LA-6014

UC-34
Reporting Date: June 1975
Issued: October 1975

A Numerical Method for Studying the Circulation Patterns of a Fluid in a Cavity

by

Linda M. Stephani
Thomas D. Butler

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87545

An Affirmative Action/Equal Opportunity Employer

UNITED STATES
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

Printed in the United States of America. Available from
National Technical Information Service
U S Department of Commerce
5285 Port Royal Road
Springfield, VA 22151
Price: Printed Copy $4.00 Microfiche $2.25

A NUMERICAL METHOD FOR STUDYING THE CIRCULATION PATTERNS OF A FLUID IN A CAVITY

by

Linda M. Stephani and Thomas D. Butler

ABSTRACT

This report describes a numerical method for studying the circulation patterns of a fluid in a cavity. The method incorporates three circulation-inducing mechanisms: (1) buoyancy induced by nonuniform initial distribution of heat throughout the fluid, (2) buoyancy induced by removal of heat from the fluid, and (3) forced convection induced by withdrawal of heated fluid and return of cooled fluid. A two-dimensional computer program, CIRCO, based on the Marker-and-Cell (MAC) technique, is used to study the circulation patterns. This report discusses the code and illustrates its capabilities by means of examples from studies conducted for the Pacer project, which investigates the concept of producing electrical power from energy released by thermonuclear explosions in a salt dome. Efficient engineering for withdrawing energy from the cavity requires an understanding of the circulation patterns of the heated fluid. CIRCO provides this information in the form of computer-generated plots.

I. INTRODUCTION

CIRCO is a transient, two-dimensional, Eulerian fluid dynamics computer program that uses a modified form of the MAC[1,2] solution technique. It is particularly applicable to study of buoyancy-induced circulation patterns in incompressible fluid. The finite difference program solves the full Navier-Stokes equations coupled with a temperature transport equation in cylindrical coordinates with azimuthal symmetry.

In CIRCO, buoyancy circulation of a fluid can be initiated by:

- Nonuniform deposition of heat energy.
- Removal of heat from the working fluid through a heat exchanger.
- Forced convection induced by withdrawal of heated fluid and return of cooled fluid.

These capabilities make CIRCO a useful research and engineering tool for a variety of applications, including postaccident heat removal studies in nuclear reactor safety analysis and that reported here, study of circulation patterns in a spherical cavity for the Pacer project.[3]

Pacer is the name of a project designed to investigate the feasibility of economically generating electrical power from energy released by thermonuclear explosions. In this concept, a nuclear device is detonated in an underground cavity filled with a working fluid such as water or steam. Subsequently, the thermal energy from the explosion is extracted using heat exchangers and converted to electrical energy. The process is repeated periodically as the excess energy in the cavity is expended, thereby permitting continuous energy conversion.

To withdraw energy from the cavity efficiently, one must determine the ideal placement and pumping requirements of the heat exchangers. These can be determined with CIRCO by studying the circulation patterns of the working fluid using various heat exchanger configurations.

This report describes the CIRCO program. Included are brief descriptions of the technique (Sec. II) and the code (Sec. III), examples illustrating the code's capabilities (Sec. IV), and a listing of the program (Appendix).

II. THE TECHNIQUE

A. Equations

Investigation of the dynamics of a circulating, incompressible fluid requires solution of the full fluid flow and heat transport equations. In cylindrical coordinates, assuming azimuthal symmetry and constant fluid density, the momenta and temperature transport equations are:

$$\frac{\partial u}{\partial t} + \frac{1}{r}\frac{\partial ru^2}{\partial r} + \frac{\partial uv}{\partial z} = -\frac{\partial p}{\partial r} + \nu\frac{\partial}{\partial z}\left(\frac{\partial u}{\partial z} - \frac{\partial v}{\partial r}\right), \tag{1}$$

$$\frac{\partial v}{\partial t} + \frac{1}{r}\frac{\partial ruv}{\partial r} + \frac{\partial v^2}{\partial z}$$

$$= -\frac{\partial p}{\partial z} - \frac{\nu}{r}\frac{\partial}{\partial r}\left[r\left(\frac{\partial u}{\partial z} - \frac{\partial v}{\partial r}\right)\right] - g_z\beta\,(T - \bar{T}), \tag{2}$$

and

$$\frac{\partial T}{\partial t} + \frac{1}{r}\frac{\partial rTu}{\partial r} + \frac{\partial Tv}{\partial z} = \frac{1}{r}\frac{\partial}{\partial r}\left(Kr\frac{\partial T}{\partial r}\right) + \frac{\partial}{\partial z}\left(K\frac{\partial T}{\partial z}\right), \tag{3}$$

and the equation of continuity is

$$\frac{1}{r}\frac{\partial ru}{\partial r} + \frac{\partial v}{\partial z} = 0. \tag{4}$$

In these equations, u and v are the radial and axial velocities, respectively; p is the normalized pressure; T, the temperature; ν, the kinematic viscosity coefficient; and K, the heat conduction coefficient. Buoyancy effects are represented in Eq. (2) using the Boussinesq approximation. Here, g_z is the gravity acceleration; β, the coefficient of thermal expansion; and $\bar{T}$, an average temperature in each horizontal layer. $\bar{T}$ is determined so that across any horizontal plane no net change of momentum results from buoyancy effects.

$$\bar{T}(z) \equiv \frac{2}{R^2}\int_0^R T(r,z)\,r\,dr,$$

where R is the radial extent of the fluid region.

B. The Solution Procedure

In CIRCO, the governing differential equations are approximated by finite difference equations related to an Eulerian mesh of computing cells. The cells are rectangular in cross section and of uniform size with radial and axial dimensions δr and δz, respectively. Each is characterized by a position index (i,j) that designates the center of the cell and its relative position within the mesh. Associated with each cell are the quantities that specify the local average fluid properties. The defined locations of the velocity components, the pressure, and the temperature in cell (i,j) are shown in Fig. 1. The velocities are located on cell boundaries, whereas the pressure and temperature are defined at the center. The relative position of each quantity within a cell is denoted by the subscripts.

To begin the calculation, initial values of the velocity, pressure, and temperature fields are specified for each cell. These are used to advance the solution one time step, δt. Thereafter, the process is repeated with the values of the dependent variables at a given time level, say n, used to obtain the solution at the new level, n + 1. Thus, the solution proceeds in time.

The solution procedure through one time cycle is composed of three separate phases: the intermediate time level velocity calculation, the pressure-velocity iteration, and the temperature calculation.

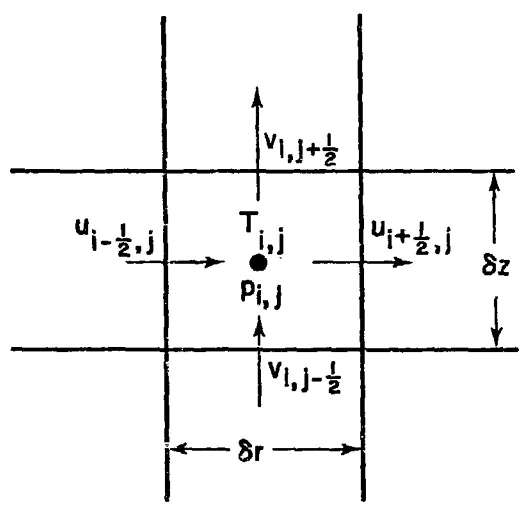

Fig. 1.
Location of the cell variables in a cell (i,j).

1. Intermediate Time Level Velocity Calculation. In this phase, intermediate values of the velocity components are determined explicitly using

$$
\begin{aligned}
\tilde{u}_{i+\frac{1}{2},j} = u^n_{i+\frac{1}{2},j} - \delta t \Bigg\{ & \frac{1}{r_{i+\frac{1}{2}}\delta r}\left(r_{i+1}\, u^n_{i+\frac{1}{2},j}\, u^n_{i+3/2,j} - r_i\, u^n_{i-\frac{1}{2},j}\, u^n_{i+\frac{1}{2},j}\right) \\
& + \frac{1}{\delta z}\left(v^n_{i+\frac{1}{2},j+\frac{1}{2}}\langle u\rangle^n_{i+\frac{1}{2},j+\frac{1}{2}} - v^n_{i+\frac{1}{2},j-\frac{1}{2}}\langle u\rangle^n_{i+\frac{1}{2},j-\frac{1}{2}}\right) + \frac{1}{\delta r}\left(p^n_{i+1,j} - p^n_{i,j}\right) \\
& - \nu\left[\frac{1}{\delta z^2}\left(u^n_{i+\frac{1}{2},j+1} - 2u^n_{i+\frac{1}{2},j} + u^n_{i+\frac{1}{2},j-1}\right) - \frac{1}{\delta r\delta z}\left(v^n_{i+1,j+\frac{1}{2}} - v^n_{i,j+\frac{1}{2}} - v^n_{i+1,j-\frac{1}{2}} + v^n_{i,j-\frac{1}{2}}\right)\right]\Bigg\} \quad ,
\end{aligned}
\tag{5}
$$

and

$$
\begin{aligned}
\tilde{v}_{i,j+\frac{1}{2}} = v^n_{i,j+\frac{1}{2}} - \delta t \Bigg(& \frac{1}{r_i\delta r}\left(r_{i+1}\, u^n_{i+\frac{1}{2},j+\frac{1}{2}}\langle v\rangle^n_{i+\frac{1}{2},j+\frac{1}{2}} - r_i\, u^n_{i-\frac{1}{2},j+\frac{1}{2}}\langle v\rangle^n_{i+\frac{1}{2},j+\frac{1}{2}}\right) \\
& + \frac{1}{\delta z}\left(v^n_{i,j+\frac{1}{2}}\, v^n_{i,j+3/2} - v^n_{i,j-\frac{1}{2}}\, v^n_{i,j+\frac{1}{2}}\right) + \frac{1}{\delta z}\left(p^n_{i,j+1} - p^n_{i,j}\right) + g_z\beta\left(\frac{T^n_{i,j+1} + T^n_{i,j}}{2} - \overline{T}^n_{j+\frac{1}{2}}\right) \\
& + \frac{\nu}{r_i}\Bigg\{\frac{1}{\delta r\delta z}\left[r_{i+\frac{1}{2}}\left(u^n_{i+\frac{1}{2},j+1} - u^n_{i+\frac{1}{2},j}\right) - r_{i-\frac{1}{2}}\left(u^n_{i-\frac{1}{2},j+1} - u^n_{i-\frac{1}{2},j}\right)\right] \\
& - \frac{1}{\delta r^2}\left[r_{i+\frac{1}{2}}\left(v^n_{i+1,j+\frac{1}{2}} - v^n_{i,j+\frac{1}{2}}\right) - r_{i-\frac{1}{2}}\left(v^n_{i,j+\frac{1}{2}} - v^n_{i-1,j+\frac{1}{2}}\right)\right]\Bigg\}\Bigg) \quad .
\end{aligned}
\tag{6}
$$

in which the tildes denote the intermediate level and the superscripts denote the time levels for the terms on the right-hand side.

A variable donor cell difference procedure is used in the cross-derivative transport terms; an example is

$$
v^n_{i+\frac{1}{2},j+\frac{1}{2}}\langle u\rangle^n_{i+\frac{1}{2},j+\frac{1}{2}} \quad .
$$

In this term, we define

$$
v^n_{i+\frac{1}{2},j+\frac{1}{2}} \equiv \frac{1}{2}\left(v^n_{i+1,j+\frac{1}{2}} + v^n_{i,j+\frac{1}{2}}\right) ,
$$

and

$$
\langle u\rangle^n_{i+\frac{1}{2},j+\frac{1}{2}} = \left(\xi_v + \frac{1}{2}\right) u^n_{i+\frac{1}{2},j} - \left(\xi_v - \frac{1}{2}\right) u_{i+\frac{1}{2},j+1} \quad ,
$$

where

$$
\xi_v = \alpha \operatorname{sign}\left(v^n_{i+\frac{1}{2},j+\frac{1}{2}}\right) + \beta_1\left(\frac{\delta t}{\delta z}\right) v_{i+\frac{1}{2},j+\frac{1}{2}} \quad ,
$$

and $0 \le \alpha \le 0.5$. The coefficient β_1 takes on a value of 0.5 when $\alpha = 0$ if so-called "interpolated donor cell differencing" is desired; otherwise, β_1 is zero.

The definition of $\overline{T}^n_{j+1/2}$ in Eq. (6) is given by

$$
\overline{T}^n_{j+\frac{1}{2}} \equiv \frac{\sum_i r_i\left(T^n_{i,j} + T^n_{i,j+1}\right)}{2\sum_i r_i} \quad , \tag{7}
$$

where the indicated summation is over all pairs of cells for which each cell in the pair is a fluid cell.

2. Pressure and Velocity Iteration. The next phase of the cycle is implicit; in it time level n + 1 values of pressure and velocity are determined. These are obtained so that the continuity equation is satisfied consistently with the boundary conditions for the problem. A complete description of the iteration process is given in Ref. 4. Only a brief outline is sketched here to indicate the basic approach.

Generally, the velocities from Eqs. (5) and (6) do not satisfy the continuity equation. The finite difference approximation to it is

$$
D_{i,j} \equiv \frac{1}{r_i\delta r}\left(r_{i+\frac{1}{2}}\, u_{i+\frac{1}{2},j} - r_{i-\frac{1}{2}}\, u_{i-\frac{1}{2},j}\right) + \frac{1}{\delta z}\left(v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}\right) = 0 \tag{8}
$$

in which the time levels are omitted. Upon substituting the tilde values into Eq. (8), we find

$$
\tilde{D}_{i,j} \ne 0 \; .
$$

The desired velocity values are found by iterating the pressure field until the resulting velocities yield

$$
\left|D^{n+1}_{i,j}\right| < \varepsilon \tag{9}
$$

for each cell. This is done by a point relaxation method in which the pressures and velocities are changed simultaneously. At each iteration level, the pressure is incremented by an amount

$$p_{i,j} = p_{i,j} + \delta p_{i,j} \quad , \tag{10}$$

where

$$\delta p_{i,j} = -\frac{\omega D_{i,j}}{\left(\frac{\partial D}{\partial p}\right)} \quad ,$$

$$\left(\frac{\partial D}{\partial p}\right) = 2\delta t \left(\frac{1}{\delta r^2} + \frac{1}{\delta z^2}\right) ,$$

and ω is an overrelaxation factor that has limits $0 < \omega < 2$. Once the pressure is found, new velocities are determined:

$$\begin{aligned} u_{i+\frac{1}{2},j} &= u_{i+\frac{1}{2},j} + \frac{\delta t}{\delta r}\,\delta p_{i,j} \quad , \\ u_{i-\frac{1}{2},j} &= u_{i-\frac{1}{2},j} - \frac{\delta t}{\delta r}\,\delta p_{i,j} \quad , \\ v_{i,j+\frac{1}{2}} &= v_{i,j+\frac{1}{2}} + \frac{\delta t}{\delta z}\,\delta p_{i,j} \quad , \\ v_{i,j-\frac{1}{2}} &= v_{i,j-\frac{1}{2}} - \frac{\delta t}{\delta z}\,\delta r_{i,j} \quad . \end{aligned} \tag{11}$$

This process is repeated until Eq. (9) is satisfied to within a specified amount ϵ, where ϵ is an acceptable error whose effects on the dynamics are negligible.

3. Temperature Calculation. With the final values of pressure and velocity determined for the cycle, the temperature transport equation is solved explicitly by

$$\begin{aligned} T^{n+1}_{i,j} = T^{n}_{i,j} - \delta t \Bigg(& \frac{1}{2r_i\delta r}\left(r_{i+\frac{1}{2}}\, T^{n}_{i+1,j}\, u^{n+1}_{i+\frac{1}{2},j} - r_{i-\frac{1}{2}}\, T^{n}_{i-1,j}\, u^{n+1}_{i-\frac{1}{2},j}\right) + \frac{1}{2\delta z}\left(T^{n}_{i,j+1}\, v^{n+1}_{i,j+\frac{1}{2}} - T^{n}_{i,j-1}\, v^{n+1}_{i,j-\frac{1}{2}}\right) \\ & - \left\{ \frac{1}{2r_i\delta r^2}\left[\left(K_{i+1,j} + K_{i,j}\right) r_{i+\frac{1}{2}} \left(T^{n}_{i+1,j} - T^{n}_{i,j}\right) - \left(K_{i,j} + K_{i-1,j}\right) r_{i-\frac{1}{2}} \left(T^{n}_{i,j} - T^{n}_{i-1,j}\right)\right] \right. \\ & \left. + \frac{1}{2\delta z^2}\left[\left(K_{i,j+1} + K_{i,j}\right)\left(T^{n}_{i,j+1} - T^{n}_{i,j}\right) - \left(K_{i,j} + K_{i,j-1}\right)\left(T^{n}_{i,j} - T^{n}_{i,j-1}\right)\right] \right\} \\ & - \frac{\delta t}{2}(1+\xi) \left\{ \frac{1}{\delta r^2}\left[\left(u^{n+1}_{i+\frac{1}{2},j}\right)^2 \left(T^{n}_{i+1,j} - T^{n}_{i,j}\right) - \left(u^{n+1}_{i-\frac{1}{2},j}\right)^2 \left(T^{n}_{i,j} - T^{n}_{i-1,j}\right)\right] \right. \\ & \left. + \frac{1}{\delta z^2}\left[\left(v^{n+1}_{i,j+\frac{1}{2}}\right)^2 \left(T^{n}_{i,j+1} - T^{n}_{i,j}\right) - \left(v^{n+1}_{i,j-\frac{1}{2}}\right)^2 \left(T^{n}_{i,j} - T^{n}_{i,j-1}\right)\right] \right\} \Bigg) \quad , \end{aligned} \tag{12}$$

in which the terms with coefficient $(1 + \xi)$ on the right-hand side have been added to overcome the effects of unfavorable truncation errors introduced by the difference approximation.[5] The value of ξ is an input constant that varies between zero and unity, with zero corresponding to exact lowest order truncation error removal, and larger values sometimes specified to introduce numerical smoothing. The difference form for the convection terms is derived using centered differences and the condition that the divergence of the velocity field must vanish.

C. Boundary Conditions

A number of different boundary conditions are permitted in CIRCO. These include conditions on both the velocity and temperature fields. Four kinds of velocity boundaries are defined:

- rigid, free-slip,
- rigid, no-slip,
- inflow,
- outflow.

Rigid boundaries are defined to coincide with cell boundaries, and the normal velocity component on the boundary vanishes. The tangential component has a vanishing gradient across a free-slip boundary and is zero at the wall in the no-slip case (see Fig. 2). Inflow and outflow boundaries also coincide with cell boundaries, and the normal velocity components are specified values on these boundaries. The tangential component vanishes at inflow and outflow boundaries.

Two temperature condition options are available for rigid boundaries. Either the temperature is kept constant, or the wall is insulated so that there is no heat flow across the boundary. At inflow boundaries, the temperature is kept constant, whereas the outflow boundary is insulated.

