)	Reciprocal of DELY(j)
RDYP(j)	$2.0/(DELY(j) + DELY(j + 1))$
RDZ(k)	Reciprocal of DELZ(k)
RDZP(k)	$2.0/(DELZ(k) + DELZ(k + 1))$
RX(i)	Reciprocal of X(i)
RXI(i)	Reciprocal of XI(i)
RY(j)	Reciprocal of Y(j)
RYJ(j)	Reciprocal of YJ(j)
RZ(k)	Reciprocal of Z(k)
RZK(l)	Reciprocal of ZK(k)
STHJ(j)	Sine of θ (at cell center) = $\sin(YJ(j)/X(IM1))$
STHJBK(j)	Sine of θ (at cell back) = $\sin(Y(j)/X(IM1))$
U(i,j,k)	x-direction velocity component in cell (i,j,k) at time level n + 1
UN(i,j,k)	x-direction velocity component in cell (i,j,k) at time level n
UVV(m,n)	x-component of velocity at center of cell (i,j,k) (used in velocity vector plots)
V(i,j,k)	y-direction velocity component in cell (i,j,k) at time level n + 1
VN(i,j,k)	y-direction velocity component in cell (i,j,k) at time level n
VVV(m,n)	y-component of velocity at center of cell (i,j,k)(used in velocity vector plots)
W(i,j,k)	z-direction velocity component in cell (i,j,k) at time level n + 1
WN(i,j,k)	z-direction velocity component in cell (i,j,k) at time level n
WWV(m,n)	z-component of velocity at center of cell (i,j,k)(used in velocity vector plots)
X(i)	Location of the right-hand boundary of the i-th cell along the x-axis
XCONV(n)	Scale factors used to convert from problem coordinates to plot coordinates (horizontal)
XI(i)	Location of the center of the i-th cell along the x-axis
XPC(m,n)	x-coordinate of center of cell (i,j,k)(used in velocity vector plots)
Y(j)	Location of the back boundary of the j-th cell along the y-axis
YCONV(n)	Scaling factors used to convert from problem coordinates to plot coordinates (vertical)
YJ(j)	Location of the center of the j-th cell along the y-axis

YPC(m,n)	y-coordinate of center of cell (i,j,k)(used in velocity vector plots)
Z(k)	Location of the top boundary of the k-th cell along the z-axis
ZCQ(n)	Contour plot parameters
ZK(k)	Location of the center of the k-th cell along the z-axis
ZPC(m,n)	z-coordinate of center of cell (i,j,k)(used in velocity vector plots)

C. Problem Input

Problem input data are furnished the code by the NAMELIST format-free option. Most have default values specified by DATA or DEFINING statements located in the subroutine containing the NAMELIST list. Data sets for problem input need only refer to variables or parameters whose values differ from these defaults.

1. **NAMELIST XPUT.** NAMELIST XPUT parameters specify the material properties of the fluid, the solution method options, problem-dependent code parameters, and output requirements. NAMELIST XPUT is read in subroutine RINPUT

INPUT PARAMETERS (NAMELIST/XPUT/)

ALPHA	Controls amount of donor-cell fluxing (= 1.0 for full donor-cell differencing, = 0.0 for central differencing)
AUTOT	Automatic time step flag (= 1.0 for automatic DELT adjustment, = 0.0 for constant DELT)
CANGLE	Contact angle, in degrees, between fluid and wall
CYL	Mesh geometry indicator (= 1.0 for cylindrical coordinates = 0.0 for Cartesian coordinates)
DELT	Initial time step for calculation (subsequently becomes current time step)
EPSI	Pressure iteration convergence criterion
FLHT	Initial fluid height in computing mesh (if appropriate)
GX	Body acceleration in positive x-direction
GY	Body acceleration in positive y-direction
GZ	Body acceleration in positive z-direction
ICLIP	Indicator used with zoom option (=0 normal, =1 eliminate extraneous lines)
ICSURF	Indicator for fluid generator = 0 , horizontal or equilibrium surface

	= 1 , axisymmetric surface
	= 2 , axisymmetric surface
IDEFM	Defoamer option flag
	= 0 , defoamer off
	= 1 , defoamer on
IEQUIB	Indicator for fluid generator
	= 0 Do not generate equilibrium free surface
	= 1 Generate equilibrium free surface
IORDER	Indicator for second-order accurate velocity calculation option
ISOR	Pressure iteration solution method (conjugate residual = 0, SOR = 1)
ISURFT	Indicator for surface tension
	= 0 No surface tension
	= 1 Calculate surface tension
IZOOM	Indicator for plot routines (= 0 normal, = 1 magnifies a portion of the mesh)
JNM	Graphics plot job identification
LPR	Print output parameter
	LPR = 0 , No paper or film data prints or film plots
	= 1 , Film plots only
	= 2 , Film plots plus film data prints
	= 3 , Paper data prints only
NAME(N)	Problem identification
NFCAL	NF Determination algorithm flag
	NFCAL = 1 , Use provisional value
	NFCAL = 2 , Use slope value
	NFCAL = 3 , Use decision-making algorithm to decide between provisional and slope values
NOWALL	Indicator for wall adhesion (= 1 no wall adhesion, = 0 wall adhesion)
NU	Coefficient of kinematic viscosity
OMG	Over-relaxation factor used in SOR pressure iteration
PLTDT	Time increment between plots and/or data prints to be output on film
PRTDT	Time increment between data prints to be output on paper
RADPS	Constant angular rotation velocity
RHOF	Fluid density
SIGMA	Surface tension coefficient

TD	Dump number on restart tape to continue problem
T	Problem time
TDDT	Time increment between restart tape dumps
TLIMD	Problem run parameter (= 1.0 terminate gracefully, = 0.0 run to system time limit)
TWFIN	Problem time to end calculation
UI	x-direction velocity used for initializing mesh
VELMX	Estimate of maximum velocity in problem, used to scale velocity vectors in plots
VI	y-direction velocity used for initializing mesh
WB	Indicator for boundary condition along bottom of mesh (K = 1)
WBK	Indicator for boundary condition along back of mesh (J = JMI)
WF	Indicator for boundary condition along front of mesh (J = 1)
WI	z-direction velocity used for initializing mesh
WL	Indicator for boundary condition along left side of mesh (I = 1)
WR	Indicator for boundary condition along right side of mesh (I = IM1)
WT	Indicator for boundary condition along top of mesh (K = KM1)

2. **NAMELIST MESHGN.** NAMELIST MESHGN is used to define the computational region dimensions and the zoning within that region. This function was performed by NAMELIST MSHSET in NASA-VOF2D. The computing mesh is constructed from a number of submeshes defined in each coordinate direction. The namelist input specifies the boundaries, the number of cells, the minimum cell dimension, and the convergence point of each submesh. Variable spacings are accommodated by linking a group of submeshes together to achieve any desired distribution of cell spacing. The number of cells is specified in each submesh on each side (i.e., to the left and to the right) of the convergence point. Both cells directly adjacent to the convergence point will have a cell spacing equal to the minimum value specified in the input as DXMN, DYMN, or DZMN. The cell spacing is then expanded quadratically from these cells to the left and right edges of the submesh in accordance with the desired number of cells (NXR,NXL,NYR,NYL, or NZR,NZL) in the input list. If the number of cells specified on the left (right) should produce a uniform cell size that is less than the minimum specified cell size DXMN (or DYMN), a uniform spacing is then used on the left (right). The number of cells to the left and to the right of the convergence point need not be equal, but there must be at least one on both sides.

When two or more submeshes are linked together, it is imperative that the location of the left edge of the right submesh be the same as the location of the right edge of the left submesh. In addition, large disparities in cell spacing should be avoided within a submesh and going from one submesh to another by adjusting the number of cells in the various submeshes. As a general rule the spacing of adjacent cells should not differ by more than 10-20%. The aspect ratio (DELX/DELY) for a given cell should not exceed 1.5 and should be greater than 0.67.

An example of the proper format to be used to specify a mesh spanning the x-dimension LW < x < RW with n submeshes is

$$\begin{aligned} NKX &= n, \quad XL = LW, \quad XL_2, \quad XL_3, \dots, XL_n, \\ XC &= XC_1, \quad XC_2, \dots, XC_n, \quad XR = XL_2, \quad XL_3, \dots, XL_n, \quad RW, \\ NXL &= NL_1, \quad NL_2, \dots, NL_n, \quad NXR = NR_1, \quad NR_2, \dots, NR_n, \\ DXMN &= DXMN_1, \quad DXMN_2, \dots, DXMN_n \end{aligned}$$

in which NL_i represents the number of cells to the left of XC_i, and NR_i is the number of cells to the right of XC_i in each submesh i, i = 1, 2, ..., n.

References 2 and 10 contain several examples of mesh generation input and resultant computational meshes.

MESH SETUP INPUT (NAMELIST/MESHGN/)

DXMN(n)	Minimum space increment in x-direction in submesh n
DYMN(n)	Minimum space increment in y-direction in submesh n
DZMN(n)	Minimum space increment in z-direction in submesh n
NKX	Number of submeshes defining radial (x) coordinate spacing
NKY	Number of submeshes defining azimuthal (y) coordinate spacing
NKZ	Number of submeshes defining axial (z) coordinate spacing
NXL(n)	Number of cells between locations XL(n) and XC(n) in submesh n
NXR(n)	Number of cells between locations XC(n) and cells between locations XC(n) and XL(n + 1) in submesh n
NYL(n)	Number of cells between locations YL(n) and YC(n) in submesh n
NYR(n)	Number of cells between locations YL(n) and YL(n + 1) in submesh n
NZL(n)	Number of cells between locations ZL(n) and ZC(n) in submesh n
NZR(n)	Number of cells between locations ZC(n) and ZL(n + 1) in submesh n
XC(n)	x-coordinates of the convergence point in submesh n

XL(n)	Location of the left edge of submesh n (NKX + 1 values required since XR(N) = XL(n + 1))
YC(n)	y-coordinate of the convergence point in submesh n
YL(n)	Location of the front edge of submesh n (NKY + 1 values required since YR(n) = YL(n + 1))
ZC(n)	Location of the convergence point of submesh n
ZL(n)	Location of the bottom edge of submesh n (NKZ + 1 values required since ZR(n) = ZL(n + 1))

NAMELIST MESHGN also provides information to the interior-obstacle-generating subroutine. The latter function was performed by NAMELIST ASETIN in NASA-VOF2D. Interior obstacles are defined as any nonflow regions within the computational mesh. As with the mesh generator, it is convenient to have the necessary flags and parameters produced automatically by the code for arbitrary-shaped axisymmetric obstacles. The generator uses a series of conic sections to define obstacle surfaces plus an additional parameter that directs the generator to designate computational cells as either flow or nonflow regions. The conic sections may overlap one another and this feature is utilized to form complex axisymmetric surfaces, representing baffles or container walls.

This procedure is as follows: coefficients of the general conic function

$$F(x,z) = a_2x^2 + a_1x + b_2z^2 + b_1z + c_2xz + c_1$$

are chosen such that some portion of the defining surface coincides with that of the desired internal obstacle. Computational cells partially or completely inside the conic surface [i.e., $F(x_c, z_c) < 0.0$, x_c and z_c being cell boundary coordinates] are flagged closed to flow, partially closed to flow, or fully open to flow depending upon the additional parameter IOH (=1.0 closed, =0.0 open) associated with each function. Generally, additional functions can be utilized to remove unwanted obstacle cells added by other functions. For example, given reasonable zoning, a 1.0×1.0 cm obstacle in the lower right corner of a 2.0×2.0 cm mesh would result from $f_1 = -x + 1.0$, $\text{IOH}_1 = 1.0$, which defines all cells, regardless of z value, to the right of $x = 1.0$ to be obstacles, and followed by $f_2 = -z + 1.0$, $\text{IOH}_2 = 0.0$, which removes all obstacle cells above $z = 1.0$. Nonaxisymmetric obstacles (or container boundaries) can be generated by inserting additional coding at the end of subroutine ASET to

define AR(i,j,k), AT(i,j,k), ABK(i,j,k), AC(i,j,k), and BETA(i,j,k) for each nonflow or partial flow cell (i,j,k).

After all sets of obstacle data have been processed, the computing mesh is swept and individual cell area and volume open-to-flow variables are assigned. Aside from geometric factors for cylindrical coordinates, they range from 0.0 (not open to flow) to 1.0 (fully open to flow).

OBSTACLE SETUP (NAMELIST/MESHGN/CONTINUED)

IOH(n)	Indicator to add obstacles inside function (n = 1) or subtract obstacles (n = 0)
NOBS	Number of sets of obstacle data input
OA2(n)	Coefficient of x^2 -term in function F
OA1(n)	Coefficient of x-term in function F
OB2(n)	Coefficient of z^2 -term in function F
OB1(n)	Coefficient of z-term in function F
OC2(n)	Coefficient of xz -term in function F
OC1(n)	Coefficient of constant term in function F

3. **NAMELIST GRAFIC.** The input information needed to prepare the data, formats, and captions for the graphics is supplied by NAMELIST GRAFIC. We define the information here, presenting it in the order required by the calling sequence associated with SUBROUTINE RGRAFIC. The information is further broken down by specific functions which are separated by lines of asterisks (***)...: first there is information connected with the eye location in perspective plots, next there is information connected with velocity vector plots, following that, contour plot data, and, finally, information for surface plots.

FORMAT PARAMETERS FOR GRAPHS (NAMELIST/GRAFIC/)

XEA(n)	The x-coordinate of the eye position in problem units.
YEA(n)	The y-coordinate of the eye position in problem units.
ZEA(n)	The z-coordinate of the eye position in problem units.
XCA(n)	The x-coordinate of the projection plane in problem units. This coordinate is usually taken near the origin.
YCA(n)	The y-coordinate of the projection plane in problem units. This coordinate is usually taken near the origin.
ZCA(n)	The z-coordinate of the projection plane in problem units. This coordinate is usually taken near the origin.

NVEWS	The number of eye views. That is the number of sets of eye point-plane points. That is XEA etc. above. Up to 10.
NVPLTS	The number of velocity vector plots. Up to 10. If NVPLTS = 0, no velocity plots will appear.
IVVIEW(n)	The eye view to be associated with this velocity vector plot. IVVIEW = 1 to 10.
IV1(n)	Defines the value of I for the left boundary of the velocity vectors to be plotted. If IV1 > imax, IV1 = IM1.
IV2(n)	Defines the value of I for the right boundary of the velocity vectors to be plotted. If IV2 > imax, IV2 = IM1.
JV1(n)	Defines the value of J for the front boundary of the velocity vectors to be plotted. If JV1 > jmax, JV1 = JM1.
JV2(n)	Defines the value for J for the back boundary of the velocity vectors to be plotted. If JV2 > jmax, JV2 = JM1.
KV1(n)	Defines the value of K for the bottom boundary of the velocity vectors to be plotted. If KV1 > kmax, KV1 = KM1.
KV2(n)	Defines the value of K for the top boundary of the velocity vectors to be plotted. If KV2 > kmax, KV2 = KM1.
NAV(n)	The number of frames on the film to be advanced before the velocity vector plot is made. Normally it is 1.
IPERV(n)	Toggle switch for perspective plot. If IPERV = 1, normal plot. If IPERV = 2, perspective plot.
NCPLTS	The number of contour plots. Up to 10. If NCPLTS = 0, no contour plots will appear.
ICVIEW(n)	The eye view to be associated with this contour plot. ICVIEW = 1 to 10.
IC1(n)	Defines the value of I for the left boundary of the contours to be plotted. If IC1 > imax, IC1 = IM1.
IC2(n)	Defines the value of I for the right boundary of the contours to be plotted. If IC2 > imax, IC2 = IM1.
JC1(n)	Defines the value of J for the front boundary of the contours to be plotted. If JC1 > jmax, JC1 = JM1.
JC2(n)	Defines the value of J for the back boundary of the contours to be plotted. If JC2 > jmax, JC2 = JM1.
KC1(n)	Defines the value of K for the bottom boundary of the contours to be plotted. If KC1 > kmax, KC1 = KM1.

KC2(n)	Defines the value of K for the top boundary of the contours to be plotted. If KC2>kmax, KC2 = KM1.
NCV(n)	The number of frames on the film to be advanced before the contour plot is made. Normally it is 1.
IPERC(n)	Toggle switch for perspective plot. Not operative. Use 1. If IPERC = 1, normal plot. If IPERC = 2, perspective plot.
NSPLTS	The number of surface plots. Up to 10. If NSPLTS = 0, no surface plots will appear.
ISVIEW(n)	The eye view to be associated with this surface plot. ISVIEW = 1 to 10.
IS1(n)	Defines the value of I for the left boundary of the surface to be plotted. If IS1>imax, IS1 = IM1.
IS2(n)	Defines the value of I for the right boundary of the surface to be plotted. If IS2>imax, IS2 = IM1.
JS1(n)	Defines the value of J for the front boundary of the surface to be plotted. If JS1>jmax, JS1 = JM1.
JS2(n)	Defines the value of J for the back boundary of the surface to be plotted. If JS2>jmax, JS2 = JM1.
KS1(n)	Defines the value of K for the bottom boundary of the surface to be plotted. If KS1>kmax, KS1 = KM1.
KS2(n)	Defines the value of K for the top boundary of the surface to be plotted. If KS2>kmax, KS2 = KM1.
NSV(n)	The number of frames on the film to be advanced before the surface plot is made. Normally it is 1.
IPERS(n)	Toggle switch for perspective plot. Not operative. Use 1. If IPERS = 1, normal plot. If IPERS = 2, perspective plot.

4. NAMELIST CONTUR. The input information needed to construct the contour plots is supplied by NAMELIST CONTUR. We enumerate and define the information here, presenting it in the order required by the calling sequence associated with SUBROUTINE CONTRJB.

INPUT PARAMETERS FOR CONTOUR PLOTS (NAMELIST/CONTUR/)

Q	Origin of matrix of values of the function to be contoured. Q should be stored so that Q(I,J) is the value of Q at [X(I), Y(J)].
NNX	Number of points in x table to be used. If NNX is negative, ZMN will be calculated by CONTRJB as the minimum value of Q.

NNY	Number of points in Y table to be used. If NNY is negative, ZMX will be calculated by CONTRJB as the maximum value of Q.
NNZ	Number of points in Z table to be used. If NNZ is negative, ZMX will be calculated by CONTRJB as the maximum value of Q.
NZX	Length of a column of Q matrix. Variable dimensioning is used. For example: If Q is dimensioned Q(10,20), then nzx = 10, and nzy = 20.
NZY	Number of columns in Q matrix. If NZY<0, and NC>0, ZCQ table is computed such that NC is the minimum of NC and (ZMX-ZMN)/DLZ. ZMX and ZMN are made multiples of DLZ.
NZZ	Similar to NZY. Used in the three dimensional extension. See the calling sequence in SUBROUTINE CNTR to CONTRJB.
NC	Number of contours to be plotted. If NC<0, ZCQ table is assumed to be user supplied and ZMN, ZMX, and DLZ are not used. The absolute value of NC must be less than or equal to 50. (See DLZ)
ZMN	Minimum value of contour to be used in contouring. If NNX<0, ZMN will be computed from minimum value of Q over contour grid.
ZMX	Maximum value of contour to be used in contouring. If NNY<0, ZMX will be computed from maximum value of Q over contour grid.
DLZ	Interval between contours. If DLZ is less than or equal to zero, DLZ will be set to (ZMX – ZMN)/(NC – 1).
ZCQ	Table of contour values. If NC>0, ZCQ table is filled by CONTRJB. If NC<0, ZCQ must be supplied by user. At least the absolute value of NC values must be supplied and values must be stored in increasing order in ZCQ, i.e., ZCQ(I + 1)>ZCQ(I).
DMPX	Length of X axes.
DMPY	Length of Y axes.
DMPZ	Length of Z axes.
IGRD	If IGRD ≤0, grid lines will appear only at intervals numbered by scaling subroutine. For a finer mesh, IGRD>0 will add IGRD-1 extra lines per interval.
ITITLE1	BCD array containing title to be written at top of graph 1.
NTITLE1	Number of characters in title 1.
ITITLE2	BCD array containing title to be written at top of graph 2.

NTITLE2	Number of characters in title 2.
XLABEL	BCD arrays containing label to be written on the X axes.
YLABEL	BCD arrays containing label to be written on the Y axes.
ZLABEL	BCD arrays containing label to be written on the Z axes.
NXLBL	Number of characters in X label.
NYLBL	Number of characters in Y label.
NZLBL	Number of characters in Z label.

5. NAMELIST FLUIDGN. Generating the initial fluid configuration is done in SUBROUTINE SETUP. If the surface is other than a horizontal or equilibrium surface, ($ICSURF > 0$), then subroutine SETFS is called.

A complex initial free surface configuration is specified through NAMELIST FLUIDGN input data parameters. The generation of the free surfaces is accomplished in a manner analogous to the generation of interior obstacles through the use of conic functions. Computational cells completely or partially enclosed by the surface defined by a function are either filled, partially filled, or empty of fluid as specified by the input parameters of that function. General functions may be used in conjunction to define very general configurations. Subroutine SETFS has two logic paths, one for axisymmetric volumes and a second for nonaxisymmetric volumes. The distinction was made for computational efficiency since the algorithms for generating off axis (i.e., nonaxisymmetric volumes) lengthens the set-up time considerably.

FREE SURFACE INPUT DATA (NAMELIST/FLUIDGN/)

IQH(n)	Indicator for adding or subtracting fluid inside function n
NQBS	Number of functions to be used in generating free surfaces A
Axisymmetric ($ICSURF = 1$)	
QA1(n)	Coefficient of x term function n
QA2(n)	Coefficient of x^2 term function n
QB1(n)	Coefficient of z term function n
QB2(n)	Coefficient of z^2 term function n
QC1(n)	Constant term function n
OC2(n)	Coefficient of xz term function n
NonAxisymmetric ($ICSURF > 1$)	
QA1(n)	Value of x-coordinate at center of volume n
QA2(n)	Lower limit of l index
QA3(n)	Upper limit of l index

QB1(n)	Value of y-coordinate at center of volume n
QB2(n)	Lower limit of J-index
QB3(n)	Upper limit of J-index
QC1(n)	Value of z-coordinate at center of volume n
QC2(n)	Lower limit of K index
QC3(n)	Upper limit of K index
QD1(n)	Constant term in function n
QD2(n)	Indicator specifying if mesh is initially fluid or void (= 1.0 fluid, = 0.0 void)

The upper and lower limits on the three indices restrict the algorithm to that portion of the computing mesh known to contain volume n.

D. User Options

We summarize here the principal user options to influence the manner in which the calculation is to be carried out. We list principally option flags, i.e., integer valued variables which specify alternate logic paths for the calculations.

ALPHA	0.0 Problem runs with central differencing of convection. 1.0 Problem runs with donor-cell differencing. 1.0 and >0.0 (usual case) partial donor-cell differencing.
AUTOT	0 Problem is to be run with constant δt . 1 Program will automatically impose time step controls; user must check for consistency of parameters.
CYL	0.0 Problem is to be run in Cartesian coordinates. (This option is not presently implemented.) 1.0 Problem is to be run in cylindrical coordinates.
ICLIP	0 Normal zoom operation (if used). 1 Eliminate extraneous lines with zoom.
ICSURF	0 Surface is horizontal or an equilibrium surface. 1 Surface is axisymmetric. Its shape is specified in SETFS by quadratic functions. 2 Surface is nonaxisymmetric. Its shape is again specified in SETFS, but by alternate quadratic function formalism.
IDEFM	0 Program will not use defoamer. 1 Defoamer option on.
IEQUIB	0 Program will not generate equilibrium free surface. 1 Program generates equilibrium free surface.
IORDER	1 Program does not call second-order accuracy option.

	2	Program calls second-order accuracy option.
ISOR	0	Program executes Conjugate Residual option.
	1	Program executes SOR option.
ISURFT	0	Program does not calculate surface tension.
	1	Program calculates a surface pressure, p_s , from surface tension algorithms.
IZOON	0	Normal (no zoom) operation of plot routines.
	1	A portion of the mesh will be magnified.
LPR	0	No paper or film data prints or film plots.
	1	Film plots only.
	2	Film plots plus film data prints.
	3	Paper data prints only.
NFCAL	1	Use provisional values for NF.
	2	Use slope determination of NF value.
	3	Use decision-making algorithm to choose between provisional and slope values for NF.
NOWALL	0	Program imposes wall adhesion (this option is not presently implemented).
	1	Program does not impose wall adhesion at physical boundaries. Only active option in present version of code.
TD	≤ 0	Problem started from initial data.
	> 0	Dump tape7 is read to start problem.
TLIMD	0.0	Problem runs to system time limit.
	1.0	Problem will terminate gracefully.

E. Example of Problem Setup

Table II is an example of a typical input data set for NASA-VOF3D. The code is to simulate a small nearly full glass vessel with cylindrical walls and hemispherical bottom draining under the influence of reduced gravity, surface-tension, and wall-adhesion. The discharge flow rate is specified and constant in time. Here cgs units are used for input and output; any consistent set of units may be employed. From the tabulated input parameters one may see that the conjugate residual method was chosen (ISOR = 0), the defoam option is on (IDEFM = 1), and the standard fluid generator is specified (ICSURF = 0). The initial fluid volume is specified by the fluid height above tank bottom (FLHT = 6.0). The surface-tension flag is on (ISURFT = 1), cylindrical coordinates are chosen (CYL = 1.0), and the equilibrium surface calculation is requested (IEQUIB = 1). The slope method for determining NF is specified

(NFCAL = 2). The initial DELT is computed from the outflow rate, discharge opening area, and mesh spacing in a fashion that satisfies stability constraints. The convergence criterion is set sufficiently small (EPSI = 0.001) that negligible pressure errors result from nonzero ($\nabla \cdot \mathbf{u}$)'s. Film prints and plots are requested (LPR = 2).

Figure 2 displays the initial problem configuration generated from XPUT and MESHGN data. The x coordinate has two submeshes with a total of 11 real cells, the spacing being finer near the tank wall. The y coordinate mesh consists of a single (wedge-shaped) physical cell of arbitrary thickness (NYL = 1, NYR = 0, DYNMN = 1.0). The z coordinate has a single submesh with 36 equally spaced real cells. The curved boundary at the bottom was drawn from cell variables and parameters produced by the obstacle generator subroutine from MESHGN data. Computational cells with areas intersected by the curve are defined to be partial flow cells whose boundaries are identified as partial flow areas. SUBROUTINE ASET computes the fraction of each face open to flow and the fractional cell volume open to flow (i.e., cell parameters AR, AT, ABK, AC). Computational mesh cells to the right and below the curve are designated obstacle cells, while those to the left and above are fluid cells. All interior obstacles and exterior, i.e., container, walls are simulated.

TABLE II
EXAMPLE OF INPUT DATA

```

$XPUT
JNM = 8H VOF3D7 .NAME = 45HV0F3D TANK DRAIN J=2 PLANE ONLY   11/13/86
NFCAL = 2, IEQUIB = 1, ICSURF = 0, IDEFM = 1, RHOF = 1.58,
CYL = 1.0, DELT = 0.0002, GZ = - 14.72, VELMX = 1.0, NU = .00443, ISOR = 0,
EPSI = 0.001, VI = 0.0, SIGMA = 18.6, LPR = 2, CANGLE = 5.0, ISURFT = 1,
WF = 1, WBK = 1, FLHT = 6.0, TWFIN = 2.5, OMG = 1.8, PLTDT = .10, TDIT = 5.0
$END
$MESHGN
NKX = 2, XL = 0.0, 0.40, 2.0, XC = 0.20, 1.90, NXL = 1, 8, NXR = 1, 1,
DXMN = 0.20, 0.10,
NKY = 1, YL = 0., 1.0, YC = 1.0, NYL = 1, NYR = 0, DYMN = 1.0,
NKZ = 1, ZL = 0.0, 7.20, ZC = 3.60, NZL = 18, NZR = 18, DZMN = 0.20,
NOBS = 2, OA2 = - 1.0, 0.0, OA1 = 0.0, OB2 = - 1.0, 0.0, OB1 = 4.0, - 1.0,
OC2 = 0.0, 0.0, OC1 = 0.0, 2.0, IOH = 1,0
$END
$GRAFIC
NVEWS = 1, XEA(1) = 700., YEA(1) = 50., ZEA(1) = 1000., XCZ(1) = 150., YCA(1) = 15.,
ZCA(1) = 400., NVPLTS = 1, NCPLTS = 1, NSPLTS = 0,
IVIEW(1) = 1, NAV(1) = 1, IPERV(1) = 1,
IV1(1) = 2, IV2(1) = 99,
JV1(1) = 2, JV2(1) = 2
KV1(1) = 2, KV2(1) = 99,
ICVIEW(1) = 1, NAC(1) = 1,
IPERC = 1,
IC1(1) = 2, IC2(1) = 99,
JC1(1) = 2, JC2(1) = 2,
KC1(1) = 2, KC2(1) = 99,
$END
$CONTUR
$END
$FLUIDGN
$END

```

The initial free surface shape is furnished by SUBROUTINE EQUIB, which solves the two-point boundary value problem for the equilibrium position of the free surface. The parameters of the equation are the contact angle (cangle) and the Bond number ($\rho g R^2 / \sigma$, where R is the tank radius). This free surface configuration is utilized to specify void cells, partially filled fluid cells, and completely full fluid cells

in the computational mesh. The free surface is represented in the plot by the $F = 0.5$ contour, which generally does not correctly display the surface meniscus at the walls.

Table III contains code changes necessary to run the example. Programs for maintaining and updating source decks (UPDATE or HISTORIAN) afford a convenient way to incorporate problem-dependent modifications into the executable code. These particular modifications arise since we wish to simulate a small pipe located on the tank bottom at the cylindrical axis, whereas the code lacks provision for partially open mesh boundaries.

First, the parameter statement (No. 2 of SLCOM1) is replaced, specifying minimum storage requirements for the various arrays. Statements inserted at the end of SUBROUTINE BETACAL redefine the cell area-open-to-flow variables for cell $i = 2$, $j = 2$, $k = 1$, so that a discharge port is created. A second insertion (in SUBROUTINE BC) specifies the constant outflow velocity for this port. Also, changes are made in SUBROUTINE PRESCR to correctly apply outflow velocity boundary conditions for discharge port cells in the pressure iteration.

Efficiency is enhanced by a problem end test inserted into SUBROUTINE SOLA and by an insertion into SUBROUTINE VFCONV which excludes the fictitious port cell from contributing to the time step limitations arising from stability considerations. Finally, the insertion into SUBROUTINE SURF10N facilitates comparison with experimental data by estimating the heights of the free surface at the axis ($i = 2$) and at the container wall ($i = IM1$).

Table IV is an echo print of the example problem input data. It displays the complete set of problem run and output control parameters.

A partial cycle print is included in **Table V** as an example of code output and as an aid to checkouts. It is the solution after one time step; the flow field is established and the pressure and velocity fields are consistent. Surface pressures are defined only for surface cells ($0 < NF(i,j,k) < 7$) and are negative, indicating that unbalanced surfaces are directed upward (acting to straighten the surface). Cell $i = 2$, $j = 2$, $k = 30$ has $F = 0.111$ and $NF = 5$, indicating that the surface passes through the cell, is nearly horizontal and is above the fluid.

Figure 3 displays the time history of the free surface heights at the axis (below) and at the container wall (above). The experimental data reported by Symons¹² were obtained from analysis of high speed photography. Calculational results

TABLE III
EXAMPLE PROBLEM CODE MODIFICATIONS

```

*IDENT TANKD
*D,SLCDM1.2
C
C      CHANGE STORAGE DIMENSIONS
C
C      PARAMETER (NQ=6, IBAR2=13, JBAR2=3, KBAR2=38, IBASC=NQ*IBAR2
*I,BC.233
C
C      IMPOSE CONSTANT VELOCITY AT OUTFLOW BOUNDARY
C
DO 600 J=2,JM1
IJK=NQ*(IMAX*(J-1)+1)+1
WOUTB=249.7936
W(IJK)=-WOUTB
600 CONTINUE
*I,BETACAL.38
C
C      MODIFY AREA OPEN TO FLOW TO CREATE OUTFLOW OPENING
C
DO 600 J=1,JMAX
IJK=NQ*(IMAX*(J-1)+1)+1
BETA(IJK)=+1.0
AT(IJK)=1.0
AC(IJK)=1.0
ABK(IJK)=1.0
600 CONTINUE
*I,PRESCHR.117
C
C      MODIFY PRESSURE SOLUTION TO IMPOSE CONSTANT VELOCITY OUTFLOW
C      BOUNDARY CONDITION
C
IF(I.EQ.2.AND.K.EQ.1) GO TO 340
*I,PRESCHR.175
C
C      MODIFY PRESSURE SOLUTION TO IMPOSE CONSTANT VELOCITY OUTFLOW
C      BOUNDARY CONDITION
C
IF(I.EQ.2.AND.K.EQ.2) GO TO 290
*I,VFCNV.21
C
C      SKIP STABILITY REQUIREMENTS FOR FICTITIOUS OUTFLOW CELL
C
IF(K.EQ.1) GO TO 20
*I,SOLA.124
C
C      END PROB IF VAPOR INGESTED
C
IJK=NQ*(IMAX*JMAX+IMAX+1)+1
IF(F(IJK).LT.0.1) TWFIN=T
*I,SURF1ON.508
C
C      OUTPUT FREE SURFACE HEIGHTS AT AXIS AND WALL
C
IF(MOD(CYCLE,20).NE.0) GO TO 9092
IF(I.EQ.2.OR.I.EQ.IM1) HISURF=Z(K-1)-DELZ(K-1)+F(IJKM)*DELZ(K-1)+1
F(IJK)*DELZ(K)+F(IJKP)*DELZ(K+1)
IF(I.EQ.2) WRITE(12,9090) I,J,K,T,HISURF
IF(I.EQ.IM1) WRITE(12,9091) I,J,K,T,HISURF
9090 FORMAT(5X,3I3,1PE12.5,*    AXIS HEIGHT**E12.5)
9091 FORMAT(5X,3I3,1PE12.5,*    WALL HEIGHT**E12.5)
9092 CONTINUE

```

TABLE IV
EXAMPLE INPUT ECHO PRINT

```

VOF3D TANK DRAIN J=2 PLANE ONLY      11/13/86      #####H'#####H'#####H'
ALPHA= 1.00000E+00
AUTOT= 1.00000E+00
CANGLE= 8.72665E-02
CYL= 1.00000E+00
DELT= 2.00000E-04
EPSI= 1.00000E-03
FLHT= 6.00000E+00
GX= 0.00000E+00
GY= 0.00000E+00
GZ= -1.47200E+01
ICLIP= 0
ICSURF= 0
IDEFM= 1
IEQUIB= 1
ISOR= 0
IORDER= 1
ISURFT= 1
IZOOM= 0
JNM= VOF3D7
LPR= 2
NFCAL= 2
NOWALL= 0
NU= 4.43000E-03
OMG= 1.80000E+00
PLTDT= 1.00000E-01
PRTDT= 1.00000E+00
RADPS= 0.00000E+00
RHOF= 1.58000E+00
SIGMA= 1.86000E+01
T= 0.00000E+00
TDDT= 5.00000E+00
TD= -1
TLIMD= 1.00000E+00
TWFIN= 2.00000E-04
UI= 0.00000E+00
VELMX= 1.00000E+00
VI= 0.00000E+00
WB= 1
WBK= 1
WF= 1
WI= 0.00000E+00
WL= 1
WR= 1
WT= 1
NKX= 2
MESH-X= 1   XL= 0.00000E+00   XC= 2.00000E-01   XR= 4.00000E-01   NXL= 1   NXR= 1   DXMN= 2.00000E-01
MESH-X= 2   XL= 4.00000E-01   XC= 1.90000E+00   XR= 2.00000E+00   NXL= 8   NXR= 1   DXMN= 1.00000E-01
NKY= 1
MESH-Y= 1   YL= 0.00000E+00   YC= 1.00000E+00   YR= 1.00000E+00   NYL= 1   NYR= 0   DYMN= 1.00000E+00
NKZ= 1
MESH-Z= 1   ZL= 0.00000E+00   ZC= 3.60000E+00   ZR= 7.20000E+00   NZL= 18  NZR= 18  DZMN= 2.00000E-01
IBAR= 11   JBAR= 1   KBAR= 36

```

TABLE V
EXAMPLE PROBLEM DATA PRINT

UOF3D TANK DRAIN J=2 PLANE ONLY										11/13/86	7070707070707070	UOF3D7	03/12/87 11:06:53			
ITER-	82	TIME-	2.04000E-04	DELT-	2.04000E-04	CYCLE-	1	PS	F	NF	NFP	NFS	NFD	BETA	PETA	D
I	J	K	U	U	W	P	U	U	U	U	U	U	U	U	U	U
1	2	2	4.3354E+01	0.0000E+00	-8.5091E+01	-2.1991E+05	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	3	1.2575E+01	0.0000E+00	-3.4791E+01	-1.3650E+05	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	4	4.2926E+00	0.0000E+00	-1.7620E+01	-1.0239E+05	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	5	1.7430E+00	0.0000E+00	-1.0648E+01	-8.5121E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	6	8.2757E-01	0.0000E+00	-7.3382E+00	-7.4685E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	7	4.4477E-01	0.0000E+00	-5.5592E+00	-6.7493E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	8	2.6175E-01	0.0000E+00	-4.5122E+00	-6.2046E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	9	1.6411E-01	0.0000E+00	-3.8558E+00	-5.7625E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	10	1.0736E-01	0.0000E+00	-3.4265E+00	-5.3848E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	11	7.2129E-02	0.0000E+00	-3.1379E+00	-5.0492E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	12	4.9215E-02	0.0000E+00	-2.9410E+00	-4.7418E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	13	3.3814E-02	0.0000E+00	-2.8057E+00	-4.4538E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	14	2.3235E-02	0.0000E+00	-2.7127E+00	-4.1790E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	15	1.5859E-02	0.0000E+00	-2.6492E+00	-3.9133E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	16	1.0633E-02	0.0000E+00	-2.6066E+00	-3.6539E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	17	6.8453E-03	0.0000E+00	-2.5793E+00	-3.3987E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	18	3.9878E-03	0.0000E+00	-2.5633E+00	-3.1461E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	19	1.6650E-03	0.0000E+00	-2.5566E+00	-2.8951E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	20	4.2962E-04	0.0000E+00	-2.5583E+00	-2.6447E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	21	2.5705E-03	0.0000E+00	-2.5685E+00	-2.3942E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	22	5.0764E-03	0.0000E+00	-2.5887E+00	-2.1427E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	23	8.2503E-03	0.0000E+00	-2.6216E+00	-1.8892E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	24	1.2521E-02	0.0000E+00	-2.6716E+00	-1.6325E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	25	1.8425E-02	0.0000E+00	-2.7453E+00	-1.3708E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	26	2.6667E-02	0.0000E+00	-2.8519E+00	-1.1020E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	27	3.8179E-02	0.0000E+00	-3.0464E+00	-8.2266E+03	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	28	5.4102E-02	0.0000E+00	-3.2210E+00	-5.2839E+03	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	29	7.5544E-02	0.0000E+00	-3.5231E+00	-2.1290E+03	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	30	7.5544E-02	0.0000E+00	-3.8253E+00	1.3221E+03	0.0000E+00	1.1147E-01	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	31	1.2422E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	32	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	33	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	34	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
1	2	38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
2	2	1	-8.6708E+01	0.0000E+00	-2.4979E+02	-1.1991E+05	0.0000E+00	1.0000E+00	0	0	0	0	1.0000E+00	1.0000E+00	0.0000E+00	
2	2	2	-8.6708E+01	0.0000E+00	-8.5091E+01	-2.1991E+05	0.0000E+00	1.0000E+00	0	0	0	0	1.210E+02	1.0000E+00	2.124E-07	
2	2	3	-2.5150E+01	0.0000E+00	-3.4791E+01	-1.3650E+05	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	-1.777E-05	
2	2	4	-8.5852E+00	0.0000E+00	-1.7620E+01	-1.0239E+05	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	-1.212E-04	
2	2	5	-3.4861E+00	0.0000E+00	-1.0648E+01	-8.5121E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	-2.212E-04	
2	2	6	-1.6551E+00	0.0000E+00	-7.3382E+00	-7.4685E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	-5.144E-04	
2	2	7	-8.8954E+01	0.0000E+00	-5.5592E+00	-6.7493E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	-3.240E-04	
2	2	8	-5.2349E+01	0.0000E+00	-4.5122E+00	-6.2046E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	-9.607E-05	
2	2	9	-3.2821E+01	0.0000E+00	-3.8558E+00	-5.7625E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	6.351E-06	
2	2	10	-2.1471E+01	0.0000E+00	-3.4265E+00	-5.3848E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	-3.225E-04	
2	2	11	-1.4426E+01	0.0000E+00	-3.1379E+00	-5.0492E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	1.003E-04	
2	2	12	-9.8430E+02	0.0000E+00	-2.9410E+00	-4.7418E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	2.017E-04	
2	2	13	-6.7628E+02	0.0000E+00	-2.8057E+00	-4.4538E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	2.240E-04	
2	2	14	-4.6469E+02	0.0000E+00	-2.7127E+00	-4.1790E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	2.521E-04	
2	2	15	-3.1719E+02	0.0000E+00	-2.6492E+00	-3.9133E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	3.545E-04	
2	2	16	-2.1266E+02	0.0000E+00	-2.6066E+00	-3.6539E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	2.104E-04	
2	2	17	-1.3691E+02	0.0000E+00	-2.5793E+00	-3.3987E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	5.721E-05	
2	2	18	-7.9756E+03	0.0000E+00	-2.5633E+00	-3.1461E+04	0.0000E+00	1.0000E+00	0	0	0	0	9.0000E+01	1.0000E+00	-6.485E-05	

TABLE V (Continued)

EXAMPLE PROBLEM DATA PRINT

VOF3D TANK DRAIN J=2 PLANE ONLY										11/13/86		VOF3D7 03/12/87 11:06:53									
ITER	J	K	U	V	W	P	F	PS	M	N	NFP	NFS	NFO	BETA	PETA	D					
2	19	-3.3299E-03	0.0000E+00	-2.5566E+00	-2.8951E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	1.819E-04					
2	20	8.5924E-04	0.0000E+00	-2.5583E+00	-2.6447E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	4.071E-04					
2	21	5.1409E-03	0.0000E+00	-2.5685E+00	-2.3942E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	1.853E-04					
2	22	1.0153E-02	0.0000E+00	-2.5887E+00	-2.1427E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	6.446E-04					
2	23	1.6501E-02	0.0000E+00	-2.6216E+00	-1.8892E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	3.576E-04					
2	24	2.5042E-02	0.0000E+00	-2.6716E+00	-1.6325E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	2.706E-04					
2	25	3.6851E-02	0.0000E+00	-2.7453E+00	-1.3708E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	2.259E-04					
2	26	5.3334E-02	0.0000E+00	-2.8519E+00	-1.1029E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	8.206E-05					
2	27	7.6358E-02	0.0000E+00	-3.0046E+00	-8.2266E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	2.430E-04					
2	28	1.0820E-01	0.0000E+00	-3.2210E+00	-5.2839E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	1.770E-04					
2	29	1.5109E-01	0.0000E+00	-3.5231E+00	-2.1298E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0.000E+01	7.742E-01	8.343E-05					
2	30	1.5199E-01	0.0000E+00	-3.8253E+00	1.3221E+03	-6.2259E+00	1.1147E-01	5	5	5	5	5	5	0.000E+01	1.635E+00	0.000E+00					
2	31	2.4844E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	0.000E+00					
2	32	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	0.000E+00					
2	33	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	0.000E+00					
2	34	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	0.000E+00					
2	35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	0.000E+00					
2	36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	0.000E+00					
2	37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	0.000E+00					
2	38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0.000E+01	1.000E+00	0.000E+00					
3	2	-2.9489E+01	0.0000E+00	0.0000E+00	-1.3491E+05	0.0000E+00	1.0000E+00	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00					
3	2	-2.9489E+01	0.0000E+00	-2.3527E+01	-1.3491E+05	0.0000E+00	1.0000E+00	0	0	0	0	0	0	1.244E+02	1.000E+00	6.924E-06					
3	2	-1.6551E+01	0.0000E+00	-1.8226E+01	-1.1184E+05	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	-2.159E-05					
3	2	-8.5713E+00	0.0000E+00	-1.2521E+01	-9.3977E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	-5.199E-05					
3	2	-4.5210E+00	0.0000E+00	-8.8175E+00	-8.1704E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	-3.466E-04					
3	2	-6.2511E+00	0.0000E+00	-6.5726E+00	-7.3062E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	-2.822E-04					
3	2	-7.1747E+00	0.0000E+00	-5.1932E+00	-6.6621E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	-2.999E-04					
3	2	-9.1890E+00	0.0000E+00	-4.3179E+00	-6.1533E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	-9.296E-05					
3	2	-9.5913E+00	0.0000E+00	-3.7423E+00	-5.7303E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	-2.387E-04					
3	2	-3.9709E+00	0.0000E+00	-3.3569E+00	-5.3637E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	-4.265E-07					
3	2	11	-2.7006E+01	0.0000E+00	-3.0921E+00	-5.0350E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	2.643E-05				
3	2	12	-1.8559E+01	0.0000E+00	-2.9102E+00	-4.7322E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	1.789E-04				
3	2	13	-1.2893E+01	0.0000E+00	-2.7846E+00	-4.4472E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	2.146E-04				
3	2	14	-8.8149E+00	0.0000E+00	-2.6980E+00	-4.1744E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	1.339E-04				
3	2	15	-6.0189E+00	0.0000E+00	-2.6388E+00	-3.9102E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	5.320E-04				
3	2	16	-4.0359E+00	0.0000E+00	-2.5991E+00	-3.6518E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	3.018E-04				
3	2	17	-2.5982E+00	0.0000E+00	-2.5735E+00	-3.3973E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	8.599E-05				
3	2	18	-1.5126E+00	0.0000E+00	-2.5587E+00	-3.1453E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	9.122E-05				
3	2	19	-6.3288E+00	0.0000E+00	-2.5524E+00	-2.8947E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	1.057E-04				
3	2	20	1.6053E+00	0.0000E+00	-2.5540E+00	-2.6448E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	1.154E-04				
3	2	21	9.7651E+00	0.0000E+00	-2.5635E+00	-2.3947E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	3.905E-04				
3	2	22	1.9240E+00	0.0000E+00	-2.5823E+00	-2.1437E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	2.199E-04				
3	2	23	3.1330E+00	0.0000E+00	-2.6131E+00	-1.8948E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	1.888E-04				
3	2	24	4.7619E+00	0.0000E+00	-2.6598E+00	-1.6349E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	1.716E-04				
3	2	25	7.9189E+00	0.0000E+00	-2.7288E+00	-1.3744E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	1.736E-04				
3	2	26	1.0179E+01	0.0000E+00	-2.8289E+00	-1.0727E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	2.693E-04				
3	2	27	1.4612E+01	0.0000E+00	-2.9728E+00	-8.3015E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	3.558E-04				
3	2	28	2.0814E+01	0.0000E+00	-3.1781E+00	-5.3900E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	2.661E-04				
3	2	29	2.9468E+01	0.0000E+00	-3.4702E+00	-2.2771E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	9.500E+01	8.034E-01	1.949E-04				
3	2	30	2.9468E+01	0.0000E+00	-3.7624E+00	-1.1221E+03	-6.3405E+00	1.6443E-01	5	5	5	5	5	5	9.500E+01	1.505E+00	0.000E+00				
3	2	31	8.7998E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	0.000E+00					
3	2	32	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	0.000E+00					
3	2	33	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	0.000E+00					
3	2	34	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	0.000E+00					
3	2	35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	9.500E+01	1.000E+00	0.000E+00					

TABLE V (Continued)

EXAMPLE PROBLEM DATA PRINT

UOF3D TANK DRAIN J=2 PLANE ONLY										11/13/86	2020202020202020	UOF3D?	03/12/87 11:06:53				
ITER.	82	TIME	2.04000E-04	DELT	2.04000E-04	CYCLE	1	P	F	NF	NFP	NFS	NFO	BETA	PETA	D	
J	K	U	V	W	P	PS											
3	26	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	9.500E+01	1.000E+00	0.0000E+00				
3	27	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.291E+02	1.000E+00	0.0000E+00				
3	28	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	-1.000E+00	1.000E+00	0.0000E+00				
4	2	1.-1.2113E+01	0.0000E+00	0.0000E+00	-1.0057E+05	0.0000E+00	1.0000E+00	0	0	0	-1.000E+00	1.000E+00	0.0000E+00				
4	2	-1.2113E+01	0.0000E+00	-8.1633E+00	-1.0057E+05	0.0000E+00	1.0000E+00	0	0	0	1.320E+02	1.000E+00	5.984E-05				
4	2	3.-9.1646E+00	0.0000E+00	-8.7509E+00	-9.2574E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.938E-05				
4	2	-6.2172E+00	0.0000E+00	-7.7116E+00	-8.3998E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	-7.044E-06				
4	2	5.-4.0813E+00	0.0000E+00	-6.4309E+00	-7.6440E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	-8.791E-05				
4	2	6.-2.6747E+00	0.0000E+00	-5.3473E+00	-7.0138E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	-1.858E-04				
4	2	7.-1.7742E+00	0.0000E+00	-4.5276E+00	-6.4899E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	-1.747E-04				
4	2	8.-1.1961E+00	0.0000E+00	-3.9325E+00	-6.0463E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	-1.106E-04				
4	2	9.-8.1841E-01	0.0000E+00	-3.5071E+00	-5.6611E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	-3.049E-05				
4	2	10.-5.6598E-01	0.0000E+00	-3.2052E+00	-5.3175E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	-1.074E-04				
4	2	11.-3.9364E-01	0.0000E+00	-2.9918E+00	-5.0036E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	5.295E-05				
4	2	12.-2.7411E-01	0.0000E+00	-2.8419E+00	-4.7106E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.242E-04				
4	2	13.-1.9042E-01	0.0000E+00	-2.7372E+00	-4.4322E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.460E-04				
4	2	14.-1.3153E-01	0.0000E+00	-2.6648E+00	-4.1642E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	4.052E-05				
4	2	15.-8.9900E-02	0.0000E+00	-2.6152E+00	-3.9032E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.253E-04				
4	2	16.-6.0241E-02	0.0000E+00	-2.5820E+00	-3.6471E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	2.232E-04				
4	2	17.-3.8731E-02	0.0000E+00	-2.5606E+00	-3.3943E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.939E-04				
4	2	18.-2.2519E-02	0.0000E+00	-2.5482E+00	-3.1435E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	2.169E-04				
4	2	19.-9.4191E-03	0.0000E+00	-2.5430E+00	-2.8940E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.890E-04				
4	2	20.-2.3792E-03	0.0000E+00	-2.5443E+00	-2.6450E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.378E-04				
4	2	21.-1.4514E-02	0.0000E+00	-2.5522E+00	-2.3958E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	8.696E-05				
4	2	22.-2.8670E-02	0.0000E+00	-2.5580E+00	-2.1459E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.873E-04				
4	2	23.-4.6784E-02	0.0000E+00	-2.5593E+00	-1.8944E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.102E-04				
4	2	24.-7.1297E-02	0.0000E+00	-2.6330E+00	-1.6405E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.997E-04				
4	2	25.-1.0541E-01	0.0000E+00	-2.6913E+00	-1.3826E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.218E-04				
4	2	26.-1.5341E-01	0.0000E+00	-2.7763E+00	-1.1191E+04	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.179E-04				
4	2	27.-2.2111E-01	0.0000E+00	-2.8991E+00	-8.4716E+03	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.695E-04				
4	2	28.-3.1631E-01	0.0000E+00	-3.0753E+00	-5.6323E+03	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	1.220E-04				
4	2	29.-4.4931E-01	0.0000E+00	-3.3262E+00	-2.6202E+03	0.0000E+00	1.0000E+00	0	0	0	1.142E+02	1.000E+00	8.719E-01	8.472E-05			
4	2	30.-4.4931E-01	0.0000E+00	-3.5771E+00	-6.3783E+02	-6.9015E+02	2.9862E-01	5	5	5	1.142E+02	1.252E+00	0.0000E+00				
4	2	31.-2.4801E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.142E+02	1.000E+00	0.0000E+00				
4	2	32.-8.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.142E+02	1.000E+00	0.0000E+00				
4	2	33.-3.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.142E+02	1.000E+00	0.0000E+00				
4	2	34.-8.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.142E+02	1.000E+00	0.0000E+00				
4	2	35.-8.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.142E+02	1.000E+00	0.0000E+00				
4	2	36.-8.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.142E+02	1.000E+00	0.0000E+00				
4	2	37.-8.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.142E+02	1.000E+00	0.0000E+00				
4	2	38.-8.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.142E+02	1.000E+00	0.0000E+00				
5	2	1.-0.0000E+00	0.0000E+00	-8.4987E+04	0.0000E+00	1.0000E+00	0	0	0	-1.000E+00	1.000E+00	1.637E+02	1.000E+00	0.0000E+00			
5	2	2.-0.0000E+00	0.0000E+00	-4.2933E+00	-8.4987E+04	0.0000E+00	1.0000E+00	0	0	0	5.857E+01	1.000E+00	2.663E-04				
5	2	3.-5.8446E+00	0.0000E+00	-4.8824E+00	-8.0781E+04	0.0000E+00	1.0000E+00	0	0	0	1.192E+02	1.000E+00	1.750E-04				
5	2	4.-4.5089E+00	0.0000E+00	-4.9082E+00	-7.5998E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	9.686E-05				
5	2	5.-3.3280E+00	0.0000E+00	-4.5847E+00	-7.1189E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	2.279E-05				
5	2	6.-2.4051E+00	0.0000E+00	-4.1654E+00	-6.6697E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	-4.481E-05				
5	2	7.-1.7235E+00	0.0000E+00	-3.7688E+00	-6.2616E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	-1.048E-04				
5	2	8.-1.2318E+00	0.0000E+00	-3.4368E+00	-5.8924E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	-6.743E-05				
5	2	9.-8.7932E-01	0.0000E+00	-3.1758E+00	-5.5558E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	-3.397E-05				
5	2	10.-6.2583E-01	0.0000E+00	-2.9790E+00	-5.2447E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	3.005E-05				
5	2	11.-4.4297E-01	0.0000E+00	-2.8349E+00	-4.9529E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	6.068E-05				
5	2	12.-3.1124E-01	0.0000E+00	-2.7320E+00	-4.6753E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.119E-04				
5	2	13.-2.1690E-01	0.0000E+00	-2.6599E+00	-4.4077E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.004E-04				
5	2	14.-1.4976E-01	0.0000E+00	-2.6101E+00	-4.1473E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	7.243E-05				

TABLE V (Continued)

EXAMPLE PROBLEM DATA PRINT

UOF3D TANK DRAIN J-2 PLANE ONLY			11/13/86	2070207020702070	UOF3D7	03/12/87 11:06:53							
ITER-	B2	TIME- 2.04000E-04	DELT= 2.04000E-04	CYCLE- 1									
J	K,	U	U	P	PS	F	NF	NFP	NFS	NFO	BETA	PETA	D
1	2 15	-1.0215E-01	0.0000E+00	-2.5763E+00	-3.8917E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.959E-04
1	2 16	-6.8282E-02	0.0000E+00	-2.5538E+00	-3.6394E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.280E-04
1	2 17	-4.3793E-02	0.0000E+00	-2.5394E+00	-3.3893E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.651E-04
1	2 18	-2.5418E-02	0.0000E+00	-2.5310E+00	-3.1406E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.366E-04
1	2 19	-1.0620E-02	0.0000E+00	-2.5275E+00	-2.8928E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	2.052E-04
1	2 20	2.6647E-03	0.0000E+00	-2.5284E+00	-2.6453E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.187E-04
1	2 21	1.6332E-02	0.0000E+00	-2.5337E+00	-2.3977E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	5.803E-05
1	2 22	3.2330E-02	0.0000E+00	-2.5442E+00	-2.1496E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	5.823E-05
1	2 23	5.2904E-02	0.0000E+00	-2.5616E+00	-1.9005E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	2.744E-05
1	2 24	8.0918E-02	0.0000E+00	-2.5883E+00	-1.6496E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	6.624E-05
1	2 25	1.2019E-01	0.0000E+00	-2.6283E+00	-1.3962E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	6.110E-05
1	2 26	1.7592E-01	0.0000E+00	-2.6874E+00	-1.1388E+04	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.233E-04
1	2 27	2.5515E-01	0.0000E+00	-2.7742E+00	-8.7561E+03	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	1.084E-04
1	2 28	3.6771E-01	0.0000E+00	-2.9008E+00	-6.0393E+03	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	6.526E-05
1	2 29	5.2944E-01	0.0000E+00	-3.0872E+00	-3.1984E+03	0.0000E+00	1.0000E+00	0	0	0	1.093E+02	1.000E+00	5.238E-05
1	2 30	5.2944E-01	0.0000E+00	-3.2737E+00	-1.7463E+02	-7.3962E+00	5.5198E-01	5	5	5	1.093E+02	9.506E-01	0.000E+00
1	2 31	4.7114E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.093E+02	1.000E+00	0.000E+00
1	2 32	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.093E+02	1.000E+00	0.000E+00
1	2 33	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.093E+02	1.000E+00	0.000E+00
1	2 34	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.093E+02	1.000E+00	0.000E+00
1	2 35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.093E+02	1.000E+00	0.000E+00
1	2 36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.093E+02	1.000E+00	0.000E+00
1	2 37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.570E+02	1.000E+00	0.000E+00
1	2 38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	-1.000E+00	1.000E+00	0.000E+00
6	2 1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0000E+00	0	0	0	-1.000E+00	1.000E+00	0.000E+00
6	2 2	0.0000E+00	0.0000E+00	0.0000E+00	-7.9482E+04	0.0000E+00	1.0000E+00	0	0	0	-1.000E+00	1.000E+00	0.000E+00
6	2 3	-4.0005E+00	0.0000E+00	-3.2962E+00	-7.3977E+04	0.0000E+00	1.0000E+00	0	0	0	1.080E+02	1.000E+00	1.056E-04
6	2 4	-3.4138E+00	0.0000E+00	-3.5059E+00	-7.0748E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	1.493E-04
6	2 5	-2.6943E+00	0.0000E+00	-3.4887E+00	-6.7314E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	7.449E-05
6	2 6	-2.0691E+00	0.0000E+00	-3.3561E+00	-6.3897E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	-4.565E-05
6	2 7	-1.5599E+00	0.0000E+00	-3.1850E+00	-6.0609E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	-8.350E-05
6	2 8	-1.1602E+00	0.0000E+00	-3.0181E+00	-5.7490E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	-2.134E-05
6	2 9	-8.5277E-01	0.0000E+00	-2.8748E+00	-5.4534E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	-1.793E-05
6	2 10	-6.1806E-01	0.0000E+00	-2.7618E+00	-5.1718E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	9.625E-06
6	2 11	-4.4087E-01	0.0000E+00	-2.6785E+00	-4.9014E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	2.692E-05
6	2 12	-3.0971E-01	0.0000E+00	-2.6200E+00	-4.6391E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	3.789E-05
6	2 13	-2.1483E-01	0.0000E+00	-2.5802E+00	-4.3825E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	1.046E-04
6	2 14	-1.4742E-01	0.0000E+00	-2.5536E+00	-4.1298E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	1.265E-04
6	2 15	-9.9505E-02	0.0000E+00	-2.5361E+00	-3.8798E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	6.272E-05
6	2 16	-6.6458E-02	0.0000E+00	-2.5247E+00	-3.6314E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	7.757E-05
6	2 17	-4.2454E-02	0.0000E+00	-2.5175E+00	-3.3842E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	7.920E-05
6	2 18	-2.4573E-02	0.0000E+00	-2.5135E+00	-3.1377E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	1.053E-04
6	2 19	-1.0262E-02	0.0000E+00	-2.5117E+00	-2.8916E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	8.470E-05
6	2 20	2.5539E-03	0.0000E+00	-2.5121E+00	-2.6456E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	1.195E-04
6	2 21	1.5730E-02	0.0000E+00	-2.5147E+00	-2.3996E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	1.118E-04
6	2 22	3.1190E-02	0.0000E+00	-2.5198E+00	-2.1534E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	1.394E-05
6	2 23	5.1194E-02	0.0000E+00	-2.5283E+00	-1.9066E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	-6.668E-06
6	2 24	7.8638E-02	0.0000E+00	-2.5416E+00	-1.6590E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	7.563E-05
6	2 25	1.1748E-01	0.0000E+00	-2.5621E+00	-1.4102E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	6.052E-05
6	2 26	1.7321E-01	0.0000E+00	-2.5933E+00	-1.1593E+04	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	2.976E-05
6	2 27	2.5309E-01	0.0000E+00	-2.6406E+00	-9.0532E+03	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	5.518E-05
6	2 28	3.6505E-01	0.0000E+00	-2.7087E+00	-6.4674E+03	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	6.996E-05
6	2 29	5.0832E-01	0.0000E+00	-2.7900E+00	-3.8147E+03	0.0000E+00	1.0000E+00	0	0	0	1.001E+02	1.000E+00	6.540E-05
6	2 30	5.0832E-01	0.0000E+00	-2.8712E+00	-1.0824E+03	-8.6578E+00	8.8990E-01	5	5	5	1.001E+02	7.195E-01	0.000E+00
6	2 31	5.0832E-01	0.0000E+00	-2.84349E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	1.001E+02	1.000E+00	0.000E+00

TABLE V (Continued)

EXAMPLE PROBLEM DATA PRINT

UOF3D TANK DRAIN J-2 PLANE ONLY										11/13/86		7070707070707070		UOF3D7		03/12/87 11106153	
ITER.	82	TIME	2.04000E-04	DELT.	2.04000E-04	CYCLE	1	P	F	PS	NF	NFP	NFS	NFO	BETA	PETA	D
I	J	K	U	V	W												
6	2	32	1.5914E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	1.001E+02	1.000E+00	0.000E+00	
6	2	33	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	1.001E+02	1.000E+00	0.000E+00	
6	2	34	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	1.001E+02	1.000E+00	0.000E+00	
6	2	35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	1.001E+02	1.000E+00	0.000E+00	
6	2	36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	1.001E+02	1.000E+00	0.000E+00	
6	2	37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	1.387E+02	1.000E+00	0.000E+00	
6	2	38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	-1.000E+00	1.000E+00	0.000E+00	
7	2	1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0000E+00	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00	
7	2	2	0.0000E+00	0.0000E+00	0.0000E+00	-5.9810E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
7	2	3	0.0000E+00	0.0000E+00	-2.6729E+00	-6.9810E+04	0.0000E+00	1.0000E+00	0	0	0	0	2.006E+01	1.000E+00	6.619E-04		
7	2	4	-2.6137E+00	0.0000E+00	-2.7414E+00	-6.7192E+04	0.0000E+00	1.0000E+00	0	0	0	0	1.082E+02	1.000E+00	1.224E-04		
7	2	5	-2.2183E+00	0.0000E+00	-2.8244E+00	-6.4507E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	1.200E-04		
7	2	6	-1.7713E+00	0.0000E+00	-2.8150E+00	-6.1741E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	1.143E-05		
7	2	7	-1.3796E+00	0.0000E+00	-2.7602E+00	-5.8984E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	-4.309E-05		
7	2	8	-1.0519E+00	0.0000E+00	-2.6915E+00	-5.6281E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	-6.416E-05		
7	2	9	-7.8764E-01	0.0000E+00	-2.6254E+00	-5.3646E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	2.528E-05		
7	2	10	-5.7452E-01	0.0000E+00	-2.5735E+00	-5.1075E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	-9.440E-06		
7	2	11	-4.0744E-01	0.0000E+00	-2.5391E+00	-4.8554E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	-9.398E-05		
7	2	12	-2.8230E-01	0.0000E+00	-2.5192E+00	-4.6068E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	-2.613E-05		
7	2	13	-1.9283E-01	0.0000E+00	-2.5086E+00	-4.3601E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	6.013E-05		
7	2	14	-1.3056E-01	0.0000E+00	-2.5032E+00	-4.1145E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	7.239E-05		
7	2	15	-8.7597E-02	0.0000E+00	-2.5005E+00	-3.8694E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	1.626E-04		
7	2	16	-5.7809E-02	0.0000E+00	-2.4992E+00	-3.6245E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	4.172E-05		
7	2	17	-3.6736E-02	0.0000E+00	-2.4986E+00	-3.3798E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	8.492E-05		
7	2	18	-2.1196E-02	0.0000E+00	-2.4983E+00	-3.1351E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	6.355E-05		
7	2	19	-8.8368E-03	0.0000E+00	-2.4981E+00	-2.8905E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	9.602E-05		
7	2	20	-2.1825E-03	0.0000E+00	-2.4981E+00	-2.6459E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	5.124E-05		
7	2	21	-1.3508E-02	0.0000E+00	-2.4982E+00	-2.4012E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	4.564E-05		
7	2	22	-2.6828E-02	0.0000E+00	-2.4985E+00	-2.1566E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	1.227E-05		
7	2	23	-4.4151E-02	0.0000E+00	-2.4991E+00	-1.9120E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	-1.240E-04		
7	2	24	-6.8114E-02	0.0000E+00	-2.5003E+00	-1.6672E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	-2.799E-05		
7	2	25	1.0244E-01	0.0000E+00	-2.5029E+00	-1.4224E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	2.845E-06		
7	2	26	1.5271E-01	0.0000E+00	-2.5085E+00	-1.1773E+04	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	8.980E-06		
7	2	27	2.2772E-01	0.0000E+00	-2.5215E+00	-9.3168E+03	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	1.618E-04		
7	2	28	3.4336E-01	0.0000E+00	-2.5565E+00	-6.8477E+03	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	1.088E-04		
7	2	29	5.3674E-01	0.0000E+00	-2.6685E+00	-4.3442E+03	0.0000E+00	1.0000E+00	0	0	0	0	8.960E+01	1.000E+00	-6.301E-05		
7	2	30	9.3262E-01	0.0000E+00	-2.1547E+00	-1.7310E+03	0.0000E+00	9.9887E-01	0	0	0	0	8.960E+01	9.046E-01	-7.118E-05		
7	2	31	9.3262E-01	0.0000E+00	-2.6943E+00	3.7853E+02	-9.5724E+00	3.1307E-01	5	5	5	5	8.960E+01	1.231E+00	0.000E+00		
7	2	32	7.5017E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	8.960E+01	1.000E+00	0.000E+00			
7	2	33	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	8.960E+01	1.000E+00	0.000E+00			
7	2	34	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	8.960E+01	1.000E+00	0.000E+00			
7	2	35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	8.960E+01	1.000E+00	0.000E+00			
7	2	36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	8.960E+01	1.000E+00	0.000E+00			
7	2	37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	1.193E+02	1.000E+00	0.000E+00			
7	2	38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	0	8	-1.000E+00	1.000E+00	0.000E+00			
8	2	1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1	0	0	0	-1.000E+00	1.000E+00	0.000E+00			
8	2	2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00			
8	2	3	0.0000E+00	0.0000E+00	0.0000E+00	-6.7300E+04	0.0000E+00	1	0	0	0	-1.000E+00	1.000E+00	0.000E+00			
8	2	4	0.0000E+00	0.0000E+00	-2.3768E+00	-6.4790E+04	0.0000E+00	1	0	0	0	3.460E+01	1.000E+00	5.498E-04			
8	2	5	-1.8192E+00	0.0000E+00	-2.4053E+00	-6.2469E+04	0.0000E+00	1	0	0	0	9.330E+01	1.000E+00	7.376E-05			
8	2	6	-1.5329E+00	0.0000E+00	-2.4479E+00	-6.0113E+04	0.0000E+00	1	0	0	0	7.764E+01	1.000E+00	5.300E-05			
8	2	7	-1.2171E+00	0.0000E+00	-2.4530E+00	-5.7716E+04	0.0000E+00	1	0	0	0	7.764E+01	1.000E+00	-4.347E-05			
8	2	8	-9.3879E-01	0.0000E+00	-2.4438E+00	-5.5314E+04	0.0000E+00	1	0	0	0	7.764E+01	1.000E+00	6.197E-06			
8	2	9	-7.1225E-01	0.0000E+00	-2.4256E+00	-5.2922E+04	0.0000E+00	1	0	0	0	7.764E+01	1.000E+00	-1.226E-05			
8	2	10	-5.1579E-01	0.0000E+00	-2.4169E+00	-5.0546E+04	0.0000E+00	1	0	0	0	7.764E+01	1.000E+00	-2.506E-05			

TABLE V (Continued)

EXAMPLE PROBLEM DATA PRINT

17

ITER-	82	UOF3D TANK DRAIN J-2 PLANE ONLY				TIME- 2.04000E-04	DELT- 2.04000E-04	11/13/86 7070707070707070 UOF3D7 03/12/87 11:06:53									
		J	K	U	U			P	F	PS	NF	NFP	NFS	NFO			
8	11	-3.5664E-01	0.0000E+00	-2.4218E+00	-4.8180E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-6.956E-06	
8	12	-2.3878E-01	0.0000E+00	-2.4353E+00	-4.5809E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-1.047E-04	
8	13	-1.5843E-01	0.0000E+00	-2.4502E+00	-4.3424E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-1.671E-05	
8	14	-1.0507E-01	0.0000E+00	-2.4629E+00	-4.1025E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	1.702E-04	
8	15	-6.9558E-02	0.0000E+00	-2.4726E+00	-3.8613E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	6.558E-05	
8	16	-4.5511E-02	0.0000E+00	-2.4794E+00	-3.6192E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-1.327E-05	
8	17	-2.8763E-02	0.0000E+00	-2.4840E+00	-3.3764E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	3.917E-06	
8	18	-1.6537E-02	0.0000E+00	-2.4867E+00	-3.1332E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	6.898E-05	
8	19	-6.8828E-03	0.0000E+00	-2.4878E+00	-2.8897E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	2.908E-05	
8	20	1.6946E-03	0.0000E+00	-2.4875E+00	-2.6461E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	9.402E-06	
8	21	1.0598E-02	0.0000E+00	-2.4857E+00	-2.4025E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	2.351E-05	
8	22	2.0891E-02	0.0000E+00	-2.4823E+00	-2.1591E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	4.358E-05	
8	23	3.4433E-02	0.0000E+00	-2.4767E+00	-1.9160E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-1.253E-04	
8	24	5.3279E-02	0.0000E+00	-2.4682E+00	-1.6735E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-5.860E-05	
8	25	8.0473E-02	0.0000E+00	-2.4558E+00	-1.4318E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-6.669E-06	
8	26	1.2063E-01	0.0000E+00	-2.4382E+00	-1.1914E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-1.313E-04	
8	27	1.8894E-01	0.0000E+00	-2.4131E+00	-9.5261E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	8.194E-06	
8	28	2.7267E-01	0.0000E+00	-2.3752E+00	-7.1633E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-5.804E-05	
8	29	4.1098E-01	0.0000E+00	-2.2974E+00	-4.8376E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-4.253E-05	
8	30	5.9266E-01	0.0000E+00	-2.0150E+00	-2.5882E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	7.764E+01	1.0000E+00	-9.530E-05	
8	31	5.9266E-01	0.0000E+00	-1.7326E+00	-6.1559E+02	-1.1403E+01	8.0438E-01	5	5	5	5	5	5	7.764E+01	7.666E-01	0.0000E+00	
8	32	5.9266E-01	0.0000E+00	-5.4748E-01	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	8	7.764E+01	1.0000E+00	0.0000E+00	
8	33	2.2037E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	8	7.764E+01	1.0000E+00	0.0000E+00	
8	34	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	8	7.764E+01	1.0000E+00	0.0000E+00	
8	35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	8	7.764E+01	1.0000E+00	0.0000E+00	
8	36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	8	7.764E+01	1.0000E+00	0.0000E+00	
8	37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	8	9.898E+01	1.0000E+00	0.0000E+00	
8	38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00	
9	2	1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0000E+00	0	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
9	2	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0000E+00	0	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
9	2	3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0000E+00	0	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
9	2	4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.0000E+00	0	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
9	2	5	0.0000E+00	0.0000E+00	0.0000E+00	-6.2905E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	3.424E+01	1.0000E+00	4.801E-04
9	2	6	-1.3097E+00	0.0000E+00	-2.1727E+00	-6.1619E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	3.424E+01	1.0000E+00	6.789E-05
9	2	7	-1.0825E+00	0.0000E+00	-2.2269E+00	-5.6747E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-3.262E-05
9	2	8	-8.3122E+01	0.0000E+00	-2.2597E+00	-5.4567E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-1.839E-05
9	2	9	-6.4398E+01	0.0000E+00	-2.2660E+00	-5.2354E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-6.963E-05
9	2	10	-4.5413E+01	0.0000E+00	-2.2876E+00	-5.0136E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-2.913E-05
9	2	11	-2.9714E+01	0.0000E+00	-2.3260E+00	-4.7896E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	3.361E-05
9	2	12	-1.8590E+01	0.0000E+00	-2.3700E+00	-4.5618E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-4.080E-05
9	2	13	-1.1787E+01	0.0000E+00	-2.4068E+00	-4.3298E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-1.518E-05
9	2	14	-7.6178E+02	0.0000E+00	-2.4340E+00	-4.0941E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	2.452E-05
9	2	15	-4.9713E+02	0.0000E+00	-2.4530E+00	-3.8558E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	8.757E-05
9	2	16	-3.2268E+02	0.0000E+00	-2.4658E+00	-3.6156E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-2.538E-05
9	2	17	-2.0299E+02	0.0000E+00	-2.4740E+00	-3.3741E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-4.559E-05
9	2	18	-1.1642E+02	0.0000E+00	-2.4788E+00	-3.1319E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-5.236E-05
9	2	19	-4.8375E+03	0.0000E+00	-2.4808E+00	-2.8891E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-7.092E-05
9	2	20	1.1892E+03	0.0000E+00	-2.4804E+00	-2.6462E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-6.150E-05
9	2	21	7.3832E+03	0.0000E+00	-2.4773E+00	-2.4033E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-4.558E-05
9	2	22	1.4683E+02	0.0000E+00	-2.4713E+00	-2.1607E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-6.298E-05
9	2	23	2.4220E+02	0.0000E+00	-2.4613E+00	-1.9188E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-1.283E-04
9	2	24	3.7544E+02	0.0000E+00	-2.4461E+00	-1.6777E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-1.270E-04
9	2	25	5.6832E+02	0.0000E+00	-2.4232E+00	-1.4382E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-1.183E-04
9	2	26	8.5442E+02	0.0000E+00	-2.3891E+00	-1.2010E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+01	1.0000E+00	-1.152E-04
9	2	27	1.2854E+01	0.0000E+00	-2.3386E+00	-9.6702E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	6.435E+		

TABLE V (Continued)

EXAMPLE PROBLEM DATA PRINT

17

UOF3D TANK DRAIN J=2 PLANE ONLY										11/13/86		2070207020702070		UOF3D7		03/12/87 11:06:53	
ITER-	82	TIME-	2.04000E-04	DELT-	2.04000E-04	CYCLE-	1	P	F	NF	NFP	NFS	NFO	BETA	PETA	D	
1	J, K,	U	V	U	P												
9	2 28	1.9399E-01	0.0000E+00	-2.2628E+00	-7.3805E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	6.435E+01	1.0000E+00	1.336E-04		
9	2 29	2.9319E-01	0.0000E+00	-2.1498E+00	-5.1649E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	6.435E+01	1.0000E+00	-4.619E-05		
9	2 30	4.4187E-01	0.0000E+00	-2.0134E+00	-3.0602E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	6.435E+01	1.0000E+00	-1.492E-05		
9	2 31	6.9187E-01	0.0000E+00	-1.2598E+00	-1.0893E+03	0.0000E+00	9.9887E-01	0	0	0	0	0	6.435E+01	9.538E-01	-3.909E-05		
9	2 32	7.3382E-01	0.0000E+00	-1.4724E+00	1.4290E+02	-1.2333E+01	3.7183E-01	5	5	5	5	5	6.435E+01	1.148E+00	0.0000E+00		
9	2 33	7.2950E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	6.435E+01	1.0000E+00	0.0000E+00		
9	2 34	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	6.435E+01	1.0000E+00	0.0000E+00		
9	2 35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	6.435E+01	1.0000E+00	0.0000E+00		
9	2 36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	6.435E+01	1.0000E+00	0.0000E+00		
9	2 37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	7.836E+01	1.0000E+00	0.0000E+00		
9	2 38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	-1.0000E+00	1.0000E+00	0.0000E+00		
10	2 1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
10	2 2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
10	2 3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
10	2 4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
10	2 5	0.0000E+00	0.0000E+00	0.0000E+00	-5.9518E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
10	2 6	0.0000E+00	0.0000E+00	-2.0412E+00	-5.8016E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	2.547E+01	1.0000E+00	-1.021E-04		
10	2 7	-9.2018E-01	0.0000E+00	-2.0541E+00	-5.5601E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	6.248E+01	1.0000E+00	-7.690E-05		
10	2 8	-7.2712E-01	0.0000E+00	-2.1310E+00	-5.4006E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	4.161E-05		
10	2 9	-5.9655E-01	0.0000E+00	-2.1355E+00	-5.1920E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-2.578E-05		
10	2 10	-3.9593E-01	0.0000E+00	-2.1797E+00	-4.9830E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-5.885E-05		
10	2 11	-2.3434E-01	0.0000E+00	-2.2496E+00	-4.7696E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-8.816E-07		
10	2 12	-1.2855E-01	0.0000E+00	-2.3232E+00	-4.5493E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	7.616E-07		
10	2 13	-7.6544E-02	0.0000E+00	-2.3782E+00	-4.3218E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-2.761E-05		
10	2 14	-4.8165E-02	0.0000E+00	-2.4158E+00	-4.0889E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	1.324E-05		
10	2 15	-3.1063E-02	0.0000E+00	-2.4410E+00	-3.8524E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	1.810E-05		
10	2 16	-2.0045E-02	0.0000E+00	-2.4576E+00	-3.6134E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-1.042E-04		
10	2 17	-1.2568E-02	0.0000E+00	-2.4681E+00	-3.3727E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-1.308E-04		
10	2 18	-7.1914E-03	0.0000E+00	-2.4741E+00	-3.1311E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-5.750E-05		
10	2 19	-2.9877E-03	0.0000E+00	-2.4767E+00	-2.8888E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-1.723E-04		
10	2 20	7.2644E-04	0.0000E+00	-2.4761E+00	-2.6463E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-1.694E-04		
10	2 21	4.5598E-03	0.0000E+00	-2.4723E+00	-2.4038E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-1.563E-04		
10	2 22	9.0627E-03	0.0000E+00	-2.4647E+00	-2.1617E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-2.344E-04		
10	2 23	1.4961E-02	0.0000E+00	-2.4522E+00	-1.9204E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-1.631E-04		
10	2 24	2.3214E-02	0.0000E+00	-2.4328E+00	-1.6803E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-2.540E-04		
10	2 25	3.5192E-02	0.0000E+00	-2.4035E+00	-1.4421E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-2.240E-05		
10	2 26	5.2996E-02	0.0000E+00	-2.3595E+00	-1.2067E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	1.382E-04		
10	2 27	7.9860E-02	0.0000E+00	-2.2936E+00	-9.7569E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	8.391E-05		
10	2 28	1.2081E-01	0.0000E+00	-2.1946E+00	-7.5112E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	6.428E-05		
10	2 29	1.8412E-01	0.0000E+00	-2.0476E+00	-5.3626E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-1.073E-05		
10	2 30	2.8685E-01	0.0000E+00	-1.8415E+00	-3.3581E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	5.007E+01	1.0000E+00	-9.113E-05		
10	2 31	4.8835E-01	0.0000E+00	-1.2629E+00	-1.5556E+03	0.0000E+00	9.9967E-01	0	0	0	0	0	5.007E+01	1.0000E+00	3.541E-05		
10	2 32	6.8286E-01	0.0000E+00	-1.2629E+00	-2.1310E+02	-5.8452E+00	9.9412E-01	2	5	2	2	2	5.007E+01	6.456E-01	0.0000E+00		
10	2 33	4.9345E-01	0.0000E+00	-9.4443E-02	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	5.007E+01	1.0000E+00	0.0000E+00		
10	2 34	3.4592E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	5.007E+01	1.0000E+00	0.0000E+00		
10	2 35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	5.007E+01	1.0000E+00	0.0000E+00		
10	2 36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	5.007E+01	1.0000E+00	0.0000E+00		
10	2 37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	8	5.816E+01	1.0000E+00	0.0000E+00		
10	2 38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
11	2 1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
11	2 2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
11	2 3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
11	2 4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
11	2 5	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		
11	2 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	-5.6763E+04	0.0000E+00	1	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00		

TABLE V (Continued)

EXAMPLE PROBLEM DATA PRINT

UOF3D TANK DRAIN J-2 PLANE ONLY										11/13/86		UOF3D7 03/12/87 11:06:53									
I	J	K	ITER.	82	TIME-	2.04000E-04	DELT-	2.04000E-04	CYCLE-	1	P	S	F	NF	NFP	NFS	NFO	BETA	PETA	D	
11	2	7	0	0.0000E+00	0.0000E+00	-1.9455E+00	-5.5510E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	1.357E+01	1.000E+00	-2.823E-04		
11	2	8	0	0.0000E+00	0.0000E+00	-2.0575E+00	-5.3605E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	5.308E+01	1.000E+00	-4.170E-05		
11	2	9	-5.8218E-01	0.0000E+00	-2.0226E+00	-5.1591E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-1.993E-04		
11	2	10	-3.4291E-01	0.0000E+00	-2.0888E+00	-4.9611E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-1.509E-04		
11	2	11	-1.7270E-01	0.0000E+00	-2.1900E+00	-4.7566E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-7.484E-05		
11	2	12	-7.1192E-02	0.0000E+00	-2.2940E+00	-4.5422E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-6.617E-05		
11	2	13	-3.9312E-02	0.0000E+00	-2.3622E+00	-4.3176E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-9.988E-05		
11	2	14	-2.4210E-02	0.0000E+00	-2.4062E+00	-4.0863E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-3.034E-05		
11	2	15	-1.5501E-02	0.0000E+00	-2.4348E+00	-3.8507E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-2.778E-05		
11	2	16	-9.9707E-03	0.0000E+00	-2.4534E+00	-3.6123E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-1.559E-04		
11	2	17	-6.2496E-03	0.0000E+00	-2.4651E+00	-3.3721E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-3.239E-04		
11	2	18	-3.5742E-03	0.0000E+00	-2.4718E+00	-3.1307E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-2.836E-04		
11	2	19	-1.4910E-03	0.0000E+00	-2.4746E+00	-2.8886E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-3.382E-04		
11	2	20	3.5947E-04	0.0000E+00	-2.4739E+00	-2.6463E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-5.947E-05		
11	2	21	2.2474E-03	0.0000E+00	-2.4698E+00	-2.4041E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-4.061E-04		
11	2	22	4.4924E-03	0.0000E+00	-2.4614E+00	-2.1622E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-2.205E-04		
11	2	23	7.4184E-03	0.0000E+00	-2.4475E+00	-1.9212E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-1.082E-04		
11	2	24	1.1512E-02	0.0000E+00	-2.4261E+00	-1.6816E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-2.522E-04		
11	2	25	1.7475E-02	0.0000E+00	-2.3935E+00	-1.4440E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	6.241E-05		
11	2	26	2.6325E-02	0.0000E+00	-2.3444E+00	-1.2096E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	7.014E-05		
11	2	27	3.9707E-02	0.0000E+00	-2.2706E+00	-9.8009E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-2.327E-05		
11	2	28	6.0165E-02	0.0000E+00	-2.1590E+00	-7.5778E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	1.808E-04		
11	2	29	9.2066E-02	0.0000E+00	-1.9898E+00	-5.4641E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	1.110E-04		
11	2	30	1.4438E-01	0.0000E+00	-1.7282E+00	-3.5163E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	-8.944E-05		
11	2	31	2.3617E-01	0.0000E+00	-1.2629E+00	-1.8249E+03	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	2.026E-04		
11	2	32	3.1804E-01	0.0000E+00	-6.0220E-01	-5.8968E+02	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.100E+00	2.085E-04		
11	2	33	2.3989E-01	0.0000E+00	-1.3472E-01	-4.0535E+01	-1.6753E+01	7.0102E-01	2	5	2	2	2	2	2	2	3.763E+01	8.326E-01	0.000E+00		
11	2	34	5.2418E-02	0.0000E+00	-2.1400E-02	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	0.000E+00		
11	2	35	3.4436E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	0.000E+00		
11	2	36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	3.763E+01	1.000E+00	0.000E+00		
11	2	37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	4.202E+01	1.000E+00	0.000E+00		
11	2	38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	5	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	7	0.0000E+00	0.0000E+00	0.0000E+00	-5.5510E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	8	0.0000E+00	0.0000E+00	0.0000E+00	-5.2456E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	-1.000E+00	1.000E+00	0.000E+00		
12	2	9	0.0000E+00	0.0000E+00	-1.9030E+00	-5.1306E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	2.704E+01	1.000E+00	-9.059E-05		
12	2	10	0.0000E+00	0.0000E+00	-2.0037E+00	-4.9443E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	4.993E+01	1.000E+00	-1.551E-04		
12	2	11	0.0000E+00	0.0000E+00	-2.1393E+00	-4.7482E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	5.918E+01	1.000E+00	-7.228E-05		
12	2	12	0.0000E+00	0.0000E+00	-2.2781E+00	-4.5387E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	6.104E+01	1.000E+00	-3.254E-05		
12	2	13	0.0000E+00	0.0000E+00	-2.3547E+00	-4.3157E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	6.104E+01	1.000E+00	-1.017E-05		
12	2	14	0.0000E+00	0.0000E+00	-2.4019E+00	-4.0851E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	6.104E+01	1.000E+00	-1.126E-04		
12	2	15	0.0000E+00	0.0000E+00	-2.4321E+00	-3.8499E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	6.104E+01	1.000E+00	2.900E-05		
12	2	16	0.0000E+00	0.0000E+00	-2.4515E+00	-3.6118E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	6.104E+01	1.000E+00	-1.618E-04		
12	2	17	0.0000E+00	0.0000E+00	-2.4637E+00	-3.3717E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	6.104E+01	1.000E+00	-1.478E-04		
12	2	18	0.0000E+00	0.0000E+00	-2.4707E+00	-3.1305E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	6.104E+01	1.000E+00	-1.999E-04		
12	2	19	0.0000E+00	0.0000E+00	-2.4737E+00	-2.8886E+04	0.0000E+00	1.0000E+00	0	0	0	0	0	0	0	0	6.104E+01	1.000E+00	-1.499E-04		
12	2	20	0																		

TABLE V (Continued)

EXAMPLE PROBLEM DATA PRINT

2

ITER-		82		TIME= 2.04000E-04		DELT= 2.04000E-04		11/13/86		?0?0?0?0?0?0?0?0?		UOF3D7		03/12/87 11:06:53	
I	J	K	U	V	W	P	PS	F	NF	MFP	NFS	NFO	BETA	PETA	D
12	2	24	0.0000E+00	0.0000E+00	-2.4231E+00	-1.6821E+04	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	-1.238E-04
12	2	25	0.0000E+00	0.0000E+00	-2.3800E+00	-1.4449E+04	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	-5.823E-05
12	2	26	0.0000E+00	0.0000E+00	-2.3377E+00	-1.2109E+04	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	1.753E-04
12	2	27	0.0000E+00	0.0000E+00	-2.2603E+00	-9.8204E+03	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	-1.444E-06
12	2	28	0.0000E+00	0.0000E+00	-2.1431E+00	-7.6073E+03	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	1.388E-04
12	2	29	0.0000E+00	0.0000E+00	-1.9636E+00	-5.5092E+03	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	3.909E-05
12	2	30	0.0000E+00	0.0000E+00	-1.6823E+00	-3.5870E+03	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	1.943E-05
12	2	31	0.0000E+00	0.0000E+00	-1.2220E+00	-1.9407E+03	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	1.831E-04
12	2	32	0.0000E+00	0.0000E+00	-6.0220E-01	-7.4558E+02	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	2.514E-04
12	2	33	0.0000E+00	0.0000E+00	-1.3472E-01	-1.5813E+02	0.0000E+00	1.0000E+00	0	0	0	0	6.104E+01	1.0000E+00	1.985E-05
12	2	34	0.0000E+00	0.0000E+00	-3.2575E-02	-2.8992E+01	-2.8973E+01	6.5696E-01	2	2	2	2	6.104E+01	1.0000E+00	0.0000E+00
12	2	35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	6.104E+01	1.0000E+00	0.0000E+00
12	2	36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	6.104E+01	1.0000E+00	0.0000E+00
12	2	37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8	8	8	8	7.351E+01	1.0000E+00	0.0000E+00
12	2	38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	5	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	8	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	9	0.0000E+00	0.0000E+00	-1.9030E+00	-5.1306E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	10	0.0000E+00	0.0000E+00	-2.0337E+00	-4.9443E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	11	0.0000E+00	0.0000E+00	-2.1393E+00	-4.7482E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	12	0.0000E+00	0.0000E+00	-2.2781E+00	-4.5387E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	13	0.0000E+00	0.0000E+00	-2.3547E+00	-4.3157E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	14	0.0000E+00	0.0000E+00	-2.4019E+00	-4.0851E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	15	0.0000E+00	0.0000E+00	-2.4321E+00	-3.8499E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	16	0.0000E+00	0.0000E+00	-2.4515E+00	-3.6118E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	17	0.0000E+00	0.0000E+00	-2.4637E+00	-3.3717E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	18	0.0000E+00	0.0000E+00	-2.4707E+00	-3.1305E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	19	0.0000E+00	0.0000E+00	-2.4737E+00	-2.8886E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	20	0.0000E+00	0.0000E+00	-2.4730E+00	-2.6463E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	21	0.0000E+00	0.0000E+00	-2.4686E+00	-2.4042E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	22	0.0000E+00	0.0000E+00	-2.4599E+00	-2.1625E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	23	0.0000E+00	0.0000E+00	-2.4455E+00	-1.9216E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	24	0.0000E+00	0.0000E+00	-2.4231E+00	-1.6821E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	25	0.0000E+00	0.0000E+00	-2.3890E+00	-1.4449E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	26	0.0000E+00	0.0000E+00	-2.3377E+00	-1.2100E+04	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	27	0.0000E+00	0.0000E+00	-2.2603E+00	-9.8204E+03	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	28	0.0000E+00	0.0000E+00	-2.1431E+00	-7.6073E+03	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	29	0.0000E+00	0.0000E+00	-1.9636E+00	-5.5092E+03	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	30	0.0000E+00	0.0000E+00	-1.6823E+00	-3.5870E+03	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	31	0.0000E+00	0.0000E+00	-1.2240E+00	-1.9407E+03	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	32	0.0000E+00	0.0000E+00	-6.0220E-01	-7.4558E+02	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	33	0.0000E+00	0.0000E+00	-1.3472E-01	-1.5813E+02	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	34	0.0000E+00	0.0000E+00	-3.2575E-02	-2.8992E+01	0.0000E+00	1.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	35	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	36	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	37	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
13	2	38	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
1	3	1	4.3354E+01	-2.9552E+01	-2.4979E+02	-2.1991E+05	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00
1	3	2	4.3354E+01	-2.9552E+01	-8.5091E+01	-2.1991E+05	0.0000E+00	0.0000E+00	0	0	0	0	-1.0000E+00	1.0000E+00	0.0000E+00

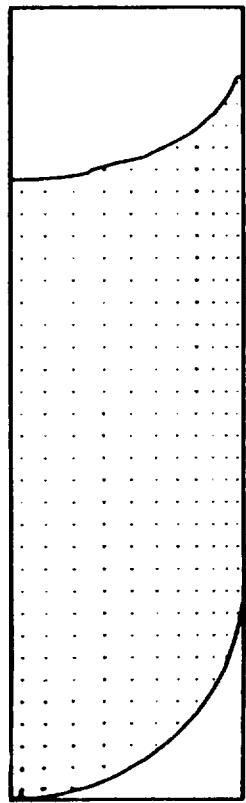


Fig. 2. Example problem: velocity vectors and free-surface contour at $t=0.0002$ s.

TANK DRAIN EXPERIMENT

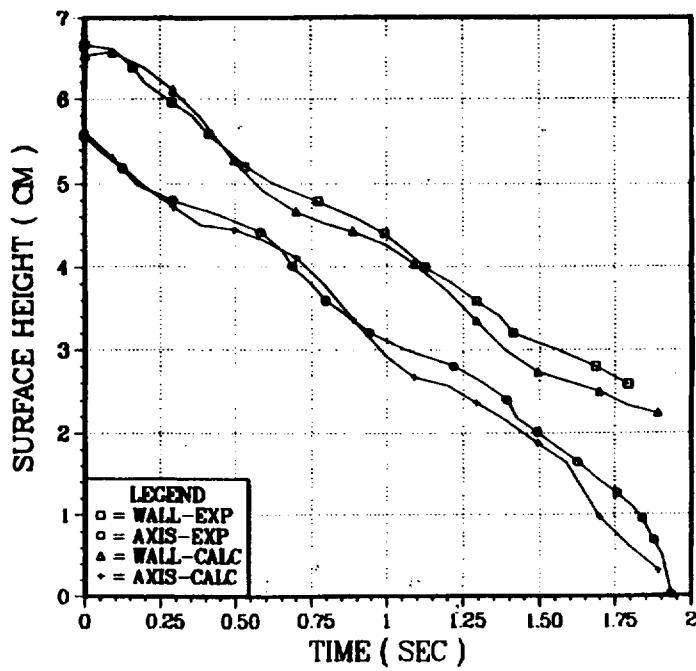


Fig. 3. Free surface height history.

reflect somewhat arbitrary definition of free surface location at the container wall. The maximal error resulting from this should be less than one cell height (0.2 cm).

IV. EXAMPLES OF SURFACE TENSION CALCULATIONS

We display the comparatively high accuracy of the present surface tension algorithms by presenting selected results for three geometrically simple examples. We also discuss the errors that can arise from the present algorithms and the (larger) errors that will arise from the more standard algorithms.

The class of example problem most difficult for our algorithms is that of the plane interface. Here the surface tension forces on opposite or adjacent faces of a computational cell should cancel exactly, and this is most difficult to arrange numerically using necessarily inexact data. The example we discuss is that of a vertical plane surface which is 0.55 cm from the axis of the ellipsoidal tank used for the experiments and simulations discussed in Sec. V. Fluid lies to the left of the interface, which, in turn, lies to the right of the axis. The fluid-vapor interface has surface-tension $\sigma = 28.2 \text{ dyne cm}^{-1}$, thus a surface pressure $|p_s| \sim 40 \text{ dyne cm}^{-2}$ corresponds to a strongly curved interface. In Table VI we display the surface pressures resulting from our present surface-tension algorithms as a function of the azimuthal zone number J . The first physical cell is $j = 2$, while $j = 15$ constitutes the last azimuthal zone intersected by the interface. It may be seen that the absolute surface pressures are mostly two to three orders of magnitude smaller than a strongly curved surface would possess, with the smallest value being $p_s = 0.015$. The multiple entries for $j = 14$ and $j = 15$ denote adjacent radial zones (values of i) that contain part of the surface.

The new surface tension algorithms generally work quite well. We have tested several hundred values of p_s , obtained from two off-axis spherical bubbles of varying radius and surface tension in different geometrical locations in the same ellipsoidal tank referred to above and in Sec. V. The tank is otherwise filled with a liquid. We obtain many values of the surface pressure, p_s , that are within 1% of the correct (analytic) values and almost all are within several percent of correct values. However, all 3D surface-tension calculations, including ours, are highly amplifying of certain types of numerical errors. As an example of the sorts of difficulty that must be handled, we note that we have seen an example in a spherical bubble where 1-2% errors on the height functions defining the surface result in 20-

TABLE VI
SURFACE PRESSURES FOR PLANE
INTERFACE IN ELLIPSOIDAL TANK

Azimuthal Zone Number	Surface Pressure P_s (dynes · cm ⁻²)
J	
2	-0.808
3	-0.228
4	-0.905
5	-0.406
6	-0.649
7	0.055
8	-0.205
9	0.248
10	-0.120
11	-0.339
12	-0.266
13	0.055
14	0.399
	-0.161
15	0.015
	-0.327
	0.068
	0.219

30% errors in some values for components of surface gradient vectors and as much as 50-60% error in a local value for p_s . By contrast, in the same example using standard older formulations for evaluating p_s (e.g., those used in the early phases of our work) it would be common to have 100-200% error in components of surface gradient vectors. So our algorithms represent an improvement in the state of the art, but they can still encounter problems. In general, it is more difficult to calculate p_s in 3D than in 2D. In 3D both surface curvatures must be evaluated by finite-difference approximations to second derivatives, whereas in 2D one of the curvatures may be obtained from a more accurate algebraic formula. Moreover, in

cylindrical geometry, the algebraic curvature is typically the largest one in the crucial region near the coordinate axis ($r = 0.0$). But in 3D the full numerical complexities of evaluating p_s must be faced routinely.

V. SAMPLE APPLICATIONS

We present here numerical simulations of three experiments¹³ done at the NASA LeRC Zero Gravity Facility to model the liquid oxygen dynamics in a Centaur G-Prime Tank during separation from the shuttle. Graphs of the data have been presented, along with numerical simulations using HYDR3-D,¹⁴ as Figs. 1, 2, and 3 of Reference 15. We will be referring to these figures as we discuss our simulations.

The experiments are done in an ellipsoidal tank which is partially filled with either Ethanol or FC-43 and then dropped from a drop tower. After two seconds of free fall the tank is accelerated along a principal axis for roughly 0.1 sec and the position of the vapor bubble is recorded on film for roughly 0.6 to 0.8 sec. The two-dimensional bubble profiles recorded on the film and displayed in Figs. 1, 2, and 3 of Reference 15 can give only a very approximate impression of the actual three-dimensional bubble contours.

