Master

LA-8095
UC-32
Issued: June 1980

SALE: A Simplified ALE Computer Program for Fluid Flow at All Speeds

A. A. Amsden
H. M. Ruppel
C. W. Hirt

CONTENTS

SALE: A SIMPLIFIED ALE COMPUTER PROGRAM FOR FLUID FLOW AT ALL SPEEDS

by

A. A. Amsden, H. M. Ruppel, and C. W. Hirt

ABSTRACT

A simplified numerical fluid-dynamics computing technique is presented for calculating two-dimensional fluid flows at all speeds. It combines an implicit treatment of the pressure equation similar to that in the Implicit Continuous-fluid Eulerian (ICE) technique with the grid rezoning philosophy of the Arbitrary Lagrangian-Eulerian (ALE) method. As a result, it can handle flow speeds from supersonic to the incompressible limit in a grid that may be moved with the fluid in typical Lagrangian fashion, or held fixed in an Eulerian manner, or moved in some arbitrary way to give a continuous rezoning capability. The report describes the combined (ICEd-ALE) technique in the framework of the SALE (Simplified ALE) computer program, for which a general flow diagram and complete FORTRAN listing are included. A set of sample problems show how to use or modify the basic code for a variety of applications. Numerical listings are provided for a sample problem run with the SALE program.

I. INTRODUCTION

Over the past decade, we have witnessed an increasing acceptance of and reliance upon numerical solutions for transient fluid flow problems. In many cases, experimental studies are prohibitively expensive, whereas high-speed computers are comparatively economical and allow a wide range of parameter variations to be examined in a short time. As a result, numerical solution techniques have become more sophisticated and the applications correspondingly more complex.

This report presents a simplified computer program to calculate two-dimensional fluid flows at all speeds, from the incompressible limit to highly supersonic. An implicit treatment of the pressure calculation similar to that in the Implicit Continuous-fluid Eulerian (ICE) technique[1] provides this flow-speed versatility. In addition, the computing mesh may move with the fluid in a typical Lagrangian fashion, be held fixed in an Eulerian manner, or move in some arbitrarily specified way to provide a continuous rezoning capability. This latitude results from the use of an Arbitrary Lagrangian-Eulerian (ALE) treatment[2] of the computing mesh. The program is named SALE, for Simplified ALE. The essential features of the ICEd-ALE combination are presented here to make this report a self-contained guide. SALE bears a strong resemblance to YAQUI, the original but more complex ICEd-ALE program.[3]

The partial differential equations solved by the SALE program are the Navier-Stokes equations,

$$\frac{\partial \rho u}{\partial t} + \frac{1}{r}\frac{\partial r\rho u^2}{\partial x} + \frac{\partial \rho uv}{\partial y}$$

$$= -\frac{\partial (p+q)}{\partial x} + \frac{1}{r}\frac{\partial r\pi_{xx}}{\partial x} + \frac{\partial \pi_{xy}}{\partial y} - \frac{\pi_\theta}{r} + \rho g_x$$

$$\frac{\partial \rho v}{\partial t} + \frac{1}{r}\frac{\partial r\rho uv}{\partial x} + \frac{\partial \rho v^2}{\partial y}$$

$$= -\frac{\partial (p+q)}{\partial y} + \frac{1}{r}\frac{\partial r\pi_{xy}}{\partial x} + \frac{\partial \pi_{yy}}{\partial y} + \rho g_y \quad ,$$

and the mass and internal energy equations,

$$\frac{\partial \rho}{\partial t} + \frac{1}{r}\frac{\partial r\rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \quad ,$$

and

$$\frac{\partial \rho I}{\partial t} + \frac{1}{r}\frac{\partial r\rho I u}{\partial x} + \frac{\partial \rho I v}{\partial y} = -(p+q)D + \pi_{xx}\frac{\partial u}{\partial x} + \pi_{xy}\frac{\partial u}{\partial y}$$

$$+ \frac{u\pi_\theta}{r} + \pi_{xy}\frac{\partial v}{\partial x} + \pi_{yy}\frac{\partial v}{\partial y} \quad ,$$

where D is the velocity divergence,

$$D = \frac{1}{r}\frac{\partial ru}{\partial x} + \frac{\partial v}{\partial y} \quad .$$

Velocity components (u,v) are in the Cartesian coordinate directions (x,y) or the cylindrical coordinate directions (r,z). When Cartesian coordinates are desired, all radii r, which appear in these equations, are set to unity. The fluid pressure p is determined from an equation of state $p = p(\rho,I)$ and supplemented with an artificial viscous pressure q for the computation of shock waves, where

$$q = \lambda_0 \, \rho \text{ Area } D \min(0,D) \quad .$$

Artificial pressures are only used in regions of compression ($D < 0$) and are scaled proportional to the area (Area) of each computational cell, with the constant of proportionality λ_0.

The stress deviator is defined according to

$$\pi_{xx} = 2\mu\frac{\partial u}{\partial x} + \lambda D \quad ,$$

$$\pi_{yy} = 2\mu\frac{\partial v}{\partial y} + \lambda D \quad ,$$

$$\pi_\theta = \text{Cyl}\left[2\mu\frac{u}{r} + \lambda D\right] \quad ,$$

and

$$\pi_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) ,$$

in which μ is the coefficient of viscosity and λ is the coefficient of dilatational viscosity. The coefficient Cyl is zero for Cartesian coordinates and unity for cylindrical coordinates.

To facilitate its use by persons with modest experience in numerical fluid dynamics, the SALE program was written in modular form with extensive annotation and input options that provide a wide range of capabilities. In addition, SALE includes several improvements to the original YAQUI scheme that have been made since its publication. We intend that the SALE program serve not only as a useful tool for many applications, but also as a teaching aid and foundation for the development of new programs with expanded capabilities.

The basic solution algorithm for SALE appears in Sec. II of this report. Section III describes the FORTRAN program. We include a general flow diagram showing the logical partitioning of the code into a set of subroutines, each responsible for a clearly definable task. Section IV presents the results of several SALE calculations chosen to illustrate the versatility of the program.

Appendix A contains a FORTRAN listing of SALE that has been liberally annotated with comment cards, beginning with a description of the input parameters, to make it as self-explanatory as possible. Appendix B describes the functions performed by operating system CALLs appearing in SALE. For users who wish to check their results with ours, App. C presents selected plots and prints from the first calculation described in Sec. IV.

II. THE SALE SOLUTION ALGORITHM

A. The Three-Phase ICEd-ALE Approach

The basic hydrodynamic part of each cycle of SALE is divided into three phases:

1. Phase 1 is a typical, explicit Lagrangian calculation, in which the velocity field is updated by the effects of all forces.
2. Phase 2 is a Newton-Raphson iteration that provides time-advanced pressures and velocities. The purpose of Phase 2 is to allow calculations in the low-speed and even completely incompressible regimes. The implicit, iterative scheme makes this possible with greater efficiency than a purely explicit calculation with reduced time step, as it offers a numerically stable means by which pressure signals can traverse more than one cell in a time step.
3. Phase 3 performs all the advective flux calculations. This phase is required for runs that are Eulerian or contain some other form of mesh rezoning.

A powerful feature of SALE is the ease with which different phases can be combined in various ways to suit the requirements of individual problems. For example, in high-speed applications, an explicit calculation is acceptable, allowing the Phase 2 iteration to be bypassed. For an explicit Lagrangian calculation, only Phase 1 is required. For an implicit Lagrangian calculation, only the first two phases are used. In neither of these two cases are advective flux calculations necessary, and the Phase 1 or 2 results are final results for the cycle. All these options may be selected by appropriately defining the input data (see beginning of code in App. A).

B. The Computing Mesh

The computing mesh consists of a two-dimensional network of quadrilateral cells for either cylindrical or plane (Cartesian) coordinates. Calculations in cylindrical coordinates are scaled to unit azimuthal angle, which allows the equations to be written without any π factors. The radial coordinate is denoted by r or x, and the axial coordinate by y, with the origin located at the lower left corner of the mesh. The coordinate names in the equations are x and y. The quantity r is used to determine the geometry: r is set equal to x for cylindrical coordinates, but the expressions automatically reduce to Cartesian form if all r's are set to unity.

The vertices of the cells are labeled with the indices i and j, which increase in the radial and axial directions, respectively. Cell centers are denoted by half-integer indices $i+1/2$ and $j+1/2$. The mesh of cells is N_x cells wide by N_y cells high.

The mesh illustrated in Fig. 1 is in cylindrical coordinates, where the cells are sections of toroids of revolution about the cylindrical axis.

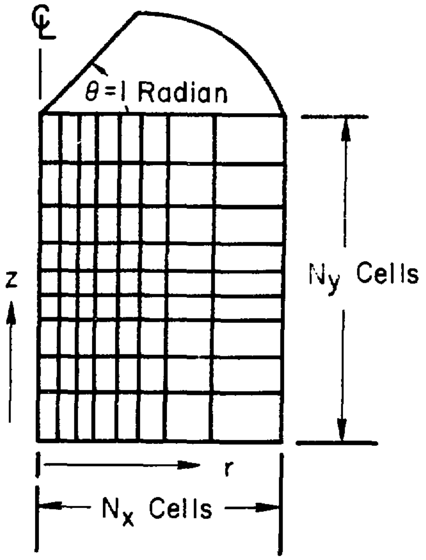

Fig. 1.
A typical SALE mesh in cylindrical coordinates.

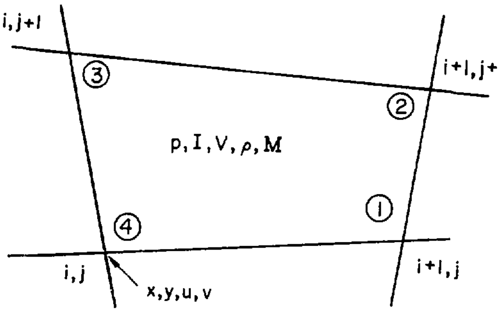

Fig. 2.
The assignment of variables about cell (i+1/2, j+1/2).

The variables in an ICEd-ALE grid are of two types: those defined at vertices and those defined at cell centers. The principal variables are shown in Fig. 2, where coordinates (x and y) and corresponding velocity components (u and v) are defined at vertices. Pressures (p), specific internal energies (I), cell volumes (V), densities (ρ), and masses (M) are all assigned at cell centers.

In the equations that follow, the superscript n refers to the beginning-of-cycle values. The advancement of the solution through a time step, of duration δt, provides values at the beginning of the next (n+1) cycle. Intermediate values are typically labeled with a subscript L for the results of Phases 1 or 2.

C. Initial Conditions and Preliminary Calculations

The input data supply the initial values of x, y, u, and v at the vertices and ρ and I for the cells.

(1) The radius r is calculated as r = x in cylindrical coordinates, or r = 1 in plane coordinates. The coordinate system is determined by the input parameter CYL, which is equal to 1 for cylindrical coordinates and is equal to zero for plane coordinates. Thus, we write

$$r_i^j = \left(x_i^j\right)\mathrm{CYL} + 1 - \mathrm{CYL} \quad .$$

(2) Cell volumes per unit azimuthal angle are given by the exact expression

$$V_{i+\frac{1}{2}}^{j+\frac{1}{2}} = \frac{1}{3}\left[(r_1 + r_2 + r_3)\mathrm{ATR} + (r_3 + r_4 + r_1)\mathrm{ABL}\right], \tag{1}$$

where

$$\mathrm{ATR} = \frac{1}{2}\,[(x_3 - x_2)(y_1 - y_2) - (x_1 - x_2)(y_3 - y_2)]$$

and

$$\mathrm{ABL} = \frac{1}{2}\,[(x_1 - x_4)(y_3 - y_4) - (x_3 - x_4)(y_1 - y_4)] \ .$$

The numerical subscript notation for vertex quantities associated with a given cell is simplified to that shown in Fig. 2. It is used throughout this report and in the SALE code.

(3) With the cell volumes defined, the masses at cell centers can be obtained from the product

$$M_{i+\frac{1}{2}}^{j+\frac{1}{2}} = \rho_{i+\frac{1}{2}}^{j+\frac{1}{2}}\, V_{i+\frac{1}{2}}^{j+\frac{1}{2}} \quad , \tag{2}$$

but it is also necessary to assign a mass to each vertex to obtain the time-advanced velocities. In SALE, we assume that the mass in each cell is shared equally between its four corner vertices, so vertex 4 in Fig. 2, for example, is given the mass

$$M_4 = \frac{1}{4}\left(M_{i+\frac{1}{2}}^{j+\frac{1}{2}} + M_{i-\frac{1}{2}}^{j+\frac{1}{2}} + M_{i-\frac{1}{2}}^{j-\frac{1}{2}} + M_{i+\frac{1}{2}}^{j-\frac{1}{2}}\right) . \tag{3}$$

D. Phase 1 of the Calculation

In this phase, velocities are advanced explicitly in time in a purely Lagrangian fashion. If viscous, elastic, or other stresses are desired, they are included in this phase as well. The updating of the specific internal energies is delayed until after the optional implicit pressure calculation of Phase 2. This delay permits time-advanced pressures to be used in computing the pdV work and ensures consistency with the velocities coming out of Phase 2.

The velocities resulting from this Lagrangian calculation phase are denoted by (u_L, v_L). Pressure, viscous, and other force contributions are computed in separate subroutines, so that in each case the (u_L, v_L) values are progressively updated with each contribution. This updating is started with the beginning-of-cycle values (u,v).

In the actual code, the order of updating is performed in the following sequence.

1. **Cycle Initialization.** This routine initializes the (u_L, v_L) velocities with the beginning-of-cycle values (u,v). In addition, cell densities ρ and ρ_L are calculated as the ratio of the cell mass to cell volume. Cell pressures (p) are calculated using an equation of state $p = f(\rho,I)$, although the equation of state is bypassed after the setup for implicit calculations, because the pressures resulting from the previous Phase 2 implicit solution generally prove to be a better initial guess for the next cycle iteration than the equation-of-state pressure. In the incompressible limit, we also bypass the equation of state in the setup and set zero pressures at time $t = 0$.

2. **Artificial Viscous Forces.** Here the (u_L,v_L) velocities are adjusted for contributions arising from a bulk artificial viscosity and from a coupling between alternate nodes.

a. ***Artificial Bulk Viscosity.*** For problems involving shock waves, an artificial pressure Q must be used to ensure mesh-resolvable shocks. This addition is required because mean kinetic energy is not conserved across a shock wave. Without dissipation, spurious velocity oscillations develop behind the shock to account for an excess of kinetic energy. We include the dissipation as a pressure addition, which models the fact that the pressure change across a shock is more than a simple adiabatic compression.

The viscous pressure used in SALE is quadratic in the velocity divergence, and is only added to cells undergoing compression,

$$Q^j_i = \min\left(0,\ D^j_i\right)\left[\lambda_0 \rho^j_i D^j_i (\text{Area})\right] \quad . \tag{4}$$

In this expression, (Area) is the area of cell (i,j), so that

$$(\text{Area}) = \frac{1}{2}\Big[(x_2 - x_4)(y_3 - y_1) - (x_1 - x_3)(y_4 - y_2)\Big] \quad ,$$

and D^j_i is its velocity divergence $\nabla \cdot \vec{u}$ defined as

$$D^j_i = \frac{1}{2(\text{Area})}\Big[(u_2 - u_4)(y_3 - y_1) - (u_1 - u_3)(y_4 - y_2) + (v_4 - v_2)(x_3 - x_1) - (v_1 - v_3)(x_2 - x_4)\Big] + \frac{u}{r} \quad , \tag{5}$$

where

$$\frac{u}{r} = \text{CYL}\left(\frac{u_1 + u_2 + u_3 + u_4}{r_1 + r_2 + r_3 + r_4}\right) \quad . \tag{6}$$

The parameter λ_0 in the above expression for Q^j_i is denoted by ARTVIS in the input data list, and should be less than 0.25 to avoid excessive viscous damping. A value of ARTVIS = 0.1 has been satisfactory for many applications.

With Q^j_i calculated, the appropriate contributions to the four vertices of cell (i,j) are

$$(u_L)_1 = u_1 + \frac{\delta t Q^j_i}{2M_1} r_1 (y_2 - y_4) \quad ,$$

$$(u_L)_2 = u_2 + \frac{\delta t Q^j_i}{2M_2} r_2 (y_3 - y_1) \quad ,$$

$$(u_L)_3 = u_3 - \frac{\delta t Q^j_i}{2M_3} r_3 (y_2 - y_4) \quad ,$$

$$(u_L)_4 = u_4 - \frac{\delta t Q^j_i}{2M_4} r_4 (y_3 - y_1) \quad ,$$

$$(v_L)_1 = v_1 - \frac{\delta t Q^j_i}{4M_1} (r_2 + r_4)(x_2 - x_4) \quad ,$$

$$(v_L)_2 = v_2 - \frac{\delta t Q^j_i}{4M_2} (r_1 + r_3)(x_3 - x_1) \quad ,$$

$$(v_L)_3 = v_3 + \frac{\delta t Q^j_i}{4M_3} (r_2 + r_4)(x_2 - x_4) \quad ,$$

and

$$(v_L)_4 = v_4 + \frac{\delta t Q^j_i}{4M_4} (r_1 + r_3)(x_3 - x_1) \quad . \tag{7}$$

The asymmetry in the geometric factors in the above expressions, which also appears in other equations for pressure accelerations, arises from the difference in the effect of the boundary of the control volume on the two directions. Accelerations in the radial direction must include the forces on the ends of the one-radian section of the torus. These contributions do not enter in the axial direction.

b. Alternate Node Coupler. In a Lagrangian calculation using quadrilateral mesh cells, there are certain degenerate mesh deformations that do not result in net pressure or viscous forces. Typically, these deformations are associated with the shortest resolvable wavelengths (2δx) in the mesh. For example, Fig. 3 illustrates two such short-wavelength deformations. Figure 3a shows the bowtie pattern and Fig. 3b shows the herringbone pattern. In each case, the deforming cells undergo no change in volume so that no pressure variations are generated. Also, it is easily verified that no net viscous or elastic strain forces are generated at vertices embedded in the bowtie type of deformation.

Thus, to prevent such deformations from slowly degrading a solution, it is sometimes necessary to couple alternate mesh nodes with a small artificial restoring force. Ideally, this force should affect flows only at the 2δx wavelength level, but have no influence on the larger, better resolved flow variations. We introduce small accelerations at each vertex, which are based on the surrounding velocity field and tend to keep the vertex velocities from deviating too strongly from their neighbors.

A fourth-order coupling scheme is effective for the bowtie mode, but a more diffusive second-order scheme must be used for the herringbone pattern. The fourth-order form is given by

$$u_i^j = u_i^j + \frac{a_{nc}}{4}\Bigl[2\left(u_{i+1}^j + u_i^{j+1} + u_{i-1}^j + u_i^{j-1}\right) - u_{i+1}^{j+1} - u_{i-1}^{j+1} - u_{i-1}^{j-1} - u_{i+1}^{j-1} - 4u_i^j\Bigr] ,$$

in which a_{nc} is a coefficient that governs the amount of coupling and implies a relaxation time of a_{nc}^{-1} time steps.

The second-order form is given by

$$u_i^j = u_i^j + \frac{a_{nc}}{4}\left[\left(u_{i+1}^j + u_j^{j+1} + u_{i-1}^j + u_i^{j-1}\right) - 4u_i^j\right] .$$

In SALE, we combine both of these forms in the following set of expressions, in which $\xi = 1$ results in the fourth-order form and $\xi = 0$ results in the second-order form. Also, rather than sweeping vertices to make the contributions, we may equivalently sweep over cells and adjust the four vertices of each cell, such that

$$(u_L)_1 = (u_L)_1 + \frac{a_{nc}}{4}\left[\left(\frac{1+\xi}{2}\right)(u_2 + u_4) - \xi u_3 - u_1\right] ,$$

$$(u_L)_2 = (u_L)_2 + \frac{a_{nc}}{4}\left[\left(\frac{1+\xi}{2}\right)(u_3 + u_1) - \xi u_4 - u_2\right] ,$$

$$(u_L)_3 = (u_L)_3 + \frac{a_{nc}}{4}\left[\left(\frac{1+\xi}{2}\right)(u_4 + u_2) - \xi u_1 - u_3\right] ,$$

and

$$(u_L)_4 = (u_L)_4 + \frac{a_{nc}}{4}\left[\left(\frac{1+\xi}{2}\right)(u_1 + u_3) - \xi u_2 - u_4\right] . \tag{8}$$

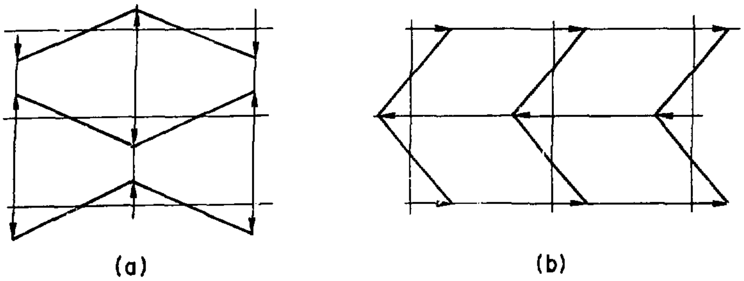
(a) (b)

Fig. 3.
Two types of instabilities between adjacent nodes: (a) bowtie, (b) herringbone.

Corresponding expressions are used for the y direction, with every u or u_L replaced by v or v_L. Note that contributions at vertices on reflective boundaries must be doubled to obtain the correct value. This is necessary here because these artificial accelerations have been defined without reference to vertex masses. In the case of all other forces, no corrections are needed for boundary vertices, because the omission of force contributions from cells on the outside of a boundary is compensated for by a corresponding omission in vertex mass.

We emphasize that node coupling is diffusive and nonphysical and should be used with discretion. In a spherical expansion, for example, the smoothing effect of too much node coupling adversely affects the sphericity. To avoid its unintentional use, we require the SALE user to supply values for ξ and a_{nc} in the input data. Rarely should a_{nc} exceed 0.05.

3. Stress Deviator Forces. At this point, the shear viscosity (μ) and bulk viscosity (λ) contributions are added, if either is specified, in terms of a stress deviator force. (This would also be the appropriate place to add material strength effects. These effects, however, have not been included in this version of SALE.)

For each cell, we define the divergence $D = \nabla \cdot \vec{u}$ as in step 1 above, and the four components of the viscous stress tensor as

$$\Pi_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda \nabla\cdot\vec{u},$$

$$\Pi_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda \nabla\cdot\vec{u},$$

$$\Pi_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \quad ,$$

and

$$\Pi_\theta = \mathrm{CYL}\left[2\mu\left(\frac{u}{r}\right) + \lambda\nabla\cdot\vec{u}\right] \quad , \tag{9}$$

where u/r is defined as in Eq. (6). The finite difference expressions to compute these quantities are

$$\frac{\partial u}{\partial x} = \frac{1}{2(\mathrm{Area})}\,[(u_2 - u_4)(y_3 - y_1) - (u_3 - u_1)(y_2 - y_4)] \quad ,$$

$$\frac{\partial v}{\partial x} = \frac{1}{2(\mathrm{Area})}\,[(v_2 - v_4)(y_3 - y_1) - (v_3 - v_1)(y_2 - y_4)] \quad ,$$

$$\frac{\partial u}{\partial y} = \frac{1}{2(\mathrm{Area})}\,[(u_3 - u_1)(x_2 - x_4) - (u_2 - u_4)(x_3 - x_1)] \quad ,$$

and

$$\frac{\partial v}{\partial y} = \frac{1}{2(\mathrm{Area})}\,[(v_3 - v_1)(x_2 - x_4) - (v_2 - v_4)(x_3 - x_1)] \quad . \tag{10}$$

Stress deviator contributions to the vertex velocities are

$$(u_L)_1 = (u_L)_1 + \frac{\delta t}{4M_1}(r_2 + r_4)\left[\Pi_{xy}(x_2 - x_4) - \Pi_{xx}(y_2 - y_4) - \frac{\mathrm{Area}}{2}\Pi_\theta\right] \quad ,$$

$$(u_L)_2 = (u_L)_2 + \frac{\delta t}{4M_2}(r_1 + r_3)\left[\Pi_{xy}(x_3 - x_1) - \Pi_{xx}(y_3 - y_1) - \frac{\mathrm{Area}}{2}\Pi_\theta\right] \quad ,$$

$$(u_L)_3 = (u_L)_3 - \frac{\delta t}{4M_3}(r_2 + r_4)\left[\Pi_{xy}(x_2 - x_4) - \Pi_{xx}(y_2 - y_4) + \frac{\mathrm{Area}}{2}\Pi_\theta\right] \quad ,$$

$$(u_L)_4 = (u_L)_4 - \frac{\delta t}{4M_4}(r_1 + r_3)\left[\Pi_{xy}(x_3 - x_1) - \Pi_{xx}(y_3 - y_1) + \frac{\mathrm{Area}}{2}\Pi_\theta\right] \quad ,$$

$$(v_L)_1 = (v_L)_1 + \frac{\delta t}{4M_1}(r_2 + r_4)[\Pi_{yy}(x_2 - x_4) - \Pi_{xy}(y_2 - y_4)] \quad ,$$

$$(v_L)_2 = (v_L)_2 + \frac{\delta t}{4M_2} (r_1 + r_3)[\Pi_{yy}(x_3 - x_1) - \Pi_{xy}(y_3 - y_1)] \quad ,$$

$$(v_L)_3 = (v_L)_3 - \frac{\delta t}{4M_3} (r_2 + r_4)[\Pi_{yy}(x_2 - x_4) - \Pi_{xy}(y_2 - y_4)] \quad ,$$

and

$$(v_L)_4 = (v_L)_4 - \frac{\delta t}{4M_4} (r_1 + r_3)[\Pi_{yy}(x_3 - x_1) - \Pi_{xy}(y_3 - y_1)] \quad . \qquad (11)$$

The Π terms are stored for later inclusion in the internal energy.

