COO-3086-19

Report No. 60

Columbia University
in the City of New York

# NUMERICAL STUDIES OF SHOCK FOCUSING IN A COAXIAL ELECTROMAGNETIC SHOCK TUBE

by

Hung Chang Lui

1973

Plasma Laboratory

School of Engineering and Applied Science

Columbia University

New York, N.Y. 10027

MASTER

## DISCLAIMER

# NUMERICAL STUDIES OF SHOCK FOCUSING
# IN A COAXIAL ELECTROMAGNETIC SHOCK TUBE

Hung Chang Lui



Submitted in Partial fulfillment
of the requirements for
The Degree of
Doctor of Engineering Science
in the Faculty of Engineering & Applied Science
Columbia University
1973



lg

ABSTRACT

The generation of a strong transverse MHD shock wave in a coaxial electromagnetic shock tube is simulated numerically. Further plasma heating is achieved by shock reflection and focusing on the end wall. The details of the two-dimensional axisymmetric flow in the shock tube and the process of shock focusing and reflection from a curved end wall are studied. In general, the drive currents can be arbitrary functions of time, and the end wall can be of arbitrary shape.

The scheme used is the fluid-in-cell method (FLIC), modified for compressible, time-dependent magnetohydrodynamic flows to include a magnetic field perpendicular to the flow directions. The governing equations are the two-dimensional axisymmetric, ideal, single fluid MHD equations coupled with Maxwell's equations. Finite differencing is carried out for cells of variable shapes to accomodate solid walls and free surfaces. On a moving vacuum-plasma interface, a mixed Lagrangian-Eulerian procedure is used.

To optimize the wall shape for shock focusing, the location of the focused region and the wall shape are selected according to Whitham's shock-ray theory. The comparisons of the computed results between different shapes of the end wall are made to maximize the plasma heating. The most efficient shock focusing and plasma heating occurs with a quarter-elliptic end wall, giving a reflected and focused shock temperature some 8 times the incident post-shock temperature, or more than twice that obtained from a flat wall reflection. The corresponding density

in the focused region is 16 times the initial density in the shock tube.

To check the accuracy of the method, computations for a plane shock wave reflecting from a semi-circular wall in two-dimensional plane flows are also made using both the FLIC method and the two-step Lax-Wendroff scheme. The two methods agree well.

# ACKNOWLEDGEMENTS

I am deeply indebted to my thesis advisors, Prof. C.K. Chu and Prof. R.T. Taussig, for their constant instruction and enlightening discussions on this research.

I am also grateful to Prof. R.A. Gross and Dr. Y.G. Chen for their help in supplying experimental data and measurements for the shock tube. To Dr. W.P. Gula and Dr. M.S. Chu, I express my thanks for their generosity in making time available for discussing computer programming at the beginning of this research. It is my pleasure to thank all members of the Plasma Physics Department for their cooperation.

I wish to thank my wife, Wenfang Wang, for her encouragement and support.

I also acknowledge the help of Mrs. Ruth Williams in correcting any grammatical errors in the original manuscript, and the expert typing of this report by Mary Black.

The research was supported by the United States Atomic Energy Commission under Contract AT(11-1)3086.

TABLE OF CONTENTS

# I. INTRODUCTION

Experimental studies of strong magnetically driven shock waves in a coaxial electromagnetic shock tube with a transverse field have been conducted for the past several years at Columbia University[1-4]. At the same time, numerical simulations of the MHD shock tube with artificial dissipation[5] and physical dissipation[6] in one-dimensional flows were also carried out. These studies are analogous to the Hain-Roberts code for pinches[7]. However, since the nature of the flow in the coaxial shock tube is obviously two-dimensional, it is important to study the flow two-dimensionally. A numerical computation of the problem will give us interesting and useful insight into the details of the flow in the tube. It is our main purpose to investigate plasma heating numerically by shock focusing and reflection from curved end walls for temperatures of fusion interest.

The general experimental setup of the shock tube is shown schematically in Fig. 4-1. The working gas fills the space of the cylindrical annulus. A bias current flows in the central rod of the inner conductor to give an azimuthal field ahead of the shock. When the circuit is closed, the capacitor banks supply a radial current between two cylindrical electrodes passing through the working gas. This current sheet heats and pushes the gas ahead, creating a shock wave that runs down the tube. To reflect the shock wave, the end wall of the tube can be designed to form a proper profile so as to produce dense plasma heating in kilovolts by shock focusing and reflection.

A number of numerical codes (see e.g. 8,9) of multi-dimensional, time-dependent, compressible flows in hydrodynamics have been developed and applied successfully in the past. Yet it is not easy to apply them directly to MHD flows with arbitrary wall shapes or with a vacuum interface. The two-step Lax-Wendroff scheme was applied by Potter[10] in plasma focus studies, in which he handled the moving vacuum-plasma interface by a minimum-density control. But this technique cannot be readily applied to MHD flows, because magnetic fields exist in both the vacuum and plasma regions, and the computed magnetic fields near the vacuum boundary will become completely inaccurate. If not properly tailored, the Lax-Wendroff scheme[11] may also induce numerical instabilities on a curved wall boundary. It has been suggested by many authors, e.g. Lapidus[12], to transform the curved wall into a straight boundary. Such variable transformations are usually very laborious to carry out even in simple geometries, and it may become prohibitive in more complicated geometries. The particle-in-cell method (PIC)[13] and the grid-and-particle method (GAP)[14] use particle-control in a surface cell to move the free boundary under an applied pressure, but they invariably give large fluctuations to the calculated results.

We have chosen to use the fluid-in-cell method (FLIC)[15], which is originally an Eulerian finite difference scheme with mass transport differencing for solving unsteady compressible flow problems[16]. But, since the system of hydrodynamic equations of a perfectly conducting fluid in the presence of a transverse magnetic field is reducible to that of ordinary fluid dynamical equations, we adapt the method rather easily. Moreover, as the magnetic field is frozen into the plasma, it can be treated in much the same way as the density.

But in the actual calculations of a plasma flow in MHD approximations, it may require some special procedures to handle boundary conditions on a curved rigid wall boundary and on a moving vacuum-plasma interface. Also for curved walls most methods to improve accuracy will result in tiny cells on the boundary. It is troublesome to handle these tiny irregular cells in the actual computations, and they cause stability problems. Since the moving vacuum-plasma interface is Lagrangian in nature, the Eulerian treatments mentioned above are less accurate. A mixed Lagrangian-Eulerian procedure is taken to handle the free boundary. Therefore, we extend the method to cells of variable shapes on solid boundaries and vacuum interfaces.

To choose an end wall shape for shock focusing, we use Whitham's shock-ray theory[17,18]. This is an approximate procedure in which the particle paths are approximated by rays perpendicular to the shock surfaces. It has been applied to various gas dynamical problems by numerous authors such as Milton[19], Lau[20], and Skews[21]. We extend this method to transverse MHD flows and use it to select end wall shapes. One of the best shapes of the end wall is a quarter-ellipse.

Chapter II summarizes the basic MHD equations and Maxwell's equations for the assumed physical model, and describes the modified fluid-in-cell method for MHD flows with a transverse magnetic field. The method is designed particularly for surface cells of different shapes having irregular neighbor cells; but usually, it is also applicable to the regular inner cells located inside the computational region.

Chapter III describes the numerical boundary conditions associated with the code. The Lagrangian contour is introduced to handle the free surface and appropriate reflective boundary conditions are applied to curved walls. The cell-mixing procedures to increase stability are given as well.

In Chapter IV the results of shock formation in a coaxial shock tube are given. The shapes of the incident shocks are obtained. While the shock is almost planar, the current sheet is strongly inclined. Hence, one dimensional results are inadequate for their description.

In Chapter V the shock waves are utilized to study their reflection from flat end walls. The solution of shock convergence and reflection from a semi-circular end wall is also obtained. Usually, a shock wave will be diffracted and converged by a curved end wall when it moves along the wall surface.

In Chapter VI, Whitham's shock-ray approximation is reviewed, and extended to transverse MHD flows. End wall shapes are selected for shock focusing and reflection.

In Chapter VII, shock focusing and reflection from a quarter-elliptic end wall is presented. The shock wave will be strengthened by the elliptic end wall to maximum limit and will reflect. Of the three shapes studied, the quarter-elliptic end wall is the most efficient shape for plasma heating, giving a reflected and focused shock temperature some 8 times the incident post-shock temperature, or more than twice that obtained from a flat wall reflection.

In Chapter VIII, we test the FLIC method by studying shock convergence and reflection from a semi-circular wall in plane transverse MHD flows, and comparing it with results from the two-step Lax-Wendroff scheme[22]. The agreement between the two methods is very good, which gives us some confidence in the accuracy of our basic code.

# II. THE FLIC MHD CODE

## 2.1 The Governing Equations

The plasma in the shock tube is assumed to be a perfect gas with infinite electrical conductivity. The fluid equations under the MHD approximation are chosen for the plasma flows. Although a two-fluid model has been used in one-dimensional flows[6], we do not feel it is easily extended to two-dimensional flows. In this study, therefore, we shall content ourselves to study only the geometrical aspects of shock focusing and plasma heating and a one fluid model is much simpler, at least as a first step.

In the simplest model, the plasma is considered as an ideal, single fluid of density $\rho$, pressure $P$, specific internal energy $I$, specific heat ratio $\gamma$, and velocity $\underset{\sim}{U}$, carrying a current $\underset{\sim}{J}$, in a magnetic field $\underset{\sim}{B}$ and electrical field $\underset{\sim}{E}$. The electrostatic forces and the displacement currents are negligible. The viscosity and heat conductivity are also neglected in the plasma. Maxwell's equations in Gaussian units become

$$\nabla \cdot \underset{\sim}{B} = 0, \tag{2.1}$$

$$\nabla \cdot \underset{\sim}{J} = 0, \tag{2.2}$$

$$\nabla \times \underset{\sim}{E} = -\frac{1}{c}\frac{\partial \underset{\sim}{B}}{\partial t}, \tag{2.3}$$

$$\nabla \times \underset{\sim}{B} = 4\pi \underset{\sim}{J}/c . \tag{2.4}$$

Ohm's law is

$$\underset{\sim}{E} + \frac{1}{c}\underset{\sim}{U} \times \underset{\sim}{B} = 0 . \tag{2.5}$$

The fluid equations are

$$\left(\frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla\right) \rho = -\rho\nabla \cdot \mathbf{U} , \tag{2.6}$$

$$\rho\left(\frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla\right) \mathbf{U} = -\nabla P + \mathbf{J} \times \mathbf{B} / c , \tag{2.7}$$

$$\rho\left(\frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla\right) I = -P\nabla \cdot \mathbf{U} , \tag{2.8}$$

$$P = (\gamma - 1) \rho I . \tag{2.9}$$

Suppose that the flow is plane $\left(\frac{\partial}{\partial z} = 0\right)$ or axisymmetric $\left(\frac{\partial}{\partial \theta} = 0\right)$ and the velocity is given initially to be in the (x-y) plane or (z-r) plane, and the magnetic field is initially perpendicular to it:

$$\begin{aligned} \mathbf{U} &= (U_x, U_y, U_z) \text{ or } (U_z, U_r, U_\theta) \\ &= (u , v , 0) , \\ \mathbf{B} &= (B_x, B_y, B_z) \text{ or } (B_z , B_r, B_\theta) \\ &= (0, 0 , B), \quad (\text{at } t = 0) \end{aligned} \tag{2.10}$$

then such a flow will remain two-dimensional (plane (x-y) or axisymmetric (z-r)) at all times,

$$\begin{aligned} \mathbf{U} &= (u , v , 0), \\ \mathbf{B} &= (0 , 0 , B). \quad (\text{for all } t) \end{aligned} \tag{2.11}$$

Since the magnetic field is in the z-direction or θ-direction, the acceleration of the plasma and the electrical field ($\mathbf{E} = -\frac{1}{c} \mathbf{U} \times \mathbf{B}$) in that direction will be zero. The statement (2.11) easily follows from Eqs. (2.3) and (2.7).

For two-dimensional plane or axisymmetric geometry , Eqs. (2.3) to (2.8) can be expressed in the form, with $\underset{\sim}{U} = (u,v,0)$ and $\underset{\sim}{B} = (0,0,B)$.

$$\frac{\partial \rho}{\partial t} = - \operatorname{div} (\rho \underset{\sim}{U}) , \tag{2.12}$$

$$\rho\left(\frac{\partial}{\partial t} + \underset{\sim}{U} \cdot \nabla\right) u = -\frac{\partial}{\partial z} (P + B^2/8\pi) , \tag{2.13}$$

$$\rho\left(\frac{\partial}{\partial t} + \underset{\sim}{U} \cdot \nabla\right) v = -\frac{\partial}{\partial r} (P + B^2/8\pi) - Cb \frac{B^2}{4\pi r} , \tag{2.14}$$

$$\rho\left(\frac{\partial}{\partial t} + \underset{\sim}{U} \cdot \nabla\right) I = - P \operatorname{div} ( \underset{\sim}{U} ) , \tag{2.15}$$

$$\frac{\partial}{\partial t} (B / r^m ) = - \operatorname{div} (\underset{\sim}{U} B / r^m ) , \tag{2.16}$$

and

$$\operatorname{div} ( \underset{\sim}{U} ) = \frac{1}{r^m} \frac{\partial}{\partial r} (r^m v) + \frac{\partial u}{\partial z} , \tag{2.17}$$

where Cb and m are equal to 1 in an axisymmetric case; equal to 0 in a plane case.

We assume that the gas just in front of a strong shock wave is already preionized. The MHD equations and Maxwell's equations apply everywhere. Actually, after a shock reflection from the end wall, the temperature of plasma is about 10 Kev, and the ion cyclotron frequency becomes larger than the ion collision frequency. The fluid approximation there is poor. We shall continue to use the fluid approximation in this study, but will be aware of these possible defects.

## 2.2 The Computing Cells

In order to carry out the computation, the volume containing the fluid is subdivided into a number of cells. The cells in the interior region are equal in shape, and we will call them regular cells. The cells on the boundaries are irregular in shape, and we will call them irregular cells. The computational cells are shown in Fig. 2-1. The location of a cell is denoted by the indices (i,j).

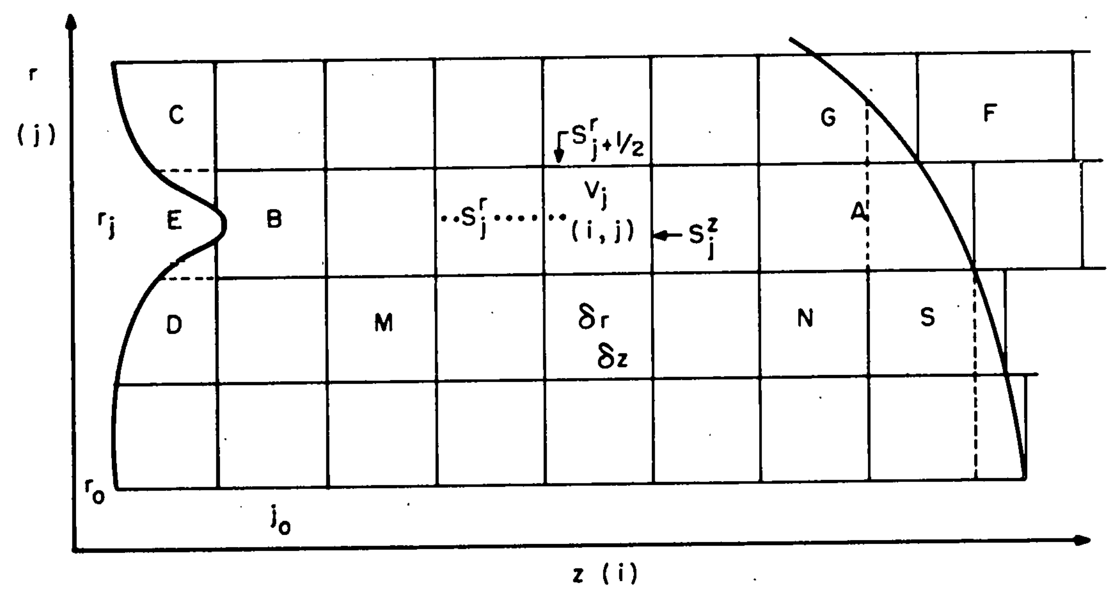



Fig. 2-1 The computing cells in (z-r) coordinates

A regular cell (i,j) in plane Cartesian coordinates is a rectangle with dimension $\delta z$, $\delta r$. In cylindrical coordinates, it is a rectangular torus with inner and outer radii $(r_o + (j-j_o - 1)\ \delta r)$ and $(r_o + (j-j_o)\ \delta r)$ respectively, and width

$\delta z$. The constants $r_o$ and $j_o$ are shown in Fig. 2-1. The regular cells have the properties shown in Table 2-1.

Table 2-1

Geometrical properties of a regular cell (i,j) in Fig. 2-1

| Properties | Plane Coordinates | Cylindrical Coordinates |
|---|---|---|
| Volume ($V_j$) | $\delta r \delta z$ | $2\pi(r_o+(j-j_o-1/2)\delta r)\delta r\delta z$ |
| Area ($S^z{}_j$) | $\delta r$ | $2\pi(r_o+(j-j_o-1/2)\delta r)\delta r$ |
| Area ($S^r{}_j$) | $\delta z$ | $2\pi(r_o+(j-j_o-1/2)\delta r)\delta z$ |
| Area ($S^r{}_{j+1/2}$) | $\delta z$ | $2\pi(r_o+(j-j_o)\delta r)\delta z$ |

Here $S^z{}_j$ is the area on the right or left side of the cell (i,j) in z-direction, $S^r{}_j$ is the area on the middle section of the cell (i,j) in r-direction, $S^r{}_{j+\frac{1}{2}}$ is the area on the top side of the cell (i,j) in r-direction, and $V_j$ is the volume of the cell (i,j).

The cells on the curved wall surface, although irregular, will not change their shapes because the wall boundary is fixed. On the other hand, the free boundary moves in space, and the cells on the free surface will change their sizes and configurations during the computational cycles. Since the motion of the free boundary must be included in the computation, these boundary cells require special treatment. The details of the resolutions on a free surface and on a curved rigid wall will be given in Chapter III.

There are a number of irregular small cells present on a vacuum-plasma interface and on a curved wall boundary. These tiny cells will limit the time increment ($\delta t$) in the computation. To avoid this problem, we join them to their adjacent cells. The combination of a tiny cell and its adjacent cell makes the resulting cell bigger than a normal cell. The combined cell on a boundary may be in contact with two or three different neighbor cells on one of its four faces. These boundary cells are irregular in shape, they have different volumes and widths. They are still regarded as rectangular in shape, but their volumes are the geometric values of the actual configurations.

In Fig. 2-1, the left curve represents the vacuum-plasma interface, and the right curve represents the curved wall boundary. The cell (B) is in contact with three different cells (C), (D), and (E) on its left side. The cell (A) is in contact with two different cells (F) and (G) on its top side; two cells (S) and (N) on its bottom side. A cell on a boundary may have two or three neighbor cells on one of its four faces. In order to consider all geometric situations, a cell $(i,j)$ is assumed to be in contact with eight neighbor cells $(i+1, j+1)$, $(i+1, j)$, $(i+1, j-1)$, $(i, j+1)$, $(i, j-1)$, $(i-1, j+1)$, $(i-1, j)$ and $(i-1, j-1)$ as shown in Fig. 2-2.

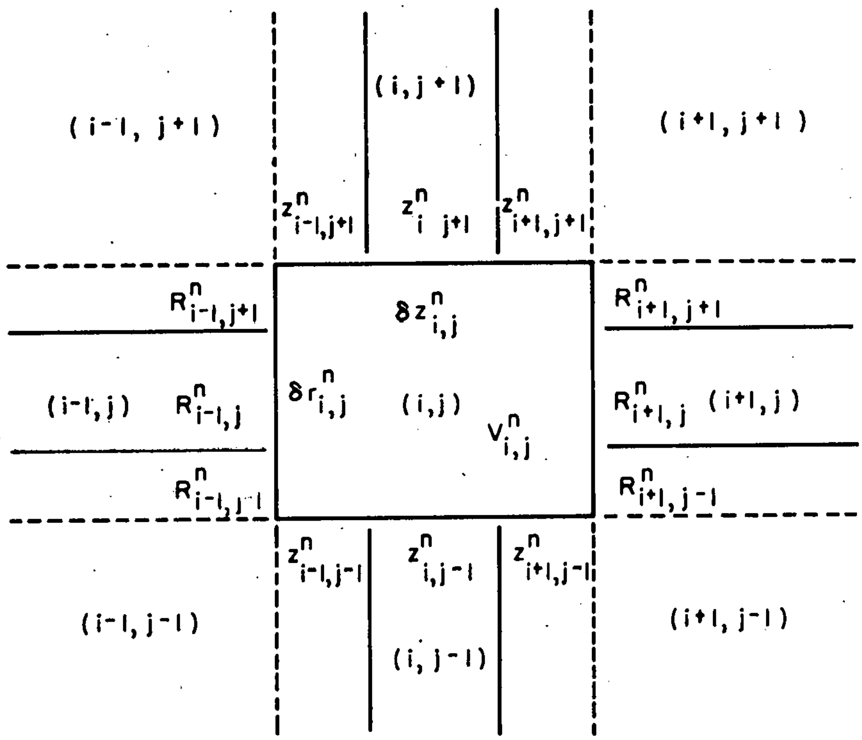



Fig. 2-2 The center cell (i, j) and its surrounding neighbors.

A 907-ME-2718

In Fig. 2-2, the center cell (i,j) has three neighbor cells on each of its four faces. The volumes and widths and configurations of all related cells are given on a wall boundary, or computed on a free boundary so that the geometric relations between them are known in advance. Therefore, the contacting surfaces of the surrounding neighbor cells on each face of the cell (i,j) are known. If a surrounding neighbor cell, say $(i+1,\ j+1)$, does not appear on a surface of the center cell, its contacting length, say $(R^n_{i+1,\ j+1})$, on that surface is set to zero. In case of a normal situation, the center cell (i,j) is surrounded by four regular neighbors; the following relations hold for the center cell and its surrounding neighbors:

$$V^n_{i,j} = V_j$$

$$\delta r^n_{i,j} = \delta r^n_{i+1,j} = R^n_{i+1,j} = \delta r\ ,$$

$$= \delta r^n_{i-1,j} = R^n_{i-1,j} = \delta r,$$

$$\delta z^n_{i,j} = \delta z^n_{i,j+1} = Z^n_{i,j+1} = \delta z\ ,$$

$$= \delta z^n_{i,j-1} = Z^n_{i,j-1} = \delta z\ ,$$

$$Z^n_{i+s,\ j\pm 1} = R^n_{i\pm 1,\ j+t'} = 0, \quad \text{with } s = +1,\ -1, \text{ and } t' = +1,\ -1.$$

The lengths $R^n_{i+s,j+t'}$ and $Z^n_{i+s,\ i+t'}$ are shown in Fig. 2-2.

## 2.3 Summary of the Fluid-in-cell (FLIC) Method

A version of the FLIC method is summarized here which is originally designed for time-dependent, one-fluid problems in plane or axisymmetric compressible flows. Since the system of equations for transverse MHD flows is reducible to that for conventional compressible flows,[23] the FLIC method is modified to include a magnetic field perpendicular to the flow directions in magnetohydrodynamics.

The particle-in-cell (PIC) method[13] is a combined Eulerian-Lagrangian scheme to solve multifluid problems in time-dependent, compressible flows. It utilizes Lagrangian particles in Eulerian cells to transport mass, momentum, and energy. The FLIC method[15] is an Eulerian scheme, in which the difference equations are similar to those used in the PIC method. However, it employs a mass transport calculation which eliminates the use of particles.

We describe the axisymmetric case, the plane case being obvious and simpler. The basic flow variables to characterize the state of the plasma in transverse MHD flows are the density $\rho$, the $z$ and $r$ components of the fluid velocity, $u$ and $v$, the magnetic field $B$, and the specific internal energy $I$. Other flow quantities are determined through their related equations from the given variables. The values of density, magnetic field, velocities, specific internal energy, and other flow quantities are given in each computing cell initially. Then, their values in each cell are advanced in time using the finite difference equations. The FLIC scheme consists of two steps for each cycle: First, intermediate values are calculated for the velocities

($\tilde{u}$, $\tilde{v}$) and specific internal energy ($\tilde{I}$) due to the effects of fluid accelerations driven by the pressure gradients and the electromagnetic forces. Second, mass transport effects are calculated due to the movement of the fluid in each cell. The mass and the magnetic flux which flow from a cell (i,j) to a cell (i + 1,j) are directly proportional to the density and magnetic field of the donor cell (i,j) respectively.

The mass and magnetic flux flowing across the cell boundaries during the time increment are determined by using the intermediate values of the fluid velocities ($\tilde{u}$ and $\tilde{v}$). The mass which has crossed the cell boundaries then carries the momentum and energy corresponding to the intermediate values of the fluid velocities ($\tilde{u}$ and $\tilde{v}$) and the specific internal energy ($\tilde{I}$). The final values for $\rho$, B, u, v, and I are computed by redistributing the mass, magnetic flux, momentum, and energy in each cell. Throughout the steps, there is conservation of mass, momentum and energy.

## 2.4 Finite Difference Equations

Assume that at time $t = n\,\delta t$ the following values are specified for a cell (i,j) shown in Fig. 2-2.

| | |
|---|---|
| density: | $\rho^n_{i,j}$ |
| z-velocity component: | $u^n_{i,j}$ |
| r-velocity component: | $v^n_{i,j}$ |
| specific internal energy: | $I^n_{i,j}$ |
| specific total energy: | $E^n_{i,j} = I^n_{i,j} + \frac{1}{2}\,(u^2 + v^2)^n_{i,j}$ |
| magnetic field: | $B^n_{i,j}$ |
| volume: | $V^n_{i,j}$ |
| widths: | $\delta r^n_{i,j}$ and $\delta z^n_{i,j}$ |
| fluid properties of its neighbors: | $\rho^n_{i+s,j+t'}$ ,etc. |
| | $s=1,\ 0,\ +1;\ t'=-1,0,+1,$ (except $s=0$ and $t'=0$) |
| contacting lengths of its neighbors: | $R^n_{i\pm1,j+t'}$, $Z^n_{i+s,j\pm1}$ |
| | $s=-1,0,+1$ and $t'=-1,0,+1$ |

The differencing procedures of the FLIC method for the cell (i,j) in Fig. 2-2 consist of two steps to calculate values for time $t = (n+1)\,\delta t$.

Step 1. The intermediate values of $\tilde{u}$, $\tilde{v}$, $\tilde{I}$ and $\tilde{E}$ due to the effects of fluid accelerations are calculated, taking into account the pressure gradients and the magnetic forces.

$$\tilde{u}^n_{i,j} = u^n_{i,j} - \delta t\, A^z_{i,j}\, ((P* + q)^n_{i+1/2,j} - (P* + q)^n_{i-1/2,j})$$

$$/ \, (\rho^n_{i,j}\, f_{i,j}\, \delta z)\,,$$

$$\tilde{v}^n_{i,j} = v^n_{i,j} - \delta t\, A^r_{i,j}\, ((S^r_{j+1/2}\, (P*^n_T - P*^n_{i,j})$$

$$-S^r_{j-1/2}\, (P*^n_B - P*^n_{i,j}))\, / \, (2\, V_j)$$

$$+ (q^n_{i,j+1/2} - q^n_{i,j-1/2})/\delta r)/(\rho^n_{i,j}\, f_{i,j}) - Cb(B^2)^n_{i,j}\, \delta t/\, (4\pi r_j \rho^n_{i,j})\,.$$

Here, $P* = P + B^2/8\pi$ ,

$$A^z_{i,j} = \delta r^n_{i,j} \, / \, \delta r\,,$$

$$A^r_{i,j} = \delta z^n_{i,j} \, / \, \delta z,$$

$$f_{i,j} = V^n_{i,j} \, / \, V_j\,,$$

$$P*^n_{i+1/2,j} = (P*^n_{i,j} + P*^n_R)\, / \, 2\,,$$

$$P*^n_{i-1/2,j} = (P*^n_{i,j} + P*^n_L)\, / \, 2\,,$$

$$P*^n_{\substack{R\\L}} = \left( \sum_{s=-1}^{+1} P*^n_{i\pm 1,j+s}\, R^n_{i\pm 1,j+s} \right) / \, \delta r^n_{i,j}\,,$$

$$P*^n_{\substack{T\\B}} = \left( \sum_{s=-1}^{+1} P*^n_{i+s,j\pm 1}\, Z^n_{i+s,j\pm 1} \right) / \, \delta z^n_{i,j}\,,$$

where $P^{*n}_{Y}$ is the total pressure of the surrounding neighbors acting on the cell (i,j) on the Y-side, the dummy subscript (Y = R, L, T, and B) denotes the right (R), left (L), top (T), and bottom (B) side of the cell (i,j) respectively. The same notation applies to all subsequent formulae.

The time-step ($\delta t$) is chosen to meet the condition

$$| \overset{\text{\tiny MM}}{U} | \; \delta t \; / \; \mathrm{Min}(\delta z^{n}_{i,j} \; , \; \delta r^{n}_{i,j}) \leq .4 \quad .$$

It was introduced for accuracy in the original FLIC method. The fluid particles in a cell (i,j) will not be allowed to pass by its adjacent neighbors to farther cells during a computing cycle. It usually covers the Courant-Friedrichs-Lewy condition

$$\delta t \; (| \overset{\text{\tiny MM}}{U} | + C') \; / \; \mathrm{Min} \; (\delta z^{n}_{i,j} \; , \; \delta r^{n}_{i,j}) \leq 1/\sqrt{2}$$

which is required for numerical stability. The notation C' represents the local magnetosonic speed.

Also,

$$\begin{aligned}
\tilde{I}^{n}_{i,j} = I^{n}_{i,j} - \delta t \; \{ & A^{r}_{i,j} \; [P^{n}_{i,j} \; (S^{r}_{j+1/2} \; \bar{v}^{n}_{i,j+1/2} \\
& - S^{r}_{j-1/2} \; \bar{v}^{n}_{i,j-1/2}) + .5 \; q^{n}_{i,j+1/2} \; (S^{r}_{j+1} \; \bar{v}^{n}_{i,j+1} \\
& + S^{r}_{j} \; \bar{v}^{n}_{i,j}) - .5 \; q^{n}_{i,j-1/2} \; (S^{r}_{j-1} \; \bar{v}^{n}_{i,j-1} \\
& + S^{r}_{j} \; \bar{v}^{n}_{i,j}) - \bar{v}^{n}_{i,j} \; S^{r}_{j} \; (q^{n}_{i,j+1/2} - q^{n}_{i,j-1/2})] \\
& + S^{z}_{j} \; A^{z}_{i,j} \; [\, \bar{u}^{n}_{i+1/2,j} \; (P^{n}_{i,j} + q^{n}_{i+1/2,j}) \\
& - \bar{u}^{n}_{i-1/2,j} \; (P^{n}_{i,j} + q^{n}_{i-1/2,j})
\end{aligned}$$

$$- \bar{u}^n_{i,j} \, (q^n_{i+1/2,j} - q^n_{i-1/2,j}) \, ] \, \} \, / \, (\rho^n_{i,j} \, f_{i,j} \, V_j) \, ,$$

$$\tilde{E}^n_{i,j} = \tilde{I}^n_{i,j} + (\tilde{u}^2 + \tilde{v}^2)^n_{i,j} \, / \, 2.$$

Here,

$$\bar{u}^n_{i,j} = (u^n_{i,j} + \tilde{u}^n_{i,j}) \, / \, 2,$$

$$\bar{v}^n_{i,j} = (v^n_{i,j} + \tilde{v}^n_{i,j}) \, / \, 2, \text{ etc.,}$$

$$\bar{u}^n_{i+1/2,j} = (u^n_{i,j} + \tilde{u}^n_{i,j} + u^n_R + \tilde{u}^n_R) \, / \, 4,$$

$$\bar{u}^n_{i-1/2,j} = (u^n_{i,j} + \tilde{u}^n_{i,j} + u^n_L + \tilde{u}^n_L) \, / \, 4,$$

$$\bar{v}^n_{i,j+1/2} = (v^n_{i,j} + \tilde{v}_{i,j} + v^n_T + \tilde{v}^n_T) \, / \, 4,$$

$$\bar{v}^n_{i,j-1/2} = (v^n_{i,j} + \tilde{v}^n_{i,j} + v^n_B + \tilde{v}^n_B) \, / \, 4,$$

The q terms that are artificial viscous pressures are of the forms.

$$q^n_{i+1/2,j} = b \, C^n_{i+1/2,j} \, \rho^n_{i+1/2,j} \, (u^n_{i,j} - u^n_R)$$

if $K \, (u^2 + v^2)^n_{i+1/2,j} < (C^2)^n_{i+1/2,j}$ and if $u^n_{i,j} > u^n_R$ ,

$= 0$ otherwise,

$$q^n_{i,j+1/2} = b \, C^n_{i,j+1/2} \, \rho^n_{i,j+1/2} \, (v^n_{i,j} - v^n_T)$$

if $K \, (u^2 + v^2)^n_{i,j+1/2} < (C^2)^n_{i,j+1/2}$ and if $v^n_{i,j} > v^n_T$ ,

$= 0$ otherwise,

$$q^n_{i-1/2,j} = b\, C^n_{i-1/2,j}\ \rho^n_{i-1/2,j}\ (u^n_L - u^n_{i,j})$$

$$\text{if } K\,(u^2+v^2)^n_{i-1/2,j} < (C^2)^n_{i-1/2,j} \text{ and if } u^n_L > u^n_{i,j},$$

$$= 0 \text{ otherwise,}$$

$$q^n_{i,j-1/2} = b\, C^n_{i,j-1/2}\ \rho^n_{i,j-1/2}\ (v^n_B - v^n_{i,j})$$

$$\text{if } K\,(u^2+v^2)^n_{i,j-1/2} < (C^2)^n_{i,j-1/2} \text{ and if } v^n_B > v^n_{i,j},$$

$$= 0 \text{ otherwise.}$$

The constants K and b are used to determine the magnitude of the artificial viscous pressure, e.g. K=1, $b \leq 0.5$. The constant (b) for cells on a wall boundary is set to be smaller than that for cells in an interior fluid region to avoid numerical viscous heating on a wall surface, say $b_{wall} \simeq (1/3)b_{fluid}$. The quantity (C) is chosen to be the local sound speed in an axisymmetric case or the local magnetosonic speed in a plane case.

In a coaxial shock tube, the initial bias magnetic field is inversely proportional to the radius (r). Notice that the fluid in the tube will not flow radially under the influence of this field alone, because it is curl-free. The value (C) in the viscous pressure term must be the local sound speed in this case. Otherwise, if one chooses the value (C) to be the local magnetosonic speed in the shock tube, the viscous pressure terms will vary with respect to the radius (r). Thus, they will result in incorrect solutions to the problems. Meanwhile, the discrete bias magnetic pressure terms ( $\propto 1/r^2$ ) acting on the inner and outer surfaces of a toroidal cell can induce errors on the radial component of the fluid velocity. To minimize such errors, one can either use fine cells in the radial direction or set Cb to a value approximately equal to 1, say

$$[\ 1 - (3/4)\times(\delta r/r)^2 + (1/4)\times(\delta r/r)^4\ ]\ [\ 1 - (\delta r/r)^2\ ]^{-2},$$

which is obtained by setting the radial acceleration of the fluid to zero in the finite difference equation with a constant initial thermal pressure and a radially varying bias magnetic pressure ( $\propto 1/r^2_j$ ) in the shock tube.

Let

$$X = (\rho,\ C,\ u,\ \tilde{u},\ v,\ \tilde{v}\ ),$$

then

$$X^n_{i\pm 1/2,j} = (\ X^n_{i,j} + X^n_{\substack{R\\L}}\ )\ /\ 2\ ,$$

$$X^n_{i,j\pm 1/2} = (\ X^n_{i,j} + X^n_{\substack{T\\B}}\ )\ /\ 2\ ,$$

$$X^n_{\substack{R\\L}} = \left( \sum_{s=-1}^{+1} X^n_{i\pm 1,j+s}\ R^n_{i\pm 1,j+s} \right) /\ \delta r^n_{i,j}\ ,$$

$$X^n_{\substack{T\\B}} = \left( \sum_{s=-1}^{+1} X^n_{i+s,j\pm 1}\ Z^n_{i+s,j\pm 1} \right) /\ \delta z^n_{i,j}\ .$$

Where $X^n_Y$ is the averaged X-quantity over the surrounding neighbors on the Y-side of the cell (i,j) by the area-weighting method, the subscript Y denotes the direction of the cell surface.

Step 2. The final values due to mass transport effects are calculated.

Let $\Delta MR^n_{i\pm1,j+s}$ be the mass flowing across the area $(R^n_{i\pm1,j+s})$ between the cell $(i,j)$ and the cell $(i\pm1,j+s)$ during the time increment $\delta t$, and $\Delta MZ^n_{i+s,j\pm1}$ be the mass flowing across the area $(Z^n_{i+s,j\pm1})$ between the cell $(i,j)$ and the cell $(i+s,j\pm1)$ during the time increment $\delta t$, (with s=-1, 0, +1).

$$\Delta MR^n_{i\pm1,j+s} = R^n_{i\pm1,j+s}\; S^z_j\; \rho^n_{i,j}\left(\tilde{u}^n_{i,j} + \tilde{u}^n_{i\pm1,j+s}\right)\delta t/(2\delta r),$$

$$\text{if } \left(\tilde{u}^n_{i,j} + \tilde{u}^n_{i\pm1,j+s}\right) \begin{matrix} > 0 \\ < 0 \end{matrix}\;,$$

$$= R^n_{i\pm1,j+s}\; S^z_{j+s}\; \rho^n_{i\pm1,j+s}\left(\tilde{u}^n_{i,j} + \tilde{u}^n_{i\pm1,j+s}\right)\delta t/(2\delta r),$$

$$\text{if } \left(\tilde{u}^n_{i,j} + \tilde{u}^n_{i\pm1,j+s}\right) \begin{matrix} < 0 \\ > 0 \end{matrix}\;,$$

$$s = -1,\ 0,\ +1$$

$$\Delta MZ^n_{i+s,j\pm1} = Z^n_{i+s,j\pm1}\; S^r_{j\pm1/2}\; \rho^n_{i,j}\left(\tilde{v}^n_{i,j} + \tilde{v}^n_{i+s,j\pm1}\right)\delta t/(2\delta z),$$

$$\text{if } \left(\tilde{v}^n_{i,j} + \tilde{v}^n_{i+s,j\pm1}\right) \begin{matrix} > 0 \\ < 0 \end{matrix}\;,$$

$$= Z^n_{i+s,j\pm1}\; S^r_{j\pm1/2}\; \rho^n_{i+s,j\pm1}\left(\tilde{v}^n_{i,j} + \tilde{v}^n_{i+s,j\pm1}\right)\delta t/(2\delta z),$$

$$\text{if } \left(\tilde{v}^n_{i,j} + \tilde{v}^n_{i+s,j\pm1}\right) \begin{matrix} < 0 \\ > 0 \end{matrix}\;,$$

$$s = -1,\ 0,\ +1$$

$$\rho^{n+1}_{i,j} = \rho^{n}_{i,j} + \Big( \sum_{s=-1}^{+1} \Delta MR^{n}_{i-1,j+s} - \sum_{s=-1}^{+1} \Delta MR^{n}_{i+1,j+s}$$

$$+ \sum_{s=-1}^{+1} \Delta MZ^{n}_{i+s,j-1} - \sum_{s=-1}^{+1} \Delta MZ^{n}_{i+s,j+1} \Big) / \left( V_j f_{i,j} \right) .$$

Let $H = B/r_j^m$ , the differencing equations for H are the same as those for the density $\rho$.

Final values for the velocity components (u,v) and for the total energy per unit mass (E) are calculated by conservation of total momentum and total energy in each cell.

$$F^{n+1}_{i,j} = \Big( \sum_{s=-1}^{+1} TR_{i-1,j+s} \;\tilde{\tilde{F}}^{n}_{i-1,j+s} \;\Delta MR^{n}_{i-1,j+s}$$

$$+ \sum_{s=-1}^{+1} TZ_{i+s,j-1} \;\tilde{\tilde{F}}^{n}_{i+s,j-1} \;\Delta MZ^{n}_{i+s,j-1}$$

$$- \sum_{s=-1}^{+1} TR_{i+1,j+s} \;\tilde{\tilde{F}}^{n}_{i+1,j+s} \;\Delta MR^{n}_{i+1,j+s}$$

$$- \sum_{s=-1}^{+1} TZ_{i+s,j+1} \;\tilde{\tilde{F}}^{n}_{i+s,j+1} \;\Delta MZ^{n}_{i+s,j+1}$$

$$+ \tilde{\tilde{F}}^{n}_{i,j} \Big( \rho^{n}_{i,j} \; f_{i,j} \; V_j$$

$$+ \sum_{s=-1}^{+1} \left( 1 - TR_{i-1,j+s} \right) \Delta MR^{n}_{i-1,j+s}$$

$$+ \sum_{s=-1}^{+1} \left( 1 - TZ_{i+s,j-1} \right) \Delta MZ^{n}_{i+s,j-1}$$

$$- \sum_{s=-1}^{+1} \left( 1 - TR_{i+1,j+s} \right) \Delta MR^{n}_{i+1,j+s}$$

$$- \sum_{s=-1}^{+1} \left( 1 - TZ_{i+s,j+1} \right) \Delta MZ^{n}_{i+s,j+1} \Big) \Big) /$$

$$\left( \rho^{n+1}_{i,j} \, f_{i,j} \, V_j \right).$$

Here,

$F = ( u, v, E)$,

$TR_{i\underline{+}1,j+s}$ = 1 if fluid flows into a cell (i,j) from a cell ($i\underline{+}1,j+s$), with s=-1, 0, +1;

= 0 otherwise.

$TZ_{i+s,j\underline{+}1}$ = 1 if fluid flows into a cell (i,j) from a cell ($i+s,j\underline{+}1$), with s=-1, 0, +1;

= 0 otherwise.

The specific internal energy I, gas pressure P, sound speed C, magnetic field B, and total pressure P* are obtained by the following relations:

$$I^{n+1}_{i,j} = E^{n+1}_{i,j} - (u^2 + v^2)^{n+1}_{i,j} / 2 ,$$

$$P^{n+1}_{i,j} = (\gamma - 1) \, \rho^{n+1}_{i,j} \, I^{n+1}_{i,j} ,$$

$$C^{n+1}_{i,j} = (\gamma \, R \, I^{n+1}_{i,j} / Cv)^{\frac{1}{2}} ,$$

$$B^{n+1}_{i,j} = H^{n+1}_{i,j} \, (r^{m}_{j}) ,$$

$$P*^{n+1}_{i,j} = P^{n+1}_{i,j} + (B^2)^{n+1}_{i,j} / 8\pi,$$

where the constants R and Cv are the gas constant and specific heat at constant volume respectively.

The most important feature of the present code is that it allows cells on boundaries to be different shapes and to be in contact with an arbitrary number of neighbor cells. The variable cells in the present modification can also be applied to other methods, say, the PIC method. It is also possible to include physical transport coefficients in the method by adding them directly to the finite difference equations.

# III. BOUNDARY CONDITIONS

## 3.1 Introduction

The finite difference equations developed in Chapter II can be applied to cells on a boundary where the boundary values have been specified. There are four types of cell boundaries: inflow, outflow, free boundary, and solid wall. In all cases, boundary conditions are specified by giving the values of the flow variables in fictitious cells outside the boundaries.

On an inflow boundary, these values are those of the fluid which flows into the boundary. On an outflow boundary, these values are equal to those of the adjacent inner cells. On a free surface, the applied pressure is specified in a vacuum. On a solid wall, the normal component of the fluid velocity is equal to that of the wall surface. There are no flows of mass or energy across a wall boundary or across a non-leaky free surface. Mathematically, the number of boundary conditions is equal to the number of the characteristic lines or cones on a boundary pointing into the fluid region. Numerically, the number of boundary conditions required is equal to the number of the difference equations, and is often more than the correct number of mathematical conditions. Care must be used so that these additional conditions have as little influence as possible. In this chapter, we describe the boundary conditions for the finite difference equations when they are applied to cells on a free boundary or on a curved wall.

In Fig. 2-1, the left curve represents a free boundary which separates the vacuum and fluid regions; the right curve represents a wall boundary which is fixed in space. The cells (A) and (C) are the surface cells on the wall and on the free boundary respectively. The cell (E) is empty in the vacuum, and the cell (F) is fictitious outside the wall boundary. The cells (M) and (N) are regular cells in the interior region. When the finite difference equations are applied to the surface cells on the boundaries, special treatment is required, which we detail below.

3.2 On a Vacuum-plasma Interface

As the free surface moves into a normal cell, that cell becomes a surface cell with a changing volume. Its cell edge in contact with the vacuum will coincide with the free surface. The free surface moves with the same velocity as the mass in the surface cell, and is a Lagrangian contour moving in the Eulerian mesh. This Lagrangian contour gives a good resolution of the movable vacuum-plasma interface if the contour does not become seiously distorted. Apparently, the surface cells on the free boundary will change their shapes, volumes, and widths when the Lagrangian contour moves into or out of them.

The finite difference equations for the center cell (i,j) in Fig. 2-2 need the following modifications when the cell (i,j) is a surface cell on the free boundary. The total pressure P* of an empty neighbor cell of the cell (i,j) is the applied magnetic pressure in the vacuum. Other flow variables in empty neighbor cells are set equal to those of the adjacent surface cell (i,j). We give a simple example as follows:

i) One of the surrounding neighbors of the cell(i,j) is empty, say the cell (i-1,j) in Fig. 2-2, i.e. $\rho^n_{i-1,j} = 0$.

Then,

$$P^{*n}_L = (\underline{P_v{}^{*n}_{i-1,j}} R^n_{i-1,j} + P^{*n}_{i-1,j+1} R^n_{i-1,j+1} + P^{*n}_{i-1,j-1} R^n_{i-1,j-1})$$
$$/\delta r^n_{i,j} \quad ,$$

$$X^n_L = (\underline{X}^n_{i,j} R^n_{i-1,j} + X^n_{i-1,j+1} R^n_{i-1,j+1} + X^n_{i-1,j-1} R^n_{i-1,j-1})$$
$$/\delta r^n_{i,j} \quad ,$$

with $X = (\rho , I , u , \tilde{u} , v , \tilde{v})$ ,

$\Delta MR^n_{i-1,j} = 0$ , etc. ,

where $P_v{}^{*n}_{i-1,j}$ is the applied magnetic pressure in the empty cell (i-1,j) at the time ($n\delta t$), and the underlined terms indicate the modification made in the usual scheme.

The values of the flow variables assigned to the empty cells do not correspond to real physical properties. It means that the flow variables on the free boundary are those of the local surface cells. For a non-leaky free surface, there are no flows of mass and magnetic flux across a cell boundary between a surface cell and an empty cell. In the computations, because of the local Lagrangian treatment, the plasma and vacuum regions do not mix. An empty cell in a vacuum is denoted by a zero-density; a cell in the plasma region is denoted by a positive density. The Lagrangian contour is invented for such problems with a vacuum-plasma interface which is treated as non-diffusive.

The contour on the free boundary is represented by a number of vertical and horizontal segments in z and r directions respectively; it may become zigzag in shape as shown in Fig. 3-1.

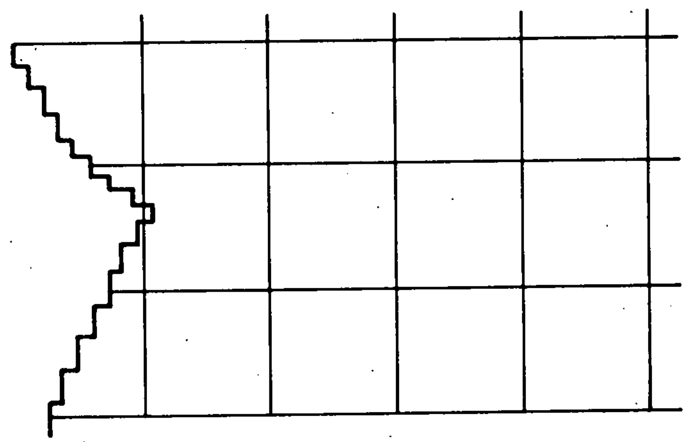


**Fig. 3-1 The vacuum-plasma interface in a zigzag.**

First, let the vertical segments move with the z-component of the fluid velocity (u) in each surface cell in the axial direction; second, let the horizontal segments move with the r-component of the fluid velocity (v) in each surface cell in the radial direction. At the end of each cycle, the new values of the cell volume ($V^{n+1}_{i,j}$), and widths ($\delta z^{n+1}_{i,j}$ and $\delta r^{n+1}_{i,j}$) of a surface cell $(i,j)$ on the free surface are reevaluated after the movements of the cell boundaries on the Lagrangian contour. The procedures are as follows:

1) Calculate the new width $(\delta z^{n+1}_{i,j})$ of a surface cell $(i,j)$ due to the movements of the vertical segments along the axial direction.

   i) Set $\delta z^{n+1}_{i,j} = \delta z^{n}_{i,j}$ unconditionally;

   ii) set $\delta z^{n+1}_{i,j} = \delta z^{n+1}_{i,j} \mp u^{n+1}_{i,j}\ \delta t$ ,

   if the left (-sign) or right (+sign) boundary of the surface cell $(i,j)$ is entirely in contact with the vacuum, or if the left or right boundary is in contact only partially with the vacuum, but the width $(\delta z^{n+1}_{i,j})$ has been changed after the movements of the vertical segments.

2) Calculate the new volume $(V^{n+1}_{i,j})$ of the surface cell $(i,j)$ due to the axial movements of the vertical segments.

   i) Set $V^{n+1}_{i,j} = V^{n}_{i,j}$ unconditionally;

   ii) set $V^{n+1}_{i,j} = V^{n+1}_{i,j} \begin{matrix} - \\ + \end{matrix} \left.\begin{matrix} \Delta r_L \\ \Delta r_R \end{matrix}\right\} u^{n+1}_{i,j}\ (2\pi r_j)\ \delta t$ ,

   where $\Delta r_L$ and $\Delta r_R$ are the total lengths of the vertical segments on the left ( - sign) and right (+ sign) boundaries of the surface cell $(i,j)$ in contact with the vacuum respectively. They are computed in advance from the previous geometrical configuration of the free boundary.

3) Calculate the final volume ($V^{n+1}_{i,j}$) of the surface cell (i,j) due to the radial movements of the horizontal segments.

i) Set $V^{n+1}_{i,j} = V^{n+1}_{i,j} \mp \frac{\Delta z_B}{\Delta z_T} \Big\} v^{n+1}_{i,j} \ (2\pi r_j) \ \delta t$ ,

where $\Delta z_B$ and $\Delta z_T$ are the total lengths of the horizontal segments on the bottom (- sign) and top (+ sign) boundaries of the cell (i,j) in contact with the vacuum respectively. They are calculated from the geometrical configuration of the free boundary after the movements of the vertical segments.

4) Calculate the new width ($\delta r^{n+1}_{i,j}$) of the surface cell (i,j) due to the movements of the horizontal segments along the radial direction.

i) Set $\delta r^{n+1}_{i,j} = \delta r^{n}_{i,j}$ unconditionally;

ii) set $\delta r^{n+1}_{i,j} = \delta r^{n+1}_{i,j} \mp v^{n+1}_{i,j} \ \delta t$ ,

if the bottom (- sign) or top (+ sign) boundary of the surface cell (i,j) is entirely in contact with the vacuum after the movements of the vertical segments in all surface cells, or if the bottom or top boundary is in contact only partially with the vacuum after the movements of the vertical segments in all surface cells, but the width ($\delta r^{n+1}_{i,j}$) has been changed after the movements of the horizontal segments.

When the new value of the volume of the surface cell (i,j) is computed, the density ($\rho^{n+1}_{i,j}$) and magnetic field ($B^{n+1}_{i,j}$) in that cell have to be reevaluated according to the continuity equations in Lagrangian form.

$$\rho^{n+1}_{i,j} = \rho^{n+1}_{i,j} \, V^{n}_{i,j} \; / \; V^{n+1}_{i,j} \quad ,$$

$$B^{n+1}_{i,j} = B^{n+1}_{i,j} \, V^{n}_{i,j} \; / \; V^{n+1}_{i,j} \quad .$$

When the free surface moves into a surface cell, the volume of that cell is decreasing. To get rid of a tiny surface cell (i,j) on the Lagrangian contour, we combine it with an adjacent inner cell when its volume is smaller than a half of the regular volume, i.e. $V^{n+1}_{i,j} < V_j/2$. When the free surface moves out of a surface cell, the volume of that cell is increasing. In order to keep the accuracy, the big surface cell will be divided into two cells when its volume ($V^{n+1}_{i,j}$) is larger than $1.4\ V_j$.

A combination of two cells and a division of the large cell will be called "cell-mixing". The combination of two cells, and the division of a cell into two cells follow the conservation laws of total mass, momentum, and energy. The cell-mixing process is also an effective way to produce an artificial viscosity on the free surface, which helps numerical stability; it naturally will smooth the fluid properties in the mixed region. The internal energies of the mixed cells increase at the expanse of the kinetic energies. On the vacuum-plasma interface, there are no viscous pressure terms between a surface cell and an empty cell because the differences of the velocity components between them are zero. The cell-mixing process is taken to produce the same effect of the artificial viscous pressure. Since it may produce extra viscous heating, it is used on the free surface to stabilize the flow only. One can also control the use of the "cell-mixing" in the computations so that there will be no extreme viscous heating on the free boundary.

The procedures of the cell-mixing are described as follows:

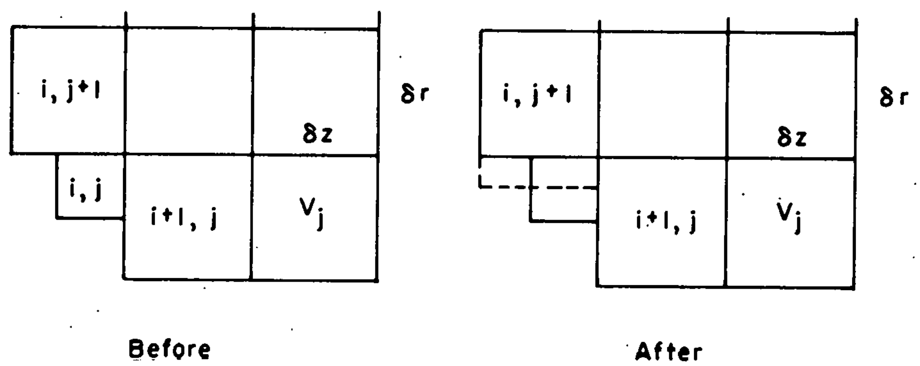



Fig. 3-2. The sample cells for cell-mixing

1) Combine the cell (i,j) with the cell (i,j+1) if $V^{n+1}_{i,j} < V_j / 2$ and $\delta r^{n+1}_{i,j} < .54\ \delta r$:

$$\underline{M}^{n+1}_{i,j+1} = \rho^{n+1}_{i,j+1} V^{n+1}_{i,j+1} + \rho^{n+1}_{i,j} V^{n+1}_{i,j},$$ (new mass of the combined cell located at (i,j+1))

Similarly,

$$\underline{M}^{n+1}_{i,j+1}\ \underline{u}^{n+1}_{i,j+1} = \rho^{n+1}_{i,j+1} V^{n+1}_{i,j+1} u^{n+1}_{i,j+1} + \rho^{n+1}_{i,j} V^{n+1}_{i,j} u^{n+1}_{i,j},$$

$$\underline{M}^{n+1}_{i,j+1}\ \underline{v}^{n+1}_{i,j+1} = \rho^{n+1}_{i,j+1} V^{n+1}_{i,j+1} v^{n+1}_{i,j+1} + \rho^{n+1}_{i,j} V^{n+1}_{i,j} v^{n+1}_{i,j},$$

$$\underline{M}^{n+1}_{i,j+1}\,\underline{E}^{n+1}_{i,j+1} = \rho^{n+1}_{i,j+1}\,V^{n+1}_{i,j+1}\,E^{n+1}_{i,j+1} + \rho^{n+1}_{i,j}\,V^{n+1}_{i,j}\,E^{n+1}_{i,j} ,$$

$$\underline{H}^{n+1}_{i,j+1} = (H^{n+1}_{i,j+1}\,V^{n+1}_{i,j+1} + H^{n+1}_{i,j}\,V^{n+1}_{i,j}) / (V^{n+1}_{i,j+1} + V^{n+1}_{i,j}) ,$$

$$\underline{\rho}^{n+1}_{i,j+1} = \underline{M}^{n+1}_{i,j+1} / (V^{n+1}_{i,j+1} + V^{n+1}_{i,j}) ,$$

$$\underline{V}^{n+1}_{i,j+1} = V^{n+1}_{i,j+1} + V^{n+1}_{i,j} ,$$

$$\underline{\delta z}^{n+1}_{i,j+1} = \delta z^{n+1}_{i,j+1} ,$$

$$\underline{\delta r}^{n+1}_{i,j+1} = V^{n+1}_{i,j+1} / (2\pi r_{j+1}\,\underline{\delta z}^{n+1}_{i,j+1}) ,$$

where the underline denotes a quantity in the combined cell (i,j+1).

2) Divide the combined cell (i,j+1) into two cells if $\underline{V}^{n+1}_{i,j+1} > 1.4V_{j+1}$ and $\underline{\delta r}^{n+1}_{i,j+1} > 1.5\,\delta r$.

Let $X = (\rho, u, v, E, H)$, then

$$X^{n+1}_{i,j} = \underline{X}^{n+1}_{i,j+1} ,$$

$$X^{n+1}_{i,j+1} = \underline{X}^{n+1}_{i,j+1} ,$$

$$V^{n+1}_{i,j} = \underline{V}^{n+1}_{i,j+1} - 2\pi r_{j+1}\underline{\delta z}^{n+1}_{i,j+1}\,\delta r ,$$

$$\delta r^{n+1}_{i,j} = \delta r^{n+1}_{i,j} ,$$

$$\delta z^{n+1}_{i,j} = V^{n+1}_{i,j} / (2\pi r_j\,\delta r^{n+1}_{i,j}) ,$$

$$V^{n+1}_{i,j+1} = 2\pi r_{j+1} \underline{\delta z}^{n+1}_{i,j+1} \delta r ,$$

$$\delta z^{n+1}_{i,j+1} = \underline{\delta z}^{n+1}_{i,j+1} ,$$

$$\delta r^{n+1}_{i,j+1} = \delta r .$$

The cell (i,j) will be combined with the cell (i+1,j) if $V^{n+1}_{i,j} < V_j / 2$ and $\delta z^{n+1}_{i,j} < .54\ \delta z$, and the combined cell (i+i,j) will be divided into two cells if $\underline{V}^{n+1}_{i+1,j} > 1.4\ V_j$ and $\underline{\delta z}^{n+1}_{i+1,j} > 1.5\delta z$. The formulae used in this case are similar to those in the previous case.

An additional exact condition is still needed to control the cell-mixing process so as not to produce numerical overheating on the free surface, but to insure the stability of the flow. Because the moving boundary changes its configuration during the computational cycles, the computer program for considering all the geometrical situations becomes very complicated. It simplifies the computer programming if the cells are made rectangular after cell-mixing. Any serious distortion of the free boundary will yield inaccurate solutions on the Lagrangian contour. Further improvements are still required for more accurate solutions on the moving vacuum-plasma interface.

### 3.3 On a Curved Rigid Wall

When the geometry of the wall shape is given, the wall boundary can be located in the computational zone. The cells on the wall surface have different shapes with the boundaries of the surface cells following the wall curve. In Fig. 3-3, the cell (S) is a surface cell on the wall boundary, which may include a regular cell and a smaller irregular cell. The cell (F) is a fictitious boundary cell. The procedures for treating the surface cells and creating the fictitious cells on a curved wall boundary are given below:

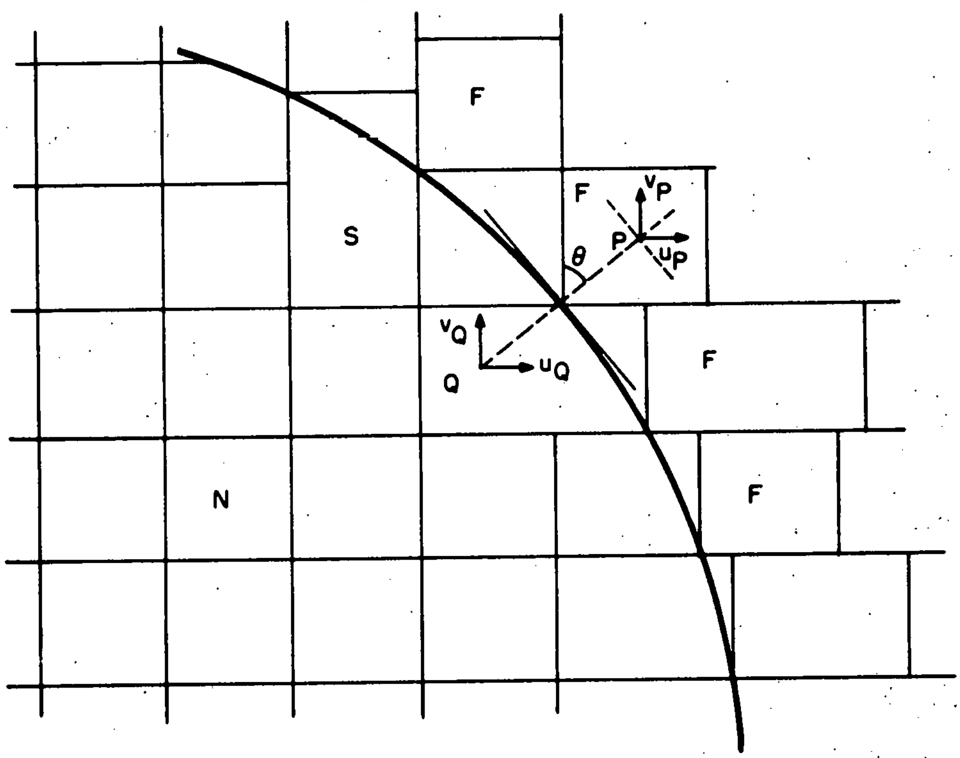



Fig. 3-3 Surface cells and fictitious cells on a curved wall.