We present data for the numerical simulations in several computational planes. Our basic plot is a composite plot of the first ($j = 2$) and the last ($j = JM1$) azimuthal planes. It is very nearly the result that would be seen at the midplane of the ellipsoid. Other plots are given in azimuthal planes ($j = \text{constant}$) and horizontal planes ($k = \text{constant}$). The calculational mesh of physical cells contains 13 radial cells, 18 azimuthal cells, and 20 vertical cells. To simplify the running of the problems, the applied acceleration is ramped linearly from zero to its experimental value in a time interval Δt_r which is typically 1% of the time interval Δt_a over which the acceleration is applied.

Experiment R-12-7 was done with Ethanol, and had an applied acceleration of 0.22 g lasting $\Delta t_a = 0.109$ sec. Numerical results of our simulation are displayed in Figs. 4-17, which give velocity vector and surface contour plots in our basic composite plot at various times during the run. A fairly rich display of computational planes ($j = \text{constant}$, and $k = \text{constant}$) is also given at $t = 0.10$ sec. It is seen from Figs. 6-12 and from Fig. 3 of Reference 15 that our simulation correctly reproduces the location of the bubble at $t = 0.10$ sec, including the jetting of the fluid into the central bubble region. The fluid jet is somewhat wider than indicated by experiment.

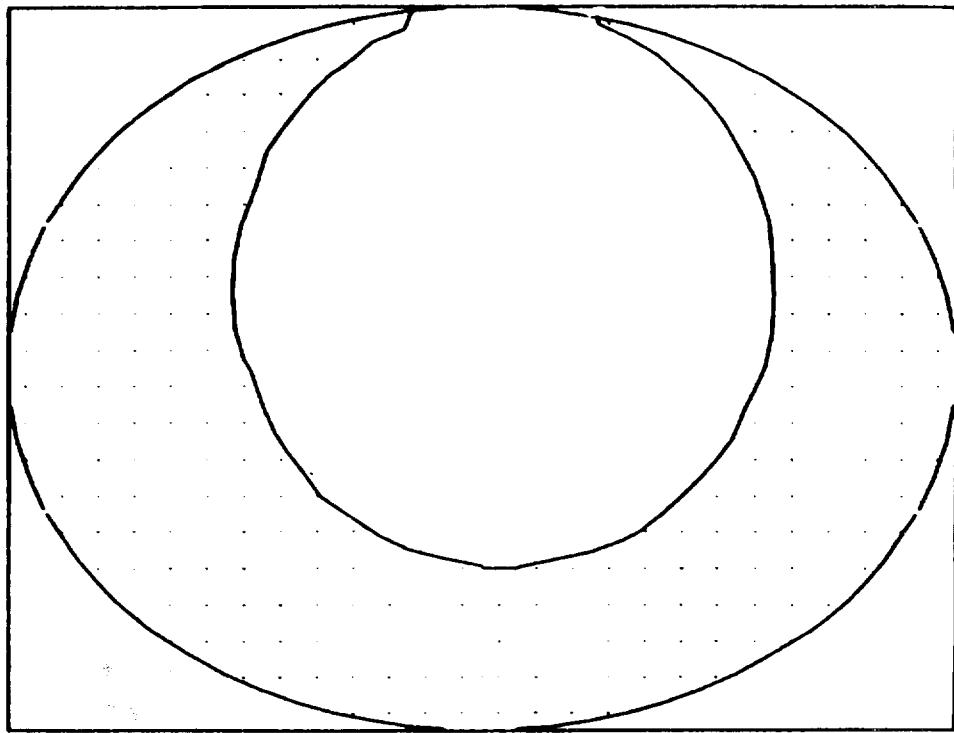


Fig. 4. Velocity vector plot for problem R-12-7: $t=0.0$ s.

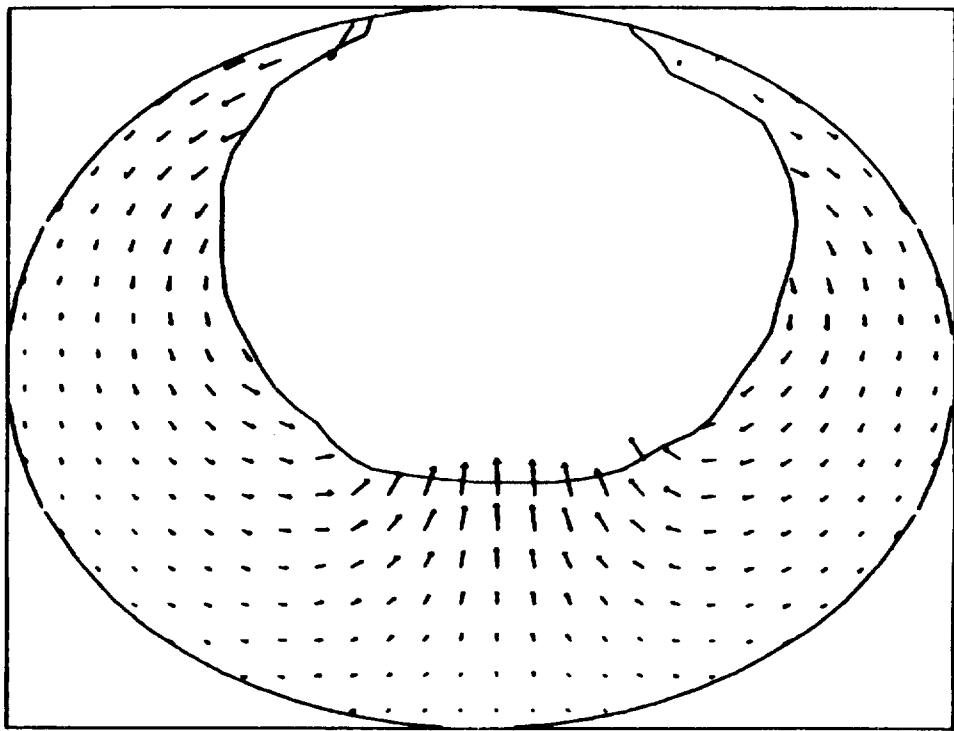


Fig. 5. Velocity vector plot for problem R-12-7: $t=0.05$ s.

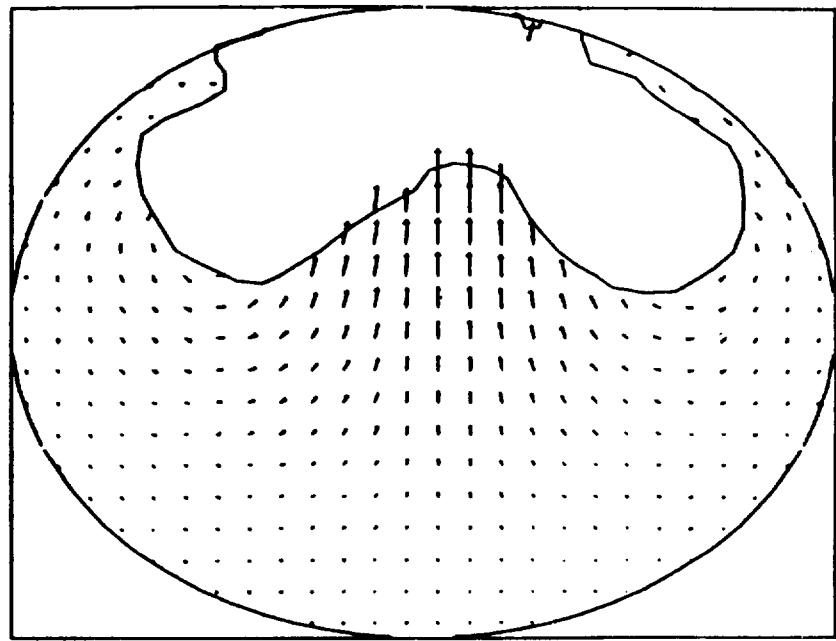


Fig. 6. Velocity vector plot for problem R-12-7: $t=0.10$ s.

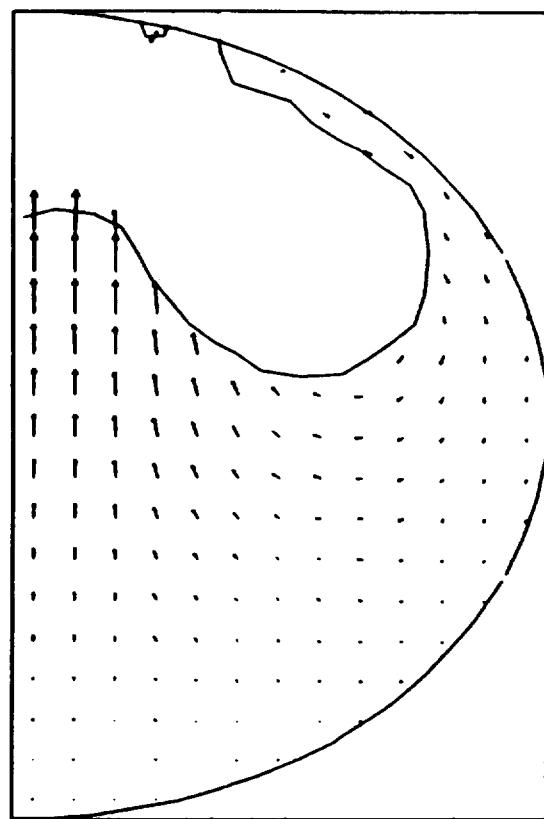


Fig. 7. Velocity vector plot for problem R-12-7: $t=0.10$ s. $J=2$ surface.

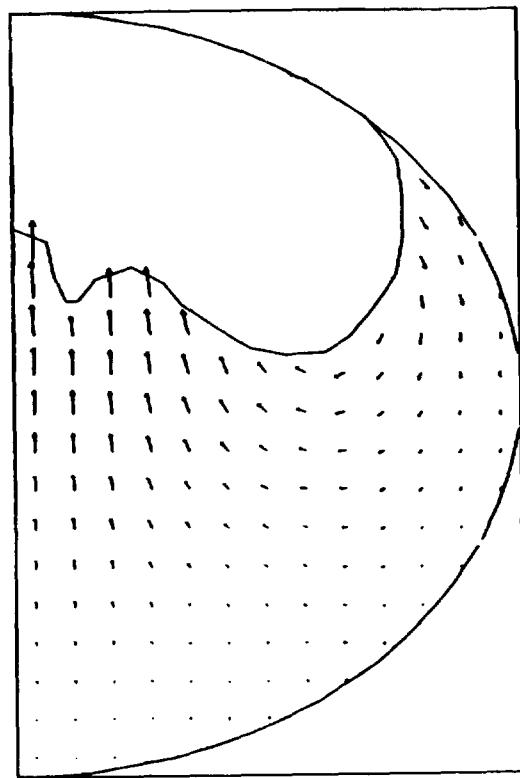


Fig. 8. Velocity vector plot for problem R-12-7: $t=0.10$ s. $J=5$ surface.

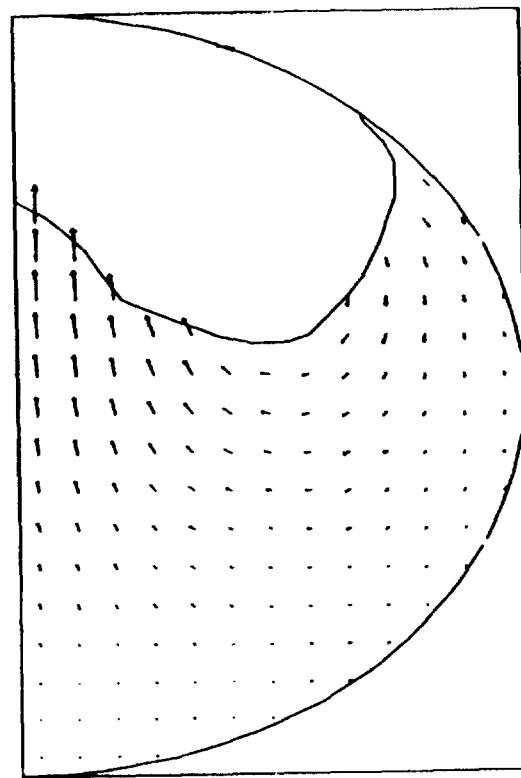


Fig. 9. Velocity vector plot for problem R-12-7: $t=0.10$ s. $J=12$ surface.

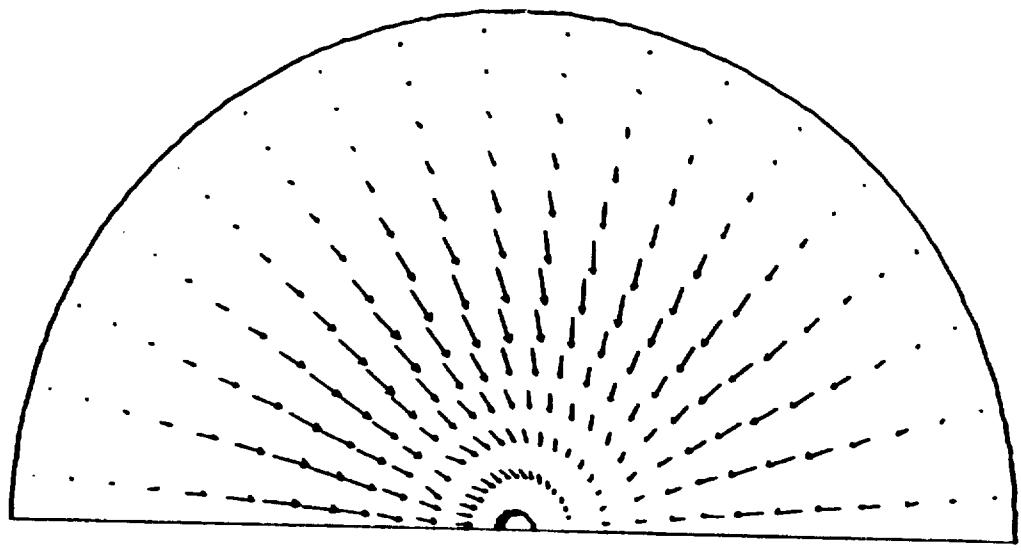


Fig. 10. Velocity vector plot for problem R-12-7: $t=0.10$ s. $K=12$ surface.

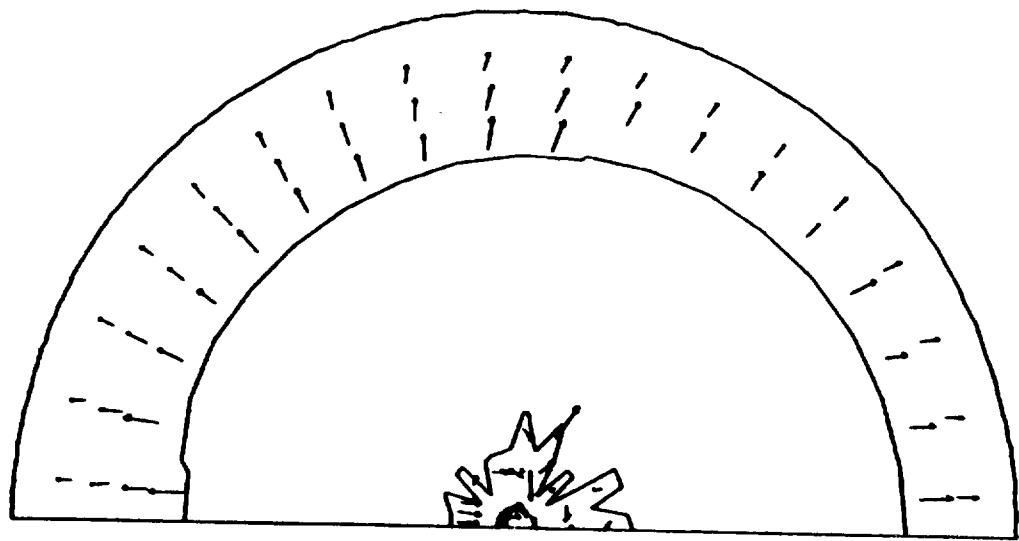


Fig. 11. Velocity vector plot for problem R-12-7: $t=0.10$ s. $K=15$ surface.

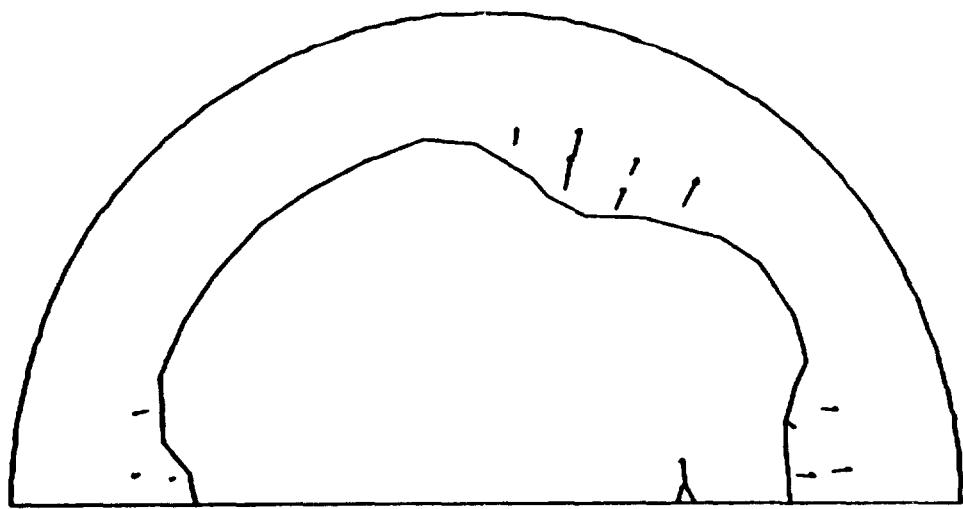


Fig. 12. Velocity vector plot for problem R-12-7: $t=0.10$ s. $K=18$ surface.

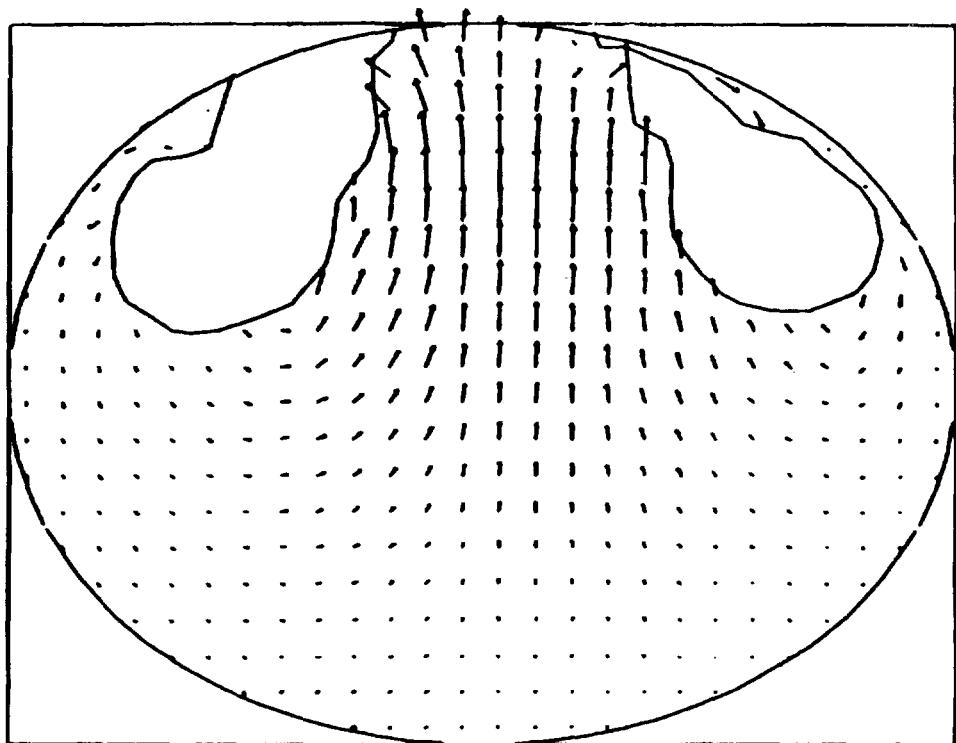


Fig. 13. Velocity vector plot for problem R-12-7: $t=0.15$ s.

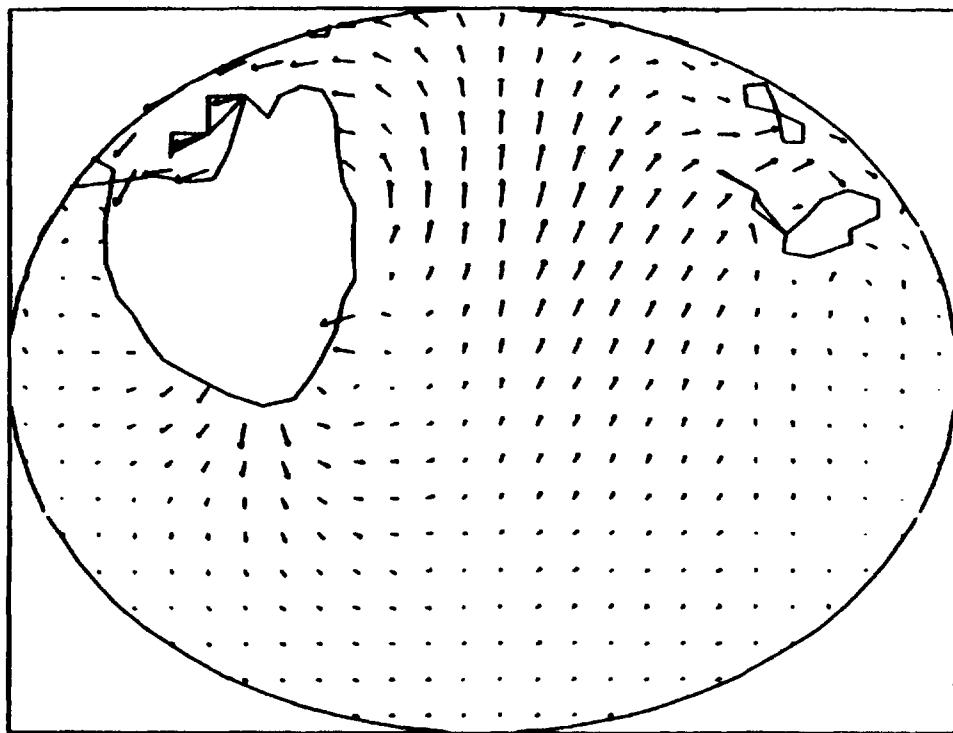


Fig. 14. Velocity vector plot for problem R-12-7: $t=0.25$ s.

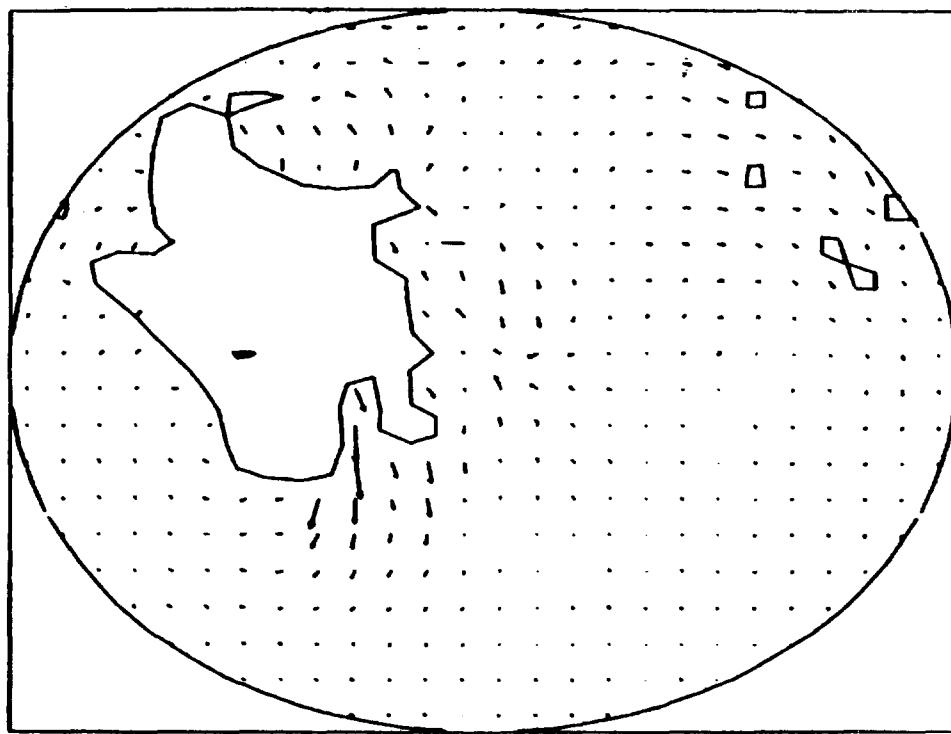


Fig. 15. Velocity vector plot for problem R-12-7: $t=0.45$ s.

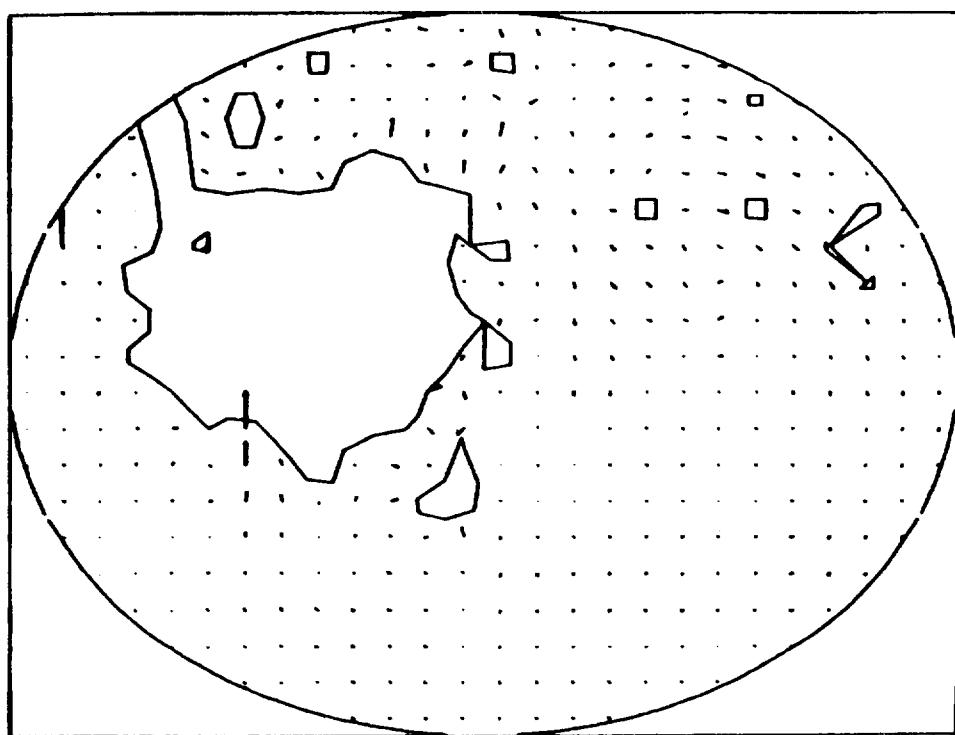


Fig. 16. Velocity vector plot for problem R-12-7: $t=0.65$ s.

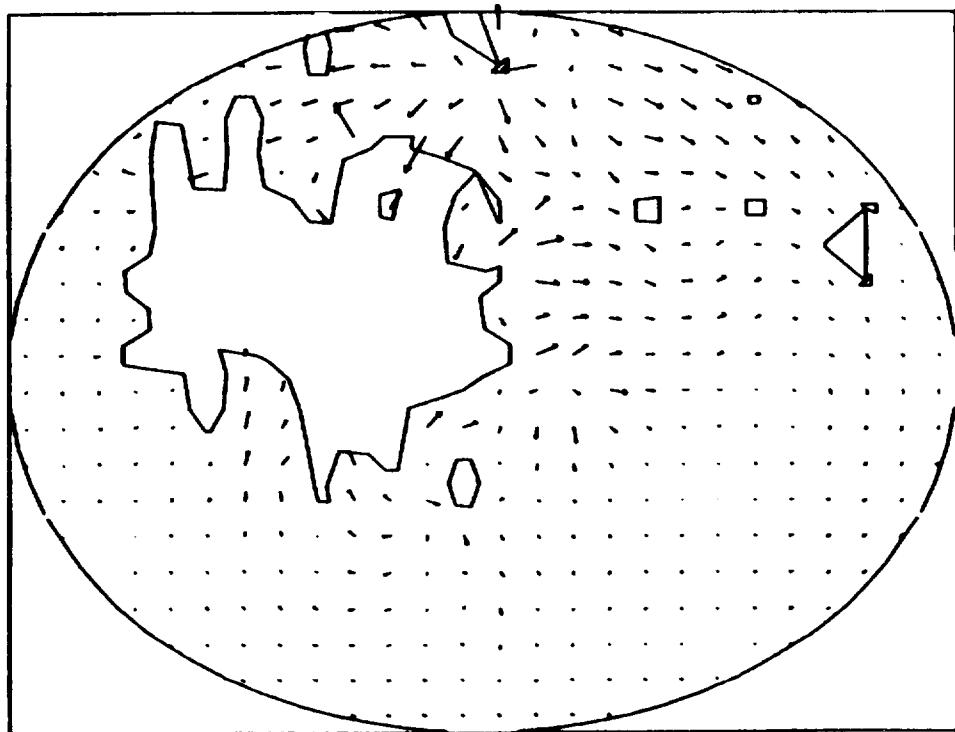


Fig. 17. Velocity vector plot for problem R-12-7: $t=0.70$ s.

Comparing our composite plot at $t = 0.65$ sec, Fig. 16, with the experimental plot at time $t = 0.67$ sec, Fig. 3 of Reference 15, we see a general breaking away of the bubble from the negative z tank wall and a general shift toward negative x values for the bubble location. There is thus a general agreement between our simulation and experiment; however, the numerical results indicate a less complete separation from the negative z tank wall and a more fragmented bubble structure than does the experiment. There appears to be a reliable long term predictive capability for this type of problem from numerical simulations based on NASA-VOF3D.

Experiment R-12-8 was done with FC-43, and had an applied acceleration of 0.14 g lasting $\Delta t_a = 0.132$ sec. Numerical results of our simulation are displayed in Figs. 18-31 and may be compared at $t = 0.13$ sec with experiment. Our simulation indicates very general qualitative agreement with experiment as to the location of the bubble. The bubble is to be found at negative z values and positive x values and has fluid jetting into its central region. However, the simulation does not agree in detail with experiment; the bubble displays too extended a structure, the jets are too wide and the bubble intersection with the tank wall is not perfectly represented. The NASA-VOF3D simulation has a bubble intersection with the tank wall more nearly in agreement with experiment, but its fluid jet has a double hump structure that is not apparent in the data. Of course, one must not place too much emphasis on detailed agreement with experiment in view of the difficulty of interpreting the two-dimensional trace of three-dimensional data (contained in the experimental plots) in terms of a truly two-dimensional structure displayed in the simulation plots. By comparing our Fig. 31 at $t = 0.85$ sec with the data at $t = 0.84$ sec one again sees general agreement; the bubble has moved toward the middle of the tank in the z coordinate and has shifted to slightly negative x values. However, the numerical results indicate some bubble contact with the negative z tank wall and a fairly complex bubble interface, in contrast to the data. There again appears to be an improvement in the long term predictive ability for general bubble location in NASA-VOF3D.

Experiment R-12-9 was done with Ethanol, and had an applied acceleration of 0.25 g lasting $\Delta t_a = 0.101$ sec. Our simulation results are displayed in Figs. 32-43 and may be compared with experiment in Fig. 1 of Reference 15. There is again qualitative agreement between simulation and data at $t = 0.10$ sec, but the left lobe of our bubble is not firmly in contact with the negative z tank wall. In contrast to the data, the bubble has not moved quite far enough to the positive x direction, and

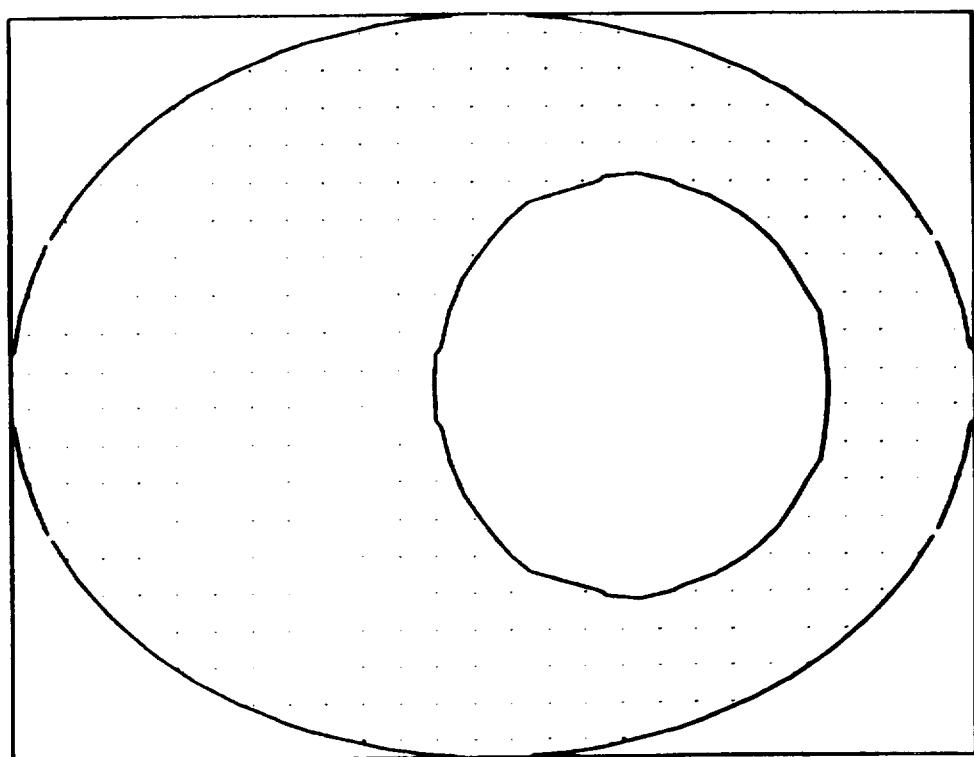


Fig. 18. Velocity vector plot for problem R-12-8: $t=0.00$ s.

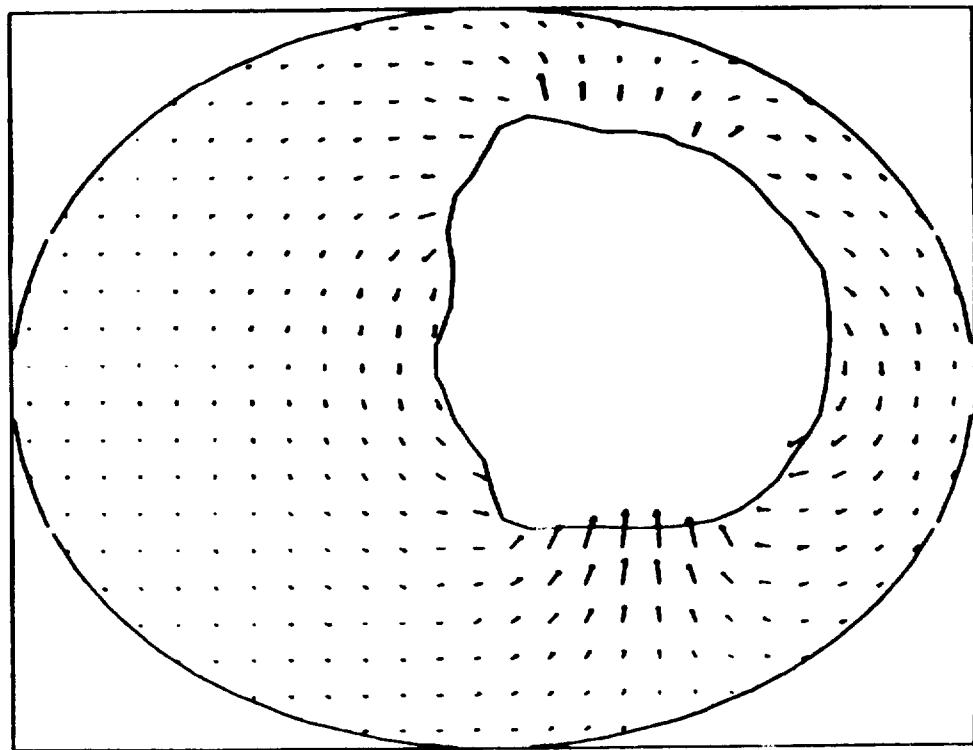


Fig. 19. Velocity vector plot for problem R-12-8: $t=0.05$ s.

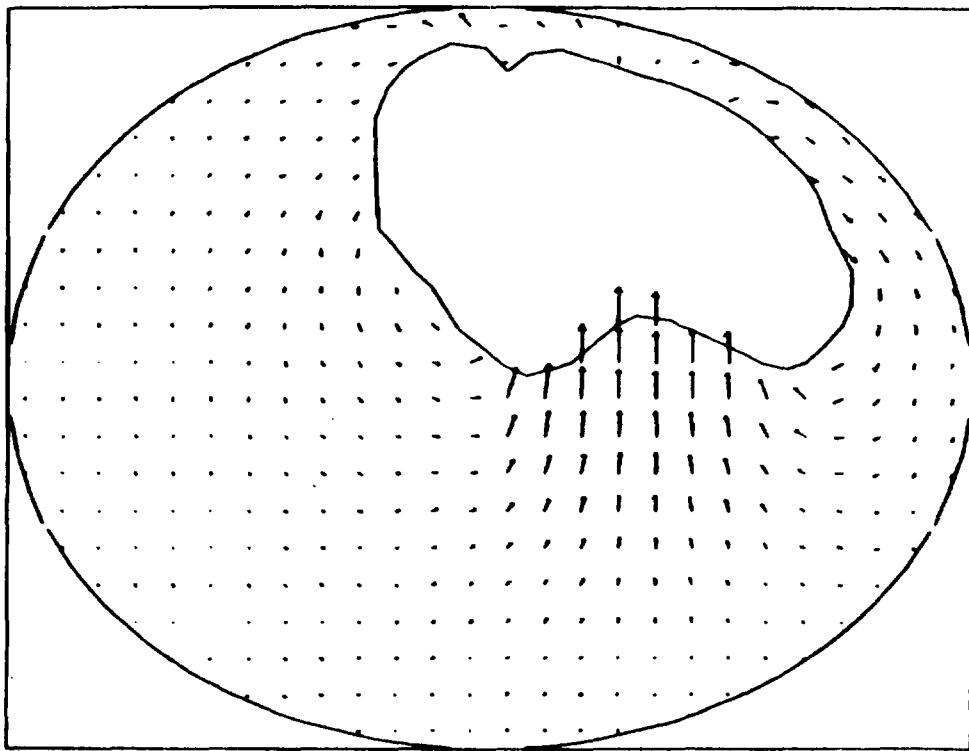


Fig. 20. Velocity vector plot for problem R-12-8: $t=0.10$ s.

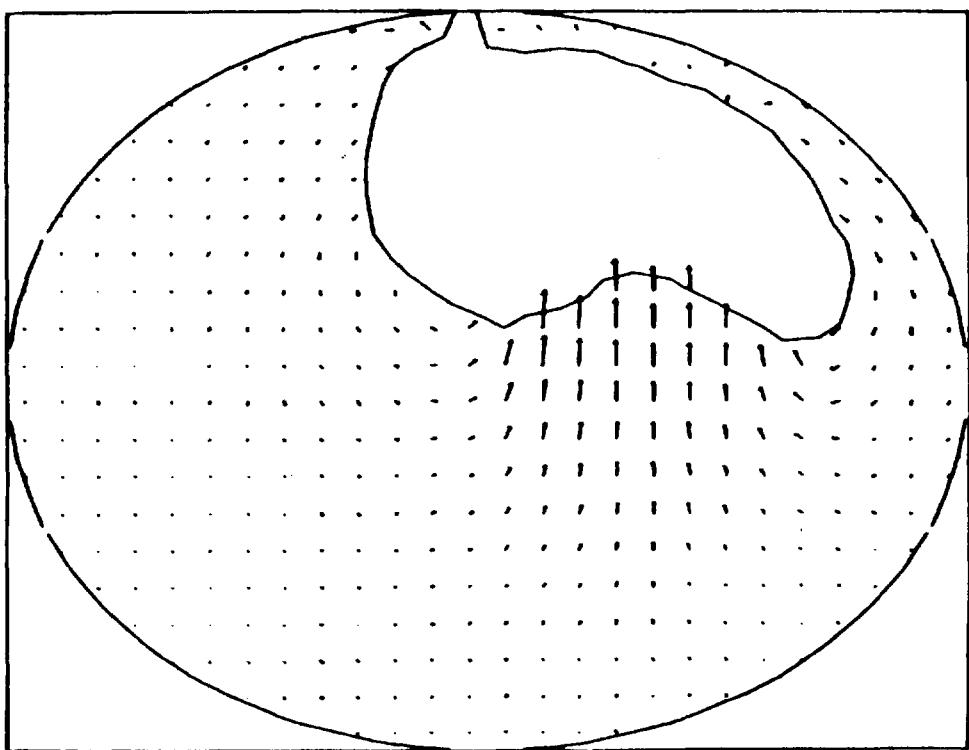


Fig. 21. Velocity vector plot for problem R-12-8: $t=0.11$ s.

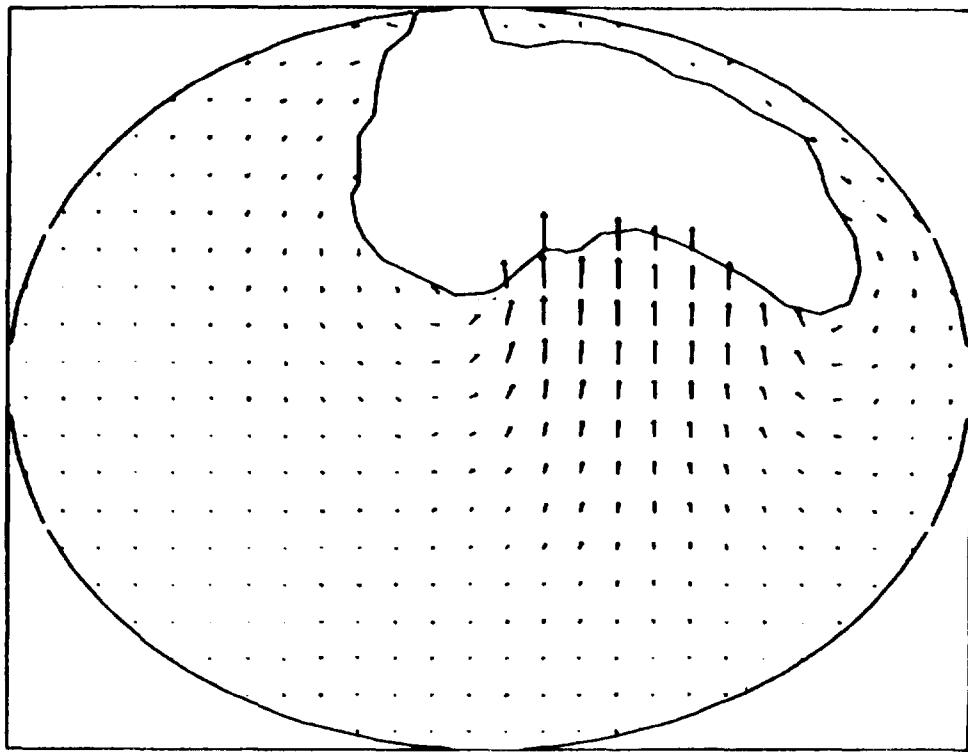


Fig. 22. Velocity vector plot for problem R-12-8: $t=0.12$ s.

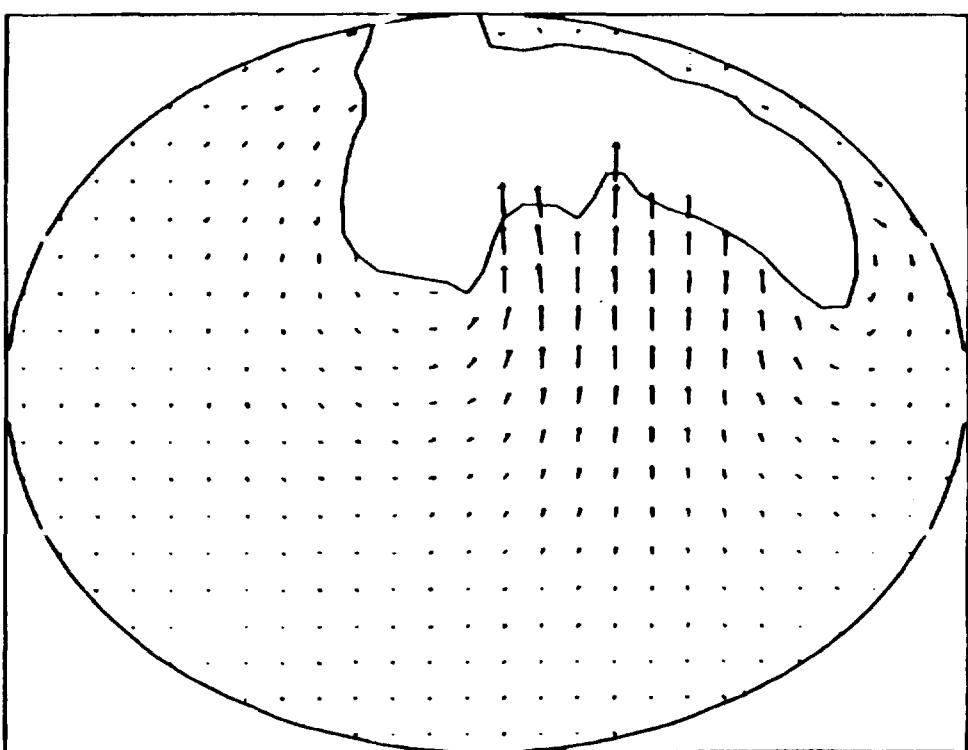


Fig. 23. Velocity vector plot for problem R-12-8: $t=0.13$ s.

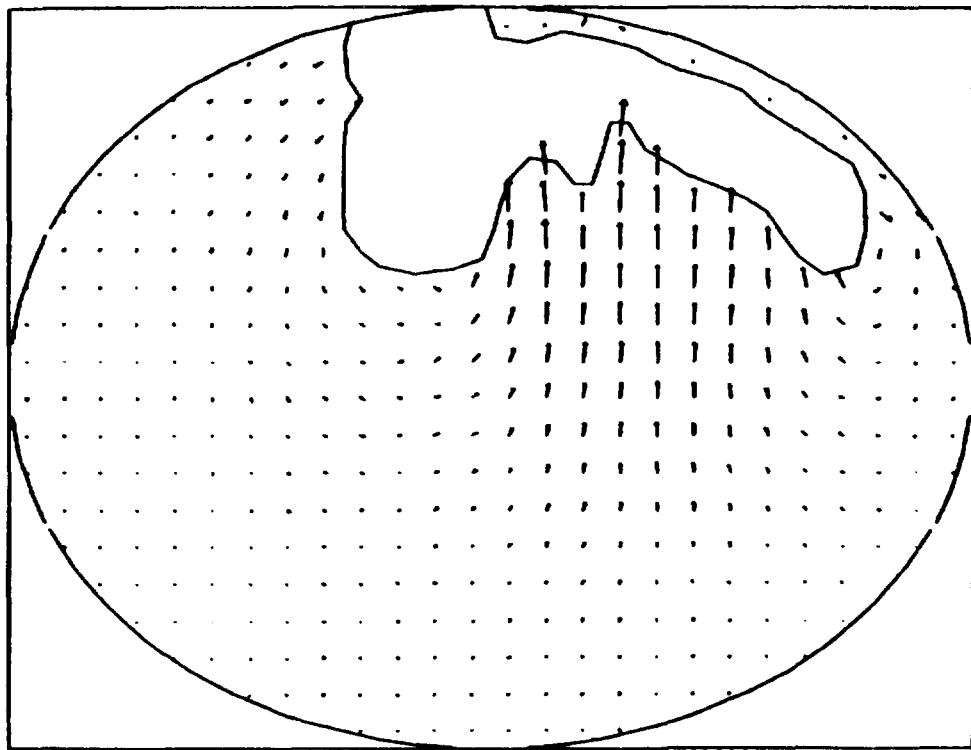


Fig. 24. Velocity vector plot for problem R-12-8: $t=0.14$ s.

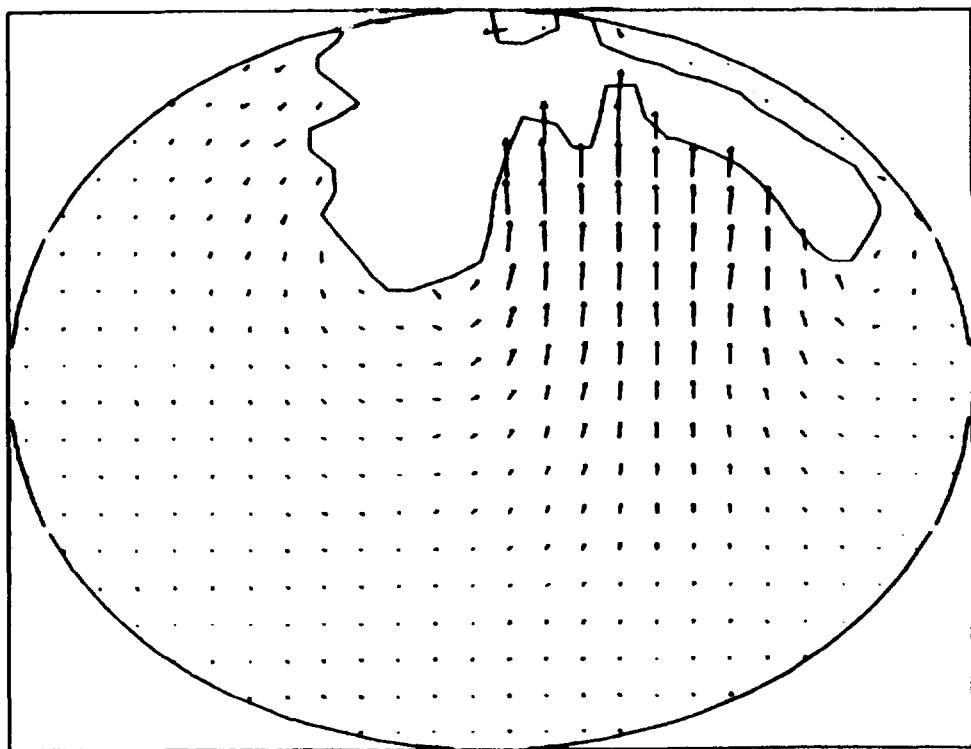


Fig. 25. Velocity vector plot for problem R-12-8: $t=0.15$ s.

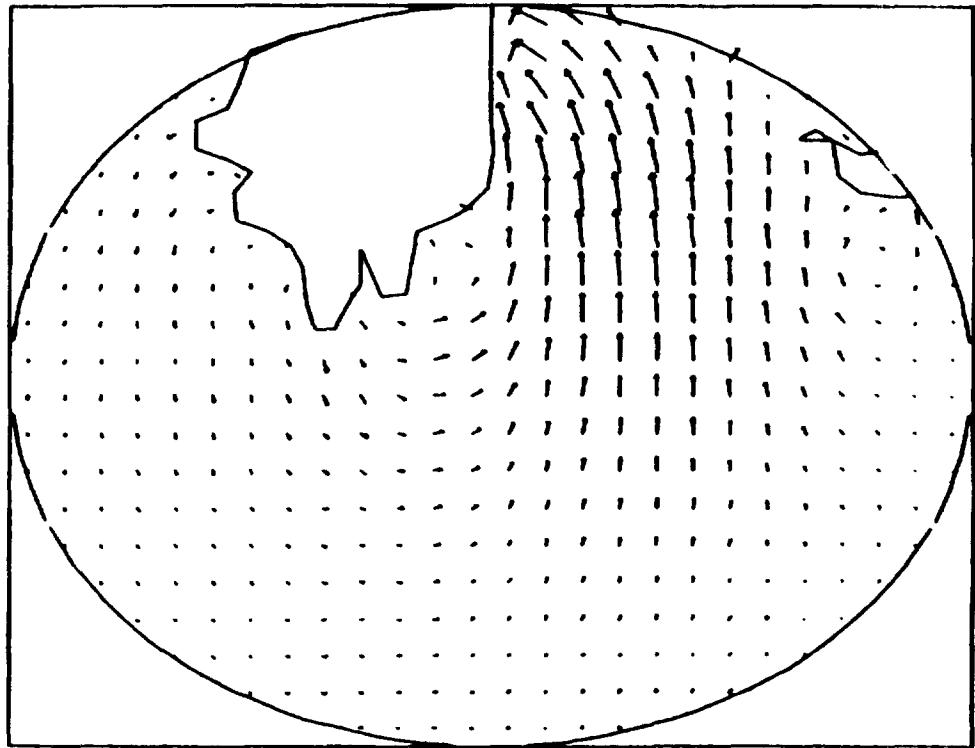


Fig. 26. Velocity vector plot for problem R-12-8: $t=0.25$ s.

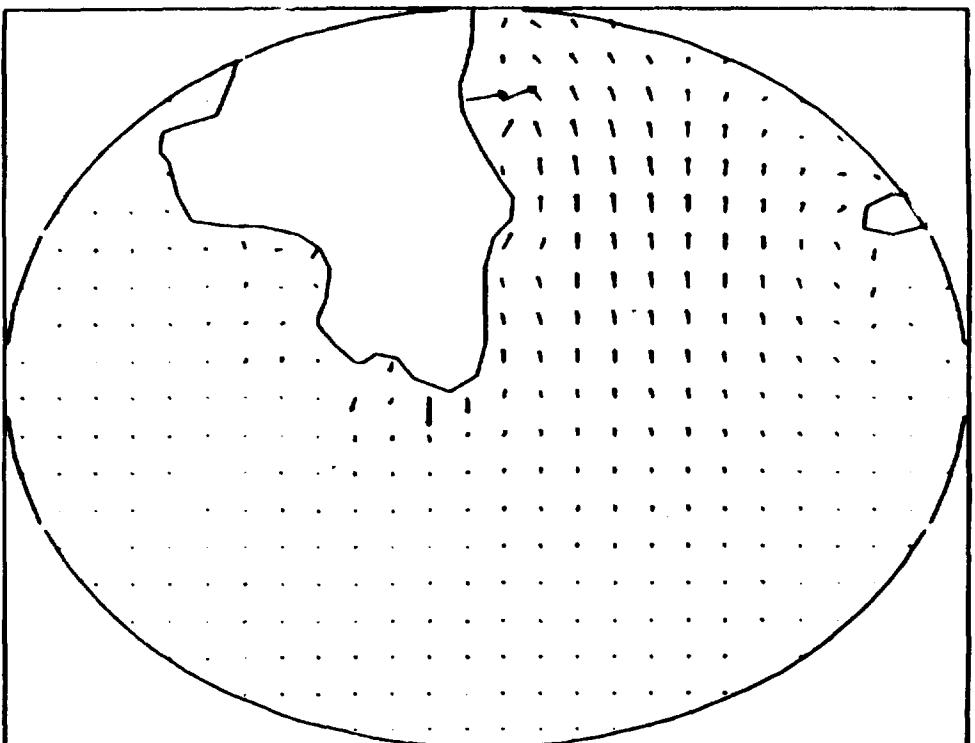


Fig. 27. Velocity vector plot for problem R-12-8: $t=0.35$ s.

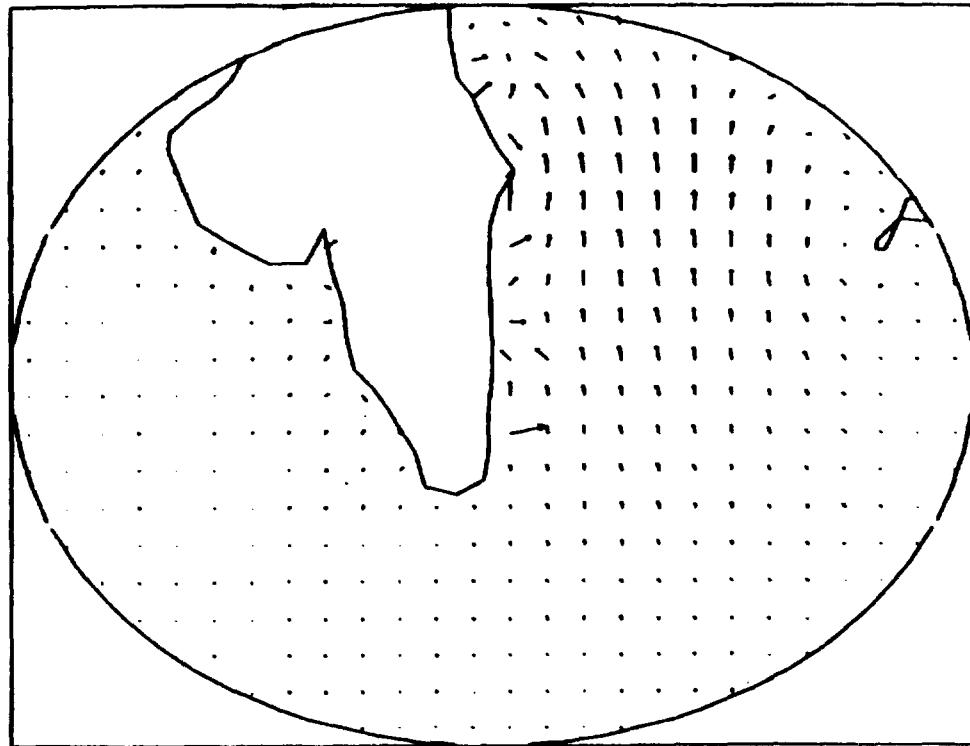


Fig. 28. Velocity vector plot for problem R-12-8: $t=0.45$ s.

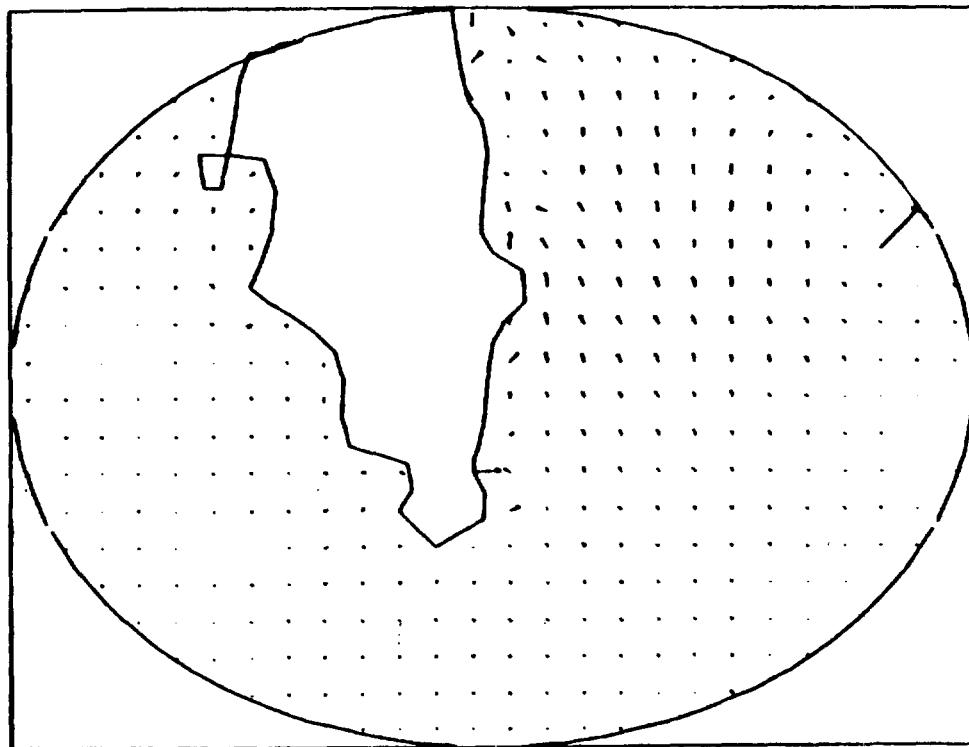


Fig. 29. Velocity vector plot for problem R-12-8: $t=0.65$ s.

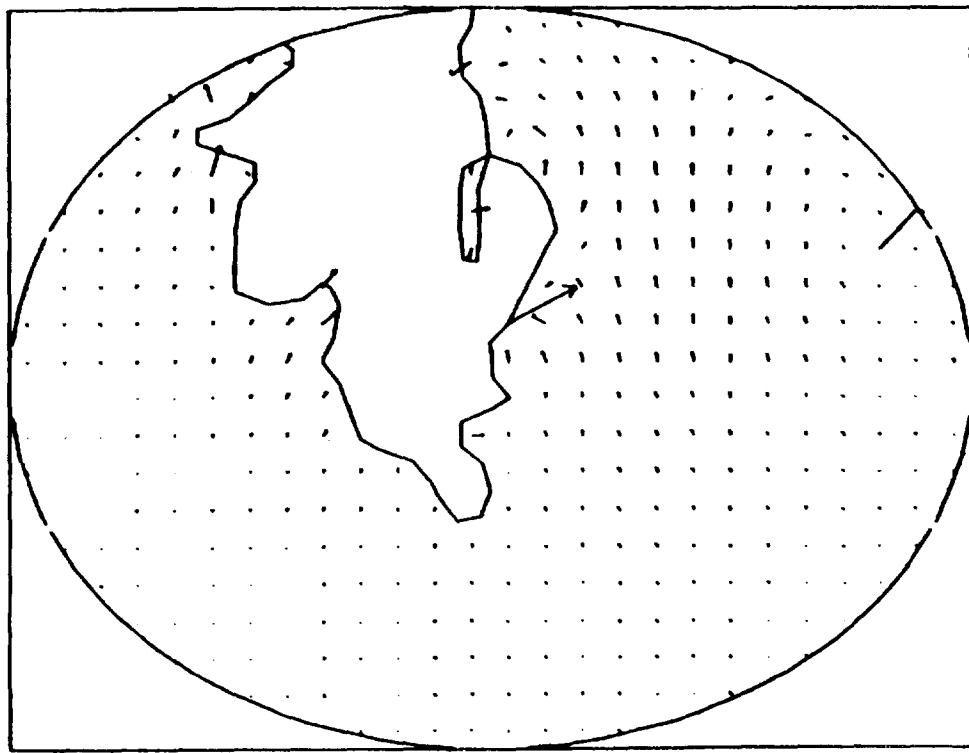


Fig. 30. Velocity vector plot for problem R-12-8: $t=0.80$ s.

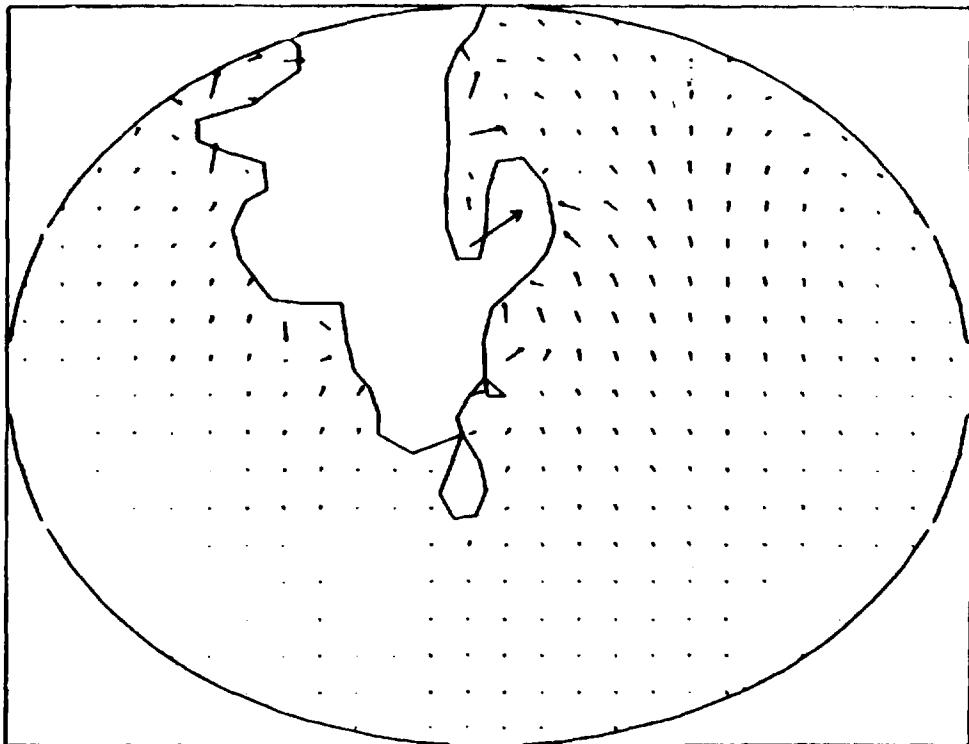


Fig. 31. Velocity vector plot for problem R-12-8: $t=0.85$ s.

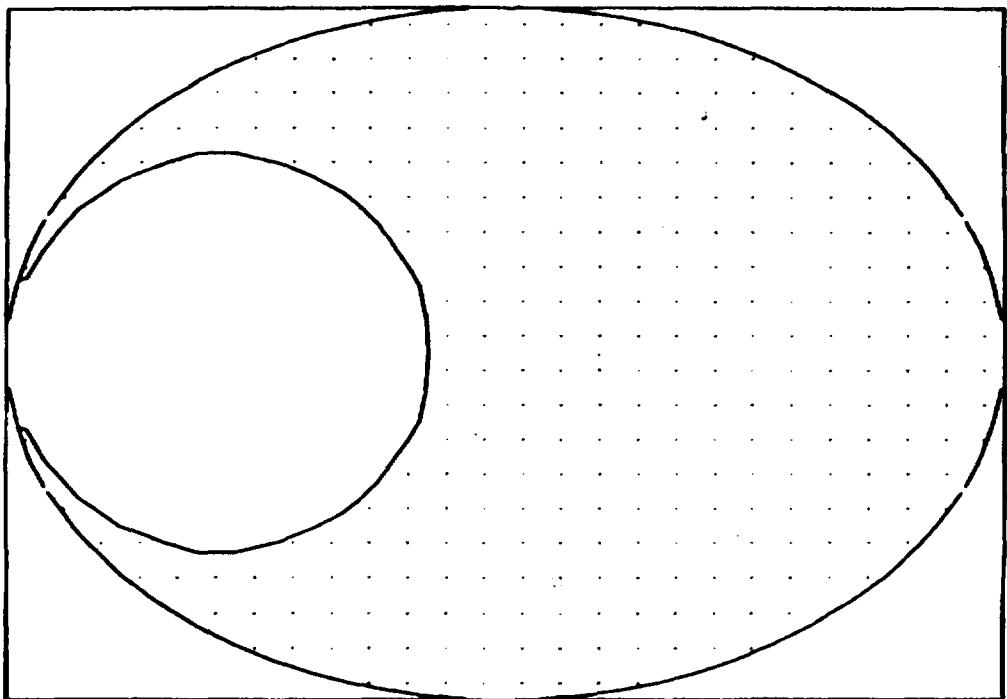


Fig. 32. Velocity vector plot for problem R-12-9: $t=0.0$ s.

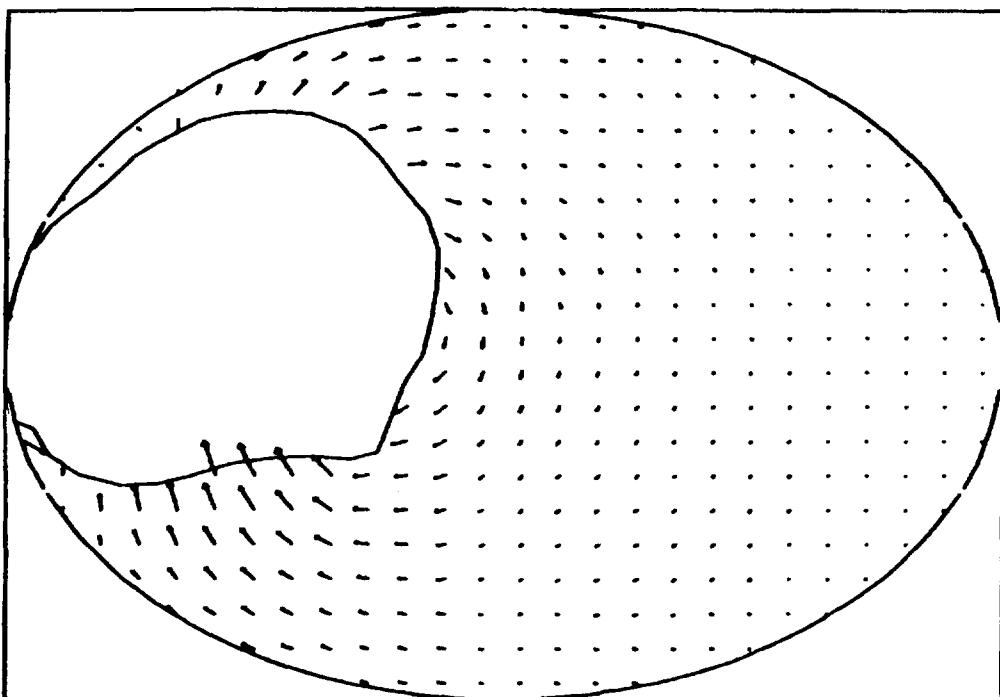


Fig. 33. Velocity vector plot for problem R-12-9: $t=0.05$ s.

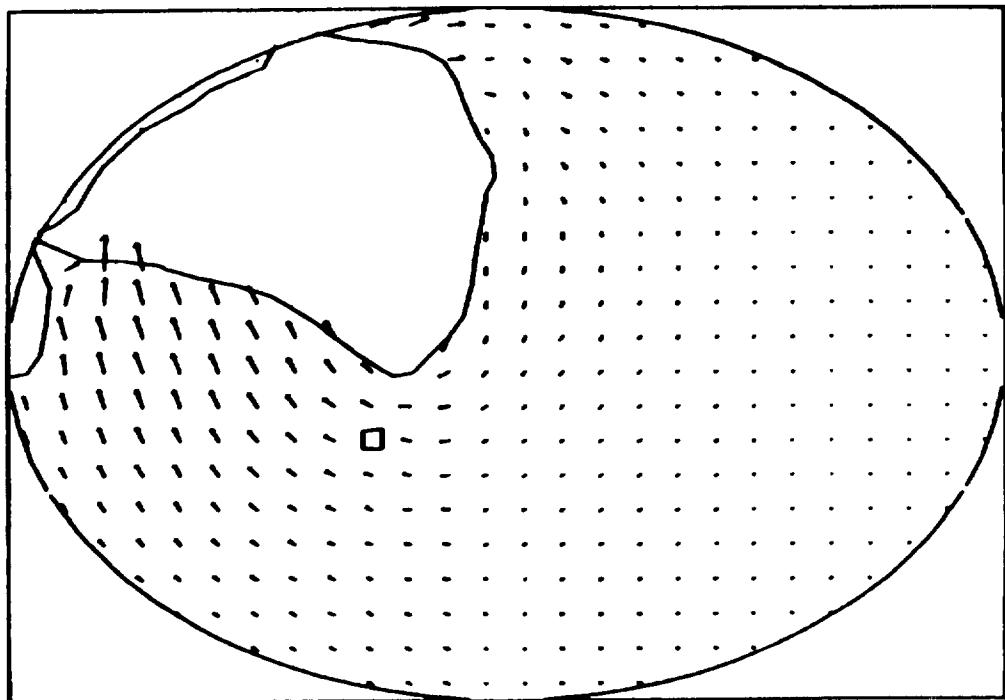


Fig. 34. Velocity vector plot for problem R-12-9: $t=0.10$ s.

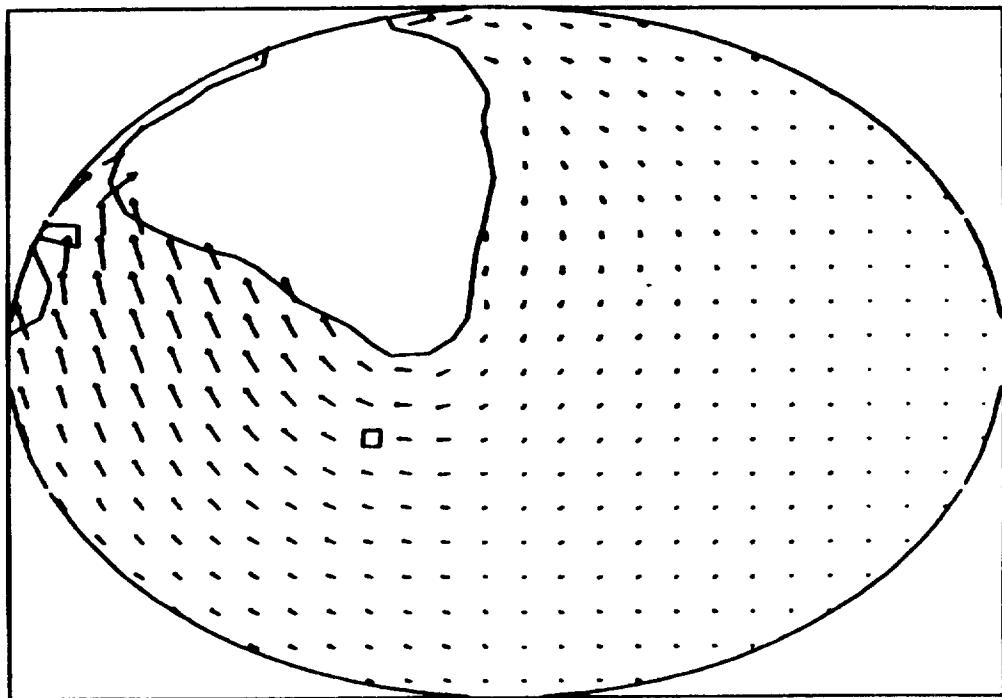


Fig. 35. Velocity vector plot for problem R-12-9: $t=0.11$ s.

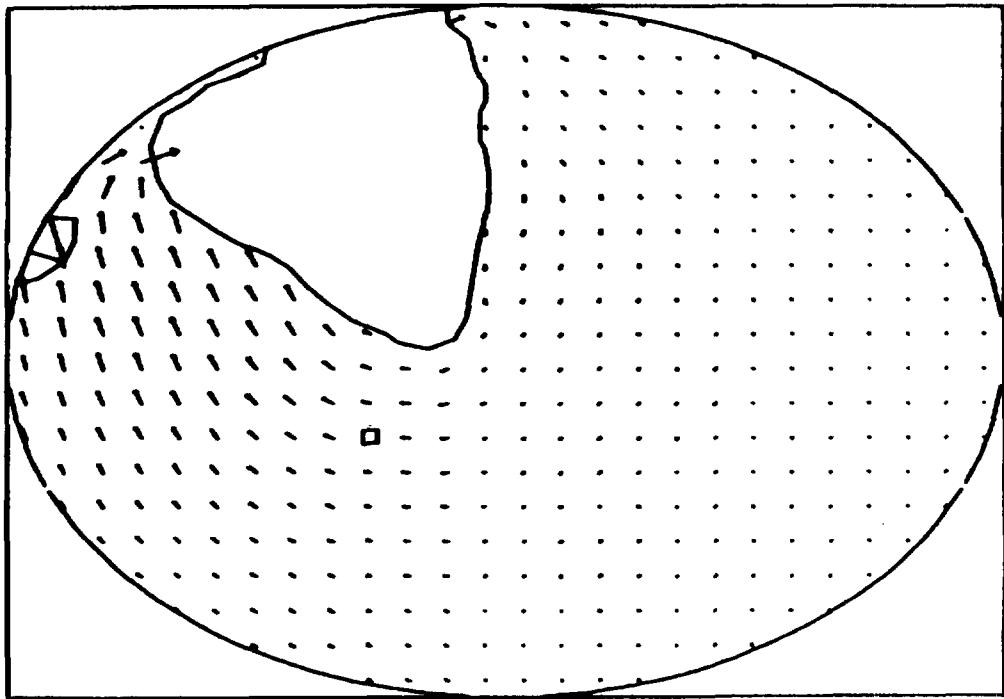


Fig. 36. Velocity vector plot for problem R-12-9: $t=0.12$ s.

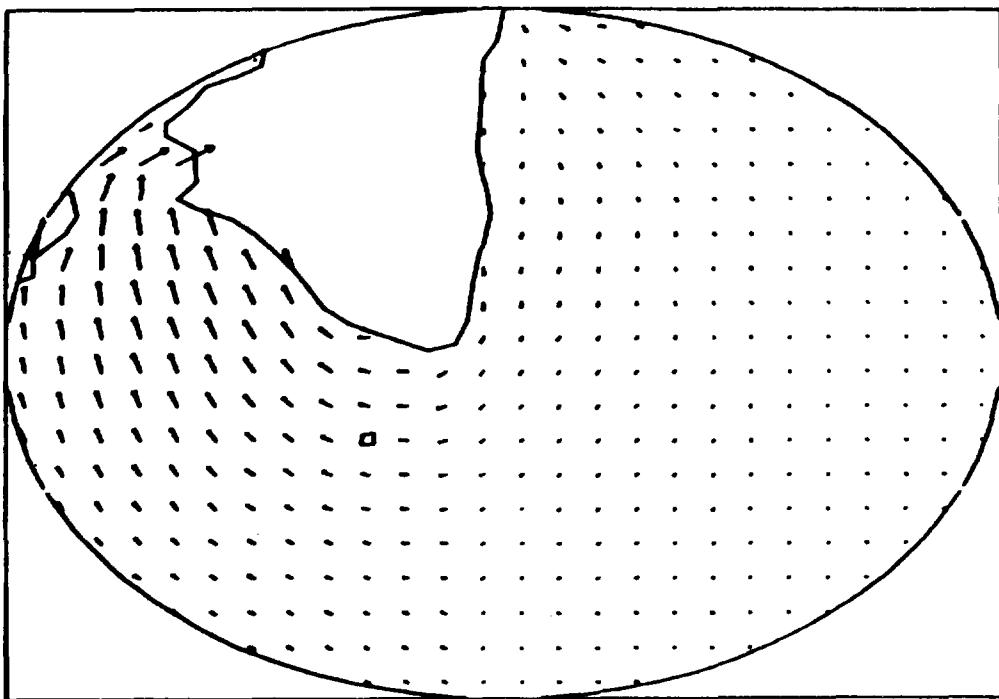


Fig. 37. Velocity vector plot for problem R-12-9: $t=0.13$ s.

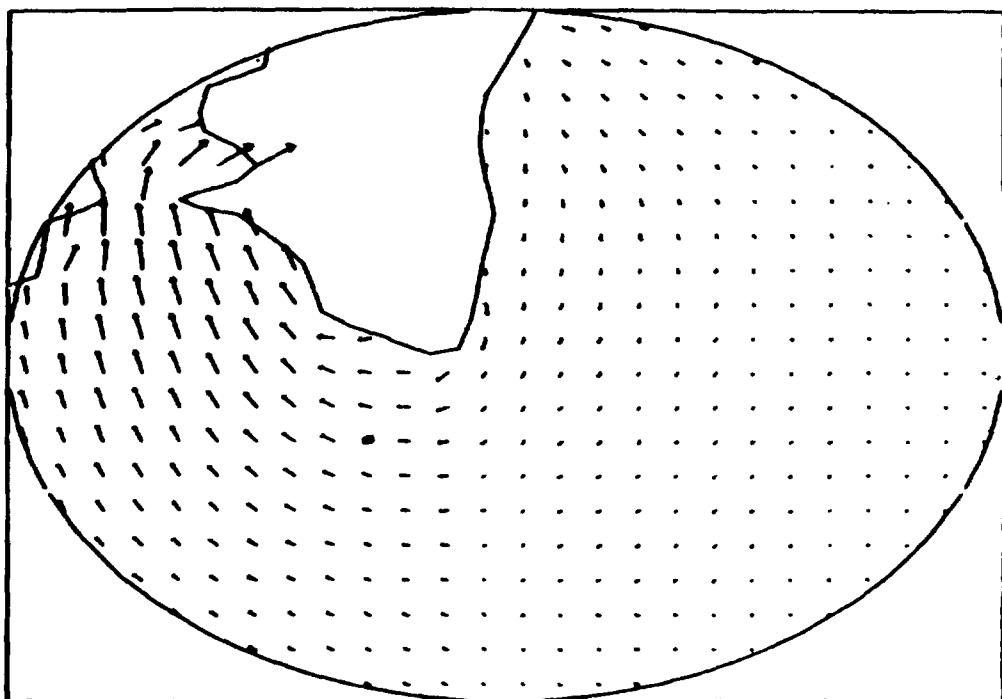


Fig. 38. Velocity vector plot for problem R-12-9: $t=0.14$ s.

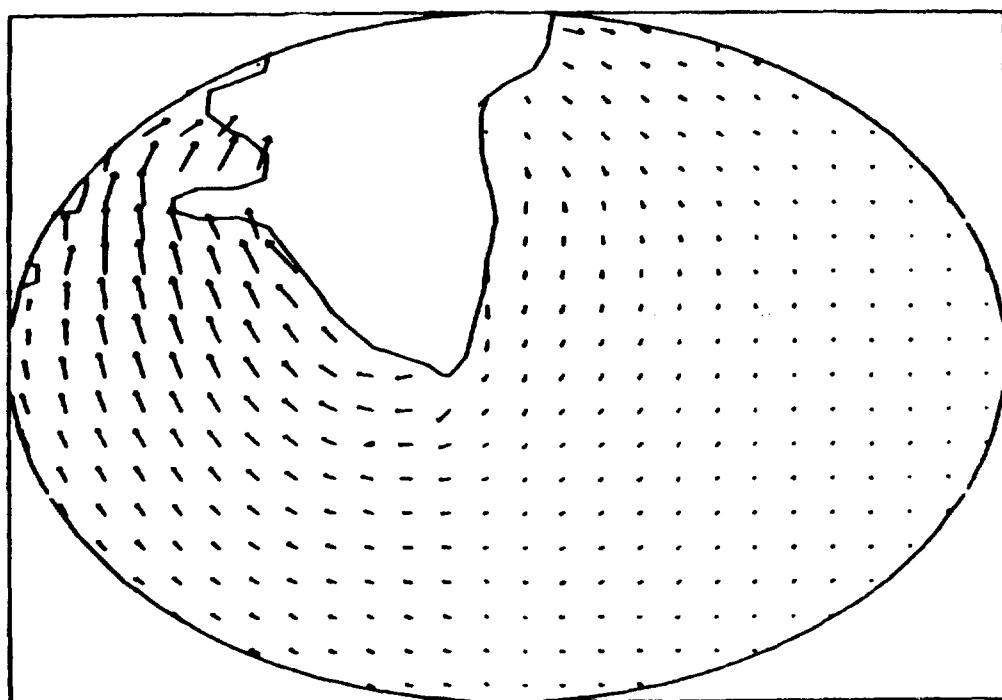


Fig. 39. Velocity vector plot for problem R-12-9: $t=0.15$ s.

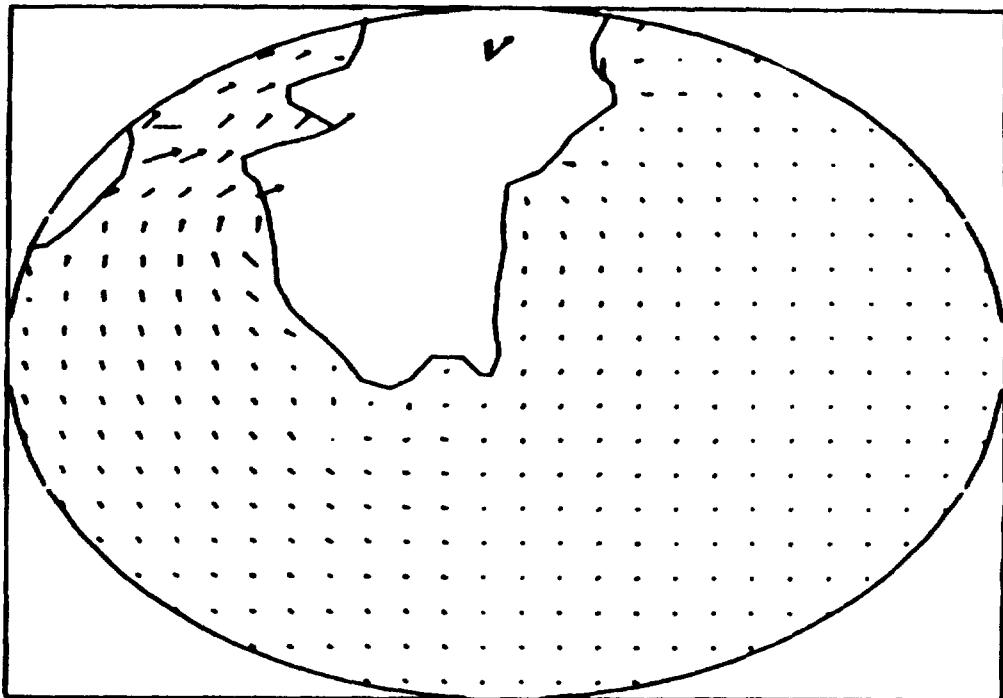


Fig. 40. Velocity vector plot for problem R-12-9: $t=0.25$ s.

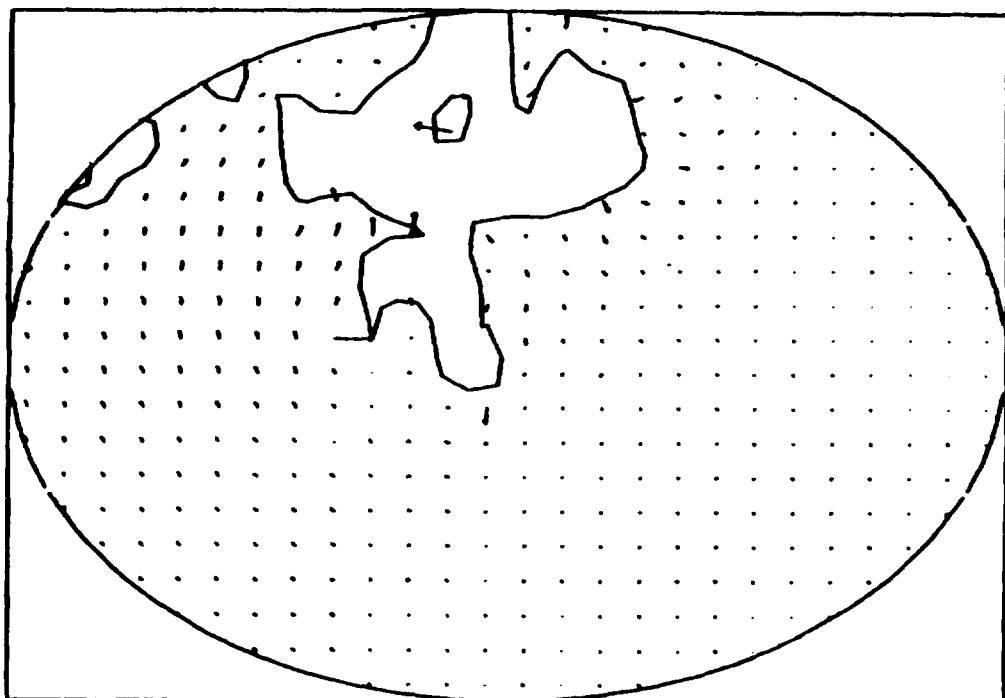


Fig. 41. Velocity vector plot for problem R-12-9: $t=0.35$ s.

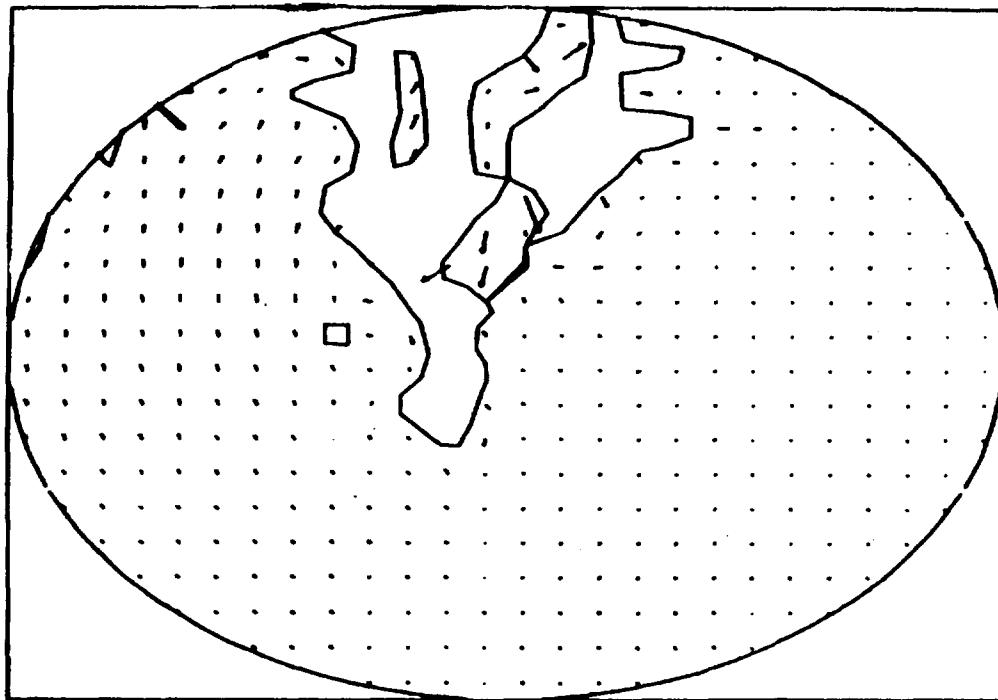


Fig. 42. Velocity vector plot for problem R-12-9: $t=0.55$ s.

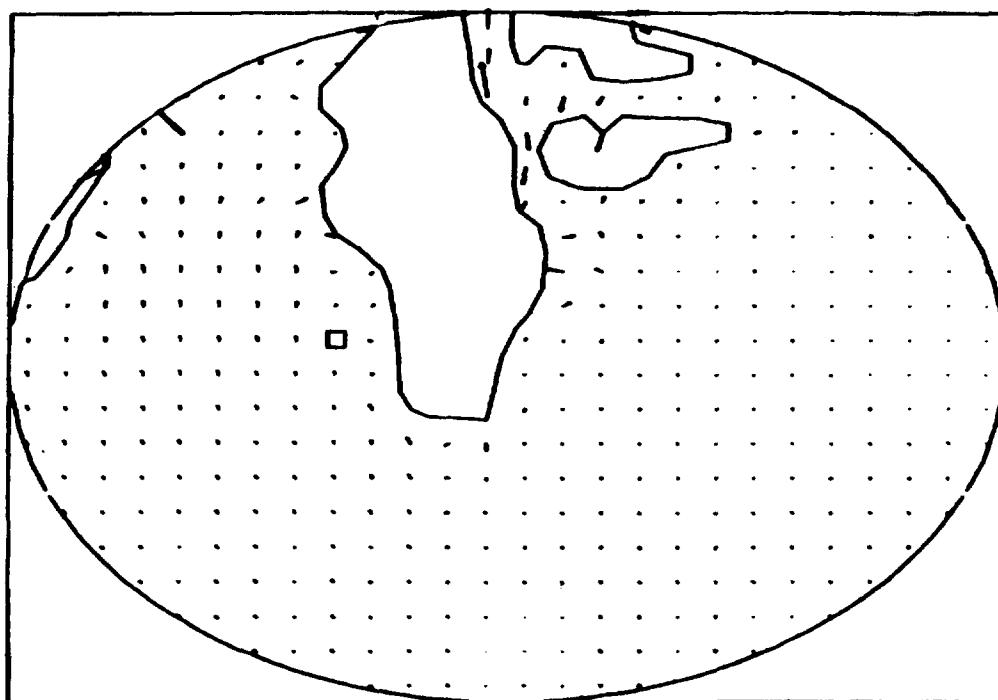


Fig. 43. Velocity vector plot for problem R-12-9: $t=0.62$ s.

the fluid jetting is too extended a structure. In comparison of our Figs. 42 and 43, at $t = 0.60$ and 0.65 sec respectively, with the data at $t = 0.62$ sec, there is a general qualitative agreement that the bubble has broken away from the negative z tank wall and is roughly in the center of the system, but the numerical bubble structure is much more fragmented than indicated by the data, the numerical bubble does not extend as far to positive x values as does the data, and the numerical bubble is concentrated at negative z values while the data are at positive z values. There seems also in this problem to be an improvement in long-term predictive capability for bubble location in NASA-VOF3D.

VI. FURTHER DEVELOPMENTS

The most urgent needs for improvements in the present version of NASA-VOF3D are: a) inclusion of a realistic, yet tractable, wall-adhesion algorithm, b) improvements in the decision-making algorithms which choose cell-by-cell between several possible evaluations of the NF array, c) a general "shakedown" period of improving the running of the code, and d) a physical model for wall-friction. We discuss these items in order.

The prototype of an acceptable and accurate wall-adhesion algorithm is given for 2D in Reference 11. The 2D version involves 1) initial calculation of trigonometric factors defining the orientation of obstacles with respect to the coordinate surfaces, 2) initial calculation of an array specifying the cell geometry of obstacles, 3) a time-dependent decision making procedure that determines whether surface-tension or wall-adhesion is to be applied at a cell interface and, if wall-adhesion is selected, whether the adhesion force points "up" or "down" the obstacle wall. In 3D there must be, in addition, a time-dependent calculation of trigonometric factors related to the angle which the adhesion force makes with the "up" direction at the obstacle wall. There will also be a somewhat more intricate procedure for selecting the wall-adhesion option and specifying the adhesion force. It has been shown in 2D that accurate specification of the wall-adhesion force is necessary for accurate prediction of some flow features.

The present decision-making algorithms for selection of surface-cell orientation can scarcely be regarded as a final solution for this important topic; they suffice for the present problems, and, presumably, for many others. We have abundant numerical evidence that there will always be some NF value that yields a

good surface-tension calculation; the problem is to find the appropriate NF value in almost all cases. The present algorithms give appropriate NF values a large percentage of the time for a number of examples. They also limit fairly effectively the damage that an inappropriate choice of NF will do to the simulation. The problem of devising algorithms that optimize the percentage of appropriate NF values in the widest variety of cases must be left for future research.

The code requires a period of time devoted to running many problems and correcting little awkwardnesses that will inevitably show up during this process. Possibly, some improvements in the conjugate residual technique will further speed up the code. Certainly the number of cases to which the code can be routinely applied would increase. The anticipated result of such a "shakedown" period would be a substantial increase in the amount of "user-friendliness" displayed by the code.

NASA-VOF3D was developed as a design tool to study bulk fluid and vapor motion in a low-g environment. It meets this goal. Even in its present form it probably suffices for many applications. But if it were to be used for precision comparison with experiment, then the aesthetically annoying feature that a physical mechanism--wall friction--is here modeled by a variety of sometimes competing numerical errors will begin to degrade the code's overall performance. In such a context it would be useful to model the wall-friction explicitly.

We close by noting that there are contexts in which one is interested in a highly viscous fluid in a low-g environment. To treat such a case by the present code would require a substantial revision of the solution algorithm. Viscous forces would have to be included more directly in the pressure iterations.

ACKNOWLEDGMENT

We are indebted to Drs. John K. Dukowicz and John R. Travis for assistance in implementing the conjugate algorithms.

REFERENCES

1. Nichols, B. D. and Hirt, C. W., "Methods for Calculating Multi-Dimensional, Transient Free Surface Flows Past Bodies," Proc. of the First Intern. Conf. Num. Ship Hydrodynamics, Gaithersburg, MD, October 1975.
2. Nichols, B. D., Hirt, C. W., and Hotchkiss, R. S., "SOLA-VOF: A Solution Algorithm for Transient Fluid Flow with Multiple Free Boundaries," Los Alamos Scientific Laboratory report LA-8355, 1980.
3. Hirt, C. W. and Nichols, B. D., 1981, *J. Comput. Phys.* 39, 201.
4. Johnson, W. E., "Development and Application of Computer Programs Related to Hypervelocity Impact," Systems, Science and Software report 3SR-353, 1970.
5. Ramshaw, J. D. and Trapp, J. A, 1976, *J. Comput. Phys.* 21, 438.
6. McMaster, W. H. and Gong, E. Y., "PELE-IC User's Manual," Lawrence Livermore Laboratory report UCRL-52609, 1979.
7. McMaster, W. H., Quiñones, D. F., Landram, C. S., Norris, D. M., Gong, E. Y., Machen, N. A., and Nickell, R. E., "Applications of the Coupled Fluid-Structure Code PELE-IC to Pressure Suppression Analysis, Annual Report to NRC for 1979," NUREG/CR-1179, UCRL-52733, 1980.
8. Kershner, J. D. and Mader, C. L., "2DE: A Two-Dimensional Continuous Eulerian Hydrodynamic Code for Computing Multicomponent Reactive Hydrodynamic Problems," Los Alamos Scientific Laboratory report LA-4846, 1972.
9. Noh, W. H. and Woodward, P., "The SLIC (Simple Line Interface Calculation)," in Lecture Notes in Physics, Vol. 59; proceedings of the Fifth Intern. Conf. on Numerical Methods in Fluid Dynamics, eds. A. I. van de Vooren and P. J. Zandbergen, Springer-Verlag, 1976.
10. Hotchkiss, R. S., "Simulation of Tank Draining Phenomena with the NASA SOLA-VOF Code," Los Alamos Scientific Laboratory report LA-8163-MS, 1979.
11. Torrey, M. D., Cloutman, L. D., Mjolsness, R. C., and Hirt, C. W., "NASA-VOF2D: A Computer Program for Incompressible Flows with Free Surfaces," Los Alamos National Laboratory report LA-10612-MS, 1985.
12. Symons, E. P., "Draining Characteristics of Hemispherically Bottomed Cylinders in a Low-Gravity Environment," NASA Technical paper 1297, 1978.
13. Carney, M., "Results of Shuttle/Centaur Zero-G Tests," NASA LeRC Internal report, March 1985.
14. Sicilian, J. M. and Hirt, C. W., "HYDR3-D, A Solution Algorithm for Transient 3D Flows," Flow Science, Inc. report FSI-84-00-3, 1985.
15. Hochstein, J. I., "Computational Prediction of Propellant Motion During Separation of a Centaur G-Prime Vehicle from the Shuttle," Washington University report WU/CFDL-85/1, December 1985.