4. **Pressure Force Contributions.** The principal contribution to the velocities in Phase 1 comes from the pressure forces and body forces acting on the vertices.

a. Pressure Accelerations. The difference approximations used for the pressure accelerations are

$$(u_L)_1 = (u_L)_1 + \frac{\delta t}{2M_1} p_i^j \, r_1(y_2 - y_4) \quad ,$$

$$(u_L)_2 = (u_L)_2 + \frac{\delta t}{2M_2} p_i^j \, r_2(y_3 - y_1) \quad ,$$

$$(u_L)_3 = (u_L)_3 - \frac{\delta t}{2M_3} p_i^j \, r_3(y_2 - y_4) \quad ,$$

$$(u_L)_4 = (u_L)_4 - \frac{\delta t}{2M_4} p_i^j \, r_4(y_3 - y_1) \quad ,$$

$$(v_L)_1 = (v_L)_1 - \frac{\delta t}{4M_1} p_i^j(r_2 + r_4)(x_2 - x_4) \quad ,$$

$$(v_L)_2 = (v_L)_2 - \frac{\delta t}{4M_2} p_i^j(r_1 + r_3)(x_3 - x_1) \quad ,$$

$$(v_L)_3 = (v_L)_3 + \frac{\delta t}{4M_3} p_i^j(r_2 + r_4)(x_2 - x_4) \quad ,$$

and

$$(v_L)_4 = (v_L)_4 + \frac{\delta t}{4M_4} p_i^j(r_1 + r_3)(x_3 - x_1) \quad . \qquad (12)$$

b. Body Accelerations. Finally, any desired body accelerations, such as those arising from gravitational effects, are added to the velocities. For example,

$$(u_L)_i^j = (u_L)_i^j + \delta t g_x$$

and

$$(v_L)_i^j = (v_L)_i^j + \delta t g_y \quad . \qquad (13)$$

E. Phase 2 of the Calculation

Phase 2 provides an implicit treatment required to eliminate Courant-like time step restrictions that would otherwise be required to ensure computational stability in low-speed or incompressible flows. This phase can be bypassed entirely when an explicit calculation will suffice. The purpose of the implicit treatment in Phase 2 is to obtain a velocity field that has been accelerated by time-advanced pressure gradients. The time-advanced pressures, in turn, depend upon the densities and energies obtained when vertices are moved with these new velocities, but because these are functions of the new pressures, the pressures are by definition implicit and are in general best determined by iteration. Our implicit approach is formulated as follows. With the subscript L again denoting time-advanced values, the desired pressure p_L of cell (i,j) will be the solution of

$$\left(p_L\right)_i^j = f\left[\left(\rho_L\right)_i^j, \left(I_L\right)_i^j\right] \quad , \qquad (14)$$

where the new cell density and energy are approximated in terms of their initial values as

$$\left(\rho_L\right)_i^j = \rho_i^j \left(V/V*\right)_i^j$$

and

$$\left(I_L\right)_i^j = I_i^j + \left(p_L\right)_i^j (1 - V*/V)/\left(\rho_L\right)_i^j \quad . \qquad (15)$$

V is the volume of the cell at time n, and V* is the volume the cell would have if its vertices were moved according to the current Lagrangian velocity field,

$$x_1^* = x_1 + (u_L)_1 \delta t \quad , \quad y_1^* = y_1 + (v_L)_1 \delta t \quad , \; \ldots . \tag{16}$$

A solution for p_L is obtained by applying a Newton-Raphson iteration, for which the Phase 1 velocities (u_L, v_L) are used as initial guesses. The iteration consists of sweeping through the mesh and applying the following adjustments to each cell, once each sweep:

(1) Compute V* using the most updated values for (u_L, v_L);

(2) Compute new guesses for ρ_L, I_L, and p_L from the above equations; and

(3) Compute a pressure change δp, according to

$$\delta p = - \frac{p_L - f\left(\rho_L, I_L\right)}{S} \quad , \tag{17}$$

where the most updated values are used for p_L, ρ_L, and I_L, and S^{-1} is a relaxation factor to be described below.

(4) Adjust the current guess for p_L by adding δp to it;

(5) Adjust the velocities at the vertices of the cell to include this pressure change:

$$(u_L)_1 = (u_L)_1 + \frac{\delta t \delta p}{2M_1} r_1 (y_2 - y_4) \quad ,$$

$$(u_L)_2 = (u_L)_2 + \frac{\delta t \delta p}{2M_2} r_2 (y_3 - y_1) \quad ,$$

$$(u_L)_3 = (u_L)_3 - \frac{\delta t \delta p}{2M_3} r_3 (y_2 - y_4) \quad ,$$

$$(u_L)_4 = (u_L)_4 - \frac{\delta t \delta p}{2M_4} r_4 (y_3 - y_1) \quad ,$$

$$(v_L)_1 = (v_L)_1 - \frac{\delta t \delta p}{4M_1} (r_2 + r_4)(x_2 - x_4) \quad ,$$

$$(v_L)_2 = (v_L)_2 - \frac{\delta t \delta p}{4M_2} (r_1 + r_3)(x_3 - x_1) \quad ,$$

$$(v_L)_3 = (v_L)_3 + \frac{\delta t \delta p}{4M_3} (r_2 + r_4)(x_2 - x_4) \quad ,$$

and

$$(v_L)_4 = (v_L)_4 + \frac{\delta t \delta p}{4M_4} (r_1 + r_3)(x_3 - x_1) \quad . \tag{18}$$

The mesh is repeatedly swept and steps (1) through (5) are preformed once for each cell each sweep, until no cell exhibits a pressure change violating the inequality

$$\frac{|\delta p|}{|p_{max}|} < \varepsilon \quad , \tag{19}$$

where p_{max} is the actual or an estimated maximum pressure in the mesh and ε is an input number (EPS), typically of order 10^{-4}.

The quantity S used in step (3) must be chosen to keep the pressure changes bounded and progressing in the right direction. In the Newton-Raphson procedure, S is the derivative of the function whose root is sought with respect to p, the iteration variable. Here, S is the rate at which the quantity $p - f(\rho, I)$ changes as the variable p changes, and is computed numerically using the same relations outlined above. For this purpose, a small pressure change Δp is chosen, scaled to the calculation:

$$\Delta p = \frac{1}{\delta t^2} \left[\frac{p_\varepsilon \rho}{2\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2}\right)} \right] \quad .$$

Here, ρ is a typical fluid density at time t = 0, and p_ε is an input quantity (PEPS), typically 10^{-4}. The velocity changes that would be induced by Δp are used to compute the corresponding volume, energy, and density changes, and from them a new pressure. Finally, S is determined from the difference between $p - f(\rho, I)$, evaluated before and after the small change in pressure, and divided by Δp. The resulting values for S^{-1} are multiplied by an optional over-relaxation coefficient, input as OM, and stored for each cell before the iteration is begun. These quantities are not recomputed during the iteration.

The above procedure works well across a broad range of low-speed flow applications, but in the incompressible limit, the procedure is modified. The reason for this is that the method is then excessively sensitive to volume changes. In this case, we replace $p = f(\rho, I)$ with

$$p = p_L + \frac{\rho_L}{\rho} - 1 \quad , \tag{20}$$

which effectively holds the densities constant and results in much faster iteration convergence. Corresponding expressions are used for the evaluation of S.

It should be noted that the densities and internal energies calculated in the pressure iteration are temporary quantities used only to update the pressure. To ensure exact mass conservation, the final new densities are computed in Phase 3 after the cell masses and volumes have been calculated. The internal energy is also recalculated with time-centered volume changes, which conserves internal energy, and viscous contributions are then included.

In some cases, when the pressure iteration does not converge within several hundred iterations, it is still possible to continue a calculation without serious error. Usually this only happens when the incompressible option is used. For example, a poor initial guess for velocities or pressures may require a high number of iterations to relax to an acceptable solution. In such cases, the code automatically terminates the iteration, continues the cycle, and then proceeds on to the next cycle and repeats this process up to 10 times. The code aborts if the pressure iteration still has not converged after 10 cycles.

The Phase 1 and 2 calculations as outlined above comprise an implicit Lagrangian method stable for any Courant number, and allow calculations at all values of sound speed vs fluid speed.

F. The Energy Calculation

The pressure work and viscous dissipation contributions to internal energy are calculated next. The pressures used in the work expression are those resulting from the Phase 2 iteration when the implicit option is used. In the case of explicit calculations, the pressures used are those coming from the equation of state at the beginning of each cycle.

The equation for the change in internal energy in a cycle is

$$I_i^j = I_i^j - \frac{\delta t}{2M_i^j}\left[\left(P_i^j + Q_i^j\right)\frac{dV}{dt} + \frac{dVIS}{dt}\right] . \qquad (21)$$

Both the dV/dt and dVIS/dt quantities are in time-centered form, using averages of beginning-of-cycle and current velocities, for example

$$(u_{TC})_1 = \frac{1}{2}\left[u_1 + (u_L)_1\right]$$

and

$$(v_{TC})_1 = \frac{1}{2}\left[v_1 + (v_L)_1\right] \quad , \ldots .$$

With this definition,

$$\frac{dV}{dt} = (y_2 - y_4)[r_1(u_{TC})_1 - r_3(u_{TC})_3]$$

$$+ (y_3 - y_1)[r_2(u_{TC})_2 - r_4(u_{TC})_4]$$

$$- \frac{1}{2}(r_2 + r_4)(x_2 - x_4)[(v_{TC})_1 - (v_{TC})_3]$$

$$- \frac{1}{2}(r_1 + r_3)(x_3 - x_1)[(v_{TC})_2 - (v_{TC})_4]$$

and

$$\frac{dVIS}{dt} = \frac{1}{2}(r_2 + r_4)[\Pi_{xy}(x_2 - x_4)$$

$$- \Pi_{xx}(y_2 - y_4)][(u_{TC})_1 - (u_{TC})_3]$$

$$+ \frac{1}{2}(r_1 + r_3)[\Pi_{xy}(x_3 - x_1)$$

$$- \Pi_{xx}(y_3 - y_1)][(u_{TC})_2 - (u_{TC})_4]$$

$$- \frac{1}{2}(\Pi_\theta \text{ Area})[(u_{TC})_1 + (u_{TC})_2 + (u_{TC})_3 + (u_{TC})_4]$$

$$+ \frac{1}{2}(r_2 + r_4)[\Pi_{yy}(x_2 - x_4)$$

$$- \Pi_{xy}(y_2 - y_4)][(v_{TC})_1 - (v_{TC})_3]$$

$$+ \frac{1}{2}(r_1 + r_3)[\Pi_{yy}(x_3 - x_1)$$

$$- \Pi_{xy}(y_3 - y_1)][(v_{TC})_2 - (v_{TC})_4] \quad .$$

The four Π terms were evaluated in the Phase 1 stress deviator calculation using Eq. (9) and stored for each cell for use here. In addition, the artificial viscous pressure Q was saved from Phase 1, Eq. (4).

G. Phase 3 of the Calculation

1. Rezone. When large fluid distortions are not expected, a purely Lagrangian approach will suffice, allowing the computing grid to follow the fluid motion exactly. In many cases, however, large fluid motions would create devastating effects, contorting cells to extreme aspect ratios or even turning cells inside out. It is often possible to ameliorate these effects by moving the mesh vertices with respect to the fluid so as to maintain a reasonable mesh structure. Whenever a vertex is moved relative to the fluid, however, there must be an exchange of material among the cells surrounding the vertex. SALE allows a broad spectrum of rezoning possibilities by treating this material exchange as an advective flux. The simplest case is that of a purely Eulerian flow, in which the vertices are moved back to their original positions every cycle. Between this extreme and the Lagrangian extreme lies whatever form of continuous or discrete rezoning the user wishes.

This latitude is made possible by defining a set of grid vertex velocities (u_G,v_G) over the entire mesh. For a purely Lagrangian calculation, $u_G \equiv u_L$ and $v_G \equiv v_L$ everywhere. For a purely Eulerian calculation, $u_G \equiv 0$ and $v_G \equiv 0$. For a continuous rezone that approximates a Lagrangian calculation, but minimizes excessive grid distortions, grid velocities are chosen to lie somewhere between these two extremes. In particular, the vertices are moved according to some relaxation rate to place vertices at the average position of the neighboring vertices. This usually maintains cells of reasonable size and proportion throughout a run. Once a set of grid velocities u_G and v_G have been defined, it is a simple matter to construct the new grid and perform whatever advective flux calculations that may be required.

SALE could also be modified to have a discontinuous rezone capability in this Phase of a cycle. For example, grid quantities could be interpolated onto another grid whenever distortions are excessive.

2. Regrid. In this step, the vertices are moved to new locations as specified by (u_G,v_G):

$$x_i^j = x_i^j + \delta t \left(u_G\right)_i^j \quad ,$$

$$y_i^j = y_i^j + \delta t \left(v_G\right)_i^j \quad ,$$

and

$$r_i^j = \left(x_i^j\right) \mathrm{CYL} + 1 - \mathrm{CYL} \quad . \tag{22}$$

We next form a set of relative velocities (u_{REL},v_{REL}) to simplify the later task of calculating advective fluxes. For this purpose, the vertex velocities with respect to the fluid are

$$\left(u_{REL}\right)_i^j = \left(u_G\right)_i^j - \left(u_L\right)_i^j$$

and

$$\left(v_{REL}\right)_i^j = \left(v_G\right)_i^j - \left(v_L\right)_i^j \quad . \tag{23}$$

New cell volumes ${}^{n+1}V$ are calculated from the new coordinates using Eq. (1), and replace the ${}^{n}V$ values in storage.

3. Advective Flux of Mass, Energy, and Momentum. This step is bypassed completely for a purely Lagrangian calculation. In all other cases the relative velocities are not zero, and we must calculate the flux of mass, energy, and momentum between cells.

The flux calculation is performed on a cell-by-cell basis. For every cell, we calculate the volume swept out by each of the four faces relative to their Lagrangian positions.

a. To calculate these volumes, it is necessary to first form the Lagrangian coordinates (x_p,x_p) given by

$$(x_p)_i = x_i - (u_{REL})_i \delta t \quad ,$$

$$(y_p)_i = y_i - (v_{REL})_i \delta t \quad ,$$

and

$$(r_p)_i = (x_p)_i \mathrm{CYL} + 1 - \mathrm{CYL}, \ \ldots . \tag{24}$$

b. Then the four volumes for the right, top, left, and bottom sides are proportional to

$$FR = \frac{1}{12}\Big([r_1 + (r_p)_1 + (r_p)_2]\{x_1[(y_p)_2 - (y_p)_1]$$

$$+ (x_p)_1[y_1 - (y_p)_2]$$

$$+ (x_p)_2[(y_p)_1 - y_1]\}$$

$$+ [r_1 + r_2 + (r_p)_2]\{x_1[y_2 - (y_p)_2]$$

$$+ x_2[(y_p)_2 - y_1]$$

$$+ (x_p)_2[y_1 - y_2]\}\Big) \quad ,$$

$$FT = \frac{1}{12}\Big([(r_p)_2 + r_3 + (r_p)_3]\{(x_p)_2[y_3 - (y_p)_3]$$

$$+ x_3[(y_p)_3 - (y_p)_2]$$

$$+ (x_p)_3[(y_p)_2 - y_3]\}$$

$$+ [r_2 + (r_p)_2 + r_3]\{x_2[y_3 - (y_p)_2]$$

$$+ (x_p)_2[y_2 - y_3]$$

$$+ x_3[(y_p)_2 - y_2]\}\Big) \quad ,$$

$$FL = \frac{1}{12}\Big([(r_p)_3 + r_4 + (r_p)_4]\{(x_p)_3[y_4 - (y_p)_4]$$

$$+ x_4[(y_p)_4 - (y_p)_3]$$

$$+ (x_p)_4[(y_p)_3 - y_4]\}$$

$$+ [r_3 + (r_p)_3 + r_4]\{x_3[y_4 - (y_p)_3]$$

$$+ (x_p)_3[y_3 - y_4]$$

$$+ x_4[(y_p)_3 - y_3]\}\Big) \quad ,$$

and

$$FB = \frac{1}{12}\Big([r_1 + (r_p)_1 + (r_p)_4]\{x_1(y_p)_1 - (y_p)_4]$$

$$+ (x_p)_1[(y_p)_4 - y_1]$$

$$+ (x_p)_4[y_1 - (y_p)_1]\}$$

$$+ [r_1 + r_4 + (r_p)_4]\{x_1[(y_p)_4 - y_4]$$

$$+ x_4[y_1 - (y_p)_4]$$

$$+ (x_p)_4[y_4 - y_1]\}\Big) \quad . \qquad (25)$$

These represent *one-half* the volumes swept over by the sides moving from their Lagrangian positions to their rezoned positions, the factor of 1/2 being included for convenience. Note also that FR for cell (i+1/2, j+1/2) is equal to −FL for cell (i+3/2, j+1/2), and FT for cell (i+1/2, j+1/2) is equal to −FB for cell (i+1/2, j+3/2). This fact is used in the code to eliminate redundant calculations.

c. Associated with each fluid volume crossing a cell face there are corresponding values of mass, energy, and momentum. For example, the mass crossing the right face of cell (i+1/2, j+1/2) might be computed as the product of the fluxing volume 2(FR) times the average fluid density of the cells (i+1/2, j+1/2) and (i+3/2, j+1/2) located on either side of the boundary. Unfortunately, this so-called 'centered differencing' leads to numerical instabilities. One way to circumvent this instability is to weight the quantity being fluxed more in favor of the upstream value. In the above example, this means the density associated with FR should be more nearly equal to the density in cell (i+1/2, j+1/2) when the flux is leaving this cell (FR < 0), or more nearly equal to the density in cell (i+3/2, j+1/2) when the flux is leaving that cell (FR > 0). In SALE, we incorporate the flux coefficients FR, FT, FL, and FB within expressions that allow various differencing forms determined from input constants a_0 and b_0:

$$a_R = a_0 \text{ sign FR} + 4b_0\text{FR} \Big/ \left(V^{j+\frac{1}{2}}_{i+3/2} + V^{j+\frac{1}{2}}_{i+\frac{1}{2}}\right) \quad ,$$

$$a_T = a_0 \text{ sign FT} + 4b_0\text{FT} \Big/ \left(V^{j+3/2}_{i+\frac{1}{2}} + V^{j+\frac{1}{2}}_{i+\frac{1}{2}}\right) \quad ,$$

$$a_L = a_0 \text{ sign FL} + 4b_0\text{FL} \Big/ \left(V^{j+\frac{1}{2}}_{i-\frac{1}{2}} + V^{j+\frac{1}{2}}_{i+\frac{1}{2}}\right) \quad ,$$

and

$$a_B = a_0 \text{ sign FB} + 4b_0\text{FB} \Big/ \left(V^{j-\frac{1}{2}}_{i+\frac{1}{2}} + V^{j+\frac{1}{2}}_{i+\frac{1}{2}}\right) \quad , \qquad (26)$$

where "sign FR," for example, equals +1 if FR $\geqslant 0$ and equals −1 if FR < 0. Both a_0 and b_0 lie in the range 0 to 1, and the limiting cases are

$a_0 = 0$ and $b_0 = 0$ → centered (unstable),

$a_0 = 1$ and $b_0 = 0$ → full donor cell or upstream differencing (stable, but diffusive),

$a_0 = 0$ and $b_0 = 1$ → interpolated donor cell (linearly stable, less diffusive),

and

$a_0 = 1$ and $b_0 = 1$ → (stable, but more diffusive).

Note that $(a_0 + b_0)$ must be sufficiently positive for numerical stability (see Sec. III.D).

d. In terms of these weighting fractions, the new mass and specific internal energy for a cell (i+1/2, j+1/2) are then given by

$$\begin{aligned}
{}^{n+1}M^{j+\frac{1}{2}}_{i+\frac{1}{2}} = {}^{n}M^{j+\frac{1}{2}}_{i+\frac{1}{2}} &+ \text{FR}(1 + a_R)\rho_L{}^{j+\frac{1}{2}}_{i+3/2} \\
&+ \text{FT}(1 + a_T)\rho_L{}^{j+3/2}_{i+\frac{1}{2}} \\
&+ \text{FL}(1 + a_L)\rho_L{}^{j+\frac{1}{2}}_{i-\frac{1}{2}} + \text{FB}(1 + a_B)\rho_L{}^{j-\frac{1}{2}}_{i+\frac{1}{2}} \\
&+ [\text{FR}(1 - a_R) + \text{FT}(1 - a_T) + \text{FL}(1 - a_L) \\
&+ \text{FB}(1 - a_B)]\rho_L{}^{j+\frac{1}{2}}_{i+\frac{1}{2}}
\end{aligned}$$

and

$$\begin{aligned}
{}^{n+1}I^{j}_{i} = \frac{1}{{}^{n+1}M^{j+\frac{1}{2}}_{i+\frac{1}{2}}} \Bigg\{ {}^{n}(MI)^{j+\frac{1}{2}}_{i+\frac{1}{2}} &+ \text{FR}(1 + a_R)\left({}^{n}I\rho_L\right)^{j+\frac{1}{2}}_{i+3/2} \\
&+ \text{FT}(1 + a_T)\left({}^{n}I\rho_L\right)^{j+3/2}_{i+\frac{1}{2}} \\
&+ \text{FL}(1 + a_L)\left({}^{n}I\rho_L\right)^{j+\frac{1}{2}}_{i-\frac{1}{2}} + \text{FB}(1 + a_B)\left({}^{n}I\rho_L\right)^{j-\frac{1}{2}}_{i+\frac{1}{2}} \\
&+ [\text{FR}(1 - a_R) + \text{FT}(1 - a_T) + \text{FL}(1 - a_L) \\
&+ \text{FB}(1 - a_B)]\left({}^{n}I\rho_L\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \Bigg\} \ .
\end{aligned}$$

e. The advection of momentum requires an extra step, because cell momenta are not carried throughout the cycle as primary field variables. Here we use the concept of a cell-centered momentum flux, which is a departure from that of a vertex-centered form previously used.[3] The cell-centered flux form has the advantage that momentum is fluxed consistently with mass and energy. The approach is to compute average cell-centered momenta based on the vertex velocities. Changes in these cell momenta resulting from advection are computed in the same way as the other cell-centered quantities. These changes are then apportioned back to the vertices. Although this scheme requires the additional calculation of cell-centered averages and their average effect back on the vertex velocities, the entire process is simpler than using another set of control volumes, because it can be easily included in the advection calculation for mass and energy. Tests with this method have shown it to be superior to all momentum advection methods based on vertex-centered control volumes. In particular, it better preserves cylindrical or spherical symmetry and does not introduce diffusion across streamlines.

The first step in the momentum flux calculation is to form the cell centered momenta. For every cell,

$$\left(\text{UMØM}_L\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} = \frac{\left(\rho_L\right)^{j}_{i}}{4}\left[(u_L)_1 + (u_L)_2 + (u_L)_3 + (u_L)_4\right]$$

and

$$\left(\text{VMØM}_L\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} = \frac{\left(\rho_L\right)^{j}_{i}}{4}\left[(v_L)_1 + (v_L)_2 + (v_L)_3 + (v_L)_4\right] .$$

Then, using the flux coefficients formed in steps (b) and (c) above, the net advection changes in cell-centered momentum components are given by

$$\left(\Delta\mathrm{UM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} = \mathrm{FR}(1 + a_R)\left(\mathrm{UMOM}_L\right)^{j+\frac{1}{2}}_{i+3/2}$$

$$+ \mathrm{FT}(1 + a_T)\left(\mathrm{UMOM}_L\right)^{j+3/2}_{i+\frac{1}{2}}$$

$$+ \mathrm{FL}(1 + a_L)\left(\mathrm{UMOM}_L\right)^{j+\frac{1}{2}}_{i-\frac{1}{2}} + \mathrm{FB}(1 + a_B)\left(\mathrm{UMOM}_L\right)^{j-\frac{1}{2}}_{i+\frac{1}{2}}$$

$$+ [\mathrm{FR}(1 - a_R) + \mathrm{FT}(1 - a_T) + \mathrm{FL}(1 - a_L)$$

$$+ \mathrm{FB}(1 - a_B)]\left(\mathrm{UMOM}_L\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}}$$

and

$$\left(\Delta\mathrm{VM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} = \mathrm{FR}(1 + a_R)\left(\mathrm{VMOM}_L\right)^{j+\frac{1}{2}}_{i+3/2}$$

$$+ \mathrm{FT}(1 + a_T)\left(\mathrm{VMOM}_L\right)^{j+3/2}_{i+\frac{1}{2}}$$

$$+ \mathrm{FL}(1 + a_L)\left(\mathrm{VMOM}_L\right)^{j+\frac{1}{2}}_{i-\frac{1}{2}} + \mathrm{FR}(1 + a_B)\left(\mathrm{VMOM}_L\right)^{j-\frac{1}{2}}_{i+\frac{1}{2}}$$

$$+ [\mathrm{FR}(1 - a_R) + \mathrm{FT}(1 - a_T) + \mathrm{FL}(1 - a_L)$$

$$+ \mathrm{FB}(1 - a_B)]\left(\mathrm{VMOM}_L\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad .$$

The momenta changes are finally converted to vertex velocity changes in the next step.

4. Updating the Vertex Quantities.

a. We first calculate new vertex masses from averages of the new cell masses, using Eq. (3).

b. To adjust the velocities, we set initial values at all vertices, where the (${}^{n}u, {}^{n}v$) values in storage are replaced by

$${}^{n+1}u^j_i = \left(\frac{{}^{n}M}{{}^{n+1}M}\right)^j_i \left(u_L\right)^j_i \quad ,$$

and

$${}^{n+1}v^j_i = \left(\frac{{}^{n}M}{{}^{n+1}M}\right)^j_i \left(v_L\right)^j_i \quad .$$

Because both the ${}^{n+1}M$ and the ${}^{n}M$ values are required, the replacement of ${}^{n}M$ values by ${}^{n+1}M$ values is deferred until after completion of this step.

c. *Finally, we distribute the cell-centered momentum* changes to the four vertices of the cell, in the same manner that we calculate vertex masses, that is, giving equal fractions to each vertex,

$${}^{n+1}u_1 = {}^{n+1}u_1 + \left(\frac{0.25}{{}^{n+1}M_1}\right)\left(\Delta\mathrm{UM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad ,$$

$${}^{n+1}u_2 = {}^{n+1}u_2 + \left(\frac{0.25}{{}^{n+1}M_2}\right)\left(\Delta\mathrm{UM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad ,$$

$${}^{n+1}u_3 = {}^{n+1}u_3 + \left(\frac{0.25}{{}^{n+1}M_3}\right)\left(\Delta\mathrm{UM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad ,$$

$${}^{n+1}u_4 = {}^{n+1}u_4 + \left(\frac{0.25}{{}^{n+1}M_4}\right)\left(\Delta\mathrm{UM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad ,$$

$${}^{n+1}v_1 = {}^{n+1}v_1 + \left(\frac{0.25}{{}^{n+1}M_1}\right)\left(\Delta\mathrm{VM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad ,$$

$${}^{n+1}v_2 = {}^{n+1}v_2 + \left(\frac{0.25}{{}^{n+1}M_2}\right)\left(\Delta\mathrm{VM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad ,$$

$${}^{n+1}v_3 = {}^{n+1}v_3 + \left(\frac{0.25}{{}^{n+1}M_3}\right)\left(\Delta\mathrm{VM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad ,$$

and

$${}^{n+1}v_4 = {}^{n+1}v_4 + \left(\frac{0.25}{{}^{n+1}M_4}\right)\left(\Delta\mathrm{VM}\right)^{j+\frac{1}{2}}_{i+\frac{1}{2}} \quad .$$

H. Completion of the Cycle

In a purely Lagrangian calculation, the advective flux is bypassed entirely, and the end-of-cycle ${}^{n+1}u$ and ${}^{n+1}v$ remain to be set. In the case of an explicit Lagrangian calculation, the Phase 1 (u_L, v_L) values replace the (${}^{n}u, {}^{n}v$) values. Similarly, for an implicit Lagrangian case, the Phase 2 (u_L,v_L) values become the final values for the cycle.

i. Summary of Solution Algorithm

The above subsections complete the description of the basic finite difference approximations to the differential equations of motion. Alternative approximations may be easily substituted for any one of the these sections, because they reside in independent subroutines. To complete the solution algorithm, it is necessary to specify suitable boundary conditions. SALE contains a variety of user options for boundary conditions, which are discussed in the next section, along with other code initialization and running instructions.

III. THE SALE FORTRAN PROGRAM

A. General Structure

The SALE computer program consists of a set of subroutines controlled by a short main program. The general structure is illustrated in Fig. 4, showing a top-to-bottom flow encompassing the three phases described in Sec. II. Beside each box in the flow diagram appears the name(s) of the primary subroutine(s) responsible for the associated task. An examination of the main program listing lines SALE.170 through SALE.193 (App. A) reveals that the program closely follows the logic path of Fig. 4.