1) A surface cell smaller than one half of the regular cell at the same radius is adjoined to an adjacent inner cell, making the resulting cell to be a surface cell.
2) Calculate its volume and widths with the wall boundary to be a part of its boundaries. But the shape of that surface cell is still regarded as a rectangle with a part outside the boundary as shown in Fig. 3-3.
3) Locate a fictitious cell adjacent to that surface cell, and make its shape equal to that of the adjacent surface cell in a rectangle.

The values of the flow variables for the fictitious cell are determined as follows:

1) Find the position of the center (P) of the fictitious cell as shown in Fig. 3-3.
2) Determine the position of its mirror image (Q) by reflection with respect to the tangent of the wall curve.
3) The values of the flow variables at the image (Q) are obtained by the area-weighting method from its surrounding cells.
4) The values of the flow variables at the center (P) of the fictitious cell are obtained by reflection from its image values at the point (Q).
5) The magnetic field at the point (P) is obtained by a special treatment, i.e., $B_P = B_Q\, r_Q \,/\, r_P$ in an axisymmetric case; $B_P = B_Q$ in a plane case.
6) There is no mass transport across the wall boundary. The

mass and magnetic flux flowing across a cell boundary between a surface cell and a fictitious cell must be set to zero.

The reflection procedure for the flow variables of the fictitious cell is as follows:

$$\rho_P = \rho_Q ,$$

$$I_P = I_Q ,$$

$$u_P = u_Q \cos(2\theta) - v_Q \sin(2\theta) ,$$

$$v_P = -u_Q \sin(2\theta) - v_Q \cos(2\theta) ,$$

$$\tilde{u}_P = \tilde{u}_Q \cos(2\theta) - \tilde{v}_Q \sin(2\theta) ,$$

$$\tilde{v}_P = -\tilde{u}_Q \sin(2\theta) - \tilde{v}_Q \cos(2\theta) ,$$

where the angle $\theta$ is shown in Fig. 3-3.

In this way, the flow variables ( $\rho$ , $u$ , $v$ , $I$ ) in the fictitious cell are obtained by reflection with respect to the tangent of the wall profile. The fluid velocity on the wall will be parallel to the wall surface and extended without change into the fictitious cell. The special treatment on the magnetic field is designed so that the total pressure ($P^*$) of the fluid near the wall will be balanced by that of the fictitious cell. In addition, we insure that there will be no mass transport between a surface cell and a fictitious cell.

The above procedures can be applied to both axisymmetric flows and plane flows. Since the cells on the wall surface are irregular in shape, we call the method an "irregular shape" approximation. The reflective boundary conditions suggested in this section make the code powerful enough to solve problems with arbitrary smooth wall boundaries. The accuracy of the code remains high on a curved wall boundary.

# IV. INCIDENT SHOCK WAVES

## 4.1 General Description

In the past, shock waves in coaxial shock tubes have been simplified as a one-dimensional propagation[5,6]. It is obvious that the plasma flow in the coaxial shock tube is two-dimensional because the drive field is a function of the radial position. In this section, we apply the FLIC MHD code to study the formation and shape of the shock wave in the two-dimensional (r-z) plane. In addition, this incident shock wave and the knowledge of its properties are needed before we study shock focusing and reflection from a curved end wall.

The schematic view of the coaxial electromagnetic shock tube is shown in Fig. 4-1. First of all, a bias current is built up in the central rod of the inner conductor to set up an azimuthal bias field. When the capacitor banks are discharged, a current sheet will form in the space between the concentric electrodes to ionize the working gas in the tube. This drive current sheet and magnetic field interact and produce electromagnetic forces acting on the plasma. The magnetic pressure serves as a magnetic piston to push the plasma down the tube. The positive pressure disturbance in the plasma will eventually steepen into a shock front. Since the bias field is azimuthal, a transverse MHD shock wave will be formed when the plasma is compressed by a sufficiently strong applied pressure.

The numerical code with the Lagrangian free boundary is now applied to this problem. When the drive current sheet is formed in the plasma, the drive magnetic field is built up in the vacuum. The corresponding magnetic pressure in the vacuum is then used in the finite difference equations as the boundary value on the free surface. We prescribe the drive current as an arbitrary function of time. We first show how shock waves will be formed by constant and time increasing sinusoidal drive currents. Since there is no physical dissipation in the basic equations, numerical viscous pressure terms are included in the code to permit the computation of shock transition.

For each case of shock formation or reflection studied, the computer results will be shown on SC4060 plottings, in which there are three planes on each page (see Fig. 4-2). In each plane, the upper and lower grid lines are the outer wall and inner conductor of the shock tube respectively. The tick marks on the upper grid line denote the sizes and locations of regular computing cells. The left blank region is the vacuum; the shadow region contains the plasma. The curves in the first plane are the constant temperature contours, their values are indicated by integers listed above the plane. The density is represented by the shading points in the second plane. The darker the region is, the higher the density. The velocity and its direction in each cell is represented by a segment vector in the third plane. The details of the flow and the shock wave in the tube can be clearly seen from these planes on the plottings at different times. In the shock transition, a number of temperature contours are present because there is a large gradient of the temperature.

In addition, we also show the profiles of magnetic field $B(z)$, temperature $T(z)$ and density $\rho(z)$ at several different radii. This is to facilitate comparison with experiments.

## 4.2 Constant Drive Current

In this section, the drive current in the shock tube is assumed to be a step function. The initial conditions and other required information of the tube are given in Table 4-1. The results on shock formation by a constant driver are presented in Fig. 4-2. Because of the radial effect of the drive magnetic field, the applied magnetic pressure in the tube is inversely proportional to the square of the radius (r). The plasma near the inner conductor will be driven by a larger current to move faster than that near the outer wall. Therefore, the shock wave has a slope (in this case, about $120^o$) with respect to the axial axis during the early stages of formation. Also, the drive current sheet on the vacuum boundary becomes inclined automatically, a radially outward force thus acts on the plasma to move it toward the outer wall. As long as the plasma behind the shock wave moves to the outer wall region continuously, compression heating occurs in the plasma in that area.

The total pressure ($P* = P+B^2 / 8\pi$) of the plasma there becomes the highest. The increase of magnetic pressure is due to the frozen-in field moving together with the plasma; the increase of thermal pressure is due to compression heating on the outer wall. This high total pressure has two essential effects on the plasma motion and on the shock wave. First, it strengthens the shock wave near the outer wall so that the shock slope diminishes. Thus, a nearly planar shock wave will always be formed by the radially varying drive magnetic field, provided the tube is long enough. Secondly, it slows down the axial motion of the plasma in the outer region behind the shock wave. The tail part of the plasma near the outer wall increases in length as the motion goes on. When the shock wave approaches a quasi-steady state, it makes an angle of about $110^\circ$ with the axial axis, but the vacuum-plasma interface continuously increases its slope with respect to the axial axis. The inclined shock wave on the outer wall will thus be reflected.

The sequence of the shock motion, flow direction of the plasma, temperature contours, and the density distributions in the shock tube are shown in Fig. 4-2. The shock speed is about 125 cm/ μsec, the Mach number is 867, and the corresponding Alfven Mach number at the mean radius is 4.63. At the mean radius, just behind the shock wave, the density of the plasma is about $1.8 \times 10^{-8}$ gram/cm$^3$, the magnetic field is about 28 kilo-Gauss (KG), the temperature is about 3.8 Kev, and $\beta$ is 0.9.

The magnetic field, density, and temperature are also plotted with respect to axial positions at several different radii in Fig. 4-3. The humps on the curves (d) again show the effect of the compression on the outer wall. The flow conditions behind the shock wave are not uniform radially or axially.

Fig. 4-4 shows the density distribution in the coaxial shock tube which was obtained by B. Marder[28] using the GAP code. There is good qualitative agreement with our results.

Table 4-1

Initial conditions and other required information of the shock tube

| | |
|---|---|
| Working gas | hydrogen |
| Density | $.5344 \times 10^{-8}$ gram/cm$^3$ |
| Pressure | 66.65 dynes/cm$^2$ |
| Temperature | 300 $^{o}$K |
| Axial component of velocity | 0 cm/sec |
| Radial component of velocity | 0 cm/sec |
| Specific internal energy | $1.87 \times 10^{10}$ ergs/gram |
| Specific heat ratio | 5/3 |
| Gas constant | $4.157 \times 10^{7}$ ergs/$^{o}$K |
| Inner radius ($r_i$) | 6.25 cm |
| Outer radius ($r_o$) | 11.25 cm |
| Mean radius ($r_m$) | $(r_i r_o)^{\frac{1}{2}}$ |
| Cell width ($\delta r = \delta z$) | 0.25 cm |
| Tube length | 32 cm |
| Bias magnetic field | $7000\ r_m / r$ Gauss |
| Drive magnetic field | $4.8 \times 7000\ r_m / r$ Gauss |

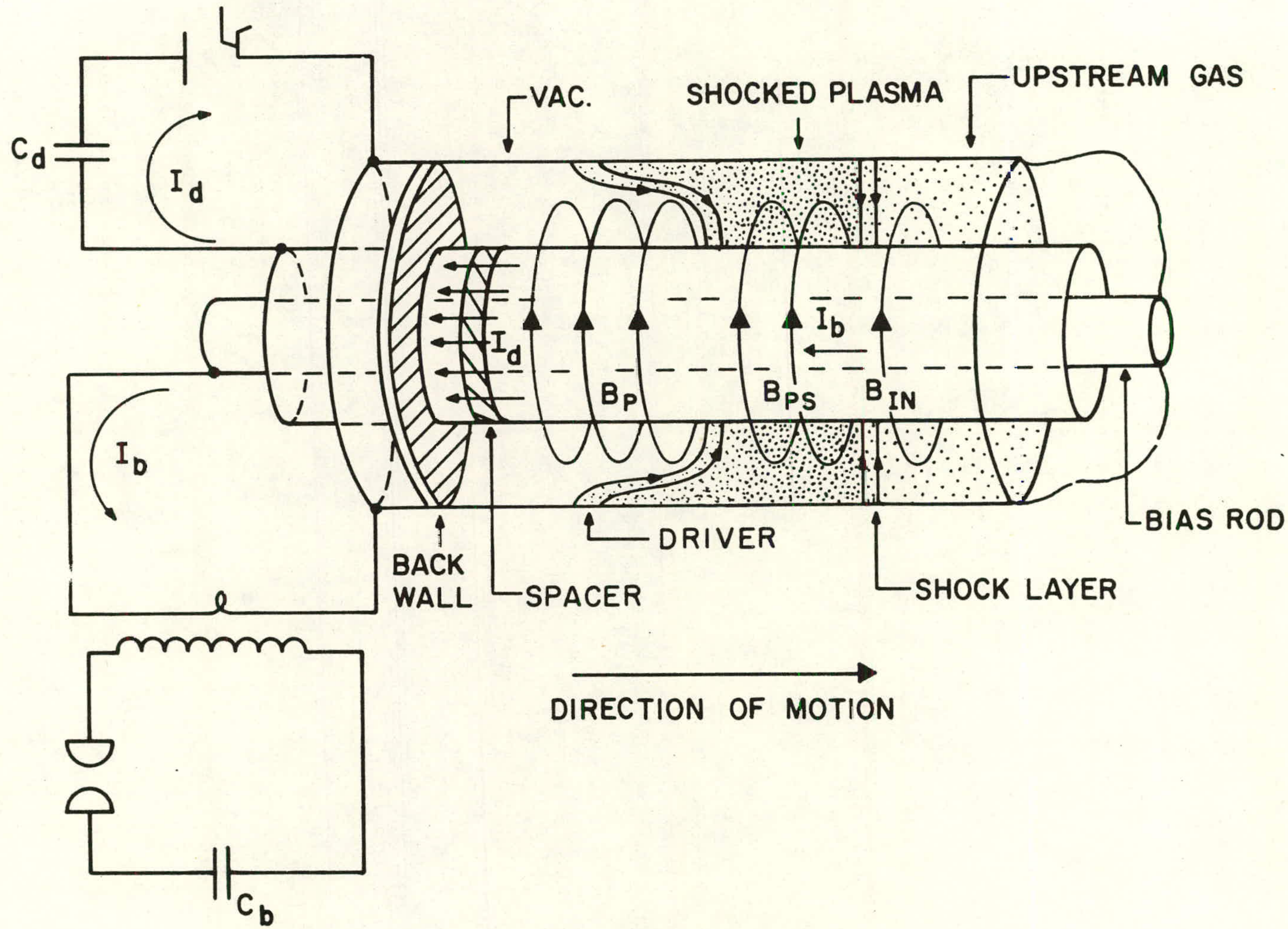



Fig. 4-1 Schematic of Shock Tube System

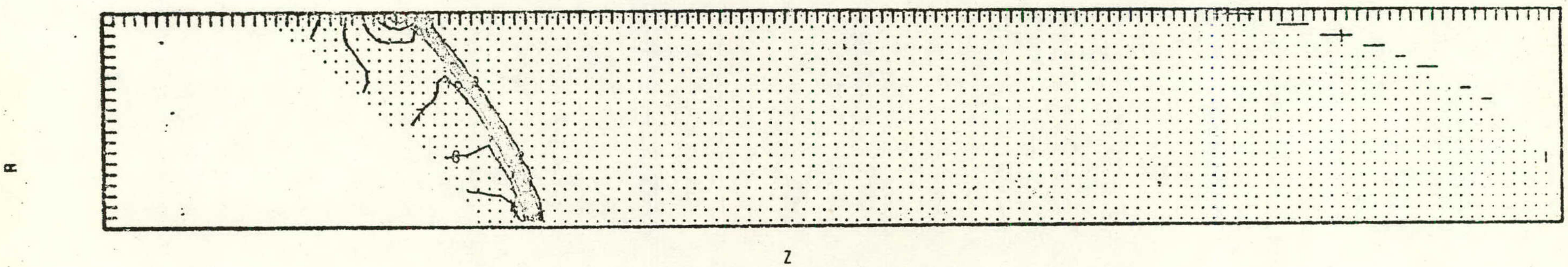
Fig. 4-2

CYCLE= 201 ,TIME= 0.799969E 07

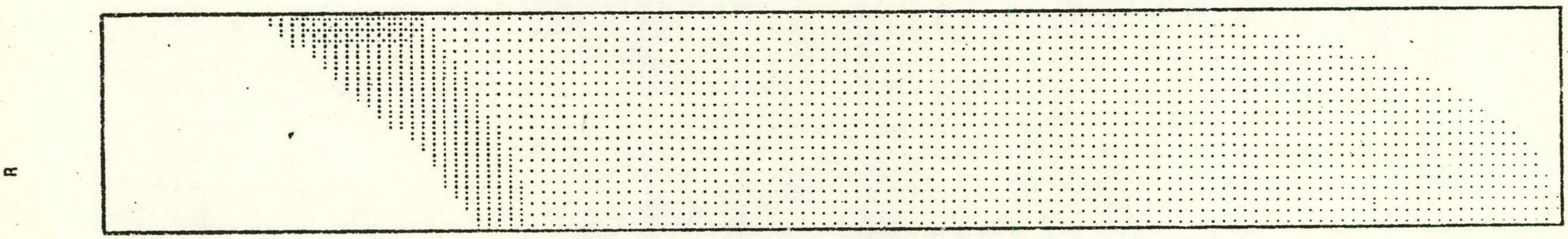
CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.498E+07 | 3 = | 0.996E+07 | 4 = | 0.149E+08 | 5 = | 0.199E+08 |
| 6 = | 0.249E+08 | 7 = | 0.299E+08 | 8 = | 0.349E+08 | 9 = | 0.398E+08 | 10 = | 0.448E+08 |

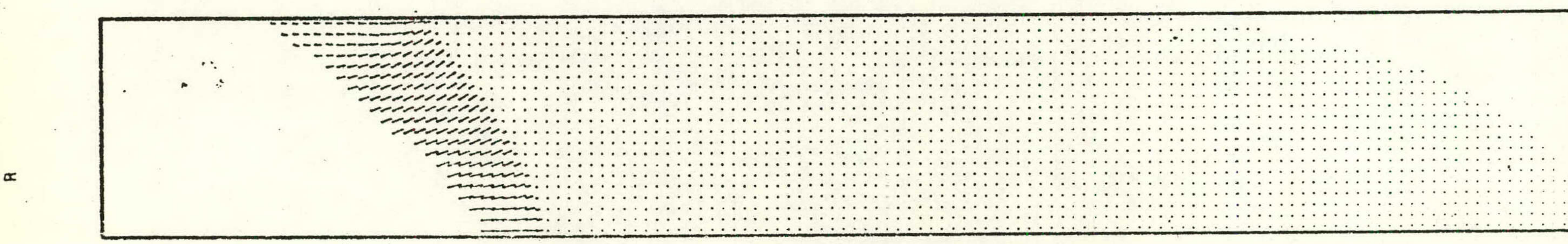
CYCLE= ,TIME= DENSITY SHADING VELO




TEMPERATURE CONTOURS




DENSITY SHADING




VELOCITY

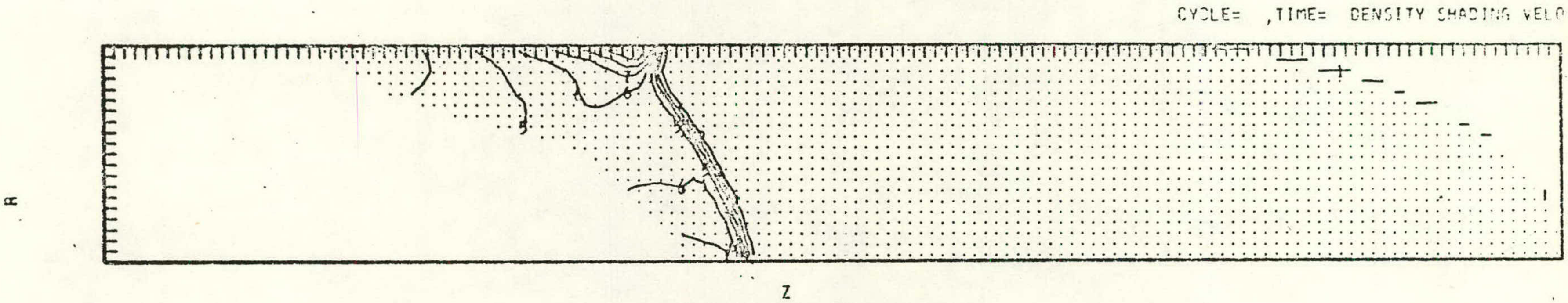
NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 4-2

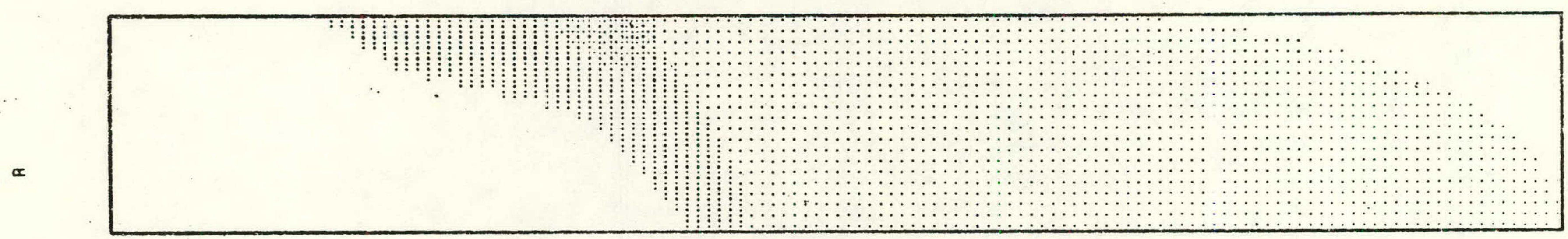
CYCLE= 301 ,TIME= 0.119997 -06

CONTOURS

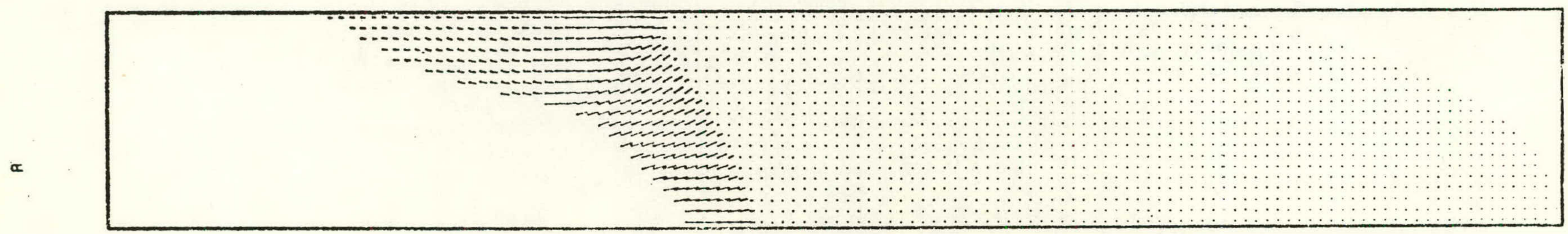
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.715E+07 | 3 = | 0.143E+08 | 4 = | 0.214E+08 | 5 = | 0.286E+08 |
| 6 = | 0.357E+08 | 7 = | 0.429E+08 | 8 = | 0.500E+08 | 9 = | 0.572E+08 | 10 = | 0.643E+08 |




TEMPERATURE CONTOURS




DENSITY SHADING




VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

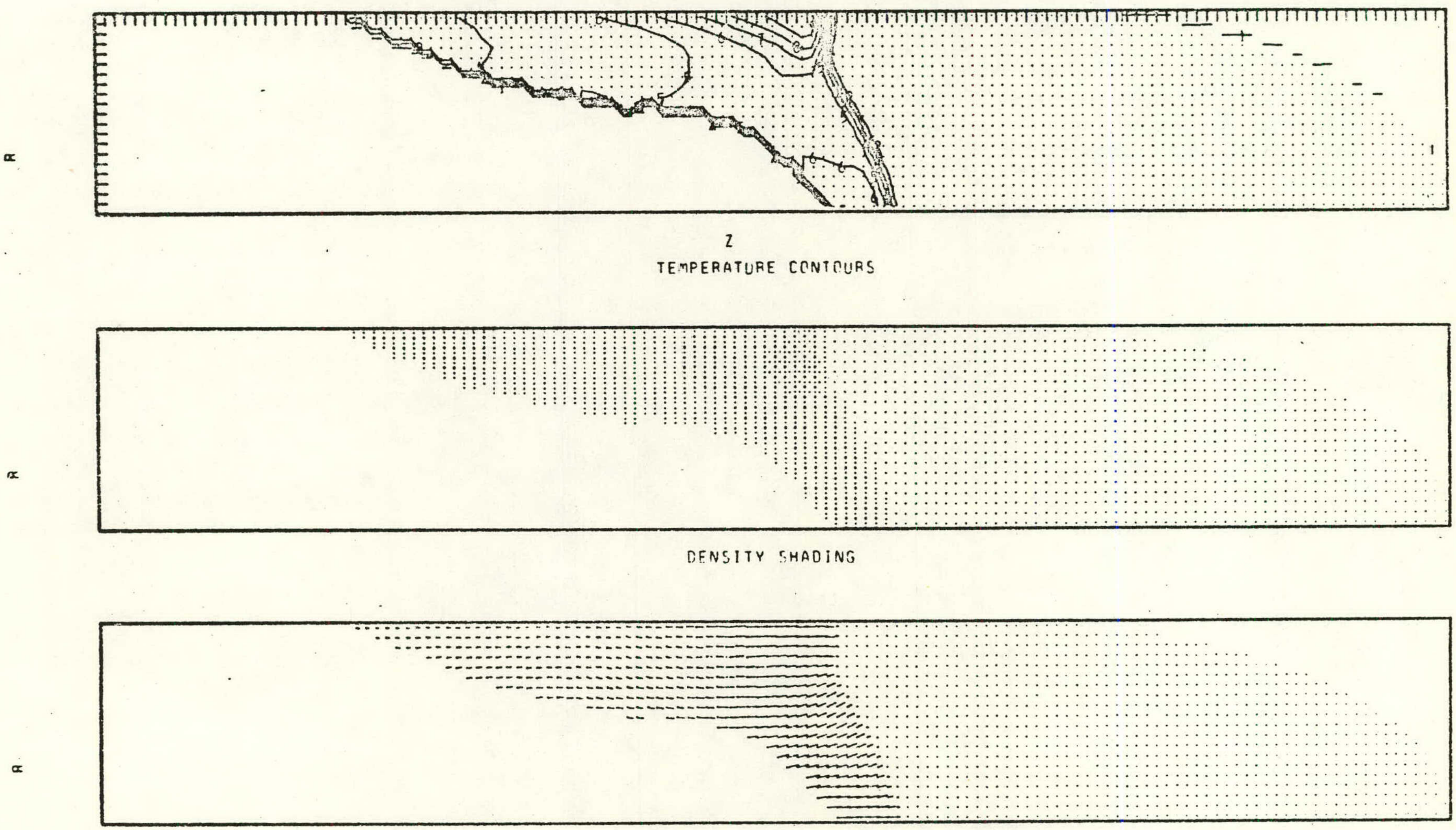
Fig. 4-2

CYCLE= 401 ,TIME= 0.159987E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.781E+07 | 3 = | 0.156E+08 | 4 = | 0.234E+08 | 5 = | 0.312E+08 |
| 6 = | 0.390E+08 | 7 = | 0.468E+08 | 8 = | 0.546E+08 | 9 = | 0.625E+08 | 10 = | 0.703E+08 |

CYCLE=YO,TIME=%CDENSITY SHADING&VELC



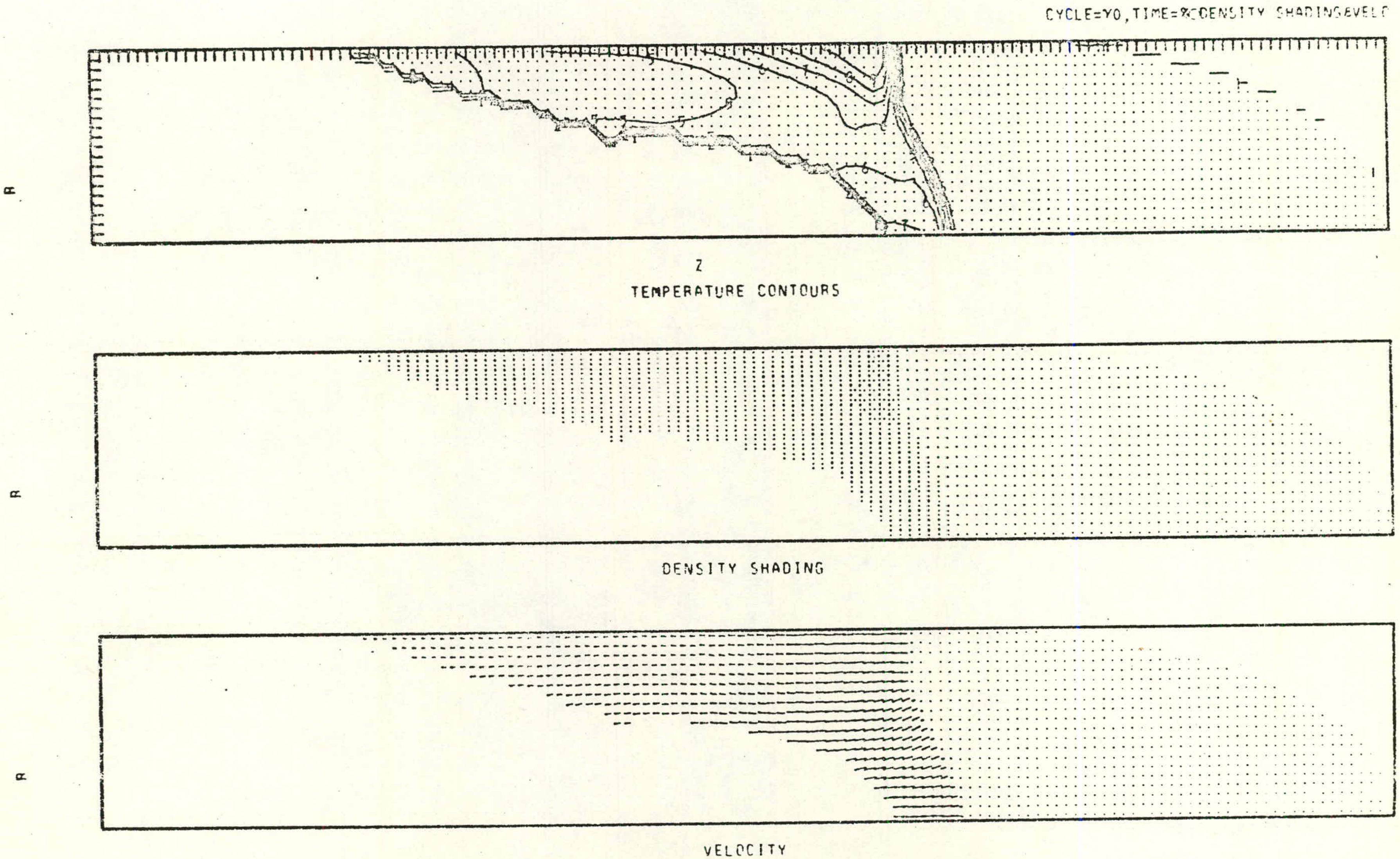

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Page 3

CYCLE= 451 ,TIME= 0.179984E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.762E+07 | 3 = | 0.152E+08 | 4 = | 0.229E+08 | 5 = | 0.305E+08 |
| 6 = | 0.381E+08 | 7 = | 0.457E+08 | 8 = | 0.533E+08 | 9 = | 0.609E+08 | 10 = | 0.686E+08 |




NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 4-2 The shock wave generated by a constant drive current in a coaxial shock tube

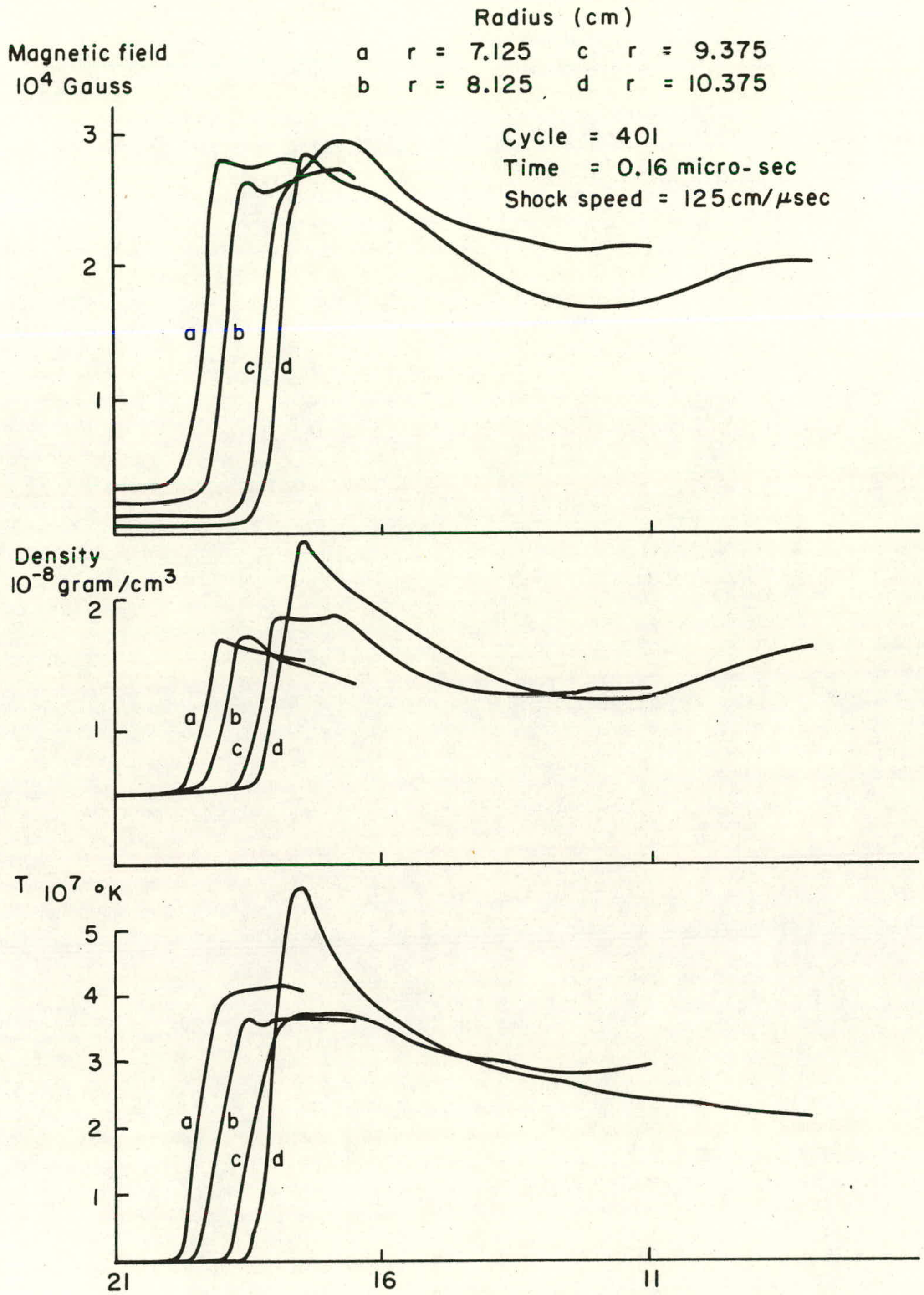



Fig. 4-3 The plots of magnetic field, density and temperature near the shock wave under a constant driver.

H-407. M. 1125 W.L.

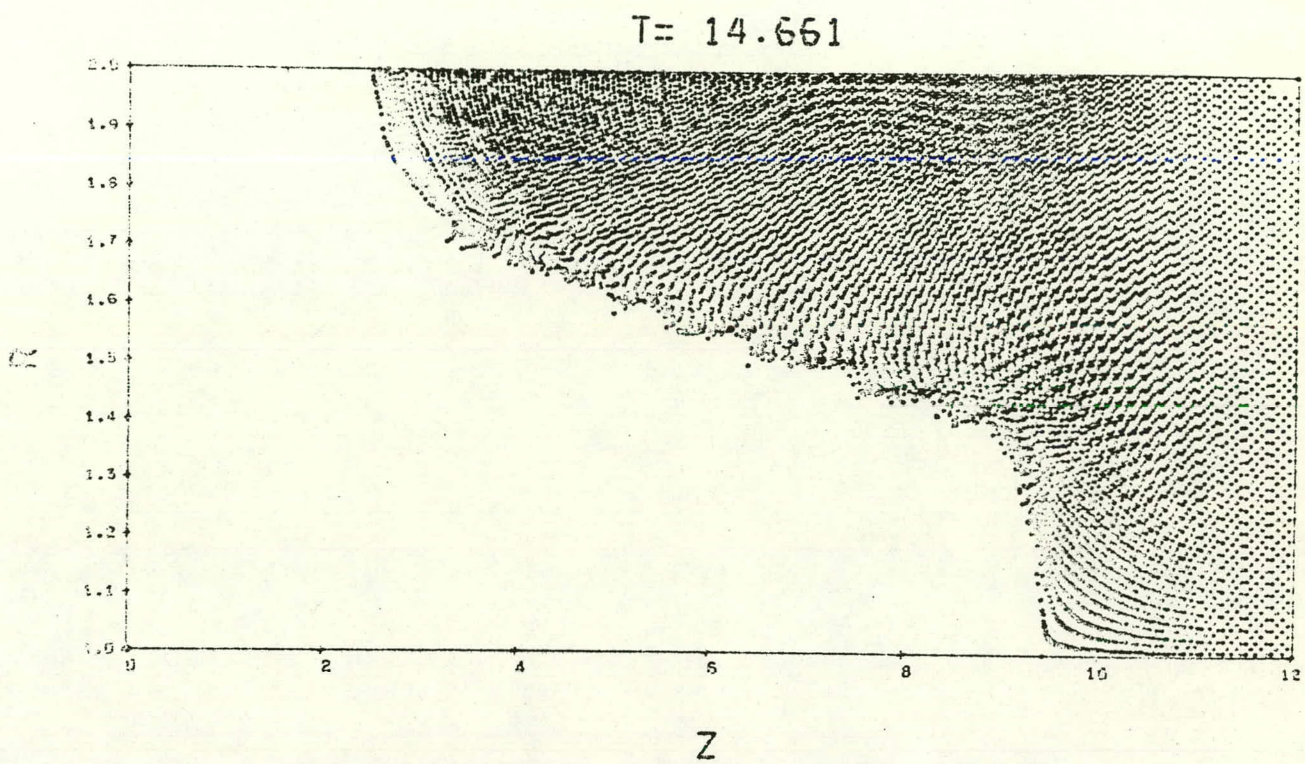



Fig. 4-4 The density distribution in the coaxial shock tube which was obtained by Marder using the GAP code. The drive magnetic field is 4 times the bias field.

## 4.3 Sinusoidal Drive Current

The drive current in the shock tube experiment is a sinusoidal function of time, this is a result of the LC circuit. For more realistic simulation and comparison with the experimental measurements, we therefore use the experimental data in Table 4-2 for the computation. Because the shock tube is 180 cm long, we use coarser cells (0.5 cm) in order to save computing time and core storages. The period of the sine function is 16 micro-seconds, the total time of the calculation is about 4 micro-seconds, at which time the drive current reaches maximum. The shock wave arrives at the end of the shock tube at 3.55 micro-seconds; but if the reflection of the shock wave is considered, the total time duration is 5 micro-seconds in our calculation.

The maximum amplitude (958.5 KA) of the sinusoidal current is smaller than the magnitude of the constant driver (1140 KA) in the previous section. Of course, the drive magnetic pressure induced by this current is smaller than that induced by the constant driver, and it will thus take a longer time to generate a shock wave. The fluid particles, and the shock wave will move more slowly under the influence of the sinusoidal driver than under the constant driver.

When the applied magnetic pressure in the vacuum is not high enough to produce a shock wave in the plasma, a compression wave will propagate down the tube and out toward the outer wall. The shock wave will be formed when the drive magnetic pressure reaches a certain value at a critical moment of time. The mechanism for shock formation under the sinusoidal driver is the same as that under the constant driver. But, the shock will increase in strength with the increase of the drive current in time. The fluid particles will also increase their velocities. These are shown in Table 4-3, in which the post-shock temperatures are measured at a radius of 8.5 cm. The temperature (2.4 Kev) of the deuterium plasma measured at 3.32 micro-seconds with the sinusoidal drive current (924.6 KA) is lower than the temperature (3.8 Kev) of the hydrogen plasma with the constant driver (1140 KA) in the previous section. However, the change of the drive current with respect to time will be small enough when it is near the maximum amplitude. The flow conditions of the shock wave will not be much different from those in the case of a constant driver with the same magnitude.

Fig. 4-5 shows the magnetic field, density and temperature plotted with respect to axial position at three different radii in the tube after 3.32 micro-seconds. At a radius of 8.5 cm, the flow conditions behind the shock wave are almost uniform. The hump in the curve (c) of the magnetic field indicates the compression of the plasma on the outer wall.

Fig. 4-6 shows the time trace of the magnetic field measured at a fixed station of 100 cm with a radius of 10 cm in the tube. It agrees with the upper trace (a) in Fig. 4-7, which was obtained in our shock tube experiment.

Table 4-2

Initial conditions in a coaxial shock tube of 180 cm

| | |
|---|---|
| Working gas | deuterium |
| Density | $1.0688 \times 10^{-8}$ gram/cm$^3$ |
| Temperature | 300 $^{o}$K |
| Pressure | 66.65 dynes/cm$^2$ |
| Bias magnetic field | 60,000/r Gauss |
| Drive magnetic field | $(300 + 958.5 \sin(2\pi t \times 10^6/16)) \times 200/r$ Gauss |
| Inner radius | 6.25 cm |
| Outer radius | 11.25 cm |
| Cell width ($\delta z=\delta r$) | 0.5 cm |

Table 4-3

The relation between shock speed and drive current (deuterium plasma)

| Time (t) | Drive Current ($958.5 \sin (2\pi t 10^6/16)$) | Shock Speed | Post-shock temp. |
|---|---|---|---|
| 1.20 μsec | 435.1 KA | 43 cm/μsec | $0.08 \times 10^{7}$ °K |
| 1.56 | 550.7 | 53 | 0.37 |
| 1.86 | 639.7 | 59.8 | 0.68 |
| 2.14 | 713.2 | 64.0 | 1.00 |
| 2.39 | 774.2 | 67.3 | 1.50 |
| 2.64 | 824.9 | 72.5 | 1.76 |
| 2.87 | 866.3 | 73.5 | 2.00 |
| 3.10 | 899.4 | 77.3 | 2.20 |
| 3.32 | 924.6 | 78.2 | 2.40 |

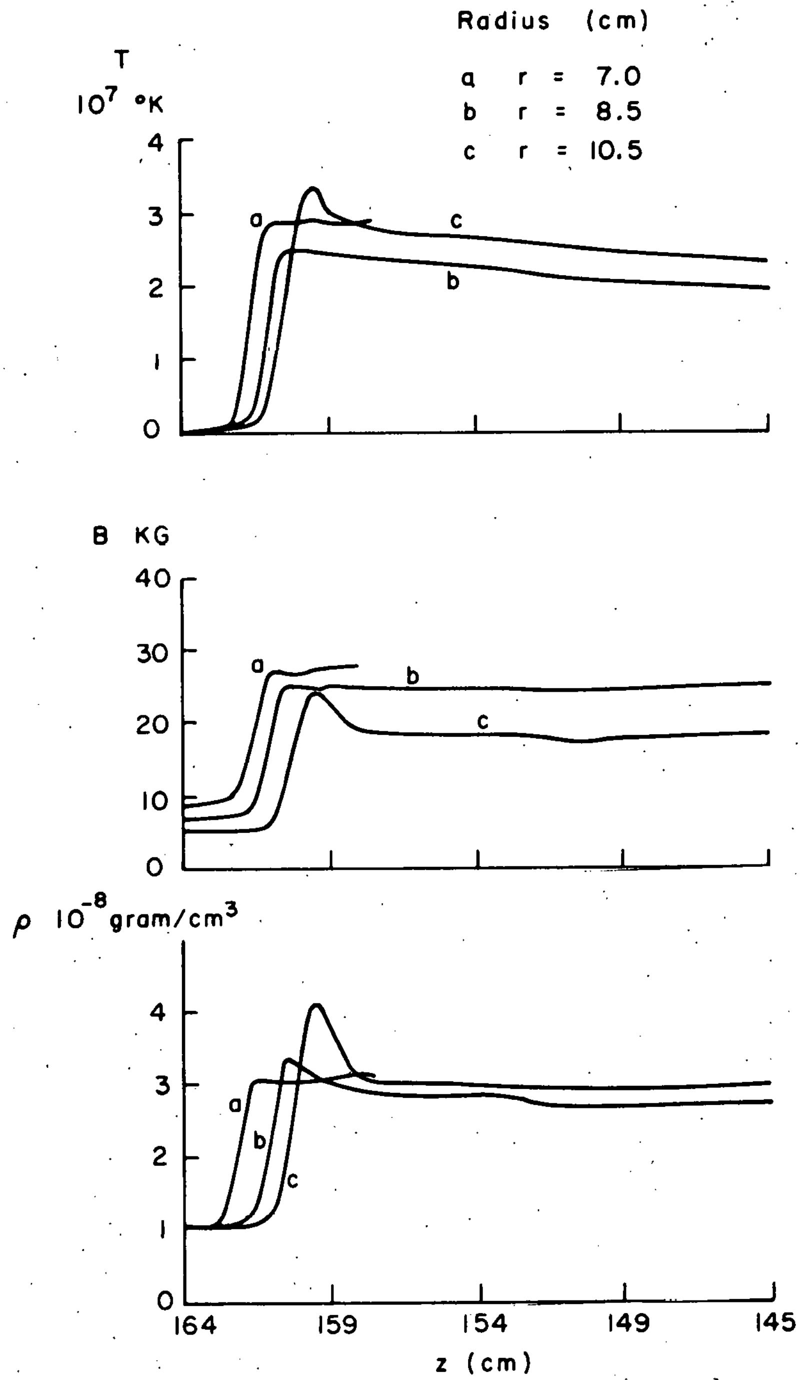



Fig. 4-5 The temperature, magnetic field and density vs. axial positions at different radii.

A·907·ME ^^27. W.L.

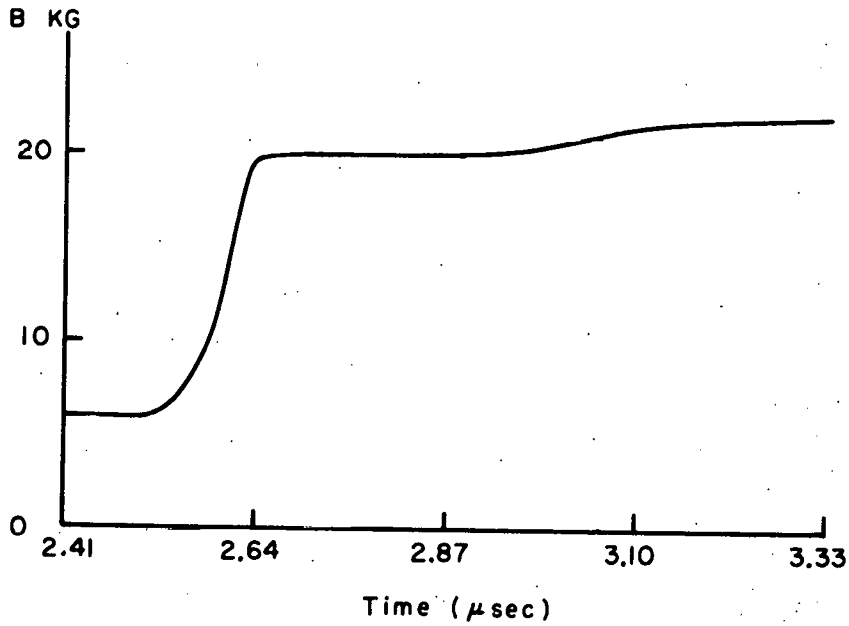



Fig. 4-6 The trace of the magnetic field measured at a fixed station (100 cm) with a radius of 10 cm.

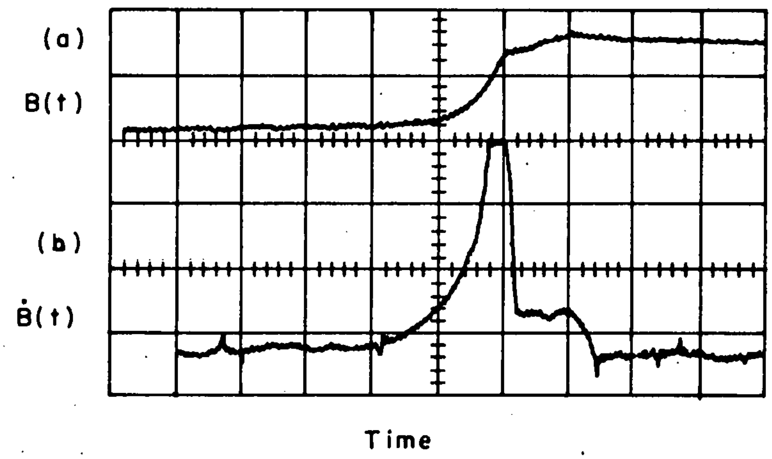



Fig. 4-7 A typical magnetic probe trace from the Columbia University shock tube, taken from a station 100 cm down the shock tube.

# V. SHOCK REFLECTION AND CONVERGENCE

## 5.1 Flat End Wall

### 5.1.1 Constant Driver

To reflect a shock wave in the simplest way, we locate a flat end wall in the shock tube. The shock wave in Section 4.2, with a constant drive current of 1140 KA, is employed to study its reflection from the end wall. The fluid particles behind the shock wave travel in the range of 110 cm/μsec, the incident shock speed is about 125 cm/μsec. Other flow conditions of the incident shock wave are given in Section 4.2.

Fig. 5-1 shows the sequence of the shock wave moving down the tube and reflecting from the flat end wall, which is at the right end. At cycle = 651, the shock wave arrives at the end wall. At cycle = 701, the shock wave just reflects off the wall surface. After shock reflection, the plasma near the outer wall again remains the hottest with the temperature $14.5 \times 10^7$ °K, density $5.1 \times 10^{-8}$ gram/cm$^3$, and magnetic field $5.3 \times 10^4$ Gauss. At cycle = 801, the reflected shock wave travels downstream, and the hot plasma expands into the vacuum near the inner conductor.

Fig. 5-2 shows the traces of the magnetic field, density and temperature with respect to time at a fixed station, which is 5 cm in front of the flat end wall with a radius of 9.375 cm. The first jump in the trace of the magnetic field indicates the arrival of the incident shock wave at the fixed station; the second jump shows the return of the reflected shock wave. The total pressure ($P^* = 2.4 \times 10^8$ dynes/cm$^2$) in the hot plasma after shock reflection is much higher than the applied magnetic pressure ($0.4 \times 10^8$ dynes/cm$^2$ at the mean radius) in the vacuum so that the reflected shock and particles are accelerated to expand into the vacuum quickly. The magnetic field, density and temperature decrease after the reflected shock wave moves away from the fixed station.

The relation between the incoming shock and the reflected shock along a radius of 9.375 cm in the tube is also demonstrated by the plots of the magnetic field, density and temperature with respect to axial position at different times in Fig. 5-3. The flat end wall is located at 32.5 cm. There are small numerical oscillations in the density and magnetic field just behind the reflected shock wave. The reflected shock speed is about 100 cm/μsec, slower than the incident shock speed of 125 cm/μsec.

Fig. 5-1

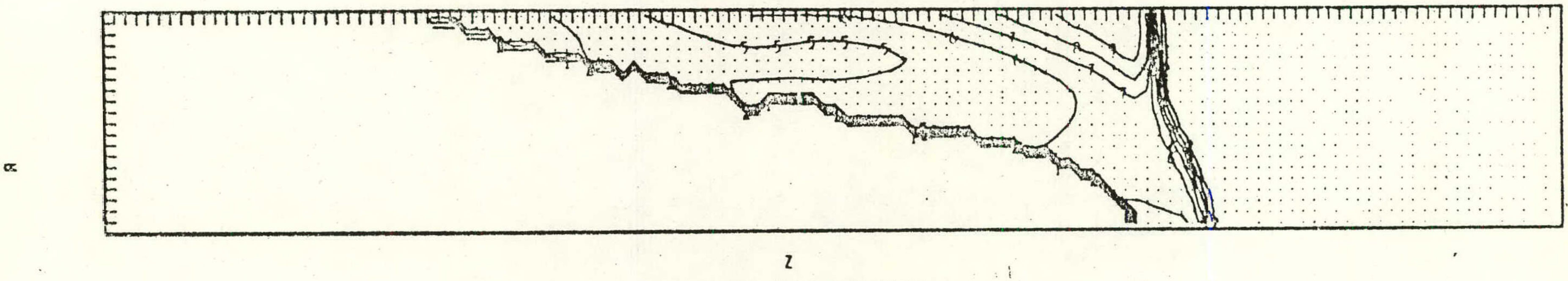


CYCLE= 501 ,TIME= 0.199982E-06

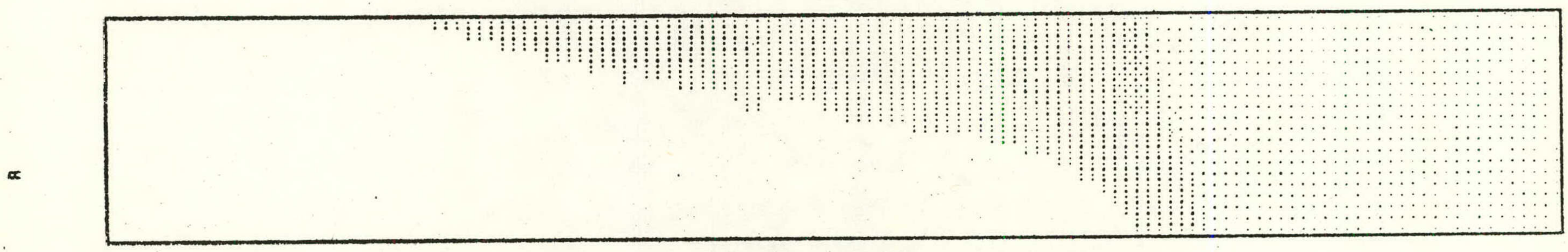
CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.734E+07 | 3 = | 0.147E+08 | 4 = | 0.220E+08 | 5 = | 0.294E+08 |
| 6 = | 0.367E+08 | 7 = | 0.441E+08 | 8 = | 0.514E+08 | 9 = | 0.587E+08 | 10 = | 0.661E+08 |

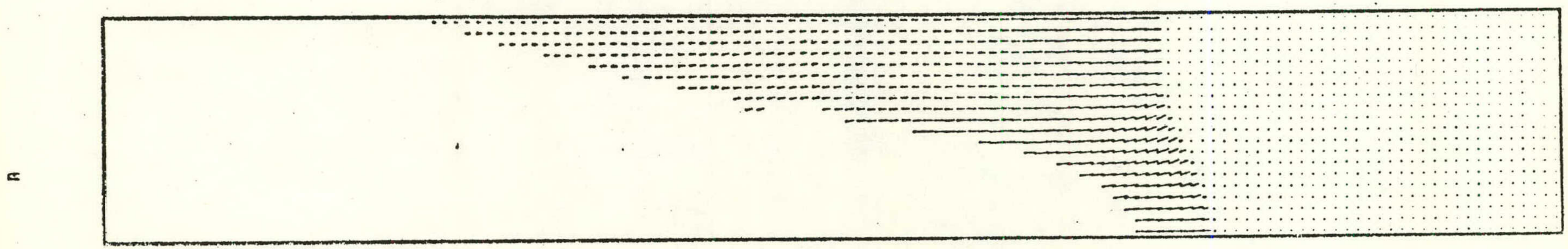
CYCLE=Y ,TIME=C DENSITY SHADING




TEMPERATURE CONTOURS




DENSITY SHADING




VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 5-1

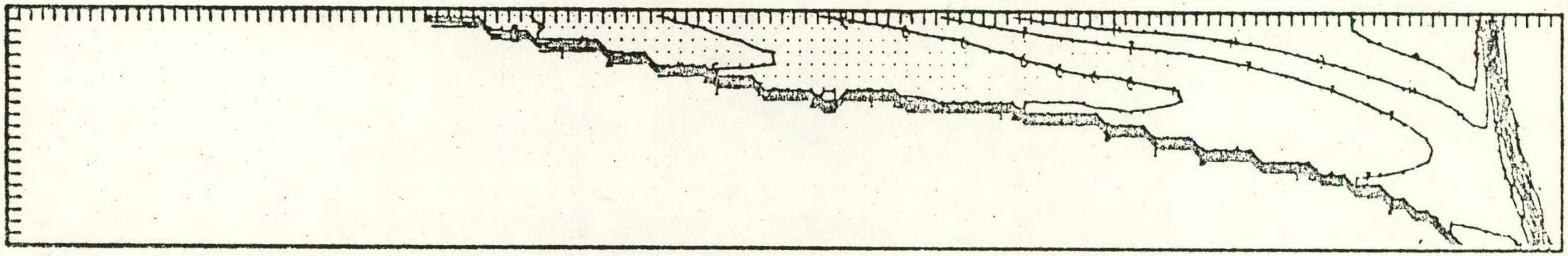
CYCLE= 651 ,TIME= 0.259974E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.652E+07 | 3 = | 0.130E+08 | 4 = | 0.196E+08 | 5 = | 0.261E+08 |
| 6 = | 0.326E+08 | 7 = | 0.391E+08 | 8 = | 0.457E+08 | 9 = | 0.522E+08 | 10 = | 0.587E+08 |

CYCLE= ,TIME= DENSITY SHADING

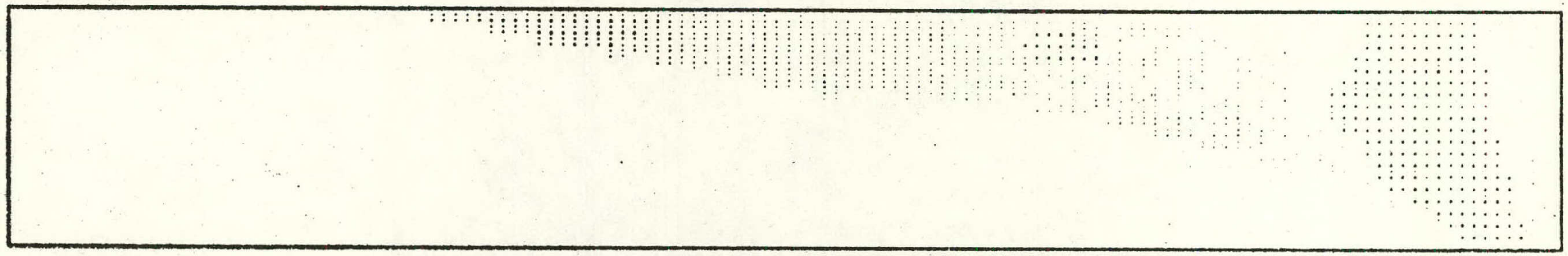
R



Z

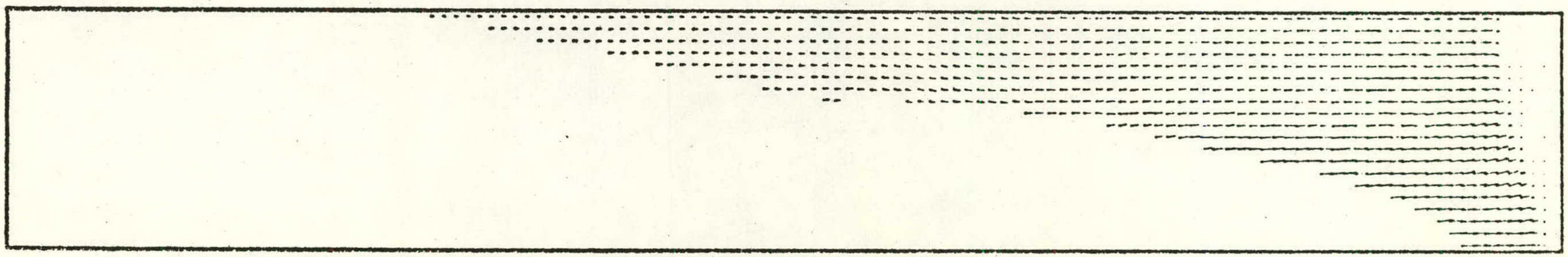
TEMPERATURE CONTOURS

R



DENSITY SHADING

R



VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 5-1

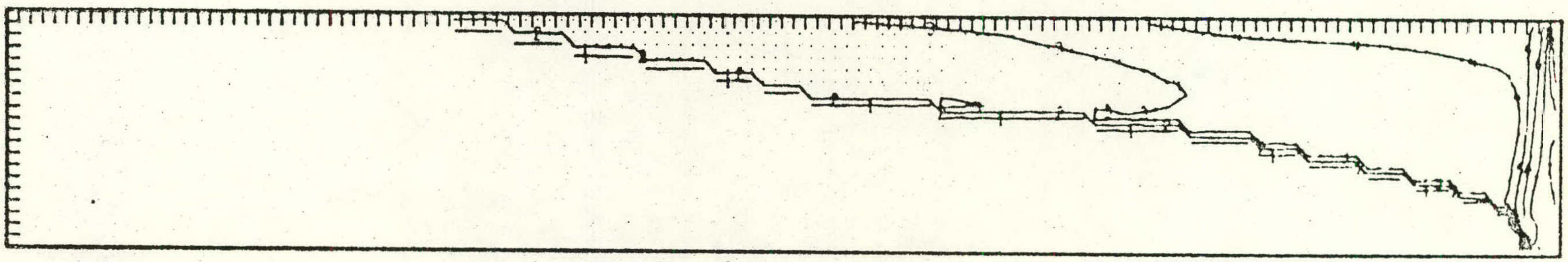
CYCLE= 701 ,TIME= 0.279972E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.164E+08 | 3 = | 0.328E+08 | 4 = | 0.492E+08 | 5 = | 0.656E+08 |
| 6 = | 0.820E+08 | 7 = | 0.984E+08 | 8 = | 0.115E+09 | 9 = | 0.131E+09 | 10 = | 0.148E+09 |