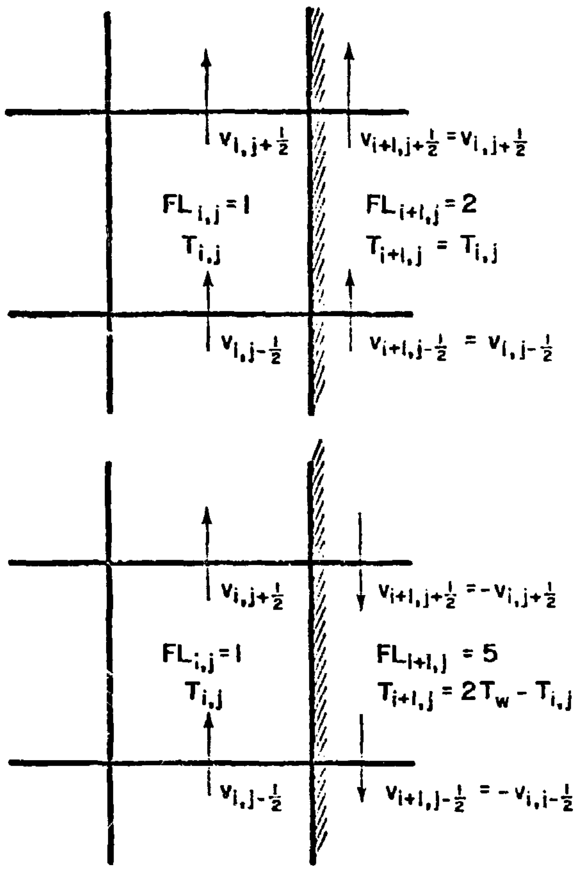

Fig. 2.
Boundary conditions applied to a rigid wall: (top) a rigid, free-slip, insulated boundary and (bottom) a rigid, no-slip, constant temperature boundary.

To facilitate the CIRCO logic, we surround the interior cells of the computation region with fictitious cells (see Fig. 4). The boundary conditions are applied by means of a cell-flagging scheme in which the computing and fictitious cells are labeled with appropriate flags to indicate the cell type and the condition imposed. Thus, associated with the computing mesh is an array of flags with the following definitions:

FL = 1 (interior fluid cell),
FL = 2 (rigid, free-slip, insulated wall or symmetry boundary),
FL = 3 (rigid, no-slip, insulated wall),
FL = 4 (rigid, free-slip, constant temperature wall),
FL = 5 (rigid, no-slip, constant temperature wall),
FL = 6 (inflow boundary),
FL = 7 (outflow boundary).

Computations are performed only for each cell with FL = 1. Some of these fictitious cells may be within the interior region, when the flow region is not rectangular in cross section (see Fig. 5). As an example, consider the situation shown by Fig. 2 (top) in which a rigid, free-slip adiabatic boundary is located on the right boundary of cell (i,j) so that $u_{i+1/2,j} = 0$. The flag of cell (i + 1,j) is $FL_{i+1,j} = 2$. The v-components of velocity are specified so that $v_{i+1,j\pm 1/2} = v_{i,j\pm 1/2}$, and $T_{i+1,j} = T_{i,j}$. In Fig. 2 (bottom), the configuration is that of a rigid, no-slip, constant temperature wall on the right of cell (i,j). In this case, $FL_{i+1,j} = 5$, $v_{i+1,j\pm 1/2} = -v_{i,j\pm 1/2}$, and $T_{i+1,j} = 2T_w - T_{i,j}$, where T_w is the constant wall temperature. Similar expressions are found for the other possible combinations.

D. Stability Conditions

The criteria for numerical stability of the finite difference equations in CIRCO are the usual ones for the MAC method[1] with an additional criterion for the temperature equation. The time step for a given cell size is chosen so that:

$$\frac{|u_{max}|\,\delta t}{\delta r} < 1 \quad ,$$

$$\frac{|v_{max}|\,\delta t}{\delta z} < 1 \quad ,$$

$$\nu\,\delta t < \Gamma$$

$$\nu > \max\left[\left(\frac{\delta t}{2}\,u^2 + \frac{\delta r^2}{2}\,\frac{1}{r}\,\frac{\partial ur}{\partial r}\right),\ \left(\frac{\delta t}{2}\,v^2 + \frac{\delta z^2}{2}\,\frac{\partial v}{\partial z}\right)\right] ,$$

$$\left[\kappa + 1/2\ (1+\xi)(u^2+v^2)\delta t\right]_{max}\delta t < \Gamma \quad .$$

Here, the subscript, max, indicates the largest values of the indicated quantities in the computing mesh. Γ is given by

$$\Gamma = \frac{\delta r^2\,\delta z^2}{2(\delta r^2 + \delta z^2)} \quad .$$

CIRCO has an automatic built-in time step control that has proved useful for application to cavity circulation problems. The time step is automatically increased by a factor TFACT, provided

$$\frac{2\,\delta t\,V_{max}}{(\delta r + \delta z)} < 0.15 \quad .$$

where V_{max} is the highest velocity in the computing mesh. Conversely, the time step is automatically decreased by a factor $(TFACT)^{-1}$, provided

$$\frac{2\,\delta t\,V_{max}}{(\delta r + \delta z)} > 0.7 \quad .$$

The quantity, TFACT, is an input number in the code, and it is typically chosen to equal 2.0.

III. THE CODE

There are two major sections of CIRCO, the setup, or initialization, section and the calculational section. Figure 3 shows the code layout. In the initialization section, requirements for a particular problem are defined, including the information needed to specify the initial and boundary conditions, and whether the problem is to use information stored on magnetic tape or data from cards. After these quantities are determined, control transfers to the calculational section where it remains for the duration of the computation. The primary cell quantities, u, v, p, and T are calculated by the procedure described in Sec. II. The rest of this section amplifies on the initialization section and describes the computing mesh, input data, output options, and magnetic tape restart capabilities.

A. The Initialization Section

Requirements for the flow problem to be calculated are specified in the initialization section. The basic steps and decision points within the code are summarized in Fig. 3. The numbers in elongated ovals are FORTRAN statement numbers within the code.

CIRCO begins by reading two input cards, the first containing the problem identification and the second including data that determine the initialization procedure. When a calculation is starting from initial data, the control proceeds to region 105, where other data cards are read, problem constants are computed, and initial quantities for the computing cells are set. Control is then transferred to the calculational section, which begins at region 1000. If, however, the problem is to be restarted from data stored on magnetic tape, the code branches to region 7500, where the tape is read. One can then change the tape information by reading additional data cards and calculating new problem constants. If such changes are desired, control returns to region 105, where the appropriate quantities are determined, and then advances to region 3499. When no changes are necessary, control proceeds directly to region 3499.

B. The Computing Mesh

Figure 4 diagrams the basic CIRCO computing mesh. Indicated are the fluid region (I × J cells) and the fictitious cells (shaded region) surrounding it. The symmetry axis is one cell in from the left boundary of the mesh. The fluid region coordinates are indicated on the interior mesh corners where R2 = I × δr and ZT = J × δz. In the figure, cell indexing for the mesh requires IM = I + 2 and JM = J + 2. This basic mesh of rectangular cross section is specified in the input data by setting SCN = 0.0 (see Sec. III-C).

To represent a spherical cavity with the CIRCO computing mesh, the curved boundaries of the cavity are approximated by straight line segments coincident with cell boundaries, as Fig. 5 indicates. Here the shaded region represents the nonfluid portion. This option is specified by setting SCN = 1.0 in the input data. The cavity radius is R2, and ZT = 2.0(R2).

Also shown in Fig. 5 is an optional spherical heated portion (see Sec. III-C) of the fluid whose circular cross section is similarly approximated by straight line segments coinciding with cell boundaries. The region of radial dimension R1 is centered at height Z1 above the cavity bottom. This represents a region whose temperature differs from that of the surrounding fluid in the cavity at initial time. When SCN = 0.0, the heated portion is rectangular in cross section with radial dimensions R1 and it extends a length, Z1/2.0, above and below the axial location Z1.

The cavity boundary conditions are applied, according to the prescriptions of Sec. II-C in two steps. First, the cells bounding the fluid region on the top, right, and bottom are flagged appropriately for the velocity and temperature conditions specified for the cavity walls. Further, the cells along the symmetry axis, the left boundary of the fluid region, are flagged as rigid, free-slip, insulated boundary cells. Second, segments of the cavity wall may require different velocity and temperature conditions to reflect the presence of inflow and outflow ports or cooling plates. This alteration is accomplished by using an optional set of input data which identifies the particular segment and the appropriate conditions (see Sec. III-C).

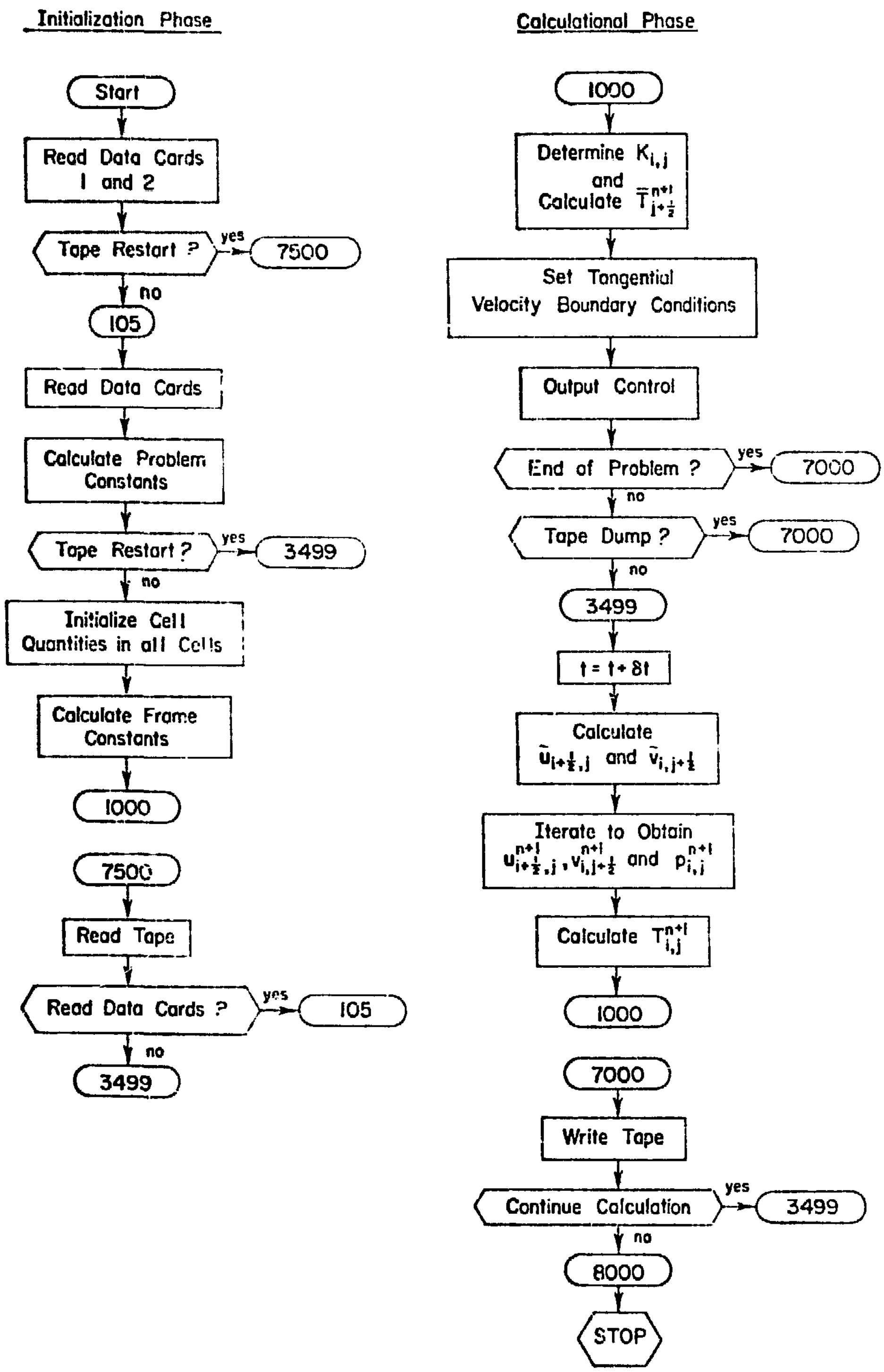

Fig. 3.
CIRCO flow diagram.

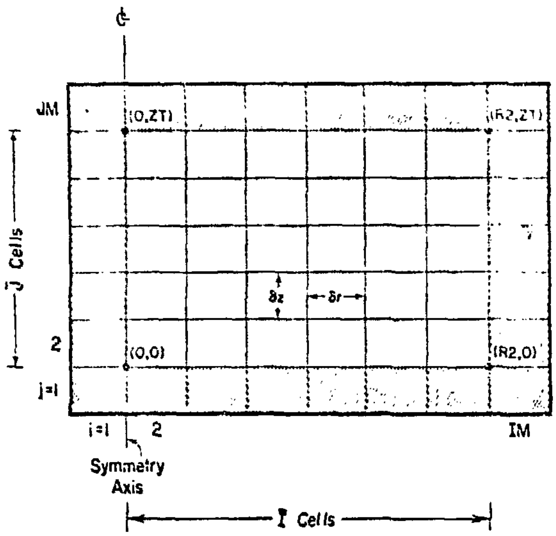

Fig. 4.
The CIRCO mesh showing a fluid region of rectangular cross section.

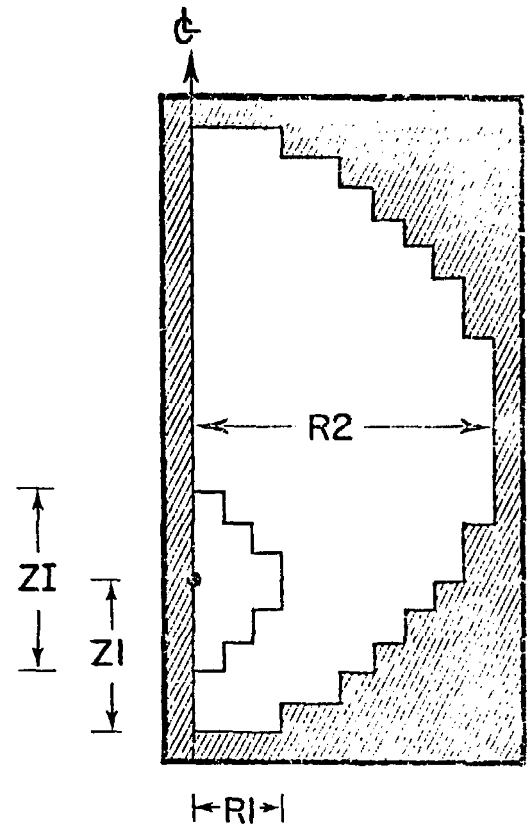

Fig. 5.
The mesh approximating a circular cavity. An optional heated part of the fluid appears in the bottom half of the mesh.

C. The Input Data

CIRCO's input card deck may consist of three categories of data. The first defines the source of the initializing information; the second, the general problem requirements; and the third, the special problem requirements. In this section we list the data cards, their FORTRAN format statements, and the quantities they contain.

Card No. 1 (Format: 10A8).

Columns 2-80 are used for the problem identification for prints and plots. To allow for carriage control, column 1 is left blank. This card is required for each problem.

Card No. 2 (Format: 3(6x, F12.5)).

This card contains the quantities that determine the source of the initializing information. It must be included for each problem.

TD1 = the source identifier, where:
  0.0 = a magnetic tape.
  1.0 = data cards.

TD2 = the tape dump number.

TD3 = the flag that determines the procedure after a tape is processed, where:
  0.0 = do not read data cards.
  1.0 = read data cards.

Card No. 3 (Format: 10A8).

This card contains the same information as Card 1, but it is included only when TD1 = 0.0. Otherwise, it is omitted from the input deck.

The quantities on Cards 4 through 14 define the general problem requirements, such as cell dimensions, initial fluid properties, boundary information, output specifications, and time-related quantities. These data are necessary if the calculation begins at a time of zero or if new data are required for problems restarted from magnetic tape. The floating point data on Cards 4 through 12 are stored in the XPUT array; the integer data on Cards 13 and 14 are stored in the NPUT array.

Card No. 4 (Format: 6x, I2, 10x, 3(6x, F12.5))

NUM = the number of floating point quantities stored in the XPUT array.

DR = δr, the cell dimension in the r direction.

DZ = δz, the cell dimension in the z direction.

DT = δt, the initial value of the time increment.

Card No. 5 (Format: 4(6x, F12.5))

R1 = the radial dimension of the optional interior heated region (default value = 0.0).

Z1 = the center position in the z-direction of the optional interior heated region measured from the bottom of the cavity (default value = 0.0).

ZI = the axial length of the optional interior heated region (default value = 0.0). With a spherical cavity (SCN = 1.0), ZI = 2R1 produces a spherical interior heated region. For a nonspherical cavity (SCN = 0.0), the interior heated region is rectangular in cross section and of extent R1 × ZI.

R2 = the radial dimension of the computing mesh.

Card No. 6 (Format: 4(6x, F12.5))

T1 = the initial temperature of the optional interior heated region (default value = 0.0).

T2 = the initial temperature of the fluid cells.

TWL = the temperature of the rigid boundary. TWL is zero for an insulated wall and a specified value for a constant temperature wall.

UIN = the initial radial component of velocity for the fluid cells.

Card No. 7 (Format: 4(6x, F12.5))

VIN = the initial z component of velocity for the fluid cells.

PIN = the initial pressure in the fluid cells.

C1 = K, the heat conduction coefficient.

VEL = the cavity wall velocity boundary condition flag, where:

0.0 = a free-slip boundary.

1.0 = a no-slip boundary.

Card No. 8 (Format: 4(6x, F12.5))

TEM = the cavity wall temperature boundary condition flag, where:

0.0 = an insulated boundary.

1.0 = a constant temperature boundary.

CPL = the constant temperature plate flag, where:

0.0 = no plate.

1.0 = at least one plate included (requires the "constant temperature plate data package," described in Sec. III-C.1).

PRTI = the inflow port flag, where:

0.0 = no port

1.0 = at least one port included (requires the "inflow port data package," described in Sec. III-C.2).

PRTO = the outflow port flag, where:

0.0 = no port.

1.0 = at least one port included (requires the "outflow port data package," described in Sec. III-C.3).

Card No. 9 (Format: 4(6x, F12.5))

SCN = the mesh configuration flag, where:

0.0 = a rectangular mesh.

1.0 = a semicircular mesh.

OM = ω, the overrelaxation parameter, usually 1.5.

NU = ν, the kinematic viscosity coefficient.

BETA = β, the coefficient of thermal expansion.

Card No. 10 (Format: 4(6x, F12.5))

GZ = g_z, the gravity acceleration.

ALPHA = α, a coefficient in the donor cell expressions.

BTA = β_1, a coefficient in the donor cell expressions.

TFACT = the factor for altering δt (see Sec. II-D).

Card No. 11 (Format: 4(6x, F12.5))

TCN = ξ, the truncation error factor in the temperature equation.

CEP = ϵ, the convergence criterion for the pressure iteration.

VCTPL = the velocity vector plot flag, where:

0.0 = no vector plots.

1.0 = vector plots.

VM = the velocity vector scaling factor for plots. The scaling constant is $VC = (\delta r + \delta z)/(2VM)$.

Card No. 12 (Format: 4(6x, F12.5))

CNPLT = the contour plot flag, where:

0.0 = no contour plots.