APPENDIX
FORTRAN LISTING OF THE NASA-VOF3D SOURCE CODE

```

1   1 *CD SLCOM1          SLCOM1  1
2   2   PARAMETER (NQ=6, IBAR2=20, JBAR2=20, KBAR2=20, IBASC=NQ*IBAR2  SLCOM1  2
3   3   1 *JBAR2*KBAR2)    SLCOM1  3
4   4 C
5   5 C   **** NOTE: IBAR2,JBAR2,KBAR2 MUST EQUAL IMAX,JMAX,KMAX OF MESH  SLCOM1  4
6   6 C   GENERATION WHEN CONJUGATE RESIDUAL SOLUTION METHOD      SLCOM1  5
7   7 C   IS USED IN THE PRESSURE ITERATION      SLCOM1  6
8   8 C
9   9   PARAMETER (NQ2=6, NVEC=NQ2*IBAR2*KBAR2)    SLCOM1  7
10 10 C
11 11 C   **** NOTE: TWX(NVEC) IS WORKING ARRAY FOR PLOT ROUTINES  SLCOM1  8
12 12 C   NVEC MUST EQUAL NQ2 TIMES THE LARGER OF      SLCOM1  9
13 13 C   IBAR2*KBAR2, IBAR2*KBAR2, JBAR2*KBAR2      SLCOM1 10
14 14 C
15 15 COMMON /SLCM1/ BASC(IBASC) /SLCM2/ BASC1(IBASC) /SLCM3/ BASC2  SLCOM1 11
16 16 1 (IBASC) /SLCM4/ BASC3(IBASC) /SLCM5/ TWX(NVEC)    SLCOM1 12
17 17 C
18 18   COMMON /SSCM1/ AA(1), ALPHA, AUTOT, CLK, CYL, CYCLE, DAT, DELMN,  SLCOM1 13
19 19   1 DELT, DELX(100), DELXRL, DELXRR, DELY(100), DELYRBK, DELYRF, DELZ  SLCOM1 14
20 20   2 (100), DELZRB, DELZRT, DLZ, DMPX, DMPY, DMPZ, DTVIS, DXMN(10),  SLCOM1 15
21 21   3 DYMN(10), DZMN(10), EPSI, FIXL(3), FIXR, FIYB, FIYT, GRDBN(24),  SLCOM1 16
22 22   4 GX, GY, GZ, I, IBAR, ICLIP, IGRD, IIO, II1, II2, II3, II4, IJK,  SLCOM1 17
23 23   5 IJKM, IJKP, IJMK, IJPK, IJPKM, IJPKP, IM1, IM2, II5, II6,  SLCOM1 18
24 24   6 II7, IMAX, IMJK, IMJKM, IMJKP, IMJPK, IPJK, IPJKM, IPJMK,  SLCOM1 19
25 25   7 IPUPK, IMJMK, IJMMK, IMUPKP, IPUMKP, IPUPKM, IPUPPK, IDEFM, ISOR,  SLCOM1 20
26 26   8 ITER, ITITLE1(3), ITITLE2(3), IZOOM, J, JBAR, JM1, JM2, JMAX, JNM  SLCOM1 21
27 27   9 , K, KBAR, KM1, KM2, KMAX, LPR, NAME(8), NC, NCR1, NCR2, NNX, NNY  SLCOM1 22
28 28 C
29 29   COMMON /SSCM2/ NNZ, NFCAL, NZX, NZY, NZZ, NTITLE1, NTITLE2, NU,  SLCOM1 23
30 30   1 NUMTD, NXL(11), NXLBL, NXR(10), NYL(11), NYLBL, NYR(10), NZL(11),  SLCOM1 24
31 31   2 NZLBL, NZR(10), OMG, PLTDT, PRTDT, Q(50,50), RDX(100), RDY(100),  SLCOM1 25
32 32   3 RDZ(100), RIJK, RX(100), RXI(100), RDXP(100), RDYP(100), RDZP(100  SLCOM1 26
33 33   4 ), RY(100), RYU(100), RZ(100), RZK(100), STIM, T, TWTD, TD,  SLCOM1 27
34 34   5 TLIMD, TLM, TWFIN, TWPLT, TWPRT, UI, VELMX, VI, WB, WBK, WF, WI,  SLCOM1 28
35 35   6 WL, WR, WT, X(100), XBL, XBLC, XBR, XC(10), XCONV(3), XI(100), XL  SLCOM1 29
36 36   7 (11), XLABLE, Y(100), YBBK, YBF, YBFC, YC(10), YCONV(3), YJ(100),  SLCOM1 30
37 37   8 YL(11), YLABLE, Z(100), ZBB, ZBT, ZC(10), ZCQ(50), ZK(100), ZL(11  SLCOM1 31
38 38   9 ), ZLABLE, ZMN, ZMX      SLCOM1 32
39 39 C
40 40   COMMON /SSCM4/ JOP(100), JC2PI, SANG, CSANG, ICSURF, IEQUIB, SIGMA  SLCOM1 33
41 41   1 , CANGLE, ISURFT, TANCA, STHJ(99), IORDER, CTHJBK(99), CTHJBK(99),  SLCOM1 34
42 42   2 STHJBK(99), FLHT, AVE, FLG, UDUM, VDUM, WDUM, EMF, EMF1, NOWALL,  SLCOM1 35
43 43   3 RHOF, VCHGT, FLGC, FNOC, PR(100), NFLGC, NOCON, VOFTOT, PI, EM6  SLCOM1 36
44 44 C
45 45   COMMON /SSCM5/ GXA(NVEC), GYA(NVEC), RADPS      SLCOM1 37
46 46 C
47 47   COMMON /SSCM5A/ NOBS, DA2(10), DA1(10), DB2(10), DB1(10), DC2(10),  SLCOM1 38
48 48   1 DC1(10), IDH(10), NQBS, QA2(10), QA1(10), QB2(10), QB1(10), QC2(1  SLCOM1 39
49 49   2 0), QC1(10), IQH(10), QA3(10), QB3(10), QC3(10), QD3(10), QD2(10)  SLCOM1 40
50 50   3 , QD1(10)      SLCOM1 41
51 51 C
52 52   COMMON /SSCM6/ IC1(10), IC2(10), ICVEW(10), IPERC(10), IPERS(10),  SLCOM1 42
53 53   1 IPERV(10), IS1(10), IS2(10), ISVEW(10), IV1(10), IV2(10), IVVEW(1  SLCOM1 43
54 54   2 0), JC1(10), JC2(10), JS1(10), JS2(10), JV1(10), JV2(10), KC1(10)  SLCOM1 44
55 55   3 , KC2(10), KS1(10), KS2(10), KV1(10), KV2(10), LPMAX, NAC(10),  SLCOM1 45
56 56   4 NAS(10), NAV(10), NCPLTS, NSPLTS, NVEWS, NVPLTS, RR(100), RRI(100  SLCOM1 46
57 57   5 ), XCA(10), XEA(10), YCA(10), YEA(10), ZCA(10), ZEA(10), ZLAST  SLCOM1 47
58 58 C
59 59 C   * * NOTE ZLAST IS END OF COMMON BLOCK DUMPPED TO TAPE      SLCOM1 48
60 60 C
61 61   DIMENSION U(1), V(1), W(1), UN(1), VN(1), WN(1), BETA(1), D(1), P(  SLCOM1 49
62 62   1 1)      SLCOM1 50
63 63 C
64 64   EQUIVALENCE (BASC(1),U), (BASC(2),V), (BASC(3),W), (BASC(4),UN),  SLCOM1 51
65 65   1 (BASC(5),VN), (BASC(6),WN), (BASC1(1),BETA), (BASC1(2),D), (BASC1  SLCOM1 52
66 66   2 (3),P)      SLCOM1 53
67 67 C
68 68   DIMENSION UVV(1), VVV(1), WVV(1), XPC(1), YPC(1), ZPC(1)      SLCOM1 54
69 69 C
70 70   EQUIVALENCE (TWX(1),UVV), (TWX(2),VVV), (TWX(3),WVV), (TWX(4),XPC)  SLCOM1 55
71 71   1 , (TWX(5),YPC), (TWX(6),ZPC)      SLCOM1 56
72 72 C
73 73   DIMENSION F(1), NF(1), PS(1), PETA(1), FN(1), AR(1), AT(1), ABK(1)  SLCOM1 57
74 74   1 , AC(1), PN(1)      SLCOM1 58
75 75 C
76 76   EQUIVALENCE (BASC2(1),F), (BASC2(2),FN), (BASC2(3),PETA), (BASC2(4  SLCOM1 59
77 77   1 ),NF), (BASC2(5),PS), (BASC2(6),PN)      SLCOM1 60
78 78 C
79 79   EQUIVALENCE (BASC3(1),AR), (BASC3(2),AT), (BASC3(3),ABK), (BASC3(4  SLCOM1 61
80 80   1 ),AC)      SLCOM1 62

```

```

81 81 C           SLCOM1      81
82 82     DIMENSION NFP(1), NFS(1), NFO(1)          SLCOM1      82
83 83 C           SLCOM1      83
84 84     EQUIVALENCE (BASC1(4),NFP), (BASC1(5),NFS), (BASC1(6),NFO)  SLCOM1      84
85 85 C           SLCOM1      85
86 86     INTEGER CYCLE, WL, WR, WT, WB, WF, WBK, TD          SLCOM1      86
87 87     REAL LONG, NU          SLCOM1      87
88 88 C           SLCOM1      88
89 89 *CD SLCOM2      SLCOM2      1
90 90 C * * SSCMB IS FOR PERSPECTIVE GRAPHICS          SLCOM2      2
91 91 C           SLCOM2      3
92 92     COMMON /SSCMB/ CSPHI, CSTH, ETAMN, ETAMX, GDRAT, SNPHI, SNTH, XCC,  SLCOM2      4
93 93     1 XEB, XIMN, XIMX, YCC, YEB, ZCC, ZEB          SLCOM2      5
94 94 C           SLCOM2      6
95 95 *DK CONTROL          CONTROL      1
96 96     PROGRAM CONTROL (INPUT,XOUTPUT,TTY,TAPE59=TTY,TAPE10=INPUT,TAPE9  CONTROL      2
97 97     1 =XOUTPUT,TAPE7,TAPE8,TAPE2)          CONTROL      3
98 98 C           CONTROL      4
99 99 *CA SLCOM1      CONTROL      5
100 100 C * * READ PROBLEM INPUT DATA          CONTROL      6
101 101 C           CONTROL      7
102 102     CALL RINPUT          CONTROL      8
103 103 C           CONTROL      9
104 104 C * * SKIP ALL SET-UP ROUTINES FOR TAPE RESTART RUN          CONTROL      10
105 105 C           CONTROL     11
106 106     IF (TD) 10,10,20          CONTROL     12
107 107     10 CONTINUE          CONTROL     13
108 108 C           CONTROL     14
109 109 C * * READ PROBLEM SET-UP DATA FOR THE MESH, GRAPHICS, ETC.  CONTROL     15
110 110 C           CONTROL     16
111 111     CALL MESHSET          CONTROL     17
112 112     CALL RGRAFIC          CONTROL     18
113 113     CALL RCONTUR          CONTROL     19
114 114     CALL SETUP          CONTROL     20
115 115     CALL FLMSET          CONTROL     21
116 116 C           CONTROL     22
117 117 C * * THE NUMERICAL SOLUTION ALGORITHM AND CYCLIC CONTROL  CONTROL     23
118 118 C OF THE CALCULATION IS PROVIDED BY SUBROUTINE SOLA          CONTROL     24
119 119 C           CONTROL     25
120 120     CALL SOLA          CONTROL     26
121 121     CALL EXITA (405)          CONTROL     27
122 122 C           CONTROL     28
123 123 C * * TAPE DUMP RESTART          CONTROL     29
124 124 C           CONTROL     30
125 125     20 CALL SOLA          CONTROL     31
126 126     CALL EXITA (100)          CONTROL     32
127 127 C           CONTROL     33
128 128     END          CONTROL     34
129 1 *DK ASET          ASET      1
130 2     SUBROUTINE ASET          ASET      2
131 3 *CA SLCOM1      ASET      3
132 4 C ***          ASET      4
133 5 C *** CONIC FCN=OA2*X*X+OA1*X+DB2*Y*Y+DB1*Y+OC2*X*Y+DC1  ASET      5
134 6 C *** INSIDE FCN=NEGATIVE VALUE          ASET      6
135 7 C *** IOH=1 ADD DBS INSIDE FCN, IOH=0 SUBTRACT DBS INSIDE FCN  ASET      7
136 8 C ***          ASET      8
137 9     DIMENSION IFLG(5), DIS(4), XM(5), YM(5)          ASET      9
138 10 C           ASET     10
139 11     EM6=1.0E-6          ASET     11
140 12     DO 20 K=1,KMAX          ASET     12
141 13     DO 20 J=1,JMAX          ASET     13
142 14     DO 20 I=1,IMAX          ASET     14
143 15     IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1          ASET     15
144 16     AR(IJK)=1.0          ASET     16
145 17     AT(IJK)=1.0          ASET     17
146 18     ABK(IJK)=1.0          ASET     18
147 19     AC(IJK)=1.0          ASET     19
148 20     BETA(IJK)=0.0          ASET     20
149 21     IF (I.EQ.IM1.AND.WR.LE.2) AR(IJK)=0.0          ASET     21
150 22     IF (K.EQ.KM1.AND.WT.LE.2) AT(IJK)=0.0          ASET     22
151 23     IF (J.EQ.JM1.AND.WBK.LE.2) ABK(IJK)=0.0          ASET     23
152 24     IF (I.EQ.1.AND.WL.LE.2) GO TO 10          ASET     24
153 25     IF (I.EQ.IMAX.AND.WR.LE.2) GO TO 10          ASET     25
154 26     IF (K.EQ.1.AND.WB.LE.2) GO TO 10          ASET     26
155 27     IF (K.EQ.KMAX.AND.WT.LE.2) GO TO 10          ASET     27
156 28     IF (J.EQ.1.AND.WF.LE.2) GO TO 10          ASET     28
157 29     IF (J.EQ.JMAX.AND.WBK.LE.2) GO TO 10          ASET     29
158 30     GO TO 20          ASET     30
159 31     10 AR(IJK)=0.0          ASET     31
160 32     AT(IJK)=0.0          ASET     32

```

```

161 33      ABK(IJK)=O.O          ASET 33
162 34      AC(IJK)=O.O          ASET 34
163 35      BETA(IJK)=-1.O       ASET 35
164 36      20 CONTINUE          ASET 36
165 37 C      IF (NOBS.LE.O) GO TO 300 ASET 37
166 38      J=2                 ASET 38
167 39 C      DD 250 N=1,NOBS   ASET 39
168 40      DD 240 K=2,KM1     ASET 40
169 41      DD 240 I=2,IM1     ASET 41
170 42      CALL IJKONLY        ASET 42
171 43      RDXDY=1.0/(DELX(I)*DELZ(K)) ASET 43
172 44      DIS(1)=DELZ(K)      ASET 44
173 45      DO 80 M=1,4         ASET 45
174 46      GO TO (30,40,50,60), M ASET 46
175 47      30 X1=X(I)          ASET 47
176 48      Y1=Z(K-1)          ASET 48
177 49      DIS(2)=DELX(I)      ASET 49
178 50      GO TO 70            ASET 50
179 51      40 Y1=Z(K)          ASET 51
180 52      X1=X(I)            ASET 52
181 53      DIS(3)=DELZ(K)      ASET 53
182 54      GO TO 70            ASET 54
183 55      50 X1=X(I-1)        ASET 55
184 56      Y1=Z(K-1)          ASET 56
185 57      DIS(4)=DELX(I)      ASET 57
186 58      GO TO 70            ASET 58
187 59      60 Y1=Z(K-1)        ASET 59
188 60      X1=X(I-1)          ASET 60
189 61      DIS(5)=DELZ(K)      ASET 61
190 62      IFLGM=0              ASET 62
191 63      FCONIC=OA2(N)*X1*X1+OA1(N)*X1+OB2(N)*Y1*Y1+OB1(N)*Y1+OC2(N)*X1*Y1 ASET 63
192 64      1 +OC1(N)          ASET 64
193 65      IF (FCONIC.LE.O.O) IFLG(M)=1 ASET 65
194 66      XM(M)=X1            ASET 66
195 67      YM(M)=Y1            ASET 67
196 68      70 IFLG(M)=0         ASET 68
197 69      80 CONTINUE          ASET 69
198 70      IFLG(5)=IFLG(1)      ASET 70
199 71      XM(5)=XM(1)          ASET 71
200 72      YM(5)=YM(1)          ASET 72
201 73      IFLGS=0              ASET 73
202 74      DD 90 M=1,4          ASET 74
203 75      90 IFLGS=IFLGS+IFLG(M) ASET 75
204 76      BRIJ=O.O            ASET 76
205 77      BTIJ=O.O            ASET 77
206 78      IF (IFLGS.EQ.O) GO TO 240 ASET 78
207 79      IF (IFLGS.LT.4) GO TO 100 ASET 79
208 80      BIJ=1.0             ASET 80
209 81      BRIJ=1.0             ASET 81
210 82      BTIJ=1.0             ASET 82
211 83      GO TO 220            ASET 83
212 84      100 IF (IFLG(1).EQ.1.AND.IFLG(2).EQ.1) BRIJ=1.0 ASET 84
213 85      IF (IFLG(2).EQ.1.AND.IFLG(3).EQ.1) BTIJ=1.0 ASET 85
214 86      DD 180 M=1,4          ASET 86
215 87      IF (IFLG(M).EQ.IFLG(M+1)) GO TO 180 ASET 87
216 88      X1=XM(M)            ASET 88
217 89      Y1=YM(M)            ASET 89
218 90      X2=XM(M+1)          ASET 90
219 91      Y2=YM(M+1)          ASET 91
220 92      IF (IFLG(M).EQ.O.O) GO TO 110 ASET 92
221 93      X2=XM(M)            ASET 93
222 94      Y2=YM(M)            ASET 94
223 95      X1=XM(M+1)          ASET 95
224 96      Y1=YM(M+1)          ASET 96
225 97      110 EPSIF=0.001*(ABS(X2-X1)+ABS(Y2-Y1)) ASET 97
226 98      SMN=O.O              ASET 98
227 99      FMN=OA2(N)*X2*X2+OA1(N)*X2+OB2(N)*Y2*Y2+OB1(N)*Y2+OC2(N)*X2*Y2+OC1 ASET 99
228 100     1 (N)                ASET 100
229 101     SMX=1.0             ASET 101
230 102     FMX=OA2(N)*X1*X1+OA1(N)*X1+OB2(N)*Y1*Y1+OB1(N)*Y1+OC2(N)*X1*Y1+DC1 ASET 102
231 103     1 (N)                ASET 103
232 104     S=0.5               ASET 104
233 105     120 XT=S*X1+(1.0-S)*X2 ASET 105
234 106     YT=S*Y1+(1.0-S)*Y2 ASET 106
235 107     FS=OA2(N)*XT*XT+OA1(N)*XT+OB2(N)*YT*YT+OB1(N)*YT+OC2(N)*XT*YT+OC1 ASET 107
236 108     1 (N)                ASET 108
237 109     IF (ABS(FS).LT.EPSIF) GO TO 150 ASET 109
238 110     IF (FS.GE.O.O) GO TO 130 ASET 110
239 111     FDEN=ABS(FS-FMN)+1.OE-10 ASET 111
240 112     SE=S-FS*(S-SMN)/FDEN ASET 112

```

```

241 113      IF (SE.GT.SMX) SE=SMX          ASET    113
242 114      FMN=FS                      ASET    114
243 115      SMN=S                       ASET    115
244 116      GO TO 140                   ASET    116
245 117      130 FDEN=ABS(FMX-FS)+1.OE-10   ASET    117
246 118      SE=S-FS*(SMX-S)/FDEN        ASET    118
247 119      IF (SE.LT.SMN) SE=SMN        ASET    119
248 120      FMX=FS                      ASET    120
249 121      SMX=S                       ASET    121
250 122      140 SI=S-FS*(SMX-SMN)/(FMX-FMN) ASET    122
251 123      S=0.5*(SE+SI)             ASET    123
252 124      GO TO 120                   ASET    124
253 125      150 DIS(M)=SQRT((XT-X2)**2+(YT-Y2)**2) ASET    125
254 126      GO TO (160,170,180,180), M     ASET    126
255 127      160 BRIJ=DIS(1)/DELZ(K)       ASET    127
256 128      GO TO 180                   ASET    128
257 129      170 BTIJ=DIS(2)/DELX(I)       ASET    129
258 130      180 CONTINUE                 ASET    130
259 131      M=0                         ASET    131
260 132      BIJ=0.0                     ASET    132
261 133      190 CONTINUE                 ASET    133
262 134      M=M+1                     ASET    134
263 135      IF (M.EQ.5) GO TO 210       ASET    135
264 136      IF (IFLG(M).EQ.0) GO TO 190   ASET    136
265 137      MP1=M+1                   ASET    137
266 138      IF (MP1.EQ.5) MP1=1         ASET    138
267 139      MM1=M-1                   ASET    139
268 140      IF (MM1.EQ.0) MM1=4         ASET    140
269 141      BIJ=BIJ+DIS(M)*DIS(MM1)     ASET    141
270 142      IF (IFLG(MP1).EQ.1) GO TO 200   ASET    142
271 143      DIS2=DIS(M)                 ASET    143
272 144      200 CONTINUE                 ASET    144
273 145      IF (IFLG(MM1).EQ.1) GO TO 190   ASET    145
274 146      DIS1=DIS(MM1)               ASET    146
275 147      GO TO 190                   ASET    147
276 148      210 CONTINUE                 ASET    148
277 149      IF (IFLGS.EQ.3) BIJ=BIJ-DIS1*DIS2   ASET    149
278 150      BIJ=0.5*BIJ*RDXDY        ASET    150
279 151      IF (BIJ.GT.1.0) BIJ=1.0         ASET    151
280 152      220 CONTINUE                 ASET    152
281 153      IF (IOH(N).EQ.0) GO TO 230       ASET    153
282 154      BIJ=-BIJ                   ASET    154
283 155      BRIJ=-BRIJ                 ASET    155
284 156      BTIJ=-BTIJ                 ASET    156
285 157      230 AC(IJK)=AC(IJK)+BIJ       ASET    157
286 158      IF (AC(IJK).GT.0.99) AC(IJK)=1.0   ASET    158
287 159      IF (AC(IJK).LT.0.01) AC(IJK)=0.0   ASET    159
288 160      ABK(IJK)=AC(IJK)            ASET    160
289 161      AR(IJK)=AR(IJK)+BRIJ        ASET    161
290 162      IF (AR(IJK).GT.0.99) AR(IJK)=1.0   ASET    162
291 163      IF (AR(IJK).LT.0.01) AR(IJK)=0.0   ASET    163
292 164      IF (I.EQ.IM1.AND.WR.LE.2) AR(IJK)=0.0   ASET    164
293 165      AT(IJK)=AT(IJK)+BTIJ        ASET    165
294 166      IF (AT(IJK).GT.0.99) AT(IJK)=1.0       ASET    166
295 167      IF (AT(IJK).LT.0.01) AT(IJK)=0.0       ASET    167
296 168      IF (K.EQ.KM1.AND.WT.LE.2) AT(IJK)=0.0   ASET    168
297 169      240 CONTINUE                 ASET    169
298 170      250 CONTINUE                 ASET    170
299 171      C
300 172      DD 270 K=1,KMAX           ASET    172
301 173      IF (WL.LE.2) GO TO 260       ASET    173
302 174      IJKL=NQ*(II5*(K-1)+IMAX+1)+1   ASET    174
303 175      IJKLM=IJKL-NQ              ASET    175
304 176      AR(IJKLM)=AR(IJKL)          ASET    176
305 177      AT(IJKLM)=AT(IJKL)          ASET    177
306 178      ABK(IJKLM)=ABK(IJKL)          ASET    178
307 179      AC(IJKLM)=AC(IJKL)          ASET    179
308 180      260 IF (WR.LE.2) GO TO 270       ASET    180
309 181      IJKR=NQ*(II5*(K-1)+IMAX+(IM1-1))+1   ASET    181
310 182      IJKRP=IJKR+NQ              ASET    182
311 183      AR(IJKRP)=AR(IJKR)          ASET    183
312 184      AT(IJKRP)=AT(IJKR)          ASET    184
313 185      ABK(IJKRP)=ABK(IJKR)          ASET    185
314 186      AC(IJKRP)=AC(IJKR)          ASET    186
315 187      270 CONTINUE                 ASET    187
316 188      C
317 189      DD 290 I=1,IMAX           ASET    188
318 190      IF (WB.LE.2) GO TO 280       ASET    189
319 191      IJKB=NQ*(II5+IMAX+(I-1))+1   ASET    190
320 192      IJKBM=IJKB-II2             ASET    191

```

321	193	AT(IJKBM)=AT(IJKB)	ASET	193
322	194	AR(IJKBM)=AR(IJKB)	ASET	194
323	195	ABK(IJKBM)=ABK(IJKB)	ASET	195
324	196	AC(IJKBM)=AC(IJKB)	ASET	196
325	197	280 IF (WT.LE.2) GO TO 290	ASET	197
326	198	IJKT=NQ*(II5*(KM1-1)+IMAX+(I-1))+1	ASET	198
327	199	IJKTP=IJKT+II2	ASET	199
328	200	AT(IJKTP)=AT(IJKT)	ASET	200
329	201	AR(IJKTP)=AR(IJKT)	ASET	201
330	202	ABK(IJKTP)=ABK(IJKT)	ASET	202
331	203	AC(IJKTP)=AC(IJKT)	ASET	203
332	204	290 CONTINUE	ASET	204
333	205	C 300 CONTINUE	ASET	205
335	207	C	ASET	207
336	208	J=2	ASET	208
337	209	DO 310 K=2,KM1	ASET	209
338	210	DO 310 I=2,IM1	ASET	210
339	211	CALL IJKAJCT	ASET	211
340	212	IF (AC(IJK).GT.EM6) GO TO 310	ASET	212
341	213	AR(IJK)=O.O	ASET	213
342	214	AR(IMJK)=O.O	ASET	214
343	215	AT(IJK)=O.O	ASET	215
344	216	AT(IJKM)=O.O	ASET	216
345	217	ABK(IJK)=O.O	ASET	217
346	218	ABK(IJMJK)=O.O	ASET	218
347	219	BETA(IJK)=-1.0	ASET	219
348	220	310 CONTINUE	ASET	220
349	221	C	ASET	221
350	222	C NOTE: ALL UPLANES HAVE SAME VALUES	ASET	222
351	223	C	ASET	223
352	224	DO 360 J=1,JMAX	ASET	224
353	225	IF (J.EQ.2) GO TO 360	ASET	225
354	226	DO 350 K=1,KMAX	ASET	226
355	227	DO 340 I=1,IMAX	ASET	227
356	228	I2K=NQ*(II5*(K-1)+IMAX+(I-1))+1	ASET	228
357	229	IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1	ASET	229
358	230	IF (J.EQ.1.AND.WF.LE.2) GO TO 330	ASET	230
359	231	IF (J.EQ.JMAX.AND.WBK.LE.2) GO TO 320	ASET	231
360	232	AR(IJK)=AR(I2K)	ASET	232
361	233	AT(IJK)=AT(I2K)	ASET	233
362	234	ABK(IJK)=ABK(I2K)	ASET	234
363	235	AC(IJK)=AC(I2K)	ASET	235
364	236	BETA(IJK)=BETA(I2K)	ASET	236
365	237	GO TO 340	ASET	237
366	238	320 IJMJK=IJK-II1	ASET	238
367	239	ABK(IJMJK)=O.O	ASET	239
368	240	330 AR(IJK)=O.O	ASET	240
369	241	AT(IJK)=O.O	ASET	241
370	242	ABK(IJK)=O.O	ASET	242
371	243	AC(IJK)=O.O	ASET	243
372	244	BETA(IJK)=-1.0	ASET	244
373	245	340 CONTINUE	ASET	245
374	246	350 CONTINUE	ASET	246
375	247	360 CONTINUE	ASET	247
376	248	C ***	ASET	248
377	249	C *** SET SPECIAL VALUES OF AR,AT,ABK HERE	ASET	249
378	250	C ***	ASET	250
379	251	RETURN	ASET	251
380	252	END	ASET	252
381	1	*DK BC	BC	1
382	2	SUBROUTINE BC	BC	2
383	3	*CA SLCOM1	BC	3
384	4	C	BC	4
385	5	C * * GENERAL-SURFACE BOUNDARY CONDITIONS	BC	5
386	6	C	BC	6
387	7	C * * SET LEFT AND RIGHT BDY CONDITIONS	BC	7
388	8	C	BC	8
389	9	DO 150 K=1,KMAX	BC	9
390	10	IF (CYL.EQ.O.O) GO TO 20	BC	10
391	11	SUM1=O.O	BC	11
392	12	SUM2=O.O	BC	12
393	13	SUM3=O.O	BC	13
394	14	DO 10 J=2,JM1	BC	14
395	15	IJK=NQ*(II5*(K-1)+IMAX*(J-1)+1)+1	BC	15
396	16	SUM1=SUM1+U(IJK)*CTHJ(J)-V(IJK)*STHJBK(J)	BC	16
397	17	SUM2=SUM2+U(IJK)*STHJ(J)+V(IJK)*CTHJBK(J)	BC	17
398	18	SUM3=SUM3+W(IJK)	BC	18
399	19	10 CONTINUE	BC	19
400	20	IF (ABS(SUM1).LT.EM6) SUM1=O.O	BC	20

```

401 21 IF (ABS(SUM2).LT.EM6) SUM2=0.0 BC 21
402 22 IF (ABS(SUM3).LT.EM6) SUM3=0.0 BC 22
403 23 SUM1=0.5*SUM1/JBAR BC 23
404 24 SUM2=0.5*SUM2/JBAR BC 24
405 25 SUM3=SUM3/JBAR BC 25
406 26 PHIO=1.570796327 BC 26
407 27 IF (SUM1.NE.0.0) PHIO=ATAN(SUM2/SUM1) BC 27
408 28 CPHIO=COS(PHIO) BC 28
409 29 SPHIO=SIN(PHIO) BC 29
410 30 VELXY=SQRT(SUM1**2+SUM2**2) BC 30
411 31 20 CONTINUE BC 31
412 32 DO 140 J=1,JMAX BC 32
413 33 ILJK=NQ*(II5*(K-1)+IMAX*(J-1)+1) BC 33
414 34 ILMJK=ILJK-NQ BC 34
415 35 ILPK=ILJK+NQ BC 35
416 36 IRJK=ILJK+IIO BC 36
417 37 IRMJK=IRJK-NQ BC 37
418 38 F(ILMJK)=F(ILJK) BC 38
419 39 P(ILMJK)=P(ILJK) BC 39
420 40 F(IRJK)=F(IRMJK) BC 40
421 41 P(IRJK)=P(IRMJK) BC 41
422 42 IF (CYL.EQ.1.0) GO TO 70 BC 42
423 43 GO TO (30,40,50,60,50), WL BC 43
424 44 30 U(ILMJK)=0.0 BC 44
425 45 V(ILMJK)=V(ILJK)*(1.0-CYL+CYL*XI(2)*RXI(1)) BC 45
426 46 W(ILMJK)=W(ILJK) BC 46
427 47 GO TO 80 BC 47
428 48 40 U(ILMJK)=0.0 BC 48
429 49 V(ILMJK)=-V(ILJK)*DELXRL BC 49
430 50 W(ILMJK)=-W(ILJK)*DELXRL BC 50
431 51 GO TO 80 BC 51
432 52 50 IF (ITER.GT.0) GO TO 80 BC 52
433 53 U(ILMJK)=U(ILJK)*RR(1)/RR(2) BC 53
434 54 V(ILMJK)=V(ILJK) BC 54
435 55 W(ILMJK)=W(ILJK) BC 55
436 56 GO TO 80 BC 56
437 57 60 U(ILMJK)=U(IRMJK) BC 57
438 58 V(ILMJK)=V(IRMJK) BC 58
439 59 W(ILMJK)=W(IRMJK) BC 59
440 60 F(ILMJK)=F(IRMJK) BC 60
441 61 NF(ILMJK)=NF(IRMJK) BC 61
442 62 P(ILMJK)=P(IRMJK) BC 62
443 63 PN(ILMJK)=PN(IRMJK) BC 63
444 64 PS(ILMJK)=PS(IRMJK) BC 64
445 65 PETA(ILMJK)=PETA(IRMJK) BC 65
446 66 GO TO 80 BC 66
447 67 70 IOPJK=NQ*(II5*(K-1)+IMAX*(JOP(J)-1)+1) BC 67
448 68 F(ILMJK)=F(IOPJK) BC 68
449 69 P(ILMJK)=P(IOPJK) BC 69
450 70 U(ILMJK)=VELXY*(CTHJ(J)*CPHIO+STHJ(J)*SPHIO) BC 70
451 71 V(ILMJK)=-VELXY*(STHJBK(J)*CPHIO-CTHJBK(J)*SPHIO) BC 71
452 72 W(ILMJK)=SUM3 BC 72
453 73 80 GO TO (90,100,110,120,110), WR BC 73
454 74 90 U(IRMJK)=0.0 BC 74
455 75 V(IRJK)=V(IRMJK)*(1.0-CYL+CYL*XI(IM1)*RXI(IMAX)) BC 75
456 76 W(IRJK)=W(IRMJK) BC 76
457 77 GO TO 130 BC 77
458 78 100 U(IRMJK)=0.0 BC 78
459 79 V(IRJK)=-V(IRMJK)*DELXRR BC 79
460 80 W(IRJK)=-W(IRMJK)*DELXRR BC 80
461 81 GO TO 130 BC 81
462 82 110 IF (ITER.GT.0) GO TO 130 BC 82
463 83 U(IRJK)=U(IRMJK)*RR(IMAX)/RR(IM1) BC 83
464 84 V(IRJK)=V(IRMJK) BC 84
465 85 W(IRJK)=W(IRMJK) BC 85
466 86 GO TO 130 BC 86
467 87 120 U(IRJK)=U(ILMJK) BC 87
468 88 V(IRJK)=V(ILMJK) BC 88
469 89 W(IRJK)=W(ILMJK) BC 89
470 90 F(IRJK)=F(ILJK) BC 90
471 91 NF(IRJK)=NF(ILJK) BC 91
472 92 P(IRJK)=P(ILJK) BC 92
473 93 PN(IRJK)=PN(ILJK) BC 93
474 94 PS(IRJK)=PS(ILJK) BC 94
475 95 PETA(IRJK)=PETA(ILJK) BC 95
476 96 130 CONTINUE BC 96
477 97 140 CONTINUE BC 97
478 98 150 CONTINUE BC 98
479 99 C 480 100 C * * SET FRONT AND BACK BDY. CONDITIONS BC 99
                                         BC 100

```

```

481 101 C
482 102 DO 270 K=1,KMAX
483 103 DO 260 I=1,IMAX
484 104 IF JK=NQ*(II5*(K-1)+IMAX+I-1)+1
485 105 IF JMK=IFJK-II1
486 106 IF JPK=IFJK+II1
487 107 IBKJK=IFJK+II3
488 108 IBKJMK=IBKJK-II1
489 109 F(IFJMK)=F(IFJK)
490 110 P(IFJMK)=P(IFJK)
491 111 F(IBKJK)=F(IBKJMK)
492 112 P(IBKJK)=P(IBKJMK)
493 113 GO TO (160,170,180,190,180), WF
494 114 160 U(IFJMK)=U(IFJK)
495 115 V(IFJMK)=O.O
496 116 W(IFJMK)=W(IFJK)
497 117 GO TO 200
498 118 170 U(IFJMK)=-U(IFJK)*DELYRF
499 119 V(IFJMK)=O.O
500 120 W(IFJMK)=-W(IFJK)*DELYRF
501 121 GO TO 200
502 122 180 IF (ITER.GT.0) GO TO 200
503 123 U(IFJMK)=U(IFJK)
504 124 V(IFJMK)=V(IFJK)
505 125 W(IFJMK)=W(IFJK)
506 126 GO TO 200
507 127 190 U(IFJMK)=U(IBKJMK)
508 128 V(IFJMK)=O.5*(V(IFJMK)+V(IBKJMK))
509 129 IF (ISOR.EQ.0) V(IFJMK)=V(IBKJMK)
510 130 W(IFJMK)=W(IBKJMK)
511 131 F(IFJMK)=F(IBKJMK)
512 132 NF(IFJMK)=NF(IBKJMK)
513 133 P(IFJMK)=P(IBKJMK)
514 134 PN(IFJMK)=PN(IBKJMK)
515 135 PS(IFJMK)=PS(IBKJMK)
516 136 PETA(IFJMK)=PETA(IBKJMK)
517 137 200 GO TO (210,220,230,240,230), WBK
518 138 210 U(IBKJK)=U(IBKJMK)
519 139 V(IBKJMK)=O.O
520 140 W(IBKJK)=W(IBKJMK)
521 141 GO TO 250
522 142 220 U(IBKJK)=-U(IBKJMK)*DELYRBK
523 143 V(IBKJMK)=O.O
524 144 W(IBKJK)=-W(IBKJMK)*DELYRBK
525 145 GO TO 250
526 146 230 IF (ITER.GT.0) GO TO 250
527 147 U(IBKJK)=U(IBKJMK)
528 148 V(IBKJK)=V(IBKJMK)
529 149 W(IBKJK)=W(IBKJMK)
530 150 GO TO 250
531 151 240 U(IBKJK)=U(IFJK)
532 152 W(IBKJK)=W(IFJK)
533 153 V(IBKJK)=V(IFJK)
534 154 V(IBKJMK)=V(IFJMK)
535 155 F(IBKJK)=F(IFJK)
536 156 NF(IBKJK)=NF(IFJK)
537 157 P(IBKJK)=P(IFJK)
538 158 PN(IBKJK)=PN(IFJK)
539 159 PS(IBKJK)=PS(IFJK)
540 160 PETA(IBKJK)=PETA(IFJK)
541 161 250 CONTINUE
542 162 260 CONTINUE
543 163 270 CONTINUE
544 164 C
545 165 C * * SET BOTTOM AND TOP BDY. CONDITIONS
546 166 C
547 167 DO 390 J=2,JM1
548 168 DO 380 I=2,IM1
549 169 IBJK=NQ*(II5+IMAX*(J-1)+I-1)+1
550 170 IBJKM=IBJK-II2
551 171 ITJK=IBJK+II4
552 172 ITJJKM=ITJK-II2
553 173 F(IBJKM)=F(IBJK)
554 174 P(IBJKM)=P(IBJK)
555 175 F(ITJK)=F(ITJJKM)
556 176 P(ITJK)=P(ITJJKM)
557 177 GO TO (280,290,300,310,300), WB
558 178 280 U(IBJKM)=U(IBJK)
559 179 V(IBJKM)=V(IBJK)
560 180 W(IBJKM)=O.O

```

```

561 181      GO TO 320
562 182      U(IBJKM)=-U(IBJK)*DELZRB          BC    181
563 183      V(IBJKM)=-V(IBJK)*DELZRB          BC    182
564 184      W(IBJKM)=0.0                      BC    183
565 185      GO TO 320                         BC    184
566 186      300 IF (ITER.GT.0) GO TO 320       BC    185
567 187      U(IBJKM)=U(IBJK)                  BC    186
568 188      V(IBJKM)=V(IBJK)                  BC    187
569 189      W(IBJKM)=W(IBJK)                  BC    188
570 190      GO TO 320                         BC    189
571 191      310 U(IBJKM)=U(ITJKM)            BC    190
572 192      V(IBJKM)=V(ITJKM)                BC    191
573 193      W(IBJKM)=0.5*(W(IBJKM)+W(ITJKM)) BC    192
574 194      F(IBJKM)=F(ITJKM)                BC    193
575 195      NF(IBJKM)=NF(ITJKM)              BC    194
576 196      P(IBJKM)=P(ITJKM)                BC    195
577 197      PN(IBJKM)=PN(ITJKM)              BC    196
578 198      PS(IBJKM)=PS(ITJKM)              BC    197
579 199      PETA(IBJKM)=PETA(ITJKM)           BC    198
580 200      320 GO TO (330,340,350,360,350), WT BC    199
581 201      330 U(ITJK)=U(ITJKM)            BC    200
582 202      V(ITJK)=V(ITJKM)                BC    201
583 203      W(ITJK)=W(ITJKM)                BC    202
584 204      GO TO 370                         BC    203
585 205      340 U(ITJK)=-U(ITJKM)*DELZRT     BC    204
586 206      V(ITJK)=-V(ITJKM)*DELZRT         BC    205
587 207      W(ITJK)=0.0                      BC    206
588 208      GO TO 370                         BC    207
589 209      350 IF (ITER.GT.0) GO TO 370       BC    208
590 210      U(ITJK)=U(ITJKM)                  BC    209
591 211      V(ITJK)=V(ITJKM)                  BC    210
592 212      W(ITJK)=W(ITJKM)                  BC    211
593 213      GO TO 370                         BC    212
594 214      360 U(ITJK)=U(IBJK)              BC    213
595 215      V(ITJK)=V(IBJK)                  BC    214
596 216      W(ITJK)=W(IBJKM)                BC    215
597 217      F(ITJK)=F(IBJK)                  BC    216
598 218      NF(ITJK)=NF(IBJK)                BC    217
599 219      P(ITJK)=P(IBJK)                  BC    218
600 220      PN(ITJK)=PN(IBJK)                BC    219
601 221      PS(ITJK)=PS(IBJK)                BC    220
602 222      PETA(ITJK)=PS(IBJK)             BC    221
603 223      370 CONTINUE                     BC    222
604 224      380 CONTINUE                     BC    223
605 225      390 CONTINUE                     BC    224
606 226      C
607 227      C * * SET FREE SURFACE BOUNDARY CONDITIONS BC    225
608 228      C
609 229      CALL BCFS                         BC    226
610 230      C
611 231      C * * SPECIAL BOUNDARY CONDITIONS BC    227
612 232      C * * (FOR SPECIFIED IN OR OUT FLOW SET BETA=+1.0 IN FICTITIOUS CELLS) BC    228
613 233      C
614 234      RETURN                          BC    229
615 235      END
616 1      *DK BCFS
617 2      SUBROUTINE BCFS
618 3      *CA SLCOM1
619 4      C
620 5      C * * SET FREE SURFACE BOUNDARY CONDITIONS BC    230
621 6      C
622 7      DO 220 K=2,KM1
623 8      DO 220 J=2,JM1
624 9      DO 220 I=2,IM1
625 10     KNFF=0
626 11     CALL CALCIJK
627 12     C
628 13     C * * SET F AND P VALUES IN OBSTACLE EDGE CELLS BC    231
629 14     C
630 15     IF (BETA(IJK).GE.0.0) GO TO 10
631 16     ABR=1.0
632 17     IF (BETA(IPJK).LT.0.0) ABR=0.0
633 18     ABL=1.0
634 19     IF (BETA(IMJK).LT.0.0) ABL=0.0
635 20     ABBK=1.0
636 21     IF (BETA(IJPK).LT.0.0) ABBK=0.0
637 22     ABF=1.0
638 23     IF (BETA(IJMK).LT.0.0) ABF=0.0
639 24     ABT=1.0
640 25     IF (BETA(IJKP).LT.0.0) ABT=0.0

```

```

641 26      ABB=1.0
642 27      IF (BETA(IJKM).LT.0.0) ABB=0.0
643 28      ABSUM=ABR+ABL+ABBK+ABF+ABT+ABB
644 29      IF (ABSUM.LT.0.5) GO TO 10
645 30      F(IJK)=(ABR+F(IPJK)+ABL+F(IMJK)+ABBK+F(IJPK)+ABF+F(IJMK)+ABT+F
646 31      1 (IJKP)+ABB+F(IJKM))/ABSUM
647 32      P(IJK)=(ABR*P(IPJK)+ABL*P(IMJK)+ABBK*P(IJPK)+ABF*P(IJMK)+ABT*P
648 33      1 (IJKP)+ABB*P(IJKM))/ABSUM
649 34      GO TO 220
650 35      10 CONTINUE
651 36      IF (NMAT.EQ.2) GO TO 230
652 37      NFF=NF(IJK)
653 38      IF (NFF.EQ.0.OR.NFF.GT.7) GO TO 220
654 39 C    * * SET VELOCITY ON CELL BOUNDARY BETWEEN PARTIALLY
655 40 C    GO TO (20,30,40,50,60,70,80), NFF
656 41      20 IF (AR(IJK).GT.EM6) U(IJK)=U(IMJK)*AR(IMJK)*X(I-1)/(AR(IJK)*X(I))
657 42      IF (ABK(IJK).GT.EM6) V(IJK)=V(IMJK)
658 43      IF (ABK(IJMK).GT.EM6) V(IJMK)=V(IPJK)
659 44      IF (AT(IJK).GT.EM6) W(IJK)=W(IMJK)
660 45      IF (AT(IJKM).GT.EM6) W(IJKM)=W(IPJK)
661 46      IF (AT(IJMK).GT.EM6) W(IJMK)=W(IPJKM)
662 47      GO TO 80
663 48      30 IF (AR(IMJK).GT.EM6) U(IMJK)=U(IJK)*AR(IJK)*X(I)/(AR(IMJK)*X(I-1))
664 49      IF (ABK(IJK).GT.EM6) V(IJK)=V(IPJK)
665 50      IF (ABK(IJMK).GT.EM6) V(IJMK)=V(IPJKM)
666 51      IF (AT(IJK).GT.EM6) W(IJK)=W(IPJK)
667 52      IF (AT(IJKM).GT.EM6) W(IJKM)=W(IPJKM)
668 53      GO TO 80
669 54      40 IF (ABK(IJK).GT.EM6) V(IJK)=V(IJMK)
670 55      IF (AR(IJK).GT.EM6) U(IJK)=U(IJMK)
671 56      IF (AR(IMJK).GT.EM6) U(IMJK)=U(IMJPK)
672 57      IF (AT(IJK).GT.EM6) W(IJK)=W(IJPK)
673 58      IF (AT(IJKM).GT.EM6) W(IJKM)=W(IJMKM)
674 59      GO TO 80
675 60      50 IF (ABK(IJK).GT.EM6) V(IJMK)=V(IJK)
676 61      IF (AR(IJK).GT.EM6) U(IJK)=U(IJPK)
677 62      IF (AR(IMJK).GT.EM6) U(IMJK)=U(IMJPK)
678 63      IF (AT(IJK).GT.EM6) W(IJK)=W(IJPK)
679 64      IF (AT(IJKM).GT.EM6) W(IJKM)=W(IJPKM)
680 65      GO TO 80
681 66      60 IF (AT(IJK).GT.EM6) W(IJK)=W(IJKM)
682 67      IF (AR(IJK).GT.EM6) U(IJK)=U(IJKM)
683 68      IF (AR(IMJK).GT.EM6) U(IMJK)=U(IMJPM)
684 69      IF (ABK(IJK).GT.EM6) V(IJK)=V(IJKM)
685 70      IF (ABK(IJKM).GT.EM6) V(IJMK)=V(IJMKM)
686 71      GO TO 80
687 72      70 IF (AT(IJKM).GT.EM6) W(IJKM)=W(IJK)
688 73      IF (AR(IJK).GT.EM6) U(IJK)=U(IJKP)
689 74      IF (AR(IMJK).GT.EM6) U(IMJK)=U(IMJKP)
690 75      IF (ABK(IJK).GT.EM6) V(IJK)=V(IJKP)
691 76      IF (ABK(IJMK).GT.EM6) V(IJMK)=V(IJMKP)
692 77      GO TO 80
693 78 C    * * CALCULATE VELOCITY ON NF SELECTED CELL BOUNDARY
694 79 C    GO TO (90,100,110,120,130,140,140), NFF
695 80 C    80 IF (NF(IPJK).LE.7.OR.AR(IJK).LT.EM6) GO TO 150
696 81      DENOM=-RRI(I)*RDX(I)*AR(IJK)/RR(I)
697 82      U(IJK)=(RRI(I)*(RDX(I)*(-U(IMJK)*AR(IMJK)/RR(I-1))+RDY(J)*(V(IJK)
698 83      1 *ABK(IJK)-V(IJMK)*ABK(IJMK)))+RDZ(K)*(W(IJK)*AT(IJK)-W(IJKM)*AT
700 84      2 (IJKM)))/DENOM
701 85      GO TO 160
702 86
703 88      100 IF (NF(IMJK).LE.7.OR.AR(IMJK).LT.EM6) GO TO 150
704 89      DENOM=-RRI(I)*RDX(I)*AR(IMJK)/RR(I-1)
705 90      U(IMJK)=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I))+RDY(J)*(V(IJK)*ABK
706 91      1 (IJK)-V(IJMK)*ABK(IJMK)))+RDZ(K)*(W(IJK)*AT(IJK)-W(IJKM)*AT(IJKM)
707 92      2 ))/DENOM
708 93      GO TO 160
709 94      110 IF (NF(IJKP).LE.7.OR.ABK(IJK).LT.EM6) GO TO 150
710 95      DENOM=-RRI(I)*RDY(J)*ABK(IJK)
711 96      V(IJK)=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR(I
712 97      1 -1))+RDY(J)*(-V(IJMK)*ABK(IJMK)))+RDZ(K)*(W(IJK)*AT(IJK)-W(IJKM)
713 98      2 *AT(IJKM)))/DENOM
714 99      GO TO 160
715 100     120 IF (NF(IJMK).LE.7.OR.ABK(IJMK).LT.EM6) GO TO 150
716 101     DENOM=RRI(I)*RDY(J)*ABK(IJMK)
717 102     V(IJMK)=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR
718 103     1 (I-1))+RDY(J)*(V(IJK)*ABK(IJK)))+RDZ(K)*(W(IJK)*AT(IJK)-W(IJKM)
719 104     2 *AT(IJKM)))/DENOM
720 105     GO TO 160

```

```

721 106 130 IF (NF(IJKP).LE.7.OR.AT(IJK).LT.EM6) GO TO 150 BCFS 106
722 107 DENOM=-RDZ(K)*AT(IJK) BCFS 107
723 108 W(IJK)=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR(I BCFS 108
724 109 1 -1))+RDY(J)*(V(IJK)*ABK(IJK)-V(IUMK)*ABK(IUMK)))+RDZ(K)*(-W(IJKM) BCFS 109
725 110 2 *AT(IJKM)))/DENOM BCFS 110
726 111 GO TO 160 BCFS 111
727 112 140 IF (NF(IJKM).LE.7.OR.AT(IJKM).LT.EM6) GO TO 150 BCFS 112
728 113 DENOM=RDZ(K)*AT(IJKM) BCFS 113
729 114 W(IJKM)=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR BCFS 114
730 115 1 (I-1))+RDY(J)*(V(IJK)*ABK(IJK)-V(IUMK)*ABK(IUMK)))+RDZ(K)*(W(IJK) BCFS 115
731 116 2 *AT(IJK)))/DENOM BCFS 116
732 117 GO TO 160 BCFS 117
733 118 150 CONTINUE BCFS 118
734 119 C KNFF=KNFF+1 BCFS 119
735 120 C * * IF NF DESIGNATED CELL IS NOT EMPTY TRY ALL NEIGHBORING BCFS 120
736 121 C CELLS TO FIND EMPTY NEIGHBOR BCFS 121
737 122 C
738 123 KNFF=KNFF+1 BCFS 123
739 124 NFF=NFF+1 BCFS 124
740 125 IF (NFF.GT.7) NFF=1 BCFS 125
741 126 IF (KNFF.GT.7) GO TO 160 BCFS 126
742 127 GO TO 80 BCFS 127
743 128 160 CONTINUE BCFS 128
744 129 C
745 130 C * * SET VELOCITIES IN EMPTY CELLS ADJACENT TO PARTIAL FLUID CELLS BCFS 130
746 131 C
747 132 IF (FLG.GT.0.5.AND.ITER.GT.0.AND.ISOR.EQ.1) GO TO 220 BCFS 132
748 133 IF (NF(IPJK).LE.7.OR.BETA(IPJK).LT.0.0) GO TO 170 BCFS 133
749 134 IF (NF(IPUPK).GT.7.AND.ABK(IPJK).GT.EM6) V(IPJK)=F(IJK)*V(IJK)*RRI BCFS 134
750 135 1 (I)/RRI(I+1) BCFS 135
751 136 IF (NF(IPUMK).GT.7.AND.ABK(IPUMK).GT.EM6) V(IPUMK)=F(IJK)*V(IUMK) BCFS 136
752 137 1 *RRI(I)/RRI(I+1) BCFS 137
753 138 IF (NF(IPUPK).GT.7.AND.AT(IPJK).GT.EM6) W(IPJK)=F(IJK)*W(IJK) BCFS 138
754 139 IF (NF(IPUPK).GT.7.AND.AT(IPUMK).GT.EM6) W(IPUMK)=F(IJK)*W(IJKM) BCFS 139
755 140 170 IF (NF(IMJK).LE.7.OR.BETA(IMJK).LT.0.0) GO TO 180 BCFS 140
756 141 IF (NF(IMUPK).GT.7.AND.ABK(IMJK).GT.EM6) V(IMJK)=F(IJK)*V(IJK)*RRI BCFS 141
757 142 1 (I)/RRI(I-1) BCFS 142
758 143 IF (NF(IMUMK).GT.7.AND.ABK(IMUMK).GT.EM6) V(IMUMK)=F(IJK)*V(IUMK) BCFS 143
759 144 1 *RRI(I)/RRI(I-1) BCFS 144
760 145 IF (NF(IMUPK).GT.7.AND.AT(IMJK).GT.EM6) W(IMJK)=F(IJK)*W(IJK) BCFS 145
761 146 IF (NF(IMJKM).GT.7.AND.AT(IMJKM).GT.EM6) W(IMJKM)=F(IJK)*W(IJKM) BCFS 146
762 147 180 IF (NF(IUPK).LE.7.OR.BETA(IUPK).LT.0.0) GO TO 190 BCFS 147
763 148 IF (NF(IUPK).GT.7.AND.AR(IMUPK).GT.EM6) U(IMUPK)=F(IJK)*U(IMJK) BCFS 148
764 149 IF (NF(IUPK).GT.7.AND.AR(IUPK).GT.EM6) U(IUPK)=F(IJK)*U(IJK) BCFS 149
765 150 IF (NF(IUPK).GT.7.AND.AT(IUPK).GT.EM6) W(IUPK)=F(IJK)*W(IJK) BCFS 150
766 151 IF (NF(IUPK).GT.7.AND.AT(IUPKM).GT.EM6) W(IUPKM)=F(IJK)*W(IJKM) BCFS 151
767 152 190 IF (NF(IUMK).LE.7.OR.BETA(IUMK).LT.0.0) GO TO 200 BCFS 152
768 153 IF (NF(IUMJK).GT.7.AND.AR(IUMJK).GT.EM6) U(IUMJK)=F(IJK)*U(IMJK) BCFS 153
769 154 IF (NF(IUPJM).GT.7.AND.AR(IUMK).GT.EM6) U(IUMK)=F(IJK)*U(IJK) BCFS 154
770 155 IF (NF(IUMK).GT.7.AND.AT(IUMK).GT.EM6) W(IUMK)=F(IJK)*W(IJK) BCFS 155
771 156 IF (NF(IUMKM).GT.7.AND.AT(IUMKM).GT.EM6) W(IUMKM)=F(IJK)*W(IJKM) BCFS 156
772 157 200 IF (NF(IUPK).LE.7.OR.BETA(IUPK).LT.0.0) GO TO 210 BCFS 157
773 158 IF (NF(IUPK).GT.7.AND.AR(IUPK).GT.EM6) U(IUPK)=F(IJK)*U(IJK) BCFS 158
774 159 IF (NF(IUMKP).GT.7.AND.AR(IUMKP).GT.EM6) U(IUMKP)=F(IJK)*U(IMJK) BCFS 159
775 160 IF (NF(IUPK).GT.7.AND.ABK(IUPK).GT.EM6) V(IUPK)=F(IJK)*V(IJK) BCFS 160
776 161 IF (NF(IUMKP).GT.7.AND.ABK(IUMKP).GT.EM6) V(IUMKP)=F(IJK)*V(IUMK) BCFS 161
777 162 210 IF (NF(IUMK).LE.7.OR.BETA(IUMK).LT.0.0) GO TO 220 BCFS 162
778 163 IF (NF(IUMK).GT.7.AND.AR(IUMK).GT.EM6) U(IUMK)=F(IJK)*U(IJK) BCFS 163
779 164 IF (NF(IUMKM).GT.7.AND.AR(IUMKM).GT.EM6) U(IUMKM)=F(IJK)*U(IMJK) BCFS 164
780 165 IF (NF(IUPK).GT.7.AND.ABK(IUMK).GT.EM6) V(IUMK)=F(IJK)*V(IJK) BCFS 165
781 166 IF (NF(IUMKM).GT.7.AND.ABK(IUMKM).GT.EM6) V(IUMKM)=F(IJK)*V(IUMK) BCFS 166
782 167 220 CONTINUE BCFS 167
783 168 230 CONTINUE BCFS 168
784 169 RETURN BCFS 169
785 170 END BCFS 170
786 1 *DK BETACAL BETACAL 1
787 2 . SUBROUTINE BETACAL BETACAL 2
788 3 *CA SLCOM1 BETACAL 3
789 4 C BETACAL 4
790 5 C * * CALCULATE BETA(IJK) FOR MESH INTERIOR BETACAL 5
791 6 C * * NOTE THAT ALL FICTITIOUS CELLS HAVE BETA SET TO ZERO IN ASET BETACAL 6
792 7 C * * NOTE THAT ALL OBSTACLE CELLS HAVE BETA SET TO -1.0 IN ASET BETACAL 7
793 8 C BETACAL 8
794 9 C * * CALCULATE BETA(IJK) FOR NON-OBSTACLE CELLS BETACAL 9
795 10 C BETACAL 10
796 11 RIJK=0.0 BETACAL 11
797 12 DO 10 K=2,KM1 BETACAL 12
798 13 DO 10 J=2,JM1 BETACAL 13
799 14 DO 10 I=2,IM1 BETACAL 14
800 15 CALL IJKAJCT BETACAL 15

```

```

801 16      IF (BETA(IJK).LT.0.0) GO TO 10          BETACAL   16
802 17      RIJK=RIJK+1.                         BETACAL   17
803 18      ABR=AR(IJK).                         BETACAL   18
804 19      IF (BETA(IPJK).LT.0.0.OR.BETA(IPJK).EQ.1.0) ABR=0.0 BETACAL   19
805 20      ABL=AR(IMJK).                         BETACAL   20
806 21      IF (BETA(IMJK).LT.0.0.OR.BETA(IMJK).EQ.1.0) ABL=0.0 BETACAL   21
807 22      ABBK=ABK(IJK).                         BETACAL   22
808 23      IF (BETA(IJPK).LT.0.0.OR.BETA(IJPK).EQ.1.0) ABBK=0.0 BETACAL   23
809 24      ABF=ABK(IJMK).                         BETACAL   24
810 25      IF (BETA(IJMK).LT.0.0.OR.BETA(IJMK).EQ.1.0) ABF=0.0 BETACAL   25
811 26      ABT=AT(IJK).                          BETACAL   26
812 27      IF (BETA(IJKP).LT.0.0.OR.BETA(IJKP).EQ.1.0) ABT=0.0 BETACAL   27
813 28      ABB=AT(IJKM).                         BETACAL   28
814 29      IF (BETA(IJKM).LT.0.0.OR.BETA(IJKM).EQ.1.0) ABB=0.0 BETACAL   29
815 30      XX=2.0*DELT*(RDX(I)*(ABR/(DELX(I)+DELX(I+1))+ABL/(DELX(I)+DELX(I-1 BETACAL   30
816 31      1 )))+RDY(J)*RRI(I)*(ABBK*RRI(I)/(DELY(J)+DELY(J+1))+ABF*RRI(I)/ BETACAL   31
817 32      2 (DELY(J)+DELY(J-1)))+RDZ(K)*(ABT/(DELZ(K)+DELZ(K+1))+ABB/(DELZ(K) BETACAL   32
818 33      3 +DELZ(K-1)))+CYL*0.5*(ABR/(DELX(I)+DELX(I+1))-ABL/(DELX(I-1)+DELX BETACAL   33
819 34      4 (I))*RRI(I)/X(IM1)).                         BETACAL   34
820 35      BETA(IJK)=OMG/XX*AC(IJK).                         BETACAL   35
821 36      10 CONTINUE.                         BETACAL   36
822 37 C
823 38      RIJK=1.0/RIJK.                         BETACAL   37
824 39      RETURN.                            BETACAL   39
825 40      END.
826 1 *DK CALCIJK.                           CALCIJK   1
827 2      SUBROUTINE CALCIJK.                   CALCIJK   2
828 3 *CA SLCOM1.                           CALCIJK   3
829 4 C
830 5 C * * CALCULATE "IJK" AND OTHER INDICES FOR CELL (I,J,K) CALCIJK   5
831 6 C
832 7      IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1 CALCIJK   7
833 8      IMJK=IJK-NQ.                          CALCIJK   8
834 9      IPJK=IJK+NQ.                          CALCIJK   9
835 10     IJMK=IJK-II1.                         CALCIJK  10
836 11     IJPK=IJK+II1.                         CALCIJK  11
837 12     IJKM=IJK-II2.                         CALCIJK  12
838 13     IJKP=IJK+II2.                         CALCIJK  13
839 14     IPJMK=IJK+NQ-II1.                      CALCIJK  14
840 15     IPJKM=IJK+NQ-II2.                      CALCIJK  15
841 16     IMUPK=IJK-NQ+II1.                      CALCIJK  16
842 17     IMJMK=IJK-NQ-II1.                      CALCIJK  17
843 18     IJPKM=IJK+II1-II2.                     CALCIJK  18
844 19     IMJKP=IJK-NQ+II2.                      CALCIJK  19
845 20     IJMKP=IJK-II1+II2.                     CALCIJK  20
846 21     IJMKM=IJK-II1-II2.                     CALCIJK  21
847 22     IPJKP=IJK+NQ+II2.                      CALCIJK  22
848 23     IMJKM=IJK-NQ-II2.                      CALCIJK  23
849 24     IPJPK=IJK+NQ+II1.                      CALCIJK  24
850 25     IJPKP=IJK+II1+II2.                     CALCIJK  25
851 26     IMJPKP=IJK-NQ+II1+II2.                  CALCIJK  26
852 27     IPJMKP=IJK+NQ-II1+II2.                  CALCIJK  27
853 28     IPJPKM=IJK+NQ+II1-II2.                  CALCIJK  28
854 29     IPJPKP=IJK+NQ+II1+II2.                  CALCIJK  29
855 30 C
856 31      RETURN.                            CALCIJK  31
857 32 C
858 33      ENTRY IJKONLY.                      CALCIJK  32
859 34 C
860 35      IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1 CALCIJK  35
861 36      RETURN.                            CALCIJK  36
862 37 C
863 38      ENTRY IJKAJCT.                      CALCIJK  37
864 39 C
865 40      IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1 CALCIJK  39
866 41      IMJK=IJK-NQ.                          CALCIJK  40
867 42      IPJK=IJK+NQ.                          CALCIJK  41
868 43      IJMK=IJK-II1.                         CALCIJK  42
869 44      IJPK=IJK+II1.                         CALCIJK  43
870 45      IJKM=IJK-II2.                         CALCIJK  44
871 46      IJKP=IJK+II2.                         CALCIJK  45
872 47      RETURN.                            CALCIJK  46
873 48 C
874 49      END.
875 1 *DK CNTR.                           CNTR      1
876 2      SUBROUTINE CNTR (I1,I2,J1,J2,K1,K2,NA,IPER) CNTR      2
877 3 *CA SLCOM1.                           CNTR      3
878 4 C
879 5 C * * CONTOUR PLOTTING ROUTINE (SET-UP IN SUBROUTINE RCONTUR) CNTR      4
880 6 C

```

```

881    7      COMMON /SSCM10/ MPLNC, IFMC, JFMC, KFMC, JNMC, DATC, CLKC, TC,      CNTR    7
882    8      1 ITERC, CYCLEC, NAMEC(10), I1C, I2C, J1C, J2C, K1C, K2C, IPERK      CNTR    8
883    9 C
884   10      INTEGER CYCLEC      CNTR    9
885   11      DO 10 I=1,50      CNTR   10
886   12      DO 10 J=1,50      CNTR   11
887   13      10 Q(I,J)=0.0      CNTR   12
888   14 C
889   15 C * * DETERMINE CONSTANT PLANE, MPLN      CNTR   13
890   16 C
891   17      IFM=0      CNTR   14
892   18      JFM=0      CNTR   15
893   19      KFM=0      CNTR   16
894   20      IF (I1.NE.I2) GO TO 20      CNTR   17
895   21      IFM=I1      CNTR   18
896   22      MPLN=1      CNTR   19
897   23      20 IF (J1.NE.J2) GO TO 30      CNTR   20
898   24      JFM=J1      CNTR   21
899   25      MPLN=2      CNTR   22
900   26      30 IF (K1.NE.K2) GO TO 40      CNTR   23
901   27      KFM=K1      CNTR   24
902   28      MPLN=3      CNTR   25
903   29      40 CONTINUE      CNTR   26
904   30 C
905   31 C * * SET CELL(2,2,2) REFERENCE INDEX      CNTR   27
906   32 C
907   33      I=2      CNTR   28
908   34      J=2      CNTR   29
909   35      K=2      CNTR   30
910   36      CALL CALCIJK      CNTR   31
911   37      IJKC=IJK      CNTR   32
912   38 C
913   39 C * * CALCULATE PRESSURE VALUES, Q-ARRAY, TO BE PLOTTED      CNTR   33
914   40 C
915   41      DO 80 K=2,KM1      CNTR   34
916   42      DO 80 J=2,JM1      CNTR   35
917   43      DO 80 I=2,IM1      CNTR   36
918   44      CALL CALCIJK      CNTR   37
919   45      IQ=I-1      CNTR   38
920   46      JQ=J-1      CNTR   39
921   47      KQ=K-1      CNTR   40
922   48      GO TO (50,60,70), MPLN      CNTR   41
923   49      50 IF (I.NE.I1) GO TO 80      CNTR   42
924   50      IF (J.LT.J1) GO TO 80      CNTR   43
925   51      IF (J.GT.J2) GO TO 80      CNTR   44
926   52      IF (K.LT.K1) GO TO 80      CNTR   45
927   53      IF (K.GT.K2) GO TO 80      CNTR   46
928   54      Q(JQ,KQ)=P(IJK)-P(IJKC)      CNTR   47
929   55      GO TO 80      CNTR   48
930   56      60 IF (J.NE.J1) GO TO 80      CNTR   49
931   57      IF (K.LT.K1) GO TO 80      CNTR   50
932   58      IF (K.GT.K2) GO TO 80      CNTR   51
933   59      IF (I.LT.I1) GO TO 80      CNTR   52
934   60      IF (I.GT.I2) GO TO 80      CNTR   53
935   61      Q(IQ,KQ)=P(IJK)-P(IJKC)      CNTR   54
936   62      GO TO 80      CNTR   55
937   63      70 IF (K.NE.K1) GO TO 80      CNTR   56
938   64      IF (J.LT.J1) GO TO 80      CNTR   57
939   65      IF (J.GT.J2) GO TO 80      CNTR   58
940   66      IF (I.LT.I1) GO TO 80      CNTR   59
941   67      IF (I.GT.I2) GO TO 80      CNTR   60
942   68      Q(IQ,JQ)=P(IJK)-P(IJKC)      CNTR   61
943   69      80 CONTINUE      CNTR   62
944   70      MPLNC=MPLN      CNTR   63
945   71      IFMC=IFM      CNTR   64
946   72      JFMC=JFM      CNTR   65
947   73      KFMC=KFM      CNTR   66
948   74      JNMC=JNM      CNTR   67
949   75      DATC=DAT      CNTR   68
950   76      CLKC=CLK      CNTR   69
951   77      TC=T      CNTR   70
952   78      ITERC=ITER      CNTR   71
953   79      CYCLEC=CYCLE      CNTR   72
954   80      DO 90 N=1,10      CNTR   73
955   81      90 NAMEC(N)=NAME(N)      CNTR   74
956   82      I1C=I1      CNTR   75
957   83      I2C=I2      CNTR   76
958   84      J1C=J1      CNTR   77
959   85      J2C=J2      CNTR   78
960   86      K1C=K1      CNTR   79

```

```

961 87      K2C=K2          CNTR   87
962 88 C
963 89 C * * CALL CONTOUR PLOTTING ROUTINES
964 90 C
965 91      NYM=NNY+1        CNTR   88
966 92      GO TO (100,110,120), MPLN    CNTR   89
967 93      100 CALL CONTRJB (YJ(2),NYM,ZK(2),NNZ,Q,NZX,NZY,NC,ZMN,ZMX,DLZ,ZCQ  CNTR   90
968 94      1 ,DMPY,DMPZ,IGRD,ITITLE1,NTITLE1,YLABEL,NYLBL,ZLABLE,NZLBL)    CNTR   91
969 95      GO TO 130         CNTR   92
970 96      110 CALL CONTRJB (XI(2),NNX,ZK(2),NNZ,Q,NZX,NZY,NC,ZMN,ZMX,DLZ,ZCQ  CNTR   93
971 97      1 ,DMPX,DMPZ,IGRD,ITITLE1,NTITLE1,XLABEL,NXLBL,ZLABLE,NZLBL)    CNTR   94
972 98      GO TO 130         CNTR   95
973 99      120 CALL CONTRJB (XI(2),NNX,YJ(2),NYM,Q,NZX,NZY,NC,ZMN,ZMX,DLZ,ZCQ  CNTR   96
974 100     1 ,DMPX,DMPY,IGRD,ITITLE1,NTITLE1,XLABEL,NXLBL,YLABEL,NYLBL)    CNTR   97
975 101     130 CONTINUE      CNTR   98
976 102 C
977 103     RETURN          CNTR   99
978 104     END              CNTR  100
979 1 *DK CONTRJB          CNTR  101
980 2      SUBROUTINE CONTRJB (X,NNX,Y,NNY,Z,NZX,NZY,NC,ZMN,ZMX,DLZ,ZPLAN  CONTRJB 102
981 3      1 ,DMPX,DMPY,IGRD,ITITLE,NTITLE,XLABEL,NXLBL,YLABEL,NYLBL)    CONTRJB 103
982 4      COMMON /CNTRCOM/ ISYM(50)        CONTRJB 104
983 5      COMMON /SSCM10/ MPLN, IFMC, JFMC, KFMC, JNMC, DATC, CLKC, TC,  CONTRJB 1
984 6      1 IERC, CYCLEC, NAMEC(10), I1C, I2C, J1C, J2C, K1C, K2C, IPERK  CONTRJB 2
985 7      DIMENSION XSCALE(2), YSCALE(2)      CONTRJB 3
986 8      EQUIVALENCE (XMIN,XSCALE(1)), (XMAX,XSCALE(2))    CONTRJB 4
987 9      EQUIVALENCE (YMIN,YSCALE(1)), (YMAX,YSCALE(2))    CONTRJB 5
988 10     DIMENSION X(1), Y(1), Z(NZX,1), ZPLAN(1)      CONTRJB 6
989 11     DIMENSION ZT(4)          CONTRJB 7
990 12     INTEGER CYCLE, CYCLEC      CONTRJB 8
991 13     NOC=MINO(IABS(NC),50)      CONTRJB 9
992 14     ZMIN=ZMN          CONTRJB 10
993 15     ZMAX=ZMX          CONTRJB 11
994 16     DLZZ=DLZ          CONTRJB 12
995 17     DMAPX=DMPX      CONTRJB 13
996 18     DMAPY=DMPY      CONTRJB 14
997 19     NOX=IABS>NNX)      CONTRJB 15
998 20     NOY=IABS>NNY)      CONTRJB 16
999 21     DO 10 I=1,50      CONTRJB 17
1000 22    10 ISYM(I)=0      CONTRJB 18
1001 23 C
1002 24 C * * ESTABLISH SCALES
1003 25 C
1004 26     CALL MINV (X,1,NOX,I,XMIN)      CONTRJB 19
1005 27     CALL MAXV (Y,1,NOY,I,YMAX)      CONTRJB 20
1006 28     CALL MAXV (X,1,NOX,I,XMAX)      CONTRJB 21
1007 29     CALL MINV (Y,1,NOY,I,YMIN)      CONTRJB 22
1008 30     FGRD=0.          CONTRJB 23
1009 31     IF (IGRD.GT.0) FGRD=-IGRD      CONTRJB 24
1010 32     CALL PLOJB (XSCALE,YSCALE,2,1,1,1,FGRD,DMAPX,DMAPY,ITITLE,-NTITLE  CONTRJB 25
1011 33     1 ,XLABEL,NXLBL,YLABEL,NYLBL)    CONTRJB 26
1012 34     IF (NC.LT.0) GO TO 50      CONTRJB 27
1013 35     IF (NNX.LE.0) CALL MINM (Z,NZX,NOX,NOY,I,J,ZMIN)    CONTRJB 28
1014 36     IF (NNY.LE.0) CALL MAXM (Z,NZX,NOX,NOY,I,J,ZMAX)    CONTRJB 29
1015 37     IF (DLZZ.GT.0) GO TO 20      CONTRJB 30
1016 38     DLZZ=(ZMAX-ZMIN)/(NOC-1.)      CONTRJB 31
1017 39     20 IF (NZY.GT.0) GO TO 30      CONTRJB 32
1018 40     ZMAX=ZMAX-AMOD(ZMAX,DLZZ)      CONTRJB 33
1019 41     ZMIN=ZMIN-AMOD(ZMIN,DLZZ)      CONTRJB 34
1020 42     NOC=MINO(NOC,IFIX((ZMAX-ZMIN)/DLZZ+1.01))    CONTRJB 35
1021 43     30 ZPLAN(1)=ZMIN      CONTRJB 36
1022 44     DO 40 I=2,NOC      CONTRJB 37
1023 45     40 ZPLAN(I)=ZPLAN(I-1)+DLZZ      CONTRJB 38
1024 46     50 DO 90 NY=2,NOY      CONTRJB 39
1025 47     IX=MOD(NY,2)      CONTRJB 40
1026 48     DY=Y(NY)-Y(NY-1)      CONTRJB 41
1027 49     DO 80 INX=2,NOX      CONTRJB 42
1028 50     NX=INX          CONTRJB 43
1029 51     IF, (IX.NE.0) NX=NOX-INX+2      CONTRJB 44
1030 52     ZT(1)=Z(NX-1,NY-1)      CONTRJB 45
1031 53     ZT(2)=Z(NX,NY-1)      CONTRJB 46
1032 54     ZT(3)=Z(NX,NY)      CONTRJB 47
1033 55     ZT(4)=Z(NX-1,NY)      CONTRJB 48
1034 56     DX=X(NX)-X(NX-1)      CONTRJB 49
1035 57     IF (ABS(ZT(3))-ZT(1))-ABS(ZT(4)-ZT(2))) 70,60,60      CONTRJB 50
1036 58     60 CALL TRICJB (X(NX),Y(NY),-DX,-DY,NOZ,ZPLAN,ZT(4),ZT(3),ZT(2))  CONTRJB 51
1037 59     CALL TRICJB (X(NX-1),Y(NY-1),DX,DY,NOZ,ZPLAN,ZT(2),ZT(1),ZT(4))  CONTRJB 52
1038 60     GO TO 80          CONTRJB 53
1039 61     70 CALL TRICJB (X(NX-1),Y(NY),DX,-DY,NOZ,ZPLAN,ZT(3),ZT(4),ZT(1))  CONTRJB 54
1040 62     CALL TRICJB (X(NX),Y(NY-1),-DX,DY,NOZ,ZPLAN,ZT(1),ZT(2),ZT(3))  CONTRJB 55

```

```

1041 63   80 CONTINUE          CONTRJB 63
1042 64   90 CONTINUE          CONTRJB 64
1043 65   CALL LINCNT (1)      CONTRJB 65
1044 66   WRITE (12,230) (NAMEC(I),I=1,5),JNMC,DATC,CLKC CONTRJB 66
1045 67   CALL LINCNT (59)      CONTRJB 67
1046 68   WRITE (12,240) ITERC,TC,CYCLEC    CONTRJB 68
1047 69   GO TO (100,110,120), MPLNC     CONTRJB 69
1048 70   100 WRITE (12,250) IFMC,J1C,J2C,K1C,K2C  CONTRJB 70
1049 71   GO TO 130             CONTRJB 71
1050 72   110 WRITE (12,260) JFMC,I1C,I2C,K1C,K2C  CONTRJB 72
1051 73   GO TO 130             CONTRJB 73
1052 74   120 WRITE (12,270) KFMC,I1C,I2C,J1C,J2C  CONTRJB 74
1053 75   130 CONTINUE          CONTRJB 75
1054 76   N=1                 CONTRJB 76
1055 77   DO 140 I=1,NDC        CONTRJB 77
1056 78   ZTT=ABS(ZPLAN(I))    CONTRJB 78
1057 79   IF (ZTT.EQ.0.OR.(ZTT.GT..01.AND.(ZTT.LT.10000))) GO TO 140 CONTRJB 79
1058 80   N=2                 CONTRJB 80
1059 81   GO TO 150             CONTRJB 81
1060 82   140 CONTINUE          CONTRJB 82
1061 83   150 I1=1              CONTRJB 83
1062 84   I2=MINO(NDC,30)       CONTRJB 84
1063 85   160 CALL FADV (1)     CONTRJB 85
1064 86   CALL LINCNT (56)      CONTRJB 86
1065 87   WRITE (12,230) (NAMEC(I),I=1,5),JNMC,DATC,CLKC CONTRJB 87
1066 88   WRITE (12,240) ITERC,TC,CYCLEC    CONTRJB 88
1067 89   GO TO (170,180,190), MPLNC     CONTRJB 89
1068 90   170 WRITE (12,250) IFMC,J1C,J2C,K1C,K2C  CONTRJB 90
1069 91   GO TO 200             CONTRJB 91
1070 92   180 WRITE (12,260) JFMC,I1C,I2C,K1C,K2C  CONTRJB 92
1071 93   GO TO 200             CONTRJB 93
1072 94   190 WRITE (12,270) KFMC,I1C,I2C,J1C,J2C  CONTRJB 94
1073 95   200 CONTINUE          CONTRJB 95
1074 96   KYS=20               CONTRJB 96
1075 97   CALL EXL              CONTRJB 97
1076 98   CALL DLCH (MAXO(0,460-NTITLE*6),KYS,NTITLE,ITITLE,1) CONTRJB 98
1077 99   KYS=KYS+60            CONTRJB 99
1078 100  CALL DLCH (250,KYS,35,35HIDENTIFICATION      CONTOUR VALUE,1) CONTRJB 100
1079 101  DO 220 I=1,I2          CONTRJB 101
1080 102  KYS=KYS+30            CONTRJB 102
1081 103  IF (N.EQ.1) ENCODE (8,280,BCDA) ZPLAN(I)    CONTRJB 103
1082 104  IF (N.EQ.2) ENCODE (8,290,BCDA) ZPLAN(I)    CONTRJB 104
1083 105  CALL EXH              CONTRJB 105
1084 106  CALL DRVEC (250,KYS,400,KYS)    CONTRJB 106
1085 107  CALL EXL              CONTRJB 107
1086 108  DO 210 K=1,4           CONTRJB 108
1087 109  210 CALL DLCH (K*50+200,KYS,0,I,1)    CONTRJB 109
1088 110  220 CALL DLCH (550,KYS-6,10,BCDA,1)    CONTRJB 110
1089 111  CALL EXH              CONTRJB 111
1090 112  IF (I2.EQ.NDC) RETURN    CONTRJB 112
1091 113  I1=I2+1              CONTRJB 113
1092 114  I2=NDC               CONTRJB 114
1093 115  GO TO 160             CONTRJB 115
1094 116 C
1095 117  230 FORMAT (1H ,2X,5A8,1X,A8,2(1X,A8))  CONTRJB 117
1096 118  240 FORMAT (1X,5HITER=,I4,18X,6HTIME= ,1PE12.5,6X,7HCYCLE= ,I4) CONTRJB 118
1097 119  250 FORMAT (13H CONSTANT I=,I4,17H SURFACE   J=,I3,3H TO,I3,7H CONTRJB 119
1098 120  1 K=.I3,4H TO,I3)      CONTRJB 120
1099 121  260 FORMAT (13H CONSTANT J=,I4,17H SURFACE   I=,I3,3H TO,I3,7H CONTRJB 121
1100 122  1 K=.I3,4H TO,I3)      CONTRJB 122
1101 123  270 FORMAT (13H CONSTANT K=,I4,17H SURFACE   I=,I3,3H TO,I3,7H CONTRJB 123
1102 124  1 J=.I3,4H TO,I3)      CONTRJB 124
1103 125  280 FORMAT (F8.3)      CONTRJB 125
1104 126  290 FORMAT (1PE8.1)    CONTRJB 126
1105 127  END                  CONTRJB 127
1106 1 *DK DELTADJ          DELTADJ 1
1107 2   SUBROUTINE DELTADJ    DELTADJ 2
1108 3 *CA SLCOM1            DELTADJ 3
1109 4 C
1110 5   DATA CON, ITMIN, ITMOST, LITER, ITCRMX /0.40,15,30,50,200/ DELTADJ 5
1111 6 C * * ADJUST TIME STEP (DELT)    DELTADJ 6
1112 7 C
1113 8   DELTN=DELT          DELTADJ 7
1114 9   IF (FLGC.LT.0.5) GO TO 20    DELTADJ 8
1115 10 C
1116 11 C * * F CONVECTION LIMIT EXCEEDED DELTADJ 10
1117 12 C
1118 13 T=T-DELT          DELTADJ 11
1119 14 CYCLE=CYCLE-1      DELTADJ 12
1120 15 DELT=0.5*DELT      DELTADJ 13

```

```

1121 16      DO 10 K=1,KMAX          DELTADJ 16
1122 17      DO 10 J=1,JMAX          DELTADJ 17
1123 18      DO 10 I=1,IMAX          DELTADJ 18
1124 19      CALL CALCIJK          DELTADJ 19
1125 20      F(IJK)=FN(IJK)          DELTADJ 20
1126 21      U(IJK)=UN(IJK)          DELTADJ 21
1127 22      V(IJK)=VN(IJK)          DELTADJ 22
1128 23      W(IJK)=WN(IJK)          DELTADJ 23
1129 24      10 CONTINUE          DELTADJ 24
1130 25      NFLGC=NFLGC+1          DELTADJ 25
1131 26      20 CONTINUE          DELTADJ 26
1132 27      IF (AUTOT.LT.0.5.AND.FNOC.LT.0.5) GO TO 90  DELTADJ 27
1133 28 C    * * DETERMINE MAXIMUM VELOCITY TERM  DELTADJ 28
1134 29 C    * * DETERMINE MAXIMUM VELOCITY TERM  DELTADJ 29
1135 30 C
1136 31      IDUMX=1          DELTADJ 30
1137 32      JDUMX=1          DELTADJ 31
1138 33      KDUMX=1          DELTADJ 32
1139 34      IDVMX=1          DELTADJ 33
1140 35      JDVMX=1          DELTADJ 34
1141 36      KDVMX=1          DELTADJ 35
1142 37      IDWMX=1          DELTADJ 36
1143 38      JDWMX=1          DELTADJ 37
1144 39      KDWMX=1          DELTADJ 38
1145 40      IMX=1          DELTADJ 39
1146 41      JMX=1          DELTADJ 40
1147 42      KMX=1          DELTADJ 41
1148 43      IJKMX=1          DELTADJ 42
1149 44      IJKUMX=1          DELTADJ 43
1150 45      IJKVMX=1          DELTADJ 44
1151 46      IJKWMX=1          DELTADJ 45
1152 47      DUMX=1.OE-10*DELX(2)/DELT          DELTADJ 46
1153 48      DVMX=1.OE-10*DELY(2)/RRI(2)/DELT          DELTADJ 47
1154 49      DWMX=1.OE-10*DELZ(2)/DELT          DELTADJ 48
1155 50      DUVW=AMAX1(DUMX,DVMX,DWMX)          DELTADJ 49
1156 51      IF (FNOC.GT.0.5) DELT=0.5*DELT          DELTADJ 50
1157 52      DO 60 K=2,KM1          DELTADJ 51
1158 53      DO 60 J=2,JM1          DELTADJ 52
1159 54      DO 60 I=2,IM1          DELTADJ 53
1160 55      CALL IJKONLY          DELTADJ 54
1161 56      IF (NF(IJK).NE.0) GO TO 60          DELTADJ 55
1162 57      IF (BETA(IJK).LT.0.0) GO TO 60          DELTADJ 56
1163 58      UDM=ABS(UN(IJK))/(XI(I+1)-XI(I))          DELTADJ 57
1164 59      VDM=ABS(VN(IJK))/(YJ(J+1)-YJ(J))*RRI(I)          DELTADJ 58
1165 60      WDM=ABS(WN(IJK))/(ZK(K+1)-ZK(K))          DELTADJ 59
1166 61      IF (UDM.LT.DUMX) GO TO 30          DELTADJ 60
1167 62      DUMX=UDM          DELTADJ 61
1168 63      IDUMX=I          DELTADJ 62
1169 64      JDUMX=J          DELTADJ 63
1170 65      KDUMX=K          DELTADJ 64
1171 66      IJKUMX=IJK          DELTADJ 65
1172 67      30 IF (VDM.LT.DVMX) GO TO 40          DELTADJ 66
1173 68      DVMX=VDM          DELTADJ 67
1174 69      JDVMX=J          DELTADJ 68
1175 70      IDVMX=I          DELTADJ 69
1176 71      KDVMX=K          DELTADJ 70
1177 72      IJKVMX=IJK          DELTADJ 71
1178 73      40 IF (WDM.LT.DWMX) GO TO 50          DELTADJ 72
1179 74      DWMX=WDM          DELTADJ 73
1180 75      KDWMX=K          DELTADJ 74
1181 76      IDWMX=I          DELTADJ 75
1182 77      JDWMX=J          DELTADJ 76
1183 78      IJKWMX=IJK          DELTADJ 77
1184 79      50 UVWS=AMAX1(UDM,VDM,WDM)          DELTADJ 78
1185 80      IF (DUVW.GT.UVWS) GO TO 60          DELTADJ 79
1186 81      DUVW=UVWS          DELTADJ 80
1187 82      IMX=I          DELTADJ 81
1188 83      JMX=J          DELTADJ 82
1189 84      KMX=K          DELTADJ 83
1190 85      IJKMX=IJK          DELTADJ 84
1191 86      60 CONTINUE          DELTADJ 85
1192 87 C    * * ADJUST DELT TO REFLECT NUMBER OF ITERATIONS  DELTADJ 86
1193 88 C
1194 89 C
1195 90      DTMP=1.02          DELTADJ 87
1196 91      IF (ITER.LT.ITMIN.AND.LITER.LT.ITMIN) DTMP=1.04  DELTADJ 88
1197 92      IF (ISOR.NE.1) GO TO 70          DELTADJ 89
1198 93      IF (ITER.GT.ITMOST.AND.LITER.GT.ITMOST) DTMP=0.99  DELTADJ 90
1199 94      GO TO 80          DELTADJ 91
1200 95      70 IF (ITER.GT.ITCRMX.AND.LITER.GT.ITCRMX) DTMP=0.98  DELTADJ 92

```

```

1201 96      80 DELTO=DELT*DTMP          DELTADJ   96
1202 97 C    * * SELECT SMALLEST DELT    DELTADJ   97
1203 98 C
1204 99 C
1205 100     DELT=AMIN1(DELTO,CON/DUMX,CON/DVMX,CON/DWMX,DTVIS) DELTADJ 100
1206 101     90 IF (DELT.EQ.DELTN) GO TO 110 DELTADJ 101
1207 102 C
1208 103 C * * ADJUST BETA ARRAY TO REFLECT DELT CHANGE DELTADJ 102
1209 104 C
1210 105     IF (ISOR.NE.1) GO TO 110 DELTADJ 103
1211 106     DO 100 K=1,KMAX DELTADJ 104
1212 107     DO 100 J=1,JMAX DELTADJ 105
1213 108     DO 100 I=1,IMAX DELTADJ 106
1214 109     CALL IJKONLY DELTADJ 107
1215 110     IF (BETA(IJK).EQ.1.0) GO TO 100 DELTADJ 108
1216 111     IF (BETA(IJK).LT.0.0) GO TO 100 DELTADJ 109
1217 112     BETA(IJK)=BETA(IJK)*DELTN/DELT DELTADJ 110
1218 113     IF (BETA(IJK).EQ.1.0) BETA(IJK)=1.10 DELTADJ 111
1219 114     100 CONTINUE DELTADJ 112
1220 115     110 CONTINUE DELTADJ 113
1221 116 C
1222 117     LITER=ITER DELTADJ 114
1223 118     IF (MOD(CYCLE,20).NE.0) RETURN DELTADJ 115
1224 119     WRITE (9,120) CYCLE,IMX,JMX,KMX,IJKMX,U(IJKMX),V(IJKMX),W(IJKMX) DELTADJ 116
1225 120     WRITE (12,120) CYCLE,IMX,JMX,KMX,IJKMX,U(IJKMX),V(IJKMX),W(IJKMX) DELTADJ 117
1226 121     RETURN DELTADJ 118
1227 122 C
1228 123     120 FORMAT (2X,*DELTADJ *,I5,3I4,I6.3(3X,1PE12.5)) DELTADJ 119
1229 124     END DELTADJ 120
1230 1 *DK DGAP DGAP 1
1231 2     SUBROUTINE DGAP (XE,YE,ZE,XC1,YC1,ZC1) DGAP 2
1232 3 *CA SLCOM2 DGAP 3
1233 4 *CA SLCOM1 DGAP 4
1234 5 C
1235 6 C * * DEFINE GRAPH AREA FOR PERSPECTIVE VIEW PLOTS (NO FILM PRODUCED) DGAP 5
1236 7 C
1237 8     XCC=XC1 DGAP 6
1238 9     YCC=YC1 DGAP 7
1239 10    ZCC=ZC1 DGAP 8
1240 11    YEMYC=YE-YC1+1.OE-20 DGAP 9
1241 12    XXSQ=(XE-XC1)**2 DGAP 10
1242 13    YYSQ=YEMYC**2 DGAP 11
1243 14    ZZSQ=(ZE-ZC1)**2 DGAP 12
1244 15    RTXY=SQRT(XXSQ+YYSQ) DGAP 13
1245 16    RHP=1.0/RTXY DGAP 14
1246 17    RHP2=1.0/SQRT(XXSQ+YYSQ+ZZSQ) DGAP 15
1247 18    CSTH=-YEMYC*RHP DGAP 16
1248 19    SNTH=(XE-XC1)*RHP DGAP 17
1249 20    TRCR=YEMYC*CSTH-(XE-XC1)*SNTH DGAP 18
1250 21    CSPHI=RTXY*RHP2 DGAP 19
1251 22    SNPFI=(ZE-ZC1)*RHP2 DGAP 20
1252 23    IF (ABS(SNPFI).GT.0.866) GO TO 10 DGAP 21
1253 24    CSPHI=1.0 DGAP 22
1254 25    SNPFI=0.0 DGAP 23
1255 26    10 CONTINUE DGAP 24
1256 27    XEB=(XE-XC1)*CSTH+(YE-YC1)*SNTH DGAP 25
1257 28    YEB=(ZC1-ZE)*SNPHI+TRCR*CSPHI DGAP 26
1258 29    ZEB=(ZE-ZC1)*CSPHI+TRCR*SNPHI DGAP 27
1259 30    IF (IZOOM.EQ.1) GO TO 30 DGAP 28
1260 31    XIMX=-1.OE+15 DGAP 29
1261 32    ETAMX=-1.OE+15 DGAP 30
1262 33    XIMN=+1.OE+15 DGAP 31
1263 34    ETAMN=+1.OE+15 DGAP 32
1264 35    DO 20 IXR=1,23,3 DGAP 33
1265 36    XBND=GRDBN(IXR) DGAP 34
1266 37    YBND=GRDBN(IXR+1) DGAP 35
1267 38    ZBND=GRDBN(IXR+2) DGAP 36
1268 39    TRCR=(YBND-YC1)*CSTH-(XBND-XC1)*SNTH DGAP 37
1269 40    XB=(XBND-XC1)*CSTH+(YBND-YC1)*SNTH DGAP 38
1270 41    YB=(ZC1-ZBND)*SNPHI+TRCR*CSPHI DGAP 39
1271 42    ZZB=(ZBND-ZC1)*CSPHI+TRCR*SNPHI DGAP 40
1272 43    YDEN=AMAX1((YB-YEB),1.OE-6) DGAP 41
1273 44    YRAT=YEB/YDEN DGAP 42
1274 45    XIBND=XEB-(XB-XBND)*YRAT DGAP 43
1275 46    ETABND=ZEB-(ZZB-ZBND)*YRAT DGAP 44
1276 47    XIMX=AMAX1(XIMX,XIBND) DGAP 45
1277 48    XIMN=AMIN1(XIMN,XIBND) DGAP 46
1278 49    ETAMX=AMAX1(ETAMX,ETABND) DGAP 47
1279 50    ETAMN=AMIN1(ETAMN,ETABND) DGAP 48
1280 51    20 CONTINUE DGAP 49
                                         DGAP 50
                                         DGAP 51

```

```

1281 52      DXI=XIMX-XIMN          DGAP   52
1282 53      DETA=ETAMX-ETAMN        DGAP   53
1283 54      XIETAMX=AMAX1(DXI,DETA) DGAP   54
1284 55      GSZ=900.0             DGAP   55
1285 56      IF (LPR.EQ.0) GSZ=1020.0 DGAP   56
1286 57      GDRAT=GSZ/XIETAMX       DGAP   57
1287 58      30 CONTINUE           DGAP   58
1288 59 C
1289 60      RETURN               DGAP   59
1290 61      END                  DGAP   61
1291 1 *DK DRAW                DRAW   1
1292 2      SUBROUTINE DRAW        DRAW   2
1293 3 *CA SLCOM1              DRAW   3
1294 4 C
1295 5 C ** PLOT VELOCITY VECTORS, PARTICLES, ETC. ON FILM DRAW   5
1296 6 C
1297 7 C ** GRID PLOT           DRAW   7
1298 8 C
1299 9 C ** VELOCITY VECTOR AND CONSTANT PLANE FREE SURFACE PLOT DRAW   9
1300 10 C
1301 11 C***** PLOT A TWO DIMENSIONAL VECTOR AND SURFACE J=2 AND J=JM1 DRAW 10
1302 12 C
1303 13      IF (CYL.GT.0.5.AND.(ABS(Y(JM1)/X(IM1)-6.2831853).LT.0.01.OR.ABS(Y DRAW 13
1304 14 1 (JM1)/X(IM1)-3.14159265).LT.0.01)) CALL DRAW0 DRAW 14
1305 15 C
1306 16      IVEWD=1              DRAW   16
1307 17      CALL DGAP (XEA(IVEWD),YEA(IVEWD),ZEA(IVEWD),XCA(IVEWD),YCA(IVEWD) DRAW 17
1308 18 1 ,ZCA(IVEWD))           DRAW   18
1309 19      IF (NVPLTS.EQ.0) GO TO 30 DRAW 19
1310 20      DO 20 NI=1,NVPLTS        DRAW 20
1311 21      IF (IVVIEW(NI).EQ.IVEWD) GO TO 10 DRAW 21
1312 22      IVEWD=IVVIEW(NI)        DRAW 22
1313 23      CALL DGAP (XEA(IVEWD),YEA(IVEWD),ZEA(IVEWD),XCA(IVEWD),YCA(IVEWD) DRAW 23
1314 24 1 ,ZCA(IVEWD))           DRAW 24
1315 25 10 CALL VELV (IV1(NI),IV2(NI),JV1(NI),JV2(NI),KV1(NI),KV2(NI),NAV(NI) DRAW 25
1316 26 1 ,IPERV(NI))            DRAW 26
1317 27 20 CONTINUE             DRAW 27
1318 28 30 CONTINUE             DRAW 28
1319 29 C
1320 30 C ** CONTOUR PLOTS      DRAW 29
1321 31 C
1322 32      IVEWD=1              DRAW 32
1323 33      IF (NCPLTS.EQ.0) GO TO 60 DRAW 33
1324 34      DO 50 NI=1,NCPLTS        DRAW 34
1325 35      IF (ICVIEW(NI).EQ.IVEWD) GO TO 40 DRAW 35
1326 36      IVEWD=ICVIEW(NI)        DRAW 36
1327 37      CALL DGAP (XEA(IVEWD),YEA(IVEWD),ZEA(IVEWD),XCA(IVEWD),YCA(IVEWD) DRAW 37
1328 38 1 ,ZCA(IVEWD))           DRAW 38
1329 39 40 CALL CNTR (IC1(NI),IC2(NI),JC1(NI),JC2(NI),KC1(NI),KC2(NI),NAC(NI) DRAW 39
1330 40 1 ,IPERC(NI))            DRAW 40
1331 41 50 CONTINUE             DRAW 41
1332 42 60 CONTINUE             DRAW 42
1333 43 C
1334 44 C ** PERSPECTIVE SURFACE PLOTS (NO HIDDEN LINES IN THIS VERSION) DRAW 44
1335 45 C
1336 46      IVEWD=1              DRAW 46
1337 47      IF (NSPLTS.EQ.0) GO TO 90 DRAW 47
1338 48      DO 80 NI=1,NSPLTS        DRAW 48
1339 49      IF (ISVIEW(NI).EQ.IVEWD) GO TO 70 DRAW 49
1340 50      IVEWD=ISVIEW(NI)        DRAW 50
1341 51      CALL DGAP (XEA(IVEWD),YEA(IVEWD),ZEA(IVEWD),XCA(IVEWD),YCA(IVEWD) DRAW 51
1342 52 1 ,ZCA(IVEWD))           DRAW 52
1343 53 70 CALL SURFPLT (IS1(NI),IS2(NI),JS1(NI),JS2(NI),KS1(NI),KS2(NI),NAS DRAW 53
1344 54 1 (NI),IPERS(NI))         DRAW 54
1345 55 80 CONTINUE             DRAW 55
1346 56 90 CONTINUE             DRAW 56
1347 57      RETURN               DRAW 57
1348 58      END                  DRAW 58
1349 1 *DK DRF
1350 2      SUBROUTINE DRF (IFM,JFM,KFM) DRF   1
1351 3 *CA SLCOM1              DRF   2
1352 4 C
1353 5 C ** DRF - DRAW FRAME (PLANE VIEW) DRF   4
1354 6 C ** DRFP - DRAW FRAME IN PERSPECTIVE (ENTRY POINT) DRF   5
1355 7 C
1356 8 C ** SELECT CONSTANT PLANE, MPLN DRF   6
1357 9 C
1358 10      IF (IFM.NE.0) MPLN=1 DRF   7
1359 11      IF (JFM.NE.0) MPLN=2 DRF   8
1360 12      IF (KFM.NE.0) MPLN=3 DRF   9