CYCLE=X ,TIME DENSITY SHADING

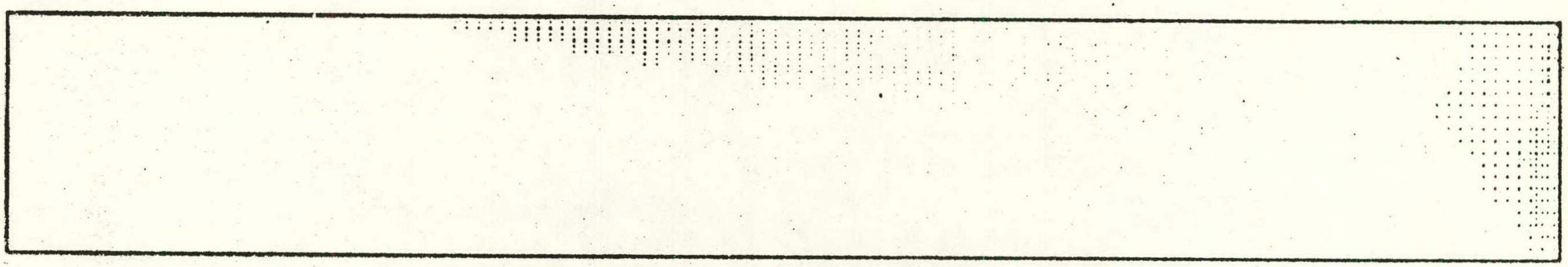
R



Z

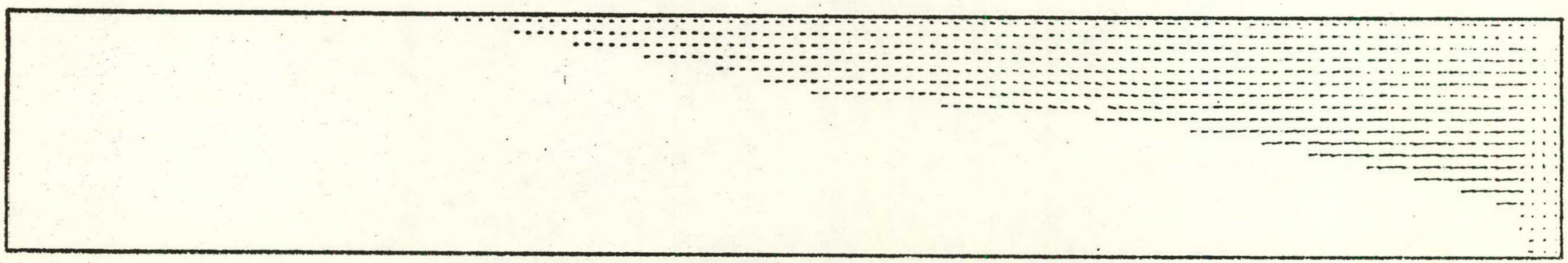
TEMPERATURE CONTOURS

R



DENSITY SHADING

R



VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

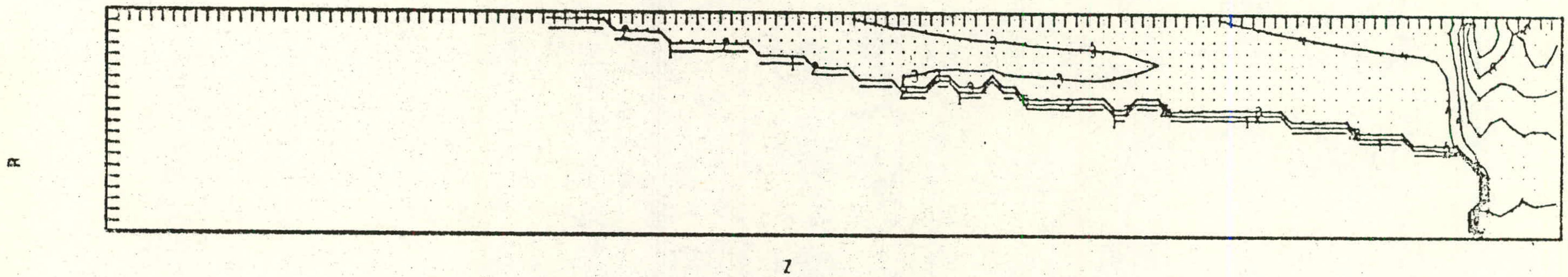
Fig. 5-1

CYCLE= 751 ,TIME= 0.299969E-06

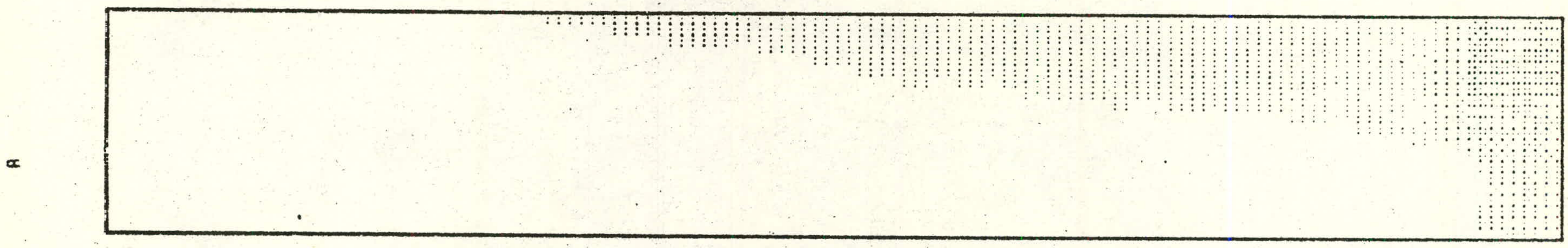
CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.149E+08 | 3 = | 0.297E+08 | 4 = | 0.446E+08 | 5 = | 0.594E+08 |
| 6 = | 0.743E+08 | 7 = | 0.892E+08 | 8 = | 0.104E+09 | 9 = | 0.119E+09 | 10 = | 0.134E+09 |

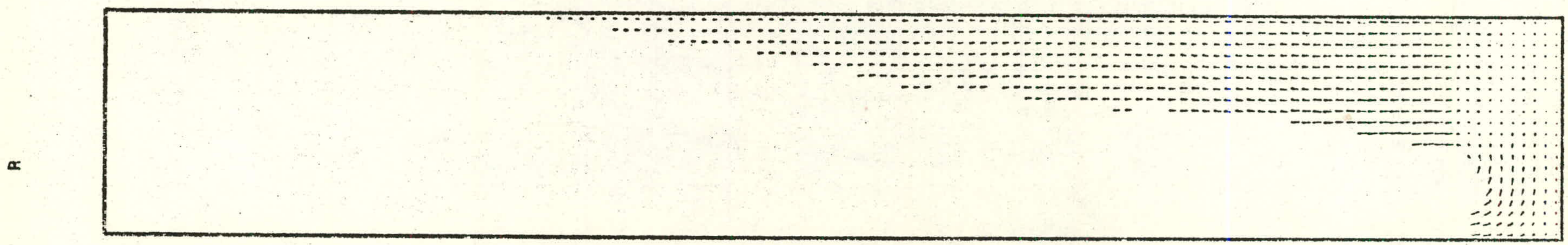
CYCLE=Y ,TIME=C DENSITY SHADING




TEMPERATURE CONTOURS




DENSITY SHADING




VELOCITY

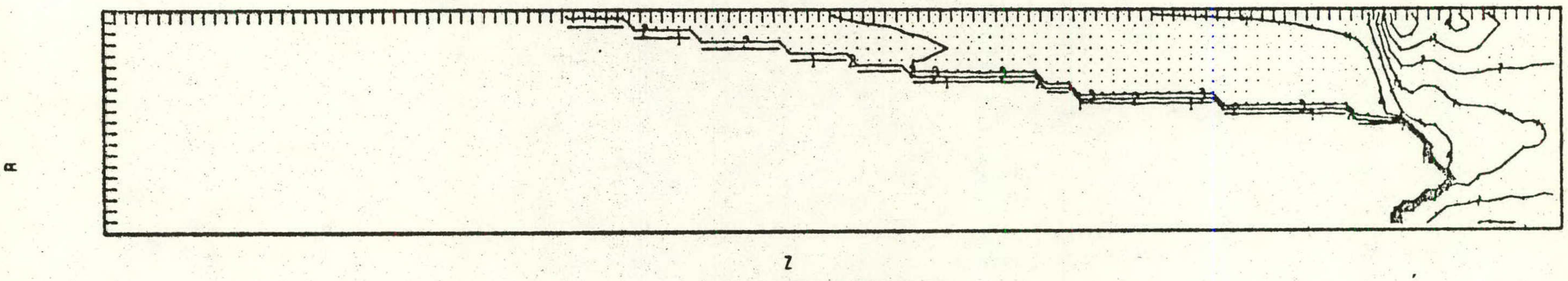
NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 5-1 

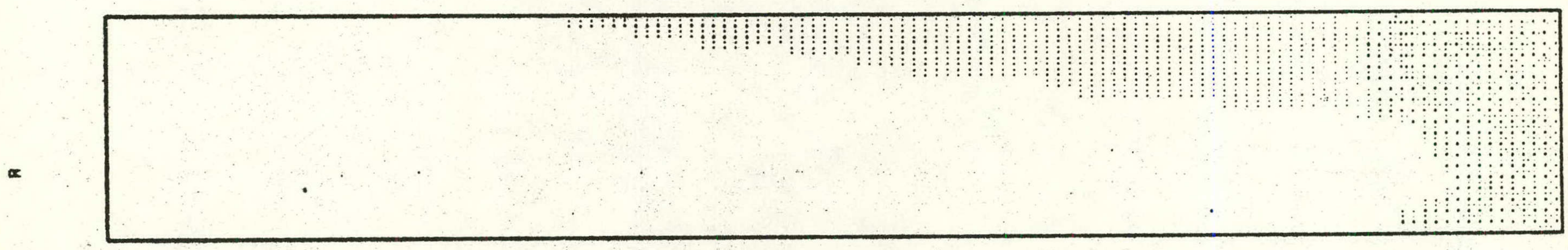
CYCLE= 801 ,TIME= 0.319967E-06
CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.136E+08 | 3 = | 0.271E+08 | 4 = | 0.407E+08 | 5 = | 0.543E+08 |
| 6 = | 0.678E+08 | 7 = | 0.814E+08 | 8 = | 0.950E+08 | 9 = | 0.109E+09 | 10 = | 0.122E+09 |

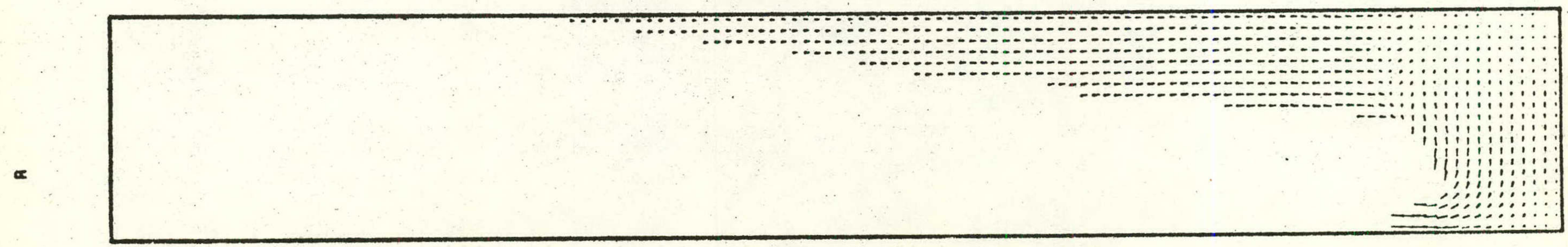
CYCLE=Y ,TIME= DENSITY SHADING




TEMPERATURE CONTOURS




DENSITY SHADING



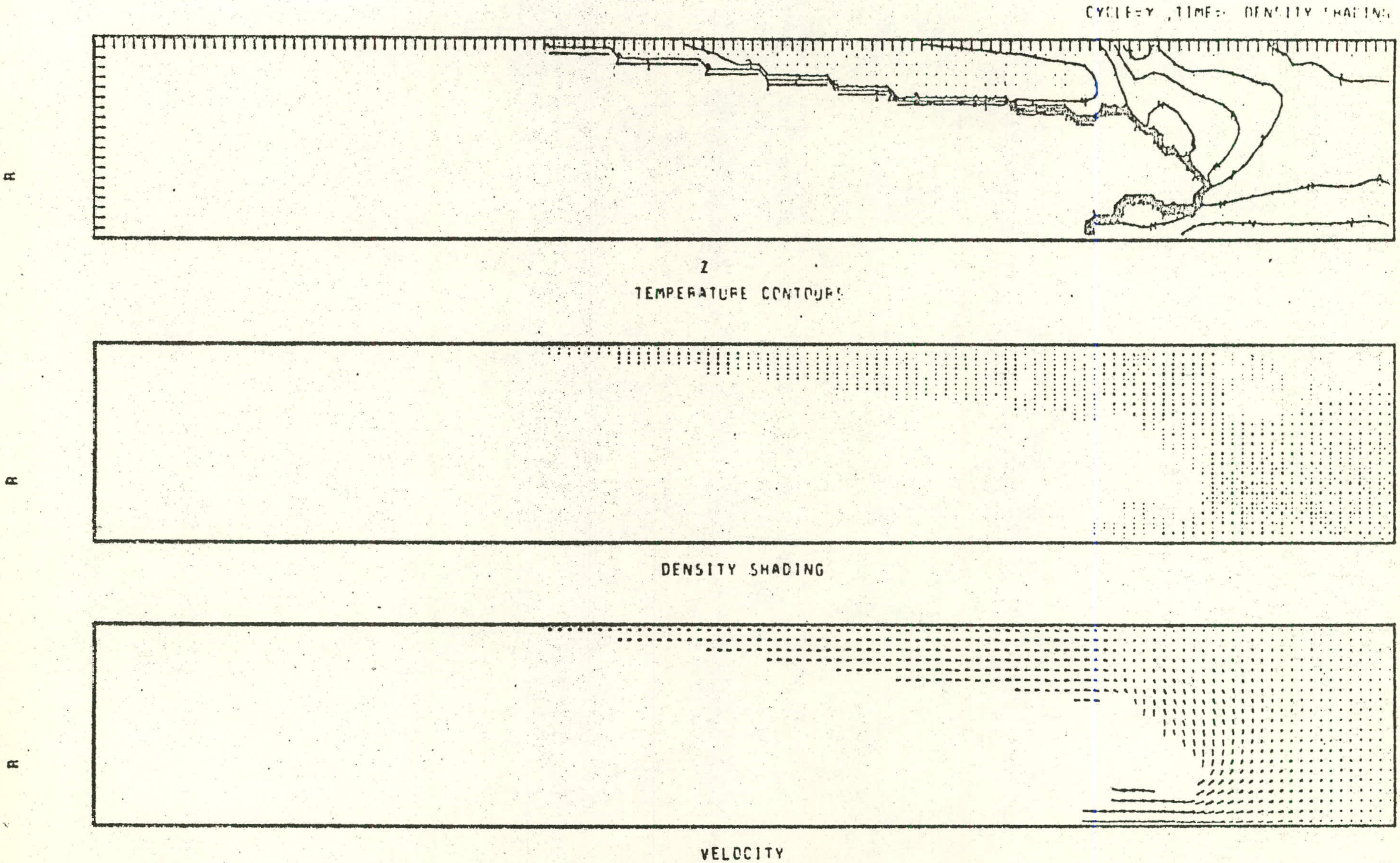

VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

CYCLE= 901 ,TIME= 0.359962E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.107E+08 | 3 = | 0.215E+08 | 4 = | 0.322E+08 | 5 = | 0.429E+08 |
| 6 = | 0.537E+08 | 7 = | 0.644E+08 | 8 = | 0.751E+08 | 9 = | 0.859E+08 | 10 = | 0.966E+08 |




NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 5-1 The incident shock wave and its reflection from a flat end wall (constant driver)

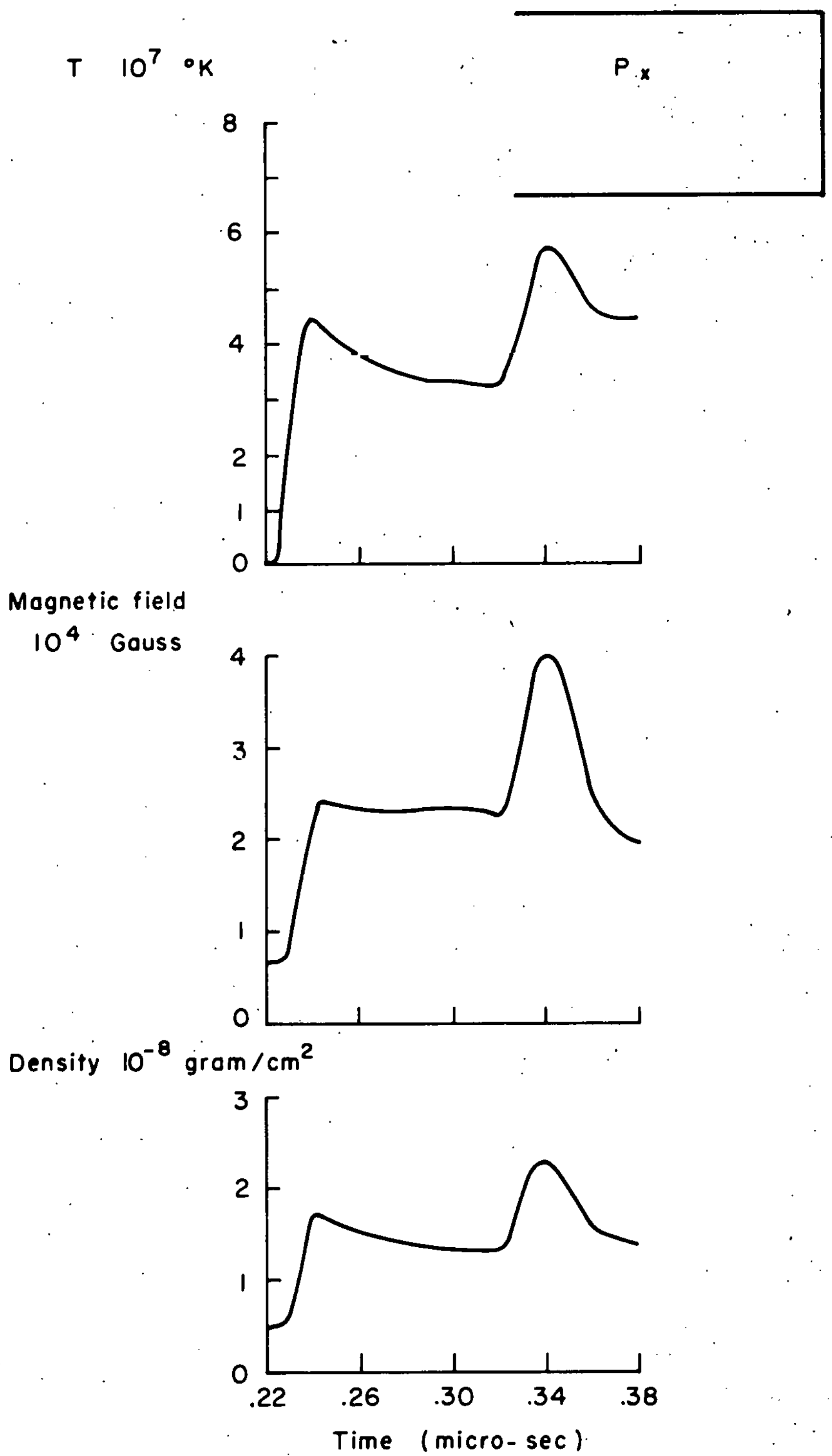



Fig. 5-2 The traces of the temperature, magnetic field and density with respect time at the fixed station (P) located at 5 cm in front of the flat end wall with the radius equal to 9.375 cm.

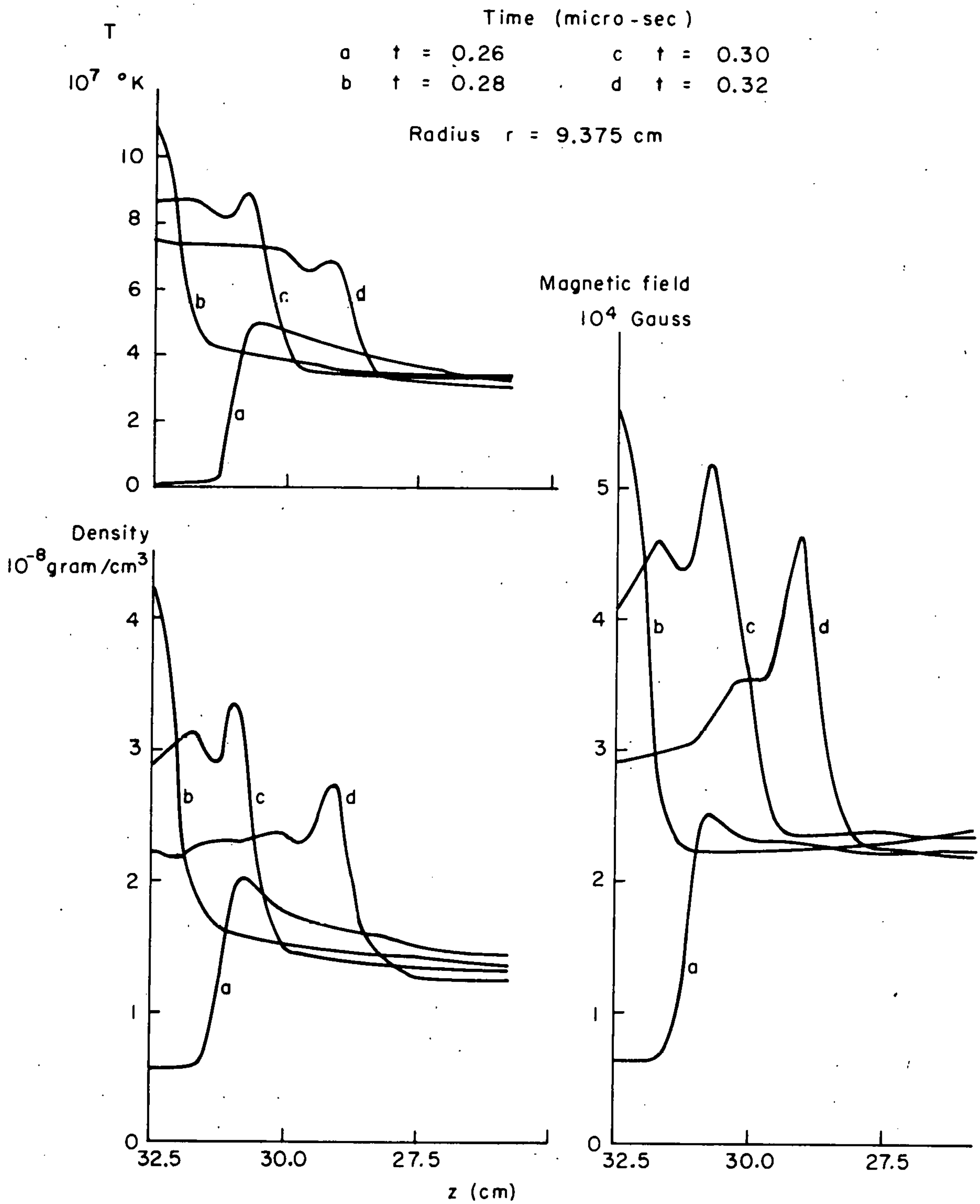



Fig. 5-3 The plots of the magnetic field, density and temperature with respect to the axial positions near the flat end wall with the constant driver.

A-907. ?724 W.L.

### 5.1.2 Sinusoidal Driver

The shock wave in Section 4.3 at 3.32 micro-seconds is utilized so that we can compare its reflection from a flat end wall with experimental measurements. The flow conditions of the incident shock wave are given in Section 4.3. The shock wave runs down the tube with a speed of about 78.2 cm/μsec.

Owing to the slope of the incident shock wave, the impact of the shock wave on the outer corner of the end wall produces a hot plasma with a temperature of about 7.8 Kev. Although the plasma in the tube may flow two-dimensionally, the reflected shock wave travels downstream almost like a simple one-dimensional reflection.

Fig. 5-4 shows the plots of the temperature, magnetic field, and density with respect to axial position in front of the flat end wall, which is at 180 cm, at a radius of 9 cm. The jumps of the temperature, magnetic field, and density between the reflected shock wave are from 2 Kev, 22 KG, and $2.6 \times 10^8$ gram/cm$^3$ to 5 Kev, 34 KG, and $4.6 \times 10^8$ gram/cm$^3$ respectively. The reflected shock speed is about 79 cm/μsec at a radius of 9 cm, slightly greater than the incident shock speed 78.2 cm/μsec. The incoming shock wave is in front of the flat end wall at 3.538 μsec. At 3.644 μsec, the shock wave reflects from the end wall and travels downstream. The motion of the shock wave and the plasma expansion are similar to those in the case of constant driver respectively.

Fig. 5-5 shows the time trace of the computed magnetic field at a fixed position of 175 cm with a radius of 9 cm in the shock tube. Qualitatively, it agrees with the experimental measurement in Fig. 5-6. The quantitative comparison between them is given below.

| | Magnetic Field (KG) | | |
|---|---|---|---|
| | bias | post-shock | maximum after reflection |
| computed (Fig. 5-5) | 6.6 | 24 | 36.8 |
| experimental (Fig. 5-6) | 6.6 | 19.8 | 42.9 |

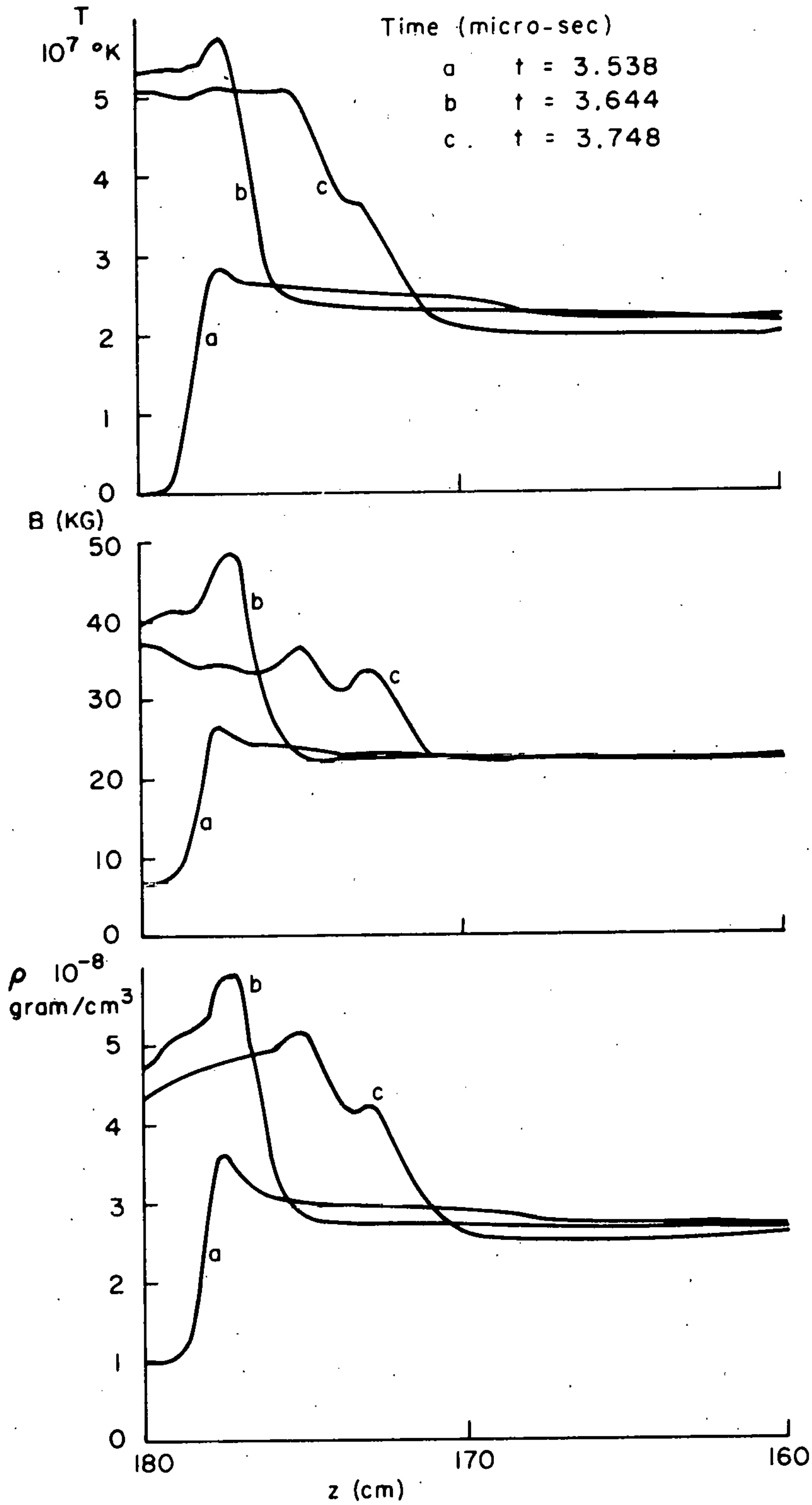



Fig. 5-4 The temperature, magnetic field and density at a radius of 9cm plotted versus axial positions in front of the flat end wall at different times.

A-907-ME-2

| | |
|---|---|
| Tube length: | 180 cm |
| Incoming shock speed: | 78 cm/micro-sec |
| End wall: | flat |
| Magnetic probe: | located at the position P |

Other flow variables:

| Time | T (°K) | $\rho$(gram/cm$^3$) |
|---|---|---|
| A | 600 | $.10688 \times 10^{-7}$ |
| B | $2.59 \times 10^7$ | $.3036 \times 10^{-7}$ |
| C | $5.04 \times 10^7$ | $.5194 \times 10^{-7}$ |
| D | $3.89 \times 10^7$ | $.323 \times 10^{-7}$ |

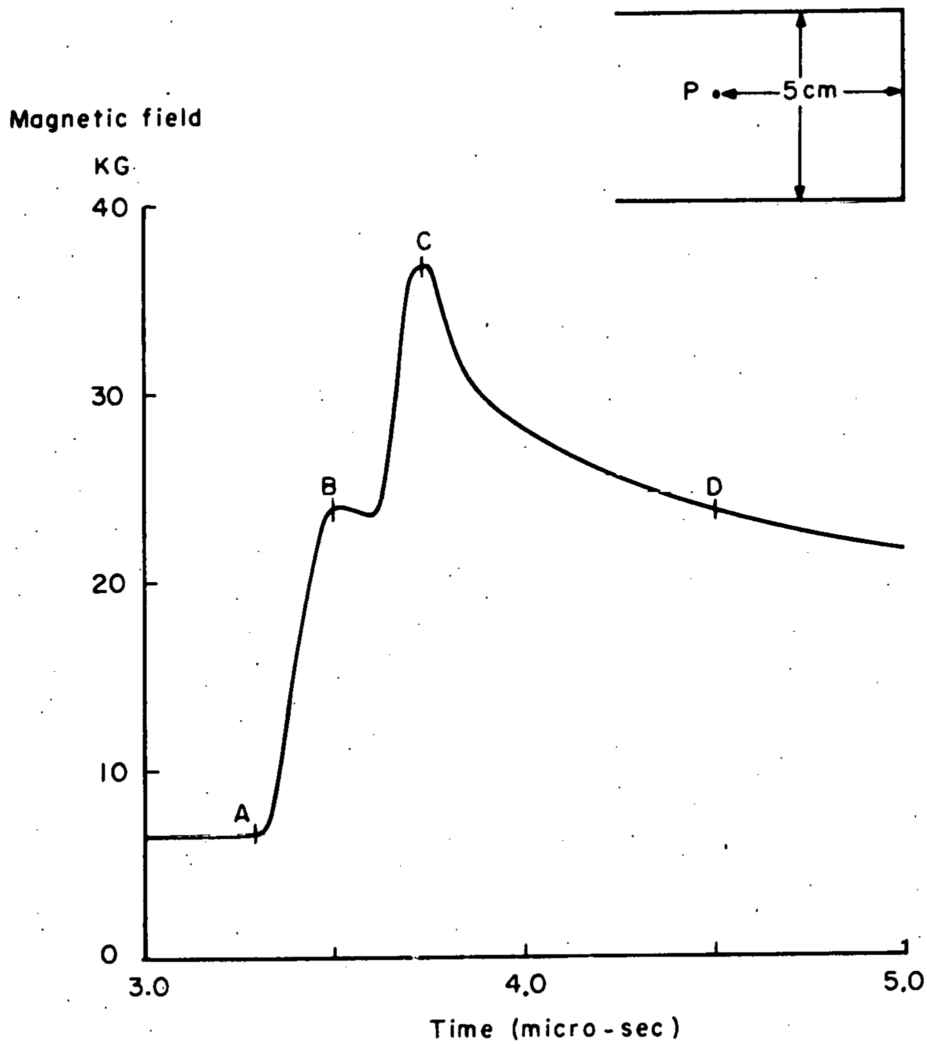



Fig. 5-5 The magnetic field vs. time measured at the position P.

A-907-ME-27
W.L.

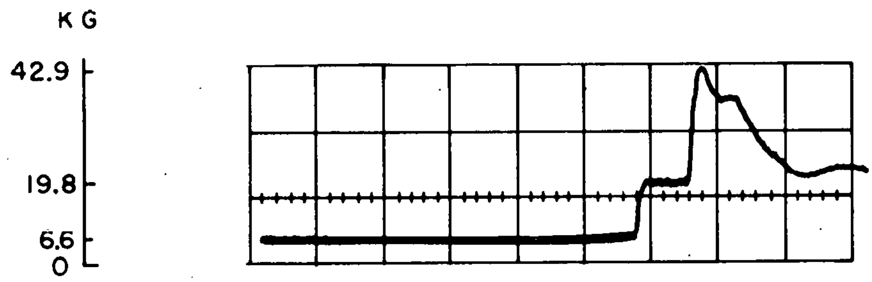



Fig. 5-6 The magnetic trace measured experimentally at the fixed station (175 cm) in front of the flat end wall in the shock tube which is 180 cm long.

A-907-ME-2735 W.L.

## 5.2 Semi-circular End Wall

In order to achieve better plasma heating by shock convergence and reflection, we consider a curved end wall instead of a flat end wall. From a practical point of view, it would be advantageous to use a semi-circular end wall instead of other curves, since it is easiest to manufacture. The same shock wave in Section 5.1.1 is again employed, so that we may study its convergence and reflection from such a semi-circular end wall, and compare the results with those of the flat end wall.

Fig. 5-7 shows the sequence of the shock flow in the tube with the wall profile at the right end. When the shock wave moves along the circular wall surface, it is strengthened near the side wall surfaces. The shock wave does not fully converge when it arrives at the end of the tube. Because the incident shock wave has a slope greater than $90^{o}$ with respect to the axial axis, the converged and reflected shock position is at a radius of 9.375 cm near the end wall surface.

At cycle = 701, the shock wave reflects off the wall surface. The plasma in front of the reflected shock still moves toward the focused position, but the hot plasma behind the reflected shock wave moves in the opposite direction. The reflected shock wave is approximately circular in shape in the focused region. Later on, its shape becomes flat when it reflects to the straight part of the tube. The maximum value of the temperature at the focused position is about 16.7 Kev, and the corresponding density and magnetic field are $5.36 \times 10^{-8}$ gram/cm$^3$

and $6.8 \times 10^4$ Gauss respectively. This temperature is only slightly higher than the 14.5 Kev obtained by flat wall reflection. Thus, the semi-circular end wall is not a very effective shape for plasma heating. (We shall see later that in plane geometry, it is actually more effective).

From the dynamical point of view, the plasma in a high pressure region will be driven to a lower pressure region. At cycle = 751, the reflected shock wave travels downstream, and the hot plasma expands into the vacuum near the inner conductor. The temperature contours indicate that there are recompressions on the side walls when the reflected shock wave reaches the straight part of the shock tube.

Fig. 5-8 shows the reflection of the shock wave from the semi-circular wall with flow variables plotted versus the axial position in front of the end wall, at a radius of 9.375 cm in the shock tube. The curve (a) of the temperature shows the incident shock wave; the curve (b) shows the reflection of the converged shock wave; and the curve (c) shows the reflected shock wave travelling downstream. When the hot plasma expands into the vacuum and the reflected shock wave moves away from the focused region, the magnetic field, density and temperature there decrease quickly.

Since the semi-circular end wall is not a good shape for plasma heating, in the next chapter, we shall find a wall shape which is more successful in converging shock waves using Whitham's shock-ray approximation.

Fig. 5-7

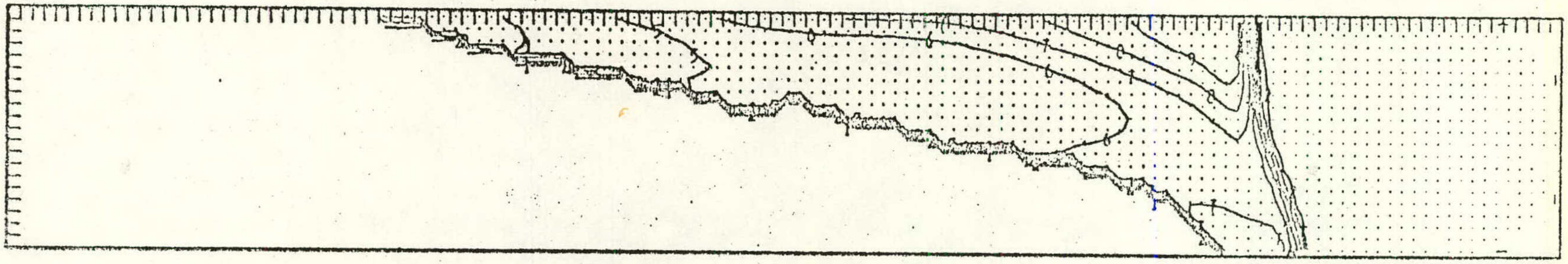
CYCLE= 551 ,TIME= 0.219979E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.702E+07 | 3 = | 0.140E+08 | 4 = | 0.210E+08 | 5 = | 0.281E+08 |
| 6 = | 0.351E+08 | 7 = | 0.421E+08 | 8 = | 0.491E+08 | 9 = | 0.561E+08 | 10 = | 0.631E+08 |

CYCLE=- ,TIME=n DENSITY SHADING

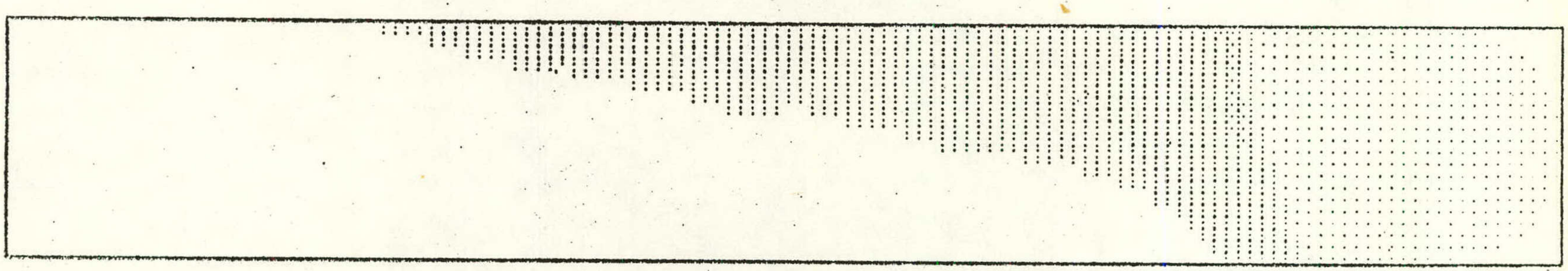
R



Z

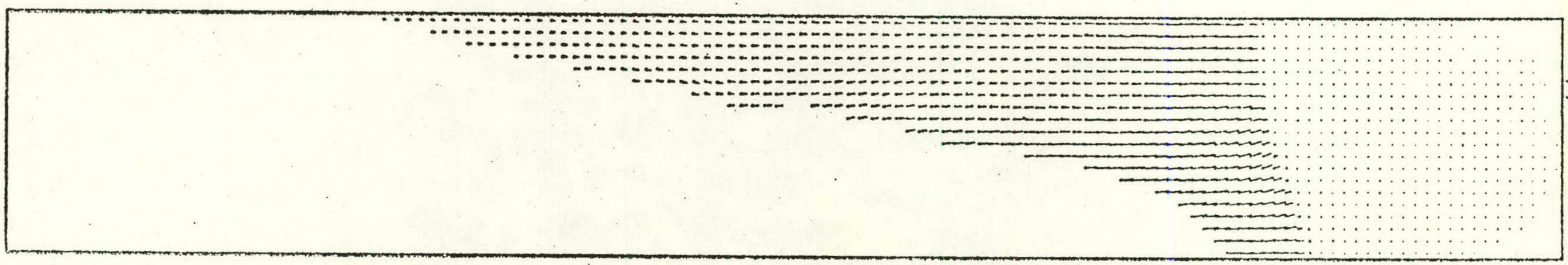
TEMPERATURE CONTOURS

R



DENSITY SHADING

R



VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 5-7

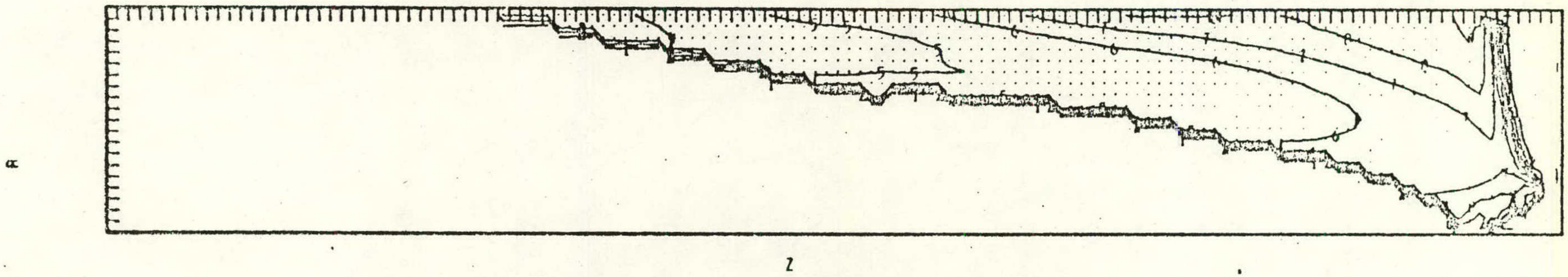
Page

CYCLE= 651 ,TIME= 0.259974E-06

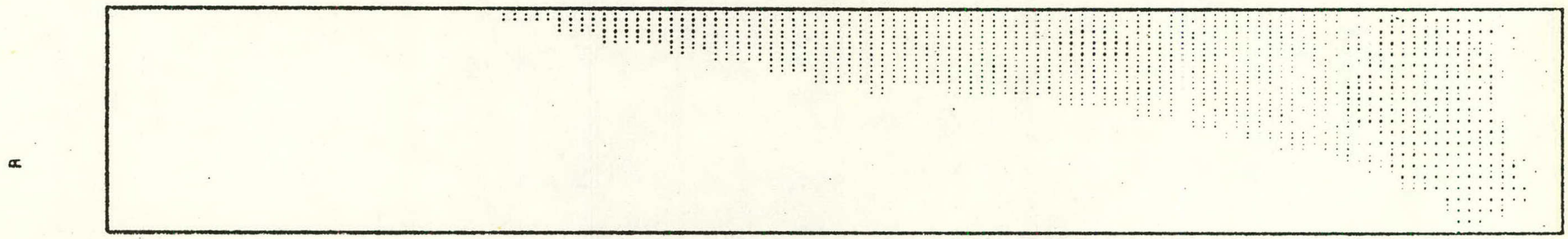
CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.719E+07 | 3 = | 0.144E+08 | 4 = | 0.216E+08 | 5 = | 0.288E+08 |
| 6 = | 0.360E+08 | 7 = | 0.432E+08 | 8 = | 0.504E+08 | 9 = | 0.576E+08 | 10 = | 0.648E+08 |

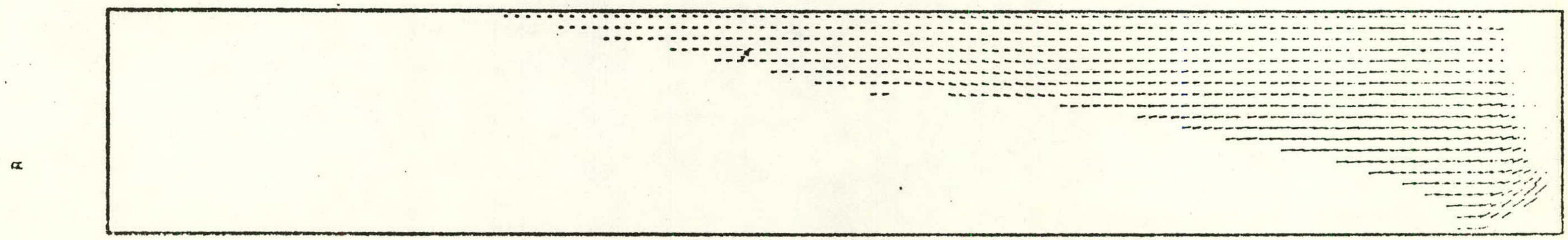
CYCLE=- ,TIME=n DENSITY SHADING




TEMPERATURE CONTOURS




DENSITY SHADING




VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 5-7

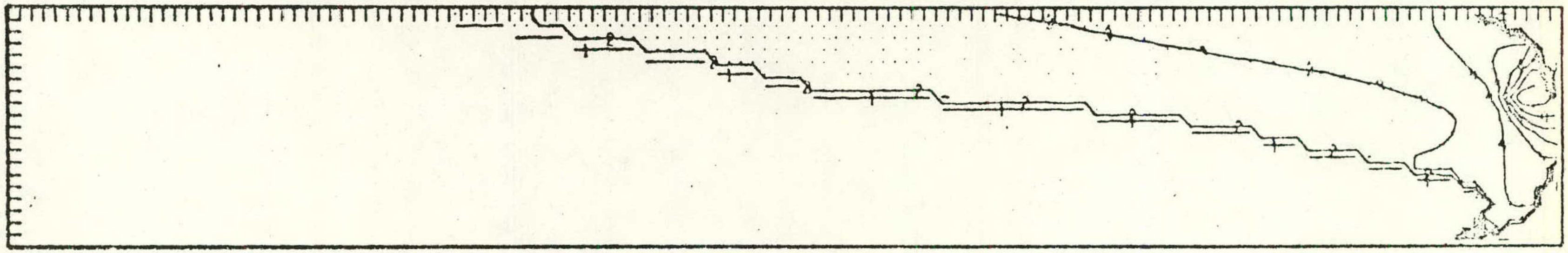
CYCLE= 701 ,TIME= 0.279972E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.185E+08 | 3 = | 0.371E+08 | 4 = | 0.556E+08 | 5 = | 0.742E+08 |
| 6 = | 0.927E+08 | 7 = | 0.111E+09 | 8 = | 0.130E+09 | 9 = | 0.148E+09 | 10 = | 0.167E+09 |

CYCLE= ,TIME= DENSITY SHADING

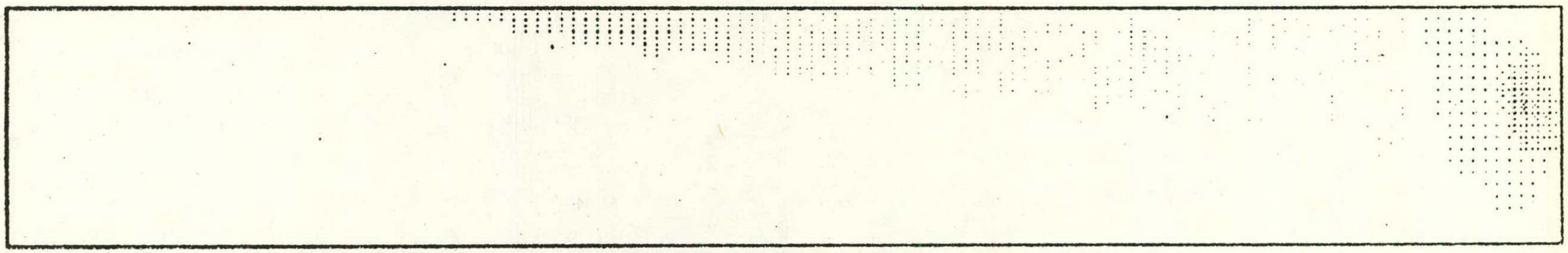
R



Z

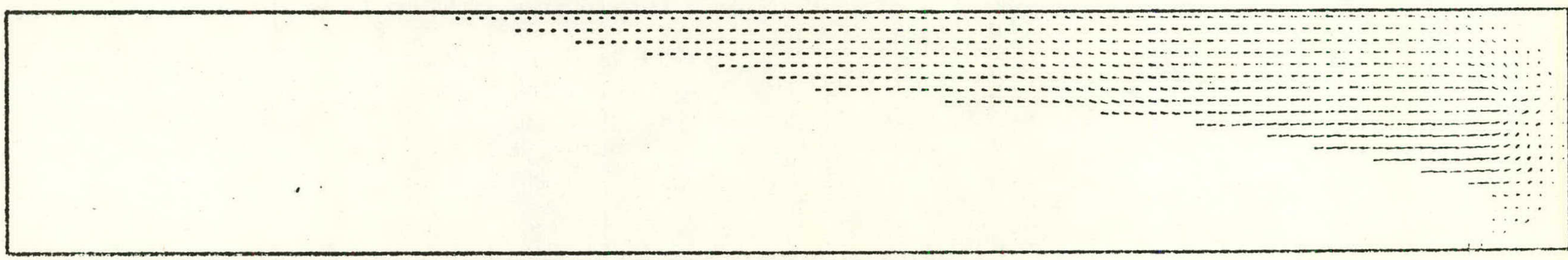
TEMPERATURE CONTOURS

R



DENSITY SHADING

R



VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

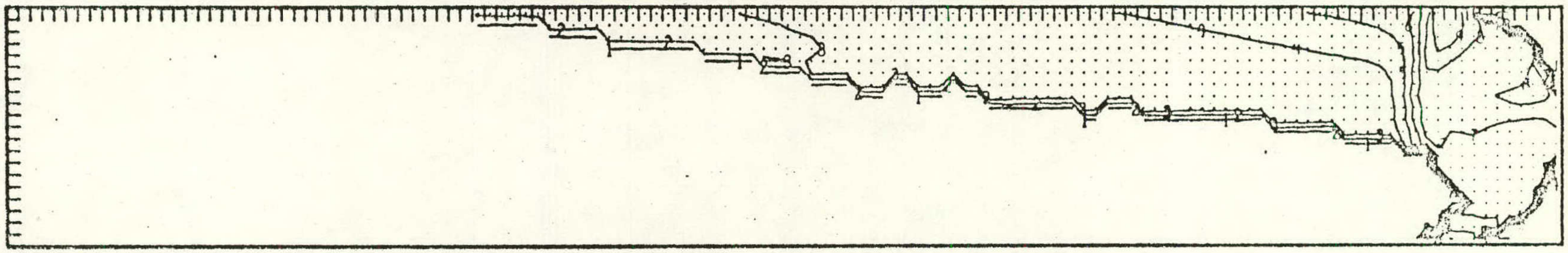
Fig. 5-7

CYCLE= 751 ,TIME= 0.299969E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.131E+08 | 3 = | 0.262E+08 | 4 = | 0.392E+08 | 5 = | 0.523E+08 |
| 6 = | 0.654E+08 | 7 = | 0.785E+08 | 8 = | 0.916E+08 | 9 = | 0.105E+09 | 10 = | 0.118E+09 |

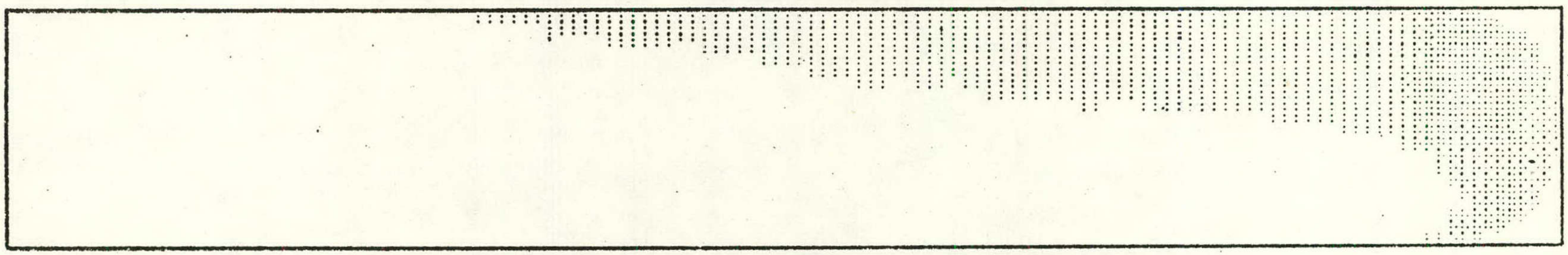
CYCLF=- ,TIME=n DENSITY SHADING



R

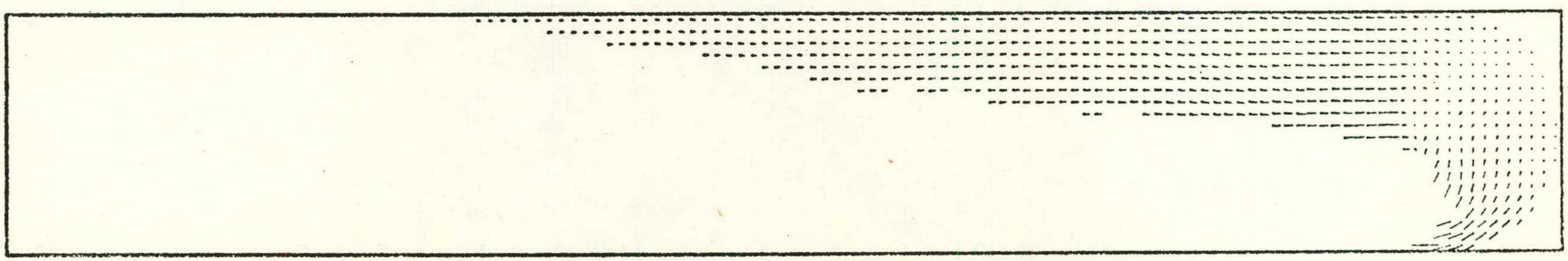
Z

TEMPERATURE CONTOURS



R

DENSITY SHADING



R

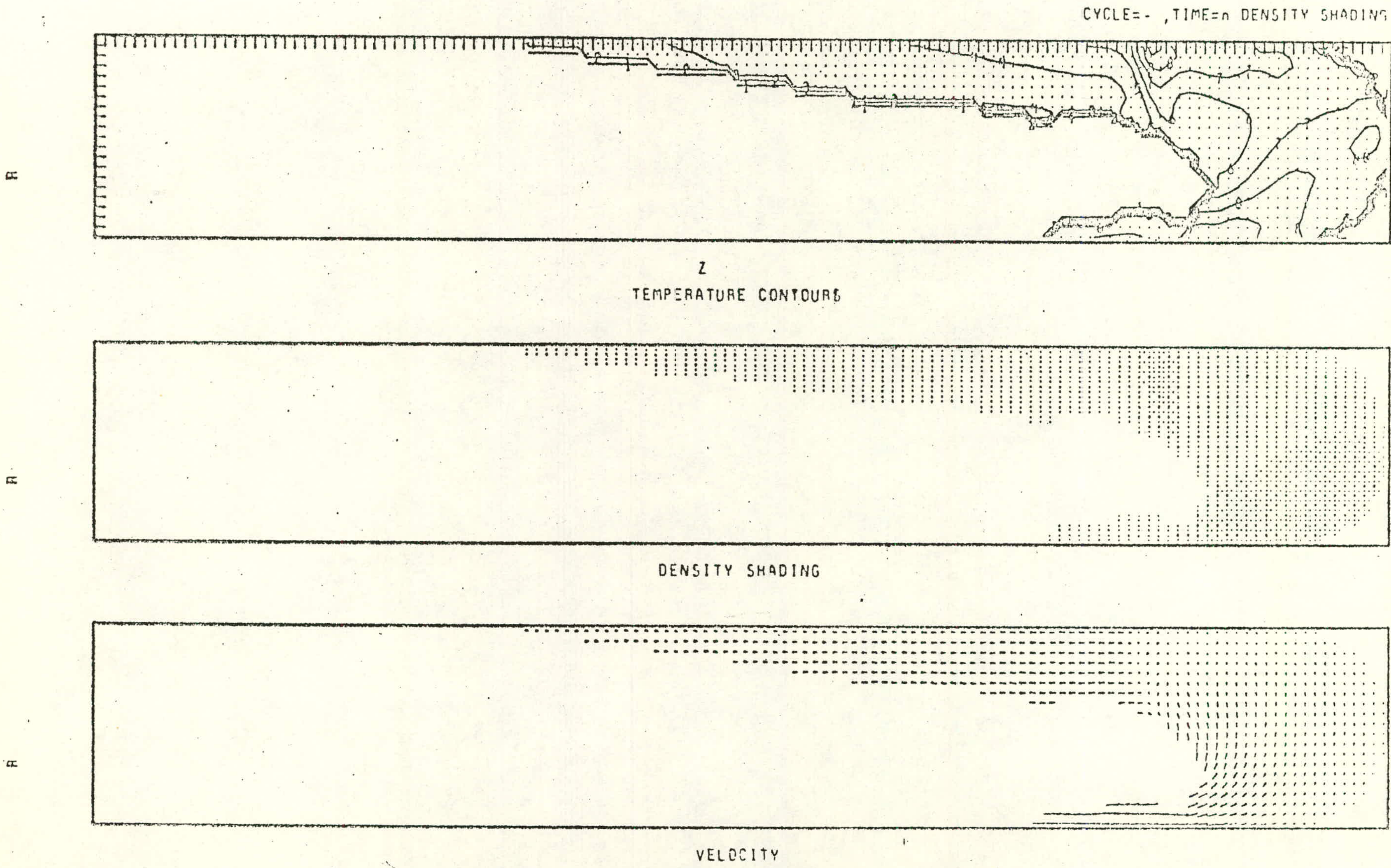
VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

CYCLE= 851 ,TIME= 0.339964E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.109E+08 | 3 = | 0.219E+08 | 4 = | 0.328E+08 | 5 = | C.438E+08 |
| 6 = | 0 547E+08 | 7 = | 0.657E+08 | 8 = | 0.766E+08 | 9 = | 0.876E+08 | 10 = | C.985E+08 |




NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 5-7 The incident shock wave and its reflection from a semi-circular end wall

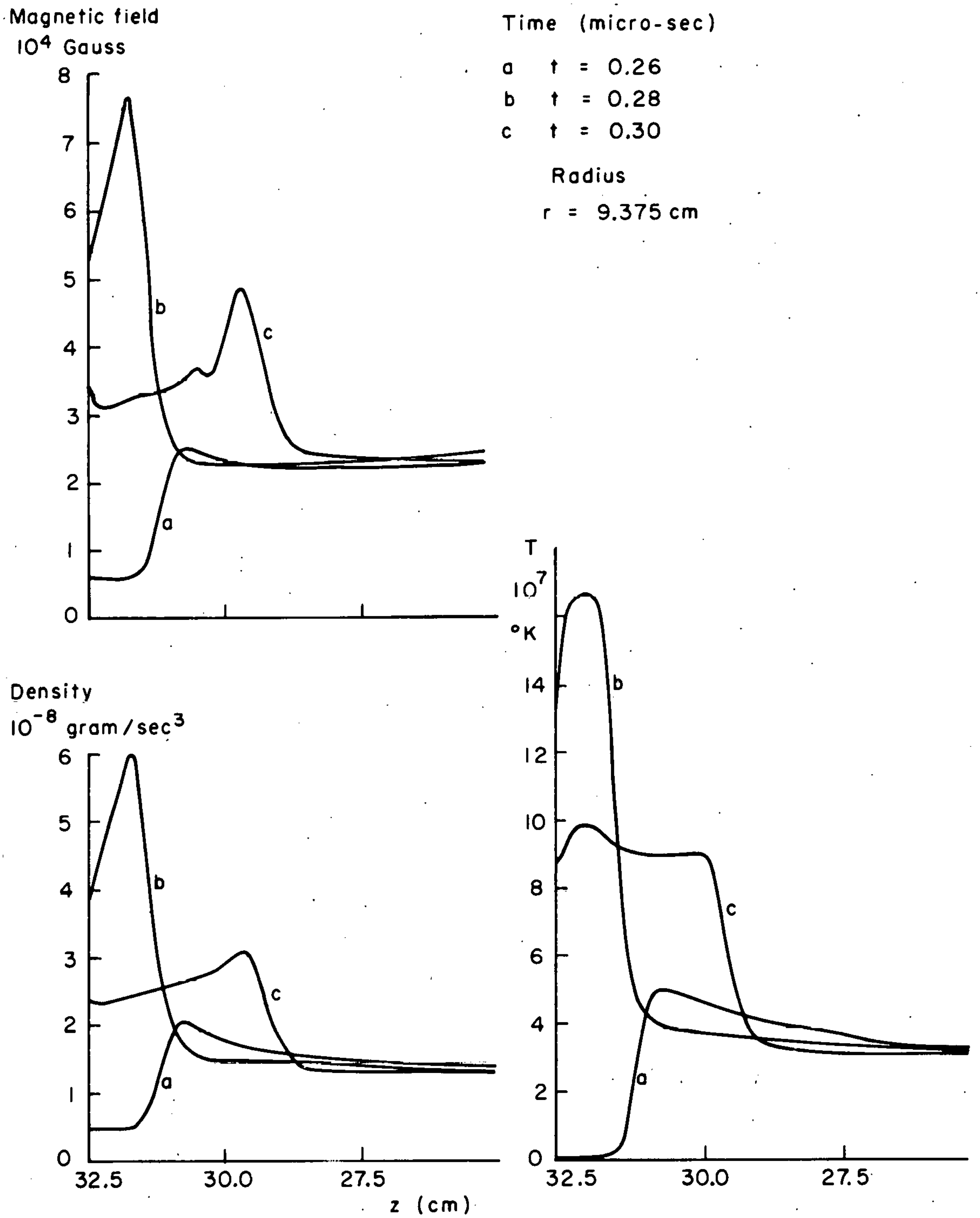



Fig. 5-8 The plots of the magnetic field, density and temperature with respect to the axial positions near the circular end wall.