1.0 = contour plots.

TFIN = the problem time to finish.

TDMP = the tape dump flag, where:

0.0 = no tape dump.

1.0 = tape dump.

DTCP = the central processor (CP) time in seconds between tape dumps (default = DTCP > the time limit requested on job card).

Card No. 13 (Format: 6(6x, I6))

INUM = the number of integer quantities stored in the NPUT array.

IBR = $\bar{I}$, the maximum number of interior fluid cells in the radial direction ($\equiv R2/\delta r$).

JBR = $\bar{J}$, the maximum number of interior fluid cells in the axial direction ($\equiv ZT/\delta z$).

NPR = the number of cycles between prints on paper.

NPL = the number of cycles between plots on microfilm.

NWR = the number of cycles between writes on microfilm.

Card No. 14 (Format 6(6x, I6))

LPR = the long print flag, where:
1 = no long prints on paper,
2 = long prints on paper.

LWR = the long write flag, where:
1 = no long writes on film,
2 = long writes on film.

The final set of data is optional and varies in content for different problems. Included in this set are the requirements for special constant temperature, inflow, and outflow boundary segments. For each segment, two data cards are required. The first contains coordinates of the segment, R_{LT}, Z_{BT}, R_{RT}, and Z_{TP} (Fig. 6). The second includes the appropriate values of velocity and temperature. The data requirements for the three boundary types are:

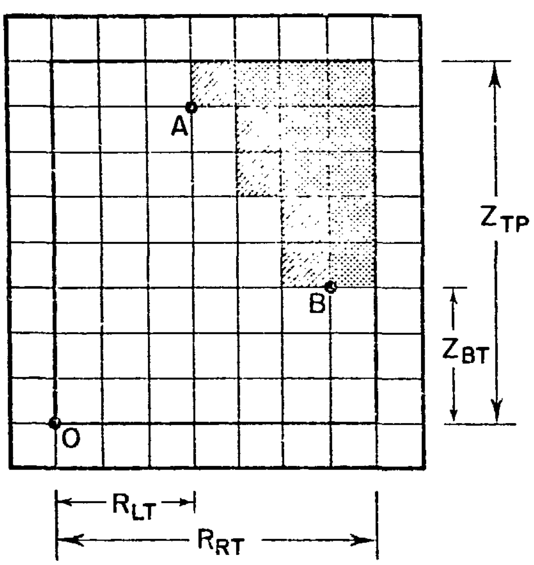

Fig. 6.
The spatial dimensions needed for applying special boundary conditions on the cavity walls between points A and B. The shaded area indicates the special fictitious cells required in the calculation. The dotted area includes cells flagged as special boundary cells but not used.

1. Constant Temperature Walls.

Card No. 1 (Format: 6x, I2)

NCP = the number of constant temperature segments.

Card No. 2 (Format: 4(6x, F12.5))

CPC = the segment coordinates, R_{LT}, Z_{BT}, R_{RT}, and Z_{TP}.

Card No. 3 (Format: 4(6x, F12.5))

TPL = the specified temperature for the segment.

(This set contains 2NCP + 1 cards. The remaining cards are of the same form as Cards 2 and 3 and are arranged in the same order.)

2. Inflow Boundaries.

Card No. 1 (Format: 6x, I2)

NIP = the number of inflow ports.

Card No. 2 (Format: 4(6x, F12.5))

IPC = the segment coordinates, R_{LT}, Z_{BT}, R_{RT}, and Z_{TP}.

Card No. 3 (Format: 4(6x, F12.5))

UPI = the radial component of the inflow velocity.

VPI = the axial component of the inflow velocity.

TPI = the specified temperature for the segment.

(Additional inflow segments require cards like Cards 2 and 3. The complete set of data for inflow ports contains 2NIP + 1 cards.)

3. Outflow Boundaries.

Card No. 1 (Format: 6x, I2)

NOP = the number of outflow ports.

Card No. 2 (Format: 4(6x, F12.5))

OPC = the segment coordinates, R_{LT}, Z_{BT}, R_{RT}, and Z_{TP}.

Card No. 3 (Format: 4(6x, F12.5))

UPO = the radial component of outflow velocity.

VPO = the axial component of outflow velocity.

(Each outflow port calls for additional cards like 2 and 3. The complete set includes 2NOP + 1 cards.)

D. Output Options

CIRCO results are recorded on printer paper and microfilm. The printer information includes the input data, long lists of values of the cell variables, and short prints containing the time, cycle number, number of iterations per cycle, and maximum velocity in the system. The input data are printed at the start of the calculation, whereas the long lists and short prints occur every NPR number of time steps.

The same information also appears on microfilm together with contour plots of the primary variables and velocity vector plots. The number of cycles between long lists on film is given by NWR, and NPL is the number of cycles between plots. The short print information appears every cycle.

The code can automatically provide long lists on paper and film of the initial cell data and those at cycle 1. Each long list contains i, j, $FL_{i,j}$, $u_{i+1/2,j}$, $v_{i,j+1/2}$, $p_{i,j}$, $D_{i,j}$, and $T_{i,j}$, in that order, for every cell in the computing mesh. Appropriate headings are provided for easy identification.

The microfilm plots are optional; velocity vector plots are provided when VCTPL = 1.0; contour plots of pressure and temperature, when CNPLT = 1.0. The velocity plots show the flow direction and relative magnitude by plotting the scaled velocity at each cell center. Two types are available, one involving a constant scaling factor that depends on the input value VM, the other involving scaling that is determined by the maximum velocity in the system. In the latter, the vector length is determined so as not to exceed a cell dimension.

For each contour plot, the matrix of cell values is scanned to determine the minimum and maximum. Then, allowing for 21 lines, the contour interval is determined using the minimum and maximum limits. The contour lines are formed by joining points of equal value (interpolated between cell centers) with line segments. These plots contain the letters L and H, which designate the locations of the minimum and maximum contour values, respectively.

E. Magnetic Tape Restarts

CIRCO can store data on magnetic tape at selected times during a run to permit restarting the problem from the stored information. The cell quantities and necessary problem constants are dumped on tape periodically during normal running of the code. The dump frequency is controlled by input quantities, DTCP and TFIN, and the time limit requested on the job card. Dumps are provided at regular intervals of elapsed central processor time by specifying DTCP to be the number of seconds between dumps. Also, when the calculation reaches the problem time to finish (t = TFIN), a dump is automatically taken. If the time limit requested on the job card is reached before t = TFIN, a dump is provided shortly before the time limit is exceeded.

To suppress the use of magnetic tape, DTCP and TFIN must be set to values in excess of those anticipated for computer time and problem time, respectively. Specifying TDMP = 0.0 in the input data suppresses the dump just before the computer time request is exceeded.

Two options are available upon restart from tape: either the problem is continued using the data stored on tape, or the input data for the problem require changes and new data cards must be read (see Sec. III-C). In the latter case, provision is made to change input quantities, but the data for the cell variables are the information stored on tape.

IV. NUMERICAL EXAMPLES

This section presents the results of three applications of CIRCO to buoyancy-driven circulation within a rigid spherical cavity. Examples serve to familiarize the reader with CIRCO's capabilities and to show graphical code output. They relate specifically to Pacer project investigations and include, in order, working fluid circulation initiated:

- From the bouyancy induced by a localized hot region, which is the result of nonuniform distribution throughout the working fluid of heat from a single explosion,
- From the buoyancy induced by removing heat from the working fluid through heat exchangers, and
- From forced convection and buoyancy induced by withdrawing the heated working fluid into heat exchangers and returning the cooled fluid to the cavity.

A. Nonuniform Deposition of Heat Energy from an Explosion

The first example is buoyancy induced by an explosion in the cavity, in the absence of heat exchangers and forced convection. The explosion distributes heat energy unevenly, but the corresponding deposition of momentum is ignored. To represent the initial conditions, a heated, 600°K, 50-m-radius spherical region is located on the axis 75 m from the bottom of the 200-m-radius spherical cavity. The initial temperature of the surrounding fluid is 500°K. As the buoyant fluid rises, clockwise circulation is produced. Gradually the temperature of the initially heated fluid decreases, the buoyant fluid hits the top of the cavity, and the circulation slows.

In this calculation, IBR = 40 and JBR = 80, so $\delta r = \delta z = 5$m. In this and the examples to follow, the heat conduction coefficient is constant, K = 1.25 m^2/s, and the coefficient of volumetric expansion is $\beta = 0.002$/°K. Initially the fluid is at rest. The temperature gradient is depicted in the contour plot of Fig. 7, which shows the relative location of the

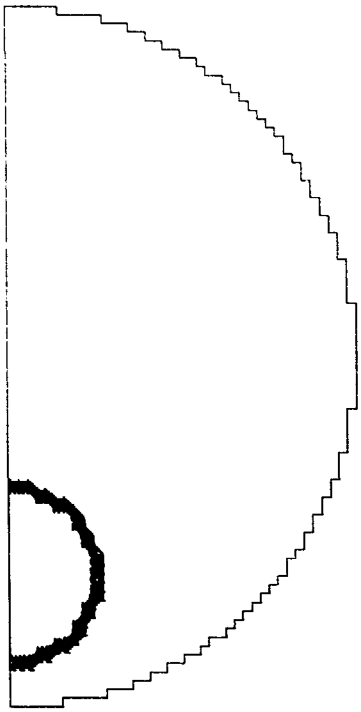

Fig. 7.
The initial temperature contour plot showing the location of the heated region in the cavity.

heated region in the cavity. Figure 8 is a composite that shows the state of the fluid at t = 20, 50, and 70 s. These plots show the outline of the cavity region in the r-z cross section. The top row depicts the velocity vectors; the middle row, the temperature contours (isotherms); and the bottom, the pressure contours (isobars) at the three times.

At t = 20 s, the induced circulation, as indicated in the velocity vector plot, is clockwise. The maximum velocity in the system, v_{max}, is 16.9 m/s. By this time, the maximum temperature, T_{max}, is 577°K. Associated with the vortex center of the circulation is a low-pressure region, whereas a high-pressure region develops at the top of the heated region. The heated fluid continues to rise until, by 50 s, it has reached the top of the cavity and expanded radially to form a torus. A clockwise circulation is still maintained; however, v_{max} is 9.5 m/s. T_{max} at this time is 516°K. The low-pressure region is still associated with the vortex center of the circulation. At 70.0 s, the fluid motion has slowed; v_{max} is 4.6 m/s. The heated fluid continues to follow the cavity wall. It has cooled to a T_{max} of 511°K. The low-pressure region remains closely associated with the vortex center.

B. The Presence of a Cooling Plate

The second example is circulation induced by the presence of a cooling plate off the cavity axis. The cavity is again 200 m in radius, and the working fluid is initially at a uniform temperature of 550°K. The cooling plate is kept at 300°K throughout the calculation. For this case, $\delta r = \delta z = 10$ m.

Table I is the printout of the input data for this problem showing the additional information needed to define the cooling plate. This printout appears both on paper and microfilm.

The cooling plate is a toroidal section of a cap on the wall in the upper part of the cavity. Starting 130 m from the axis and extending to a radius of 160 m, it is located 330-360 m above the bottom. Figure 9 shows the isotherm plot at cycle 1 when the fluid immediately around the plate is cooled to 524°K. The temperature difference between the wall and the fluid induces a circulation that at first rotates generally clockwise in the r-z cross-sectional plane. Subsequently, the circulation patterns become very complex showing breakup and coalescence of eddies within the cavity. This behavior is seen in Fig. 10, which shows the velocities at 120 s (200 time steps) and 520 s (1000 time steps). The maximum velocity at 120 s is 11.1 m/s; that at 520 s is 5.1 m/s. The circulation pattern at the later time shows that the single large vortex formed initially breaks up into a series of vortices that continue to interact, causing further breakup and coalescence.

C. Forced Convection

The final sample calculation is that of coupled forced convection and buoyancy in which working fluid is withdrawn from the cavity at 1450 kg/s. Fluid, cooled to 100°K after passage through a heat exchanger, is introduced through a one-cell opening at the top left of the computing mesh. The heated fluid is withdrawn through an adjacent cell. This configuration represents inflow and outflow through coaxial cylindrical pipes.

Unlike the circulation patterns observed in the cooling plate problem, this calculation shows a counterclockwise circulation in the r-z plane. The general pattern formed in the first 5 min of cavity operation is maintained throughout 12 h of operation. Furthermore, there is no short-circuiting of the

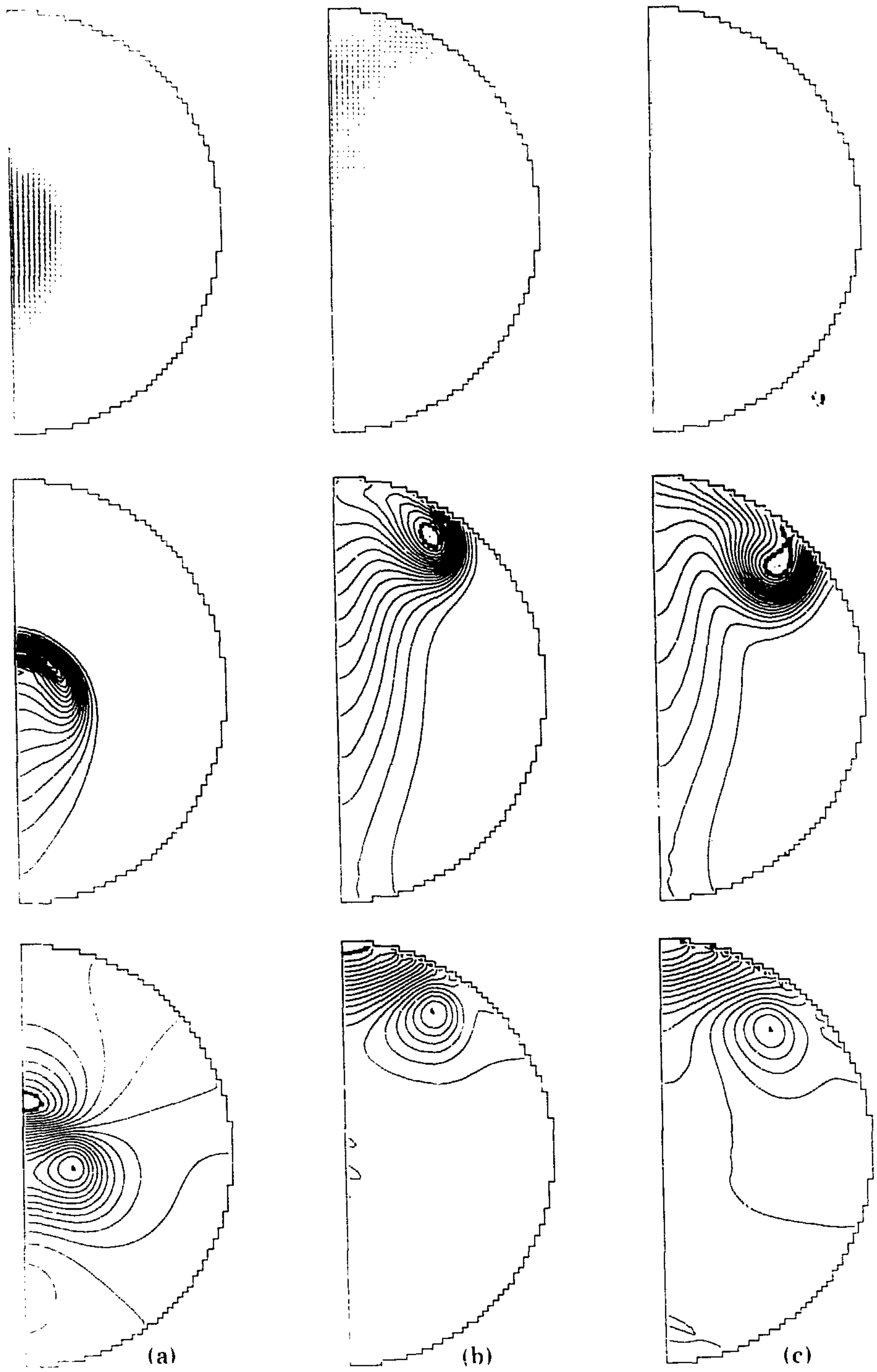

Fig. 8.
A composite of velocity vector and temperature and pressure contour plots showing the state of the fluid whose circulation was initiated by an uneven distribution of heat energy in the cavity: (a) t = 20 s, (b) t = 50 s, and (c) t = 70 s.

TABLE I

CIRCO INPUT DATA

TD1 = 1.00000E+00	TD2 = 1.00000E+00	TD3 = 0.				
DR = 1.00000E+01	DZ = 1.00000E+01	DT = 2.00000E+00	R1 = 0.	Z1 = 0.	ZI = 0.	R2 = 2.00000E+02
T4 = 0.	T2 = 5.50000E+02	TWL = 0.	UIN = 0.	VIN = 0.	PIN = 0.	CI = 1.25000E+00
VEL = 0.	TEM = 0.	CPL = 1.00000E+00	PRTI = 0.	PRTO = 0.	SCN = 1.00000E+00	OM = 1.50000E+00
NU = 1.00000E+00	BETA = 2.00000E-03	GZ = 9.80000E+00	ALPHA = 5.00000E-01	BTA = 0.	TFACT = 2.00000E+00	TCN = 2.50000E-01
CEP = 5.00000E-03	VCTPL = 1.00000E+00	VM = 5.00000E+00	CNPLT = 1.00000E+00	TFIN = 1.00000+06	TDMP = 1.00000E+00	DTCP = 3.00000E+02

IBR = 20 JBR = 40 NPR = 10 NPL = 25 NWR = 25 LPR = 1 LWR = 2

NCP =	1 NPC =	4		
CPC	130.00000	330.00000	160.00000	360.00000
TPL	300.00000			

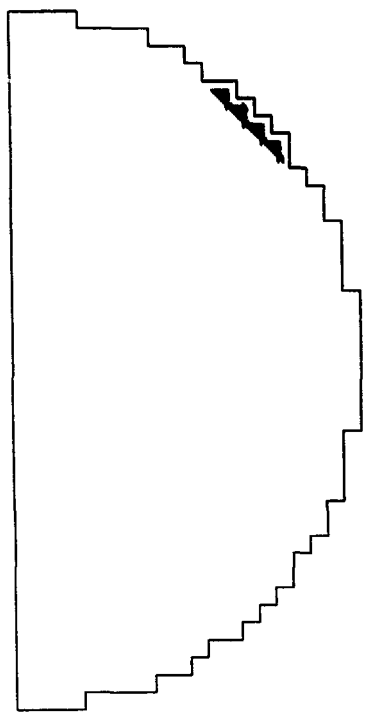

Fig. 9.
The initial isotherm plot showing the location of the cooling plate on the cavity wall.