In addition to the primary subroutines, Fig. 4 also identifies a number of supporting subroutines that perform tasks for the primaries. These are concerned with boundary conditions, equation of state, and program output. Comment cards at the beginning of every subroutine in the listing describe its purpose.

The listing in App. A is for the FORTRAN Extended (FTN) compiler in use at the Los Alamos Scientific Laboratory (LASL) on the CDC-7600 computer, which operates under the Livermore Time-Sharing System (LTSS). The code is close to ANSI standard and is generally compatible with other compilers. The principal incompatibility with other systems lies in the calls that communicate with the operating system. Most of these concern the film-plotting routines. WRITE(59,-) statements refer to the user's remote terminal at LASL. The functions of all calls to local routines in SALE are described in App. B to aid users at other installations.

The input quantities to set up a problem are described in the listing (lines SALE.11 through SALE.62), using the formats appearing in subroutine RINPUT. Provision is made for tape dump and subsequent restart. In this case, a modified input file is used, where $N_x \equiv 0$ and N_y = the dump number.

B. The Indexing Notation

Figure 2 shows some variables centered at vertices and some at cell centers, typical of Lagrangian methods. In FORTRAN, one can reference x_i^j simply as X(I,J), but $p_{i+1/2}^{j+1/2}$ cannot be referenced by a "half-integer" index, so the convention is that P(I,J) refers to this pressure. Thus, the indices I and J refer to a quantity lying at the lower left vertex of a cell, or at the cell center, depending upon where the quantity is defined. In SALE, (I,J) is replaced by (IJ), as only single subscripts are used for computer efficiency. In the SALE subscript notation, the letter P stands for + and M for −. Thus, we write

IMJ for $(i-1,j)$,
IPJ for $(i+1,j)$,
IJP for $(i,j+1)$,
IPJP for $(i+1, j+1)$, ...

This notation allows programmed equations in the listing to be quickly comprehended.

As the number of vertices in either direction is one greater than the number of cells, it is apparent that the grid in computer storage must be $(N_x + 1)$ by $N_y + 1)$ in size. Because our indexing refers to cell centers and lower left vertices, we must allow one extra column of storage on the right and one extra row across the top of the mesh.

C. Boundary Conditions

Seven types of boundary conditions are provided in the SALE listing of App. A, identified on lines BC.6-BC.12. Except for special situations, as in the sample problem of Sec. IV.D, the user only needs to define the boundary types in the input data file and supply values for specified flow or applied pressure boundaries, and SALE will automatically do the rest. The input data flags indicating what conditions are to be applied along each edge of the mesh are WB, WL, WR, and WT, which correspond to the bottom, left, right, and top

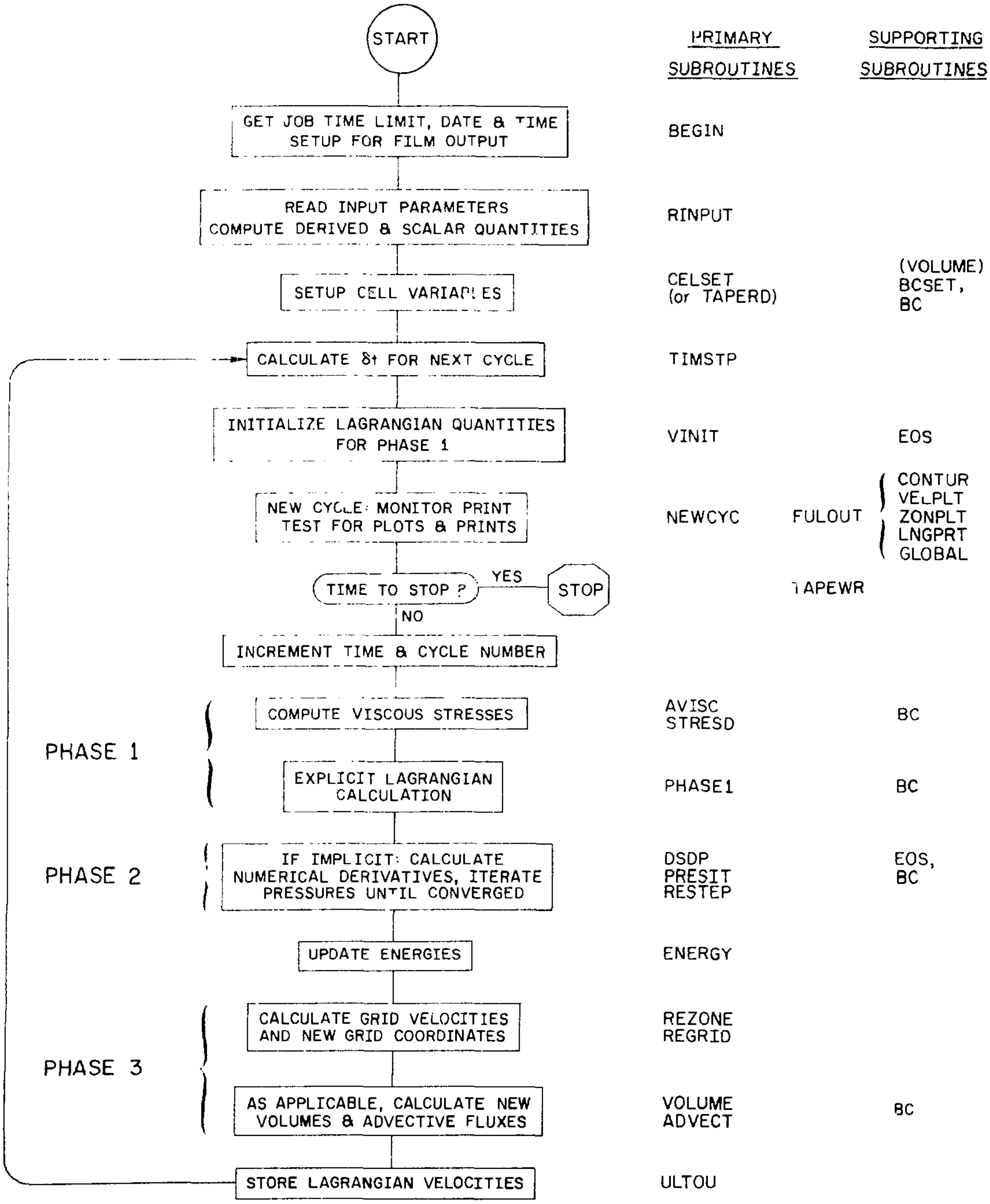

Fig. 4.
General flow diagram for the SALE program.

boundaries. Integer values assigned to these flags range from zero to six and have the following meanings.

0 = a free Lagrangian surface,
1 = a free-slip vertical or horizontal wall,
2 = a curved or tilted free-slip wall,
3 = a no-slip wall,
4 = a continuative outflow boundary,
5 = a specified inflow or outflow boundary, and
6 = a specified pressure boundary.

In general, application of a boundary condition implies some appropriate setting of the velocity components for boundary vertices. However, for a free Lagrangian surface, no adjustment is required, as the u_L and v_L from Phase 1 or 2 are the desired values, as in the sample calculation of Sec. IV.A.

For a symmetry boundary or a free-slip wall, the normal wall velocities must be kept at zero throughout the calculation. If such a boundary is parallel to the x or y axis, the u is set to zero on a left or right boundary, or the v is set to zero on a bottom or top boundary. If the wall is slanted or curved (as in the sample calculation of Sec. IV.C), both velocity components must be adjusted to make the flow tangent to the local slope. This is done by first computing an average tangent at a boundary vertex from the neighboring vertices. Then the vertex velocity is changed in such a way that its component along this tangent remains unchanged, but its normal component is zero.

Currently, the general free-slip condition calculates the tangential direction in terms of the fluid-mesh coordinates. For this reason, the boundary keeps its original shape only when the vertices lying along it are treated as Eulerian grid points. It would be straightforward to modify the code to specify a fixed boundary, not tied to the calculational mesh, in terms of which the boundary condition would be applied. This would allow a fixed, curved boundary to be used with fully Lagrangian fluid motion.

It should be noted that general free-slip curved boundaries can cause difficulties in the limit of incompressible flow. The reason is that tangential motion at the curved boundary will not precisely conserve volume. If ε, the convergence parameter, is such that it requires total volume changes less than the error committed at the curved boundary, the pressure iteration will not converge. It is necessary, then, to decrease δt, increase ε, or improve resolution along the curved boundary.

For a rigid no-slip wall, both velocity components are set to zero, regardless of wall orientation or curvature.

Inflow and outflow boundaries are more complex than those described above, as an inflow boundary requires not only specified velocity components (UIN, VIN), but also a density (ROIN) and energy (SIEIN). To have these cell quantities readily available for fluxing, SALE automatically makes the first column or row of cells at such a boundary into fictitious cells, and adjusts DO-loop limits accordingly so that the calculations bypass such cells.

The typical treatment of an outflow boundary is either to specify the outflow velocities or, if it is continuative, to set the velocity components equal to those located one vertex in. This generally works properly for high-speed flows, but for low-speed or incompressible flows, the prescription may require adjustment if it affects the upstream flow. Note also that an outflow boundary can only be used in a full-donor cell calculation ($a_0 = 1$, $b_0 = 0$), because the flux expressions will refer to outside quantities for any other a_0 and b_0. SALE provides a warning message if the input specifies other than a full-donor cell treatment in combination with an outflow boundary. This can be dealt with, as we did in the sample calculation of Sec. IV.D, by storing the necessary values in outside cells. In such instances, it is simplest to choose the right or top boundaries for the outflow, as they provide true outside cells (N_x+1 and N_y+1), and inside fictitious cells are not provided for continuative outflow boundaries. One should also be aware that a continuative outflow boundary could become an inflow boundary if the velocity field unexpectedly reverses during the calculation. Should this happen, the flux calculations will be incorrect at the boundary. Continuative boundaries, therefore, must be used with caution.

The final type of boundary condition we offer is a specified pressure boundary. It too uses inside fictitious cells for the pressure definition. Again, DO-loop limits are automatically adjusted to bypass such cells in the calculation, except in Phase 1, where these cells are included so that pressure accelerations are correctly calculated at the boundary. When a specified pressure boundary is used, the pressure must be defined in the input data list as the value of PAP.

The boundary conditions applied on the edges of the logical SALE mesh are set in the sequence of left, right, bottom, and top. A corner vertex, common to two edges, is therefore set to the condition of the edge that is treated second.

The boundary conditions of both edges meeting at a corner are considered when the general free-slip condition is applied to either edge. If both edges are general free-slip, we 'wrap around' and reference the neighboring points on each side when calculating the slope at the corner. This is especially helpful for meshes with connected curved edges, as it maintains tangential flow.

D. Numerical Stability and Accuracy

Numerical stability and accuracy are essential factors to consider when assessing the application of numerical simulation models. Time and experience have proven that first-order methods such as SALE are perfectly adequate for many problems of interest. The researcher, however, should realistically evaluate his needs when he considers the suitability of any computing technique or program. It is no wiser to pay the expense for an overly sophisticated technique than it is to struggle with trying to apply a technique to a situation for which it is not suited.

The useful accuracy of a given numerical solution may be difficult to determine analytically.[4] In such cases, it is sometimes possible to use a spectrum of computer runs with different meshes, time steps, donor-cell coefficients, artificial viscosities, convergence criteria, etc., to determine if a calculated effect is physical or simply a numerical artifice. Aside from this type of 'brute force' approach, there are several general rules one should follow in using the SALE program. For example, if a solution exhibits large variations over distances comparable to a cell width or over times comparable to the time step, it is probably not very reliable. Thus, when computing time or memory limitations preclude the use of a cell size and time step that are fine enough to resolve all spatial and temporal variations of interest in the dependent variables, then the results must be interpreted with care. In spite of such limitations, the investigator often has some choices. For example, a thin boundary layer in a large region may be resolved by employing much finer zoning at the wall and a no-slip boundary condition. If, however, its resolution is unimportant, then coarser zoning and a free-slip condition at the wall may be a valid approximation.

Numerical methods may give solutions that develop large, high-frequency oscillations in space or time. If the physical problem being modeled is known not to exhibit such behavior, the source may be numerical instability, caused by violation of one or more restrictions that should be used to limit the size of the time step.

Implicit methods have less stringent stability requirements than explicit methods, which typically require that the Courant condition on sound signal propagation,

$$\max\left(\frac{c\delta t}{\delta x}\ ,\ \frac{c\delta t}{\delta y}\right) < 1 \quad ,$$

is not violated. When SALE is run implicitly, the Courant condition can be avoided because it uses time-advanced quantities, which are determined by iteration. The resulting increase in running time per cycle can be more than compensated for by the reduction in number of time steps, but the time step must always be chosen to ensure that important short time-scale phenomena are resolved. When the Mach number of the flow exceeds the 0.1 range, an explicit solution is often more efficient, but below this range, the restriction on sound signal propagation would require a large number of time steps to move the fluid even one cell width, and the implicit solution becomes preferable.

Whether run implicitly or explicitly, the time step in SALE is always restricted by the well-verified condition that fluid cannot be moved more than approximately one cell width per time step, that is,

$$\delta t < \min\left(\frac{\delta x}{|u|}\ ,\ \frac{\delta y}{|v|}\right) \quad .$$

The minimum implies that every cell in the mesh must be considered to ensure that the δt satisfies the most restrictive case. Our general advice is to start by choosing δt equal to one-fifth of the minimum cell transit time (DTF = 0.2 in the code), being more liberal if experience allows it for the application at hand. Although Lagrangian calculations do not have as restrictive a stability limit due to advection, we still recommend that the time step satisfy this requirement, primarily for reasons of accuracy, and secondarily for efficiency by avoiding negative cell volumes at the end of Phase 1, as negative cell volumes automatically force the code to back up and restart the cycle with a smaller time step.

The donor cell or upstream component of the advection terms contributes numerical diffusion-like effects that influence the stability conditions. In particular, space-centered differencing ($a_0 = b_0 = 0$) leads to unstable results,[2] whereas full donor-cell differencing ($a_0 = 1$, $b_0 = 0$) may be too diffusive for some circumstances.

Our general recommendation is to use a_0 as small as possible without generating an instability.

Another numerical stability condition relates to the stress tensor. When viscous effects are included, the crucial condition to be satisfied is that

$$\delta t < \left[\frac{2(\lambda + 2\mu)}{\rho}\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2}\right)\right]^{-1}$$

is met in every cell, which roughly states that momentum must diffuse less than one cell width per time step. Also to be considered in Phase 1 is the related stability condition on the alternate node coupler, which can be shown to be $a_{nc} < 1$.

In SALE, we automatically choose a new time step every cycle (subroutine TIMSTP) that must satisfy four requirements: (1) advective flux, (2) viscous effects of λ and μ, (3) no more than 5% increase from the time step of the previous cycle, for reasons of accuracy, and (4) no greater than the maximum time step for the calculation (DTMAX *in the input file*). We assume that the user has satisfied upstream (a_0,b_0) and node coupling (ANC) requirements in the input data.

A final comment regarding the selection of the time step is that we specifically do *not* recommend tailoring the time step to precisely fit a specified output time. Occasionally, this may result in suddenly having, for an output cycle, a time step several orders of magnitude smaller than the problem has been running with. Experience has shown that such a discontinuous drop can adversely affect the results and cause some computing methods to blow up. It is better to let the four requirements above determine the time step, and do the output when the problem time equals or first exceeds the specified output time.

IV. SAMPLE CALCULATIONS

SALE is written in a very general fashion and is not geared to any specific type of calculation, as many codes are. It has an unusually wide range of capabilities, but as a result, most problems will require at least some code modification. Typically, this is either in the setup, boundary conditions, or the rezone. Techniques for mesh generation, boundary treatment, and rezoning are explored in the examples that follow. These include input data and code modifications, from which it is apparent that the necessary code changes for most problems are quite straightforward. In these examples, code changes are written in CDC UPDATE format. A line of the form *I, *deckname*.n means the FORTRAN statement(s) that appear between the *I line and the next line beginning with a * are *inserted* following line *deckname*.n. The line *D, *deckname*.m, *deckname*.n means to *delete* statements *deckname*.m through *deckname*.n and replace them with any lines between the *D and the next * line.

The Central Processor Unit (CPU) times and the grind times are given for each example. The grind time, defined as the CPU time per cell per cycle, $\delta CPU/(N_x*N_y)$, is a useful indicator of the computing efficiency of the code. It should be noted that CPU usage is influenced by the dynamic load on the resources in a time-sharing environment.

A. Broken Dam

This problem can be solved by the SALE program exactly as listed in App. A. Because no code modifications are required, it has been selected as the test problem in App. C, which presents several pages of numerical output for verification of results at other installations. The setup consists of a 10-cell by 10-cell uniform mesh, in plane geometry, resting on a rigid free-slip surface. The left boundary is a rigid free-slip wall, representing a plane of symmetry. At time t = 0, a dam at the right wall is removed, and the fluid is free to flow outward under the influence of gravity. The mesh is allowed to move in a purely Lagrangian fashion, and the top and right boundaries are treated as Lagrangian free surfaces. The calculation is run implicitly, in the incompressible limit (see Sec. II.E). Figure 5 shows the mesh configuration at three selected times: t = 2.0 (cycle 20), t = 5.0 (cycle 71), and at the specified finish time, t = 10.0 (cycle 207). The plots in the figure create the illusion of decreasing volume, solely because we scaled each plot to fit a specified maximum width on the film frame. The actual rightward progress of the flow can be determined from the numerical printout of cell data. The x coordinate of the lower right corner vertex (11,1), which was at position x = 10.00 at t = 0, is located at x = 14.40 at t = 2.0, at x = 28.94 at t = 5.0, and at x = 58.10 at t = 10.0. The y coordinate of the upper left corner vertex (1,11), which was at position y = 10.00 at t = 0, has dropped to y = 9.127 by t = 2.0, to y = 6.782 by t = 5.0, and down to y = 3.379 at t = 10.0.

The calculation required 42.5 s of CDC-7600 CPU time to run to completion, and created 110 frames of plots and printed information on microfiche. The input file had the following appearance.

```
T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1.
 NX        10
 NY        10
  IMP       1
  INC       1
  IREZ      1
 LPR        1
 WB         1
 WL         1
 WR         0
 WT         0
 DX         1.0
 DY         1.0
 CYL        0.0
 DT         0.10
 DTMAX      0.10
 TLIMD      0.0
 TWFILM     1.0
 TWPRTR   200.0
 TWFIN     10.0
 OM         1.00
 PEPS       1.0E-4
```

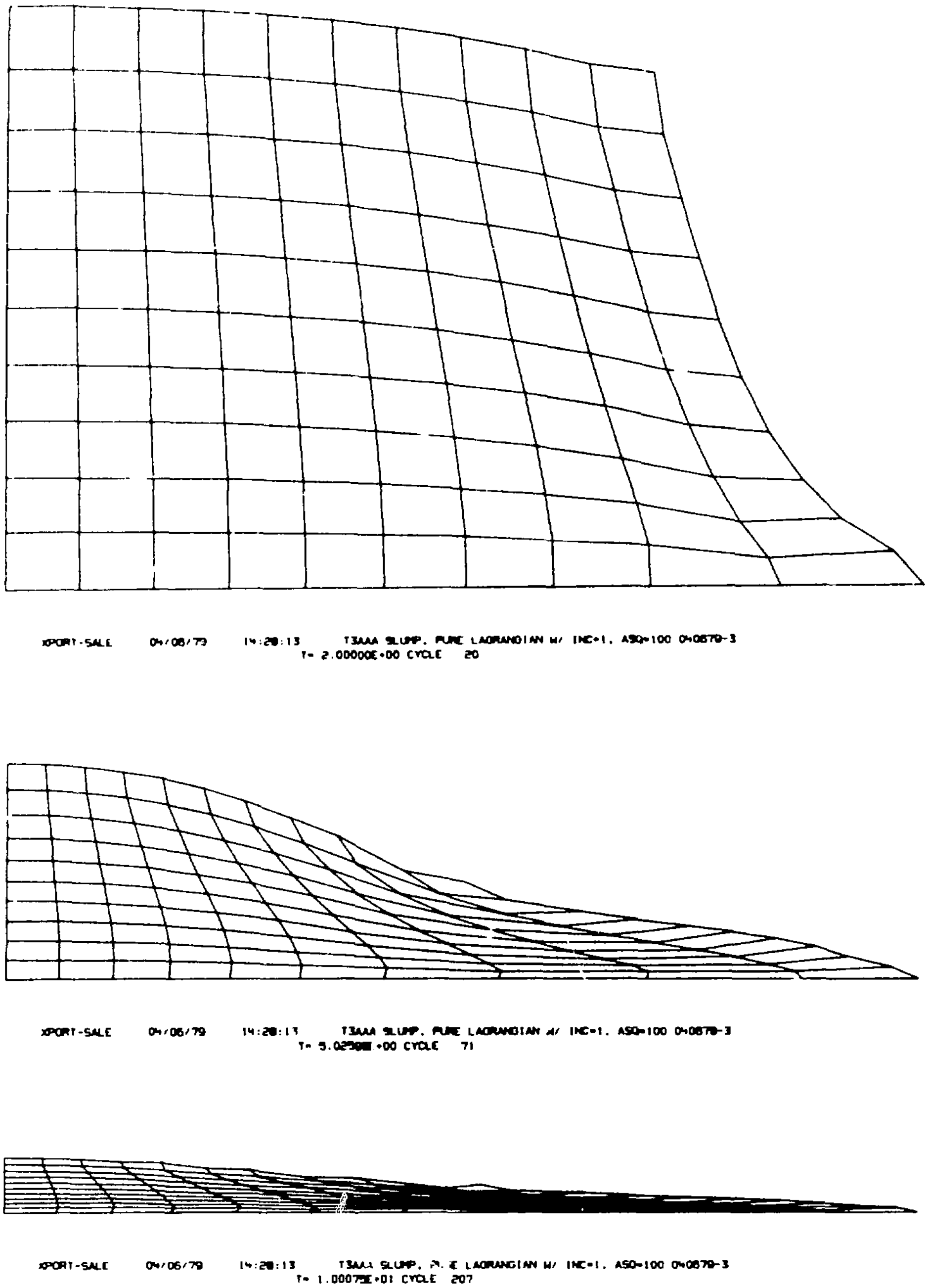

Fig. 5.

The broken dam calculation showing the configuration of the Lagrangian mesh at times $t = 2.0$, $t = 5.0$, and $t = 10.0$.

```
EPS       1.0E-4
RF        0.05
ARTVIS    0.1
LAMBDA    0.0
MU        0.0
ANC       0.05
XI        1.0
GX        0.0
GY       -1.0
A0        1.0
B0        0.0
ASQ       1.0E+2
RON       1.0
GM1       0.0
RO1       1.C
SIE1      0.0
UIN       0.0
VIN       0.0
ROIN      0.0
SIEIN     0.0
PAP       0.0
```

For further information on this calculation, refer to App. C.

B. One-Dimensional Shock Tube

The two graphs plotted in Fig. 6 illustrate density profiles from Lagrangian (upper profile) and Eulerian (lower profile) calculations of a 2:1 density-ratio shock tube. No attempt was made to obtain the best solutions that SALE can produce for this problem; rather, our intent is to illustrate that satisfactory solutions can be obtained in both limits. In Fig. 6, the SALE solutions are plotted with a heavy line and the theoretical solution[5] is plotted with a light line.

The calculations were performed in a plane mesh 60 cells long by 1 cell high, allowing 30 cells for each fluid region. The initial density was 0.2 on the left and 0.1 on the right, and the initial specific internal energy was 0.18. The gas was polytropic with $\gamma = 5/3$. The initial cell size was $\delta x = \delta y = 1/3$. Both calculations were completely inviscid ($\lambda_0 = \lambda = \mu = a_{nc} = 0$), but were run with full donor-cell differencing ($a_0 = 1$, $b_0 = 0$). The only difference between the two calculations is that the first is Lagrangian implicit (Phases 1 and 2 only), and the second is Eulerian explicit (Phases 1 and 3 only).

At $t = 0$, the diaphragm separating the two fluid regions was instantaneously removed, causing a shock to advance into the lower density region and a rarefaction to propagate back from the contact surface into the higher density region. In both calculations, δt was held constant at 0.1, and the profiles shown in Fig. 6 are at $t = 10.0$. The Lagrangian calculation required 8.6 s of CDC 7600 CPU time to run to $t = 15.0$, running at 3 iterations per cycle, with a grind time around 0.315 ms. The Eulerian calculation required 7.4 s, with a grind time around 0.210 ms. The one code modification required was to distinguish the two density regions in the setup:

```
*IDENT S020279
*I,CELSET.35
         IF(I.GT.30) RO(IJ)=0.1
```

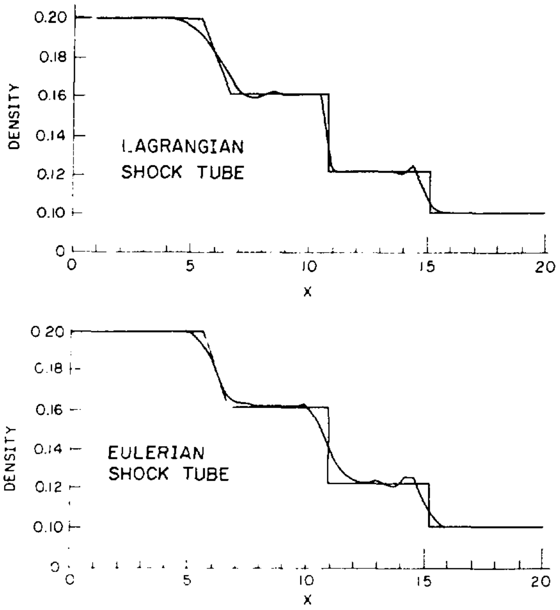

Fig. 6.
Density profiles at time t = 10.0 from an implicit Lagrangian (upper) and an explicit Eulerian (lower) calculation of a shock tube with a 2:1 density ratio.

The input file for the Lagrangian shock tube appeared as follows.

```
T3AAA LAG ST, IMP=1, LAM=0 YAQUI

     NX        60
     NY         1
     IMP        1
     INC        0
     IREZ       1
     LPR        2
     WB         1
     WL         1
     WR         1
     WT         1
     DX     .333333333
     DY     .333333333
     CYL       0.0
     DT        0.1
     DTMAX     0.1
     TLIMD     0.0
     TWFILM    1.0
     TWPRTR    1.0
     TWFIN    15.0
     OM        1.00
     PEPS      1.0E-4
     EPS       1.0E-4
     RF        0.0
     ARTVIS    0.0
     LAMBDA    0.0
     MU        0.0
     ANC       0.0
     XI        1.0
     GX        0.0
     GY        0.0
     A0        1.0
     B0        0.0
     ASQ       0.0
     RON       0.0
     GMI     .666666667
     ROI       0.2
     SIEI      0.18
     UIN       0.0
     VIN       0.0
     ROIN      0.0
     SIEIN     0.0
     PAP       0.0
```

C. Supersonic Flow Through a Curved Duct

In this example we illustrate supersonic flow through a curved two-dimensional duct. Of interest here is the procedure for creating the mesh, and the usefulness of the general free-slip boundary condition. The schematic for the mesh generation (Fig. 7) calls for a 35-cell-long by 6-cell-high Eulerian mesh, of which a 15-cell portion of the central region is to be deformed into a 90° curve, leaving a 10-cell straight section at either end. The right mesh boundary, now visually appearing at the bottom, is the inflow boundary, and the left mesh boundary allows a continuative outflow. The top and bottom boundaries employ the general free-slip condition, which is intended for arbitrarily curved or straight regions.