# VI. SHOCK-RAY APPROXIMATION IN TRANSVERSE MHD FLOWS

In order to select a wall profile and to determine the location for shock focusing and reflection from a curved end wall in the coaxial shock tube, Whitham's shock-ray approximation is reviewed and extended to plane transverse MHD flows. The system of equations in a two-dimensional transverse MHD flow is reducible to the case of a conventional two-dimensional unsteady fluid flow[23]. Therefore, we can apply Whitham's formulations and Lau's shaping method[20] to design the end wall of the shock tube for shock convergence in transverse MHD flows.

According to the theory of sound, a wave front carrying a disturbance from a surface of arbitrary shape moves with the local speed of sound along the rays which are normal to the surface. Rays are orthogonal trajectories of successive positions of the wave front. They may be considered as carrying the discontinuities. Similarly, the particles immediately behind a shock wave move perpendicularly to the shock front, i.e. along the ray directions. In particular, a wall profile is a ray.

Whitham extended the theory of geometrical acoustics[24] to the shock-ray approximation[17,18] by constructing shock-ray coordinates in the Cartesian space and by formulating the kinematic relations. He made the assumption that the shock Mach number and the area between ray tubes were functionally related. A smoothly converging wall will increase the shock Mach number and generate a geometrical wave along the shock surface. For a properly shaped wall, focusing of the shock into a small region will occur.

We now extend this approximation to the transverse MHD flows. Let the shock position be defined by the curves $\alpha$ = const and the rays by $\beta$ = const, with $\alpha(x,y,z) = C_0 t$. $C_0$ is the local magnetosonic speed in front of the shock wave. The $\alpha - \beta$ curves in two-dimensional axisymmetric $(r - z)$ coordinates are shown in Fig. 6-1.

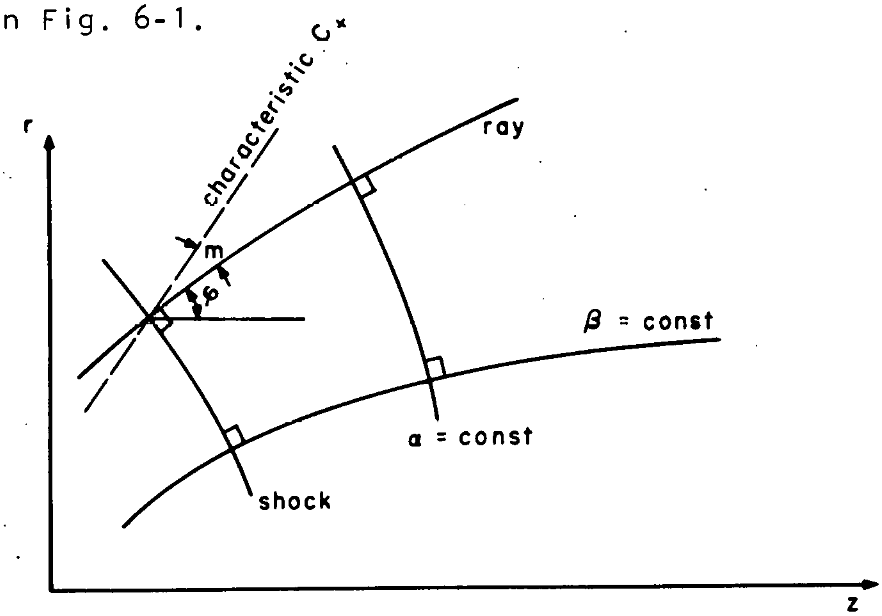



**Fig. 6-1 Orthogonal shock-ray curves.**

Using Whitham's formulations[18], the following kinematic relations can be obtained for transverse MHD shock flows in axisymmetric geometry.

$$\alpha_z = \cos\varphi / M^* , \quad \alpha_r = \sin\varphi / M^* , \tag{6.1}$$

$$\frac{\partial}{\partial z}\left(\frac{\sin\varphi}{M^*}\right) - \frac{\partial}{\partial r}\left(\frac{\cos\varphi}{M^*}\right) = 0 , \tag{6.2}$$

$$\frac{\partial}{\partial z}\left(\frac{r\cos\varphi}{A}\right) + \frac{\partial}{\partial r}\left(\frac{r\sin\varphi}{A}\right) = 0\,, \tag{6.3}$$

where the subscripts denote derivatives, $M^*$ is the magnetosonic Mach number and $A$ is the area between ray tubes. $\varphi$ is the angle between a ray and the z-axis as shown in Fig. 6-1. The following derivation of the dynamical equation, $A = A(M^*)$, is similar to Whitham's work[25].

In a transverse MHD flow, the equations of motion in a non-uniform tube are

$$\rho_t + u\rho_z + \rho\left(u_z + \frac{u}{A}\frac{dA(z)}{dz}\right) = 0\,, \tag{6.4}$$

$$u_t + uu_z + \frac{1}{\rho}P^*_z = 0\,, \tag{6.5}$$

$$P^*_t + uP^*_z - C^2(\rho_t + u\rho_z) = 0\,, \tag{6.6}$$

$$\frac{B}{\rho} = \text{const}\,, \tag{6.7}$$

$$P^* = P + \frac{B^2}{8\pi}\,, \tag{6.8}$$

$$C^2 = \frac{\partial P^*}{\partial \rho}\,, \tag{6.9}$$

where $C$ is the magnetosonic speed, $C^2 = a^2 + b'^2$, with $a$ being the sound speed and $b'\left(=\frac{B}{\sqrt{4\pi\rho}}\right)$ being the Alfven speed. The pressure term due to the area change does not appear in Eq. (6.5) because it can be neglected in a smoothly converging tube.

The characteristic form in (z - t) plane is

$$\left(\frac{P^*_t}{u \pm C} + P^*_z\right) \pm \rho C \left(\frac{u_t}{u \pm C} + u_z\right) + \frac{\rho C^2 u}{(u \pm C) A} \frac{dA(z)}{dz} = 0, \quad (6.10)$$

$$\left(\frac{P^*_t}{u} + P^*_z\right) - C^2 \left(\frac{\rho_t}{u} + \rho_z\right) = 0 . \quad (6.11)$$

The equations (6.10) and (6.11) can be rewritten as follows:

$$dP^* + \rho C du + \frac{\rho C^2 u}{u + C} \frac{dA}{A} = 0 \qquad \text{on } C_+ : \frac{dz}{dt} = u + C, \quad (6.12)$$

$$dP^* - \rho C du + \frac{\rho C^2 u}{u - C} \frac{dA}{A} = 0 \qquad \text{on } C_- : \frac{dz}{dt} = u - C, \quad (6.13)$$

$$dP^* - C^2 d\rho = 0 \qquad \text{on } P : \frac{dz}{dt} = u . \quad (6.14)$$

The transverse MHD shock jump conditions with an infinite shock strength ($M^* \approx \infty$) are

$$P^* = P^*_o + \rho_o C_o^2 M^{*2} \frac{2}{\gamma+1} , \quad (6.15)$$

$$\rho = \rho_o \frac{\gamma + 1}{\gamma - 1} , \quad (6.16)$$

$$u = M^* \frac{2C_o}{\gamma + 1} , \quad (6.17)$$

$$C = C_o M^* \sqrt{2\gamma(\nu-1)} / (\gamma+1) , \quad (6.18)$$

where the subscript (o) denotes a quantity in front of the shock wave.

Since the shock wave is along $C_+$, we get the following dynamical relation by substituting Eqs. (6.15),(6.16), (6.17), and (6.18) into Eq. (6.12).

$$\frac{dA}{A}\frac{M^*}{dM^*} = f(\gamma) = -\left\{ \frac{2}{\gamma} + 1 + \sqrt{\frac{1}{\gamma(\gamma-1)}} + \sqrt{\left(1-\frac{1}{\gamma}\right)} \right\} \tag{6.19}$$

The result is similar to Whitham's formula for very strong shocks[25]. Although the dynamical relation (6.19) is derived from the equations of plane transverse MHD flows, we still use it as an approximation to the coaxial shock tube by treating the propagation of the shock wave down each ray tube separately.

With the dynamical equation (6.19), we can rewrite hyperbolic Eqs. (6.2) and (6.3) in the characteristic forms.

$$d\varphi + \frac{1}{\tan(\pm m)}\frac{dM^*}{M^*} + \frac{\tan\varphi\tan(\pm m)}{\tan\varphi + \tan(\pm m)}\frac{dr}{r} = 0 ,$$

$$\text{along } \frac{dr}{dz} = \tan(\varphi \pm m) \tag{6.20}$$

$$\tan m = \sqrt{\frac{-A}{M^*}\frac{dM^*}{dA}} = \frac{1}{2.15} \quad (\text{with } \gamma = 5/3) , \tag{6.21}$$

where m is the angle between a ray and a characteristic line $C_+$. These equations are of the same form as the gas dynamical equations, with their characteristics, Riemann invariants, shocks, etc. The propagation speed of the geometrical wave,

$$\frac{d\beta}{d\alpha} = \pm\left( - \frac{dM^{*2}}{dA^2} \right)^{\frac{1}{2}} r$$

in an axisymmetric case, is an increasing function of magnetosonic Mach number M*. The geometrical waves with increased magnetosonic Mach number will eventually break and form a shock, in this case, a "shock-shock" called by Whitham. Such a "shock-shock" which propagates on the shock surface is a discontinuity in magnetosonic Mach number ($M^*$) and shock slope ($\varphi$).

With Eqs. (6.20) and (6.21), we follow Lau's wall shaping method using the theory of characteristics to find the end wall profile for shock focusing in the coaxial shock tube. The incident shock wave is assumed to be a plane shape. Because the Eqs. (6.20) and (6.21) are completely similar to those used by Lau, the wall shape for shock focusing in our case will be exactly the same as that found by him. It is approximately a quarter of the ellipse with the ratio of the major axis to the minor axis equal to 2.25 [= (2.15 + 0.1)]. This is the optimal ratio, as the shock focuses to a point in the r - z plane, or actually to a ring. The difference between two slightly different wall shapes will eventually become negligible when a shock wave is converged into a same location of the bottom of the end wall. The elliptic shape makes the numerical computation particularly easy. In Fig. 6-2, the line BC is the first positive characteristic due to the wall curvature. The dimension of the triangle ABC is given by Eq. (6.21), i.e., $\tan(m) = AC/AB = 1/2.15$. The numerical value (0.1) is chosen to make the ellipse following Lau's wall shape fairly. It can be changed to other small values. The focused location is arranged to be near the inner conductor according to Eq. (6.20). The end wall (CE) in the shock tube is shown in Fig. 6-2.

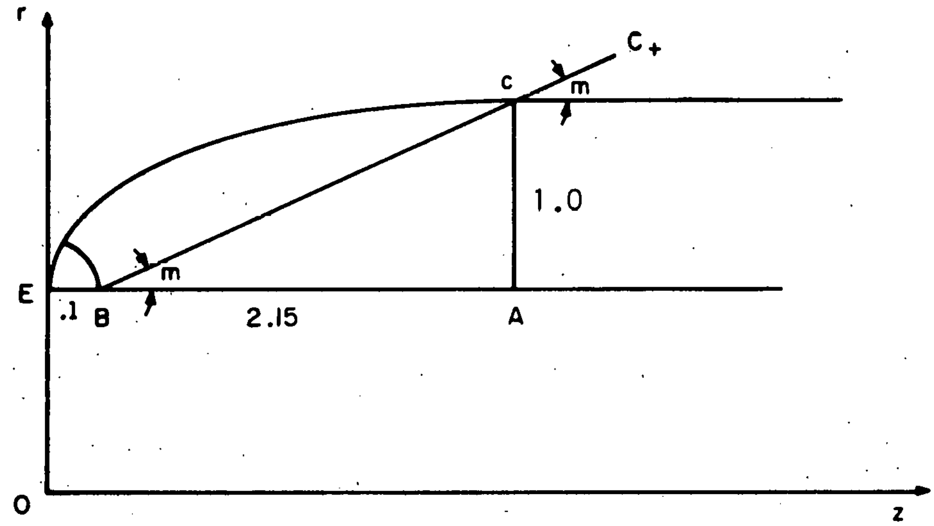



Fig. 6-2 The Selected Elliptic End Wall.

# VII. SHOCK FOCUSING AND REFLECTION

## 7.1 Quarter-elliptic End Wall

The elliptic wall designed in the previous chapter will now be checked out by computation. The shock wave of Section 4.2 is used again so that we can study its focusing and reflection. The solution is presented in the plottings such as Figs. 7-1, 7-2, and 7-3.

Fig. 7-1 shows the sequence of the shock flow in the tube with the elliptic end wall located at the right end. These pictures are helpful to us in understanding the details of the flow, the process of shock focusing and reflection, and plasma heating. At cycle = 501, the incident shock wave runs down the tube with a speed of 125 cm/μsec. At cycle = 601, the shock wave moving through the converging channel is diffracted and strengthened on the side wall, and the plasma behind the wall shock is further compressed. The fluid particles behind the wall shock travel along the wall surface.

At cycle = 651, the wall shock and the incoming main shock wave make an angle of about $110^{o}$. The geometrical wave on the shock surface now breaks into a geometric shock (Mach reflection). Whitham called such a geometric discontinuity on the shock surface a "shock-shock". The magnetosonic Mach number and the angle (φ) of the ray on the wall shock differ with those on the main shock wave respectively. No matter what the shock shape is at the bottom of the end wall, the shock will inevitably reach a maximum in strength, i.e., shock focusing. Then, the high kinetic energy of a fast moving particle suddenly will change into plasma thermal energy in the focused region.

At cycle = 701, the reflected shock wave has indeed focused and produces a small region of hot plasma with a temperature of 32 Kev, or some 8 times the incident post-shock temperature. The maximum density in the focused region is $8.12 \times 10^{-8}$ gram/cm$^3$, or 16 times the initial density in the shock tube. The duration of the hot plasma is quite short, being equal to the transit time of the reflected shock wave across the major axis of the ellipse.

At cycle = 751, the shock again reflects from the focused region, and behind the reflected shock is a rarefaction wave. The velocities of the incoming particles and the reflected particles between the reflected shock wave are in opposite directions. Then, the reflected shock wave propagates downstream near the outer wall, and moves into the vacuum near the inner conductor. The plasma expands very rapidly along the surface of the inner conductor.

Fig. 7-2 shows the plots of the magnetic field, density and temperature with respect to axial position in the focused region with a radius of 6.625 cm at different times. The bottom of the elliptic end wall is at 33.75 cm. At 0.28 microseconds, shock focusing and reflection make all the flow variables peak at the focused position.

Fig. 7-3 shows the traces of the magnetic field, density and temperature with respect to time at a fixed station near the focused region. The station is shown on the first page of Fig. 7-1 as the point (G). A shock compression raises the total pressure (P*) due to the arrival of the incoming shock wave. When the shock wave moves away from the station, the total pressure drops since the magnetic field and density of the plasma just behind the incident shock wave are higher than those of the downstream plasma; this is due to the effects of the side wall and drive current. A furhter shock compression occurs when the reflected shock wave returns to the station. Finally, the total pressure goes down quickly as the reflected shock wave moves away from the station. Milton[19] pointed out that ordinary gas shock convergence and reflection by a curved wall in plane flows follows purely the cylindrical implosion cycle[26,27], and that the flow conditions in the area contraction of the shock wave are not uniform. In our case, the magnetic field in front of the incident shock wave varies radially. However, some features of cylindrical implosion appear in this case.

To keep the temperature of the plasma high, it is necessary to keep the plasma from expanding. An externally applied magnetic field after shock reflection at a suitable downstream position will slow the plasma expansion resulting in extended heating. The plasma under the applied magnetic fields will not expand when the applied magnetic energy density is much greater than the total gas energy density.

Fig. 7-1

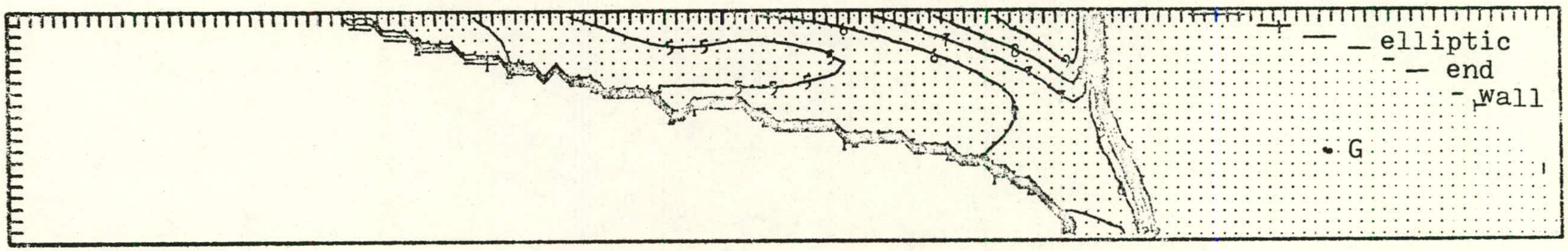
CYCLE= 501 ,TIME= 0.199982E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.739E+07 | 3 = | 0.148E+08 | 4 = | 0.222E+08 | 5 = | 0.296E+08 |
| 6 = | 0.369E+08 | 7 = | 0.443E+08 | 8 = | 0.517E+08 | 9 = | 0.591E+08 | 10 = | 0.665E+08 |

CYCLE=YC,TIME=%CDENSITY SHADING&VELO

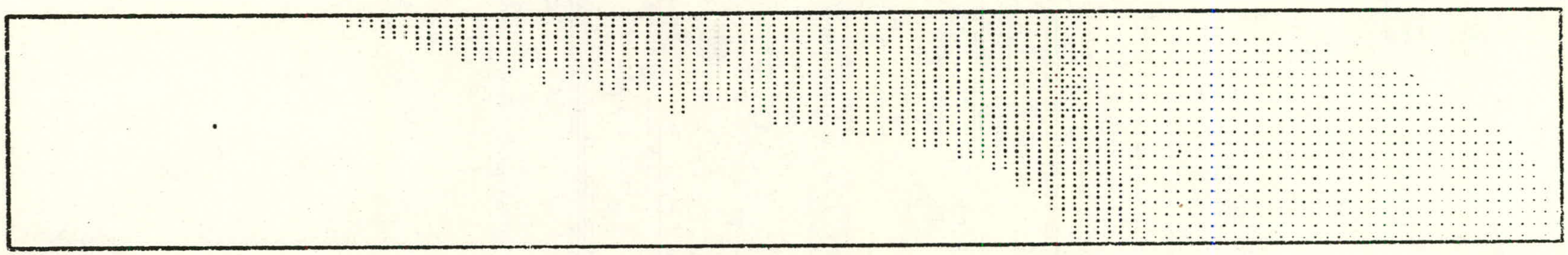
R




Z

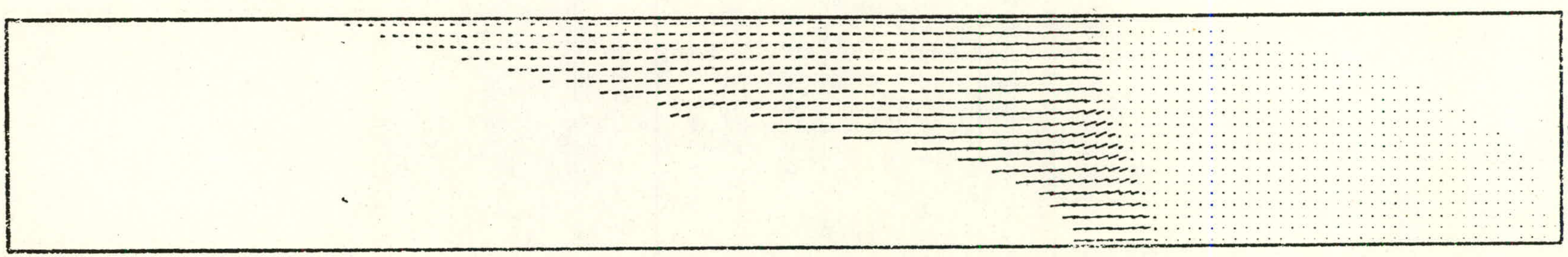
TEMPERATURE CONTOURS

R



DENSITY SHADING

R



VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 7-1

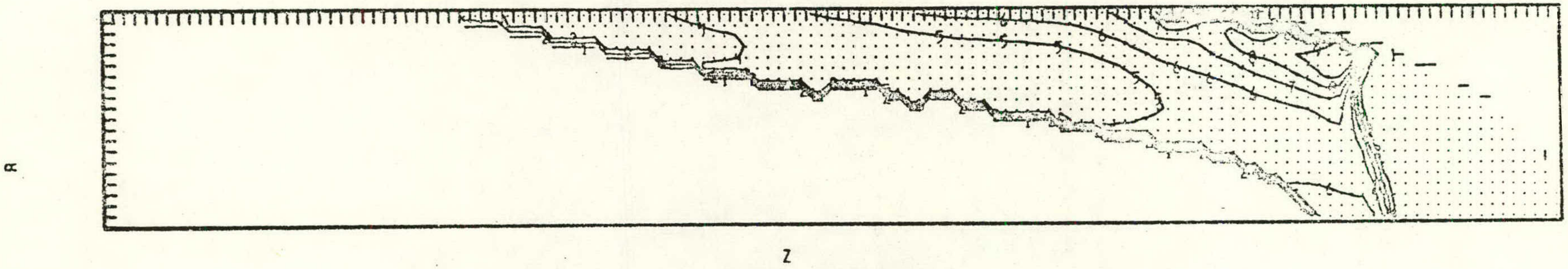


CYCLE= 601 ,TIME= 0.239??71 06

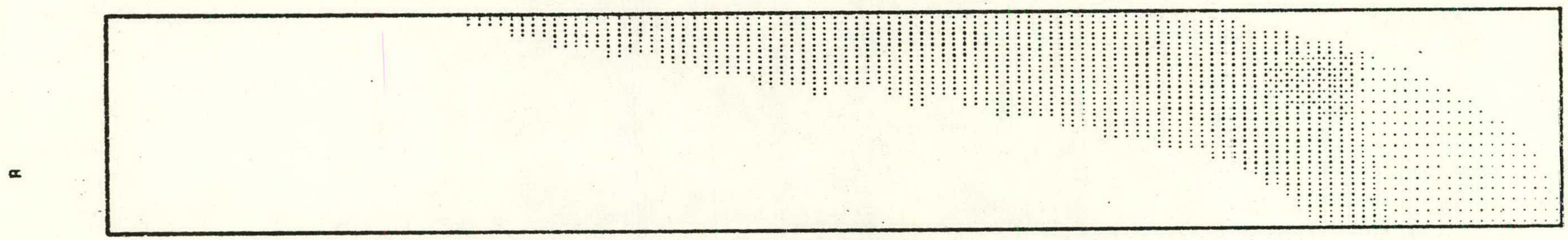
CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.867E+07 | 3 = | 0.173E+08 | 4 = | 0.260E+08 | 5 = | 0.347E+08 |
| 6 = | 0.433E+08 | 7 = | 0.520E+08 | 8 = | 0.607E+08 | 9 = | 0.694E+08 | 10 = | 0.780E+08 |

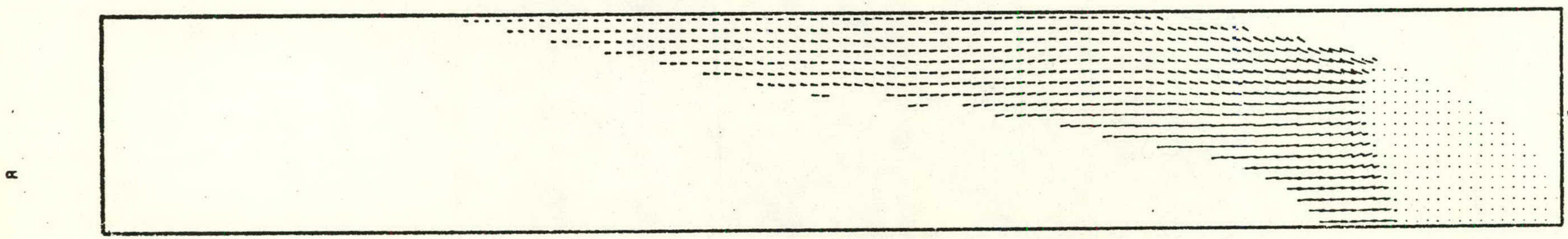
CYCLE=Y0,TIME=%CDENSITY SHADINGAVFL?




TEMPERATURE CONTOURS




DENSITY SHADING




VELOCITY

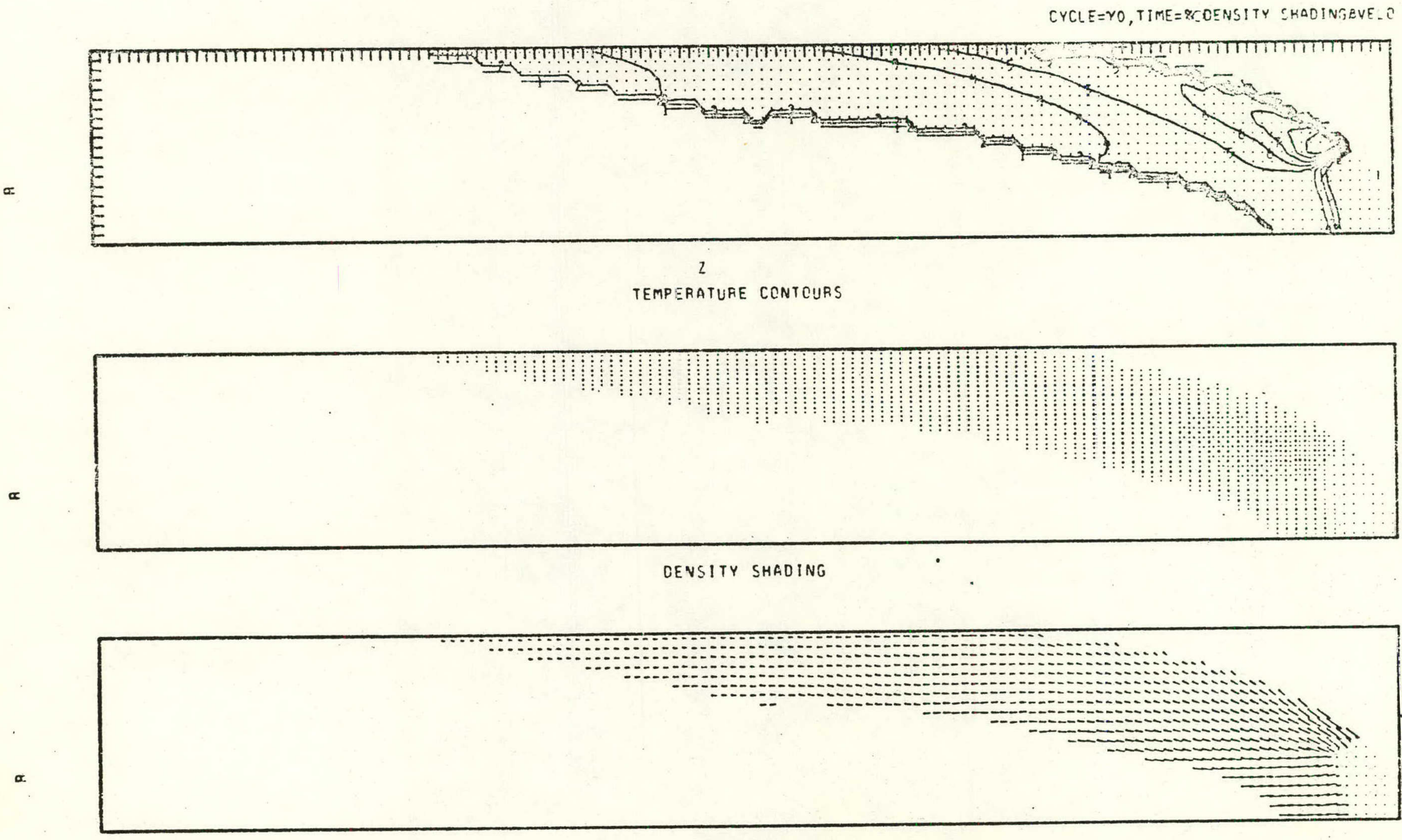
NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 7-1

CYCLE= 651 ,TIME= 0.259974[illegible]06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.124E+08 | 3 = | 0.248E+08 | 4 = | 0.371E+08 | 5 = | 0.495E+08 |
| 6 = | 0.619E+08 | 7 = | 0.743E+08 | 8 = | 0.867E+08 | 9 = | 0.990E+08 | 10 = | 0.111E+09 |




NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 7-1

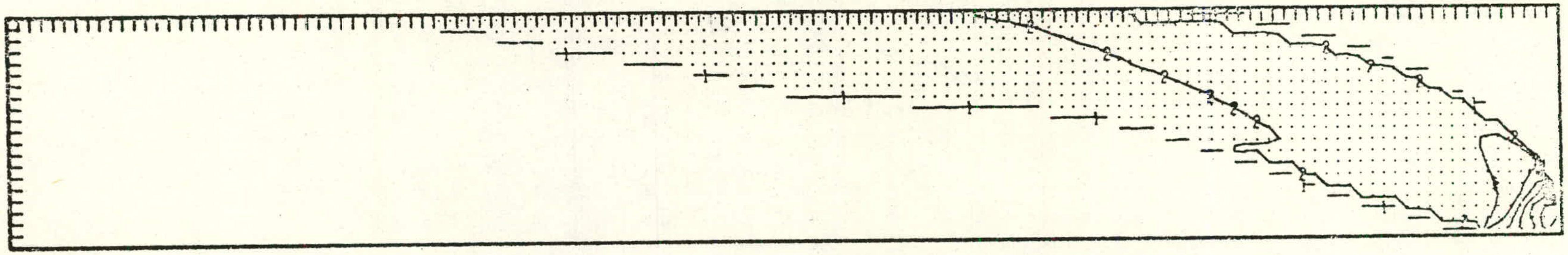
CYCLE= 701 ,TIME= 0.279972E-06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.381E+08 | 3 = | 0.761E+08 | 4 = | 0.114E+09 | 5 = | 0.152E+09 |
| 6 = | 0.190E+09 | 7 = | 0.228E+09 | 8 = | 0.266E+09 | 9 = | 0.304E+09 | 10 = | 0.342E+09 |

CYCLE=YO,TIME=%CDENSITY SHADING&VELO

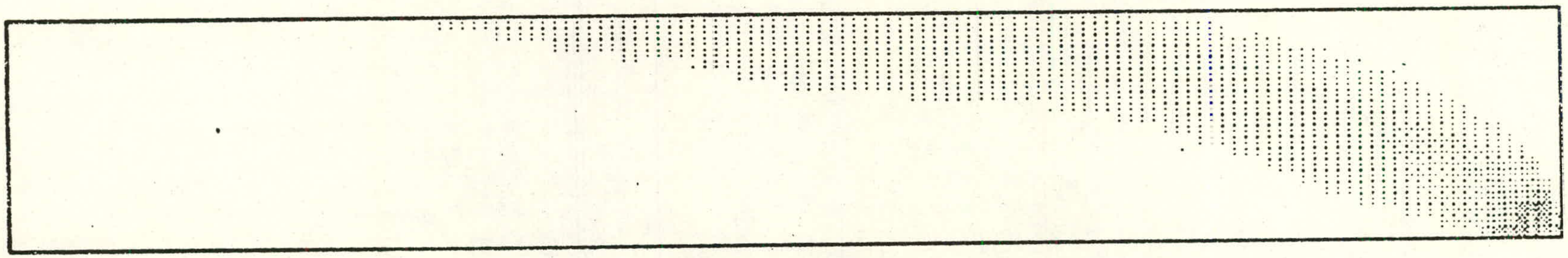
R



Z

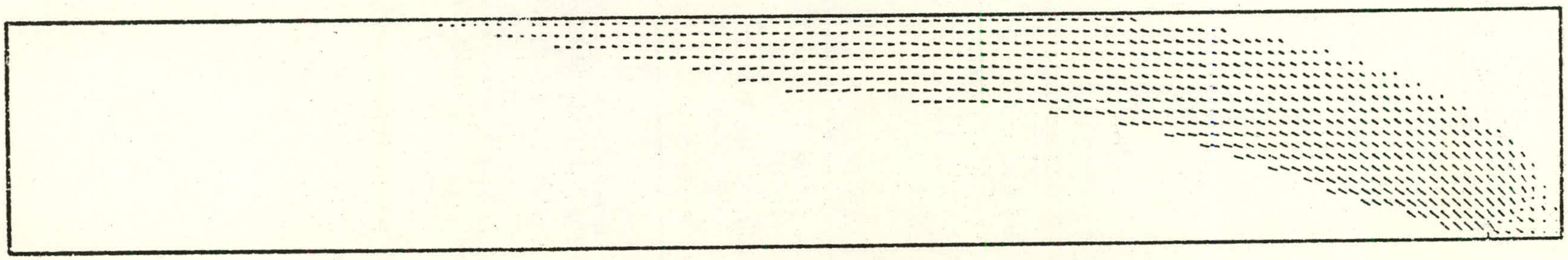
TEMPERATURE CONTOURS

R



DENSITY SHADING

R



VELOCITY

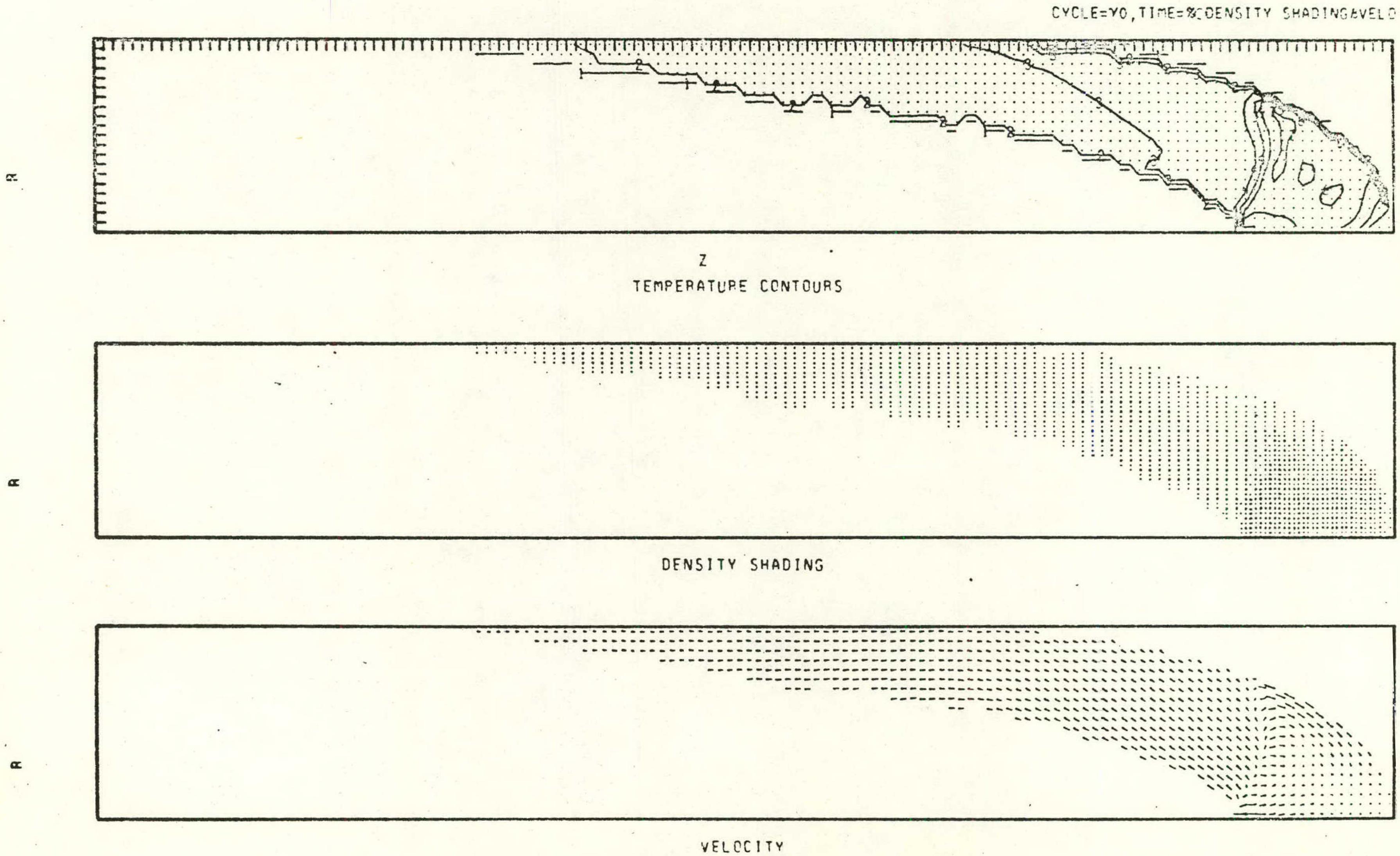
NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 7-1

CYCLE= 751 ,TIME= 0.2999691 06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.203E+08 | 3 = | 0.407E+08 | 4 = | 0.610E+08 | 5 = | 0.813E+08 |
| 6 = | 0.102E+09 | 7 = | 0.122E+09 | 8 = | 0.142E+09 | 9 = | 0.163E+09 | 10 = | 0.183E+09 |




NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 7-1

Page * 11

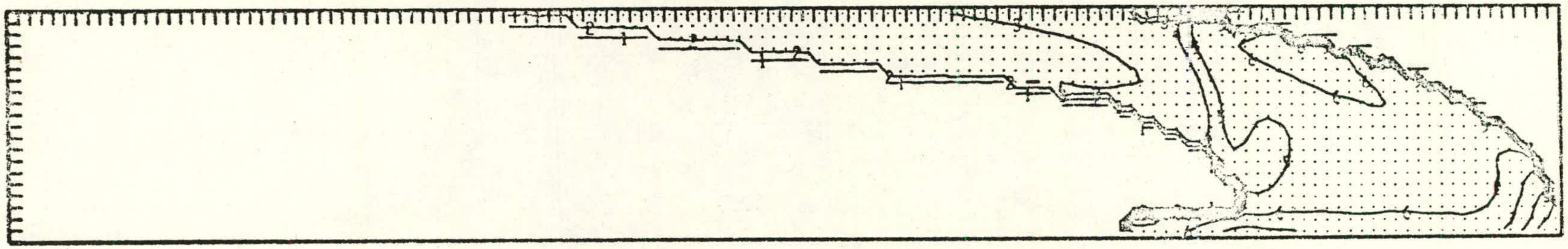
CYCLE= 851 ,TIME= 0.339964E 06

CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.166E+08 | 3 = | 0.332E+08 | 4 = | 0.498E+08 | 5 = | 0.665E+08 |
| 6 = | 0.831E+08 | 7 = | 0.997E+08 | 8 = | 0.116E+09 | 9 = | 0.133E+09 | 10 = | 0.150E+09 |

CYCLE=YO,TIME=%CDENSITY SHADINGAVELD

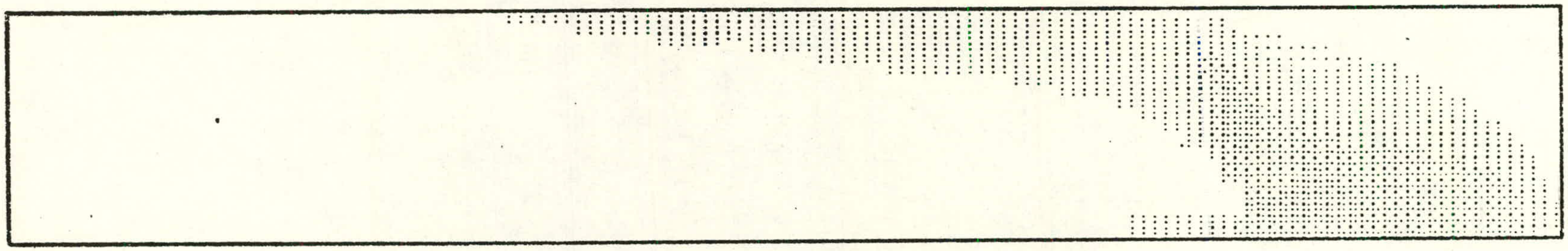
R



Z

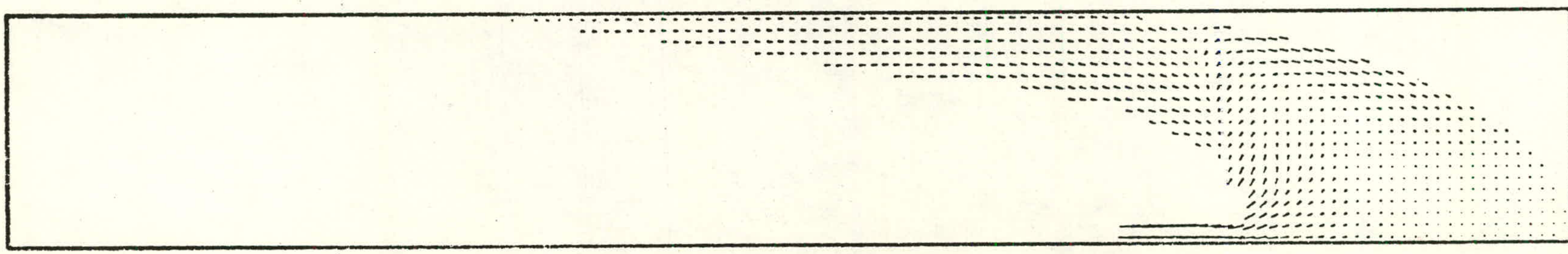
TEMPERATURE CONTOURS

R



DENSITY SHADING

R



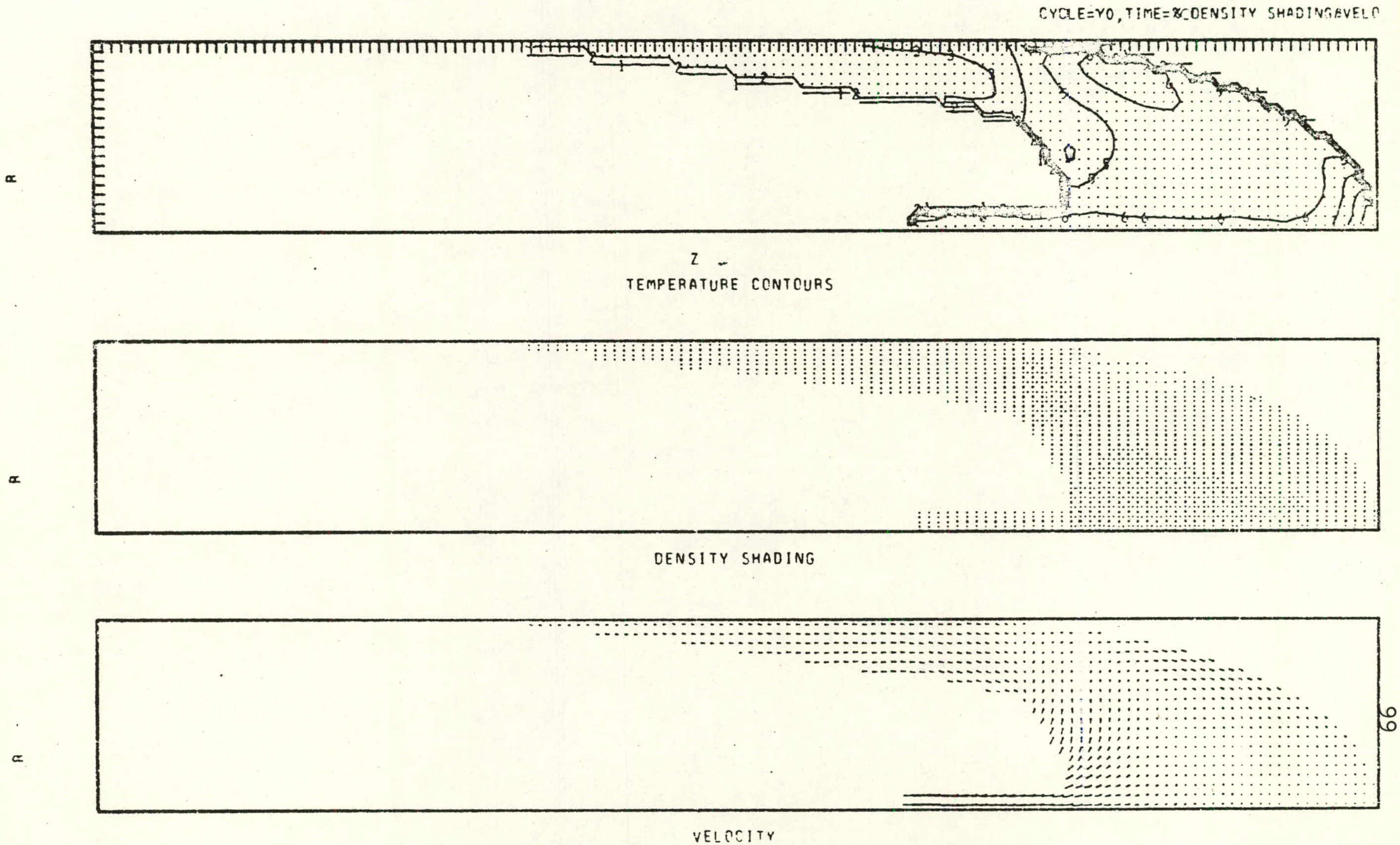
VELOCITY

NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

98

CYCLE= 901 ,TIME= 0.359962E-06
CONTOURS

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = | 0.000 | 2 = | 0.158E+08 | 3 = | 0.315E+08 | 4 = | 0.473E+08 | 5 = | 0.631E+08 |
| 6 = | 0.788E+08 | 7 = | 0.946E+08 | 8 = | 0.110E+09 | 9 = | 0.126E+09 | 10 = | 0.142E+09 |




NUMERICAL SHOCK WAVES SIMULATIONS AT COLUMBIA

Fig. 7-1 Shock focusing and reflection from the selected quarter-elliptic end wall

A-907- -2731 W.L.

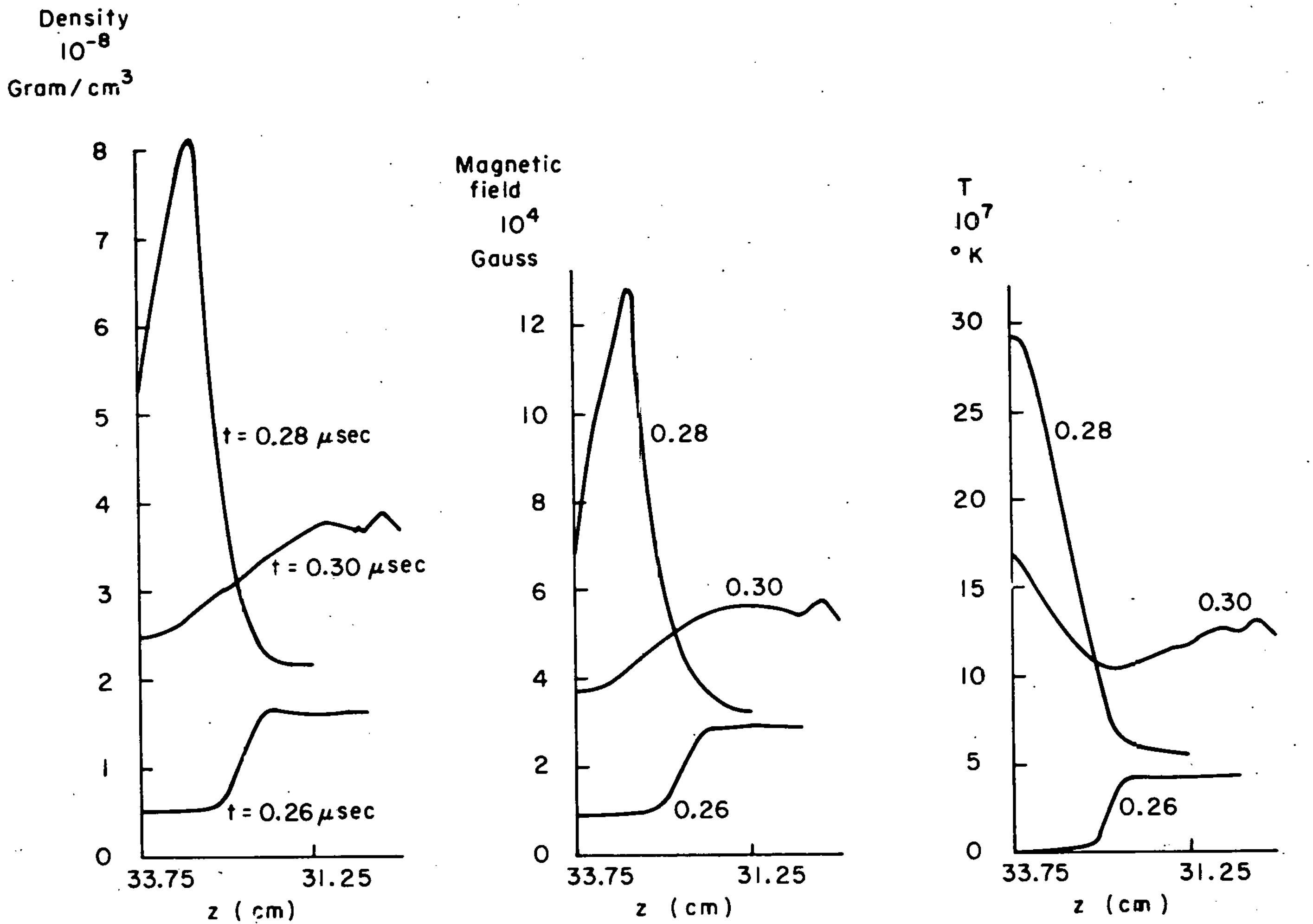



Fig. 7-2 Plots of the magnetic field, the density and the temperature with respect to the axial positions near the focused region with the radius r = 6.625 cm at different sequences. The incident shock speed is 125 cm/$\mu$sec.

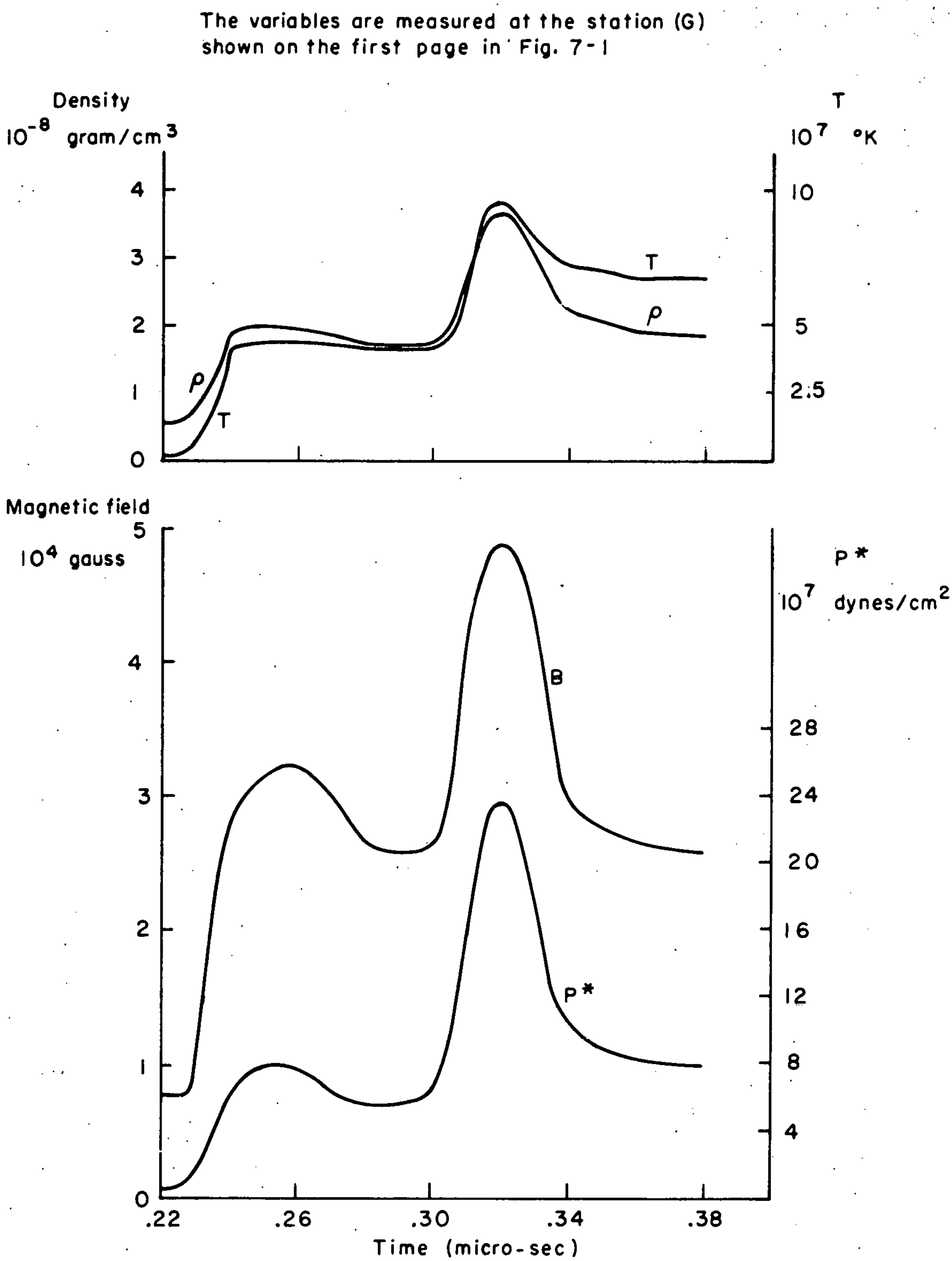



Fig. 7-3 The traces of the temperature, density, magnetic field and total pressure (P*) at a fixed position.

## 7.2 Summary

As we have seen, the end wall profile in a coaxial shock tube obviously plays an important rule in plasma heating. The convergence of the incident shock wave, the motions of fluid particles, and shock flows depend strongly on the wall profile. Generally speaking, the flat wall does not converge a plane shock; the semi-circular wall converges an incident shock to some degree; and the selected quarter-elliptic end wall makes an incident shock wave fully convergent. The maximum temperature and density of plasma achieved during shock focusing and reflection from the quarter-elliptic end wall, are much higher than those due to shock reflection from the semi-circular and flat end walls respectively. The maximum temperature and the corresponding density and magnetic field are listed below for the shock wave which has been reflected from three different wall shapes. They are measured at the focused locations of the quarter-elliptic ($r = 6.625$ cm), semi-circular ($r = 9.375$ cm), and flat($r = 11$ cm) end walls at the moment of maximum plasma heating.

| | flow variables measured in the focused region at the moment of maximum plasma heating | | |
|---|---|---|---|
| wall shape | maximum temp.($^{o}$K) | corresponding density (gram/cm$^3$) | corresponding magnetic field (KG) |
| elliptic | $29.0 \times 10^7$ | $5.30 \times 10^{-8}$ | 78.7 |
| circular | $16.7 \times 10^7$ | $5.36 \times 10^{-8}$ | 68 |
| flat | $14.5 \times 10^7$ | $5.1 \times 10^{-8}$ | 53 |

For the convenience to see the motions of the incident and reflected shock waves, Fig. 7-4 shows the whole sequence of the temperature contours for shock focusing and reflection from the selected quarter-elliptic end wall in the coaxial shock tube.

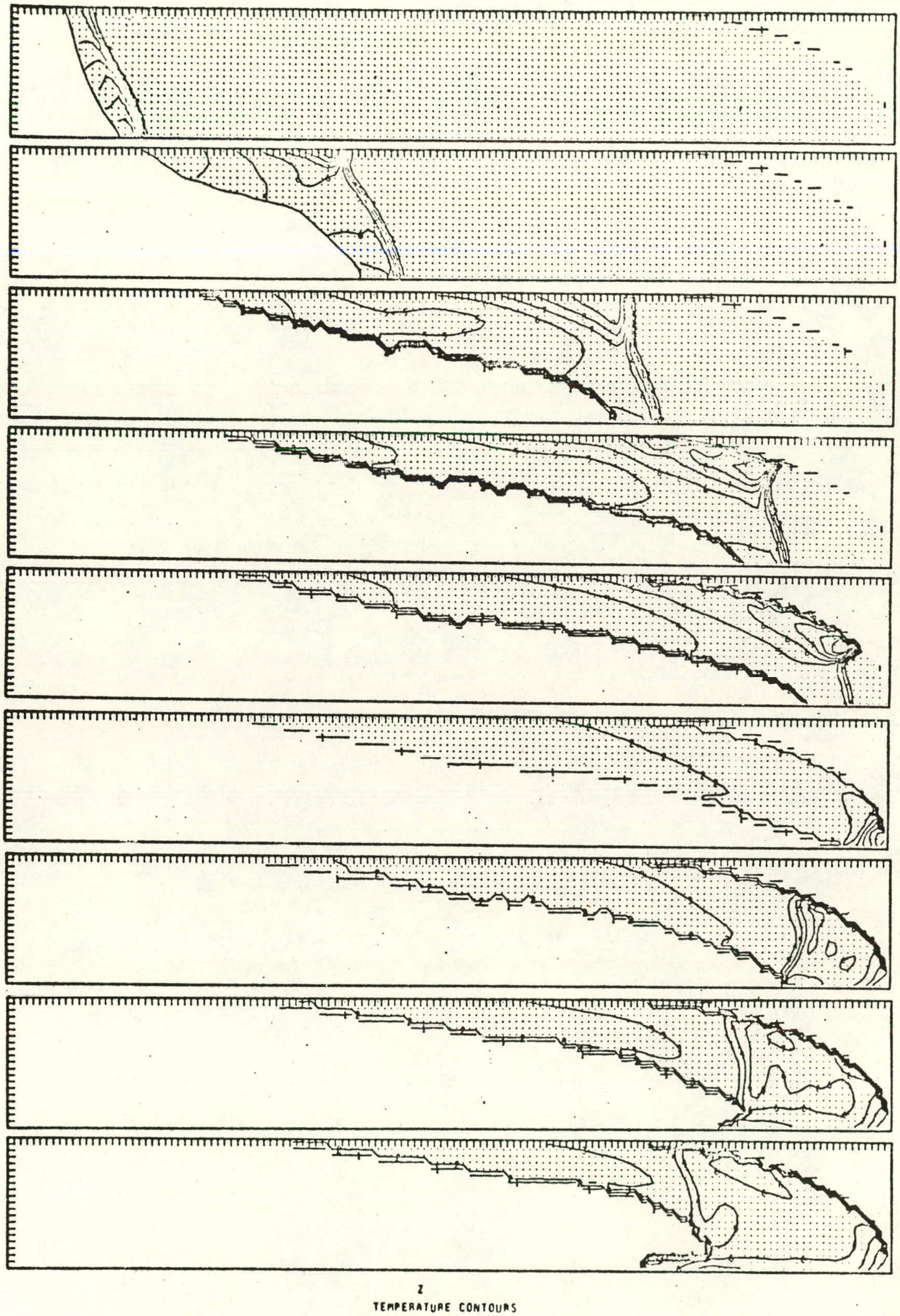



Fig. 7-4 Shock focusing and reflection from the quarter-elliptic end wall in the coaxial shock tube

# VIII. PLANE FLOWS

We did some preliminary work on plane flows to check the accuracy of the FLIC method. The code was applied to shock reflection in plane flows, and the results were compared with those obtained by the two-step Lax-Wendroff scheme[22], which was given in Appendix A.

The treatment of boundary conditions using irregular cells on the curved wall described in Chapter III is designed for the FLIC method only. For the Lax-Wendroff scheme, we use a simpler approximation, using only regular cells. This same approximation is also used for the FLIC method; and in plane flows, it does not give much difference in results from the more accurate method of Chapter III. More specifically, on a curved wall boundary, all surface cells and fictitious boundary cells are treated in the same way as the interior cells, in spite of the fact that they are split by the wall boundary. They are kept in the same shape as the interior regular cells. The values of the flow variables in a fictitious cell, which has at least one half of its area outside the wall boundary, are calculated by reflection with respect to the tangent of the wall profile. In this way, there are no flows of mass, momentum and energy across the wall boundary as the normal component of the fluid velocity on a fixed wall vanishes. This method makes it easy to handle boundary cells on a curved wall in plane flows. We call it a "regular shape" approximation.

The problem, with which we compare the two methods, is the reflection of a plane shock wave from a semi-circular wall in a transverse MHD flow. In Table 8-1, we give the required parameters for both schemes. The downstream conditions are obtained from the upstream conditions in front of the shock wave. The initial shock position of the incident shock wave is given to be at 3.2 cm in front of the center of the semi-circular wall as shown in Fig. 8-1. The radius of the wall curvature is 2.5 cm. Then, the computations of the shock flow can be made by using a continuous inflow boundary at the left end. Actually, reflection is applied to the center line because of the symmetry, so that only one half of the flow region is in the computations.

To select the right amount of numerical dissipation, for this set of tests as well as for Chapters IV, V and VII, we test the FLIC method on shock reflection from a flat wall using various values for the coefficient of the numerical dissipation, until it is large enough to dampen significantly the numerical oscillations. The coefficient (b) in the viscous pressure term, say $q^n_{i+\frac{1}{2},j}$ , is chosen to be 0.5. The detailed relation between b and $q^n_{i+\frac{1}{2},j}$ is given in Chapter II. Similar tests are carried out for the two-step Lax-Wendroff scheme, for which we need a coefficient (k=1.2) in a viscosity term which is shown in Appendix A. These values of the coefficients (b and k) are used in the computations of shock reflection from the semi-circular wall.

Fig. 8-2 shows the results obtained by the FLIC method for a shock wave reflecting from a flat wall using the initial conditions given in Table 8-1. The reflected shock wave is shown on the lower curves. The density profile is on the left, the temperature profile is on the right. The solid curves are a large cell result ($\delta x=\delta y=0.2$ cm), the dotted curves are a fine cell result ($\delta x=\delta y=0.1$ cm). The shock thickness in the fine cell result is narrower than that of the large cell result since the numerical diffusion is proportional to the cell size. An ordinary numerical scheme will spread out a step function into several cells. But the amount of inherent numerical dissipation in the large cell result is greater than that of the fine cell result. These are the common nature of the numerical schemes. However, these computed results agree well with the analytical solution.