```

1361	13	GO TO (10,20,30), MPLN	DRF	13
1362	14 C		DRF	14
1363	15 C	* * CONSTANT I PLANE	DRF	15
1364	16 C		DRF	16
1365	17	10 YLEFT=0.0	DRF	17
1366	18	YRIGHT=YBBK-YBF	DRF	18
1367	19	ZBOT=0.0	DRF	19
1368	20	ZTOP=ZBT-ZBB	DRF	20
1369	21	IXL=FIXL(MPLN)+YLEFT*XCONV(MPLN)	DRF	21
1370	22	IXR=FIXL(MPLN)+YRIGHT*XCONV(MPLN)	DRF	22
1371	23	IXB=FIYB+ZBOT*YCONV(MPLN)	DRF	23
1372	24	IXT=FIYB+ZTOP*YCONV(MPLN)	DRF	24
1373	25	CALL FRAME (IXL,IXR,IXT,IXB)	DRF	25
1374	26	CALL FRAME (IXL,IXR,IXT,IXB)	DRF	26
1375	27	GO TO 50	DRF	27
1376	28 C		DRF	28
1377	29 C	* * CONSTANT J PLANE	DRF	29
1378	30 C		DRF	30
1379	31	20 XLEFT=0.0	DRF	31
1380	32	XRIGHT=XBR-XBL	DRF	32
1381	33	ZBOT=0.0	DRF	33
1382	34	ZTOP=ZBT-ZBB	DRF	34
1383	35	IXL=FIXL(MPLN)+XLEFT*XCONV(MPLN)	DRF	35
1384	36	IXR=FIXL(MPLN)+XRIGHT*XCONV(MPLN)	DRF	36
1385	37	IXB=FIYB+ZBOT*YCONV(MPLN)	DRF	37
1386	38	IXT=FIYB+ZTOP*YCONV(MPLN)	DRF	38
1387	39	CALL FRAME (IXL,IXR,IXT,IXB)	DRF	39
1388	40	CALL FRAME (IXL,IXR,IXT,IXB)	DRF	40
1389	41	GO TO 50	DRF	41
1390	42 C		DRF	42
1391	43 C	* * CONSTANT K PLANE	DRF	43
1392	44 C		DRF	44
1393	45	30 IF (CYL.EQ.1.0) GO TO 40	DRF	45
1394	46	XLEFT=0.0	DRF	46
1395	47	XRIGHT=XBR-XBL	DRF	47
1396	48	YBOT=0.0	DRF	48
1397	49	YTOP=YBBK-YBF	DRF	49
1398	50	IXL=FIXL(MPLN)+XLEFT*XCONV(MPLN)	DRF	50
1399	51	IXR=FIXL(MPLN)+XRIGHT*XCONV(MPLN)	DRF	51
1400	52	IXB=FIYB+YBOT*YCONV(MPLN)	DRF	52
1401	53	IXT=FIYB+YTOP*YCONV(MPLN)	DRF	53
1402	54	CALL FRAME (IXL,IXR,IXT,IXB)	DRF	54
1403	55	CALL FRAME (IXL,IXR,IXT,IXB)	DRF	55
1404	56	GO TO 50	DRF	56
1405	57 C		DRF	57
1406	58 C	* * DRAW FRAME FOR CONSTANT K-PLANE IN CYLINDRICAL COORDINATES	DRF	58
1407	59 C		DRF	59
1408	60	40 CONTINUE	DRF	60
1409	61	CALL DRFCYL (MPLN)	DRF	61
1410	62 C		DRF	62
1411	63 C	* * LABEL COORDINATE AXES	DRF	63
1412	64 C		DRF	64
1413	65	50 CALL LINCNT (59)	DRF	65
1414	66	IX1=AMAX1(0.0,(FIXL(MPLN)-8))	DRF	66
1415	67	IY1=FIYB-32	DRF	67
1416	68	IX2=FIXL(MPLN)+16	DRF	68
1417	69 C		DRF	69
1418	70 C	DRAW OBSTACLE BOUNDARIES FOR JPLANE PLOT	DRF	70
1419	71 C		DRF	71
1420	72	CALL DRWOBJS (MPLN)	DRF	72
1421	73 C		DRF	73
1422	74	GO TO (60,70,80), MPLN	DRF	74
1423	75 C		DRF	75
1424	76 C	* * CONSTANT I PLANE	DRF	76
1425	77 C		DRF	77
1426	78	60 CALL PLT (IX1,IY1,57)	DRF	78
1427	79	CALL PLT (IX2,IY2,56)	DRF	79
1428	80	RETURN	DRF	80
1429	81 C		DRF	81
1430	82 C	* * CONSTANT J PLANE	DRF	82
1431	83 C		DRF	83
1432	84	70 CALL PLT (IX1,IY1,57)	DRF	84
1433	85	CALL PLT (IX2,IY2,55)	DRF	85
1434	86	RETURN	DRF	86
1435	87 C		DRF	87
1436	88 C	* * CONSTANT K PLANE	DRF	88
1437	89 C		DRF	89
1438	90	80 CONTINUE	DRF	90
1439	91	RETURN	DRF	91
1440	92 C		DRF	92

1441	93	END	DRF	93
1442	1	*DK DRFCYL	DRFCYL	1
1443	2	SUBROUTINE DRFCYL (MPLN)	DRFCYL	2
1444	3	*CA SLCOM1	DRFCYL	3
1445	4	C	DRFCYL	4
1446	5	C * * DRAW FRAME FOR CONSTANT K-PLANE IN CYLINDRICAL COORDINATES	DRFCYL	5
1447	6	C	DRFCYL	6
1448	7	X1=X(1)	DRFCYL	7
1449	8	X2=X(IM1)	DRFCYL	8
1450	9	CRX=CYL/X2	DRFCYL	9
1451	10	XLEFT=XBLC	DRFCYL	10
1452	11	YLEFT=YBFC	DRFCYL	11
1453	12	JPL=JM1	DRFCYL	12
1454	13	DO 20 J=2,JPL	DRFCYL	13
1455	14	TH1=CRX*Y(J-1)	DRFCYL	14
1456	15	TH2=CRX*Y(J)	DRFCYL	15
1457	16	STH1=SIN(TH1)	DRFCYL	16
1458	17	CTH1=COS(TH1)	DRFCYL	17
1459	18	STH2=SIN(TH2)	DRFCYL	18
1460	19	CTH2=COS(TH2)	DRFCYL	19
1461	20	XX1=X1*CTH1	DRFCYL	20
1462	21	YY1=X1*STH1+Y(J-1)*(1.0-CYL)	DRFCYL	21
1463	22	XX2=X1*CTH2	DRFCYL	22
1464	23	YY2=X1*STH2+Y(J)*(1.0-CYL)	DRFCYL	23
1465	24	XRIGHT1=XX1-XLEFT	DRFCYL	24
1466	25	YRIGHT1=YY1-YLEFT	DRFCYL	25
1467	26	IX1=FIXL(MPLN)+XRIGHT1*XCONV(MPLN)	DRFCYL	26
1468	27	IY1=FIYB+YRIGHT1*YCONV(MPLN)	DRFCYL	27
1469	28	XRIGHT2=XX2-XLEFT	DRFCYL	28
1470	29	YRIGHT2=YY2-YLEFT	DRFCYL	29
1471	30	IX2=FIXL(MPLN)+XRIGHT2*XCONV(MPLN)	DRFCYL	30
1472	31	IY2=FIYB+YRIGHT2*YCONV(MPLN)	DRFCYL	31
1473	32	CALL DRV (IX1,IY1,IX2,IY2)	DRFCYL	32
1474	33	DTH21=(TH2-TH1)/10.0	DRFCYL	33
1475	34	THO=TH1	DRFCYL	34
1476	35	DO 10 L=1,10	DRFCYL	35
1477	36	THO1=THO+(L-1)*DTH21	DRFCYL	36
1478	37	THO2=THO+L*DTH21	DRFCYL	37
1479	38	CTH1=COS(THO1)	DRFCYL	38
1480	39	CTH2=COS(THO2)	DRFCYL	39
1481	40	STH1=SIN(THO1)	DRFCYL	40
1482	41	STH2=SIN(THO2)	DRFCYL	41
1483	42	XX1=X2*CTH1	DRFCYL	42
1484	43	XX2=X2*CTH2	DRFCYL	43
1485	44	YY1=X2*STH1+Y(J-1)*(1.0-CYL)	DRFCYL	44
1486	45	YY2=X2*STH2+Y(J)*(1.0-CYL)	DRFCYL	45
1487	46	XRIGHT1=XX1-XLEFT	DRFCYL	46
1488	47	YRIGHT1=YY1-YLEFT	DRFCYL	47
1489	48	IX1=FIXL(MPLN)+XRIGHT1*XCONV(MPLN)	DRFCYL	48
1490	49	IY1=FIYB+YRIGHT1*YCONV(MPLN)	DRFCYL	49
1491	50	XRIGHT2=XX2-XLEFT	DRFCYL	50
1492	51	YRIGHT2=YY2-YLEFT	DRFCYL	51
1493	52	IX2=FIXL(MPLN)+XRIGHT2*XCONV(MPLN)	DRFCYL	52
1494	53	IY2=FIYB+YRIGHT2*YCONV(MPLN)	DRFCYL	53
1495	54	10 CALL DRV (IX1,IY1,IX2,IY2)	DRFCYL	54
1496	55	20 CONTINUE	DRFCYL	55
1497	56	IF (ABS(Y(JPL))*CRX-6.28327853).LT.0.0001) RETURN	DRFCYL	56
1498	57	Y1=Y(1)	DRFCYL	57
1499	58	Y2=Y(JM1)	DRFCYL	58
1500	59	TH1=CRX*Y1	DRFCYL	59
1501	60	TH2=CRX*Y2	DRFCYL	60
1502	61	STH1=SIN(TH1)	DRFCYL	61
1503	62	CTH1=COS(TH1)	DRFCYL	62
1504	63	STH2=SIN(TH2)	DRFCYL	63
1505	64	CTH2=COS(TH2)	DRFCYL	64
1506	65	XX1=X1*CTH1	DRFCYL	65
1507	66	YY1=X1*STH1+Y1*(1.0-CYL)	DRFCYL	66
1508	67	XX2=X2*CTH1	DRFCYL	67
1509	68	YY2=X2*STH1+Y1*(1.0-CYL)	DRFCYL	68
1510	69	XRIGHT1=XX1-XLEFT	DRFCYL	69
1511	70	YRIGHT1=YY1-YLEFT	DRFCYL	70
1512	71	IX1=FIXL(MPLN)+XRIGHT1*XCONV(MPLN)	DRFCYL	71
1513	72	IY1=FIYB+YRIGHT1*YCONV(MPLN)	DRFCYL	72
1514	73	XRIGHT2=XX2-XLEFT	DRFCYL	73
1515	74	YRIGHT2=YY2-YLEFT	DRFCYL	74
1516	75	IX2=FIXL(MPLN)+XRIGHT2*XCONV(MPLN)	DRFCYL	75
1517	76	IY2=FIYB+YRIGHT2*YCONV(MPLN)	DRFCYL	76
1518	77	CALL DRV (IX1,IY1,IX2,IY2)	DRFCYL	77
1519	78	XX1=X1*CTH2	DRFCYL	78
1520	79	YY1=X1*STH2+Y2*(1.0-CYL)	DRFCYL	79

1521	80	XX2=X2*CTH2	DRFCYL	80
1522	81	YY2=X2*STH2+Y2*(1.0-CYL)	DRFCYL	81
1523	82	XRIGHT1=XX1-XLEFT	DRFCYL	82
1524	83	YRIGHT1=YY1-YLEFT	DRFCYL	83
1525	84	IX1=FIXL(MPLN)+XRIGHT1*XCONV(MPLN)	DRFCYL	84
1526	85	IY1=FIYB+YRIGHT1*YCONV(MPLN)	DRFCYL	85
1527	86	XRIGHT2=XX2-XLEFT	DRFCYL	86
1528	87	YRIGHT2=YY2-YLEFT	DRFCYL	87
1529	88	IX2=FIXL(MPLN)+XRIGHT2*XCONV(MPLN)	DRFCYL	88
1530	89	IY2=FIYB+YRIGHT2*YCONV(MPLN)	DRFCYL	89
1531	90	CALL DRV (IX1,IY1,IX2,IY2)	DRFCYL	90
1532	91 C	RETURN	DRFCYL	91
1533	92	END	DRFP	92
1534	93	1 *DK DRFP	DRFP	1
1535	2	SUBROUTINE DRFP	DRFP	2
1536	3 *CA SLCOM1		DRFP	3
1537	4 C		DRFP	4
1538	5 C	* * DRAW FRAME IN PERSPECTIVE	DRFP	5
1539	6 C		DRFP	6
1541	7	X1=X(1)	DRFP	7
1542	8	X2=X(IM1)	DRFP	8
1543	9	Y1=Y(1)	DRFP	9
1544	10	Y2=Y(JM1)	DRFP	10
1545	11	Z1=Z(1)	DRFP	11
1546	12	Z2=Z(KM1)	DRFP	12
1547	13	CRX=CYL/X2	DRFP	13
1548	14	TH1=CRX*Y1	DRFP	14
1549	15	TH2=CRX*Y2	DRFP	15
1550	16	CTH1=COS(TH1)	DRFP	16
1551	17	CTH2=COS(TH2)	DRFP	17
1552	18	STH1=SIN(TH1)	DRFP	18
1553	19	STH2=SIN(TH2)	DRFP	19
1554	20	XX1=X1*CTH1	DRFP	20
1555	21	YY1=X1*STH1+Y1*(1.0-CYL)	DRFP	21
1556	22	XX2=X2*CTH1	DRFP	22
1557	23	YY2=X2*STH1+Y1*(1.0-CYL)	DRFP	23
1558	24	CALL PCNV (IXI1,IETA1,XX1,YY1,Z1)	DRFP	24
1559	25	CALL PCNV (IXI2,IETA2,XX2,YY2,Z1)	DRFP	25
1560	26	CALL DRVEC (IXI1,IETA1,IXI2,IETA2)	DRFP	26
1561	27	CALL PCNV (IXI3,IETA3,XX1,YY1,Z2)	DRFP	27
1562	28	CALL PCNV (IXI4,IETA4,XX2,YY2,Z2)	DRFP	28
1563	29	CALL DRVEC (IXI3,IETA3,IXI4,IETA4)	DRFP	29
1564	30	CALL PCNV (IXI5,IETA5,XX1,YY1,Z1)	DRFP	30
1565	31	CALL PCNV (IXI6,IETA6,XX1,YY1,Z2)	DRFP	31
1566	32	CALL DRVEC (IXI5,IETA5,IXI6,IETA6)	DRFP	32
1567	33	CALL PCNV (IXI7,IETA7,XX2,YY2,Z1)	DRFP	33
1568	34	CALL PCNV (IXI8,IETA8,XX2,YY2,Z2)	DRFP	34
1569	35	CALL DRVEC (IXI7,IETA7,IXI8,IETA8)	DRFP	35
1570	36	XX1=X1*CTH2	DRFP	36
1571	37	YY1=X1*STH2+Y2*(1.0-CYL)	DRFP	37
1572	38	XX2=X2*CTH2	DRFP	38
1573	39	YY2=X2*STH2+Y2*(1.0-CYL)	DRFP	39
1574	40	CALL PCNV (IXI1,IETA1,XX1,YY1,Z1)	DRFP	40
1575	41	CALL PCNV (IXI2,IETA2,XX2,YY2,Z1)	DRFP	41
1576	42	CALL DRVEC (IXI1,IETA1,IXI2,IETA2)	DRFP	42
1577	43	CALL PCNV (IXI3,IETA3,XX1,YY1,Z2)	DRFP	43
1578	44	CALL PCNV (IXI4,IETA4,XX2,YY2,Z2)	DRFP	44
1579	45	CALL DRVEC (IXI3,IETA3,IXI4,IETA4)	DRFP	45
1580	46	CALL PCNV (IXI5,IETA5,XX1,YY1,Z1)	DRFP	46
1581	47	CALL PCNV (IXI6,IETA6,XX1,YY1,Z2)	DRFP	47
1582	48	CALL DRVEC (IXI5,IETA5,IXI6,IETA6)	DRFP	48
1583	49	CALL PCNV (IXI7,IETA7,XX2,YY2,Z1)	DRFP	49
1584	50	CALL PCNV (IXI8,IETA8,XX2,YY2,Z2)	DRFP	50
1585	51	CALL DRVEC (IXI7,IETA7,IXI8,IETA8)	DRFP	51
1586	52	JPL=JM1	DRFP	52
1587	53	DO 10 J=2,JPL	DRFP	53
1588	54	TH1=CRX*Y(J-1)	DRFP	54
1589	55	TH2=CRX*Y(J)	DRFP	55
1590	56	STH1=SIN(TH1)	DRFP	56
1591	57	CTH1=COS(TH1)	DRFP	57
1592	58	STH2=SIN(TH2)	DRFP	58
1593	59	CTH2=COS(TH2)	DRFP	59
1594	60	XX1=X1*CTH1	DRFP	60
1595	61	YY1=X1*STH1+Y(J-1)*(1.0-CYL)	DRFP	61
1596	62	XX2=X1*CTH2	DRFP	62
1597	63	YY2=X1*STH2+Y(J)*(1.0-CYL)	DRFP	63
1598	64	CALL PCNV (IXI1,IETA1,XX1,YY1,Z1)	DRFP	64
1599	65	CALL PCNV (IXI2,IETA2,XX2,YY2,Z1)	DRFP	65
1600	66	CALL DRVEC (IXI1,IETA1,IXI2,IETA2)	DRFP	66

```

1601 67      CALL PCNV (IXI3,IETA3,XX1,YY1,Z2)          DRFP   67
1602 68      CALL PCNV (IXI4,IETA4,XX2,YY2,Z2)          DRFP   68
1603 69      CALL DRVEC (IXI3,IETA3,IXI4,IETA4)        DRFP   69
1604 70      XX1=X2*CTH1                                DRFP   70
1605 71      XX2=X2*CTH2                                DRFP   71
1606 72      YY1=X2*STH1+Y(J-1)*(1.0-CYL)             DRFP   72
1607 73      YY2=X2*STH2+Y(J)*(1.0-CYL)               DRFP   73
1608 74      CALL PCNV (IXI1,IETA1,XX1,YY1,Z1)          DRFP   74
1609 75      CALL PCNV (IXI2,IETA2,XX2,YY2,Z1)          DRFP   75
1610 76      CALL DRVEC (IXI1,IETA1,IXI2,IETA2)        DRFP   76
1611 77      CALL PCNV (IXI3,IETA3,XX1,YY1,Z2)          DRFP   77
1612 78      CALL PCNV (IXI4,IETA4,XX2,YY2,Z2)          DRFP   78
1613 79      CALL DRVEC (IXI3,IETA3,IXI4,IETA4)        DRFP   79
1614 80      10 CONTINUE                               DRFP   80
1615 81 C
1616 82 C * * DRAW OBSTACLE IN PERSPECTIVE           DRFP   81
1617 83 C
1618 84 C      CALL DROBSP                         DRFP   84
1619 85 C
1620 86      RETURN                                 DRFP   86
1621 87      END                                    DRFP   87
1622 1 *DK DROBSP
1623 2      SUBROUTINE DROBSP
1624 3 *CA SLCOM1
1625 4 C
1626 5 C * * DRAW OBSTACLES IN PERSPECTIVE          DRFBSP  5
1627 6 C
1628 7      IPL=IM1                                DRFBSP  6
1629 8      JPL=JM1                                DRFBSP  7
1630 9      KPL=KM1                                DRFBSP  8
1631 10 C
1632 11      DO 40 I=1,IPL                         DRFBSP 10
1633 12      DO 40 J=1,JPL                         DRFBSP 11
1634 13      DO 40 K=1,KPL                         DRFBSP 12
1635 14      ACS=1.0                                DRFBSP 13
1636 15      ARS=1.0                                DRFBSP 14
1637 16      ABKS=1.0                               DRFBSP 15
1638 17      ATS=1.0                                DRFBSP 16
1639 18      IF (BETA(IJK).LT.0.0) ACS=0.0          DRFBSP 17
1640 19      IF (BETA(IPJK).LT.0.0) ARS=0.0          DRFBSP 18
1641 20      IF (BETA(IJPK).LT.0.0) ABKS=0.0         DRFBSP 19
1642 21      IF (BETA(IJKP).LT.0.0) ATS=0.0          DRFBSP 20
1643 22 C
1644 23      IPLX=0                                 DRFBSP 21
1645 24      IF (ACS.EQ.0.0) IPLX=1                  DRFBSP 22
1646 25      IF (BETA(IJPKP).LT.0.0) IPLX=IPLX+1    DRFBSP 23
1647 26      IPLD=0                                 DRFBSP 24
1648 27      IF (ATS.EQ.0.0) IPLD=1                  DRFBSP 25
1649 28      IF (ABKS.EQ.0.0) IPLD=IPLD+1            DRFBSP 26
1650 29      IPLX=IPLX-IPLD                         DRFBSP 27
1651 30 C
1652 31      IPLY=0                                 DRFBSP 28
1653 32      IF (ACS.EQ.0.0) IPLY=1                  DRFBSP 29
1654 33      IF (BETA(IPJKP).LT.0.0) IPLY=IPLY+1    DRFBSP 30
1655 34      IPLD=0                                 DRFBSP 31
1656 35      IF (ARS.EQ.0.0) IPLD=1                  DRFBSP 32
1657 36      IF (ATS.EQ.0.0) IPLD=IPLD+1            DRFBSP 33
1658 37      IPLY=IPLY-IPLD                         DRFBSP 34
1659 38 C
1660 39      IPLZ=0                                 DRFBSP 35
1661 40      IF (ACS.EQ.0.0) IPLZ=1                  DRFBSP 36
1662 41      IF (BETA(IPUPK).LT.0.0) IPLZ=IPLZ+1    DRFBSP 37
1663 42      IPLD=0                                 DRFBSP 38
1664 43      IF (ARS.EQ.0.0) IPLD=1                  DRFBSP 39
1665 44      IF (ABKS.EQ.0.0) IPLD=IPLD+1            DRFBSP 40
1666 45      IPLZ=IPLZ-IPLD                         DRFBSP 41
1667 46 C
1668 47      IF (IPLX.EQ.0.AND.IPLY.EQ.0.AND.IPLZ.EQ.0) GO TO 40 DRFBSP 42
1669 48      XB1=X(I)                                DRFBSP 43
1670 49      YB1=Y(J)                                DRFBSP 44
1671 50      ZZ1=Z(K)                                DRFBSP 45
1672 51      CRX=CYL/X(IM1)                         DRFBSP 46
1673 52      TH1=CRX*YB1                            DRFBSP 47
1674 53      CTH1=COS(TH1)                           DRFBSP 48
1675 54      STH1=SIN(TH1)                           DRFBSP 49
1676 55      XX1=XB1*CTH1                           DRFBSP 50
1677 56      YY1=XB1*STH1+YB1*(1.0-CYL)             DRFBSP 51
1678 57      CALL PCNV (IXI1,IETA1,XX1,YY1,ZZ1)       DRFBSP 52
1679 58      IRET=1                                 DRFBSP 53
1680 59      IF (IPLX.EQ.0.OR.I.EQ.1) GO TO 10      DRFBSP 54

```

1681	60	XX2=X(I-1)*CTH1	DROBSP	60
1682	61	YY2=X(I-1)*STH1+YB1*(1.0-CYL)	DROBSP	61
1683	62	ZZ2=ZZ1	DROBSP	62
1684	63	GO TO 30	DROBSP	63
1685	64	10 IRET=2	DROBSP	64
1686	65	IF (IPLZ.EQ.0.OR.K.EQ.1.) GO TO 20	DROBSP	65
1687	66	XX2=XX1	DROBSP	66
1688	67	YY2=YY1	DROBSP	67
1689	68	ZZ2=Z(K-1)	DROBSP	68
1690	69	GO TO 30	DROBSP	69
1691	70	20 IRET=3	DROBSP	70
1692	71	IF (IPLY.EQ.0.OR.J.EQ.1.) GO TO 40	DROBSP	71
1693	72	YB2=Y(J-1)	DROBSP	72
1694	73	TH2=CRX*YB2	DROBSP	73
1695	74	CTH2=COS(TH2)	DROBSP	74
1696	75	STH2=SIN(TH2)	DROBSP	75
1697	76	XX2=XB1*CTH2	DROBSP	76
1698	77	YY2=XB1*STH2+YB2*(1.0-CYL)	DROBSP	77
1699	78	ZZ2=ZZ1	DROBSP	78
1700	79	30 CALL PCNV (IXI2,IETA2,XX2,YY2,ZZ2)	DROBSP	79
1701	80	CALL DRVEC (IXI1,IETA1,IXI2,IETA2)	DROBSP	80
1702	81	GO TO (10,20,40), IRET	DROBSP	81
1703	82	40 CONTINUE	DROBSP	82
1704	83 C	RETURN	DROBSP	83
1705	84	END	DROBSP	84
1706	85	1 *DK DRWOBS	DRWOBS	1
1707	2	SUBROUTINE DRWOBS (MPLN)	DRWOBS	2
1709	3	*CA SLCOM1	DRWOBS	3
1710	4 C	*** DRAW AROUND ALL OBSTACLES	DRWOBS	4
1711	5 C	TPI=2.0*PI	DRWOBS	5
1712	6 C	GO TO (200,10,200), MPLN	DRWOBS	6
1716	10 C	10 CONTINUE	DRWOBS	10
1717	11	J=2	DRWOBS	11
1718	12	DO 190 I=2,IM1	DRWOBS	12
1719	13	ATR=1.0-CYL+CYL*(1.0-EM6)	DRWOBS	13
1720	14	ATL=1.0-CYL+CYL*(1.0-EM6)	DRWOBS	14
1721	15	ATC=1.0-CYL+CYL*(1.0-EM6)	DRWOBS	15
1722	16	DO 190 K=2,KM1	DRWOBS	16
1723	17	CALL IJKAJCT	DRWOBS	17
1724	18	IF (AC(IJK).LT.EM6) GO TO 190	DRWOBS	18
1725	19	AFR=1.0	DRWOBS	19
1726	20	AFT=1.0	DRWOBS	20
1727	21	AFL=1.0	DRWOBS	21
1729	23	AFB=1.0	DRWOBS	23
1730	24	IF (AR(IJK).LT.ATR) AFR=AR(IJK)/ATR	DRWOBS	24
1731	25	IF (AT(IJK).LT.ATC) AFT=AT(IJK)/ATC	DRWOBS	25
1732	26	IF (AR(IMJK).LT.ATL) AFL=AR(IMJK)/ATL	DRWOBS	26
1733	27	IF (AT(IJKM).LT.ATC) AFB=AT(IJKM)/ATC	DRWOBS	27
1734	28	IF (AC(IJK).GE.ATC) GO TO 130	DRWOBS	28
1735	29	IF (I.EQ.2) AFL=AFR-EM6	DRWOBS	29
1736	30	IF (I.EQ.IM1) AFR=AFL-EM6	DRWOBS	30
1737	31	IF (K.EQ.2) AFB=AFT-EM6	DRWOBS	31
1738	32	IF (K.EQ.KM1) AFT=AFB-EM6	DRWOBS	32
1739	33	IF ((AFT+AFB).LT.EM6.OR.(AFL+AFR).LT.EM6) GO TO 190	DRWOBS	33
1740	34	M=1	DRWOBS	34
1741	35	AMN=AFB+AFR	DRWOBS	35
1742	36	IF ((AFR+AFT).GT.AMN) GO TO 20	DRWOBS	36
1743	37	M=2	DRWOBS	37
1744	38	AMN=AFT+AFT	DRWOBS	38
1745	39	20 IF ((AFT+AFL).GT.AMN) GO TO 30	DRWOBS	39
1746	40	M=3	DRWOBS	40
1747	41	AMN=AFT+AFL	DRWOBS	41
1748	42	30 IF ((AFL+AFB).GT.AMN) GO TO 40	DRWOBS	42
1749	43	M=4	DRWOBS	43
1750	44	40 GO TO (50,70,90,110), M	DRWOBS	44
1751	45	50 X1=X(I-1)+AFT*DELX(I)	DRWOBS	45
1752	46	Y1=Z(K)	DRWOBS	46
1753	47	IF (AFT.LT.1.0) GO TO 60	DRWOBS	47
1754	48	Y1=Y1-AFR*DELZ(K)	DRWOBS	48
1755	49	60 X2=X(I-1)	DRWOBS	49
1756	50	Y2=Z(K)-AFL*DELZ(K)	DRWOBS	50
1757	51	IF (AFL.LT.1.0) GO TO 170	DRWOBS	51
1758	52	X2=X2+AFB*DELX(I)	DRWOBS	52
1759	53	GO TO 170	DRWOBS	53
1760	54	70 X1=X(I-1)	DRWOBS	54

```

1761 55      Y1=Z(K-1)+AFL*DELZ(K)          DRW0BS   55
1762 56      IF (AFL.LT.1.0) GO TO 80        DRW0BS   56
1763 57      X1=X1+AFT*DELX(I)             DRW0BS   57
1764 58      80 X2=X(I-1)+AFB*DELX(I)       DRW0BS   58
1765 59      Y2=Z(K-1)                      DRW0BS   59
1766 60      IF (AFB.LT.1.0) GO TO 170       DRW0BS   60
1767 61      Y2=Y2+AFR*DELZ(K)             DRW0BS   61
1768 62      GO TO 170                      DRW0BS   62
1769 63      90 X1=X(I)-AFB*DELX(I)       DRW0BS   63
1770 64      Y1=Z(K-1)                      DRW0BS   64
1771 65      IF (AFB.LT.1.0) GO TO 100       DRW0BS   65
1772 66      Y1=Y1+AFL*DELZ(K)             DRW0BS   66
1773 67      100 X2=X(I)                   DRW0BS   67
1774 68      Y2=Z(K-1)+AFR*DELZ(K)       DRW0BS   68
1775 69      IF (AFR.LT.1.0) GO TO 170       DRW0BS   69
1776 70      X2=X2-AFT*DELX(I)             DRW0BS   70
1777 71      GO TO 170                      DRW0BS   71
1778 72      110 X1=X(I)                   DRW0BS   72
1779 73      Y1=Z(K)-AFR*DELZ(K)           DRW0BS   73
1780 74      IF (AFR.LT.1.0) GO TO 120       DRW0BS   74
1781 75      X1=X1-AFB*DELX(I)             DRW0BS   75
1782 76      120 X2=X(I)-AFT*DELX(I)       DRW0BS   76
1783 77      Y2=Z(K)                      DRW0BS   77
1784 78      IF (AFT.LT.1.0) GO TO 170       DRW0BS   78
1785 79      Y2=Y2-AFL*DELZ(K)             DRW0BS   79
1786 80      GO TO 170                      DRW0BS   80
1787 81      130 IF (AFR.GT.EM6) GO TO 140    DRW0BS   81
1788 82      X1=X(I)                      DRW0BS   82
1789 83      Y1=Z(K-1)                      DRW0BS   83
1790 84      X2=X1                      DRW0BS   84
1791 85      Y2=Z(K)                      DRW0BS   85
1792 86      ASSIGN 140 TO KR1            DRW0BS   86
1793 87      GO TO 180                      DRW0BS   87
1794 88      140 IF (AFT.GT.EM6) GO TO 150    DRW0BS   88
1795 89      X1=X(I-1)                    DRW0BS   89
1796 90      Y1=Z(K)                      DRW0BS   90
1797 91      X2=X(I)                      DRW0BS   91
1798 92      Y2=Y1                      DRW0BS   92
1799 93      ASSIGN 150 TO KR1            DRW0BS   93
1800 94      GO TO 180                      DRW0BS   94
1801 95      150 IF (AFL.GT.EM6) GO TO 160    DRW0BS   95
1802 96      X1=X(I-1)                    DRW0BS   96
1803 97      Y1=Z(K)                      DRW0BS   97
1804 98      X2=X1                      DRW0BS   98
1805 99      Y2=Z(K-1)                    DRW0BS   99
1806 100     ASSIGN 160 TO KR1            DRW0BS  100
1807 101     GO TO 180                      DRW0BS  101
1808 102     160 IF (AFB.GT.EM6) GO TO 190    DRW0BS  102
1809 103     X1=X(I-1)                    DRW0BS  103
1810 104     Y1=Z(K-1)                    DRW0BS  104
1811 105     X2=X(I)                      DRW0BS  105
1812 106     Y2=Y1                      DRW0BS  106
1813 107     ASSIGN 170 TO KR1            DRW0BS  107
1814 108     GO TO 180                      DRW0BS  108
1815 109     170 ASSIGN 190 TO KR1            DRW0BS  109
1816 110     180 IX1=FIXL(MPLN)+X1*XCONV(MPLN) DRW0BS  110
1817 111     IX2=FIXL(MPLN)+X2*XCONV(MPLN)   DRW0BS  111
1818 112     IY1=FIYB+Y1*YCONV(MPLN)         DRW0BS  112
1819 113     IY2=FIYB+Y2*YCONV(MPLN)         DRW0BS  113
1820 114     CALL DRV (IX1,IY1,IX2,IY2)       DRW0BS  114
1821 115     GO TO KR1. (140,150,160,170,190) DRW0BS  115
1822 116     190 CONTINUE                  DRW0BS  116
1823 117     200 CONTINUE                  DRW0BS  117
1824 118     RETURN                      DRW0BS  118
1825 119     END                         DRW0BS  119
1826 1      *DK DRVEC                  DRVEC   1
1827 2      SUBROUTINE DRVEC (IXI1,IETA1,IXI2,IETA2) DRVEC   2
1828 3      *CA SLCOM1                  DRVEC   3
1829 4      C                           DRVEC   4
1830 5      C * * AFTER CLIPPING,DRAW VECTOR (A PART OF ZOOM FEATURE) DRVEC   5
1831 6      C                           DRVEC   6
1832 7      C * * WORKING IN FLOATING POINT NUMBERS DRVEC   7
1833 8      C                           DRVEC   8
1834 9      IF (ICLIP.EQ.0) GO TO 90        DRVEC   9
1835 10     IF ((IXI1+IETA1).EQ.0.OR.(IXI2+IETA2).EQ.0) GO TO 100    DRVEC  10
1836 11     XIL=123.0                     DRVEC  11
1837 12     XIR=1023.0                     DRVEC  12
1838 13     IF (LPR.EQ.0) XIL=2.0          DRVEC  13
1839 14     IF (LPR.EQ.0) XIR=1022.0        DRVEC  14
1840 15     ETAT=1.0                      DRVEC  15

```

```

1841 16      ETAB=901.0
1842 17      IF (LPR.EQ.0) ESTAT=2.0
1843 18      IF (LPR.EQ.0) ETAB=1022.0
1844 19      CX1=IXI1
1845 20      X11=CX1
1846 21      CX2=IXI2
1847 22      X22=CX2
1848 23      CY1=IETA1
1849 24      Y11=CY1
1850 25      CY2=IETA2
1851 26      Y22=CY2
1852 27      DX=X22-X11
1853 28      DY=Y22-Y11
1854 29      IF (ABS(DX).LT.1.0E-3) GO TO 40
1855 30      SM=DY/DX
1856 31      IF (CX1.GE.XIL) GO TO 10
1857 32      CY1=Y11+SM*(XIL-X11)
1858 33      CX1=XIL
1859 34      10 IF (CX1.LE.XIR) GO TO 20
1860 35      CY1=Y11+SM*(XIR-X11)
1861 36      CX1=XIR
1862 37      20 IF (CX2.GE.XIL) GO TO 30
1863 38      CY2=Y11+SM*(XIL-X11)
1864 39      CX2=XIL
1865 40      30 IF (CX2.LE.XIR) GO TO 40
1866 41      CY2=Y11+SM*(XIR-X11)
1867 42      CX2=XIR
1868 43      40 IF (CX1.LT.XIL.OR.CX1.GT.XIR) GO TO 100
1869 44      IF (ABS(DY).LT.1.0E-3) GO TO 80
1870 45      SM=DX/DY
1871 46      IF (CY1.LE.ETAB) GO TO 50
1872 47      CX1=X11+SM*(ETAB-Y11)
1873 48      CY1=ETAB
1874 49      50 IF (CY1.GE.ETAT) GO TO 60
1875 50      CX1=X11+SM*(ETAT-Y11)
1876 51      CY1=ETAT
1877 52      60 IF (CY2.LE.ETAB) GO TO 70
1878 53      CX2=X11+SM*(ETAB-Y11)
1879 54      CY2=ETAB
1880 55      70 IF (CY2.GE.ETAT) GO TO 80
1881 56      CX2=X11+SM*(ETAT-Y11)
1882 57      CY2=ETAT
1883 58      80 CONTINUE
1884 59      IF (CY1.LT.ETAT.OR.CY1.GT.ETAB) GO TO 100
1885 60      DS=ABS(CX2-CX1)+ABS(CY2-CY1)
1886 61      IF (DS.LT.1.0E-3) GO TO 100
1887 62      IXI1=CX1
1888 63      IXI2=CX2
1889 64      IETA1=CY1
1890 65      IETA2=CY2
1891 66      90 CALL DRV (IXI1,IETA1,IXI2,IETA2)
1892 67      100 RETURN
1893 68      END
1894 1 *DK EQUIB
1895 2      SUBROUTINE EQUIB (Y,Z,NX1,BOND,CANGLE,CYL)
1896 3      DIMENSION Y(1), Z(1)
1897 4      DATA PI /3.141592654/, EPS /1.E-04/
1898 5 C +++
1899 6 C +++ USEFUL INTERMEDIATE QUANTITIES
1900 7 C +++
1901 8      RANGLE=CANGLE
1902 9      COSTST=COS(RANGLE)
1903 10     COSCA=(1.+CYL)*COS(RANGLE)
1904 11     OMCYL=1.-CYL
1905 12     NX=NX1-1
1906 13     DR=1./FLOAT(NX)
1907 14     ITER=0
1908 15 C +++
1909 16 C +++ SET INITIAL GUESS FOR Y(1)
1910 17 C +++
1911 18     Y(1)=0.
1912 19     IF (ABS(BOND).GT.0.) Y(1)=-COSCA/(2.*ABS(BOND))
1913 20 C +++
1914 21 C +++ NUMERICAL INTEGRATION
1915 22 C +++
1916 23     10 Z(1)=0.
1917 24     ITER=ITER+1
1918 25     Z(NX1+1)=0.
1919 26     WRITE (2,100) Y(1)
1920 27     DD 20 J=2,NX1

```

```

1921 28      RJM=DR*FLOAT(J-2)*CYL+OMCYL          EQUIB   28
1922 29      RJ=DR*FLOAT(J-1)*CYL+OMCYL          EQUIB   29
1923 30      RJH=0.5*(RJM+RJ)                   EQUIB   30
1924 31      Z(J)=Z(J-1)*RJM/RJ+DR*(RJM/RJ)*(COSCA-BOND*(Y(J-1)+0.5*DR*Z(J-1)
1925 32      1/SQRT(1.-Z(J-1)**2)))           EQUIB   31
1926 33      IF (Z(J).GE.1.) GO TO 30            EQUIB   32
1927 34      SLOPE=Z(J)/SQRT(1.-Z(J)**2)        EQUIB   33
1928 35      SLOPF=Z(J-1)/SQRT(1.-Z(J-1)**2)    EQUIB   34
1929 36      Z(NX1+J)=(SLOPE-SLOPF)/(DR*(1.+SLOPE**2)**1.5)  EQUIB   35
1930 37      Y(J)=Y(J-1)+0.5*DR*(Z(J)/SQRT(1.-Z(J)**2)+Z(J-1)/SQRT(1.-Z(J-1)**2
1931 38      1))                                EQUIB   36
1932 39      20 CONTINUE                         EQUIB   37
1933 40      GO TO 40                            EQUIB   38
1934 41      30 CONTINUE                         EQUIB   39
1935 42      WRITE (2,110) J,Z(J)                 EQUIB   40
1936 43      Y(1)=Y(1)*1.05                      EQUIB   41
1937 44      WRITE (2,120) Z(NX1)                EQUIB   42
1938 45      IF (ITER.GT.400) GO TO 60          EQUIB   43
1939 46      GO TO 10                            EQUIB   44
1940 47 C +++
1941 48 C +++ CHECK CONSTRAINT AND CONVERGENCE
1942 49 C +++
1943 50      40 CONTINUE                         EQUIB   45
1944 51      WRITE (2,130)                      EQUIB   46
1945 52      YSUM=0.5*DR*Y(NX1)                EQUIB   47
1946 53      DO 50 J=2,NX                      EQUIB   48
1947 54      YSUM=YSUM+(FLOAT(J-1)*DR*CYL+OMCYL)*DR*Y(J)  EQUIB   49
1948 55      50 CONTINUE                         EQUIB   50
1949 56      WRITE (2,140) YSUM,Y(1),Z(NX1),COSTST  EQUIB   51
1950 57      IF (ABS(Z(NX1)-COS(RANGLE)).LE.EPS.AND.BOND.NE.O.) GO TO 70  EQUIB   52
1951 58      IF (ABS(YSUM).LT.EPS.AND.BOND.EQ.O.) GO TO 70  EQUIB   53
1952 59      Y(1)=Y(1)-YSUM/(OMCYL+CYL*2.)  EQUIB   54
1953 60      IF (ITER.GT.400) GO TO 60          EQUIB   55
1954 61      GO TO 10                            EQUIB   56
1955 62 C +++
1956 63 C +++ EXIT IF CONVERGENCE FAILS
1957 64 C +++
1958 65      60 CONTINUE                         EQUIB   57
1959 66      WRITE (59,150)                      EQUIB   58
1960 67      WRITE (12,150)                      EQUIB   59
1961 68      WRITE (9,150)                       EQUIB   60
1962 69      CALL EXIT                          EQUIB   61
1963 70      70 CONTINUE                         EQUIB   62
1964 71      WRITE (2,80) (I,Y(I),Z(I),Z(NX1+I),I=1,NX1)  EQUIB   63
1965 72      WRITE (59,90)                      EQUIB   64
1966 73      WRITE (2,90)                       EQUIB   65
1967 74      RETURN                           EQUIB   66
1968 75 C
1969 76      80 FORMAT (//3X,1HI,BX,1HY,11X,1HZ,9X,3HKXY/(1X,I3,1P3E12.4))  EQUIB   67
1970 77      90 FORMAT (15H FINISHED EQUIB)        EQUIB   68
1971 78      100 FORMAT (1X,17HSTEP B WITH Y(1)=,1PE14.6)  EQUIB   69
1972 79      110 FORMAT (1X,9HZ TOO BIG,I5,1PE12.4)    EQUIB   70
1973 80      120 FORMAT (1X,17HGO TO STEP B  Z=,1PE12.4)  EQUIB   71
1974 81      130 FORMAT (1X,23HPARTS B AND C CONVERGED)  EQUIB   72
1975 82      140 FORMAT (1X,5HYSUM=,1PE13.5,4H Y1=,E13.5,4H ZN=E13.5,5H COS=,E13.5) EQUIB   73
1976 83      150 FORMAT (1X,28H*** EQUIB FAILED TO CONVERGE)  EQUIB   74
1977 84      END                               EQUIB   75
1978 1 *DK FLMSET
1979 2      SUBROUTINE FLMSET
1980 3 *CA SLCDM1
1981 4 C
1982 5 C ** CALCULATE MESH BOUNDARIES FOR PLOTTING
1983 6 C
1984 7 C ** SET BOUNDARIES FOR RECTANGULAR MESH
1985 8 C
1986 9      XBR=X(IM1)
1987 10     Y2=Y(JM1)
1988 11     YBBK=Y2
1989 12     YBF=Y(1)
1990 13     XBL=X(1)
1991 14     ZBT=Z(KM1)
1992 15     ZBB=Z(1)
1993 16     IF (CYL.LT.0.5) GO TO 10
1994 17 C
1995 18 C ** RESET BOUNDARIES FOR CYCLINDRICAL COORDINATES
1996 19 C
1997 20     XBL=XBR*COS(Y2/XBR)
1998 21     IF (Y2/XBR.GT.0.5*PI) XBL=XBR*COS(Y2/XBR)
1999 22     IF ((Y2/XBR).GT.PI) XBL=-XBR
2000 23     YBBK=XBR*SIN(Y2/XBR)

```

```

2001 24      IF ((Y2/XBR).GT.(0.5*PI)) YBBK=XBR*SIN(Y(1)/XBR)          FLMSET 24
2002 25      IF (Y2/XBR.GE.PI) YBBK=XBR                                FLMSET 25
2003 26      YBF=X(1)*SIN(Y(1)/XBR)                                 FLMSET 26
2004 27      IF ((Y2/XBR).LT.0.5*PI) YBF=X(1)*SIN(Y2/XBR)                FLMSET 27
2005 28      IF (Y2/XBR.GT.PI) YBF=-YBBK                               FLMSET 28
2006 29      10 CONTINUE                                         FLMSET 29
2007 30 C
2008 31 C * * SET BOUNDARY VALUES INTO GRDBN-ARRAY FOR USE IN DGAP SUBROUTINE FLMSET 30
2009 32 C
2010 33      GRDBN(1)=XBL                                         FLMSET 31
2011 34      GRDBN(2)=YBF                                         FLMSET 32
2012 35      GRDBN(3)=ZBB                                         FLMSET 33
2013 36      GRDBN(4)=XBR                                         FLMSET 34
2014 37      GRDBN(5)=YBF                                         FLMSET 35
2015 38      GRDBN(6)=ZBB                                         FLMSET 36
2016 39      GRDBN(7)=XBL                                         FLMSET 37
2017 40      GRDBN(8)=YBBK                                         FLMSET 38
2018 41      GRDBN(9)=ZBB                                         FLMSET 39
2019 42      GRDBN(10)=XBR                                         FLMSET 40
2020 43      GRDBN(11)=YBBK                                         FLMSET 41
2021 44      GRDBN(12)=ZBB                                         FLMSET 42
2022 45      GRDBN(13)=XBL                                         FLMSET 43
2023 46      GRDBN(14)=YBF                                         FLMSET 44
2024 47      GRDBN(15)=ZBT                                         FLMSET 45
2025 48      GRDBN(16)=XBR                                         FLMSET 46
2026 49      GRDBN(17)=YBF                                         FLMSET 47
2027 50      GRDBN(18)=ZBT                                         FLMSET 48
2028 51      GRDBN(19)=XBL                                         FLMSET 49
2029 52      GRDBN(20)=YBBK                                         FLMSET 50
2030 53      GRDBN(21)=ZBT                                         FLMSET 51
2031 54      GRDBN(22)=XBR                                         FLMSET 52
2032 55      GRDBN(23)=YBBK                                         FLMSET 53
2033 56      GRDBN(24)=ZBT                                         FLMSET 54
2034 57 C
2035 58 C * * SCALE MESH BOUNDARY VALUES FOR CRT PLOTTING DEVICE FLMSET 55
2036 59 C
2037 60 C * * FOR K PLANE                                         FLMSET 56
2038 61 C
2039 62      FIYB=916.0                                         FLMSET 57
2040 63      XD=(XBR-XBL)/(YBBK-YBF)                           FLMSET 58
2041 64      YY=0.0                                           FLMSET 59
2042 65      IF (XD.LT.1.13556) YY=1.0                         FLMSET 60
2043 66      FIXL(3)=AMAX1(0.0,(511.0-450.0*XD)*YY)           FLMSET 61
2044 67      FIXR=(511.0+450.0*XD)*YY+1022.0*(1.0-YY)         FLMSET 62
2045 68      FIYT=16.0*YY+(916.0-1022.0/XD)*(1.0-YY)          FLMSET 63
2046 69      XCONV(3)=(FIXR-FIXL(3))/(XBR-XBL)                 FLMSET 64
2047 70      YCONV(3)=(FIYT-FIYB)/(YBBK-YBF)                   FLMSET 65
2048 71 C
2049 72 C * * RESET LIMITS FOR RECTANGULAR MESH FOR I AND J PLANE FLMSET 66
2050 73 C
2051 74      XBLC=XBL                                         FLMSET 67
2052 75      YBFC=YBF                                         FLMSET 68
2053 76      XBR=X(IM1)                                         FLMSET 69
2054 77      YBBK=Y(JM1)                                         FLMSET 70
2055 78      XBL=X(1)                                           FLMSET 71
2056 79      YBF=Y(1)                                           FLMSET 72
2057 80 C
2058 81 C * * FOR J PLANE                                         FLMSET 73
2059 82 C
2060 83      XD=(XBR-XBL)/(ZBT-ZBB)                           FLMSET 74
2061 84      YY=0.0                                           FLMSET 75
2062 85      IF (XD.LT.1.13556) YY=1.0                         FLMSET 76
2063 86      FIXL(2)=AMAX1(0.0,(511.0-450.0*XD)*YY)           FLMSET 77
2064 87      FIXR=(511.0+450.0*XD)*YY+1022.0*(1.0-YY)          FLMSET 78
2065 88      FIYT=16.0*YY+(916.0-1022.0/XD)*(1.0-YY)          FLMSET 79
2066 89      XCONV(2)=(FIXR-FIXL(2))/(XBR-XBL)                 FLMSET 80
2067 90      YCONV(2)=(FIYT-FIYB)/(ZBT-ZBB)                   FLMSET 81
2068 91 C
2069 92 C * * FOR I PLANE                                         FLMSET 82
2070 93 C
2071 94      XD=(YBBK-YBF)/(ZBT-ZBB)                           FLMSET 83
2072 95      YY=0.0                                           FLMSET 84
2073 96      IF (XD.LT.1.13556) YY=1.0                         FLMSET 85
2074 97      FIXL(1)=AMAX1(0.0,(511.0-450.0*XD)*YY)           FLMSET 86
2075 98      FIXR=(511.0+450.0*XD)*YY+1022.0*(1.0-YY)          FLMSET 87
2076 99      FIYT=16.0*YY+(916.0-1022.0/XD)*(1.0-YY)          FLMSET 88
2077 100     XCONV(1)=(FIXR-FIXL(1))/(YBBK-YBF)                 FLMSET 89
2078 101     YCONV(1)=(FIYT-FIYB)/(ZBT-ZBB)                   FLMSET 90
2079 102 C
2080 103     RETURN                                         FLMSET 91

```

2081	104	END	FLMSET	104
2082	1	*DK LPRT	LPRT	1
2083	2	SUBROUTINE LPRT	LPRT	2
2084	3	*CA SLCOM1	LPRT	3
2085	4	IF (LPR.EQ.0) RETURN	LPRT	4
2086	5	C	LPRT	5
2087	6	C * * PRINT DATA ON FILM	LPRT	6
2088	7	C	LPRT	7
2089	8	CALL LINCNT (64)	LPRT	8
2090	9	CALL FADV (1)	LPRT	9
2091	10	ASSIGN 30 TO KR1	LPRT	10
2092	11	GO TO 20	LPRT	11
2093	12	10 LINES=LINES+1	LPRT	12
2094	13	IF (LINES.LT.55) GO TO 40	LPRT	13
2095	14	20 LINES=0	LPRT	14
2096	15	WRITE (12,70)	LPRT	15
2097	16	WRITE (12,100) NAME,JNM,DAT,CLK	LPRT	16
2098	17	WRITE (12,110) ITER,T,DELT,CYCLE	LPRT	17
2099	18	WRITE (12,90)	LPRT	18
2100	19	GO TO KR1, (30,40)	LPRT	19
2101	20	30 ASSIGN 40 TO KR1	LPRT	20
2102	21	DO 50 J=1,JMAX	LPRT	21
2103	22	DO 50 I=1,IMAX	LPRT	22
2104	23	DO 50 K=1,KMAX	LPRT	23
2105	24	CALL IJKONLY	LPRT	24
2106	25	WRITE (12,80) I,J,K,U(IJK),V(IJK),W(IJK),P(IJK),PS(IJK),F(IJK),NF	LPRT	25
2107	26	1 (IJK),NFP(IJK),NFS(IJK),NFO(IJK),BETA(IJK),PETA(IJK),D(IJK)	LPRT	26
2108	27	GO TO 10	LPRT	27
2109	28	40 CONTINUE	LPRT	28
2110	29	50 CONTINUE	LPRT	29
2111	30	IF (LPR.LE.2) RETURN	LPRT	30
2112	31	C	LPRT	31
2113	32	C * * PRINT DATA ON PAPER	LPRT	32
2114	33	C	LPRT	33
2115	34	ENTRY LPRT2	LPRT	34
2116	35	WRITE (9,70)	LPRT	35
2117	36	WRITE (9,100) NAME,JNM,DAT,CLK	LPRT	36
2118	37	WRITE (9,110) ITER,T,DELT,CYCLE	LPRT	37
2119	38	WRITE (9,90)	LPRT	38
2120	39	DO 60 J=1,JMAX	LPRT	39
2121	40	DO 60 I=1,IMAX	LPRT	40
2122	41	DO 60 K=1,KMAX	LPRT	41
2123	42	CALL IJKONLY	LPRT	42
2124	43	WRITE (9,80) I,J,K,U(IJK),V(IJK),W(IJK),P(IJK),PS(IJK),F(IJK),NF	LPRT	43
2125	44	1 (IJK),NFP(IJK),NFS(IJK),NFO(IJK),BETA(IJK),PETA(IJK),D(IJK)	LPRT	44
2126	45	60 CONTINUE	LPRT	45
2127	46	RETURN	LPRT	46
2128	47	C	LPRT	47
2129	48	70 FORMAT (1H1)	LPRT	48
2130	49	80 FORMAT (3I3.6(1X,1PE11:4),2I3,2I4,3(1X,1PE10:3))	LPRT	49
2131	50	90 FORMAT (1OH I J K,6X,1HU,11X,1HV,11X,1HW,11X,1HP,10X,2HPS,10X,1	LPRT	50
2132	51	1 HF,6X,2HNF,1X,3HNFP,1X,3HNFS,1X,3HNFO,4X,4HBETA,7X,4HPETA,8X,1HD)	LPRT	51
2133	52	100 FORMAT (1H ,18X,8AB,1X,A8,2(1X,A8))	LPRT	52
2134	53	110 FORMAT (5X,6HITER= ,I5,5X,6HTIME= ,1PE12.5,5X,6HDELT= ,1PE12.5,5X,	LPRT	53
2135	54	1 7HCYCLE= ,14)	LPRT	54
2136	55	END	LPRT	55
2137	1	*DK MESHSET	MESHSET	1
2138	2	SUBROUTINE MESHSET	MESHSET	2
2139	3	*CA SLCOM1	MESHSET	3
2140	4	C	MESHSET	4
2141	5	C * * MESH GENERATOR FOR X,Y,Z-COORDINATES	MESHSET	5
2142	6	C	MESHSET	6
2143	7	NAMELIST /MESHGN/ NKX,XL,XC,NXL,NXR,DXMN,NKY,YL,YC,NYL,NYR,DYMN	MESHSET	7
2144	8	1 ,NKZ,ZL,ZC,NZL,NZR,DZMN,NOBS,OA2,OA1,OB2,OB1,DC2,DC1,IOH	MESHSET	8
2145	9	DATA NOBS /O/	MESHSET	9
2146	10	C	MESHSET	10
2147	11	C * * READ IN DATA FROM NAMELIST / MESHGN /	MESHSET	11
2148	12	C	MESHSET	12
2149	13	READ (10,MESHGN)	MESHSET	13
2150	14	C	MESHSET	14
2151	15	C * * WRITE INPUT DATA FOR MESH GENERATOR ONTO TAPES 9 AND 12.	MESHSET	15
2152	16	C	MESHSET	16
2153	17	WRITE (9,220) NKX	MESHSET	17
2154	18	WRITE (12,220) NKX	MESHSET	18
2155	19	DO 10 L=1,NKX	MESHSET	19
2156	20	WRITE (9,230) L,XL(L),XC(L),XL(L+1),NXL(L),NXR(L),DXMN(L)	MESHSET	20
2157	21	WRITE (12,230) L,XL(L),XC(L),XL(L+1),NXL(L),NXR(L),DXMN(L)	MESHSET	21
2158	22	10 CONTINUE	MESHSET	22
2159	23	C	MESHSET	23
2160	24	WRITE (9,240) NKY	MESHSET	24

```

2161 25      WRITE (12,240) NKY                         MESHSET 25
2162 26      DO 20 L=1,NKY                         MESHSET 26
2163 27      WRITE (9,250) L,YL(L),YC(L),YL(L+1),NYL(L),NYR(L),DYMN(L) MESHSET 27
2164 28      WRITE (12,250) L,YL(L),YC(L),YL(L+1),NYL(L),NYR(L),DYMN(L) MESHSET 28
2165 29      20 CONTINUE                           MESHSET 29
2166 30 C     WRITE (9,260) NKZ                         MESHSET 30
2167 31      WRITE (12,260) NKZ                         MESHSET 31
2168 32      DO 30 L=1,NKZ                         MESHSET 32
2169 33      WRITE (9,270) L,ZL(L),ZC(L),ZL(L+1),NZL(L),NZR(L),DZMN(L) MESHSET 33
2170 34      WRITE (12,270) L,ZL(L),ZC(L),ZL(L+1),NZL(L),NZR(L),DZMN(L) MESHSET 34
2171 35      30 CONTINUE                           MESHSET 35
2172 36      30 CONTINUE                           MESHSET 36
2173 37 C     * * COMPUTE X-COORDINATE VALUES        MESHSET 37
2174 38 C     CALL MESHX (NUMXP1,NKX)                 MESHSET 38
2175 39 C     * * COMPUTE Y-COORDINATE VALUES        MESHSET 39
2176 40      CALL MESHY (NUMYP1,NKY)                 MESHSET 40
2177 41 C     * * COMPUTE Z-COORDINATE VALUES        MESHSET 41
2178 42 C     CALL MESHZ (NUMZP1,NKZ)                 MESHSET 42
2179 43 C     * * COMPUTE CONSTANT TERMS            MESHSET 43
2180 44      CALL MESHZ (NUMZP1,NKZ)                 MESHSET 44
2181 45 C     * * COMPUTE CONSTANT TERMS            MESHSET 45
2182 46 C     * * COMPUTE CONSTANT TERMS            MESHSET 46
2183 47 C     * * COMPUTE CONSTANT TERMS            MESHSET 47
2184 48      CALL MESHZ (NUMZP1,NKZ)                 MESHSET 48
2185 49 C     * * COMPUTE CONSTANT TERMS            MESHSET 49
2186 50      WRITE (9,280) IBAR,JBAR,KBAR           MESHSET 50
2187 51      WRITE (12,280) IBAR,JBAR,KBAR           MESHSET 51
2188 52      WRITE (59,280) IBAR,JBAR,KBAR           MESHSET 52
2189 53 C     * * COMPUTE CONSTANT TERMS            MESHSET 53
2190 54 C     * * COMPUTE CONSTANT TERMS            MESHSET 54
2191 55 C     * * COMPUTE CONSTANT TERMS            MESHSET 55
2192 56      IMAX=IBAR+2                          MESHSET 56
2193 57      JMAX=JBAR+2                          MESHSET 57
2194 58      KMAX=KBAR+2                          MESHSET 58
2195 59      IM1=IMAX-1                           MESHSET 59
2196 60      JM1=JMAX-1                           MESHSET 60
2197 61      KM1=KMAX-1                           MESHSET 61
2198 62      IM2=IMAX-2                           MESHSET 62
2199 63      JM2=JMAX-2                           MESHSET 63
2200 64      KM2=KMAX-2                           MESHSET 64
2201 65      II5=IMAX*JMAX                      MESHSET 65
2202 66      II0=NQ*IBAR                        MESHSET 66
2203 67      II1=NQ*IMAX                        MESHSET 67
2204 68      II2=NQ*II5                         MESHSET 68
2205 69      II3=NQ*IMAX*JBAR                     MESHSET 69
2206 70      II4=NQ*II5*KBAR                     MESHSET 70
2207 71      II6=IMAX*JBAR                       MESHSET 71
2208 72      II7=IMAX*JM1                        MESHSET 72
2209 73 C     * * SET ARRAYS TO INDEFINITE       MESHSET 73
2210 74 C     * * SET ARRAYS TO INDEFINITE       MESHSET 74
2211 75 C     CALL XINDF                         MESHSET 75
2212 76      CALL DATEH (DAT)                     MESHSET 76
2213 77 C     CALL CLOCK1 (CLK)                   MESHSET 77
2214 78      CALL DATEH (DAT)                     MESHSET 78
2215 79      CALL CLOCK1 (CLK)                   MESHSET 79
2216 80 C     * * INITIALIZE RADII FOR CYL. COORDINATES MESHSET 80
2217 81 C     * * INITIALIZE RADII FOR CYL. COORDINATES MESHSET 81
2218 82 C     NOTE, Y=X(IM1)*THETA                  MESHSET 82
2219 83 C     * * INITIALIZE RADII FOR CYL. COORDINATES MESHSET 83
2220 84      DO 40 L=2,IMAX                      MESHSET 84
2221 85      RR(L)=1.0-CYL+CYL*X(IM1)/X(L)       MESHSET 85
2222 86      40 RRI(L)=1.0-CYL+CYL*X(IM1)/XI(L)   MESHSET 86
2223 87      RR(1)=1.0                           MESHSET 87
2224 88      RRI(1)=1.0                           MESHSET 88
2225 89      IF (X(1).EQ.0.0) GO TO 50          MESHSET 89
2226 90      RR(1)=X(IM1)/X(1)                   MESHSET 90
2227 91      RRI(1)=X(IM1)/XI(1)                 MESHSET 91
2228 92      50 CONTINUE                         MESHSET 92
2229 93 C     DO 60 J=1,JMAX                      MESHSET 93
2230 94      STHJ(J)=SIN(YJ(J)/X(IM1))          MESHSET 94
2231 95      STHJBK(J)=SIN(Y(J)/X(IM1))         MESHSET 95
2232 96      CTHJ(J)=COS(YJ(J)/X(IM1))          MESHSET 96
2233 97      CTHJBK(J)=COS(Y(J)/X(IM1))         MESHSET 97
2234 98      60 CONTINUE                         MESHSET 98
2235 99      JC2PI=0                            MESHSET 99
2236 100     IF (CYL.LT.0.5) GO TO 110          MESHSET 100
2237 101     IF (ABS(Y(JM1)/X(IM1)-6.2831853).GT.0.01) GO TO 90 MESHSET 101
2238 102     JC2PI=1                            MESHSET 102
2239 103     INC=JBAR/2                          MESHSET 103
2240 104

```

```

2241 105      JLAST=INC+1          MESHSET 105
2242 106      JSTART=JLAST+1       MESHSET 106
2243 107      DO 70 J=1,JLAST      MESHSET 107
2244 108      JOP(J)=J+INC        MESHSET 108
2245 109      70 CONTINUE         MESHSET 109
2246 110      DD 80 J=JSTART,JMAX MESHSET 110
2247 111      JOP(J)=J-INC        MESHSET 111
2248 112      80 CONTINUE         MESHSET 112
2249 113      GO TO 110          MESHSET 113
2250 114      90 DD 100 J=1,JMAX MESHSET 114
2251 115      JDP(J)=JMAX-J+1    MESHSET 115
2252 116      100 CONTINUE        MESHSET 116
2253 117      110 CONTINUE        MESHSET 117
2254 118 C * * WRITE MESH GENERATOR OUTPUT MESHSET 118
2255 119 C
2256 120      WRITE (9,290)        MESHSET 120
2257 121      WRITE (12,290)        MESHSET 121
2258 122      DO 120 N=1,NUMXP1     MESHSET 122
2259 123      WRITE (9,300) N,X(N),N,RX(N),N,DELX(N),N,RDX(N),N,XI(N),N,RXI(N) MESHSET 123
2260 124      WRITE (12,300) N,X(N),N,RX(N),N,DELX(N),N,RDX(N),N,XI(N),N,RXI(N) MESHSET 124
2261 125      120 CONTINUE        MESHSET 125
2262 126      WRITE (9,290)        MESHSET 126
2263 127      WRITE (12,290)        MESHSET 127
2264 128      DO 130 N=1,NUMYP1     MESHSET 128
2265 129      WRITE (9,310) N,Y(N),N,DELY(N),N,RDY(N),N,YU(N),N,RYU(N) MESHSET 129
2266 130      WRITE (12,310) N,Y(N),N,DELY(N),N,RDY(N),N,YU(N),N,RYU(N) MESHSET 130
2267 131      130 CONTINUE        MESHSET 131
2268 132      WRITE (9,290)        MESHSET 132
2269 133      WRITE (12,290)        MESHSET 133
2270 134      DO 140 N=1,NUMZP1     MESHSET 134
2271 135      WRITE (9,320) N,Z(N),N,DELZ(N),N,RDZ(N),N,ZK(N),N,RZK(N) MESHSET 135
2272 136      WRITE (12,320) N,Z(N),N,DELZ(N),N,RDZ(N),N,ZK(N),N,RZK(N) MESHSET 136
2273 137      140 CONTINUE        MESHSET 137
2274 138 C
2275 139 C      CALCULATE PARTIAL AREAS MESHSET 139
2276 140 C
2277 141      CALL ASET           MESHSET 141
2278 142 C
2279 143 C      PRINT OBSTACLE DATA MESHSET 143
2280 144 C
2281 145      IF (LPR.EQ.0) GO TO 160 MESHSET 145
2282 146      IF (NOBS.LE.0) GO TO 160 MESHSET 146
2283 147      WRITE (9,210)        MESHSET 147
2284 148      WRITE (12,210)        MESHSET 148
2285 149      DO 150 I=1,NOBS      MESHSET 149
2286 150      WRITE (9,200) I,OA2(I),OA1(I),OB2(I),OB1(I),OC2(I),OC1(I),IOH(I) MESHSET 150
2287 151      WRITE (12,200) I,OA2(I),OA1(I),OB2(I),OB1(I),OC2(I),OC1(I),IOH(I) MESHSET 151
2288 152      150 CONTINUE        MESHSET 152
2289 153      160 CONTINUE        MESHSET 153
2290 154 C * * CALC RATIOS FOR USE IN NO SLIP BDY. CONDITIONS IN SUBROUTINE BC MESHSET 154
2291 155 C
2292 156      DELXRL=DELX(1)/DELX(2) MESHSET 156
2293 157      DELXRR=DELX(IMAX)/DELX(IM1) MESHSET 157
2294 158      DELYRF=DELY(1)/DELY(2) MESHSET 158
2295 159      DELYRBK=DELY(JMAX)/DELY(JM1) MESHSET 159
2296 160      DELZRB=DELZ(1)/DELZ(2) MESHSET 160
2297 161      DELZRT=DELZ(KMAX)/DELZ(KM1) MESHSET 161
2298 162 C
2299 163 C * * DETERMINE MINIMUM CELL SIZE (LENGTH) MESHSET 163
2300 164 C
2301 165      DELXMN=1.OE+20 MESHSET 165
2302 166      DELYMN=1.OE+20 MESHSET 166
2303 167      DELZMN=1.OE+20 MESHSET 167
2304 168      DO 170 N=1,IMAX MESHSET 168
2305 169      170 DELXMN=AMIN1(DELX(N),DELXMN) MESHSET 169
2306 170      DD 180 N=1,JMAX MESHSET 170
2307 171      180 DELYMN=AMIN1(DELY(N),DELYMN) MESHSET 171
2308 172      DO 190 N=1,KMAX MESHSET 172
2309 173      190 DELZMN=AMIN1(DELZ(N),DELZMN) MESHSET 173
2310 174      DELMN=AMIN1(DELXMN,DELYMN,DELZMN) MESHSET 174
2311 175      RETURN            MESHSET 175
2312 176 C
2313 177      200 FORMAT (2X,2HI=,I1,2X,5HOA2= ,1PE12.5,2X,5HOA1= ,E12.5,2X,5HOB2= MESHSET 177
2314 178      1 ,E12.5,2X,5HOB1= ,E12.5,2X,5HOC2= ,E12.5,2X,5HOC1= ,E12.5,2X,5HIO MESHSET 178
2315 179      2H= ,I2) MESHSET 179
2316 180      210 FORMAT (//20X,25HCONIC OBSTACLE PARAMETERS) MESHSET 180
2317 181      220 FORMAT (2X,5HNKX= ,I4) MESHSET 181
2318 182      230 FORMAT (2X,8HMESH-X= ,I4,3X,4HXL= ,1PE12.5,3X,4HXC= ,E12.5,3X,4HXR MESHSET 182
2319 183      1= ,E12.5,3X,5HNXL= ,I4,3X,5HNXR= ,I4,3X,6HDXMN= ,E12.5) MESHSET 183
2320 184      240 FORMAT (2X,5HNKY= ,I4) MESHSET 184

```

```

2321 185 250 FORMAT (2X,8HMESH-Y= ,I4,3X,4HYL= ,1PE12.5,3X,4HYC= ,E12.5,3X,4HYR MESHSET 185
2322 186 1= ,E12.5,3X,5HNYL= ,I4,3X,5HNYR= ,I4,3X,6HDYMN= ,E12.5) MESHSET 186
2323 187 260 FORMAT (2X,5HNKZ= .14) MESHSET 187
2324 188 270 FORMAT (2X,8HMESH-Z= ,I4,3X,4HZL= ,1PE12.5,3X,4HZC= ,E12.5,3X,4HZR MESHSET 188
2325 189 1= ,E12.5,3X,5HNZL= ,I4,3X,5HNZR= ,I4,3X,6HDZMN= ,E12.5) MESHSET 189
2326 190 280 FORMAT (/,7H IBAR=,I3,8H JBAR=,I3,8H KBAR=,I3) MESHSET 190
2327 191 290 FORMAT (1H1) MESHSET 191
2328 192 300 FORMAT (1X,2HX(,I2,2H)=,1PE12.5,2X,3HRX(,I2,2H)=,1PE12.5,2X,5HDELX MESHSET 192
2329 193 1(,I2,2H)=,1PE12.5,1X,4HRDX(,I2,2H)=,1PE12.5,2X,3HXI(,I2,2H)=,1PE12 MESHSET 193
2330 194 2 .5,2X,4HRXI(,I2,2H)=,1PE12.5) MESHSET 194
2331 195 310 FORMAT (1X,2HY(,I2,2H)=,1PE12.5,3X,5HDELY(,I2,2H)=,1PE12.5,3X,4HRD MESHSET 195
2332 196 1Y(,I2,2H)=,1PE12.5,3X,3HYJ(,I2,2H)=,1PE12.5,3X,4HRYJ(,I2,2H)=,1PE1 MESHSET 196
2333 197 2 2.5) MESHSET 197
2334 198 320 FORMAT (1X,2HZ(,I2,2H)=,1PE12.5,3X,5HDELZ(,I2,2H)=,1PE12.5,3X,4HRD MESHSET 198
2335 199 1Z(,I2,2H)=,1PE12.5,3X,3HZK(,I2,2H)=,1PE12.5,3X,4HRZK(,I2,2H)=,1PE1 MESHSET 199
2336 200 2 2.5) MESHSET 200
2337 201 END MESHSET 201
2338 1 *DK MESHX MESHX 1
2339 2 SUBROUTINE MESHX (NUMXP1,NKX) MESHX 2
2340 3 *CA SLCOM1 MESHX 3
2341 4 C MESHX 4
2342 5 C * * COMPUTE X-COORDINATE VALUES AND RECIPROCAL MESHX 5
2343 6 C MESHX 6
2344 7 X(1)=XL(1) MESHX 7
2345 8 I11=1 MESHX 8
2346 9 DO 30 L=1,NKX MESHX 9
2347 10 DXML=(XC(L)-XL(L))/NXL(L) MESHX 10
2348 11 DXMN1=DXMN(L) MESHX 11
2349 12 NT=NXL(L) MESHX 12
2350 13 TN=NT MESHX 13
2351 14 TN=AMAX1(TN, 1.0+1.OE-14) MESHX 14
2352 15 DXMN(L)=AMIN1(DXMN1,DXML) MESHX 15
2353 16 CMC=(XC(L)-XL(L)-TN*DXMN(L))*TN/(TN-1.0) MESHX 16
2354 17 IF (NT.EQ.1) CMC=0.0 MESHX 17
2355 18 BMC=XC(L)-XL(L)-CMC MESHX 18
2356 19 DO 10 L1=1,NT MESHX 19
2357 20 I11=I11+1 MESHX 20
2358 21 RLN=(FLOAT(L1)-TN)/TN MESHX 21
2359 22 X(I11)=XC(L)+BMC*RLN-CMC*RLN*RLN MESHX 22
2360 23 DXMR=(XL(L+1)-XC(L))/NXR(L) MESHX 23
2361 24 DXMN(L)=AMIN1(DXMN1,DXMR) MESHX 24
2362 25 NT=NXR(L) MESHX 25
2363 26 TN=NT MESHX 26
2364 27 TN=AMAX1(TN, 1.0+1.OE-14) MESHX 27
2365 28 CMC=(XL(L+1)-XC(L)-TN*DXMN(L))*TN/(TN-1.0) MESHX 28
2366 29 IF (NT.EQ.1) CMC=0.0 MESHX 29
2367 30 BMC=XL(L+1)-XC(L)-CMC MESHX 30
2368 31 DO 20 L1=1,NT MESHX 31
2369 32 I11=I11+1 MESHX 32
2370 33 RLN=FLOAT(L1)/TN MESHX 33
2371 34 X(I11)=XC(L)+BMC*RLN+CMC*RLN*RLN MESHX 34
2372 35 20 CONTINUE MESHX 35
2373 36 NUMX=I11 MESHX 36
2374 37 NUMXM1=NUMX-1 MESHX 37
2375 38 NUMXP1=NUMX+1 MESHX 38
2376 39 IBAR=NUMX-1 MESHX 39
2377 40 DO 40 I11=2,NUMX MESHX 40
2378 41 40 DELX(I11)=X(I11)-X(I11-1) MESHX 41
2379 42 DELX(1)=DELX(2) MESHX 42
2380 43 DELX(NUMXP1)=DELX(NUMX) MESHX 43
2381 44 X(NUMXP1)=X(NUMX)+DELX(NUMXP1) MESHX 44
2382 45 C MESHX 45
2383 46 DO 60 I11=1,NUMXP1 MESHX 46
2384 47 IF (X(I11).EQ.0.0) GO TO 50 MESHX 47
2385 48 RX(I11)=1.0/X(I11) MESHX 48
2386 49 GO TO 60 MESHX 49
2387 50 50 RX(I11)=0.0 MESHX 50
2388 51 60 CONTINUE MESHX 51
2389 52 DO 70 I11=2,NUMXP1 MESHX 52
2390 53 XI(I11)=0.5*(X(I11-1)+X(I11)) MESHX 53
2391 54 RXI(I11)=1.0/XI(I11) MESHX 54
2392 55 IF (I11.LT.NUMXP1) RDXP(I11)=2.0/(DELX(I11)+DELX(I11+1)) MESHX 55
2393 56 70 RD(X(I11)=1.0/DELX(I11) MESHX 56
2394 57 XI(1)=XI(2)-DELX(2) MESHX 57
2395 58 RXI(1)=1.0/XI(1) MESHX 58
2396 59 RD(X(1)=1.0/DELX(1) MESHX 59
2397 60 RD(XP(1)=2.0/(DELX(1)+DELX(2)) MESHX 60
2398 61 C MESHX 61
2399 62 RETURN MESHX 62
2400 63 END MESHX 63

```

```

2401 1 *DK MESHY
2402 2 SUBROUTINE MESHY (NUMYP1,NKY)
2403 3 *CA SLCOM1
2404 4 C
2405 5 C * * COMPUTE Y-COORDINATE VALUES AND RECIPROCALS
2406 6 C
2407 7 IF (NKY.EQ.1.AND.(NYL(1).EQ.0.OR.NYR(1).EQ.0)) GO TO 40
2408 8 Y(1)=YL(1)
2409 9 J11=1
2410 10 DO 30 L=1,NKY
2411 11 DYML=(YC(L)-YL(L))/NYL(L)
2412 12 DYMN1=DYMN(L)
2413 13 NT=NYL(L)
2414 14 TN=NT
2415 15 TN=AMAX1(TN,1.0+1.0E-14)
2416 16 DYMN(L)=AMIN1(DYMN1,DYML)
2417 17 CMC=(YC(L)-YL(L)-TN*DYMN(L))*TN/(TN-1.0)
2418 18 IF (NT.EQ.1) CMC=0.0
2419 19 BMC=YC(L)-YL(L)-CMC
2420 20 DO 10 L1=1,NT
2421 21 J11=J11+1
2422 22 RLN=(FLOAT(L1)-TN)/TN
2423 23 10 Y(J11)=YC(L)+BMC*RLN-CMC*RLN*RLN
2424 24 DYMRE=(YL(L+1)-YC(L))/NYR(L)
2425 25 DYMN(L)=AMIN1(DYMN1,DYMRE)
2426 26 NT=NYR(L)
2427 27 TN=NT
2428 28 TN=AMAX1(TN,1.0+1.0E-14)
2429 29 CMC=(YL(L+1)-YC(L)-TN*DYMN(L))*TN/(TN-1.0)
2430 30 IF (NT.EQ.1) CMC=0.0
2431 31 BMC=YL(L+1)-YC(L)-CMC
2432 32 DO 20 L1=1,NT
2433 33 J11=J11+1
2434 34 RLN=FLOAT(L1)/TN
2435 35 20 Y(J11)=YC(L)+BMC*RLN+CMC*RLN*RLN
2436 36 30 CONTINUE
2437 37 GO TO 50
2438 38 40 J11=2
2439 39 Y(J11)=1.0
2440 40 Y(J11-1)=0.0
2441 41 50 CONTINUE
2442 42 NUMY=J11
2443 43 NUMYM1=NUMY-1
2444 44 NUMYP1=NUMY+1
2445 45 UBAR=NUMY-1
2446 46 DO 60 J11=2,NUMY
2447 47 60 DELY(J11)=Y(J11)-Y(J11-1)
2448 48 DELY(1)=DELY(2)
2449 49 DELY(NUMYP1)=DELY(NUMY)
2450 50 Y(NUMYP1)=Y(NUMY)+DELY(NUMYP1)
2451 51 C
2452 52 DO 80 J11=1,NUMYP1
2453 53 IF (Y(J11).EQ.0.0) GO TO 70
2454 54 RY(J11)=1.0/Y(J11)
2455 55 GO TO 80
2456 56 70 RY(J11)=0.0
2457 57 80 CONTINUE
2458 58 DO 90 J11=2,NUMYP1
2459 59 YJ(J11)=0.5*(Y(J11-1)+Y(J11))
2460 60 RYJ(J11)=1.0/YJ(J11)
2461 61 IF (J11.LT.NUMYP1) RDYP(J11)=2.0/(DELY(J11)+DELY(J11+1))
2462 62 90 RDY(J11)=1.0/DELY(J11)
2463 63 YJ(1)=YJ(2)-DELY(2)
2464 64 RYJ(1)=1.0/YJ(1)
2465 65 RDY(1)=1.0/DELY(1)
2466 66 RDYP(1)=2.0/(DELY(1)+DELY(2))
2467 67 C
2468 68 RETURN
2469 69 END
2470 1 *DK MESHZ
2471 2 SUBROUTINE MESHZ (NUMZP1,NKZ)
2472 3 *CA SLCOM1
2473 4 C
2474 5 C * * COMPUTE Z-COORDINATE VALUES AND RECIPROCALS
2475 6 C
2476 7 Z(1)=ZL(1)
2477 8 K11=1
2478 9 DO 30 L=1,NKZ
2479 10 DZML=(ZC(L)-ZL(L))/NZL(L)
2480 11 DZMN1=DZMN(L)

```

```

2481 12      NT=NZL(L)                                MESHZ   12
2482 13      TN=NT                                  MESHZ   13
2483 14      TN=AMAX1(TN, 1.0+1.OE-14)              MESHZ   14
2484 15      DZMN(L)=AMIN1(DZMN1,DZML)             MESHZ   15
2485 16      CMC=(ZC(L)-ZL(L)-TN*DZMN(L))*TN/(TN-1.0) MESHZ   16
2486 17      IF (NT.EQ.1) CMC=0.0                  MESHZ   17
2487 18      BMC=ZC(L)-ZL(L)-CMC                 MESHZ   18
2488 19      DO 10 L1=1,NT                           MESHZ   19
2489 20      K11=K11+1                            MESHZ   20
2490 21      RLN=(FLOAT(L1)-TN)/TN                MESHZ   21
2491 22      10 Z(K11)=ZC(L)+BMC*RLN-CMC*RLN*RLN  MESHZ   22
2492 23      DZMR=(ZL(L+1)-ZC(L))/NZR(L)          MESHZ   23
2493 24      DZMN(L)=AMIN1(DZMN1,DZMR)            MESHZ   24
2494 25      NT=NZR(L)                            MESHZ   25
2495 26      TN=NT                                  MESHZ   26
2496 27      TN=AMAX1(TN, 1.0+1.OE-14)              MESHZ   27
2497 28      CMC=(ZL(L+1)-ZC(L)-TN*DZMN(L))*TN/(TN-1.0) MESHZ   28
2498 29      IF (NT.EQ.1) CMC=0.0                  MESHZ   29
2499 30      BMC=ZL(L+1)-ZC(L)-CMC               MESHZ   30
2500 31      DO 20 L1=1,NT                           MESHZ   31
2501 32      K11=K11+1                            MESHZ   32
2502 33      RLN=FLOAT(L1)/TN                      MESHZ   33
2503 34      20 Z(K11)=ZC(L)+BMC*RLN+CMC*RLN*RLN  MESHZ   34
2504 35      30 CONTINUE                          MESHZ   35
2505 36      NUMZ=K11                             MESHZ   36
2506 37      NUMZM1=NUMZ-1                         MESHZ   37
2507 38      NUMZP1=NUMZ+1                         MESHZ   38
2508 39      KBAR=NUMZ-1                          MESHZ   39
2509 40      DO 40 K11=2,NUMZ                      MESHZ   40
2510 41      40 DELZ(K11)=Z(K11)-Z(K11-1)        MESHZ   41
2511 42      DELZ(1)=DELZ(2)                      MESHZ   42
2512 43      DELZ(NUMZP1)=DELZ(NUMZ)            MESHZ   43
2513 44      Z(NUMZP1)=Z(NUMZ)+DELZ(NUMZP1)       MESHZ   44
2514 45      C
2515 46      DO 60 K11=1,NUMZP1                   MESHZ   46
2516 47      IF (Z(K11).EQ.0.0) GO TO 50          MESHZ   47
2517 48      RZ(K11)=1.0/Z(K11)                  MESHZ   48
2518 49      GO TO 60                            MESHZ   49
2519 50      50 RZ(K11)=0.0                      MESHZ   50
2520 51      60 CDNTINUE                        MESHZ   51
2521 52      DO 70 K11=2,NUMZP1                   MESHZ   52
2522 53      ZK(K11)=0.5*(Z(K11-1)+Z(K11))       MESHZ   53
2523 54      RZK(K11)=1.0/ZK(K11)                MESHZ   54
2524 55      IF (K11.LT.NUMZP1) RDZP(K11)=2.0/(DELZ(K11)+DELZ(K11+1)) MESHZ   55
2525 56      70 RDZ(K11)=1.0/DELZ(K11)           MESHZ   56
2526 57      ZK(1)=ZK(2)-DELZ(2)                 MESHZ   57
2527 58      RZK(1)=1.0/ZK(1)                   MESHZ   58
2528 59      RDZ(1)=1.0/DELZ(1)                 MESHZ   59
2529 60      RDZP(1)=2.0/(DELZ(1)+DELZ(2))       MESHZ   60
2530 61      C
2531 62      RETURN
2532 63      END
2533 1 *DK PCNV
2534 2      SUBROUTINE PCNV (IXI,IETA,XX1,YY1,ZZ1)
2535 3 *CA SLCOM1
2536 4 *CA SLCOM2
2537 5 C
2538 6 C * * PERSPECTIVE CONVERT SUBR - PRODUCES 4020 COORDINATES FOR
2539 7 C     PERSPECTIVE VIEWS
2540 8 C
2541 9      IXI=0
2542 10     IETA=0
2543 11     TRCR=(YY1-YCC)*CSTH-(XX1-XCC)*SNTH
2544 12     XB=(XX1-XCC)*CSTH+(YY1-YCC)*SNTH
2545 13     YB=(ZCC-ZZ1)*SNPHI+TRCR*CSPHI
2546 14     ZZB=(ZZ1-ZCC)*CSPHI+TRCR*SNPHI
2547 15     YDEN=YB-YEB
2548 16     IF (YDEN.LT.1.OE-6) GO TO 10
2549 17     YRAT=YEB/YDEN
2550 18     XID=XEB-(XB-XEB)*YRAT
2551 19     ETA=ZEB-(ZZB-ZEB)*YRAT
2552 20     ISH=61
2553 21     IF (LPR.EQ.0) ISH=0
2554 22     XIA=0.5*(XIMX+XIMN)
2555 23     ETAA=0.5*(ETAMX+ETAMN)
2556 24     IETA=512-(ETA-ETAA)*GDRAT-ISH
2557 25     IXI=512+(XID-XIA)*GDRAT+ISH
2558 26     10 CONTINUE
2559 27 C
2560 28     RETURN

```

2561	29	END	PCNV	29
2562	1	*DK PDFCALC	PDFCALC	1
2563	2	*DK PETACAL	PETACAL	1
2564	3	SUBROUTINE PETACAL	PETACAL	2
2565	4	*CA SLCOM1	PETACAL	3
2566	5	C	PETACAL	4
2567	6	C * * DETERMINE PROVISIONAL NF VALUES, SURFACE TENSION PRESSURES, AND	PETACAL	5
2568	7	C * * PRESSURE INTERPOLATION FACTOR PETA FOR SURFACE AND NEIGHBOR CELLS	PETACAL	6
2569	8	C	PETACAL	7
2570	9	DO 10 K=1,KMAX	PETACAL	8
2571	10	DO 10 J=1,JMAX	PETACAL	9
2572	11	DO 10 I=1,IMAX	PETACAL	10
2573	12	CALL CALCIJK	PETACAL	11
2574	13	NFO(IJK)=NF(IJK)	PETACAL	12
2575	14	NF(IJK)=0	PETACAL	13
2576	15	PETA(IJK)=1.0	PETACAL	14
2577	16	PS(IJK)=0.0	PETACAL	15
2578	17	NFP(IJK)=0	PETACAL	16
2579	18	NFS(IJK)=0	PETACAL	17
2580	19	10 CONTINUE	PETACAL	18
2581	20	C	PETACAL	19
2582	21	CALL PCAL	PETACAL	20
2583	22	C CALC SURFACE TENSION	PETACAL	21
2584	23	IF (ISURFT.EQ.1) CALL SURF1ON	PETACAL	22
2585	24	C	PETACAL	23
2586	25	DO 100 K=2,KM1	PETACAL	24
2587	26	DO 100 J=2,JM1	PETACAL	25
2588	27	DO 100 I=2,IM1	PETACAL	26
2589	28	CALL IJKAJCT	PETACAL	27
2590	29	IF (BETA(IJK).LT.0.0) GO TO 100	PETACAL	28
2591	30	L=I	PETACAL	29
2592	31	M=J	PETACAL	30
2593	32	N=K	PETACAL	31
2594	33	NFC=NF(IJK)+1	PETACAL	32
2595	34	GO TO (100,20,30,40,50,60,70,80,100), NFC	PETACAL	33
2596	35	20 DSUR=DELX(I)	PETACAL	34
2597	36	DNBR=DELX(I-1)	PETACAL	35
2598	37	L=I-1	PETACAL	36
2599	38	GO TO 90	PETACAL	37
2600	39	30 DSUR=DELX(I)	PETACAL	38
2601	40	DNBR=DELX(I+1)	PETACAL	39
2602	41	L=I+1	PETACAL	40
2603	42	GO TO 90	PETACAL	41
2604	43	40 DSUR=DELY(J)	PETACAL	42
2605	44	DNBR=DELY(J-1)	PETACAL	43
2606	45	M=J-1	PETACAL	44
2607	46	GO TO 90	PETACAL	45
2608	47	C RADIAL DISTANCE FACTOR CANCELS	PETACAL	46
2609	48	50 DSUR=DELY(J)	PETACAL	47
2610	49	DNBR=DELY(J+1)	PETACAL	48
2611	50	M=J+1	PETACAL	49
2612	51	GO TO 90	PETACAL	50
2613	52	60 DSUR=DELZ(K)	PETACAL	51
2614	53	DNBR=DELZ(K-1)	PETACAL	52
2615	54	N=K-1	PETACAL	53
2616	55	GO TO 90	PETACAL	54
2617	56	70 DSUR=DELZ(K)	PETACAL	55
2618	57	DNBR=DELZ(K+1)	PETACAL	56
2619	58	N=K+1	PETACAL	57
2620	59	GO TO 90	PETACAL	58
2621	60	80 P(IJK)=0.1666667*(P(IPJK)+P(IUPK)+P(IJKP)+P(IMJK)+P(IJMK)+P(IJKM))	PETACAL	59
2622	61	GO TO 100	PETACAL	60
2623	62	90 CONTINUE	PETACAL	61
2624	63	C	PETACAL	62
2625	64	C * * CALCULATE PETA	PETACAL	63
2626	65	C	PETACAL	64
2627	66	LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1	PETACAL	65
2628	67	SDIS=F(IJK)*DSUR+F(LMN)*DNBR*0.5	PETACAL	66
2629	68	DCNT=0.5*(DSUR+DNBR)	PETACAL	67
2630	69	SDIS=AMAX1(SDIS,0.5*DCNT)	PETACAL	68
2631	70	PETA(IJK)=DCNT/SDIS	PETACAL	69
2632	71	IF (BETA(LMN).LT.0.0.OR.NF(LMN).NE.0) PETA(IJK)=1.0	PETACAL	70
2633	72	IF (PETA(IJK).GT.2.0) PETA(IJK)=2.0	PETACAL	71
2634	73	IF (PETA(IJK).LT.0.0) PETA(IJK)=0.0	PETACAL	72
2635	74	100 CONTINUE	PETACAL	73
2636	75	C	PETACAL	74
2637	76	C * * CALCULATE PETA IN NEIGHBORING INTERPOLATION CELLS	PETACAL	75
2638	77	C	PETACAL	76
2639	78	CALL PETASET	PETACAL	77
2640	79	C	PETACAL	78

```

2641 80      IF (CYCLE.GT.0) CALL PRESCK          PETACAL 79
2642 81 C
2643 82      RETURN                           PETACAL 80
2644 83      END                               PETACAL 81
2645 1 *DK PCAL                           PETACAL 82
2646 2      SUBROUTINE PCAL                   PCAL   1
2647 3 *CA SLCOM1                         PCAL   2
2648 4 C
2649 5 C * * DETERMINE PROVISIONAL NF VALUES PCAL   3
2650 6 C
2651 7      DO 10 K=1,KMAX                  PCAL   4
2652 8      DO 10 J=1,JMAX                  PCAL   5
2653 9      DO 10 I=1,IMAX                  PCAL   6
2654 10     CALL IJKONLY                   PCAL   7
2655 11     NF(IJK)=7                     PCAL   8
2656 12     IF (I.EQ.1.OR.I.EQ.IMAX.OR.J.EQ.1.OR.J.EQ.JMAX.OR.K.EQ.1.OR.K.EQ.
2657 13     1 .KMAX.OR.BETA(IJK).LT.0.0) NF(IJK)=0 PCAL   9
2658 14 10 CONTINUE                      PCAL 10
2659 15     DO 160 K=2,KM1                 PCAL 11
2660 16     DO 160 J=2,JM1                 PCAL 12
2661 17     DO 160 I=2,IM1                 PCAL 13
2662 18     CALL CALCijk                   PCAL 14
2663 19     IF (BETA(IJK).LT.0.0) GO TO 20 PCAL 15
2664 20     IF (F(IJK).GT.EMF1.OR.AC(IJK).LT.EMF) GO TO 20 PCAL 16
2665 21     IF (F(IJK).GT.EMF) GO TO 30    PCAL 17
2666 22     NF(IJK)=8                     PCAL 18
2667 23     NFP(IJK)=NF(IJK)               PCAL 19
2668 24     GO TO 160                     PCAL 20
2669 25 20 NF(IJK)=0                   PCAL 21
2670 26     NFP(IJK)=NF(IJK)             PCAL 22
2671 27     GO TO 160                     PCAL 23
2672 28 C
2673 29 C      LOOKING ONLY AT PARTIALLY FILLED CELLS TO FIND SURFACE CELLS PCAL 24
2674 30 C      DETERMINE NF VALUE FOR PROVISIONAL SURFACE ORIENTATION PCAL 25
2675 31 C
2676 32 30 FXM=F(IJK)                  PCAL 26
2677 33 FXP=F(IJK)                  PCAL 27
2678 34 FZM=F(IJK)                  PCAL 28
2679 35 FZP=F(IJK)                  PCAL 29
2680 36 FYP=F(IJK)                  PCAL 30
2681 37 FYM=F(IJK)                  PCAL 31
2682 38 IF (CYL.EQ.1.0.AND.I.EQ.2) IMJK=NQ*(II5*(K-1)+IMAX*(JDP(J)-1)+(I-1
2683 39 1 ))+1                    PCAL 32
2684 40 IF (AR(IMJK).GT.EMF) FXM=F(IMJK) PCAL 33
2685 41 IF (AR(IJK).GT.EMF) FXP=F(IPJK) PCAL 34
2686 42 IF (AT(IJKM).GT.EMF) FZM=F(IJKM) PCAL 35
2687 43 IF (AT(IJK).GT.EMF) FZP=F(IJKP) PCAL 36
2688 44 IF (ABK(IJK).GT.EMF) FYP=F(IJPK) PCAL 37
2689 45 IF (ABK(IUMK).GT.EMF) FYM=F(IUMK) PCAL 38
2690 46 C
2691 47 C      INITIALIZE 3 INDICES FOR NF DECISIONS AND 3 VARIABLES FOR PCAL 39
2692 48 C      VOLUME OF FLUID CALCULATIONS                   PCAL 40
2693 49 C
2694 50     MOBS=1                      PCAL 41
2695 51     INF=1                       PCAL 42
2696 52     IOBS=1                      PCAL 43
2697 53     VF=0.0                      PCAL 44
2698 54     VFXM=0.0                    PCAL 45
2699 55     VFXP=0.0                    PCAL 46
2700 56 C
2701 57 C      SET UP DO LOOPS FOR VOLUME OF FLUID CALCULATIONS AT XM AND XP PCAL 47
2702 58 C
2703 59     DO 50 KK=1,3                 PCAL 48
2704 60     N=K-2+KK                  PCAL 49
2705 61     DO 40 JJ=1,3                 PCAL 50
2706 62     M=J-2+JJ                  PCAL 51
2707 63     LMN=NQ*(IMAX*JMAX*(N-1)+IMAX*(M-1)+(I-1))+1 PCAL 52
2708 64     LMNM=LMN-NQ                PCAL 53
2709 65     LMNP=LMN+NQ                PCAL 54
2710 66     VFXM=VFXM+F(LMN)           PCAL 55
2711 67     VFXP=VFXP+F(LMNP)          PCAL 56
2712 68 40 CONTINUE                  PCAL 57
2713 69 50 CONTINUE                  PCAL 58
2714 70 C
2715 71 C      CALCULATE NF=1 AND RESET INDICES IF APPROPRIATE PCAL 59
2716 72 C
2717 73     IF (BETA(IMJK).LT.0.0) IOBS=2 PCAL 60
2718 74     MOBS=MOBS+(IOBS-1)          PCAL 61
2719 75     IF (IOBS.EQ.2) GO TO 60   PCAL 62
2720 76     IF (FXM.LT.EMF) GO TO 60  PCAL 63