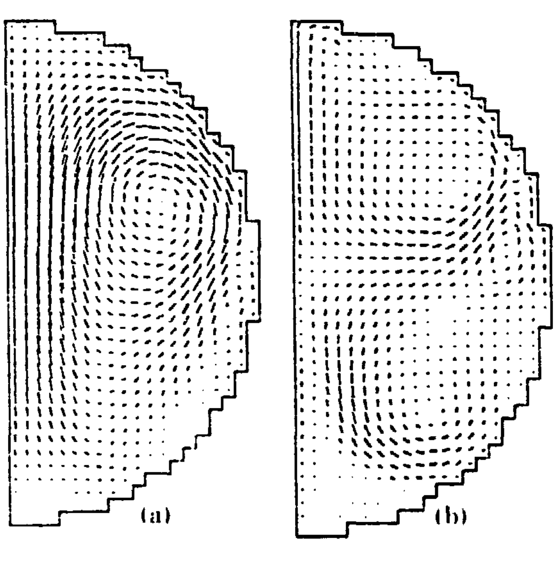

Fig. 10.
Velocity vector plots of the induced circulations caused by a cooling plate: (a) t = 120 s and (b) t = 520 s.

cooled fluid from the inflow to the outflow ports, which indicates the likelihood of efficient cavity cooling.

Figure 11 summarizes the results of this calculation by showing the velocity vectors and temperature contours at 1 and 12 h during cavity operation. The cooled fluid introduced on the cavity axis accelerates as it flows down the axis to the bottom. The velocity at the inflow port is held constant at 0.17 m/s. After 1 h of operation, the maximum velocity in the cavity is 6.8 m/s, and it is still 6.6 m/s after 12 h of operation. At t = 12 h, the average cavity temperature is 532°K, down from the initial 550°K. The isotherm plots at the two times are similar in shape, the maximum cell temperature being 552°K and the minimum 492°K after 1 h. These temperatures reduce to 533 and 474°K, respectively, at 12 h.

ACKNOWLEDGMENTS

We thank Francis H. Harlow for his many valuable contributions to development of the code.

REFERENCES

1. F. H. Harlow and J. E. Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow," Phys. Fluids 8, 2182 (1965); J. E. Welch, F. H. Harlow, J. P. Shannon, and B. J. Daly, "The MAC Method: A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-Flow Problems Involving Free Surfaces," Los Alamos Scientific Laboratory report LA-3425 (1966).

2. A. A. Amsden and F. H. Harlow, "The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows," Los Alamos Scientific Laboratory report LA-4370 (1970).

3. R. G. Shreffler and R. E. Roush, "PACER Program FY-1974 LASL Activity," Los Alamos Scientific Laboratory report LA-5754-MS (1974).

4. C. W. Hirt and J. L. Cook, "Calculating Three-Dimensional Flows around Structures and over Rough Terrain," J. Comp. Phys. **10**, 324-340 (1972).

5. W. C. Rivard, O. A. Farmer, T. D. Butler, and P. J. O'Rourke, "A Method for Increased Accuracy in Eulerian Fluid Dynamics Calculations," Los Alamos Scientific Laboratory report LA-5426-MS (1973).

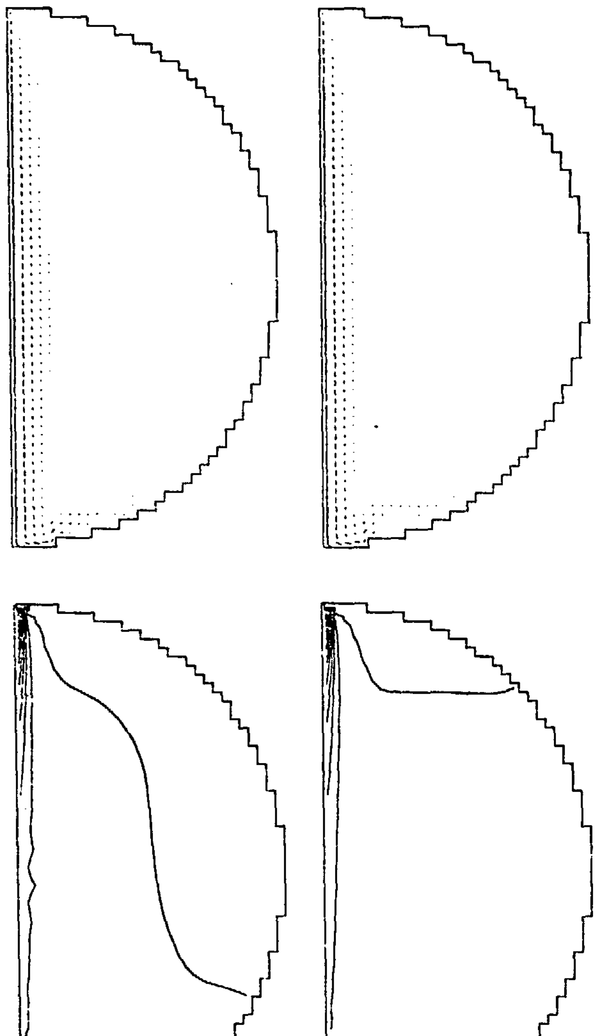

Fig. 11.
A composite of velocity vector and isotherm plots showing the effects of adding cooled fluid and withdrawing heated fluid through inflow and outflow ports in the cavity walls: (a) t = 1 h and (b) t = 12 h.