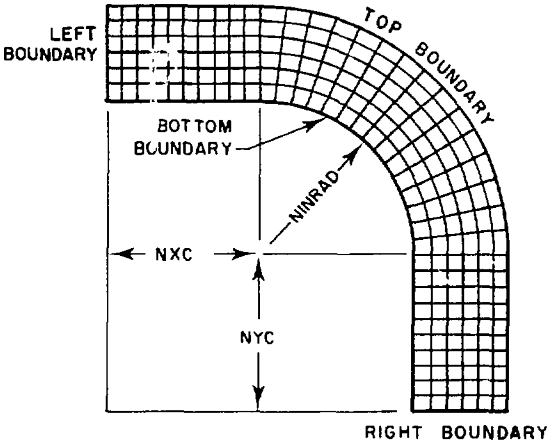

Fig. 7.
The curved-duct mesh. The three dimensions are defined in numbers of cells.

The fluid is a polytropic gas with $\gamma = 1.4$. Initial conditions state that the fluid in the mesh is at rest, with density and specific internal energy both equal to unity, and that at time $t = 0$, a shock with a Mach number $M = 10$ enters at the right boundary. The shock relations for a polytropic gas[5] allow the required conditions to be calculated as functions of M alone. In this case, the shock speed is 7.4833, while behind the shock, the fluid speed is 6.1737, the density is 5.7143, and the specific internal energy is 20.3874.

Figure 8 shows velocity vectors and isobars at three selected times in the calculation. The high and low contour values at the time of each contour plot are labeled with an H and L. At the first time, $t = 0.25$ (cycle 33), the shock is midway through the curved portion of the duct, and a high-pressure region has developed where it has encountered the wall. By time 0.50 (cycle 71), in the second set of plots, the shock front is approaching the outflow boundary and shows a strong deformation as a result of having been turned 90°. At the last time shown, $t = 3.00$ (cycle 445), well after the shock front has passed the outflow boundary, a nearly steady-state

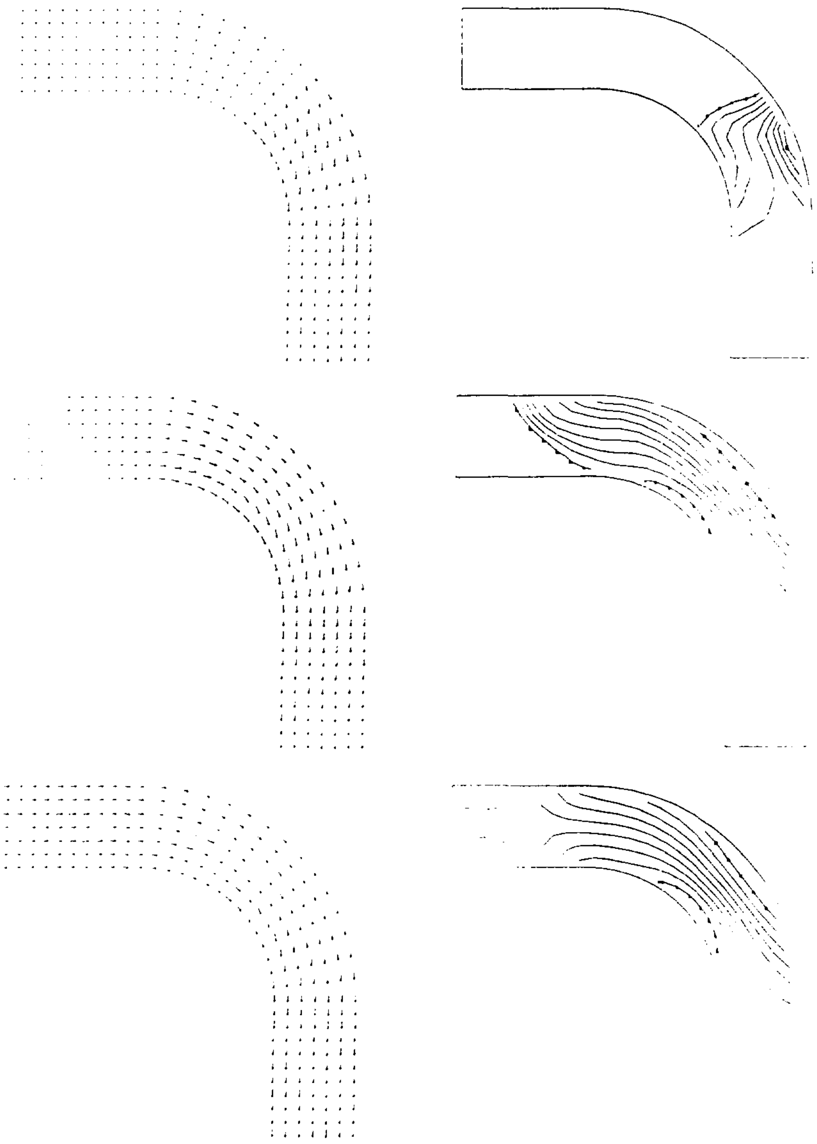

Fig. 8.
Velocity vectors (left) and isobars (right) for the supersonic duct calculation, at times t = 0.25 (top), t = 0.50 (middle), and t = 3.00 (bottom).

configuration has been established. The pressures (and similarly the densities and specific internal energies) are highest at the outer radius of the bend where the fluid reflects off the wall in the turning process.

This calculation required 28.0 s of CDC-7600 CPU time to run to time t = 3.0, with a grind time around 0.200 ms. The code modifications and input file for this problem are listed below. The first change, to DTF, allows the use of a larger time step than the standard conservative value in the code. Although this is not possible for all problems, we did realize a substantial increase in computing efficiency for the duct problem. The remaining changes, in subroutine CELSET, create the mesh using the three dimensions NXC, NYC, and NINRAD identified in Fig. 7.

By simply varying the values of the five parameters (NX, NY, NXC, NYC, and NINRAD), one can alter the configuration considerably without having to revise the setup logic. In addition, it would be possible to create a mesh as shown here, then make another pass to further modify it, for example, necking the outflow end to a nozzle. Iterative or direct schemes for grid generation[6] are quite useful for such purposes.

```
 *IDENT S041379
 *D,RINPUT.58
       DTF=0.5
 *I,CELSET.8
       NXC=NYC=10
       NINRAD=10
       RI=FLOAT(NINRAD)*DY
       CENX=FLOAT(NXC)*DX
       CENY=FLOAT(NYC)*DY
       NARC=(NXP-NYC)-(NXC+1)
       TH=90./FLOAT(NARC) * 0.0174532925
 *D,CELSET.12,CELSET.13
       IF(I.LE.NXC+1) GO TO 2
       IF(I.GE.NXP-NYC) GO TO 3
       THFAC=FLOAT(I-(NXC+1))*TH
       SINFAC=SIN(THFAC)
       COSFAC=COS(THFAC)
       RADIUS=RI+FLOAT(J-1)*DY
       X(IJ)=CENX + RADIUS*SINFAC
       Y(IJ)=CENY + RADIUS*COSFAC
       GO TO 5
     2 X(IJ)=FLOAT(I-1)*DX
       Y(IJ)=FLOAT(J-1+NYC+NINRAD)*DY
       GO TO 5
     3 X(IJ)=FLOAT(J-1+NXC+NINRAD)*DX
       Y(IJ)=FLOAT(NXP-1)*DY
     5 CONTINUE
```

```
T3AAA 35X6 SUPERSONIC DUCT 122078. IMP=0. 050779
  NX        35
  NY         6
  IMP        0
  INC        0
  IREZ       0
  LPR        1
  WB         2
  WL         4
  WR         5
  WT         2
  DX         0.1
  DY         0.1
  CYL        0.0
  DT         0.01
  DTMAX      1.0
  TLIMD      0.0
  TWFILM     0.25
  TWPRTR   100.0
  TWFIN      3.0
  OM         1.0
  PEPS       1.0E-4
  EPS        1.0E-4
  RF         0.0
  ARTVIS     0.2
  LAMBDA     0.0
  MU         0.0
  ANC        0.05
  XI         1.0
  GX         0.0
  GY         0.0
  A0         1.0
  B0         0.0
  ASQ        0.0
  RON        0.0
  GM1        0.4
  RO1        1.0
  SIE1       1.0
  UIN        0.0
  VIN        6.1737
  ROIN       5.7143
  SIEIN     20.3874
  PAP        0.0
```

D. von Karman Vortex Street

The sequence of velocity vector plots in Fig. 9 show the development of a von Karman vortex street. This fluid-flow phenomenon represents a true physical instability that is seldom seen because of fluid transparency, but is a common occurrence when air or water flows past an object. If the Reynolds number (Re) of the flow lies in a specific range,[7] the wake will depart from uniform laminar flow, and vortices will be shed alternately from each corner of the object in a regular pattern.

In this example, the initial condition was a uniform upward flow of fluid past both sides of a rigid obstacle centered across the inflow boundary of a 16 × 46 cell mesh. The problem was run implicitly in the incompressible limit (see Sec. II.E).

After the laminar flow was well established, at time t = 10 (cycle 210, Fig. 9a), we perturbed the flow by decreasing the incoming velocity on one side of the obstacle by 5%, while increasing it by 5% on the other side. This perturbation was allowed to decay immediately as a function of time, and went to zero at t = 30. Its effect is evident by t = 20 (cycle 432, Fig. 9b), and by t = 30 (cycle 654, Fig. 9c) the flow is noticeably asymmetric. Figures 9d-9f show the appearance at t = 40 (cycle 925), t = 65 (cycle 1585), and t = 70 (cycle 1709). The frequency of shedding at each corner is about 10 units of time, thus the plots at t =50 and t = 60 are similar to that at t = 70, and those at t = 45 and t = 55 to that at t = 65.

The mean flow velocity is about 0.5, the fluid density is 1.0, the effective obstacle width is 1.75, and the shear viscosity $\mu = 3.3 \times 10^{-3}$. This corresponds to Re ≈ 260, but the effective viscosity in the system is probably closer to 10^{-2}, due to truncation error effects from donor-cell differencing. Although a truncation error

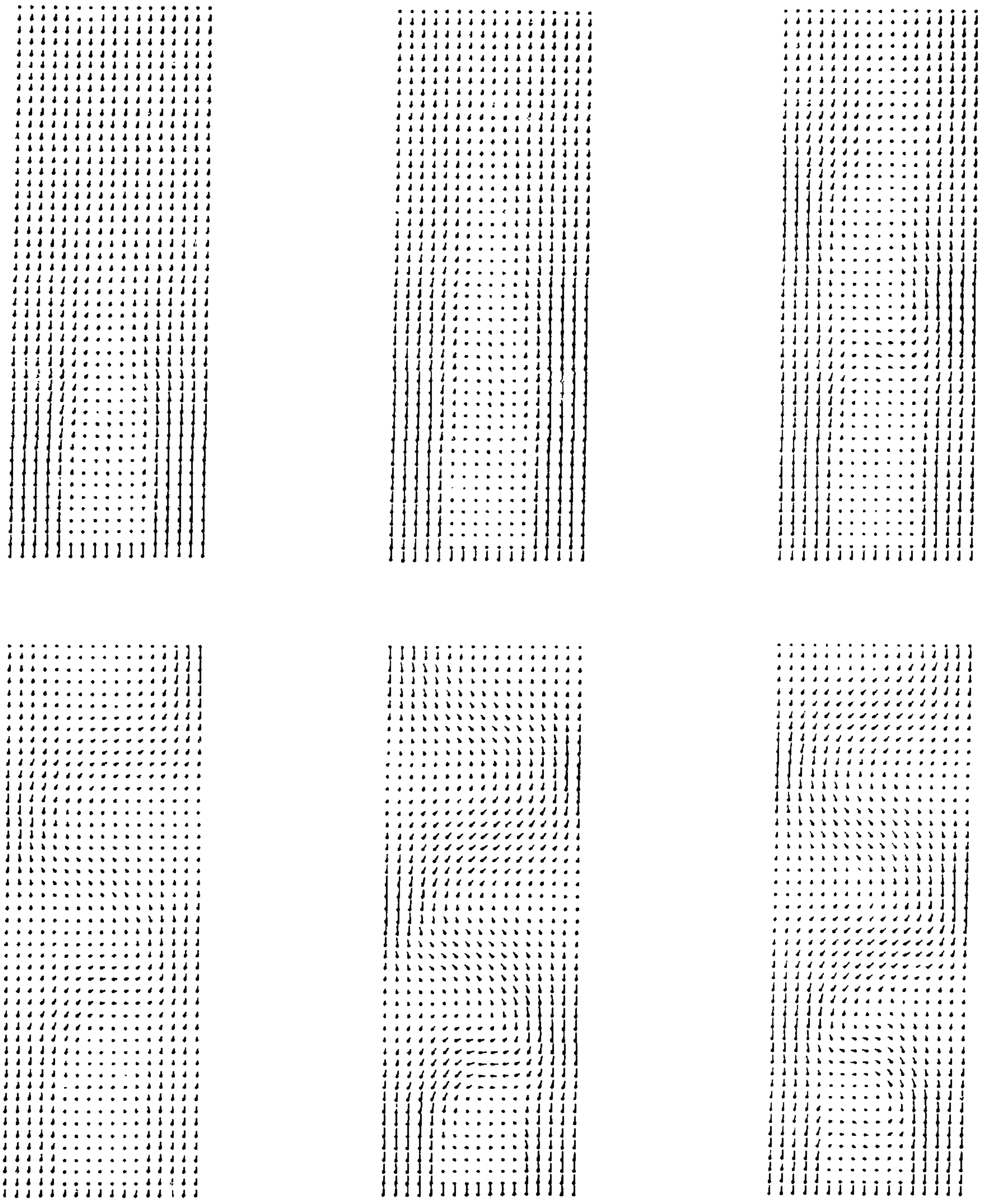

Fig. 9.

Velocity vectors for the von Karman vortex street calculation, at times $t = 10$, $t = 20$, $t = 30$, $t = 40$, $t = 65$, and $t = 70$.

analysis[4] has not been performed for SALE, it is safe to say the actual Reynolds number of the flow is closer to 100. Our results are also influenced by the narrow channel width and the placement of the obstacle at the inflow boundary, rather than up in the mesh where the flow could pass completely around it.

Of importance are the modifications required at the continuative outflow boundary. Outflow boundaries always pose a problem for low speed flows, because whatever prescription is chosen has the potential to affect the entire flow field upstream. Our problem is compounded in the vortex street calculation, as the vortices significantly change the outflow velocity profile from one side of the channel to the other. The usual treatment for a continuative outflow boundary is to set both velocity components equal to those located one cell in from the boundary, except during the iteration, when they are allowed to vary with changes in pressure, as any interior velocity component. However, for the vortex street, the outflow velocities across the top must be continually scaled to the inflow to preserve the mass balance in the system. This must be repeated every iteration, because if the velocities are allowed to float, the number of iterations will double. Note also in this example that the open length across the inlet, required for the scaling, is 9δx rather than the 8δx it would appear to be, as two half cells are effectively available for mass flux at the obstacle corners.

A small amount of fourth order node coupling was required to prevent difficulty at the trailing edges of the obstacle.[2] To minimize diffusion, we ran with only a small amount of donor cell differencing, $a_0 = 0.1$, $b_0 = 0$. For any departure from full donor-cell ($a_0 = 1$, $b_0 = 0$), the density and energy *must* be set in the cells "outside" of the continuative boundary to prevent erroneous values from being computed in the flux calculation at the boundary.

The foregoing special considerations are dealt with in the UPDATE modifications listed below.

The calculation required 16.5 min of CDC-7600 CPU time to reach the completion time of t = 70. At early times, the iteration number was typically 5 per cycle with a grind time of 0.46 ms. By t = 30, as the flow became unsteady, the iteration number was around 10 (0.66-ms grind). By t = 50, 20 iterations (1.00-ms grind) were required. Thereafter, the iteration history oscillated, ranging as high as 27 (1.30-ms grinds), down to 13 (0.77-ms grind).

```
*IDENT KARMAN
*D,BC.13
      VAVG=0.
      DO 100 I=1,NX
      IJ=NY*NXP+I
  100 VAVG=VAVG+(VV(IJ-NXP)+VV(IJ-NXP+1))*0.5*DX
      CAPK=VIN*9.*DX/VAVG
*D,BC.103
      VEPS=AMAX1((30.-T)*2.5E-3,0.0)
      IF(T.LT.10.) VEPS=0.
      VEFF=1.-VEPS
      IF(I.GT.5) VEFF=0.
      IF(I.GT.12) VEFF=1.+VEPS
      VV(IJ+NXP)=VEFF
*D,BC.127
  540 CONTINUE
*I,BC.129
      RO(IJ)=RO(IJ-NXP)
      ROL(IJ)=ROL(IJ-NXP)
      SIE(IJ)=SIE(IJ-NXP)
      VV(IJ)=CAPK*VV(IJ)

*I,CELSET.24
      V(IJ)=1.
*D,CONTUR.137
      IF(J.EQ.1 .AND. (I.GT.4.AND.I.LT.13))
     1      CALL DRV(IX4,IY4,IX1,IY1)


T3AAA KVS 16X46 CELLCEN MOMFLX, AO=.10, 041379
   NX        16
   NY        46
   IMP        1
   INC        1
   IREZ       0
   LPR        1
   WB         5
   WL         1
   WR         1
   WT         4
   DX        0.25
   DY        0.25
   CYL       0.0
   DT        0.05
   DTMAX     1.0
   TLIMD     0.0
   TWFILM    5.0
   TWPRTR  200.0
   TWFIN    70.0
   OM        1.80
   PEPS      1.0E-4
   EPS       1.0E-4
   RF        0.00
   ARTVIS    0.0
   LAMBDA    0.0
   MU      3.33333E-3
   ANC       0.05
   XI        1.0
   GX        0.0
   GY        0.0
   AO        0.1
   BO        0.0
```

```
ASQ      0.0
RON      0.0
GM1      0.4
RO1      1.0
SIE1     0.0
UIN      0.0
VIN      1.0
ROIN     1.0
SIEIN    0.0
PAP      0.0
```

E. Strong Shock Passing Over a Mercury Drop in Water

This final example concerns the calculation of a strong isothermal shock passing over a drop of mercury in water. It involves a more complex continuous rezone that illustrates what is sometimes required in order to obtain a solution, but it also illustrates the possibilities that can be realized in SALE with some imagination.

Figure 10 shows the initial configuration of the mesh. The drop has a 1-cm radius and all units are in the CGS system. The incident shock, input at the lower mesh boundary, has speed 3.62×10^5, well into the supersonic regime, as the sound speed of water is 1.5×10^5. The solution of the isothermal shock relations[5] gives the inflow conditions behind the shock—a water velocity of 3×10^5 and density of 5.83. The mercury has density 13.5 and a slightly lower sound speed, 1.45×10^5.

Mesh configuration, velocity vectors, and pressure contours at selected times are shown in Fig. 11. In Fig. 11a, at $t = 5 \times 10^{-6}$ (cycle 100), the shock has encountered the leading edge of the drop and a large pressure increase develops there because of the sudden increase in inertial resistance. By $t = 7.5 \times 10^{-6}$ (cycle 150, Fig. 11b), significant deformation of the drop has already occurred when the shock has reached its back side. By $t = 1 \times 10^{-5}$ (cycle 200, Fig. 11c) the shock is noticeably diffracting, and at $t = 1.25 \times 10^{-5}$ (cycle 250, Fig. 11d) it collapses on the symmetry axis behind the drop, and sends out a radial pressure wave evident in the $t = 1.5 \times 10^{-5}$ plot (cycle 300, Fig. 11e). A Rayleigh-Taylor type of instability is to be expected at the leading surface of the drop. The instability may be developing in the $t = 1.5 \times 10^{-5}$ plot, but the numerical resolution is too coarse and the time too short to show this.

The UPDATE modifications to set up and run this calculation are listed below. The modifications to subroutine CELSET principally concern the creation of the initial grid, which required 158 iterations.[6] The input file only allows for the definition of one material in our two-material problem. We defined the mercury in the input and supplied values for the water where needed (subroutines CELSET and VINIT). For efficiency, we bypassed the equation of state in the setup and the entire energy subroutine, taking advantage of the isothermal conditions.

A major portion of the UPDATE concerns the continuous rezone. The DATA statement identifies the vertices along the surface of the mercury drop, in single-subscript code notation, where 81 and 241 are the vertices on the axis, at the leading and trailing edges, respectively. Except for the boundary of the mercury drop, continuous rezoning was employed to keep mesh distortions under control. A relaxation factor (RF) of 1.0 was required to keep the vertices sufficiently separated in the shock front. In an initial attempt, a value of RF = 0.05 resulted in cells going to zero height across several rows.

Initially, we ran with the boundary of the mercury drop treated as purely Lagrangian (that is, moving with the fluid), but by a time of 5×10^{-6}, as the shock was encountering the drop, vertices (6,6), (6,7), (6,8), and (6,9) pinched together and crossed over. To counter this, we added a DO loop at the end of the rezone to keep the interface vertices better distributed. Considering three vertices at a time, we passed a circular arc through them and rezoned the center vertex so that its new position would lie on the arc. The procedure involved determining the coordinates (x_0,y_0) and radius (r_0) of the circular arc for use in a quadratic equation, whose correct root may be either the positive or the negative one, depending upon the circumstances. Choice of the wrong root would place the new position of the vertex on the opposite side of the circle, so care must be taken to select the root that places the new position closer to the original position.

The additional loop completely eliminated all problems of interface vertex spacing. Eventually, however, the shock deformed the drop so severely that at time 1.71×10^{-5} (cycle 383), cell inversions took place in the folded region around vertex (16,6) and we ended the run. To proceed further would require a slide-line treatment and/or moving cells from one side of the interface to the other, but a project of this magnitude was beyond the scope of this study. The CPU time required for the run was 41.5 s on the CDC-7600, with a grind time of 0.174 ms.

It can be noted in Fig. 11 that our rezone allowed the vertices along the inflow boundary (the $j = 2$ line) to

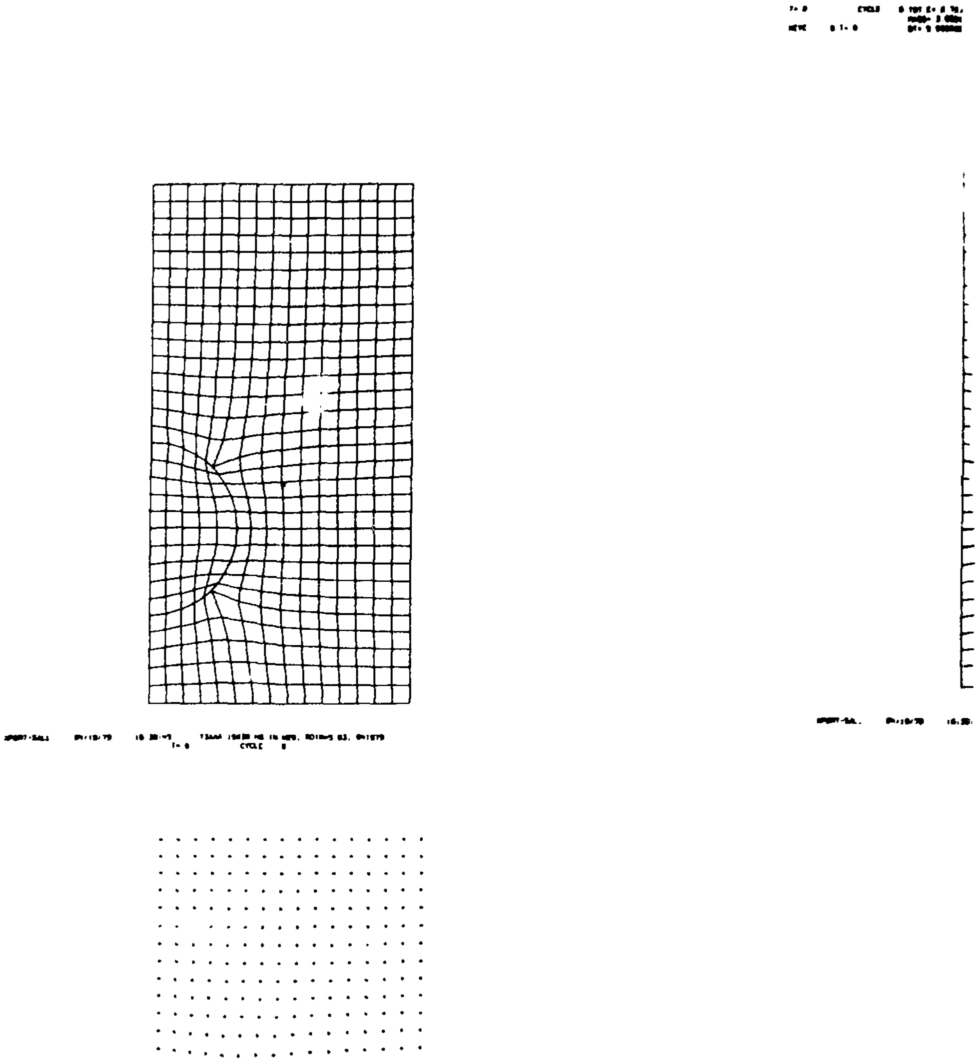

Fig. 10.
Initial mesh configuration for the mercury drop calculation.

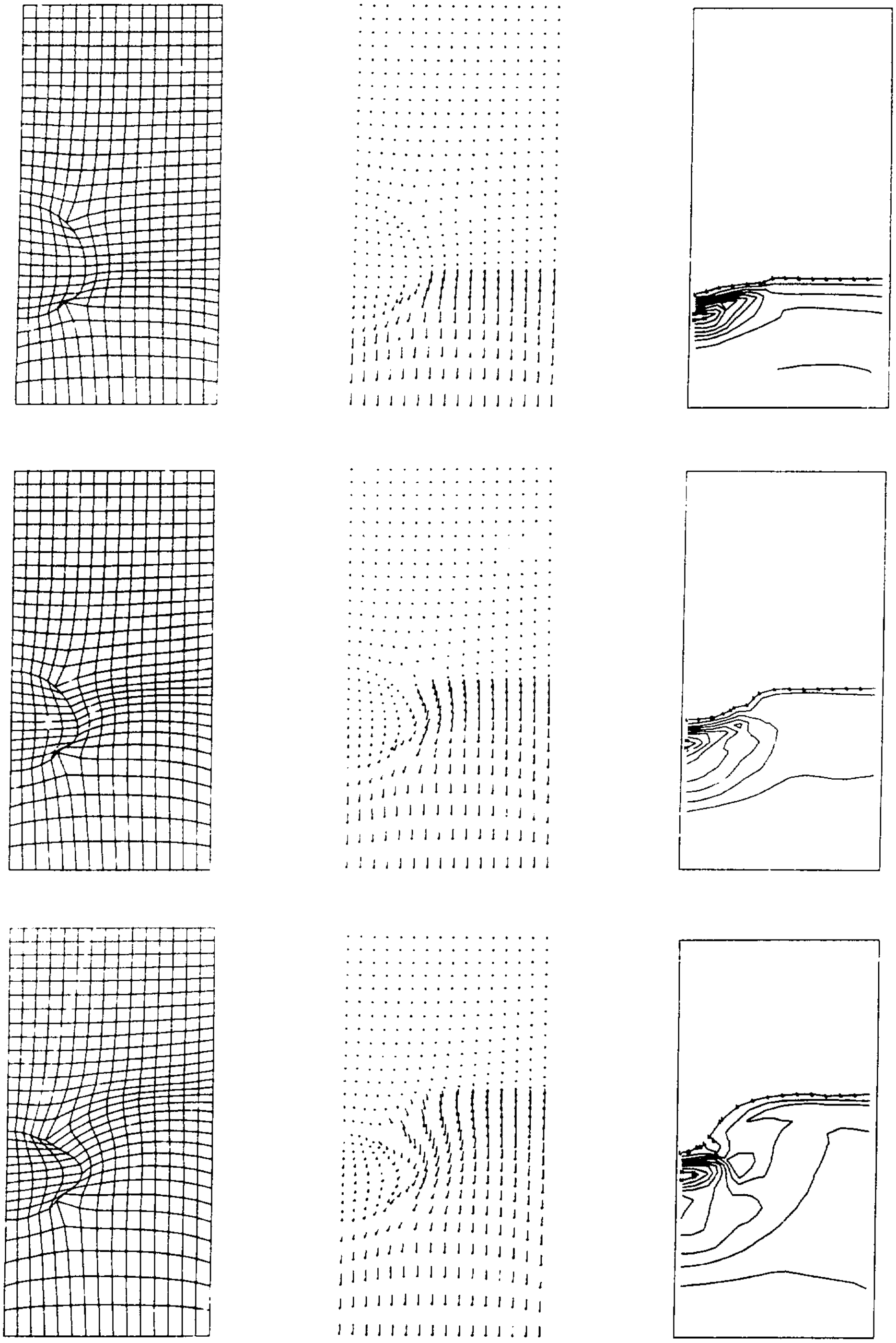

Fig. 11.