The sequence of temperature contours obtained by the FLIC method for the plane shock wave reflecting from the semi-circular wall is shown in Fig. 8-3. The comparison of the computed results between the two methods in this problem is given in Fig. 8-4 and Table 8-2. Fig. 8-4 shows the temperature and density contours obtained from three different calculations at the same instant of time: a large grid FLIC method ($\delta x=\delta y=0.2$ cm), a fine grid FLIC method ($\delta x=\delta y=0.1$ cm), and a large grid two-step Lax-Wendroff scheme ($\delta x=\delta y=0.2$ cm). The agreement between the three results is good. In addition, we also tabulate the density and temperature measured at the position B (shown in Fig. 8-1) with respect to time under the two methods in Table 8-2. The average errors between them based on the FLIC result is about 2.34% in density and about 1.36% in temperature. This gives us some confidence on the basic soundness of the FLIC method for this class of problems.

Table 8-1

The required parameters and initial conditions for a plane shock wave reflecting from a semi-circular wall in plane flows

| | |
|---|---|
| Gas | hydrogen |
| Mach number | 1000 |
| Shock speed | 144 cm/$\mu$sec |
| Cell width ($\delta x=\delta y$) | 0.2 cm |

| | Upstream | Downstream |
|---|---|---|
| Density | $.5344 \times 10^{-8}$ gram/cm$^3$ | $1.86 \times 10^{-8}$ gram/cm$^3$ |
| Pressure | 66.65 dynes/cm$^2$ | $5.74 \times 10^{7}$ dynes/cm$^2$ |
| Temperature | 300 $^{o}$K | $7.43 \times 10^{7}$ $^{o}$K |
| Transverse field | 7000 Gauss | 24364.7 Gauss |
| Axial component of fluid velocity | 0 cm/$\mu$sec | 102.63 cm/$\mu$sec |

Table 8-2

The density and temperature measured at the position B (shown in Fig. 8-1) with respect to time under both schemes

| | FLIC ($\delta x=\delta y=0.2$ cm) | | Lax-Wendroff ($\delta x=\delta y=0.2$ cm) | |
|---|---|---|---|---|
| Time µsec | Density $10^{-8}$ gram/cm$^3$ | Temp. $10^8$ $^oK$ | Density $10^{-8}$ gram/cm$^3$ | Temp. $10^8$ $^oK$ |
| 0 | .53 | 0 | .53 | 0 |
| .0155 | 1.93 | .78 | 2.03 | .81 |
| .0255 | 1.83 | .70 | 1.87 | .72 |
| .0355 | 5.04 | 2.33 | 4.67 | 2.44 |
| .0455 | 4.18 | 2.00 | 3.91 | 2.07 |
| .0555 | 3.97 | 1.89 | 3.92 | 1.88 |
| .0655 | 3.95 | 1.87 | 4.01 | 1.78 |

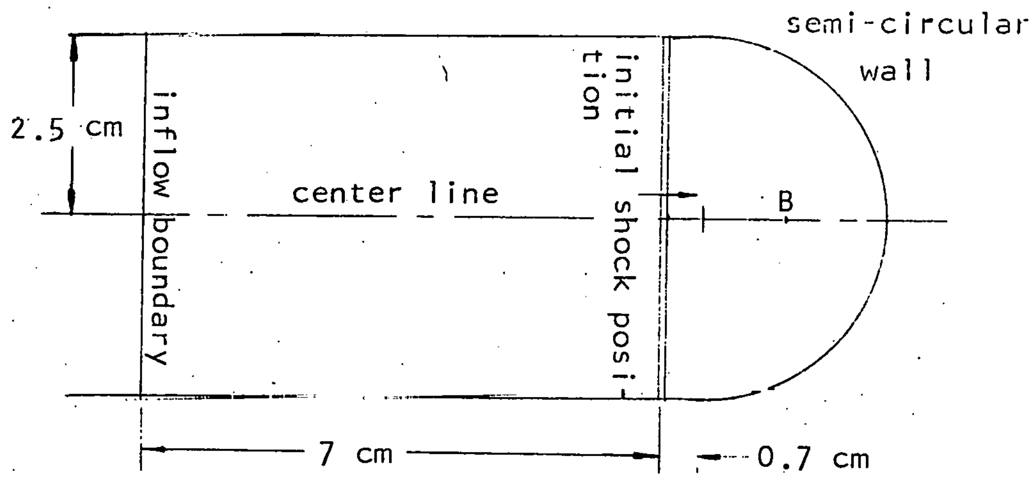



Fig. 8-1 The initial shock position in the flow region

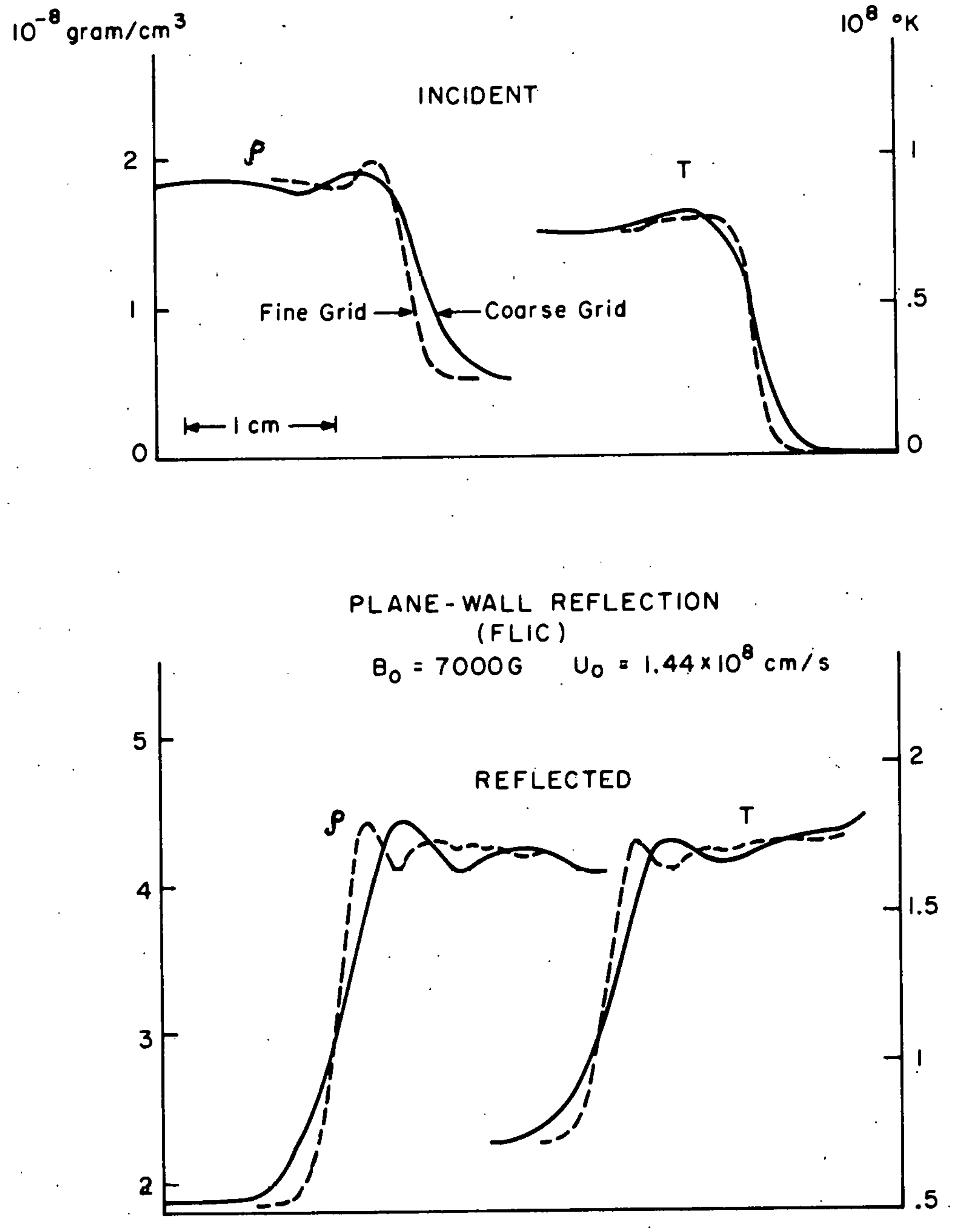



Fig. 8-2 The computer results of an incident shock wave reflecting from a flat wall in a plane flow.

A-907- -2730 W.C.

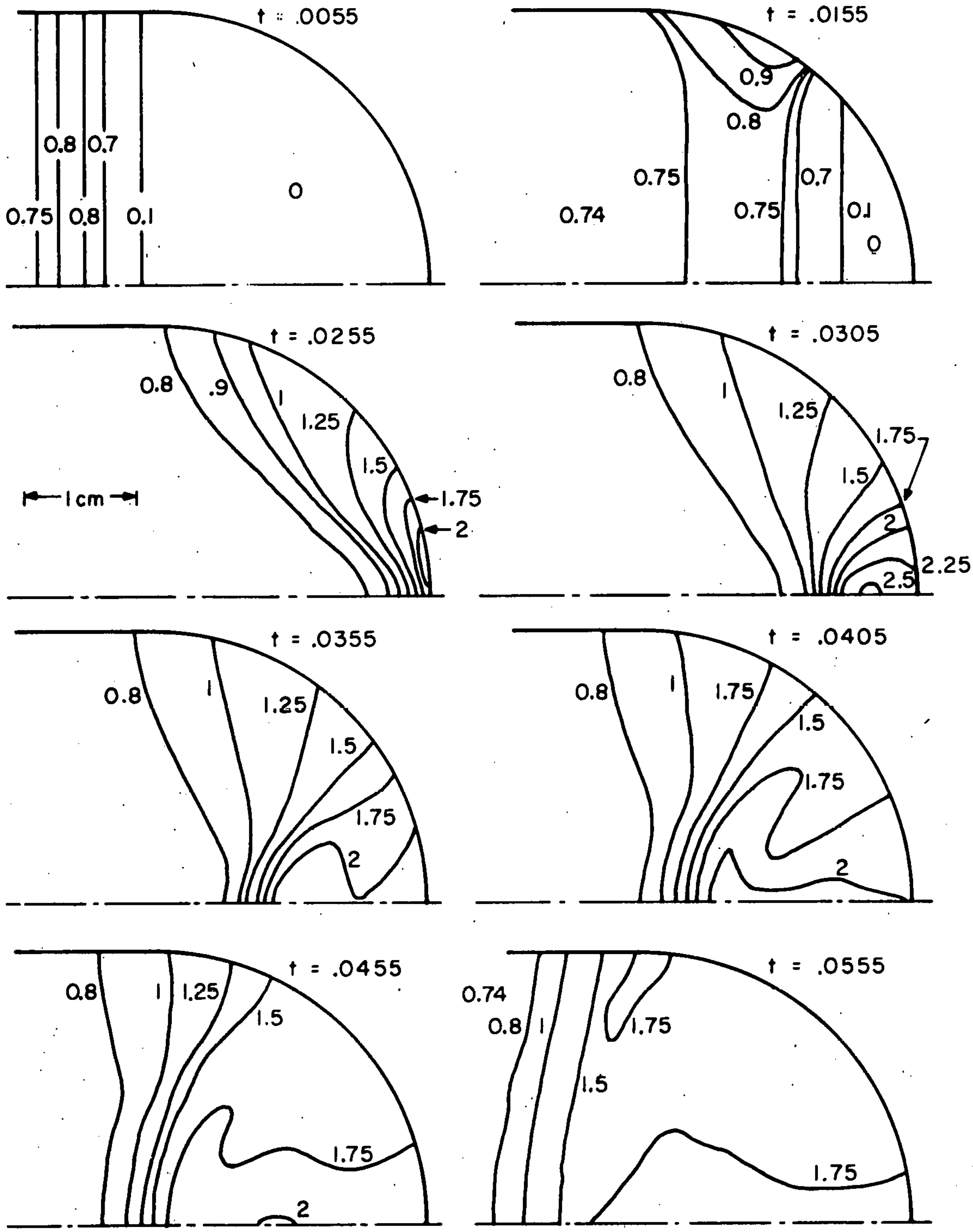



Fig. 8-3 The sequence of temperature contours for a plane shock wave reflecting from a semi-circular wall in a plane flow, with $B_0$ = 7000G, $U_s$ = $1.44 \times 10^8$ cm/sec, t in $\mu$sec, T in $10^8$ °K

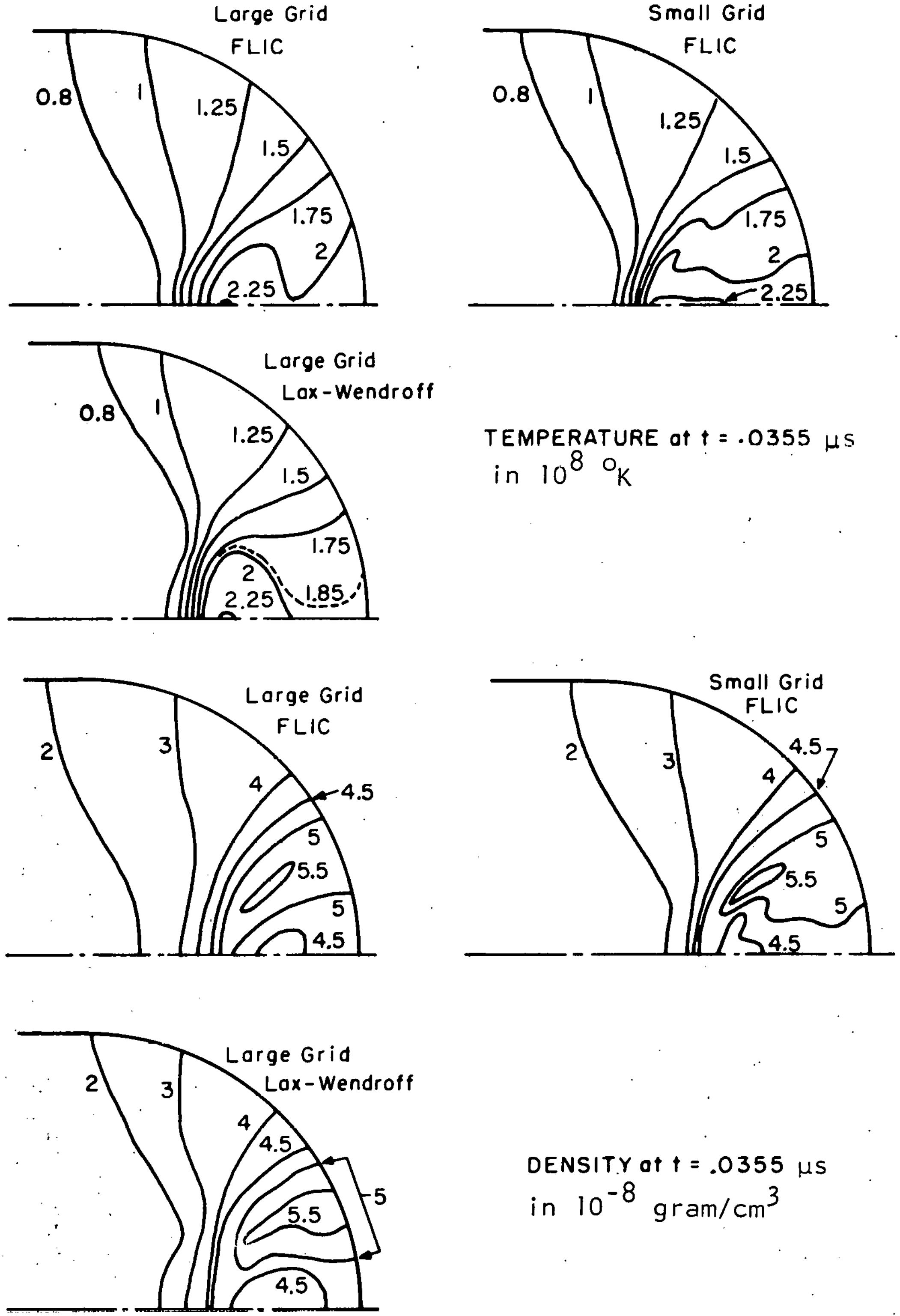



Fig. 8-4 The temperature and density contours obtained from three different calculations for the same instant of time.

W.L.

A-907-ME-2

REFERENCES

1. Meehan, R. J., Ph. D. Thesis, Report No. 44, Plasma Lab., Columbia Univ., (1969).

2. Halmoy, E., Phys. Fluids, 14, 2134, (1971).

3. Moriette, P. R., Phys. Fluids, 15, 51, (1972).

4. Gross, R. A., Chen, Y. G., Halmoy, E., and Moriette, P. R., Phys. Rev. Letters, 25, 575, (1970).

5. Schneider, S. H., Chu, C. K., and Leonard, B. P., Phys. Fluids, 14, 1103, (1971).

6. Schneider, S. H., Phys. Fluids, 15, 805, (1972).

7. Hain, K., Hain, G., Roberts, K. V., Roberts, S. J., and Koppendorfer, W., Z. Naturforsch, 15a, 1039, (1960).

8. Alder, B., Methods in Computational Physics, Vol. 3, Academic Press, (1964).

9. Chu, C. K., Computational Fluid Dynamics, AIAA Selected Reprint Series.

10. Potter, D. E., Phys. Fluids, 14, 1971, (1971).

11. Burstein, S. Z., AIAA Journal, 2, 2111, (1964).

12. Lapidus, A., J. Computational Phys., 2, 154, (1967).

13. Amsden, A. A., Los Alamos Rep. La-3466, (1966).

14. Marder, B. M., Los Alamos Rep. LA-DC 13103, (1971).

15. Gentry, R. A., Martin, R. E., and Daly, B. J., J. Computational Phys., 1, 87, (1966).

16. Butler, T. D., Phys. Fluids, 10, 1205, (1967).

17. Whitham, G. B., J. Fluid Mech., 2, 145, (1957).

18. Whitham, G. B., J. Fluid Mech., 5, 369, (1959).

19. Milton, B. E., and Archer, R. D., AIAA Journal, 7, 779, (1969).
Milton, B. E., and Archer, R. D., 8th International Symposium, London, (1971).

20. Lau, J., CASI Transaction, 4, 13, (1971).

21. Skews, B. W., AIAA Journal, 10, 839, (1972).

22. Burstein, S. Z., J. Computational Phys., 2, 198, (1967).

23. Chu, C. K., and Harold Grad, in Research Frontiers in Fluid Dynamics, (R. Seeger and G. Temple, eds.), Wiley-Interscience, pp. 308-349, (1964).

24. Jeffrey, A., and Taniuti, T., Non-Linear Wave Propagation, Academic Press, p. 159, (1964).

25. Whitham, G. B., J. Fluid Mech., 2, 185, (1958).

26. Payne, R. B., J. Fluid Mech., 2, 185, (1957).

27. Lapidus, A., J. Computational Phys., 8, 106, (1971).

28. Private communication.

29. Richtmyer, R. D., and Morton, K. W., Difference Methods for Initial-Value Problems, second edition, Interscience Publishers, pp. 365-368, (1967).

# APPENDIX A

## The Two-Step Lax-Wendroff Scheme

The basic equations of a two-dimensional transverse MHD flow in Cartesian coordinates (x,y) can be expressed as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(m) + \frac{\partial}{\partial y}(n) = 0 , \qquad \text{(A-1)}$$

$$\frac{\partial m}{\partial t} + \frac{\partial}{\partial x}\left(\frac{m^2}{\rho} + P^*\right) + \frac{\partial}{\partial y}\left(\frac{mn}{\rho}\right) = 0 , \qquad \text{(A-2)}$$

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x}\left(\frac{mn}{\rho}\right) + \frac{\partial}{\partial y}\left(\frac{n^2}{\rho} + P^*\right) = 0 , \qquad \text{(A-3)}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}\left(\frac{m}{\rho}(E + P^*)\right) + \frac{\partial}{\partial y}\left(\frac{n}{\rho}(E + P^*)\right) = 0 , \qquad \text{(A-4)}$$

$$\frac{\partial B}{\partial t} + \frac{\partial}{\partial x}\left(\frac{mB}{\rho}\right) + \frac{\partial}{\partial y}\left(\frac{nB}{\rho}\right) = 0 . \qquad \text{(A-5)}$$

Here,

$$m = \rho u,$$

$$n = \rho v,$$

$$\begin{aligned} P^* &= P + B^2/8\pi , \\ &= (\gamma-1)\,(E - \tfrac{1}{2}(m^2 + n^2)/\rho) \\ &\quad + (2-\gamma)\,B^2/8\pi , \end{aligned}$$

$$P = (\gamma-1)\,\rho\, I ,$$

$$E = \rho\,(I + \tfrac{1}{2}(u^2 + v^2)) + \frac{B^2}{8\pi} ,$$

where $\rho$ , $u$ , $v$ , $P$ , $B$ and $I$ are the density, x-component of velocity, y-component of velocity, thermal pressure, magnetic field and specific internal energy respectively.

These equations fit in the following form,

$$\frac{\partial w}{\partial t} = \frac{\partial}{\partial x}[f] + \frac{\partial}{\partial y}[g] . \tag{A-6}$$

The difference schemes[22] for Eq. (A-6) consists of two steps as follows:

Step 1: $$w_{h,l}(t+\delta t) = \tfrac{1}{4}(w_{h+\frac{1}{2},l+\frac{1}{2}} + w_{h+\frac{1}{2},l-\frac{1}{2}} + w_{h-\frac{1}{2},l+\frac{1}{2}} + w_{h-\frac{1}{2},l-\frac{1}{2}}) + \frac{\lambda}{2}(f_{h+\frac{1}{2},l+\frac{1}{2}} - f_{h-\frac{1}{2},l+\frac{1}{2}} + f_{h+\frac{1}{2},l-\frac{1}{2}} - f_{h-\frac{1}{2},l-\frac{1}{2}}) + \frac{\lambda}{2}(g_{h+\frac{1}{2},l+\frac{1}{2}} - g_{h+\frac{1}{2},l-\frac{1}{2}} + g_{h-\frac{1}{2},l+\frac{1}{2}} - g_{h-\frac{1}{2},l-\frac{1}{2}}) ,$$

with $\delta x = \delta y$ , $\lambda = \delta t/\delta x$ ,

$h = i \pm \frac{1}{2}$ , $l = j \pm \frac{1}{2}$ .

Step 2: $$w_{i,j}(t+\delta t) = w_{i,j}(t) + \frac{\lambda}{4}\left(f_{i+1,j} - f_{i-1,j} + \underline{f}_x^{t+\delta t}\right) + \frac{\lambda}{4}\left(g_{i,j+1} - g_{i,j-1} + \underline{g}_y^{t+\delta t}\right) + q^t_{i,j} ,$$

$$\underline{f}_x^{t+\delta t} = f^{t+\delta t}_{i+\frac{1}{2},j+\frac{1}{2}} - f^{t+\delta t}_{i-\frac{1}{2},j+\frac{1}{2}} + f^{t+\delta t}_{i+\frac{1}{2},j-\frac{1}{2}} - f^{t+\delta t}_{i-\frac{1}{2},j-\frac{1}{2}} ,$$

$$\underline{g}_y^{t+\delta t} = g^{t+\delta t}_{i+\frac{1}{2},j+\frac{1}{2}} - g^{t+\delta t}_{i+\frac{1}{2},j-\frac{1}{2}} + g^{t+\delta t}_{i-\frac{1}{2},j+\frac{1}{2}} - g^{t+\delta t}_{i-\frac{1}{2},j-\frac{1}{2}} ,$$

$$q^t_{i,j} = \frac{0.5}{4}\lambda \cdot k \cdot \left(\left|u^t_{i+1,j} - u^t_{i,j}\right| \times \left(w^t_{i+1,j} - w^t_{i,j}\right)\right)$$

$$-|u^t_{i,j} - u^t_{i-1,j}|\times\left(w^t_{i,j} - w^t_{i-1,j}\right)$$

$$+|v^t_{i,j+1} - v^t_{i,j}|\times\left(w^t_{i,j+1} - w^t_{i,j}\right)$$

$$-|v^t_{i,j} - v^t_{i,j-1}|\times\left(w^t_{i,j} - w^t_{i,j-1}\right)\Big) ,$$

with

$$k = \frac{1.2}{2}\left[1 + \exp\left(1 - \frac{\sqrt{u^2 + v^2}}{U_2}\right)\right] \text{ if } \sqrt{u^2 + v^2}\,/U_2 < 1 ;$$

$$= 1.2 \text{ otherwise.}$$

where $q^t_{i,j}$ is the viscosity term[29], $U_2$ is the speed of fluid behind an incoming shock wave.

# APPENDIX B

The computer program for shock focusing and reflection from the selected quarter-elliptic end wall in a coaxial shock tube with a constant drive current

```
//*SETUP DEVICE=2400-9,ID=(MI0015,RING),VERIFY=MI0452
//*SETUP UNIT=2400-7,ID=(CSJ032,RING,,NL)
// EXEC FORTGCG
//FORT.SYSIN DD *
C     PROGRAM FOR SHOCK WAVE GENERATION AND REFLECTION IN AXISYMMETRY
C   FOR USE ONLY WHEN VACUUM-CARPET INTERFACE DOESNT DISTORT.
      DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
     1FF(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),RB(2,137,22),
     2GU(1,137,22),GV(1,137,22),GI(1,137,22),GF(1,137,22),TMP(2,137,22),
     3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPP(22),BV(137,22),
     4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
     5ZSX(137,22)
      COMMON DR,FU,FV,FI,FF,TC,PP,P,RB,GU,GV,GI,GF,TMP,SYA,SXB,SYB,VOL,
     1SI,CPP,BV,CSX,DSX,CSY,DSY,RSY, ZSX,Q1,Q2,Q3,Q4,M,I,J
      DIMENSION F(137,22) ,Z(200)
      N=1
      IS=3
      IB=3
      JB=2
      IE=5
      JE=21
      JM=1
      JM1=2
      IW=136
      IC=134
      DX=.25
      CK=1.
      M=0
      TIME=0.
      IM=9
      IM1=33
      JC=9
      RA=5.0
      EPSI=1.5E-30
      FR=.4157E+08
      DDT=.1E-12
      DT=.4E-09
      FK=1.66666
      ID=IW+1
      JD=JE+1
      RMEAN=(5.25*11.25)**.5
      CV=FR/(FK-1.)
      DY=DX
      IMMM=1
      DO 2 J=1,JD
      FJ=J
      SI(J)=IS
      SXB(J)=2.*3.1416*(6.25+(FJ-1.5)*DX)*DX
      SYA(J)=2.*3.1416*(6.25+(FJ-1.)*DX)*DX
      SYB(J)=SXB(J)
      XC=2.*3.1416*DX*DX/SXB(J)
      CPP(J)=(4.-7.*XC**2*XC**4 )/(4.*(1.-XC**2)**2)
      DO 2 I=IMMM,ID
      DSY(I,J)=DY
      DSX(I,J)=DX
      VOL(I,J)=SXB(J)*DX
      CSX(I,J)=DX
      CSY(I,J)=DY
      RSY(I,J)=DX
      ZSX(I,J)=DX
```



```
      DR(1,I,J)=.5344E-02
      BV(I,J)=0.
      FU(1,I,J)=0.
      FV(1,I,J)=0.
      FI(1,I,J)=.187E+11
      P (1,I,J)=66.65
      GU(1,I,J)=0.
      GV(1,I,J)=0.
      GF(1,I,J)=FI(1,I,J)
      FF(1,I,J)=FI(1,I,J)
      TC(1,I,J)=.144E6
      RB(1,I,J)=7000.*RMEAN/(SXB(J)/(2.*3.1416*DX))
      PP(1,I,J)=65.5*RB(1,I,J)**2/(8.*3.1416)
      TMP(1,I,J)=300.
    2 CONTINUE
      DO 3 I=1,2
      DO 3 J=1,JD
      CALL VACUUM(RMEAN,DX)
    3 CONTINUE
      DO 5 J=2,JD
      DO 5 I=IM1,ID
      CALL DISTAN(I,J,DA,CB,DEL,DX)
      IF (DEL .GT. .21*DX) CALL VACUUM(RMEAN,DX)
      IF (DEL .GT. .21*DX) BV(I,J)=0.
    5 CONTINUE
      CALL CIRCLE(DX,IM1,IC,JC,JM,RA,RMEAN)
      CALL MODESG(Z,0,'*          ','*','H.LUI',8,'ELLIPSES')   *project_number
    9 M=M+1
      DO 47 J=JB,JE
      DO 47 I=IS,IE
      IF (ZSX(I,J).EQ.0..AND.RSY(I,J).EQ.0.) GO TO 47
      IF (DR(1,I-1,J).EQ.0..AND.DR(1,I+1,J).EQ.0.) GO TO 488
      IF (DR(1,I-1,J).GT.0..AND.DR(1,I+1,J).GT.0..AND.J.NE.JE) GO TO 488
      IF (ZSX(I,J).LE. .6*DX) GO TO 61
      IF (VOL(I,J) .LE. VOL(2,J)/2.0) GO TO 61
      GO TO 483
   61 CALL        RIGHT(DY,FK,FR,CV,IW,RMEAN)
      IF (DR(1,I,J) .EQ. 0.) GO TO 431
  488 CONTINUE
      IF (DR(1,I,J-1).EQ.0..AND.DR(1,I,J+1)).EQ.0.) GO TO 481
      IF (DR(1,I,J-1).GT.0..AND.DR(1,I,J+1).GT.0.) GO TO 481
      IF (RSY(I,J) .LE. .6*DY) GO TO 51
      IF (VOL(I,J) .LE. VOL(2,J)/2.0) GO TO 51
      GO TO 481
   51 CALL        UPPER(DX FK,FR,CV,JB,JE,RMEAN)
  481 CONTINUE
   47 CONTINUE
      CALL        DIVIDE(DX,DY,FK,FR,CV,JB,JE,IS,IE,IM,JM,IC)
      DO 46 J=JB,JE
      IE=MAX0(SI(J),SI(JE))
      IF (FU(1,IE,J) .GT.   1.0) IE=IE+1
      IF (IE.GE.IW) IE=IW
   46 CONTINUE
      IS=IB-1
      DO 12 J=JB,JE
      DO 12 I=IS,IE
```

```
      IF (M2(1,I,J) .EQ. 0.) GO TO  12
      IF (ABS(FU(1,I,J)) .LE. EPSI) FU(1,I,J)=0.
      IF (ABS(FV(1,I,J)) .LE. EPSI) GO TO 14
      FVV1=FV(1,I,J)**2
      GO TO 15
   14 FVV1=0.
   15 UABS=(FU(1,I,J)**2+FVV1)**.5
   16 IF (DT*UABS-.3*DX) 17,19,18
   19 DT=DT-DDT
      GO TO 16
   17 RT=DT/DX
   12 CONTINUE
      IF (DT .LE. 0.) GO TO 226
      IF (MOD(M,50) .EQ. 1) GO TO 27
      [illegible]
      IF (M .GE. 552) GO TO 31
      GO TO 28
   27 N=1
      FMIN=0.
      FMAX=300.
      NX=ID
      NY=JD
      XMIN=2.
      XMAX=ID
      VMIN=2.
      VMAX=JD
      NCONT=10
      ITICK=1
      ICROSS=0
      IDEBUG=0
      DO 347 J=2,JD
      DO 347 I=1,ID
      F(I,J)=TMP(1,I,J)
      FMAX=AMAX1(F(I,J),FMAX)
      IF ( I .GT. IW .OR. J .GT. JE) RSY(I,J)=0.
  347 CONTINUE
      CALL CNTLM(Z,F,NX,NY,XMIN,XMAX,VMIN,VMAX,FMAX,FMIN,NCONT,ITICK,
     1ICROSS,'Z          ','R          ','TEMPERATURE CONTOURS',IDEBUG,DR,
     2RSY,FU,FV,TIME,M)
      TMASS=0.
      HMASS=0.
      DO 33 J=JB,JE
      DO 33 I=IS,IE
      IF (RSY(I,J).EQ.0..AND.ZSX(I,J).EQ.0..AND.M.GT.2) GO TO 33
      TMASS=TMASS+DR(1,I,J)*VCL(I,J)
      HMASS=HMASS+BB(1,I,J)*VCL(I,J)*2.*3.1416*DX/SXB(J)
      WRITE (6,32) I,J,FU(N,I,J),FV(N,I,J),DR(N,I,J),FE(N,I,J),
     1 BB(N,I,J),TMP(N,I,J),PP(N,I,J)
   32 FORMAT (1H ,'I=',I3,'J=',I3,'U=',E14.8,'V=',E15.8,'DR=',E14.8,
     1'FE=',E14.8,'BB=',E14.8,'T=',E14.8,'PP=',F14.8)
   33 CONTINUE
      WRITE (6,634) TMASS,HMASS
  634 FORMAT (1H ,'MASS=',E15.8,'HMASS=',E15.8)
  226 WRITE (6,34) M,TIME,DT ,IP
   34 FORMAT (1H ,'M=',I4,'TIME=',E14.8,'DT=',F14.8,I12)
      IF (DT .LE. 0.) GO TO 31
```



```
      CALL       PLOTS(IB,IE,JB,JE,DX,DT,TIME,N,IW,JD,ID)
   28 CONTINUE
      DO 9 I=IB,IE
      DSX(I,JB-1)=DSX(I,JB)
      CSX(I,JB-1)=DSX(I,JB)
      ZSX(I,JB-1)=DSX(I,JB)
      RSY(I,JB-1)=0.
      CSY(I,JB-1)=0.
      DSY(I,JB-1)=0.
      DR(1,I,JB-1)= DR(1,I,JB)
      FU(1,I,JB-1)= FU(1,I,JB)
      FV(1,I,JB-1)=-FV(1,I,JB)
      FI(1,I,JB-1)= FI(1,I,JB)
      BB(1,I,JB-1)= BB(1,I,JB)*SXB(JB)/SXB(JB-1)
       P(1,I,JB-1)=  P(1,I,JB)
      PP(1,I,JB-1)=P (1,I,JB-1)+BB(1,I,JB-1)**2/(8.*3.1416)
      TC(1,I,JB-1)= TC(1,I,JB)
      DSX(I,JE+1)=CSX(I,JE)
      ZSX(I,JE+1)=CSX(I,JE)
      CSX(I,JE+1)=CSX(I,JE)
      RSY(I,JE+1)=0.
      CSY(I,JE+1)=0.
      DSY(I,JE+1)=0.
      DR(1,I,JE+1)= DR(1,I,JE)
      FU(1,I,JE+1)= FU(1,I,JE)
      FV(1,I,JE+1)=-FV(1,I,JE)
      FI(1,I,JE+1)= FI(1,I,JE)
      BB(1,I,JE+1)= BB(1,I,JE)*SXB(JE)/SXB(JE+1)
       P(1,I,JE+1)=  P(1,I,JE)
      PP(1,I,JE+1)=P (1,I,JE+1)+BB(1,I,JE+1)**2/(8.*3.1416)
      TC(1,I,JE+1)= TC(1,I,JE)
    9 CONTINUE
      IF (IE.GT.IM) CALL CIRCLE(DX,IM1,IC,JC,JM,RA,RMEAN)
      MP=1
      DO 21 J=JB,JE
      DO 21 I=IS,IE
      IF (ZSX(I,J) .EQ. 0. .AND. RSY(I,J) .EQ. 0.) GO TO 21
      CB=.5
      IF (J .EQ. JE .OR. J .EQ. JB) CB=0.18
      IF (I .LE. IN) GO TO 2222
      CALL DISTAN(I,J,CA,CB,DEL,DX)
      IF (ABS(DEL).LE.1.20*DX) CB=0.18
 2222 CONTINUE
      CALL       FRAC(SFIJ,AXIJ,AYIJ,      DX,DY)
      CALL       VISCOS(CB,CK,MP,FUR,FUL,GUR,GUL,FVT,FVB,GVT,GVB,DT,DX,
     1IM,JM)
      TPP=PP(1,I+1,J)
      TPL=PP(1,I-1,J)
      TPU=PP(1,I,J+1)
      TPB=PP(1,I,J-1)
      IF ((RSY(I,J).EQ.RSY(I+1,J)).AND.(RSY(I,J).EQ.RSY(I-1,J)).AND.
     1(ZSX(I,J).EQ.ZSX(I,J+1)).AND.(ZSX(I,J).EQ.ZSX(I,J-1))) GO TO 6
      IF ( RSY(I+1,J) .GT. 0. .AND. DR(1,I+1,J-1) .EQ. 0. .AND. RSY(I,J)
     1.GT. RSY(I+1,J)) TPR=(PP(1,I+1,J)*RSY(I+1,J)+PP(1,2,J)*(RSY(I,J)
     2-RSY(I+1,J)))/RSY(I,J)
      IF ( DSY(I+1,J) .GT. 0. .AND. PP(1,I+1,J+1) .EQ. 0. .AND. RSY(I,J)
```