APPENDIX

FORTRAN IV INDEX LISTING OF THE CIRCO PROGRAM

LASL CODE NO. LP-0550

```
RUN-LCM97  U                    75/06/12     11.02.22     T3LMSZZ2CD    PAGE NO. 1

         PROGRAM CIRCO (INP,OUT,FILM,FSET10=INP,FSET9=OUT,FSET12=FILM,FSET7 CIRCO       2
        1,FSET8)                                                            CIRCO       3
2        COMMON/A// A1(3444),A2(3444),A3(3444),A4(3444),A5(3444),A6(3444),A COMM        2
        17(3444),A8(3444),A9(3444),A10(82),XPUT(50),NPUT(25),FA(50),IA(50), COMM        3
        2RD(170),ZD(170),NAME(10),IFRC(210),ZZZ                             COMM        4
2        COMMON/SPCLQN/ CPC(40),IPC(40),OPC(40),UPI(10),VPI(10),TPI(10),TPL COMM        5
        1(10),UPO(10),VPO(10),SPC                                           COMM        6
2        EQUIVALENCE (A1,FL),(A2,U),(A3,UT),(A4,V),(A5,VT),(A6,P),(A7,T),(A COMM        7
        18,K),(A9,D),(A10,TBAR),(A5,TN),(RD,RCC),(RD(43),RBC),(RD(85),RRCC) COMM        8
        2,(RD(127),RRBC),(ZD,ZCC),(ZD(83),ZBC),(IFRC,IRPC),(IFRC(43),IZRC), COMM        9
        3(IFRC(125),IZTC)                                                   COMM       10
2        EQUIVALENCE (FA(1),RDR),(FA(2),RDRH),(FA(3),RDRQ),(FA(4),RRR),(FA( COMM       11
        15),RRRH),(FA(6),RRR2),(FA(7),RT),(FA(8),RRT),(FA(9),BTR),(FA(10),R COMM       12
        2DZ),(FA(11),RDZH),(FA(12),RDZQ),(FA(13),RZZ),(FA(14),RZZH),(FA(15) COMM       13
        3,RZZ2),(FA(16),ZT),(FA(17),RZT),(FA(18),BTZ),(FA(19),PDRZ),(FA(20) COMM       14
        4,RDT),(FA(21),DTRDR),(FA(22),DTRDZ),(FA(23),DTRZ),(FA(24),TIME),(F COMM       15
        5A(25),RTFCT),(FA(26),EM6),(FA(27),FC),(FA(28),FC1),(FA(29),B),(FA( COMM       16
        630),UCON),(FA(31),DTEST),(FA(32),DM),(FA(33),VC),(FA(34),VELMX),(F COMM       17
        7A(35),DAT),(FA(36),CLK),(FA(37),TD),(FA(38),TCN1)                  COMM       18
2        EQUIVALENCE (IA(1),INUM),(IA(2),NUM),(IA(3),IB1),(IA(4),IM),(IA(5) COMM       19
        1,JB1),(IA(6),JM),(IA(7),IRL),(IA(8),IRR),(IA(9),IZB),(IA(10),IZT), COMM       20
        2(IA(11),ICNT),(IA(12),JCNT),(IA(13),NCYC),(IA(14),NPRT),(IA(15),NP COMM       21
        3LT),(IA(16),NWRT),(IA(17),JNM),(IA(18),IT),(IA(19),ITMX),(IA(20),N COMM       22
        4CP),(IA(21),NPC),(IA(22),NIP),(IA(23),NIC),(IA(24),NOP),(IA(25),NO COMM       23
        5C)                                                                 COMM       24
2        EQUIVALENCE (XPUT(1),DR),(XPUT(2),DZ),(XPUT(3),DT),(XPUT(4),R1),(X COMM       25
        1PUT(5),Z1),(XPUT(6),Z1),(XPUT(7),R2),(XPUT(8),T1),(XPUT(9),T2),(XP COMM       26
        2UT(10),TWL),(XPUT(11),UIN),(XPUT(12),VIN),(XPUT(13),PIN),(XPUT(14) COMM       27
        3,C1),(XPUT(15),VEL),(XPUT(16),TEM),(XPUT(17),CPL),(XPUT(18),PRTI), COMM       28
        4(XPUT(19),PRTO),(XPUT(20),SCN),(XPUT(21),OM),(XPUT(22),NU),(XPUT(2 COMM       29
        53),BETA),(XPUT(24),GZ),(XPUT(25),ALPHA),(XPUT(26),BTA),(XPUT(27),T COMM       30
        6FACT),(XPUT(28),TCN),(XPUT(29),CEP),(XPUT(30),VCTPL),(XPUT(31),VM) COMM       31
        7,(XPUT(32),CNPLT),(XPUT(33),TFIN),(XPUT(34),TDMP),(XPUT(35),DTCP)  COMM       32
2        EQUIVALENCE (NPUT(1),IBR),(NPUT(2),JBR),(NPUT(3),NPR),(NPUT(4),NPL COMM       33
        1),(NPUT(5),NWR),(NPUT(6),LPR),(NPUT(7),LWR)                        COMM       34
2        INTEGER TFI,TFO                                                    COMM       35
2        INTEGER FL,FC,FC1                                                  COMM       36
2        REAL IPC,K,NU                                                      COMM       37
2        DIMENSION U(32,62),UT(32,62),V(32,62),VT(32,62),P(32,62),T(32,62), COMM       38
        1TN(32,62),K(32,62),FL(32,62),TBAR(62),D(32,62)                     COMM       39
2        DIMENSION RCC(32),RBC(32),RRCC(32),RRBC(32),ZCC(62),ZBC(62)        COMM       40
2        DIMENSION IRPC(32),IZBC(62),IZTC(62)                               COMM       41
       1 FORMAT(10A8)                                                       CIRCO       5
       2 FORMAT(6X,I2,10X,3(6X,F12.5)/(4(6X,F12.5)))                        CIRCO       6
       3 FORMAT(6(6X,I6))                                                   CIRCO       7
       4 FORMAT(2(6X,I2))                                                   CIRCO       8
       5 FORMAT(4(6X,F12.5))                                                CIRCO       9
      10 FORMAT(3(6X,F12.5))                                                CIRCO      10
      51 FORMAT(*  DR =*1PE12.5*  DZ =*E12.5*  DT =*E12.5*  R1 =*E12.5*  Z1 CIRCO      11
        1 =*E12.5*  Z1 =*E12.5*  R2 =*E12.5/*  T1 =*E12.5*  T2 =*E12.5* TWL CIRCO      12
        2 =*E12.5* UIN =*E12.5* VIN =*E12.5* PIN =*E12.5*  C1 =*E12.5/* VEL CIRCO      13
        3 =*E12.5* TEM =*E12.5* CPL =*E12.5* PRTI=*E12.5* PRTO=*E12.5* SCN  CIRCO      14
        4=*E12.5*  OM =*E12.5/*  NU =*E12.5* BETA=*E12.5*  GZ =*E12.5*ALPHA CIRCO      15
        5=*E12.5* BTA =*E12.5*TFACT=*E12.5* TCN =*E12.5/* CEP =*E12.5*VCTPL CIRCO      16
        6=*E12.5*  VM =*E12.5*CNPLT=*E12.5* TFIN=*E12.5* TDMP=*E12.5* DTCP= CIRCO      17
```

```
          7*E12.5)
      52 FOR AT(1H0,* IBR =*I5* JBR =*I5* NPR =*I5* NPL =*I5* NWR =*I5* LPR
        1 =*I5* LWR =*I5)
      53 FORMAT(1H0,* NCP =*I5* NPC =*I5)
      54 FORMAT( *    CPC    *4(F12.5))
      55 FORMAT( *    TPL    *F12.5)
      56 FORMAT(1H0,* NIP =*I5* NIC =*I5)
      57 FORMAT( *    IPC    *4(F12.5))
      58 FORMAT( *    UPI    *F12.5*    VPI    *F12.5*    TPI    *F12.5)
      61 FORMAT(1H0,* NOP =*I5* NOC =*I5)
      62 FORMAT( *    OPC    *4(F12.5))
      63 FORMAT( *    UPO    *F12.5*    VPO    *F12.5)
      74 FORMAT(1H0,4X,*I*,5X,*J*,4X,*FL*,11X,*U*,16X,*V*,16X,*P*,16X,*D*,1
        16X,*I*)
      75 FORMAT(3(3X,I3),5(5X,1PE12.5))
      76 FORMAT(1H0,*    IT =*I5)
      80 FORMAT(* TD1 =*1PE12.5* TD2 =*E12.5* TD3 =*E12.5)
     100 FORMAT(1H1)
C***INITIALIZATION PHASE OF CIRCO*****
  2         CALL GETQ(4LKTLM,NTIM)
  4         TLIM=FLOAT(NTIM)
  6         CALL SECOND(TBASE)
  7         TBASE1=TBASE
  7         TF0=8
  7         TFI=7
  7         TD=0.0
  7         TDF=0.0
  7         TIME=0.0
  7         IT=0
  7         NCYC=0
 16         PRINT 100
C***SECTION FOR PROCESSING INPUT DATA*****
 21         READ 1,NAME
 27         PRINT 1,NAME
 35         CALL ADV(1)
 37         WRITE(12,1)NAME
C*****READ DATA WHICH DETERMINES INITIAL INFORMATION SOURCE*****
 45         READ 10,TD1,TD2,TD3
 57         PRINT 80,TD1,TD2,TD3
 71         WRITE (12,80)TD1,TD2,TD3
103         IF(TD1.LT.0.000001) GO TO 7500
106     105 CONTINUE
C*****READ XPUT ARRAY WHICH CONTAINS FLOATING POINT DATA*****
106         READ 2,NUM,(XPUT(N),N=1,NUM)
117         PRINT 51,(XPUT(N),N=1,NUM)
126         WRITE(12,51)(XPUT(N),N=1,NUM)
C*****READ NPUT ARRAY WHICH CONTAINS INTEGER DATA*****
135         READ 3,INUM,(NPUT(N),N=1,INUM)
146         PRINT 52,(NPUT(N),N=1,INUM)
155         WRITE(12,52)(NPUT(N),N=1,INUM)
C***SECTION FOR CALCULATING CONSTANTS*****
164         TLIM=0.9667*TLIM*27.9E-09-30.+(1.0-TDMP)*1.0E+10
173         CALL GETQ(4LKJON,JNM)
175         CALL DATE1(DAT)
177         CALL CLOCK1(CLK)
```

```
201        EM6=1.0E-06                                CIRCO
201        IB1=IBR+1                                  CIRCO
201        IM=IBR+2                                   CIRCO
201        JB1=JBR+1                                  CIRCO
201        JM=JBR+2                                   CIRCO
201        VC=(DR+DZ)/(2.*VM)                         CIRCO
201        RTFCT=1.0/TFACT                            CIRCO
      C*****DR DEPENDENT CONSTANTS*****               CIRCO
201        RDR=1.0/DR                                 CIRCO
201        RRR=1.0/DR**2                              CIRCO
201        RDRH=0.5*RDR                               CIRCO
201        RDRQ=0.25*RDR                              CIRCO
201        RRRH=0.5*RRR                               CIRCO
201        RRR2=2.0*RRR                               CIRCO
201        RT=IBR*DR                                  CIRCO
201        RRT=1.0/RT                                 CIRCO
234        DO 150 I=1,IM                              CIRCO
243        RCC(I)=(I-1.5)*DR                          CIRCO
243        RBC(I)=(I-1.0)*DR                          CIRCO
243    150 CONTINUE                                   CIRCO
250        DO 155 I=2,IM                              CIRCO
257        RRCC(I)=1.0/RCC(I)                         CIRCO
257        RRBC(I)=1.0/RBC(I)                         CIRCO
257    155 CONTINUE                                   CIRCO
      C*****DZ DEPENDENT CONSTANTS*****               CIRCO
263        RDZ=1.0/DZ                                 CIRCO
263        RZZ=1.0/DZ**2                              CIRCO
263        RDZH=0.5*RDZ                               CIRCO
263        RDZQ=0.25*RDZ                              CIRCO
263        RZZH=0.5*RZZ                               CIRCO
263        RZZ2=2.0*RZZ                               CIRCO
263        ZT=JBR*DZ                                  CIRCO
263        RZT=1.0/ZT                                 CIRCO
277        DO 160 J=1,JM                              CIRCO
306        ZCC(J)=(J-1.5)*DZ                          CIRCO
306        ZBC(J)=(J-1.0)*DZ                          CIRCO
306    160 CONTINUE                                   CIRCO
313        RDRZ=RDR*RDZ                               CIRCO
313        DCON=(2.0*CEP)/(DR*DZ)                     CIRCO
      C*****DT DEPENDENT CONSTANTS*****               CIRCO
313        RDT=1.0/DT                                 CIRCO
313        DTRDR=DT*RDR                               CIRCO
313        DTRDZ=DT*RDZ                               CIRCO
313        DTRZ=4.0*DT/(DR+DZ)                        CIRCO
313        TCN1=0.5*RDT*(1.0+TCN)                     CIRCO
313        B=-(0.5*RDT*OM)/(RRR+RZZ)                  CIRCO
313        BTR=BIA*DTRDR                              CIRCO
313        BTZ=BIA*DTRDZ                              CIRCO
      C*****DETERMINING PLOTTING CONSTANTS*****       CIRCO
313        DM=ZT                                      CIRCO
350        IF(ZT.LT.RT) DM=RT                         CIRCO
353        IRL=123                                    CIRCO
353        IRR=RT/DM*900.+123.+EM6                    CIRCO
353        IZT=900.-ZT/DM*900.+EM6                    CIRCO
353        IZB=900                                    CIRCO
```

```
366         IF(TOF.GT.0.0) GO TO 3495
371         NPRT=NPR
371         NPLT=NPL
371         NWRT=NWR
      C***SECTION FOR DETERMINING CELL FLAGS AND INITIAL CELL QUANTITIES*****
375         DO 250 J=1,JM
376         DO 250 I=1,IM
405         U(I,J)=0.0
405         V(I,J)=0.0
405         P(I,J)=0.0
405         T(I,J)=0.0
405         K(I,J)=0.0
405     250 CONTINUE
      C*****FLAGGING SCHEME*****
      C        FL=1  FLUID CELL
      C        FL=2  FREESLIP-INSULATED
      C        FL=3  NOSLIP-INSULATED
      C        FL=4  FREESLIP-CONSTANT TEMPERATURE
      C        FL=5  NOSLIP-CONSTANT TEMPERATURE
      C        FL=6  INFLOW PORT
      C        FL=7  OUTFLOW PORT
413         FC=0
414         IF(VEL.EQ.0.0.AND.TEM.EQ.0.0) FC=2
423         IF(VEL.EQ.1.0.AND.TEM.EQ.0.0) FC=3
433         IF(VEL.EQ.0.0.AND.TEM.EQ.1.0) FC=4
443         IF(VEL.EQ.1.0.AND.TEM.EQ.1.0) FC=5
      C*****DETERMINING FLAGS FOR LEFT FICTITIOUS CELLS*****
452         DO 260 J=1,JM
461         FL(1,J)=2
461     260 CONTINUE
      C*****DETERMINING FLAGS AND TEMPERATURES FOR BOTTOM AND TOP ROWS OF FICT
      C      CELLS*****
463         DO 270 I=1,IM
471         FL(I,1)=FC
471         FL(I,JM)=FC
471         T(I,1)=TWL
471         T(I,JM)=TWL
471     270 CONTINUE
      C*****DETERMINING FLAGS AND CELL QUANTITIES FOR FLUID AND BOUNDARY CELLS
504         RTEST=R2**2
504         ZC=-0.5*DZ
507         DO 300 J=2,JM1
511         ZC=ZC+DZ
512         IF(ZC.GT.ZT) GO TO 301
516         RC=-0.5*DR
520         DO 300 I=2,IM
521         RC=RC+DR
521         RSQ=SCN*(ZC-R2)**2
521         RSQ=RSQ+RC**2
527         IF(RSQ.LE.RTEST) GO TO 290
532         DO 280 II=I,IM
542         FL(II,J)=FC
542         T(II,J)=TWL
542     280 CONTINUE
545         GO TO 300
```

```
 550     290 FL(I,J)=1                                                        CIRCO
 550         P(I,J)=PIN                                                       CIRCO
 550         T(I,J)=T2                                                        CIRCO
 554     300 CONTINUE                                                         CIRCO
 561     301 CONTINUE                                                         CIRCO
       C*****SETTING SPECIAL INTERIOR TEMPERATURES*****                       CIRCO
 561         RTEST=R1**2                                                      CIRCO
 561         ZC=-0.5*DZ                                                       CIRCO
 565         DO 311 J=2,JB1                                                   CIRCO
 566         ZC=ZC+DZ                                                         CIRCO
 566         ZTEST1=Z1-0.5*ZI                                                 CIRCO
 566         ZTEST2=Z1+0.5*ZI                                                 CIRCO
 574         IF(ZC.GT.ZTEST2) GO TO 312                                       CIRCO
 600         IF(ZC.LT.ZTEST1) GO TO 311                                       CIRCO
 602         RC=-0.5*DR                                                       CIRCO
 604         DO 310 I=2,IB1                                                   CIRCO
 605         RC=RC+DR                                                         CIRCO
 605         RSQ=SCN*(ZC-Z1)**2                                               CIRCO
 605         RSQ=RSQ+RC**2                                                    CIRCO
 613         IF(RSQ.LE.RTEST) T(I,J)=T1                                       CIRCO
 622     310 CONTINUE                                                         CIRCO
 625     311 CONTINUE                                                         CIRCO
 630     312 CONTINUE                                                         CIRCO
       C*****SETTING NON-BOUNDARY FLUID CELL VELOCITIIES*****                 CIRCO
 630         DO 325 J=2,JB1                                                   CIRCO
 632         DO 320 I=2,IB1                                                   CIRCO
 633         IF(FL(I,J).NE.1) GO TO 325                                       CIRCO
 637         IF(FL(I+1,J).EQ.1) U(I,J)=UIN                                    CIRCO
 644         IF(FL(I,J+1).EQ.1) V(I,J)=VIN                                    CIRCO
 653     320 CONTINUE                                                         CIRCO
 656     325 CONTINUE                                                         CIRCO
 661         FC1=5                                                            CIRCO
 662         IF(VEL.EQ.0.0) FC1=4                                             CIRCO
 664         IF(CPL.EQ.0.0) GO TO 455                                         CIRCO
       C*****SETTING FLAGS AND TEMPERATURES FOR SPECIAL CONSTANT TEMPERATURE CE CIRCO
 665         READ 4,NCP,NPC                                                   CIRCO
 675         PRINT 53,NCP,NPC                                                 CIRCO
 705         WRITE(12,53)NCP,NPC                                              CIRCO
 715         DO 450 N=1,NCP                                                   CIRCO
 717         M=4*(N-1)+1                                                      CIRCO
 717         M1=M+1                                                           CIRCO
 717         M2=M+2                                                           CIRCO
 717         M3=M+3                                                           CIRCO
 725         READ 5,(CPC(NN),NN=M,M3)                                         CIRCO
 735         PRINT 54,(CPC(NN),NN=M,M3)                                       CIRCO
 745         WRITE(12,54)(CPC(NN),NN=M,M3)                                    CIRCO
 755         READ 5,TPL(N)                                                    CIRCO
 764         PRINT 55,TPL(N)                                                  CIRCO
 773         WRITE(12,55)TPL(N)                                               CIRCO
1002         RC1=CPC(M)                                                       CIRCO
1002         ZC1=CPC(M1)                                                      CIRCO
1002         RC2=CPC(M2)                                                      CIRCO
1002         ZC2=CPC(M3)                                                      CIRCO
1002         ZC=-1.5*DZ                                                       CIRCO
1016         DO 450 J=1,JM                                                    CIRCO
```

```
1020          ZC=ZC+DZ
1020          RC=-0.5*DR
1024          DO 450 I=2,IM
1025          RC=RC+DR
1026          IF(FL(I,J).EQ.1) GO TO 450
1032          IF(RC.LT.RC1.OR.RC.GT.RC2) GO TO 450
1043          IF(ZC.LT.ZC1.OR.ZC.GT.ZC2) GO TO 450
1056          FL(I,J)=FC1
1056          T(I,J)=TPL(N)
1062      450 CONTINUE
1072      455 CONTINUE
1072          IF(PR(I.EQ.0.0) GO TO 555
     C*****SETTING FLAGS,TEMPERATURES AND VELOCITIES FOR INFLOW CELLS*****
1073          READ 4,NIP,NIC
1103          PRINT 56,NIP,NIC
1113          WRITE(12,56)NIP,NIC
1123          DO 550 N=1,NIP
1125          M=4*(N-1)+1
1125          M1=M+1
1125          M2=M+2
1125          M3=M+3
1133          READ 5,(IPC(NN),NN=M,M3)
1143          PRINT 57,(IPC(NN),NN=M,M3)
1153          WRITE(12,57)(IPC(NN),NN=M,M3)
1163          READ 3,UPI(N),VPI(N),TPI(N)
1200          PRINT 58,UPI(N),VPI(N),TPI(N)
1215          WRITE(12,58)UPI(N),VPI(N),TPI(N)
1232          RC1=IPC(M)
1232          ZC1=IPC(M1)
1232          RC2=IPC(M2)
1232          ZC2=IPC(M3)
1232          ZC=-1.5*DZ
1246          DO 550 J=1,JM
1250          ZC=ZC+DZ
1250          RC=-0.5*DR
1254          DO 550 I=2,IM
1255          RC=RC+DR
1256          IF(FL(I,J).EQ.1) GO TO 550
1262          IF(RC.LT.RC1.OR.RC.GT.RC2) GO TO 550
1273          IF(ZC.LT.ZC1.OR.ZC.GT.ZC2) GO TO 550
1306          FL(I,J)=6
1306          T(I,J)=TPI(N)
1311          IF(FL(I-1,J).NE.1) GO TO 500
1317          U(I-1,J)=UPI(N)
1317          V(I,J-1)=VPI(N)
1317          V(I,J)=VPI(N)
1323          GO TO 550
1324      500 IF(FL(I,J-1).NE.1) GO TO 510
1333          V(I,J-1)=VPI(N)
1333          U(I,J)=UPI(N)
1336          IF(I.GT.2) U(I-1,J)=UPI(N)
1343          GO TO 550
1344      510 IF(FL(I,J+1).NE.1) GO TO 550
1353          V(I,J)=VPI(N)
1353          U(I,J)=UPI(N)
```

```
1356        IF(I.GT.2) U(I-1,J)=UPI(N)                                   CIRCO
1363    550 CONTINUE                                                     CIRCO
1373    555 CONTINUE                                                     CIRCO
1373        IF(PRTO.EQ.0.0) GO TO 655                                    CIRCO
     C*****SETTING FLAGS AND VELOCITIES FOR OUTFLOW CELLS*****           CIRCO
1374        READ 4,NOP,NOC                                               CIRCO
1404        PRINT 61,NOP,NOC                                             CIRCO
1414        WRITE(12,61)NOP,NOC                                          CIRCO
1424        DO 659 N=1,NOP                                               CIRCO
1426        M=4*(N-1)+1                                                  CIRCO
1426        M1=M+1                                                       CIRCO
1426        M2=M+2                                                       CIRCO
1426        M3=M+3                                                       CIRCO
1434        READ 5,(OPC(NN),NN=M,M3)                                     CIRCO
1444        PRINT 62,(OPC(NN),NN=M,M3)                                   CIRCO
1454        WRITE(12,62)(OPC(NN),NN=M,M3)                                CIRCO
1464        READ 5,UPO(N),VPO(N)                                         CIRCO
1476        PRINT 63,UPO(N),VPO(N)                                       CIRCO
1510        WRITE(12,63)UPO(N),VPO(N)                                    CIRCO
1522        RC1=OPC(M)                                                   CIRCO
1522        ZC1=OPC(M1)                                                  CIRCO
1522        RC2=OPC(M2)                                                  CIRCO
1522        ZC2=OPC(M3)                                                  CIRCO
1522        ZC=-1.5*DZ                                                   CIRCO
1536        DO 659 J=1,JM                                                CIRCO
1540        ZC=ZC+DZ                                                     CIRCO
1540        RC=-0.5*DR                                                   CIRCO
1544        DO 659 I=2,IM                                                CIRCO
1545        RC=RC+DR                                                     CIRCO
1546        IF(FL(I,J).EQ.1) GO TO 650                                   CIRCO
1552        IF(RC.LT.RC1.OR.RC.GT.RC2) GO TO 650                         CIRCO
1563        IF(ZC.LT.ZC1.OR.ZC.GT.ZC2) GO TO 650                         CIRCO
1574        FL(I,J)=7                                                    CIRCO
1577        IF(FL(I-1,J).NE.1) GO TO 600                                 CIRCO
1605        U(I-1,J)=UPO(N)                                              CIRCO
1605        V(I,J-1)=VPO(N)                                              CIRCO
1605        V(I,J)=VPO(N)                                                CIRCO
1611        GO TO 650                                                    CIRCO
1612    600 IF(FL(I,J-1).NE.1) GO TO 610                                 CIRCO
1621        V(I,J-1)=VPO(N)                                              CIRCO
1621        U(I,J)=UPO(N)                                                CIRCO
1624        IF(I.GT.2) U(I-1,J)=UPO(N)                                   CIRCO
1631        GO TO 650                                                    CIRCO
1632    610 IF(FL(I,J+1).NE.1) GO TO 650                                 CIRCO
1641        V(I,J)=VPO(N)                                                CIRCO
1641        U(I,J)=UPO(N)                                                CIRCO
1644        IF(I.GT.2) U(I-1,J)=UPO(N)                                   CIRCO
1651    650 CONTINUE                                                     CIRCO
1661    655 CONTINUE                                                     CIRCO
     C***DETERMINING FRAME CONSTANTS FOR PLOTTING*****                   CIRCO
1661        ICNT=1                                                       CIRCO
1661        JCNT=1                                                       CIRCO
1661        IRPC(1)=IRL                                                  CIRCO
1661        IZBC(1)=IZB                                                  CIRCO
1661        IZTC(1)=IZT                                                  CIRCO
```

```
1661          J=2                                                          CIRCO
1670          DO 660 I=2,IB1                                               CIRCO
1671          IF(FL(I,J).EQ.1) GO TO 660                                   CIRCO
1675          ICNT=ICNT+1                                                  CIRCO
1675          RPC=(I-2.)*DR                                                CIRCO
1675          IRPC(ICNT)=IRL+RPC*RRT*(IRR-IRL)+EM6                         CIRCO
1711      659 J=J+1                                                        CIRCO
1713          IF(FL(I,J).NE.1) GO TO 659                                   CIRCO
1716          JCNT=JCNT+1                                                  CIRCO
1716          ZPC=(J-2.)*DZ                                                CIRCO
1716          IZBC(JCNT)=IZB+ZPC*RZT*(IZT-IZB)+EM6                         CIRCO
1716          ZPC=ZT-ZPC                                                   CIRCO
1716          IZTC(JCNT)=IZB+ZPC*RZT*(IZT-IZB)+EM6                         CIRCO
1743      660 CONTINUE                                                     CIRCO
1746          ICNT=ICNT+1                                                  CIRCO
1746          IRPC(ICNT)=IRR                                               CIRCO
        C***CALCULATIONAL PHASE OF CIRCO*****                              CIRCO
1751     1000 CONTINUE                                                     CIRCO
        C***DETERMINING MAXIMUM VELOCITY IN SYSTEM*****                    CIRCO
1751          VELMX=0.0                                                    CIRCO
1752          DO 1100 J=2,JB1                                              CIRCO
1754          DO 1100 I=2,IB1                                              CIRCO
1757          UCM=ABS(U(I,J))                                              CIRCO
1757          VCM=ABS(V(I,J))                                              CIRCO
1757          VMX=UCM                                                      CIRCO
1763          IF(VCM.GT.VMX) VMX=VCM                                       CIRCO
1767          IF(VELMX.GT.VMX) GO TO 1100                                  CIRCO
1773          VELMX=VMX                                                    CIRCO
1773          IMX=I                                                        CIRCO
1773          JMX=J                                                        CIRCO
1776     1100 CONTINUE                                                     CIRCO
2003          DTEST=DCON*VELMX+1.0E-09                                     CIRCO
        C***DETERMINING HEAT CONDUCTION COEFFICIENT,K*****                 CIRCO
2006          DO 1200 J=1,JM                                               CIRCO
2010          DO 1200 I=1,IM                                               CIRCO
2011          K(I,J)=C1                                                    CIRCO
2014          IF(FL(I,J).EQ.6) K(I,J)=0.0                                  CIRCO
2022     1200 CONTINUE                                                     CIRCO
        C***DETERMINING TBAR*****                                          CIRCO
2027          DO 1320 J=2,JBR                                              CIRCO
2031          SUM=0.0                                                      CIRCO
2031          RSUM= .0                                                     CIRCO
2032          DO 1300 I=2,IB1                                              CIRCO
2034          IF(FL(I,J).NE.1) GO TO 1310                                  CIRCO
2040          IF(FL(I,J+1).NE.1) GO TO 1310                                CIRCO
2044          TEM1=0.5*(T(I,J)+T(I,J+1))                                   CIRCO
2044          SUM=SUM+RCC(I)*TEM1                                          CIRCO
2044          RSUM=RSUM+RCC(I)                                             CIRCO
2053     1300 CONTINUE                                                     CIRCO
2055     1310 TBAR(J)=SUM/RSUM                                             CIRCO
2060     1320 CONTINUE                                                     CIRCO
        C***SETTING TANGENTIAL VELOCITY BOUNDARY CONDITIONS--FREESLIP OR NOSLIP* CIRCO
2062          I=1                                                          CIRCO
2062          II=IM                                                        CIRCO
2064          DO 1500 J=1,JB1                                              CIRCO
```

```
2066          V(I,J)=V(I+1,J)
2073          IF(FL(II,J).GE.6.OR.FL(II,J+1).GE.6) GO TO 1500
2107          IF(FL(II,J).EQ.3.OR.FL(II,J).EQ.5) GO TO 1400
2120          IF(FL(II-1,J).EQ.1) V(II,J)=V(II-1,J)
2130          GO TO 1500
2131     1400 IF(FL(II-1,J).EQ.1) V(II,J)=-V(II-1,J)
2141     1500 CONTINUE
2144          J=1
2144          JJ=J+1
2146          DO 2000 M=1,2
2147          DO 1800 I=2,IB1
2150          IF(FL(I,J).GE.6.OR.FL(I+1,J).GE.6) GO TO 1800
2164          IF(FL(I,J).EQ.3.OR.FL(I,J).EQ.5) GO TO 1700
2175          IF(FL(I,JJ).EQ.1) U(I,J)=U(I,JJ)
2205          GO TO 1800
2206     1700 IF(FL(I,JJ).EQ.1) U(I,J)=-U(I,JJ)
2217     1800 CONTINUE
2222          J=JM
2222          JJ=J-1
2224     2000 CONTINUE
2226          DO 3000 J=2,JB1
2230          DO 3000 I=2,IB1
2231          IF(FL(I,J).EQ.1.OR.FL(I,J).GE.6) GO TO 3000
2242          IF(FL(I,J).EQ.3.OR.FL(I,J).EQ.5) GO TO 2400
2253          IF(FL(I+1,J).GE.6) GO TO 2300
2257          IF(FL(I,J-1).NE.1) GO TO 2200
2262          U(I,J)=U(I,J-1)
2266          GO TO 2300
2267     2200 IF(FL(I,J+1).EQ.1) U(I,J)=U(I,J+1)
2277     2300 IF(FL(I,J+1).GE.6.OR.FL(I,J+1).EQ.1.OR.FL(I-1,J).NE.1) GO TO 3000
2315          V(I,J)=V(I-1,J)
2322          GO TO 3000
2322     2400 IF(FL(I+1,J).GE.6) GO TO 2600
2327          IF(FL(I,J-1).NE.1) GO TO 2500
2332          U(I,J)=-U(I,J-1)
2336          GO TO 2600
2336     2500 IF(FL(I,J+1).EQ.1) U(I,J)=-U(I,J+1)
2346     2600 IF(FL(I,J+1).GE.6.OR.FL(I,J+1).EQ.1.OR.FL(I-1,J).NE.1) GO TO 3000
2364          V(I,J)=-V(I-1,J)
2371     3000 CONTINUE
         C***SECTION FOR PRINTED OUTPUT*****
2376          IF(NCYC.LE.1) GO TO 3099
2401          IF(NCYC.NE.NPRT) GO TO 3110
2403          NPRT=NPRT+NPR
2404          PRINT 70,TIME,NCYC,IT,VELMX
2420          GO TO (3110,3099), LPR
2426     3099 CONTINUE
           70 FORMAT(5X,*TIME =*1PE12.5,5X,*NCYC =*I5,5X,*IT =*I5,5X,*VELMX =*E1
             12.5)
2426          PRINT 74
2432          DO 3100 J=1,JM
2434          DO 3100 I=1,IM
2435          PRINT 75,I,J,FL(I,J),U(I,J),V(I,J),P(I,J),D(I,J),T(I,J)
2470     3100 CONTINUE
2475     3110 CONTINUE
```

```
      C***SECTION FOR PLOTTED OUTPUT*****
2475        IF(NCYC.LE.1) GO TO 3115
2500        IF(NCYC.NE.NPLT) GO TO 3120
2502        NPLT=NPLT+NPL
2503   3115 CONTINUE
2503        IF(VCTPL.NE.0.0) CALL VCTPLT
2505        IF(CNPLT.NE.0.0) CALL CONTUR
2507   3120 CONTINUE
      C***SECTION FOR WRITTEN OUTPUT*****
2507        WRITE(12,70)TIME,NCYC,IT,VELMX
2523        IF(NCYC.LE.1) GO TO 3125
2526        IF(NCYC.NE.NWRT) GO TO 3130
2530        NWRT=NWRT+NWR
2531        GO TO (3130,3125), LWR
2537   3125 CONTINUE
2537        CALL LINCNT(1)
2541        WRITE(12,70)TIME,NCYC,IT,VELMX
2555        WRITE(12,74)
2561        DO 3126 J=1,JM
2563        DO 3126 I=1,IM
2564        WRITE(12,75)I,J,FL(I,J),U(I,J),V(I,J),P(I,J),D(I,J),T(I,J)
2617   3126 CONTINUE
2624   3130 CONTINUE
2624        TMTST=DTRZ*VELMX
      C***TEST FOR DECREASING DT TO MAINTAIN STABILITY*****
2626        IF(TMTST.LT.0.70) GO TO 3150
2631        DT=RIFCT*DT
2631        DTRZ=2.0*DT/(DR+DZ)
2631        RDT=1.0/DT
2631        DTRDR=DT*RDR
2631        DTRDZ=DT*RDZ
2631        BTR=BTA*DTRDR
2631        BTZ=BTA*DTRDZ
2631        B=-(0.5*RDT*DM)/(RRR+RZZ)
2631        TCN1=0.5*DT*(1.0+TCN)
      C***TEST FOR INCREASING DT WHEN VELOCITIES ARE SMALL*****
2657   3150 IF(TMTST.GT.0.15) GO TO 3160
2663        IF(NCYC.LT.50) GO TO 3160
2665        TMTST=TFACT*DT
2667        IF(TMTST.GT.0.25) GO TO 3160
2673        DT=TFACT*DT
2673        DTRZ=2.0*DT/(DR+DZ)
2673        RDT=1.0/DT
2673        DTRDR=DT*RDR
2673        DTRDZ=DT*RDZ
2673        BTR=BTA*DTRDR
2673        BTZ=BTA*DTRDZ
2673        B=-(0.5*RDT*DM)/(RRR+RZZ)
2673        TCN1=0.5*DT*(1.0+TCN)
2721   3160 CONTINUE
      C***SECTION FOR TAPE DUMPS*****
2721        ASSIGN 3499 TO KTD
2722        IF(TIME+EM6.LT.TFIN) GO TO 3300
2726        ASSIGN 8000 TO KTD
2727        GO TO 7000
```

```
2727   3300 CALL SECOND(TCYC)
2731        IF((TCYC-TBASE).LT.TLIM) GO TO 3310
2735        ASSIGN 8000 TO KTD
2736        GO TO 7000
2736   3310 IF((TCYC-TBASE1).LT.DTCP) GO TO 3499
2742        TBASE1=TCYC
2742        GO TO 7000
2743   3499 CONTINUE
2743        TIME=TIME+DT
2743        NCYC=NCYC+1
      C***SECTION FOR CALCULATING INTERMEDIATE VELOCITIES*****
2747        DO 3699 J=2,JB1
2750        DO 3698 I=2,IB1
2751        IF(FL(I,J).NE.1) GO TO 3699
2755        IF(FL(I+1,J).NE.1) GO TO 3500
      C*****CALCULATION OF U VELOCITIES*****
2762        UCON1=RDR*RRBC(I)*U(I,J)*(RCC(I+1)*U(I+1,J)-RCC(I)*U(I-1,J))
2762        TEM1=0.5*(V(I+1,J)+V(I,J))
2762        EPS=ALPHA*SIGN(1.0,TEM1)+BTZ*TEM1
2762        UCONT=TEM1*((EPS+0.5)*U(I,J)-(EPS-0.5)*U(I,J+1))
2762        TEM1=0.5*(V(I+1,J-1)+V(I,J-1))
2762        EPS=ALPHA*SIGN(1.0,TEM1)+BTZ*TEM1
2762        UCONB=TEM1*((EPS+0.5)*U(I,J-1)-(EPS-0.5)*U(I,J))
2762        UCON2=RDZ*(UCONT-UCONB)
2762        UPRES=RDR*(P(I+1,J)-P(I,J))
2762        UDIF1=RZZ*(U(I,J+1)-2.0*U(I,J)+U(I,J-1))
2762        UDIF2=RDRZ*((V(I+1,J)-V(I,J))-(V(I+1,J-1)-V(I,J-1)))
2762        UT(I,J)=U(I,J)-DT*(UCON1+UCON2+UPRES-NU*(UDIF1-UDIF2))
3051   3500 IF(FL(I,J+1).NE.1) GO TO 3698
      C*****CALCULATION OF V VELOCITIES*****
3061        TEM1=0.5*(U(I,J+1)+U(I,J))
3061        EPS=ALPHA*SIGN(1.0,TEM1)+BTR*TEM1
3061        VCONR=RBC(I)*TEM1*((EPS+0.5)*V(I,J)-(EPS-0.5)*V(I+1,J))
3061        TEM1=0.5*(U(I-1,J+1)+U(I-1,J))
3061        EPS=ALPHA*SIGN(1.0,TEM1)+BTR*TEM1
3061        VCONL=RBC(I-1)*TEM1*((EPS+0.5)*V(I-1,J)-(EPS-0.5)*V(I,J))
3061        VCON1=RDR*RRCC(I)*(VCONR-VCONL)
3061        VCON2=RDZ*V(I,J)*(V(I,J+1)-V(I,J-1))
3061        VPRES=RDZ*(P(I,J+1)-P(I,J))
3061        VDIF1=RDRZ*RRCC(I)*(RBC(I)*(U(I,J+1)-U(I,J))-RBC(I-1)*(U(I-1,J+1)
           1-U(I-1,J)))
3061        VDIF2=RRR*RRCC(I)*(RBC(I)*(V(I+1,J)-V(I,J))-RBC(I-1)*(V(I,J)-
           1V(I-1,J)))
3061        VBOUY=(0.5*(T(I,J+1)+T(I,J))-TBAR(J))
3061        VT(I,J)=V(I,J)-DT*(VCON1+VCON2+VPRES+NU*(VDIF1-VDIF2)+BETA*GZ*
           1VBOUY)
3161   3698 CONTINUE
3164   3699 CONTINUE
3167        DO 3750 J=2,JB1
3170        DO 3749 I=2,IB1
3171        IF(FL(I,J).NE.1) GO TO 3750
3175        IF(FL(I+1,J).EQ.1) U(I,J)=UT(I,J)
3203        IF(FL(I,J+1).EQ.1) V(I,J)=VT(I,J)
      C***PRESSURE-VELOCITY ITERATION SECTION*****
3213   3749 CONTINUE
```

```
3216   3750 CONTINUE
3221        ITMX=1500
3221        IT=0
3223   4000 CONTINUE
3223        ITFL=0
3224        DO 4599 J=2,JM1
3226        DO 4598 I=2,IM1
3227        IF(FL(I,J).NE.1) GO TO 4598
3235        D(I,J)=RDR*RRCC(I)*(RBC(I)*U(I,J)-RBC(I-1)*U(I-1,J))+RDZ*(V(I,J)-
           1V(I,J-1))
3235        DP=R*D(I,J)
3246        IF(FL(I+1,J).EQ.1) U(I,J)=U(I,J)+DTRDR*DP
3257        IF(FL(I,J+1).EQ.1) V(I,J)=V(I,J)+DTRDZ*DP
3267        IF(FL(I-1,J).EQ.1) U(I-1,J)=U(I-1,J)-DTRDR*DP
3277        IF(FL(I,J-1).EQ.1) V(I,J-1)=V(I,J-1)-DTRDZ*DP
3312        P(I,J)=P(I,J)+DP
3312        ABD=ABS(D(I,J))
3315        IF(ABD.LE.DTEST) GO TO 4598
3320        ITFL=1
3321   4598 CONTINUE
3324   4599 CONTINUE
3327        IT=IT+1
3330        IF(IT.EQ.ITMX) GO TO 4799
3332        IF(IT.GT.ITMX) GO TO 7799
3335        IF(ITFL.EQ.1) GO TO 4000
3336        GO TO 5000
3336   4799 CONTINUE
3336        PRINT 76,IT
3344        PRINT 74
3350        DO 4798 J=1,JM
3352        DO 4798 I=1,IM
3353        PRINT 75,I,J,FL(I,J),U(I,J),V(I,J),P(I,J),D(I,J)
3403   4798 CONTINUE
3410        IF(ITFL.EQ.1) GO TO 4000
3412   5000 CONTINUE
      C****SECTION FOR CALCULATING TEMPERATURES*****
3412        DO 5999 J=2,JM1
3414        DO 5998 I=2,IM1
3415        IF(FL(I,J).NE.1) GO TO 5999
3421        IF(FL(I-1,J).NE.1) GO TO 5200
3425        TLFT=T(I-1,J)*U(I-1,J)*RBC(I-1)
3425        TEM1=T(I,J)-T(I-1,J)
3425        TTRNL=(U(I-1,J)**2)*TEM1
3425        TCONL=RBC(I-1)*0.5*(K(I,J)+K(I-1,J))*TEM1
3440        GO TO 5300
3440   5200 CONTINUE
3443        TLFT=T(I,J)*U(I-1,J)*RBC(I-1)
3443        TTRNL=0.0
3443        TCONL=0.0
3447   5300 IF(FL(I+1,J).NE.1) GO TO 5350
3455        TRGT=T(I+1,J)*U(I,J)*RBC(I)
3455        TEM1=T(I+1,J)-T(I,J)
3455        TTRNR=(U(I,J)**2)*TEM1
3455        TCONR=RBC(I)*0.5*(K(I+1,J)+K(I,J))*TEM1
3470        GO TO 5500
```

```
3470  5350 IF(FL(I-1,J).LT.4.OR.FL(I-1,J).GT.6) GO TO 5375
3504       TRGT=(2.0*T(I-1,J)-T(I,J))*U(I,J)*RBC(I)
3504       TEM1=4.0*(T(I-1,J)-T(I,J))
3504       TTRNR=(U(I,J)**2)*TEM1
3504       TCONR=RC(I)*K(I-1,J)*TEM1
3517       GO TO 5500
3517  5375 CONTINUE
3522       TRGT=T(I,J)*U(I,J)*RBC(I)
3522       TTRNR=0.0
3522       TCONR=0.0
3524  5500 IF(FL(I,J-1).NE.1) GO TO 5550
3536       TBTM=T(I,J-1)*V(I,J-1)
3536       TEM1=T(I,J)-T(I,J-1)
3536       TTRNB=(V(I,J-1)**2)*TEM1
3536       TCONB=0.5*(K(I,J)+K(I,J-1))*TEM1
3544       GO TO 5700
3544  5550 IF(FL(I,J-1).LT.4.OR.FL(I,J-1).GT.6) GO TO 5575
3545       TBTM=(2.0*T(I,J-1)-T(I,J))*V(I,J-1)
3540       TEM1=4.0*(T(I,J)-T(I,J-1))
3540       TTRNB=(V(I,J-1)**2)*TEM1
3540       TCONB=K(I,J-1)*TEM1
3547       GO TO 5700
3570  5575 CONTINUE
3573       TBTM=T(I,J)*V(I,J-1)
3573       TTRNB=0.0
3573       TCONB=0.0
3574  5700 IF(FL(I,J+1).NE.1) GO TO 5750
3606       TTOP=T(I,J+1)*V(I,J)
3606       TEM1=T(I,J+1)-T(I,J)
3606       TTRNT=(V(I,J)**2)*TEM1
3606       TCONT=0.5*(K(I,J+1)+K(I,J))*TEM1
3614       GO TO 5900
3614  5750 IF(FL(I,J+1).LT.4.OR.FL(I,J+1).GT.6) GO TO 5775
3630       TTOP=(2.0*T(I,J+1)-T(I,J))*V(I,J)
3630       TEM1=4.0*(T(I,J+1)-T(I,J))
3630       TTRNT=(V(I,J)**2)*TEM1
3630       TCONT=K(I,J+1)*TEM1
3637       GO TO 5900
3640  5775 CONTINUE
3643       TTOP=T(I,J)*V(I,J)
3643       TTRNT=0.0
3643       TCONT=0.0
3646  5900 CONTINUE
3646       TN(I,J)=T(I,J)-DT*(RACC(I)*RDRR*(TRGT-TLFT)+RDZR*(TTOP-TBTM)-TCN1*
          1(RRR*(TTRNR-TTRNL)+RZZ*(TTRNT-TTRNB))-(RACC(I)*RRR*(TCONR-TCONL)+R
          2ZZ*(TCONT-TCONB)))
3702  5998 CONTINUE
3704  5999 CONTINUE
3707       DO 6010 J=2,JM1
3710       DO 6000 I=2,IM1
3711       IF(FL(I,J).NE.1) GO TO 6010
3715       T(I,J)=TN(I,J)
3721  6000 CONTINUE
3723  6010 CONTINUE
3726       GO TO 1000
```

```
3726   7000 TD=TU+1.0                                                CIRCO
       C***SECTION FOR TRANSFERRING INFORMATION TO A TAPE*****        CIRCO
3730        BUFFER OUT(TFO,1) (A1,ZZZ)                               CIRCO
3737   7010 IF(UNIT,TFO) 7010,7011,7100                              CIRCO
3743   7011 BUFFER OUT(TFO,1) (CPC,SPC)                              CIRCO
3752   7012 IF(UNIT,TFO) 7012,7020,7100                              CIRCO
3756   7020 CONTINUE                                                 CIRCO
3756        PRINT 7030,TU,TIME                                       CIRCO
       7030 FORMAT(5X,*TU =*F12.5,5X,*TIME =*F12.5,5X,*TAPE DUMP*)   CIRCO
3766        WRITE (12,7030)TU,TIME                                   CIRCO
3776        GO TO 810                                                CIRCO
4001   7100 WRITE(12,7101)                                           CIRCO
4005        PRINT 7101                                               CIRCO
       7101 FORMAT(1H0,* EOF SENSED ON BUFFER OPERATION*)            CIRCO
4011        GO TO 8000                                               CIRCO
4012   7500 CONTINUE                                                 CIRCO
       C***SECTION FOR TRANSFERRING INFORMATION FROM A TAPE TO CIRCO*****   CIRCO
4012   7510 BUFFER IN(TFI,1) (A1,ZZZ)                                CIRCO
4021   7511 IF(UNIT,TFI) 7511,7512,7100                              CIRCO
4025   7512 BUFFER IN(TFI,1) (CPC,SPC)                               CIRCO
4034   7513 IF(UNIT,TFI) 7513,7520,7100                              CIRCO
4042   7520 IF(TUZ-10) 7600,7525,7510                                CIRCO
4043   7525 CONTINUE                                                 CIRCO
4043        WRITE (12,7530)TU,TIME                                   CIRCO
4053        PRINT 7530,TU,TIME                                       CIRCO
       7530 FORMAT(1H0,5X,*TU =*F12.5,5X,*TIME =*F12.5,5X,* PICK-UP TAPE*)   CIRCO
4063        TOF=1.0                                                  CIRCO
4065        READ 1,NAME                                              CIRCO
4072        IF(TU3.GT.0.0) GO TO 105                                 CIRCO
4075        CALL GETQ(ALKJHN,JNM)                                    CIRCO
4077        CALL DATE1(DAT)                                          CIRCO
4101        CALL CLOCK1(CLK)                                         CIRCO
4103        TLIM=0.9667*TLIM*27.9E-09-30.*(1.0-TDMP)*1.0E+10         CIRCO
4112        GO TO 3499                                               CIRCO
4113   7600 WRITE (12,7601)                                          CIRCO
4117        PRINT 7601                                               CIRCO
       7601 FORMAT(1H0,* TU NUMBER INCORRECT*)                       CIRCO
4123   7799 CONTINUE                                                 CIRCO
4123        PRINT 99                                                 CIRCO
         99 FORMAT(1H0,*     NUMBER OF ITERATIONS EXCEEDS MAXIMUM*)  CIRCO
4127        PRINT 76,IT                                              CIRCO
4135        PRINT 74                                                 CIRCO
4141        DO 7800 J=1,JM                                           CIRCO
4143        DO 7800 I=1,IM                                           CIRCO
4144        PRINT 75,I,J,FL(I,J),U(I,J),V(I,J),P(I,J),D(I,J),T(I,J)  CIRCO
4177   7800 CONTINUE                                                 CIRCO
4204   8000 CONTINUE                                                 CIRCO
4205        END                                                      CIRCO
```