SALE calculation of a strong shock passing over a mercury drop in water. From left to right: the computing mesh, velocity vectors, and isobars. From top to bottom, the sequences are at times 5.0 × 10^{-6}, 7.5 × 10^{-6}, and 1.0 × 10^{-5}.

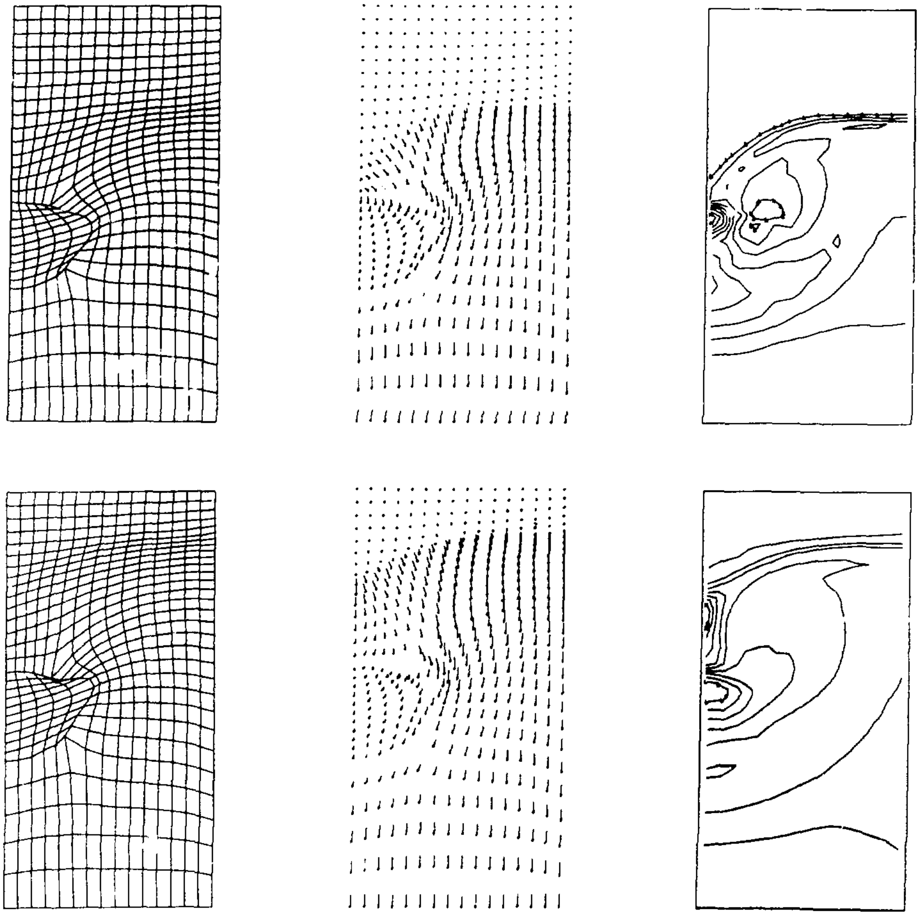

Fig. 11. (cont)
The times are 1.25 × 10^{-5} and 1.50 × 10^{-5}.

move. As a result, the inflow conditions were not perfectly maintained. This particular calculation was not significantly affected, but as a general rule the vertices on a specified inflow boundary should not be allowed to move. A slight modification to the rezone logic to ensure zero grid velocities at these vertices can easily be made.

```
*IDENT S040679
*D,CELSET.9,CELSET.15
      ITER=0
      X0=0.0
      Y0=2.0
      RAD=1.0
      RDX=1./DX
      RDY=1./DY
    5 IND=0
      DO 19 J=1,NYP
      IJ=(J-1)*NXP+1
      IJP=IJ+NXP
```

```
      IJM=IJ-NXP
      DO 18 I=1,NXP
      IPJ=IJ+1
      IMJ=IJ-1
      IF(ITER.GT.0) GO TO 10
      X(IJ)=FLOAT(I-1)*DX
      Y(IJ)=FLOAT(J-1)*DY
      IND=1
      GO TO 17
   10 IF(J.EQ.1 .OR. J.EQ.NYP) GO TO 12
      IF(I.EQ.1) GO TO 13
      IF(I.EQ.NXP) GO TO 14
      IF(I.LE.6 .AND. (J.EQ.6 .OR. J.EQ.16)) GO TO 15
      IF(I.EQ.6 .AND. (J.GE.6 .AND. J.LE.16)) GO TO 15
      XN=.25*(X(IPJ)+X(IJP)+X(IMJ)+X(IJM))
      YN=.25*(Y(IPJ)+Y(IJP)+Y(IMJ)+Y(IJM))
   11 DELX=XN-X(IJ)
      DELY=YN-Y(IJ)
      X(IJ)=XN
      Y(IJ)=YN
      GO TO 16
   12 IF(I.EQ.1 .OR. I.EQ.NXP) GO TO 17
      XN=.5*(X(IMJ)+X(IPJ))
      YN=Y(IJ)
      GO TO 11
   13 IF(J.EQ.6 .OR. J.EQ.16) GO TO 17
   14 XN=X(IJ)
      YN=.5*(Y(IJP)+Y(IJM))
      GO TO 11
   15 BETA=.02*((X(IJ)-X0)**2+(Y(IJ)-Y0)**2-RAD**2)
      DELX=(X0-X(IJ))*BETA
      DELY=(Y0-Y(IJ))*BETA
      X(IJ)=X(IJ)+DELX
      Y(IJ)=Y(IJ)+DELY
   16 XX=ABS(DELX)*RDX
      YY=ABS(DELY)*RDY
      IF(XX.GT.1.E-4 .OR. YY.GT.1.E-4) IND=1
   17 IJ=IJ+1
      IJP=IJP+1
   18 IJM=IJM+1
   19 CONTINUE
      ITER=ITER+1
      IF(IND.GT.0) GO TO 5
      WRITE(59,20) ITER
   20 FORMAT(I10)
*I,CELSET.35
      IF(I.GT.5 .OR. J.LE.5 .OR. J.GE.16) RO(IJ)=1.0
*D,CELSET.41
      F(IJ)=0.0
*I,REZONE.8
      DIMENSION IHG(21)
      DATA IHG /81,82,83,84,85,86,102,118,134,150,166,182,
     1          198,214,230,246,245,244,243,242,241/
*D,REZONE.45,REZONE.46
      UG(IJ)=VG(IJ)=0.
*I,REZONE.48
      IF(IJ.EQ.81 .OR. IJ.EQ.241) GO TO 71
*I,REZONE.53
   71 UG(IJ)=0.
      VG(IJ)=VL(IJ)
```

```
      GO TO 78
*I,REZONE.60
      IF(I.EQ.1 .OR. I.EQ.NXP) UG(IJ)=0.
      IF(J.EQ.1 .OR. J.EQ.NYP) VG(IJ)=0.
*I,REZONE.64
      DO 95 K=2,20
      K1=IHG(K-1)
      K2=IHG(K)
      K3=IHG(K+1)
      X1=X(K1)+UL(K1)*DT
      X2=X(K2)+UL(K2)*DT
      X3=X(K3)+UL(K3)*DT
      Y1=Y(K1)+VL(K1)*DT
      Y2=Y(K2)+VL(K2)*DT
      Y3=Y(K3)+VL(K3)*DT
      XA=.5*(X1+X2)
      XB=.5*(X2+X3)
      YA=.5*(Y1+Y2)
      YB=.5*(Y2+Y3)
      X2MX1=X2-X1
      Y2MY1=Y2-Y1
      X3MX1=X3-X1
      Y3MY1=Y3-Y1
      RM1= X2MX1 / Y2MY1
      RM2=(X3-X2)/(Y3-Y2)
      X0=(YB-YA-XA*RM1+XB*RM2)/(RM2-RM1)
      Y0=YA-RM1*(X0-XA)
      R0=SQRT((Y2-Y0)**2+(X2-X0)**2)
      AL=2.*Y3MY1
      BE=2.*X3MX1
      GA=X1**2+Y1**2-X3**2-Y3**2
      A=-GA/AL
      B=-BE/AL
      EMU=A-Y0
      ERO=X0**2-R0**2+EMU**2
      BTERM=B*EMU-X0
      ATERM=1.+B**2
      RADCL=SQRT(BTERM**2-ATERM*ERO)
      XN=XN1=(-BTERM+RADCL)/ATERM
         XN2=(-BTERM-RADCL)/ATERM
      YN=YN1=B*XN1+A
         YN2=B*XN2+A
      R1=SQRT((Y2-YN1)**2+(X2-XN1)**2)
      R2=SQRT((Y2-YN2)**2+(X2-XN2)**2)
      IF(R1.LT.R2) GO TO 94
      XN=XN2
      YN=YN2
   94 UG(K2)=UL(K2)+RFODT*(XN-X2)
   95 VG(K2)=VL(K2)+RFODT*(YN-Y2)
*D,VINIT.23
      P(IJ)=ASQ*(RO(IJ)-RON)
      IF(I.GT.5 .OR. J.LE.5 .OR. J.GE.16)
     1      P(IJ)=2.25E+10*(RO(IJ)-1.0)
*D,SALE.187
   30 IF(GM1.GT.0.) CALL ENERGY

T3AAA 15X30 HG IN H20, ROIN=5.83 VIN=3.0E5,

   NX        15
   NY        30
   IMP        0
   INC        0
   IREZ       2
   LPR        1
   WB         5
   WL         1
   WR         1
   WT         4
   DX        0.2
   DY        0.2
   CYL       1.0
   DT        5.0E-8
   DTMAX     5.0E-8
   TLIMD     0.0
   TWFILM    5.0E-6
   TWPRTR  100.0
   TWFIN     4.0E-5
   OM        0.0
   PEPS      0.0
   EPS       0.0
   RF        1.0
   ARTVIS    0.1
   LAMBDA    0.0
   MU        0.0
   ANC       0.05
   XI        1.0
   GX        0.0
   GY        0.0
   A0        0.5
   B0        0.0
   ASQ     2.1025E+10
   RON      13.5
   GM1       0.0
   RO1      13.5
   SIE1      0.0
   UIN       0.0
   VIN       3.0E+5
   ROIN      5.83
   SIEIN     0.0
   PAP       0.0
```

REFERENCES

1. F. H. Harlow and A. A. Amsden, "A Numerical Fluid Dynamics Calculation Method for All Flow Speeds," J. Comput. Phys. 8, 197-213 (1971). Reprinted in AIAA Selected Reprints, Vol. XV, "Computer Fluid Dynamics—Recent Advances," 83-91 (1973).

2. C. W. Hirt, A. A. Amsden, and J. L. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," J. Comput. Phys. 14, 227-253 (1974).

3. A. A. Amsden and C. W. Hirt, "YAQUI: An Arbitrary Lagrangian-Eulerian Computer Program for Fluid Flow at All Speeds," Los Alamos Scientific Laboratory report LA-5100 (March 1973).

4. C. W. Hirt, "Heuristic Stability Theory for Finite Difference Equations," J. Comput. Phys. 2, 339 (1968).

5. F. H. Harlow and A. A. Amsden, "Fluid Dynamics," Los Alamos Scientific Laboratory report LA-4700 (June 1971).

6. A. A. Amsden and C. W. Hirt, "A Simple Scheme for Generating General Curvilinear Grids," J. Comput. Phys. 11, 348-359 (1973).

7. H. Schlichting, *Boundary Layer Theory* (McGraw-Hill Book Co., New York, 1960), pp. 16, 18.

APPENDIX A

FORTRAN LISTING OF THE SALE PROGRAM

Note added in proof:
Line CELSET.53 should read: DO 180 J=JFIRST,JLAST
Line CELSET.54 should read: IJ=(J−1)*NXP+IFIRST
Line CELSET.56 should read: DO 170 I=IFIRST,ILAST
Line CELSET.67 should read: DO 200 J=JFIRST,JLASTV
Line CELSET.68 should read: IJ=(J−1)*NXP+IFIRST
Line CELSET.[illegible] should read: DO 190 I=IFIRST,ILASTV
Line TIMSTP.9 should read: DO 40 J=JFIRST,JLAST
Line TIMSTP.10 should read: IJ=(J−1)*NXP+IFIRST
Line TIMSTP.12 should read: DO 30 I=IFIRST,ILAST