3

```
              1.GT. DSY(I+1,J)) TPR=(PP(1,I+1,J)*DSY(I+1,J)+PP(1,2,J)*(RSY(I,J)
              2-DSY(I+1,J)))/RSY(I,J)
 0209         IF (DR(1,I+1,J) .EQ. 0. .AND. DSY(I+1,J+1) .GT. CSY(I,J+1) .AND.
              1DSY(I+1,J-1) .GT. CSY(I,J-1)) TPR=(PP(1,I+1,J+1)*(DSY(I+1,J+1)
              2-CSY(I,J+1))+PP(1,I+1,J-1)*(DSY(I+1,J-1)-CSY(I,J-1))+PP(1,2, J)*
              3(RSY(I,J)+CSY(I,J+1)+CSY(I,J-1)-DSY(I+1,J+1)-DSY(I+1,J-1)))/RSY(I
              4,J)
 0210         IF (DR(1,I+1,J) .EQ. 0. .AND. DSY(I+1,J+1) .GT. CSY(I,J+1) .AND.
              1DSY(I+1,J-1) .LE. CSY(I,J-1)) TPR=(PP(1,I+1,J+1)*(DSY(I+1,J+1)-
              2CSY(I,J+1))+PP(1,2,J)*(RSY(I,J)+CSY(I,J+1)-DSY(I+1,J+1)))/RSY(I,J)
 0211         IF (DR(1,I+1,J) .EQ. 0. .AND. DSY(I+1,J+1) .LE. CSY(I,J+1) .AND.
              1DSY(I+1,J-1) .GT. CSY(I,J-1)) TPR=(PP(1,I+1,J-1)*(DSY(I+1,J-1)-
              2CSY(I,J-1))+PP(1,2,J)*(RSY(I,J)+CSY(I,J-1)-DSY(I+1,J-1)))/RSY(I,J)
 0212         IF ( CSY(I-1,J) .GT. 0. .AND. DR(1,I-1,J-1) .EQ. 0. .AND. RSY(I,J)
              1.GT. CSY(I-1,J)) TPL=(PP(1,I-1,J)*CSY(I-1,J)+PP(1,2,J)*(RSY(I,J)
              2-CSY(I-1,J)))/RSY(I,J)
 0213         IF ( CSY(I-1,J) .GT. 0. .AND. DR(1,I-1,J+1) .EQ. 0. .AND. RSY(I,J)
              1.GT. CSY(I-1,J)) TPL=(PP(1,I-1,J)*CSY(I-1,J)+PP(1,2,J)*(RSY(I,J)
              2-CSY(I-1,J)))/RSY(I,J)
 0214         IF (DR(1,I-1,J) .EQ. 0. .AND. CSY(I-1,J+1) .GT. DSY(I,J+1) .AND.
              1CSY(I-1,J-1) .GT. DSY(I,J-1)) TPL=(PP(1,I-1,J+1)*(CSY(I-1,J+1)
              2-DSY(I,J+1))+PP(1,I-1,J-1)*(CSY(I-1,J-1)-DSY(I,J-1))+PP(1,2, J)*
              3(RSY(I,J)+DSY(I,J+1)+DSY(I,J-1)-CSY(I-1,J+1)-CSY(I-1,J-1)))/RSY(I
              4,J)
 0215         IF (DR(1,I-1,J) .EQ. 0. .AND. CSY(I-1,J+1) .GT. DSY(I,J+1) .AND.
              1CSY(I-1,J-1) .LE. DSY(I,J-1)) TPL=(PP(1,I-1,J+1)*(CSY(I-1,J+1)-
              2DSY(I,J+1))+PP(1,2,J)*(RSY(I,J)+DSY(I,J+1)-CSY(I-1,J+1)))/RSY(I,J)
 0216         IF (DR(1,I-1,J) .EQ. 0. .AND. CSY(I-1,J+1) .LE. DSY(I,J+1) .AND.
              1CSY(I-1,J-1) .GT. DSY(I,J-1)) TPL=(PP(1,I-1,J-1)*(CSY(I-1,J-1)-
              2DSY(I,J-1))+PP(1,2,J)*(RSY(I,J)+DSY(I,J-1)-CSY(I-1,J-1)))/RSY(I,J)
 0217         IF ( DSX(I,J+1).GT.0..AND.DR(1,I-1,J+1).EQ.0..AND.ZSX(I,J).GT.
              1DSX(I,J+1)) TPU=(PP(1,I,J+1)*DSX(I,J+1)+PP(1,2,J+1)*(ZSX(I,J)-
              2DSX(I,J+1)))/ZSX(I,J)
 0218         IF ( DSX(I,J+1) .GT.0..AND.DR(1,I+1,J+1).EQ.0..AND.ZSX(I,J).GT.
              1DSX(I,J+1)) TPU=(PP(1,I,J+1)*DSX(I,J+1)+PP(1,2,J+1)*(ZSX(I,J)-
              2DSX(I,J+1)))/ZSX(I,J)
 0219         IF (DR(1,I,J+1).EQ.0..AND.DSX(I+1,J+1).GT.CSX(I+1,J).AND.DSX(I-1,J
              1+1).GT.CSX(I-1,J)) TPU=(PP(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))
              2+PP(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+PP(1,2,J+1)*(ZSX(I,J)+
              3CSX(I+1,J)+CSX(I-1,J)-DSX(I+1,J+1)-DSX(I-1,J+1)))/ZSX(I,J)
 0220         IF (DR(1,I,J+1).EQ.0..AND.DSX(I+1,J+1).GT.CSX(I+1,J).AND.
              1DSX(I-1,J+1) .LE.CSX(I-1,J)) TPU=(PP(1,I+1,J+1)*(DSX(I+1,J+1)-
              2CSX(I+1,J))+PP(1,2,J+1)*(ZSX(I,J)+CSX(I+1,J)-DSX(I+1,J+1)))/ZSX
              3(I,J)
 0221         IF (DR(1,I,J+1) .EQ.0..AND.DSX(I+1,J+1).LE.CSX(I+1,J).AND.
              1DSX(I-1,J+1) .GT.CSX(I-1,J)) TPU=(PP(1,I-1,J+1)*(DSX(I-1,J+1)-
              2CSX(I-1,J))+PP(1,2,J+1)*(ZSX(I,J)+CSX(I-1,J)-DSX(I-1,J+1)))/
              3ZSX(I,J)
 0222         IF ( CSX(I,J-1).GT.0..AND.DR(1,I-1,J-1).EQ.0..AND.ZSX(I,J).GT.
              1CSX(I,J-1)) TPB=(PP(1,I,J-1)*CSX(I,J-1)+PP(1,2,J-1)*(ZSX(I,J)-
              2CSX(I,J-1)))/ZSX(I,J)
 0223         IF ( CSX(I,J-1) .GT.0..AND.DR(1,I+1,J-1).EQ.0..AND.ZSX(I,J).GT.
              1CSX(I,J-1)) TPB=(PP(1,I,J-1)*CSX(I,J-1)+PP(1,2,J-1)*(ZSX(I,J)-
              2CSX(I,J-1)))/ZSX(I,J)
 0224         IF (DR(1,I,J-1).EQ.0..AND.CSX(I+1,J-1).GT.DSX(I+1,J).AND.CSX(I-1,J
              1-1).GT.DSX(I-1,J)) TPB=(PP(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))
```



```
              2+PP(1,I-1,J-1)*(CSX(I-1,J-1)-DSX(I-1,J))+PP(1,2,J-1)*(ZSX(I,J)+
              3DSX(I+1,J)+DSX(I-1,J)-CSX(I+1,J-1)-CSX(I-1,J-1)))/ZSX(I,J)
 0225         IF (DR(1,I,J-1).EQ.0..AND.CSX(I+1,J-1).GT.DSX(I+1,J).AND.
              1CSX(I-1,J-1) .LE.DSX(I-1,J)) TPB=(PP(1,I+1,J-1)*(CSX(I+1,J-1)-
              2DSX(I+1,J))+PP(1,2,J-1)*(ZSX(I,J)+DSX(I+1,J)-CSX(I+1,J-1)))/ZSX
              3(I,J)
 0226         IF (DR(1,I,J-1) .EQ.0..AND.CSX(I+1,J-1).LE.DSX(I+1,J).AND.
              1CSX(I-1,J-1) .GT.DSX(I-1,J)) TPB=(PP(1,I-1,J-1)*(CSX(I-1,J-1)-
              2DSX(I-1,J))+PP(1,2,J-1)*(ZSX(I,J)+DSX(I-1,J)-CSX(I-1,J-1)))/
              3ZSX(I,J)
 0227         IF (RSY(I,J).GT.DSY(I+1,J) .AND. DSY(I+1,J).GT.0..AND.DR(1,I+1,
              1J-1).GT.0..AND.DR(1,I,J-1).EQ.0.) TPR=(PP(1,I+1,J-1)*(RSY(I,J)-
              2DSY(I+1,J))+PP(1,I+1,J)*DSY(I+1,J))/RSY (I,J)
 0228         IF (RSY(I,J).GT.DSY(I+1,J) .AND. DSY(I+1,J).GT.0..AND.DR(1,I+1,
              1J+1).GT.0..AND.DR(1,I,J+1).EQ.0.) TPR=(PP(1,I+1,J+1)*(RSY(I,J)-
              2DSY(I+1,J))+PP(1,I+1,J)*DSY(I+1,J))/RSY (I,J)
 0229         IF (ZSX(I,J).GT.DSX(I,J+1).AND. DSX(I,J+1).GT.0..AND.DR(1,I-1,J+1)
              1.GT.0..AND.DR(1,I-1,J).EQ.0.)TPU=(PP(1,I-1,J+1)*(ZSX(I,J)-DSX(I,
              2J+1))+PP(1,I,J+1)*DSX(I,J+1))/ZSX(I,J)
 0230         IF (ZSX(I,J).GT.DSX(I,J+1).AND. DSX(I,J+1).GT.0..AND.DR(1,I+1,J+1)
              1.GT.0..AND.DR(1,I+1,J).EQ.0.)TPU=(PP(1,I+1,J+1)*(ZSX(I,J)-DSX(I,
              2J+1))+PP(1,I,J+1)*DSX(I,J+1))/ZSX(I,J)
 0231         IF (RSY(I,J) .GT. CSY(I-1,J) .AND.  CSY(I-1,J) .GT. 0. .AND.
              1DR(1,I-1,J-1) .GT. 0. .AND. DR(1,I,J-1) .EQ. 0.) TPL=(PP(1,I-1,J-1
              2)*(RSY(I,J)-CSY(I-1,J))+PP(1,I-1,J)*CSY(I-1,J))/RSY(I,J)
 0232         IF (RSY(I,J) .GT. CSY(I-1,J) .AND.  CSY(I-1,J) .GT. 0. .AND.
              1DR(1,I-1,J+1) .GT. 0. .AND. DR(1,I,J+1) .EQ. 0.) TPL=(PP(1,I-1,J+1
              2)*(RSY(I,J)-CSY(I-1,J))+PP(1,I-1,J)*CSY(I-1,J))/RSY(I,J)
 0233         IF (ZSX(I,J).GT.CSX(I,J-1).AND. CSX(I,J-1).GT.0..AND.DR(1,I-1,J-1)
              1.GT.0..AND.DR(1,I-1,J).EQ.0.) TPB=(PP(1,I-1,J-1)*(ZSX(I,J)-
              2CSX(I,J-1))+PP(1,I,J-1)*CSX(I,J-1))/ZSX(I,J)
 0234         IF (ZSX(I,J).GT.CSX(I,J-1).AND. CSX(I,J-1).GT.0..AND.DR(1,I+1,J-1)
              1.GT.0..AND.DR(1,I+1,J).EQ.0.) TPB=(PP(1,I+1,J-1)*(ZSX(I,J)-
              2CSX(I,J-1))+PP(1,I,J-1)*CSX(I,J-1))/ZSX(I,J)
 0235         IF (I .LE. IM) GO TO 6
 0236         IF (ABS(DEL) .GT. 1.75*DX) GO TO 6
 0237         IF (J .GT. JM) K=-1
 0238         IF (J .LE. JM) K=1
 0239         IF (DSY(I,J).GT.CSY(I-1,J).AND. CSY(I-1,J).GT.0..AND.
              1DR(1,I-1,J-K).GT.0.) TPL=(PP(1,I-1,J)*CSX(I-1,J)+PP(1,I-1,J-K)*
              2(DSY(I,J)-CSY(I-1,J)))/DSY(I,J)
 0240         IF (DSY(I+1,J).GT.0..AND.DSY(I,J).GT.DSY(I+1,J).AND.
              1DR(1,I+1,J-K).GT.0.) TPR=(PP(1,I+1,J)*DSY(I+1,J)+PP(1,I+1,J-K)*
              2(DSY(I,J)-DSY(I+1,J)))/DSY(I,J)
 0241         IF (CSY(I,J+K) .LT.DSY(I+1,J+K).AND.DR(1,I+1,J).GT.0..AND.
              1 CSY(I,J+K).GT.0.) TPR=(PP(1,I+1,J+K)*(DSY(I+1,J+K)-CSY(I,J+K))+
              2PP(1,I+1,J)*(DSY(I,J)+CSY(I,J+K)-DSY(I+1,J+K)))/CSY(I,J)
 0242         IF (J .GT. JM) GO TO 42
 0243         IF (CSX(I,J) .GT. DSX(I,J+1).AND.DSX (I,J+1).GT.0..AND.DR(1,I+1,
              1J+1).GT.0.) TPU=(PP(1,I,J+1)*DSX(I,J+1)+PP(1,I+1,J+1)*(CSX(I,J)-
              2DSX(I,J+1)))/CSX(I,J)
 0244         IF (CSX(I,J-1).GT.0..AND.CSX(I,J).GT.CSX(I,J-1).AND.DR(1,I+1,J-1)
              1.GT.0.) TPB=(PP(1,I,J-1)*CSX(I,J-1)+PP(1,I+1,J-1)*(CSX(I,J)-
              2CSX(I,J-1)))/CSX(I,J)
 0245         IF (DSX(I-1,J).LT.CSX(I-1,J-1).AND.DR(1,I,J-1).GT.0..AND.
              1 DSX(I-1,J).GT.0.) TPB=(PP(1,I-1,J-1)*(CSX(I-1,J-1)-DSX(I-1,J))+
```

```
                2PP(1,I,J-1)*(DSX(I-1,J)+CSX(I,J)-CSX(I-1,J-1)))/CSX(I,J)
0246              GO TO 6
0247           62 IF (DSX(I,J).GT.CSX(I,J-1).AND. CSX(I,J-1).GT.0..AND.PP(1,I+1,J-1)
                 1.GT.0.)   TPB=(PP(1,I,J-1)*CSX(I,J-1)+PP(1,I+1,J-1)*(DSX(I,J)-
                 2CSX(I,J-1)))/DSX(I,J)
0248              IF (DSX(I,J+1).GT.0..AND.DSX(I,J).GT.DSX(I,J+1).AND.DP(1,I+1,J+1)
                 1.GT.0.)   TPU=(PP(1,I,J+1)*CSX(I,J+1)+PP(1,I+1,J+1)*(DSX(I,J)-
                 2DSX(I,J+1)))/DSX(I,J)
0249              IF (CSX(I-1,J).LT.CSX(I-1,J+1).AND.CP(1,I,J+1).GT.0..AND.
                 1 CSX(I-1,J).GT.0.) TPU=(PP(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+
                 2PP(1,I,J+1)*(DSX(I,J)+CSX(I-1,J)-DSX(I-1,J+1)))/DSX(I,J)
0250            6 CONTINUE
0251              GU(1,I,J)=FU(1,I,J)-DT*AXIJ*(.5*(TPR-TPL)+C1-C2)
                 1 /(PP(1,I,J)*DX*SFIJ)
0252              GV(1,I,J)=FV(1,I,J)-DT*AYIJ*((SYA(J)*(TPU-PP(1,I,J))-SYA(J-1)*
                 1(TPB-PP(1,I,J)))/(2.*VOL(2,J))+(C3-C4)/DY)/(PP(1,I,J)*SFIJ)
                 2-PP(1,I,J)**2/(2.*SXB(J)/DX)*DT/PP(1,I,J)*CPR(J)
0253              IF (ABS(GU(1,I,J)) .LE. EPSI) GU(1,I,J)=0.
0254              IF (ABS(GV(1,I,J)) .LE. EPSI) GV(1,I,J)=0.
0255           21 CONTINUE
0256              DO 22 I=I3,IE
0257              GU(1,I,JB-1)= GU(1,I,JB)
0258              GV(1,I,JB-1)=-GV(1,I,JB)
0259              GU(1,I,JE+1)= GU(1,I,JE)
0260              GV(1,I,JE+1)=-GV(1,I,JE)
0261           22 CONTINUE
0262              IF (IE.GT.IM) CALL CIRCLE(DX,IM1,IC,JC,JM,RA,RMEAN)
0263              MP=2
0264              DO 23 J=JB,JE
0265              DO 23 I=IS,IE
0266              IF (ZSX(I,J) .EQ. 0. .AND. RSY(I,J) .EQ. C.) GO TO 23
0267              CB=.5
0268              IF (J .EQ. JE .OR. J .EQ. JB) CB=C.18
0269              IF (I .LE. IM) GO TO 2223
0270              CALL DISTAN(I,J,DA,CB,DEL,DX)
0271              IF (ABS(DEL).LE.1.20*DX) CB=0.18
0272         2223 CONTINUE
0273              CALL        FRAC(SFIJ,AXIJ,AYIJ,     DX,DY)
0274              CALL        VISCOS(CB,CK,MP,FUR,FUL,GUR,GUL,FVT,FVB,GVT,GVB,DT,DX,
                 1IM,JM)
0275              GI(1,I,J)=FI(1,I,J)-DT*(AYIJ*(P(1,I,J)*(SYA(J)*(FVT+
                 1 GVT          +FV(1,I,J)+GV(1,I,J))/4.-SYA(J-1)*(FVB+
                 2 GVB          +FV(1,I,J)+GV(1,I,J))/4.)+.5*C3*.5*(SYB(J+1)*(
                 3 FVT         +GVT          )+SYB(J)*(FV(1,I,J)+GV(1,I,J)))-.5*C4*.5*
                 4 (SYB(J)*(FV(1,I,J)+GV(1,I,J))+SYB(J-1)*(FVB          +GVB))
                 5-.5*(FV(1,I,J)+GV(1,I,J))*SYB(J)*(C3-C4))+SXB(J)*AXIJ*(.25*
                 6 (FUR         +GUR          +FU(1,I,J)+GU(1,I,J))*(P(1,I,J)+Q1)
                 7-.25*(FUL          +GUL          +FU(1,I,J)+GU(1,I,J))*(P(1,I,J)+Q2)
                 8-.5*(FU(1,I,J)+GU(1,I,J))*(Q1-Q2)))/(PP(1,I,J)*SFIJ*VOL(2,J))
0276              GF(1,I,J)=GI(1,I,J)+.5*(GU(1,I,J)**2+GV(1,I,J)**2)
0277           23 CONTINUE
0278              CALL        TRANSP(DX,DY,DT,JB,JE,IE,EPSI,IS,RMEAN,IM,JM)
0279              DO 477 J=JB,JE
0280              DO 477 I=IS,IE
0281              IF (ZSX(I,J) .EQ. C. .AND. RSY(I,J) .EQ. C.) GO TO 477
0282              IF ((RSY(I,J).EQ.RSY(I+1,J)).AND.(RSY(I,J).EQ.RSY(I-1,J)).AND.
```



```
                 1(ZSX(I,J).EQ.ZSX(I,J+1)).AND.(ZSX(I,J).EQ.ZSX(I,J-1))) GO TO 477
0283              IF (DR(1,I-1,J).EQ.0..AND.RSY(I-1,J+1).LE.RSY(I,J+1).AND.
                 1RSY(I-1,J-1).LE.RSY(I,J-1)) GO TO 9102
0284              IF (DR(1,I-1,J).EQ.C..AND.FU(2,I,J).LT.0..AND.VOL(I,J).GE.
                 1RSY(I,J)*SYB(J)) GO TO 9102
0285              GO TO 9105
0286         9102 ZSX(I,J)=ZSX(I,J)-FU(2,I,J)*DT
0287              CSX(I,J)=ZSX(I,J)
0288              DSX(I,J)=ZSX(I,J)
0289         9105 CONTINUE
0290              IF (DR(1,I+1,J).EQ.0..AND.RSY(I+1,J+1).LE.RSY(I,J+1).AND.
                 1RSY(I+1,J-1).LE.RSY(I,J-1)) GO TO 9104
0291              IF (DR(1,I+1,J).EQ.0..AND.FU(2,I,J).GT.0..AND.VOL(I,J).GE.
                 1RSY(I,J)*SYB(J)) GO TO 9104
0292              GO TO 9107
0293         9104 ZSX(I,J)=ZSX(I,J)+FU(2,I,J)*DT
0294              CSX(I,J)=ZSX(I,J)
0295              DSX(I,J)=ZSX(I,J)
0296         9107 CONTINUE
0297          477 CONTINUE
0298              DO 25 J=JB,JE
0299              DO 25 I=IS,IE
0300              IF (ZSX(I,J) .EQ. C. .AND. RSY(I,J) .EQ. 0.) GO TO 25
0301              SFV=VOL(I ,J)
0302              IF ((RSY(I,J).EQ.RSY(I+1,J)).AND.(RSY(I,J).EQ.RSY(I-1,J)).AND.
                 1(ZSX(I,J).EQ.ZSX(I,J+1)).AND.(ZSX(I,J).EQ.ZSX(I,J-1))) GO TO 8551
0303              IF (DR(1,I-1,J).EQ.0..AND.CSY(I-1,J+1).LE.DSY(I,J+1).AND.CSY(I-1,
                 1J-1).LE.DSY(I,J-1)) VOL(I,J)=VOL(I,J)-RSY(I,J)*FU(2,I,J)*DT*SXB(J)
                 2/DX
0304              IF (DR(1,I+1,J).EQ.0..AND.DSY(I+1,J+1).LE.CSY(I,J+1).AND.DSY(I+1,
                 1J-1).LE.CSY(I,J-1)) VOL(I,J)=VOL(I,J)+RSY(I,J)*FU(2,I,J)*DT*SXB(J)
                 2/DX
0305              IF (DR(1,I-1,J).EQ.0..AND.CSY(I-1,J+1).GT.DSY(I,J+1).AND.CSY(I-1,
                 1J-1).GT.DSY(I,J-1)) VOL(I,J)=VOL(I,J)-(RSY(I,J)+DSY(I,J+1)+DSY(I,
                 2J-1)-CSY(I-1,J+1)-CSY(I-1,J-1))*FU(2,I,J)*DT*SXB(J)/DX
0306              IF (DR(1,I+1,J).EQ.0..AND.DSY(I+1,J+1).GT.CSY(I,J+1).AND.DSY(I+1,
                 1J-1).GT.CSY(I,J-1)) VOL(I,J)=VOL(I,J)+(RSY(I,J)+CSY(I,J+1)+
                 2CSY(I,J-1)-DSY(I+1,J+1)-DSY(I+1,J-1))*FU(2,I,J)*DT*SXB(J)/DX
0307              IF (DR(1,I-1,J).EQ.0..AND.CSY(I-1,J+1).GT.DSY(I,J+1).AND.
                 1CSY(I-1,J-1).LE.DSY(I,J-1)) VOL(I,J)=VOL(I,J)-(RSY(I,J)+
                 2DSY(I,J+1)-CSY(I-1,J+1))*FU(2,I,J)*DT*SXB(J)/DX
0308              IF (DR(1,I+1,J).EQ.0..AND.DSY(I+1,J+1).GT.CSY(I,J+1).AND.
                 1DSY(I+1,J-1).LE.CSY(I,J-1)) VOL(I,J)=VOL(I,J)+(RSY(I,J)+
                 2CSY(I,J+1)-DSY(I+1,J+1))*FU(2,I,J)*DT*SXB(J)/DX
0309              IF (DR(1,I-1,J).EQ.0..AND.CSY(I-1,J+1).LE.DSY(I,J+1).AND.
                 1CSY(I-1,J-1).GT.DSY(I,J-1)) VOL(I,J)=VOL(I,J)-(RSY(I,J)+
                 2DSY(I,J-1)-CSY(I-1,J-1))*FU(2,I,J)*DT*SXB(J)/DX
0310              IF (DR(1,I+1,J).EQ.0..AND.DSY(I+1,J+1).LE.CSY(I,J+1).AND.
                 1DSY(I+1,J-1).GT.CSY(I,J-1)) VOL(I,J)=VOL(I,J)+(RSY(I,J)+
                 2CSY(I,J-1)-DSY(I+1,J-1))*FU(2,I,J)*DT*SXB(J)/DX
0311              IF (DR(1,I-1,J+1).GT.0..AND.DR(1,I-1,J-1).GT.0.) GO TO 9000
0312              IF (CSY(I-1,J).GT.0..AND.CSY(I-1,J).LT.DSY(I,J).AND.FU(2,I,J).LT.
                 10..AND.VOL(I,J).GE.RSY(I,J)*SYB(J)) GO TO 9561
0313              GO TO 9562
0314         9561 CSY(I-1,J)=DSY(I,J)
0315              DSY(I-1,J)=DSY(I,J)
```

```
0216           DSY(I-1,J)=DSY(I,J)
0217      9562 CONTINUE
0218           IF ( CSY(I-1,J).GT.0..AND.CSY(I-1,J).LT.RSY(I,J)) VOL(I,J)=
              1VOL(I,J)-(RSY(I,J)-CSY(I-1,J))*FU(2,I,J)*DT*SXB(J)/DX
0219      9000 CONTINUE
0220           IF (DP(1,I+1,J+1).GT.0..AND.DP(1,I+1,J-1).GT.0.) GO TO 9001
0221           IF (DSY(I+1,J).GT.0..AND.DSY(I+1,J).LT.CSY(I,J).AND.FU(2,I,J).GT.
              10..AND.VOL(I,J).GE.RSY(I,J)*SXB(J)) GO TO 9563
0222           GO TO 9564
0223      9563 DSY(I+1,J)=CSY(I,J)
0224           CSY(I+1,J)=CSY(I,J)
0225           SSY(I+1,J)=CSY(I,J)
0226      9564 CONTINUE
0227           IF ( DSY(I+1,J).GT.0..AND.CSY(I+1,J).LT.RSY(I,J)) VOL(I,J)=
              1VOL(I,J)+(RSY(I,J)-DSY(I+1,J))*FU(2,I,J)*DT*SXB(J)/DX
0228      9001 CONTINUE
0229           IF (DP(1,I,J-1).EQ.0..AND.CSX(I+1,J-1).LE.DSX(I+1,J).AND.
              1CSX(I-1,J-1).LE.DSX(I-1,J)) VOL(I,J)=VOL(I,J)-ZSX(I,J)*FV(2,I,J)
              2*DT*SXB(J)/DX
0230           IF (DP(1,I,J+1).EQ.0..AND.DSX(I+1,J+1).LE.CSX(I+1,J).AND.
              1DSX(I-1,J+1).LE.CSX(I-1,J)) VOL(I,J)=VOL(I,J)+ZSX(I,J)*FV(2,I,J)
              2*DT*SXB(J)/DX
0231           IF (DR(1,I,J-1).EQ.0..AND.CSX(I+1,J-1).GT.DSX(I+1,J).AND.
              1CSX(I-1,J-1).GT.DSX(I-1,J)) VOL(I,J)=VOL(I,J)-(ZSX(I,J)+
              2DSX(I+1,J)+DSX(I-1,J)-CSX(I+1,J-1)-CSX(I-1,J-1))*FV(2,I,J)*DT*
              3SXB(J)/DX
0232           IF (DR(1,I,J+1).EQ.0..AND.CSX(I+1,J+1).GT.CSX(I+1,J).AND.
              1DSX(I-1,J+1).GT.CSX(I-1,J)) VOL(I,J)=VOL(I,J)+(ZSX(I,J)+CSX(I+1,J)
              2+CSX(I-1,J)-CSX(I+1,J+1)-DSX(I-1,J+1))*FV(2,I,J)*DT*SXB(J)/DX
0233           IF (DP(1,I,J-1).EQ.0..AND.CSX(I+1,J-1).GT.DSX(I+1,J).AND.
              1CSX(I-1,J-1).LE.DSX(I-1,J)) VOL(I,J)=VOL(I,J)-(ZSX(I,J)+DSX(I+1,J)
              2-CSX(I+1,J-1))*FV(2,I,J)*DT*SXB(J)/DX
0234           IF (DR(1,I,J+1).EQ.0..AND.CSX(I+1,J+1).GT.CSX(I+1,J).AND.
              1DSX(I-1,J+1).LE.CSX(I-1,J)) VOL(I,J)=VOL(I,J)+(ZSX(I,J)+
              2CSX(I+1,J)-DSX(I+1,J+1))*FV(2,I,J)*DT*SXB(J)/DX
0235           IF (DP(1,I,J-1).EQ.0..AND.CSX(I+1,J-1).LE.DSX(I+1,J).AND.
              1CSX(I-1,J-1).GT.DSX(I-1,J)) VOL(I,J)=VOL(I,J)-(ZSX(I,J)+
              2DSX(I-1,J)-CSX(I-1,J-1))*FV(2,I,J)*DT*SXB(J)/DX
0236           IF (DR(1,I,J+1).EQ.0..AND.DSX(I+1,J+1).LE.CSX(I+1,J).AND.
              1DSX(I-1,J+1).GT.CSX(I-1,J)) VOL(I,J)=VOL(I,J)+(ZSX(I,J)+
              2CSX(I-1,J)-DSX(I-1,J+1))*FV(2,I,J)*DT*SXB(J)/DX
0237           IF (DP(1,I+1,J-1).GT.0..AND.DR(1,I-1,J-1).GT.0.) GO TO 9003
0238           IF (CSX(I,J-1).GT.0..AND.DSX(I,J).GT.CSX(I,J-1).AND.FV(2,I,J).LT.
              10..AND.VOL(I,J).GE.ZSX(I,J)*SYB(J)) GO TO 9565
0239           GO TO 9566
0240      9565 CSX(I,J-1)=DSX(I,J)
0241           DSX(I,J-1)=DSX(I,J)
0242           ZSX(I,J-1)=DSX(I,J)
0243      9566 CONTINUE
0244           IF ( CSX(I,J-1).GT.0..AND.ZSX(I,J).GT.CSX(I,J-1)) VOL(I,J)=
              1VOL(I,J)-(ZSX(I,J)-CSX(I,J-1))*FV(2,I,J)*DT*SXB(J)/DX
0245      9003 CONTINUE
0246           IF (DR(1,I+1,J+1).GT.0..AND.DR(1,I-1,J+1).GT.0.) GO TO 9002
0247           IF (DSX(I,J+1).GT.0..AND.CSX(I,J).GT.DSX(I,J+1).AND.FV(2,I,J).GT.
              10..AND.VOL(I,J).GE.ZSX(I,J)*SYB(J)) GO TO 9567
0248           GO TO 9568
```



```
0249      9567 DSX(I,J+1)=CSX(I,J)
0250           CSX(I,J+1)=CSX(I,J)
0251           ZSX(I,J+1)=CSX(I,J)
0252      9568 CONTINUE
0253           IF ( DSX(I,J+1).GT.0..AND.ZSX(I,J).GT.DSX(I,J+1)) VOL(I,J)=
              1VOL(I,J)+(ZSX(I,J)-CSX(I,J+1))*FV(2,I,J)*DT*SXB(J)/DX
0254      9002 CONTINUE
0255      9551 CONTINUE
0256           DP(2,I,J)=DP(2,I,J)*SFV/VOL(I,J)
0257           DR(2,I,J)=PB(2,I,J)*SFV/VOL(I,J)
0258           IF (FI(2,I,J) .LE. 0.) WRITE (6,256) I,J,FI(2,I,J)
0259       256 FORMAT (1H ,I3,I4,F15.8)
0260           IF (FI(2,I,J) .LE. 0.) GO TO 31
0261           P(2,I,J)=(FK-1.)*DR(2,I,J)*FI(2,I,J)
0262           PP(2,I,J)=P(2,I,J)+PE(2,I,J)**2/(8.*3.1416)
0263           TMP(2,I,J)=FI(2,I,J)/CV
0264           SS2=(FK*FF*TMP(2,I,J))**.5
0265           TC(2,I,J)=SS2
0266        25 CONTINUE
0267           DO 482 J=JB,JE
0268           DO 482 I=IS,IE
0269           IF (ZSX(I,J) .EQ. 0. .AND. RSY(I,J) .EQ. 0.) GO TO 482
0270           IF ((RSY(I,J).EQ.RSY(I+1,J)).AND.(RSY(I,J).EQ.RSY(I-1,J)).AND.
              1(ZSX(I,J).EQ.ZSX(I,J+1)).AND.(ZSX(I,J).EQ.ZSX(I,J-1))) GO TO 482
0271           IF (DR(1,I,J-1).EQ.0..AND.ZSX(I+1,J-1).LE.ZSX(I+1,J).AND.
              1ZSX(I-1,J-1).LE.ZSX(I-1,J)) GO TO 9201
0272           IF (DP(1,I,J-1).EQ.0..AND.FV(2,I,J).LT.0..AND.VOL(I,J)
              1.GE.ZSX(I,J)*SYB(J)) GO TO 9201
0273           GO TO 9205
0274      9201 RSY(I,J)=RSY(I,J)-FV(2,I,J)*DT
0275           CSY(I,J)=RSY(I,J)
0276           DSY(I,J)=RSY(I,J)
0277      9205 CONTINUE
0278           IF (DR(1,I,J+1).EQ.0..AND.ZSX(I+1,J+1).LE.ZSX(I+1,J).AND.
              1ZSX(I-1,J+1).LE.ZSX(I-1,J)) GO TO 9203
0279           IF (DR(1,I,J+1).EQ.0..AND.FV(2,I,J).GT.0..AND.VOL(I,J)
              1.GE.ZSX(I,J)*SYB(J)) GO TO 9203
0280           GO TO 9207
0281      9203 RSY(I,J)=RSY(I,J)+FV(2,I,J)*DT
0282           CSY(I,J)=RSY(I,J)
0283           DSY(I,J)=RSY(I,J)
0284      9207 CONTINUE
0285       482 CONTINUE
0286           DO 35 J=JB,JE
0287           DO 35 I=IS,IE
0288           IF (ZSX(I,J) .EQ. 0. .AND. RSY(I,J) .EQ. 0.) GO TO 35
0289           FU(1,I,J)=FU(2,I,J)
0290           FV(1,I,J)=FV(2,I,J)
0291           FI(1,I,J)=FI(2,I,J)
0292           FE(1,I,J)=FE(2,I,J)
0293           DR(1,I,J)=DR(2,I,J)
0294           PP(1,I,J)=PP(2,I,J)
0295           P (1,I,J)=P (2,I,J)
0296           PR(1,I,J)=PR(2,I,J)
0297           TC(1,I,J)=TC(2,I,J)
0298           TMP(1,I,J)=TMP(2,I,J)
```

```
      IF (ABS(FU(1,I,J)) .LE. EPSI) FU(1,I,J)=0.
      IF (ABS(FV(1,I,J)) .LE. EPSI) FV(1,I,J)=0.
      IF (I.LE. IM) GO TO 9559
      CALL DISTAN(I,J,CA,CP,DEL,CX)
      IF (ABS(DEL).LE.1.74*CX) GO TO 9559
 9559 CONTINUE
      IF (DR(1,I,J+1).GT.0..AND.CR(1,I,J-1).GT.0..AND.RSY(I,J).NE.DY)
     1GO TO 9108
      GO TO 9109
 9108 RSY(I,J)=DY
      CSY(I,J)=DY
      DSY(I,J)=DY
 9109 CONTINUE
      IF (DR(1,I+1,J).GT.0..AND.CR(1,I-1,J).GT.0..AND.ZSX(I,J).NE.DX) GO
     1 TO 9208
      GO TO 9209
 9208 ZSX(I,J)=DX
      CSX(I,J)=DX
      DSX(I,J)=DX
 9209 CONTINUE
      IF (RSY(I,J).GT.DY.AND.VOL(I,J).LT. VOL(2,J)/2.) GO TO 9551
      GO TO 9552
 9551 RSY(I,J)=DY
      CSY(I,J)=DY
      DSY(I,J)=DY
 9552 IF (ZSX(I,J).GT.DX .AND.VOL(I,J).LT.VOL(2,J)/2.) GO TO 9553
      GO TO 9554
 9553 ZSX(I,J)=DX
      CSX(I,J)=DX
      DSX(I,J)=DX
 9554 IF (RSY(I,J) .GT. 1.2*DY) GO TO 9555
      GO TO 9556
 9555 RSY(I,J)=VOL(I,J)*DX/SXR(J)/ZSX(I,J)
      CSY(I,J)=RSY(I,J)
      DSY(I,J)=RSY(I,J)
 9556 IF (ZSX(I,J) .GT. 1.2*DX) GO TO 9557
      GO TO 9558
 9557 ZSX(I,J)=VOL(I,J)*DX/SXR(J)/RSY(I,J)
      CSX(I,J)=ZSX(I,J)
      DSX(I,J)=ZSX(I,J)
 9558 CONTINUE
   35 CONTINUE
      TIME=TIME+DT
      IF (M .EQ. 400) GO TO 1237
      GO TO 1238
 1237 CONTINUE
      REWIND 8
      WRITE (8,1229) IE,M,TIME,DT
 1229 FORMAT (I4,I4,E15.8,E15.8/)
      DO 1106 J=1,JD
      DO 1106 I=1,IE
      WRITE (8,1109) DR(1,I,J),FU(1,I,J),FV(1,I,J),GE(1,I,J),FI(1,I,J),
     1TC(1,I,J),PP(1,I,J),P(1,I,J),RB(1,I,J),IMP(1,I,J),VOL(I,J),
     2CSX(I,J),DSX(I,J),ZSX(I,J),CSY(I,J),DSY(I,J),RSY(I,J),RV(I,J),
     3SY(J),FE(1,I,J)
 1106 CONTINUE
```



```
 1109 FORMAT (5E15.8/)
      END FILE 8
 1238 CONTINUE
      GO TO 8
   31 CONTINUE
      CALL EXIT(Z)
      STOP
      END
```

```
      SUBROUTINE CIRCLE(DX,IM1,IC,JC,JM,RA,RMEAN)
      DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
     1FF(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BE(2,137,22),
     2GU(1,137,22),GV(1,137,22),GI(1,137,22),CF(1,137,22),TMP(2,137,22),
     3SYA(22),SXB(22),SYP(22),VCL(137,22),SI(22),CIE(22),EV(137,22),
     4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
     5ZSX(137,22)
      COMMON DR,FU,FV,FI,FF,TC,PP,P,BE,GU,GV,GI,CF,TMP,SYA,SXP,SYP,VCL,
     1SI,CPB,BV,CSX,DSX,CSY,CSY,RSY, ZSX,Q1,Q2,Q3,Q4,M,I,J
      DIMENSION ZITA(137),CF(137),ZITF(11),CC(11)
      FO=0.000002
      JM1=JM+1
      XIM1=IM1
      DO 1 I=IM1,IC
      FN=1.
      XI=I
      XB=(XI-XIM1)*DX
      XA=(XI-XIM1+1.)*DX
      YA=SQRT(RA**2-(XA/2.25)**2)
      YB=SQRT(RA**2-(XB/2.25)**2)
    2 YN=FN*DX
      DL=YB-YN
      IF (DL .LE. 1.716*DX) GO TO 4
      FN=FN+1.
      GO TO 2
    4 NF=FN
      J=JM1+NF
      IF (M .GT. 2) GO TO 222
      DSY(I,J)=DL
      CSY(I,J)=YA-YN
      IF (I .EQ. IC) CSY(I,J)=DX
      VOL(I,J)=(DSY(I,J)+CSY(I,J))*SXB(J)/2.
      RSY(I,J)=AMAX1(CSY(I,J),DSY(I,J))
  222 CONTINUE
      DB=YB+.5*CSY(I,J)
      DA=(XI-XIM1+.5)*DX
      IF (M .GT. 2) GO TO 131
      Y=YA
  117 Y=Y+ED
      X=2.25*SQRT(RA**2-Y**2)
      XX=DA*Y*(2.25**2)/(Y*2.25**2+DB-Y)
      IF (ABS(X-XX) .LE.150.*ED) GO TO 118
      GO TO 117
  118 TA=(DA-X)/(DB-Y)
      ANGL=ATAN(TA)
      DS=SQRT((DA-X)**2+(DB-Y)**2)
      ZITA(I)=ANGL
      DP(I)=DS
  131 ANGL=ZITA(I)
      DS=DP(I)
      DM=2.*DS*SIN(ANGL)
      DN=2.*DS*COS(ANGL)
      IB=I
      JB=J+1
      GZ=DA-DM
      GR=DB-DN
```



```
      JG=GR/DX
      IG=GZ/DX
      IW=IM1+IG
      JW=JM1+JG
      QI=IG
      QJ=JG
      CALL      AREAV(QI,QJ,GZ,GR,DX,IW,JW,W1,W2,W3,W4,W5,W6,W7,W8,W9)
      DR(1,IB,JB)=W1
      P(1,IB,JB)=W6
      BE(1,IB,JB)=W7*(6.25     +GR)/(6.25     +DB)
      TC(1,IB,JB)=W8
      GE(1,IB,JB)=W9
      FU(1,IB,JB)=-W3*SIN(2.*ANGL)+W2*COS(2.*ANGL)
      FV(1,IB,JB)=-W2*SIN(2.*ANGL)-W3*COS(2.*ANGL)
      GV(1,IB,JB)=-W4*SIN(2.*ANGL)-W5*COS(2.*ANGL)
      GU(1,IB,JB)=-W5*SIN(2.*ANGL)+W4*COS(2.*ANGL)
      PP(1,IB,JB)=P(1,IB,JB)+BE(1,IB,JB)**2/25.1328
      ZSX(IB,JB)=0.
      RSY(IB,JB)=0.
      CSX(IB,JB)=0.
      DSX(IB,JB)=0.
      CSY(IB,JB)=0.
      DSY(IB,JB)=0.
      IF (M .LE. 2) WRITE (6,9) CSX(I,J),DSX(I,J),CSY(I,J),DSY(I,J),
     1I,J,IB,JB,IW,JW
    1 CONTINUE
    9 FORMAT (1H , 4F10.5,6I4)
      XJM=JM1
      DO 5 J=JM1,JC
      XJ=J
      YD=(XJ-XJM)*DX
      YC=(XJ-XJM+1.)*DX
      XD=2.25*SQRT(RA**2-YC**2)
      XC=2.25*SQRT(RA**2-YC**2)
      FN=1.
    6 XN=FN*DX
      DL=XD-XN
      IF (DL .LE. 1.716*DX) GO TO 7
      FN=FN+1.
      GO TO 6
    7 NF=FN
      I=IM1+NF
      IF (M .GT. 2) GO TO 224
      DSX(I,J)=DL
      CSX(I,J)=XC-XN
      VOL(I,J )=(DSX(I,J )+CSX(I,J ))*SXB(J)/2.
      IF (J .EQ. JC) VOL(I,J)=.5*CSY(I,J)*DSX(I,J)*SXB(J)/DX
      ZSX(I,J)=AMAX1(CSX(I,J),CSX(I,J))
  224 CONTINUE
      DA=XD+.5*CSX(I,J)
      DB=(XJ-XJM+.5)*DX
      IF (M .GT. 2) GO TO 132
      X=XC
  119 X=X+ED
      Y=SQRT(RA**2-(X/2.25)**2)
      YY=X*DB/((DA-X)*2.25**2+X)
```

```
      IF (ABS(Y-YY) .LE.150.*ED) GO TO  120
      GO TO 119
  120 TA=(DA-X)/(DB-Y)
      ANGL=ATAN(TA)
      DS=SQRT((DA-X)**2+(DB-Y)**2)
      ZITB(J)=ANGL
      DQ(J)=DS
  132 ANGL=ZITB(J)
      DS=DQ(J)
      DM=2.*DS*SIN(ANGL)
      DN=2.*DS*COS(ANGL)
      IB=I+1
      JB=J
      GZ=DA-DM
      GR=DB-DN
      IG=GZ/DX
      JG=GR/DX
      IW=IM1+IG
      JW=JM1+JG
      CI=IG
      CJ=JG
      CALL       AREAV(CI,CJ,GZ,GR,DX,IW,JW,W1,W2,W3,W4,W5,W6,W7,W8,W9)
      DR(1,IB,JB)=W1
      P(1,IB,JB)=W6
      BB(1,IB,JB)=W7*(6.25    +GR)/(6.25    +DB)
      TC(1,IB,JB)=W8
      GE(1,IB,JB)=W9
      FU(1,IB,JB)=-W3*SIN(2.*ANGL)+W2*COS(2.*ANGL)
      FV(1,IB,JB)=-W2*SIN(2.*ANGL)-W3*COS(2.*ANGL)
      GU(1,IB,JB)=-W5*SIN(2.*ANGL)+W4*COS(2.*ANGL)
      GV(1,IB,JB)=-W4*SIN(2.*ANGL)-W5*COS(2.*ANGL)
      PP(1,IB,JB)=P(1,IB,JB)+BB(1,IB,JB)**2/25.1329
      ZSX(IB,JB)=0.
      RSY(IB,JB)=0.
      CSX(IB,JB)=0.
      DSX(IB,JB)=0.
      CSY(IB,JB)=0.
      DSY(IB,JB)=0.
      IF (M .LE. 2) WRITE (6,9) CSX(I,J),DSX(I,J),CSY(I,J),DSY(I,J),
     1I,J,IB,JB,IW,JW
    5 CONTINUE
      RETURN
      END
```



```
      SUBROUTINE CNTLN(Z,F,NX,NY,XMIN,XMAX,YMIN,YMAX,CNTMX,CNTMN,NCNT,
     1ITICK,ICRS,LX,LY,LZ,IDEBUG,DR,RSY,FU,FV,TIME,M)
      DIMENSION PX(17),PY(17),XAX(137,22),YAY(137,22),XPX(137,22),
     1YPY(137,22),F(137,22),DR(2,137,22),RSY(137,22),FU(2,137,22),
     2FV(2,137,22)
      DIMENSION Z(200),        CNT(10),MCC(4),MCS(4),LX(2),LY(2),
     1LZ(5),        XPT(2),YPT(2),XPT1(2),YPT1(2),CNTR(2),CNTR1(5),CNTR2
     2(3)
      INTEGER*4 IBK(25)
      EQUIVALENCE (CNTR2(1),CNTR1(3))
      DATA IBK/25*'    '/,CNTR/'CONT','OURS'/,CNTR1/'    ',' =  ',
     1'    ','    ','    '/
C
C  Z IS MODE ARRAY
C MODESG MUST BE CALLED IN THE MAIN CALLING ROUTINE
C  F IS ARRAY OF VALUES
C  NX IS NUMBER OF POINTS IN HORIZONTAL DIRECTION
C  NY IS NUMBER OF POINTS IN VERTICAL DIRECTION
C   FOR GOOD PLOTS, NX AND NY SHOULD BE ABOUT 50
C  SMIN,SMAX,YMIN,YMAX SPECIFY LIMITS OF GRID
C  CNTMX IS MAXIMUM VALUED CONTOUR
C  CNTMN IS MINIMUM VALUED CONTOUR
C  NCNT IS NUMBER OF CONTOURS, IT MUST BE LESS THAN 11
C  ITICK=1 PLOT TICK MARKS, OTHERWISE NOT
C  ICRS=1 DRAW LINES FOR X AND Y=0 AXES, OTHERWISE NOT
C  LX,LY ARE AXIS LABELS, UP TO 8 CHARACTERS
C  LZ IS GRAPH TITLE UP TO 20 CHARACTERS
C  SET IDEBUG=0 FOR NORMAL RUNS.  IF IDEBUG=1,ROUTINE WILL PRINT OUT
C  IX,IY,X,Y,A,B,C,D, AND ANY POINTS IT IS GOING TO PLOT
C  WHEN OPERATING ON EACH GRID BOX, THE CORNERS AND SIDES ARE LABELLED
C  COUNTER-CLOCKWISE FROM LOWER LEFT CORNER AND BOTTOMSIDE=1
      NXX=NX-1
      NYY=NY-1
      YDIS=1.2*(YMAX-YMIN)/(XMAX-XMIN)
      IF(NCNT .LE. 10   .AND.  NCNT .GE. 1)GO TO 10
      WRITE(6,101)NCNT
      RETURN
   10 YEN=0.64*YDIS
      CALL OBJCTG(Z,0.1,0.64,1.2,YEN)
      IE=12
      WRITE (6,999) IE
      CALL SUBJEG(Z,XMIN,YMIN,XMAX,YMAX)
      IE=13
      WRITE (6,999) IE
C  THIS SETS UP A GRID 2458 RASTERS SQUARE AT THE LOWER LEFT
      DX=XMAX-XMIN
      DY =YMAX-YMIN
      CALL GRIDG(Z,DX,DY,0,0)
      IE=14
      WRITE (6,999) IE
C  DRAWS GRID, NO LABELLING, NO GRID LINES
      CALL TITLEG(Z,8,LX,8,LY,20,LZ)
      IE=15
      WRITE (6,999) IE
C  TITLE ABOVE THE GRID
      CALL SETSMG(Z,104,1.5)
```

```
            IE=16
            WRITE (6,999) IE
C     LABELS AXES AND TITLES GRID
            IF(ITICK .NE. 1)GO TO 1
            CALL SETSMG(Z,102,.01)
            CALL SETSMG(Z,103,.01)
C     HEIGHT OF TICK MARKS ABOUT 30 RASTERS
            DX=DX/(NX-2)
            DY=DY/(NY-2)
            CALL LABELG(Z,0,DX,1,IEK)
            IE=17
            WRITE (6,999) IE
            CALL LABELG(Z,1,DY,1,IEK)
            IE=18
            WRITE (6,999) IE
  1         IF(ICRS .NE. 1)GO TO 2
            CALL SETSMG(Z,14,0.)
C     SCALING IN SUBJECT SPACE UNITS
            CALL SEGMTG(Z,1,XMIN,0.,XMAX,0.)
            CALL SEGMTG(Z,1,0.,YMIN,0.,YMAX)
C     DRAWS CROSS LINES
  2         CONTINUE
            DCNT=(CNTMX-CNTMN)/(NCNT-1)
            CNT(1)=CNTMN
            DO 3 I=2,NCNT
  3         CNT(I)=CNT(I-1)+DCNT
            WRITE(6,103)(I,CNT(I),I=1,NCNT)
            CALL OBJCTG(Z,0.1,0.845,1.33,1.)
            IE=19
            WRITE (6,999) IE
C     NWO OBJECT SPACE IS TO RIGHT OF GRID WHERE WE WILL LIST CONTOUR
C     VALUES.  MODE NO. 14 ALREADY=0 FROM A PREVIOUS CALL (SUBJECT SPACE UNITS)
C     CHANGE IT BACK TO 39 (OBJECT SPACE UNITS)
            CALL SETSMG(Z,14,3.)
            IE=20
            WRITE (6,999) IE
            CALL LEGNDG(Z,0.1,0.98,6,'CYCLE=')
            IE=21
            WRITE (6,999) IE
            CALL NUMBRG(Z,0.2,0.98,4,M)
            IE=22
            WRITE (6,999) IE
            CALL LEGNDG(Z,0.3,0.98,6,',TIME=')
            IE=23
            WRITE (6,999) IE
            CALL NUMBRG(Z,0.4,0.98,-14.6,TIME)
            IE=24
            WRITE (6,999) IE
            CALL LEGNDG(Z,0.1,0.96,8,CNTR)
            IE=25
            WRITE (6,999) IE
            YY=0.94
            XX=0.1
            DO 4 I=1,NCNT
            IF (I    .EQ. 6) XX=0.1
            IF (I    .EQ. 6) YY=0.92
```



```
            CALL FMTSG(Z,1,4,0,I,CNTR1(1))
            CALL FMTSG(Z,3,12,3,CNT(I),CNTR2)
            CALL LEGNDG(Z,XX,YY,20,CNTR1)
            XX=XX+0.2
  4         CONTINUE
            CALL OBJCTG(Z,0.1,0.64,1.3,YEN)
            IE=26
            WRITE (6,999) IE
C     RESETS OBJECT SPACE TO SQUARE
C     NOW START WORK DRAWING CONTOURS
             CALL SETSMG(Z,14,0.)
            IE=27
            WRITE (6,999) IE
            NX1=NX-2
            NY1=NY-2
            DO 5 I=1,NCNT
            CALL FMTSG(Z,1,1,0,I,ICHT)
            ICCUNT=0
            X=XMIN+DX/2.
C     WHEN ICOUNT IS A MULTIPLE OF 10, PRINT THE CONTOUR NUMBER ON THE CONTOUR
            DO 6 IX=2,NX1
            Y=YMIN+DY/2.
            DO 7 JY=2,NY1
            DO 8 JJ=1,4
            MCC(JJ)=0.
  8         MCS(JJ)=0.
            IF(IDEBUG .EQ. 1)WRITE(6,104)IX,JY,X,Y
            IF(F(IX,JY) .EQ. CNT(I))      MCC(1)=1
            IF(F(IX,JY+1) .EQ. CNT(I))    MCC(2)=1
            IF(F(IX+1,JY+1) .EQ. CNT(I))  MCC(3)=1
            IF(F(IX+1,JY) .EQ. CNT(I))    MCC(4)=1
            A=CNT(I)-F(IX,JY)
            B=CNT(I)-F(IX,JY+1)
            C=CNT(I)-F(IX+1,JY+1)
            D=CNT(I)-F(IX+1,JY)
            IF(IDEBUG .EQ. 1)WRITE(6,105)A,B,C,D
            IF(A*D .LT. 0.)MCS(1)=1
            IF(A*B .LT. 0.)MCS(2)=1
            IF(B*C .LT. 0.)MCS(3)=1
            IF(C*D .LT. 0.)MCS(4)=1
C     THAT IS, IF THE PRODUCT OF THE DIFFEREN ES IS NEGATIVE, THE
C     TWO ENDS LIE ON UPPOSITE SIDES OF CNT(I), OR THE LINE INTERSECTS
C     THE GRID BOX SIDE.  A,B,C,AND D WILL BE USED LATER TO CALCULATE
C     INTERPOLATIONS
            IMCC=MCC(1)+MCC(2)+MCC(3)+MCC(4)
            IMCS=MCS(1)+MCS(2)+MCS(3)+MCS(4)
            IF(IMCC .GE. 3)GO TO 7
C     IF MORE THAT 2 CORNERS EQUAL TO THE CONTOUR, DONT DRAW ANY LINES IS THIS
C     BOX, BECAUSE IT IS AN AREA OF CONSTANT F
            IF(IMCS .NE. 1  .AND. IMCS .NE. 3)GO TO 9
            IF(IMCC .NE. 0)GO TO 9
            WRITE(6,100)IMCS
C     THE LOGIC OF THE SITUATION MAKES IT OBVIOUS THAT THE CONTOUR LINE
C     MAY ONLY INTERSECT AN EVEN NUMBER OF SIDES, UNLESS IT ALSO INTERSECTS
C     A CORNER
            RETURN
```

```
 9     IF(IMCS .EQ. 0  .AND.  IMCC .EQ. 0)GO TO 7
C  LINE DOES NOT CROSS THIS BOX, GO ON TO NEXT
       IF(IMCS .EQ. 2)GO TO 11
C  GO TO 11 IF 2 SIDES INTERSECTED.  IF THE OPPOSITE CORNER IS ALSO
C  INTERSECTED, IGNORE IT
       IF(IMCS .EQ. 4)GO TO 12
C  GO TO STATEMENT 12 IF ALL FOUR SIDES INTERSECTED
       IF(IMCS .EQ. 1  .AND.  IMCC .EQ. 1)GO TO 13
C  TO 13 IF ONE SIDE AND ONE CORNER INTERSECTED
       IF(IMCC .EQ. 1)GO TO 7
C  IF ONE CORNER INTERSECTED ONLY, IGNORE IT.  IT WILL BE PICKED UP
C  IN ANOTHER BOX.
C  AT THIS POINT IN THE CODE, IMCC=2, IMCS=0, OR TWO CORNERS INTERSECTED,
C  NO SIDES
C  IF THE TWO CORNERS ARE OPPOSITE, IGNORE THEM IF THE OTHER TWO
C  CORNERS ARE ON THE SAME SIDE OF THE CONTOUR
       IF(MCC(1) .EQ. MCC(3)  .AND.  MCC(2) .EQ. MCC(4))GO TO 71
       X1=X
       Y1=Y
       X2=X
       Y2=Y
       IF(MCC(1) .EQ. 1)GO TO 14
       IF(MCC(2) .NE. 1)GO TO 15
       Y1=Y1+DY
       Y2=Y2+DY
       X2=X2+DX
       GO TO 17
15     IF(MCC(3) .NE. 1)GO TO 14
       X1=X1+DX
       Y1=Y1+DY
       X2=X2+DX
       GO TO 17
14     IF(MCC(4) .NE. 1) GO TO 16
       X2=X2+DX
       GO TO 17
16     Y2=Y2+DY
17     CONTINUE
C
       IF(IDEBUG .EQ. 1)WRITE(6,106)X1,Y1,X2,Y2
C
       CALL SEGMTG(Z,1,X1,Y1,X2,Y2)
       ICOUNT=ICOUNT+1
       IF(ICOUNT .NE. 10)GO TO 7
       ICOUNT=0
       CALL LEGNDG(Z,X1,Y1,1,ICHT)
       GO TO 7
C  HERE TWO OPPOSITE CORNERS HAVE A CONTOUR GOING THROUGH THEM.
C  TEST IF OTHER TWO CORNERS ARE ON SAME SIDE OF THE CONTOUR.  IF SO
C  SKIP OUT.  THE CORNERS WILL BE PICKED UP IN OTHER BOXES.
71     IF(MCC(2) .EQ. 1)GO TO 72
       IF(B*D .GT. 0)GO TO 7
C  IF CORNERS 2 AND 4 ARE ON SAME SIDE OF CONTOUR GO OUT, OTHERWISE
C  CONNECT 1 AND 3
       X1=X
       Y1=Y
       X2=X+DX
```



```
       Y2=Y+DY
       GO TO 17
72     IF(A*C .GT. 0)GO TO 7
C  IF 1 AND 3 ON SAME SIDE GO OUT, OTHERWISE CONNECT 2 AND 4
       X1=X
       Y1=Y+DY
       X2=X+DX
       Y2=Y
       GO TO 17
C  NOW TAKE IMCS=2, IMCC=1,0, BUT DOESNT PLAY A PART
11     IO=1
       IF(MCS(1) .NE. 1)GO TO 151
       XPT(IO)=X+DX*A/(F(IX+1,JY)-F(IX,JY))
       YPT(IO)=Y
       IO=IO+1
151    IF(MCS(2) .NE. 1)GO TO 161
       XPT(IO)=X
       YPT(IO)=Y+DY*A/(F(IX,JY+1)-F(IX,JY))
       IO=IO+1
       IF(IO .EQ. 3)GO TO 18
161    IF(MCS(3) .NE. 1)GO TO 171
       XPT(IO)=X+DX*B/(F(IX+1,JY+1)-F(IX,JY+1))
       YPT(IO)=Y+DY
       IO=IO+1
       IF(IO .EQ. 3)GO TO 18
171    IF(MCS(4) .NE. 1)GO TO 19
       XPT(IO)=X+DX
        YPT(IO)=Y+DY*D/(F(IX+1,JY+1)-F(IX+1,JY))
18     CONTINUE
C
       IF(IDEBUG .EQ. 1)WRITE(6,107)(XPT(II),YPT(II),II=1,2)
C
       CALL LINESG(Z,2,XPT,YPT)
       ICOUNT =ICOUNT+1
       IF(ICOUNT .NE. 10)GO TO 7
       ICOUNT=0
       CALL LEGNDG(Z,XPT(1),YPT(1),1,ICHT)
       GO TO 7
19      WRITE(6,102)
       GO TO 7
C  NOW CASE OF ALL SIDES INTERSECTED, IMCC=0
    12 DX1=DX*A/(F(IX+1,JY)-F(IX,JY))
       DX2=DX*B/(F(IX+1,JY+1)-F(IX,JY+1))
       DY1=DY*A/(F(IX,JY+1)-F(IX,JY))
       DY2=DY*D/(F(IX+1,JY+1)-F(IX+1,JY))
C  NOW TEST NEXT BOX TO SEE WHICH PAIRS TO CONNECT.
C  IF F(IX,JY+2) AND F(IX,JY+1) ARE ON THE SAME SIDE OF CNT(I),
C  THEN CONNECT SIDES 2 AND 3 AND SIDES 1 AND 4, OTHERWISE 1-2 AND 3-4.
C  IF JY+2 .GT. NY, USE F(IX-1,JY) AND F(IX,JY). IF THEY ARE ON THE SAME
C  SIDE OF CNT(I), CONNECT 1-2,3-4,OTHERWISE 1-4,2-3
C
       IF (JY+2 .GT. NY-1) GO TO 20
       XX=B*(CNT(I)-F(IX,JY+2))
       IF(XX .LT. 0.)GO TO 21
23     XPT(1)=X+DX1
       YPT(1)=Y
```

```
          XPT(2)=X
          YPT(2)=Y+DY1
          XPT1(1)=X+DX
          YPT1(1)=Y+DY2
          XPT1(2)=X+DX2
          YPT1(2)=Y+DY
          GO TO 22
C   THAT CONNECTS 1-4,2-3
21        XPT(1)=X+CX1
          YPT(1)=Y
          XPT(2)=X+CX2
          YPT(2)=Y+CY
          XPT1(1)=X
          YPT1(1)=Y+DY1
          XPT1(2)=X+DX
          YPT1(2)=Y+DY2
C   THAT CONNECTED 1-2,3-4
22        CONTINUE
C
          IF(IDEBUG .EQ. 1)WRITE(6,108)(XPT(II),YPT(II),XPT1(II),YPT1(II),
         1II=1,2)
C
          CALL SEGMTG(Z,2,XPT,YPT,XPT1,YPT1)
          ICOUNT=ICCUNT+1
          IF(ICOUNT .NE. 10)GO TO 7
           ICCUNT=0
          CALL LEGNDG(Z,XPT(1),YPT(1),1,ICHT)
          GO TO 7
20        XX=A*(CNT(I)-F(IX,JY-1))
          IF(XX .LT. C.)GO TO 23
          GO TO 21
C   NOW WE TAKE CASE OF CNE SICE ANC CNE COFNER
13        X1=X
          Y1=Y
          IF(MCC(2) .EC. 1)Y1=Y1+CY
          IF(MCC(3) .EQ. 1)X1=X1+DX
          IF(MCC(1) .EQ. 1)GC TC 24
          IF(MCC(4) .NE. 1)GC TO 24
          X1=X+DX
          Y1=Y+DY
24        IF(MCS(1) .NE. 1)GC TO 25
          X2=X+DX*A/(F(IX+1,JY)-F(IX,JY))
          Y2=Y
          GO TO 28
25        IF(MCS(2) .NE. 1)GC TO 26
          X2=X
          Y2=Y+DY*A/(F(IX,JY+1)-F(IX,JY))
          GO TO 28
26        IF(MCS(3) .NE. 1)GC TO 27
          X2=X+DX*B/(F(IX+1,JY+1)-F(IX,JY+1))
          Y2=Y+CY
          GO TO 28
27        X2=X+DX
          Y2=Y+DY*D/(F(IX+1,JY+1)-F(IX+1,JY))
28        CONTINUE
C
```



```
          IF(IDEBUG .EQ. 1)WRITE(6,109)X1,Y1,X2,Y2
C
          CALL SEGMTG(Z,1,X1,Y1,X2,Y2)
          ICOUNT=ICOUNT+1
          IF(ICOUNT .NE. 1)GC TO 7
          ICOUNT=0
          CALL LEGNDG(Z,X1,Y1,1,ICHT)
7         Y=Y+DY
6         X=X+DX
5         CONTINUE
          DO 201 I=2,NXX
          XF=I
          DO 201 J=2,NYY
          YG=J
          XAX(I,J)=XMIN+(XF*2.-3.)*CX/2.
          YAY(I,J)=YMIN+(YG*2.-3.)*DY/2.
          XPX(I,J)=XAX(I,J)+FU(1,I,J)/0.8E+08*CX
          YPY(I,J)=YAY(I,J)+FV(1,I,J)/0.8E+08*DY
      201 CONTINUE
          DO 209 I=2,NXX
          DO 209 J=2,NYY
          IF (RSY(I,J) .EC. 0.) GO TC 209
          CALL POINTG(Z,1,XAX(I,J),YAY(I,J))
      209 CONTINUE
          CALL RSETMG(Z)
          IE=6
          WRITE (6,999) IE
          YEN=0.36*YDIS
          CALL OBJCTG(Z,0.1,C.36,1.3,YEN)
          IE=9
          WRITE (6,999) IE
          CALL SUBJEG(Z,XMIN,YMIN,XMAX,YMAX)
          IE=2
          WRITE (6,999) IE
          DX=XMAX-XMIN
          CY=YMAX-YMIN
          CALL GRIDG(Z,DX,DY,0,0)
          IE=10
          WRITE (6,999) IE
          CALL TITLEG(Z,15,15HDENSITY SHACING,8,LY,0,0)
          IE=11
          WRITE (6,999) IE
          DX=DX/(NX-2)
          CY=DY/(NY-2)
          CALL SETSMG(Z,14,0.)
          IE=5
          WRITE (6,999) IE
          DO 221 I=2,NXX
          DO 221 J=2,NYY
          IF (RSY(I,J) .EC. 0.) GO TO 221
          NN=DR(1,I,J)/0.05CCOF-C7
          IF (NN .GT. 17) NN=17
          PX(1)=XAX(I,J)
          PY(1)=YAY(I,J)
          PX(2)=XAX(I,J)
          PY(2)=YAY(I,J)+DY/2.22
```

```
      PX(3)=XAX(I,J)
      PY(3)=YAY(I,J)-DY/2.22
      PX(4)=XAX(I,J)+DX/2.22
      PY(4)=PY(1)
      PX(5)=PX(4)
      PY(5)=PY(2)
      PX(6)=PX(4)
      PY(6)=PY(3)
      PX(7)=XAX(I,J)-DX/2.22
      PY(7)=PY(1)
      PX(8)=PX(7)
      PY(8)=PY(2)
      PX(9)=PX(7)
      PY(9)=PY(3)
      PX(10)=PX(1)+DX/4.6
      PY(10)=PY(1)+DY/4.6
      PX(11)=PX(10)
      PY(11)=PY(1)-DY/4.6
      PX(12)=PX(1)-DX/4.6
      PY(12)=PY(10)
      PX(13)=PX(12)
      PY(13)=PY(11)
      PX(14)=PX(4)-DX/4.6
      PY(14)=PY(10)
      PX(15)=PX(14)
      PY(15)=PY(11)
      PX(16)=PX(7)+DX/4.6
      PY(16)=PY(12)
      PX(17)=PX(16)
      PY(17)=PY(13)
      DO 222 NPT=1,NN
      CALL POINTG(Z,1,PX(NPT),PY(NPT))
  222 CONTINUE
  221 CONTINUE
      YEN=0.1+YCIS
      CALL OBJCTG(Z,C11,C,2,1.3,YEN)
      IE=1
      WRITE (6,999) IE
      DX=XMAX-XMIN
      DY=YMAX-YMIN
      CALL GRIDG(Z,DX,DY,0,0)
      IE=3
      WRITE (6,999) IE
      CALL TITLEG(Z,8,8HVELOCITY,8,LY,47,47H NUMERICAL SHOCK WAVES SIMUL
     1ATIONS AT COLUMBIA )
      IE=4
      WRITE (6,999) IE
      DX=DX/(NX-2)
      DY=DY/(NY-2)
      CALL SETSMG(Z,30,C.5)
      IE=7
      WRITE (6,999) IE
      DO 312 I=2,NXX
      DO 312 J=2,NYY
      IF (RSY(I,J) .EQ. 0.) GO TO 312
      CALL SEGMTG(Z,1,XAX(I,J),YAY(I,J),XPX(I,J),YPY(I,J))
```



```
      CALL POINTG(Z,1,XAX(I,J),YAY(I,J))
  312 CONTINUE
      CALL PAGEG(Z,0,1,1)
      IE=230
      WRITE (6,999) IE
      CALL RSETMG(Z)
      IE=31
      WRITE (6,999) IE
      RETURN
  100 FORMAT ('0ERROR, IPCS= ',I2)
  101 FORMAT('0ERROR,NCNT= ',I5)
  102 FORMAT('0ERROR, IPCS=2,BUT LESS THAN TWO PCS =1')
  103 FORMAT('0I',10X,'CNT(I)'/(I5,5X,1PE12.3))
  104 FORMAT('0IX= ',I5,' JY= ',I5/' X= ',1PE12.3,' Y= ',1PE12.3)
  105 FORMAT('0A= ',1PE12.3,' B= ',1PE12.3,' C= ',1PE12.3,' D= ',1PE12.3)
     1)
  106 FORMAT('02 CORNERS'/' X1= ',1PE12.3,' Y1= ',1PE12.3,' X2= ',1PE12.
     13,' Y2= ',1PE12.3)
  107 FORMAT('02 SIDES'/' XPT',10X,'YPT'/(1P2E12.3))
  108 FORMAT('0FOUR SIDES'/' XPT',10X,'YPT',10X,'XPT1',10X,'YPT1'/
     1(1P4E12.3))
  109 FORMAT('0ONE SIDE,ONE CORNER'/' X1= ',1PE12.3,' Y1= ',1PE12.3,
     1' X2= ',1PE12.3,' Y2= ',1PE12.3)
  999 FORMAT (1H ,'IE=',I4)
      END
```

```
0001        SUBROUTINE DIVIDE(DX,DY,FK,FR,CV,JP,JE,IS,IE,IP,JP,IC)
0002        DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
           1FE(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BB(2,137,22),
           2GU(1,137,22),GV(1,137,22),GI(1,137,22),GF(1,137,22),TMP(2,137,22),
           3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPB(22),BV(137,22),
           4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
           5ZSX(137,22)
0003        COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,GU,GV,GI,GF,TMP,SYA,SXB,SYB,VOL,
           1SI,CPB,BV,CSX,DSX,CSY,DSY,RSY, ZSX,C1,C2,C3,C4,P,I,J
0004        DO 744 J=JP,JE
0005        DO 744 I=IS,IE
0006        IF (ZSX(I,J).EQ.0..AND.RSY(I,J).EQ.0.) GO TO 744
0007        IF (RSY(I,J).EQ.DX.AND.ZSX(I,J).EQ.DX.AND.VOL(I,J).EQ.VOL(2,J))
           1 GO TO 744
0008        L=0
0009        VY=0.
0010        VZ=0.
0011        IF (DR(1,I+1,J) .EQ. 0.) L=-1
0012        IF (DR(1,I-1,J) .EQ. 0.) L=+1
0013        IF (L .EQ. 0) GO TO 744
0014        IF (ZSX(I,J) .GT. 1.5*DX .AND. VOL(I,J) .GT. VOL(2,J) .AND.
           1VOL(I,J) .GT. 1.4*RSY(I,J)*SYB(J)) GO TO 761
0015        IF (RSY(I-L,J+1) .GT. 1.5*DX .AND. VOL(I-L,J+1) .GE. VOL(2,J+1)
           1.AND. VOL(I-L,J+1) .GT. 1.4*ZSX(I-L,J+1)*SYB(J+1)) GO TO 762
0016        IF (RSY(I-L,J-1) .GT. 1.5*DX .AND. VOL(I-L,J-1) .GE. VOL(2,J-1)
           1.AND. VOL(I-L,J-1) .GT. 1.4*ZSX(I-L,J-1)*SYB(J-1)) GO TO 762
0017        GO TO 744
0018    762 CONTINUE
0019        IF (ZSX(I,J) .GT. DX .AND. VOL(I,J) .GT. RSY(I,J)*SYB(J)) GO TO861
0020        GO TO 744
0021    761 IF (ZSX(I,J+1).GT.1.5*DX.OR.ZSX(I,J-1).GT.1.5*DX) GO TO 861
0022        IF (DR(1,I-L,J+1).EQ.0..AND.DR(1,I-L,J-1).EQ.0.) GO TO 744
0023    861 CONTINUE
0024        VX=VOL(I,J)-RSY(I,J)*SYB(J)
0025        IF (VOL(I-L,J+1) .GT. ZSX(I-L,J+1)*SYB(J+1) .AND. RSY(I-L,J+1)
           1.GT. DY) VY=VOL(I-L,J+1)-ZSX(I-L,J+1)*SYB(J+1)
0026        IF (VOL(I-L,J-1) .GT. ZSX(I-L,J-1)*SYB(J-1) .AND. RSY(I-L,J-1)
           1.GT. DY) VZ=VOL(I-L,J-1)-ZSX(I-L,J-1)*SYB(J-1)
0027        VOL(I-L,J)=VX+VY+VZ
0028        IF (VY.EQ.0..AND.VZ.EQ.0.) GO TO 881
0029        GO TO 882
0030    881 DSX(I-L,J)=DSX(I,J)-DX
0031        CSX(I-L,J)=CSX(I,J)-DX
0032        ZSX(I-L,J)=AMAX1(DSX(I-L,J),CSX(I-L,J))
0033        GO TO 981
0034    882 IF (VY.GT.0..AND.VZ.EQ.0.) GO TO 883
0035        GO TO 884
0036    883 CSX(I-L,J)=ZSX(I-L,J+1)
0037        DSX(I-L,J)=ZSX(I-L,J+1)
0038        ZSX(I-L,J)=ZSX(I-L,J+1)
0039        VOL(I-L,J+1)=VOL(I-L,J+1)-VY
0040        DSY(I-L,J+1)=DY
0041        CSY(I-L,J+1)=DY
0042        RSY(I-L,J+1)=DY
0043        CSX(I-L,J+1)=ZSX(I-L,J+1)
0044        DSX(I-L,J+1)=ZSX(I-L,J+1)