STMT NO=	LOCATION	STMT NO=	LOCATION	STMT NO=	LOCATION	STMT NO=	LOCATION
1	4221	2	4223	3	4230	4	4233
5	4236	10	4241	51	4244	52	4324
53	4335	54	4341	55	4345	56	4351
57	4355	58	4361	61	4370	62	4374
63	4400	70	4461	74	4405	75	4415
76	4421	80	4424	99	4535	100	4432
105	107	290	546	300	555	301	562
310	623	311	626	312	631	320	654
325	657	450	1063	455	1073	500	1325
510	1345	550	1364	555	1374	600	1613
610	1633	650	1652	655	1662	659	1712
660	1744	1000	1752	1100	1776	1200	2022
1310	2055	1400	2131	1500	2141	1700	2206
1800	2217	2200	2267	2300	2277	2400	2322
2500	2336	2600	2346	3000	2371	3099	2426
3110	2475	3115	2503	3120	2507	3125	2537
3130	2624	3150	2657	3160	2721	3300	2727
3310	2736	3499	2743	3500	3051	3698	3161
3699	3164	3749	3213	3750	3216	4000	3223
4598	3321	4599	3324	4799	3336	5000	3412
5200	3440	5300	3447	5350	3470	5375	3517
5500	3526	5550	3544	5575	3570	5700	3576
5750	3614	5775	3640	5900	3646	5999	3704
6010	3723	7000	3726	7010	3737	7011	3743
7012	3752	7020	3756	7030	4477	7100	4001
7101	4505	7500	4012	7510	4012	7511	4021
7512	4025	7513	4034	7520	4040	7525	4043
7530	4513	7600	4112	7601	4531	7799	4122
8000	4203						