LASL Identification No. LP-2023

```
      PROGRAM SALE (ITAPE,TAPE5=ITAPE,TAPE6,TAPE7,TAPE8,TTY,TAPE59=TTY)
C +++
C +++ SALE 2D - A SIMPLIFIED ICED ALE PROGRAM IN 2 DIMENSIONS;
C +++ A.A.AMSDEN, LASL T-3. LA-8095 REPORT VERSION, 011780/1045
C +++
C     * * * * LIST OF VARIABLES- * * * *
C
C (1) INPUT QUANTITIES-
C
C     NAME       PROBLEM IDENTIFICATION LINE
C     NX         NO. OF CELLS IN X-DIRECTION  (OR 0 IF TAPE RESTART)
C     NY         NO. OF CELLS IN Y-DIRECTION (OR DUMP NO. IF TAPE RESTART)
C     IMP        =1 FOR IMPLICIT PRESSURE CALCULATION,
C                =0 FOR PURELY EXPLICIT CALCULATION
C     INC        =1 FOR INCOMPRESSIBLE LIMIT VARIANT OF IMP=1 CALCULATION
C     IREZ       REZONE FLAG- 0=EULERIAN, 1=LAGRANGIAN, 2 OR GREATER
C                FOR SOME SPECIFIED CONTINUOUS REZONE
C     LPR        LONG PRINT CONTROL- 0=OMIT, 1=FILM, 2=FILM AND PRINTER,
C                3=PRINTER
C     WB,WL,WR,WT      INDICATORS FOR BOUNDARY CONDITION TO BE USED
C                      ALONG THE BOTTOM,LEFT,RIGHT, AND TOP EDGES
C                      OF THE MESH (0=LAGRANGIAN SURFACE, 1=SIMPLE FREESLIP
C                      2=GENERAL FREESLIP FOR CURVILINEAR BOUNDARIES, 3=
C                      NOSLIP, 4=CONTINUATIVE OUTFLOW, 5=SPECIFIED INFLOW
C                      OR OUTFLOW, 6=APPLIED PRESSURE)
C     DX         CELL SIZE IN THE X-DIRECTION IF UNIFORMLY ZONED
C     DY         CELL SIZE IN THE Y-DIRECTION IF UNIFORMLY ZONED
C     CYL        =1.0 FOR CYLINDRICAL GEOMETRY, =0.0 FOR PLANE GEOMETRY
C     DT         TIME STEP, SUBJECT TO AUTOMATIC RECALCULATION DURING RUN
C     DTMAX      MAXIMUM DT ALLOWED
C     TLIMD      =1.0 FORCES A TAPE DUMP EXIT BEFORE JOB TIME LIMIT
C     TWFILM     PROBLEM TIME INTERVAL BETWEEN FILM PLOTS
C     TWPRTR     PROBLEM TIME INTERVAL BETWEEN LONG PRINTS
C     TWFIN      PROBLEM TIME WHEN TO TERMINATE THE CALCULATION
C     OM         RELAXATION COEFFICIENT USED IN PRESSURE ITERATION
C     PEPS       PRESSURE FRACTION SCALING THE RELAXATION FACTOR RDSDP
C     EPS        ALLOWED RELATIVE ERROR IN THE PRESSURE ITERATION
C     RF         RELAXATION FACTOR FOR CONTINUOUS GRID REZONING
C     ARTVIS     ARTIFICIAL (BULK) VISCOSITY COEFFICIENT
C     LAMBDA     BULK VISCOSITY COEFFICIENT
C     MU         SHEAR VISCOSITY COEFFICIENT
C     ANC        ALTERNATE NODE-COUPLER COEFFICIENT
C     XI         =1. FOR 4TH-ORDER NODE COUPLING,
C                =0. FOR 2ND ORDER NODE-COUPLING
C     GX         BODY ACCELERATION IN X-DIRECTION, + OR -
C     GY         BODY ACCELERATION IN Y-DIRECTION, + OR -
C     A0, B0     FACTORS CONTROLLING CONVECTIVE FLUXING. LIMITING CASES-
C      0   0        CENTERED; UNSTABLE
C      1   0        FULL DONOR CELL; STABLE; DIFFUSIVE
C      0   1        INTERPOLATED DONOR CELL; LINEARLY STABLE; NON-DIFFUSIVE
C      1   1        STABLE; DIFFUSIVE
C     ASQ, RON, GM1     PARAMETERS FOR STIFFENED POLYTROPIC GAS EQUATION
C                       OF STATE- ASQ IS THE SQUARE OF THE ZERO-TEMPERA-
C                       TURE SOUND SPEED, RON IS THE FLUID NORMAL
C                       DENSITY, AND GM1 IS GAMMA-1, IN WHICH GAMMA IS THE
C                       RATIO OF SPECIFIC HEATS
C     ROI, SIEI    INITIAL FLUID DENSITY AND SPECIFIC INTERNAL ENERGY
C     UIN, VIN, ROIN, SIEIN   DESCRIPTORS FOR A SPECIFIED FLOW BOUNDARY-
C                             X-DIRECTION VELOCITY, Y-DIRECTION VELOCITY,
C                             DENSITY, AND SPECIFIC INTERNAL ENERGY
```

```
C     PAP       APPLIED PRESSURE FOR PRESSURE BOUNDARY CONDITION           SALE  62
C                                                                          SALE  63
C (2) DERIVED QUANTITIES:                                                  SALE  64
C                                                                          SALE  65
C     OMCYL     1-CYL                                                      SALE  66
C     ANCO      ANC/4                                                      SALE  67
C     XICOF     (1+XI)/2                                                   SALE  68
C     COLAMU    LAMBDA+2*MU                                                SALE  69
C     DPCOF     PEPS*ROI/(2(1/(DX*DX) + 1/(DY*DY)))                        SALE  70
C     GRFAC     1/(NX*NY)                                                  SALE  71
C     TFILM     PROBLEM TIME INTERVAL BETWEEN FILM PLOTS                   SALE  72
C     TPRTR     PROBLEM TIME INTERVAL BETWEEN LONG PRINTS                  SALE  73
C     NYP       NY+1                                                       SALE  74
C     NXP       NX+1                                                       SALE  75
C   THE FOLLOWING 12 INDICES DEFINE DO-LOOP LIMITS, TO                     SALE  76
C   PROVIDE FICTITIOUS CELLS FOR CERTAIN BOUNDARY TYPES:                   SALE  77
C     IFIRST    =1  IF WL=1,2,3 OR 4; =2   IF WL=5 OR 6                    SALE  78
C     JFIRST    =1  IF WB=1,2,3 OR 4; =2   IF WB=5 OR 6                    SALE  79
C     ILAST     =NX IF WR=1,2,3 OR 4; =NX-1 IF WR=5 OR 6                   SALE  80
C     JLAST     =NY IF WT=1,2,3 OR 4; =NY-1 IF WT=5 OR 6                   SALE  81
C     IFPHI     =1  IF WL=1,2,3,4 OR 6; =2   IF WL=5                       SALE  82
C     JFPHI     =1  IF WB=1,2,3,4 OR 6; =2   IF WB=5                       SALE  83
C     ILPHI     =NX IF WR=1,2,3,4 OR 6; =NX-1 IF WR=5                      SALE  84
C     JLPHI     =NY IF WT=1,2,3,4 OR 6; =NY-1 IF WT=5                      SALE  85
C     ILASTV    ILAST+1                                                    SALE  86
C     JLASTV    JLAST+1                                                    SALE  87
C     ILASTM    ILAST-1                                                    SALE  88
C     JLASTM    JLAST-1                                                    SALE  89
C                                                                          SALE  90
C (3) SCALAR QUANTITIES:                                                   SALE  91
C                                                                          SALE  92
C     T         PROBLEM TIME                                               SALE  93
C     NCYC      CYCLE COUNTER                                              SALE  94
C     IPRES     =1 DURING ITERATION TO LET CONTINUATIVE BDRY. UL,VL FLOAT  SALE  95
C     MAXIT     MAXIMUM NO. OF ITERATIONS ALLOWED BEFORE DT CUT IN HALF    SALE  96
C     NUMIT     NO. OF ITERATIONS REQD. FOR PRESSURE ITERATION CONVERGENCESALE  97
C     LOOPS     NO. OF DT CUTS FOR NON-CONVERGENCE PERFORMED               SALE  98
C               IN A GIVEN CYCLE                                           SALE  99
C     LOOPMX    LIMITS NO. OF TIMES BEFORE ERROR STOP THAT DT CAN BE       SALE 100
C               CUT DUE TO NON-CONVERGENCE IN A GIVEN CYCLE                SALE 101
C     THIRD     1/3                                                        SALE 102
C     TWLFTH    1/12                                                       SALE 103
C     PMAX      MAXIMUM PRESSURE IN THE SYSTEM                             SALE 104
C     GRIND     CENTRAL PROCESSOR TIME PER CELL PER CYCLE, IN MSEC         SALE 105
C     NDUMP     TAPE DUMP COUNTER                                          SALE 106
C     DROU      SCALING FACTOR FOR THE VELOCITY VECTOR PLOT                SALE 107
C     VMAX      MAXIMUM VELOCITY IN THE SYSTEM                             SALE 108
C     DTF       TIME-STEP FACTOR FOR CONVECTIVE STABILITY AND ACCURACY     SALE 109
C     C1        WALL CLOCK HRS/MIN/SEC WHEN THE JOB BEGAN                  SALE 110
C     D1        MONTH/DAY/YEAR WHEN THE JOB BEGAN                          SALE 111
C     XL        X-COORDINATE OF THE LEFTMOST VERTEX                        SALE 112
C     YB        Y-COORDINATE OF THE BOTTOMMOST VERTEX                      SALE 113
C     FIXL      FILM-FRAME COORDINATE ANALOG OF XL                         SALE 114
C     FIYB      FILM-FRAME COORDINATE ANALOG OF YB                         SALE 115
C     XCONV     FILM-FRAME CONVERSION COEFFICIENT IN X-DIRECTION           SALE 116
C     YCONV     FILM-FRAME CONVERSION COEFFICIENT IN Y-DIRECTION           SALE 117
C     TIMLMT    JOB TIME LIMIT, IN SECONDS                                 SALE 118
C     DTMIN     MINIMUM DT ALLOWED, = (DT OF CYCLE 1)*E-10                 SALE 119
C     IDDT      SYMBOL IDENTIFYING THE RESTRICTION CONTROLLING THE         SALE 120
C               CURRENT TIME STEP, WHERE: G=105% OF PREVIOUS DT,           SALE 121
```

```
C             C=CONVECTION (DTF), V=SHEAR VISCOSITY, OR                 SALE 122
C             M=MAXIMUM DT (DTMAX)                                      SALE 123
C    JNM      OPTIONAL, IDENTIFIER FOR CODE    SION, USER, ETC.         SALE 124
C    AA       DUMMY WORD IDENTIFYING BEGINNI       TAPE-DUMP DATA       SALE 125
C    ZZ       DUMMY WORD IDENTIFYING END OF       DUMP DATA             SALE 126
C                                                                       SALE 127
C (4) VECTOR QUANTITIES:                                                SALE 128
C                                                                       SALE 129
C    X        CARTESIAN COORDINATE IN RADIAL DIRECTION                  SALE 130
C    R        RADIAL COORDINATE (R=1 FOR CARTESIAN GEOMETRY,            SALE 131
C                                R=X FOR CYLINDRICAL GEOMETRY)          SALE 132
C    Y        CARTESIAN COORDINATE IN AXIAL DIRECTION                   SALE 133
C    U        X-DIRECTION VERTEX VELOCITY COMPONENT                     SALE 134
C    V        Y-DIRECTION VERTEX VELOCITY COMPONENT                     SALE 135
C    MC       CELL MASS                                                 SALE 136
C    MV       VERTEX MASS                                               SALE 137
C    RMV      RECIPROCAL OF VERTEX MASS MV                              SALE 138
C    RO       CELL DENSITY                                              SALE 139
C    VOL      CELL VOLUME                                               SALE 140
C    P        CELL PRESSURE                                             SALE 141
C    SIE      CELL SPECIFIC INTERNAL ENERGY                             SALE 142
C    UL       LAGRANGIAN U, CALCULATED IN PHASES 1 AND 2                SALE 143
C    VL       LAGRANGIAN V, CALCULATED IN PHASES 1 AND 2                SALE 144
C    ROL      LAGRANGIAN DENSITY, UPDATED IN PHASE 2                    SALE 145
C    PL       LAGRANGIAN PRESSURE, UPDATED IN PHASE 2                   SALE 146
C    D        VELOCITY DIVERGENCE                                       SALE 147
C    Q        CELL ARTIFICIAL VISCOUS PRESSURE                          SALE 148
C    RRSUM    1/(R1+R2+R3+R4) OF THE CELL                               SALE 149
C    PIXX     STRESS DEVIATOR TERM                                      SALE 150
C    PIXY     STRESS DEVIATOR TERM                                      SALE 151
C    PIYY     STRESS DEVIATOR TERM                                      SALE 152
C    PITH     STRESS DEVIATOR TERM                                      SALE 153
C    RDSDP    RECIPROCAL OF NUMERICAL DERIVATIVE FOR PHASE 2 ITERATION  SALE 154
C    UG       GRID VELOCITY IN THE X-DIRECTION                          SALE 155
C    VG       GRID VELOCITY IN THE Y-DIRECTION                          SALE 156
C    UREL     RELATIVE VELOCITY BETWEEN GRID AND FLUID, =UG-UL          SALE 157
C    VREL     RELATIVE VELOCITY BETWEEN GRID AND FLUID, =VG-VL          SALE 158
C    MP       INTERMEDIATE STORAGE FOR NEW MC IN PHASE 3                SALE 159
C    MVP      INTERMEDIATE STORAGE FOR NEW MV IN PHASE 3                SALE 160
C    SIEP     INTERMEDIATE STORAGE FOR SIE IN PHASE 3                   SALE 161
C    UMOM     X-DIRECTION COMPONENT OF CELL MOMENTUM                    SALE 162
C    VMOM     Y-DIRECTION COMPONENT OF CELL MOMENTUM                    SALE 163
C    UMOMP    INTERMEDIATE STORAGE FOR NEW UMOM IN PHASE 3              SALE 164
C    VMOMP    INTERMEDIATE STORAGE FOR NEW VMOM IN PHASE 3              SALE 165
C                                                                       SALE 166
C                                                                       SALE 167
C                                                                       SALE 168
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
```

```
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ            COMD  15
       REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
       INTEGER WB,WL,WR,WT                                               COMD  17
       CALL BEGIN                                                        SALE 170
       CALL RINPUT                                                       SALE 171
       IF( .(.GT.0) CALL CELSET                                          SALE 172
       IF(NX.EQ.0) CALL TAPERD                                           SALE 173
    10 CALL TIMSTP                                                       SALE 174
    20 CALL VINIT                                                        SALE 175
       CALL NEWCYC                                                       SALE 176
       IF(ARTVIS+ANC.GT.0.) CALL AVISC                                   SALE 177
       IF(LAMBDA+MU.GT.0.) CALL STRESD                                   SALE 178
       CALL PHASE1                                                       SALE 179
       IF(IMP.EQ.0) GO TO 30                                             SALE 180
       CALL DSDP                                                         SALE 181
       CALL PRESIT                                                       SALE 182
       IF(NUMIT.EQ.MAXIT .AND. INC.EQ.1) CALL RESTEP                     SALE 183
       IF(NUMIT.EQ.MAXIT .AND. INC.EQ.1) GO TO 30                        SALE 184
       IF(NUMIT.GE.MAXIT) CALL RESTEP                                    SALE 185
       IF(NUMIT.GE.MAXIT) GO TO 20                                       SALE 186
    30 CALL ENERGY                                                       SALE 187
       CALL REZONE                                                       SALE 188
       CALL REGRID                                                       SALE 189
       IF(IREZ.GT.0) CALL VOLUME                                         SALE 190
       IF(IREZ.NE.1) CALL ADVECT                                         SALE 191
       IF(IREZ.EQ.1) CALL ULTOU                                          SALE 192
       GO TO 10                                                          SALE 193
       END                                                               SALE 194
```

```
      SUBROUTINE ADVECT                                                 ADVECT 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
      DIMENSION AT(100),FT(100)                                         ADVECT 4
C +++                                                                   ADVECT 5
C +++ PHASE 3.  FLUXING OF CELL CENTERED QUANTITIES - MASS, ENERGY,     ADVECT 6
C +++ AND MOMENTUM, THEN CONVERT MOMENTA TO VELOCITIES.                 ADVECT 7
C +++ CALLED ONLY IF EULERIAN (IREZ=0) OR OTHER REZONE (IREZ.GE.2)      ADVECT 8
C +++                                                                   ADVECT 9
C +++ NOTE DO-LOOP LIMITS - INFLOW BOUNDARIES REQUIRE A MOMENTUM THAT   ADVECT10
C +++ CAN BE FLUXED IN    . .                                           ADVECT11
C +++                                                                   ADVECT12
      DO 20 J=1,NY                                                      ADVECT13
      IJ=(J-1)*NXP+1                                                    ADVECT14
      IJP=IJ+NXP                                                        ADVECT15
      DO 10 I=1,NX                                                      ADVECT16
      IPJ=IJ+1                                                          ADVECT17
      IPJP=IJP+1                                                        ADVECT18
      UMOM(IJ)=.25*ROL(IJ)*(UL(IPJ)+UL(IPJP)+UL(IJP)+UL(IJ))            ADVECT19
      VMOM(IJ)=.25*ROL(IJ)*(VL(IPJ)+VL(IPJP)+VL(IJP)+VL(IJ))            ADVECT20
      IJ=IPJ                                                            ADVECT21
   10 IJP=IPJP                                                          ADVECT22
   20 CONTINUE                                                          ADVECT23
      DO 60 J=JFIRST,JLAST                                              ADVECT24
      IJ=(J-1)*NXP+IFIRST                                               ADVECT25
      IJP=IJ+NXP                                                        ADVECT26
      IJM=IJ-NXP                                                        ADVECT27
      DO 50 I=IFIRST,ILAST                                              ADVECT28
      IMJ=IJ-1                                                          ADVECT29
      IPJ=IJ+1                                                          ADVECT30
      IPJP=IJP+1                                                        ADVECT31
      X1=X(IPJ)                                                         ADVECT32
      Y1=Y(IPJ)                                                         ADVECT33
      R1=R(IPJ)                                                         ADVECT34
      X2=X(IPJP)                                                        ADVECT35
      Y2=Y(IPJP)                                                        ADVECT36
      R2=R(IPJP)                                                        ADVECT37
      X3=X(IJP)                                                         ADVECT38
      Y3=Y(IJP)                                                         ADVECT39
      R3=R(IJP)                                                         ADVECT40
      X4=X(IJ)                                                          ADVECT41
      Y4=Y(IJ)                                                          ADVECT42
      R4=R(IJ)                                                          ADVECT43
      XP1=X1-UREL(IPJ)*DT                                               ADVECT44
      XP2=X2-UREL(IPJP)*DT                                              ADVECT45
      XP3=X3-UREL(IJP)*DT                                               ADVECT46
      XP4=X4-UREL(IJ)*DT                                                ADVECT47
      YP1=Y1-VREL(IPJ)*DT                                               ADVECT48
      YP2=Y2-VREL(IPJP)*DT                                              ADVECT49
```

```
      YP3=Y3-VREL(IJP)*DT
      YP4=Y4-VREL(IJ)*DT
      RP1=XP1*CYL+OMCYL
      RP2=XP2*CYL+OMCYL
      RP3=XP3*CYL+OMCYL
      RP4=XP4*CYL+OMCYL
      VOLL=VOLB=VOLR=VOLT=VOLC=VOL(IJ)
      IF(I.NE.1) VOLL=VOL(IMJ)
      IF(J.NE.1) VOLB=VOL(IJM)
      IF(I.NE.NX) VOLR=VOL(IPJ)
      IF(J.NE.NY) VOLT=VOL(IJP)
      FL=-FR
      AL=-AR
      IF(I.EQ.1) FL=AL=0.
      IF(WL.LT.4) GO TO 30
      IF(I.GT.IFIRST) GO TO 30
      FL=((RP3+R4+RP4)*(XP3*(Y4-YP4)+X4*(YP4-YP3)+XP4*(YP3-Y4))
     1 +(R3+RP3+R4)*(X3*(Y4-YP3)+XP3*(Y3-Y4)+X4*(YP3-Y3)))*TWLFTH
      AL=A0*SIGN(1.,FL)+B0*4.*FL/(VOLL+VOLC)
   30 FB=-FT(I)
      AB=-AT(I)
      IF(J.EQ.1) FB=AB=0.
      IF(WB.LT.4) GO TO 40
      IF(J.GT.JFIRST) GO TO 40
      FB=((R1+RP1+RP4)*(X1*(YP1-YP4)+XP1*(YP4-Y1)+XP4*(Y1-YP1))
     1 +(R1+R4+RP4)*(X1*(YP4-Y4)+X4*(Y1-YP4)+XP4*(Y4-Y1)))*TWLFTH
      AB=A0*SIGN(1.,FB)+B0*4.*FB/(VOLB+VOLC)
   40 FR=((R1+RP1+RP2)*(X1*(YP2-YP1)+XP1*(Y1-YP2)+XP2*(YP1-Y1))
     1 +(R1+R2+RP2)*(X1*(Y2-YP2)+X2*(YP2-Y1)+XP2*(Y1-Y2)))*TWLFTH
      AR=A0*SIGN(1.,FR)+B0*4.*FR/(VOLR+VOLC)
      FT(I)=((RP2+R3+RP3)*(XP2*(Y3-YP3)+X3*(YP3-YP2)+XP3*(YP2-Y3))
     1 +(R2+RP2+R3)*(X2*(Y3-YP2)+XP2*(Y2-Y3)+X3*(YP2-Y2)))*TWLFTH
      AT(I)=A0*SIGN(1.,FT(I))+B0*4.*FT(I)/(VOLT+VOLC)
      S1=FR*(1.-AR)
      S2=FR*(1.+AR)
      S3=FT(I)*(1.-AT(I))
      S4=FT(I)*(1.+AT(I))
      S5=FL*(1.-AL)
      S6=FL*(1.+AL)
      S7=FB*(1.-AB)
      S8=FB*(1.+AB)
      S9=S1+S3+S5+S7
      MP(IJ)=MC(IJ)+S9*ROL(IJ)  +S2*ROL(IPJ)+S4*ROL(IJP)
     1              +S6*ROL(IMJ)+S8*ROL(IJM)
      SIEP(IJ)=(MC(IJ)*SIE(IJ)+S9*ROL(IJ)  *SIE(IJ)  +S2*ROL(IPJ)*SIE(IPJ)
     1                          +S4*ROL(IJP)*SIE(IJP)+S6*ROL(IMJ)*SIE(IMJ)
     2                          +S8*ROL(IJM)*SIE(IJM)) / MP(IJ)
      UMOMP(IJ)=S9*UMOM(IJ)  +S2*UMOM(IPJ)  +S4*UMOM(IJP)
     1          +S6*UMOM(IMJ)+S8*UMOM(IJM)
      VMOMP(IJ)=S9*VMOM(IJ)  +S2*VMOM(IPJ)  +S4*VMOM(IJP)
     1          +S6*VMOM(IMJ)+S8*VMOM(IJM)
      IJ=IPJ
      IJP=IPJP
   50 IJM=IJM+1
   60 CONTINUE
C +++
C +++ COMPUTE NEW VERTEX MASSES
C +++
      DO 80 J=JFIRST,JLAST
      IJ=(J-1)*NXP+IFIRST
      IJP=IJ+NXP
      DO 70 I=IFIRST,ILAST
      IPJ=IJ+1
```

```
      IPJP=IJP+1
      MC(IJ)=MP(IJ)
      SIE(IJ)=SIEP(IJ)
      QM=0.25*MC(IJ)
      MVP(IPJ) =MVP(IPJ) +QM
      MVP(IPJP)=MVP(IPJP)+QM
      MVP(IJP) =MVP(IJP) +QM
      MVP(IJ)  =MVP(IJ)  +QM
      IJ=IPJ
   70 IJP=IPJP
   80 CONTINUE
C +++
C +++ STORE VERTEX MASSES AND CALCULATE NEW VELOCITIES . . .
C +++
      DO 100 J=JFIRST,JLASTV
      IJ=(J-1)*NXP+IFIRST
      DO 90 I-IFIRST,ILASTV
      RMV(IJ)=1./MVP(IJ)
      U(IJ)=UL(IJ)*MV(IJ)*RMV(IJ)
      V(IJ)=VL(IJ)*MV(IJ)*RMV(IJ)
      MV(IJ)=MVP(IJ)
   90 IJ=IJ+1
  100 CONTINUE
      DO 120 J=JFIRST,JLAST
      IJ=(J-1)*NXP+IFIRST
      IJP=IJ+NXP
      DO 110 I=IFIRST,ILAST
      IPJ=IJ+1
      IPJP=IJP+1
      QMOMU=.25*UMOMP(IJ)
      U(IPJ) =U(IPJ) +QMOMU*RMV(IPJ)
      U(IPJP)=U(IPJP)+QMOMU*RMV(IPJP)
      U(IJP) =U(IJP) +QMOMU*RMV(IJP)
      U(IJ)  =U(IJ)  +QMOMU*RMV(IJ)
      QMOMV=.25*VMOMP(IJ)
      V(IPJ) =V(IPJ) +QMOMV*RMV(IPJ)
      V(IPJP)=V(IPJP)+QMOMV*RMV(IPJP)
      V(IJP) =V(IJP) +QMOMV*RMV(IJP)
      V(IJ)  =V(IJ)  +QMOMV*RMV(IJ)
      IJ=IPJ
  110 IJP=IPJP
  120 CONTINUE
      CALL BC(U,V)
      RETURN
      END
```

```
      SUBROUTINE AVISC                                                  AVISC   2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),     COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                               COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,        COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPH1,ILAST,ILASTM,ILASTV,ILPH1,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPH1,JLAST,JLASTM,JLASTV,JLPH1,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
      IF(ARTVIS.EQ.0.) GO TO 50                                         AVISC   4
C +++                                                                   AVISC   5
C +++ CALCULATE ARTIFICIAL (BULK) VISCOUS CONTRIBUTIONS TO UL AND VL    AVISC   6
C +++                                                                   AVISC   7
      DO 20 J=JFIRST,JLAST                                              AVISC   8
      IJ=(J-1)*NXP+IFIRST                                               AVISC   9
      IJP=IJ+NXP                                                        AVISC  10
      DO 10 I=IFIRST,ILAST                                              AVISC  11
      IPJ=IJ+1                                                          AVISC  12
      IPJP=IJP+1                                                        AVISC  13
      X1=X(IPJ)                                                         AVISC  14
      Y1=Y(IPJ)                                                         AVISC  15
      U1=UL(IPJ)                                                        AVISC  16
      V1=VL(IPJ)                                                        AVISC  17
      X2=X(IPJP)                                                        AVISC  18
      Y2=Y(IPJP)                                                        AVISC  19
      U2=UL(IPJP)                                                       AVISC  20
      V2=VL(IPJP)                                                       AVISC  21
      X3=X(IJP)                                                         AVISC  22
      Y3=Y(IJP)                                                         AVISC  23
      U3=UL(IJP)                                                        AVISC  24
      V3=VL(IJP)                                                        AVISC  25
      X4=X(IJ)                                                          AVISC  26
      Y4=Y(IJ)                                                          AVISC  27
      U4=UL(IJ)                                                         AVISC  28
      V4=VL(IJ)                                                         AVISC  29
      AREA=0.5*((X2-X4)*(Y3-Y1)-(X1-X3)*(Y4-Y2))                        AVISC  30
      RAREA=1./AREA                                                     AVISC  31
      DUDX=0.5*RAREA*((U2-U4)*(Y3-Y1)-(U1-U3)*(Y4-Y2))                  AVISC  32
      DVDY=0.5*RAREA*((V4-V2)*(X3-X1)-(V1-V3)*(X2-X4))                  AVISC  33
      RRSUM(IJ)=1./(R(IPJ)+R(IPJP)+R(IJP)+R(IJ))                        AVISC  34
      UOR=(U1+U2+U3+U4)*RRSUM(IJ)                                       AVISC  35
      D(IJ)=DUDX+DVDY+UOR*CYL                                           AVISC  36
      Q(IJ)=ARTVIS*RO(IJ)*AREA*D(IJ)                                    AVISC  37
C +++                                                                   AVISC  38
C +++ Q IS AN ARTIFICIAL VISCOUS PRESSURE FOR USE WITH SHOCKS.  THE     AVISC  39
C +++ FORM USED HERE IS QUADRATIC IN THE VELOCITY DIVERGENCE D=DEL DOT UAVISC  40
C +++ Q IS ZERO IN EXPANDING CELLS (D POSITIVE).                        AVISC  41
C +++                                                                   AVISC  42
      IJP=IPJP                                                          AVISC  43
   10 IJ=IPJ                                                            AVISC  44
   20 CONTINUE                                                          AVISC  45
      DO 40 J=JFIRST,JLAST                                              AVISC  46
      IJ=(J-1)*NXP+IFIRST                                               AVISC  47
      IJP=IJ+NXP                                                        AVISC  48
      DO 30 I=IFIRST,ILAST                                              AVISC  49
```

```
      IPJ=IJ+1
      IPJP=IJP+1
      Q(IJ)=AMIN1(0.,D(IJ))*Q(IJ)
      XX=0.5*DT*Q(IJ)
      R1=R(IPJ)
      R2=R(IPJP)
      R3=R(IJP)
      R4=R(IJ)
      RM1=RMV(IPJ)
      RM2=RMV(IPJP)
      RM3=RMV(IJP)
      RM4=RMV(IJ)
      Y24=Y(IPJP)-Y(IJ)
      Y31=Y(IJP)-Y(IPJ)
      XR24=(X(IPJP)-X(IJ) )*.5*(R2+R4)
      XR31=(X(IJP) -X(IPJ))*.5*(R3+R1)
      UL(IPJ) =UL(IPJ) +XX*RM1*Y24*R1
      UL(IPJP)=UL(IPJP)+XX*RM2*Y31*R2
      UL(IJP) =UL(IJP) -XX*RM3*Y24*R3
      UL(IJ)  =UL(IJ)  -XX*RM4*Y31*R4
      VL(IPJ) =VL(IPJ) -XX*RM1*XR24
      VL(IPJP)=VL(IPJP)-XX*RM2*XR31
      VL(IJP) =VL(IJP) +XX*RM3*XR24
      VL(IJ)  =VL(IJ)  +XX*RM4*XR31
      IJP=IPJP
   30 IJ=IPJ
   40 CONTINUE
      CALL BC(UL,VL)
   50 IF(ANC.EQ.0.) RETURN
C +++
C +++ OPTIONAL NODE COUPLER - A VELOCITY DIFFUSION TO SUPPRESS
C +++ VERTEX COASTING.  USE XI=1.0 (4TH ORDER) TO COMBAT BOWTIES.
C +++ USE XI=0.0 (2ND ORDER) TO COMBAT HERRINGBONE PATTERN . . .
C +++
      DO 70 J=JFIRST,JLAST
      IJ=(J-1)*NXP+IFIRST
      IJP=IJ+NXP
      DO 60 I=IFIRST,ILAST
      IPJ=IJ+1
      IPJP=IJP+1
      U1=U(IPJ)
      U2=U(IPJP)
      U3=U(IJP)
      U4=U(IJ)
      V1=V(IPJ)
      V2=V(IPJP)
      V3=V(IJP)
      V4=V(IJ)
      IVL=IUB=IVR=IUT=1.
      IF(I.EQ.1) IVL=2.
      IF(J.EQ.1) IUB=2.
      IF(I.EQ.NX) IVR=2.
      IF(J.EQ.NY) IUT=2.
      UL(IPJ) =UL(IPJ) +ANCO*IUB*(XICOF*(U2+U4)-XI*U3-U1)
      UL(IPJP)=UL(IPJP)+ANCO*IUT*(XICOF*(U1+U3)-XI*U4-U2)
      UL(IJP) =UL(IJP) +ANCO*IUT*(XICOF*(U2+U4)-XI*U1-U3)
      UL(IJ)  =UL(IJ)  +ANCO*IUB*(XICOF*(U1+U3)-XI*U2-U4)
      VL(IPJ) =VL(IPJ) +ANCO*IVR*(XICOF*(V2+V4)-XI*V3-V1)
      VL(IPJP)=VL(IPJP)+ANCO*IVR*(XICOF*(V1+V3)-XI*V4-V2)
      VL(IJP) =VL(IJP) +ANCO*IVL*(XICOF*(V2+V4)-XI*V1-V3)
      VL(IJ)  =VL(IJ)  +ANCO*IVL*(XICOF*(V1+V3)-XI*V2-V4)
      IJP=IPJP
   60 IJ=IPJ
```

```
   70 CONTINUE
      CALL BC(UL,VL)
      RETURN
      END
```

```
      SUBROUTINE BC(UU,VV)
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZI
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GMI,GRFAC,GRIND,
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES,
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ
      REAL LAMBDA,MC,MP,MU,MV,MVP
      INTEGER WB,WL,WR,WT
      DIMENSION UU(1),VV(1)
C +++
C +++ SET VELOCITY BOUNDARY CONDITIONS FOR ALL 4 SIDES, WHERE
C +++ WL, WR, WB, AND WT ARE INPUT INTEGERS DEFINED AS FOLLOWS:
C +++ 0 = LAGRANGIAN SURFACE (NO VELOCITY ADJUSTMENT WHATSOEVER),
C +++ 1 = SIMPLE FREESLIP, 2 = GENERAL FREESLIP, 3 = NOSLIP,
C +++ 4 = CONTINUATIVE OUTFLOW, 5 = SPECIFIED INFLOW OR OUTFLOW,
C +++ 6 = SPECIFIED PRESSURE. (NOTE - THE CONTINUATIVE OUTFLOW BOUNDARY
C +++ APPROXIMATION GIVEN HERE MAY NOT WORK FOR ALL APPLICATIONS.)
C +++
      DO 400 J=1,NYP
      IJ=(J-1)*NXP+1
C +++
C +++ THE LEFT EDGE . . .
C +++
      IF(WL.EQ.0) GO TO 300
      GO TO (210,220,230,240,250,240),WL
  210 UU(IJ)=0.
      GO TO 300
  220 IJM=IJP=IJ
      IF(J.GT.1) IJM=IJ-NXP
      IF(J.LT.NYP) IJP=IJ+NXP
      IF(J.EQ.1   .AND. WB.EQ.2) IJM=IJ+1
      IF(J.EQ.NYP .AND. WT.EQ.2) IJP=IJ+1
      EM=1.E+20
      XTE=X(IJP)-X(IJM)
      IF(XTE.NE.0.) EM=(Y(IJP)-Y(IJM))/XTE
      RDEN=1./(1.+EM*EM)
      UOLD=UU(IJ)
      UU(IJ)=(EM*VV(IJ)+UOLD)*RDEN
      VV(IJ)=(EM*EM*VV(IJ)+EM*UOLD)*RDEN
      GO TO 300
```

```
  230 UU(IJ)=VV(IJ)=0.
      GO TO 300
  240 IF(IPRES.EQ.1) GO TO 300
      UU(IJ)=UU(IJ+1)
      VV(IJ)=VV(IJ+1)
      GO TO 300
  250 UU(IJ+1)=UIN
      VV(IJ+1)=VIN
  300 IJ=IJ+NX
C +++
C +++ THE RIGHT EDGE . . .
C +++
      IF(WR.EQ.0) GO TO 400
      GO TO (310,320,330,340,350,340),WR
  310 UU(IJ)=0.
      GO TO 400
  320 IJP=IJM=IJ
      IF(J.GT.1) IJM=IJ-NXP
      IF(J.LT.NYP) IJP=IJ+NXP
      IF(J.EQ.1    .AND. WB.EQ.2) IJM=IJ-1
      IF(J.EQ.NYP  .AND. WT.EQ.2) IJP=IJ-1
      EM=1.E+20
      XTE=X(IJP)-X(IJM)
      IF(XTE.NE.0.) EM=(Y(IJP)-Y(IJM))/XTE
      RDEN=1./(1.+EM*EM)
      UOLD=UU(IJ)
      UU(IJ)=(EM*VV(IJ)+UOLD)*RDEN
      VV(IJ)=(EM*EM*VV(IJ)+EM*UOLD)*RDEN
      GO TO 400
  330 UU(IJ)=VV(IJ)=0.
      GO TO 400
  340 IF(IPRES.EQ.1) GO TO 400
      UU(IJ)=UU(IJ-1)
      VV(IJ)=VV(IJ-1)
      GO TO 400
  350 UU(IJ-1)=UIN
      VV(IJ-1)=VIN
  400 CONTINUE
      DO 600 I=1,NXP
      IJ=I
C +++
C +++ THE BOTTOM EDGE . . .