```

```

2721 77      INF=INF+1                                PCAL    77
2722 78      IF (VFXM.GT.VF) NF(IJK)=1                PCAL    78
2723 79      IF (NF(IJK).EQ.1) VF=VFXM                PCAL    79
2724 80      60 CONTINUE                               PCAL    80
2725 81      IOBS=1                                  PCAL    81
2726 82 C     CALCULATE NF=2 AND RESET INDICES IF APPROPRIATE PCAL    82
2727 83 C     IF (BETA(IPJK).LT.O.O) IOBS=2          PCAL    83
2728 84 C     MOBS=MOBS+(IOBS-1)                      PCAL    84
2729 85      IF (IOBS.EQ.2) GO TO 70                  PCAL    85
2730 86      IF (FXP.LT.EMF) GO TO 70                  PCAL    86
2731 87      INF=INF+1                                PCAL    87
2732 88      IF (VFXP.GT.VF) NF(IJK)=2                PCAL    88
2733 89      IF (NF(IJK).EQ.2) VF=VFXP                PCAL    89
2734 90      70 CONTINUE                               PCAL    90
2735 91      IOBS=1                                  PCAL    91
2736 92      70 CONTINUE                               PCAL    92
2737 93      IOBS=1                                  PCAL    93
2738 94 C     INITIALIZE 2 VARIABLES FOR VOLUME OF FLUID CALCULATIONS PCAL    94
2739 95 C     VFZM=O.O                                PCAL    95
2740 96 C     VFZP=O.O                                PCAL    96
2741 97      VFZM=O.O                                PCAL    97
2742 98      VFZP=O.O                                PCAL    98
2743 99 C     SET UP DO LOOPS FOR VOLUME OF FLUID CALCULATIONS AT ZM AND ZP PCAL    99
2744 100 C    DO 90 II=1,3                            PCAL   100
2745 101 C    L=I-2+II                             PCAL   101
2746 102      DO 80 JJ=1,3                            PCAL   102
2747 103      L=M-J+JJ                             PCAL   103
2748 104      LMN=NQ*(IMAX*JMAX*(K-1)+IMAX*(M-1)+(L-1))+1 PCAL   104
2749 105      LMNM=LMN-II2                           PCAL   105
2750 106      LMNP=LMN+II2                           PCAL   106
2751 107      VFZM=VFZM+F(LMNM)                     PCAL   107
2752 108      VFZP=VFZP+F(LMNP)                     PCAL   108
2753 109      VFZM=VFZM+F(LMNM)                     PCAL   109
2754 110      VFZP=VFZP+F(LMNP)                     PCAL   110
2755 111      80 CONTINUE                            PCAL   111
2756 112      90 CONTINUE                            PCAL   112
2757 113 C     CALCULATE NF=5 AND RESET INDICES IF APPROPRIATE PCAL   113
2758 114 C     IF (BETA(IJGM).LT.O.O) IOBS=2          PCAL   114
2759 115 C     MOBS=MOBS+(IOBS-1)                      PCAL   115
2760 116      IF (IOBS.EQ.2) GO TO 100               PCAL   116
2761 117      IF (FZM.LT.EMF) GO TO 100               PCAL   117
2762 118      INF=INF+1                                PCAL   118
2763 119      IF (VFZM.GT.VF) NF(IJK)=5                PCAL   119
2764 120      IF (NF(IJK).EQ.5) VF=VFZM                PCAL   120
2765 121      100 CONTINUE                            PCAL   121
2766 122      IOBS=1                                  PCAL   122
2767 123      100 CONTINUE                            PCAL   123
2768 124      IOBS=1                                  PCAL   124
2769 125 C     CALCULATE NF=6 AND RESET INDICES IF APPROPRIATE PCAL   125
2770 126 C     IF (BETA(IJKP).LT.O.O) IOBS=2          PCAL   126
2771 127 C     MOBS=MOBS+(IOBS-1)                      PCAL   127
2772 128      IF (IOBS.EQ.2) GO TO 110               PCAL   128
2773 129      IF (FZP.LT.EMF) GO TO 110               PCAL   129
2774 130      INF=INF+1                                PCAL   130
2775 131      IF (VFZP.GT.VF) NF(IJK)=6                PCAL   131
2776 132      IF (NF(IJK).EQ.6) VF=VFZP                PCAL   132
2777 133      110 CONTINUE                            PCAL   133
2778 134      IOBS=1                                  PCAL   134
2779 135      110 CONTINUE                            PCAL   135
2780 136      IOBS=1                                  PCAL   136
2781 137 C     INITIALIZE 2 VARIABLES FOR VOLUME OF FLUID CALCULATIONS PCAL   137
2782 138 C     VFYM=O.O                                PCAL   138
2783 139 C     VFYP=O.O                                PCAL   139
2784 140      VFYM=O.O                                PCAL   140
2785 141      VFYP=O.O                                PCAL   141
2786 142 C     SET UP DO LOOPS FOR VOLUME OF FLUID CALCULATIONS AT YM AND YP PCAL   142
2787 143 C     DO 130 KK=1,3                            PCAL   143
2788 144 C     N=K-2+KK                             PCAL   144
2789 145      DO 120 II=1,3                            PCAL   145
2790 146      L=I-2+II                             PCAL   146
2791 147      LMN=NQ*(IMAX*JMAX*(N-1)+IMAX*(J-1)+(L-1))+1 PCAL   147
2792 148      LMNM=LMN-II1                           PCAL   148
2793 149      LMNP=LMN+II1                           PCAL   149
2794 150      VFYM=VFYM+F(LMNM)                     PCAL   150
2795 151      VFYP=VFYP+F(LMNP)                     PCAL   151
2796 152      VFYM=VFYM+F(LMNM)                     PCAL   152
2797 153      VFYP=VFYP+F(LMNP)                     PCAL   153
2798 154      120 CONTINUE                            PCAL   154
2799 155      130 CONTINUE                            PCAL   155
2800 156 C

```

```

2801 157 C      CALCULATE NF=3 AND RESET INDICES IF APPROPRIATE      PCAL    157
2802 158 C      IF (BETA(IJMK).LT.0.0) IOBS=2                      PCAL    158
2803 159         MOBS=MOBS+(IOBS-1)                                PCAL    159
2804 160         IF (IOBS.EQ.2) GO TO 140                           PCAL    160
2805 161         IF (FYM.LT.EMF) GO TO 140                           PCAL    161
2806 162         INF=INF+1                                         PCAL    162
2807 163         IF (VFYM.GT.VF) NF(IJK)=3                         PCAL    163
2808 164         IF (NF(IJK).EQ.3) VF=VFYM                          PCAL    164
2809 165         140 CONTINUE                                     PCAL    165
2810 166         IOBS=1                                         PCAL    166
2811 167
2812 168 C      CALCULATE NF=4 AND RESET INDICES IF APPROPRIATE      PCAL    168
2813 169 C      IF (BETA(IJPK).LT.0.0) IOBS=2                      PCAL    169
2814 170 C      MOBS=MOBS+(IOBS-1)                                PCAL    170
2815 171         IF (IOBS.EQ.2) GO TO 150                           PCAL    171
2816 172         IF (FYP.LT.EMF) GO TO 150                           PCAL    172
2817 173         INF=INF+1                                         PCAL    173
2818 174         IF (VFYP.GT.VF) NF(IJK)=4                         PCAL    174
2819 175         IF (NF(IJK).EQ.4) VF=VFYP                          PCAL    175
2820 176         150 CONTINUE                                     PCAL    176
2821 177
2822 178
2823 179 C      IF WE DO NOT RESET NF VALUE DEFAULT CORRECTLY GIVES ISOLATED      PCAL    179
2824 180 C      CELL : NF=7                                       PCAL    180
2825 181 C
2826 182 C
2827 183 C      IF WE DO RESET NF VALUE WE WILL HAVE CORRECT NF VALUE - EXCEPT      PCAL    183
2828 184 C      WHEN THERE IS FLUID IN EACH NHBR CELL THAT IS NOT AN OBSTACLE      PCAL    184
2829 185 C      CELL                                         PCAL    185
2830 186 C
2831 187 C      IN THIS EXCEPTIONAL CASE WE SHOULD HAVE NF=0             PCAL    187
2832 188 C
2833 189 C      THIS IS DONE BY DEFINING A CRITICAL VALUE FOR INF AND          PCAL    189
2834 190 C      TESTING ON THAT VALUE                               PCAL    190
2835 191 C
2836 192         INFCR=8-MOBS                                PCAL    192
2837 193         IF (INF.EQ.INFCR.AND.INFCR.GT.1) NF(IJK)=0          PCAL    193
2838 194 C
2839 195         NFP(IJK)=NF(IJK)                            PCAL    194
2840 196 C
2841 197         160 CONTINUE                                     PCAL    195
2842 198         RETURN                                         PCAL    196
2843 199         END                                           PCAL    197
2844 1 *DK PETASET
2845 2           SUBROUTINE PETASET
2846 3 *CA SLCOM1
2847 4 C
2848 5 C * * CALCULATE PETA IN NEIGHBORING INTERPOLATION CELLS      PETASET  1
2849 6 C
2850 7           DO 90 K=1,KMAX                                PETASET  2
2851 8           DO 90 J=1,JMAX                                PETASET  3
2852 9           DO 90 I=1,IMAX                                PETASET  4
2853 10          CALL CALCIJK                                PETASET  5
2854 11          NFF=NF(IJK)                                PETASET  6
2855 12          IF (NFF.EQ.0.OR.BETA(IJK).LT.0.0) GO TO 90      PETASET  7
2856 13          IF (NFF.GT.7) GO TO 80                         PETASET  8
2857 14          L=I                                         PETASET  9
2858 15          M=J                                         PETASET 10
2859 16          N=K                                         PETASET 11
2860 17          GO TO (10,20,30,40,50,60,90), NFF          PETASET 12
2861 18          10 L=I-1                                    PETASET 13
2862 19          AMN=AR(IMJK)                                PETASET 14
2863 20          DNBR=DELX(L)                                PETASET 15
2864 21          DCNT=0.5*(DNBR+DELX(I))                  PETASET 16
2865 22          GO TO 70                                    PETASET 17
2866 23          20 L=I+1                                    PETASET 18
2867 24          AMN=AR(IJK)                                PETASET 19
2868 25          DNBR=DELX(L)                                PETASET 20
2869 26          DCNT=0.5*(DNBR+DELX(I))                  PETASET 21
2870 27          GO TO 70                                    PETASET 22
2871 28          30 M=J-1                                    PETASET 23
2872 29          AMN=ABK(IJMK)                                PETASET 24
2873 30          DNBR=DELY(M)/RRI(I)                          PETASET 25
2874 31          DCNT=0.5*(DNBR+DELY(J)/RRI(I))            PETASET 26
2875 32          GO TO 70                                    PETASET 27
2876 33          40 M=J+1                                    PETASET 28
2877 34          AMN=ABK(IJK)                                PETASET 29
2878 35          DNBR=DELY(M)/RRI(I)                          PETASET 30
2879 36          DCNT=0.5*(DNBR+DELY(J)/RRI(I))            PETASET 31
2880 37          GO TO 70                                    PETASET 32

```

```

2881 38      50 N=K-1                                PETASET 38
2882 39      AMN=AT(IJKM)                            PETASET 39
2883 40      DNBR=DELZ(N)                            PETASET 40
2884 41      DCNT=0.5*(DNBR+DELZ(K))                PETASET 41
2885 42      GO TO 70                               PETASET 42
2886 43      60 N=K+1                                PETASET 43
2887 44      AMN=AT(IJK)                            PETASET 44
2888 45      DNBR=DELZ(N)                            PETASET 45
2889 46      DCNT=0.5*(DNBR+DELZ(K))                PETASET 46
2890 47      70 CONTINUE                            PETASET 47
2891 48      LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1 PETASET 48
2892 49      IF (NF(LMN).GT.O.OR.AMN.LT.EM6) GO TO 90 PETASET 49
2893 50      BPD=1.0/PETA(LMN)-BETA(LMN)*(1.0-PETA(IJK))*AMN/AC(LMN)*DELT/(DNBR PETASET 50
2894 51      1 *DCNT)                                PETASET 51
2895 52      PETA(LMN)=AMIN1(1.0/BPD,1.98/OMG)        PETASET 52
2896 53      GO TO 90                               PETASET 53
2897 54      80 CONTINUE                            PETASET 54
2898 55 C   * * SET VOID REGION NFF PRESSURE INTO CELL IJK PETASET 55
2899 56 C   * * SET VOID REGION NFF PRESSURE INTO CELL IJK PETASET 56
2900 57 C
2901 58      P(IJK)=PR(NFF)                          PETASET 57
2902 59      90 CONTINUE                            PETASET 59
2903 60 C
2904 61      RETURN                                PETASET 60
2905 62      END                                    PETASET 62
2906 1 *DK PRESCK                                PRESCK 1
2907 2   SUBROUTINE PRESCK                         PRESCK 2
2908 3 C
2909 4 *CA SLCOM1                                PRESCK 3
2910 5   DATA PSAT /O.O/                           PRESCK 4
2911 6 C
2912 7      DO 150 K=2,KM1                          PRESCK 5
2913 8      DO 150 J=2,JM1                          PRESCK 6
2914 9      DO 150 I=2,IM1                          PRESCK 7
2915 10     CALL IJKAJCT                           PRESCK 8
2916 11     IF (AC(IJK).LT.EM6.OR.BETA(IJK).LE.O.O) GO TO 130 PRESCK 9
2917 12     NFF=NF(IJK)                            PRESCK 10
2918 13     NFFO=NFO(IJK)                           PRESCK 11
2919 14     IF (NFF.EQ.NFFO) GO TO 150             PRESCK 12
2920 15     IF (NFF.EQ.O) GO TO 140               PRESCK 13
2921 16     IF (NFF.GT.7) GO TO 100               PRESCK 14
2922 17     L=I                                  PRESCK 15
2923 18     M=J                                  PRESCK 16
2924 19     N=K                                  PRESCK 17
2925 20     GO TO (10,20,30,40,50,60,70), NFF    PRESCK 18
2926 21     10 L=I-1                             PRESCK 19
2927 22     GO TO 80                             PRESCK 20
2928 23     20 L=I+1                             PRESCK 21
2929 24     GO TO 80                             PRESCK 22
2930 25     30 M=J-1                             PRESCK 23
2931 26     GO TO 80                             PRESCK 24
2932 27     40 M=J+1                             PRESCK 25
2933 28     GO TO 80                             PRESCK 26
2934 29     50 N=K-1                             PRESCK 27
2935 30     GO TO 80                             PRESCK 28
2936 31     60 N=K+1                             PRESCK 29
2937 32     GO TO 80                             PRESCK 30
2938 33     70 GO TO 90                           PRESCK 31
2939 34     80 LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1 PRESCK 32
2940 35     NFEL=NF(IMJK)                          PRESCK 33
2941 36     NFER=NF(IPJK)                          PRESCK 34
2942 37     NFEF=NF(IUMK)                          PRESCK 35
2943 38     NFEBK=NFBK(IUPK)                      PRESCK 36
2944 39     NFEB=NF(IJKM)                          PRESCK 37
2945 40     NFET=NF(IJKP)                          PRESCK 38
2946 41     NFE=MAXO(NFEL,NFER,NFEBK,NFEF,NFEB,NFET,1) PRESCK 39
2947 42     PSURF=PS(IJK)+PR(NFE)                 PRESCK 40
2948 43     PLMN=P(LMN)                           PRESCK 41
2949 44     IF (NF(LMN).NE.O.AND.BETA(IJK).GT.O.O) PLMN=PSURF PRESCK 42
2950 45     P(IJK)=(1.0-PETA(IJK))*PLMN+PETA(IJK)*PSURF PRESCK 43
2951 46     GO TO 150                           PRESCK 44
2952 47     90 IF (PSAT.LE.O.O) GO TO 120          PRESCK 45
2953 48     IF (F(IJK).LT.EMF1) GO TO 110          PRESCK 46
2954 49     PMPS=O.O                            PRESCK 47
2955 50     IF (PN(IJK).LT.PSAT) PMPS=P(IJK)-PSAT PRESCK 48
2956 51     DIJK=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR(I-1 PRESC 49
2957 52     1 ))+RDY(J)*(V(IJK)*ABK(IJK)-V(IUMK)*ABK(IUMK)))+RDZ(K)*(W(IJK)*AT PRESC 50
2958 53     2 (IJK)-W(IJKM)*AT(IJKM)))/AC(IJK)       PRESC 51
2959 54     DIJK=DIJK-PMPS**2/DELT                  PRESC 52
2960 55     P(IJK)=PN(IJK)-DIJK*BETA(IJK)/(1.0-2.0*PMPS*BETA(IJK)/DELT)    PRESC 53

```

```

2961 56      GO TO 150                               PRESCK 56
2962 57      100 P(IJK)=PR(NFF)                   PRESCK 57
2963 58      GO TO 150                               PRESCK 58
2964 59      110 P(IJK)=PSAT                      PRESCK 59
2965 60      GO TO 150                               PRESCK 60
2966 61      120 P(IJK)=PN(IJK)                   PRESCK 61
2967 62      GO TO 150                               PRESCK 62
2968 63      130 P(IJK)=O.O                      PRESCK 63
2969 64      140 CONTINUE                         PRESCK 64
2970 65      150 CONTINUE                         PRESCK 65
2971 66 C
2972 67      IF (CYCLE.LT.9999) RETURN             PRESCK 66
2973 68 C
2974 69      DO 160 K=2,KM1                         PRESCK 67
2975 70      DO 160 J=2,JM1                         PRESCK 68
2976 71      DO 160 I=2,IM1                         PRESCK 69
2977 72      CALL IJKAJCT                          PRESCK 70
2978 73      IF (AC(IJK).LT.EM6.OR.BETA(IJK).LE.O.O) GO TD 160
2979 74      NFF=NF(IJK)                           PRESCK 71
2980 75      NFFO=NFO(IJK)                          PRESCK 72
2981 76      IF (NFF.EQ.NFFD) GO TD 160             PRESCK 73
2982 77      WRITE (9,170) CYCLE,I,J,K,NF(IJK),NFO(IJK),P(IJK),PN(IJK),U
2983 78      1 (IJK),U(IMJK),V(IJK),V(IUMK),W(IJK),W(IJMK)          PRESCK 74
2984 79      160 CONTINUE                         PRESCK 75
2985 80      RETURN                                PRESCK 76
2986 81 C
2987 82      170 FORMAT (* NF SWITCH AFTER PETACAL *,6I4/9(2X,1PE12.5)) PRESCK 77
2988 83      END                                     PRESCK 78
2989 1 *DK PLTPT
2990 2      SUBROUTINE PLTPT (XONE,YONE,ICHAR,ISYM) PLTPT 79
2991 3 *CA SLCOM1
2992 4      PARAMETER (IBAR2Q=2*IBAR2-2)           PLTPT 80
2993 5      COMMON BETAQ(IBAR2Q,KBAR2), FQ(IBAR2Q,KBAR2), ACQ(IBAR2Q,KBAR2),
2994 6      1 ARO(IBAR2Q,KBAR2), ATQ(IBAR2Q,KBAR2), UQ(IBAR2Q,KBAR2), VQ(IBAR2Q
2995 7      ,KBAR2), XQ(IBAR2Q), XIQ(IBAR2Q), YQ(KBAR2), YUQ(KBAR2), IM1Q,
2996 8      3 JM1Q, UBAR2Q, UMAXQ, IMAXQ, SF, XSHFT, YSHFT, DELXQ(IBAR2Q),
2997 9      4 DELYQ(KBAR2), XMINQ, XMAXQ, YMINQ, YMAXQ
2998 10 C
2999 11 C +++ PLOT (DRAW) A POINT                PLTPT 81
3000 12 C +++ PROVIDES A SYSTEM DEPENDANT CALL   PLTPT 82
3001 13 C
3002 14      IC=0                                 PLTPT 83
3003 15      X1=XONE                            PLTPT 84
3004 16      Y1=YONE                            PLTPT 85
3005 17      X01=(X1-XMINQ)*SF+XSHFT            PLTPT 86
3006 18      Y01=(Y1-YMINQ)*SF+YSHFT            PLTPT 87
3007 19      IX1=16.+900.0*X01                  PLTPT 88
3008 20      IY1=16.+900.0*(1.0-Y01)            PLTPT 89
3009 21      CALL PLT (IX1,IY1,42)               PLTPT 90
3010 22      IF (ABS(X1).LE.EM6) GO TD 10        PLTPT 91
3011 23      10 RETURN                           PLTPT 92
3012 24      END                                   PLTPT 93
3013 1 *DK PRESCR
3014 2      SUBROUTINE PRESCR                  PRESCR 94
3015 3 *CA SLCOM1
3016 4 C
3017 5      DIMENSION FF(IBAR2*UBAR2*KBAR2), AF(IBAR2*UBAR2*KBAR2), AD(IBAR2
3018 6      1 *UBAR2*KBAR2), QQ(IBAR2*UBAR2*KBAR2), VLM(IBAR2*UBAR2*KBAR2), SSM
3019 7      2 (IBAR2*UBAR2*KBAR2), RXR(IBAR2*UBAR2*KBAR2), RXL(IBAR2*UBAR2
3020 8      ,KBAR2), RZT(IBAR2*UBAR2*KBAR2), RZB(IBAR2*UBAR2*KBAR2), RYBK
3021 9      4 (IBAR2*UBAR2*KBAR2), RYF(IBAR2*UBAR2*KBAR2), DIV(IBAR2*UBAR2
3022 10     ,KBAR2), DP(IBAR2*UBAR2*KBAR2), CQ(IBAR2*UBAR2*KBAR2)
3023 11 C
3024 12     DATA ADEFM, BDEFM /100.0.0.10/
3025 13     DATA ITMAX, ITMIN, ITMOST /500,5,90/
3026 14     DATA KTRAN /1/
3027 15     DATA PSAT /0.0/
3028 16 C
3029 17 C      THIS SUBROUTINE MUST BE MODIFIED FOR BOUNDARY CONDITIONS
3030 18 C      OTHER THAN NORMAL VELOCITY EQUAL TO ZERO
3031 19 C
3032 20      GO TO (10,200), KTRAN              PRESCR 17
3033 21      10 KTRAN=2                         PRESCR 18
3034 22 C
3035 23     LVEC=II5*KMAX                      PRESCR 19
3036 24     IF (LVEC.NE.IBAR2*UBAR2*KBAR2) GO TD 760
3037 25     EPSIP=2.0*EPSI                      PRESCR 20
3038 26     EPSIM=0.5*EPSI                     PRESCR 21
3039 27     DO 20 K=1,KMAX                      PRESCR 22
3040 28     DO 20 J=1,UMAX                      PRESCR 23

```

```

3041 29      DD 20 I=1,IMAX          PRESCR   29
3042 30      CALL IJKONLY        PRESCR   30
3043 31      LIJK=II5*(K-1)+IMAX*(J-1)+I    PRESCR   31
3044 32      VLM(LIJK)=AC(IJK)*DELX(I)*DELZ(K)*DELY(J)/RRI(I)  PRESCR   32
3045 33      20 CONTINUE          PRESCR   33
3046 34 C     IF (CYCLE.GT. 1) GO TO 200  PRESCR   34
3047 35      IF (GZ.EQ.O.O) GO TO 40       PRESCR   35
3048 36      DO 30 J=2,JM1          PRESCR   36
3049 37      DO 30 I=2,IM1          PRESCR   37
3050 38      PSADD=O.O          PRESCR   38
3051 39      PSADD=O.O          PRESCR   39
3052 40      DO 30 KK=2,KM1          PRESCR   40
3053 41      K=KM1-KK+2          PRESCR   41
3054 42      CALL IJKONLY          PRESCR   42
3055 43      IF (PS(IJK).NE.O.O) PSADD=PS(IJK)-P(IJK)  PRESCR   43
3056 44      P(IJK)=P(IJK)+PSADD    PRESCR   44
3057 45      30 CONTINUE          PRESCR   45
3058 46 C     40 DO 190 K=1,KMAX          PRESCR   46
3059 47      DO 190 J=1,JMAX          PRESCR   47
3060 48      DO 190 I=1,IMAX          PRESCR   48
3061 49      CALL IJKAJCT        PRESCR   49
3062 50      LIJK=II5*(K-1)+IMAX*(J-1)+I    PRESCR   50
3063 51      VLM(LIJK)=AC(IJK)*DELX(I)*DELZ(K)*DELY(J)/RRI(I)  PRESCR   51
3064 52      IF (AC(IJK).LT.EM6.DR.BETA(IJK).LE.O.O) GO TO 170  PRESCR   52
3065 53      NFF=NF(IJK)          PRESCR   53
3066 54      IF (NFF.EQ.O) GO TO 180    PRESCR   54
3067 55      IF (NFF.GT.7) GO TO 140    PRESCR   55
3068 56      L=I          PRESCR   56
3069 57      M=J          PRESCR   57
3070 58      N=K          PRESCR   58
3071 59      GO TO (50,60,70,80,90,100,110), NFF  PRESCR   59
3072 60      50 L=I-1          PRESCR   60
3073 61      GO TO 120          PRESCR   61
3074 62      60 L=I+1          PRESCR   62
3075 63      GO TO 120          PRESCR   63
3076 64      70 M=J-1          PRESCR   64
3077 65      GO TO 120          PRESCR   65
3078 66      80 M=J+1          PRESCR   66
3079 67      GO TO 120          PRESCR   67
3080 68      90 N=K-1          PRESCR   68
3081 69      GO TO 120          PRESCR   69
3082 70      100 N=K+1         PRESCR   70
3083 71      GO TO 120          PRESCR   71
3084 72      110 GO TO 130        PRESCR   72
3085 73      120 LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1  PRESCR   73
3086 74      NFEL=NF(IMJK)        PRESCR   74
3087 75      NFER=NF(IPJK)        PRESCR   75
3088 76      NFEF=NF(IUMK)        PRESCR   76
3089 77      NFEBK=NF(IJPK)        PRESCR   77
3090 78      NFEB=NF(IJKM)        PRESCR   78
3091 79      NFET=NF(IJKP)        PRESCR   79
3092 80      NFE=MAX0(NFEL,NFER,NFEBK,NFEF,NFEB,NFET,1)  PRESCR   80
3093 81      PSURF=PS(IJK)+PR(NFE)  PRESCR   81
3094 82      PLMN=P(LMN)          PRESCR   82
3095 83      IF (NF(LMN).NE.O.AND.BETA(IJK).GT.O.O) PLMN=PSURF  PRESCR   83
3096 84      P(IJK)=(1.O-PETA(IJK))*PLMN+PETA(IJK)*PSURF  PRESCR   84
3097 85      GO TO 180          PRESCR   85
3098 86      130 IF (PSAT.LE.O.O) GO TO 160    PRESCR   86
3099 87      IF (F(IJK).LT.EMF1) GO TO 150    PRESCR   87
3100 88      PMPS=O.O          PRESCR   88
3101 89      IF (PN(IJK).LT.PSAT) PMPS=P(IJK)-PSAT    PRESCR   89
3102 90      DIJK=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR(I-1
3103 91      1 ))+RDY(J)*(V(IJK)*ABK(IJK)-V(IUMK)*ABK(IUMK)))+RDZ(K)*(W(IJK)*AT
3104 92      2 (IJK)-W(IJKM)*AT(IJKM)))/AC(IJK)    PRESCR   92
3105 93      DIJK=DIJK-PMPS**2/DELT    PRESCR   93
3106 94      P(IJK)=PN(IJK)-DIJK*BETA(IJK)/(1.O-2.O*PMPS*BETA(IJK)/DELT)  PRESCR   94
3107 95      GO TO 180          PRESCR   95
3108 96      140 P(IJK)=PR(NFF)    PRESCR   96
3109 97      GO TO 180          PRESCR   97
3110 98      150 P(IJK)=PSAT    PRESCR   98
3111 99      GO TO 180          PRESCR   99
3112 100     160 P(IJK)=PN(IJK)    PRESCR  100
3113 101     GO TO 180          PRESCR  101
3114 102     170 P(IJK)=O.O      PRESCR  102
3115 103     PN(IJK)=P(IJK)      PRESCR  103
3116 104     180 PN(IJK)=P(IJK)    PRESCR  104
3117 105     190 CONTINUE          PRESCR  105
3118 106 C     CALL BC          PRESCR  106
3119 107     PRESCR  107
3120 108 C     PRESCR  108

```

```

3121 109 200 IF (CYCLE.LT.1) RETURN          PRESCR 109
3122 110 C                                     PRESCR 110
3123 111 C1=1.0                                PRESCR 111
3124 112 DD=1.0                                PRESCR 112
3125 113 DO 360 K=1,KMAX                      PRESCR 113
3126 114 DO 360 J=1,JMAX                      PRESCR 114
3127 115 DO 360 I=1,IMAX                      PRESCR 115
3128 116 CALL IJKAJCT                         PRESCR 116
3129 117 LIJK=I15*(K-1)+IMAX*(J-1)+I           PRESCR 117
3130 118 IF (AC(IJK).LT.EM6.OR.BETA(IJK).LE.O.O) GO TO 340 PRESCR 118
3131 119 IF (NF(IJK).GT.0) GO TO 350          PRESCR 119
3132 120 DIJK=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR(I-1
1 ))+RDY(J)*(V(IJK)*ABK(IJK)-V(IJKM)*ABK(IJKM)))+RDZ(K)*(W(IJK)*AT
2 (IJK)-W(IJKM)*AT(IJKM)))/AC(IJK)          PRESCR 120
3133 121                                         PRESCR 121
3134 122                                         PRESCR 122
3135 123 C                                     PRESCR 123
3136 124 IF (IDEFM.EQ.O.OR.F(IJK).GE.EMF1) GO TO 210 PRESCR 124
3137 125 C                                     PRESCR 125
3138 126 DIJK=DIJK+AMIN1(ADEFM*EPSI,BDEFM*(1.0-F(IJK))/DELT) PRESCR 126
3139 127 C                                     PRESCR 127
3140 128 210 DIV(LIJK)=DIJK*VLM(LIJK)        PRESCR 128
3141 129 C                                     PRESCR 129
3142 130 SSM(LIJK)=O.O                        PRESCR 130
3143 131 RXR(LIJK)=O.O                        PRESCR 131
3144 132 RXRA=O.O                            PRESCR 132
3145 133 RXRB=O.O                            PRESCR 133
3146 134 IF (AR(IJK).LT.EM6) GO TO 230        PRESCR 134
3147 135 RXR(LIJK)=DELZ(K)*DELY(J)*RDXP(I)*AR(IJK)/RR(I) PRESCR 135
3148 136 SSM(LIJK)=SSM(LIJK)+DELT*RXR(LIJK)    PRESCR 136
3149 137 IF (NF(IPJK).EQ.O) GO TO 230          PRESCR 137
3150 138 IF (NF(IPJK).GT.7) GO TO 220          PRESCR 138
3151 139 IF (NF(IPJK).NE.1) GO TO 220          PRESCR 139
3152 140 RXRA=(1.0-PETA(IPJK))*RXR(LIJK)      PRESCR 140
3153 141 RXRB=((1.0-PETA(IPJK))*PN(IJK)+PETA(IPJK)*(PS(IPJK)+PSAT)-PN(IPJK)
1 )*RXR(LIJK)                                PRESCR 141
3154 142                                         PRESCR 142
3155 143 GO TO 220                           PRESCR 143
3156 144 220 RXR(LIJK)=O.O                  PRESCR 144
3157 145 230 RXL(LIJK)=O.O                  PRESCR 145
3158 146 RXLA=O.O                            PRESCR 146
3159 147 RXLB=O.O                            PRESCR 147
3160 148 IF (AR(IMJK).LT.EM6) GO TO 250        PRESCR 148
3161 149 RXL(LIJK)=DELZ(K)*DELY(J)*RDXP(I-1)*AR(IMJK)/RR(I-1) PRESCR 149
3162 150 SSM(LIJK)=SSM(LIJK)+DELT*RXL(LIJK)    PRESCR 150
3163 151 IF (NF(IMJK).EQ.O) GO TO 250          PRESCR 151
3164 152 IF (NF(IMJK).GT.7) GO TO 240          PRESCR 152
3165 153 IF (NF(IMJK).NE.2) GO TO 240          PRESCR 153
3166 154 RXLA=(1.0-PETA(IMJK))*RXL(LIJK)      PRESCR 154
3167 155 RXLB=((1.0-PETA(IMJK))*PN(IJK)+PETA(IMJK)*(PS(IMJK)+PSAT)-PN(IMJK)
1 )*RXL(LIJK)                                PRESCR 155
3168 156                                         PRESCR 156
3169 157 GO TO 240                           PRESCR 157
3170 158 240 RXL(LIJK)=O.O                  PRESCR 158
3171 159 250 RZT(LIJK)=O.O                  PRESCR 159
3172 160 RZTA=O.O                            PRESCR 160
3173 161 RZTB=O.O                            PRESCR 161
3174 162 IF (AT(IJK).LT.EM6) GO TO 270        PRESCR 162
3175 163 RZT(LIJK)=DELX(I)*DELY(J)/RRI(I)*RDZP(K)*AT(IJK) PRESCR 163
3176 164 SSM(LIJK)=SSM(LIJK)+DELT*RZT(LIJK)    PRESCR 164
3177 165 IF (NF(IJKP).EQ.O) GO TO 270          PRESCR 165
3178 166 IF (NF(IJKP).GT.7) GO TO 260          PRESCR 166
3179 167 IF (NF(IJKP).NE.5) GO TO 260          PRESCR 167
3180 168 RZTA=(1.0-PETA(IJKP))*RZT(LIJK)      PRESCR 168
3181 169 RZTB=((1.0-PETA(IJKP))*PN(IJK)+PETA(IJKP)*(PS(IJKP)+PSAT)-PN(IJKP)
1 )*RZT(LIJK)                                PRESCR 169
3182 170                                         PRESCR 170
3183 171 GO TO 260                           PRESCR 171
3184 172 260 RZT(LIJK)=O.O                  PRESCR 172
3185 173 270 RZB(LIJK)=O.O                  PRESCR 173
3186 174 RZBA=O.O                            PRESCR 174
3187 175 RZBB=O.O                            PRESCR 175
3188 176 IF (AT(IJKM).LT.EM6) GO TO 290        PRESCR 176
3189 177 RZB(LIJK)=DELX(I)*DELY(J)/RRI(I)*RDZP(K-1)*AT(IJKM) PRESCR 177
3190 178 SSM(LIJK)=SSM(LIJK)+DELT*RZB(LIJK)    PRESCR 178
3191 179 IF (NF(IJKM).EQ.O) GO TO 290          PRESCR 179
3192 180 IF (NF(IJKM).GT.7) GO TO 280          PRESCR 180
3193 181 IF (NF(IJKM).NE.6) GO TO 280          PRESCR 181
3194 182 RZBA=(1.0-PETA(IJKM))*RZB(LIJK)      PRESCR 182
3195 183 RZBB=((1.0-PETA(IJKM))*PN(IJK)+PETA(IJKM)*(PS(IJKM)+PSAT)-PN(IJKM)
1 )*RZB(LIJK)                                PRESCR 183
3196 184                                         PRESCR 184
3197 185 GO TO 280                           PRESCR 185
3198 186 280 RZB(LIJK)=O.O                  PRESCR 186
3199 187 290 RYBK(LIJK)=O.O                  PRESCR 187
3200 188 RYBKA=O.O                           PRESCR 188

```

```

3201 189 RYBKB=0.0 PRESCR 189
3202 190 IF (ABK(IJK).LT.EM6) GO TO 310 PRESCR 190
3203 191 RYBK(LIJK)=DELX(I)*DELZ(K)*RDYP(J)*RRI(I)*ABK(IJK) PRESCR 191
3204 192 SSM(LIJK)=SSM(LIJK)+DELT*RYBK(LIJK) PRESCR 192
3205 193 IF (NF(IJPK).EQ.0) GO TO 310 PRESCR 193
3206 194 IF (NF(IJPK).GT.7) GO TO 300 PRESCR 194
3207 195 IF (NF(IJPK).NE.3) GO TO 300 PRESCR 195
3208 196 RYBKA=(1.0-PETA(IJPK))*RYBK(LIJK) PRESCR 196
3209 197 RYBKB=((1.0-PETA(IJPK))*PN(IJK)+PETA(IJPK)*(PS(IJPK)+PSAT)-PN(IJPK) PRESCR 197
3210 198 1))*RYBK(LIJK) PRESCR 198
3211 199 GO TO 300 PRESCR 199
3212 200 300 RYBK(LIJK)=0.0 PRESCR 200
3213 201 310 RYF(LIJK)=0.0 PRESCR 201
3214 202 RYFA=0.0 PRESCR 202
3215 203 RYFB=0.0 PRESCR 203
3216 204 IF (ABK(IJMK).LT.EM6) GO TO 330 PRESCR 204
3217 205 RYF(LIJK)=DELX(I)*DELZ(K)*RDYP(J-1)*RRI(I)*ABK(IJMK) PRESCR 205
3218 206 SSM(LIJK)=SSM(LIJK)+DELT*RYF(LIJK) PRESCR 206
3219 207 IF (NF(IJMK).EQ.0) GO TO 330 PRESCR 207
3220 208 IF (NF(IJMK).GT.7) GO TO 320 PRESCR 208
3221 209 IF (NF(IJMK).NE.4) GO TO 320 PRESCR 209
3222 210 RYFA=(1.0-PETA(IJMK))*RYF(LIJK) PRESCR 210
3223 211 RYFB=((1.0-PETA(IJMK))*PN(IJK)+PETA(IJMK)*(PS(IJMK)+PSAT)-PN(IJMK) PRESCR 211
3224 212 1))*RYF(LIJK) PRESCR 212
3225 213 GO TO 320 PRESCR 213
3226 214 320 RYF(LIJK)=0.0 PRESCR 214
3227 215 330 SSM(LIJK)=SSM(LIJK)-DELT*(RXRA+RXLA+RZTA+RZBA+RYBKA+RYFA) PRESCR 215
3228 216 DIV(LIJK)=(DIV(LIJK)-DELT*(RXRB+RXLB+RZTB+RZBB+RYBKB+RYFB))/SSM PRESCR 216
3229 217 1 (LIJK) PRESCR 217
3230 218 FF(LIJK)=0.0 PRESCR 218
3231 219 AF(LIJK)=0.0 PRESCR 219
3232 220 QQ(LIJK)=0.0 PRESCR 220
3233 221 AD(LIJK)=0.0 PRESCR 221
3234 222 DP(LIJK)=0.0 PRESCR 222
3235 223 GO TO 360 PRESCR 223
3236 224 340 VLM(LIJK)=1.0 PRESCR 224
3237 225 350 DIV(LIJK)=0.0 PRESCR 225
3238 226 FF(LIJK)=0.0 PRESCR 226
3239 227 AF(LIJK)=0.0 PRESCR 227
3240 228 AD(LIJK)=0.0 PRESCR 228
3241 229 QQ(LIJK)=0.0 PRESCR 229
3242 230 SSM(LIJK)=1.0 PRESCR 230
3243 231 RXR(LIJK)=0.0 PRESCR 231
3244 232 RXL(LIJK)=0.0 PRESCR 232
3245 233 RZT(LIJK)=0.0 PRESCR 233
3246 234 RZB(LIJK)=0.0 PRESCR 234
3247 235 RYBK(LIJK)=0.0 PRESCR 235
3248 236 RYF(LIJK)=0.0 PRESCR 236
3249 237 DP(LIJK)=0.0 PRESCR 237
3250 238 360 CONTINUE PRESCR 238
3251 239 C IF (WBK.NE.4) GO TO 380 PRESCR 239
3252 240 DO 370 K=1,KMAX PRESCR 240
3253 241 DO 370 I=1,IMAX PRESCR 241
3254 242 LIJKF=I*I5*(K-1)+I PRESCR 242
3255 243 LIJKFP=LIJKF+IMAX PRESCR 243
3256 244 LIJKBK=LIJKFP+IMAX*JBAR PRESCR 244
3257 245 LIJKBKM=LIJKBK-IMAX PRESCR 245
3258 246 SSM(LIJKF)=SSM(LIJKBKM) PRESCR 246
3259 247 VLM(LIJKF)=VLM(LIJKBKM) PRESCR 247
3260 248 DIV(LIJKF)=DIV(LIJKBKM) PRESCR 248
3261 249 RXR(LIJKF)=RXR(LIJKBKM) PRESCR 249
3262 250 RXL(LIJKF)=RXL(LIJKBKM) PRESCR 250
3263 251 RZT(LIJKF)=RZT(LIJKBKM) PRESCR 251
3264 252 RZB(LIJKF)=RZB(LIJKBKM) PRESCR 252
3265 253 RYBK(LIJKF)=RYBK(LIJKBKM) PRESCR 253
3266 254 RYF(LIJKF)=RYF(LIJKBKM) PRESCR 254
3267 255 SSM(LIJKBK)=SSM(LIJKFP) PRESCR 255
3268 256 VLM(LIJKBK)=VLM(LIJKFP) PRESCR 256
3269 257 DIV(LIJKBK)=DIV(LIJKFP) PRESCR 257
3270 258 RXR(LIJKBK)=RXR(LIJKFP) PRESCR 258
3271 259 RXL(LIJKBK)=RXL(LIJKFP) PRESCR 259
3272 260 RZT(LIJKBK)=RZT(LIJKFP) PRESCR 260
3273 261 RZB(LIJKBK)=RZB(LIJKFP) PRESCR 261
3274 262 RYBK(LIJKBK)=RYBK(LIJKFP) PRESCR 262
3275 263 RYF(LIJKBK)=RYF(LIJKFP) PRESCR 263
3276 264 370 CONTINUE PRESCR 264
3277 265 380 CONTINUE PRESCR 265
3278 266 C **** PRESCR 266
3279 267 C ***** PRESCR 267
3280 268 380 CONTINUE PRESCR 268

```

```

3281 269 C *****
3282 270 390 A=C1/DD PRESCR 269
3283 271 CALL SAXPY (LVEC,A,FF,1,DP,1) PRESCR 270
3284 272 CALL SAXPY (LVEC,-A,QQ,1,DIV,1) PRESCR 271
3285 273 C PRESCR 272
3286 274 DO 400 K=1,KMAX PRESCR 273
3287 275 KKK=II5*(K-1) PRESCR 274
3288 276 DO 400 J=1,JMAX PRESCR 275
3289 277 JJJ=KKK+IMAX*(J-1) PRESCR 276
3290 278 DO 400 I=1,IMAX PRESCR 277
3291 279 C PRESCR 278
3292 280 C THE FOLLOWING COMMENTED STATEMENTS REPLACED IN ORDER TO VECTORIZE PRESCR 279
3293 281 C IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1 PRESCR 280
3294 282 C LIJK=II5*(K-1)+IMAX*(J-1)+I PRESCR 281
3295 283 C CQ(LIJK)=DIV(LIJK)*SSM(LIJK)/VLM(LIJK) PRESCR 282
3296 284 C IF((NF(IJK).NE.0).OR.(BETA(IJK).LT.0.0))CQ(LIJK)=0.0 PRESCR 283
3297 285 C PRESCR 284
3298 286 C CQ(JJJ+I)=DIV(JJJ+I)*SSM(JJJ+I)/VLM(JJJ+I) PRESCR 285
3299 287 IF ((NF(NQ*(JJJ+I-1)+1).NE.0).OR.(BETA(NQ*(JJJ+I-1)+1).LT.0.0)) CQ PRESCR 286
3300 288 1 (JJJ+I)=O.O PRESCR 287
3301 289 400 CONTINUE PRESCR 288
3302 290 C PRESCR 289
3303 291 IJMAX=ISAMAX(LVEC,CQ,1) PRESCR 290
3304 292 DMAX=ABS(CQ(IJMAX)) PRESCR 291
3305 293 IF (DMAX.LE.EPSI) GO TO 450 PRESCR 292
3306 294 C PRESCR 293
3307 295 DO 410 K=1,KMAX PRESCR 294
3308 296 KKK=II5*(K-1) PRESCR 295
3309 297 DO 410 J=1,JMAX PRESCR 296
3310 298 JJJ=KKK+IMAX*(J-1) PRESCR 297
3311 299 DO 410 I=1,IMAX PRESCR 298
3312 300 C PRESCR 299
3313 301 C THE FOLLOWING COMMENTED STATEMENTS REPLACED IN ORDER TO VECTORIZE PRESCR 300
3314 302 C IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1 PRESCR 301
3315 303 C LIJK=II5*(K-1)+IMAX*(J-1)+I PRESCR 302
3316 304 C LIPJK=LIJK+1 PRESCR 303
3317 305 C LIMJK=LIJK-1 PRESCR 304
3318 306 C LIUPK=LIJK+IMAX PRESCR 305
3319 307 C LIUMK=LIJK-IMAX PRESCR 306
3320 308 C LIUKP=LIJK+II5 PRESCR 307
3321 309 C LIUKM=LIJK-II5 PRESCR 308
3322 310 C AD(LIJK)=DELT*(RXR(LIJK)*DIV(LIPJK)+RXL(LIJK)*DIV(LIMJK)+RZT(LIJK) PRESCR 309
3323 311 C 1 *DIV(LIJKP)+RZB(LIJK)*DIV(LIJKM)+RYBK(LIJK)*DIV(LIUPK)+RYF(LIJK)* PRESCR 310
3324 312 C 2 DIV(LIUMK))-SSM(LIJK)*DIV(LIJK) PRESCR 311
3325 313 C IF((NF(IJK).NE.0).OR.(BETA(IJK).LT.0.0))AD(LIJK)=0.0 PRESCR 312
3326 314 C PRESCR 313
3327 315 AD(JJJ+I)=DELT*(RXR(JJJ+I)*DIV(JJJ+I+1)+RXL(JJJ+I)*DIV(JJJ+I-1) PRESCR 314
3328 316 1 +RZT(JJJ+I)*DIV(JJJ+I+II5)+RZB(JJJ+I)*DIV(JJJ+I-II5)+RYBK(JJJ+I) PRESCR 315
3329 317 2 *DIV(JJJ+I+IMAX)+RYF(JJJ+I)*DIV(JJJ+I-IMAX))-SSM(JJJ+I)*DIV(JJJ+I) PRESCR 316
3330 318 3 ) PRESCR 317
3331 319 IF ((NF(NQ*(JJJ+I-1)+1).NE.0).OR.(BETA(NQ*(JJJ+I-1)+1).LT.0.0)) AD PRESCR 318
3332 320 1 (JJJ+I)=O.O PRESCR 319
3333 321 410 CONTINUE PRESCR 320
3334 322 C PRESCR 321
3335 323 IF (WBK.NE.4) GO TO 430 PRESCR 322
3336 324 DO 420 K=1,KMAX PRESCR 323
3337 325 KKK=II5*(K-1) PRESCR 324
3338 326 CDIR$ IVDEP PRESCR 325
3339 327 DO 420 I=1,IMAX PRESCR 326
3340 328 C PRESCR 327
3341 329 C THE FOLLOWING COMMENTED STATEMENTS REPLACED IN ORDER TO VECTORIZE PRESCR 328
3342 330 C LIJKF=II5*(K-1)+I PRESCR 329
3343 331 C LIJKFP=LIJKF+IMAX PRESCR 330
3344 332 C LIJKBK=LIJKFP+IMAX*JBAR PRESCR 331
3345 333 C LIJKBKM=LIJKBK-IMAX PRESCR 332
3346 334 C AD(LIJKF)=AD(LIJKBK) PRESCR 333
3347 335 C AD(LIJKBK)=AD(LIJKFP) PRESCR 334
3348 336 C PRESCR 335
3349 337 AD(KKK+I)=AD(KKK+II6+I) PRESCR 336
3350 338 AD(KKK+II7+I)=AD(KKK+IMAX+I) PRESCR 337
3351 339 420 CONTINUE PRESCR 338
3352 340 C PRESCR 339
3353 341 430 C2=SDOT(LVEC,DIV,1,AD,1) PRESCR 340
3354 342 B=C2/C1 PRESCR 341
3355 343 CALL SSCAL (LVEC,B,FF,1) PRESCR 342
3356 344 CALL SAXPY (LVEC,1.0,DIV,1,FF,1) PRESCR 343
3357 345 C PRESCR 344
3358 346 CALL SSCAL (LVEC,B,AF,1) PRESCR 345
3359 347 CALL SAXPY (LVEC,1.0,AD,1,AF,1) PRESCR 346
3360 348 C PRESCR 347

```

```

3361 349      DO 440 K=1,KMAX          PRESCR 349
3362 350      DO 440 J=1,JMAX          PRESCR 350
3363 351      DO 440 I=1,IMAX          PRESCR 351
3364 352      LIJK=II5*(K-1)+IMAX*(J-1)+I  PRESCR 352
3365 353      QQ(LIJK)=AF(LIJK)/SSM(LIJK)  PRESCR 353
3366 354      440 CONTINUE          PRESCR 354
3367 355 C     DD=SDOT(LVEC,QQ,1,AF,1)  PRESCR 355
3368 356      C1=C2                PRESCR 356
3369 357      ITER=ITER+1          PRESCR 357
3370 358      IF (ITER.GT.ITMAX) GO TO 750  PRESCR 358
3371 359      GO TO 390            PRESCR 359
3372 360      C
3373 361 C     *****
3374 362      450 IF (ITER.GT.ITMOST) EPSI=1.05*EPSI  PRESCR 362
3375 363      IF (ITER.LT.ITMIN) EPSI=0.95*EPSI  PRESCR 363
3376 364      EPSI=AMIN1(EPSI,EPSIP)    PRESCR 364
3377 365      EPSI=AMAX1(EPSI,EPSIM)    PRESCR 365
3378 366 C     *****
3379 367 C     *****
3380 368 C     +--+ UNFOLD PRESSURE FOR INTERIOR FLUID CELLS
3382 370 C
3383 371      DO 460 K=1,KMAX          PRESCR 370
3384 372      KKK=II5*(K-1)          PRESCR 372
3385 373      DO 460 J=1,JMAX          PRESCR 373
3386 374      JJJ=KKK+IMAX*(J-1)    PRESCR 374
3387 375      DO 460 I=1,IMAX          PRESCR 375
3388 376      D(NQ*(JJJ+I-1)+1)=CQ(JJJ+I)  PRESCR 376
3389 377      IF ((NF(NQ*(JJJ+I-1)+1).LE.0).AND.(AC(NQ*(JJJ+I-1)+1).GE.EM6).AND. PRESCR 377
3390 378      1 (BETA(NQ*(JJJ+I-1)+1).GE.0.0)) P(NQ*(JJJ+I-1)+1)=PN(NQ*(JJJ+I-1)+  PRESCR 378
3391 379      2 1)+DP(JJJ+I)          PRESCR 379
3392 380      460 CONTINUE          PRESCR 380
3393 381 C     +--+ CALCULATE PRESSURE CHANGES FOR SURFACE CELLS
3395 383 C
3396 384      DO 610 K=1,KMAX          PRESCR 383
3397 385      DO 610 J=1,JMAX          PRESCR 384
3398 386      DO 610 I=1,IMAX          PRESCR 385
3399 387      CALL IJKAJCT          PRESCR 386
3400 388      LIJK=II5*(K-1)+IMAX*(J-1)+I  PRESCR 387
3401 389      IF (AC(IJK).LT.EM6.OR.BETA(IJK).LT.0.0) GO TO 590  PRESCR 388
3402 390      NFF=NF(IJK)          PRESCR 389
3403 391      IF (NFF.EQ.0) GO TO 600  PRESCR 390
3404 392      IF (NFF.GT.7) GO TO 560  PRESCR 391
3405 393      L=I                PRESCR 392
3406 394      M=J                PRESCR 393
3407 395      N=K                PRESCR 394
3408 396      GO TO (470,480,490,500,510,520,530), NFF  PRESCR 395
3409 397      470 L=I-1          PRESCR 396
3410 398      GO TO 540            PRESCR 397
3411 399      480 L=I+1          PRESCR 398
3412 400      GO TO 540            PRESCR 399
3413 401      490 M=J-1          PRESCR 400
3414 402      GO TO 540            PRESCR 401
3415 403      500 M=J+1          PRESCR 402
3416 404      GO TO 540            PRESCR 403
3417 405      510 N=K-1          PRESCR 404
3418 406      GO TO 540            PRESCR 405
3419 407      520 N=K+1          PRESCR 406
3420 408      GO TO 540            PRESCR 407
3421 409      530 GO TO 550          PRESCR 408
3422 410      540 LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1  PRESCR 409
3423 411      NFEL=NF(IMJK)    PRESCR 410
3424 412      NFER=NF(IPJK)    PRESCR 411
3425 413      NFEF=NF(IJMK)    PRESCR 412
3426 414      NFEBK=NF(IJPK)    PRESCR 413
3427 415      NFEB=NF(IJKM)    PRESCR 414
3428 416      NFET=NF(IJKP)    PRESCR 415
3429 417      NFE=MAXC(NFEL,NFER,NFEBK,NFEF,NFEB,NFET,1)  PRESCR 416
3430 418      PSURF=PS(IJK)+PR(NFE)  PRESCR 417
3431 419      PLMN=P(LMN)          PRESCR 418
3432 420      IF (NF(LMN).NE.0.AND.BETA(IJK).GT.0.0) PLMN=PSURF  PRESCR 419
3433 421      DP(LIJK)=(1.0-PETA(IJK))*PLMN+PETA(IJK)*PSURF-PN(IJK)  PRESCR 420
3434 422      GO TO 600            PRESCR 421
3435 423      550 IF (PSAT.LE.0.0) GO TO 580  PRESCR 422
3436 424      IF (F(IJK).LT.EMF1) GO TO 570  PRESCR 423
3437 425      PMPS=0.0            PRESCR 424
3438 426      IF (PN(IJK).LT.PSAT) PMPS=P(IJK)-PSAT  PRESCR 425
3439 427      DIJK=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR(I-1  PRESCR 426
3440 428      1 ))+RDY(J)*(V(IJK)*ABK(IJK)-V(IJMK)*ABK(IJMK))+RDZ(K)*(W(IJK)*AT  PRESCR 427
                                         )+PRES 428

```

```

3441 429      2 (IJK)-W(IJKM)*AT(IJKM))/AC(IJK)          PRESCR 429
3442 430      DIJK=DIJK-PMPS**2/DELT                  PRESCR 430
3443 431      DP(LIJK)=-DIJK*BETA(IJK)/(1.0-2.0*PMPS*BETA(IJK)/DELT)  PRESCR 431
3444 432      GO TO 600                                PRESCR 432
3445 433      560 DP(LIJK)=PR(NFF)-PN(IJK)            PRESCR 433
3446 434      GO TO 600                                PRESCR 434
3447 435      570 DP(LIJK)=PSAT-PN(IJK)              PRESCR 435
3448 436      GO TO 600                                PRESCR 436
3449 437      580 DP(LIJK)=O.O                      PRESCR 437
3450 438      GO TO 600                                PRESCR 438
3451 439      590 DP(LIJK)=O.O                      PRESCR 439
3452 440      600 CONTINUE                            PRESCR 440
3453 441      610 CONTINUE                            PRESCR 441
3454 442 C
3455 443 C      CALC NEW VELOCITIES CONSISTENT WITH SURFACE PRESSURE ASSUMPTIONS  PRESCR 443
3456 444 C      NOTE: YIELDS DIV=O.O FOR INTERIOR CELLS ADJACENT TO SURFACE CELLS  PRESCR 444
3457 445 C
3458 446      DO 730 K=1,KM1                      PRESCR 445
3459 447      DO 730 J=1,JM1                      PRESCR 446
3460 448      DO 730 I=1,IM1                      PRESCR 447
3461 449      CALL IJKAJCT                         PRESCR 448
3462 450      LIJK=II5*(K-1)+IMAX*(J-1)+I        PRESCR 449
3463 451      LIPJK=LIJK+1                        PRESCR 450
3464 452      LIUPK=LIJK+IMAX                     PRESCR 451
3465 453      LIJKP=LIJK+II5                       PRESCR 452
3466 454      IF (AC(IJK).LT.EM6.OR.BETA(IJK).LT.O.O) GO TO 730  PRESCR 453
3467 455      IF (AR(IJK).LT.EM6) GO TO 650          PRESCR 454
3468 456      IF (NF(IJK).EQ.O.AND.NF(IPJK).EQ.O) GO TO 620  PRESCR 455
3469 457      IF (NF(IJK).EQ.O.AND.NF(IPJK).NE.1) GO TO 630  PRESCR 456
3470 458      IF (NF(IJK).NE.2.AND.NF(IPJK).EQ.O) GO TO 640  PRESCR 457
3471 459      620 U(IJK)=U(IJK)+DELT*(1.0*(DP(LIJK)-DP(LIPJK)))*RDXP(I)  PRESCR 458
3472 460      GO TO 650                                PRESCR 459
3473 461      630 U(IJK)=U(IJK)+DELT*DP(LIJK)*RDXP(I)  PRESCR 460
3474 462      GO TO 650                                PRESCR 461
3475 463      640 U(IJK)=U(IJK)+DELT*(-DP(LIPJK))*RDXP(I)  PRESCR 462
3476 464      GO TO 650                                PRESCR 463
3477 465      650 IF (ABK(IJK).LT.EM6) GO TO 690          PRESCR 464
3478 466      IF (NF(IJK).EQ.O.AND.NF(IJPK).EQ.O) GO TO 660  PRESCR 465
3479 467      IF (NF(IJK).EQ.O.AND.NF(IJPK).NE.3) GO TO 670  PRESCR 466
3480 468      IF (NF(IJK).NE.4.AND.NF(IJPK).EQ.O) GO TO 680  PRESCR 467
3481 469      660 V(IJK)=V(IJK)+DELT*(1.0*(DP(LIJK)-DP(LIUPK)))*RDYP(J)*RRI(I)  PRESCR 468
3482 470      GO TO 690                                PRESCR 469
3483 471      670 V(IJK)=V(IJK)+DELT*DP(LIJK)*RDYP(J)*RRI(I)  PRESCR 470
3484 472      GO TO 690                                PRESCR 471
3485 473      680 V(IJK)=V(IJK)+DELT*(-DP(LIUPK))*RDYP(J)*RRI(I)  PRESCR 472
3486 474      690 IF (AT(IJK).LT.EM6) GO TO 730          PRESCR 473
3487 475      IF (NF(IJK).EQ.O.AND.NF(IJPK).EQ.O) GO TO 700  PRESCR 474
3488 476      IF (NF(IJK).EQ.O.AND.NF(IJPK).NE.5) GO TO 710  PRESCR 475
3489 477      IF (NF(IJK).NE.6.AND.NF(IJPK).EQ.O) GO TO 720  PRESCR 476
3490 478      700 W(IJK)=W(IJK)+DELT*(1.0*(DP(LIJK)-DP(LIJKP)))*RDZP(K)  PRESCR 477
3491 479      GO TO 730                                PRESCR 478
3492 480      710 W(IJK)=W(IJK)+DELT*DP(LIJK)*RDZP(K)  PRESCR 479
3493 481      GO TO 730                                PRESCR 480
3494 482      720 W(IJK)=W(IJK)+DELT*(-DP(LIUPK))*RDZP(K)  PRESCR 481
3495 483      730 CONTINUE                            PRESCR 482
3496 484 C
3497 485 C      +++ PUT IN PRESSURES FOR SURFACE CELLS, ISOLATED CELLS, VOIDS  PRESCR 483
3498 486 C
3499 487      DO 740 K=1,KMAX                      PRESCR 484
3500 488      KKK=II5*(K-1)                      PRESCR 485
3501 489      DO 740 J=1,JMAX                      PRESCR 486
3502 490      JJJ=KKK+IMAX*(J-1)                    PRESCR 487
3503 491      DO 740 I=1,IMAX                      PRESCR 488
3504 492      IF ((NF(NQ*(JJJ+I-1)+1).EQ.O).OR.(AC(NQ*(JJJ+I-1)+1).LT.EM6).OR.  PRESCR 489
3505 493      1(BETA(NQ*(JJJ+I-1)+1).LT.O.O)) GO TO 740  PRESCR 490
3506 494      P(NQ*(JJJ+I-1)+1)=PN(NQ*(JJJ+I-1)+1)+DP(JJJ+I)  PRESCR 491
3507 495      740 CONTINUE                            PRESCR 492
3508 496 C
3509 497      RETURN                               PRESCR 493
3510 498 C
3511 499      750 WRITE (59,770) CYCLE             PRESCR 494
3512 500      WRITE (12,770) CYCLE               PRESCR 495
3513 501      WRITE (9,770) CYCLE                PRESCR 496
3514 502      CALL EXIT                           PRESCR 497
3515 503 C
3516 504      760 WRITE (59,780) CYCLE             PRESCR 498
3517 505      WRITE (12,780) CYCLE               PRESCR 499
3518 506      WRITE (9,780) CYCLE                PRESCR 500
3519 507      CALL EXIT                           PRESCR 501
3520 508 C

```

```

3521 509    770 FORMAT (1X,29HTDD MANY CR ITERATIONS, CYCLE,I7)      PRESCR   509
3522 510    780 FORMAT (1X,44HMESH DIMENSIONS AND STORAGE NOT EQUAL, CYCLE,I7)  PRESCR   510
3523 511    END      PRESCR   511
3524 1 *DK PRESRC      PRESRC   1
3525 2 *DK PRESSIT      PRESSIT   1
3526 3      SUBROUTINE PRESSIT      PRESSIT   2
3527 4 *CA SLCOM1      PRESSIT   3
3528 5      DATA ITMAX /1000/      PRESSIT   4
3529 6 C      PRESSIT   5
3530 7 C * * PRESSURE ITERATION      PRESSIT   6
3531 8 C      PRESSIT   7
3532 9      10 IF (IDRDER.EQ.2) ALPHA=1.0      PRESSIT   8
3533 10     IF (FLG.EQ.0.0) GO TO 180      PRESSIT   9
3534 11     ITER=ITER+1      PRESSIT  10
3535 12     IF (ITER.LT.ITMAX) GO TO 20      PRESSIT  11
3536 13     FNOC=1.0      PRESSIT  12
3537 14     NDCON=NDCON+1      PRESSIT  13
3538 15     CALL LPRT2      PRESSIT  14
3539 16     GO TO 180      PRESSIT  15
3540 17     20 FLG=0.0      PRESSIT  16
3541 18 C      PRESSIT  17
3542 19 C * * COMPUTE UPDATED CELL PRESSURE AND VELOCITIES      PRESSIT  18
3543 20 C      PRESSIT  19
3544 21     DO 170 K=2,KM1      PRESSIT  20
3545 22     DO 170 J=2,JM1      PRESSIT  21
3546 23     DO 170 I=2,IM1      PRESSIT  22
3547 24     CALL CALCIJK      PRESSIT  23
3548 25     IF (BETA(IJK).LT.0.0) GO TO 170      PRESSIT  24
3549 26     IF (F(IJK).LT.EMF) GO TO 170      PRESSIT  25
3550 27     IF (NF(IJK).EQ.0) GO TO 100      PRESSIT  26
3551 28 C      PRESSIT  27
3552 29 C * * CALCULATE PRESSURE CHANGE FOR SURFACE CELLS      PRESSIT  28
3553 30 C      PRESSIT  29
3554 31     NFF=NF(IJK)      PRESSIT  30
3555 32     L=I      PRESSIT  31
3556 33     M=J      PRESSIT  32
3557 34     N=K      PRESSIT  33
3558 35     GO TO (30,40,50,60,70,80,170), NFF      PRESSIT  34
3559 36     30 L=I-1      PRESSIT  35
3560 37     GO TO 90      PRESSIT  36
3561 38     40 L=I+1      PRESSIT  37
3562 39     GO TO 90      PRESSIT  38
3563 40     50 M=J-1      PRESSIT  39
3564 41     GO TO 90      PRESSIT  40
3565 42     60 M=J+1      PRESSIT  41
3566 43     GO TO 90      PRESSIT  42
3567 44     70 N=K-1      PRESSIT  43
3568 45     GO TO 90      PRESSIT  44
3569 46     80 N=K+1      PRESSIT  45
3570 47     90 CONTINUE      PRESSIT  46
3571 48     LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1      PRESSIT  47
3572 49     NFER=NF(IPJK)      PRESSIT  48
3573 50     NFEL=NF(IMJK)      PRESSIT  49
3574 51     NFEBK=NF(IJPK)      PRESSIT  50
3575 52     NFEF=NF(IJMK)      PRESSIT  51
3576 53     NFEB=NF(IJKM)      PRESSIT  52
3577 54     NFET=NF(IJKP)      PRESSIT  53
3578 55     NFE=MAX(NFEL,NFER,NFEBK,NFEF,NFEB,NFET,1)      PRESSIT  54
3579 56     PSURF=PR(NFE)+PS(IJK)      PRESSIT  55
3580 57     PLMN=P(LMN)      PRESSIT  56
3581 58     IF (NF(LMN).NE.0.AND.BETA(IJK).GT.0.0) PLMN=PSURF      PRESSIT  57
3582 59     DELP=(1.0-PETA(IJK))*PLMN+PETA(IJK)*PSURF-P(IJK)      PRESSIT  58
3583 60     GO TO 110      PRESSIT  59
3584 61     100 CONTINUE      PRESSIT  60
3585 62 C      PRESSIT  61
3586 63 C * * CALCULATE PRESSURE CHANGE FOR FULL CELLS      PRESSIT  62
3587 64 C AND TEST FOR CONVERGENCE      PRESSIT  63
3588 65 C      PRESSIT  64
3589 66     D(IJK)=(RRI(I)*(RDX(I)*(U(IJK)*AR(IJK)/RR(I)-U(IMJK)*AR(IMJK)/RR(I)
3590 67     1 -1))+RDY(J)*(V(IJK)+ABK(IJK)-V(IJMK)*ABK(IJMK)))+RDZ(K)*(W(IJK)
3591 68     2 *AT(IJK)-W(IJKM)*AT(IJKM))/AC(IJK)      PRESSIT  65
3592 69     DIJK=D(IJK)      PRESSIT  66
3593 70     IF (ABS(DIJK).GE.EPSI) FLG=1.0      PRESSIT  67
3594 71     DELP=-BETA(IJK)*DIJK*PETA(IJK)      PRESSIT  68
3595 72 C      PRESSIT  69
3596 73 C * * UPDATE CELL PRESSURE AND VELOCITY      PRESSIT  70
3597 74 C      PRESSIT  71
3598 75     110 DTDELP=DELT*DELP      PRESSIT  72
3599 76     P(IJK)=P(IJK)+DELP      PRESSIT  73
3600 77     IF (BETA(IPJK).LT.0.0.OR.BETA(IPJK).EQ.1.0) GO TO 120      PRESSIT  74

```

3601	78	U(IJK)=U(IJK)+DTDELP*RDXP(I)	PRESSIT	77
3602	79	120 IF (BETA(IMJK).LT.0.0.OR.BETA(IMJK).EQ.1.0) GO TO 130	PRESSIT	78
3603	80	U(IMJK)=U(IMJK)-DTDELP*RDXP(I-1)	PRESSIT	79
3604	81	130 IF (BETA(IJPK).LT.0.0.OR.BETA(IJPK).EQ.1.0) GO TO 140	PRESSIT	80
3605	82	V(IJK)=V(IJK)+DTDELP*RDYP(J)*RRI(I)	PRESSIT	81
3606	83	140 IF (BETA(IJMK).LT.0.0.OR.BETA(IJMK).EQ.1.0) GO TO 150	PRESSIT	82
3607	84	V(IJMK)=V(IJMK)-DTDELP*RDYP(J-1)*RRI(I)	PRESSIT	83
3608	85	150 IF (BETA(IJKP).LT.0.0.OR.BETA(IJKP).EQ.1.0) GO TO 160	PRESSIT	84
3609	86	W(IJK)=W(IJK)+DTDELP*RDZP(K)	PRESSIT	85
3610	87	160 IF (BETA(IJKM).LT.0.0.OR.BETA(IJKM).EQ.1.0) GO TO 170	PRESSIT	86
3611	88	W(IJKM)=W(IJKM)-DTDELP*RDZP(K-1)	PRESSIT	87
3612	89	170 CONTINUE	PRESSIT	88
3613	90 C	* * IMPOSE BOUNDARY CONDITIONS	PRESSIT	89
3614	91 C	CALL BC	PRESSIT	90
3615	92 C	GO TO 10	PRESSIT	91
3616	93	180 CONTINUE	PRESSIT	92
3617	94 C	RETURN	PRESSIT	93
3618	95	END	PRESSIT	94
3619	96	1 *DK RCONTUR	RCONTUR	1
3620	97 C	SUBROUTINE RCONTUR	RCONTUR	2
3621	98	*CA SLCOM1	RCONTUR	3
3622	99	4 C	RCONTUR	4
3623	1	* * CONTOUR SET-UP	RCONTUR	5
3624	2	6 C	RCONTUR	6
3625	3	7 C * * THIS SET UP IS FOR CELL CENTERED QUANTITIES	RCONTUR	7
3626	4 C	8 C NAMELIST /CONTUR/ NNX,NNY,NNZ,NZX,NZY,NZZ,NC,ZMN,ZMX,DLZ,ZCQ,DMPX	RCONTUR	8
3627	9	1 ,DMPY,DMPZ,IGRD,ITITLE1,NTITLE1,ITITLE2,NTITLE2,XLABLE,YLABLE	RCONTUR	9
3628	10	2 ,ZLABLE,NXLBL,NYLBL,NZLBL	RCONTUR	10
3629	11	12 C	RCONTUR	11
3630	13 C	* * DEFAULT VALUES FOR CONTUR	RCONTUR	12
3631	14 C	NNX=-IBAR	RCONTUR	13
3632	15	NNY=-JBAR	RCONTUR	14
3633	16	NNZ=-KBAR	RCONTUR	15
3634	17	NZX=50	RCONTUR	16
3635	18	NZY=50	RCONTUR	17
3636	19	NZZ=50	RCONTUR	18
3637	20	NC=10	RCONTUR	19
3638	21	ZMN=-1.0	RCONTUR	20
3639	22	ZMX=-1.0	RCONTUR	21
3640	23	DLZ=0.0	RCONTUR	22
3641	24	ZCQ(1)=1.0	RCONTUR	23
3642	25	DMPX=X(IM1)	RCONTUR	24
3643	26	DMPY=Y(JM1)	RCONTUR	25
3644	27	DMPZ=Z(KM1)	RCONTUR	26
3645	28	IGRD=0	RCONTUR	27
3646	29	ITITLE1(1)=8HPPRESSURE	RCONTUR	28
3647	30	ITITLE1(2)=8HCONTOURS	RCONTUR	29
3648	31	NTITLE1=16	RCONTUR	30
3649	32	ITITLE2(1)=8HVORTICIT	RCONTUR	31
3650	33	ITITLE2(2)=8HY CONTOU	RCONTUR	32
3651	34	ITITLE2(3)=8HRS	RCONTUR	33
3652	35	NTITLE2=18	RCONTUR	34
3653	36	XLABLE=6HX-AXIS	RCONTUR	35
3654	37	YLABLE=6HY-AXIS	RCONTUR	36
3655	38	ZLABLE=6HZ-AXIS	RCONTUR	37
3656	39	NXLBL=6	RCONTUR	38
3657	40	NYLBL=6	RCONTUR	39
3658	41	NZLBL=6	RCONTUR	40
3659	42	NC=10	RCONTUR	41
3660	43 C	* * READ VALUES FOR CONTOUR FROM NAMELIST INPUT / CONTUR /	RCONTUR	42
3661	44 C	READ (10,CONTUR)	RCONTUR	43
3662	45 C	46	RCONTUR	44
3663	46	READ (10,CONTUR)	RCONTUR	45
3664	47 C	48 C * * WRITE CONTOUR INPUT VALUES ONTO TAPE 9 AND TAPE 12.	RCONTUR	46
3665	48 C	49 C	RCONTUR	47
3666	50 C	* * WRITE EFFECTIVE DATA FROM "CONTUR" TO TAPES 9 AND 12.	RCONTUR	48
3667	51 C	52	RCONTUR	49
3668	53 C	IF (LPR.EQ.0) GO TO 30	RCONTUR	50
3669	54	WRITE (9,40)	RCONTUR	51
3670	55	WRITE (12,40)	RCONTUR	52
3671	56	WRITE (9,300)	RCONTUR	53
3672	57	WRITE (12,300)	RCONTUR	54
3673	58 C		RCONTUR	55
3674			RCONTUR	56
3675			RCONTUR	57
3676			RCONTUR	58

3681	59	KPR=9	RCONTUR	59
3682	60	ASSIGN 20 TO KRET	RCONTUR	60
3683	61	10 WRITE (KPR,50) NNX	RCONTUR	61
3684	62	WRITE (KPR,60) NNY	RCONTUR	62
3685	63	WRITE (KPR,70) NNZ	RCONTUR	63
3686	64	WRITE (KPR,80) NZX	RCONTUR	64
3687	65	WRITE (KPR,90) NZY	RCONTUR	65
3688	66	WRITE (KPR,100) NZZ	RCONTUR	66
3689	67	WRITE (KPR,110) NC	RCONTUR	67
3690	68	WRITE (KPR,120) ZMN	RCONTUR	68
3691	69	WRITE (KPR,130) ZMX	RCONTUR	69
3692	70	WRITE (KPR,140) DLZ	RCONTUR	70
3693	71	WRITE (KPR,150) ZCQ	RCONTUR	71
3694	72	WRITE (KPR,160) DMPX	RCONTUR	72
3695	73	WRITE (KPR,170) DMPY	RCONTUR	73
3696	74	WRITE (KPR,180) DMPZ	RCONTUR	74
3697	75	WRITE (KPR,190) IGRD	RCONTUR	75
3698	76	WRITE (KPR,200) (ITITLE1(L),L=1,3)	RCONTUR	76
3699	77	WRITE (KPR,210) NTITLE1	RCONTUR	77
3700	78	WRITE (KPR,220) (ITITLE2(L),L=1,3)	RCONTUR	78
3701	79	WRITE (KPR,230) NTITLE2	RCONTUR	79
3702	80	WRITE (KPR,240) XLABEL	RCONTUR	80
3703	81	WRITE (KPR,250) YLABEL	RCONTUR	81
3704	82	WRITE (KPR,260) ZLABEL	RCONTUR	82
3705	83	WRITE (KPR,270) NXLBL	RCONTUR	83
3706	84	WRITE (KPR,280) NYLBL	RCONTUR	84
3707	85	WRITE (KPR,290) NZLBL	RCONTUR	85
3708	86	GO TO KRET, (20,30)	RCONTUR	86
3709	87	20 ASSIGN 30 TO KRET	RCONTUR	87
3710	88	KPR=12	RCONTUR	88
3711	89	GO TO 10	RCONTUR	89
3712	90	30 CONTINUE	RCONTUR	90
3713	91	RETURN	RCONTUR	91
3714	92	C	RCONTUR	92
3715	93	40 FORMAT (1H1)	RCONTUR	93
3716	94	50 FORMAT (10X,1OH NNX= ,I3)	RCONTUR	94
3717	95	60 FORMAT (10X,1OH NNY= ,I3)	RCONTUR	95
3718	96	70 FORMAT (10X,1OH NNZ= ,I3)	RCONTUR	96
3719	97	80 FORMAT (10X,1OH NZX= ,I3)	RCONTUR	97
3720	98	90 FORMAT (10X,1OH NZY= ,I3)	RCONTUR	98
3721	99	100 FORMAT (10X,1OH NZZ= ,I3)	RCONTUR	99
3722	100	110 FORMAT (10X,1OH NC= ,I3)	RCONTUR	100
3723	101	120 FORMAT (10X,1OH ZMN= ,1PE12.5)	RCONTUR	101
3724	102	130 FORMAT (10X,1OH ZMX= ,1PE12.5)	RCONTUR	102
3725	103	140 FORMAT (10X,1OH DLZ= ,1PE12.5)	RCONTUR	103
3726	104	150 FORMAT (10X,1OH ZCQ= ,1PE12.5)	RCONTUR	104
3727	105	160 FORMAT (10X,1OH DMPX= ,1PE12.5)	RCONTUR	105
3728	106	170 FORMAT (10X,1OH DMPY= ,1PE12.5)	RCONTUR	106
3729	107	180 FORMAT (10X,1OH DMPZ= ,1PE12.5)	RCONTUR	107
3730	108	190 FORMAT (10X,1OH IGRD= ,I3)	RCONTUR	108
3731	109	200 FORMAT (10X,1OH ITITLE1= ,3A8)	RCONTUR	109
3732	110	210 FORMAT (10X,1OH NTITLE1= ,I3)	RCONTUR	110
3733	111	220 FORMAT (10X,1OH ITITLE2= ,3A8)	RCONTUR	111
3734	112	230 FORMAT (10X,1OH NTITLE2= ,I3)	RCONTUR	112
3735	113	240 FORMAT (10X,1OH XLABEL= ,A8)	RCONTUR	113
3736	114	250 FORMAT (10X,1OH YLABEL= ,A8)	RCONTUR	114
3737	115	260 FORMAT (10X,1OH ZLABEL= ,A8)	RCONTUR	115
3738	116	270 FORMAT (10X,1OH NXLBL= ,I3)	RCONTUR	116
3739	117	280 FORMAT (10X,1OH NYLBL= ,I3)	RCONTUR	117
3740	118	290 FORMAT (10X,1OH NZLBL= ,I3)	RCONTUR	118
3741	119	300 FORMAT (1HO,/,25X,26HCNTUR PLOTTING SPECIFICS,/) END	RCONTUR	119
3742	120	RCONTUR	120	
3743	1	*DK RDTAPE	RDTAPE	1
3744	2	SUBROUTINE RDTAPE	RDTAPE	2
3745	3	*CA SLCOM1	RDTAPE	3
3746	4	C	RDTAPE	4
3747	5	C * * READ INPUT TAPE	RDTAPE	5
3748	6	C	RDTAPE	6
3749	7	REWIND 7	RDTAPE	7
3750	8	JTD=TD	RDTAPE	8
3751	9	JNSC=LOCF(ZLAST)-LOCF(AA)+1	RDTAPE	9
3752	10	10 READ (7) (AA(N),N=1,JNSC)	RDTAPE	10
3753	11	READ (7) (BASC(N),N=1,NCR2)	RDTAPE	11
3754	12	READ (7) (BASC1(N),N=1,NCR2)	RDTAPE	12
3755	13	READ (7) (BASC2(N),N=1,NCR2)	RDTAPE	13
3756	14	READ (7) (BASC3(N),N=1,NCR2)	RDTAPE	14
3757	15	IF (JTD.NE.NUMTD) GO TO 10	RDTAPE	15
3758	16	CALL CLOSE (7)	RDTAPE	16
3759	17	CALL SECOND (STIM)	RDTAPE	17
3760	18	TRL=STIM	RDTAPE	18

3761	19	T1=TRL	RDTAPE	19
3762	20	T2=T1	RDTAPE	20
3763	21	CALL DATEH (DAT)	RDTAPE	21
3764	22	CALL CLOCK1 (CLK)	RDTAPE	22
3765	23	CALL GETJTL (TL)	RDTAPE	23
3766	24	TLM=TL-150.0+(1.0-TLIMD)*1.0E+10	RDTAPE	24
3767	25	WRITE (9,20) JTD	RDTAPE	25
3768	26	WRITE (9,30) NAME,JNM,DAT,CLK	RDTAPE	26
3769	27	WRITE (9,40) ITER,T,CYCLE	RDTAPE	27
3770	28	WRITE (12,20) JTD	RDTAPE	28
3771	29	WRITE (12,30) NAME,JNM,DAT,CLK	RDTAPE	29
3772	30	WRITE (12,40) ITER,T,CYCLE	RDTAPE	30
3773	31	WRITE (59,20) JTD	RDTAPE	31
3774	32	WRITE (59,30) NAME,JNM,DAT,CLK	RDTAPE	32
3775	33	WRITE (59,40) ITER,T,CYCLE	RDTAPE	33
3776	34 C	* * * CHANGES TO TAPE DUMP INSERTED HERE	RDTAPE	34
3777	35 C	CHANGES TO ALLOW INPUT PARAMETER MODIFICATION ON RESTART	RDTAPE	35
3778	36 C	CALL RINPUT	RDTAPE	36
3779	37 C	CALL RGRAFIC	RDTAPE	37
3780	38 C	RETURN	RDTAPE	38
3781	39	FORMAT (22HO RESTARTING FROM TD= ,I3)	RDTAPE	39
3782	40	30 FORMAT (1H ,18X,1O8,1X,A8,2(1X,A8))	RDTAPE	40
3783	41	40 FORMAT (6X,6HITER= ,I5,18X,6HTIME= ,1PE12.5,12X,7HCYCLE= ,I4)	RDTAPE	41
3784	42 C	END	RDTAPE	42
3785	43	*DK RGRAFIC	RGRAFIC	1
3786	44	SUBROUTINE RGRAFIC	RGRAFIC	2
3787	45	*CA SLCOM1	RGRAFIC	3
3788	46	C	RGRAFIC	4
3789	1	NAMELIST /GRAFIC/ XEA,YEA,ZEA,XCA,YCA,ZCA,NVEWS,NVPLTS,IVVIEW,IV1	RGRAFIC	5
3790	2	1 ,IV2,JV1,JV2,KV1,KV2,NAV,IPERV,NCPLTS,ICVIEW,IC1,IC2,JC1,JC2,KC1	RGRAFIC	6
3791	3	2 ,KC2,NAC,IPERC,NSPLTS,ISVIEW,IS1,IS2,JS1,JS2,KS1,KS2,NAS,IPERS	RGRAFIC	7
3792	4 C	C*****	RGRAFIC	8
3793	5	C***** DEFAULT VALUES FOR NAMELIST /GRAFIC/	RGRAFIC	9
3794	6	10 C*****	RGRAFIC	10
3795	7	11 C***** DEFAULT VALUES ARE FOR ONE VELOCITY VECTOR VIEW. WITH FULL	RGRAFIC	11
3796	8 C*****	12 C***** RANGE ON I AND FULL RANGE ON J WITH K=10 CONSTANT.	RGRAFIC	12
3797	9	13 C*****	RGRAFIC	13
3798	10	IF (TD.GE.1) GO TO 10	RGRAFIC	14
3799	11	XEA(1)=50.0	RGRAFIC	15
3800	12	YEA(1)=50.0	RGRAFIC	16
3801	13	ZEA(1)=50.0	RGRAFIC	17
3802	14	XCA(1)=0.05	RGRAFIC	18
3803	15	YCA(1)=0.05	RGRAFIC	19
3804	16	ZCA(1)=0.05	RGRAFIC	20
3805	17	NVEWS=1	RGRAFIC	21
3806	18	NVPLTS=1	RGRAFIC	22
3807	19	IV1(1)=2	RGRAFIC	23
3808	20	IV2(1)=99	RGRAFIC	24
3809	21	JV1(1)=2	RGRAFIC	25
3810	22	JV2(1)=99	RGRAFIC	26
3811	23	KV1(1)=10	RGRAFIC	27
3812	24	KV2(1)=10	RGRAFIC	28
3813	25	NCPLTS=0	RGRAFIC	29
3814	26	NSPLTS=0	RGRAFIC	30
3815	27	C	RGRAFIC	31
3816	28	* * * READ IN PLOT-OUTPUT SPECIFICS	RGRAFIC	32
3817	29	33 C	RGRAFIC	33
3818	30	10 READ (10,GRAFIC)	RGRAFIC	34
3819	31 C	34 C	RGRAFIC	35
3820	32 C	IPMX=IM1	RGRAFIC	36
3821	33 C	JPMX=JM1	RGRAFIC	37
3822	34	KPMX=KM1	RGRAFIC	38
3823	35 C	IF (NVPLTS.EQ.0) GO TO 30	RGRAFIC	39
3824	36	DO 20 N=1,NVPLTS	RGRAFIC	40
3825	37	IF (IV1(N).GT.IMAX) IV1(N)=IPMX	RGRAFIC	41
3826	38	IF (IV2(N).GT.IMAX) IV2(N)=IPMX	RGRAFIC	42
3827	39	IF (JV1(N).GT.JMAX) JV1(N)=JPMX	RGRAFIC	43
3828	40	IF (JV2(N).GT.JMAX) JV2(N)=JPMX	RGRAFIC	44
3829	41	IF (KV1(N).GT.KMAX) KV1(N)=KPMX	RGRAFIC	45
3830	42	IF (KV2(N).GT.KMAX) KV2(N)=KPMX	RGRAFIC	46
3831	43	20 IF (NCPLTS.EQ.0) GO TO 50	RGRAFIC	47
3832	44	30 IF (NCPLTS.EQ.0) GO TO 50	RGRAFIC	48
3833	45	DO 40 N=1,NCPLTS	RGRAFIC	49
3834	46	IF (IC1(N).GT.IMAX) IC1(N)=IPMX	RGRAFIC	50
3835	47	IF (IC2(N).GT.IMAX) IC2(N)=IPMX	RGRAFIC	51
3836	48	IF (JC1(N).GT.JMAX) JC1(N)=JPMX	RGRAFIC	52
3837	49	IF (JC2(N).GT.JMAX) JC2(N)=JPMX	RGRAFIC	52

```

3841 53      IF (KC1(N).GT.KMAX) KC1(N)=KPMX          RGRAFIC 53
3842 54      40 IF (KC2(N).GT.KMAX) KC2(N)=KPMX          RGRAFIC 54
3843 55      50 IF (NSPLTS.EQ.0) GO TO 70             RGRAFIC 55
3844 56      DO 60 N=1,NSPLTS                         RGRAFIC 56
3845 57      IF (IS1(N).GT.IMAX) IS1(N)=IPMX          RGRAFIC 57
3846 58      IF (IS2(N).GT.IMAX) IS2(N)=IPMX          RGRAFIC 58
3847 59      IF (JS1(N).GT.JMAX) JS1(N)=JPMX          RGRAFIC 59
3848 60      IF (JS2(N).GT.JMAX) JS2(N)=JPMX          RGRAFIC 60
3849 61      IF (KS1(N).GT.KMAX) KS1(N)=KPMX          RGRAFIC 61
3850 62      60 IF (KS2(N).GT.KMAX) KS2(N)=KPMX          RGRAFIC 62
3851 63      70 CONTINUE                            RGRAFIC 63
3852 64 C
3853 65 C * * PRINT/WRITE(4020) PLOTTING INPUT DATA RGRAFIC 64
3854 66 C
3855 67      KPR=9                                RGRAFIC 65
3856 68      ASSIGN 90 TO KRET                      RGRAFIC 66
3857 69      80 WRITE (KPR,110)                      RGRAFIC 67
3858 70      WRITE (KPR,120) (XEA(N),YEA(N),ZEA(N),XCA(N),YCA(N),ZCA(N),N=1 RGRAFIC 68
3859 71      1 ,NVEWS)                             RGRAFIC 69
3860 72      WRITE (KPR,130) (IVVIEW(N),IV1(N),IV2(N),JV1(N),JV2(N),KV1(N),KV2(N RGRAFIC 70
3861 73      1 ),NAV(N),IPERV(N),N=1,NVPLTS)        RGRAFIC 71
3862 74      WRITE (KPR,140) (ICVIEW(N),IC1(N),IC2(N),JC1(N),JC2(N),KC1(N),KC2(N RGRAFIC 72
3863 75      1 ),NAC(N),IPERC(N),N=1,NCPLTS)        RGRAFIC 73
3864 76      WRITE (KPR,150) (ISVIEW(N),IS1(N),IS2(N),JS1(N),JS2(N),KS1(N),KS2(N RGRAFIC 74
3865 77      1 ),NAS(N),IPERS(N),N=1,NSPLTS)        RGRAFIC 75
3866 78      GO TO KRET, (90,100)                   RGRAFIC 76
3867 79      90 ASSIGN 100 TO KRET                  RGRAFIC 77
3868 80      KPR=12                               RGRAFIC 78
3869 81      GO TO 80
3870 82      100 CONTINUE                          RGRAFIC 79
3871 83      RETURN                               RGRAFIC 80
3872 84 C
3873 85      110 FORMAT (1HO,3OX,32H PLOTTING OUTPUT SPECIFICATIONS//) RGRAFIC 81
3874 86      120 FORMAT (38HO EYE POINT-VIEW PLANE TABLE/(5H XE=,1PE12.5 RGRAFIC 82
3875 87      1 .5H YE=,E12.5,5H ZE=,E12.5,5H XC=,E12.5,5H YC=,E12.5,5H ZC= RGRAFIC 83
3876 88      2 ,E12.5))                           RGRAFIC 84
3877 89      130 FORMAT (32HO VELOCITY VECTOR PLOTS/(8H IVVIEW=,I3,6H IV1 RGRAFIC 85
3878 90      1 =,I3,6H IV2=,I3,6H JV1=,I3,6H JV2=,I3,6H KV1=,I3,6H KV2=,I3,6 RGRAFIC 86
3879 91      2 H NAV=,I3,8H IPERV=,I3))           RGRAFIC 87
3880 92      140 FORMAT (24HO CONTOUR PLOTS /(8H ICVIEW=,I3,6H IC1=,I3,6H RGRAFIC 88
3881 93      1 IC2=,I3,6H JC1=,I3,6H JC2=,I3,6H KC1=,I3,6H KC2=,I3,6H NAC= RGRAFIC 89
3882 94      2 ,I3,8H IPERC=,I3))                RGRAFIC 90
3883 95      150 FORMAT (35HO PERSPECTIVE SURFACE PLOTS/(8H ISVIEW=,I3,6H RGRAFIC 91
3884 96      IIS1=,I3,6H IS2=,I3,6H JS1=,I3,6H JS2=,I3,6H KS1=,I3,6H KS2= RGRAFIC 92
3885 97      2 ,I3,6H NAS=,I3,8H IPERS=,I3))       RGRAFIC 93
3886 98      END                                  RGRAFIC 94
3887 1 *DK RINPUT                                RINPUT 95
3888 2 SUBROUTINE RINPUT                         RINPUT 96
3889 3 *CA SLCOM1                                RINPUT 97
3890 4 C
3891 5 C * * READ PROBLEM INPUT DATA            RINPUT 98
3892 6 C
3893 7      NAMELIST /XPUT/ ALPHA,PLTDT,PRTDT,CYL,DELT,EPSI,GX,GY,GZ,ICSURF RINPUT 99
3894 8      1 .IEQUIB,JNM,LPR,NAME,NFCAL,NU,OMG,TDDT,TD,TLIMD,AUTOT,SIGMA RINPUT 100
3895 9      2 .CANGLE,IDEFM,ISOR,ISURFT,T,RADPS,NOWALL,TWFIN,UI,VELMX,VI,WB,WBK RINPUT 101
3896 10     3 .WF,WI,WL,WR,WT,FLHT,IORDER,ICLIP,IZOOM,RHOF RINPUT 102
3897 11 C
3898 12      DATA KTRAN /1/                      RINPUT 103
3899 13 C * * DEFAULT VALUES FOR NAMELIST / XPUT / . RINPUT 104
3900 14 C
3901 15      GO TO (10,20). KTRAN                 RINPUT 105
3902 16     10 CONTINUE                            RINPUT 106
3903 17      ALPHA=1.0                            RINPUT 107
3904 18      AUTOT=1.0                            RINPUT 108
3905 19      PLTDT=1.0                            RINPUT 109
3906 20      PRTDT=1.0                            RINPUT 110
3907 21      CYL=0.0                              RINPUT 111
3908 22      DELT=0.02                            RINPUT 112
3909 23      EPSI=0.001                           RINPUT 113
3910 24      GX=0.0                              RINPUT 114
3911 25      GY=0.0                              RINPUT 115
3912 26      GZ=0.0                              RINPUT 116
3913 27      ICLIP=0                            RINPUT 117
3914 28      ICSURF=0                           RINPUT 118
3915 29      IEQUIB=0                           RINPUT 119
3916 30      IORDER=1                           RINPUT 120
3917 31      IDEFM=0                            RINPUT 121
3918 32      ISOR=0                             RINPUT 122
3919 33      IZOOM=0                            RINPUT 123
3920 34      JNM=BH RUN 1                        RINPUT 124

```

3921	35	LPR=2	RINPUT	35			
3922	36	NAME(1)=8HPROB. NO	RINPUT	36			
3923	37	NAME(2)=8HNAME	RINPUT	37			
3924	38	NFCAL=3	RINPUT	38			
3925	39	NOWALL=1	RINPUT	39			
3926	40	NU=0.0	RINPUT	40			
3927	41	OMG=1.0	RINPUT	41			
3928	42	FLHT=1.0	RINPUT	42			
3929	43	RADPS=0.0	RINPUT	43			
3930	44	RHOFF=1.0	RINPUT	44			
3931	45	SIGMA=0.0	RINPUT	45			
3932	46	ISURFT=0	RINPUT	46			
3933	47	CANGLE=0.0	RINPUT	47			
3934	48	T=0.0	RINPUT	48			
3935	49	TDDT=+10	RINPUT	49			
3936	50	TD=-1.0	RINPUT	50			
3937	51	TLIMD=1.0	RINPUT	51			
3938	52	TWFIN=10.0	RINPUT	52			
3939	53	UI=0.0	RINPUT	53			
3940	54	VELMX=2.0	RINPUT	54			
3941	55	VI=0.0	RINPUT	55			
3942	56	WB=1	RINPUT	56			
3943	57	WBK=1	RINPUT	57			
3944	58	WF=1	RINPUT	58			
3945	59	WI=0.0	RINPUT	59			
3946	60	WL=1	RINPUT	60			
3947	61	WR=1	RINPUT	61			
3948	62	WT=1	RINPUT	62			
3949	63	20 CONTINUE	RINPUT	63			
3950	64	C	*	*	READ INPUT DATA FROM NAMELIST /XPUT /.	RINPUT	64
3951	65	C	*	*	READ (10,XPUT)	RINPUT	65
3952	66	C	*	*	IF (CANGLE.EQ.90.0) CANGLE=CANGLE-EM6	RINPUT	66
3953	67		*	*	CANGLE=CANGLE*0.0174532925	RINPUT	67
3954	68	C	*	*	TANCA=TAN(CANGLE)	RINPUT	68
3955	69		*	*	SANG=SIN(CANGLE)	RINPUT	69
3956	70		*	*	CSANG=COS(CANGLE)	RINPUT	70
3957	71		*	*	74 C * * SET UP FILM (CGSCFT) PACKAGE	RINPUT	71
3958	72		*	*	75 C	RINPUT	72
3959	73		*	*	GO TO (30,40), KTRAN	RINPUT	73
3960	74	C	*	*	30 KTRAN=2	RINPUT	74
3961	75	C	*	*	CALL GRPHCFT	RINPUT	75
3962	76		*	*	CALL GPLOT (1HU, 14HT3MDT SOLA-3DV, 14)	RINPUT	76
3963	77		*	*	CALL LIB4020	RINPUT	77
3964	78		*	*	CALL GRPHLUN (12,4HPLOT)	RINPUT	78
3965	79		*	*	CALL SETFLSH	RINPUT	79
3966	80		*	*	83 40 CONTINUE	RINPUT	80
3967	81		*	*	84 C * * WRITE EFFECTIVE DATA FROM XPUT TO TAPES 9 AND 12.	RINPUT	81
3968	82		*	*	85 C * * IF (LPR.EQ.0) GO TO 70	RINPUT	82
3969	83		*	*	86 C	RINPUT	83
3970	84	C	*	*	87 IF (LPR.EQ.0) GO TO 70	RINPUT	84
3971	85	C	*	*	88 C	RINPUT	85
3972	86	C	*	*	89 KPR=9	RINPUT	86
3973	87		*	*	90 ASSIGN 60 TO KRET	RINPUT	87
3974	88	C	*	*	91 50 WRITE (KPR,80) (NAME(I),I=1,8)	RINPUT	88
3975	89		*	*	92 WRITE (KPR,90) ALPHA	RINPUT	89
3976	90		*	*	93 WRITE (KPR,430) AUTOT	RINPUT	90
3977	91		*	*	94 WRITE (KPR,480) CANGLE	RINPUT	91
3978	92		*	*	95 WRITE (KPR,120) CYL	RINPUT	92
3979	93		*	*	96 WRITE (KPR,130) DELT	RINPUT	93
3980	94		*	*	97 WRITE (KPR,140) EPSI	RINPUT	94
3981	95		*	*	98 WRITE (KPR,360) FLHT	RINPUT	95
3982	96		*	*	99 WRITE (KPR,150) GX	RINPUT	96
3983	97		*	*	100 WRITE (KPR,160) GY	RINPUT	97
3984	98		*	*	101 WRITE (KPR,170) GZ	RINPUT	98
3985	99		*	*	102 WRITE (KPR,380) ICLIP	RINPUT	99
3986	100		*	*	103 WRITE (KPR,390) ICSURF	RINPUT	100
3987	101		*	*	104 WRITE (KPR,490) IDEFM	RINPUT	101
3988	102		*	*	105 WRITE (KPR,400) IEQUIB	RINPUT	102
3989	103		*	*	106 WRITE (KPR,500) ISOR	RINPUT	103
3990	104		*	*	107 WRITE (KPR,370) IORDER	RINPUT	104
3991	105		*	*	108 WRITE (KPR,510) ISURFT	RINPUT	105
3992	106		*	*	109 WRITE (KPR,420) IZOOM	RINPUT	106
3993	107		*	*	110 WRITE (KPR,180) JNM	RINPUT	107
3994	108		*	*	111 WRITE (KPR,190) LPR	RINPUT	108
3995	109		*	*	112 WRITE (KPR,410) NFCAL	RINPUT	109
3996	110		*	*	113 WRITE (KPR,450) NDWALL	RINPUT	110
3997	111		*	*	114 WRITE (KPR,200) NU	RINPUT	111