BLOCK NAMES AND LENGTHS

AZ	= 76106	SPCLON =	265

VARIABLE ASSIGNMENTS

NAME	=	LOCATION	NAME	=	LOCATION	NAME	=	LOCATION	NAME	=	LOCATION
ABD	=R	4553	ALPHA	=R	74576C01	A1	=R	0C01	A10	=R	74424C01
A2	=R	6564C01	A3	=R	15350C01	A4	=R	24134C01	A5	=R	32720C01
A6	=R	41504C01	A7	=R	50270C01	A8	=R	57054C01	A9	=R	65640C01
B	=R	74715C01	BETA	=R	74574C01	BTA	=R	74577C01	BTR	=R	74671C01
BTZ	=R	74702C01	CEP	=R	74602C01	CLK	=R	74724C01	CNPLT	=R	74605C01
CPC	=R	0C02	CPL	=R	74566C01	C1	=R	74563C01	D	=R	65640C01
DAT	=R	74723C01	DCON	=R	74716C01	DM	=R	74720C01	DP	=R	4554
DR	=R	74546C01	DT	=R	74550C01	DTCP	=R	74610C01	DTEST	=R	74717C01
DTRDR	=R	74705C01	DTRDZ	=R	74706C01	DTRZ	=R	74707C01	DZ	=R	74547C01
EM6	=R	74712C01	EPS	=R	4555	FA	=R	74661C01	FC	=I	74713C01
FC1	=I	74714C01	FL	=I	0C01	GZ	=R	74575C01	I	=I	4556
IA	=I	74743C01	IBR	=I	74630C01	IB1	=I	74745C01	ICNT	=I	74755C01
IFRC	=I	75563C01	II	=I	4557	IM	=I	74746C01	IMX	=I	4560
INUM	=I	74743C01	IPC	=R	50C02	IRL	=I	74751C01	IRPC	=I	75563C01
IRR	=I	74752C01	IT	=I	74764C01	ITFL	=I	4561	ITMX	=I	74765C01
IZR	=I	74753C01	IZBC	=I	75635C01	IZT	=I	74754C01	IZTC	=I	75757C01
J	=I	4562	JBR	=I	74631C01	JB1	=I	74747C01	JCNT	=I	74756C01
JJ	=I	4563	JM	=I	74750C01	JMX	=I	4564	JNM	=I	74763C01

```
K       *R  57054C01     KTD     *I   4565         LPR     *I  74635C01     LWR     *I  74636C01
M       *I   4566        M1      *I   4567         M2      *I   4570        M3      *I   4571
N       *I   4572        NAME    *I  75551C01      NCP     *I  74766C01     NCYC    *I  74757C01
NIC     *I  74771C01     NIP     *I  74770C01      NN      *I   4573        NOC     *I  74773C01
NOP     *I  74772C01     NPC     *I  74767C01      NPL     *I  74633C01     NPLI    *I  74761C01
NPR     *I  74632C01     NPRT    *I  74760C01      NPUT    *I  74630C01     NTIM    *I   4574
NU      *R  74573C01     NUM     *I  74744C01      NWR     *I  74634C01     NWRI    *I  74762C01
OM      *R  74572C01     OPC     *R    120C02      P       *R  41504C01     PIN     *R  74562C01
PRTI    *R  74567C01     PRTO    *R  74570C01      RHC     *R  75077C01     RC      *R   4575
RCC     *R  75025C01     RC1     *R   4576         RC2     *R   4577        RD      *R  75025C01
RDR     *R  74661C01     RDRR    *R  74662C01      RDRQ    *R  74663C01     RDRZ    *R  74703C01
RDT     *R  74704C01     RDZ     *R  74672C01      RDZH    *R  74673C01     RDZU    *R  74674C01
RPC     *R   4600        RRHC    *R  75223C01      RRCC    *R  75151C01     RRR     *R  74664C01
RRRH    *R  74665C01     RRR2    *R  74666C01      RRT     *R  74670C01     RSO     *R   4601
RSUM    *R   4602        RT      *R  74667C01      RTEST   *R   4603        RTFCT   *R  74711C01
RZT     *R  74701C01     RZZ     *R  74675C01      RZZH    *R  74676C01     RZZZ    *R  74677C01
R1      *R  74551C01     R2      *R  74554C01      SCN     *R  74571C01     SPC     *R    264C02
SUM     *R   4604        T       *R  50270C01      TBAR    *R  74424C01     TBASE   *R   4605
TBASE1  *R   4606        TBIM    *R   4607         TCN     *R  74601C01     TCNI    *R  74726C01
TCONR   *R   4610        TCONL   *R   4611         TCONR   *R   4612        TCONT   *R   4613
TCYC    *R   4614        TD      *R  74725C01      TDF     *R   4615        TDMP    *R  74607C01
TD1     *R   4616        TD2     *R   4617         TD3     *R   4620        TEM     *R  74565C01
TEND    *R   4621        TFACT   *R  74600C01      TFI     *I   4622        TFIN    *R  74606C01
TFO     *I   4623        TIME    *R  74710C01      TLFT    *R   4624        TLIM    *R   4625
TMTST   *R   4626        TN      *R  32720C01      TPI     *R    214C02     TPL     *R    226C02
TRST    *R   4627        TTOP    *R   4630         TTRNH   *R   4631        TTRNL   *R   4632
TTRNR   *R   4633        TTRNI   *R   4634         TWL     *R  74557C01     T1      *R  74555C01
T2      *R  74556C01     U       *R   6564C01      UCM     *R   4635        UCONR   *R   4636
UCONT   *R   4637        UCON1   *R   4640        UCON2   *R   4641        UDIF1   *R   4642
UDIF2   *R   4643        UIN     *R  74560C01      UPI     *R    170C02     UPO     *R    240C02
UPRES   *R   4644        UT      *R  15350C01      V       *R  24134C01     VROUT   *R   4645
VC      *R  74721C01     VCM     *R   4646         VCONL   *R   4647        VCONR   *R   4650
VCON1   *R   4651        VCON2   *R   4652         VCTPL   *R  74603C01     VDIF1   *R   4653
VDIF2   *R   4654        VFL     *R  74564C01      VELMX   *R  74722C01     VIN     *R  74561C01
VM      *R  74604C01     VMX     *R   4655         VPI     *R    202C02     VPO     *R    252C02
VPRES   *R   4656        VT      *R  32720C01      XPUT    *R  74546C01     ZAC     *R  75421C01
ZC      *R   4657        ZCC     *R  75277C01      ZC1     *R   4660        ZC2     *R   4661
ZD      *R  75277C01     ZI      *R  74553C01      ZPC     *R   4662        ZT      *R  74700C01
ZTEST1  *R   4663        ZTEST2  *R   4664         ZZZ     *R  76105C01     Z1      *R  74552C01

EXTERNAL ASSIGNMENTS
GETQ    *R               SECOND  *R                OUTPTC  *R               INPUTC  *I
ADV     *R               DATE1   *R                CLOCK1  *R               ACGUER  *R
VCTPLT  *R               CONTUR  *R                LINCNT  *I               BUFFEO  *R
IOCHEK  *I               BUFFEI  *R                END     *R               QRNTRY

START OF      -      CONSTANTS      TEMPORARIES      INDIRECTS      -      UNUSED COMPILER SPACE
                       4205            4544             4551                     53000
```

```
          SUBROUTINE VCTPLT                                                      VCTPLT
1         COMMON/A// A1(3444),A2(3444),A3(3444),A4(3444),A5(3444),A6(3444),A COMM
         17(3444),A8(3444),A9(3444),A10(82),XPUT(50),NPUT(25),FA(50),IA(50), COMM
         2RD(170),ZD(170),NAME(10),IFRC(210),ZZZ                              COMM
1         COMMON/SPCLUN/ CPC(40),IPC(40),UPC(40),UPI(10),VPI(10),TPI(10),TPL COMM
         1(10),UPO(10),VPO(10),SPC                                            COMM
1         EQUIVALENCE (A1,FL),(A2,U),(A3,UT),(A4,V),(A5,VT),(A6,P),(A7,T),(A COMM
         18,K),(A9,D),(A10,THAR),(A5,TN),(RD,RCC),(RD(43),RRC),(RD(85),RRCC) COMM
         2,(RD(127),RRHC),(ZD,ZCC),(ZD(83),ZBC),(IFRC,IRPC),(IFRC(43),IZBC), COMM
         3(IFRC(125),IZTC)                                                    COMM
1         EQUIVALENCE (FA(1),RDR),(FA(2),RDRR),(FA(3),RURU),(FA(4),RRR),(FA( COMM
         15),RRRR),(FA(6),RRR2),(FA(7),RT),(FA(8),RRT),(FA(9),RTR),(FA(10),R COMM
         2DZ),(FA(11),RDZR),(FA(12),RDZD),(FA(13),RZZ),(FA(14),RZZR),(FA(15) COMM
         3,RZZZ),(FA(16),ZT),(FA(17),RZT),(FA(18),RTZ),(FA(19),RDRZ),(FA(20) COMM
         4,RDT),(FA(21),DTRDR),(FA(22),DTRDZ),(FA(23),DTRZ),(FA(24),TIME),(F COMM
         5A(25),RIFCT),(FA(26),EM6),(FA(27),FC),(FA(28),FC1),(FA(29),B),(FA( COMM
         630),UCON),(FA(31),DTEST),(FA(32),D4),(FA(33),VC),(FA(34),VELMX),(F COMM
         7A(35),DAT),(FA(36),CLK),(FA(37),TD),(FA(38),TCN1)                   COMM
1         EQUIVALENCE (IA(1),INUM),(IA(2),NUM),(IA(3),IM1),(IA(4),IM),(IA(5) COMM
         1,JM1),(IA(6),JM),(IA(7),IRL),(IA(8),IRR),(IA(9),IZB),(IA(10),IZT), COMM
         2(IA(11),ICNT),(IA(12),JCNT),(IA(13),NCYC),(IA(14),NPRT),(IA(15),NP COMM
         3LT),(IA(16),NWRT),(IA(17),JNM),(IA(18),IT),(IA(19),ITMX),(IA(20),N COMM
         4CP),(IA(21),NPC),(IA(22),NTP),(IA(23),NIC),(IA(24),NOP),(IA(25),NO COMM
         5C)                                                                  COMM
1         EQUIVALENCE (XPUT(1),DR),(XPUT(2),DZ),(XPUT(3),DT),(XPUT(4),R1),(X COMM
         1PUT(5),Z1),(XPUT(6),ZI),(XPUT(7),R2),(XPUT(8),T1),(XPUT(9),T2),(XP COMM
         2UT(10),TWL),(XPUT(11),UIN),(XPUT(12),VIN),(XPUT(13),PIN),(XPUT(14) COMM
         3,C1),(XPUT(15),VEL),(XPUT(16),TEM),(XPUT(17),CPL),(XPUT(18),PRTI), COMM
         4(XPUT(19),PRTO),(XPUT(20),SCN),(XPUT(21),ON),(XPUT(22),NU),(XPUT(2 COMM
         53),BETA),(XPUT(24),GZ),(XPUT(25),ALPHA),(XPUT(26),RTA),(XPUT(27),T COMM
         6FACT),(XPUT(28),TCN),(XPUT(29),CEP),(XPUT(30),VCTPL),(XPUT(31),VM) COMM
         7,(XPUT(32),CNPLT),(XPUT(33),TFIN),(XPUT(34),IUMP),(XPUT(35),DTCP)   COMM
1         EQUIVALENCE (NPUT(1),IRR),(NPUT(2),JRR),(NPUT(3),NPR),(NPUT(4),NPL COMM
         1),(NPUT(5),NWR),(NPUT(6),LPR),(NPUT(7),LWR)                         COMM
1         INTEGER IFI,IFO                                                     COMM
1         INTEGER FL,FC,FC1                                                   COMM
1         REAL IPC,K,NU                                                       COMM
1         DIMENSION U(32,62),UT(32,62),V(32,62),VT(32,62),P(32,62),T(32,62), COMM
         1TN(32,62),K(32,62),FL(32,62),THAR(62),D(32,62)                      COMM
1         DIMENSION RCC(32),RHC(32),RRCC(32),RRHC(32),ZCC(62),ZBC(62)         COMM
1         DIMENSION IRPC(32),IZBC(62),IZTC(62)                                COMM
        1 FORMAT(1H0,18X,10A8,1X,A10,2(1X,A8))                                VCTPLT
        2 FORMAT(1H0,18X,* VELOCITY VECTORS *15X,*TIME =*1PE12.5,1X,*CYCLE = VCTPLT
         1*I5)                                                                VCTPLT
        3 FORMAT(1H0,18X,* CONSTANT SCALING FACTOR *15X,*U HORIZONTAL V VERT VCTPLT
         1ICAL*)                                                              VCTPLT
        4 FORMAT(1H0,18X,* VARIABLE SCALING FACTOR *5X,*VELMX =*1PE12.5,5X,* VCTPLT
         1U HORIZONTAL V VERTICAL*)                                           VCTPLT
      C***SUBROUTINE FOR PLOTTING VELOCITY VECTORS*****                       VCTPLT
1         NPLOT=0                                                             VCTPLT
2     110 CONTINUE                                                            VCTPLT
2         NPLOT=NPLOT+1                                                       VCTPLT
      C*****SECTION FOR DRAWING FRAME*****                                    VCTPLT
4         CALL ADV(1)                                                         VCTPLT
5         ICNM=ICNT-1                                                         VCTPLT
```

```
  5          JCNM=JCNT-1                                                  VCTPLT
 10          DO 150 L=1,ICNM                                              VCTPLT
 12          CALL DRV(IRPC(L),IZHC(L),IRPC(L+1),IZHC(L))                  VCTPLT
 15          CALL DRV(IRPC(L),IZTC(L),IRPC(L+1),IZTC(L))                  VCTPLT
 21      150 CONTINUE                                                     VCTPLT
 24          DO 160 L=1,JCNM                                              VCTPLT
 25          CALL DRV(IRPC(L+1),IZHC(L),IRPC(L+1),IZHC(L+1))              VCTPLT
 31          CALL DRV(IRPC(L+1),IZTC(L),IRPC(L+1),IZTC(L+1))              VCTPLT
 36      160 CONTINUE                                                     VCTPLT
 41          CALL DRV(IRPC(1),IZHC(1),IRPC(1),IZTC(1))                    VCTPLT
 44          CALL DRV(IRPC(ICNT),IZHC(JCNT),IRPC(ICNT),IZTC(JCNT))        VCTPLT
 51          CALL LINCNT(5H)                                              VCTPLT
 53          GO TO (190,290),NPLOT                                        VCTPLT
       C*****SECTION FOR PLOTTING CONSTANT SCALED VELOCITY VECTORS*****   VCTPLT
 61      190 WRITE(12,2)TIME,NCYC                                         VCTPLT
 71          WRITE(12,3)                                                  VCTPLT
 75          WRITE(12,1)NAME,JNM,DAT,CLK                                  VCTPLT
111          DO 210 J=2,JB1                                               VCTPLT
113          DO 200 I=2,IB1                                               VCTPLT
114          IF(FL(I,J).NE.1) GO TO 210                                   VCTPLT
122          UVEC=RCC(I)+VC*0.5*(U(I,J)+U(I-1,J))                         VCTPLT
122          VVEC=ZCC(J)+VC*0.5*(V(I,J)+V(I,J-1))                         VCTPLT
122          IR=IRL+RCC(I)*RRT*(IRR-IRL)+EM6                              VCTPLT
122          IZ=IZB+ZCC(J)*RZT*(IZT-IZB)+EM6                              VCTPLT
122          IU=IRL+UVEC*RRT*(IRR-IRL)+EM6                                VCTPLT
122          IV=IZB+VVEC*RZT*(IZT-IZB)+EM6                                VCTPLT
171          CALL DRV(IR,IZ,IU,IV)                                        VCTPLT
174      200 CONTINUE                                                     VCTPLT
177      210 CONTINUE                                                     VCTPLT
202          IF(VELMX.GT.0.0) GO TO 110                                   VCTPLT
204          GO TO 350                                                    VCTPLT
       C*****SECTION FOR PLOTTING VARIABLE SCALED VELOCITY VECTORS*****   VCTPLT
204      290 WRITE(12,2)TIME,NCYC                                         VCTPLT
214          WRITE(12,4)VELMX                                             VCTPLT
222          WRITE(12,1)NAME,JNM,DAT,CLK                                  VCTPLT
236          VSC=0.5*(DR+DZ)/VELMX                                        VCTPLT
242          DO 310 J=2,JB1                                               VCTPLT
244          DO 300 I=2,IB1                                               VCTPLT
245          IF(FL(I,J).NE.1) GO TO 310                                   VCTPLT
256          UVEC=RCC(I)+VSC*0.5*(U(I,J)+U(I-1,J))                        VCTPLT
256          VVEC=ZCC(J)+VSC*0.5*(V(I,J)+V(I,J-1))                        VCTPLT
256          IR=IRL+RCC(I)*RRT*(IRR-IRL)+EM6                              VCTPLT
256          IZ=IZB+ZCC(J)*RZT*(IZT-IZB)+EM6                              VCTPLT
256          IU=IRL+UVEC*RRT*(IRR-IRL)+EM6                                VCTPLT
256          IV=IZB+VVEC*RZT*(IZT-IZB)+EM6                                VCTPLT
322          CALL DRV(IR,IZ,IU,IV)                                        VCTPLT
325      300 CONTINUE                                                     VCTPLT
330      310 CONTINUE                                                     VCTPLT
333      350 CONTINUE                                                     VCTPLT
333          RETURN                                                       VCTPLT
334          END                                                          VCTPLT
```

427

STATEMENT ASSIGNMENTS

STMT NO*	LOCATION	STMT NO*	LOCATION	STMT NO*	LOCATION	STMT NO*	LOCATION
1 *	337	2 *	344	3 *	354	4 *	364
110 *	3	190 *	62	210 *	200	290 *	205
310 *	331	350 *	334				