C +++
      IF(WB.EQ.0) GO TO 500
      GO TO (410,420,430,440,450,440),WB
  410 VV(IJ)=0.
      GO TO 500
  420 IMJ=IPJ=IJ
      IF(I.GT.1) IMJ=IJ-1
      IF(I.LT.NXP) IPJ=IJ+1
      IF(I.EQ.1    .AND. WL.EQ.2) IMJ=IJ+NXP
      IF(I.EQ.NXP  .AND. WR.EQ.2) IPJ=IJ+NXP
      EM=1.E+20
      XTE=X(IPJ)-X(IMJ)
      IF(XTE.NE.0.) EM=(Y(IPJ)-Y(IMJ))/XTE
      RDEN=1./(1.+EM*EM)
      UOLD=UU(IJ)
      UU(IJ)=(EM*VV(IJ)+UOLD)*RDEN
      VV(IJ)=(EM*EM*VV(IJ)+EM*UOLD)*RDEN
      GO TO 500
  430 UU(IJ)=VV(IJ)=0.
      GO TO 500
  440 IF(IPRES.EQ.1) GO TO 500
```

```
      UU(IJ)=UU(IJ+NXP)
      VV(IJ)=VV(IJ+NXP)
      GO TO 500
  450 UU(IJ+NXP)=UIN
      VV(IJ+NXP)=VIN
  500 IJ=NY*NXP+1
C +++
C +++ THE TOP EDGE . . .
C +++
      IF(WT.EQ.0) GO TO 600
      GO TO (510,520,530,540,550,540),WT
  510 VV(IJ)=0.
      GO TO 600
  520 IMJ=IPJ=IJ
      IF(I.GT.1) IMJ=IJ-1
      IF(I.LT.NXP) IPJ=IJ+1
      IF(I.EQ.1    .AND. WL.EQ.2) IMJ=IJ-NXP
      IF(I.EQ.NXP .AND. WR.EQ.2) IPJ=IJ-NXP
      EM=1.E+20
      XTE=X(IPJ)-X(IMJ)
      IF(XTE.NE.0.) EM=(Y(IPJ)-Y(IMJ))/XTE
      RDEN=1./(1.+EM*EM)
      UOLD=UU(IJ)
      UU(IJ)=(EM*VV(IJ)+UOLD)*RDEN
      VV(IJ)=(EM*EM*VV(IJ)+EM*UOLD)*RDEN
      GO TO 600
  530 UU(IJ)=VV(IJ)=0.
      GO TO 600
  540 IF(IPRES.EQ.1) GO TO 600
      UU(IJ)=UU(IJ-NXP)
      VV(IJ)=VV(IJ-NXP)
      GO TO 600
  550 UU(IJ-NXP)=UIN
      VV(IJ-NXP)=VIN
  600 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BCSET
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,
     2 GX,GY,IDD1,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES,
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,Xi,XICOF,XL,YB,YCONV,ZZ
      REAL LAMBDA,MC,MP,MU,MV,MVP
      INTEGER WB,WL,WR,WT
C +++
```

```
C +++ ADJUST U,V,RO,MC,SIE,P AS REQUIRED FOR INFLOW OR SPECIFIED
C +++ PRESSURE BOUNDARIES.  (CALLED ONCE, BY SUBROUTINE CELSET). . .
C +++
      DO 400 J=1,NYP
      IJ=(J-1)*NXP+1
C +++
C +++ THE LEFT EDGE . . .
C +++
      IF(WL.EQ.0) GO TO 300
      GO TO (300,300,300,300,250,260),WL
  250 U(IJ)=UIN
      V(IJ)=VIN
      RO(IJ)=ROIN
      MC(IJ)=VOL(IJ)*ROIN
      SIE(IJ)=SIEIN
      GO TO 300
  260 P(IJ)=PAP
  300 IJ=IJ+NX
C +++
C +++ THE RIGHT EDGE . . .
C +++
      IMJ=IJ-1
      IF(WR.EQ.0) GO TO 400
      GO TO (400,400,400,400,350,360),WR
  350 U(IJ)=UIN
      V(IJ)=VIN
      RO(IMJ)=ROIN
      MC(IMJ)=VOL(IMJ)*ROIN
      SIE(IMJ)=SIEIN
      GO TO 400
  360 P(IMJ)=PAP
  400 CONTINUE
      DO 600 I=1,NXP
      IJ=I
C +++
C +++ THE BOTTOM EDGE . . .
C +++
      IF(WB.EQ.0) GO TO 500
      GO TO (500,500,500,500,450,460),WB
  450 U(IJ)=UIN
      V(IJ)=VIN
      RO(IJ)=ROIN
      MC(IJ)=VOL(IJ)*ROIN
      SIE(IJ)=SIEIN
      GO TO 500
  460 P(IJ)=PAP
  500 IJ=NY*NXP+I
C +++
C +++ THE TOP EDGE . . .
C +++
      IJM=IJ-NXP
      IF(WT.EQ.0) GO TO 600
      GO TO (600,600,600,600,550,560),WT
  550 U(IJ)=UIN
      V(IJ)=VIN
      RO(IJM)=ROIN
      MC(IJM)=VOL(IJM)*ROIN
      SIE(IJM)=SIEIN
      GO TO 600
  560 P(IJM)=PAP
  600 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BEGIN                                                  BEGIN  2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,   COMD   9
     2 GX,GY,IDOT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES,  COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,     COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,              COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,         COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,      COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ            COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   BEGIN  4
C +++ GET JOB IDENTIFICATION, RUN-TIME LIMIT, DATE AND TIME OF DAY.     BEGIN  5
C +++                                                                   BEGIN  6
      READ(5,100) NAME                                                  BEGIN  7
      JNM=10HXPORT-SALE                                                 BEGIN  8
      CALL GETJTL(TIMLMT)                                               BEGIN  9
      CALL DATEH(D1)                                                    BEGIN 10
      CALL TIMEH(C1)                                                    BEGIN 11
C +++                                                                   BEGIN 12
C +++ THE FOLLOWING 5 CALLS REFER STRICTLY TO LASL COM SOFTWARE.        BEGIN 13
C +++ FILM CHOICES ARE:  3H105, 2H16, 3H16C, 2H35, 3H35C . . .          BEGIN 14
C +++                                                                   BEGIN 15
      CALL GFR80 (1HU,NAME,60,3H105,5HT3AAA,4HKEEP)                     BEGIN 16
      CALL GRPHLUN(12)                                                  BEGIN 17
      CALL LIB4020                                                      BEGIN 18
      CALL GRPHFTN                                                      BEGIN 19
      CALL SETFLSH                                                      BEGIN 20
C +++                                                                   BEGIN 21
C +++ CLEAR SCM CELL STORAGE BLOCK . . .                                BEGIN 22
C +++                                                                   BEGIN 23
      NWSC1=LOCF(ZZ1)-LOCF(AA)+1                                        BEGIN 24
      DO 10 N=1,NWSC1                                                   BEGIN 25
   10 AA(N)=0.                                                          BEGIN 26
      RETURN                                                            BEGIN 27
  100 FORMAT(8A10)                                                      BEGIN 28
      END                                                               BEGIN 29
```

```
      SUBROUTINE CELSET                                                 CELSET 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   CELSET 4
C +++ INITIALIZE X AND Y IN THIS FIRST LOOP.  THIS LOOP SHOULD BE       CELSET 5
C +++ REPLACED BY THE USER WHEN DOING A SPECIAL GRID GENERATION.  THE   CELSET 6
C +++ REMAINING LOOPS OF CELSET ARE MORE GENERAL IN THEIR APPLICABILITY.CELSET 7
C +++                                                                   CELSET 8
      DO 20 J=1,NYP                                                     CELSET 9
      IJ=(J-1)*NXP+1                                                    CELSET10
      DO 10 I=1,NXP                                                     CELSET11
      X(IJ)=FLOAT(I-1)*DX                                               CELSET12
      Y(IJ)=FLOAT(J-1)*DY                                               CELSET13
   10 IJ=IJ+1                                                           CELSET14
   20 CONTINUE                                                          CELSET15
C +++                                                                   CELSET16
C +++ NEXT, INITIALIZE THE REMAINING VERTEX QUANTITIES                  CELSET17
C +++ (SPECIFIED INFLOW VELS. WILL BE SET IN SUBR. BC)                  CELSET18
C +++                                                                   CELSET19
      DO 140 J=1,NYP                                                    CELSET20
      IJ=(J-1)*NXP+1                                                    CELSET21
      DO 130 I=1,NXP                                                    CELSET22
      R(IJ)=X(IJ)*CYL+OMCYL                                             CELSET23
      U(IJ)=V(IJ)=MV(IJ)=0.                                             CELSET24
  130 IJ=IJ+1                                                           CELSET25
  140 CONTINUE                                                          CELSET26
C +++                                                                   CELSET27
C +++ INITIALIZE THE CELL CENTERED QUANTITIES                           CELSET28
C +++ FIRST, GET THE VOLUMES, REQUIRED FOR MC CALCULATION               CELSET29
C +++                                                                   CELSET30
      CALL VOLUME                                                       CELSET31
      DO 160 J=1,NY                                                     CELSET32
      IJ=(J-1)*NXP+1                                                    CELSET33
      DO 150 I=1,NX                                                     CELSET34
      RO(IJ)=ROI                                                        CELSET35
      SIE(IJ)=SIEI                                                      CELSET36
C +++                                                                   CELSET37
C +++ INITIAL PRESSURE IS E.O.S. PRESSURE, EXCEPT FOR INCOMPRESSIBLE    CELSET38
C +++ CASE (INC=1), FOR WHICH INITIAL PRESSURE IS ZERO...               CELSET39
C +++                                                                   CELSET40
      CALL EOS(P(IJ),RO(IJ),SIE(IJ),0.,RO(IJ))                          CELSET41
      MC(IJ)=VOL(IJ)*RO(IJ)                                             CELSET42
  150 IJ=IJ+1                                                           CELSET43
  160 CONTINUE                                                          CELSET44
C +++                                                                   CELSET45
C +++ BCSET WILL ADJUST BOUNDARY VALUES OF U,V,MC,RO,SIE,P AS REQD.     CELSET46
C +++ IF INFLOW OR PRESSURE BOUNDARIES ARE SPECIFIED . . .              CELSET47
C +++                                                                   CELSET48
      CALL BCSET                                                        CELSET49
```

```
C +++
C +++ INITIALIZE VERTEX MASSES AND THEIR RECIPROCALS
C +++
      DO 180 J=1,NY
      IJ=(J-1)*NXP+1
      IJP=IJ+NXP
      DO 170 I=1,NX
      IPJ=IJ+1
      IPJP=IJP+1
      QM=0.25*MC(IJ)
      MV(IPJ) =MV(IPJ) +QM
      MV(IPJP)=MV(IPJP)+QM
      MV(IJP) =MV(IJP) +QM
      MV(IJ)  =MV(IJ)  +QM
      IJ=IPJ
  170 IJP=IPJP
  180 CONTINUE
      DO 200 J=1,NYP
      IJ=(J-1)*NXP+1
      DO 190 I=1,NXP
      RMV(IJ)=1./MV(IJ)
  190 IJ=IJ+1
  200 CONTINUE
      CALL BC(U,V)
      RETURN
      END
```

```
      SUBROUTINE CONTUR(L,CQ)                                           CONTUR 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                               COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,X1,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
      DIMENSION CQ(1),IX1(2),IY1(2),XCO(4),YCO(4),CON(11)               CONTUR 4
      DIMENSION IDEN(3)                                                 CONTUR 5
      DATA (IDEN(I),I=1,3)/10HISOBARS   ,10HISOPYCNICS,10HISOTHERMS /   CONTUR 6
C +++                                                                   CONTUR 7
C +++ CONTOUR PLOT OF ARRAY CQ.  (CALLED FROM SUBR. FULOUT)             CONTUR 8
C +++                                                                   CONTUR 9
      IF(NX.EQ.1 .OR. NY.EQ.1) RETURN                                   CONTUR10
C +++                                                                   CONTUR11
C +++ SET CONTOUR VALUES                                                CONTUR12
C +++                                                                   CONTUR13
      QMN=1.E+200                                                       CONTUR14
      QMX=-QMN                                                          CONTUR15
      DO 20 J=JFIRST,JLAST                                              CONTUR16
      IJ=(J-1)*NXP+1                                                    CONTUR17
      DO 10 I=IFIRST,ILAST                                              CONTUR18
      QMN=AMIN1(CQ(IJ),QMN)                                             CONTUR19
      QMX=AMAX1(CQ(IJ),QMX)                                             CONTUR20
   10 IJ=IJ+1                                                           CONTUR21
   20 CONTINUE                                                          CONTUR22
      XX=QMX-QMN                                                        CONTUR23
      IF(XX.LE.0.001*AMAX1(ABS(QMX),ABS(QMN))) RETURN                   CONTUR24
      DQ=0.1*(XX+1.E-50)                                                CONTUR25
      DO 30 K=1,11                                                      CONTUR26
   30 CON(K)=QMN+(FLOAT(K-1))*DQ                                        CONTUR27
C +++                                                                   CONTUR28
C +++ PRINT THE LABELS ON THE PLOT                                      CONTUR29
C +++                                                                   CONTUR30
      CALL ADV (1)                                                      CONTUR31
      CALL LINCNT(59)                                                   CONTUR32
      WRITE(12,170) JNM,D1,C1,NAME,T,NCYC                               CONTUR33
      WRITE(12,180) IDEN(L),QMN,QMX,CON(2),CON(10),DQ                   CONTUR34
C +++                                                                   CONTUR35
C +++ DRAW THE CONTOURS, CONSIDERING CELLS GROUPED IN QUADRANTS -       CONTUR36
C +++ IJ,IPJ,IJP,IPJP                                                   CONTUR37
C +++                                                                   CONTUR38
      DO 130 J=JFIRST,JLASTM                                            CONTUR39
      IJ=(J-1)*NX    FIRST                                              CONTUR40
      IJP=IJ+NXP                                                        CONTUR41
      DO 120 I=IFIRST,ILASTM                                            CONTUR42
      IPJ=IJ+1                                                          CONTUR43
      IPJP=IJP+1                                                        CONTUR44
      N=0                                                               CONTUR45
C +++                                                                   CONTUR46
C +++ DRAW ALL CONTOUR SEGMENTS PASSING THRU THE AREA BOUNDED           CONTUR47
C +++ BY THE CENTERS OF THE FOUR CELLS . . .                            CONTUR48
C +++                                                                   CONTUR49
```

```
      DO 110 K=2,11
      K1=K2=K3=K4=0
      IF(CQ(IJ)  .LE.CON(K)) K1=1
      IF(CQ(IPJ) .LE.CON(K)) K2=1
      IF(CQ(IJP) .LE.CON(K)) K3=1
      IF(CQ(IPJP).LE.CON(K)) K4=1
C +++
C +++ IF PRODUCT .NE. 0, THEN ALL 4 ARE = 1.
C +++ IF SUM. EQ. 0, THEN ALL 4 ARE = 0.  FOR EITHER OF THESE,
C +++ THE CONTOUR DOES NOT PASS THRU, SO GO TO 110 TO TRY NEXT
C +++ VALUE ON K LOOP . . .
C +++
      IF(K1*K2*K3*K4.NE.0.OR.K1+K2+K3+K4.EQ.0) GO TO 110
      IF(N.GT.0) GO TO 60
      IJB=IJ
      IJA=IJP
      DO 50 JJ=1,2
      DO 40 II=1,2
      IPJB=IJB+1
      IPJA=IJA+1
      N=N+1
      XCO(N)=.25*(X(IPJB)+X(IPJA)+X(IJA)+X(IJB))
      YCO(N)=.25*(Y(IPJB)+Y(IPJA)+Y(IJA)+Y(IJB))
      IJA=IPJA
   40 IJB=IPJB
      IJB=IJP
   50 IJA=IJP+NXP
   60 LL=0
      IF(K1+K3.NE.1) GO TO 70
      IC1=1
      IC2=3
      IJ1=IJ
      IJ2=IJP
      KR1=1
      GO TO 100
   70 IF(K1+K2.NE.1) GO TO 80
      IC1=1
      IC2=2
      IJ1=IJ
      IJ2=IPJ
      KR1=2
      GO TO 100
   80 IF(K2+K4.NE.1) GO TO 90
      IC1=2
      IC2=4
      IJ1=IPJ
      IJ2=IPJP
      KR1=3
      GO TO 100
   90 IF(K3+K4.NE.1) GO TO 110
      IC1=3
      IC2=4
      IJ1=IJP
      IJ2=IPJP
      KR1=4
  100 LL=LL+1
      XX=(CON(K)-CQ(IJ1))/(CQ(IJ2)-CQ(IJ1))
      IX1(LL)=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XL)*XCONV
      IY1(LL)=FIYB-(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YB)*YCONV
      IF(LL.LT.2) GO TO (70,80,90,110),KR1
      CALL DRV (IX1(1),IY1(1),IX1(2),IY1(2))
      IF(K.EQ.2) CALL PLT (IX1(1),IY1(1),35)
      IF(K.EQ.10) CALL PLT (IX1(1),IY1(1),24)
```

```
      LL=0                                                              CONTU113
      IF(IJ2.EQ.IPJ) GO TO 80                                           CONTU114
  110 CONTINUE                                                          CONTU115
      IJ=IPJ                                                            CONTU116
  120 IJP=IPJP                                                          CONTU117
  130 CONTINUE                                                          CONTU118
C +++                                                                   CONTU119
C +++ DRAW THE FRAME TO OUTLINE THE MESH PERIPHERY                      CONTU120
C +++                                                                   CONTU121
      DO 150 J=1,NY                                                     CONTU122
      IJ=(J-1)*NXP+1                                                    CONTU123
      IJP=IJ+NXP                                                        CONTU124
      DO 140 I=1,NX                                                     CONTU125
      IPJ=IJ+1                                                          CONTU126
      IPJP=IJP+1                                                        CONTU127
      IX1(1)=FIXL+(X(IPJ)-XL)*XCONV                                     CONTU128
      IY1(1)=FIYB-(Y(IPJ)-YB)*YCONV                                     CONTU129
      IX2=FIXL+(X(IPJP)-XL)*XCONV                                       CONTU130
      IY2=FIYB-(Y(IPJP)-YB)*YCONV                                       CONTU131
      IX3=FIXL+(X(IJP)-XL)*XCONV                                        CONTU132
      IY3=FIYB-(Y(IJP)-YB)*YCONV                                        CONTU133
      IX4=FIXL+(X(IJ)-XL)*XCONV                                         CONTU134
      IY4=FIYB-(Y(IJ)-YB)*YCONV                                         CONTU135
      IF(I.EQ.1) CALL DRV (IX3,IY3,IX4,IY4)                             CONTU136
      IF(J.EQ.1) CALL DRV (IX4,IY4,IX1(1),IY1(1))                       CONTU137
      IF(I.EQ.NX) CALL DRV (IX1(1),IY1(1),IX2,IY2)                      CONTU138
      IF(J.EQ.NY) CALL DRV (IX2,IY2,IX3,IY3)                            CONTU139
      IJ IPJ                                                            CONTU140
  140 IJP=IPJP                                                          CONTU141
  150 CONTINUE                                                          CONTU142
      RETURN                                                            CONTU143
  170 FORMAT(2X,A10,2(2X,A8),2X,8A10/40X,3H T=,1PE12.5,6H CYCLE,I5)     CONTU144
  180 FORMAT(1X,A10,5H MIN=,1PE12.5,5H MAX=,E12.5,3H L=,E12.5,3H H=,    CONTU145
     1 E12.5,4H DQ=,E12.5)                                              CONTU146
      END                                                               CONTU147
```

```
      SUBROUTINE DSDP                                                   DSDP   2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZI                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,CI,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GMI,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   DSDP   4
C +++ PHASE 2.  NUMERICAL EVALUATION OF RELAXATION FACTOR               DSDP   5
C +++ TO BE USED IN THE PRESSURE ITERATION (SUBR. PRESIT) . . .         DSDP   6
C +++                                                                   DSDP   7
C +++                                                                   DSDP   8
      DATA PTEMP,PSTAR /0.,0./                                          DSDP   9
      DP=DPCOF/(DT*DT)                                                  DSDP  10
      DTP=0.5*DT*DP                                                     DSDP  11
      DO 20 J=JFIRST,JLAST                                              DSDP  12
      IJ=(J-1)*NXP+IFIRST                                               DSDP  13
      IJP=IJ+NXP                                                        DSDP  14
      DO 10 I=IFIRST,ILAST                                              DSDP  15
      IPJ=IJ+1                                                          DSDP  16
      IPJP=IJP+1                                                        DSDP  17
      X1=X(IPJ)                                                         DSDP  18
      X2=X(IPJP)                                                        DSDP  19
      X3=X(IJP)                                                         DSDP  20
      X4=X(IJ)                                                          DSDP  21
      R1=R(IPJ)                                                         DSDP  22
      R2=R(IPJP)                                                        DSDP  23
      R3=R(IJP)                                                         DSDP  24
      R4=R(IJ)                                                          DSDP  25
      Y1=Y(IPJ)                                                         DSDP  26
      Y2=Y(IPJP)                                                        DSDP  27
      Y3=Y(IJP)                                                         DSDP  28
      Y4=Y(IJ)                                                          DSDP  29
      U1=UL(IPJ)                                                        DSDP  30
      U2=UL(IPJP)                                                       DSDP  31
      U3=UL(IJP)                                                        DSDP  32
      U4=UL(IJ)                                                         DSDP  33
      V1=VL(IPJ)                                                        DSDP  34
      V2=VL(IPJP)                                                       DSDP  35
      V3=VL(IJP)                                                        DSDP  36
      V4=VL(IJ)                                                         DSDP  37
      X1P=X1+U1*DT                                                      DSDP  38
      Y1P=Y1+V1*DT                                                      DSDP  39
      X2P=X2+U2*DT                                                      DSDP  40
      Y2P=Y2+V2*DT                                                      DSDP  41
      X3P=X3+U3*DT                                                      DSDP  42
      Y3P=Y3+V3*DT                                                      DSDP  43
      X4P=X4+U4*DT                                                      DSDP  44
      Y4P=Y4+V4*DT                                                      DSDP  45
      R1P=X1P*CYL+OMCYL                                                 DSDP  46
      R2P=X2P*CYL+OMCYL                                                 DSDP  47
      R3P=X3P*CYL+OMCYL                                                 DSDP  48
      R4P=X4P*CYL+OMCYL                                                 DSDP  49
```

```
      ATR=.5*((X3P-X2P)*(Y1P-Y2P)-(X1P-X2P)*(Y3P-Y2P))
      ABL=.5*((X1P-X4P)*(Y3P-Y4P)-(X3P-X4P)*(Y1P-Y4P))
      VOLLZ=THIRD*((R1P+R2P+R3P)*ATR+(R3P+R4P+R1P)*ABL)
      XITEMP=SIE(IJ)+P(IJ)*(1.0-VOLLZ/VOL(IJ))/RO(IJ)
      ROTEMP=RO(IJ)*VOL(IJ)/VOLLZ
      CALL EOS(PTEMP,ROTEMP,XITEMP,P(IJ),RO(IJ))
      RM1=RMV(IPJ)
      RM2=RMV(IPJP)
      RM3=RMV(IJP)
      RM4=RMV(IJ)
      Y24=Y2-Y4
      Y31=Y3-Y1
      XR24=0.5*(R2+R4)*(X2-X4)
      XR31=0.5*(R3+R1)*(X3-X1)
      U1P=U1+DTP*RM1*Y24*R1
      U2P=U2+DTP*RM2*Y31*R2
      U3P=U3-DTP*RM3*Y24*R3
      U4P=U4-DTP*RM4*Y31*R4
      V1P=V1-DTP*RM1*XR24
      V2P=V2-DTP*RM2*XR31
      V3P=V3+DTP*RM3*XR24
      V4P=V4+DTP*RM4*XR31
      X1P=X1+U1P*DT
      Y1P=Y1+V1P*DT
      X2P=X2+U2P*DT
      Y2P=Y2+V2P*DT
      X3P=X3+U3P*DT
      Y3P=Y3+V3P*DT
      X4P=X4+U4P*DT
      Y4P=Y4+V4P*DT
      R1P=X1P*CYL+OMCYL
      R2P=X2P*CYL+OMCYL
      R3P=X3P*CYL+OMCYL
      R4P=X4P*CYL+OMCYL
      ATR=.5*((X3P-X2P)*(Y1P-Y2P)-(X1P-X2P)*(Y3P-Y2P))
      ABL=.5*((X1P-X4P)*(Y3P-Y4P)-(X3P-X4P)*(Y1P-Y4P))
      VOLL=THIRD*((R1P+R2P+R3P)*ATR+(R3P+R4P+R1P)*ABL)
      X1STAR=XITEMP+PTEMP*(1.0-VOLL/VOLLZ)/ROTEMP
      ROSTAR=RO(IJ)*VOL(IJ)/VOLL
      PPDP=P(IJ)+DP
      CALL EOS(PSTAR,ROSTAR,XISTAR,PPDP,RO(IJ))
      RDSDP(IJ)=DP/(PTEMP+DP-PSTAR)*OM
      IJ=IPJ
   10 IJP=IPJP
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ENERGY                                                 ENERGY 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   ENERGY 4
C +++ CALCULATE ENERGY CHANGES DUE TO PDV WORK AND VISCOUS STRESSES     ENERGY 5
C +++                                                                   ENERGY 6
      DO 20 J=JFIRST,JLAST                                              ENERGY 7
      IJ=(J-1)*NXP+IFIRST                                               ENERGY 8
      IJP=IJ+NXP                                                        ENERGY 9
      DO 10 I=IFIRST,ILAST                                              ENERGY10
      IPJ=IJ+1                                                          ENERGY11
      IPJP=IJP+1                                                        ENERGY12
      X1=X(IPJ)                                                         ENERGY13
      Y1=Y(IPJ)                                                         ENERGY14
      R1=R(IPJ)                                                         ENERGY15
      UTC1=U(IPJ)+UL(IPJ)                                               ENERGY16
      VTC1=V(IPJ)+VL(IPJ)                                               ENERGY17
      X2=X(IPJP)                                                        ENERGY18
      Y2=Y(IPJP)                                                        ENERGY19
      R2=R(IPJP)                                                        ENERGY20
      UTC2=U(IPJP)+UL(IPJP)                                             ENERGY21
      VTC2=V(IPJP)+VL(IPJP)                                             ENERGY22
      X3=X(IJP)                                                         ENERGY23
      Y3=Y(IJP)                                                         ENERGY24
      R3=R(IJP)                                                         ENERGY25
      UTC3=U(IJP)+UL(IJP)                                               ENERGY26
      VTC3=V(IJP)+VL(IJP)                                               ENERGY27
      X4=X(IJ)                                                          ENERGY28
      Y4=Y(IJ)                                                          ENERGY29
      R4=R(IJ)                                                          ENERGY30
      UTC4=U(IJ)+UL(IJ)                                                 ENERGY31
      VTC4=V(IJ)+VL(IJ)                                                 ENERGY32
      Y24=Y2-Y4                                                         ENERGY33
      Y31=Y3-Y1                                                         ENERGY34
      X24=X2-X4                                                         ENERGY35
      X31=X3-X1                                                         ENERGY36
      HR24=0.5*(R2+R4)                                                  ENERGY37
      HR13=0.5*(R1+R3)                                                  ENERGY38
      XX1=HR24*(PIXY(IJ)*X24-PIXX(IJ)*Y24)                              ENERGY39
      XX2=HR13*(PIXY(IJ)*X31-PIXX(IJ)*Y31)                              ENERGY40
      XX3=HR24*(PIYY(IJ)*X24-PIXY(IJ)*Y24)                              ENERGY41
      XX4=HR13*(PIYY(IJ)*X31-PIXY(IJ)*Y31)                              ENERGY42
      DV=Y24*(UTC1*R1-UTC3*R3) + Y31*(UTC2*R2-UTC4*R4)                  ENERGY43
     1   -HR24*X24*(VTC1-VTC3) - HR13*X31*(VTC2-VTC4)                   ENERGY44
      AREA=0.5*(X24*Y31-X31*Y24)                                        ENERGY45
      DVIS=XX1*(UTC1-UTC3)+XX2*(UTC2-UTC4)                              ENERGY46
     1     -PITH(IJ)*(UTC1+UTC2+UTC3+UTC4)*0.5*AREA                     ENERGY47
     2     +XX3*(VTC1-VTC3)+XX4*(VTC2-VTC4)                             ENERGY48
      SIE(IJ)=SIE(IJ)-((P(IJ)+Q(IJ))*DV + DVIS)*0.25*DT/MC(IJ)          ENERGY49
```

```
      IJP=IPJP                                                          ENERGY50
   10 IJ=IPJ                                                            ENERGY51
   20 CONTINUE                                                          ENERGY52
      RETURN                                                            ENERGY53
      END                                                               ENERGY54
```

```
      SUBROUTINE EOS(PTEMP,ROTEMP,SIETMP,PLCUR,ROLCUR)                  EOS    2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
      IF(INC.EQ.1) GO TO 100                                            EOS    4
C +++                                                                   EOS    5
C +++ EQUATION OF STATE FOR REAL MATERIAL GOES HERE . . .               EOS    6
C +++ (STIFFENED GAS + IDEAL GAS E.O.S. IS SHOWN HERE AS AN EXAMPLE)    EOS    7
C +++                                                                   EOS    8
      PTEMP=ASQ*(ROTEMP-RON)+GM1*ROTEMP*SIETMP                          EOS    9
      RETURN                                                            EOS   10
C +++                                                                   EOS   11
C +++ PSEUDO-PRESSURE CALCULATION FOR INCOMPRESSIBLE FLOWS (INC=1) . . .EOS   12
C +++                                                                   EOS   13
  100 IF(ROLCUR.EQ.0.) GO TO 110                                        EOS   14
      PTEMP=PLCUR + ROTEMP/ROLCUR - 1.0                                 EOS   15
      RETURN                                                            EOS   16
  110 PTEMP=0.                                                          EOS   17
      RETURN                                                            EOS   18
      END                                                               EOS   19
```

```
      SUBROUTINE FULOUT (LUN)                                           FULOUT 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
```

```
      IF(LUN.EQ.0) GO TO 10                                             FULOUT 4
      IF(LUN.EQ.6) GO TO 40                                             FULOUT 5
      TWFILM=TWFILM+TFILM                                               FULOUT 6
C +++                                                                   FULOUT 7
C +++ SETUP PLOT SCALING AND CALL THE VARIOUS PLOT AND PRINT SUBRS.     FULOUT 8
C +++                                                                   FULOUT 9
   10 XL=YB=1.E+100                                                     FULOUT10
      XR=YT=VMAX=-XL                                                    FULOUT11
      DO 30 J=1,NYP                                                     FULOUT12
      IJ=(J-1)*NXP+1                                                    FULOUT13