4001	115	WRITE (KPR, 210) OMG	RINPUT	115
4002	116	WRITE (KPR, 100) PLTDT	RINPUT	116
4003	117	WRITE (KPR, 110) PRTDT	RINPUT	117
4004	118	WRITE (KPR, 440) RADPS	RINPUT	118
4005	119	WRITE (KPR, 460) RHOF	RINPUT	119
4006	120	WRITE (KPR, 470) SIGMA	RINPUT	120
4007	121	WRITE (KPR, 520) T	RINPUT	121
4008	122	WRITE (KPR, 220) TDDT	RINPUT	122
4009	123	WRITE (KPR, 230) TD	RINPUT	123
4010	124	WRITE (KPR, 240) TLIMD	RINPUT	124
4011	125	WRITE (KPR, 250) TWFIN	RINPUT	125
4012	126	WRITE (KPR, 260) UI	RINPUT	126
4013	127	WRITE (KPR, 270) VELMX	RINPUT	127
4014	128	WRITE (KPR, 280) VI	RINPUT	128
4015	129	WRITE (KPR, 290) WB	RINPUT	129
4016	130	WRITE (KPR, 300) WBK	RINPUT	130
4017	131	WRITE (KPR, 310) WF	RINPUT	131
4018	132	WRITE (KPR, 320) WI	RINPUT	132
4019	133	WRITE (KPR, 330) WL	RINPUT	133
4020	134	WRITE (KPR, 340) WR	RINPUT	134
4021	135	WRITE (KPR, 350) WT	RINPUT	135
4022	136	GO TO KRET, (60,70)	RINPUT	136
4023	137	60 ASSIGN 70 TO KRET	RINPUT	137
4024	138	KPR=12	RINPUT	138
4025	139	GO TO 50	RINPUT	139
4026	140	70 CONTINUE	RINPUT	140
4027	141	RETURN	RINPUT	141
4028	142 C		RINPUT	142
4029	143	80 FORMAT (1H1,10A8,//)	RINPUT	143
4030	144	90 FORMAT (10X,10H ALPHA= ,1PE12.5)	RINPUT	144
4031	145	100 FORMAT (10X,10H PLTDT= ,1PE12.5)	RINPUT	145
4032	146	110 FORMAT (10X,10H PRTDT= ,1PE12.5)	RINPUT	146
4033	147	120 FORMAT (10X,10H CYL= ,1PE12.5)	RINPUT	147
4034	148	130 FORMAT (10X,10H DELT= ,1PE12.5)	RINPUT	148
4035	149	140 FORMAT (10X,10H EPSI= ,1PE12.5)	RINPUT	149
4036	150	150 FORMAT (10X,10H GX= ,1PE12.5)	RINPUT	150
4037	151	160 FORMAT (10X,10H GY= ,1PE12.5)	RINPUT	151
4038	152	170 FORMAT (10X,10H GZ= ,1PE12.5)	RINPUT	152
4039	153	180 FORMAT (10X,10H JNM= ,A8)	RINPUT	153
4040	154	190 FORMAT (10X,10H LPR= ,I2)	RINPUT	154
4041	155	200 FORMAT (10X,10H NU= ,1PE12.5)	RINPUT	155
4042	156	210 FORMAT (10X,10H OMG= ,1PE12.5)	RINPUT	156
4043	157	220 FORMAT (10X,10H TDDT= ,1PE12.5)	RINPUT	157
4044	158	230 FORMAT (10X,10H TD= ,I4)	RINPUT	158
4045	159	240 FORMAT (10X,10H TLIMD= ,1PE12.5)	RINPUT	159
4046	160	250 FORMAT (10X,10H TWFIN= ,1PE12.5)	RINPUT	160
4047	161	260 FORMAT (10X,10H UI= ,1PE12.5)	RINPUT	161
4048	162	270 FORMAT (10X,10H VELMX= ,1PE12.5)	RINPUT	162
4049	163	280 FORMAT (10X,10H VI= ,1PE12.5)	RINPUT	163
4050	164	290 FORMAT (10X,10H WB= ,I2)	RINPUT	164
4051	165	300 FORMAT (10X,10H WBK= ,I2)	RINPUT	165
4052	166	310 FORMAT (10X,10H WF= ,I2)	RINPUT	166
4053	167	320 FORMAT (10X,10H WI= ,1PE12.5)	RINPUT	167
4054	168	330 FORMAT (10X,10H WL= ,I2)	RINPUT	168
4055	169	340 FORMAT (10X,10H WR= ,I2)	RINPUT	169
4056	170	350 FORMAT (10X,10H WT= ,I2)	RINPUT	170
4057	171	360 FORMAT (10X,10H FLHT= ,1PE12.5)	RINPUT	171
4058	172	370 FORMAT (10X,10H IQORDER= ,I2)	RINPUT	172
4059	173	380 FORMAT (10X,10H ICLIP= ,I2)	RINPUT	173
4060	174	390 FORMAT (10X,10H ICSURF= ,I2)	RINPUT	174
4061	175	400 FORMAT (10X,10H IEQUIB= ,I2)	RINPUT	175
4062	176	410 FORMAT (10X,10H NFCAL= ,I2)	RINPUT	176
4063	177	420 FORMAT (10X,10H IZOOM= ,I2)	RINPUT	177
4064	178	430 FORMAT (10X,10H AUTOT= ,1PE12.5)	RINPUT	178
4065	179	440 FORMAT (10X,10H RADPS= ,1PE12.5)	RINPUT	179
4066	180	450 FORMAT (10X,10H NOWALL= ,I2)	RINPUT	180
4067	181	460 FORMAT (10X,10H RHOF= ,1PE12.5)	RINPUT	181
4068	182	470 FORMAT (10X,10H SIGMA= ,1PE12.5)	RINPUT	182
4069	183	480 FORMAT (10X,10H CANGLE= ,1PE12.5)	RINPUT	183
4070	184	490 FORMAT (10X,10H IDEFM= ,I2)	RINPUT	184
4071	185	500 FORMAT (10X,10H ISOR= ,I2)	RINPUT	185
4072	186	510 FORMAT (10X,10H ISURFT= ,I2)	RINPUT	186
4073	187	520 FORMAT (10X,10H T= ,1PE12.5)	RINPUT	187
4074	188	END	RINPUT	188
4075	1	*DK SECORD	SECORD	1
4076	2	SUBROUTINE SECORD	SECORD	2
4077	3	*CA SLCOM1	SECORD	3
4078	4 C		SECORD	4
4079	5 C	* * SECOND ORDER OPTION	SECORD	5
4080	6 C		SECORD	6

```

4081    7      DO 20 J=1,JMAX          SECORD   7
4082    8      DO 20 I=1,IMAX          SECORD   8
4083    9      KT1=KT1+1             SECORD   9
4084   10      DO 10 K=1,KMAX          SECORD  10
4085   11      CALL CALCIJK          SECORD  11
4086   12      UDUM=UN(IJK)          SECORD  12
4087   13      UN(IJK)=U(IJK)          SECORD  13
4088   14      U(IJK)=UDUM          SECORD  14
4089   15      VDUM=VN(IJK)          SECORD  15
4090   16      VN(IJK)=V(IJK)          SECORD  16
4091   17      V(IJK)=VDUM          SECORD  17
4092   18      WDUM=WN(IJK)          SECORD  18
4093   19      WN(IJK)=W(IJK)          SECORD  19
4094   20      W(IJK)=WDUM          SECORD  20
4095   21      10 CONTINUE          SECORD  21
4096   22      20 CONTINUE          SECORD  22
4097   23      ALPHA=-1.0          SECORD  23
4098   24      AVE=0.5           SECORD  24
4099   25 C
4100   26      RETURN          SECORD  25
4101   27      END          SECORD  26
4102   1 *DK SETACC          SETACC  1
4103   2      SUBROUTINE SETACC      SETACC  2
4104   3 C
4105   4 *CA SLCOM1          SETACC  3
4106   5      IJ=I+(J-1)*IMAX      SETACC  4
4107   6      GXA(IJ)=GX*CTHJ(J)+GY*STHJ(J)      SETACC  5
4108   7      GYA(IJ)=GX*STHJBK(J)+GY*CTHJBK(J)      SETACC  6
4109   8      RETURN          SETACC  7
4110   9      END          SETACC  8
4111   1 *DK SETFS           SETFS  9
4112   2      SUBROUTINE SETFS        SETFS 10
4113   3 C
4114   4 *CA SLCOM1          SETFS 11
4115   5      DIMENSION IFLG(5), DIS(4), XM(5), ZM(5)      SETFS 12
4116   6 C
4117   7      NAMELIST /FLUIDGN/ NQBS,QA3,QA2,QA1,QB3,QB2,QB1,QC3,QC2,QC1,QD3      SETFS 13
4118   8      1 ,QD2, QD1, IQH          SETFS 14
4119   9      DATA NQBS /0/          SETFS 15
4120  10      DATA QA2, QA1, QB2, QB1, QC2, QC1, IQH /10*0.0, 10*0.0, 10*0.0, 10*0.0, 10*0.0, 10*0.0, 10*0.0, 10*0.0/      SETFS 16
4121  11      1 0, 10*0.0, 10*0.0, 10*0.0/          SETFS 17
4122  12      DATA QA3, QB3, QC3, QD3, QD2, QD1 /10*0.0, 10*0.0, 10*0.0, 10*0.0, 10*0.0, 10*0.0/      SETFS 18
4123  13      1 0.0, 10*0.0/          SETFS 19
4124  14 C
4125  15 C * * READ IN DATA FROM NAMELIST / FLUIDGN /
4126  16 C
4127  17      READ (10,FLUIDGN)
4128  18 C
4129  19 C * * WRITE INPUT DATA FOR FLUID GENERATOR ONTO TAPES 9 AND 12.
4130  20 C
4131  21      IF (NQBS.LE.0) GO TO 410          SETFS 21
4132  22      IF (LPR.EQ.0) GO TO 20          SETFS 22
4133  23      WRITE (9,450)          SETFS 23
4134  24      WRITE (12,450)          SETFS 24
4135  25      WRITE (9,420) NQBS          SETFS 25
4136  26      WRITE (12,420) NQBS          SETFS 26
4137  27      DO 10 I=1,NQBS          SETFS 27
4138  28      WRITE (9,430) I,QA2(I),QA1(I),QB2(I),QB1(I),QC2(I),QC1(I),IQH(I)      SETFS 28
4139  29      WRITE (9,440) I,QA3(I),QB3(I),QC3(I),QD3(I),QD2(I),QD1(I)      SETFS 29
4140  30      WRITE (12,430) I,QA2(I),QA1(I),QB2(I),QB1(I),QC2(I),QC1(I),IQH(I)      SETFS 30
4141  31      WRITE (12,440) I,QA3(I),QB3(I),QC3(I),QD3(I),QD2(I),QD1(I)      SETFS 31
4142  32      10 CONTINUE          SETFS 32
4143  33 C
4144  34      20 IF (ICSURF.GT.1) GO TO 300          SETFS 33
4145  35 C
4146  36 C *** THIS PORTION FOR AXISYMMETRIC SURFACES          SETFS 34
4147  37 C *** CONIC FCN=QA2*X*X+QA1*X+QB2*Z*Z+QB1*Z+QC2*X*Z+QC1          SETFS 35
4148  38 C *** INSIDE FCN=NEGATIVE VALUE          SETFS 36
4149  39 C *** IQH=1 ADD FLUID INSIDE FCN, IQH=0 SUBTRACT FLUID INSIDE FCN          SETFS 37
4150  40 C ***          SETFS 38
4151  41      J=2          SETFS 39
4152  42      DO 250 L=1,NQBS          SETFS 40
4153  43      DO 240 K=2,KM1          SETFS 41
4154  44      DO 240 I=2,IM1          SETFS 42
4155  45      CALL IJKONLY          SETFS 43
4156  46      RDXDZ=1.0/(DELX(I)*DELZ(K))          SETFS 44
4157  47      DO 80 M=1,4          SETFS 45
4158  48      GO TO (30,40,50,60), M          SETFS 46
4159  49      30 X1=X(I)          SETFS 47
4160  50      Z1=Z(K-1)          SETFS 48

```

```

4161 51      DIS(1)=DELZ(K)          SETFS 51
4162 52      GO TO 70             SETFS 52
4163 53      40 Z1=Z(K)           SETFS 53
4164 54      X1=X(I)            SETFS 54
4165 55      DIS(2)=DELX(I)        SETFS 55
4166 56      GO TO 70             SETFS 56
4167 57      50 X1=X(I-1)         SETFS 57
4168 58      Z1=Z(K)            SETFS 58
4169 59      DIS(3)=DELZ(K)        SETFS 59
4170 60      GO TO 70             SETFS 60
4171 61      60 Z1=Z(K-1)         SETFS 61
4172 62      X1=X(I-1)           SETFS 62
4173 63      DIS(4)=DELX(I)        SETFS 63
4174 64      70 IFLG(M)=0          SETFS 64
4175 65      FCONIC=QA2(L)*X1*X1+QA1(L)*X1+QB2(L)*Z1*Z1+QB1(L)*Z1+QC2(L)*X1*Z1
4176 66      +QC1(L)              SETFS 65
4177 67      IF (FCONIC.LE.O.O) IFLG(M)=1 SETFS 66
4178 68      XM(M)=X1            SETFS 67
4179 69      ZM(M)=Z1            SETFS 68
4180 70      80 CONTINUE          SETFS 69
4181 71      IFLG(5)=IFLG(1)        SETFS 70
4182 72      XM(5)=XM(1)          SETFS 71
4183 73      ZM(5)=ZM(1)          SETFS 72
4184 74      IFLGS=0              SETFS 73
4185 75      DO 90 M=1,4           SETFS 74
4186 76      90 IFLGS=IFLGS+IFLG(M) SETFS 75
4187 77      BRIK=0.0             SETFS 76
4188 78      BTIK=0.0             SETFS 77
4189 79      IF (IFLGS.EQ.0) GO TO 240 SETFS 78
4190 80      IF (IFLGS.LT.4) GO TO 100 SETFS 79
4191 81      BIK=1.0              SETFS 80
4192 82      BRIK=1.0              SETFS 81
4193 83      BTIK=1.0              SETFS 82
4194 84      GO TO 220             SETFS 83
4195 85      100 IF (IFLG(1).EQ.1.AND.IFLG(2).EQ.1) BRIK=1.0 SETFS 84
4196 86      IF (IFLG(2).EQ.1.AND.IFLG(3).EQ.1) BTIK=1.0 SETFS 85
4197 87      DO 180 M=1,4           SETFS 86
4198 88      IF (IFLG(M).EQ.IFLG(M+1)) GO TO 180 SETFS 87
4199 89      X1=XM(M)             SETFS 88
4200 90      Z1=ZM(M)             SETFS 89
4201 91      X2=XM(M+1)           SETFS 90
4202 92      Z2=ZM(M+1)           SETFS 91
4203 93      IF (IFLG(M).EQ.0) GO TO 110 SETFS 92
4204 94      X2=XM(M)             SETFS 93
4205 95      Z2=ZM(M)             SETFS 94
4206 96      X1=XM(M+1)           SETFS 95
4207 97      Z1=ZM(M+1)           SETFS 96
4208 98      110 EPSIF=0.001*(ABS(X2-X1)+ABS(Z2-Z1)) SETFS 97
4209 99      SMN=0.0              SETFS 98
4210 100     FMN=QA2(L)*X2*X2+QA1(L)*X2+QB2(L)*Z2*Z2+QB1(L)*Z2+QC2(L)*X2*Z2+QC1
4211 101     1 (L)                  SETFS 99
4212 102     SMX=1.0              SETFS 100
4213 103     FMX=QA2(L)*X1*X1+QA1(L)*X1+QB2(L)*Z1*Z1+QB1(L)*Z1+QC2(L)*X1*Z1+QC1
4214 104     1 (L)                  SETFS 101
4215 105     S=0.5                SETFS 102
4216 106     120 XT=S*X1+(1.0-S)*X2 SETFS 103
4217 107     ZT=S*Z1+(1.0-S)*Z2 SETFS 104
4218 108     FS=QA2(L)*XT*XT+QA1(L)*XT+QB2(L)*ZT*ZT+QB1(L)*ZT+QC2(L)*XT*ZT+QC1
4219 109     1 (L)                  SETFS 105
4220 110     IF (ABS(FS).LT.EPSIF) GO TO 150 SETFS 106
4221 111     IF (FS.GE.0.0) GO TO 130 SETFS 107
4222 112     FDEN=ABS(FS-FMN)+1.0E-10 SETFS 108
4223 113     SE=S-FS*(S-SMN)/FDEN SETFS 109
4224 114     IF (SE.GT.SMX) SE=SMX SETFS 110
4225 115     FMN=FS              SETFS 111
4226 116     SMN=S              SETFS 112
4227 117     GO TO 140             SETFS 113
4228 118     130 FDEN=ABS(FMX-FS)+1.0E-10 SETFS 114
4229 119     SE=S-FS*(SMX-S)/FDEN SETFS 115
4230 120     IF (SE.LT.SMN) SE=SMN SETFS 116
4231 121     FMX=FS              SETFS 117
4232 122     SMX=S              SETFS 118
4233 123     140 SI=S-FS*(SMX-SMN)/(FMX-FMN) SETFS 119
4234 124     S=0.5*(SE+SI)        SETFS 120
4235 125     GO TO 120             SETFS 121
4236 126     150 DIS(M)=SQRT((XT-X2)**2+(ZT-Z2)**2) SETFS 122
4237 127     GO TO (160,170,180,180), M SETFS 123
4238 128     160 BRIK=DIS(1)/DELZ(K) SETFS 124
4239 129     GO TO 180             SETFS 125
4240 130     170 BTIK=DIS(2)/DELX(I) SETFS 126

```

4241	131	180	CONTINUE	SETFS	131
4242	132		M=0	SETFS	132
4243	133		BIK=0.0	SETFS	133
4244	134	190	CONTINUE	SETFS	134
4245	135		M=M+1	SETFS	135
4246	136		IF (M.EQ.5) GO TO 210	SETFS	136
4247	137		IF (IFLG(M).EQ.0) GO TO 190	SETFS	137
4248	138		MP1=M+1	SETFS	138
4249	139		IF (MP1.EQ.5) MP1=1	SETFS	139
4250	140		MM1=M-1	SETFS	140
4251	141		IF (MM1.EQ.0) MM1=4	SETFS	141
4252	142		BIK=BIK+DIS(M)*DIS(MM1)	SETFS	142
4253	143		IF (IFLG(MP1).EQ.1) GO TO 200	SETFS	143
4254	144		DIS2=DIS(M)	SETFS	144
4255	145	200	CONTINUE	SETFS	145
4256	146		IF (IFLG(MM1).EQ.1) GO TO 190	SETFS	146
4257	147		DIS1=DIS(MM1)	SETFS	147
4258	148		GO TO 190	SETFS	148
4259	149	210	CONTINUE	SETFS	149
4260	150		IF (IFLGS.EQ.3) BIK=BIK-DIS1*DIS2	SETFS	150
4261	151		BIK=0.5*BIK*RDXDZ	SETFS	151
4262	152		IF (BIK.GT.1.0) BIK=1.0	SETFS	152
4263	153	220	CONTINUE	SETFS	153
4264	154		IF (IQH(L).EQ.1) GO TO 230	SETFS	154
4265	155		BIK=-BIK	SETFS	155
4266	156	230	F(IJK)=F(IJK)+BIK	SETFS	156
4267	157		IF (F(IJK).GT.0.99) F(IJK)=1.0	SETFS	157
4268	158		IF (F(IJK).LT.0.01) F(IJK)=0.0	SETFS	158
4269	159	240	CONTINUE	SETFS	159
4270	160	250	CONTINUE	SETFS	160
4271	161	C	NOTE: ALL UPLANES HAVE SAME VALUES	SETFS	161
4272	162	C		SETFS	162
4273	163	C		SETFS	163
4274	164		DO 290 J=1,JMAX	SETFS	164
4275	165		IF (J.EQ.2) GO TO 290	SETFS	165
4276	166		DO 280 K=1,KMAX	SETFS	166
4277	167		DO 270 I=1,IMAX	SETFS	167
4278	168		I2K=NQ*(II5*(K-1)+IMAX+(I-1))+1	SETFS	168
4279	169		IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1	SETFS	169
4280	170		IF (J.EQ.1.AND.WF.LE.2) GO TO 260	SETFS	170
4281	171		IF (J.EQ.JMAX.AND.WBK.LE.2) GO TO 260	SETFS	171
4282	172		F(IJK)=F(I2K)	SETFS	172
4283	173		GO TO 270	SETFS	173
4284	174	260	F(IJK)=0.0	SETFS	174
4285	175	270	CONTINUE	SETFS	175
4286	176	280	CONTINUE	SETFS	176
4287	177	290	CONTINUE	SETFS	177
4288	178	C		SETFS	178
4289	179		GO TO 410	SETFS	179
4290	180	C		SETFS	180
4291	181	C	+++ THIS PORTION FOR NON-AXISYMMETRIC SURFACES	SETFS	181
4292	182	C		SETFS	182
4293	183	C	+++ FCN=(X-QA1)**2+(Y-QB1)**2+(Z-QC1)**2+QD1	SETFS	183
4294	184	C		SETFS	184
4295	185	C	FOR GENERAL SURFACES CHANGE DEFINITION OF FCN	SETFS	185
4296	186	C		SETFS	186
4297	187	C	QA2,QA3 ARE USED IN THIS FORMULATION TO DENOTE APPROXIMATE	SETFS	187
4298	188	C	UPPER AND LOWER LIMITS FOR I INDEX OF MESH CONTAING SURFACE FOR	SETFS	188
4299	189	C	EFFICIENCY PURPOSES. QB3,QB2 FOR J INDEX, QC3,QC2 FOR K INDEX	SETFS	189
4300	190	C		SETFS	190
4301	191	C	QD2 IS USED TO INDICATE IF MESH IS INITIALLY FLUID OR VOID	SETFS	191
4302	192	C	=1.0 FLUID, =0.0 VOID	SETFS	192
4303	193	C		SETFS	193
4304	194	C	+++ INSIDE FCN=NEGATIVE VALUE	SETFS	194
4305	195	C	+++ IQH=1 ADD FLUID INSIDE FCN, IQH=0 SUBTRACT FLUID INSIDE FCN	SETFS	195
4306	196	C	+++	SETFS	196
4307	197	C		SETFS	197
4308	198		DATA LMAX, MMAX, NMAX /20,40,20/	SETFS	198
4309	199		DATA SUMVOID, SUMLIQD, SUMVOL /0.0,0.0,0.0/	SETFS	199
4310	200	C		SETFS	200
4311	201	300	RXIM1=1.0/X(IM1)	SETFS	201
4312	202		FLMAX=1.0/FLOAT(LMAX)	SETFS	202
4313	203		FMMAX=1.0/FLOAT(MMAX)	SETFS	203
4314	204		FNMAX=1.0/FLOAT(NMAX)	SETFS	204
4315	205		DO 400 NS=1,NQBS	SETFS	205
4316	206		IBOT=QA2(NS)	SETFS	206
4317	207		ITOP=QA3(NS)	SETFS	207
4318	208		JBOT=QB2(NS)	SETFS	208
4319	209		JTOP=QB3(NS)	SETFS	209
4320	210		KBOT=QC2(NS)	SETFS	210

4321	211	KTOP=QC3(NS)	SETFS	211
4322	212	FIQH=IQH(NS)	SETFS	212
4323	213 C	DO 390 K=1,KMAX	SETFS	213
4324	214	DO 390 J=1,JMAX	SETFS	214
4325	215	DO 390 I=1,IMAX	SETFS	215
4326	216	CALL IJKONLY	SETFS	216
4327	217	IF (NS.GT.1) GO TO 310	SETFS	217
4328	218	F(IJK)=1.0	SETFS	218
4329	219	IF (QD2(NS).EQ.0.0) F(IJK)=0.0	SETFS	219
4330	220	310 IF (I.EQ.1.OR.I.EQ.IMAX) GO TO 390	SETFS	220
4331	221	IF (J.EQ.1.OR.J.EQ.JMAX) GO TO 390	SETFS	221
4332	222	IF (K.EQ.1.OR.K.EQ.KMAX) GO TO 390	SETFS	222
4333	223	VOLLC=DELX(I)*DELZ(K)*DELY(J)/RRI(I)	SETFS	223
4334	224	IF (I.LT.IBOT.OR.I.GT.ITOP) GO TO 380	SETFS	224
4335	225	IF (J.LT.JBOT.OR.J.GT.JTOP) GO TO 380	SETFS	225
4336	226	IF (K.LT.KBOT.OR.K.GT.KTOP) GO TO 380	SETFS	226
4337	227	328 C	SETFS	227
4338	228	CTEST=0.0	SETFS	228
4339	229	DO 320 N=K-1,K	SETFS	229
4340	230	DO 320 M=J-1,J	SETFS	230
4341	231	DO 320 L=I-1,I	SETFS	231
4342	232	X1=X(L)*CTHJBK(M)	SETFS	232
4343	233	Y1=X(L)*STHJBK(M)	SETFS	233
4344	234	Z1=Z(N)	SETFS	234
4345	235	FNC=(X1-QA1(NS))**2+(Y1-QB1(NS))**2+(Z1-QC1(NS))**2+QD1(NS)	SETFS	235
4347	237	CTEST=CTEST+CVMGP(1.0,0.0,FNC)	SETFS	237
4348	238	320 CONTINUE	SETFS	238
4349	239	IF (CTEST.GT.7.5) GO TO 380	SETFS	239
4350	240	IF (CTEST.GT.0.5) GO TO 330	SETFS	240
4351	241	F(IJK)=FIQH	SETFS	241
4352	242	GO TO 380	SETFS	242
4353	243 C	330 SVOLSC=0.0	SETFS	243
4354	244	DELTAY=DELY(J)*FMMAX*RXIM1	SETFS	244
4355	245	DELTAX=DELX(I)*FLMAX	SETFS	245
4356	246	DELTAZ=DELZ(K)*FNMAX	SETFS	246
4357	247	RSC=X(I-1)-0.5*DELTAX	SETFS	247
4358	248	RSD2=X(I-1)	SETFS	248
4359	249	DO 360 L=1,LMAX	SETFS	249
4360	250	RSC=RSC+DELTAX	SETFS	250
4361	251	RSD1=RSD2	SETFS	251
4362	252	RSD2=RSD1+DELTAX	SETFS	252
4363	253	RA=SQRT(0.5*(RSD2**2+RSD1**2))	SETFS	253
4365	255	DELTAZ=DELTAY*RSC	SETFS	255
4366	256	VOLSC=DELTAX*DELTAY*DELTAZ	SETFS	256
4367	257	YSC=Y(J-1)*RSC*RXIM1-0.5*DELTAZ	SETFS	257
4368	258	DO 350 M=1,MMAX	SETFS	258
4369	259	YSC=YSC+DELTAZ	SETFS	259
4370	260	THSC=YSC/RSC	SETFS	260
4371	261	X1=RA*COS(THSC)	SETFS	261
4372	262	Y1=RA*SIN(THSC)	SETFS	262
4373	263	ZSC=Z(K-1)-0.5*DELTAZ	SETFS	263
4374	264	DO 340 N=1,NMAX	SETFS	264
4375	265	ZSC=ZSC+DELTAZ	SETFS	265
4376	266	Z1=ZSC	SETFS	266
4377	267 C	340 CONTINUE	SETFS	267
4378	268	350 CONTINUE	SETFS	268
4379	269 C	360 CONTINUE	SETFS	269
4380	270	SVOLSC=SVOLSC+CVMGP(0.0,VOLSC,FNC)	SETFS	270
4381	271 C	340 CONTINUE	SETFS	271
4382	272	350 CONTINUE	SETFS	272
4383	273	360 CONTINUE	SETFS	273
4384	274	370 IF (F(IJK).LE.1.0) GO TO 370	SETFS	274
4385	275 C	FVOL=FVOL=SVOLSC/VOLLC	SETFS	275
4386	276	IF (IQH(NS).EQ.1) F(IJK)=F(IJK)+FVOL	SETFS	276
4387	277	IF (IQH(NS).EQ.0) F(IJK)=F(IJK)-FVOL	SETFS	277
4388	278	IF (F(IJK).LE.1.0) GO TO 370	SETFS	278
4389	279	F(IJK)=1.0	SETFS	279
4390	280	370 IF (F(IJK).GE.0.0) GO TO 380	SETFS	280
4391	281	F(IJK)=0.0	SETFS	281
4392	282	380 CONTINUE	SETFS	282
4393	283	SUMVOID=SUMVOID+(1.0-F(IJK))*VOLLC	SETFS	283
4394	284	SUMLIQD=SUMLIQD+F(IJK)*VOLLC	SETFS	284
4395	285	SUMVOL=SUMVOL+VOLLC	SETFS	285
4396	286	390 CONTINUE	SETFS	286
4397	287 C	400 CONTINUE	SETFS	287
4398	288	390 CONTINUE	SETFS	288
4399	289 C	400 CONTINUE	SETFS	289
4400	290	400 CONTINUE	SETFS	290

```

4401 291 C          SETFS 291
4402 292          410 RETURN          SETFS 292
4403 293 C          SETFS 293
4404 294          420 FORMAT (2X,6HNOBS= ,I2)          SETFS 294
4405 295          430 FORMAT (2X,2HI=,I1,2X,5HQA2= ,1PE12.5,2X,5HQA1= ,E12.5,2X,5HQB2=          SETFS 295
4406 296          1 ,E12.5,2X,5HQB1= ,E12.5,2X,5HQC2= ,E12.5,2X,5HQC1= ,E12.5,2X,5HQC2=          SETFS 296
4407 297          2H= ,I2)          SETFS 297
4408 298          440 FORMAT (2X,2HI=,I1,2X,5HQA3= ,1PE12.5,2X,5HQB3= ,E12.5,2X,5HQC3=          SETFS 298
4409 299          1 ,E12.5,2X,5HQC3= ,E12.5,2X,5HQC2= ,E12.5,2X,5HQC1= ,E12.5)          SETFS 299
4410 300          450 FORMAT (///20X,25HCONIC FLUID PARAMETERS )          SETFS 300
4411 301          END          SETFS 301
4412 1 *DK SETUP          SETUP 1
4413 2          SUBROUTINE SETUP          SETUP 2
4414 3 *CA SLCOM1          SETUP 3
4415 4 C          SETUP 4
4416 5 C * * SET CONSTANT PARAMETERS          SETUP 5
4417 6 C          SETUP 6
4418 7          PI=3.141592654          SETUP 7
4419 8          EMF=1.OE-6          SETUP 8
4420 9          EMF1=1.0-EMF          SETUP 9
4421 10         FNOC=0.0          SETUP 10
4422 11         FLGC=0.0          SETUP 11
4423 12         NFLGC=0.0          SETUP 12
4424 13         NOCON=0          SETUP 13
4425 14         FLG=0.0          SETUP 14
4426 15         VCHGT=0.0          SETUP 15
4427 16         NUMTD=1          SETUP 16
4428 17         T=0.0          SETUP 17
4429 18         ITER=0          SETUP 18
4430 19         CYCLE=0          SETUP 19
4431 20         IF (IORDER.EQ.2) ALPHA=1.0          SETUP 20
4432 21         TWTD=TDDT          SETUP 21
4433 22         TWPLT=PLTDT          SETUP 22
4434 23         TWPRP=PRPTDT          SETUP 23
4435 24         SIGMA=SIGMA/RHO          SETUP 24
4436 25 C          SETUP 25
4437 26 C * * SPECIAL INPUT DATA          SETUP 26
4438 27 C          SETUP 27
4439 28 C * * SET PRESSURE FIELD AND OTHER ARRAYS TO ZERO          SETUP 28
4440 29 C          SETUP 29
4441 30         DO -10 K=1,KMAX          SETUP 30
4442 31         DO 10 J=1,JMAX          SETUP 31
4443 32         DO 10 I=1,IMAX          SETUP 32
4444 33         CALL IJKONLY          SETUP 33
4445 34         P(IJK)=0.0          SETUP 34
4446 35         PN(IJK)=0.0          SETUP 35
4447 36         F(IJK)=0.0          SETUP 36
4448 37         NF(IJK)=0          SETUP 37
4449 38         PETA(IJK)=0.0          SETUP 38
4450 39         FN(IJK)=0.0          SETUP 39
4451 40         D(IJK)=0.0          SETUP 40
4452 41         10 CONTINUE          SETUP 41
4453 42         DO 20 L=1,100          SETUP 42
4454 43         20 PR(L)=0.0          SETUP 43
4455 44 C          SETUP 44
4456 45 C * * COMPUTE INITIAL VOLUME FRACTION FUNCTION F IN CELLS          SETUP 45
4457 46 C          SETUP 46
4458 47         IF (ICSURF.GT.0) GO TO 70          SETUP 47
4459 48         IF (IEQUIB.EQ.0) GO TO 30          SETUP 48
4460 49         BOND=GZ*X(IM1)**2/SIGMA          SETUP 49
4461 50         CALL EQUIB (BASC(1),BASC(1001),1000,BOND,CANGLE,CYL)          SETUP 50
4462 51         SFLHT=FLHT          SETUP 51
4463 52         30 DO 60 I=1,IMAX          SETUP 52
4464 53         DO 60 J=1,JMAX          SETUP 53
4465 54         IBJK=NQ*(II5+IMAX*(J-1)+I-1)+1          SETUP 54
4466 55         IBJKM=IBJK-II2          SETUP 55
4467 56         DO 50 K=2,KMAX          SETUP 56
4468 57         CALL IJKONLY          SETUP 57
4469 58         F(IJK)=1.0          SETUP 58
4470 59         IF (IEQUIB.EQ.0) GO TO 40          SETUP 59
4471 60         LDCK=1000.*XI(I)/X(IM1)+1.000001          SETUP 60
4472 61         LDCK=MINO(1000,LDCK)          SETUP 61
4473 62         LDCK=MAXO(1,LDCK)          SETUP 62
4474 63         FLHT=SFLHT+BASC(LDCK)*X(IM1)          SETUP 63
4475 64         40 IF (FLHT.GT.Z(K-1).AND.FLHT.LT.Z(K)) F(IJK)=RDZ(K)*(FLHT-Z(K-1))          SETUP 64
4476 65         IF (Z(K-1).GE.FLHT) F(IJK)=0.0          SETUP 65
4477 66         50 CONTINUE          SETUP 66
4478 67         F(IBJKM)=F(IBJK)          SETUP 67
4479 68         60 CONTINUE          SETUP 68
4480 69         GO TO 80          SETUP 69

```

```

4481 70 C * * GENERATE SPECIAL F-FUNCTION (FLUID) CONFIGURATION      SETUP 70
4482 71    70 CALL SETFS                                         SETUP 71
4483 72 C
4484 73 C * * CALCULATE HYDROSTATIC PRESSURE                      SETUP 72
4485 74 C
4486 75    80 DO 90 I=2,IM1                                         SETUP 75
4487 76        DO 90 J=2,JM1                                         SETUP 76
4488 77        DO 90 KK=2,KM1                                         SETUP 77
4489 78        K=KM1-KK+2                                         SETUP 78
4490 79        CALL IJKAJCT                                         SETUP 79
4491 80        P(IJK)=P(IJKP)-GZ*(AMIN1(F(IJKP),0.5)*DELZ(K+1)+AMAX1(F(IJK)-0.5,O  SETUP 80
4492 81            1 .O)*DELZ(K))                                         SETUP 81
4493 82        90 CONTINUE                                         SETUP 82
4494 83 C
4495 84 C * * CALCULATE DELT TO ENSURE VISCOSITY DIFFUSION STABILITY   SETUP 83
4496 85 C
4497 86        DTVIS=DELT*1.OE+10                                     SETUP 86
4498 87        DO 100 K=2,KM1                                         SETUP 87
4499 88        DO 100 J=2,JM1                                         SETUP 88
4500 89        DO 100 I=2,IM1                                         SETUP 89
4501 90        DXSQ=DELX(I)**2                                         SETUP 90
4502 91        DYSQ=DELY(J)**2                                         SETUP 91
4503 92        DZSQ=DELZ(K)**2                                         SETUP 92
4504 93        RDSQ1=DXSQ*DYSQ*DZSQ/(DXSQ+DYSQ+DZSQ)                   SETUP 93
4505 94        RDSQ=RDSQ1/(3.O*NU+DELT*1.OE-10)                         SETUP 94
4506 95        DTVIS=AMIN1(DTVIS,RDSQ)                               SETUP 95
4507 96        100 CONTINUE                                         SETUP 96
4508 97 C
4509 98 C     CALCULATE PARTIAL AREAS                           SETUP 98
4510 99 C
4511 100 CALL ASET                                         SETUP 99
4512 101 C
4513 102 C * * CALCULATE BETA(IJK) FOR MESH                      SETUP 102
4514 103 C
4515 104 CALL BETACAL                                         SETUP 104
4516 105 C
4517 106 C ***** SKIP PRINT ***** REMOVE FOR DEBUG *****           SETUP 106
4518 107 C
4519 108 GO TO 140                                         SETUP 108
4520 109 C
4521 110 C     PRINT BETAS AC AR AT ABK FOR ALL PLANES             SETUP 109
4522 111 KPR=9                                         SETUP 110
4523 112 ASSIGN 130 TO KRET                                         SETUP 111
4524 113 IF (LPR.NE.3) KPR=12                                       SETUP 112
4525 114 IF (LPR.NE.3) ASSIGN 140 TO KRET                         SETUP 113
4526 115 110 WRITE (KPR,160)                                         SETUP 114
4527 116 WRITE (KPR,170)                                         SETUP 115
4528 117 DO 120 K=1,KMAX                                         SETUP 116
4529 118 DO 120 J=1,JMAX                                         SETUP 117
4530 119 DO 120 I=1,IMAX                                         SETUP 118
4531 120 IJK=NQ*(II5*(K-1)+IMAX*(J-1)+(I-1))+1                 SETUP 119
4532 121 WRITE (KPR,180) I,J,K,BETA(IJK),AC(IJK),AR(IJK),AT(IJK),ABK(IJK)   SETUP 120
4533 122 120 CONTINUE                                         SETUP 121
4534 123 GO TO KRET, (130,140)                                     SETUP 122
4535 124 130 ASSIGN 140 TO KRET                         SETUP 123
4536 125 KPR=12                                         SETUP 124
4537 126 GO TO 110                                         SETUP 125
4538 127 140 CONTINUE                                         SETUP 126
4539 128 C * * SET INITIAL VELOCITY FIELDS INTO U,V AND W ARRAYS   SETUP 127
4540 129 C
4541 130 DO 150 J=1,JMAX                                         SETUP 128
4542 131 DO 150 I=1,IMAX                                         SETUP 129
4543 132 DO 150 K=1,KMAX                                         SETUP 130
4544 133 CALL IJKONLY                                         SETUP 131
4545 134 CALL SETACC                                         SETUP 132
4546 135 U(IJK)=O.O                                         SETUP 133
4547 136 V(IJK)=O.O                                         SETUP 134
4548 137 W(IJK)=O.O                                         SETUP 135
4549 138 IF (F(IJK).LE.EMF) GO TO 150                         SETUP 136
4550 139 IF (BETA(IPJK).GT.0.O.O.AND.BETA(IPJK).NE.1.O) U(IJK)=UI   SETUP 137
4551 140 IF (BETA(IJKP).GT.0.O.O.AND.BETA(IJKP).NE.1.O) V(IJK)=VI   SETUP 138
4552 141 IF (BETA(IJKP).GT.0.O.O.AND.BETA(IJKP).NE.1.O) W(IJK)=WI   SETUP 139
4553 142 150 CONTINUE                                         SETUP 140
4554 143 C
4555 144 RETURN                                         SETUP 141
4556 145 C
4557 146 160 FORMAT (1H1)                                         SETUP 142
4558 147 170 FORMAT (4X,1HI,4X,1HU,4X,1HK,7X,4HBETA,12X,2HAC,13X,2HAR,13X,2HAT,   SETUP 143
4559 148          1 13X,3HABK)                                         SETUP 144
4560 149 180 FORMAT (3I5,1P5E15.5)                               SETUP 145

```

4561	150	END	SETUP	150
4562	1	*DK SOLA	SOLA	1
4563	2	SUBROUTINE SOLA	SOLA	2
4564	3	*CA SLCOM1	SOLA	3
4565	4	C	SOLA	4
4566	5	C * * NUMERICAL SOLUTION ALGORITHM	SOLA	5
4567	6	C	SOLA	6
4568	7	MSGCODE=0	SOLA	7
4569	8	C * * REQUEST THAT MSGCODE BE SET TO 1	SOLA	8
4570	9	C * * WHEN A MESSAGE FROM A CONTROLLER ARRIVES.	SOLA	9
4571	10	CALL MSGFLAG (MSGCODE,2)	SOLA	10
4572	11	CALL SECOND (STIM)	SOLA	11
4573	12	CALL FADV (1)	SOLA	12
4574	13	WRITE (59,190)	SOLA	13
4575	14	TRL=STIM	SOLA	14
4576	15	T1=TRL	SOLA	15
4577	16	T2=T1	SOLA	16
4578	17	CALL GETJTL (TL)	SOLA	17
4579	18	TLM=TL-150.0+(1.0-TLIMD)*1.0E+10	SOLA	18
4580	19	IF (TD.GT.0) GO TO 160	SOLA	19
4581	20	C	SOLA	20
4582	21	C * * SET INITIAL BOUNDARY CONDITIONS	SOLA	21
4583	22	C	SOLA	22
4584	23	CALL BC	SOLA	23
4585	24	C	SOLA	24
4586	25	GO TO 40	SOLA	25
4587	26	C	SOLA	26
4588	27	C * * START TIME CYCLE	SOLA	27
4589	28	C	SOLA	28
4590	29	10 CONTINUE	SOLA	29
4591	30	T=T+DELT	SOLA	30
4592	31	AVE=1.0	SOLA	31
4593	32	CYCLE=CYCLE+1	SOLA	32
4594	33	IF (NFLGC.GE.200.OR.NOCON.GE.25) T=1.0E+10	SOLA	33
4595	34	IF ((MOD(CYCLE,20).EQ.0) WRITE (59,180) T,CYCLE,ITER,DELT,EPSI	SOLA	34
4596	35	ITER=0	SOLA	35
4597	36	FLG=1.0	SOLA	36
4598	37	FNOC=0	SOLA	37
4599	38	20 CONTINUE	SOLA	38
4600	39	C	SOLA	39
4601	40	C * * EXPLICITLY APPROXIMATE NEW TIME LEVEL VELOCITIES	SOLA	40
4602	41	C	SOLA	41
4603	42	CALL TILDE	SOLA	42
4604	43	C	SOLA	43
4605	44	C * * SET BOUNDARY CONDITIONS	SOLA	44
4606	45	C	SOLA	45
4607	46	CALL BC	SOLA	46
4608	47	C	SOLA	47
4609	48	C * * SECOND ORDER OPTION	SOLA	48
4610	49	C	SOLA	49
4611	50	IF (IORDER.EQ.1.OR.AVE.LT.0.75) GO TO 30	SOLA	50
4612	51	CALL SECORD	SOLA	51
4613	52	GO TO 20	SOLA	52
4614	53	C	SOLA	53
4615	54	C * * ITERATIVELY ADJUST CELL PRESSURE AND VELOCITY	SOLA	54
4616	55	C	SOLA	55
4617	56	30 IF (ISOR.EQ.1) CALL PRESSIT	SOLA	56
4618	57	IF (ISOR.EQ.0) CALL PRESCR	SOLA	57
4619	58	CALL BC	SOLA	58
4620	59	C * * UPDATE FLUID CONFIGURATION	SOLA	59
4621	60	C	SOLA	60
4622	61	40 CALL VFCONV	SOLA	61
4623	62	IF (FLGC.GT.0.5) GO TO 110	SOLA	62
4624	63	C	SOLA	63
4625	64	C * * SET BOUNDARY CONDITIONS	SOLA	64
4626	65	C	SOLA	65
4627	66	CALL BC	SOLA	66
4628	67	C	SOLA	67
4629	68	C * * DETERMINE PRESSURE INTERPOLATION FACTOR AND NEIGHBOR	SOLA	68
4630	69	C	SOLA	69
4631	70	CALL PETACAL	SOLA	70
4632	71	CALL BC	SOLA	71
4633	72	IF (CYCLE.LT.1.AND.ISOR.NE.2) CALL PRESCR	SOLA	72
4634	73	C	SOLA	73
4635	74	C * * PLOT VELOCITY VECTORS AND FREE SURFACE ON FILM	SOLA	74
4636	75	C * * LPR=1, DRAW ONLY AT TWPLT	SOLA	75
4637	76	C * * LPR=2, DRAW AND PRINT DATA ON FILM AT TWPLT	SOLA	76
4638	77	C * * PRINT DATA ON LISTING ONLY AT TWPRT	SOLA	77
4639	78	C	SOLA	78
4640	79	IF (CYCLE.EQ.0.OR.CYCLE.EQ.1) GO TO 60	SOLA	79

```

4641 80      IF (T+1.OE-10.GE.TWFIN) GO TO 50          SOLA 80
4642 81      IF (T+1.OE-6.LT.TWPLT) GO TO 70          SOLA 81
4643 82      50 TWPLT=TWPLT+PLTDT                  SOLA 82
4644 83      60 IF (LPR.EQ.0) GO TO 70          SOLA 83
4645 84      IF (LPR.EQ.1.OR.LPR.EQ.2) CALL DRAW    SOLA 84
4646 85      IF (LPR.EQ.2) CALL LPRT                SOLA 85
4647 86 C
4648 87 C * * PRINT FIELD VARIABLE DATA ON PAPER   SOLA 86
4649 88 C
4650 89      70 CONTINUE                         SOLA 89
4651 90      IF (CYCLE.EQ.0.OR.CYCLE.EQ.1) GO TO 80  SOLA 90
4652 91      IF (T+1.OE-6.LT.TWPRT) GO TO 90          SOLA 91
4653 92      IF (TWPRT.EQ.TWPLT-PLTDT.AND.LPR.EQ.2) GO TO 90  SOLA 92
4654 93      TWPRT=TWPRT+PRTDT                  SOLA 93
4655 94      80 IF (LPR.NE.3) GO TO 90          SOLA 94
4656 95      CALL LPRT2                          SOLA 95
4657 96 C
4658 97 C * * SET THE ADVANCE TIME ARRAYS INTO THE TIME-N ARRAYS  SOLA 96
4659 98 C
4660 99      90 CONTINUE                         SOLA 99
4661 100     DO 100 K=1,KMAX                   SOLA 100
4662 101     DO 100 J=1,JMAX                   SOLA 101
4663 102     DO 100 I=1,IMAX                   SOLA 102
4664 103     CALL IJKONLY                     SOLA 103
4665 104     UN(IJK)=U(IJK)                   SOLA 104
4666 105     VN(IJK)=V(IJK)                   SOLA 105
4667 106     WN(IJK)=W(IJK)                   SOLA 106
4668 107     FN(IJK)=F(IJK)                   SOLA 107
4669 108     PN(IJK)=P(IJK)                   SOLA 108
4670 109     100 CONTINUE                     SOLA 109
4671 110 C
4672 111 C * * ADJUST DELT                 SOLA 110
4673 112 C
4674 113     110 CALL DELTADJ                 SOLA 113
4675 114 C
4676 115     IF (MSGCODE.NE.1) GO TO 120        SOLA 115
4677 116 C * * GET THE MESSAGE FROM THE CONTROLLER.  SOLA 116
4678 117 C * * THIS ALSO RESETS MSGCODE TO ZERO.  SOLA 117
4679 118     CALL GETMSGR (IBUFF,NW,1)           SOLA 118
4680 119 C * * CHECK FOR THE MESSAGE "HALT"       SOLA 119
4681 120     IF (IBUFF.EQ.4HHALT) GO TO 140      SOLA 120
4682 121     120 CONTINUE                     SOLA 121
4683 122 C
4684 123 C * * TEST IF PROBLEM IS FINISHED  SOLA 122
4685 124 C
4686 125     IF (T+1.OE-10.GE.TWFIN) GO TO 140  SOLA 125
4687 126     IF (CYCLE.LE.0) GO TO 10          SOLA 126
4688 127 C
4689 128 C * * CALCULATE GRIND TIME AND PRINT OUT PROBLEM DATA  SOLA 128
4690 129 C
4691 130     TOLD=T2                         SOLA 129
4692 131     IF (MOD(CYCLE,20).NE.0) GO TO 130  SOLA 130
4693 132     CALL SECOND (T2)                 SOLA 131
4694 133     XXX=(T2-TOLD)                   SOLA 132
4695 134     XX=XXX*RIJK                    SOLA 133
4696 135     IF (LPR.NE.0) WRITE (12,170) T,CYCLE,ITER,XX,DELT,EPSI,VCHGT  SOLA 134
4697 136     1 ,VOFTOT,XXX                  SOLA 135
4698 137     IF (LPR.NE.0) WRITE (9,170) T,CYCLE,ITER,XX,DELT,EPSI,VCHGT,VOFTOT  SOLA 136
4699 138     1 ,XXX                      SOLA 137
4700 139     130 CONTINUE                   SOLA 138
4701 140     IF (T2-STIM.GE.TLM) GO TO 140      SOLA 139
4702 141 C
4703 142 C * * WRITE PROBLEM DUMP TAPE  SOLA 140
4704 143 C
4705 144     IF (T.GE.TWTD) CALL WRTAPE        SOLA 141
4706 145     GO TO 10                      SOLA 142
4707 146     140 CALL WRTAPE                 SOLA 143
4708 147     IF (T+1.OE-10.GE.TWFIN) GO TO 150  SOLA 144
4709 148     IF (LPR.EQ.1.OR.LPR.EQ.2) CALL DRAW  SOLA 145
4710 149     IF (LPR.EQ.2) CALL LPRT                SOLA 146
4711 150     150 CONTINUE                   SOLA 147
4712 151     CALL GDONE                     SOLA 148
4713 152     RETURN                        SOLA 149
4714 153 C
4715 154 C * * READ TAPE DUMP             SOLA 150
4716 155 C
4717 156     160 CALL RDTAPE                 SOLA 151
4718 157     GO TO 10                      SOLA 152
4719 158 C
4720 159     170 FORMAT (3H T=,1PE12.5,7H CYCLE=,I4,6H ITER=,I4,8H GRINDS=,E10.3,6H SOLA 153

```

```

4721 160      1 DELT=,E11.4,6H EPSI=,E10.3,7H VCHGT=,E10.3,8H VOFTOT=,E12.5,/5X,* SOLA 160
4722 161      2TOTAL TIME FOR CYCLE = *,E11.4) SOLA 161
4723 162      180 FORMAT (/,3H T=,1PE10.3,7H CYCLE=,I5,6H ITER=,I5,6H DELT=,E10.3,6H SOLA 162
4724 163      1 EPSI=,E10.3) SOLA 163
4725 164      190 FORMAT (/,25HENTERING SUBROUTINE SOLA:,/) SOLA 164
4726 165      END SOLA 165
4727 1 *DK SURCNTR SURCNTR 1
4728 2          SUBROUTINE SURCNTR (I1,I2,J1,J2,K1,K2,MPLN,AXP,AYP) SURCNTR 2
4729 3 *CA SLCOM1 SURCNTR 3
4730 4 C SURCNTR 4
4731 5 C * * DRAW FREE SURFACE ON SELECTED PLANE VELOCITY VECTOR PLOT SURCNTR 5
4732 6 C SURCNTR 6
4733 7          FPL=0.5 SURCNTR 7
4734 8 C SURCNTR 8
4735 9 C * * SELECT INDICES AND COORDINATES FOR SURFACE PLANE TO BE PLOTTED SURCNTR 9
4736 10 C SURCNTR 10
4737 11          GO TO (10,20,30), MPLN SURCNTR 11
4738 12      10 L1=J1 SURCNTR 12
4739 13          L2=J2 SURCNTR 13
4740 14          M1=I1 SURCNTR 14
4741 15          N1=K1 SURCNTR 15
4742 16          N2=K2 SURCNTR 16
4743 17          GO TO 40 SURCNTR 17
4744 18      20 L1=I1 SURCNTR 18
4745 19          L2=I2 SURCNTR 19
4746 20          M1=J1 SURCNTR 20
4747 21          N1=K1 SURCNTR 21
4748 22          N2=K2 SURCNTR 22
4749 23          GO TO 40 SURCNTR 23
4750 24      30 L1=I1 SURCNTR 24
4751 25          L2=I2 SURCNTR 25
4752 26          M1=K1 SURCNTR 26
4753 27          N1=J1 SURCNTR 27
4754 28          N2=J2 SURCNTR 28
4755 29          40 CONTINUE SURCNTR 29
4756 30 C SURCNTR 30
4757 31 C * * MULTIPLIERS OF 1 OR 0 TO DETERMINE PLANE PLOTTED SURCNTR 31
4758 32 C SURCNTR 32
4759 33          AXPY=AXP*AYP SURCNTR 33
4760 34          AXP1=1.0-AXP SURCNTR 34
4761 35          AYP1=1.0-AYP SURCNTR 35
4762 36          M=M1 SURCNTR 36
4763 37 C SURCNTR 37
4764 38 C * * DRAW FREE SURFACE SURCNTR 38
4765 39 C SURCNTR 39
4766 40          DO 140 L=L1,L2 SURCNTR 40
4767 41          DO 140 N=N1,N2 SURCNTR 41
4768 42          XLR=1.0 SURCNTR 42
4769 43          XLL=1.0 SURCNTR 43
4770 44          GO TO (50,60,70), MPLN SURCNTR 44
4771 45      50 I=M SURCNTR 45
4772 46          J=L SURCNTR 46
4773 47          K=N SURCNTR 47
4774 48          XLCP=YJ(J+1) SURCNTR 48
4775 49          XLC=YJ(J) SURCNTR 49
4776 50          XLCM=YJ(J-1) SURCNTR 50
4777 51          YNCP=ZK(K+1) SURCNTR 51
4778 52          YNC=ZK(K) SURCNTR 52
4779 53          YNCM=ZK(K-1) SURCNTR 53
4780 54          GO TO 80 SURCNTR 54
4781 55      6C J=M SURCNTR 55
4782 56          I=L SURCNTR 56
4783 57          K=N SURCNTR 57
4784 58          XLCP=XI(I+1) SURCNTR 58
4785 59          XLC=XI(I) SURCNTR 59
4786 60          XLCM=XI(I-1) SURCNTR 60
4787 61          YNCP=ZK(K+1) SURCNTR 61
4788 62          YNC=ZK(K) SURCNTR 62
4789 63          YNCM=ZK(K-1) SURCNTR 63
4790 64          GO TO 80 SURCNTR 64
4791 65      70 K=M SURCNTR 65
4792 66          I=L SURCNTR 66
4793 67          J=N SURCNTR 67
4794 68          XLCP=XI(I+1) SURCNTR 68
4795 69          XLC=XI(I) SURCNTR 69
4796 70          XLCM=XI(I-1) SURCNTR 70
4797 71          YNCP=YJ(J+1)/RR(I) SURCNTR 71
4798 72          YNC=YJ(J)/RR(I) SURCNTR 72
4799 73          YNCM=YJ(J-1)/RR(I) SURCNTR 73
4800 74          IF (CYL.LT.0.5.OR.MPLN.NE.3) GO TO 80 SURCNTR 74

```

4801	75	YNCP=YJ(J+1)/X(IM1)	SURCNTR	75
4802	76	YNC=YJ(J)/X(IM1)	SURCNTR	76
4803	77	YNCM=YJ(J-1)/X(IM1)	SURCNTR	77
4804	78	XLR=X(I)	SURCNTR	78
4805	79	XLL=X(I-1)	SURCNTR	79
4806	80	IF (XLL.EQ.O.O) XLL=EM6	SURCNTR	80
4807	81	80 CONTINUE	SURCNTR	81
4808	82	CALL CALCIJK	SURCNTR	82
4809	83	LMN=IJK	SURCNTR	83
4810	84	LPMN=IPJK*AXP+IUPK*AXP1	SURCNTR	84
4811	85	LMNP=IJKP*AYP1+IJKP*AYP	SURCNTR	85
4812	86	LPMNP=IJPKP*AXP1+IPJKP*AXYP+IPJPK*AYP1	SURCNTR	86
4813	87	LMNM=IJMK*AYP1+IJKM*AYP	SURCNTR	87
4814	88	LMMN=IMJK*AXP+IJMK*AXP1	SURCNTR	88
4815	89	LPMNM=IJKP*AXP1+IPJKM*AXYP+IPJMK*AYP1	SURCNTR	89
4816	90	LMMNP=IJMKP*AXP1+IMJKP*AXYP+IMUPK*AYP1	SURCNTR	90
4817	91	IF (BETA(LMN).LT.O.O) GO TO 140	SURCNTR	91
4818	92	FATR=0.25*(F(LMN)+F(LPMN)+F(LMNP)+F(LPMNP))	SURCNTR	92
4819	93	FXTR=0.5*(F(LPMNP)+F(LPMN)-F(LMNP)-F(LMN))/(XLCP-XLC)	SURCNTR	93
4820	94	FYTR=0.5*(F(LMN)+F(LPMNP)-F(LMN)-F(LPMN))/(YNCP-YNC)	SURCNTR	94
4821	95	FYTR=FYTR/XLR	SURCNTR	95
4822	96	FTRS=FXTR**2+FYTR**2	SURCNTR	96
4823	97	IF (FTRS.EQ.O.O) FTRS=1.OE+10	SURCNTR	97
4824	98	XTR=0.5*(XLCP+XLC)+(FPL-FATR)*FXTR/FTRS	SURCNTR	98
4825	99	XTR=AMAX1(XTR,XLC)	SURCNTR	99
4826	100	XTR=AMIN1(XTR,XLC)	SURCNTR	100
4827	101	YTR=0.5*(YNC+YNCP)+(FPL-FATR)*FYTR/(FTRS*XLR)	SURCNTR	101
4828	102	YTR=AMAX1(YTR,YNC)	SURCNTR	102
4829	103	YTR=AMIN1(YTR,YNCP)	SURCNTR	103
4830	104	X2EF=XTR-(AXP*XBL+AXP1*YBF)	SURCNTR	104
4831	105	Y2EF=YTR-(AYP*ZBB+AYP1*YBF)	SURCNTR	105
4832	106	IF (F(LMN).GT.O.5.AND.F(LPMN).GT.O.5) GO TO 110	SURCNTR	106
4833	107	IF (F(LMN).LT.O.5.AND.F(LPMN).LT.O.5) GO TO 110	SURCNTR	107
4834	108	FABR=0.25*(F(LMN)+F(LPMN)+F(LMNM)+F(LPMNM))	SURCNTR	108
4835	109	FXBR=0.5*(F(LPMN)+F(LPMNM)-F(LMN)-F(LMNM))/(XLCP-XLC)	SURCNTR	109
4836	110	FYBR=0.5*(F(LMN)+F(LPMN)-F(LMNM)-F(LPMNM))/(YNC-YNCM)	SURCNTR	110
4837	111	FYBR=FYBR/XLR	SURCNTR	111
4838	112	FBRS=FXBR**2+FYBR**2	SURCNTR	112
4839	113	IF (FBRS.EQ.O.O) FBRS=1.OE+10	SURCNTR	113
4840	114	XBMR=0.5*(XLCP+XLC)+(FPL-FABR)*FXBR/FBRS	SURCNTR	114
4841	115	XBMR=AMAX1(XBMR,XLC)	SURCNTR	115
4842	116	XBMR=AMIN1(XBMR,XLC)	SURCNTR	116
4843	117	YBMR=0.5*(YNC+YNCM)+(FPL-FABR)*FYBR/(FBRS*XLR)	SURCNTR	117
4844	118	YBMR=AMAX1(YBMR,YNCM)	SURCNTR	118
4845	119	YBMR=AMIN1(YBMR,YNC)	SURCNTR	119
4846	120	X1EF=XBMR-(AXP*XBL+AXP1*YBF)	SURCNTR	120
4847	121	Y1EF=YBMR-(AYP*ZBB+AYP1*YBF)	SURCNTR	121
4848	122	GO TO (100,100,90), MPLN	SURCNTR	122
4849	123	90 IF (CYL.LT.O.5) GO TO 100	SURCNTR	123
4850	124	X2EF=XTR*COS(YTR)-XBLC	SURCNTR	124
4851	125	Y2EF=XTR*SIN(YTR)-YBFC	SURCNTR	125
4852	126	X1EF=XBMR*COS(YBMR)-XBLC	SURCNTR	126
4853	127	Y1EF=XBMR*SIN(YBMR)-YBFC	SURCNTR	127
4854	128	100 CONTINUE	SURCNTR	128
4855	129	IX1=FIXL(MPLN)+X1EF*XCONV(MPLN)	SURCNTR	129
4856	130	IY1=FIYB+Y1EF*YCONV(MPLN)	SURCNTR	130
4857	131	IX2=FIXL(MPLN)+X2EF*XCONV(MPLN)	SURCNTR	131
4858	132	IY2=FIYB+Y2EF*YCONV(MPLN)	SURCNTR	132
4859	133	CALL DRV (IX1,IY1,IX2,IY2)	SURCNTR	133
4860	134	110 CONTINUE	SURCNTR	134
4861	135	IF (F(LMN).GT.O.5.AND.F(LMNP).GT.O.5) GO TO 140	SURCNTR	135
4862	136	IF (F(LMN).LT.O.5.AND.F(LMNP).LT.O.5) GO TO 140	SURCNTR	136
4863	137	FATL=0.25*(F(LMN)+F(LMNP)+F(LMMN)+F(LMMNP))	SURCNTR	137
4864	138	FXTL=0.5*(F(LMNP)+F(LMN)-F(LMMNP)-F(LMMN))/(XLC-XLCM)	SURCNTR	138
4865	139	FYTL=0.5*(F(LMMNP)+F(LMN)-F(LMMN)-F(LMN))/(YNC-YNC)	SURCNTR	139
4866	140	FYTL=FYTL/XLL	SURCNTR	140
4867	141	FTLS=FXTL**2+FYTL**2	SURCNTR	141
4868	142	IF (FTLS.EQ.O.O) FTLS=1.OE+10	SURCNTR	142
4869	143	XTL=0.5*(XLCM+XLC)+(FPL-FATL)*FXTL/FTLS	SURCNTR	143
4870	144	XTL=AMAX1(XTL,XLCM)	SURCNTR	144
4871	145	XTL=AMIN1(XTL,XLC)	SURCNTR	145
4872	146	YTL=0.5*(YNC+YNCP)+(FPL-FATL)*FYTL/(FTLS*XLL)	SURCNTR	146
4873	147	YTL=AMAX1(YTL,YNC)	SURCNTR	147
4874	148	YTL=AMIN1(YTL,YNCP)	SURCNTR	148
4875	149	X1EF=XTL-(AXP*XBL+AXP1*YBF)	SURCNTR	149
4876	150	Y1EF=YTL-(AYP*ZBB+AYP1*YBF)	SURCNTR	150
4877	151	GO TO (130,130,120), MPLN	SURCNTR	151
4878	152	120 IF (CYL.LT.O.5) GO TO 130	SURCNTR	152
4879	153	X2EF=XTR*COS(YTR)-XBLC	SURCNTR	153
4880	154	Y2EF=XTR*SIN(YTR)-YBFC	SURCNTR	154

```

4881 155      X1EF=XTL*COS(YTL)-XBLC          SURCNTR 155
4882 156      Y1EF=XTL*SIN(YTL)-YBFC          SURCNTR 156
4883 157      130 CONTINUE                     SURCNTR 157
4884 158      IX1=FIXL(MPLN)+X1EF*XCONV(MPLN)  SURCNTR 158
4885 159      IY1=FIYB+Y1EF*YCONV(MPLN)        SURCNTR 159
4886 160      IX2=FIXL(MPLN)+X2EF*XCONV(MPLN)  SURCNTR 160
4887 161      IY2=FIYB+Y2EF*YCONV(MPLN)        SURCNTR 161
4888 162      CALL DRV (IX1,IY1,IX2,IY2)       SURCNTR 162
4889 163      140 CONTINUE                     SURCNTR 163
4890 164      C                               SURCNTR 164
4891 165      RETURN                         SURCNTR 165
4892 166      END                            SURCNTR 166
4893 1 *DK SURFPLT                         SURFPLT 1
4894 2         SUBROUTINE SURFPLT (I1,I2,J1,J2,K1,K2,NA,IPER) SURFPLT 2
4895 3 *CA SLCOM1                          SURFPLT 3
4896 4 C                               SURFPLT 4
4897 5 C * * DRAW FREE SURFACE IN PERSPECTIVE SURFPLT 5
4898 6 C                               SURFPLT 6
4899 7         CALL FADV (NA)                SURFPLT 7
4900 8         CALL DRFP                 SURFPLT 8
4901 9         CALL LINCNT (59)              SURFPLT 9
4902 10        WRITE (12,70)               SURFPLT 10
4903 11        WRITE (12,80) NAME,JNM,DAT,CLK  SURFPLT 11
4904 12        WRITE (12,90) ITER,T,CYCLE    SURFPLT 12
4905 13        FPL=0.5                  SURFPLT 13
4906 14        I1M=I1-1                  SURFPLT 14
4907 15        J1M=J1-1                  SURFPLT 15
4908 16        K1M=K1-1                  SURFPLT 16
4909 17        DO 60 I=I1M,I2            SURFPLT 17
4910 18        DO 60 J=J1M,J2            SURFPLT 18
4911 19        DO 60 K=K1M,K2            SURFPLT 19
4912 20        CALL CALCIJK             SURFPLT 20
4913 21        FAIM=0.0                 SURFPLT 21
4914 22        FAJM=0.0                 SURFPLT 22
4915 23        FAKM=0.0                 SURFPLT 23
4916 24        CRX=CYL/X(IM1)           SURFPLT 24
4917 25        IF (BETA(IJK).LT.O.O.AND.BETA(IJPK).LT.O.O.AND.BETA(IJPKP).LT.O.O SURFPLT 25
4918 26        .AND.BETA(IJKP).LT.O.O) GO TO 10 SURFPLT 26
4919 27        IF (F(IJK).GT.FPL.AND.F(IJPK).GT.FPL.AND.F(IJPKP).GT.FPL.AND.F SURFPLT 27
4920 28        1 (IJKP).GT.FPL) GO TO 10 SURFPLT 28
4921 29        IF (F(IJK).LT.FPL.AND.F(IJPK).LT.FPL.AND.F(IJPKP).LT.FPL.AND.F SURFPLT 29
4922 30        1 (IJKP).LT.FPL) GO TO 10 SURFPLT 30
4923 31        FAIM=0.125*(F(IJK)+F(IJPK)+F(IJPKP)+F(IJKP)+F(IMJK)+F(IMUPK)+F SURFPLT 31
4924 32        1 (IMJPKP)+F(IMJKP))          SURFPLT 32
4925 33        FAIX=0.5*(F(IJK)+F(IJPK)+F(IJPKP)+F(IJKP)-4.0*FAIM)/(XI(I)-XI(I-1) SURFPLT 33
4926 34        1 )                           SURFPLT 34
4927 35        FAIY=0.5*(F(IMJK)+F(IJK)+F(IJKP)+F(IMJKP)-4.0*FAIM)*RR(I-1)/(YJ(J) SURFPLT 35
4928 36        1 -YJ(J+1))                 SURFPLT 36
4929 37        FAIZ=0.5*(F(IJK)+F(IJPK)+F(IMJPKP)+F(IMJK)-4.0*FAIM)/(ZK(K)-ZK(K+1) SURFPLT 37
4930 38        1 )                           SURFPLT 38
4931 39        FAIS=FAIX**2+FAIY**2+FAIZ**2 SURFPLT 39
4932 40        IF (FAIS.EQ.O.O) FAIS=1.0E+10 SURFPLT 40
4933 41        XAIM=0.5*(XI(I-1)+XI(I))+(FPL-FAIM)*FAIX/FAIS          SURFPLT 41
4934 42        XAIM=AMAX1(XAIM,XI(I-1))          SURFPLT 42
4935 43        XAIM=AMIN1(XAIM,XI(I))           SURFPLT 43
4936 44        YAIM=0.5*(YJ(J)+YJ(J+1))+(FPL-FAIM)*FAIY*RR(I-1)/FAIS          SURFPLT 44
4937 45        YAIM=AMAX1(YAIM,YJ(J))           SURFPLT 45
4938 46        YAIM=AMIN1(YAIM,YJ(J+1))          SURFPLT 46
4939 47        TH1=CYL*RX(IM1)*YAIM          SURFPLT 47
4940 48        YAIM=YAIM*(1.0-CYL)+XAIM*SIN(TH1)          SURFPLT 48
4941 49        XAIM=XAIM*COS(TH1)           SURFPLT 49
4942 50        ZAIM=0.5*(ZK(K)+ZK(K+1))+(FPL-FAIM)*FAIZ/FAIS          SURFPLT 50
4943 51        ZAIM=AMAX1(ZAIM,ZK(K))           SURFPLT 51
4944 52        ZAIM=AMIN1(ZAIM,ZK(K+1))          SURFPLT 52
4945 53        TH1=CRX*YAIM                 SURFPLT 53
4946 54        YAIM=XAIM*SIN(TH1)+YAIM*(1.0-CYL)          SURFPLT 54
4947 55        XAIM=XAIM*COS(TH1)           SURFPLT 55
4948 56        10 CONTINUE                     SURFPLT 56
4949 57        IF (BETA(IJK).LT.O.O.AND.BETA(IPJK).LT.O.O.AND.BETA(IPJKP).LT.O.O SURFPLT 57
4950 58        .AND.BETA(IJKP).LT.O.O) GO TO 20 SURFPLT 58
4951 59        IF (F(IJK).GT.FPL.AND.F(IPJK).GT.FPL.AND.F(IPJKP).GT.FPL.AND.F SURFPLT 59
4952 60        1 (IJKP).GT.FPL) GO TO 20 SURFPLT 60
4953 61        IF (F(IJK).LT.FPL.AND.F(IPJK).LT.FPL.AND.F(IPJKP).LT.FPL.AND.F SURFPLT 61
4954 62        1 (IJKP).LT.FPL) GO TO 20 SURFPLT 62
4955 63        FAJM=0.125*(F(IJK)+F(IPJK)+F(IPJKP)+F(IJKP)+F(IUMK)+F(IPUMK)+F SURFPLT 63
4956 64        1 (IPUMKP)+F(IUMKP))          SURFPLT 64
4957 65        FAJX=0.5*(F(IJK)+F(IJPK)+F(IJPKP)+F(IJKP)-4.0*FAJM)/(XI(I)-XI(I+1) SURFPLT 65
4958 66        1 )                           SURFPLT 66
4959 67        FAJY=0.5*(F(IJK)+F(IPJK)+F(IPJKP)+F(IJKP)-4.0*FAJM)*RR(I)/(YJ(J) SURFPLT 67
4960 68        1 -YJ(J-1))                 SURFPLT 68

```

```

4961 69 FAJZ=O.5*(F(IJK)+F(IJMK)+F(IPJMK)+F(IPJK)-4.O*FAJM)/(ZK(K)-ZK(K+1)) SURFPLT 69
4962 70 1 ) SURFPLT 70
4963 71 FAJS=FAJX**2+FAJY**2+FAJZ**2 SURFPLT 71
4964 72 IF (FAJS.EQ.O.O) FAJS=1.OE+10 SURFPLT 72
4965 73 XAJM=O.5*(XI(I)+XI(I+1))+(FPL-FAJM)*FAJX/FAJS SURFPLT 73
4966 74 XAJM=AMAX1(XAJM,XI(I)) SURFPLT 74
4967 75 XAJM=AMIN1(XAJM,XI(I+1)) SURFPLT 75
4968 76 YAJM=O.5*(YJ(J-1)+YJ(J))+(FPL-FAJM)*FAJY*RR(I)/FAJS SURFPLT 76
4969 77 YAJM=AMAX1(YAJM,YJ(J-1)) SURFPLT 77
4970 78 YAJM=AMIN1(YAJM,YJ(J)) SURFPLT 78
4971 79 TH1=CYL*RX(IM1)*YAJM SURFPLT 79
4972 80 YAJM=YAJM*(1.O-CYL)+XAJM*SIN(TH1) SURFPLT 80
4973 81 XAJM=XAJM*COS(TH1) SURFPLT 81
4974 82 ZAJM=O.5*(ZK(K)+ZK(K+1))+(FPL-FAJM)*FAJZ/FAJS SURFPLT 82
4975 83 ZAJM=AMAX1(ZAJM,ZK(K)) SURFPLT 83
4976 84 ZAJM=AMIN1(ZAJM,ZK(K+1)) SURFPLT 84
4977 85 TH1=CRX*YAJM SURFPLT 85
4978 86 YAJM=XAJM*SIN(TH1)+YAJM*(1.O-CYL) SURFPLT 86
4979 87 XAJM=XAJM*COS(TH1) SURFPLT 87
4980 88 20 CONTINUE SURFPLT 88
4981 89 IF (BETA(IJK).LT.O.O.AND.BETA(IPJK).LT.O.O.AND.BETA(IPJPK).LT.O.O SURFPLT 89
4982 90 1 .AND.BETA(IJPK).LT.O.O) GO TO 30 SURFPLT 90
4983 91 IF (F(IJK).GT.FPL.AND.F(IPJK).GT.FPL.AND.F(IPJPK).GT.FPL.AND.F SURFPLT 91
4984 92 1 (IJPK).GT.FPL) GO TO 30 SURFPLT 92
4985 93 IF (F(IJK).LT.FPL.AND.F(IPJK).LT.FPL.AND.F(IPJPK).LT.FPL.AND.F SURFPLT 93
4986 94 1 (IPJPK).LT.FPL) GO TO 30 SURFPLT 94
4987 95 FAKM=O.125*(F(IJK)+F(IPJK)+F(IPJPK)+F(IJKM)+F(IPJMK)+F SURFPLT 95
4988 96 1 (IPJPKM)+F(IJPKM)) SURFPLT 96
4989 97 FAKX=O.5*(F(IJK)+F(IJPK)+F(IJPKM)+F(IJKM)-4.O*FAKM)/(XI(I)-XI(I+1)) SURFPLT 97
4990 98 1 ) SURFPLT 98
4991 99 FAKY=O.5*(F(IJK)+F(IPJK)+F(IPJMK)+F(IJKM)-4.O*FAKM)*RR(I)/(YJ(J) SURFPLT 99
4992 100 1 -YJ(J+1)) SURFPLT 100
4993 101 FAKZ=O.5*(F(IJK)+F(IPJK)+F(IJPK)-4.O*FAKM)/(ZK(K)-ZK(K-1)) SURFPLT 101
4994 102 1 ) SURFPLT 102
4995 103 FAKS=FAKX**2+FAKY**2+FAKZ**2 SURFPLT 103
4996 104 IF (FAKS.EQ.O.O) FAKS=1.OE+10 SURFPLT 104
4997 105 XAKM=O.5*(XI(I)+XI(I+1))+(FPL-FAKM)*FAKX/FAKS SURFPLT 105
4998 106 XAKM=AMAX1(XAKM,XI(I)) SURFPLT 106
4999 107 XAKM=AMIN1(XAKM,XI(I+1)) SURFPLT 107
5000 108 YAKM=O.5*(YJ(J)+YJ(J+1))+(FPL-FAKM)*FAKY*RR(I)/FAKS SURFPLT 108
5001 109 YAKM=AMAX1(YAKM,YJ(J)) SURFPLT 109
5002 110 YAKM=AMIN1(YAKM,YJ(J+1)) SURFPLT 110
5003 111 TH1=CYL*RX(IM1)*YAKM SURFPLT 111
5004 112 YAKM=YAKM*(1.O-CYL)+XAKM*SIN(TH1) SURFPLT 112
5005 113 XAKM=XAKM*COS(TH1) SURFPLT 113
5006 114 ZAKM=O.5*(ZK(K-1)+ZK(K))+(FPL-FAKM)*FAKZ/FAKS SURFPLT 114
5007 115 ZAKM=AMAX1(ZAKM,ZK(K-1)) SURFPLT 115
5008 116 ZAKM=AMIN1(ZAKM,ZK(K)) SURFPLT 116
5009 117 TH1=CRX*YAKM SURFPLT 117
5010 118 YAKM=XAKM*SIN(TH1)+YAKM*(1.O-CYL) SURFPLT 118
5011 119 XAKM=XAKM*COS(TH1) SURFPLT 119
5012 120 30 CONTINUE SURFPLT 120
5013 121 IF (FAIM.EQ.O.O.AND.FAJM.EQ.O.O.AND.FAKM.EQ.O.O) GO TO 60 SURFPLT 121
5014 122 FAC=O.125*(F(IJK)+F(IPJK)+F(IPJPK)+F(IJPK)+F(IPJPKP)+F SURFPLT 122
5015 123 1 (IPJPKP)+F(IJPKP)) SURFPLT 123
5016 124 FACX=O.5*(F(IJK)+F(IJPK)+F(IJPKP)+F(IJKP)-4.O*FAC)/(XI(I)-XI(I+1)) SURFPLT 124
5017 125 FACY=O.5*(F(IJK)+F(IPJK)+F(IPJPK)+F(IJPK)-4.O*FAC)*RR(I)/(YJ(J)-YJ SURFPLT 125
5018 126 1 (J+1)) SURFPLT 126
5019 127 FACZ=O.5*(F(IJK)+F(IPJK)+F(IPJPK)+F(IJKP)-4.O*FAC)/(ZK(K)-ZK(K+1)) SURFPLT 127
5020 128 FACS=FACX**2+FACY**2+FAcz**2 SURFPLT 128
5021 129 IF (FACS.EQ.O.O) FACS=1.OE+10 SURFPLT 129
5022 130 XAC=O.5*(XI(I)+XI(I+1))+(FPL-FAC)*FACX/FACS SURFPLT 130
5023 131 XAC=AMAX1(XAC,XI(I)) SURFPLT 131
5024 132 XAC=AMIN1(XAC,XI(I+1)) SURFPLT 132
5025 133 YAC=O.5*(YJ(J)+YJ(J+1))+(FPL-FAC)*FACY*RR(I)/FACS SURFPLT 133
5026 134 YAC=AMAX1(YAC,YJ(J)) SURFPLT 134
5027 135 YAC=AMIN1(YAC,YJ(J+1)) SURFPLT 135
5028 136 TH1=CYL*RX(IM1)*YAC SURFPLT 136
5029 137 YAC=YAC*(1.O-CYL)+XAC*SIN(TH1) SURFPLT 137
5030 138 XAC=XAC*COS(TH1) SURFPLT 138
5031 139 ZAC=O.5*(ZK(K)+ZK(K+1))+(FPL-FAC)*FACZ/FACS SURFPLT 139
5032 140 ZAC=AMAX1(ZAC,ZK(K)) SURFPLT 140
5033 141 ZAC=AMIN1(ZAC,ZK(K+1)) SURFPLT 141
5034 142 TH1=CRX*YAC SURFPLT 142
5035 143 YAC=XAC*SIN(TH1)+YAC*(1.O-CYL) SURFPLT 143
5036 144 XAC=XAC*COS(TH1) SURFPLT 144
5037 145 CALL PCNV (IXIC,IETAC,XAC,YAC,ZAC) SURFPLT 145
5038 146 IF (FAIM.EQ.O.O.OR.I.EQ.1M) GO TO 40 SURFPLT 146
5039 147 CALL PCNV (IXIIM,IETAIM,XAIM,YAIM,ZAIM) SURFPLT 147
5040 148 CALL DRVEC (IXIC,IETAC,IXIIM,IETAIM) SURFPLT 148

```

```

5041 149    40 IF (FAJMJ.EQ.O.O.DR.J.EQ.J1M) GO TO 50          SURFPLT 149
5042 150    CALL PCNV (IXIJM,IETAJM,XAJM,YAJM,ZAJM)        SURFPLT 150
5043 151    CALL DRVEC (IXIC,IETAC,IXIJM,IETAJM)        SURFPLT 151
5044 152    50 IF (FAKM.EQ.O.O.OR.K.EQ.K1M) GO TO 60        SURFPLT 152
5045 153    CALL PCNV (IXIKM,IETAKM,XAKM,YAKM,ZAKM)        SURFPLT 153
5046 154    CALL DRVEC (IXIC,IETAC,IXIKM,IETAKM)        SURFPLT 154
5047 155    60 CONTINUE                                     SURFPLT 155
5048 156    RETURN                                       SURFPLT 156
5049 157 C
5050 158    70 FORMAT (2H PERSPECTIVE SURFACE )           SURFPLT 157
5051 159    80 FORMAT (1H ,2X,10A8,1X,A8,2(1X,A8))        SURFPLT 158
5052 160    90 FORMAT (6X,5HITER=,I5,18X,6HTIME= ,1PE12.5,12X,7HCYCLE= ,I4) SURFPLT 159
5053 161    END                                         SURFPLT 160
5054 1 *DK SURF1ON                                     SURFPLT 161
5055 2      SUBROUTINE SURF1ON                         SURF1ON  1
5056 3 C
5057 4 *CA SLCOM1                                     SURF1ON  2
5058 5      DIMENSION HX(3,3), HY(3,3), HZ(3,3)          SURF1ON  3
5059 6      DIMENSION ANE(3,3), ANW(3,3), AEN(3,3), AES(3,3), CEE(3,3), CEW(3, SURF1ON  4
5060 7      1 3), CNN(3,3), CNS(3,3)                   SURF1ON  5
5061 8      DATA NOWALL /1/                          SURF1ON  6
5062 9      DATA SANG, CSANG /0.087155742,0.996194698/ SURF1ON  7
5063 10     PSLIM=25.0/AMIN1(X(IM1),Y(JM1),Z(KM1))    SURF1ON  8
5064 11     DO 1070 K=2,KM1                           SURF1ON  9
5065 12     DO 1070 J=2,JM1                           SURF1ON 10
5066 13     DO 1070 I=2,IM1                           SURF1ON 11
5067 14     CALL CALCijk                         SURF1ON 12
5068 15     IF (BETA(IJK).LT.0.0) GO TO 1070         SURF1ON 13
5069 16     NFC=NF(IJK)                            SURF1ON 14
5070 17     IF (NFC.EQ.O.OR.NFC.GT.6) GO TO 1070       SURF1ON 15
5071 18 C
5072 19 C      CHECK IF SUFFICIENT FLUID PRESENT      SURF1ON 16
5073 20 C
5074 21     FSUM=O.O                           SURF1ON 17
5075 22     DO 10 N=K-1,K+1                      SURF1ON 18
5076 23     DO 10 M=J-1,J+1                      SURF1ON 19
5077 24     DO 10 L=I-1,I+1                      SURF1ON 20
5078 25     LMN=NO*(IMAX*JMAX*(N-1)+IMAX*(M-1)+(L-1))+1 SURF1ON 21
5079 26     FSUM=FSUM+F(LMN)                     SURF1ON 22
5080 27 10 CONTINUE                               SURF1ON 23
5081 28     IF (FSUM.LT.1.0) GO TO 1070           SURF1ON 24
5082 29 C
5083 30 C      VARIABLES FOR USE IN SIX AND SEVEN POINT DERIVATIVE FORMULAS SURF1ON 25
5084 31 C
5085 32 C      Z COORDINATE PATTERN                 SURF1ON 26
5086 33     DZN=0.5*DELZ(K+1)                      SURF1ON 27
5087 34     DZ=0.5*DELZ(K)                        SURF1ON 28
5088 35     DZS=0.5*DELZ(K-1)                      SURF1ON 29
5089 36     DZP=DZN+DZ                         SURF1ON 30
5090 37     DZM=DZS+DZ                         SURF1ON 31
5091 38     ZP=DZP/DZM                         SURF1ON 32
5092 39     ZM=1.0/ZP                         SURF1ON 33
5093 40     RDZS=1.0/(DZP+DZM)                    SURF1ON 34
5094 41     DZD=DZP-DZM                         SURF1ON 35
5095 42     RPDZ=1.0/(DZP*DZM)                    SURF1ON 36
5096 43     FRZP=DZ/DZP                         SURF1ON 37
5097 44     FRZM=DZ/DZM                         SURF1ON 38
5098 45     RATZP=DZP*RDZS                      SURF1ON 39
5099 46     RATZM=DZM*RDZS                      SURF1ON 40
5100 47     DZPA=DZP+DZ                         SURF1ON 41
5101 48     DZMA=DZM+DZ                         SURF1ON 42
5102 49     DZDN=DZN-DZ                         SURF1ON 43
5103 50     DZDS=DZS-DZ                         SURF1ON 44
5104 51 C
5105 52 C      MIRROR PATTERN FOR THETA COORDINATE (Y) SURF1ON 45
5106 53 C
5107 54     DTHE=0.5*DELY(J+1)*RX(IM1)          SURF1ON 46
5108 55     DTH=0.5*DELY(J)*RX(IM1)            SURF1ON 47
5109 56     DTHW=0.5*DELY(J-1)*RX(IM1)          SURF1ON 48
5110 57     DTHP=DTHE+DTH                         SURF1ON 49
5111 58     DTHM=DTH+DTHW                       SURF1ON 50
5112 59     THP=DTHP/DTHM                      SURF1ON 51
5113 60     THM=1.0/THP                        SURF1ON 52
5114 61     RDTHS=1.0/(DTHP+DTHM)                SURF1ON 53
5115 62     DTHD=DTHP-DTHM                      SURF1ON 54
5116 63     RPDTH=1.0/(DTHP*DTHM)                SURF1ON 55
5117 64     FRTHP=DTH/DTHP                      SURF1ON 56
5118 65     FRTHM=DTH/DTHM                      SURF1ON 57
5119 66     RATTHP=DTHP*RDTHS                  SURF1ON 58
5120 67     RATTHM=DTHM*RDTHS                  SURF1ON 59

```

5121	68	DTHPA=DTHP+DTH	SURF 1ON	68
5122	69	DTHMA=DTHM+DTH	SURF 1ON	69
5123	70	DTHDE=DTHE-DTH	SURF 1ON	70
5124	71	DTHDW=DTHW-DTH	SURF 1ON	71
5125	72 C		SURF 1ON	72
5126	73 C	MIRROR PATTERN FOR RADIAL COORDINATE	SURF 1ON	73
5127	74 C		SURF 1ON	74
5128	75	DRD=0.5*DELX(I+1)	SURF 1ON	75
5129	76	DR=0.5*DELX(I)	SURF 1ON	76
5130	77	DRI=0.5*DELX(I-1)	SURF 1ON	77
5131	78	IF (I.NE.2) GO TO 20	SURF 1ON	78
5132	79	DRO=0.5*DELX(4)	SURF 1ON	79
5133	80	DR=0.5*DELX(3)	SURF 1ON	80
5134	81	DRI=0.5*DELX(2)	SURF 1ON	81
5135	82	20 CONTINUE	SURF 1ON	82
5136	83	DRP=DRD+DR	SURF 1ON	83
5137	84	DRM=DR+DRI	SURF 1ON	84
5138	85	RP=DRP/DRM	SURF 1ON	85
5139	86	RM=1.0/RP	SURF 1ON	86
5140	87	RDRS=1.0/(DRP+DRM)	SURF 1ON	87
5141	88	DRD=DRP-DRM	SURF 1ON	88
5142	89	RPDR=1.0/(DRP*DRM)	SURF 1ON	89
5143	90	FRRP=DR/DRP	SURF 1ON	90
5144	91	FRRM=DR/DRM	SURF 1ON	91
5145	92	RATRP=DRP*RDRS	SURF 1ON	92
5146	93	RATRM=DRM*RDRS	SURF 1ON	93
5147	94	DRPA=DRP+DR	SURF 1ON	94
5148	95	DRMA=DRM+DR	SURF 1ON	95
5149	96	DRDO=DRO-DR	SURF 1ON	96
5150	97	DRDI=DRI-DR	SURF 1ON	97
5151	98 C		SURF 1ON	98
5152	99	DXR=0.5*(DELX(I)+DELX(I+1))	SURF 1ON	99
5153	100	DXL=0.5*(DELX(I)+DELX(I-1))	SURF 1ON	100
5154	101	IF (I.EQ.2) DXL=0.5*(DELX(2)+DELX(3))	SURF 1ON	101
5155	102	IF (I.EQ.2) DRX=0.5*(DELX(3)+DELX(4))	SURF 1ON	102
5156	103	DZT=0.5*(DELZ(K)+DELZ(K+1))	SURF 1ON	103
5157	104	DZB=0.5*(DELZ(K)+DELZ(K-1))	SURF 1ON	104
5158	105	DYBK=0.5*(DELY(J)+DELY(J+1))/RRI(I)	SURF 1ON	105
5159	106	DYBKP=0.5*(DELY(J)+DELY(J+1))/RRI(I+1)	SURF 1ON	106
5160	107	DYBKM=0.5*(DELY(J)+DELY(J+1))/RRI(I-1)	SURF 1ON	107
5161	108	IF (CYL.EQ.1.0.AND.I.EQ.2) DYBKM=DYBK	SURF 1ON	108
5162	109	DYF=0.5*(DELY(J)+DELY(J-1))/RRI(I)	SURF 1ON	109
5163	110	DYFP=0.5*(DELY(J)+DELY(J-1))/RRI(I+1)	SURF 1ON	110
5164	111	DYFM=0.5*(DELY(J)+DELY(J-1))/RRI(I-1)	SURF 1ON	111
5165	112	IF (CYL.EQ.1.0.AND.I.EQ.2) DYFM=DYF	SURF 1ON	112
5166	113	RXDEN=1.0/(DXL*DXR*(DXL+DXR))	SURF 1ON	113
5167	114	RYDEN=1.0/(DYF*DYBK*(DYF+DYBK))	SURF 1ON	114
5168	115	RYDENP=1.0/(DYFP*DYBKP*(DYFP+DYBKP))	SURF 1ON	115
5169	116	RYDENM=1.0/(DYFM*DYBKM*(DYFM+DYBKM))	SURF 1ON	116
5170	117	RZDEN=1.0/(DZT*DZB*(DZT+DZB))	SURF 1ON	117
5171	118 C		SURF 1ON	118
5172	119 C	ASSUME NEAR CYLINDRICAL SURFACE - COLLAPSE IN RADIAL DIRECTION	SURF 1ON	119
5173	120 C		SURF 1ON	120
5174	121 C	ISCR, ISCV, ISCH ARE FLAGS SET DURING COLLAPSING DOWN LOOPS	SURF 1ON	121
5175	122 C	IF EQUAL TO 0 THEN SUBSEQUENT SLOPES MAY BE ACCURATE	SURF 1ON	122
5176	123 C	IF EQUAL TO 1 THEN SLOPES CANNOT BE CORRECT : AT LEAST ONE ROW	SURF 1ON	123
5177	124 C	OR COLUMN OF THE COLLAPSING ARRAYS DOES NOT CONTAIN A	SURF 1ON	124
5178	125 C	SURFACE CELL	SURF 1ON	125
5179	126 C		SURF 1ON	126
5180	127	ISCR=0	SURF 1ON	127
5181	128	ISCV=0	SURF 1ON	128
5182	129	ISCH=0	SURF 1ON	129
5183	130 C		SURF 1ON	130
5184	131	ISCRE=0	SURF 1ON	131
5185	132	ISCRF=0	SURF 1ON	132
5186	133	DO 150 KK=1,3	SURF 1ON	133
5187	134	N=K-2+KK	SURF 1ON	134
5188	135	DO 140 JU=1,3	SURF 1ON	135
5189	136	M=J-2+JJ	SURF 1ON	136
5190	137	LMN=NQ*(IMAX*JMAX*(N-1)+IMAX*(M-1)+(I-1))+1	SURF 1ON	137
5191	138	LMNM=LMN-NQ	SURF 1ON	138
5192	139	LMNP=LMN+NQ	SURF 1ON	139
5193	140	IF (CYL.LT.0.5) GO TO 70	SURF 1ON	140
5194	141	IF (M.EQ.1) GO TO 30	SURF 1ON	141
5195	142	IF (M.EQ.JMAX) GO TO 50	SURF 1ON	142
5196	143	GO TO 70	SURF 1ON	143
5197	144 C		SURF 1ON	144
5198	145 C	M=1	SURF 1ON	145
5199	146 C		SURF 1ON	146
5200	147	30 MM=2	SURF 1ON	147

```

5201 148      IF (WF.EQ.4) MM=JM1
5202 149      40 LMN=NQ*(IMAX*JMAX*(N-1)+IMAX*(MM-1)+(I-1))+1
5203 150      GO TO 60
5204 151 C
5205 152 C
5206 153 C
5207 154      M=JMAX
5208 155      IF (WBK.EQ.4) MM=2
5209 156      GO TO 40
5210 157      60 LMNM=LMN-NQ
5211 158      LMNP=LMN+NQ
5212 159      70 CONTINUE
5213 160      FOFM=1.0
5214 161      IF (AC(LMNM).LE.EM6.AND.F(IMJK).LT.EMF) FOFM=0.0
5215 162      FOFP=1.0
5216 163      IF (AC(LMNP).LE.EM6.AND.F(IPJK).LT.EMF) FOFP=0.0
5217 164      IF (I.GE.3.AND.I.LE.IM2) GO TO 100
5218 165      IF (I.EQ.2) GO TO 80
5219 166      FOFMM=1.0
5220 167      LMNMM=LMNM-NQ
5221 168      IF (AC(LMNMM).LT.EM6.AND.F(IMJK).LT.EMF) FOFMM=0.0
5222 169      DELTA=2.0*((1.0+AC(LMNM)*(F(LMNM)-1.0))*XI(I-1)*DELX(I-1)*FOFM+(1.
5223 170      1.0+AC(LMN)*(F(LMN)-1.0))*XI(I)*DELX(I)+(1.0+AC(LMNMM)*(F(LMNMM)-1.
5224 171      2.0))*XI(I-2)*DELX(I-2)*FOFMM)
5225 172      DELTB=2.0*(XI(I-2)*DELX(I-2)*FOFMM+XI(I-1)*DELX(I-1)*FOFM+XI(I)
5226 173      1.*DELX(I))
5227 174      IF (NFC.NE.2) HX(KK,JJ)=SQRT(X(I-3)**2+DELTA)-X(I-3)
5228 175      IF (NFC.EQ.2) HX(KK,JJ)=X(I)-SQRT(X(I)**2-DELTA)
5229 176      IF (NFC.NE.2) HXCOL=SQRT(X(I-3)**2+DELTB)-X(I-3)-EM6
5230 177      IF (NFC.EQ.2) HXCOL=X(I)-SQRT(X(I)**2-DELTB)-EM6
5231 178      IF (HX(KK,JJ).LE.0.0) ISCRE=ISCRE+1
5232 179      IF (HX(KK,JJ).GE.HXCOL) ISCRF=ISCRF+1
5233 180      GO TO 90
5234 181      80 FOFPP=1.0
5235 182      LMNPP=LMNP+NQ
5236 183      IF (AC(LMNPP).LT.EM6.AND.F(IPJK).LT.EMF) FOFPP=0.0
5237 184      DELTA=2.0*((1.0+AC(LMN)*(F(LMN)-1.0))*XI(I)*DELX(I)+(1.0+AC(LMNP)*
5238 185      1.(F(LMNP)-1.0))*XI(I+1)*DELX(I+1)*FOFP+(1.0+AC(LMNPP)*(F(LMNPP)-1.
5239 186      2.0))*XI(I+2)*DELX(I+2)*FOFPP)
5240 187      DELTB=2.0*(XI(I+2)*DELX(I+2)*FOFPP+XI(I+1)*DELX(I+1)*FOFP+XI(I)
5241 188      1.*DELX(I))
5242 189      IF (NFC.NE.2) HX(KK,JJ)=SQRT(DELTA)
5243 190      IF (NFC.EQ.2) HX(KK,JJ)=X(I+2)-SQRT(X(I+2)**2-DELTA)
5244 191      IF (NFC.NE.2) HXCOL=SQRT(DELTB)-EM6
5245 192      IF (NFC.EQ.2) HXCOL=X(I+2)-SQRT(X(I+2)**2-DELTB)-EM6
5246 193      IF (HX(KK,JJ).LE.0.0) ISCRE=ISCRE+1
5247 194      IF (HX(KK,JJ).GE.HXCOL) ISCRF=ISCRF+1
5248 195      90 CONTINUE
5249 196      GO TO 140
5250 197      100 LMNMM=LMNM-NQ
5251 198      LMNPP=LMNP+NQ
5252 199      FOFMM=1.0
5253 200      FOFPP=1.0
5254 201      IMM=I-3
5255 202      IPP=I+2
5256 203      IF (I.NE.3) GO TO 110
5257 204      LMNMM=LMNM
5258 205      FOFMM=0.0
5259 206      IMM=I-2
5260 207      IF (AC(LMNPP).LT.EM6.AND.F(IPJK).LT.EMF) FOFPP=0.0
5261 208      GO TO 130
5262 209      110 IF (I.NE.IM2) GO TO 120
5263 210      LMNPP=LMNP
5264 211      FOFPP=0.0
5265 212      IPP=I+1
5266 213      IF (AC(LMNMM).LT.EM6.AND.F(IMJK).LT.EMF) FOFMM=0.0
5267 214      GO TO 130
5268 215      120 IF (AC(LMNMM).LT.EM6.AND.F(IMJK).LT.EMF) FOFMM=0.0
5269 216      IF (AC(LMNPP).LT.EM6.AND.F(IPJK).LT.EMF) FOFPP=0.0
5270 217      130 DELTA=2.0*((1.0+AC(LMNM)*(F(LMNM)-1.0))*XI(I-1)*DELX(I-1)*FOFM+(1.
5271 218      1.0+AC(LMN)*(F(LMN)-1.0))*XI(I)*DELX(I)+(1.0+AC(LMNP)*(F(LMNP)-1.0)
5272 219      2.0))*XI(I+1)*DELX(I+1)*FOFP+(1.0+AC(LMNMM)*(F(LMNMM)-1.0))*XI(I-2)
5273 220      3.*DELX(I-2)*FOFMM+(1.0+AC(LMNPP)*(F(LMNPP)-1.0))*XI(I+2)*DELX(I+2)
5274 221      4.*FOFPP)
5275 222      DELTB=2.0*(XI(I-2)*DELX(I-2)*FOFMM+XI(I-1)*DELX(I-1)*FOFM+XI(I)
5276 223      1.*DELX(I)+XI(I+1)*DELX(I+1)*FOFP+XI(I+2)*DELX(I+2)*FOFPP)
5277 224      IF (NFC.NE.2) HX(KK,JJ)=SQRT(X(IMM)**2+DELTA)-X(IMM)
5278 225      IF (NFC.EQ.2) HX(KK,JJ)=X(IPP)-SQRT(X(IPP)**2-DELTB)
5279 226      IF (NFC.NE.2) HXCOL=SQRT(X(IMM)**2+DELTB)-X(IMM)-EM6
5280 227      IF (NFC.EQ.2) HXCOL=X(IPP)-SQRT(X(IPP)**2-DELTB)-EM6