```



```
0045        GO TO 981
0046    884 IF (VY.EQ.0..AND.VZ.GT.0.) GO TO 885
0047        GO TO 886
0048    885 CSX(I-L,J)=ZSX(I-L,J-1)
0049        DSX(I-L,J)=ZSX(I-L,J-1)
0050        ZSX(I-L,J)=ZSX(I-L,J-1)
0051        VOL(I-L,J-1)=VOL(I-L,J-1)-VZ
0052        CSY(I-L,J-1)=DY
0053        DSY(I-L,J-1)=DY
0054        RSY(I-L,J-1)=DY
0055        CSX(I-L,J-1)=ZSX(I-L,J-1)
0056        DSX(I-L,J-1)=ZSX(I-L,J-1)
0057        GO TO 981
0058    886 IF (VY.GT.0..AND.VZ.GT.0.) GO TO 887
0059        GO TO 981
0060    887 ZSX(I-L,J)=VOL(I-L,J)/SYB(J)
0061        CSX(I-L,J)=ZSX(I-L,J)
0062        DSX(I-L,J)=ZSX(I-L,J)
0063        RSY(I-L,J+1)=DY
0064        CSY(I-L,J+1)=DY
0065        DSY(I-L,J+1)=DY
0066        VOL(I-L,J+1)=VOL(I-L,J+1)-VY
0067        DSY(I-L,J-1)=DY
0068        RSY(I-L,J-1)=DY
0069        CSY(I-L,J-1)=DY
0070        VOL(I-L,J-1)=VOL(I-L,J-1)-VZ
0071        CSX(I-L,J+1)=ZSX(I-L,J+1)
0072        DSX(I-L,J+1)=ZSX(I-L,J+1)
0073        CSX(I-L,J-1)=ZSX(I-L,J-1)
0074        DSX(I-L,J-1)=ZSX(I-L,J-1)
0075    981 CONTINUE
0076        CSY(I,J)=RSY(I,J)
0077        DSY(I,J)=RSY(I,J)
0078        VOL(I,J)=VOL(I,J)-VX
0079        DSX(I,J)=DX
0080        CSX(I,J)=DX
0081        ZSX(I,J)=DX
0082        RSY(I-L,J)=VOL(I-L,J)*DX/ZSX(I-L,J)/SXB(J)
0083        CSY(I-L,J)=RSY(I-L,J)
0084        DSY(I-L,J)=RSY(I-L,J)
0085        DR(1,I-L,J)=DR(1,I,J)*VX+DR(1,I-L,J+1)*VY+DR(1,I-L,J-1)*VZ
0086        BB(1,I-L,J)=BB(1,I,J)*VX+BB(1,I-L,J+1)*VY*SYB(J)/SYB(J+1)+
           1BB(1,I-L,J-1)*VZ*SYB(J)/SYB(J-1)
0087        FE(1,I-L,J)=DR(1,I,J)*VX*FE(1,I,J)+DR(1,I-L,J+1)*VY*FE(1,I-L,J+1)
           1+DR(1,I-L,J-1)*VZ*FE(1,I-L,J-1)
0088        FU(1,I-L,J)=DR(1,I,J)*VX*FU(1,I,J)+DR(1,I-L,J+1)*VY*FU(1,I-L,J+1)
           1+DR(1,I-L,J-1)*VZ*FU(1,I-L,J-1)
0089        FV(1,I-L,J)=DR(1,I,J)*VX*FV(1,I,J)+DR(1,I-L,J+1)*VY*FV(1,I-L,J+1)
           1+DR(1,I-L,J-1)*VZ*FV(1,I-L,J-1)
0090        FE(1,I-L,J)=FE(1,I-L,J)/DR(1,I-L,J)
0091        FU(1,I-L,J)=FU(1,I-L,J)/DR(1,I-L,J)
0092        FV(1,I-L,J)=FV(1,I-L,J)/DR(1,I-L,J)
0093        DR(1,I-L,J)=DR(1,I-L,J)/VOL(I-L,J)
0094        BB(1,I-L,J)=BB(1,I-L,J)/VOL(I-L,J)
0095        FI(1,I-L,J)=FE(1,I-L,J)-.5*(FU(1,I-L,J)**2+FV(1,I-L,J)**2)
0096        TC(1,I-L,J)=(FK*FR*FI(1,I-L,J)/CV)**.5
```

```
      P(1,I-L,J)=(FK-1.)*DR(1,I-L,J)*FI(1,I-L,J)
      PP(1,I-L,J)=P(1,I-L,J)+PP(1,I-L,J)**2/25.1328
      TMP(1,I-L,J)=FI(1,I-L,J)/CV
      BV(I-L,J)=0.
      SI(J)=SI(J)-1.
  744 CONTINUE
      DO 748 J=JB,JE
      DO 748 I=IS,IE
      IF (ZSX(I,J).EQ.O..ANC.RSY(I,J).EQ.C.) GO TO 748
      IF (RSY(I,J).EQ.DX.AND.ZSX(I,J).EQ.CX.AND.VOL(I,J).EQ.VOL(2,J))
     1 GO TO 748
      L=0
      IF (DR(1,I,J+1) .EC. 0.) L=-1
      IF (DR(1,I,J-1) .EC. 0.) L=+1
      IF (L .EC. 0) GO TC 748
      IF (RSY(I,J) .GT. 1.5*DY .AND. VCL(I,J) .GE. VCL(2,J) .AND.
     1VOL(I,J) .GT. 1.4*ZSX(I,J)*SYB(J)) GO TO 763
      GC TO 748
  763 IF (RSY(I+1,J) .GT. 1.5*DY .CR.  RSY(I-1,J) .GT. 1.5*DY) GO TO 764
      IF (DR(1,I+1,J-L).EC.0..AND.DR(1,I-1,J-L).EQ.0.) GO TO 748
  764 CONTINUE
      CSX(I,J-L)=ZSX(I,J)
      DSX(I,J-L)=ZSX(I,J)
      ZSX(I,J-L)=ZSX(I,J)
      VOL(I,J-L)=VCL(I,J)-ZSX(I,J)*SYB(J)
      RSY(I,J-L)=VCL(I,J-L)*CX/SXB(J-L)/ZSX(I,J-L)
      CSY(I,J-L)=RSY(I,J-L)
      DSY(I,J-L)=RSY(I,J-L)
      VOL(I,J)=ZSX(I,J)*SYB(J)
      CSY(I,J)=DY
      DSY(I,J)=DY
      RSY(I,J)=DY
      CSX(I,J)=ZSX(I,J)
      DSX(I,J)=ZSX(I,J)
      FE(1,I,J-L)=FE(1,I,J)
      FU(1,I,J-L)=FU(1,I,J)
      FV(1,I,J-L)=FV(1,I,J)
      FI(1,I,J-L)=FI(1,I,J)
      DR(1,I,J-L)=CR(1,I,J)
      TC(1,I,J-L)=TC(1,I,J)
      P (1,I,J-L)= P(1,I,J)
      BB(1,I,J-L)=BB(1,I,J)*SYB(J-L)/SYB(J)
      PP(1,I,J-L)=P(1,I,J-L)+BB(1,I,J-L)**2/25.1328
      TMP(1,I,J-L)=TMP(1,I,J)
      BV(I,J-L)=0.
      SI(J-L)=SI(J-L)-1.
  748 CONTINUE
      RETURN
      END
```



```
      SUBROUTINE DISTAN(I,J,CA,CB,DEL,DX)
      RA=5.
      U=1.
      XIM1=92.
      XJM=2.
      XI=I
      XJ=J
      DA=(XI-XIM1+.5)*DX
      DB=(XJ-XJM+.5)*DX
      DAB=SQRT(CA**2+DB**2)
      ZN=ATAN(DA/CB)
      DS=RA*SCRT(U/(COS(ZN)**2+(SIN(ZN)/2.25)**2))
      DEL=CAB-DS
      RETURN
      END
```

```
0001    SUBROUTINE RIGHT(DY,FK,FR,CV,IW,RMEAN)
0002    DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
       1FE(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BB(2,137,22),
       2GU(1,137,22),GV(1,137,22),GI(1,137,22),GE(1,137,22),TMP(2,137,22),
       3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPB(22),RV(137,22),
       4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
       5ZSX(137,22)
0003    COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,GU,GV,GI,GE,TMP,SYA,SXB,SYB,VOL,
       1SI,CPB,RV,CSX,DSX,CSY,DSY,RSY, ZSX,Q1,Q2,Q3,Q4,N,I,J
0004    L=0
0005    IF (ZSX(I-1,J).GT..1E-02) L=-1
0006    IF (ZSX(I+1,J).GT..1E-02) L=+1
0007    IF (L .EQ. 0) GO TO 1
0008    IF ((I+L) .GT. IW) GO TO 1
0009    VS=0.
0010    K=0
0011    IF (DR(1,I,J+1) .EQ. 0.) K=-1
0012    IF (DR(1,I,J-1) .EQ. 0.) K =+1
0013    IF (K .EQ. 0 .OR. (DR(1,I,J+1).EQ. 0..AND.DR(1,I,J-1).EQ.0.)) GO
       1 TO 2
0014    IF (RSY(I,J).GT.RSY(I+L,J).AND. ZSX(I+L,J-K).GT.0..AND.VOL(I,J)
       1.GT.RSY(I+L,J)*(CSX(I,J)+DSX(I,J))*SXB(J)/DY/2.) VS=VOL(I,J)-
       2RSY(I+L,J)*(CSX(I,J)+DSX(I,J))*SXB(J)/DY/2.
0015    VOL(I,J)=VOL(I,J)-VS
0016  2 CONTINUE
0017    FU(1,I +L,J)=(VOL(I ,J)*DR(1,I ,J)*FU(1,I ,J)+VOL(I +L,J)*
       1DR(1,I +L,J)*FU(1,I +L,J))/(VOL(I ,J)+VOL(I +L,J))
0018    FV(1,I +L,J)=(VOL(I ,J)*DR(1,I ,J)*FV(1,I ,J)+VOL(I +L,J)*
       1DR(1,I +L,J)*FV(1,I +L,J))/(VOL(I ,J)+VOL(I +L,J))
0019    FE(1,I +L,J)=(VOL(I ,J)*DR(1,I ,J)*FE(1,I ,J)+VOL(I +L,J)*
       1DR(1,I +L,J)*FE(1,I +L,J))/(VOL(I ,J)+VOL(I +L,J))
0020    DR(1,I +L,J)=(VOL(I ,J)*DR(1,I ,J)+VOL(I +L,J)*DR(1,I +L,J))
       1/(VOL(I,J)+VOL(I+L,J))
0021    BB(1,I +L,J)=(VOL(I ,J)*BB(1,I ,J)+VOL(I +L,J)*BB(1,I +L,J))
       1/(VOL(I,J)+VOL(I+L,J))
0022    FU(1,I+L,J)=FU(1,I+L,J)/DR(1,I+L,J)
0023    FV(1,I+L,J)=FV(1,I+L,J)/DR(1,I+L,J)
0024    FE(1,I+L,J)=FE(1,I+L,J)/DR(1,I+L,J)
0025    FI(1,I+L,J)=FE(1,I+L,J)-.5*(FU(1,I+L,J)**2+FV(1,I+L,J)**2)
0026    TC(1,I+L,J)=(FK*FR*FI(1,I+L,J)/CV)**.5
0027    P (1,I+L,J)=(FK-1.)*DR(1,I+L,J)*FI(1,I+L,J)
0028    PP(1,I+L,J)=P (1,I+L,J)+BB(1,I+L,J)**2/25.1328
0029    TMP(1,I+L,J)=FI(1,I+L,J)/CV
0030    VOL(I+L,J)=VOL(I,J)+VOL(I+L,J)
0031    ZSX(I+L,J)=VOL(I+L,J)*DY/SYB(J)/RSY(I+L,J)
0032    CSX(I+L,J)=ZSX(I+L,J)
0033    DSX(I+L,J)=ZSX(I+L,J)
0034    SI(J)=SI(J)+1.
0035    IF (VS .LE. 0. .OR. K .EQ. 0) GO TO 11
0036    FU(1,I+L,J-K)=(VS*DR(1,I,J)*FU(1,I,J)+VOL(I+L,J-K)*DR(1,I+L,J-K)
       1*FU(1,I+L,J-K))/(VOL(I+L,J-K)+VS)
0037    FV(1,I+L,J-K)=(VS*DR(1,I,J)*FV(1,I,J)+VOL(I+L,J-K)*DR(1,I+L,J-K)
       1*FV(1,I+L,J-K))/(VOL(I+L,J-K)+VS)
0038    FE(1,I+L,J-K)=(VS*DR(1,I,J)*FE(1,I,J)+VOL(I+L,J-K)*DR(1,I+L,J-K)
       1*FE(1,I+L,J-K))/(VOL(I+L,J-K)+VS)
0039    DR(1,I+L,J-K)=(VS*DR(1,I,J)+VOL(I+L,J-K)*DR(1,I+L,J-K))/
```



```
       1(VOL(I+L,J-K)+VS)
0040    BB(1,I+L,J-K)=(VS*BB(1,I,J)*SXB(J-K)/SXB(J)+VOL(I+L,J-K)*
       1BB(1,I+L,J-K))/(VOL(I+L,J-K)+VS)
0041    FU(1,I+L,J-K)=FU(1,I+L,J-K)/DR(1,I+L,J-K)
0042    FV(1,I+L,J-K)=FV(1,I+L,J-K)/DR(1,I+L,J-K)
0043    FE(1,I+L,J-K)=FE(1,I+L,J-K)/DR(1,I+L,J-K)
0044    FI(1,I+L,J-K)=FE(1,I+L,J-K)-.5*(FU(1,I+L,J-K)**2+FV(1,I+L,J-K)**2)
0045    TC(1,I+L,J-K)=(FK*FR*FI(1,I+L,J-K)/CV)**.5
0046    P(1,I+L,J-K)=(FK-1.)*DR(1,I+L,J-K)*FI(1,I+L,J-K)
0047    PP(1,I+L,J-K)=P(1,I+L,J-K)+BB(1,I+L,J-K)**2/25.1328
0048    TMP(1,I+L,J-K)=FI(1,I+L,J-K)/CV
0049    VOL(I+L,J-K)=VOL(I+L,J-K)+VS
0050    ZSX(I+L,J-K)=VOL(I+L,J-K)*DY/SXB(J-K)/RSY(I+L,J-K)
0051    CSX(I+L,J-K)=ZSX(I+L,J-K)
0052    DSX(I+L,J-K)=ZSX(I+L,J-K)
0053 11 CALL  VACUUM(RMEAN,DY)
0054  1 RETURN
0055    END
```

```
0001      SUBROUTINE UPPER(DX,FK,FR,CV,JB,JF,RMEAN)
0002      DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
         1FE(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BB(2,137,22),
         2CU(1,137,22),CV(1,137,22),GI(1,137,22),GL(1,137,22),TMP(2,137,22),
         3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPR(22),BV(137,22),
         4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
         5ZSX(137,22)
0003      COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,CU,CW,GI,GL,TMP,SYA,SXB,SYB,VOL,
         1SI,CPR,BV,CSX,DSX,CSY,DSY,RSY, ZSX,Q1,Q2,Q3,Q4,M,I,J
0004      L=0
0005      IF (RSY(I,J-1).GT..1E-02) L=-1
0006      IF (RSY(I,J+1).GT..1E-02) L=+1
0007      IF (L .EQ. 0) GO TO 1
0008      IF ((J+L) .GT. JF) GO TO 1
0009      IF ((J+L) .LT. JB) GO TO 1
0010      VS=0.
0011      K=0
0012      IF (DR(1,I+1,J) .EQ. 0.) K=-1
0013      IF (DR(1,I-1,J) .EQ. 0.) K=+1
0014      IF (K .EQ. 0 .OR. (DR(1,I+1,J).EQ.0..AND.DR(1,I-1,J).EQ.0.))GO TO2
0015      IF (ZSX(I,J).GT.ZSX(I,J+L).AND. RSY(I-K,J+L).GT.0..AND.VOL(I,J)
         1.GT.ZSX(I,J+L)*(CSY(I,J)+DSY(I,J))*SXB(J)/DX/2.) VS=VOL(I,J)-
         1ZSX(I,J+L)*(CSY(I,J)+DSY(I,J))*SXB(J)/DX/2.
0016      VOL(I,J)=VOL(I,J)-VS
0017    2 CONTINUE
0018      FU(1,I,J+L)=(VOL(I,J)*DR(1,I,J)*FU(1,I,J)+VOL(I,J+L)*DR(1,I,J+L)
         1 *FU(1,I,J+L))/(VOL(I,J)+VOL(I,J+L))
0019      FV(1,I,J+L)=(VOL(I,J)*DR(1,I,J)*FV(1,I,J)+VOL(I,J+L)*DR(1,I,J+L)
         1 *FV(1,I,J+L))/(VOL(I,J)+VOL(I,J+L))
0020      FE(1,I,J+L)=(VOL(I,J)*DR(1,I,J)*FE(1,I,J)+VOL(I,J+L)*DR(1,I,J+L)
         1 *FE(1,I,J+L))/(VOL(I,J)+VOL(I,J+L))
0021      DR(1,I,J+L)=(VOL(I,J)*DR(1,I,J)+VOL(I,J+L)*DR(1,I,J+L))/
         1(VOL(I,J)+VOL(I,J+L))
0022      BB(1,I,J+L)=(VOL(I,J)*SXB(J+L)/SXB(J)*BB(1,I,J)+VOL(I,J+L)*
         1BB(1,I,J+L))/(VOL(I,J)+VOL(I,J+L))
0023      FU(1,I,J+L)=FU(1,I,J+L)/DR(1,I,J+L)
0024      FV(1,I,J+L)=FV(1,I,J+L)/DR(1,I,J+L)
0025      FE(1,I,J+L)=FE(1,I,J+L)/DR(1,I,J+L)
0026      FI(1,I,J+L)=FE(1,I,J+L)-.5*(FU(1,I,J+L)**2+FV(1,I,J+L)**2)
0027      TC(1,I,J+L)=(FK*FR*FI(1,I,J+L)/CV)**.5
0028      P (1,I,J+L)=(FK-1.)*DR(1,I,J+L)*FI(1,I,J+L)
0029      PP(1,I,J+L)=P (1,I,J+L)+BB(1,I,J+L)**2/25.1328
0030      TMP(1,I,J+L)=FI(1,I,J+L)/CV
0031      VOL(I,J+L)=VOL(I,J)+VOL(I,J+L)
0032      RSY(I,J+L)=VOL(I,J+L)*DX/SXB(J+L)/ZSX(I,J+L)
0033      CSY(I,J+L)=RSY(I,J+L)
0034      DSY(I,J+L)=RSY(I,J+L)
0035      SI(J)=SI(J)+1.
0036      IF (VS .LE. 0. .OR. K .EQ. 0) GO TO 11
0037      FU(1,I-K,J+L)=(VS*DR(1,I,J)*FU(1,I,J)+VOL(I-K,J+L)*DR(1,I-K,J+L)
         1*FU(1,I-K,J+L))/(VOL(I-K,J+L)+VS)
0038      FV(1,I-K,J+L)=(VS*DR(1,I,J)*FV(1,I,J)+VOL(I-K,J+L)*DR(1,I-K,J+L)
         1*FV(1,I-K,J+L))/(VOL(I-K,J+L)+VS)
0039      FE(1,I-K,J+L)=(VS*DR(1,I,J)*FE(1,I,J)+VOL(I-K,J+L)*DR(1,I-K,J+L)
         1*FE(1,I-K,J+L))/(VOL(I-K,J+L)+VS)
0040      DR(1,I-K,J+L)=(VS*DR(1,I,J)+VOL(I-K,J+L)*DR(1,I-K,J+L))/
```



```
         1(VOL(I-K,J+L)+VS)
0041      BB(1,I-K,J+L)=(VS*BB(1,I,J)*SXB(J+L)/SXB(J)+VOL(I-K,J+L)*
         1BB(1,I-K,J+L))/(VOL(I-K,J+L)+VS)
0042      FU(1,I-K,J+L)=FU(1,I-K,J+L)/DR(1,I-K,J+L)
0043      FV(1,I-K,J+L)=FV(1,I-K,J+L)/DR(1,I-K,J+L)
0044      FE(1,I-K,J+L)=FE(1,I-K,J+L)/DR(1,I-K,J+L)
0045      FI(1,I-K,J+L)=FE(1,I-K,J+L)-.5*(FU(1,I-K,J+L)**2+FV(1,I-K,J+L)**2)
0046      TC(1,I-K,J+L)=(FK*FR*FI(1,I-K,J+L)/CV)**.5
0047      P(1,I-K,J+L)=(FK-1.)*DR(1,I-K,J+L)*FI(1,I-K,J+L)
0048      PP(1,I-K,J+L)=P(1,I-K,J+L)+BB(1,I-K,J+L)**2/25.1328
0049      TMP(1,I-K,J+L)=FI(1,I-K,J+L)/CV
0050      VOL(I-K,J+L)=VOL(I-K,J+L)+VS
0051      RSY(I-K,J+L)=VOL(I-K,J+L)*DX/SXB(J+L)/ZSX(I-K,J+L)
0052      CSY(I-K,J+L)=RSY(I-K,J+L)
0053      DSY(I-K,J+L)=RSY(I-K,J+L)
0054   11 CALL  VACUUM(RMEAN,DX)
0055    1 RETURN
0056      END
```

```
      SUBROUTINE TRANSP(DX,DY,DT,JB,JE,IF,EPSI,IS,EMEAN,IM,JM)
      DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
     1FE(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BB(2,137,22),
     2GU(1,137,22),GV(1,137,22),GI(1,137,22),GL(1,137,22),TMP(2,137,22),
     3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPB(22),BV(137,22),
     4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
     5ZSX(137,22)
      COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,GU,GV,GI,GE,TMP,SYA,SXB,SYB,VOL,
     1SI,CPB,BV,CSX,DSX,CSY,DSY,RSY, ZSX,Q1,Q2,Q3,Q4,M,I,J
      DO 25 J=JB,JE
      DO 25 I=IS,IE
      IF (ZSX(I,J) .EQ. 0. .AND. RSY(I,J) .EQ. 0.) GO TO 25
      SF(J=VOL(I,J)/VOL(I,J)
      IF ((RSY(I,J).EQ.RSY(I+1,J)).AND.(RSY(I,J).EQ.RSY(I-1,J)).AND.
     1(ZSX(I,J).EQ.ZSX(I,J+1)).AND.(ZSX(I,J).EQ.ZSX(I,J-1))) GO TO 6
      IF (DR(1,I-1,J+1).GT.0..AND.DR(1,I-1,J-1).GT.0.) GO TO 153
      IF (CSY(I-1,J).GT.0..AND.CSY(I-1,J).LT.DSY(I,J).AND.
     1(GU(1,I,J)+GU(1,I-1,J)).LT.0..AND.VOL(I,J).GE.RSY(I,J)*SYB(J))
     2GO TO 9571
      GO TO 153
 9571 CSY(I-1,J)=DSY(I,J)
      DSY(I-1,J)=DSY(I,J)
      RSY(I-1,J)=DSY(I,J)
  153 CONTINUE
      IF (DR(1,I+1,J+1).GT.0..AND.DR(1,I+1,J-1).GT.0.) GO TO 155
      IF (DSY(I+1,J).GT.0..AND.DSY(I+1,J).LT.CSY(I,J).AND.
     1(GU(1,I,J)+GU(1,I+1,J)).GT.0..AND.VOL(I,J).GE.RSY(I,J)*SYB(J))
     2GO TO 9573
      GO TO 155
 9573 DSY(I+1,J)=CSY(I,J)
      CSY(I+1,J)=CSY(I,J)
      RSY(I+1,J)=CSY(I,J)
  155 CONTINUE
      IF (DR(1,I-1,J-1) .GT. 0. .AND. DR(1,I+1,J-1) .GT. 0.) GO TO 152
      IF (CSX(I,J-1).GT.0..AND.CSX(I,J-1).LT.DSX(I,J).AND.
     1(GV(1,I,J)+GV(1,I,J-1)).LT.0..AND.VOL(I,J).GE.ZSX(I,J)*SYB(J))
     2GO TO 9575
      GO TO 152
 9575 CSX(I,J-1)=DSX(I,J)
      DSX(I,J-1)=DSX(I,J)
      ZSX(I,J-1)=DSX(I,J)
  152 CONTINUE
      IF (DR(1,I-1,J+1) .GT. 0. .AND. DR(1,I+1,J+1) .GT. 0.) GO TO 156
      IF (CSX(I,J+1).GT.0..AND.DSX(I,J+1).LT.CSX(I,J).AND.
     1(GV(1,I,J)+GV(1,I,J+1)).GT.0..AND.VOL(I,J).GE.ZSX(I,J)*SYB(J))
     2GO TO 9577
      GO TO 156
 9577 DSX(I,J+1)=CSX(I,J)
      ZSX(I,J+1)=CSX(I,J)
      CSX(I,J+1)=CSX(I,J)
  156 CONTINUE
    6 CONTINUE
      AA=AMIN1(CSY(I,J),DSY(I+1,J))/DX
      AB=AMIN1(DSY(I,J),CSY(I-1,J))/DX
      AC=AMIN1(CSX(I,J),DSX(I,J+1))/DX
      AD=AMIN1(DSX(I,J),CSX(I,J-1))/DX
```



```
      TTR=0.
      TTL=0.
      TTT=0.
      TTG=0.
      TTA=0.
      TTB=0.
      TTC=0.
      TTD=0.
      TT1=0.
      TT2=0.
      TT3=0.
      TT4=0.
      DMR=0.
      DML=0.
      DMT=0.
      DMG=0.
      DBR=0.
      DBL=0.
      DBT=0.
      DBG=0.
      DM1=0.
      DM2=0.
      DM3=0.
      DM4=0.
      DMA=0.
      DMB=0.
      DMC=0.
      DMD=0.
      DB1=0.
      DB2=0.
      DB3=0.
      DB4=0.
      DBA=0.
      DBB=0.
      DBC=0.
      DBD=0.
      IF (.5*(GU(1,I,J)+GU(1,I+1,J)) .GT. 0.) DM1=SXB(J)*DR(1,I,J)*.5*
     1 (GU(1,I,J)+GU(1,I+1,J))*DT*AA
      IF (.5*(GU(1,I,J)+GU(1,I+1,J)) .LT. 0.) DM1=SXB(J)*DR(1,I+1,J)*.5*
     1 (GU(1,I,J)+GU(1,I+1,J))*DT*AA
      IF (.5*(GV(1,I,J)+GV(1,I,J+1)) .GT. 0.) DM3=SYA(J)*DR(1,I,J)*.5*
     1 (GV(1,I,J)+GV(1,I,J+1))*DT*AC
      IF (.5*(GV(1,I,J)+GV(1,I,J+1)) .LT. 0.) DM3=SYA(J)*DR(1,I,J+1)*.5*
     1 (GV(1,I,J)+GV(1,I,J+1))*DT *AC
      IF (.5*(GU(1,I,J)+GU(1,I-1,J)) .GT. 0.) DM2=SXB(J)*DR(1,I-1,J)*
     1 (GU(1,I,J)+GU(1,I-1,J))*DT *.5 *AB
      IF (.5*(GU(1,I,J)+GU(1,I-1,J)) .LT. 0.) DM2=SXB(J)*DR(1,I,J)*
     1 (GU(1,I,J)+GU(1,I-1,J))*DT *.5 *AB
      IF (.5*(GV(1,I,J)+GV(1,I,J-1)) .GT. 0.) DM4=SYA(J-1)*DR(1,I,J-1)*
     1.5*(GV(1,I,J)+GV(1,I,J-1))*DT *AD
      IF (.5*(GV(1,I,J)+GV(1,I,J-1)) .LT. 0.) DM4=SYA(J-1)*DR(1,I,J)*
     1.5*(GV(1,I,J)+GV(1,I,J-1))*DT *AD
      IF (.5*(GU(1,I,J)+GU(1,I+1,J)) .GT. 0.) DB1=SXB(J)*BB(1,I,J)*.5*
     1 (GU(1,I,J)+GU(1,I+1,J))*DT*2.*3.1416*DX/SXB(J)*AA
      IF (.5*(GU(1,I,J)+GU(1,I+1,J)) .LT. 0.) DB1=SXB(J)*BB(1,I+1,J)*.5*
     1 (GU(1,I,J)+GU(1,I+1,J))*DT*2.*3.1416*DX/SXB(J)*AA
```

```
0088           IF (.5*(GV(1,I,J)+GV(1,I,J+1)) .GT. 0.) DB3=SYA(J)*BB(1,I,J)*.5*
              1 (GV(1,I,J)+GV(1,I,J+1))*DT*2.*3.1416*DX/SXB(J)*AC
0089           IF (.5*(GV(1,I,J)+GV(1,I,J+1)) .LT. 0.) DB3=SYA(J)*BB(1,I,J+1)*.5*
              1 (GV(1,I,J)+GV(1,I,J+1))*DT*2.*3.1416*DX/SXB(J+1)*AC
0090           IF (.5*(GU(1,I,J)+GU(1,I-1,J)) .GT. 0.) DB2=SXB(J)*BB(1,I-1,J)*
              1 (GU(1,I,J)+GU(1,I-1,J))*DT *.5 *2.*3.1416*DX/SXB(J)  *AB
0091           IF (.5*(GU(1,I,J)+GU(1,I-1,J)) .LT. 0.) DB2=SXB(J)*BB(1,I,J)*
              1 (GU(1,I,J)+GU(1,I-1,J))*DT *.5 *2.*3.1416*DX/SXB(J)  *AB
0092           IF (.5*(GV(1,I,J)+GV(1,I,J-1)) .GT. 0.) DB4=SYA(J-1)*BB(1,I,J-1)*
              1.5*(GV(1,I,J)+GV(1,I,J-1))*DT *2.*3.1416*DX/SXB(J-1) *AC
0093           IF (.5*(GV(1,I,J)+GV(1,I,J-1)) .LT. 0.) DB4=SYA(J-1)*BB(1,I,J)*
              1.5*(GV(1,I,J)+GV(1,I,J-1))*DT *2.*3.1416*DX/SXB(J)  *AD
0094           IF (.5*(GU(1,I,J)+GU(1,I+1,J)) .LT. 0.) TT1=1.
0095           IF (.5*(GU(1,I,J)+GU(1,I-1,J)) .GT. 0.) TT2=1.
0096           IF (.5*(GV(1,I,J)+GV(1,I,J+1)) .LT. 0.) TT3=1.
0097           IF (.5*(GV(1,I,J)+GV(1,I,J-1)) .GT. 0.) TT4=1.
0098           IF ((RSY(I,J).EQ.RSY(I+1,J)).AND.(RSY(I,J).EQ.RSY(I-1,J)).AND.
              1(ZSX(I,J).EQ.ZSX(I,J+1)).AND.(ZSX(I,J).EQ.ZSX(I,J-1))) GO TO 39
0099           IF (DR(1,I+1,J+1) .EQ. 0.) GO TO     9404
0100           IF (DR(1,I,J+1).EQ.0..AND. CSY(I,J).GT.DSY(I+1,J) .AND.(GU(1,I,J)
              1+GU(1,I+1,J+1)).GT.0.)  GO TO  9301
0101           GO TO 9302
0102      9301 DMR=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*(CSY(I,J)-
              1DSY(I+1,J))/DX
0103           DBR=BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*(CSY(I,J)-DSY(I+1,J)
              1 )*6.2832
0104           GO TO 9404
0105      9302 IF (DR(1,I,J+1).EQ.0..AND. CSY(I,J).GT.DSY(I+1,J).AND.(GU(1,I,J)
              1+GU(1,I+1,J+1)).LT.0.)  GO TO  9303
0106           GO TO 9404
0107      9303 DMR=SXB(J+1)*DR(1,I+1,J+1)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*
              1 (CSY(I,J)-DSY(I+1,J))/DX
0108           DBR=BB(1,I+1,J+1)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*(CSY(I,J)-
              1 DSY(I+1,J))*6.2832
0109           TTR=1.
0110      9404 CONTINUE
0111           IF (DR(1,I+1,J-1) .EQ. 0.) GO TO  9408
0112           IF (DR(1,I,J-1).EQ.0..AND. CSY(I,J).GT.DSY(I+1,J).AND.(GU(1,I,J)
              1+GU(1,I+1,J-1)).GT.0.)  GO TO  9405
0113           GO TO 9406
0114      9405 DMA=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*(CSY(I,J)-
              1 DSY(I+1,J))/DX
0115           DBA=BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*(CSY(I,J)-DSY(I+1,J)
              1 )*6.2832
0116           GO TO 9408
0117      9406 IF (DR(1,I,J-1).EQ.0..AND. CSY(I,J).GT.DSY(I+1,J).AND.(GU(1,I,J)
              1+GU(1,I+1,J-1)).LT.0.)  GO TO  9407
0118           GO TO 9408
0119      9407 DMA=SXB(J-1)*DR(1,I+1,J-1)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*
              1 (CSY(I,J)-DSY(I+1,J))/DX
0120           DBA=BB(1,I+1,J-1)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*(CSY(I,J)-
              1 DSY(I+1,J))*6.2832
0121           TTA=1.
0122      9408 CONTINUE
0123           IF (DR(1,I-1,J-1) .EQ. 0.) GO TO 9315
0124           IF (DR(1,I-1,J).EQ.0..AND.CSY(I-1,J-1).GT.DSY(I,J-1).AND.
```



```
              1(GU(1,I,J)+GU(1,I-1,J-1)).GT.0.) GO TO  9312
0125           GO TO 9313
0126      9312 DML=SXB(J-1)*DR(1,I-1,J-1)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*
              1 (CSY(I-1,J-1)-DSY(I,J-1))/DX
0127           DBL=BB(1,I-1,J-1)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*(CSY(I-1,J-1)-
              1 DSY(I,J-1))*6.2832
0128           TTL=1.
0129           GO TO 9315
0130      9313 IF (DR(1,I-1,J).EQ. 0..AND.CSY(I-1,J-1).GT.DSY(I,J-1).AND.
              1(GU(1,I,J)+GU(1,I-1,J-1)).LT.0.)  GO  TO  9314
0131           GO TO  9315
0132      9314 DML=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*(CSY(I-1,J-1)
              1-DSY(I,J-1))/DX
0133           DBL=BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*(CSY(I-1,J-1)-
              1DSY(I,J-1))*6.2832
0134      9315 CONTINUE
0135           IF (DR(1,I-1,J+1) .EQ.  0.)  GO TO  9319
0136           IF (DR(1,I-1,J).EQ.0..AND.CSY(I-1,J+1).GT.DSY(I,J+1).AND.
              1(GU(1,I,J)+GU(1,I-1,J+1)).GT.0.) GO TO 9316
0137           GO TO  9317
0138      9316 DMB=SXB(J+1)*DR(1,I-1,J+1)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*(
              1 CSY(I-1,J+1)-DSY(I,J+1))/DX
0139           DBB=BB(1,I-1,J+1)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*(CSY(I-1,J+1)-
              1DSY(I,J+1))*6.2832
0140           TTB=1.
0141           GO TO 9319
0142      9317 IF (DR(1,I-1,J).EQ.0..AND.CSY(I-1,J+1).GT. DSY(I,J+1) .AND.
              1(GU(1,I,J)+GU(1,I-1,J+1)).LT.0.) GO TO  9318
0143           GO TO  9319
0144      9318 DMB=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*
              1(CSY(I-1,J+1)-DSY(I,J+1))/DX
0145           DBB=BB(1,I,J )*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*(CSY(I-1,J+1)-
              1DSY(I,J+1))*6.2832
0146      9319 CONTINUE
0147           IF (DR(1,I+1,J+1) .EQ. 0.) GO TO 9324
0148           IF (DR(1,I+1,J).EQ.0..AND.CSX(I,J).GT.DSX(I,J+1).AND. (GV(1,I,J)+
              1GV(1,I+1,J+1)).GT.0.) GO TO 9321
0149           GO TO 9322
0150      9321 DMT=SYA(J)*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*(CSX(I,J)-
              1DSX(I,J+1))/DX
0151           DBT=SYA(J)*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*(CSX(I,J)-
              1 DSX(I,J+1))*6.2832/SYB(J)
0152           GO TO 9324
0153      9322 IF(DR(1,I+1,J).EQ.0..AND.CSX(I,J).GT.DSX(I,J+1).AND.(GV(1,I,J)+
              1 GV(1,I+1,J+1)).LT.0. ) GO TO 9323
0154           GO TO  9324
0155      9323 DMT=SYA(J)*DR(1,I+1,J+1)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*(CSX(I,J)
              1-DSX(I,J+1))/DX
0156           DBT=SYA(J)*BB(1,I+1,J+1)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*(CSX(I,J)
              1-DSX(I,J+1))*6.2832/SYB(J+1)
0157           TTT=1.
0158      9324 CONTINUE
0159           IF (DR(1,I-1,J+1) .EQ. 0.) GO TO 9328
0160           IF (DR(1,I-1,J) .EQ.0..AND.CSX(I,J).GT.DSX(I,J+1).AND.(GV(1,I,J)+
              1GV(1,I-1,J+1)).GT.0.)  GO TO 9325
0161           GO TO 9326
```

```
0162      9325 DMC=SYA(J)*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*(CSX(I,J)-
               1 DSX(I,J+1))/DX
0163           DBC=SYA(J)*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*(CSX(I,J)-
               1 DSX(I,J+1))*6.2832/SYB(J)
0164           GO TO 9328
0165      9326 IF (DR(1,I-1,J).EQ.0..AND.CSX(I,J).GT.DSX(I,J+1).AND.(GV(1,I,J)+
               1GV(1,I-1,J+1)).LT.0.) GO TO 9327
0166           GO TO 9328
0167      9327 DMC=SYA(J)*DR(1,I-1,J+1)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*(CSX(I,J)
               1-DSX(I,J+1))/DX
0169           DBC=SYA(J)*BB(1,I-1,J+1)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*(CSX(I,J)
               1-DSX(I,J+1))*6.2832/SYB(J+1)
0169           TTC=1.
0170      9328 CONTINUE
0171           IF (DR(1,I+1,J-1).EQ. 0.) GO TO 9334
0172           IF (DR(1,I,J-1).EQ.0. .AND.CSX(I+1,J-1).GT.DSX(I+1,J).AND.
               1(GV(1,I,J)+GV(1,I+1,J-1)).GT.0.) GO TO 9331
0173           GO TO 9332
0174      9331 DMD=SYA(J-1)*DR(1,I+1,J-1)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*
               1(CSX(I+1,J-1)-DSX(I+1,J))/DX
0175           DBD=SYA(J-1)*BB(1,I+1,J-1)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*
               1(CSX(I+1,J-1)-DSX(I+1,J))*6.2832/SYB(J-1)
0176           TTD=1.
0177           GO TO 9334
0178      9332 IF (DR(1,I,J-1) .EQ.0..AND.CSX(I+1,J-1).GT.DSX(I+1,J).AND.
               1(GV(1,I,J)+GV(1,I+1,J-1)).LT.0.) GO TO 9333
0179           GO TO 9334
0180      9333 DMD=SYA(J-1)*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*
               1(CSX(I+1,J-1)-DSX(I+1,J))/DX
0181           DBD=SYA(J-1)*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*
               1(CSX(I+1,J-1)-DSX(I+1,J))*6.2832/SYB(J)
0182      9334 CONTINUE
0183           IF (DR(1,I-1,J-1) .EQ. 0.) GO TO 9338
0184           IF (DR(1,I,J-1).EQ.0..AND.CSX(I-1,J-1).GT.DSX(I-1,J).AND.
               1(GV(1,I,J)+GV(1,I-1,J-1)) .GT. 0.) GO TO 9335
0185           GO TO 9336
0186      9335 DMG=SYA(J-1)*DR(1,I-1,J-1)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1(CSX(I-1,J-1)-DSX(I-1,J))/DX
0187           DBG=SYA(J-1)*BB(1,I-1,J-1)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1(CSX(I-1,J-1)-DSX(I-1,J))*6.2832/SYB(J-1)
0188           TTG=1.
0189           GO TO 9338
0190      9336 IF (DR(1,I,J-1).EQ.0..AND.CSX(I-1,J-1).GT.DSX(I-1,J).AND.
               1(GV(1,I,J)+GV(1,I-1,J-1)).LT.0.) GO TO 9337
0191           GO TO 9338
0192      9337 DMG=SYA(J-1)*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1(CSX(I-1,J-1)-DSX(I-1,J))/DX
0193           DBG=SYA(J-1)*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1(CSX(I-1,J-1)-DSX(I-1,J))*6.2832/SYB(J)
0194      9338 CONTINUE
0195           IF (DR(1,I-1,J+1) .EQ. 0.) GO TO 8404
0196           IF (DR(1,I,J+1).EQ.0..AND. DSY(I,J).GT.CSY(I-1,J) .AND.(GU(1,I,J)
               1+GU(1,I-1,J+1)).LT.0.) GO TO 8301
0197           GO TO 8302
0198      8301 DMB=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*(DSY(I,J)-
               1CSY(I-1,J))/DX
```



```
0199           DBB=BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*(DSY(I,J)-CSY(I+1,J)
               1 )*6.2832
0200           GO TO 8404
0201      8302 IF (DR(1,I,J+1 ).EQ.0..AND. DSY(I,J).GT.CSY(I-1,J).AND.(GU(1,I,J)
               1+GU(1,I-1,J+1)).GT.0.) GO TO 8303
0202           GO TO 8404
0203      8303 DMB=SXB(J+1)*DR(1,I-1,J+1)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*
               1 (DSY(I,J)-CSY(I-1,J))/DX
0204           DBB=BB(1,I-1,J+1)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*(DSY(I,J)-
               1 CSY(I-1,J))*6.2832
0205           TTB=1.
0206      8404 CONTINUE
0207           IF (DR(1,I-1,J-1) .EQ. 0.) GO TO 8409
0208           IF (DR(1,I,J-1).EQ.0..AND. DSY(I,J).GT.CSY(I-1,J).AND.(GU(1,I,J)
               1+GU(1,I-1,J-1)).LT.0.) GO TO 8405
0209           GO TO 8406
0210      8405 DML=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J-1 ))*DT*(DSY(I,J)-
               1 CSY(I-1,J))/DX
0211           DBL=BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*(DSY(I,J)-CSY(I-1,J)
               1 )* 6.2832
0212           GO TO 8409
0213      8406 IF (DR(1,I,J-1).EQ.0..AND. DSY(I,J).GT.CSY(I-1,J).AND.(GU(1,I,J)
               1+GU(1,I-1,J-1)).GT.0.) GO TO 8407
0214           GO TO 8408
0215      8407 DML=SXB(J-1)*DR(1,I-1,J-1)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*
               1 (DSY(I,J)-CSY(I-1,J))/DX
0216           DBL=BB(1,I-1,J-1)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*(DSY(I,J)-
               1 CSY(I-1,J))*6.2832
0217           TTL=1.
0218      8408 CONTINUE
0219           IF (DR(1,I+1,J-1) .EQ. 0.) GO TO 8315
0220           IF (DR(1,I+1,J).EQ. 0..AND.DSY(I+1,J-1).GT. CSY(I,J-1).AND.
               1(GU(1,I,J)+GU(1,I+1,J-1)).LT.0.) GO TO 8312
0221           GO TO 8313
0222      8312 DMA=SXB(J-1)*DR(1,I+1,J-1)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*
               1 (DSY(I+1,J-1)-CSY(I,J-1))/DX
0223           DBA=BB(1,I+1,J-1)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*(DSY(I+1,J-1)-
               1 CSY(I,J-1))*6.2832
0224           TTA=1.
0225           GO TO 8315
0226      8313 IF (DR(1,I+1,J).EQ. 0..AND.DSY(I+1,J-1).GT. CSY(I,J-1).AND.
               1(GU(1,I,J)+GU(1,I+1,J-1)).GT.0.) GO TO 8314
0227           GO TO 8315
0228      8314 DMA=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*(DSY(I+1,J-1)
               1-CSY(I,J-1))/DX
0229           DBA=BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*(DSY(I+1,J-1)-
               1CSY(I,J-1))*6.2832
0230      8315 CONTINUE
0231           IF (DR(1,I+1,J+1) .EQ. 0.) GO TO 8319
0232           IF (DR(1,I+1,J).EQ.0..AND.DSY(I+1,J+1).GT.CSY(I,J+1).AND.
               1(GU(1,I,J)+GU(1,I+1,J+1)).LT.0.) GO TO 8316
0233           GO TO 8317
0234      8316 DMR=SXB(J+1)*DR(1,I+1,J+1)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*(
               1 DSY(I+1,J+1)-CSY(I,J+1))/DX
0235           DBR=BB(1,I+1,J+1)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*(DSY(I+1,J+1)-
               1CSY(I,J+1))*6.2832
```

```
0236            TTR=1.
0237            GO TO 8319
0238       8317 IF (CR(1,I+1,J).EQ.0..AND.DSY(I+1,J+1).GT. CSY(I,J+1).AND.
               1(GU(1,I,J)+GU(1,I+1,J+1)).GT.0.) GO TO 8318
0239            GO TO 8319
0240       8318 DMR=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*
               1(DSY(I+1,J+1)-CSY(I,J+1))/DX
0241            DBR=BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*(DSY(I+1,J+1)-
               1CSY(I,J+1))*6.2832
0242       8319 CONTINUE
0243            IF (CR(1,I+1,J-1) .EQ. 0.)  GO TO  8324
0244            IF (CR(1,I+1,J).EQ.0..AND.DSX(I,J).GT.CSX(I,J-1).AND. (GV(1,I,J)+
               1 GV(1,I+1,J-1)) .LT. 0.)  GO TO  8321
0245            GO TO 8322
0246       8321 DMD=SYA(J-1)*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*(DSX(I,J)-
               1 CSX(I,J-1))/DX
0247            DBD=SYA(J-1)*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*(DSX(I,J)-
               1 CSX(I,J-1))*6.2832/SYB(J)
0248            GO TO 8324
0249       8322 IF (CR(1,I+1,J).EQ.0..AND.DSX(I,J).GT.CSX(I,J-1).AND. (GV(1,I,J)+
               1 GV(1,I+1,J-1)).GT. 0.) GO TO 8323
0250            GO TO 8324
0251       8323 DMD=SYA(J-1)*DR(1,I+1,J-1)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*
               1 (DSX(I,J)-CSX(I,J-1))/DX
0252            DBD=SYA(J-1)*BB(1,I+1,J-1)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*
               1 (DSX(I,J)-CSX(I,J-1))*6.2832/ SYB(J-1)
0253            TTD=1.
0254       8324 CONTINUE
0255            IF (DR(1,I-1,J-1) .EQ. 0.) GO TO 8328
0256            IF (CR(1,I-1,J).EQ.0..AND.DSX(I,J).GT.CSX(I,J-1).AND.(GV(1,I,J)+
               1 GV(1,I-1,J-1)).LT.0.) GO TO  8325
0257            GO TO  8326
0258       8325 DMG=SYA(J-1)*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*(DSX(I,J)-
               1 CSX(I,J-1))/DX
0259            DBG=SYA(J-1)*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*(DSX(I,J)-
               1 CSX(I,J-1))*6.2832/SYB(J)
0260            GO TO 8328
0261       8326 IF (DR(1,I-1,J).EQ.0..AND.DSX(I,J).GT.CSX(I,J-1).AND. (GV(1,I,J)+
               1 GV(1,I-1,J-1)).GT. 0.)  GO TO 8327
0262            GO TO 8328
0263       8327 DMG=SYA(J-1)*DR(1,I-1,J-1)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1 (DSX(I,J)-CSX(I,J-1))/DX
0264            DBG=SYA(J-1)*BB(1,I-1,J-1)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1 (DSX(I,J)-CSX(I,J-1))*6.2832/SYB(J-1)
0265            TTG=1.
0266       8328 CONTINUE
0267            IF (DR(1,I+1,J+1).EQ. 0.) GO TO  8334
0268            IF (CR(1,I,J+1).EQ.0. .AND.DSX(I+1,J+1).GT.CSX(I+1,J).AND.
               1(GV(1,I,J)+GV(1,I+1,J+1)).LT.0.) GO TO  8331
0269            GO TO 8332
0270       8331 DMT=SYA(J  )*DR(1,I+1,J+1)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*
               1(DSX(I+1,J+1)-CSX(I+1,J))/DX
0271            DBT=SYA(J  )*BB(1,I+1,J+1)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*
               1(DSX(I+1,J+1)-CSX(I+1,J))*6.2832/SYB(J+1)
0272            TTT=1.
0273            GO TO 8334
```



```
0274       8332 IF (CR(1,I,J+1) .EQ.0..AND.DSX(I+1,J+1).GT.CSX(I+1,J).AND.
               1(GV(1,I,J)+GV(1,I+1,J+1)).GT.0.) GO TO 8333
0275            GO TO 8334
0276       8333 DMT=SYA(J  )*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*
               1(DSX(I+1,J+1)-CSX(I+1,J))/DX
0277            DBT=SYA(J  )*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*
               1(DSX(I+1,J+1)-CSX(I+1,J))*6.2832/SYB(J)
0278       8334 CONTINUE
0279            IF (CR(1,I-1,J+1) .EQ. 0.) GO TO 8338
0280            IF (CR(1,I,J+1).EQ.0..AND.DSX(I-1,J+1).GT.CSX(I-1,J).AND.
               1(GV(1,I,J)+GV(1,I-1,J+1)) .LT. 0.) GO TO 8335
0281            GO TO  8336
0282       8335 DMC=SYA(J  )*DR(1,I-1,J+1)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*
               1(DSX(I-1,J+1)-CSX(I-1,J))/DX
0283            DBC=SYA(J  )*BB(1,I-1,J+1)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*
               1(DSX(I-1,J+1)-CSX(I-1,J))*6.2832/SYB(J+1)
0284            TTC=1.
0285            GO TO 8338
0286       8336 IF (CR(1,I,J+1).EQ.0..AND.DSX(I-1,J+1).GT.CSX(I-1,J).AND.
               1(GV(1,I,J)+GV(1,I-1,J+1)).GT.0.) GO TO 8337
0287            GO TO 8338
0288       8337 DMC=SYA(J  )*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*
               1(DSX(I-1,J+1)-CSX(I-1,J))/DX
0289            DBC=SYA(J  )*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*
               1(DSX(I-1,J+1)-CSX(I-1,J))*6.2832/SYB(J)
0290       8338 CONTINUE
0291            IF (I .LE. IM) GO TO 654
0292            IF (J .GT. JM) GO TO 653
0293            IF (DSY(I,J).GT.CSY(I-1,J).AND.CSY(I-1,J).GT.0. .AND.DR(1,I,J-1)
               1 .GT.0..AND.ZSX(I,J-1).EQ.0..AND.(GU(1,I,J)+GU(1,I-1,J-1)).GT.0.)
               2 GO TO  7441
0294            GO TO  7442
0295       7441 DML=SXB(J-1)*DR(1,I-1,J-1)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*
               1 (DSY(I,J)-CSY(I-1,J))/DX
0296            DBL=SXB(J-1)*BB(1,I-1,J-1)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*
               1 (DSY(I,J)-CSY(I-1,J))*6.2832/SXB(J-1)
0297            TTL=1.
0298            GO TO 7444
0299       7442 IF (DSY(I,J).GT.CSY(I-1,J).AND.CSY(I-1,J).GT.0. .AND.DR(1,I,J-1)
               1 .GT.0..AND.ZSX(I,J-1).EQ.0..AND.(GU(1,I,J)+GU(1,I-1,J-1)).LT.0.)
               2 GO TO 7443
0300            GO TO 7444
0301       7443 DML=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*(DSY(I,J)
               1-CSY(I-1,J))/DX
0302            DBL=SXB(J)*BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J-1))*DT*(DSY(I,J)
               1-CSY(I-1,J))*6.2832/SXB(J)
0303       7444 CONTINUE
0304            IF (DSY(I+1,J+1).GT.CSY(I,J+1).AND.CSY(I,J+1).GT.0..AND.
               1 DR(1,I+1,J).GT.0..AND.ZSX(I+1,J).EQ.0..AND.(GU(1,I,J)+
               2 GU(1,I+1,J+1)).GT.0.) GO TO  7445
0305            GO TO 7446
0306       7445 DMR=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*(DSY(I+1,J+1)
               1 -CSY(I,J+1))/DX
0307            DBR=SXB(J)*BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*(DSY(I+1,J+1)
               1 -CSY(I,J+1))*6.2832/SXB(J)
0308            GO TO 7448
```

```
0309       7446 IF (DSY(I+1,J+1).GT.CSY(I,J+1).AND.CSY(I,J+1).GT.0..AND.
               1 DR(1,I+1,J).GT.0..AND.ZSX(I+1,J) .EQ.0..AND.(GU(1,I,J)+
               2 GU(1,I+1,J+1)).LT.0.)  GO TO  7447
0310            GO TO 7448
0311       7447 DMR=SXP(J+1)*DR(1,I+1,J+1)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*
               1 (DSY(I+1,J+1)-CSY(I,J+1))/DX
0312            DBR=SXB(J+1)*BB(1,I+1,J+1)*.5*(GU(1,I,J)+GU(1,I+1,J+1))*DT*
               1 (DSY(I+1,J+1)-CSY(I,J+1))*6.2832/SXB(J+1)
0313            TTR=1.
0314       7448 CONTINUE
0315            IF (CSX(I,J).GT.DSX(I,J+1).AND.DSX(I,J+1).GT.0..AND.DR(1,I+1,J)
               1.GT.0..AND.RSY(I+1,J).EQ.0..AND.(GV(1,I,J)+GV(1,I+1,J+1)).GT.0.)
               2GO TO 7451
0316            GO TO 7452
0317       7451 DMT=SYA(J   )*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*(CSX(I,J)-
               1DSX(I,J+1))/DX
0318            DBT=SYA(J   )*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*(CSX(I,J)-
               1DSX(I,J+1))*6.2832/SXB(J)
0319            GO TO 7454
0320       7452 IF (CSX(I,J).GT.DSX(I,J+1).AND.DSX(I,J+1).GT.0..AND.DR(1,I+1,J)
               1.GT.0..AND.RSY(I+1,J).EQ.0..AND.(GV(1,I,J)+GV(1,I+1,J+1)).LT.0.)
               2GO TO 7453
0321            GO TO 7454
0322       7453 DMT=SYA(J   )*DR(1,I+1,J+1)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*
               1(CSX(I,J)-DSX(I,J+1))/DX
0323            DBT=SYA(J   )*BB(1,I+1,J+1)*.5*(GV(1,I,J)+GV(1,I+1,J+1))*DT*
               1(CSX(I,J)-DSX(I,J+1))*6.2832/SXB(J+1)
0324            TTT=1.
0325       7454 CONTINUE
0326            IF (CSX(I-1,J).LT.CSX(I-1,J-1).AND.DSX(I-1,J).GT.0..AND.
               1DR(1,I,J-1).GT.0..AND.RSY(I,J-1).EQ.0..AND.(GV(1,I,J)+
               2GV(1,I-1,J-1)).GT.0.) GO TO 7461
0327            GO TO 7462
0328       7461 DMG=SYA(J-1)*DR(1,I-1,J-1)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1(CSX(I-1,J-1)-DSX(I-1,J))/DX
0329            DBG=SYA(J-1)*BB(1,I-1,J-1)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1(CSX(I-1,J-1)-DSX(I-1,J))*6.2832/SXB(J-1)
0330            TTG=1.
0331            GO TO 7464
0332       7462 IF (DSX(I-1,J).LT.CSX(I-1,J-1).AND.DSX(I-1,J).GT.0..AND.
               1DR(1,I,J-1).GT.0..AND.RSY(I,J-1).EQ.0..AND.(GV(1,I,J)+
               2GV(1,I-1,J-1)).LT.0.) GO TO 7463
0333            GO TO 7464
0334       7463 DMG=SYA(J-1)*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1(CSX(I-1,J-1)-DSX(I-1,J))/DX
0335            DBG=SYA(J-1)*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J-1))*DT*
               1(CSX(I-1,J-1)-DSX(I-1,J))*6.2832/SXB(J)
0336       7464 CONTINUE
0337            GO TO 654
0338        653 CONTINUE
0339            IF (DSY(I,J).GT.CSY(I-1,J).AND.CSY(I-1,J).GT.0..AND.DR(1,I,J+1)
               1 .GT.0..AND.ZSX(I,J+1).EQ.0..AND.(GU(1,I,J)+GU(1,I-1,J+1)).GT.0.)
               2 GO TO  6441
0340            GO TO  6442
0341       6441 DMB=SXB(J+1)*DP(1,I-1,J+1)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*
               1 (DSY(I,J)-CSY(I-1,J))/DX
```



```
0342            DBB=SXB(J+1)*BB(1,I-1,J+1)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*
               1 (DSY(I,J)-CSY(I-1,J))*6.2832/SXB(J+1)
0343            TTB=1.
0344            GO TO 6444
0345       6442 IF (DSY(I,J).GT.CSY(I-1,J).AND.CSY(I-1,J).GT.0..AND.DR(1,I,J+1)
               1 .GT.0..AND.ZSX(I,J+1).EQ.0..AND.(GU(1,I,J)+GU(1,I-1,J+1)).LT.0.)
               2 GO TO 6443
0346            GO TO 6444
0347       6443 DMB=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*(DSY(I,J)
               1-CSY(I-1,J))/DX
0348            DBB=SXB(J)*BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I-1,J+1))*DT*(DSY(I,J)
               1-CSY(I-1,J))*6.2832/SXB(J)
0349       6444 CONTINUE
0350            IF (CSY(I+1,J-1).GT.CSY(I,J-1).AND.CSY(I,J-1).GT.0..AND.
               1 DR(1,I+1,J).GT.0..AND.ZSX(I+1,J).EQ.0..AND.(GU(1,I,J)+
               2 GU(1,I+1,J-1)).GT.0.) GO TO  6445
0351            GO TO 6446
0352       6445 DMA=SXB(J)*DR(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*(DSY(I+1,J-1)
               1 -CSY(I,J-1))/DX
0353            BBA=SXB(J)*BB(1,I,J)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*(DSY(I+1,J-1)
               1 -CSY(I,J-1))*6.2832/SXB(J)
0354            GO TO 6449
0355       6446 IF (CSY(I+1,J-1).GT.CSY(I,J-1).AND.CSY(I,J-1).GT.0..AND.
               1 DR(1,I+1,J).GT.0..AND.ZSX(I+1,J) .EQ.0..AND.(GU(1,I,J)+
               2 GU(1,I+1,J-1)).LT.0.)  GO TO  6447
0356            GO TO 6449
0357       6447 DMA=SXB(J-1)*DR(1,I+1,J-1)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*
               1 (DSY(I+1,J-1)-CSY(I,J-1))/DX
0358            DBA=SXB(J-1)*BB(1,I+1,J-1)*.5*(GU(1,I,J)+GU(1,I+1,J-1))*DT*
               1 (DSY(I+1,J-1)-CSY(I,J-1))*6.2832/SXB(J-1)
0359            TTA=1.
0360       6448 CONTINUE
0361            IF (DSX(I,J).GT.CSX(I,J-1).AND.CSX(I,J-1).GT.0..AND.DR(1,I+1,J)
               1.GT.0..AND.RSY(I+1,J).EQ.0..AND.(GV(1,I,J)+GV(1,I+1,J-1)).LT.0.)
               2GO TO 6451
0362            GO TO 6452
0363       6451 DMD=SYA(J-1)*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*(DSX(I,J)-
               1CSX(I,J-1))/DX
0364            DBD=SYA(J-1)*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*(DSX(I,J)-
               1CSX(I,J-1))*6.2832/SXB(J)
0365            GO TO 6454
0366       6452 IF (DSX(I,J).GT.CSX(I,J-1).AND.CSX(I,J-1).GT.0..AND.DR(1,I+1,J)
               1.GT.0..AND.RSY(I+1,J).EQ.0..AND.(GV(1,I,J)+GV(1,I+1,J-1)).GT.0.)
               2GO TO 6453
0367            GO TO 6454
0368       6453 DMD=SYA(J-1)*DR(1,I+1,J-1)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*
               1(DSX(I,J)-CSX(I,J-1))/DX
0369            DBD=SYA(J-1)*BB(1,I+1,J-1)*.5*(GV(1,I,J)+GV(1,I+1,J-1))*DT*
               1(DSX(I,J)-CSX(I,J-1))*6.2832/SXB(J-1)
0370            TTD=1.
0371       6454 CONTINUE
0372            IF (CSX(I-1,J).LT.DSX(I-1,J+1).AND.CSX(I-1,J).GT.0..AND.
               1DR(1,I,J+1).GT.0..AND.RSY(I,J+1).EQ.0..AND.(GV(1,I,J)+
               2GV(1,I-1,J+1)).LT.0.) GO TO 6461
0373            GO TO 6462
0374       6461 DMC=SYA(J   )*DR(1,I-1,J+1)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*
```

```
                   1(DSX(I-1,J+1)-CSX(I-1,J))/DX
 0375              DBC=SYA(J  )*BP(1,I-1,J+1)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*
                   1(DSX(I-1,J+1)-CSX(I-1,J))*6.2832/SXB(J+1)
 0376              TTC=1.
 0377              GO TO 6464
 0378         6462 IF (CSX(I-1,J).LT.DSX(I-1,J+1).AND.CSX(I-1,J).GT.0..AND.
                   1DR(1,I,J+1).GT.0..AND.RSY(I,J+1).EQ.0..AND.(GV(1,I,J)+
                   2GV(1,I-1,J+1)).GT.0.) GO TO 6463
 0379              GO TO 6464
 0380         6463 DMC=SYA(J  )*DR(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*
                   1(DSX(I-1,J+1)-CSX(I-1,J))/DX
 0381              DBC=SYA(J  )*BB(1,I,J)*.5*(GV(1,I,J)+GV(1,I-1,J+1))*DT*
                   1(DSX(I-1,J+1)-CSX(I-1,J))*6.2832/SXB(J)
 0382         6464 CONTINUE
 0383          654 CONTINUE
 0384           39 CONTINUE
 0385              DR(2,I,J)=DR(1,I,J)+(DM4+DM2+DMD+DMB+DMG+DML-DM3-DM1-DMC-DMA-
                   1DMR-DMT)/(VOL(I,J)*SF(J))
 0386              BB(2,I,J)=BB(1,I,J)+(DB4+DB2+DBD+DBB+DBL+DBG-DB3-DB1-DBC-DBA-
                   1DBR-DBT)/(VOL(I,J)*SF(J)*SYB(J)/(6.2832*DX))
 0387              FU(2,I,J)=(TT2*DM2*GU(1,I-1,J)+TT4*DM4*GU(1,I,J-1)-TT1*DM1*GU(1,
                   1I+1,J)-TT3*DM3*GU(1,I,J+1)+GU(1,I-1,J+1)*(TTB*DMB-TTC*DMC)
                   2+GU(1,I+1,J-1)*(TTD*DMD-TTA*DMA)-GU(1,I+1,J+1)*(TTT*DMT+TTR*DMR)
                   3+GU(1,I-1,J-1)*(TTL*DML+TTG*DMG)+GU(1,I,J)*(DR(1,I,J)*VOL(I,J)*
                   4SF(J)+(1.-TT4)*DM4+(1.-TT2)*DM2+(1.-TTB)*DMB+(1.-TTL)*DML+(1.-TTG)*
                   5DMG+(1.-TTD)*DMD-(1.-TT1)*DM1-(1.-TT3)*DM3-(1.-TTA)*DMA-(1.-TTC)*
                   6DMC-(1.-TTT)*DMT-(1.-TTR)*DMR))/(VOL(I,J)*DR(2,I,J)*SF(J))
 0388              FV(2,I,J)=(TT2*DM2*GV(1,I-1,J)+TT4*DM4*GV(1,I,J-1)-TT1*DM1*GV(1,
                   1I+1,J)-TT3*DM3*GV(1,I,J+1)+GV(1,I-1,J+1)*(TTB*DMB-TTC*DMC)
                   2+GV(1,I+1,J-1)*(TTD*DMD-TTA*DMA)-GV(1,I+1,J+1)*(TTT*DMT+TTR*DMR)
                   3+GV(1,I-1,J-1)*(TTL*DML+TTG*DMG)+GV(1,I,J)*(DR(1,I,J)*VOL(I,J)*
                   4SF(J)+(1.-TT4)*DM4+(1.-TT2)*DM2+(1.-TTB)*DMB+(1.-TTL)*DML+(1.-TTG)*
                   5DMG+(1.-TTD)*DMD-(1.-TT1)*DM1-(1.-TT3)*DM3-(1.-TTA)*DMA-(1.-TTC)*
                   6DMC-(1.-TTT)*DMT-(1.-TTR)*DMR))/(VOL(I,J)*DR(2,I,J)*SF(J))
 0389              FE(2,I,J)=(TT2*DM2*GE(1,I-1,J)+TT4*DM4*GE(1,I,J-1)-TT1*DM1*GE(1,
                   1I+1,J)-TT3*DM3*GE(1,I,J+1)+GE(1,I-1,J+1)*(TTB*DMB-TTC*DMC)
                   2+GE(1,I+1,J-1)*(TTD*DMD-TTA*DMA)-GE(1,I+1,J+1)*(TTT*DMT+TTR*DMR)
                   3+GE(1,I-1,J-1)*(TTL*DML+TTG*DMG)+GE(1,I,J)*(DR(1,I,J)*VOL(I,J)*
                   4SF(J)+(1.-TT4)*DM4+(1.-TT2)*DM2+(1.-TTB)*DMB+(1.-TTL)*DML+(1.-TTG)*
                   5DMG+(1.-TTD)*DMD-(1.-TT1)*DM1-(1.-TT3)*DM3-(1.-TTA)*DMA-(1.-TTC)*
                   6DMC-(1.-TTT)*DMT-(1.-TTR)*DMR))/(VOL(I,J)*DR(2,I,J)*SF(J))
 0390              IF (ABS(FU(2,I,J)) .LE. EPS1) FU(2,I,J)=0.
 0391              IF (ABS(FV(2,I,J)) .LE. EPS1) GO TO 1130
 0392              FVV2=FV(2,I,J)**2
 0393              GO TO 1131
 0394         1130 FVV2=0.
 0395         1131 FI(2,I,J)=FE(2,I,J)-.5*(FU(2,I,J)**2+FVV2)
 0396              IF (DR(1,I+1,J).EQ.0..AND.DR(1,I-1,J).EQ.0..AND.DR(1,I,J+1).EQ.
                   10..AND.DR(1,I,J-1).EQ.0.) CALL VACUUM(RMEAN,DX)
 0397           25 CONTINUE
 0398              RETURN
 0399              END
```