BLOCK NAMES AND LENGTHS

AZ *	76106	SPCLON *	265

VARIABLE ASSIGNMENTS

NAME	*	LOCATION	NAME	*	LOCATION	NAME	*	LOCATION	NAME	*	LOCATION
ALPHA	*R	74576C01	A1	*R	0C01	A10	*R	74424C01	A2	*R	6564C01
A3	*R	15350C01	A4	*R	24134C01	A5	*R	32720C01	A6	*R	41504C01
A7	*R	50270C01	A8	*R	57054C01	A9	*R	65640C01	B	*R	74715C01
BETA	*R	74574C01	BTA	*R	74577C01	BTR	*R	74671C01	BTZ	*R	74702C01
CEP	*R	74602C01	CLK	*R	74724C01	CNPLT	*R	74605C01	CPC	*R	0C02
CPL	*R	74566C01	C1	*R	74563C01	D	*R	65640C01	DAT	*R	74723C01
DCON	*R	74716C01	DM	*R	74720C01	DR	*R	74546C01	DT	*R	74550C01
DTCP	*R	74610C01	DTEST	*R	74717C01	DTRDR	*R	74705C01	DTRDZ	*R	74706C01
DTRZ	*R	74707C01	DZ	*R	74547C01	EM6	*R	74712C01	FA	*R	74661C01
FC	*I	74713C01	FC1	*I	74714C01	FL	*I	0C01	GZ	*R	74575C01
I	*I	410	IA	*I	74743C01	IBR	*I	74630C01	IB1	*I	74745C01
ICNM	*I	411	ICNT	*I	74755C01	IFRC	*I	75563C01	IM	*I	74746C01
INUM	*I	74743C01	IPC	*R	50C02	IR	*I	412	IRL	*I	74751C01
IRPC	*I	75563C01	IRR	*I	74752C01	IT	*I	74764C01	ITMX	*I	74765C01
IU	*I	413	IV	*I	414	IZ	*I	415	IZH	*I	74753C01
IZRC	*I	75635C01	IZT	*I	74754C01	IZTC	*I	75757C01	J	*I	416
JBR	*I	74631C01	JB1	*I	74747C01	JCNM	*I	417	JCNT	*I	74756C01
JM	*I	74750C01	JNM	*I	74763C01	K	*R	57054C01	L	*I	420
LPR	*I	74635C01	LWR	*I	74636C01	NAME	*I	75551C01	NCP	*I	74766C01
NCYC	*I	74757C01	NIC	*I	74771C01	NIP	*I	74770C01	NOC	*I	74773C01
NOP	*I	74772C01	NPC	*I	74767C01	NPL	*I	74633C01	NPLOT	*I	421
NPLT	*I	74761C01	NPR	*I	74632C01	NPRT	*I	74760C01	NPUT	*I	74630C01
NJ	*R	74573C01	NUM	*I	74744C01	NWR	*I	74634C01	NWRT	*I	74762C01
OM	*R	74572C01	OPC	*R	120C02	P	*R	41504C01	PIN	*R	74562C01
PRTI	*R	74567C01	PRTO	*R	74570C01	RBC	*R	75077C01	RCC	*R	75025C01
RD	*R	75025C01	RDR	*R	74661C01	RDRR	*R	74662C01	RDRU	*R	74663C01
RDRZ	*R	74703C01	RDT	*R	74704C01	RDZ	*R	74672C01	RDZR	*R	74673C01
RDZQ	*R	74674C01	RRBC	*R	75223C01	RRCC	*R	75151C01	RRR	*R	74664C01
RRRR	*R	74665C01	RRR2	*R	74666C01	RRT	*R	74670C01	RT	*R	74667C01
RTFCT	*R	74711C01	RZT	*R	74701C01	RZZ	*R	74675C01	RZZR	*R	74676C01
RZZ2	*R	74677C01	R1	*R	74551C01	R2	*R	74554C01	SCN	*R	74571C01
T	*R	50270C01	TBAR	*R	74424C01	TCN	*R	74601C01	TCN1	*R	74726C01
TD	*R	74725C01	TOMP	*R	74607C01	TEM	*R	74565C01	TFACT	*R	74600C01
TFI	*I	422	TFIN	*R	74606C01	TFO	*I	423	TIME	*R	74710C01
TN	*R	32720C01	TPI	*R	214C02	TPL	*R	226C02	TWL	*R	74557C01
T1	*R	74555C01	T2	*R	74556C01	U	*R	6564C01	UIN	*R	74560C01
UPI	*R	170C02	UPO	*R	240C02	UT	*R	15350C01	UVEC	*R	424
V	*R	24134C01	VC	*R	74721C01	VCTPL	*R	74603C01	VEL	*R	74564C01
VELMX	*R	74722C01	VIN	*R	74561C01	VM	*R	74604C01	VPI	*R	202C02
VPO	*R	252C02	VSC	*R	425	VT	*R	32720C01	VVEC	*R	426
XPUT	*R	74546C01	ZRC	*R	75421C01	ZCC	*R	75277C01	ZD	*R	75277C01

```
ZI      *R  74553C01        ZI      *P  74700C01        Z1      *R  74552C01

EXTERNAL ASSIGNMENTS
ADV     *R                  DRV     *R                  LINCNT *I                  ACGOER *R
OUTPTC  *R

START OF       -       CONSTANTS       TEMPORARIES       INDIRECTS       -       UNUSED COMPILER SPACE
                          336              401               407                        67600
```

```
          SUBROUTINE CONTUR                                                   CONTUR      2
 1        COMMON/A2/ A1(3444),A2(3444),A3(3444),A4(3444),A5(3444),A6(3444),A  COMM        2
         17(3444),A8(3444),A9(3444),A10(82),XPUT(50),NPUT(25),FA(50),IA(50),  COMM        3
         2RD(170),ZD(170),NAME(10),IFRC(210),ZZZ                              COMM        4
 1        COMMON/SPCLUN/ CPC(40),IPC(40),OPC(40),UPI(10),VPI(10),TPI(10),TPL  COMM        5
         1(10),UPO(10),VPO(10),SPC                                            COMM        6
 1        EQUIVALENCE (A1,FL),(A2,U),(A3,UT),(A4,V),(A5,VT),(A6,P),(A7,T),(A  COMM        7
         18,K),(A9,D),(A10,TBAR),(A5,TN),(RD,RCC),(RD(43),RBC),(RD(85),RRCC)  COMM        8
         2,(RD(127),RRBC),(ZD,ZCC),(ZD(83),ZBC),(IFRC,IRPC),(IFRC(43),IZBC),  COMM        9
         3(IFRC(125),IZTC)                                                    COMM       10
 1        EQUIVALENCE (FA(1),RDR),(FA(2),RDRH),(FA(3),RDRQ),(FA(4),RRR),(FA(  COMM       11
         15),RRRH),(FA(6),RRR2),(FA(7),RT),(FA(8),RRT),(FA(9),BTR),(FA(10),R  COMM       12
         2DZ),(FA(11),RDZH),(FA(12),RDZQ),(FA(13),RZZ),(FA(14),RZZH),(FA(15)  COMM       13
         3,RZZ2),(FA(16),ZT),(FA(17),RZT),(FA(18),BTZ),(FA(19),RDRZ),(FA(20)  COMM       14
         4,RDT),(FA(21),UTRDR),(FA(22),DTRDZ),(FA(23),DTRZ),(FA(24),TIME),(F  COMM       15
         5A(25),RTFCT),(FA(26),EM6),(FA(27),FC),(FA(28),FC1),(FA(29),B),(FA(  COMM       16
         630),UCON),(FA(31),DTEST),(FA(32),DM),(FA(33),VC),(FA(34),VELMX),(F  COMM       17
         7A(35),DAT),(FA(36),CLK),(FA(37),TD),(FA(38),TCN1)                   COMM       18
 1        EQUIVALENCE (IA(1),INUM),(IA(2),NUM),(IA(3),IB1),(IA(4),IM),(IA(5)  COMM       19
         1,JB1),(IA(6),JM),(IA(7),IRL),(IA(8),IRR),(IA(9),IZB),(IA(10),IZT),  COMM       20
         2(IA(11),ICNT),(IA(12),JCNT),(IA(13),NCYC),(IA(14),NPRT),(IA(15),NP  COMM       21
         3LT),(IA(16),NWRT),(IA(17),JNM),(IA(18),IT),(IA(19),ITMX),(IA(20),N  COMM       22
         4CP),(IA(21),NPC),(IA(22),NIP),(IA(23),NIC),(IA(24),NOP),(IA(25),NO  COMM       23
         5C)                                                                  COMM       24
 1        EQUIVALENCE (XPUT(1),DR),(XPUT(2),DZ),(XPUT(3),DT),(XPUT(4),R1),(X  COMM       25
         1PUT(5),Z1),(XPUT(6),ZI),(XPUT(7),R2),(XPUT(8),T1),(XPUT(9),T2),(XP  COMM       26
         2UT(10),IWL),(XPUT(11),UIN),(XPUT(12),VIN),(XPUT(13),PIN),(XPUT(14)  COMM       27
         3,C1),(XPUT(15),VEL),(XPUT(16),TEM),(XPUT(17),CPL),(XPUT(18),PRTI),  COMM       28
         4(XPUT(19),PRTO),(XPUT(20),SCN),(XPUT(21),OM),(XPUT(22),NU),(XPUT(2  COMM       29
         53),BETA),(XPUT(24),GZ),(XPUT(25),ALPHA),(XPUT(26),BTA),(XPUT(27),T  COMM       30
         6FACT),(XPUT(28),TCN),(XPUT(29),CEP),(XPUT(30),VCTPL),(XPUT(31),VM)  COMM       31
         7,(XPUT(32),CNPLT),(XPUT(33),TFIN),(XPUT(34),TUMP),(XPUT(35),DTCP)   COMM       32
 1        EQUIVALENCE (NPUT(1),IBR),(NPUT(2),JBR),(NPUT(3),NPR),(NPUT(4),NPL  COMM       33
         1),(NPUT(5),NWR),(NPUT(6),LPR),(NPUT(7),LWR)                         COMM       34
 1        INTEGER TFI,IFO                                                     COMM       35
 1        INTEGER FL,FC,FC1                                                   COMM       36
 1        REAL IPC,K,NU                                                       COMM       37
 1        DIMENSION U(32,62),UT(32,62),V(32,62),VT(32,62),P(32,62),T(32,62),  COMM       38
         1TN(32,62),K(32,62),FL(32,62),TBAR(62),D(32,62)                      COMM       39
 1        DIMENSION RCC(32),RBC(32),RRCC(32),RRBC(32),ZCC(62),ZBC(62)         COMM       40
 1        DIMENSION IRPC(32),IZBC(62),IZTC(62)                                COMM       41
 1        DIMENSION CQ(32,62),CON(21),R(4),Z(4),RP(2),ZP(2)                   CONTUR      4
    C***SUBROUTINE FOR PLOTTING TEMPERATURE AND PRESSURE CONTOURS*****        CONTUR      5
 1        NPLOT=0                                                             CONTUR      6
 2    150 CONTINUE                                                            CONTUR      7
 2        NPLOT=NPLOT+1                                                       CONTUR      8
 4        GO TO (200,230,750),NPLOT                                           CONTUR      9
12    200 DO 210 J=1,JM                                                       CONTUR     10
14        DO 210 I=1,IM                                                       CONTUR     11
22        CQ(I,J)=P(I,J)                                                      CONTUR     12
22    210 CONTINUE                                                            CONTUR     13
27        GO TO 250                                                           CONTUR     14
27    230 DO 240 J=1,JM                                                       CONTUR     15
31        DO 240 I=1,IM                                                       CONTUR     16
37        CQ(I,J)=T(I,J)                                                      CONTUR     17
```

```
37      240 CONTINUE
44      250 CONTINUE
44          CQMN=1.0E+10
44          CQMX=-1.0E+10
47          DO 305 J=2,JB1
50          DO 300 I=2,IB1
51          IF(FL(I,J).NE.1) GO TO 305
56          CQMN=AMIN1(CQ(I,J),CQMN)
56          CQMX=AMAX1(CQ(I,J),CQMX)
65      300 CONTINUE
67      305 CONTINUE
72          DCQ=0.05*(CQMX-CQMN)
74          IF(DCQ.LE.0.00001) GO TO 600
77          DO 310 II=1,21
105         CON(II)=CQMN+(II-1)*DCQ
105     310 CONTINUE
      C*****SECTION FOR DRAWING FRAME*****
111         CALL ADV(1)
113         ICNM=ICNT-1
113         JCNM=JCNT-1
116         DO 320 L=1,ICNM
120         CALL DRV(IRPC(L),IZBC(L),IRPC(L+1),IZBC(L))
123         CALL DRV(IRPC(L),IZTC(L),IRPC(L+1),IZTC(L))
127     320 CONTINUE
132         DO 330 L=1,JCNM
133         CALL DRV(IRPC(L+1),IZBC(L),IRPC(L+1),IZBC(L+1))
137         CALL DRV(IRPC(L+1),IZTC(L),IRPC(L+1),IZTC(L+1))
144     330 CONTINUE
147         CALL DRV(IRPC(1),IZBC(1),IRPC(1),IZTC(1))
152         CALL DRV(IRPC(ICNT),IZBC(JCNT),IRPC(ICNT),IZTC(JCNT))
157         CALL LINCNT(58)
161         GO TO (350,360),NPLOT
167     350 WRITE(12,1)TIME,NCYC
177         GO TO 400
200     360 WRITE(12,2)TIME,NCYC
210     400 CONTINUE
210         WRITE(12,3)CQMN,CQMX,DCQ
222         WRITE(12,4)NAME
          1 FORMAT(1H0,18X,* ISOBARS *15X,*TIME =*1PE12.5,1X,*CYCLE =*I5)
          2 FORMAT(1H0,18X,* ISOTHERMS*15X,*TIME =*1PE12.5,1X,*CYCLE =*I5)
          3 FORMAT(1H0,18X,* CQMN =*1PE12.5* CQMX =*E12.5* DCQ =*E12.5)
          4 FORMAT(1H0,18X,10A8)
      C*****SECTION FOR DETERMINING CONTOUR VALUES*****
230         DO 550 J=2,JBR
232         Z(1)=Z(2)=ZCC(J)
235         Z(3)=Z(4)=ZCC(J+1)
237         DO 540 I=2,IBR
241         R(1)=R(3)=RCC(I)
244         R(2)=R(4)=RCC(I+1)
246         IF(FL(I+1,J).NE.1.OR.FL(I+1,J+1).NE.1) GO TO 550
260         DO 530 II=1,21
261         KK=22-II
262         K1=K2=K3=K4=0
266         IF(CQ(I,J).LE.CON(KK))K1=1
274         IF(CQ(I+1,J).LE.CON(KK))K2=1
```

```
301         IF(CQ(I,J+1).LE.CON(KK))K3=1
306         IF(CQ(I+1,J+1).LE.CON(KK))K4=1
313         IF(K1*K2*K3*K4.NE.0.OR.K1+K2+K3+K4.EQ.0) GO TO 530
325         NP=0
325         IF(K1+K3.NE.1) GO TO 450
330         L1=1
330         L2=3
332         I1=I2=I
334         J1=J
334         J2=J+1
336         ASSIGN 450 TO KRT
337         GO TO 500
340     450 IF(K1+K2.NE.1) GO TO 460
343         L1=1
343         L2=2
343         I1=I
343         I2=I+1
347         J1=J2=J
352         ASSIGN 460 TO KRT
353         GO TO 500
353     460 IF(K2+K4.NE.1) GO TO 470
356         L1=2
356         L2=4
360         I1=I2=I+1
362         J1=J
362         J2=J+1
364         ASSIGN 470 TO KRT
365         GO TO 500
366     470 IF(K3+K4.NE.1) GO TO 530
371         L1=3
371         L2=4
371         I1=I
371         I2=I+1
375         J1=J2=J+1
400         ASSIGN 530 TO KRT
411     500 NP=NP+1
      C*****SECTION FOR DRAWING CONTOUR LINES*****
411         XM=(CON(KK)-CQ(I1,J1))/(CQ(I2,J2)-CQ(I1,J1))
411         RP(NP)=R(L1)+XM*(R(L2)-R(L1))
411         ZP(NP)=Z(L1)+XM*(Z(L2)-Z(L1))
426         IF(NP.NE.2) GO TO KRT
433         IR1=123.+(RP(1))*(IRR-123)*RRT+EM6
433         IZ1=900.+(ZP(1))*(IZT-900)*RZT+EM6
433         IR2=123.+(RP(2))*(IRR-123)*RRT+EM6
433         IZ2=900.+(ZP(2))*(IZT-900)*RZT+EM6
467         CALL DRV(IR1,IZ1,IR2,IZ2)
472         IF(KK.EQ.1) CALL PLT(IR1,IZ1,35)
477         IF(KK.EQ.21) CALL PLT(IR1,IZ1,24)
504     530 CONTINUE
506     540 CONTINUE
511     550 CONTINUE
514     600 CONTINUE
514         GO TO 150
515     750 CONTINUE
515         RETURN
```

516 END CONTUR 128

SUBPROGRAM LENGTH - CONTUR
4566

STATEMENT ASSIGNMENTS

STMT NO*	LOCATION	STMT NO*	LOCATION	STMT NO*	LOCATION	STMT NO*	LOCATION
1 *	527	2 *	536	3 *	545	4 *	554
150 *	3	200 *	13	230 *	30	250 *	45
305 *	67	350 *	167	360 *	200	400 *	210
450 *	340	460 *	353	470 *	366	500 *	401
530 *	504	550 *	511	600 *	514	750 *	515

BLOCK NAMES AND LENGTHS
AZ * 76106 SPCLON * 265

VARIABLE ASSIGNMENTS

NAME	*	LOCATION	NAME	*	LOCATION	NAME	*	LOCATION	NAME	*	LOCATION
ALPHA	*R	74576C01	A1	*R	0C01	A10	*R	74424C01	A2	*R	6564C01
A3	*R	15350C01	A4	*R	24134C01	A5	*R	32720C01	A6	*R	41504C01
A7	*R	50270C01	A8	*R	57054C01	A9	*R	65640C01	B	*R	74715C01
BETA	*R	74574C01	BTA	*R	74577C01	BTR	*R	74671C01	BTZ	*R	74702C01
CEP	*R	74602C01	CLK	*R	74724C01	CNPLT	*R	74605C01	CON	*R	567
CPC	*R	0C02	CPL	*R	74566C01	CQ	*R	614	COMN	*R	4514
COMX	*R	4515	C1	*R	74563C01	D	*R	65640C01	DAT	*R	74723C01
DCON	*R	74716C01	DCQ	*R	4516	DM	*R	74720C01	DR	*R	74546C01
DT	*R	74550C01	DTCP	*R	74610C01	DTEST	*R	74717C01	DTRDR	*R	74705C01
DTRDZ	*R	74706C01	DTRZ	*R	74707C01	DZ	*R	74547C01	FM6	*R	74712C01
FA	*R	74661C01	FC	*I	74713C01	FC1	*I	74714C01	FL	*I	0C01
GZ	*R	74575C01	I	*I	4517	IA	*I	74743C01	IBR	*I	74630C01
IB1	*I	74745C01	ICNM	*I	4520	ICNT	*I	74755C01	IFRC	*I	75563C01
II	*I	4521	IM	*I	74746C01	INUM	*I	74743C01	IPC	*R	50C02
IRL	*I	74751C01	IRPC	*I	75563C01	IRR	*I	74752C01	IR1	*I	4522
IR2	*I	4523	IT	*I	74764C01	ITMX	*I	74765C01	IZB	*I	74753C01
IZRC	*I	75635C01	IZT	*I	74754C01	IZTC	*I	75757C01	IZ1	*I	4524
IZ2	*I	4525	I1	*I	4526	I2	*I	4527	J	*I	4530
JB4	*I	74631C01	JB1	*I	74747C01	JCNM	*I	4531	JCNT	*I	74756C01
JM	*I	74750C01	JNM	*I	74763C01	J1	*I	4532	J2	*I	4533
K	*R	57054C01	KK	*I	4534	KRT	*I	4535	K1	*I	4536
K2	*I	4537	K3	*I	4540	K4	*I	4541	L	*I	4542
LPR	*I	74635C01	LWR	*I	74636C01	L1	*I	4543	L2	*I	4544
NAME	*I	75551C01	NCP	*I	74766C01	NCYC	*I	74757C01	NIC	*I	74771C01
NIP	*I	74770C01	NOC	*I	74773C01	NOP	*I	74772C01	NP	*I	4545
NPC	*I	74767C01	NPL	*I	74633C01	NPLOT	*I	4546	NPLT	*I	74761C01
NPR	*I	74632C01	NPRT	*I	74760C01	NPUT	*I	74630C01	NU	*R	74573C01
NUM	*I	74744C01	NWR	*I	74634C01	NWRT	*I	74762C01	OM	*R	74572C01
OPC	*R	120C02	P	*R	41504C01	PIN	*R	74562C01	PRTI	*R	74567C01
PRTO	*R	74570C01	R	*R	4547	RBC	*R	75077C01	RCC	*R	75025C01
RD	*R	75025C01	RDR	*R	74661C01	RDRH	*R	74662C01	RDRQ	*R	74663C01
RDRZ	*R	74703C01	RDT	*R	74704C01	RDZ	*R	74672C01	RDZH	*R	74673C01
RDZQ	*R	74674C01	RP	*R	4553	RRBC	*R	75223C01	RRCC	*R	75151C01
RRR	*R	74664C01	RRRH	*R	74665C01	RRR2	*R	74666C01	RRT	*R	74670C01