```

```

5281 228      IF (HX(KK,JJ).LE.0.0) ISCRE=ISCRE+1          SURF 1ON 228
5282 229      IF (HX(KK,JJ).GE.HXCOL) ISCRF=ISCRF+1        SURF 1ON 229
5283 230      140 CONTINUE                                SURF 1ON 230
5284 231      150 CONTINUE                                SURF 1ON 231
5285 232 C     150 CONTINUE                                SURF 1ON 232
5286 233      IF (K.NE.KM1) GO TO 160                      SURF 1ON 233
5287 234      HX(3,1)=HX(2,1)                            SURF 1ON 234
5288 235      HX(3,2)=HX(2,2)                            SURF 1ON 235
5289 236      HX(3,3)=HX(2,3)                            SURF 1ON 236
5290 237      GO TO 170                                 SURF 1ON 237
5291 238      160 IF (K.NE.2) GO TO 170                  SURF 1ON 238
5292 239      HX(1,1)=HX(2,1)                            SURF 1ON 239
5293 240      HX(1,2)=HX(2,2)                            SURF 1ON 240
5294 241      HX(1,3)=HX(2,3)                            SURF 1ON 241
5295 242      170 CONTINUE                                SURF 1ON 242
5296 243 C     170 CONTINUE                                SURF 1ON 243
5297 244      PHXZ=RZDEN*((HX(3,2)-HX(2,2))*DZB**2+(HX(2,2)-HX(1,2))*DZT**2) SURF 1ON 244
5298 245      PHXY=RYDEN*((HX(2,3)-HX(2,2))*DYF**2+(HX(2,2)-HX(2,1))*DYBK**2) SURF 1ON 245
5299 246 C     ASSUME NEAR VERTICAL SURFACE - COLLAPSE IN AZMUTHAL DIRECTION SURF 1ON 246
5300 247 C     180 CONTINUE                                SURF 1ON 247
5301 248 C     180 CONTINUE                                SURF 1ON 248
5302 249      IF (JBAR.EQ.1) GO TO 350                  SURF 1ON 249
5303 250      ISCVE=0                                  SURF 1ON 250
5304 251      ISCVF=0                                  SURF 1ON 251
5305 252      DO 320 KK=1,3                           SURF 1ON 252
5306 253      N=K-2+KK                               SURF 1ON 253
5307 254      DO 310 II=1,3                           SURF 1ON 254
5308 255      L=I-2+II                               SURF 1ON 255
5309 256      IF (I.EQ.2) L=I+II                         SURF 1ON 256
5310 257      LMN=NQ*(IMAX*JMAX*(N-1)+IMAX*(J-1)+(L-1))+1 SURF 1ON 257
5311 258      LMNM=LMN-II1                            SURF 1ON 258
5312 259      LMNP=LMN+II1                            SURF 1ON 259
5313 260      IF (CYL.LT.0.5) GO TO 250                SURF 1ON 260
5314 261      IF (J.EQ.2) GO TO 180                  SURF 1ON 261
5315 262      IF (J.EQ.JM1) GO TO 210                  SURF 1ON 262
5316 263      IF (I.EQ.2) GO TO 240                  SURF 1ON 263
5317 264      GO TO 250                                 SURF 1ON 264
5318 265 C     180 IF (I.EQ.2) GO TO 200                SURF 1ON 265
5319 266 C     J=2                                    SURF 1ON 266
5320 267 C     190 LL=L                                SURF 1ON 267
5321 268      180 IF (I.EQ.2) GO TO 200                SURF 1ON 268
5322 269      190 LL=L                                SURF 1ON 269
5323 270      M=2                                    SURF 1ON 270
5324 271      IF (WF.EQ.4) M=JM1                         SURF 1ON 271
5325 272      LMNM=NQ*(IMAX*JMAX*(N-1)+IMAX*(M-1)+(LL-1))+1 SURF 1ON 272
5326 273      GO TO 250                               SURF 1ON 273
5327 274      200 IF (L.GT.1) GO TO 190                SURF 1ON 274
5328 275      M=JDP(J)                             SURF 1ON 275
5329 276      LL=2                                    SURF 1ON 276
5330 277      LMN=NQ*(IMAX*JMAX*(N-1)+IMAX*(M-1)+(LL-1))+1 SURF 1ON 277
5331 278      LMNP=LMN-II1                            SURF 1ON 278
5332 279      LMNM=LMN+II1                            SURF 1ON 279
5333 280      IF (WF.EQ.4) GO TO 250                SURF 1ON 280
5334 281      LMNP=LMN                            SURF 1ON 281
5335 282      LMNM=LMN-II1                            SURF 1ON 282
5336 283      GO TO 250                               SURF 1ON 283
5337 284 C     180 IF (I.EQ.2) GO TO 200                SURF 1ON 284
5338 285 C     J=JM1                                SURF 1ON 285
5339 286 C     190 LL=L                                SURF 1ON 286
5340 287      210 IF (I.EQ.2) GO TO 230                SURF 1ON 287
5341 288      220 LL=L                                SURF 1ON 288
5342 289      M=JM1                                SURF 1ON 289
5343 290      IF (WBK.EQ.4) M=2                         SURF 1ON 290
5344 291      LMNP=NQ*(IMAX*JMAX*(N-1)+IMAX*(M-1)+(LL-1))+1 SURF 1ON 291
5345 292      GO TO 250                               SURF 1ON 292
5346 293      230 IF (L.GT.1) GO TO 220                SURF 1ON 293
5347 294      M=JDP(J)                             SURF 1ON 294
5348 295      LL=2                                    SURF 1ON 295
5349 296      LMN=NQ*(IMAX*JMAX*(N-1)+IMAX*(M-1)+(LL-1))+1 SURF 1ON 296
5350 297      LMNM=LMN-II1                            SURF 1ON 297
5351 298      LMNP=LMN-II1                            SURF 1ON 298
5352 299      IF (WBK.EQ.4) GO TO 250                SURF 1ON 299
5353 300      LMNM=LMN                            SURF 1ON 300
5354 301      LMNP=LMN+II1                            SURF 1ON 301
5355 302      GO TO 250                               SURF 1ON 302
5356 303 C     180 IF (I.EQ.2) GO TO 200                SURF 1ON 303
5357 304 C     J=2                                    SURF 1ON 304
5358 305 C     190 LL=L                                SURF 1ON 305
5359 306      240 IF (L.GT.1) GO TO 250                SURF 1ON 306
5360 307      LL=2                                    SURF 1ON 307

```

```

5361 308      M=JOP(J)
5362 309      LMN=NQ*(IMAX*JMAX*(N-1)+IMAX*(M-1)+(LL-1))+1
5363 310      LMNM=LMN+II1
5364 311      LMNP=LMN-II1
5365 312      IF (WF.EQ.4) GO TO 250
5366 313      LMNM=LMN-II1
5367 314      LMNP=LMN+II1
5368 315      250 CONTINUE
5369 316      FOFM=1.0
5370 317      IF (AC(LMNM).LE.EM6.AND.F(IJMK).LT.EMF) FOFM=0.0
5371 318      IF (J.EQ.2.AND.WF.NE.4.AND.F(LMN).LT.EMF) FOFM=0.0
5372 319      FDFP=1.0
5373 320      IF (AC(LMNP).LE.EM6.AND.F(IJPK).LT.EMF) FDFP=0.0
5374 321      IF (J.EQ.JM1.AND.WBK.NE.4.AND.F(LMN).LT.EMF) FDFP=0.0
5375 322      IF (J.GE.3.AND.J.LE.JM2) GO TO 270
5376 323      IF (J.EQ.2) GO TO 260
5377 324      FOFMM=1.0
5378 325      LMNMM=LMNM-II1
5379 326      IF (AC(LMNMM).LT.EM6.AND.F(IJMK).LT.EMF) FOFMM=0.0
5380 327      HY(II,KK)=(1.0+AC(LMNM)*(F(LMN)-1.0))*DELY(J-1)*FOFM+(1.0+AC(LMN)
1 *(F(LMN)-1.0))*DELY(J)+(1.0+AC(LMNMM)*(F(LMNMM)-1.0))*DELY(J-2)
2 *FOFMM
5382 329      HY(II,KK)=HY(II,KK)/X(IM1)
5383 330      HYCOL=(DELY(J-2)*FOFMM+DELY(J-1)*FOFM+DELY(J))/X(IM1)-EM6
5384 331      IF (HY(II,KK).LE.0.0) ISCVE=ISCVE+1
5385 332      IF (HY(II,KK).GE.HYCOL) ISCVF=ISCVF+1
5386 333      GO TO 310
5387 334      260 FOFPP=1.0
5388 335      LMNPP=LMNP+II1
5389 336      IF (AC(LMNPP).LT.EM6.AND.F(IJPK).LT.EMF) FOFPP=0.0
5390 337      HY(II,KK)=(1.0+AC(LMN)*(F(LMN)-1.0))*DELY(J)+(1.0+AC(LMNP)*(F(LMNP
1 )-1.0))*DELY(J+1)+(1.0+AC(LMNPP)*(F(LMNPP)-1.0))*DELY(J+2)
5392 339      HY(II,KK)=HY(II,KK)/X(IM1)
5393 340      HYCOL=(DELY(J+2)*FOFPP+DELY(J+1)*FOFP+DELY(J))/X(IM1)-EM6
5394 341      IF (HY(II,KK).LE.0.0) ISCVE=ISCVE+1
5395 342      IF (HY(II,KK).GE.HYCOL) ISCVF=ISCVF+1
5396 343      GO TO 310
5397 344      270 LMNMM=LMNM-II1
5399 346      LMNPP=LMNP+II1
5400 347      FOFMM=1.0
5401 348      FOFPP=1.0
5402 349      IF (J.NE.3) GO TO 280
5403 350      LMNMM=LMNM
5404 351      FDFMM=0.0
5405 352      IF (AC(LMNPP).LT.EM6.AND.F(IJPK).LT.EMF) FDFPP=0.0
5406 353      GO TO 300
5407 354      280 IF (J.NE.JM2) GO TO 290
5408 355      LMNPP=LMNP
5409 356      FOFPP=0.0
5410 357      IF (AC(LMNMM).LT.EM6.AND.F(IJMK).LT.EMF) FOFMM=0.0
5411 358      GO TO 300
5412 359      290 IF (AC(LMNMM).LT.EM6.AND.F(IJMK).LT.EMF) FDFMM=0.0
5413 360      IF (AC(LMNPP).LT.EM6.AND.F(IJPK).LT.EMF) FOFPP=0.0
5414 361      300 HY(II,KK)=(1.0+AC(LMNM)*(F(LMN)-1.0))*DELY(J-1)*FOFM+(1.0+AC(LMN
1 *(F(LMN)-1.0))*DELY(J)+(1.0+AC(LMNP)*(F(LMNP)-1.0))*DELY(J+1)
2 *FOFP+(1.0+AC(LMNMM)*(F(LMNMM)-1.0))*DELY(J-2)*FOFMM+(1.0+AC
3 *(LMNPP)*(F(LMNPP)-1.0))*DELY(J+2)*FOFPP
5415 362      HY(II,KK)=HY(II,KK)/X(IM1)
5416 363      HYCOL=(DELY(J-2)*FOFMM+DELY(J-1)*FOFM+DELY(J)+DELY(J+1)*FOFP+DELY
1 *(J+2)*FOFPP)/X(IM1)-EM6
5421 368      IF (HY(II,KK).LE.0.0) ISCVE=ISCVE+1
5422 369      IF (HY(II,KK).GE.HYCOL) ISCVF=ISCVF+1
5423 370      310 CONTINUE
5424 371      320 CONTINUE
5425 372      C
5426 373      IF (K.NE.KM1) GO TO 330
5427 374      HY(1,3)=HY(1,2)
5428 375      HY(2,3)=HY(2,2)
5429 376      HY(3,3)=HY(3,2)
5430 377      GO TO 340
5431 378      330 IF (K.NE.2) GO TO 340
5432 379      HY(1,1)=HY(1,2)
5433 380      HY(2,1)=HY(2,2)
5434 381      HY(3,1)=HY(3,2)
5435 382      340 CONTINUE
5436 383      C
5437 384      PHYX=RXDEN*((HY(3,2)-HY(2,2))*DXL**2+(HY(2,2)-HY(1,2))*DXR**2)
5438 385      PHYZ=RZDEN*((HY(2,3)-HY(2,2))*DZB**2+(HY(2,2)-HY(2,1))*DZT**2)
5439 386      IF (I.EQ.2) PHYX=RXDEN*((HY(2,2)-HY(3,2))*(DXL**2)+(HY(2,2)-HY(1,2
1 ))*DXR*(DXR+2.0*DXL))

```

```

5441 388      IF ( I.EQ.2) PHYZ=RZDEN*((HY(1,3)-HY(1,2))*(DZB**2)+(HY(1,2)-HY(1,1)
5442 389      1 ))*(DZT**2))
5443 390      PHYX=XI(I)*PHYX
5444 391      PHYZ=XI(I)*PHYZ
5445 392      GO TO 360
5446 393      350 PHYX=1.OE10
5447 394      PHYZ=1.OE10
5448 395      360 CONTINUE
5449 396 C      ASSUME NEAR HORIZONTAL SURFACE - COLLAPSE IN AXIAL DIRECTION
5450 397 C
5451 398 C
5452 399      ISCHE=0
5453 400      DO 520 II=1,3
5454 401      L=I-2+II
5455 402      IF ( I.EQ.2) L=1+II
5456 403      DO 510 JJ=1,3
5457 404      M=J-2+JJ
5458 405      LMN=NQ*(IMAX*JMAX*(K-1)+IMAX*(M-1)+(L-1))+1
5459 406      LMNM=LMN-II2
5460 407      LMNP=LMN+II2
5461 408      IF ( CYL.LT.0.5) GO TO 450
5462 409      IF ( M.EQ.1) GO TO 370
5463 410      IF ( M.EQ.JMAX) GO TO 400
5464 411      IF ( L.EQ.1) GO TO 430
5465 412      GO TO 450
5466 413 C      M=1
5467 414 C
5468 415 C
5469 416      370 IF ( L.EQ.1) GO TO 390
5470 417      LL=L
5471 418      MM=2
5472 419      IF ( WF.EQ.4) MM=JM1
5473 420      380 LMN=NQ*(IMAX*JMAX*(K-1)+IMAX*(MM-1)+(LL-1))+1
5474 421      GO TO 440
5475 422      390 LL=2
5476 423      MM=JM2
5477 424      IF ( WF.EQ.4) MM=JOP(J)+1
5478 425      GO TO 380
5479 426 C
5480 427 C      M=JMAX
5481 428 C
5482 429      400 IF ( L.EQ.1) GO TO 420
5483 430      LL=L
5484 431      MM=JM1
5485 432      IF ( WBK.EQ.4) M=2
5486 433      410 LMN=NQ*(IMAX*JMAX*(K-1)+IMAX*(MM-1)+(LL-1))+1
5487 434      GO TO 440
5488 435      420 LL=2
5489 436      MM=3
5490 437      IF ( WBK.EQ.4) MM=JOP(JM1)-1
5491 438      GO TO 410
5492 439 C      I=2
5493 440 C
5494 441 C
5495 442      430 LL=2
5496 443      IF ( JJ.EQ.1) MM=JOP(J+1)
5497 444      IF ( JJ.EQ.2) MM=JOP(J)
5498 445      IF ( JJ.EQ.3) MM=JOP(J-1)
5499 446      GO TO 380
5500 447      440 LMNM=LMN-II2
5501 448      LMNP=LMN+II2
5502 449      450 CONTINUE
5503 450      FOFM=1.0
5504 451      IF ( AC(LMNM).LE.EM6.AND.F(IJKM).LT.EMF) FOFM=0.0
5505 452      FOFP=1.0
5506 453      IF ( AC(LMNP).LE.EM6.AND.F(IJKP).LT.EMF) FOFP=0.0
5507 454      IF ( K.GE.3.AND.K.LE.KM2) GO TO 470
5508 455      IF ( K.EQ.2) GO TO 460
5509 456      FOFMM=1.0
5510 457      LMNMM=LMNM-II2
5511 458      IF ( AC(LMNMM).LT.EM6.AND.F(IJKM).LT.EMF) FOFMM=0.0
5512 459      HZ(II,JJ)=(1.0+AC(LMNM)*(F(LMNM)-1.0))*DELZ(K-1)*FOFM+(1.0+AC(LMN
5513 460      1 *(F(LMN)-1.0))*DELZ(K)+(1.0+AC(LMNMM)*(F(LMNMM)-1.0))*DELZ(K-2)
5514 461      2 *FOFMM
5515 462      HZCOL=DELZ(K-2)*FOFMM+DELZ(K-1)*FOFM+DELZ(K)-EM6
5516 463      IF ( HZ(II,JJ).LE.0.0) ISCHE=ISCHE+1
5517 464      IF ( HZ(II,JJ).GE.HZCOL) ISCHF=ISCHF+1
5518 465      GO TO 510
5519 466      460 FOFPP=1.0
5520 467      LMNPP=LMNP+II2

```

```

5521 468 IF ( AC(LMNPP).LT.EM6.AND.F(IJKP).LT.EMF) FOFPP=0.0 SURF1ON 468
5522 469 HZ(II,JJ)=(1.0+AC(LMN)*(F(LMN)-1.0))*DELZ(K)+(1.0+AC(LMNP)*(F(LMNP SURF1ON 469
5523 470 1 )-1.0))*DELZ(K+1)+(1.0+AC(LMNPP)*(F(LMNPP)-1.0))*DELZ(K+2) SURF1ON 470
5524 471 HZCOL=DELZ(K+2)*FOFPP+DELZ(K+1)*FOFP+DELZ(K)-EM6 SURF1ON 471
5525 472 IF (HZ(II,JJ).LE.0.0) ISCHE=ISCHE+1 SURF1ON 472
5526 473 IF (HZ(II,JJ).GE.HZCOL) ISCHF=ISCHF+1 SURF1ON 473
5527 474 GO TO 510 SURF1ON 474
5528 475 470 LMNMM=LMNM-II2 SURF1ON 475
5529 476 LMNPP=LMNP+II2 SURF1ON 476
5530 477 FOFMM=1.0 SURF1ON 477
5531 478 FOFPP=1.0 SURF1ON 478
5532 479 IF (K.NE.3) GO TO 480 SURF1ON 479
5533 480 LMNMM=LMNM SURF1ON 480
5534 481 FOFMM=0.0 SURF1ON 481
5535 482 IF (AC(LMNPP).LT.EM6.AND.F(IJKP).LT.EMF) FOFPP=0.0 SURF1ON 482
5536 483 GO TO 500 SURF1ON 483
5537 484 480 IF (K.NE.KM2) GO TO 490 SURF1ON 484
5538 485 LMNPP=LMNP SURF1ON 485
5539 486 FOFPP=0.0 SURF1ON 486
5540 487 IF (AC(LMNMM).LT.EM6.AND.F(IJKM).LT.EMF) FOFMM=0.0 SURF1ON 487
5541 488 GO TO 500 SURF1ON 488
5542 489 490 IF (AC(LMNMM).LT.EM6.AND.F(IJKM).LT.EMF) FOFMM=0.0 SURF1ON 489
5543 490 IF (AC(LMNPP).LT.EM6.AND.F(IJKP).LT.EMF) FOFPP=0.0 SURF1ON 490
5544 491 500 HZ(II,JJ)=(1.0+AC(LMN)*(F(LMN)-1.0))*DELZ(K-1)*FOFM+(1.0+AC(LMN) SURF1ON 491
5545 492 1 *(F(LMN)-1.0))*DELZ(K)+(1.0+AC(LMNP)*(F(LMN)-1.0))*DELZ(K+1) SURF1ON 492
5546 493 2 *FOFP+(1.0+AC(LMNMM)*(F(LMNMM)-1.0))*DELZ(K-2)*FOFMM+(1.0+AC SURF1ON 493
5547 494 3 (LMNPP)*(F(LMNPP)-1.0))*DELZ(K+2)*FOFPP SURF1ON 494
5548 495 HZCOL=DELZ(K-2)*FOFMM+DELZ(K-1)*FOFM+DELZ(K)+DELZ(K+1)*FOFP+DELZ(K SURF1ON 495
5549 496 1 +2)*FOFPP-EM6 SURF1ON 496
5550 497 IF (HZ(II,JJ).LE.0.0) ISCHE=ISCHE+1 SURF1ON 497
5551 498 IF (HZ(II,JJ).GE.HZCOL) ISCHF=ISCHF+1 SURF1ON 498
5552 499 510 CONTINUE SURF1ON 499
5553 500 520 CONTINUE SURF1ON 500
5554 501 C
5555 502 PHZX=RXDEN*((HZ(3,2)-HZ(2,2))*DXL**2+(HZ(2,2)-HZ(1,2))*DXR**2) SURF1ON 501
5556 503 PHZY=RYDEN*((HZ(2,3)-HZ(2,2))*DYF**2+(HZ(2,2)-HZ(2,1))*DYBK**2) SURF1ON 502
5557 504 IF (I.EQ.2) PHZX=RXDEN*((HZ(2,2)-HZ(3,2))*(DXL**2)+(HZ(2,2)-HZ(1,2 SURF1ON 504
5558 505 1 )*(DXR+2.0*DXL)*DXR) SURF1ON 505
5559 506 IF (I.EQ.2) PHZY=RYDEN*((HZ(1,3)-HZ(1,2))*(DYF**2)+(HZ(1,2)-HZ(1,1 SURF1ON 506
5560 507 1 )*(DYBK**2)) SURF1ON 507
5561 508 C
5562 509 IF (ISCR+ISCV+ISCH.EQ.3) GO TO 720 SURF1ON 509
5563 510 QX=PHXY**2+PHXZ**2 SURF1ON 510
5564 511 QY=PHYX**2+PHYZ**2 SURF1ON 511
5565 512 QZ=PHZX**2+PHZY**2 SURF1ON 512
5566 513 IF (QX+QY+QZ.GT.1.OE-10) GO TO 530 SURF1ON 513
5567 514 PS(IJK)=0.0 SURF1ON 514
5568 515 GO TO 1070 SURF1ON 515
5569 516 C
5570 517 530 IF (QX.LT.QY.AND.QX.LT.QZ) GO TO 540 SURF1ON 517
5571 518 IF (QY.LT.QX.AND.QY.LT.QZ) GO TO 600 SURF1ON 518
5572 519 IF (QZ.LT.QX.AND.QZ.LT.QY) GO TO 660 SURF1ON 519
5573 520 C
5574 521 540 NFSLP=1 SURF1ON 521
5575 522 IF (F(IPJK).GT.F(IMJK)) NFSLP=2 SURF1ON 522
5576 523 IF (NFSLP.EQ.NF(IJK)) GO TO 720 SURF1ON 523
5577 524 IF (NFCAL.EQ.2) GO TO 590 SURF1ON 524
5578 525 IF (NFCAL.EQ.3) GO TO 550 SURF1ON 525
5579 526 GO TO 720 SURF1ON 526
5580 527 C
      CONFLICT SURF1ON 527
5581 528 550 IF (BETA(IMJK).LT.0.0.OR.BETA(IPJK).LT.0.0) GO TO 590 SURF1ON 528
5582 529 C
5583 530 INB=0 SURF1ON 530
5584 531 NFREF=0 SURF1ON 531
5585 532 DO 560 JJ=1,3 SURF1ON 532
5586 533 DD 560 II=1,3 SURF1ON 533
5587 534 N=K-1 SURF1ON 534
5588 535 M=J-2+JJ SURF1ON 535
5589 536 L=I-2+II SURF1ON 536
5590 537 LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1 SURF1ON 537
5591 538 IF (NF(LMN).LT.1.OR.NF(LMN).GT.6) GO TO 560 SURF1ON 538
5592 539 NFNB=NF(LMN) SURF1ON 539
5593 540 IF (NFREF.EQ.0) NFREF=NFNB SURF1ON 540
5594 541 IF (NFNB.NE.NFREF) INB=1 SURF1ON 541
5595 542 560 CONTINUE SURF1ON 542
5596 543 DO 570 KK=2,3 SURF1ON 543
5597 544 DO 570 II=1,3 SURF1ON 544
5598 545 N=K-2+KK SURF1ON 545
5599 546 M=J-1 SURF1ON 545
5600 547 L=I-2+II SURF1ON 547

```

```

5601 548      LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1          SURF 1ON 548
5602 549      IF (NF(LMN).LT.1.OR.NF(LMN).GT.6) GO TO 570    SURF 1ON 549
5603 550      NFNB=NF(LMN)                                     SURF 1ON 550
5604 551      IF (NFREF.EQ.O) NFREF=NFNB                      SURF 1ON 551
5605 552      IF (NFNB.NE.NFREF) INB=1                      SURF 1ON 552
5606 553      570 CONTINUE                                     SURF 1ON 553
5607 554 C      IF (NF(IMJK).LT.1.OR.NF(IMJK).GT.6) GO TO 580  SURF 1ON 554
5608 555      NFNB=NF(IMJK)                                    SURF 1ON 555
5609 556      IF (NFREF.EQ.O) NFREF=NFNB                      SURF 1ON 556
5610 557      IF (NFNB.NE.NFREF) INB=1                      SURF 1ON 557
5611 558      580 CONTINUE                                     SURF 1ON 558
5612 559 C      IF (NFREF.EQ.O) GO TO 720                      SURF 1ON 559
5613 560 C      IF (INB.EQ.1) GO TO 720                      SURF 1ON 560
5614 561      IF (NFREF.NE.NFSLP) GO TO 720                  SURF 1ON 561
5615 562 C      IF (NFREF.NE.NFSLP) GO TO 720                  SURF 1ON 562
5616 563 C      IF (INB.EQ.1) GO TO 720                      SURF 1ON 563
5617 564 C      IF (NFREF.NE.NFSLP) GO TO 720                  SURF 1ON 564
5618 565 C      IF (NFREF.NE.NFSLP) GO TO 720                  SURF 1ON 565
5619 566 C      590 NF(IJK)=NFSLP                         SURF 1ON 566
5620 567      GO TO 720                                      SURF 1ON 567
5621 568 C      590 NF(IJK)=NFSLP                         SURF 1ON 568
5622 569 C      GO TO 720                                      SURF 1ON 569
5623 570      600 NFSLP=3                                     SURF 1ON 570
5624 571      IF (F(IJPK).GT.F(IJMK)) NFSLP=4             SURF 1ON 571
5625 572      IF (NFSLP.EQ.NF(IJK)) GO TO 720              SURF 1ON 572
5626 573      IF (NFCAL.EQ.2) GO TO 650                   SURF 1ON 573
5627 574      IF (NFCAL.EQ.3) GO TO 610                   SURF 1ON 574
5628 575      GO TO 720                                      SURF 1ON 575
5629 576 C      CONFLICT                                     SURF 1ON 576
5630 577      610 IF (BETA(IJMK).LT.0.0.OR.BETA(IJPK).LT.0.0) GO TO 650  SURF 1ON 577
5631 578 C      610 IF (BETA(IJMK).LT.0.0.OR.BETA(IJPK).LT.0.0) GO TO 650  SURF 1ON 578
5632 579      INB=0                                         SURF 1ON 579
5633 580      NFREF=0                                       SURF 1ON 580
5634 581      DO 620 JJ=1,3                                SURF 1ON 581
5635 582      DO 620 II=1,3                                SURF 1ON 582
5636 583      N=K-1                                       SURF 1ON 583
5637 584      M=J-2+JJ                                     SURF 1ON 584
5638 585      L=I-2+II                                     SURF 1ON 585
5639 586      LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1          SURF 1ON 586
5640 587      IF (NF(LMN).LT.1.OR.NF(LMN).GT.6) GO TO 620    SURF 1ON 587
5641 588      NFNB=NF(LMN)                                     SURF 1ON 588
5642 589      IF (NFREF.EQ.O) NFREF=NFNB                      SURF 1ON 589
5643 590      IF (NFNB.NE.NFREF) INB=1                      SURF 1ON 590
5644 591      620 CONTINUE                                     SURF 1ON 591
5645 592      DO 630 KK=2,3                                SURF 1ON 592
5646 593      DO 630 II=1,3                                SURF 1ON 593
5647 594      N=K-2+KK                                     SURF 1ON 594
5648 595      M=J-1                                       SURF 1ON 595
5649 596      L=I-2+II                                     SURF 1ON 596
5650 597      LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1          SURF 1ON 597
5651 598      IF (NF(LMN).LT.1.OR.NF(LMN).GT.6) GO TO 630    SURF 1ON 598
5652 599      NFNB=NF(LMN)                                     SURF 1ON 599
5653 600      IF (NFREF.EQ.O) NFREF=NFNB                      SURF 1ON 600
5654 601      IF (NFNB.NE.NFREF) INB=1                      SURF 1ON 601
5655 602      630 CONTINUE                                     SURF 1ON 602
5656 603 C      630 CONTINUE                                     SURF 1ON 603
5657 604      IF (NF(IMJK).LT.1.OR.NF(IMJK).GT.6) GO TO 640    SURF 1ON 604
5658 605      NFNB=NF(IMJK)                                    SURF 1ON 605
5659 606      IF (NFREF.EQ.O) NFREF=NFNB                      SURF 1ON 606
5660 607      IF (NFNB.NE.NFREF) INB=1                      SURF 1ON 607
5661 608      640 CONTINUE                                     SURF 1ON 608
5662 609 C      IF (NFREF.EQ.O) GO TO 720                  SURF 1ON 609
5663 610 C      IF (INB.EQ.1) GO TO 720                  SURF 1ON 610
5664 611 C      IF (INB.EQ.1) GO TO 720                  SURF 1ON 611
5665 612 C      IF (NFREF.NE.NFSLP) GO TO 720              SURF 1ON 612
5666 613 C      IF (NFREF.NE.NFSLP) GO TO 720              SURF 1ON 613
5667 614 C      IF (NFREF.NE.NFSLP) GO TO 720              SURF 1ON 614
5668 615 C      650 NF(IJK)=NFSLP                         SURF 1ON 615
5669 616      GO TO 720                                      SURF 1ON 616
5670 617 C      650 NF(IJK)=NFSLP                         SURF 1ON 617
5671 618 C      660 NFSLP=5                                     SURF 1ON 618
5672 619      IF (F(IJPK).GT.F(IJMK)) NFSLP=6             SURF 1ON 619
5673 620      IF (NFSLP.EQ.NF(IJK)) GO TO 720              SURF 1ON 620
5674 621      IF (NFCAL.EQ.2) GO TO 710                   SURF 1ON 621
5675 622      IF (NFCAL.EQ.3) GO TO 670                   SURF 1ON 622
5676 623 C      GO TO 720                                      SURF 1ON 623
5677 624 C      CONFLICT                                     SURF 1ON 624
5678 625 C      670 IF (BETA(IJMK).LT.0.0.OR.BETA(IJPK).LT.0.0) GO TO 710  SURF 1ON 625
5679 626 C      670 IF (BETA(IJMK).LT.0.0.OR.BETA(IJPK).LT.0.0) GO TO 710  SURF 1ON 626
5680 627 C      670 IF (BETA(IJMK).LT.0.0.OR.BETA(IJPK).LT.0.0) GO TO 710  SURF 1ON 627

```

5681	628	INB=0	SURF 1ON	628
5682	629	NREF=0	SURF 1ON	629
5683	630	DO 680 JJ=1,3	SURF 1ON	630
5684	631	DO 680 II=1,3	SURF 1ON	631
5685	632	N=K-1	SURF 1ON	632
5686	633	M=J-2+JJ	SURF 1ON	633
5687	634	L=I-2+II	SURF 1ON	634
5688	635	LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1	SURF 1ON	635
5689	636	IF (NF(LMN).LT.1.OR.NF(LMN).GT.6) GO TO 680	SURF 1ON	636
5690	637	NFB=NF(LMN)	SURF 1ON	637
5691	638	IF (NFREF.EQ.0) NFREF=NFB	SURF 1ON	638
5692	639	IF (NFB.NE.NFREF) INB=1	SURF 1ON	639
5693	640	680 CONTINUE	SURF 1ON	640
5694	641	DO 690 KK=2,3	SURF 1ON	641
5695	642	DO 690 II=1,3	SURF 1ON	642
5696	643	N=K-2+KK	SURF 1ON	643
5697	644	M=J-1	SURF 1ON	644
5698	645	L=I-2+II	SURF 1ON	645
5699	646	LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1	SURF 1ON	646
5700	647	IF (NF(LMN).LT.1.OR.NF(LMN).GT.6) GO TO 690	SURF 1ON	647
5701	648	NFB=NF(LMN)	SURF 1ON	648
5702	649	IF (NFREF.EQ.0) NFREF=NFB	SURF 1ON	649
5703	650	IF (NFB.NE.NFREF) INB=1	SURF 1ON	650
5704	651	690 CONTINUE	SURF 1ON	651
5705	652	C IF (NF(IMJK).LT.1.OR.NF(IMJK).GT.6) GO TO 700	SURF 1ON	652
5706	653	NFB=NF(IMJK)	SURF 1ON	653
5707	654	IF (NFREF.EQ.0) NFREF=NFB	SURF 1ON	654
5708	655	IF (NFB.NE.NFREF) INB=1	SURF 1ON	655
5709	656	700 CONTINUE	SURF 1ON	656
5710	657	C IF (NFREF.EQ.0) GO TO 720	SURF 1ON	657
5711	658	C IF (INB.EQ.1) GO TO 720	SURF 1ON	658
5712	659	C IF (NFREF.NE.NFSLP) GO TO 720	SURF 1ON	659
5713	660	C 710 NF(IJK)=NFSLP	SURF 1ON	660
5714	661	GO TO 720	SURF 1ON	661
5715	662	C 5716 663 C 720 CONTINUE	SURF 1ON	662
5717	664	C 5718 665 710 NF(IJK)=NFSLP	SURF 1ON	663
5719	666	GO TO 720	SURF 1ON	664
5720	667	C 5721 668 720 CONTINUE	SURF 1ON	665
5722	669	C 5723 670 NFS(IJK)=NFSLP	SURF 1ON	666
5724	671	NFF=NF(IJK)	SURF 1ON	667
5725	672	CTH=1.0	SURF 1ON	668
5726	673	STH=0.0	SURF 1ON	669
5727	674	FLGE=1.0	SURF 1ON	670
5728	675	FLGW=1.0	SURF 1ON	671
5729	676	FLGN=1.0	SURF 1ON	672
5730	677	FLGS=1.0	SURF 1ON	673
5731	678	GO TO (730,730,790,790,880,880), NFF	SURF 1ON	674
5732	679	730 CONTINUE	SURF 1ON	675
5733	680	RDEW=RRI(I)*RDY(J)	SURF 1ON	676
5734	681	RDNS=RDZ(K)	SURF 1ON	677
5735	682	IF (CYL.LT.0.5) GO TO 760	SURF 1ON	678
5736	683	HXE=(DTH*DTHMA*RDTHS/DTHP)*HX(2,3)+(DTHE*DTHMA*RPDTH)*HX(2,2)-	SURF 1ON	679
5737	684	1 (DTHE*DTH*RDTHS/DTHM)*HX(2,1)	SURF 1ON	680
5738	685	HXW=(DTH*DTHPA*RDTHS/DTHM)*HX(2,1)+(DTHW*DTHPA*RPDTH)*HX(2,2)-	SURF 1ON	681
5739	686	1 (DTHW*DTH*RDTHS/DTHP)*HX(2,3)	SURF 1ON	682
5740	687	TH=0.5*DELY(J)*RX(IM1)	SURF 1ON	683
5741	688	CTH=COS(TH)	SURF 1ON	684
5742	689	STH=SIN(TH)	SURF 1ON	685
5743	690	RDEW=1.0/(XI(1)*2.0*STH)	SURF 1ON	686
5744	691	IF (NFF.EQ.2) GO TO 740	SURF 1ON	687
5745	692	IF (I.GT.3) RSURF=X(I-3)+HX(2,2)	SURF 1ON	688
5746	693	IF (I.GT.3) RSURFE=X(I-3)+HXE	SURF 1ON	689
5747	694	IF (I.GT.3) RSURFW=X(I-3)+HXW	SURF 1ON	690
5748	695	IF (I.EQ.3) RSURF=X(I-2)+HX(2,2)	SURF 1ON	691
5749	696	IF (I.EQ.3) RSURFE=X(I-2)+HXE	SURF 1ON	692
5750	697	IF (I.EQ.3) RSURFW=X(I-2)+HXW	SURF 1ON	693
5751	698	IF (I.EQ.2) RSURF=X(I-1)+HX(2,2)	SURF 1ON	694
5752	699	IF (I.EQ.2) RSURF=AMAX1(RSURF,O.1*DELX(2))	SURF 1ON	695
5753	700	IF (I.EQ.2) RSURFE=X(I-1)+HXE	SURF 1ON	696
5754	701	IF (I.EQ.2) RSURFE=AMAX1(RSURFE,O.1*DELX(2))	SURF 1ON	697
5755	702	IF (I.EQ.2) RSURFW=X(I-1)+HXW	SURF 1ON	698
5756	703	IF (I.EQ.2) RSURFW=AMAX1(RSURFW,O.1*DELX(2))	SURF 1ON	699
5757	704	GO TO 750	SURF 1ON	700
5758	705	740 IF (I.LT.IM2) RSURF=X(I+2)-HX(2,2)	SURF 1ON	701
5759	706	IF (I.LT.IM2) RSURFE=X(I+2)-HXE	SURF 1ON	702
5760	707	IF (I.LT.IM2) RSURFW=X(I+2)-HXW	SURF 1ON	703

```

5761 708      IF (I.EQ.IM2) RSURF=X(I+1)-HX(2,2)          SURF 1ON   708
5762 709      IF (I.EQ.IM2) RSURFE=X(I+1)-HXE           SURF 1ON   709
5763 710      IF (I.EQ.IM2) RSURFW=X(I+1)-HXW           SURF 1ON   710
5764 711      IF (I.EQ.IM1) RSURF=X(I)-HX(2,2)          SURF 1ON   711
5765 712      IF (I.EQ.IM1) RSURFE=X(I)-HXE           SURF 1ON   712
5766 713      IF (I.EQ.IM1) RSURFW=X(I)-HXW           SURF 1ON   713
5767 714      750 RDEW=1.0/(RSURF*2.0*STH)          SURF 1ON   714
5768 715      IF (NFF.EQ.2) STH=-STH                 SURF 1ON   715
5769 716      760 CONTINUE                         SURF 1ON   716
5770 717      AFE=ABK(IJK)                         SURF 1ON   717
5771 718      IF (CYL.EQ.1.0.AND.AFE.LT.EM6) AFE=1.0    SURF 1ON   718
5772 719      AFW=ABK(IJMK)                         SURF 1ON   719
5773 720      IF (CYL.EQ.1.0.AND.AFW.LT.EM6) AFW=1.0    SURF 1ON   720
5774 721      AFN=AT(IJK)                           SURF 1ON   721
5775 722      AFS=AT(IJMK)                         SURF 1ON   722
5776 723      AFLOR=AR(IJMK)                         SURF 1ON   723
5777 724      IF (NFF.EQ.2) AFLOR=AR(IJK)           SURF 1ON   724
5778 725      IF (NOWALL.EQ.1) AFLOR=1.0            SURF 1ON   725
5779 726 C   ** USE FLG- TO INDICATE WALL ADHESION AT CELL BOTTOM SURF 1ON   726
5780 727      IF (AFE.GT.EM6.AND.F(IJPK).LT.EMF) FLGE=AFLOR    SURF 1ON   727
5781 728      IF (AFW.GT.EM6.AND.F(IJMK).LT.EMF) FLGW=AFLOR    SURF 1ON   728
5782 729      IF (AFN.GT.EM6.AND.F(IJKP).LT.EMF) FLGN=AFLOR    SURF 1ON   729
5783 730      IF (AFS.GT.EM6.AND.F(IJKM).LT.EMF) FLGS=AFLOR    SURF 1ON   730
5784 731      PHXZB=(HX(2,2)-HX(1,2))/DZB          SURF 1ON   731
5785 732      PHXZT=(HX(3,2)-HX(2,2))/DZT          SURF 1ON   732
5786 733      IF (CYL.LT.0.5) GO TO 990           SURF 1ON   733
5787 734      DHNA=0.5*(PHXZT+PHXZB)           SURF 1ON   734
5788 735      FCYL=0.5*DELZ(K)*DHNA*RXI(I)        SURF 1ON   735
5789 736      IF (NFF.EQ.2) FCYL=-FCYL           SURF 1ON   736
5790 737      FLGN=FLGN*(1.0+FCYL)             SURF 1ON   737
5791 738      FLGS=FLGS*(1.0-FCYL)             SURF 1ON   738
5792 739 C   BUILDING FOUR SETS OF SIX POINT COEFFICIENTS SURF 1ON   739
5793 740 C   COORDINATES ARE SUCH THAT (E,W) GOES WITH (J+1,J-1) SURF 1ON   740
5794 741 C   COORDINATES ARE SUCH THAT (N,S) GOES WITH (K+1,K-1) SURF 1ON   741
5795 742 C   Z DERIVATIVE ON E FACE (FOR DHNE)          SURF 1ON   742
5796 743 C   5797 744 C   SURF 1ON   743
5798 745 C   SURF 1ON   744
5799 746 C   SURF 1ON   745
5800 747      ANE(3,3)=(1.0+3.0*FRTHP*RATZM)/(3.0*DZP)    SURF 1ON   746
5801 748      ANE(2,3)=-(1.0+ZP)*ANE(3,3)+(FRTHP/DZM)     SURF 1ON   747
5802 749      ANE(1,3)=ZP*ANE(3,3)-(FRTHP/DZM)           SURF 1ON   748
5803 750      ANE(3,2)=-ANE(3,3)+ZM*RDZS              SURF 1ON   749
5804 751      ANE(2,2)=-(1.0+ZP)*ANE(3,3)-(FRTHP/DZM)+DZD*RPDZ SURF 1ON   750
5805 752      ANE(1,2)=-ZP*ANE(3,3)+(FRTHP/DZM)-ZP*RDZS    SURF 1ON   751
5806 753      ANE(3,1)=0.0                            SURF 1ON   752
5807 754      ANE(2,1)=0.0                            SURF 1ON   753
5808 755      ANE(1,1)=0.0                            SURF 1ON   754
5809 756 C   5810 757 C   Z DERIVATIVE ON W FACE (FOR DHNW)          SURF 1ON   755
5811 758 C   5812 759      ANW(1,1)=-(1.0+3.0*FRTHM*RATZP)/(3.0*DZM)    SURF 1ON   756
5813 760      ANW(2,1)=-(1.0+ZM)*ANW(1,1)-(FRTHM/DZP)     SURF 1ON   757
5814 761      ANW(3,1)=ZM*ANW(1,1)+(FRTHM/DZP)           SURF 1ON   758
5815 762      ANW(1,2)=-ANW(1,1)-ZP*RDZS              SURF 1ON   759
5816 763      ANW(2,2)=(1.0+ZM)*ANW(1,1)+(FRTHM/DZP)+DZD*RPDZ SURF 1ON   760
5817 764      ANW(3,2)=-ZM*ANW(1,1)-(FRTHM/DZP)+ZM*RDZS    SURF 1ON   761
5818 765      ANW(1,3)=0.0                            SURF 1ON   762
5819 766      ANW(2,3)=0.0                            SURF 1ON   763
5820 767      ANW(3,3)=0.0                            SURF 1ON   764
5821 768 C   5822 769 C   THETA DERIVATIVE ON N FACE (FOR DHEN)          SURF 1ON   765
5823 770 C   5824 771      AEN(3,3)=(1.0+3.0*FRZP*RATTHM)/(3.0*DTHP)    SURF 1ON   766
5825 772      AEN(3,2)=-(1.0+THP)*AEN(3,3)+(FRZP/DTHM)     SURF 1ON   767
5826 773      AEN(3,1)=THP*AEN(3,3)-(FRZP/DTHM)           SURF 1ON   768
5827 774      AEN(2,3)=-AEN(3,3)+THM*RDTHS              SURF 1ON   769
5828 775      AEN(2,2)=(1.0+THP)*AEN(3,3)-(FRZP/DTHM)+DTHD*RPDTH SURF 1ON   770
5829 776      AEN(2,1)=-THP*AEN(3,3)+(FRZP/DTHM)-THP*RDTHS    SURF 1ON   771
5830 777      AEN(1,3)=0.0                            SURF 1ON   772
5831 778      AEN(1,2)=0.0                            SURF 1ON   773
5832 779      AEN(1,1)=0.0                            SURF 1ON   774
5833 780 C   5834 781 C   THETA DERIVATIVE ON S FACE (FOR DHES)          SURF 1ON   775
5835 782 C   5836 783      AES(1,1)=-(1.0+3.0*FRZM*RATTHP)/(3.0*DTHM)    SURF 1ON   776
5837 784      AES(1,2)=-(1.0+THM)*AES(1,1)-(FRZM/DTHP)     SURF 1ON   777
5838 785      AES(1,3)=THM*AES(1,1)+(FRZM/DTHP)           SURF 1ON   778
5839 786      AES(2,1)=-AES(1,1)-THP*RDTHS              SURF 1ON   779
5840 787      AES(2,2)=(1.0+THM)*AES(1,1)+(FRZM/DTHP)+DTHD*RPDTH SURF 1ON   780

```

5841	788	AES(2,3)=-THM*AES(1,1)-(FRZM/DTHP)+THM*RDTHS	SURF 1ON	788
5842	789	AES(3,1)=0.0	SURF 1ON	789
5843	790	AES(3,2)=0.0	SURF 1ON	790
5844	791	AES(3,3)=0.0	SURF 1ON	791
5845	792 C	BUILDING FOUR SETS OF SEVEN POINT COEFFICIENTS	SURF 1ON	792
5846	793 C	THETA DERIVATIVE ON E FACE (FOR DHEE)	SURF 1ON	793
5847	794 C	CEE(3,3)=RPDTH*RDTHS*RATZM*(DTHDE*(DTHMA**2)+(DTHMA+DTH)*(DTHE**2)) 1)/(3.0*DTHP)	SURF 1ON	794
5848	795 C	CEE(2,3)=-(1.0+ZP)*CEE(3,3)+(DTHMA+DTH)*RDTHS/DTHP	SURF 1ON	795
5849	796 C	CEE(1,3)=ZP*CEE(3,3)	SURF 1ON	796
5850	797	CEE(3,2)=-CEE(3,3)	SURF 1ON	797
5851	798	CEE(2,2)=-(1.0+ZP)*CEE(3,3)+(DTHDE-DTHM)*RPDTH	SURF 1ON	798
5852	799	CEE(1,2)=-ZP*CEE(3,3)	SURF 1ON	799
5853	800	CEE(3,1)=0.0	SURF 1ON	800
5854	801	CEE(2,1)=-DTHDE*RDTHS/DTHM	SURF 1ON	801
5855	802	CEE(1,1)=0.0	SURF 1ON	802
5856	803	THETA DERIVATIVE ON W FACE (FOR DHEW)	SURF 1ON	803
5857	804	CEW(1,1)=-RPDTH*RDTHS*RATZP*(DTHDW*(DTHPA**2)+(DTHPA+DTH)*(DTHW**2)) 1)/(3.0*DTHM)	SURF 1ON	804
5858	805	CEW(2,1)=-(1.0+ZM)*CEW(1,1)-(DTHPA+DTH)*RDTHS/DTHM	SURF 1ON	805
5859	806	CEW(3,1)=ZM*CEW(1,1)	SURF 1ON	806
5860	807 C	CEW(1,2)=-CEW(1,1)	SURF 1ON	807
5861	808 C	CEW(2,2)=-(1.0+ZM)*CEW(1,1)-(DTHDW-DTHP)*RPDTH	SURF 1ON	808
5862	809 C	CEW(3,2)=-ZM*CEW(1,1)	SURF 1ON	809
5863	810	CEW(1,3)=0.0	SURF 1ON	810
5864	811	CEW(2,3)=DTHDW*RDTHS/DTHP	SURF 1ON	811
5865	812	CEW(3,3)=0.0	SURF 1ON	812
5866	813	Z DERIVATIVE ON N FACE (FOR DHNN)	SURF 1ON	813
5867	814	CNN(1,1)=-RPDZ*RDZS*RATTHM*(DZDN*(DZMA**2)+(DZMA+DZ)*(DZN**2))/(3.0 1 *DZP)	SURF 1ON	814
5868	815	CNN(3,2)=-(1.0+THP)*CNN(3,3)+(DZMA+DZ)*RDZS/DZP	SURF 1ON	815
5869	816	CNN(3,1)=THP*CNN(3,3)	SURF 1ON	816
5870	817	CNN(2,3)=-CNN(3,3)	SURF 1ON	817
5871	818	CNN(2,2)=-(1.0+THP)*CNN(3,3)+(DZDN-DZM)*RPDZ	SURF 1ON	818
5872	819	CNN(2,1)=-THP*CNN(3,3)	SURF 1ON	819
5873	820 C	CNN(1,3)=0.0	SURF 1ON	820
5874	821 C	CNN(1,2)=-DZDN*RDZS/DZM	SURF 1ON	821
5875	822 C	CNN(1,1)=0.0	SURF 1ON	822
5876	823	Z DERIVATIVE ON S FACE (FOR DHNS)	SURF 1ON	823
5877	824	CNS(1,1)=-RPDZ*RDZS*RATTHP*(DZDS*(DZPA**2)+(DZPA+DZ)*(DZS**2))/(3. 1 O*DZM)	SURF 1ON	824
5878	825	CNS(1,2)=-(1.0+THM)*CNS(1,1)-(DZPA+DZ)*RDZS/DZM	SURF 1ON	825
5879	826	CNS(1,3)=THM*CNS(1,1)	SURF 1ON	826
5880	827	CNS(2,1)=-CNS(1,1)	SURF 1ON	827
5881	828	CNS(2,2)=-(1.0+THM)*CNS(1,1)-(DZDS-DZP)*RPDZ	SURF 1ON	828
5882	829	CNS(2,1)=-THP*CNS(1,1)	SURF 1ON	829
5883	830	CNS(3,1)=0.0	SURF 1ON	830
5884	831	CNS(1,2)=-DZDN*RDZS/DZM	SURF 1ON	831
5885	832	CNS(1,1)=0.0	SURF 1ON	832
5886	833 C	INITIALIZE GRADIENTS FOR USE IN DO LOOPS	SURF 1ON	833
5887	834 C	DHEE=0.0	SURF 1ON	834
5888	835 C	DHEW=0.0	SURF 1ON	835
5889	836	DHEN=0.0	SURF 1ON	836
5890	837	DHES=0.0	SURF 1ON	837
5891	838	DHNE=0.0	SURF 1ON	838
5892	839	DHNN=0.0	SURF 1ON	839
5893	840	DHNS=0.0	SURF 1ON	840
5894	841	RRSURF=1.0/RSURF	SURF 1ON	841
5895	842	RRSURFE=1.0/RSURFE	SURF 1ON	842
5896	843	RRSURFW=1.0/RSURFW	SURF 1ON	843
5897	844	NESTED DO LOOPS FOR CONSTRUCTION OF GRADIENTS	SURF 1ON	844
5898	845	SIX OR SEVEN POINT FORMULAS FOR GRADIENTS IMPOSED BY	SURF 1ON	845
5899	846 C	SURF 1ON	846	
5900	847 C	SURF 1ON	847	
5901	848 C	SURF 1ON	848	
5902	849 C	SURF 1ON	849	
5903	850	SURF 1ON	850	
5904	851	SURF 1ON	851	
5905	852	SURF 1ON	852	
5906	853 C	SURF 1ON	853	
5907	854 G	SURF 1ON	854	
5908	855 C	SURF 1ON	855	
5909	856	SURF 1ON	856	
5910	857	SURF 1ON	857	
5911	858	SURF 1ON	858	
5912	859	SURF 1ON	859	
5913	860	SURF 1ON	860	
5914	861	SURF 1ON	861	
5915	862	SURF 1ON	862	
5916	863	SURF 1ON	863	
5917	864 C	SURF 1ON	864	
5918	865 C	SURF 1ON	865	
5919	866 C	SURF 1ON	866	
5920	867 C	SURF 1ON	867	

5921	868	C	NINE POINT SUMMATIONS (3 OR 2 TERMS HAVE BEEN SET TO ZERO)	SURF 1ON	868
5922	869	C	DO 780 KK=1,3	SURF 1ON	869
5923	870		DO 770 JJ=1,3	SURF 1ON	870
5924	871		DHNE=DHNE+ANE(KK,JJ)*HX(KK,JJ)	SURF 1ON	871
5925	872		DHNW=DHNW+ANW(KK,JJ)*HX(KK,JJ)	SURF 1ON	872
5926	873		DHEN=DHEN+AEN(KK,JJ)*HX(KK,JJ)*RRSURF	SURF 1ON	873
5927	874		DHES=DHES+AES(KK,JJ)*HX(KK,JJ)*RRSURF	SURF 1ON	874
5928	875		DHEE=DHEE+CEE(KK,JJ)*HX(KK,JJ)*RRSURFE	SURF 1ON	875
5929	876		DHEW=DHEW+C EW(KK,JJ)*HX(KK,JJ)*RRSURFW	SURF 1ON	876
5930	877		DHNN=DHNN+CNN(KK,JJ)*HX(KK,JJ)	SURF 1ON	877
5931	878		DHNS=DHNS+C NS(KK,JJ)*HX(KK,JJ)	SURF 1ON	878
5932	879		770 CONTINUE	SURF 1ON	879
5933	880		780 CONTINUE	SURF 1ON	880
5934	881		GO TO 990	SURF 1ON	881
5935	882		790 CONTINUE	SURF 1ON	882
5936	883		RDEW=RDX(I)	SURF 1ON	883
5937	884		RDNS=RDZ(K)	SURF 1ON	884
5938	885		PHINW=0.25*(HY(1,3)+HY(2,3)+HY(1,2)+HY(2,2))	SURF 1ON	885
5939	886		PHINE=0.25*(HY(2,3)+HY(3,3)+HY(2,2)+HY(3,2))	SURF 1ON	886
5940	887		PHISW=0.25*(HY(1,2)+HY(2,2)+HY(1,1)+HY(2,1))	SURF 1ON	887
5941	888		PHISE=0.25*(HY(2,2)+HY(3,2)+HY(2,1)+HY(3,1))	SURF 1ON	888
5942	889		PHIAVE=0.5*(PHINW+PHISE)	SURF 1ON	889
5943	890		PHIAVW=0.5*(PHINE+PHISW)	SURF 1ON	890
5944	891		PHIAVN=0.5*(PHINW+PHINE)	SURF 1ON	891
5945	892		PHIAVS=0.5*(PHISW+PHISE)	SURF 1ON	892
5946	893		IF (J.EQ.2) PHIM=(YU(2)+DELY(1))/X(IM1)	SURF 1ON	893
5947	894		IF (J.NE.2) PHIM=(YU(J)-Y(U-2))/X(IM1)	SURF 1ON	894
5948	895		5949 896 C ***** Y=2 SPECIAL CASE *****	SURF 1ON	895
5950	897		DELE=PHIAVE-PHIM	SURF 1ON	896
5951	898		DELW=PHIAVW-PHIM	SURF 1ON	897
5952	899		IF (I.EQ.2) DELW=0.0	SURF 1ON	898
5953	900		DELN=PHIAVN-PHIM	SURF 1ON	899
5954	901		DELS=PHIAVS-PHIM	SURF 1ON	900
5955	902		ACOSE=COS(DELE)	SURF 1ON	901
5956	903		ASINE=SIN(DELE)	SURF 1ON	902
5957	904		ACOSW=COS(DELW)	SURF 1ON	903
5958	905		ASINW=SIN(DELW)	SURF 1ON	904
5959	906		ACOSN=COS(DELN)	SURF 1ON	905
5960	907		ASINN=SIN(DELN)	SURF 1ON	906
5961	908		ACOSS=COS(DELS)	SURF 1ON	907
5962	909		ASINS=SIN(DELS)	SURF 1ON	908
5963	910		A FE=AR(IJK)	SURF 1ON	909
5964	911		AFW=AR(IMJK)	SURF 1ON	910
5965	912		IF (CYL.GT.0.5.AND.I.EQ.2) AFW=1.0	SURF 1ON	911
5966	913		AFN=AT(IJK)	SURF 1ON	912
5967	914		AFS=AT(IJKM)	SURF 1ON	913
5968	915		AFLLOOR=ABK(IJKM)	SURF 1ON	914
5969	916		IF (NFF.EQ.4) AFLLOOR=ABK(IJK)	SURF 1ON	915
5970	917		IF (NOWALL.EQ.1) AFLLOOR=1.0	SURF 1ON	916
5971	918		IF (CYL.EQ.1.0.AND.AFLLOOR.LT.EM6) AFLLOOR=1.0	SURF 1ON	917
5972	919		IF (A FE.GT.EM6.AND.F(IPJK).LT.EMF) FLGE=AFLLOOR	SURF 1ON	918
5973	920		IF (AFW.GT.EM6.AND.F(IMJK).LT.EMF) FLGW=AFLLOOR	SURF 1ON	919
5974	921		IF (AFN.GT.EM6.AND.F(IJKP).LT.EMF) FLGN=AFLLOOR	SURF 1ON	920
5975	922		IF (AFS.GT.EM6.AND.F(IJKM).LT.EMF) FLGS=AFLLOOR	SURF 1ON	921
5976	923	C	BUILDING FOUR SETS OF SIX POINT COEFFICIENTS	SURF 1ON	922
5977	924	C	COORDINATES ARE SUCH THAT (E,W) GOES WITH (I+1,I-1)	SURF 1ON	923
5978	925	C	COORDINATES ARE SUCH THAT (N,S) GOES WITH (K+1,K-1)	SURF 1ON	924
5979	926	C	Z DERIVATIVE ON E FACE (FOR DHNE)	SURF 1ON	925
5980	927	C	ANE(3,3)=(1.0+3.0*FRRP*RATZM)/(3.0*DZP)	SURF 1ON	926
5981	928	C	ANE(2,3)=-(1.0+ZP)*ANE(3,3)+(FRRP/DZM)	SURF 1ON	927
5982	929	C	ANE(1,3)=ZP*ANE(3,3)-(FRRP/DZM)	SURF 1ON	928
5983	930	C	ANE(3,2)=-ANE(3,3)+ZM*RDZS	SURF 1ON	929
5984	931		ANE(2,2)=(1.0+ZP)*ANE(3,3)-(FRRP/DZM)+DZD*RPDZ	SURF 1ON	930
5985	932		ANE(1,2)=-ZP*ANE(3,3)+(FRRP/DZM)-ZP*RDZS	SURF 1ON	931
5986	933		ANE(3,1)=O.O	SURF 1ON	932
5987	934		ANE(2,1)=O.O	SURF 1ON	933
5988	935		ANE(1,1)=O.O	SURF 1ON	934
5989	936		IF (I.NE.2) GO TO 800	SURF 1ON	935
5990	937		ANE(3,1)=ANE(3,2)	SURF 1ON	936
5991	938		ANE(2,1)=ANE(2,2)	SURF 1ON	937
5992	939		ANE(1,1)=ANE(1,2)	SURF 1ON	938
5993	940		ANE(3,2)=ANE(3,3)	SURF 1ON	939
5994	941		ANE(2,2)=ANE(2,3)	SURF 1ON	940
5995	942		ANE(1,2)=ANE(1,3)	SURF 1ON	941
5996	943		ANE(3,3)=O.O	SURF 1ON	942
5997	944		ANE(2,1)=ANE(2,0)	SURF 1ON	943
5998	945		ANE(1,1)=ANE(1,0)	SURF 1ON	944
5999	946		ANE(3,2)=ANE(3,0)	SURF 1ON	945
6000	947		ANE(2,2)=ANE(2,0)	SURF 1ON	946
			ANE(1,2)=ANE(1,0)	SURF 1ON	947

6001	948	ANE(2,3)=0.0	SURF 1ON	948
6002	949	ANE(1,3)=0.0	SURF 1ON	949
6003	950	800 CONTINUE	SURF 1ON	950
6004	951 C	Z DERIVATIVE ON W FACE (FOR DHNW)	SURF 1ON	951
6006	953 C	ANW(1,1)=-(1.0+3.0*FRRM*RATZP)/(3.0*DZM)	SURF 1ON	953
6007	954	ANW(2,1)=-(1.0+ZM)*ANW(1,1)-(FRRM/DZP)	SURF 1ON	954
6008	955	ANW(3,1)=ZM*ANW(1,1)+(FRRM/DZP)	SURF 1ON	955
6009	956	ANW(1,2)=-ANW(1,1)-ZP*RDZS	SURF 1ON	956
6010	957	ANW(2,2)=(1.0+ZM)*ANW(1,1)+(FRRM/DZP)+DZD*RPDZ	SURF 1ON	957
6012	959	ANW(3,2)=-ZM*ANW(1,1)-(FRRM/DZP)+ZM*RDZS	SURF 1ON	959
6013	960	ANW(1,3)=0.0	SURF 1ON	960
6014	961	ANW(2,3)=0.0	SURF 1ON	961
6015	962	ANW(3,3)=0.0	SURF 1ON	962
6016	963 C	R DERIVATIVE ON N FACE (FOR DHEN)	SURF 1ON	963
6017	964 C	AEN(3,3)=(1.0+3.0*FRZP*RATRM)/(3.0*DRP)	SURF 1ON	964
6018	965 C	IF (I.EQ.2) AEN(3,3)=-(1.0+RM+3.0*FRZP*RATRM)/(3.0*DRP)	SURF 1ON	965
6020	967	AEN(3,2)=-(1.0+RP)*AEN(3,3)+(FRZP/DRM)	SURF 1ON	967
6021	968	AEN(3,1)=RP*AEN(3,3)-(FRZP/DRM)	SURF 1ON	968
6022	969	AEN(2,3)=-AEN(3,3)+RM*RDRS	SURF 1ON	969
6023	970	AEN(2,2)=(1.0+RP)*AEN(3,3)-(FRZP/DRM)+DRD*RPDR	SURF 1ON	970
6024	971	AEN(2,1)=-RP*AEN(3,3)+(FRZP/DRM)-RP*RDRS	SURF 1ON	971
6026	973	AEN(1,3)=0.0	SURF 1ON	973
6027	974	AEN(1,2)=0.0	SURF 1ON	974
6028	975	AEN(1,1)=0.0	SURF 1ON	975
6029	976	IF (I.NE.2) GO TO 810	SURF 1ON	976
6030	977	AEN(2,3)=AEN(2,3)-2.0*RM*RDRS	SURF 1ON	977
6031	978	AEN(2,2)=AEN(2,2)+(2.0/DRP)	SURF 1ON	978
6032	979	AEN(2,1)=AEN(2,1)-(2.0*RDRS)	SURF 1ON	979
6033	980	810 CONTINUE	SURF 1ON	980
6034	981 C	R DERIVATIVE ON S FACE (FOR DHES)	SURF 1ON	981
6035	982 C	AES(1,1)=-(1.0+3.0*FRZM*RATRP)/(3.0*DRM)	SURF 1ON	982
6036	983 C	AES(1,2)=-(1.0+RM)*AES(1,1)-(FRZM/DRP)	SURF 1ON	983
6037	984	AES(1,3)=RM*AES(1,1)+(FRZM/DRP)	SURF 1ON	984
6038	985	AES(2,1)=-AES(1,1)-RP*RDRS	SURF 1ON	985
6039	986	AES(2,2)=(1.0+RM)*AES(1,1)+(FRZM/DRP)+DRD*RPDR	SURF 1ON	986
6040	987	AES(2,3)=-RM*AES(1,1)-(FRZM/DRP)+RM*RDRS	SURF 1ON	987
6041	988	AES(3,1)=0.0	SURF 1ON	988
6042	989	AES(3,2)=0.0	SURF 1ON	989
6043	990	AES(3,3)=0.0	SURF 1ON	990
6044	991	IF (I.NE.2) GO TO 820	SURF 1ON	991
6045	992	AES(1,3)=(1.0+RM-3.0*FRZM*RATRM)/(3.0*DRP)	SURF 1ON	992
6046	993	AES(1,2)=-(1.0+RP)*AES(1,3)+(FRZM/DRM)	SURF 1ON	993
6047	994	AES(1,1)=RP*AES(1,3)-(FRZM/DRM)	SURF 1ON	994
6048	995	AES(2,3)=-AES(1,3)-(RM*RDRS)	SURF 1ON	995
6049	996	AES(2,2)=(1.0+RP)*AES(1,3)-(FRZM/DRM)+(1.0/DRP)+(1.0/DRM)	SURF 1ON	996
6050	997	AES(2,1)=-RP*AES(1,3)+(FRZM/DRM)-(1.0/DRM)-(RDRS)	SURF 1ON	997
6053	1000	820 CONTINUE	SURF 1ON	999
6054	1001 C	BUILDING FOUR SETS OF SEVEN POINT COEFFICIENTS	SURF 1ON	1000
6055	1002 C	R DERIVATIVE ON E FACE (FOR DHEE)	SURF 1ON	1001
6056	1003 C	CEE(3,3)=RPDR*RDRS*RATZM*(DRDO*(DRMA**2)+(DRMA+DR)*(DRO**2))/(3.0*DRP)	SURF 1ON	1002
6057	1004 C	CEE(2,3)=-(1.0+ZP)*CEE(3,3)+(DRMA+DR)*RDRS/DRP	SURF 1ON	1003
6058	1005 C	CEE(1,3)=ZP*CEE(3,3)	SURF 1ON	1004
6059	1006	CEE(3,2)=-CEE(3,3)	SURF 1ON	1005
6060	1007	CEE(2,2)=(1.0+ZP)*CEE(3,3)+(DRDO-DRM)*RPDR	SURF 1ON	1006
6061	1008	CEE(1,2)=-ZP*CEE(3,3)	SURF 1ON	1007
6062	1009	CEE(3,1)=O.0	SURF 1ON	1008
6063	1010	CEE(2,1)=-DRDO*RDRS/DRM	SURF 1ON	1009
6064	1011	CEE(1,1)=O.0	SURF 1ON	1010
6065	1012	IF (I.NE.2) GO TO 830	SURF 1ON	1011
6066	1013	CEE(1,1)=-ZP*CEE(3,3)	SURF 1ON	1012
6067	1014	CEE(2,1)=-DRDO*RDRS/DRM	SURF 1ON	1013
6068	1015	CEE(1,1)=O.0	SURF 1ON	1014
6069	1016	CEE(1,1)=-ZP*CEE(3,3)	SURF 1ON	1015
6070	1017	CEE(2,1)=-(1.0+ZM)*CEE(1,1)-(DRPA+DR)*RDRS/DRM	SURF 1ON	1016
6071	1018	CEE(3,1)=ZM*CEE(1,1)	SURF 1ON	1017
6072	1019	CEE(1,2)=-CEE(1,1)	SURF 1ON	1018
6073	1020	CEE(2,2)=(1.0+ZM)*CEE(1,1)-(DRDI-DRP)*RPDR	SURF 1ON	1019
6074	1021	CEE(3,2)=-ZM*CEE(1,1)	SURF 1ON	1020
6075	1022	CEE(1,3)=O.0	SURF 1ON	1021
6076	1023	CEE(2,3)=DRDI*RDRS/DRP	SURF 1ON	1022
6077	1024	CEE(3,3)=O.0	SURF 1ON	1023
6078	1025	830 CONTINUE	SURF 1ON	1024
6079	1026	CEE(1,1)=O.0	SURF 1ON	1025
6080	1027 C	SURF 1ON	1026	

```

6081 1028 C R DERIVATIVE ON W FACE (FOR DHEW) SURF 1ON 1028
6082 1029 C CEW(1,1)=-RPDR*RDRS*RATZP*(DRDI*(DRPA**2)+(DRPA+DR)*(DRI**2))/(3.0 SURF 1ON 1029
6083 1030 1 *DRM) SURF 1ON 1030
6084 1031 CEW(2,1)=-(1.0+ZM)*CEW(1,1)-(DRPA+DR)*RDRS/DRM SURF 1ON 1031
6085 1032 CEW(3,1)=ZM*CEW(1,1) SURF 1ON 1032
6086 1033 CEW(1,2)=-CEW(1,1) SURF 1ON 1033
6087 1034 CEW(2,2)=(1.0+ZM)*CEW(1,1)-(DRDI-DRP)*RPDR SURF 1ON 1034
6088 1035 CEW(3,2)=-ZM*CEW(1,1) SURF 1ON 1035
6089 1036 CEW(1,3)=O.O SURF 1ON 1036
6090 1037 CEW(2,3)=DRDI*RDRS/DRP SURF 1ON 1037
6091 1038 CEW(3,3)=O.O SURF 1ON 1038
6092 1039 C Z DERIVATIVE ON N FACE (FOR DHNN) SURF 1ON 1039
6093 1040 C 6094 1041 C SURF 1ON 1040
6095 1042 C 6096 1043 CNN(3,3)=RPDZ*RDZS*RATRM*(DZDN*(DZMA**2)+(DZMA+DZ)*(DZN**2))/(3.0 SURF 1ON 1043
6097 1044 1 *DZP) SURF 1ON 1044
6098 1045 CNN(3,2)=-(1.0+RP)*CNN(3,3)+(DZMA+DZ)*RDZS/DZP SURF 1ON 1045
6099 1046 CNN(3,1)=RP*CNN(3,3) SURF 1ON 1046
6100 1047 CNN(2,3)=-CNN(3,3) SURF 1ON 1047
6101 1048 CNN(2,2)=(1.0+RP)*CNN(3,3)+(DZDN-DZM)*RPDZ SURF 1ON 1048
6102 1049 CNN(2,1)=-RP*CNN(3,3) SURF 1ON 1049
6103 1050 CNN(1,3)=O.O SURF 1ON 1050
6104 1051 CNN(1,2)=-DZDN*RDZS/DZM SURF 1ON 1051
6105 1052 CNN(1,1)=O.O SURF 1ON 1052
6106 1053 IF (I.NE.2) GO TO 840 SURF 1ON 1053
6107 1054 CNN(3,2)=-(1.0+RP)*CNN(3,3) SURF 1ON 1054
6108 1055 CNN(3,1)=RP*CNN(3,3)+(1.0/DZP)-DZDN*RDZS/DZP SURF 1ON 1055
6109 1056 CNN(2,3)=-CNN(3,3) SURF 1ON 1056
6110 1057 CNN(2,2)=(1.0+RP)*CNN(3,3) SURF 1ON 1057
6111 1058 CNN(2,1)=-RP*CNN(3,3)+(DZDN-DZM)*RPDZ SURF 1ON 1058
6112 1059 CNN(1,2)=O.O SURF 1ON 1059
6113 1060 CNN(1,1)=-DZDN*RDZS/DZM SURF 1ON 1060
6114 1061 840 CONTINUE SURF 1ON 1061
6115 1062 C 6116 1063 C Z DERIVATIVE ON S FACE (FOR DHNS) SURF 1ON 1062
6117 1064 C 6118 1065 CNS(1,1)=-RPDZ*RDZS*RATRP*(DZDS*(DZPA**2)+(DZPA+DZ)*(DZS**2))/(3.0 SURF 1ON 1065
6119 1066 1 *DZM) SURF 1ON 1066
6120 1067 CNS(1,2)=-(1.0+RM)*CNS(1,1)-(DZPA+DZ)*RDZS/DZM SURF 1ON 1067
6121 1068 CNS(1,3)=RM*CNS(1,1) SURF 1ON 1068
6122 1069 CNS(2,1)=-CNS(1,1) SURF 1ON 1069
6123 1070 CNS(2,2)=(1.0+RM)*CNS(1,1)-(DZDS-DZP)*RPDZ SURF 1ON 1070
6124 1071 CNS(2,3)=-RM*CNS(1,1) SURF 1ON 1071
6125 1072 CNS(3,1)=O.O SURF 1ON 1072
6126 1073 CNS(3,2)=DZDS*RDZS/DZP SURF 1ON 1073
6127 1074 CNS(3,3)=O.O SURF 1ON 1074
6128 1075 IF (I.NE.2) GO TO 850 SURF 1ON 1075
6129 1076 CNS(1,3)=RM*CNS(1,1) SURF 1ON 1076
6130 1077 CNS(1,2)=-(1.0+RP)*CNS(1,3) SURF 1ON 1077
6131 1078 CNS(1,1)=RP*CNS(1,3)-(1.0/DZM)+DZDS*RDZS/DZM SURF 1ON 1078
6132 1079 CNS(2,3)=-CNS(1,3) SURF 1ON 1079
6133 1080 CNS(2,2)=(1.0+RP)*CNS(1,3) SURF 1ON 1080
6134 1081 CNS(2,1)=-RP*CNS(1,3)-(DZDS-DZP)*RPDZ SURF 1ON 1081
6135 1082 CNS(3,2)=O.O SURF 1ON 1082
6136 1083 CNS(3,1)=DZDS*RDZS/DZP SURF 1ON 1083
6137 1084 850 CONTINUE SURF 1ON 1084
6138 1085 C 6139 1086 C INITIALIZE GRADIENTS FOR USE IN DO LOOPS SURF 1ON 1085
6140 1087 C 6141 1088 DHEE=O.O SURF 1ON 1087
6142 1089 DHEW=O.O SURF 1ON 1088
6143 1090 DHEN=O.O SURF 1ON 1089
6144 1091 DHES=O.O SURF 1ON 1090
6145 1092 DHNE=O.O SURF 1ON 1091
6146 1093 DHNW=O.O SURF 1ON 1092
6147 1094 DHNN=O.O SURF 1ON 1093
6148 1095 DHNS=O.O SURF 1ON 1094
6149 1096 C 6150 1097 C NESTED DO LOOPS FOR CONSTRUCTION OF GRADIENTS SURF 1ON 1095
6151 1098 C 6152 1099 C MULTIPLY GRADIENTS BY APPROPRIATE RADIUS ON EACH FACE SURF 1ON 1096
6153 1100 C TO OBTAIN VARIABLES WITH DIMENSIONS OF ANGLES SURF 1ON 1097
6154 1101 C 6155 1102 C SIX OR SEVEN POINT FORMULAS FOR GRADIENTS IMPOSED BY SURF 1ON 1098
6156 1103 C NINE POINT SUMMATIONS (3 OR 2 TERMS HAVE BEEN SET TO ZERO) SURF 1ON 1099
6157 1104 C 6158 1105 DO 870 KK=1,3 SURF 1ON 1100
6159 1106 DO 860 II=1,3 SURF 1ON 1101
6160 1107 DHNE=DHNE+ANE(KK,II)*HY(II,KK)*X(I) SURF 1ON 1102
6161 1108

```

```

6161 1108 DHNW=DHNW+ANW(KK,II)*HY(II,KK)*X(I-1) SURF 1ON 1108
6162 1109 DHEN=DHEN+AEN(KK,II)*HY(II,KK)*XI(I) SURF 1ON 1109
6163 1110 DHEs=DHES+AES(KK,II)*HY(II,KK)*XI(I) SURF 1ON 1110
6164 1111 DHEE=DHEE+CEE(KK,II)*HY(II,KK)*X(I) SURF 1ON 1111
6165 1112 DHEW=DHEW+C EW(KK,II)*HY(II,KK)*X(I-1) SURF 1ON 1112
6166 1113 DHNN=DHNN+CNN(KK,II)*HY(II,KK)*XI(I) SURF 1ON 1113
6167 1114 DHNS=DHNS+CNS(KK,II)*HY(II,KK)*XI(I) SURF 1ON 1114
6168 1115 860 CONTINUE SURF 1ON 1115
6169 1116 870 CONTINUE SURF 1ON 1116
6170 1117 GO TO 990 SURF 1ON 1117
6171 1118 880 CONTINUE SURF 1ON 1118
6172 1119 RDEW=RDX(I) SURF 1ON 1119
6173 1120 RDNS=RDY(J)*RRI(I) SURF 1ON 1120
6174 1121 AFE=AR(IJK) SURF 1ON 1121
6175 1122 AFW=AR(IMJK) SURF 1ON 1122
6176 1123 IF (I.EQ.2.AND.CYL.EQ.1.0.AND.X(1).EQ.0.0) AFW=1.0 SURF 1ON 1123
6177 1124 AFN=ABK(IJK) SURF 1ON 1124
6178 1125 IF (J.EQ.JM1.AND.CYL.EQ.1.0.AND.JC2PI.EQ.0) AFN=1.0 SURF 1ON 1125
6179 1126 AFS=ABK(IUMK) SURF 1ON 1126
6180 1127 IF (J.EQ.2.AND.CYL.EQ.1.0.AND.JC2PI.EQ.0) AFS=1.0 SURF 1ON 1127
6181 1128 AFLLOOR=AT(IJKM) SURF 1ON 1128
6182 1129 IF (NFF.EQ.6) AFLLOOR=AT(IJK) SURF 1ON 1129
6183 1130 IF (NOWALL.EQ.1) AFLLOOR=1.0 SURF 1ON 1130
6184 1131 IF (AFE.GT.EM6.AND.F(IPUK).LT.EMF) FLGE=AFLLOOR SURF 1ON 1131
6185 1132 IF (AFW.GT.EM6.AND.F(IMUK).LT.EMF) FLGW=AFLLOOR SURF 1ON 1132
6186 1133 IF (AFN.GT.EM6.AND.F(IUPK).LT.EMF) FLGN=AFLLOOR SURF 1ON 1133
6187 1134 IF (AFS.GT.EM6.AND.F(IUMK).LT.EMF) FLGS=AFLLOOR SURF 1ON 1134
6188 1135 IF (CYL.GT.0.5) FLGE=FLGE*X(I)*RXI(I) SURF 1ON 1135
6189 1136 IF (CYL.GT.0.5) FLGW=FLGW*X(I-1)*RXI(I) SURF 1ON 1136
6190 1137 C
6191 1138 C
6192 1139 C
6193 1140 C
6194 1141 C
6195 1142 C
6196 1143 C
6197 1144 C
6198 1145
6199 1146
6200 1147
6201 1148
6202 1149
6203 1150
6204 1151
6205 1152
6206 1153
6207 1154
6208 1155
6209 1156
6210 1157
6211 1158
6212 1159
6213 1160
6214 1161
6215 1162
6216 1163
6217 1164
6218 1165 C
6219 1166 C
6220 1167 C
6221 1168
6222 1169
6223 1170
6224 1171
6225 1172
6226 1173
6227 1174
6228 1175
6229 1176
6230 1177 C
6231 1178 C
6232 1179 C
6233 1180
6234 1181
6235 1182
6236 1183
6237 1184
6238 1185
6239 1186
6240 1187
     890 CONTINUE
     BUILDING FOUR SETS OF SIX POINT COEFFICIENTS
COORDINATES ARE SUCH THAT (E,W) GOES WITH (I+1,I-1)
COORDINATES ARE SUCH THAT (N,S) GOES WITH (J+1,J-1)
THETA DERIVATIVE ON E FACE (FOR DHNE)
ANE(3,3)=(1.0+3.0*FRRP*RATTHM)/(3.0*DTHP)
ANE(2,3)=-(1.0+THP)*ANE(3,3)+(FRRP/DTHM)
ANE(1,3)=THP*ANE(3,3)-(FRRP/DTHM)
ANE(3,2)=-ANE(3,3)+THM*RDTHS
ANE(2,2)=(1.0+THP)*ANE(3,3)-(FRRP/DTHM)+DTHD*RPDT
ANE(1,2)=-THP*ANE(3,3)+(FRRP/DTHM)-THP*RDTHS
ANE(3,1)=0.0
ANE(2,1)=0.0
ANE(1,1)=0.0
IF (I.NE.2) GO TO 890
ANE(3,1)=ANE(3,2)
ANE(2,1)=ANE(2,2)
ANE(1,1)=ANE(1,2)
ANE(3,2)=ANE(3,3)
ANE(2,2)=ANE(2,3)
ANE(1,2)=ANE(1,3)
ANE(3,3)=0.0
ANE(2,3)=0.0
ANE(1,3)=0.0
890 CONTINUE
THETA DERIVATIVE ON W FACE (FOR DHNW)
ANW(1,1)=-(1.0+3.0*FRRM*RATTHP)/(3.0*DTHM)
ANW(2,1)=-(1.0+THM)*ANW(1,1)-(FRRM/DTHP)
ANW(3,1)=THM*ANW(1,1)+(FRRM/DTHP)
ANW(1,2)=-ANW(1,1)-THP*RDTHS
ANW(2,2)=(1.0+THM)*ANW(1,1)+(FRRM/DTHP)+DTHD*RPDT
ANW(3,2)=-THM*ANW(1,1)-(FRRM/DTHP)+THM*RDTHS
ANW(1,3)=0.0
ANW(2,3)=0.0
ANW(3,3)=0.0
R DERIVATIVE ON N FACE (FOR DHEN)
AEN(3,3)=(1.0+3.0*FRTHP*RATRM)/(3.0*DRP)
IF (I.EQ.2) AEN(3,3)=-(1.0+RM+3.0*FRTHP*RATRM)/(3.0*DRP)
AEN(3,2)=-(1.0+RP)*AEN(3,3)+(FRTHP/DRM)
AEN(3,1)=RP*AEN(3,3)-(FRTHP/DRM)
AEN(2,3)=-AEN(3,3)+RM*RDRS
AEN(2,2)=(1.0+RP)*AEN(3,3)-(FRTHP/DRM)+DRD*RPDR
AEN(2,1)=-RP*AEN(3,3)+(FRTHP/DRM)-RP*RDRS
AEN(1,3)=0.0

```

```

6241 1188 AEN(1,2)=O.O SURF 1ON 1188
6242 1189 AEN(1,1)=O.O SURF 1ON 1189
6243 1190 IF (I.NE.2) GO TO 900 SURF 1ON 1190
6244 1191 AEN(2,3)=AEN(2,3)-2.0*RM*RDRS SURF 1ON 1191
6245 1192 AEN(2,2)=AEN(2,2)+(2.0/DRP) SURF 1ON 1192
6246 1193 AEN(2,1)=AEN(2,1)-(2.0*RDRS) SURF 1ON 1193
6247 1194 900 CONTINUE SURF 1ON 1194
6248 1195 C SURF 1ON 1195
6249 1196 C R DERIVATIVE ON S FACE (FOR DHES) SURF 1ON 1196
6250 1197 C SURF 1ON 1197
6251 1198 AES(1,1)=-(1.0+3.0*FRTHM*RATRP)/(3.0*DRM) SURF 1ON 1198
6252 1199 AES(1,2)=-(1.0+RM)*AES(1,1)-(FRTHM/DRP) SURF 1ON 1199
6253 1200 AES(1,3)=RM*AES(1,1)+(FRTHM/DRP) SURF 1ON 1200
6254 1201 AES(2,1)=-AES(1,1)-RP*RDRS SURF 1ON 1201
6255 1202 AES(2,2)=(1.0+RM)*AES(1,1)+(FRTHM/DRP)+DRD*RPDR SURF 1ON 1202
6256 1203 AES(2,3)=-RM*AES(1,1)-(FRTHM/DRP)+RM*RDRS SURF 1ON 1203
6257 1204 AES(3,1)=O.O SURF 1ON 1204
6258 1205 AES(3,2)=O.O SURF 1ON 1205
6259 1206 AES(3,3)=O.O SURF 1ON 1206
6260 1207 IF (I.NE.2) GO TO 910 SURF 1ON 1207
6261 1208 AES(1,3)=(1.0+RM-3.0*FRTHM*RATRM)/(3.0*DRP) SURF 1ON 1208
6262 1209 AES(1,2)=-(1.0+RP)*AES(1,3)+(FRTHM/DRM) SURF 1ON 1209
6263 1210 AES(1,1)=RP*AES(1,3)-(FRTHM/DRM) SURF 1ON 1210
6264 1211 AES(2,3)=-AES(1,3)-(RM*RDRS) SURF 1ON 1211
6265 1212 AES(2,2)=(1.0+RP)*AES(1,3)-(FRTHM/DRM)+(1.0/DRP)+(1.0/DRM) SURF 1ON 1212
6266 1213 AES(2,1)=-RP*AES(1,3)+(FRTHM/DRM)-(1.0/DRM)-(RDRS) SURF 1ON 1213
6267 1214 910 CONTINUE SURF 1ON 1214
6268 1215 C SURF 1ON 1215
6269 1216 C BUILDING FOUR SETS OF SEVEN POINT COEFFICIENTS SURF 1ON 1216
6270 1217 C SURF 1ON 1217
6271 1218 C R DERIVATIVE ON E FACE (FOR DHEE) SURF 1ON 1218
6272 1219 C SURF 1ON 1219
6273 1220 CEE(3,3)=RPDR*RDRS*RATTHM*(DRDO*(DRMA**2)+(DRMA+DR)*(DR0**2))/(3.0 SURF 1ON 1220
6274 1221 1*DRP) SURF 1ON 1221
6275 1222 CEE(2,3)=-(1.0+THP)*CEE(3,3)+(DRMA+DR)*RDRS/DRP SURF 1ON 1222
6276 1223 CEE(1,3)=THP*CEE(3,3) SURF 1ON 1223
6277 1224 CEE(3,2)=-CEE(3,3) SURF 1ON 1224
6278 1225 CEE(2,2)=(1.0+THP)*CEE(3,3)+(DRDO-DRM)*RPDR SURF 1ON 1225
6279 1226 CEE(1,2)=-THP*CEE(3,3) SURF 1ON 1226
6280 1227 CEE(3,1)=O.O SURF 1ON 1227
6281 1228 CEE(2,1)=-DRDO*RDRS/DRM SURF 1ON 1228
6282 1229 CEE(1,1)=O.O SURF 1ON 1229
6283 1230 IF (I.NE.2) GO TO 920 SURF 1ON 1230
6284 1231 CEE(1,1)=-THP*CEE(3,3) SURF 1ON 1231
6285 1232 CEE(2,1)=-(1.0+THM)*CEE(1,1)-(DRPA+DR)*RDRS/DRM SURF 1ON 1232
6286 1233 CEE(3,1)=THM*CEE(1,1) SURF 1ON 1233
6287 1234 CEE(1,2)=-CEE(1,1) SURF 1ON 1234
6288 1235 CEE(2,2)=(1.0+THM)*CEE(1,1)-(DRDI-DRP)*RPDR SURF 1ON 1235
6289 1236 CEE(3,2)=-THM*CEE(1,1) SURF 1ON 1236
6290 1237 CEE(1,3)=O.O SURF 1ON 1237
6291 1238 CEE(2,3)=DRDI*RDRS/DRP SURF 1ON 1238
6292 1239 CEE(3,3)=O.O SURF 1ON 1239
6293 1240 920 CONTINUE SURF 1ON 1240
6294 1241 C SURF 1ON 1241
6295 1242 C R DERIVATIVE ON W FACE (FOR DHEW) SURF 1ON 1242
6296 1243 C SURF 1ON 1243
6297 1244 CEW(1,1)=-RPDR*RDRS*RATTHP*(DRDI*(DRPA**2)+(DRPA+DR)*(DRI**2))/(3. SURF 1ON 1244
6298 1245 1*O*DRM) SURF 1ON 1245
6299 1246 CEW(2,1)=-(1.0+THM)*CEW(1,1)-(DRPA+DR)*RDRS/DRM SURF 1ON 1246
6300 1247 CEW(3,1)=THM*CEW(1,1) SURF 1ON 1247
6301 1248 CEW(1,2)=-CEW(1,1) SURF 1ON 1248
6302 1249 CEW(2,2)=(1.0+THM)*CEW(1,1)-(DRDI-DRP)*RPDR SURF 1ON 1249
6303 1250 CEW(3,2)=-THM*CEW(1,1) SURF 1ON 1250
6304 1251 CEW(1,3)=O.O SURF 1ON 1251
6305 1252 CEW(2,3)=DRDI*RDRS/DRP SURF 1ON 1252
6306 1253 CEW(3,3)=O.O SURF 1ON 1253
6307 1254 C SURF 1ON 1254
6308 1255 C THETA DERIVATIVE ON N FACE (FOR DHNN) SURF 1ON 1255
6309 1256 C SURF 1ON 1256
6310 1257 CNN(3,3)=RPDTH*RDTHS*RATRM*(DTHDE*(DTHMA**2)+(DTHMA+DTH)*(DTHE**2) SURF 1ON 1257
6311 1258 1)/(3.0*DTHP) SURF 1ON 1258
6312 1259 CNN(3,2)=-(1.0+RP)*CNN(3,3)+(DTHMA+DTH)*RDTHS/DTHP SURF 1ON 1259
6313 1260 CNN(3,1)=RP*CNN(3,3) SURF 1ON 1260
6314 1261 CNN(2,3)=-CNN(3,3) SURF 1ON 1261
6315 1262 CNN(2,2)=(1.0+RP)*CNN(3,3)+(DTHDE-DTHM)*RPDTH SURF 1ON 1262
6316 1263 CNN(2,1)=-RP*CNN(3,3) SURF 1ON 1263
6317 1264 CNN(1,3)=O.O SURF 1ON 1264
6318 1265 CNN(1,2)=-DTHDE*RDTHS/DTHM SURF 1ON 1265
6319 1266 CNN(1,1)=O.O SURF 1ON 1266
6320 1267 IF (I.NE.2) GO TO 930 SURF 1ON 1267

```

```

6321 1268 CNN(3,2)=-(1.0+RP)*CNN(3,3) SURF 1 ON 1268
6322 1269 CNN(3,1)=RP*CNN(3,3)+(1.0/DTHP)-DTHDE*RDTHS/DTHP SURF 1 ON 1269
6323 1270 CNN(2,3)=-CNN(3,3) SURF 1 ON 1270
6324 1271 CNN(2,2)=(1.0+RP)*CNN(3,3) SURF 1 ON 1271
6325 1272 CNN(2,1)=-RP*CNN(3,3)+(DTHDE-DTHM)*RPDTH SURF 1 ON 1272
6326 1273 CNN(1,2)=0.0 SURF 1 ON 1273
6327 1274 CNN(1,1)=-DTHDE*RDTHS/DTHM SURF 1 ON 1274
6328 1275 930 CONTINUE SURF 1 ON 1275
6329 1276 C THETA DERIVATIVE ON S FACE (FOR DHNS) SURF 1 ON 1276
6330 1277 C 1277
6331 1278 C SURF 1 ON 1278
6332 1279 CNS(1,1)=-RPDTH*RDTHS*RATRP*(DTHDW*(DTHPA**2)+(DTHPA+DTH)*(DTHW**2 SURF 1 ON 1279
6333 1280 1 )/(3.0*DTHM)) SURF 1 ON 1280
6334 1281 CNS(1,2)=-(-1.0+RM)*CNS(1,1)-(DTHPA+DTH)*RDTHS/DTHM SURF 1 ON 1281
6335 1282 CNS(1,3)=RM*CNS(1,1) SURF 1 ON 1282
6336 1283 CNS(2,1)=-CNS(1,1) SURF 1 ON 1283
6337 1284 CNS(2,2)=(1.0+RM)*CNS(1,1)-(DTHDW-DTHP)*RPDTH SURF 1 ON 1284
6338 1285 CNS(2,3)=-RM*CNS(1,1) SURF 1 ON 1285
6339 1286 CNS(3,1)=0.0 SURF 1 ON 1286
6340 1287 CNS(3,2)=DTHDW*RDTHS/DTHP SURF 1 ON 1287
6341 1288 CNS(3,3)=0.0 SURF 1 ON 1288
6342 1289 IF (I.NE.2) GO TO 940 SURF 1 ON 1289
6343 1290 CNS(1,3)=RM*CNS(1,1) SURF 1 ON 1290
6344 1291 CNS(1,2)=-(-1.0+RP)*CNS(1,3) SURF 1 ON 1291
6345 1292 CNS(1,1)=RP*CNS(1,3)-(1.0/DTHM)+DTHDW*RDTHS/DTHM SURF 1 ON 1292
6346 1293 CNS(2,3)=-CNS(1,3) SURF 1 ON 1293
6347 1294 CNS(2,2)=(1.0+RP)*CNS(1,3) SURF 1 ON 1294
6348 1295 CNS(2,1)=-RP*CNS(1,3)-(DTHDW-DTHP)*RPDTH SURF 1 ON 1295
6349 1296 CNS(3,2)=0.0 SURF 1 ON 1296
6350 1297 CNS(3,1)=DTHDW*RDTHS/DTHP SURF 1 ON 1297
6351 1298 940 CONTINUE SURF 1 ON 1298
6352 1299 C SURF 1 ON 1299
6353 1300 C INITIALIZE GRADIENTS FOR USE IN DO LOOPS SURF 1 ON 1300
6354 1301 C SURF 1 ON 1301
6355 1302 DHEE=0.0 SURF 1 ON 1302
6356 1303 DHEW=0.0 SURF 1 ON 1303
6357 1304 DHEN=0.0 SURF 1 ON 1304
6358 1305 DHES=0.0 SURF 1 ON 1305
6359 1306 DHNE=0.0 SURF 1 ON 1306
6360 1307 DHNW=0.0 SURF 1 ON 1307
6361 1308 DHNN=0.0 SURF 1 ON 1308
6362 1309 DHNS=0.0 SURF 1 ON 1309
6363 1310 C SURF 1 ON 1310
6364 1311 C NESTED DO LOOPS FOR CONSTRUCTION OF GRADIENTS SURF 1 ON 1311
6365 1312 C SURF 1 ON 1312
6366 1313 C SIX OR SEVEN POINT FORMULAS FOR GRADIENTS IMPOSED BY SURF 1 ON 1313
6367 1314 C NINE POINT SUMMATIONS (3 OR 2 TERMS HAVE BEEN SET TO ZERO) SURF 1 ON 1314
6368 1315 C SURF 1 ON 1315
6369 1316 DO 980 JJ=1,3 SURF 1 ON 1316
6370 1317 DO 970 II=1,3 SURF 1 ON 1317
6371 1318 DHNE=DHNE+ANE(JJ,II)*HZ(II,JJ)*RX(I) SURF 1 ON 1318
6372 1319 IF (I.EQ.2) GO TO 950 SURF 1 ON 1319
6373 1320 DHNW=DHNW+ANW(JJ,II)*HZ(II,JJ)*RX(I-1) SURF 1 ON 1320
6374 1321 GO TO 960 SURF 1 ON 1321
6375 1322 950 CONTINUE SURF 1 ON 1322
6376 1323 DHNW=1.0 SURF 1 ON 1323
6377 1324 960 CONTINUE SURF 1 ON 1324
6378 1325 DHEN=DHEN+AEN(JJ,II)*HZ(II,JJ) SURF 1 ON 1325
6379 1326 DHES=DHES+AES(JJ,II)*HZ(II,JJ) SURF 1 ON 1326
6380 1327 DHEE=DHEE+CEE(JJ,II)*HZ(II,JJ) SURF 1 ON 1327
6381 1328 DHEW=DHEW+CEW(JJ,II)*HZ(II,JJ) SURF 1 ON 1328
6382 1329 DHNN=DHNN+CNN(JJ,II)*HZ(II,JJ)*RX(I) SURF 1 ON 1329
6383 1330 DHNS=DHNS+CNS(JJ,II)*HZ(II,JJ)*RX(I) SURF 1 ON 1330
6384 1331 970 CONTINUE SURF 1 ON 1331
6385 1332 980 CONTINUE SURF 1 ON 1332
6386 1333 990 CONTINUE SURF 1 ON 1333
6387 1334 IF (NOWALL.EQ.0) GO TO 1000 SURF 1 ON 1334
6388 1335 AFE=1.0 SURF 1 ON 1335
6389 1336 AFW=1.0 SURF 1 ON 1336
6390 1337 AFN=1.0 SURF 1 ON 1337
6391 1338 AFS=1.0 SURF 1 ON 1338
6392 1339 1000 CONTINUE SURF 1 ON 1339
6393 1340 TERM1=1.0+DHNE*DHNE SURF 1 ON 1340
6394 1341 TERM2=1.0+DHNW*DHNW SURF 1 ON 1341
6395 1342 RHDE=SQRT(TERM1+DHEE*DHEE) SURF 1 ON 1342
6396 1343 RHDW=SQRT(TERM2+DHEW*DHEW) SURF 1 ON 1343
6397 1344 IF (NFF.EQ.3.OR.NFF.EQ.4) GO TO 1020 SURF 1 ON 1344
6398 1345 FEW=FLGE*(AFE*DHEE/RHDE+(1.0-AFE)*CSANG)-FLGW*(AFW*DHEW/RHDW-(1.0 SURF 1 ON 1345
6399 1346 1 -AFW)*CSANG) SURF 1 ON 1346
6400 1347 C ADD CYL TERMS FOR NF=1 OR 2 SURF 1 ON 1347