FORTRAN IV G LEVEL 19 PLOTS DATE = 72173 19/11/47 PAGE 0001

```
 0001              SUBROUTINE PLOTS(IB,IE,JB,JE,DX,DT,TIME,N,IW,JD,ID)
 0002              DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
                   1FE(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BB(2,137,22),
                   2GU(1,137,22),GV(1,137,22),GI(1,137,22),GE(1,137,22),TMP(2,137,22),
                   3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPB(22),BV(137,22),
                   4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
                   5ZSX(137,22)
 0003              DIMENSION PDR(2,137,22),PBB(2,137,22),PTMP(2,137,22)
 0004              COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,GU,GV,GI,GE,TMP,SYA,SXB,SYB,VOL,
                   1SI,CPB,BV,CSX,DSX,CSY,DSY,RSY, ZSX,C1,Q2,Q3,Q4,M,I,J
 0005              SCR=1.E+04
 0006              TOR=1.0E+09
 0007              IBN=IB-1
 0008              IF (N.EQ.1) NN=2
 0009              IF (N.EQ. 2) NN=1
 0010              DO 77 J=JB,JE
 0011              DO 77 I=IBN,IE
 0012              IF (FU(N,I,J) .NE. 0. .AND. DR(1,I,J) .EQ. 0.) FU(N,I,J)=0.
 0013              IF (FV(N,I,J) .NE. 0. .AND. DR(1,I,J) .EQ. 0.) FV(N,I,J)=0.
 0014              PDR(N,I,J)=DR(N,I,J)*TOR
 0015              PBB(N,I,J)=(BB(N,I,J)+BV(I,J))/SCR
 0016              PTMP(N,I,J)=TMP(N,I,J)/1000000.
 0017              PDR(NN,I,J)=FU(N,I,J)/TOR
 0018              PBB(NN,I,J)=FV(N,I,J)/TOR
 0019           77 CONTINUE
 0020              WRITE (6,84)
 0021           84 FORMAT (1H0,'DISTRIBUTION OF TEMPERATURE IN SHOCK TUBE')
 0022              DO 64 I=IBN,IE
 0023              WRITE (6,63) (PTMP(N,I,J),J=JB,JE)
 0024           64 CONTINUE
 0025           63 FORMAT (1H0,33F4.2)
 0026              WRITE (6,85)
 0027           85 FORMAT (1H0,'DENSITY VALUES')
 0028              DO 65 I=IBN,IE
 0029              WRITE (6,63) (PDR(N,I,J),J=JB,JE)
 0030           65 CONTINUE
 0031              WRITE (6,86)
 0032           86 FORMAT (1H0,'MAGNETIC FIELD DISTRIBUTION')
 0033              DO 46 I=IBN,IE
 0034              WRITE (6,63) (PBB(N,I,J),J=JB,JE)
 0035           46 CONTINUE
 0036              WRITE (6,95)
 0037           95 FORMAT (1H0,'U-COMPONENT.')
 0038              DO 55 I=IBN,IE
 0039              WRITE (6,53) (PDR(NN,I,J),J=JB,JE)
 0040           55 CONTINUE
 0041              WRITE (6,96)
 0042           96 FORMAT (1H0,'V-COMPONENT.')
 0043              DO 56 I=IBN,IE
 0044              WRITE (6,53) (PBB(NN,I,J),J=JB,JE)
 0045           56 CONTINUE
 0046           53 FORMAT (1H0,26F5.2)
 0047              RETURN
 0048              END
```

```
0001      SUBROUTINE VISCOS(CR,CK,MP,FUR,FUL,GUR,GUL,FVT,FVB,GVT,GVB,DT,DX,
         1IP,JP)
0002      DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
         1FF(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BR(2,137,22),
         2GU(1,137,22),GV(1,137,22),GI(1,137,22),GE(1,137,22),TMP(2,137,22),
         3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPB(22),BV(137,22),
         4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
         5ZSX(137,22)
0003      COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,GU,GV,GI,GE,TMP,SYA,SXB,SYB,VOL,
         1SI,CPB,BV,CSX,DSX,CSY,DSY,RSY, ZSX,Q1,Q2,Q3,Q4,M,I,J
0004      Q1=0.
0005      Q2=0.
0006      Q3=0.
0007      Q4=0.
0008      DRR=DR(1,I+1,J)
0009      DRL=DR(1,I-1,J)
0010      DRT=DR(1,I,J+1)
0011      DRB=DR(1,I,J-1)
0012      IF(DR(1,I+1,J) .EQ. 0.) GO TO 1
0013      GO TO 2
0014    1 CONTINUE
0015      FU(1,I+1,J)=FU(1,I,J)
0016      FV(1,I+1,J)=FV(1,I,J)
0017      GU(1,I+1,J)=GU(1,I,J)
0018      TC(1,I+1,J)=TC(1,I,J)
0019      DRR=DR(1,I,J)
0020    2 CONTINUE
0021      IF(DR(1,I-1,J) .EQ. 0.) GO TO 3
0022      GO TO 4
0023    3 CONTINUE
0024      FU(1,I-1,J)=FU(1,I,J)
0025      FV(1,I-1,J)=FV(1,I,J)
0026      GU(1,I-1,J)=GU(1,I,J)
0027      TC(1,I-1,J)=TC(1,I,J)
0028      DRL=DR(1,I,J)
0029    4 CONTINUE
0030      IF (DR(1,I,J+1) .EQ. 0.) GO TO 5
0031      GO TO 6
0032    5 CONTINUE
0033      FU(1,I,J+1)=FU(1,I,J)
0034      FV(1,I,J+1)=FV(1,I,J)
0035      GV(1,I,J+1)=GV(1,I,J)
0036      TC(1,I,J+1)=TC(1,I,J)
0037      DRT=DR(1,I,J)
0038    6 CONTINUE
0039      IF (DR(1,I,J-1) .EQ. 0.) GO TO 7
0040      GO TO 8
0041    7 CONTINUE
0042      FU(1,I,J-1)=FU(1,I,J)
0043      FV(1,I,J-1)=FV(1,I,J)
0044      GV(1,I,J-1)=GV(1,I,J)
0045      TC(1,I,J-1)=TC(1,I,J)
0046      DRB=DR(1,I,J)
0047    8 CONTINUE
0048      FUR=FU(1,I+1,J)
0049      FUL=FU(1,I-1,J)
```



```
0050      FUT=FU(1,I,J+1)
0051      FUB=FU(1,I,J-1)
0052      FVR=FV(1,I+1,J)
0053      FVL=FV(1,I-1,J)
0054      FVT=FV(1,I,J+1)
0055      FVB=FV(1,I,J-1)
0056      GUR=GU(1,I+1,J)
0057      GUL=GU(1,I-1,J)
0058      GVT=GV(1,I,J+1)
0059      GVB=GV(1,I,J-1)
0060      TCR=TC(1,I+1,J)
0061      TCL=TC(1,I-1,J)
0062      TCT=TC(1,I,J+1)
0063      TCB=TC(1,I,J-1)
0064      IF ((RSY(I,J).EQ.RSY(I+1,J)).AND.(RSY(I,J).EQ.RSY(I-1,J)).AND.
         1(ZSX(I,J).EQ.ZSX(I,J+1)).AND.(ZSX(I,J).EQ.ZSX(I,J-1))) GO TO 59
0065      IF ( DSY(I+1,J) .GT. 0. .AND. DR(1,I+1,J+1) .EQ. 0. .AND. RSY(I,J)
         1.GT. DSY(I+1,J)) GO TO 11
0066      IF ( DSY(I+1,J) .GT. 0. .AND. DR(1,I+1,J-1) .EQ. 0. .AND. RSY(I,J)
         1.GT. DSY(I+1,J)) GO TO 11
0067      GO TO 41
0068   11 CONTINUE
0069      FUR =(FU(1,I+1,J)*DSY(I+1,J)+FU(1,I,J)*(RSY(I,J)-DSY(I+1,J)))
         1 /RSY(I,J)
0070      FVR =(FV(1,I+1,J)*DSY(I+1,J)+FV(1,I,J)*(RSY(I,J)-DSY(I+1,J)))
         1 /RSY(I,J)
0071      DRR =(DR(1,I+1,J)*DSY(I+1,J)+DR(1,I,J)*(RSY(I,J)-DSY(I+1,J)))
         1 /RSY(I,J)
0072      TCR =(TC(1,I+1,J)*DSY(I+1,J)+TC(1,I,J)*(RSY(I,J)-DSY(I+1,J)))
         1 /RSY(I,J)
0073      IF (MP .EQ. 1) GO TO 41
0074      GUR =(GU(1,I+1,J)*DSY(I+1,J)+GU(1,I,J)*(RSY(I,J)-DSY(I+1,J)))
         1 /RSY(I,J)
0075   41 CONTINUE
0076      IF (DR(1,I+1,J) .EQ. 0. .AND. DSY(I+1,J+1) .GT. CSY(I,J+1) .AND.
         1DSY(I+1,J-1) .GT. CSY(I,J-1)) GO TO 12
0077      GO TO 42
0078   12 CONTINUE
0079      FUR =(FU(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+FU(1,I+1,J-1)*
         1(DSY(I+1,J-1)-CSY(I,J-1))+FU(1,I,J)*(RSY(I,J)+CSY(I,J+1)+CSY(I,
         2J-1)-DSY(I+1,J+1)-DSY(I+1,J-1)))/RSY(I,J)
0080      FVR =(FV(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+FV(1,I+1,J-1)*
         1(DSY(I+1,J-1)-CSY(I,J-1))+FV(1,I,J)*(RSY(I,J)+CSY(I,J+1)+CSY(I,
         2J-1)-DSY(I+1,J+1)-DSY(I+1,J-1)))/RSY(I,J)
0081      DRR =(DR(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+DR(1,I+1,J-1)*
         1(DSY(I+1,J-1)-CSY(I,J-1))+DR(1,I,J)*(RSY(I,J)+CSY(I,J+1)+CSY(I,
         2J-1)-DSY(I+1,J+1)-DSY(I+1,J-1)))/RSY(I,J)
0082      TCR =(TC(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+TC(1,I+1,J-1)*
         1(DSY(I+1,J-1)-CSY(I,J-1))+TC(1,I,J)*(RSY(I,J)+CSY(I,J+1)+CSY(I,
         2J-1)-DSY(I+1,J+1)-DSY(I+1,J-1)))/RSY(I,J)
0083      IF (MP .EQ. 1) GO TO 42
0084      GUR =(GU(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+GU(1,I+1,J-1)*
         1(DSY(I+1,J-1)-CSY(I,J-1))+GU(1,I,J)*(RSY(I,J)+CSY(I,J+1)+CSY(I,
         2J-1)-DSY(I+1,J+1)-DSY(I+1,J-1)))/RSY(I,J)
0085   42 CONTINUE
0086      IF (DR(1,I+1,J) .EQ. 0. .AND. DSY(I+1,J+1) .GT. CSY(I,J+1) .AND.
```

```
                 1DSY(I+1,J-1) .LE. CSY(I,J-1)) GO TO 14
 0087            GO TO 43
 0088         14 CONTINUE
 0089            FUR =(FU(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+FU(1,I,J)*(RSY(I,J)
                1+CSY(I,J+1)-DSY(I+1,J+1)))/RSY(I,J)
 0090            FVR =(FV(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+FV(1,I,J)*(RSY(I,J)
                1+CSY(I,J+1)-DSY(I+1,J+1)))/RSY(I,J)
 0091            DRR =(DR(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+DR(1,I,J)*(RSY(I,J)
                1+CSY(I,J+1)-DSY(I+1,J+1)))/RSY(I,J)
 0092            TCR =(TC(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+TC(1,I,J)*(RSY(I,J)
                1+CSY(I,J+1)-DSY(I+1,J+1)))/RSY(I,J)
 0093            IF (MP .EQ. 1) GO TO 43
 0094            GUR =(GU(1,I+1,J+1)*(DSY(I+1,J+1)-CSY(I,J+1))+GU(1,I,J)*(RSY(I,J)
                1+CSY(I,J+1)-DSY(I+1,J+1)))/RSY(I,J)
 0095         43 CONTINUE
 0096            IF (DR(1,I+1,J) .EQ. 0. .AND. DSY(I+1,J+1) .LE. CSY(I,J+1) .AND.
                1DSY(I+1,J-1) .GT. CSY(I,J-1)) GO TO 15
 0097            GO TO 44
 0098         15 CONTINUE
 0099            FUR =(FU(1,I+1,J-1)*(DSY(I+1,J-1)-CSY(I,J-1))+FU(1,I,J)*(RSY(I,J)
                1+CSY(I,J-1)-DSY(I+1,J-1)))/RSY(I,J)
 0100            FVR =(FV(1,I+1,J-1)*(DSY(I+1,J-1)-CSY(I,J-1))+FV(1,I,J)*(RSY(I,J)
                1+CSY(I,J-1)-DSY(I+1,J-1)))/RSY(I,J)
 0101            DRR =(DR(1,I+1,J-1)*(DSY(I+1,J-1)-CSY(I,J-1))+DR(1,I,J)*(RSY(I,J)
                1+CSY(I,J-1)-DSY(I+1,J-1)))/RSY(I,J)
 0102            TCR =(TC(1,I+1,J-1)*(DSY(I+1,J-1)-CSY(I,J-1))+TC(1,I,J)*(RSY(I,J)
                1+CSY(I,J-1)-DSY(I+1,J-1)))/RSY(I,J)
 0103            IF (MP .EQ. 1) GO TO 44
 0104            GUR =(GU(1,I+1,J-1)*(DSY(I+1,J-1)-CSY(I,J-1))+GU(1,I,J)*(RSY(I,J)
                1+CSY(I,J-1)-DSY(I+1,J-1)))/RSY(I,J)
 0105         44 CONTINUE
 0106            IF ( CSY(I-1,J) .GT. 0. .AND. DR(1,I-1,J-1) .EQ. 0. .AND. RSY(I,J)
                1.GT. CSY(I-1,J)) GO TO 16
 0107            IF ( CSY(I-1,J) .GT. 0. .AND. DR(1,I-1,J+1) .EQ. 0. .AND. RSY(I,J)
                1.GT. CSY(I-1,J)) GO TO 16
 0108            GO TO 45
 0109         16 CONTINUE
 0110            FUL =(FU(1,I-1,J)*CSY(I-1,J)+FU(1,I,J)*(RSY(I,J)-CSY(I-1,J)))
                1 /RSY(I,J)
 0111            FVL =(FV(1,I-1,J)*CSY(I-1,J)+FV(1,I,J)*(RSY(I,J)-CSY(I-1,J)))
                1 /RSY(I,J)
 0112            DRL =(DR(1,I-1,J)*CSY(I-1,J)+DR(1,I,J)*(RSY(I,J)-CSY(I-1,J)))
                1 /RSY(I,J)
 0113            TCL =(TC(1,I-1,J)*CSY(I-1,J)+TC(1,I,J)*(RSY(I,J)-CSY(I-1,J)))
                1 /RSY(I,J)
 0114            IF (MP .EQ. 1) GO TO 45
 0115            GUL =(GU(1,I-1,J)*CSY(I-1,J)+GU(1,I,J)*(RSY(I,J)-CSY(I-1,J)))
                1 /RSY(I,J)
 0116         45 CONTINUE
 0117            IF (DR(1,I-1,J) .EQ. 0. .AND. CSY(I-1,J+1) .GT. DSY(I,J+1) .AND.
                1CSY(I-1,J-1) .GT. DSY(I,J-1)) GO TO 17
 0118            GO TO 46
 0119         17 CONTINUE
 0120            FUL =(FU(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+FU(1,I-1,J-1)*
                1(CSY(I-1,J-1)-DSY(I,J-1))+FU(1,I,J)*(RSY(I,J)+DSY(I,J+1)+DSY(I,
                2J-1)-CSY(I-1,J+1)-CSY(I-1,J-1)))/RSY(I,J)
```



```
 0121            FVL =(FV(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+FV(1,I-1,J-1)*
                1(CSY(I-1,J-1)-DSY(I,J-1))+FV(1,I,J)*(RSY(I,J)+DSY(I,J+1)+DSY(I,
                2J-1)-CSY(I-1,J+1)-CSY(I-1,J-1)))/RSY(I,J)
 0122            DRL =(DR(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+DR(1,I-1,J-1)*
                1(CSY(I-1,J-1)-DSY(I,J-1))+DR(1,I,J)*(RSY(I,J)+DSY(I,J+1)+DSY(I,
                2J-1)-CSY(I-1,J+1)-CSY(I-1,J-1)))/RSY(I,J)
 0123            TCL =(TC(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+TC(1,I-1,J-1)*
                1(CSY(I-1,J-1)-DSY(I,J-1))+TC(1,I,J)*(RSY(I,J)+DSY(I,J+1)+DSY(I,
                2J-1)-CSY(I-1,J+1)-CSY(I-1,J-1)))/RSY(I,J)
 0124            IF (MP .EQ. 1) GO TO 46
 0125            GUL =(GU(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+GU(1,I-1,J-1)*
                1(CSY(I-1,J-1)-DSY(I,J-1))+GU(1,I,J)*(RSY(I,J)+DSY(I,J+1)+DSY(I,
                2J-1)-CSY(I-1,J+1)-CSY(I-1,J-1)))/RSY(I,J)
 0126         46 CONTINUE
 0127            IF (DR(1,I-1,J) .EQ. 0. .AND. CSY(I-1,J+1) .GT. DSY(I,J+1) .AND.
                1CSY(I-1,J-1) .LE. DSY(I,J-1)) GO TO 18
 0128            GO TO 47
 0129         18 CONTINUE
 0130            FUL =(FU(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+FU(1,I,J)*(RSY(I,J)
                1+DSY(I,J+1)-CSY(I-1,J+1)))/RSY(I,J)
 0131            FVL =(FV(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+FV(1,I,J)*(RSY(I,J)
                1+DSY(I,J+1)-CSY(I-1,J+1)))/RSY(I,J)
 0132            DRL =(DR(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+DR(1,I,J)*(RSY(I,J)
                1+DSY(I,J+1)-CSY(I-1,J+1)))/RSY(I,J)
 0133            TCL =(TC(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+TC(1,I,J)*(RSY(I,J)
                1+DSY(I,J+1)-CSY(I-1,J+1)))/RSY(I,J)
 0134            IF (MP .EQ. 1) GO TO 47
 0135            GUL =(GU(1,I-1,J+1)*(CSY(I-1,J+1)-DSY(I,J+1))+GU(1,I,J)*(RSY(I,J)
                1+DSY(I,J+1)-CSY(I-1,J+1)))/RSY(I,J)
 0136         47 CONTINUE
 0137            IF (DR(1,I-1,J) .EQ. 0. .AND. CSY(I-1,J+1) .LE. DSY(I,J+1) .AND.
                1CSY(I-1,J-1) .GT. DSY(I,J-1)) GO TO 19
 0138            GO TO 48
 0139         19 CONTINUE
 0140            FUL =(FU(1,I-1,J-1)*(CSY(I-1,J-1)-DSY(I,J-1))+FU(1,I,J)*(RSY(I,J)
                1+DSY(I,J-1)-CSY(I-1,J-1)))/RSY(I,J)
 0141            FVL =(FV(1,I-1,J-1)*(CSY(I-1,J-1)-DSY(I,J-1))+FV(1,I,J)*(RSY(I,J)
                1+DSY(I,J-1)-CSY(I-1,J-1)))/RSY(I,J)
 0142            DRL =(DR(1,I-1,J-1)*(CSY(I-1,J-1)-DSY(I,J-1))+DR(1,I,J)*(RSY(I,J)
                1+DSY(I,J-1)-CSY(I-1,J-1)))/RSY(I,J)
 0143            TCL =(TC(1,I-1,J-1)*(CSY(I-1,J-1)-DSY(I,J-1))+TC(1,I,J)*(RSY(I,J)
                1+DSY(I,J-1)-CSY(I-1,J-1)))/RSY(I,J)
 0144            IF (MP .EQ. 1) GO TO 48
 0145            GUL =(GU(1,I-1,J-1)*(CSY(I-1,J-1)-DSY(I,J-1))+GU(1,I,J)*(RSY(I,J)
                1+DSY(I,J-1)-CSY(I-1,J-1)))/RSY(I,J)
 0146         48 CONTINUE
 0147            IF ( DSX(I,J+1).GT.0..AND.DR(1,I-1,J+1).EQ.0..AND.ZSX(I,J).GT.
                1DSX(I,J+1)) GO TO 20
 0148            IF ( DSX(I,J+1) .GT.0..AND.DR(1,I+1,J+1).EQ.0..AND.ZSX(I,J).GT.
                1DSX(I,J+1)) GO TO 20
 0149            GO TO 49
 0150         20 CONTINUE
 0151            FUT =(FU(1,I,J+1)*DSX(I,J+1)+FU(1,I,J)*(ZSX(I,J)-DSX(I,J+1)))
                1 /ZSX(I,J)
 0152            FVT =(FV(1,I,J+1)*DSX(I,J+1)+FV(1,I,J)*(ZSX(I,J)-DSX(I,J+1)))
                1 /ZSX(I,J)
```

```
0153         DRT =(DR(1,I,J+1)*DSX(I,J+1)+DR(1,I,J)*(ZSX(I,J)-DSX(I,J+1)))
            1 /ZSX(I,J)
0154         TCT =(TC(1,I,J+1)*DSX(I,J+1)+TC(1,I,J)*(ZSX(I,J)-DSX(I,J+1)))
            1 /ZSX(I,J)
0155         IF (MP .EQ. 1) GO TO 49
0156         GVT =(GV(1,I,J+1)*DSX(I,J+1)+GV(1,I,J)*(ZSX(I,J)-DSX(I,J+1)))
            1 /ZSX(I,J)
0157      49 CONTINUE
0158         IF (DR(1,I,J+1).EQ.0..AND.DSX(I+1,J+1).GT.CSX(I+1,J).AND.
            1DSX(I-1,J+1) .GT.CSX(I-1,J))  GO TO 21
0159         GO TO 50
0160      21 CONTINUE
0161         FUT =(FU(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+FU(1,I-1,J+1)*
            1(DSX(I-1,J+1)-CSX(I-1,J))+FU(1,I,J)*(ZSX(I,J)+CSX(I+1,J)+
            2CSX(I-1,J)-DSX(I+1,J+1)-DSX(I-1,J+1)))/ZSX(I,J)
0162         FVT =(FV(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+FV(1,I-1,J+1)*
            1(DSX(I-1,J+1)-CSX(I-1,J))+FV(1,I,J)*(ZSX(I,J)+CSX(I+1,J)+
            2CSX(I-1,J)-DSX(I+1,J+1)-DSX(I-1,J+1)))/ZSX(I,J)
0163         DRT =(DR(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+DR(1,I-1,J+1)*
            1(DSX(I-1,J+1)-CSX(I-1,J))+DR(1,I,J)*(ZSX(I,J)+CSX(I+1,J)+
            2CSX(I-1,J)-DSX(I+1,J+1)-DSX(I-1,J+1)))/ZSX(I,J)
0164         TCT =(TC(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+TC(1,I-1,J+1)*
            1(DSX(I-1,J+1)-CSX(I-1,J))+TC(1,I,J)*(ZSX(I,J)+CSX(I+1,J)+
            2CSX(I-1,J)-DSX(I+1,J+1)-DSX(I-1,J+1)))/ZSX(I,J)
0165         IF (MP .EQ. 1) GO TO 50
0166         GVT =(GV(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+GV(1,I-1,J+1)*
            1(DSX(I-1,J+1)-CSX(I-1,J))+GV(1,I,J)*(ZSX(I,J)+CSX(I+1,J)+
            2CSX(I-1,J)-DSX(I+1,J+1)-DSX(I-1,J+1)))/ZSX(I,J)
0167      50 CONTINUE
0168         IF (DR(1,I,J+1).EQ.0..AND.DSX(I+1,J+1).GT.CSX(I+1,J).AND.
            1DSX(I-1,J+1) .LE.CSX(I-1,J))  GO TO 22
0169         GO TO 51
0170      22 CONTINUE
0171         FUT =(FU(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+FU(1,I,J)*
            1(ZSX(I,J)+CSX(I+1,J)-DSX(I+1,J+1)))/ZSX(I,J)
0172         FVT =(FV(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+FV(1,I,J)*
            1(ZSX(I,J)+CSX(I+1,J)-DSX(I+1,J+1)))/ZSX(I,J)
0173         DRT =(DR(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+DR(1,I,J)*
            1(ZSX(I,J)+CSX(I+1,J)-DSX(I+1,J+1)))/ZSX(I,J)
0174         TCT =(TC(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+TC(1,I,J)*
            1(ZSX(I,J)+CSX(I+1,J)-DSX(I+1,J+1)))/ZSX(I,J)
0175         IF (MP .EQ. 1) GO TO 51
0176         GVT =(GV(1,I+1,J+1)*(DSX(I+1,J+1)-CSX(I+1,J))+GV(1,I,J)*
            1(ZSX(I,J)+CSX(I+1,J)-DSX(I+1,J+1)))/ZSX(I,J)
0177      51 CONTINUE
0178         IF (DR(1,I,J+1) .EQ.0..AND.DSX(I+1,J+1).LE.CSX(I+1,J).AND.
            1DSX(I-1,J+1) .GT.CSX(I-1,J))  GO TO 23
0179         GO TO 52
0180      23 CONTINUE
0181         FUT=(FU(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+FU(1,I,J)*(
            1ZSX(I,J)+CSX(I-1,J)-DSX(I-1,J+1)))/ZSX(I,J)
0182         FVT=(FV(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+FV(1,I,J)*(
            1ZSX(I,J)+CSX(I-1,J)-DSX(I-1,J+1)))/ZSX(I,J)
0183         DRT=(DR(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+DR(1,I,J)*(
            1ZSX(I,J)+CSX(I-1,J)-DSX(I-1,J+1)))/ZSX(I,J)
0184         TCT=(TC(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+TC(1,I,J)*(
```



```
            1ZSX(I,J)+CSX(I-1,J)-DSX(I-1,J+1)))/ZSX(I,J)
0185         IF (MP .EQ. 1) GO TO 52
0186         GVT=(GV(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+GV(1,I,J)*(
            1ZSX(I,J)+CSX(I-1,J)-DSX(I-1,J+1)))/ZSX(I,J)
0187      52 CONTINUE
0188         IF ( CSX(I,J-1).GT.0..AND.DR(1,I-1,J-1).EQ.0..AND.ZSX(I,J).GT.
            1CSX(I,J-1)) GO TO 24
0189         IF ( CSX(I,J-1) .GT.0..AND.DR(1,I+1,J-1).EQ.0..AND.ZSX(I,J).GT.
            1CSX(I,J-1)) GO TO 24
0190         GO TO 53
0191      24 CONTINUE
0192         FUB =(FU(1,I,J-1)*CSX(I,J-1)+FU(1,I,J)*(ZSX(I,J)-CSX(I,J-1)))
            1 /ZSX(I,J)
0193         FVB =(FV(1,I,J-1)*CSX(I,J-1)+FV(1,I,J)*(ZSX(I,J)-CSX(I,J-1)))
            1 /ZSX(I,J)
0194         DRB =(DR(1,I,J-1)*CSX(I,J-1)+DR(1,I,J)*(ZSX(I,J)-CSX(I,J-1)))
            1 /ZSX(I,J)
0195         TCB =(TC(1,I,J-1)*CSX(I,J-1)+TC(1,I,J)*(ZSX(I,J)-CSX(I,J-1)))
            1 /ZSX(I,J)
0196         IF (MP .EQ. 1) GO TO 53
0197         GVB =(GV(1,I,J-1)*CSX(I,J-1)+GV(1,I,J)*(ZSX(I,J)-CSX(I,J-1)))
            1 /ZSX(I,J)
0198      53 CONTINUE
0199         IF (DR(1,I,J-1).EQ.0..AND.CSX(I+1,J-1).GT.DSX(I+1,J).AND.CSX(I-1,J
            1-1).GT.DSX(I-1,J)) GO TO 26
0200         GO TO 54
0201      26 CONTINUE
0202         FUB =(FU(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+FU(1,I-1,J-1)*
            1(CSX(I-1,J-1)-DSX(I-1,J))+FU(1,I,J)*(ZSX(I,J)+DSX(I+1,J)+
            2DSX(I-1,J)-CSX(I+1,J-1)-CSX(I-1,J-1)))/ZSX(I,J)
0203         FVB =(FV(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+FV(1,I-1,J-1)*
            1(CSX(I-1,J-1)-DSX(I-1,J))+FV(1,I,J)*(ZSX(I,J)+DSX(I+1,J)+
            2DSX(I-1,J)-CSX(I+1,J-1)-CSX(I-1,J-1)))/ZSX(I,J)
0204         DRB =(DR(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+DR(1,I-1,J-1)*
            1(CSX(I-1,J-1)-DSX(I-1,J))+DR(1,I,J)*(ZSX(I,J)+DSX(I+1,J)+
            2DSX(I-1,J)-CSX(I+1,J-1)-CSX(I-1,J-1)))/ZSX(I,J)
0205         TCB =(TC(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+TC(1,I-1,J-1)*
            1(CSX(I-1,J-1)-DSX(I-1,J))+TC(1,I,J)*(ZSX(I,J)+DSX(I+1,J)+
            2DSX(I-1,J)-CSX(I+1,J-1)-CSX(I-1,J-1)))/ZSX(I,J)
0206         IF (MP .EQ. 1) GO TO 54
0207         GVB =(GV(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+GV(1,I-1,J-1)*
            1(CSX(I-1,J-1)-DSX(I-1,J))+GV(1,I,J)*(ZSX(I,J)+DSX(I+1,J)+
            2DSX(I-1,J)-CSX(I+1,J-1)-CSX(I-1,J-1)))/ZSX(I,J)
0208      54 CONTINUE
0209         IF (DR(1,I,J-1).EQ.0..AND.CSX(I+1,J-1).GT.DSX(I+1,J).AND.
            1CSX(I-1,J-1) .LE.DSX(I-1,J))  GO TO 27
0210         GO TO 55
0211      27 CONTINUE
0212         FUB =(FU(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+FU(1,I,J)*
            1(ZSX(I,J)+DSX(I+1,J)-CSX(I+1,J-1)))/ZSX(I,J)
0213         FVB =(FV(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+FV(1,I,J)*
            1(ZSX(I,J)+DSX(I+1,J)-CSX(I+1,J-1)))/ZSX(I,J)
0214         DRB =(DR(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+DR(1,I,J)*
            1(ZSX(I,J)+DSX(I+1,J)-CSX(I+1,J-1)))/ZSX(I,J)
0215         TCB =(TC(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+TC(1,I,J)*
            1(ZSX(I,J)+DSX(I+1,J)-CSX(I+1,J-1)))/ZSX(I,J)
```

```
0216            IF (MP .EQ. 1) GO TO 55
0217            GVB =(GV(1,I+1,J-1)*(CSX(I+1,J-1)-DSX(I+1,J))+GV(1,I,J)*
               1(ZSX(I,J)+DSX(I+1,J)-CSX(I+1,J-1)))/ZSX(I,J)
0218         55 CONTINUE
0219            IF (DR(1,I,J-1) .EQ.0..AND.CSX(I+1,J-1).LE.DSX(I+1,J).AND.
               1CSX(I-1,J-1) .GT.DSX(I-1,J)) GO TO 28
0220            GO TO 56
0221         28 CONTINUE
0222            FUB=(FU(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+FU(1,I,J)*(
               1ZSX(I,J)+DSX(I-1,J)-CSX(I-1,J-1)))/ZSX(I,J)
0223            FVB=(FV(1,I-1,J-1)*(CSX(I-1,J-1)-DSX(I-1,J))+FV(1,I,J)*(
               1ZSX(I,J)+DSX(I-1,J)-CSX(I-1,J-1)))/ZSX(I,J)
0224            DRB=(DR(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+DR(1,I,J)*(
               1ZSX(I,J)+DSX(I-1,J)-CSX(I-1,J-1)))/ZSX(I,J)
0225            TCB=(TC(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+TC(1,I,J)*(
               1ZSX(I,J)+DSX(I-1,J)-CSX(I-1,J-1)))/ZSX(I,J)
0226            IF (MP .EQ. 1) GO TO 56
0227            GVB=(GV(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+GV(1,I,J)*(
               1ZSX(I,J)+DSX(I-1,J)-CSX(I-1,J-1)))/ZSX(I,J)
0228         56 CONTINUE
0229            IF (RSY(I,J).GT.DSY(I+1,J) .AND. DSY(I+1,J).GT.0..AND.DR(1,I+1,
               1J-1).GT.0..AND.DR(1,I,J-1).EQ.0.) GO TO 30
0230            GO TO 57
0231         30 CONTINUE
0232            FUR =(FU(1,I+1,J-1)*(RSY(I,J)-DSY(I+1,J))+FU(1,I+1,J)*DSY(I+1,J))
               1 /RSY(I,J)
0233            FVR =(FV(1,I+1,J-1)*(RSY(I,J)-DSY(I+1,J))+FV(1,I+1,J)*DSY(I+1,J))
               1 /RSY(I,J)
0234            DRR =(DR(1,I+1,J-1)*(RSY(I,J)-DSY(I+1,J))+DR(1,I+1,J)*DSY(I+1,J))
               1 /RSY(I,J)
0235            TCR =(TC(1,I+1,J-1)*(RSY(I,J)-CSY(I+1,J))+TC(1,I+1,J)*DSY(I+1,J))
               1 /RSY(I,J)
0236            IF (MP .EQ. 1) GO TO 57
0237            GUR =(GU(1,I+1,J-1)*(RSY(I,J)-CSY(I+1,J))+GU(1,I+1,J)*DSY(I+1,J))
               1 /RSY(I,J)
0238         57 CONTINUE
0239            IF (ZSX(I,J).GT.DSX(I,J+1).AND. DSX(I,J+1).GT.0..AND.DR(1,I-1,J+1)
               1.GT.0..AND.DR(1,I-1,J).EQ.0.)GO TO 31
0240            GO TO 58
0241         31 CONTINUE
0242            FUT =(FU(1,I-1,J+1)*(ZSX(I,J)-DSX(I,J+1))+FU(1,I,J+1)*
               1 DSX(I,J+1))/ZSX(I,J)
0243            FVT =(FV(1,I-1,J+1)*(ZSX(I,J)-DSX(I,J+1))+FV(1,I,J+1)*
               1 DSX(I,J+1))/ZSX(I,J)
0244            DRT =(DR(1,I-1,J+1)*(ZSX(I,J)-DSX(I,J+1))+DR(1,I,J+1)*
               1 DSX(I,J+1))/ZSX(I,J)
0245            TCT =(TC(1,I-1,J+1)*(ZSX(I,J)-DSX(I,J+1))+TC(1,I,J+1)*
               1 DSX(I,J+1))/ZSX(I,J)
0246            IF (MP .EQ. 1) GO TO 58
0247            GVT =(GV(1,I-1,J+1)*(ZSX(I,J)-DSX(I,J+1))+GV(1,I,J+1)*
               1 DSX(I,J+1))/ZSX(I,J)
0248         58 CONTINUE
0249            IF (RSY(I,J).GT.DSY(I+1,J).AND. DSY(I+1,J).GT.0..AND.
               1DR(1,I+1,J+1).GT.0..AND.DR(1,I,J+1).EQ.0.) GO TO 61
0250            GO TO 81
0251         61 CONTINUE
```



```
0252            FUR=(FU(1,I+1,J+1)*(RSY(I,J)-DSY(I+1,J))+FU(1,I+1,J)*DSY(I+1,J))
               1/RSY(I,J)
0253            FVR=(FV(1,I+1,J+1)*(RSY(I,J)-DSY(I+1,J))+FV(1,I+1,J)*DSY(I+1,J))
               1/RSY(I,J)
0254            DRR=(DR(1,I+1,J+1)*(RSY(I,J)-DSY(I+1,J))+DR(1,I+1,J)*DSY(I+1,J))
               1/RSY(I,J)
0255            TCR=(TC(1,I+1,J+1)*(RSY(I,J)-DSY(I+1,J))+TC(1,I+1,J)*DSY(I+1,J))
               1/RSY(I,J)
0256            GUR=(GU(1,I+1,J+1)*(RSY(I,J)-DSY(I+1,J))+GU(1,I+1,J)*DSY(I+1,J))
               1/RSY(I,J)
0257         81 CONTINUE
0258            IF (ZSX(I,J).GT.DSX(I,J+1).AND. DSX(I,J+1).GT.0..AND.DR(1,I+1,J+1)
               1.GT.0..AND.DR(1,I+1,J).EQ.0.) GO TO 62
0259            GO TO 82
0260         62 CONTINUE
0261            FUT=(FU(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1))+FU(1,I,J+1)*DSX(I,J+1))
               1/ZSX(I,J)
0262            FVT=(FV(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1))+FV(1,I,J+1)*DSX(I,J+1))
               1/ZSX(I,J)
0263            DRT=(DR(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1))+DR(1,I,J+1)*DSX(I,J+1))
               1/ZSX(I,J)
0264            TCT=(TC(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1))+TC(1,I,J+1)*DSX(I,J+1))
               1/ZSX(I,J)
0265            GVT=(GV(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1))+GV(1,I,J+1)*DSX(I,J+1))
               1/ZSX(I,J)
0266         82 CONTINUE
0267            IF (RSY(I,J).GT.CSY(I-1,J).AND. CSY(I-1,J).GT.0..AND.
               1DR(1,I-1,J+1).GT.0..AND.DR(1,I,J+1).EQ.0.) GO TO 63
0268            GO TO 83
0269         63 CONTINUE
0270            FUL=(FU(1,I-1,J+1)*(RSY(I,J)-CSY(I-1,J))+FU(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0271            FVL=(FV(1,I-1,J+1)*(RSY(I,J)-CSY(I-1,J))+FV(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0272            DRL=(DR(1,I-1,J+1)*(RSY(I,J)-CSY(I-1,J))+DR(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0273            TCL=(TC(1,I-1,J+1)*(RSY(I,J)-CSY(I-1,J))+TC(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0274            GUL=(GU(1,I-1,J+1)*(RSY(I,J)-CSY(I-1,J))+GU(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0275         83 CONTINUE
0276            IF (RSY(I,J).GT.CSY(I-1,J).AND. CSY(I-1,J).GT.0..AND.
               1DR(1,I-1,J-1).GT.0..AND.DR(1,I,J-1).EQ.0.) GO TO 64
0277            GO TO 84
0278         64 CONTINUE
0279            FUL=(FU(1,I-1,J-1)*(RSY(I,J)-CSY(I-1,J))+FU(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0280            FVL=(FV(1,I-1,J-1)*(RSY(I,J)-CSY(I-1,J))+FV(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0281            DRL=(DR(1,I-1,J-1)*(RSY(I,J)-CSY(I-1,J))+DR(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0282            TCL=(TC(1,I-1,J-1)*(RSY(I,J)-CSY(I-1,J))+TC(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0283            GUL=(GU(1,I-1,J-1)*(RSY(I,J)-CSY(I-1,J))+GU(1,I-1,J)*CSY(I-1,J))
               1/RSY(I,J)
0284         84 CONTINUE
```

```
0285          IF (ZSX(I,J).GT.CSX(I,J-1).AND. CSX(I,J-1).GT.0..AND.
             1DR(1,I+1,J-1).GT.0..AND.DR(1,I+1,J).EQ.0.) GO TO 65
0286          GO TC 85
0287       65 CONTINUE
0288          FUB=(FU(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1))+FU(1,I,J-1)*CSX(I,J-1))
             1/ZSX(I,J)
0289          FVB=(FV(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1))+FV(1,I,J-1)*CSX(I,J-1))
             1/ZSX(I,J)
0290          DRB=(DR(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1))+DR(1,I,J-1)*CSX(I,J-1))
             1/ZSX(I,J)
0291          TCB=(TC(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1))+TC(1,I,J-1)*CSX(I,J-1))
             1/ZSX(I,J)
0292          GVB=(GV(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1))+GV(1,I,J-1)*CSX(I,J-1))
             1/ZSX(I,J)
0293       85 CONTINUE
0294          IF (ZSX(I,J).GT.CSX(I,J-1).AND. CSX(I,J-1).GT.0..AND.DR(1,I-1,J-1)
             1.GT.0..AND.DR(1,I-1,J).EQ.0.) GO TO 32
0295          GO TC 159
0296       32 CONTINUE
0297          FUB =(FU(1,I-1,J-1)*(ZSX(I,J)-CSX(I,J-1))+FU(1,I,J-1)*CSX(I,J-1))
             1 /ZSX(I,J)
0298          FVB =(FV(1,I-1,J-1)*(ZSX(I,J)-CSX(I,J-1))+FV(1,I,J-1)*CSX(I,J-1))
             1 /ZSX(I,J)
0299          DRB =(DR(1,I-1,J-1)*(ZSX(I,J)-CSX(I,J-1))+DR(1,I,J-1)*CSX(I,J-1))
             1 /ZSX(I,J)
0300          TCB =(TC(1,I-1,J-1)*(ZSX(I,J)-CSX(I,J-1))+TC(1,I,J-1)*CSX(I,J-1))
             1 /ZSX(I,J)
0301          IF (MP .EQ. 1) GO TO 159
0302          GVB =(GV(1,I-1,J-1)*(ZSX(I,J)-CSX(I,J-1))+GV(1,I,J-1)*CSX(I,J-1))
             1 /ZSX(I,J)
0303      159 CONTINUE
0304          IF (I .LE. IM) GO TO 777
0305          IF (J .GT. JM) K=-1
0306          IF (J .LE. JM) K=1
0307          IF (RSY(I,J).GT.CSY(I-1,J).AND.CSY(I-1,J).GT.0..AND.DR(1,I,J-K)
             1.GT.0..AND.ZSX(I,J-K).EQ.0.) GC TC 101
0308          GO TO 102
0309      101 FUL=(FU(1,I-1,J)*CSY(I-1,J)+FU(1,I-1,J-K)*(RSY(I,J)-CSY(I-1,J)))
             1/RSY(I,J)
0310          FVL=(FV(1,I-1,J)*CSY(I-1,J)+FV(1,I-1,J-K)*(RSY(I,J)-CSY(I-1,J)))
             1/RSY(I,J)
0311          DRL=(DR(1,I-1,J)*CSY(I-1,J)+DR(1,I-1,J-K)*(RSY(I,J)-CSY(I-1,J)))
             1/RSY(I,J)
0312          TCL=(TC(1,I-1,J)*CSY(I-1,J)+TC(1,I-1,J-K)*(RSY(I,J)-CSY(I-1,J)))
             1/RSY(I,J)
0313          GUL=(GU(1,I-1,J)*CSY(I-1,J)+GU(1,I-1,J-K)*(RSY(I,J)-CSY(I-1,J)))
             1/RSY(I,J)
0314      102 IF (RSY(I,J).GT.DSY(I+1,J).AND.DSY(I+1,J).GT.0..AND.DR(1,I,J-K)
             1.GT.0..AND.ZSX(I,J-K).EQ.0.) GO TO 103
0315          GO TO 104
0316      103 FUR=(FU(1,I+1,J)*DSY(I+1,J)+FU(1,I+1,J-K)*(RSY(I,J)-DSY(I+1,J)))
             1/RSY(I,J)
0317          FVR=(FV(1,I+1,J)*DSY(I+1,J)+FV(1,I+1,J-K)*(RSY(I,J)-DSY(I+1,J)))
             1/RSY(I,J)
0318          DRR=(DR(1,I+1,J)*DSY(I+1,J)+DR(1,I+1,J-K)*(RSY(I,J)-DSY(I+1,J)))
             1/RSY(I,J)
```



```
0319          TCR=(TC(1,I+1,J)*DSY(I+1,J)+TC(1,I+1,J-K)*(RSY(I,J)-DSY(I+1,J)))
             1/RSY(I,J)
0320          GUR=(GU(1,I+1,J)*DSY(I+1,J)+GU(1,I+1,J-K)*(RSY(I,J)-DSY(I+1,J)))
             1/RSY(I,J)
0321      104 IF (DSY(I+1,J+K).GT.CSY(I,J+K).AND.CSY(I,J+K).GT.0..AND.
             1DR(1,I+1,J).GT.0..AND.ZSX(I+1,J).EQ.0.) GO TO 105
0322          GO TC 106
0323      105 FUR=(FU(1,I+1,J+K)*(DSY(I+1,J+K)-CSY(I,J+K))+FU(1,I+1,J)*(RSY(I,J)
             1+CSY(I,J+K)-DSY(I+1,J+K)))/RSY(I,J)
0324          FVR=(FV(1,I+1,J+K)*(DSY(I+1,J+K)-CSY(I,J+K))+FV(1,I+1,J)*(RSY(I,J)
             1+CSY(I,J+K)-DSY(I+1,J+K)))/RSY(I,J)
0325          DRR=(DR(1,I+1,J+K)*(DSY(I+1,J+K)-CSY(I,J+K))+DR(1,I+1,J)*(RSY(I,J)
             1+CSY(I,J+K)-DSY(I+1,J+K)))/RSY(I,J)
0326          TCR=(TC(1,I+1,J+K)*(DSY(I+1,J+K)-CSY(I,J+K))+TC(1,I+1,J)*(RSY(I,J)
             1+CSY(I,J+K)-DSY(I+1,J+K)))/RSY(I,J)
0327          GUR=(GU(1,I+1,J+K)*(DSY(I+1,J+K)-CSY(I,J+K))+GU(1,I+1,J)*(RSY(I,J)
             1+CSY(I,J+K)-DSY(I+1,J+K)))/RSY(I,J)
0328      106 IF (J .GT. JM) GO TO 778
0329          IF (ZSX(I,J).GT.DSX(I,J+1).AND.DSX(I,J+1).GT.0..AND.DR(1,I+1,J)
             1.GT.0..AND.RSY(I+1,J).EC.0.) GC TC 107
0330          GO TO 108
0331      107 FUT=(FU(1,I,J+1)*DSX(I,J+1)+FU(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0332          FVT=(FV(1,I,J+1)*DSX(I,J+1)+FV(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0333          DRT=(DR(1,I,J+1)*DSX(I,J+1)+DR(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0334          TCT=(TC(1,I,J+1)*DSX(I,J+1)+TC(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0335          GVT=(GV(1,I,J+1)*DSX(I,J+1)+GV(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0336      108 IF (ZSX(I,J).GT.CSX(I,J-1).AND.CSX(I,J-1).GT.0..AND.DR(1,I+1,J)
             1.GT.0..AND.RSY(I+1,J).EQ.0.) GC TC 109
0337          GO TC 110
0338      109 FUB=(FU(1,I,J-1)*CSX(I,J-1)+FU(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0339          FVB=(FV(1,I,J-1)*CSX(I,J-1)+FV(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0340          DRB=(DR(1,I,J-1)*CSX(I,J-1)+DR(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0341          TCB=(TC(1,I,J-1)*CSX(I,J-1)+TC(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0342          GVB=(GV(1,I,J-1)*CSX(I,J-1)+GV(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0343      110 IF (CSX(I-1,J-1).GT.DSX(I-1,J).AND.DSX(I-1,J).GT.0..AND.
             1DR(1,I+1,J).GT.0..AND.RSY(I+1,J).EC.0.) GO TO 111
0344          GO TO 122
0345      111 FUB=(FU(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+FU(1,I,J-1)*(DSX(I-1,
             1J)+ZSX(I,J)-CSX(I-1,J-1)))/ZSX(I,J)
0346          FVB=(FV(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+FV(1,I,J-1)*(DSX(I-1,
             1J)+ZSX(I,J)-CSX(I-1,J-1)))/ZSX(I,J)
0347          DRB=(DR(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+DR(1,I,J-1)*(DSX(I-1,
             1J)+ZSX(I,J)-CSX(I-1,J-1)))/ZSX(I,J)
0348          TCB=(TC(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+TC(1,I,J-1)*(DSX(I-1,
             1J)+ZSX(I,J)-CSX(I-1,J-1)))/ZSX(I,J)
0349          GVB=(GV(1,I-1,J-1)*(CSX(I-1,J-1)-CSX(I-1,J))+GV(1,I,J-1)*(DSX(I-1,
```

```
              1J)+ZSX(I,J)-CSX(I-1,J-1)))/ZSX(I,J)
0350          GO TO 122
0351      778 CONTINUE
0352          IF (ZSX(I,J).GT.CSX(I,J-1).AND.CSX(I,J-1).GT.0..AND.DR(1,I+1,J)
             1.GT.0..AND.RSY(I+1,J).EQ.0.) GO TO 117
0353          GO TO 119
0354      117 FUB=(FU(1,I,J-1)*CSX(I,J-1)+FU(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0355          FVB=(FV(1,I,J-1)*CSX(I,J-1)+FV(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0356          DRB=(DR(1,I,J-1)*CSX(I,J-1)+DR(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0357          TCB=(TC(1,I,J-1)*CSX(I,J-1)+TC(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0358          GVB=(GV(1,I,J-1)*CSX(I,J-1)+GV(1,I+1,J-1)*(ZSX(I,J)-CSX(I,J-1)))
             1/ZSX(I,J)
0359      118 IF (ZSX(I,J).GT.DSX(I,J+1).AND.DSX(I,J+1).GT.0..AND.DR(1,I+1,J)
             1.GT.0..AND.RSY(I+1,J).EQ.0.) GO TO 119
0360          GO TO 120
0361      119 FUT=(FU(1,I,J+1)*DSX(I,J+1)+FU(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0362          FVT=(FV(1,I,J+1)*DSX(I,J+1)+FV(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0363          DRT=(DR(1,I,J+1)*DSX(I,J+1)+DR(1,I+1,J+1)*(ZSY(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0364          TCT=(TC(1,I,J+1)*DSX(I,J+1)+TC(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0365          GVT=(GV(1,I,J+1)*DSX(I,J+1)+GV(1,I+1,J+1)*(ZSX(I,J)-DSX(I,J+1)))
             1/ZSX(I,J)
0366      120 IF (DSX(I-1,J+1).GT.CSX(I-1,J).AND.CSX(I-1,J).GT.0..AND.
             1DR(1,I+1,J).GT.0..AND.RSY(I+1,J).EQ.0.) GO TO 121
0367          GO TO 122
0368      121 FUT=(FU(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+FU(1,I,J+1)*(CSX(I-1,
             1J)+ZSX(I,J)-DSX(I-1,J+1)))/ZSX(I,J)
0369          FVT=(FV(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+FV(1,I,J+1)*(CSX(I-1,
             1J)+ZSX(I,J)-DSX(I-1,J+1)))/ZSX(I,J)
0370          DRT=(DR(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+DR(1,I,J+1)*(CSX(I-1,
             1J)+ZSX(I,J)-DSX(I-1,J+1)))/ZSX(I,J)
0371          TCT=(TC(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+TC(1,I,J+1)*(CSX(I-1,
             1J)+ZSX(I,J)-DSX(I-1,J+1)))/ZSX(I,J)
0372          GVT=(GV(1,I-1,J+1)*(DSX(I-1,J+1)-CSX(I-1,J))+GV(1,I,J+1)*(CSX(I-1,
             1J)+ZSX(I,J)-DSX(I-1,J+1)))/ZSX(I,J)
0373      122 CONTINUE
0374      777 CONTINUE
0375       59 CONTINUE
0376          IF (((.5*(FUR       +FU(1,I,J)))**2+(.5*(FVR       +FV(1,I,J)))
             1**2)*CK .LT. (.5*(TCR       +TC(1,I,J)))**2 .AND. FU(1,I,J) .GT.
             2FUR        ) Q1=.25*CB*(TCR       +TC(1,I,J))*(DRR       +
             3DR(1,I,J))*(FU(1,I,J)-FUR       )
0377          IF (((.5*(FUT       +FU(1,I,J)))**2+(.5*(FVT       +FV(1,I,J)))
             1**2)*CK .LT. (.5*(TCT       +TC(1,I,J)))**2 .AND. FV(1,I,J)
             2.GT. FVT        ) Q3=.25*CB*(TCT       +TC(1,I,J))*(DRT
             3+DR(1,I,J))*(FV(1,I,J)-FVT       )
0378          IF (((.5*(FU(1,I,J)+FUL       ))**2+(.5*(FV(1,I,J)+FVL       ))
             1**2)*CK .LT. (.5*(TC(1,I,J)+TCL       ))**2 .AND. FUL
             2 .GT. FU(1,I,J)) Q2=.25*CB*(TC(1,I,J)+TCL       )*(DR(1,I,J)+
```



```
             3DRL       )*(FUL       -FU(1,I,J))
0379          IF (((.5*(FU(1,I,J)+FUB       ))**2+(.5*(FV(1,I,J)+FVB       ))
             1**2)*CK .LT. (.5*(TC(1,I,J)+TCB       ))**2 .AND. FVB
             2 .GT. FV(1,I,J)) Q4=.25*CB*(TC(1,I,J)+TCB       )*(DR(1,I,J)+
             3DRB       )*(FVB       -FV(1,I,J))
0380          RETURN
0381          END
```

```
0001          SUBROUTINE AREAV(CI,CJ,GZ,GR,DX,IW,JW,W1,W2,W3,W4,W5,W6,W7,W8,W9)
0002          DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
             1FE(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BB(2,137,22),
             2GU(1,137,22),GV(1,137,22),GI(1,137,22),GE(1,137,22),TMF(2,137,22),
             3SYA(22),SXB(22),SYB(22),VOL(137,22),S(22),CPB(22),BV(137,22),
             4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
             5ZSX(137,22)
0003          COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,GU,GV,GI,GE,TMP,SYA,SXB,SYB,VOL,
             1SI,CPB,BV,CSX,DSX,CSY,DSY,RSY, ZSX,C1,C2,C3,C4,M,I,J
0004          IF ((GZ .GE. (QI+.5)*DX) .AND. GR .GE. (CJ+.5)*DX) GO TO 1
0005          IF ((GZ .LT. (CI+.5)*DX) .AND. GR .GE. (QJ+.5)*DX) GO TO 2
0006          IF ((GZ .LT. (CI+.5)*DX) .AND. GR .LT. (CJ+.5)*DX) GO TO 3
0007          IF ((GZ .GE. (QI+.5)*DX) .AND. GR .LT. (CJ+.5)*DX) GO TO 4
0008          WRITE (6,7) IW,JW,GZ,GR,CI,QJ
0009        7 FORMAT (1H ,2I4,4F15.8)
0010        4 A4=((QJ+.5)*DX-GR)*(GZ-(CI+.5)*DX)
0011          A3=((QJ+.5)*DX-GR)*((CI+1.5)*DX-GZ)
0012          A2=(GR-(QJ-.5)*DX)*((CI+1.5)*DX-GZ)
0013          A1=(GR-(CJ-.5)*DX)*(GZ-(QI+.5)*DX)
0014          IF (DSY(IW+1,JW-1).GT.DX .AND. RSY(IW+1,JW).EQ.0.) A4=A4+A1
0015          IF (DSY(IW+1,JW-1).GT.DX .AND. RSY(IW+1,JW).EQ.0.) A1=0.
0016          IF (DSX(IW,JW).GT.DX .AND. RSY(IW+1,JW).EQ.0.) A2=A2+A1
0017          IF (DSX(IW,JW).GT.DX .AND. RSY(IW+1,JW).EQ.0.) A1=0.
0018          IF (CSX(IW,JW-1).GT.DX .AND. RSY(IW+1,JW-1).EQ. 0.) A3=A3+A4
0019          IF (CSX(IW,JW-1).GT.DX .AND. RSY(IW+1,JW-1).EQ. 0.) A4=0.
0020          W1=(A1*DR(1,IW+1,JW)+A2*DR(1,IW,JW)+A3*DR(1,IW,JW-1)+A4*DR(1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0021          W2=(A1*FU(1,IW+1,JW)+A2*FU(1,IW,JW)+A3*FU(1,IW,JW-1)+A4*FU(1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0022          W3=(A1*FV(1,IW+1,JW)+A2*FV(1,IW,JW)+A3*FV(1,IW,JW-1)+A4*FV(1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0023          W4=(A1*GU(1,IW+1,JW)+A2*GU(1,IW,JW)+A3*GU(1,IW,JW-1)+A4*GU(1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0024          W5=(A1*GV(1,IW+1,JW)+A2*GV(1,IW,JW)+A3*GV(1,IW,JW-1)+A4*GV(1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0025          W6=(A1*P (1,IW+1,JW)+A2*P (1,IW,JW)+A3*P (1,IW,JW-1)+A4*P (1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0026          W7=(A1*BB(1,IW+1,JW)+A2*BB(1,IW,JW)+A3*BB(1,IW,JW-1)+A4*BB(1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0027          W8=(A1*TC(1,IW+1,JW)+A2*TC(1,IW,JW)+A3*TC(1,IW,JW-1)+A4*TC(1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0028          W9=(A1*GE(1,IW+1,JW)+A2*GE(1,IW,JW)+A3*GE(1,IW,JW-1)+A4*GE(1,
             1IW+1,JW-1))/(A1+A2+A3+A4)
0029          GO TO 5
0030        3 A4=((QJ+.5)*DX-GR)*(GZ-(CI-.5)*DX)
0031          A3=((QJ+.5)*DX-GR)*((CI+.5)*DX-GZ)
0032          A2=(GR-(QJ-.5)*DX)*((CI+.5)*DX-GZ)
0033          A1=(GR-(QJ-.5)*DX)*(GZ-(CI-.5)*DX)
0034          W1=(A1*DR(1,IW,JW)+A2*DR(1,IW-1,JW)+A3*DR(1,IW-1,JW-1)+A4*DR(1,
             1IW,JW-1))/(A1+A2+A3+A4)
0035          W2=(A1*FU(1,IW,JW)+A2*FU(1,IW-1,JW)+A3*FU(1,IW-1,JW-1)+A4*FU(1,
             1IW,JW-1))/(A1+A2+A3+A4)
0036          W3=(A1*FV(1,IW,JW)+A2*FV(1,IW-1,JW)+A3*FV(1,IW-1,JW-1)+A4*FV(1,
             1IW,JW-1))/(A1+A2+A3+A4)
0037          W4=(A1*GU(1,IW,JW)+A2*GU(1,IW-1,JW)+A3*GU(1,IW-1,JW-1)+A4*GU(1,
             1IW,JW-1))/(A1+A2+A3+A4)
```



```
0038          W5=(A1*GV(1,IW,JW)+A2*GV(1,IW-1,JW)+A3*GV(1,IW-1,JW-1)+A4*GV(1,
             1IW,JW-1))/(A1+A2+A3+A4)
0039          W6=(A1*P (1,IW,JW)+A2*P (1,IW-1,JW)+A3*P (1,IW-1,JW-1)+A4*P (1,
             1IW,JW-1))/(A1+A2+A3+A4)
0040          W7=(A1*BB(1,IW,JW)+A2*BB(1,IW-1,JW)+A3*BB(1,IW-1,JW-1)+A4*BB(1,
             1IW,JW-1))/(A1+A2+A3+A4)
0041          W8=(A1*TC(1,IW,JW)+A2*TC(1,IW-1,JW)+A3*TC(1,IW-1,JW-1)+A4*TC(1,
             1IW,JW-1))/(A1+A2+A3+A4)
0042          W9=(A1*GE(1,IW,JW)+A2*GE(1,IW-1,JW)+A3*GE(1,IW-1,JW-1)+A4*GE(1,
             1IW,JW-1))/(A1+A2+A3+A4)
0043          GO TO 5
0044        2 A4=((QJ+1.5)*DX-GR)*(GZ-(CI-.5)*DX)
0045          A3=((QJ+1.5)*DX-GR)*((CI+.5)*DX-GZ)
0046          A2=(GR-(QJ+.5)*DX)*((CI+.5)*DX-GZ)
0047          A1=(GR-(QJ+.5)*DX)*(GZ-(CI-.5)*DX)
0048          IF (DSX(IW-1,JW+1).GT.DX .AND. RSY(IW,JW+1).EQ.0.) A2=A2+A1
0049          IF (DSX(IW-1,JW+1).GT.DX .AND. RSY(IW,JW+1).EQ.0.) A1=0.
0050          IF (DSY(IW,JW).GT.DX .AND. RSY(IW,JW+1).EQ.0.) A4=A4+A1
0051          IF (DSY(IW,JW).GT.DX .AND. RSY(IW,JW+1).EQ.0.) A1=0.
0052          IF (CSY(IW-1,JW).GT.DX .AND. RSY(IW-1,JW+1).EQ.0.) A3=A3+A2
0053          IF (CSY(IW-1,JW).GT.DX .AND. RSY(IW-1,JW+1).EQ.0.) A2=0.
0054          W1=(A1*DR(1,IW,JW+1)+A2*DR(1,IW-1,JW+1)+A3*DR(1,IW-1,JW)+A4*DR(1,
             1IW,JW))/(A1+A2+A3+A4)
0055          W2=(A1*FU(1,IW,JW+1)+A2*FU(1,IW-1,JW+1)+A3*FU(1,IW-1,JW)+A4*FU(1,
             1IW,JW))/(A1+A2+A3+A4)
0056          W3=(A1*FV(1,IW,JW+1)+A2*FV(1,IW-1,JW+1)+A3*FV(1,IW-1,JW)+A4*FV(1,
             1IW,JW))/(A1+A2+A3+A4)
0057          W4=(A1*GU(1,IW,JW+1)+A2*GU(1,IW-1,JW+1)+A3*GU(1,IW-1,JW)+A4*GU(1,
             1IW,JW))/(A1+A2+A3+A4)
0058          W5=(A1*GV(1,IW,JW+1)+A2*GV(1,IW-1,JW+1)+A3*GV(1,IW-1,JW)+A4*GV(1,
             1IW,JW))/(A1+A2+A3+A4)
0059          W6=(A1*P (1,IW,JW+1)+A2*P (1,IW-1,JW+1)+A3*P (1,IW-1,JW)+A4*P (1,
             1IW,JW))/(A1+A2+A3+A4)
0060          W7=(A1*BB(1,IW,JW+1)+A2*BB(1,IW-1,JW+1)+A3*BB(1,IW-1,JW)+A4*BB(1,
             1IW,JW))/(A1+A2+A3+A4)
0061          W8=(A1*TC(1,IW,JW+1)+A2*TC(1,IW-1,JW+1)+A3*TC(1,IW-1,JW)+A4*TC(1,
             1IW,JW))/(A1+A2+A3+A4)
0062          W9=(A1*GE(1,IW,JW+1)+A2*GE(1,IW-1,JW+1)+A3*GE(1,IW-1,JW)+A4*GE(1,
             1IW,JW))/(A1+A2+A3+A4)
0063          GO TO 5
0064        1 A4=((QJ+1.5)*DX-GR)*(GZ-(CI+.5)*DX)
0065          A3=((QJ+1.5)*DX-GR)*((QI+1.5)*DX-GZ)
0066          A2=(GR-(CJ+.5)*DX)*((CI+1.5)*DX-GZ)
0067          A1=(GR-(QJ+.5)*DX)*(GZ-(QI+.5)*DX)
0068          IF (DSX(IW  ,JW+1).GT.DX .AND. RSY(IW+1,JW+1).EQ.0.) A2=A2+A1
0069          IF (DSX(IW  ,JW+1).GT.DX .AND. RSY(IW+1,JW+1).EQ.0.) A1=0.
0070          IF (DSY(IW+1,JW).GT.DX .AND. RSY(IW+1,JW+1).EQ.0.) A4=A4+A1
0071          IF (DSY(IW+1,JW).GT.DX .AND. RSY(IW+1,JW+1).EQ.0.) A1=0.
0072          IF (CSY(IW,JW).GT.DX .AND. RSY(IW,JW+1).EQ. 0.) A3=A3+A2
0073          IF (CSY(IW,JW).GT.DX .AND. RSY(IW,JW+1).EQ. 0.) A2=0.
0074          IF (CSX(IW,JW).GT.DX .AND. RSY(IW+1,JW).EQ.0.) A3=A3+A4
0075          IF (CSX(IW,JW).GT.DX .AND. RSY(IW+1,JW).EQ.0.) A4=0.
0076          W1=(A1*DR(1,IW+1,JW+1)+A2*DR(1,IW,JW+1)+A3*DR(1,IW,JW)+A4*DR(1,
             1IW+1,JW))/(A1+A2+A3+A4)
0077          W2=(A1*FU(1,IW+1,JW+1)+A2*FU(1,IW,JW+1)+A3*FU(1,IW,JW)+A4*FU(1,
             1IW+1,JW))/(A1+A2+A3+A4)
```

```
FORTRAN IV G LEVEL  21                  AREAV           DATE = 72210        11/03/32              PAGE 0003
      0078              W3=(A1*FV(1,IW+1,JW+1)+A2*FV(1,IW,JW+1)+A3*FV(1,IW,JW)+A4*FV(1,
                       1IW+1,JW))/(A1+A2+A3+A4)
      0079              W4=(A1*GU(1,IW+1,JW+1)+A2*GU(1,IW,JW+1)+A3*GU(1,IW,JW)+A4*GU(1,
                       1IW+1,JW))/(A1+A2+A3+A4)
      0080              W5=(A1*GV(1,IW+1,JW+1)+A2*GV(1,IW,JW+1)+A3*GV(1,IW,JW)+A4*GV(1,
                       1IW+1,JW))/(A1+A2+A3+A4)
      0081              W6=(A1*P (1,IW+1,JW+1)+A2*P (1,IW,JW+1)+A3*P (1,IW,JW)+A4*P (1,
                       1IW+1,JW))/(A1+A2+A3+A4)
      0082              W7=(A1*BB(1,IW+1,JW+1)+A2*BB(1,IW,JW+1)+A3*BB(1,IW,JW)+A4*BB(1,
                       1IW+1,JW))/(A1+A2+A3+A4)
      0083              W8=(A1*TC(1,IW+1,JW+1)+A2*TC(1,IW,JW+1)+A3*TC(1,IW,JW)+A4*TC(1,
                       1IW+1,JW))/(A1+A2+A3+A4)
      0084              W9=(A1*GE(1,IW+1,JW+1)+A2*GE(1,IW,JW+1)+A3*GE(1,IW,JW)+A4*GE(1,
                       1IW+1,JW))/(A1+A2+A3+A4)
      0085            5 RETURN
      0086              END
```

```
FORTRAN IV G LEVEL  21                  FRAC            DATE = 72210        11/03/32              PAGE 0001
      0001              SUBROUTINE FRAC(SFIJ,AXIJ,AYIJ,    DX,DY)
      0002              DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
                       1FE(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BB(2,137,22),
                       2GU(1,137,22),GV(1,137,22),GI(1,137,22),GE(1,137,22),TMP(2,137,22),
                       3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPB(22),BV(137,22),
                       4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
                       5ZSX(137,22)
      0003              COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,GU,GV,GI,GE,TMP,SYA,SXB,SYB,VOL,
                       1SI,CPB,BV,CSX,DSX,CSY,DSY,RSY, ZSX,C1,C2,C3,C4,M,I,J
      0004              SFIJ=VOL(I,J)/VOL(2,J)
      0005              AXIJ=RSY(I,J)/DY
      0006              AYIJ=ZSX(I,J)/DX
      0007              RETURN
      0008              END
```

```
FORTRAN IV G LEVEL  19                  VACUUM          DATE = 72173        19/11/47              PAGE 0001
      0001              SUBROUTINE VACUUM(RMEAN,DX)
      0002              DIMENSION DR(2,137,22),FU(2,137,22),FV(2,137,22),FI(2,137,22),
                       1FE(2,137,22),TC(2,137,22),PP(2,137,22),P(2,137,22),BB(2,137,22),
                       2GU(1,137,22),GV(1,137,22),GI(1,137,22),GE(1,137,22),TMP(2,137,22),
                       3SYA(22),SXB(22),SYB(22),VOL(137,22),SI(22),CPB(22),BV(137,22),
                       4CSX(137,22),DSX(137,22),CSY(137,22),DSY(137,22),RSY(137,22),
                       5ZSX(137,22)
      0003              COMMON DR,FU,FV,FI,FE,TC,PP,P,BB,GU,GV,GI,GE,TMP,SYA,SXB,SYB,VOL,
                       1SI,CPB,BV,CSX,DSX,CSY,DSY,RSY, ZSX,Q1,Q2,Q3,Q4,M,I,J
      0004              BV(I,J)=7000.*6.2832*DX*RMEAN/SXB(J)*4.8
      0005              PP(1,I,J)=BV(I,J)**2/25.1328
      0006              BB(1,I,J)=0.
      0007              BB(2,I,J)=0.
      0008              GE(1,I,J)=0.
      0009              FE(1,I,J)=0.
      0010              P(1,I,J)=0.
      0011              FI(1,I,J)=0.
      0012              FI(2,I,J)=0.
      0013              GU(1,I,J)=0.
      0014              GV(1,I,J)=0.
      0015              FV(1,I,J)=0.
      0016              FU(1,I,J)=0.
      0017              TC(1,I,J)=0.
      0018              DR(2,I,J)=0.
      0019              DR(1,I,J)=0.
      0020              TMP(1,I,J)=0.
      0021              TMP(2,I,J)=0.
      0022              DSX(I,J)=0.
      0023              DSY(I,J)=0.
      0024              CSX(I,J)=0.
      0025              CSY(I,J)=0.
      0026              ZSX(I,J)=0.
      0027              RSY(I,J)=0.
      0028              RETURN
      0029              END
//GO.FTCBF001 DD DSN=H.LUT,VOL=SER=MT0467,UNIT=2400-9,
// DISP=(NEW,KEEP),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),
// LABEL=(1,SL)
//GO.SC406027  DD  DISP=NEW,UNIT=2400-7,LABEL=(1,NL),
//   VOLUME=SER=CS0037,DCB=(DEN=2,TRTCH=0)
//GC.SYSIN DD *
/*
```