```

```

6401 1348 IF (STH.NE.0.0) GO TO 1010 SURF 1ON 1348
6402 1349 GO TO 1030 SURF 1ON 1349
6403 1350 1010 CONTINUE SURF 1ON 1350
6404 1351 FEW=FEW*CTH-STH*(FLGE*(AFE*TERM1/RHDE+(1.0-AFE)*SANG)+FLGW*(AFW SURF 1ON 1351
6405 1352 1 *TERM2/RHDW+(1.0-AFW)*SANG)) SURF 1ON 1352
6406 1353 GO TO 1030 SURF 1ON 1353
6407 1354 1020 CONTINUE SURF 1ON 1354
6408 1355 T2RE=(1.0+DHNE**2)/RHDE SURF 1ON 1355
6409 1356 T2RW=(1.0+DHNW**2)/RHDW SURF 1ON 1356
6410 1357 FEW=FLGE*(AFE*(DHEE/RHDE*ACOSE+T2RE*ASINE)+(1.0-AFE)*CSANG)-FLGW* SURF 1ON 1357
6411 1358 1 (AFW*(DHEW/RHDW*ACOSW+T2RW*ASINW)-(1.0-AFW)*CSANG) SURF 1ON 1358
6412 1359 C SURF 1ON 1359
6413 1360 1030 CONTINUE SURF 1ON 1360
6414 1361 TERM1=1.0+DHEN*DHEN SURF 1ON 1361
6415 1362 TERM2=1.0+DHES*DHES SURF 1ON 1362
6416 1363 RHDN=SQRT(TERM1+DHNN*DHN) SURF 1ON 1363
6417 1364 RHDS=SQRT(TERM2+DHNS*DHN) SURF 1ON 1364
6418 1365 IF (NFF.EQ.3.OR.NFF.EQ.4) GO TO 1040 SURF 1ON 1365
6419 1366 FNS=FLGN*(AFN*DHN/RHDN+(1.0-AFN)*CSANG)-FLGS*(AFS*DHN/RHDS-(1.0 SURF 1ON 1366
6420 1367 1 -AFS)*CSANG) SURF 1ON 1367
6421 1368 GO TO 1050 SURF 1ON 1368
6422 1369 1040 CONTINUE SURF 1ON 1369
6423 1370 T2RN=-DHEN*DHN/RHDN SURF 1ON 1370
6424 1371 T2RS=-DHES*DHN/RHDS SURF 1ON 1371
6425 1372 FNS=FLGN*(AFN*(DHNN/RHDN*ACOSN+T2RN*ASINN)+(1.0-AFN)*CSANG)-FLGS* SURF 1ON 1372
6426 1373 1 (AFS*(DHNS/RHDS*ACOSS+T2RS*ASINS)-(1.0-AFS)*CSANG) SURF 1ON 1373
6427 1374 1050 CONTINUE SURF 1ON 1374
6428 1375 C ADD WALL ADHESION AT BOTTOM OF CELL SURF 1ON 1375
6429 1376 IF (NOWALL.EQ.1) GO TO 1060 SURF 1ON 1376
6430 1377 FEW=FEW-(2.0-FLGE-FLGW)*SANG SURF 1ON 1377
6431 1378 FNS=FNS-(2.0-FLGN-FLGS)*SANG SURF 1ON 1378
6432 1379 C LIMIT CURVATURE BY CELL SIZE SURF 1ON 1379
6433 1380 1060 AFew=ABS(FEW) SURF 1ON 1380
6434 1381 FEW=RDEW*AMIN1(Afew,2.0)*SIGN(1.0,FEW) SURF 1ON 1381
6435 1382 AFNS=ABS(FNS) SURF 1ON 1382
6436 1383 FNS=RDNS*AMIN1(AFNS,2.0)*SIGN(1.0,FNS) SURF 1ON 1383
6437 1384 C SURF 1ON 1384
6438 1385 FEWFNS=FEW+FNS SURF 1ON 1385
6439 1386 PS(IJK)=-SIGMA*AMIN1(ABS(FEW),PSLIM)*SIGN(1.0,FEWFNS) SURF 1ON 1386
6440 1387 C SURF 1ON 1387
6441 1388 1070 CONTINUE SURF 1ON 1388
6442 1389 C SURF 1ON 1389
6443 1390 C FIX FLOOR AND CEILING FOR ALL J AND K I=3,4.....IM1 SURF 1ON 1390
6444 1391 C SURF 1ON 1391
6445 1392 PSLIMI=PSLIM*SIGMA SURF 1ON 1392
6446 1393 DO 1150 K=2,KM1 SURF 1ON 1393
6447 1394 DO 1150 J=2,JM1 SURF 1ON 1394
6448 1395 DO 1150 I=2,IM1 SURF 1ON 1395
6449 1396 CALL IJKAJCT SURF 1ON 1396
6450 1397 PN(IJK)=0.0 SURF 1ON 1397
6451 1398 IF (BETA(IJK).LE.0.0) GO TO 1150 SURF 1ON 1398
6452 1399 IF (NF(IJK).LT.1.OR.NF(IJK).GT.6) GO TO 1150 SURF 1ON 1399
6453 1400 C SURF 1ON 1400
6454 1401 C TREAT INTERIOR CELLS OF I,J,K MESH BY GENERAL RULE, SURF 1ON 1401
6455 1402 C EXCEPTION FOR EDGE CELLS SURF 1ON 1402
6456 1403 C SURF 1ON 1403
6457 1404 PSAV=0.0 SURF 1ON 1404
6458 1405 RIAV=0.0 SURF 1ON 1405
6459 1406 DO 1140 MM=1,3 SURF 1ON 1406
6460 1407 DO 1140 NN=1,3 SURF 1ON 1407
6461 1408 DO 1140 LL=1,3 SURF 1ON 1408
6462 1409 L=I-2+LL SURF 1ON 1409
6463 1410 M=J-2+MM SURF 1ON 1410
6464 1411 N=K-2+NN SURF 1ON 1411
6465 1412 LMN=NQ*(II5*(N-1)+IMAX*(M-1)+(L-1))+1 SURF 1ON 1412
6466 1413 IF (LL.EQ.2.AND.MM.EQ.2.AND.NN.EQ.2) GO TO 1140 SURF 1ON 1413
6467 1414 IF (L.NE.1) GO TO 1080 SURF 1ON 1414
6468 1415 LPMN=LMN+NQ SURF 1ON 1415
6469 1416 PS(LMN)=PS(LPMN) SURF 1ON 1416
6470 1417 1080 IF (L.NE.IMAX) GO TO 1090 SURF 1ON 1417
6471 1418 LMMN=LMN-NQ SURF 1ON 1418
6472 1419 PS(LMN)=PS(LMMN) SURF 1ON 1419
6473 1420 1090 IF (M.NE.1) GO TO 1100 SURF 1ON 1420
6474 1421 LMPN=LMMN+NQ+IMAX SURF 1ON 1421
6475 1422 PS(LMN)=PS(LMPN) SURF 1ON 1422
6476 1423 1100 IF (M.NE.JMAX) GO TO 1110 SURF 1ON 1423
6477 1424 LMMN=LMN-NQ+IMAX SURF 1ON 1424
6478 1425 PS(LMN)=PS(LMMN) SURF 1ON 1425
6479 1426 1110 IF (N.NE.1) GO TO 1120 SURF 1ON 1426
6480 1427 LMNP=LMN+NQ+II5 SURF 1ON 1427

```

```

6481 1428      PS(LMN)=PS(LMNP)          SURF 1ON 1428
6482 1429      1120 IF (N.NE.KMAX) GO TO 1130    SURF 1ON 1429
6483 1430      LMNM=LMN-NQ*II5          SURF 1ON 1430
6484 1431      PS(LMN)=PS(LMNM)          SURF 1ON 1431
6485 1432      1130 CONTINUE           SURF 1ON 1432
6486 1433      IF (ABS(PS(LMN)).LT.1.0) GO TO 1140    SURF 1ON 1433
6487 1434      RIAV=RIAVER1.0          SURF 1ON 1434
6488 1435      PSAV=PSAV+PS(LMN)          SURF 1ON 1435
6489 1436      1140 CONTINUE           SURF 1ON 1436
6490 1437      IF (RIAVER.EQ.0.0) GO TO 1150    SURF 1ON 1437
6491 1438      PSAVI=PSAV/RIAVER        SURF 1ON 1438
6492 1439      PN(IJK)=PS(IJK)          SURF 1ON 1439
6493 1440      IF (ABS(PS(IJK)).GT.1.5*ABS(PSAVI).OR.ABS(PS(IJK)).GT.PSLIMI) PN    SURF 1ON 1440
6494 1441      1 (IJK)=AMIN1(PSLIMI,1.5*ABS(PSAVI))*SIGN(1.0,PSAVI)    SURF 1ON 1441
6495 1442      IF (ABS(PS(IJK)).LT.0.5*ABS(PSAVI)) PN(IJK)=0.5*ABS(PSAVI)*SIGN(1.0,PSAVI)    SURF 1ON 1442
6496 1443      1 O,PSAVI)          SURF 1ON 1443
6497 1444      1150 CONTINUE           SURF 1ON 1444
6498 1445      C
6499 1446      DO 1160 K=1,KMAX          SURF 1ON 1445
6500 1447      DO 1160 J=1,JMAX          SURF 1ON 1446
6501 1448      DO 1160 I=1,IMAX          SURF 1ON 1447
6502 1449      CALL IJKONLY           SURF 1ON 1448
6503 1450      PS(IJK)=PN(IJK)          SURF 1ON 1449
6504 1451      IF (I.EQ.1.OR.I.EQ.IMAX.OR.J.EQ.1.OR.J.EQ.JMAX.OR.K.EQ.1.OR.K.EQ.1.KMAX) PS(IJK)=0.0    SURF 1ON 1450
6505 1452      1 .KMAX) PS(IJK)=0.0    SURF 1ON 1451
6506 1453      1160 CONTINUE           SURF 1ON 1452
6507 1454      C
6508 1455      C      FIX FOR I=2 CASE    SURF 1ON 1453
6509 1456      C
6510 1457      I=2
6511 1458      DO 1180 K=2,KM1          SURF 1ON 1457
6512 1459      DO 1180 J=2,JM1          SURF 1ON 1458
6513 1460      CALL IJKAUJT           SURF 1ON 1459
6514 1461      IF (BETA(IJK).LE.0.0) GO TO 1180    SURF 1ON 1460
6515 1462      IF (NF(IJK).LT.1.0.R.NF(IJK).GT.6) GO TO 1180    SURF 1ON 1461
6516 1463      C
6517 1464      C      TREAT INTERIOR CELLS OF J,K MESH BY GENERAL RULE,    SURF 1ON 1462
6518 1465      C      EXCEPTION FOR EDGE CELLS          SURF 1ON 1463
6519 1466      C
6520 1467      PSAV=0.0
6521 1468      RIAV=0.0
6522 1469      DO 1170 MM=1,3          SURF 1ON 1468
6523 1470      DO 1170 NN=1,3          SURF 1ON 1469
6524 1471      M=J-2+MM
6525 1472      N=K-2+NN
6526 1473      LMN=NQ*(II5*(N-1)+IMAX*(M-1)+2)+1    SURF 1ON 1472
6527 1474      LMPN=LMN+NQ*IMAX          SURF 1ON 1473
6528 1475      LMMN=LMN-NQ*IMAX          SURF 1ON 1474
6529 1476      LMNP=LMN+NQ*II5          SURF 1ON 1475
6530 1477      LMNM=LMN-NQ*II5          SURF 1ON 1476
6531 1478      IF (M.EQ.1) PS(LMN)=PS(LMPN)    SURF 1ON 1477
6532 1479      IF (N.EQ.1) PS(LMN)=PS(LMNP)    SURF 1ON 1478
6533 1480      IF (M.EQ.JMAX) PS(LMN)=PS(LMMN)    SURF 1ON 1479
6534 1481      IF (N.EQ.KMAX) PS(LMN)=PS(LMNM)    SURF 1ON 1480
6535 1482      IF (ABS(PS(LMN)).LT.1.0) GO TO 1170    SURF 1ON 1481
6536 1483      RIAV=RIAVER1.0          SURF 1ON 1482
6537 1484      PSAV=PSAV+PS(LMN)          SURF 1ON 1483
6538 1485      1170 CONTINUE           SURF 1ON 1484
6539 1486      IF (RIAVER.EQ.0.0) GO TO 1180    SURF 1ON 1485
6540 1487      PSAVI=PSAV/RIAVER        SURF 1ON 1486
6541 1488      IF (ABS(PS(IJK)).GT.1.5*ABS(PSAVI).OR.ABS(PS(IJK)).GT.PSLIMI) PS    SURF 1ON 1487
6542 1489      1 (IJK)=AMIN1(PSLIMI,1.5*ABS(PSAVI))*SIGN(1.0,PSAVI)    SURF 1ON 1488
6543 1490      IF (ABS(PS(IJK)).LT.0.5*ABS(PSAVI)) PS(IJK)=0.5*ABS(PSAVI)*SIGN(1.0,PSAVI)    SURF 1ON 1489
6544 1491      1 O,PSAVI)          SURF 1ON 1490
6545 1492      1180 CONTINUE           SURF 1ON 1491
6546 1493      C
6547 1494      I=2
6548 1495      DO 1190 K=1,KMAX,KM1    SURF 1ON 1492
6549 1496      DO 1190 J=1,JMAX,JM1    SURF 1ON 1493
6550 1497      CALL IJKONLY           SURF 1ON 1494
6551 1498      PS(IJK)=0.0
6552 1499      1190 CONTINUE           SURF 1ON 1495
6553 1500      C
6554 1501      C      *****      SURF 1ON 1496
6555 1502      RETURN           SURF 1ON 1497
6556 1503      END
6557 1 *DK SURF 100          SURF 100 1
6558 2 *DK TILDE           TILDE 1
6559 3      SUBROUTINE TILDE        TILDE 2
6560 4 *CA SLCOM1          TILDE 3

```

```

6561 5 C      TILDE   4
6562 6 C * * EXPLICITLY APPROXIMATE NEW TIME LEVEL VELOCITIES TILDE   5
6563 7 C      TILDE   6
6564 8 DO 50 J=2,JM1 TILDE   7
6565 9 DO 50 I=2,IM1 TILDE   8
6566 10 DO 40 K=2,KM1 TILDE   9
6567 11 CALL CALCIJK TILDE 10
6568 12 UDUM=O.O TILDE 11
6569 13 VDUM=O.O TILDE 12
6570 14 WDUM=O.O TILDE 13
6571 15 IF (BETA(IJK).LT.O.O) GO TO 30 TILDE 14
6572 16 C      TILDE 15
6573 17 IJ=I+(J-1)*IMAX TILDE 16
6574 18 GX=GXA(IJ) TILDE 17
6575 19 GY=GYA(IJ) TILDE 18
6576 20 C * * U-VELOCITY TILDE 19
6577 21 C      TILDE 20
6578 22 IF (BETA(IPJK).LT.O.O) GO TO 10 TILDE 21
6579 23 IF (F(IJK)+F(IPJK).LT.EMF) GO TO 10 TILDE 22
6580 24 SGU=SIGN(1.0,UN(IJK)) TILDE 23
6581 25 DUDR=(UN(IPJK)-UN(IJK))*RDX(I+1) TILDE 24
6582 26 DUDL=(UN(IJK)-UN(IMJK))*RDX(I) TILDE 25
6583 27 IF (AR(IPJK).LT.EM6) DUDR=O.O TILDE 26
6584 28 IF (AR(IMJK).LT.EM6) DUDL=O.O TILDE 27
6585 29 DXAV=DELX(I)+DELX(I+1)+ALPHA*SGU*(DELX(I+1)-DELX(I)) TILDE 28
6586 30 FUX=UN(IJK)*(DELX(I)*DUDR+DELX(I+1)*DUDL+ALPHA*SGU*(DELX(I+1)*DUDL TILDE 29
6587 31 1 -DELX(I)*DUDR))/DXAV TILDE 30
6588 32 VBDYT=(DELX(I)*VN(IPJK)+DELX(I+1)*VN(IJK))/(DELX(I)+DELX(I+1)) TILDE 31
6589 33 VBDYB=(DELX(I)*VN(IPJMK)+DELX(I+1)*VN(IJMK))/(DELX(I)+DELX(I+1)) TILDE 32
6590 34 DVDY=VBDYT-VBDYB TILDE 33
6591 35 IF (ABK(IJK).LT.EM6.OR.ABK(IPJK).LT.EM6.OR.ABK(IJMK).LT.EM6.OR.ABK TILDE 34
6592 36 1 (IPJMK).LT.EM6) DVDY=O.O TILDE 35
6593 37 VAV=O.5*(VBDYT+VBDYB) TILDE 36
6594 38 DYF=0.5*(DELY(J)+DELY(J+1)) TILDE 37
6595 39 DYA=0.5*(DELY(J-1)+DELY(J)) TILDE 38
6596 40 DUDF=RR(I)*(UN(IPJK)-UN(IJK))/DYF TILDE 39
6597 41 DUDA=RR(I)*(UN(IJK)-UN(IJMK))/DYA TILDE 40
6598 42 IF (AR(IPJK).LT.EM6) DUDF=O.O TILDE 41
6599 43 IF (AR(IJMK).LT.EM6) DUDA=O.O TILDE 42
6600 44 SGV=SIGN(1.0,VAV) TILDE 43
6601 45 DYAV=DYF+DYA+ALPHA*SGV*(DYF-DYA) TILDE 44
6602 46 FUY=(DYA*DUDF+DYF*DUDA+ALPHA*SGV*(DYF*DUDA-DYA*DUDF))*(VAV/DYAV) TILDE 45
6603 47 WBDYT=(DELX(I+1)*WN(IJK)+DELX(I)*WN(IPJK))/(DELX(I)+DELX(I+1)) TILDE 46
6604 48 WBDYB=(DELX(I+1)*WN(IJMK)+DELX(I)*WN(IPJMK))/(DELX(I)+DELX(I+1)) TILDE 47
6605 49 WAV=0.5*(WBDYT+WBDYB) TILDE 48
6606 50 DZT=0.5*(DELZ(K)+DELZ(K+1)) TILDE 49
6607 51 DZB=0.5*(DELZ(K-1)+DELZ(K)) TILDE 50
6608 52 DUDT=(UN(IPJK)-UN(IJK))/DZT TILDE 51
6609 53 DUDB=(UN(IJK)-UN(IJMK))/DZB TILDE 52
6610 54 IF (AR(IPJK).LT.EM6) DUDT=O.O TILDE 53
6611 55 IF (AR(IJMK).LT.EM6) DUDB=O.O TILDE 54
6612 56 SGW=SIGN(1.0,WAV) TILDE 55
6613 57 DZAV=DZT+DZB+ALPHA*SGW*(DZT-DZB) TILDE 56
6614 58 FUZ=(DZB*DUDT+DZT*DUDB+ALPHA*SGW*(DZT*DUDB-DZB*DUDT))*(WAV/DZAV) TILDE 57
6615 59 DUDXSQ=2.0*(DUDR-DUDL)/(DELX(I)+DELX(I+1)) TILDE 58
6616 60 DUDYSQ=RDY(J)*(DUDF-DUDA)*RR(I) TILDE 59
6617 61 DUDZSQ=RDZ(K)*(DUDT-DUDB) TILDE 60
6618 62 DUDCYL=1.0*RX(I)*(DELX(I)*DUDR+DELX(I+1)*DUDL)/(DELX(I)+DELX(I+1)) TILDE 61
6619 63 1 -2.0*RR(I)*RX(I)*RDY(J)*DVDT-UN(IJK)*RX(I)**2 TILDE 62
6620 64 VISX=NU*(DUDXSQ+DUDYSQ+DUDZSQ+CYL*DUDCYL) TILDE 63
6621 65 FUC=CYL*(-0.25*(VBDYT+VBDYB)**2)*RX(I) TILDE 64
6622 66 FRU=-RADPS*(VBDYT+VBDYB)-X(I)*RADPS**2 TILDE 65
6623 67 UDUM=UN(IJK)+DELT*((P(IJK)-P(IPJK))*2.0/(DELX(I)+DELX(I+1))+GX-FUX TILDE 66
6624 68 1 -FUY-FUZ+VISX-FUC-FRU) TILDE 67
6625 69 10 CONTINUE TILDE 68
6626 70 C      TILDE 69
6627 71 C * * V-VELOCITY TILDE 70
6628 72 C      TILDE 71
6629 73 IF (BETA(IPJK).LT.O.O) GO TO 20 TILDE 72
6630 74 IF (F(IJK)+F(IPJK).LT.EMF) GO TO 20 TILDE 73
6631 75 VBDYR=(DELX(I+1)*VN(IJK)+DELX(I)*VN(IPJK))/(DELX(I)+DELX(I+1)) TILDE 74
6632 76 VBDYL=(DELX(I)*VN(IMJK)+DELX(I-1)*VN(IJK))/(DELX(I-1)+DELX(I)) TILDE 75
6633 77 UBDYR=(DELY(J)*UN(IPJK)+DELY(J+1)*UN(IJK))/(DELY(J)+DELY(J+1)) TILDE 76
6634 78 UBDYL=(DELY(J)*UN(IMJPK)+DELY(J+1)*UN(IMJK))/(DELY(J)+DELY(J+1)) TILDE 77
6635 79 UAV=O.5*(UBDYR+UBDYL) TILDE 78
6636 80 DXR=0.5*(DELX(I)+DELX(I+1)) TILDE 79
6637 81 DXL=0.5*(DELX(I-1)+DELX(I)) TILDE 80
6638 82 DVDR=(VN(IPJK)-VN(IJK))/DXR TILDE 81
6639 83 DVDL=(VN(IJK)-VN(IMJK))/DXL TILDE 82
6640 84 IF (ABK(IPJK).LT.EM6) DVDR=O.O TILDE 83

```

```

6641 85 IF (ABK(IJK).LT.EM6) DVDL=0.0 TILDE 84
6642 86 SGU=SIGN(1.0,UAV) TILDE 85
6643 87 DXAV=DXR+DXL+ALPHA*SGU*(DXR-DXL) TILDE 86
6644 88 FVX=(DXR*DVML+DXL*DVR+ALPHA*SGU*(DXR*DVML-DXL*DVR))*(UAV/DXAV) TILDE 87
6645 89 FVC=CYL*(VBDYR*UBDYR+VBDYL*UBDYL)*O.5*RXI(I) TILDE 88
6646 90 SGV=SIGN(1.0,VN(IJK)) TILDE 89
6647 91 DVDF=(VN(IJPK)-VN(IJK))*RDY(J+1)*RRI(I) TILDE 90
6648 92 DVDA=(VN(IJK)-VN(IJMK))*RDY(J)*RRI(I) TILDE 91
6649 93 IF (ABK(IJPK).LT.EM6) DVDF=0.0 TILDE 92
6650 94 IF (ABK(IJMK).LT.EM6) DVDA=0.0 TILDE 93
6651 95 DYAV=DELY(J)+DELY(J+1)+ALPHA*SGV*(DELY(J+1)-DELY(J)) TILDE 94
6652 96 FVY=VN(IJK)*(DELY(J)*DVDF+DELY(J+1)*DVDA+ALPHA*SGV*(DELY(J+1)*DVDA TILDE 95
6653 97 1 -DELY(J)*DVDF))/DYAV TILDE 96
6654 98 WBDYT=(DELY(J+1)*WN(IJK)+DELY(J)*WN(IJPK))/(DELY(J)+DELY(J+1)) TILDE 97
6655 99 WBDYB=(DELY(J+1)*WN(IJKM)+DELY(J)*WN(IJPKM))/(DELY(J)+DELY(J+1)) TILDE 98
6656 100 WAV=0.5*(WBDYT+WBDYB) TILDE 99
6657 101 DZT=0.5*(DELZ(K)+DELZ(K+1)) TILDE 100
6658 102 DZB=0.5*(DELZ(K-1)+DELZ(K)) TILDE 101
6659 103 DVDT=(VN(IJPK)-VN(IJK))/DZT TILDE 102
6660 104 DVDB=(VN(IJK)-VN(IJKM))/DZB TILDE 103
6661 105 IF (ABK(IJPK).LT.EM6) DVDT=0.0 TILDE 104
6662 106 IF (ABK(IJMK).LT.EM6) DVDB=0.0 TILDE 105
6663 107 SGW=SIGN(1.0,WAV) TILDE 106
6664 108 DZAV=DZT+DZB+ALPHA*SGW*(DZT-DZB) TILDE 107
6665 109 FVZ=(DZT*DVB+DZB*DVT+ALPHA*SGW*(DZT*DVB-DZB*DVT))*(WAV/DZAV) TILDE 108
6666 110 DVDXSQ=RDX(I)*(DVDR-DVDL) TILDE 109
6667 111 DVDXSQ=2.0*RRI(I)*(DVDR-DVDA)/(DELY(J)+DELY(J+1)) TILDE 110
6668 112 DVDXSQ=RDX(K)*(DVDT-DVDB) TILDE 111
6669 113 DUDY=UN(IJPK)+UN(IMJPK)-UN(IJK)-UN(IMJK) TILDE 112
6670 114 IF (AR(IJPK).LT.EM6.OR.AR(IMJPK).LT.EM6.OR.AR(IJK).LT.EM6.OR.AR TILDE 113
6671 115 1 (IMJK).LT.EM6) DUDY=0.0 TILDE 114
6672 116 DVDCYL=RXI(I)*O.5*(DVDR+DVDL)+2.0*RXI(I)*RRI(I)*DUDY/(DELY(J)+DELY TILDE 115
6673 117 1 (J+1))-VN(IJK)*RXI(I)**2 TILDE 116
6674 118 VISY=NU*(DVDXSQ+DVDXSQ+DVDZSQ+CYL*DVCYL) TILDE 117
6675 119 FRV=RADPS*(UBDYR+UBDYL) TILDE 118
6676 120 VDUM=VN(IJK)+DELT*((P(IJK)-P(IJPK))*2.0/(DELY(J)+DELY(J+1))*RRI(I) TILDE 119
6677 121 1 +GY-FVX-FVY-FVZ+VISY-FVC-FRV) TILDE 120
6678 122 20 CONTINUE TILDE 121
6679 123 C
6680 124 C * * W-VELOCITY TILDE 122
6681 125 C
6682 126 IF (BETA(IJPK).LT.0.0) GO TO 30 TILDE 125
6683 127 IF (F(IJK)+F(IJPK).LT.EMF) GO TO 30 TILDE 126
6684 128 UBDYR=(DELZ(K+1)*UN(IJK)+DELZ(K)*UN(IJPK))/(DELZ(K)+DELZ(K+1)) TILDE 127
6685 129 UBDYL=(DELZ(K+1)*UN(IMJK)+DELZ(K)*UN(IMJPK))/(DELZ(K)+DELZ(K+1)) TILDE 128
6686 130 UAV=0.5*(UBDYR+UBDYL) TILDE 129
6687 131 DXR=0.5*(DELX(I)+DELX(I+1)) TILDE 130
6688 132 DXL=0.5*(DELX(I-1)+DELX(I)) TILDE 131
6689 133 DWDR=(WN(IPJK)-WN(IJK))/DXR TILDE 132
6690 134 DWDL=(WN(IJK)-WN(IMJK))/DXL TILDE 133
6691 135 IF (AT(IPJK).LT.EM6) DWDR=0.0 TILDE 134
6692 136 IF (AT(IMJK).LT.EM6) DWDL=0.0 TILDE 135
6693 137 SGU=SIGN(1.0,UAV) TILDE 136
6694 138 DXAV=DXR+DXL+ALPHA*SGU*(DXR-DXL) TILDE 137
6695 139 FWX=(DXR*DWDL+DXL*DWDR+ALPHA*SGU*(DXR*DWDL-DXL*DWDR))*(UAV/DXAV) TILDE 138
6696 140 VBDYF=(DELZ(K+1)*VN(IJK)+DELZ(K)*VN(IJPK))/(DELZ(K)+DELZ(K+1)) TILDE 139
6697 141 VBDYA=(DELZ(K+1)*VN(IJKM)+DELZ(K)*VN(IJMKP))/(DELZ(K)+DELZ(K+1)) TILDE 140
6698 142 VAV=0.5*(VBDYF+VBDYA) TILDE 141
6699 143 DYF=0.5*(DELY(J)+DELY(J+1)) TILDE 142
6700 144 DYA=0.5*(DELY(J-1)+DELY(J)) TILDE 143
6701 145 DWDF=RRI(I)*(WN(IJPK)-WN(IJK))/DYF TILDE 144
6702 146 DWDA=RRI(I)*(WN(IJK)-WN(IJMK))/DYA TILDE 145
6703 147 IF (AT(IJPK).LT.EM6) DWDF=0.0 TILDE 146
6704 148 IF (AT(IMJK).LT.EM6) DWDA=0.0 TILDE 147
6705 149 SGV=SIGN(1.0,VAV) TILDE 148
6706 150 DYAV=DYF+DYA+ALPHA*SGV*(DYF-DYA) TILDE 149
6707 151 FWY=(DYF*DWDA+DYA*DWDF+ALPHA*SGV*(DYF*DWDA-DYA*DWDF))*(VAV/DYAV) TILDE 150
6708 152 SGW=SIGN(1.0,WN(IJK)) TILDE 151
6709 153 DWDT=(WN(IJPK)-WN(IJK))*RDZ(K+1) TILDE 152
6710 154 DWDB=(WN(IJK)-WN(IJKM))*RDZ(K) TILDE 153
6711 155 IF (AT(IJPK).LT.0.0) DWDT=0.0 TILDE 154
6712 156 IF (AT(IMJK).LT.EM6) DWDB=0.0 TILDE 155
6713 157 DZAV=DELZ(K)+DELZ(K+1)+ALPHA*SGW*(DELZ(K+1)-DELZ(K)) TILDE 156
6714 158 FWZ=(DELZ(K)+DWDT+DELZ(K+1)*DWDB+ALPHA*SGW*(DELZ(K+1)*DWDB-DELZ(K) TILDE 157
6715 159 1 *DWDT))*(WN(IJK)/DZAV) TILDE 158
6716 160 DWDXSQ=RDX(I)*(DWDR-DWDL) TILDE 159
6717 161 DWDXSQ=RRI(I)*RDY(J)*(DWDF-DWDA) TILDE 160
6718 162 DWDXSQ=2.0*(DWDT-DWDB)/(DELZ(K+1)+DELZ(K)) TILDE 161
6719 163 DWDCYL=RXI(I)*O.5*(DWDR+DWDL) TILDE 162
6720 164 VISZ=NU*(DWDXSQ+DWDXSQ+DVDZSQ+CYL*DVCYL) TILDE 163

```

```

6721 165      WDUM=WN(IJK)+DELT*((P(IJK)-P(IJKP))*2.0/(DELZ(K)+DELZ(K+1))+GZ-FWX TILDE 164
6722 166      1 -FWY-FWZ+VISZ) TILDE 165
6723 167      30 CONTINUE TILDE 166
6724 168 C
6725 169 C * * COMPUTE VELOCITIES CONSIDERING SECOND ORDER TILDE 167
6726 170 C ACCURATE OPTION PARAMETER AVE TILDE 168
6727 171 C
6728 172      U(IJK)=AVE*UDUM+(1.0-AVE)*U(IJK) TILDE 171
6729 173      V(IJK)=AVE*VDUM+(1.0-AVE)*V(IJK) TILDE 172
6730 174      W(IJK)=AVE*WDUM+(1.0-AVE)*W(IJK) TILDE 173
6731 175      40 CONTINUE TILDE 174
6732 176      50 CONTINUE TILDE 175
6733 177 C
6734 178 C ****RETURN TILDE 176
6735 179      RETURN TILDE 177
6736 180      END TILDE 178
6737 1 *DK VELV TILDE 179
6738 2      SUBROUTINE VELV (I1,I2,J1,J2,KK1,KK2,NA,IPER) VELV 1
6739 3 *CA SLCOM1 VELV 2
6740 4 C VELV 3
6741 5 C * * DRAW VELOCITY VECTORS IN PLANE OR PERSPECTIVE PLOTS VELV 4
6742 6 C VELV 5
6743 7 C * * DETERMINE PLANE TO BE PLOTTED VELV 6
6744 8 C VELV 7
6745 9      IFM=0 VELV 8
6746 10     JFM=0 VELV 9
6747 11     KFM=0 VELV 10
6748 12     IF (I1.NE.I2) GO TO 10 VELV 11
6749 13     IFM=I1 VELV 12
6750 14     MPLN=1 VELV 13
6751 15     ICON=J2-J1+1 VELV 14
6752 16     JCON=KK2-KK1+1 VELV 15
6753 17     10 IF (J1.NE.J2) GO TO 20 VELV 16
6754 18     JFM=J1 VELV 17
6755 19     MPLN=2 VELV 18
6756 20     ICON=I2-I1+1 VELV 19
6757 21     JCON=KK2-KK1+1 VELV 20
6758 22     20 IF (KK1.NE.KK2) GO TO 30 VELV 21
6759 23     KFM=KK1 VELV 22
6760 24     MPLN=3 VELV 23
6761 25     ICON=I2-I1+1 VELV 24
6762 26     JCON=J2-J1+1 VELV 25
6763 27     30 CONTINUE VELV 26
6764 28 C VELV 27
6765 29 C * * DRAW PLANE OR PERSPECTIVE PLOT FRAME VELV 28
6766 30 C VELV 29
6767 31     CALL FADV (NA) VELV 30
6768 32     IF (IPER.EQ.2) GO TO 40 VELV 31
6769 33     AXP=1.0 VELV 32
6770 34     IF (MPLN.EQ.1) AXP=0.0 VELV 33
6771 35     AYP=1.0 VELV 34
6772 36     IF (MPLN.EQ.3) AYP=0.0 VELV 35
6773 37     IF (NA.GT.0) CALL DRF (IFM,JFM,KFM) VELV 36
6774 38     GO TO 50 VELV 37
6775 39     40 IF (NA.GT.0) CALL DRFP VELV 38
6776 40 C VELV 39
6777 41 C * * SET UP TO DRAW VELOCITY VECTORS VELV 40
6778 42 C VELV 41
6779 43     50 IJ=1 VELV 42
6780 44     KNTIJ=1 VELV 43
6781 45     VMAX=0.0 VELV 44
6782 46     VMAXI=0.0 VELV 45
6783 47     VMAXJ=0.0 VELV 46
6784 48     VMAXK=0.0 VELV 47
6785 49     DO 110 K=2,KMAX VELV 48
6786 50     DO 110 J=2,JMAX VELV 49
6787 51     DO 110 I=2,IMAX VELV 50
6788 52     CALL CALCijk VELV 51
6789 53     IF (F(IJK).LT.0.5) GO TO 110 VELV 52
6790 54     IF (AC(IJK).LT.0.5) GO TO 110 VELV 53
6791 55     GO TO (60,70,80), MPLN VELV 54
6792 56     60 IF (I.NE.I1) GO TO 110 VELV 55
6793 57     IF (J.LT.J1) GO TO 110 VELV 56
6794 58     IF (J.GT.J2) GO TO 110 VELV 57
6795 59     IF (K.LT.KK1) GO TO 110 VELV 58
6796 60     IF (K.GT.KK2) GO TO 110 VELV 59
6797 61     GO TO 90 VELV 60
6798 62     70 IF (J.NE.J1) GO TO 110 VELV 61
6799 63     IF (K.LT.KK1) GO TO 110 VELV 62
6800 64     IF (K.GT.KK2) GO TO 110 VELV 63

```

```

6801 65      IF (I.LT.I1) GO TO 110          VELV   65
6802 66      IF (I.GT.I2) GO TO 110          VELV   66
6803 67      GO TO 90                      VELV   67
6804 68      80 IF (K.NE.KK1) GO TO 110          VELV   68
6805 69      IF (J.LT.J1) GO TO 110          VELV   69
6806 70      IF (J.GT.J2) GO TO 110          VELV   70
6807 71      IF (I.LT.I1) GO TO 110          VELV   71
6808 72      IF (I.GT.I2) GO TO 110          VELV   72
6809 73      90 CONTINUE                  VELV   73
6810 74      STH1=O.O                      VELV   74
6811 75      CTH1=1.O                      VELV   75
6812 76      CYLM=1.O                      VELV   76
6813 77      IF (MPLN.NE.3.AND.IPER.EQ.1) GO TO 100    VELV   77
6814 78      TH1=CYLM*RX(IM1)*YJ(J)          VELV   78
6815 79      STH1=SIN(TH1)                  VELV   79
6816 80      CTH1=COS(TH1)                  VELV   80
6817 81      CYLM=1.O-CYL                 VELV   81
6818 82      100 CONTINUE                  VELV   82
6819 83 C
6820 84 C * * DETERMINE MAXIMUM VECTOR LENGTH VELV   83
6821 85 C
6822 86      VINER=XI(I)*RADPS            VELV   86
6823 87      VINER=O.O                      VELV   87
6824 88      UAVE=0.5*(U(IJK)+U(IMJK))        VELV   88
6825 89      VAVE=0.5*(V(IJK)+V(IUMK))+VINER    VELV   89
6826 90      UVV(IJ)=UAVE*CTH1-VAVE*STH1        VELV   90
6827 91      VVV(IJ)=UAVE*STH1+VAVE*CTH1        VELV   91
6828 92      WVV(IJ)=0.5*(W(IJK)+W(IJMK))        VELV   92
6829 93      XPC(IJ)=XI(I)*CTH1                VELV   93
6830 94      YPC(IJ)=YJ(J)*CYLM+XI(I)*STH1        VELV   94
6831 95      ZPC(IJ)=ZK(K)                  VELV   95
6832 96      VMAX=AMAX1(VMAX,ABS(UVV(IJ)),ABS(VVV(IJ)),ABS(WVV(IJ)))    VELV   96
6833 97      VMAXI=AMAX1(VMAXI,ABS(VVV(IJ)),ABS(WVV(IJ)))          VELV   97
6834 98      VMAXJ=AMAX1(VMAXJ,ABS(UVV(IJ)),ABS(WVV(IJ)))          VELV   98
6835 99      VMAXK=AMAX1(VMAXK,ABS(UVV(IJ)),ABS(VVV(IJ)))          VELV   99
6836 100     IF (BETA(IJK).LT.0.0.OR.F(IJK).LT.0.5) UVV(IJ)=1.OE+10    VELV   100
6837 101     KT1=KM1                      VELV   101
6838 102     IF (K.GT.KT1) UVV(IJ)=+1.OE+10        VELV   102
6839 103     KNTIJ=KNTIJ+1                  VELV   103
6840 104     IJ=IJ+NQ2                      VELV   104
6841 105     IF (IJ.GT.NVEC) GO TO 310        VELV   105
6842 106     110 CONTINUE                  VELV   106
6843 107     KNTIJ=KNTIJ-1                  VELV   107
6844 108 C
6845 109 C * * SCALE VECTOR LENGTH VELV   108
6846 110 C
6847 111     GO TO (120,130,140), MPLN          VELV   111
6848 112     120 DROU=VELMX*DELMN/(VMAXI+1.OE-10)    VELV   112
6849 113     GO TO 150                      VELV   113
6850 114     130 DROU=VELMX*DELMN/(VMAXJ+1.OE-10)    VELV   114
6851 115     GO TO 150                      VELV   115
6852 116     140 DROU=VELMX*DELMN/(VMAXK+1.OE-10)    VELV   116
6853 117     150 CONTINUE                  VELV   117
6854 118     IJ=1                        VELV   118
6855 119     IF (MPLN.NE.3) GO TO 160        VELV   119
6856 120     XBL=XBLC                      VELV   120
6857 121     YBF=YBFC                      VELV   121
6858 122     160 CONTINUE                  VELV   122
6859 123     DO 210 L=1,KNTIJ            VELV   123
6860 124     X1P=XPC(IJ)                  VELV   124
6861 125     Y1P=YPC(IJ)                  VELV   125
6862 126     Z1P=ZPC(IJ)                  VELV   126
6863 127     U1P=UVV(IJ)                  VELV   127
6864 128     V1P=VVV(IJ)                  VELV   128
6865 129     W1P=WVV(IJ)                  VELV   129
6866 130     IF (U1P.GT.1.OE+06) GO TO 210    VELV   130
6867 131     GO TO (170,190), IPER          VELV   131
6868 132 C
6869 133 C * * PLANE VECTOR PLOT VELV   133
6870 134 C
6871 135     170 CONTINUE                  VELV   135
6872 136     X1EF=AXP*(X1P-XBL)+(1.O-AXP)*(Y1P-YBF)    VELV   136
6873 137     Y1EF=AYP*(Z1P-ZBB)+(1.O-AYP)*(Y1P-YBF)    VELV   137
6874 138     X2EF=AXP*(X1P+U1P*DROU-XBL)+(1.O-AXP)*(Y1P+V1P*DROU-YBF)    VELV   138
6875 139     Y2EF=AYP*(Z1P+W1P*DROU-ZBB)+(1.O-AYP)*(Y1P+V1P*DROU-YBF)    VELV   139
6876 140     IX1=FIXL(MPLN)+X1EF*XCONV(MPLN)          VELV   140
6877 141     IY1=FIYB+Y1EF*YCONV(MPLN)          VELV   141
6878 142     IX2=FIXL(MPLN)+X2EF*XCONV(MPLN)          VELV   142
6879 143     IY2=FIYB+Y2EF*YCONV(MPLN)          VELV   143
6880 144     CALL DRV (IX1,IY1,IX2,IY2)          VELV   144

```

```

6881 145 C DRAW ARROWHEADS VELV 145
6882 146 FIX1=IX1 VELV 146
6883 147 FIX2=IX2 VELV 147
6884 148 FIY1=IY1 VELV 148
6885 149 FIY2=IY2 VELV 149
6886 150 X34=FIX1+O.85*(FIX2-FIX1) VELV 150
6887 151 Y34=FIY1+O.85*(FIY2-FIY1) VELV 151
6888 152 DX234=FIX2-X34 VELV 152
6889 153 DY234=FIY2-Y34 VELV 153
6890 154 D234=SQRT(DX234**2+DY234**2) VELV 154
6891 155 DA234=D234*0.57735027 VELV 155
6892 156 C TAN(15 DEG)=0.57735027 VELV 156
6893 157 CDSB=DA234 VELV 157
6894 158 SINB=O.O VELV 158
6895 159 IF (DX234.EQ.0.0) GO TO 180 VELV 159
6896 160 BETA234=1.57079633-ATAN(DY234/DX234) VELV 160
6897 161 CDSB=COS(BETA234)*DA234 VELV 161
6898 162 SINB=SIN(BETA234)*DA234 VELV 162
6899 163 180 IX3=X34-COSB VELV 163
6900 164 IX4=X34+COSB VELV 164
6901 165 IY3=Y34+SINB VELV 165
6902 166 IY4=Y34-SINB VELV 166
6903 167 CALL DRV (IX2,IY2,IX3,IY3) VELV 167
6904 168 CALL DRV (IX2,IY2,IX4,IY4) VELV 168
6905 169 GO TO 210 VELV 169
6906 170 C VELV 170
6907 171 C * * PERSPECTIVE VECTOR PLOT VELV 171
6908 172 C VELV 172
6909 173 190 CALL PCNV (IXI1,IETA1,X1P,Y1P,Z1P) VELV 173
6910 174 X2P=X1P+U1P*DROU VELV 174
6911 175 Y2P=Y1P+V1P*DROU VELV 175
6912 176 Z2P=Z1P+W1P*DROU VELV 176
6913 177 CALL PCNV (IXI2,IETA2,X2P,Y2P,Z2P) VELV 177
6914 178 CALL DRVEC (IXI1,IETA1,IXI2,IETA2) VELV 178
6915 179 C DRAW ARROWHEADS VELV 179
6916 180 FIX1=IXI1 VELV 180
6917 181 FIX2=IXI2 VELV 181
6918 182 FIY1=IETA1 VELV 182
6919 183 FIY2=IETA2 VELV 183
6920 184 X34=FIX1+O.85*(FIX2-FIX1) VELV 184
6921 185 Y34=FIY1+O.85*(FIY2-FIY1) VELV 185
6922 186 DX234=FIX2-X34 VELV 186
6923 187 DY234=FIY2-Y34 VELV 187
6924 188 D234=SQRT(DX234**2+DY234**2) VELV 188
6925 189 DA234=D234*0.57735027 VELV 189
6926 190 C TAN(15 DEG)=0.57735027 VELV 190
6927 191 CDSB=DA234 VELV 191
6928 192 SINB=O.O VELV 192
6929 193 IF (DX234.EQ.0.0) GO TO 200 VELV 193
6930 194 BETA234=1.57079633-ATAN(DY234/DX234) VELV 194
6931 195 CDSB=COS(BETA234)*DA234 VELV 195
6932 196 SINB=SIN(BETA234)*DA234 VELV 196
6933 197 200 IX3=X34-COSB VELV 197
6934 198 IX4=X34+COSB VELV 198
6935 199 IY3=Y34+SINB VELV 199
6936 200 IY4=Y34-SINB VELV 200
6937 201 CALL DRV (IXI2,IETA2,IX3,IY3) VELV 201
6938 202 CALL DRV (IXI2,IETA2,IX4,IY4) VELV 202
6939 203 210 IJ=IJ+NQ2 VELV 203
6940 204 C VELV 204
6941 205 C * * DRAW CONSTANT PLANE FREE SURFACE VELV 205
6942 206 C VELV 206
6943 207 IF (IPER.NE.1) GO TO 220 VELV 207
6944 208 CALL SURCNTR (I1,I2,J1,J2,KK1,KK2,MPLN,AXP,AYP) VELV 208
6945 209 C VELV 209
6946 210 C * * WRITE TITLES AND INFO ON PLOT VELV 210
6947 211 C VELV 211
6948 212 220 CONTINUE VELV 212
6949 213 XBL=X(1) VELV 213
6950 214 YBF=Y(1) VELV 214
6951 215 CALL LINCNT (59) VELV 215
6952 216 GO TO (230,240,250), MPLN VELV 216
6953 217 230 WRITE (12,350) VMAX,VMAXI VELV 217
6954 218 GO TO 260 VELV 218
6955 219 240 WRITE (12,360) VMAX,VMAXJ VELV 219
6956 220 GO TO 260 VELV 220
6957 221 250 WRITE (12,370) VMAX,VMAXK VELV 221
6958 222 260 CONTINUE VELV 222
6959 223 WRITE (12,380) CYCLE,T,(NAME(I),I=1,5),JNM VELV 223
6960 224 IF (IPER.EQ.2) GO TO 300 VELV 224

```

```

6961 225      GO TO (270,280,290), MPLN          VELV   225
6962 226      270 WRITE (12,320) IFM,J1,J2,KK1,KK2  VELV   226
6963 227      GO TO 300                          VELV   227
6964 228      280 WRITE (12,330) JFM,I1,I2,KK1,KK2  VELV   228
6965 229      GO TO 300                          VELV   229
6966 230      290 WRITE (12,340) KFM,I1,I2,J1,J2    VELV   230
6967 231      300 RETURN                         VELV   231
6968 232 C
6969 233 C * * ERROR - STOP AND PRINT           VELV   233
6970 234 C
6971 235      310 CONTINUE                         VELV   235
6972 236      WRITE (9,390)                        VELV   236
6973 237      WRITE (59,390)                       VELV   237
6974 238      WRITE (12,390)                       VELV   238
6975 239      CALL EXITA (3020)                     VELV   239
6976 240 C
6977 241      320 FORMAT (13H CONSTANT I=,I4,17H   SURFACE   I=,I3,3H TO,I3,7H VELV   241
6978 242      1 K=,I3,3H TO,I3)                   VELV   242
6979 243      330 FORMAT (13H CONSTANT J=,I4,17H   SURFACE   I=,I3,3H TO,I3,7H VELV   243
6980 244      1 K=,I3,3H TO,I3)                   VELV   244
6981 245      340 FORMAT (13H CONSTANT K=,I4,17H   SURFACE   I=,I3,3H TO,I3,7H VELV   245
6982 246      1 J=,I3,3H TO,I3)                   VELV   246
6983 247      350 FORMAT (18H VELOCITY VECTORS,4X,7H VMAX=,1PE12.5,4X,6HVMAXI=,E12 VELV   247
6984 248      1 .5)                                VELV   248
6985 249      360 FORMAT (18H VELOCITY VECTORS,4X,7H VMAX=,1PE12.5,4X,6HVMAXJ=,E12 VELV   249
6986 250      1 .5)                                VELV   250
6987 251      370 FORMAT (18H VELOCITY VECTORS,4X,7H VMAX=,1PE12.5,4X,6HVMAXK=,E12 VELV   251
6988 252      1 .5)                                VELV   252
6989 253      380 FORMAT (8H CYCLE=,I5,4H T=,1PE12.5,5X,5A8,1X,A8) VELV   253
6990 254      390 FORMAT (51HO - - - ERROR - - TWX ARRAY FOR PLOTTING EXCEEDED) VELV   254
6991 255      END                                 VELV   255
6992 1 *DK VCHGCAL                           VCHGCAL 1
6993 2          SUBROUTINE VCHGCAL             VCHGCAL 2
6994 3 *CA SLCOM1                            VCHGCAL 3
6995 4 C
6996 5 C * * CALCULATE TOTAL VOLUME CHANGE VCHGCAL 4
6997 6          VOFTOT=0.0                      VCHGCAL 5
6998 7 C
6999 8          DO 40 K=2,KM1                  VCHGCAL 6
7000 9          DO 40 J=2,JM1                  VCHGCAL 7
7001 10         DO 40 I=2,IM1                  VCHGCAL 8
7002 11         CALL CALCIJK                VCHGCAL 9
7003 12         IF (BETA(IJK).LT.0.0) GO TO 40 VCHGCAL 10
7004 13         VOFTOT=VOFTOT+F(IJK)*DELX(I)*DELY(J)/RRI(I)*DELZ(K)*AC(IJK) VCHGCAL 11
7005 14         VCHG=0.0                      VCHGCAL 12
7006 15         IF (F(IJK).GT.EMF.AND.F(IJK).LT.EMF1) GO TO 20 VCHGCAL 13
7007 16         IF (F(IJK).GE.EMF1) GO TO 10 VCHGCAL 14
7008 17         VCHG=F(IJK)                  VCHGCAL 15
7009 18         F(IJK)=0.0                      VCHGCAL 16
7010 19         GO TO 20                      VCHGCAL 17
7011 20         10 CONTINUE                   VCHGCAL 18
7012 21         VCHG=-(1.0-F(IJK))            VCHGCAL 19
7013 22         F(IJK)=1.0                      VCHGCAL 20
7014 23         20 CONTINUE                   VCHGCAL 21
7015 24         VCHGT=VCHGT+VCHG*DELX(I)*DELY(J)/RRI(I)*DELZ(K)*AC(IJK) VCHGCAL 22
7016 25         IF (F(IJK).LT.EMF1) GO TO 40 VCHGCAL 23
7017 26         IF (F(IPJK).LT.EMF) GO TO 30 VCHGCAL 24
7018 27         IF (F(IMJK).LT.EMF) GO TO 30 VCHGCAL 25
7019 28         IF (F(IUPK).LT.EMF) GO TO 30 VCHGCAL 26
7020 29         IF (F(IUMK).LT.EMF) GO TO 30 VCHGCAL 27
7021 30         IF (F(IUPK).LT.EMF) GO TO 30 VCHGCAL 28
7022 31         IF (F(IUMK).LT.EMF) GO TO 30 VCHGCAL 29
7023 32         GO TO 40                      VCHGCAL 30
7024 33         30 F(IJK)=F(IJK)-1.1*EMF        VCHGCAL 31
7025 34         VCHG=1.1*EMF                  VCHGCAL 32
7026 35         VCHGT=VCHGT+VCHG*DELX(I)*DELY(J)/RRI(I)*DELZ(K)*AC(IJK) VCHGCAL 33
7027 36         40 CONTINUE                   VCHGCAL 34
7028 37 C
7029 38         RETURN                      VCHGCAL 35
7030 39         END                         VCHGCAL 36
7031 1 *DK VFCONV                           VFCONV  1
7032 2          SUBROUTINE VFCONV             VFCONV  2
7033 3 *CA SLCOM1                            VFCONV  3
7034 4 C
7035 5 C * * CONVECT VOLUME OF FLUID FUNCTION F VFCONV  4
7036 6 C
7037 7          DATA FLGCS, KOUNT /0.0,0/       VFCONV  5
7038 8          IF (CYCLE.LT.1) GO TO 100      VFCONV  6
7039 9          KOUNT=0                      VFCONV  7
7040 10         FLGCS=FLGCS+FLGC              VFCONV  8
                                         VFCONV  9
                                         VFCONV 10

```

```

7041 11      FLGC=0.0          VFCONV 11
7042 12      DO 90 K=1,KM1    VFCONV 12
7043 13      DO 90 J=1,JM1    VFCONV 13
7044 14      DO 90 I=1,IM1    VFCONV 14
7045 15      CALL CALCijk    VFCONV 15
7046 16      VX=U(IJK)*DELT  VFCONV 16
7047 17      VY=V(IJK)*DELT  VFCONV 17
7048 18      VZ=W(IJK)*DELT  VFCONV 18
7049 19      ABVX=ABS(VX)    VFCONV 19
7050 20      ABVY=ABS(VY)    VFCONV 20
7051 21      ABVZ=ABS(VZ)    VFCONV 21
7052 22      IF (NF(IJK).GT.7) GO TO 20  VFCONV 22
7053 23      IF (BETA(IJK).LT.0.0) GO TO 20  VFCONV 23
7054 24      IF (ABVX.LT.0.9*DELX(I).AND.ABVY.LT.0.9*DELY(J)/RRI(I).AND.ABVZ.LT
7055 25      1.0*DELZ(K)) GO TO 20  VFCONV 24
7056 26      FLGC=1.0          VFCONV 25
7057 27      WRITE (9,130) I,J,K,U(IJK),V(IJK),W(IJK),T,DELT,CYCLE,ITER  VFCONV 26
7058 28      WRITE (12,130) I,J,K,U(IJK),V(IJK),W(IJK),T,DELT,CYCLE,ITER  VFCONV 27
7059 29      IF (FLGCS.LT.100.0) GO TO 10  VFCONV 28
7060 30      WRITE (12,110) FLGCS  VFCONV 29
7061 31      WRITE (9,110) FLGCS  VFCONV 30
7062 32      CALL EXIT        VFCONV 31
7063 33      10 CONTINUE       VFCONV 32
7064 34      KOUNT=KOUNT+1    VFCONV 33
7065 35      IF (KOUNT.LT.20) GO TO 20  VFCONV 34
7066 36      WRITE (12,120) KOUNT  VFCONV 35
7067 37      WRITE (9,120) KOUNT  VFCONV 36
7068 38      CALL EXIT        VFCONV 37
7069 39      20 IF (BETA(IJK).LE.0.0) GO TO 90  VFCONV 38
7070 40      IF (BETA(IPJK).LE.0.0) GO TO 40  VFCONV 39
7071 41 C
7072 42 C * * CONVECT F IN X-DIRECTION  VFCONV 40
7073 43 C
7074 44      IA=I+1          VFCONV 41
7075 45      ID=I            VFCONV 42
7076 46      IAJK=IPJK       VFCONV 43
7077 47      IDJK=IJK         VFCONV 44
7078 48      IDM=MAXO(I-1,1)  VFCONV 45
7079 49      RB=AR(IJK)*(1.0-CYL+CYL*X(I))  VFCONV 46
7080 50      RA=AC(IPJK)*(1.0-CYL+CYL*XI(I+1))  VFCONV 47
7081 51      RD=AC(IJK)*(1.0-CYL+CYL*XI(I))  VFCONV 48
7082 52      IF (VX.GE.0.0) GO TO 30  VFCONV 49
7083 53      IA=I            VFCONV 50
7084 54      ID=I            VFCONV 51
7085 55      IAJK=IJK         VFCONV 52
7086 56      IDJK=IPJK       VFCONV 53
7087 57      IDM=MINO(I+2,IMAX)  VFCONV 54
7088 58      RA=AC(IJK)*(1.0-CYL+CYL*XI(I))  VFCONV 55
7089 59      RD=AC(IPJK)*(1.0-CYL+CYL*XI(I+1))  VFCONV 56
7090 60      30 CONTINUE       VFCONV 57
7091 61      IAD=IA          VFCONV 58
7092 62      IDMJK=NQ*(II5*(K-1)+IMAX*(J-1)+(IDM-1))+1  VFCONV 59
7093 63      IF (NF(IDJK).EQ.3.OR.NF(IDJK).EQ.4) IAD=ID  VFCONV 60
7094 64      IF (NF(IDJK).EQ.5.OR.NF(IDJK).EQ.6) IAD=ID  VFCONV 61
7095 65      IF (FN(IAJK).LT.EMF.OR.FN(IDMJK).LT.EMF) IAD=IA  VFCONV 62
7096 66      IADJK=NQ*(II5*(K-1)+IMAX*(J-1)+(IAD-1))+1  VFCONV 63
7097 67      FDM=AMAX1(FN(IDMJK),FN(IDJK))  VFCONV 64
7098 68      IF (F(IDMJK).LT.EMF.AND.F(IAJK).LT.EMF) FDM=AMAX1(FDM,0.10)  VFCONV 65
7099 69      IF (BETA(IDMJK).LT.0.0) FDM=1.0  VFCONV 66
7100 70      FX1=FN(IADJK)*ABVX+AMAX1((FDM-FN(IADJK))*ABVX-(FDM-FN(IDJK))*DELX
7101 71      1 (ID),O.O)  VFCONV 67
7102 72      FX=AMIN1(FX1,FN(IDJK)*DELX(ID))  VFCONV 68
7103 73      F(IDJK)=F(IDJK)-FX*RDX(ID)*(RB/RD)  VFCONV 69
7104 74      F(IAJK)=F(IAJK)+FX*RDX(IA)*(RB/RA)  VFCONV 70
7105 75      40 IF (BETA(IPJK).LE.0.0) GO TO 60  VFCONV 71
7106 76 C
7107 77 C * * CONVECT F IN Y-DIRECTION  VFCONV 72
7108 78 C
7109 79      JA=J+1          VFCONV 73
7110 80      JD=J            VFCONV 74
7111 81      IAJK=IPJK       VFCONV 75
7112 82      IDJK=IJK         VFCONV 76
7113 83      JDM=MAXO(J-1,1)  VFCONV 77
7114 84      RB=ABK(IJK)      VFCONV 78
7115 85      RA=AC(IPJK)     VFCONV 79
7116 86      RD=AC(IJK)       VFCONV 80
7117 87      IF (VY.GE.0.0) GO TD 50  VFCONV 81
7118 88      JA=J            VFCONV 82
7119 89      JD=J+1          VFCONV 83
7120 90      IAJK=IJK         VFCONV 84

```

```

7121 91      IJDK=IJKP
7122 92      JDM=MINO(J+2,JMAX)
7123 93      RA=AC(IJK)
7124 94      RD=AC(IJKP)
7125 95      50 CONTINUE
7126 96      JAD=JA
7127 97      IJDMK=NQ*(II5*(K-1)+IMAX*(JDM-1)+(I-1))+1
7128 98      IF (NF(IJDK).EQ.5.OR.NF(IJDK).EQ.6) JAD=JD
7129 99      IF (NF(IJDK).EQ.1.OR.NF(IJDK).EQ.2) JAD=JD
7130 100     IF (FN(IJAK).LT.EMF.OR.FN(IJDMK).LT.EMF) JAD=JA
7131 101     IJADK=NQ*(II5*(K-1)+IMAX*(JAD-1)+(I-1))+1
7132 102     FDM=AMAX1(FN(IJDMK),FN(IJDK))
7133 103     IF (F(IJDMK).LT.EMF.AND.F(IJAK).LT.EMF) FDM=AMAX1(FDM,O.10)
7134 104     IF (BETA(IJDMK).LT.O.O) FDM=1.0
7135 105     FY1=FN(IJADK)*ABVY+AMAX1((FDM-FN(IJADK))*ABVY-(FDM-FN(IJDK))*DELY
7136 106     1 (JD)/RRI(I),O.O)
7137 107     FY=AMIN1(FY1,FN(IJDK)*DELY(JD)/RRI(I))
7138 108     F(IJDK)=F(IJDK)-FY*RDY(JD)*RRI(I)*(RB/RD)
7139 109     F(IJAK)=F(IJAK)+FY*RDY(JA)*RRI(I)*(RB/RA)
7140 110 C    VFCONV 110
7141 111 C   * * CONVECT F IN Z-DIRECTION
7142 112 C   VFCONV 112
7143 113     60 IF (BETA(IJKP).LE.O.O) GO TO 80
7144 114     KA=K+1
7145 115     KD=K
7146 116     IJKA=IJKP
7147 117     IJKD=IJK
7148 118     KDM=MAXO(K-1,1)
7149 119     RB=AT(IJK)
7150 120     RA=AC(IJKP)
7151 121     RD=AC(IJK)
7152 122     IF (VZ.GE.O.O) GO TO 70
7153 123     KA=K
7154 124     KD=K+1
7155 125     IJKA=IJK
7156 126     IJKD=IJKP
7157 127     KDM=MINO(K+2,KMAX)
7158 128     RA=AC(IJK)
7159 129     RD=AC(IJKP)
7160 130     70 CONTINUE
7161 131     KAD=KA
7162 132     IJKDM=NQ*(II5*(KDM-1)+IMAX*(J-1)+(I-1))+1
7163 133     IF (NF(IJKD).EQ.1.OR.NF(IJKD).EQ.2) KAD=KD
7164 134     IF (NF(IJKD).EQ.3.OR.NF(IJKD).EQ.4) KAD=KD
7165 135     IF (FN(IJKA).LT.EMF.OR.FN(IJKDM).LT.EMF) KAD=KA
7166 136     IJKAD=NQ*(II5*(KAD-1)+IMAX*(J-1)+(I-1))+1
7167 137     FDM=AMAX1(FN(IJKDM),FN(IJKD))
7168 138     IF (F(IJKDM).LT.EMF.AND.F(IJKA).LT.EMF) FDM=AMAX1(FDM,O.10)
7169 139     IF (BETA(IJKDM).LT.O.O) FDM=1.0
7170 140     FZ1=FN(IJKAD)*ABVZ+AMAX1((FDM-FN(IJKAD))*ABVZ-(FDM-FN(IJKD))*DELZ
7171 141     1 (KD),O.O)
7172 142     FZ=AMIN1(FZ1,FN(IJKD)*DELZ(KD))
7173 143     F(IJKD)=F(IJKD)-FZ*RDZ(KD)*(RB/RD)
7174 144     F(IJKA)=F(IJKA)+FZ*RDZ(KA)*(RB/RA)
7175 145     80 IF (NF(IJK).NE.O) GO TO 90
7176 146     IF (IDEFM.GT.O) GO TO 90
7177 147     IF (BETA(IJK).EQ.1.O) GO TO 90
7178 148     F(IJK)=F(IJK)+DELT*FN(IJK)*D(IJK)
7179 149     90 CONTINUE
7180 150 C    VFCONV 150
7181 151 C   * * CALCULATE TOTAL VOLUME CHANGE
7182 152 C    VFCONV 152
7183 153     100 CALL VCHGCAL
7184 154 C    VFCONV 154
7185 155     RETURN
7186 156 C    VFCONV 156
7187 157     .110 FORMAT (*      VFCONV - TOO MANY LARGE VELOCITIES  FLGCS=*,F10.5)
7188 158     120 FORMAT (*      VFCONV - TOO MANY LARGE VELOCITIES  KOUNT=*,F10.5)
7189 159     130 FORMAT (*      VFCONV - VELOCITIES TOO LARGE    */3X,3I3,5F12.5,2I5)
7190 160     END
7191 1 *DK WRTAPE
7192 2      SUBROUTINE WRTAPE
7193 3 *CA SLCOM1
7194 4 C
7195 5 C   * * WRITE MAGMETIC TAPE
7196 6 C
7197 7      WRITE (9,10) NUMTD,T,CYCLE
7198 8      IF (LPR.GT.O) WRITE (12,10) NUMTD,T,CYCLE
7199 9      WRITE (59,10) NUMTD,T,CYCLE
7200 10     TWTD=T+TDDT
                                         VFCONV 91
                                         VFCONV 92
                                         VFCONV 93
                                         VFCONV 94
                                         VFCONV 95
                                         VFCONV 96
                                         VFCONV 97
                                         VFCONV 98
                                         VFCONV 99
                                         VFCONV 100
                                         VFCONV 101
                                         VFCONV 102
                                         VFCONV 103
                                         VFCONV 104
                                         VFCONV 105
                                         VFCONV 106
                                         VFCONV 107
                                         VFCONV 108
                                         VFCONV 109
                                         VFCONV 110
                                         VFCONV 111
                                         VFCONV 112
                                         VFCONV 113
                                         VFCONV 114
                                         VFCONV 115
                                         VFCONV 116
                                         VFCONV 117
                                         VFCONV 118
                                         VFCONV 119
                                         VFCONV 120
                                         VFCONV 121
                                         VFCONV 122
                                         VFCONV 123
                                         VFCONV 124
                                         VFCONV 125
                                         VFCONV 126
                                         VFCONV 127
                                         VFCONV 128
                                         VFCONV 129
                                         VFCONV 130
                                         VFCONV 131
                                         VFCONV 132
                                         VFCONV 133
                                         VFCONV 134
                                         VFCONV 135
                                         VFCONV 136
                                         VFCONV 137
                                         VFCONV 138
                                         VFCONV 139
                                         VFCONV 140
                                         VFCONV 141
                                         VFCONV 142
                                         VFCONV 143
                                         VFCONV 144
                                         VFCONV 145
                                         VFCONV 146
                                         VFCONV 147
                                         VFCONV 148
                                         VFCONV 149
                                         VFCONV 150
                                         VFCONV 151
                                         VFCONV 152
                                         VFCONV 153
                                         VFCONV 154
                                         VFCONV 155
                                         VFCONV 156
                                         VFCONV 157
                                         VFCONV 158
                                         VFCONV 159
                                         VFCONV 160
                                         WRTAPE 1
                                         WRTAPE 2
                                         WRTAPE 3
                                         WRTAPE 4
                                         WRTAPE 5
                                         WRTAPE 6
                                         WRTAPE 7
                                         WRTAPE 8
                                         WRTAPE 9
                                         WRTAPE 10

```

```

7201 11      WRITE (B) (AA(N),N=1,NCR1)          WRTAPE 11
7202 12      WRITE (B) (BASC(N),N=1,NCR2)        WRTAPE 12
7203 13      WRITE (B) (BASC1(N),N=1,NCR2)       WRTAPE 13
7204 14      WRITE (B) (BASC2(N),N=1,NCR2)       WRTAPE 14
7205 15      WRITE (B) (BASC3(N),N=1,NCR2)       WRTAPE 15
7206 16      NUMTD=NUMTD+1                      WRTAPE 16
7207 17 C
7208 18      RETURN                            WRTAPE 17
7209 19 C
7210 20      10 FORMAT (12HO TAPE DUMP=,I3,7H AT T=,1PE12.5,7H CYCLE=,I5) WRTAPE 20
7211 21      END                               WRTAPE 21
7212 1 *DK XINDF                           XINDF   1
7213 2           SUBROUTINE XINDF             XINDF   2
7214 3 *CA SLCOM1                           XINDF   3
7215 4 C
7216 5 C * * SET ARRAYS TO INDEFINITE XINDF   4
7217 6 C
7218 7           ILIM=II5*KMAX*NQ            XINDF   5
7219 8           DATA XINDEF /17770000000000777777B/ XINDF   6
7220 9           IUP=IBASC                           XINDF   7
7221 10          DO 10 I=1,IUP                  XINDF   8
7222 11          BASC(I)=XINDEF                XINDF   9
7223 12          BASC2(I)=XINDEF               XINDF  10
7224 13          10 BASC3(I)=XINDEF             XINDF  11
7225 14          NCR1=LOCF(ZLAST)-LOCF(AA)+1    XINDF  12
7226 15          NCR2=LOCF(BASC(ILIM))-LOCF(BASC)+1 XINDF  13
7227 16 C
7228 17      RETURN                            XINDF  14
7229 18      END                               XINDF  15
7230 1 *DK DRAWQ                           DRAWQ   1
7231 2           SUBROUTINE DRAWQ              DRAWQ   2
7232 3 *CA SLCOM1                           DRAWQ   3
7233 4 C
7234 5           PARAMETER (IBAR2Q=2*IBAR2-2)  DRAWQ   4
7235 6           COMMON BETAQ(IBAR2Q,KBAR2), FQ(IBAR2Q,KBAR2), ACQ(IBAR2Q,KBAR2), DRAWQ   5
7236 7           1 ARQ(IBAR2Q,KBAR2), ATQ(IBAR2Q,KBAR2), UQ(IBAR2Q,KBAR2), VQ(IBAR2Q DRAWQ   6
7237 8           2 ,KBAR2), XQ(IBAR2Q), XIQ(IBAR2Q), YQ(KBAR2), YJQ(KBAR2), IM1Q, DRAWQ   7
7238 9           3 JM1Q, JBAR2Q, JMAXQ, IMAXQ, SF, XSHFT, YSHFT, DELXQ(IBAR2Q), DRAWQ   8
7239 10          4 DELYQ(KBAR2), XMINQ, XMAXQ, YMINQ, YMAXQ                   DRAWQ   9
7240 11 C
7241 12 C           MOVE QUANITIES FROM 3D MESH TO 2D PLOT MESH DRAWQ  10
7242 13 C
7243 14 C           PLOTTING J=2 AND J=JM1 PLANES FOR 180 DEG PROBLEM DRAWQ  11
7244 15 C           PLOTTING J=2 AND J=JDP(2) PLANES FOR 360 DEG PROBLEM DRAWQ  12
7245 16 C
7246 17          IM1Q=IBAR2Q-1                  DRAWQ  13
7247 18          JM1Q=KM1                     DRAWQ  14
7248 19          JBAR2Q=KBAR2                 DRAWQ  15
7249 20          JMAXQ=KMAX                   DRAWQ  16
7250 21          IMAXQ=IBAR2Q                 DRAWQ  17
7251 22 C
7252 23          J=2                         DRAWQ  18
7253 24          DO 10 L=IMAX,IBAR2Q          DRAWQ  19
7254 25          DO 10 K=1,KMAX                 DRAWQ  20
7255 26          I=L-IMAX+2                  DRAWQ  21
7256 27          CALL IJKONLY                DRAWQ  22
7257 28          BETAQ(L,K)=BETA(IJK)           DRAWQ  23
7258 29          FQ(L,K)=F(IJK)                 DRAWQ  24
7259 30          ARQ(L,K)=AR(IJK)               DRAWQ  25
7260 31          ATQ(L,K)=AT(IJK)               DRAWQ  26
7261 32          ACQ(L,K)=AC(IJK)               DRAWQ  27
7262 33          UQ(L,K)=U(IJK)                 DRAWQ  28
7263 34          VQ(L,K)=W(IJK)                 DRAWQ  29
7264 35          10 CONTINUE                DRAWQ  30
7265 36 C
7266 37          J=JDP(2)                   DRAWQ  31
7267 38 C
7268 39          DO 20 L=1,IM1                DRAWQ  32
7269 40          DO 20 K=1,KMAX               DRAWQ  33
7270 41          I=IMAX-L+1                  DRAWQ  34
7271 42          CALL IJKAJCT                DRAWQ  35
7272 43          BETAQ(L,K)=BETA(IJK)           DRAWQ  36
7273 44          FQ(L,K)=F(IJK)                 DRAWQ  37
7274 45          ARQ(L,K)=AR(IMUK)              DRAWQ  38
7275 46          IF (L.EQ.IM1) ARQ(L,K)=1.0   DRAWQ  39
7276 47          ATQ(L,K)=AT(IJK)               DRAWQ  40
7277 48          ACQ(L,K)=AC(IJK)               DRAWQ  41
7278 49          UQ(L,K)=-U(IMJK)              DRAWQ  42
7279 50          IF (L.EQ.IM1) UQ(L,K)=0.0    DRAWQ  43
7280 51          VQ(L,K)=W(IJK)                 DRAWQ  44

```

```

7281 52      20 CONTINUE
7282 53 C
7283 54      DO 30 L=IMAX,IBAR20
7284 55      I=L-IM1+1
7285 56      XQ(L)=X(I)+X(IM1)
7286 57      DELXQ(L)=DELX(I)
7287 58      30 CONTINUE
7288 59 C
7289 60      DO 40 L=1,IM1
7290 61      I=IMAX-L
7291 62      XQ(L)=X(IM1)-X(I)
7292 63      DELXQ(L)=DELX(I)
7293 64      40 CONTINUE
7294 65 C
7295 66      DO 50 L=2,IBAR20
7296 67      XIQ(L)=(XQ(L)+XQ(L-1))*0.5
7297 68      50 CONTINUE
7298 69      XIQ(1)=-XIQ(2)
7299 70 C
7300 71      DO 60 K=1,KMAX
7301 72      YQ(K)=Z(K)
7302 73      YQ(K)=ZK(K)
7303 74      DELYQ(K)=DELZ(K)
7304 75      60 CONTINUE
7305 76 C
7306 77 C    +++ SET CONSTANT TERMS FOR PLOTTING
7307 78 C
7308 79      XMINQ=XQ(1)
7309 80      XMAXQ=XQ(IM1Q)
7310 81      YMINQ=YQ(1)
7311 82      YMAXQ=YQ(JM1Q)
7312 83      D1=XMAXQ-XMINQ
7313 84      D2=YMAXQ-YMINQ
7314 85      D3=AMAX1(D1,D2)
7315 86      SF=1.0/D3
7316 87      XSHFT=0.5*(1.0-D1*SF)
7317 88      YSHFT=0.5*(1.0-D2*SF)
7318 89 C
7319 90 C    * * DETERMINE MAXIMUM VECTOR LENGTH
7320 91 C
7321 92      VMAX=0.0
7322 93      VMAXJ=0.0
7323 94      DO 70 K=2,KM1
7324 95      DO 70 J=2,JM1
7325 96      DO 70 I=2,IM1
7326 97      CALL IJKAJCT
7327 98      IF (F(IJK).LT.-0.5) GO TO 70
7328 99      IF (AC(IJK).LT.0.5) GO TO 70
7329 100     IF (J.NE.2.AND.J.NE.JOP(2)) GO TO 70
7330 101     UAVE=0.5*(U(IJK)+U(IMJK))
7331 102     VAVE=0.5*(V(IJK)+V(IJMJK))
7332 103     WAVE=0.5*(W(IJK)+W(IJMJK))
7333 104     VMAX=AMAX1(VMAX,ABS(UAVE),ABS(VAVE),ABS(WAVE))
7334 105     VMAXJ=AMAX1(VMAXJ,ABS(UAVE),ABS(WAVE))
7335 106     70 CONTINUE
7336 107 C
7337 108 C    * * SCALE VECTOR LENGTH
7338 109 C
7339 110     VELMX1=VELMX*DELMN/(VMAXJ+1.0E-10)
7340 111 C
7341 112 C    ***
7342 113 C    +++ DRAW VELOCITY VECTOR PLOT
7343 114 C    ***
7344 115     CALL ADV (1)
7345 116     CALL LINCNT (1)
7346 117 C     WRITE(12,200) T,CYCLE,NAME
7347 118     CALL FRAMEQ (XMINQ,XMAXQ,YMAXQ,YMINQ)
7348 119     CALL DRWOBSSQ
7349 120     DO 80 I=2,IM1Q
7350 121     ACELH=0.5*CYL+1.0-CYL
7351 122     DO 80 J=2,JM1Q
7352 123     IF (FO(I,J).LT.-0.5) GO TO 80
7353 124     IF (ACQ(I,J).LT.ACELH) GO TO 80
7354 125     XXCC=XIQ(I)
7355 126     YYCC=0.5*(YQ(J)+YQ(J-1))
7356 127     UMPL=0.0
7357 128     VMPL=0.0
7358 129     IF (ARQ(I,J).GT.EM6) UMPL=1.0
7359 130     IF (ARQ(I-1,J).GT.EM6) UMPL=1.0/(1.0+UMPL)
7360 131     IF (ATQ(I,J).GT.EM6) VMPL=1.0

```

```

7361 132      IF (ATQ(I,J-1).GT.EM6) VMPL=1.0/(1.0+VMPL)          DRAWQ   132
7362 133      UPLT=UMPL*(UQ(I-1,J)+UQ(I,J))                      DRAWQ   133
7363 134      VPLT=VMPL*(VQ(I,J-1)+VQ(I,J))                      DRAWQ   134
7364 135      UVEC=UPLT*VELMX1+XXCC                           DRAWQ   135
7365 136      VVEC=VPLT*VELMX1+YYCC                           DRAWQ   136
7366 137      CALL DRWVECA (XXCC,YYCC,UVEC,VVEC,1)             DRAWQ   137
7367 138      CALL PLPTP (XXCC,YYCC,53B,1)                      DRAWQ   138
7368 139      BO CONTINUE                                     DRAWQ   139
7369 140 C
7370 141 C     +++ DRAW FREE SURFACE
7371 142 C
7372 143      FPL=0.5
7373 144      DO 100 I=2,IM1Q                                DRAWQ   144
7374 145      DO 100 J=2,JM1Q                                DRAWQ   145
7375 146      IF (BETAQ(I,J).LT.0.0) GO TO 100               DRAWQ   146
7376 147      FATR=0.25*(FQ(I,J)+FQ(I+1,J)+FQ(I,J+1)+FQ(I+1,J+1)) DRAWQ   147
7377 148      FXTR=0.5*(FQ(I+1,J+1)+FQ(I+1,J)-FQ(I,J+1)-FQ(I,J))/(XIQ(I+1)-XIQ(I) DRAWQ   148
7378 149      1 )
7379 150      FYTR=0.5*(FQ(I,J+1)+FQ(I+1,J+1)-FQ(I,J)-FQ(I+1,J))/(YJQ(J+1)-YJQ(J) DRAWQ   149
7380 151      1 )
7381 152      FTRS=FXTR**2+FYTR**2                         DRAWQ   152
7382 153      IF (FTRS.EQ.0.0) FTRS=1.OE+10                  DRAWQ   153
7383 154      XTR=0.5*(XIQ(I+1)+XIQ(I))+(FPL-FATR)*FXTR/FTRS    DRAWQ   154
7384 155      XTR=AMAX1(XTR,XIQ(I))                      DRAWQ   155
7385 156      XTR=AMIN1(XTR,XIQ(I+1))                      DRAWQ   156
7386 157      YTR=0.5*(YJQ(J)+YJQ(J+1))+(FPL-FATR)*FYTR/FTRS    DRAWQ   157
7387 158      YTR=AMAX1(YTR,YJQ(J))                      DRAWQ   158
7388 159      YTR=AMIN1(YTR,YJQ(J+1))                      DRAWQ   159
7389 160      IF (FQ(I,J).GT.0.5.AND.FQ(I+1,J).GT.0.5) GO TO 90  DRAWQ   160
7390 161      IF (FQ(I,J).LT.0.5.AND.FQ(I+1,J).LT.0.5) GO TO 90  DRAWQ   161
7391 162      FABR=0.25*(FQ(I,J)+FQ(I+1,J)+FQ(I,J-1)+FQ(I+1,J-1)) DRAWQ   162
7392 163      FBXR=0.5*(FQ(I+1,J)+FQ(I+1,J-1)-FQ(I,J)-FQ(I,J-1))/(XIQ(I+1)-XIQ(I) DRAWQ   163
7393 164      1 )
7394 165      FYBR=0.5*(FQ(I,J)+FQ(I+1,J)-FQ(I,J-1)-FQ(I+1,J-1))/(YJQ(J)-YJQ(J-1) DRAWQ   165
7395 166      1 )
7396 167      FBRS=FXBR**2+FYBR**2                         DRAWQ   167
7397 168      IF (FBRS.EQ.0.0) FBRS=1.OE+10                  DRAWQ   168
7398 169      XXBR=0.5*(XIQ(I+1)+XIQ(I))+(FPL-FABR)*FXBR/FBRS    DRAWQ   169
7399 170      XXBR=AMAX1(XXBR,XIQ(I))                      DRAWQ   170
7400 171      XXBR=AMIN1(XXBR,XIQ(I+1))                      DRAWQ   171
7401 172      YBR=0.5*(YJQ(J)+YJQ(J-1))+(FPL-FABR)*FYBR/FBRS    DRAWQ   172
7402 173      YBR=AMAX1(YBR,YJQ(J-1))                      DRAWQ   173
7403 174      YBR=AMIN1(YBR,YJQ(J))                      DRAWQ   174
7404 175      CALL DRWVEC (XXBR,YBR,XTR,YTR,1)              DRAWQ   175
7405 176      90 CONTINUE                                     DRAWQ   176
7406 177      IF (FQ(I,J).GT.0.5.AND.FQ(I,J+1).GT.0.5) GO TO 100  DRAWQ   177
7407 178      IF (FQ(I,J).LT.0.5.AND.FQ(I,J+1).LT.0.5) GO TO 100  DRAWQ   178
7408 179      FTL=0.25*(FQ(I,J)+FQ(I,J+1)+FQ(I-1,J)+FQ(I-1,J+1)) DRAWQ   179
7409 180      FXTL=0.5*(FQ(I,J+1)+FQ(I,J)-FQ(I-1,J+1)-FQ(I-1,J))/(XIQ(I)-XIQ(I-1) DRAWQ   180
7410 181      1 )
7411 182      FYTL=0.5*(FQ(I-1,J+1)+FQ(I,J+1)-FQ(I-1,J)-FQ(I,J))/(YJQ(J+1)-YJQ(J) DRAWQ   182
7412 183      1 )
7413 184      FTLS=FXTL**2+FYTL**2                         DRAWQ   184
7414 185      IF (FTLS.EQ.0.0) FTLS=1.OE+10                  DRAWQ   185
7415 186      XTL=0.5*(XIQ(I-1)+XIQ(I))+(FPL-FATL)*FXTL/FTLS    DRAWQ   186
7416 187      XTL=AMAX1(XTL,XIQ(I-1))                      DRAWQ   187
7417 188      XTL=AMIN1(XTL,XIQ(I))                      DRAWQ   188
7418 189      YTL=0.5*(YJQ(J)+YJQ(J+1))+(FPL-FATL)*FYTL/FTLS    DRAWQ   189
7419 190      YTL=AMAX1(YTL,YJQ(J))                      DRAWQ   190
7420 191      YTL=AMIN1(YTL,YJQ(J+1))                      DRAWQ   191
7421 192      CALL DRWVEC (XTL,YTL,XTR,YTR,1)              DRAWQ   192
7422 193      100 CONTINUE                                     DRAWQ   193
7423 194 C
7424 195      CALL LINCNT (59)
7425 196      WRITE (12,120) VMAX,VMAXJ
7426 197      WRITE (12,130) CYCLE,T,(NAME(I),I=1,5),JNM
7427 198      WRITE (12,110) 2,IM1,2,KM1
7428 199      RETURN
7429 200 C
7430 201      110 FORMAT (21H CONSTANT J=2 AND JM1,17H SURFACE I=,I3,3H TD,I3,
7431 202      1 7H K=,I3,3H TD,I3)                          DRAWQ   201
7432 203      120 FORMAT (1BH VELOCITY VECTORS,4X,7H VMAX=,1PE12.5,4X,6HVMAXJ=,E12 DRAWQ   203
7433 204      1 .5)
7434 205      130 FORMAT (8H CYCLE=,I5,4H T=,1PE12.5,5X,5A8,1X,AB)
7435 206      END
7436 1 *DK DRWOBSQ
7437 2       SUBROUTINE DRWOBSQ
7438 3 *CA SLCOM1
7439 4 C
7440 5       PARAMETER (IBAR2Q=2*IBAR2-2)

```

```

7441   6      COMMON BETAO(IBAR2Q,KBAR2), FQ(IBAR2Q,KBAR2), ACQ(IBAR2Q,KBAR2),
7442   7      1 ARQ(IBAR2Q,KBAR2), ATQ(IBAR2Q,KBAR2), UQ(IBAR2Q,KBAR2), VQ(IBAR2Q
7443   8      2 ,KBAR2), XQ(IBAR2Q), XIQ(IBAR2Q), YQ(KBAR2), YUQ(KBAR2), IM1Q,
7444   9      3 JM1Q, JBAR2Q, JMAXQ, IMAXQ, SF, XSHFT, YSHFT, DELXQ(IBAR2Q),
7445  10      4 DELYQ(KBAR2), XMINQ, XMAXQ, YMINQ, YMAXQ
7446  11 C
7447  12 C +++ DRAW AROUND ALL OBSTACLES
7448  13 C
7449  14      DO 170 I=2,IM1Q
7450  15      ATR=1.0-EM6
7451  16      ATL=1.0-EM6
7452  17      IF (I.EQ.2.) ATL=-EM6
7453  18      ATC=1.0-EM6
7454  19      DO 170 J=2,JM1Q
7455  20      IF (ACQ(I,J).LT.EM6) GO TO 170
7456  21      AFR=1.0
7457  22      AFT=1.0
7458  23      AFL=1.0
7459  24      AFB=1.0
7460  25      IF (ARQ(I,J).LT.ATR) AFR=ARQ(I,J)/ATR
7461  26      IF (ATQ(I,J).LT.ATC) AFT=ATQ(I,J)/ATC
7462  27      IF (ARQ(I-1,J).LT.ATL) AFL=ARQ(I-1,J)/ATL
7463  28      IF (ATQ(I,J-1).LT.ATC) AFB=ATQ(I,J-1)/ATC
7464  29      IF (ACQ(I,J).GE.ATC) GO TO 120
7465  30      IF (I.EQ.2) AFL=AFR-EM6
7466  31      IF (I.EQ.IM1Q) AFR=AFL-EM6
7467  32      IF (J.EQ.2) AFB=AFT-EM6
7468  33      IF (J.EQ.JM1Q) AFT=AFB-EM6
7469  34      IF ((AFT+AFB).LT.EM6.OR.(AFL+AFR).LT.EM6) GO TO 170
7470  35      M=1
7471  36      AMN=AFB+AFT
7472  37      IF ((AFR+AFT).GT.AMN) GO TO 10
7473  38      M=2
7474  39      AMN=AFT+AFT      *
7475  40      10 IF ((AFT+AFL).GT.AMN) GO TO 20
7476  41      M=3
7477  42      AMN=AFT+AFL
7478  43      20 IF ((AFL+AFB).GT.AMN) GO TO 30
7479  44      M=4
7480  45      30 GO TO (40,60,80,100), M
7481  46      40 X1=XQ(I-1)+AFT*DELXQ(I)
7482  47      Y1=YQ(J)
7483  48      IF (AFT.LT.1.0) GO TO 50
7484  49      Y1=Y1-AFR*DELYQ(J)
7485  50      50 X2=XQ(I-1)
7486  51      ZY2=YQ(J)-AFL*DELYQ(J)
7487  52      IF (AFL.LT.1.0) GO TO 160
7488  53      X2=X2+AFB*DELXQ(I)
7489  54      GO TO 160
7490  55      60 X1=XQ(I-1)
7491  56      Y1=YQ(J-1)+AFL*DELYQ(J)
7492  57      IF (AFL.LT.1.0) GO TO 70
7493  58      X1=X1+AFT*DELXQ(I)
7494  59      70 X2=XQ(I-1)+AFB*DELXQ(I)
7495  60      ZY2=YQ(J-1)
7496  61      IF (AFB.LT.1.0) GO TO 160
7497  62      ZY2=ZY2+AFL*DELYQ(J)
7498  63      GO TO 160
7499  64      80 X1=XQ(I)-AFB*DELXQ(I)
7500  65      Y1=YQ(J-1)
7501  66      IF (AFB.LT.1.0) GO TO 90
7502  67      Y1=Y1+AFL*DELYQ(J)
7503  68      90 X2=XQ(I)
7504  69      ZY2=YQ(J-1)+AFR*DELYQ(J)
7505  70      IF (AFR.LT.1.0) GO TO 160
7506  71      X2=X2-AFT*DELXQ(I)
7507  72      GO TO 160
7508  73      100 X1=XQ(I)
7509  74      Y1=YQ(J)-AFR*DELYQ(J)
7510  75      IF (AFR.LT.1.0) GO TO 110
7511  76      X1=X1-AFB*DELXQ(I)
7512  77      110 X2=XQ(I)-AFT*DELXQ(I)
7513  78      ZY2=YQ(J)
7514  79      IF (AFT.LT.1.0) GO TO 160
7515  80      ZY2=ZY2-AFL*DELYQ(J)
7516  81      GO TO 160
7517  82      120 IF (AFR.GT.EM6) GO TO 130
7518  83      X1=XQ(I)
7519  84      Y1=YQ(J-1)
7520  85      X2=X1
                                         DRWOBSQ   6
                                         DRWOBSQ   7
                                         DRWOBSQ   8
                                         DRWOBSQ   9
                                         DRWOBSQ  10
                                         DRWOBSQ  11
                                         DRWOBSQ  12
                                         DRWOBSQ  13
                                         DRWOBSQ  14
                                         DRWOBSQ  15
                                         DRWOBSQ  16
                                         DRWOBSQ  17
                                         DRWOBSQ  18
                                         DRWOBSQ  19
                                         DRWOBSQ  20
                                         DRWOBSQ  21
                                         DRWOBSQ  22
                                         DRWOBSQ  23
                                         DRWOBSQ  24
                                         DRWOBSQ  25
                                         DRWOBSQ  26
                                         DRWOBSQ  27
                                         DRWOBSQ  28
                                         DRWOBSQ  29
                                         DRWOBSQ  30
                                         DRWOBSQ  31
                                         DRWOBSQ  32
                                         DRWOBSQ  33
                                         DRWOBSQ  34
                                         DRWOBSQ  35
                                         DRWOBSQ  36
                                         DRWOBSQ  37
                                         DRWOBSQ  38
                                         DRWOBSQ  39
                                         DRWOBSQ  40
                                         DRWOBSQ  41
                                         DRWOBSQ  42
                                         DRWOBSQ  43
                                         DRWOBSQ  44
                                         DRWOBSQ  45
                                         DRWOBSQ  46
                                         DRWOBSQ  47
                                         DRWOBSQ  48
                                         DRWOBSQ  49
                                         DRWOBSQ  50
                                         DRWOBSQ  51
                                         DRWOBSQ  52
                                         DRWOBSQ  53
                                         DRWOBSQ  54
                                         DRWOBSQ  55
                                         DRWOBSQ  56
                                         DRWOBSQ  57
                                         DRWOBSQ  58
                                         DRWOBSQ  59
                                         DRWOBSQ  60
                                         DRWOBSQ  61
                                         DRWOBSQ  62
                                         DRWOBSQ  63
                                         DRWOBSQ  64
                                         DRWOBSQ  65
                                         DRWOBSQ  66
                                         DRWOBSQ  67
                                         DRWOBSQ  68
                                         DRWOBSQ  69
                                         DRWOBSQ  70
                                         DRWOBSQ  71
                                         DRWOBSQ  72
                                         DRWOBSQ  73
                                         DRWOBSQ  74
                                         DRWOBSQ  75
                                         DRWOBSQ  76
                                         DRWOBSQ  77
                                         DRWOBSQ  78
                                         DRWOBSQ  79
                                         DRWOBSQ  80
                                         DRWOBSQ  81
                                         DRWOBSQ  82
                                         DRWOBSQ  83
                                         DRWOBSQ  84
                                         DRWOBSQ  85

```

```

7521  86      ZY2=YQ(J)
7522  87      CALL DRWVEC (X1,Y1,X2,ZY2,1)
7523  88  130 IF (AFT.GT.EM6) GO TO 140
7524  89      X1=XQ(I-1)
7525  90      Y1=YQ(J)
7526  91      X2=XQ(I)
7527  92      ZY2=Y1
7528  93      CALL DRWVEC (X1,Y1,X2,ZY2,1)
7529  94  140 IF (AFL.GT.EM6) GO TO 150
7530  95      X1=XQ(I-1)
7531  96      Y1=YQ(J)
7532  97      X2=X1
7533  98      ZY2=YQ(J-1)
7534  99      CALL DRWVEC (X1,Y1,X2,ZY2,1)
7535 100  150 IF (AFB.GT.EM6) GO TO 170
7536 101      X1=XQ(I-1)
7537 102      Y1=YQ(J-1)
7538 103      X2=XQ(I)
7539 104      ZY2=Y1
7540 105      CALL DRWVEC (X1,Y1,X2,ZY2,1)
7541 106  160 CALL DRWVEC (X1,Y1,X2,ZY2,1)
7542 107  170 CONTINUE
7543 108      RETURN
7544 109      END
7545  1 *DK DRWVEC
7546  2      SUBROUTINE DRWVEC (XONE,YONE,XTWO,YTWO,ISYM)
7547  3 *CA SLCOM1
7548  4      PARAMETER (IBAR2Q=2*IBAR2-2)
7549  5      COMMON BETAQ(IBAR2Q,KBAR2), FQ(IBAR2Q,KBAR2), ACQ(IBAR2Q,KBAR2),
7550  6      1 ARO(IBAR2Q,KBAR2), ATQ(IBAR2Q,KBAR2), UQ(IBAR2Q,KBAR2), VQ(IBAR2Q
7551  7      2 ,KBAR2), XQ(IBAR2Q), XIQ(IBAR2Q), YQ(KBAR2), YUQ(KBAR2), IM1Q,
7552  8      3 UM1Q, UBAR2Q, UMAXQ, IMAXQ, SF, XSHFT, YSHFT, DELXQ(IBAR2Q),
7553  9      4 DELYQ(KBAR2), XMINQ, XMAXQ, YMINQ, YMAXQ
7554 10 C
7555 11 C +++ DRAW A VECTOR
7556 12 C +++ PROVIDES A SYSTEM DEPENDANT CALL
7557 13 C
7558 14      IC=0
7559 15      X1=XONE
7560 16      Y1=YONE
7561 17      X2=XTWO
7562 18      ZY2=YTWO
7563 19      X01=(X1-XMINQ)*SF+XSHFT
7564 20      Y01=(Y1-YMINQ)*SF+YSHFT
7565 21      X02=(X2-XMINQ)*SF+XSHFT
7566 22      Y02=(ZY2-YMINQ)*SF+YSHFT
7567 23      IX1=16.+900.0*X01
7568 24      IX2=16.+900.0*X02
7569 25      IY1=16.+900.0*(1.0-Y01)
7570 26      IY2=16.+900.0*(1.0-Y02)
7571 27      CALL DRV (IX1,IY1,IX2,IY2)
7572 28      RETURN
7573 29 C
7574 30      ENTRY DRWVECA(XONE,YONE,XTWO,YTWO,ISYM)
7575 31 C
7576 32 C +++ DRAW A VECTOR WITH ARROWHEADS
7577 33 C
7578 34      IC=0
7579 35      X1=XONE
7580 36      Y1=YONE
7581 37      X2=XTWO
7582 38      ZY2=YTWO
7583 39      X01=(X1-XMINQ)*SF+XSHFT
7584 40      Y01=(Y1-YMINQ)*SF+YSHFT
7585 41      X02=(X2-XMINQ)*SF+XSHFT
7586 42      Y02=(ZY2-YMINQ)*SF+YSHFT
7587 43      IX1=16.+900.0*X01
7588 44      IX2=16.+900.0*X02
7589 45      IY1=16.+900.0*(1.0-Y01)
7590 46      IY2=16.+900.0*(1.0-Y02)
7591 47      CALL DRV (IX1,IY1,IX2,IY2)
7592 48 C      DRAW ARROWHEADS
7593 49      FIX1=IX1
7594 50      FIX2=IX2
7595 51      FIY1=IY1
7596 52      FIY2=IY2
7597 53      X34=FIX1+0.85*(FIX2-FIX1)
7598 54      Y34=FIY1+0.85*(FIY2-FIY1)
7599 55      DX234=FIX2-X34
7600 56      DY234=FIY2-Y34

```

7601	57	D234=SQRT(DX234**2+DY234**2)	DRWVEC	57
7602	58	DA234=D234*0.57735027	DRWVEC	58
7603	59 C	TAN(15 DEG)=0.57735027	DRWVEC	59
7604	60	COSB=DA234	DRWVEC	60
7605	61	SINB=0.0	DRWVEC	61
7606	62	IF (DX234.EQ.0.0) GO TO 10	DRWVEC	62
7607	63	BETA234=1.57079633-ATAN(DY234/DX234)	DRWVEC	63
7608	64	COSB=COS(BETA234)*DA234	DRWVEC	64
7609	65	SINB=SIN(BETA234)*DA234	DRWVEC	65
7610	66	10 IX3=X34-COSB	DRWVEC	66
7611	67	IX4=X34+COSB	DRWVEC	67
7612	68	IY3=Y34+SINB	DRWVEC	68
7613	69	IY4=Y34-SINB	DRWVEC	69
7614	70	CALL DRV (IX2,IY2,IX3,IY3)	DRWVEC	70
7615	71	CALL DRV (IX2,IY2,IX4,IY4)	DRWVEC	71
7616	72	RETURN	DRWVEC	72
7617	73	END	DRWVEC	73
7618	1	*DK FRAME	FRAME	1
7619	2	SUBROUTINE FRAMEQ (XXL,XXR,YYT,YYB)	FRAME	2
7620	3 C		FRAME	3
7621	4 C	+++ DRAW A FRAME AROUND THE PLOT	FRAME	4
7622	5 C		FRAME	5
7623	6	CALL DRWVEC (XXL,YYT,XXR,YYT,O)	FRAME	6
7624	7	CALL DRWVEC (XXL,YYT,XXL,YYB,O)	FRAME	7
7625	8	CALL DRWVEC (XXL,YYB,XXR,YYB,O)	FRAME	8
7626	9	CALL DRWVEC (XXR,YYB,XXR,YYT,O)	FRAME	9
7627	10	RETURN	FRAME	10
7628	11	END	FRAME	11

11000