      DO 20 I=1,NXP                                                     FULOUT14
      XL=AMIN1(XL,X(IJ))                                                FULOUT15
      XR=AMAX1(XR,X(IJ))                                                FULOUT16
      YB=AMIN1(YB,Y(IJ))                                                FULOUT17
      YT=AMAX1(YT,Y(IJ))                                                FULOUT18
      VMAX=AMAX1(VMAX,ABS(U(IJ)),ABS(V(IJ)))                            FULOUT19
   20 IJ=IJ+1                                                           FULOUT20
   30 CONTINUE                                                          FULOUT21
      IF(VMAX.NE.0.) DROU=0.9*XR/(FLOAT(NX)*VMAX)                       FULOUT22
      FIYB=900.                                                         FULOUT23
C +++                                                                   FULOUT24
C +++ MAKE PLOTS SLIGHTLY UNDERSIZE TO ENSURE VELOCITY VECTORS          FULOUT25
C +++ WILL BE PLOTTED PROPERLY AT MESH BOUNDARIES . . .                 FULOUT26
C +++                                                                   FULOUT27
      XTE=0.025*(XR-XL)                                                 FULOUT28
      YTE=0.025*(YT-YB)                                                 FULOUT29
      XL=XL-XTE                                                         FULOUT30
      XR=XR+XTE                                                         FULOUT31
      YB=YB-YTE                                                         FULOUT32
      YT=YT+YTE                                                         FULOUT33
      XD=(XR-XL)/(YT-YB)                                                FULOUT34
      YY=0.                                                             FULOUT35
      IF(XD.LT.1.13556) YY=1.                                           FULOUT36
      FIXL=AMAX1(0.,(511.-450.*XD)*YY)                                  FULOUT37
      FIXR=(511.+450.*XD)*YY + 1022.*(1.-YY)                            FULOUT38
      FIYT=(900.-1022./XD)*(1.-YY)                                      FULOUT39
      XCONV=(FIXR-FIXL)/(XR-XL)                                         FULOUT40
      YCONV=(FIYB-FIYT)/(YT-YB)                                         FULOUT41
      CALL ZONPLT                                                       FULOUT42
      CALL VELPLT                                                       FULOUT43
      CALL CONTUR(1,P)                                                  FULOUT44
      CALL CONTUR(2,RO)                                                 FULOUT45
      CALL CONTUR(3,SIE)                                                FULOUT46
      IF(LPR.EQ.1 .OR. LPR.EQ.2) CALL LNGPRT(12)                        FULOUT47
      CALL GLOBAL(12)                                                   FULOUT48
      IF(LUN.EQ.0) GO TO 50                                             FULOUT49
      RETURN                                                            FULOUT50
   40 TWPRTR=TWPRTR+TPRTR                                               FULOUT51
   50 IF(LPR.GT.1) CALL LNGPRT(6)                                       FULOUT52
      CALL GLOBAL(6)                                                    FULOUT53
      RETURN                                                            FULOUT54
      END                                                               FULOUT55
```

```
      SUBROUTINE GLOBAL (LUN)                                           GLOBAL 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,CI,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GMI,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,X1,X1COF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   GLOBAL 4
C +++ COMPUTE TOTAL MASS, MOMENTUM, AND ENERGY OF THE SYSTEM            GLOBAL 5
C +++                                                                   GLOBAL 6
      TOTM=TOTI=TOTU=TOTV=TOTK=0.                                       GLOBAL 7
      DO 20 J=1,NY                                                      GLOBAL 8
      IJ=(J-1)*NXP+1                                                    GLOBAL 9
      DO 10 I=1,NX                                                      GLOBAL10
      TOTM=TOTM+MC(IJ)                                                  GLOBAL11
      TOTI=TOTI+MC(IJ)*SIE(IJ)                                          GLOBAL12
   10 IJ=IJ+1                                                           GLOBAL13
   20 CONTINUE                                                          GLOBAL14
      DO 40 J=1,NYP                                                     GLOBAL15
      IJ=(J-1)*NXP+1                                                    GLOBAL16
      DO 30 I=1,NXP                                                     GLOBAL17
      TOTU=TOTU+MV(IJ)*U(IJ)                                            GLOBAL18
      TOTV=TOTV+MV(IJ)*V(IJ)                                            GLOBAL19
      TOTK=TOTK+MV(IJ)*.5*(U(IJ)*U(IJ)+V(IJ)*V(IJ))                     GLOBAL20
   30 IJ=IJ+1                                                           GLOBAL21
   40 CONTINUE                                                          GLOBAL22
      TOTE=TOTK+TOTI                                                    GLOBAL23
      WRITE(LUN,50) T,NCYC,TOTE,TOTI,TOTM,TOTU,TOTV                     GLOBAL24
      RETURN                                                            GLOBAL25
   50 FORMAT(3H T=,1PE12.5,6H CYCLE,I5,7H TOT E=,E15.8,5H SIE=,E15.8/   GLOBAL26
     1 26X,6H MASS=,E15.8,7H U MOM=,E15.8,7H V MOM=,E15.8)              GLOBAL27
      END                                                               GLOBAL28
```

```
      SUBROUTINE LNGPRT (LUN)                                           LNGPRT 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,CI,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDOT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   LNGPRT 4
C +++ LONG PRINT OF X,Y,U,V,SIE,RO,MC,VOL,P FOR ALL CELLS IN MESH.      LNGPRT 5
C +++ DESTINATION IS FICHE (LUN=12), AND/OR LINE PRINTER (LUN=6)        LNGPRT 6
C +++                                                                   LNGPRT 7
      LPR1=LPR2=0                                                       LNGPRT 8
      IF(LPR.GT.1 .AND. LUN.EQ.6) LPR1=1                                LNGPRT 9
      IF(LPR.LT.3 .AND. LUN.EQ.12) LPR2=1                               LNGPRT10
      LINES=99                                                          LNGPRT11
      DO 50 J=1,NYP                                                     LNGPRT12
      IJ=(J-1)*NXP+1                                                    LNGPRT13
      DO 40 I=1,NXP                                                     LNGPRT14
      IF(LINES.LT.56) GO TO 10                                          LNGPRT15
      LINES=0                                                           LNGPRT16
      IF(LPR1.GT.0) WRITE(6,100) JNM,DI,CI,NAME,T,NCYC                  LNGPRT17
      IF(LPR2.GT.0) WRITE(12,100) JNM,DI,CI,NAME,T,NCYC                 LNGPRT18
   10 IF(LOOPS.LT.LOOPMX) GO TO 20                                      LNGPRT19
C +++                                                                   LNGPRT20
C +++ SPECIAL PRINT FOR RUN-ABORT CASE . . .                            LNGPRT21
C +++                                                                   LNGPRT22
      IF(LPR1.GT.0) WRITE (6,110) I,J,X(IJ),Y(IJ),UL(IJ),VL(IJ),        LNGPRT23
     1 SIE(IJ),ROL(IJ),MC(IJ),VOL(IJ),PL(IJ)                            LNGPRT24
      IF(LPR2.GT.0) WRITE(12,110) I,J,X(IJ),Y(IJ),UL(IJ),VL(IJ),        LNGPRT25
     1 SIE(IJ),ROL(IJ),MC(IJ),VOL(IJ),PL(IJ)                            LNGPRT26
      GO TO 30                                                          LNGPRT27
C +++                                                                   LNGPRT28
C +++ NORMAL OUTPUT CASE . . .                                          LNGPRT29
C +++                                                                   LNGPRT30
   20 IF(LPR1.GT.0) WRITE(6,110) I,J,X(IJ),Y(IJ),U(IJ),V(IJ),SIE(IJ),   LNGPRT31
     1 RO(IJ),MC(IJ),VOL(IJ),P(IJ)                                      LNGPRT32
      IF(LPR2.GT.0) WRITE(12,110) I,J,X(IJ),Y(IJ),U(IJ),V(IJ),SIE(IJ),  LNGPRT33
     1 RO(IJ),MC(IJ),VOL(IJ),P(IJ)                                      LNGPRT34
   30 LINES=LINES+1                                                     LNGPRT35
   40 IJ=IJ+1                                                           LNGPRT36
   50 CONTINUE                                                          LNGPRT37
      IF(LPR2.GT.0) CALL ADV(1)                                         LNGPRT38
      RETURN                                                            LNGPRT39
  100 FORMAT(1H1,2X,A10,2(2X,A8),2X,8A10/40X,3H T=,1PE12.5,6H CYCLE,I5/ LNGPRT40
     1 1H0,7H  I    J,6X,1HX,10X,1HY,10X,1HU,10X,1HV,                   LNGPRT41
     2 10X,3HSIE,8X,3HRHO,7X,4HMASS,8X,3HVOL,8X,1HP)                    LNGPRT42
  110 FORMAT(2I4,9(1PE11.3))                                            LNGPRT43
      END                                                               LNGPRT44
```

```
      SUBROUTINE NEWCYC                                                 NEWCYC 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,CI,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GMI,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
      DATA TIME /0./                                                    NEWCYC 4
C +++                                                                   NEWCYC 5
C +++ BEGIN CYCLE - PROVIDE MONITOR PRINT, THEN TEST FOR                NEWCYC 6
C +++ OUTPUT AND RUN TERMINATION.  IF CONTINUING, INCREMENT             NEWCYC 7
C +++ TIME AND CYCLE NUMBER . . .                                       NEWCYC 8
C +++                                                                   NEWCYC 9
      IF(NCYC.LE.1) CALL FULOUT(0)                                      NEWCYC10
      IF((MOD(NCYC,25).EQ.0) .OR. (T.GE.TWFIN))                         NEWCYC11
     1WRITE(59,100) NCYC,T,DT,NUMIT,GRIND,IDDT                          NEWCYC12
      WRITE (6,100) NCYC,T,DT,NUMIT,GRIND,IDDT                          NEWCYC13
      WRITE(12,100) NCYC,T,DT,NUMIT,GRIND,IDDT                          NEWCYC14
      IF(T.GE.TWFILM) CALL FULOUT(12)                                   NEWCYC15
      IF(T.GE.TWPRTR) CALL FULOUT(6)                                    NEWCYC16
      TOLD=TIME                                                         NEWCYC17
      CALL SECOND(TIME)                                                 NEWCYC18
      GRIND=(TIME-TOLD)*GRFAC                                           NEWCYC19
      TLEFT=TIMLMT-TIME                                                 NEWCYC20
      IF(TLEFT.LT.180. .AND. TLIMD.EQ.1.) CALL TAPEWR                   NEWCYC21
      IF(T.GE.TWFIN) GO TO 10                                           NEWCYC22
      T=T+DT                                                            NEWCYC23
      NCYC=NCYC+1                                                       NEWCYC24
      RETURN                                                            NEWCYC25
   10 WRITE(59,110)                                                     NEWCYC26
      WRITE (6,110)                                                     NEWCYC27
      WRITE(12,110)                                                     NEWCYC28
      IF(T.LT.TWFILM) CALL FULOUT(12)                                   NEWCYC29
      IF(T.LT.TWPRTR) CALL FULOUT(6)                                    NEWCYC30
      CALL EXITA(1)                                                     NEWCYC31
  100 FORMAT(5H NCYC,I6,3H T=,1PE12.5,4H DT=,E12.5,7H NUMIT=,I4,        NEWCYC32
     1 7H GRIND=,0PF7.3,1X,A1)                                          NEWCYC33
  110 FORMAT(19H NORMAL TERMINATION)                                    NEWCYC34
      END                                                               NEWCYC35
```

```
      SUBROUTINE PHASE1                                                 PHASE1 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPH1,ILAST,ILASTM,ILASTV,ILPH1,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPH1,JLAST,JLASTM,JLASTV,JLPH1,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   PHASE1 4
C +++ PHASE 1.  EXPLICIT LAGRANGIAN CALCULATION, IN WHICH WE ADJUST     PHASE1 5
C +++ THE LAGRANGIAN VELOCITIES BY PRESSURE GRADIENTS AND BODY FORCES.  PHASE1 6
C +++                                                                   PHASE1 7
C +++ THE PRESSURE ACCELERATIONS . . .                                  PHASE1 6
C +++                                                                   PHASE1 9
      DO 20 J=JFPH1,JLPH1                                               PHASE110
      IJ=(J-1)*NXP+IFPH1                                                PHASE111
      IJP=IJ+NXP                                                        PHASE112
      DO 10 I=IFPH1,ILPH1                                               PHASE113
      IPJ=IJ+1                                                          PHASE114
      IPJP=IJP+1                                                        PHASE115
      DTP=0.5*DT*P(IJ)                                                  PHASE116
      RM1=RMV(IPJ)                                                      PHASE117
      RM2=RMV(IPJP)                                                     PHASE118
      RM3=RMV(IJP)                                                      PHASE119
      RM4=RMV(IJ)                                                       PHASE120
      R1=R(IPJ)                                                         PHASE121
      R2=R(IPJP)                                                        PHASE122
      R3=R(IJP)                                                         PHASE123
      R4=R(IJ)                                                          PHASE124
      XR24=(X(IPJP)-X(IJ) )*.5*(R2+R4)                                  PHASE125
      XR31=(X(IJP) -X(IPJ))*.5*(R3+R1)                                  PHASE126
      Y24=Y(IPJP)-Y(IJ)                                                 PHASE127
      Y31=Y(IJP)-Y(IPJ)                                                 PHASE128
      UL(IPJ) =UL(IPJ) +DTP*RM1*Y24*R1                                  PHASE129
      UL(IPJP)=UL(IPJP)+DTP*RM2*Y31*R2                                  PHASE130
      UL(IJP) =UL(IJP) -DTP*RM3*Y24*R3                                  PHASE131
      UL(IJ)  =UL(IJ)  -DTP*RM4*Y31*R4                                  PHASE132
      VL(IPJ) =VL(IPJ) -DTP*RM1*XR24                                    PHASE133
      VL(IPJP)=VL(IPJP)-DTP*RM2*XR31                                    PHASE134
      VL(IJP) =VL(IJP) +DTP*RM3*XR24                                    PHASE135
      VL(IJ)  =VL(IJ)  +DTP*RM4*XR31                                    PHASE136
      IJ=IPJ                                                            PHASE137
   10 IJP=IPJP                                                          PHASE138
   20 CONTINUE                                                          PHASE139
      IF(GX.EQ.0. .AND. GY.EQ.0.) GO TO 50                              PHASE140
C +++                                                                   PHASE141
C +++ THE BODY ACCELERATIONS . . .                                      PHASE142
C +++                                                                   PHASE143
      DTGX=DT*GX                                                        PHASE144
      DTGY=DT*GY                                                        PHASE145
      DO 40 J=JFIRST,JLASTV                                             PHASE146
      IJ=NXP*(J-1)+IFIRST                                               PHASE147
      DO 30 I=IFIRST,ILASTV                                             PHASE148
      VL(IJ)=VL(IJ)+DTGY                                                PHASE149
```

```
      UL(IJ)=UL(IJ)+DTGX                                                PHASE150
   30 IJ=IJ+1                                                           PHASE151
   40 CONTINUE                                                          PHASE152
   50 CALL BC(UL,VL)                                                    PHASE153
      RETURN                                                            PHASE154
      END                                                               PHASE155
```

```
      SUBROUTINE PRESIT                                                 PRESIT 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,CI,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   PRESIT 4
C +++ PHASE 2.  THE NEWTON-RAPHSON PRESSURE ITERATION . . .             PRESIT 5
C +++                                                                   PRESIT 6
      DATA PSTAR /0./                                                   PRESIT 7
      NUMIT=0                                                           PRESIT 8
   10 NUMIT=NUMIT+1                                                     PRESIT 9
      MUSTIT=0                                                          PRESIT10
      PMAXN=0.                                                          PRESIT11
      DO 30 J=JFIRST,JLAST                                              PRESIT12
      IJ=(J-1)*NXP+IFIRST                                               PRESIT13
      IJP=IJ+NXP                                                        PRESIT14
      DO 20 I=IFIRST,ILAST                                              PRESIT15
      IPJ=IJ+1                                                          PRESIT16
      IPJP=IJP+1                                                        PRESIT17
      X1=X(IPJ)                                                         PRESIT18
      R1=R(IPJ)                                                         PRESIT19
      Y1=Y(IPJ)                                                         PRESIT20
      U1=UL(IPJ)                                                        PRESIT21
      V1=VL(IPJ)                                                        PRESIT22
      X2=X(IPJP)                                                        PRESIT23
      R2=R(IPJP)                                                        PRESIT24
      Y2=Y(IPJP)                                                        PRESIT25
      U2=UL(IPJP)                                                       PRESIT26
      V2=VL(IPJP)                                                       PRESIT27
      X3=X(IJP)                                                         PRESIT28
      R3=R(IJP)                                                         PRESIT29
      Y3=Y(IJP)                                                         PRESIT30
      U3=UL(IJP)                                                        PRESIT31
      V3=VL(IJP)                                                        PRESIT32
      X4=X(IJ)                                                          PRESIT33
      R4=R(IJ)                                                          PRESIT34
      Y4=Y(IJ)                                                          PRESIT35
      U4=UL(IJ)                                                         PRESIT36
      V4=VL(IJ)                                                         PRESIT37
      X1P=X1+UL(IPJ)*DT                                                 PRESIT38
      X2P=X2+UL(IPJP)*DT                                                PRESIT39
      X3P=X3+UL(IJP)*DT                                                 PRESIT40
      X4P=X4+UL(IJ)*DT                                                  PRESIT41
      Y1P=Y1+VL(IPJ)*DT                                                 PRESIT42
      Y2P=Y2+VL(IPJP)*DT                                                PRESIT43
      Y3P=Y3+VL(IJP)*DT                                                 PRESIT44
      Y4P=Y4+VL(IJ)*DT                                                  PRESIT45
      R1P=X1P*CYL+OMCYL                                                 PRESIT46
      R2P=X2P*CYL+OMCYL                                                 PRESIT47
      R3P=X3P*CYL+OMCYL                                                 PRESIT48
      R4P=X4P*CYL+OMCYL                                                 PRESIT49
```

```
      ATR=.5*((X3P-X2P)*(Y1P-Y2P)-(X1P-X2P)*(Y3P-Y2P))
      ABL=.5*((X1P-X4P)*(Y3P-Y4P)-(X3P-X4P)*(Y1P-Y4P))
      VW=THIRD*((R1P+R2P+R3P)*ATR+(R3P+R4P+R1P)*ABL)
      IF(VW.LE.0.) GO TO 90
      ROL(IJ)=RO(IJ)*VOL(IJ)/VW
      XISTAR=SIE(IJ)+PL(IJ)*(1.-VW/VOL(IJ))/ROL(IJ)
      CALL EOS(PSTAR,ROL(IJ),XISTAR,PL(IJ),RO(IJ))
      S=PL(IJ)-PSTAR
      DP=-S*RDSDP(IJ)
      PL(IJ)=PL(IJ)+DP
      PMAXN=AMAX1(PMAXN,ABS(PL(IJ)))
      Y24=Y2-Y4
      Y31=Y3-Y1
      XR24=(X2-X4)*.5*(R2+R4)
      XR31=(X3-X1)*.5*(R3+R1)
      XX=0.5*DT*DP
      RM1=RMV(IPJ)
      RM2=RMV(IPJP)
      RM3=RMV(IJP)
      RM4=RMV(IJ)
      UL(IPJ) =U1+XX*RM1*Y24*R1
      UL(IPJP)=U2+XX*RM2*Y31*R2
      UL(IJP) =U3-XX*RM3*Y24*R3
      UL(IJ)  =U4-XX*RM4*Y31*R4
      VL(IPJ) =V1-XX*RM1*XR24
      VL(IPJP)=V2-XX*RM2*XR31
      VL(IJP) =V3+XX*RM3*XR24
      VL(IJ)  =V4+XX*RM4*XR31
      IF(ABS(DP).GT.EPS*PMAX) MUSTIT=MUSTIT+1
      IJ=IPJ
   20 IJP=IPJP
   30 CONTINUE
C +++
C +++ IPRES=1 LETS CONTINUATIVE BOUNDARY UL,VL FLOAT DURING ITERATION.
C +++
      IPRES=1
      CALL BC(UL,VL)
      IPRES=0
C +++
C +++ NUMIT=(MAXIT) IN PRESIT-PROBLEM COMMENT SIGNIFIES THAT
C +++ CONVERGENCE FAILURE IS FORCING A DT CUT . . .
C +++ BUT IF INC=1, SIMPLY EXIT ITERATION IF NUMIT=MAXIT.
C +++
      PMAX=PMAXN
      IF(NUMIT.EQ.MAXIT .AND. INC.EQ.1) GO TO 60
      IF(NUMIT.EQ.MAXIT) RETURN
      IF((MUSTIT.GT.0) .OR. (NCYC*NUMIT.EQ.1)) GO TO 10
   60 DO 80 J=JFIRST,JLAST
      IJ=(J-1)*NXP+IFIRST
      DO 70 I=IFIRST,ILAST
      P(IJ)=PL(IJ)
   70 IJ=IJ+1
   80 CONTINUE
      IF(WB.EQ.4 .OR. WL.EQ.4 .OR. WR.EQ.4 .OR. WT.EQ.4) CALL BC(UL,VL)
      RETURN
C +++
C +++ NUMIT=9999 IN PRESIT-PROBLEM COMMENT SIGNIFIES THAT A
C +++ NEGATIVE VOLUME IS FORCING A DT CUT . . .
C +++
   90 NUMIT=9999
      RETURN
      END
```

```
      SUBROUTINE REGRID                                                 REGRID 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,CI,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GMI,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   REGRID 4
C +++ MOVE VERTICES AND COMPUTE RELATIVE VELOCITY BETWEEN FLUID AND GRIDREGRID 5
C +++                                                                   REGRID 6
      DO 20 J=1,NYP                                                     REGRID 7
      IJ=(J-1)*NXP+1                                                    REGRID 8
      DO 10 I=1,NXP                                                     REGRID 9
      X(IJ)=X(IJ)+DT*UG(IJ)                                             REGRID10
      Y(IJ)=Y(IJ)+DT*VG(IJ)                                             REGRID11
      R(IJ)=X(IJ)*CYL+OMCYL                                             REGRID12
      UREL(IJ)=UG(IJ)-UL(IJ)                                            REGRID13
      VREL(IJ)=VG(IJ)-VL(IJ)                                            REGRID14
      MVP(IJ)=0.                                                        REGRID15
   10 IJ=IJ+1                                                           REGRID16
   20 CONTINUE                                                          REGRID17
      RETURN                                                            REGRID18
      END                                                               REGRID19
```

```
      SUBROUTINE RESTEP                                                 RESTEP 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   RESTEP 4
C +++ CALLED WHEN PRESSURE ITERATION EITHER FAILS TO CONVERGE OR        RESTEP 5
C +++ CALCULATES A NEGATIVE VOLUME.  TREATMENT IS TO HALVE DT AND       RESTEP 6
C +++ RESTART THE CYCLE, ALLOWING UP TO (LOOPMX) ATTEMPTS PER CYCLE.    RESTEP 7
C +++                                                                   RESTEP 8
      DATA LOOPS,NCYOLD /0,0/                                           RESTEP 9
      IF(INC.EQ.1) GO TO 30                                             RESTEP10
      IF(NCYC.NE.NCYOLD) LOOPS=0                                        RESTEP11
      NCYOLD=NCYC                                                       RESTEP12
      LOOPS=LOOPS+1                                                     RESTEP13
      DTNEW=DT*0.5                                                      RESTEP14
      IF(NUMIT.EQ.9999) GO TO 10                                        RESTEP15
      WRITE(59,100) T,NCYC,NUMIT,DT,DTNEW                               RESTEP16
      WRITE (6,100) T,NCYC,NUMIT,DT,DTNEW                               RESTEP17
      WRITE(12,100) T,NCYC,NUMIT,DT,DTNEW                               RESTEP18
      GO TO 20                                                          RESTEP19
   10 WRITE(59,110) T,NCYC,DT,DTNEW                                     RESTEP20
      WRITE (6,110) T,NCYC,DT,DTNEW                                     RESTEP21
      WRITE(12,110) T,NCYC,DT,DTNEW                                     RESTEP22
   20 T=T-DT                                                            RESTEP23
      DT=DTNEW                                                          RESTEP24
      NCYC=NCYC-1                                                       RESTEP25
      IF(LOOPS.LT.LOOPMX) RETURN                                        RESTEP26
      WRITE(59,120)                                                     RESTEP27
      WRITE (6,120)                                                     RESTEP28
      WRITE(12,120)                                                     RESTEP29
      GO TO 40                                                          RESTEP30
   30 LOOPS=LOOPS+1                                                     RESTEP31
      IF(LOOPS.LT.10) RETURN                                            RESTEP32
      WRITE(59,130)                                                     RESTEP33
      WRITE (6,130)                                                     RESTEP34
      WRITE(12,130)                                                     RESTEP35
   40 CALL FULOUT(12)                                                   RESTEP36
      CALL FULOUT(6)                                                    RESTEP37
      CALL EXITA(2)                                                     RESTEP38
  100 FORMAT(26H CONVERGENCE FAILURE AT T=,1PE12.5,4H CYC,I4,7H NUMIT=, RESTEP39
     1 I4/19X,7HOLD DT=,E12.5,8H NEW DT=,E12.5)                         RESTEP40
  110 FORMAT(22H NEGATIVE VOLUME AT T=,1PE12.5,4H CYC,                  RESTEP41
     1 I4/15X,7HOLD DT=,E12.5,8H NEW DT=,E12.5)                         RESTEP42
  120 FORMAT(34H JOB ABORTED - TIMESTEP TOO SMALL.)                     RESTEP43
  130 FORMAT(50H JOB ABORTED - INCOMPRESSIBLE FLOW NOT CONVERGING.)     RESTEP44
      END                                                               RESTEP45
```

```
      SUBROUTINE REZONE                                                 REZONE 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPH1,ILAST,ILASTM,ILASTV,ILPH1,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPH1,JLAST,JLASTM,JLASTV,JLPH1,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   REZONE 4
C +++ COMPUTE GRID VELOCITIES UG AND VG FOR USE IN REGRID               REZONE 5
C +++ IREZ=0 IS EULERIAN, IREZ=1 IS LAGRANGIAN, IREZ=2 IS AVERAGING RE- REZONE 6
C +++ ZONE BY RELAXATION, IREZ=3 LEFT VACANT FOR SPECIFICATION BY USER. REZONE 7
C +++                                                                   REZONE 8
      JREZ=IREZ+1                                                       REZONE 9
      GO TO (10,40,70,100),JREZ                                         REZONE10
C +++                                                                   REZONE11
C +++ EULERIAN                                                          REZONE12
C +++                                                                   REZONE13
   10 DO 30 J=1,NYP                                                     REZONE14
      IJ=(J-1)*NXP+1                                                    REZONE15
      DO 20 I=1,NXP                                                     REZONE16
      UG(IJ)=0.                                                         REZONE17
      VG(IJ)=0.                                                         REZONE18
   20 IJ=IJ+1                                                           REZONE19
   30 CONTINUE                                                          REZONE20
      RETURN                                                            REZONE21
C +++                                                                   REZONE22
C +++ LAGRANGIAN                                                        REZONE23
C +++                                                                   REZONE24
   40 DO 60 J=1,NYP                                                     REZONE25
      IJ=(J-1)*NXP+1                                                    REZONE26
      DO 50 I=1,NXP                                                     REZONE27
      UG(IJ)=UL(IJ)                                                     REZONE28
      VG(IJ)=VL(IJ)                                                     REZONE29
   50 IJ=IJ+1                                                           REZONE30
   60 CONTINUE                                                          REZONE31
      RETURN                                                            REZONE32
C +++                                                                   REZONE33
C +++ SAMPLE CONTINUOUS REZONE - RELAX ALL VERTICES EXCEPT THE 4 CORNERSREZONE34
C +++ TOWARD THE AVERAGE POSITION OF THE 4 OR 2 CLOSEST NEIGHBORS...    REZONE35
C +++                                                                   REZONE36
   70 RFODT=RF/DT                                                       REZONE37
      DO 90 J=1,NYP                                                     REZONE38
      IJ=(J-1)*NXP+1                                                    REZONE39
      IJP=IJ+NXP                                                        REZONE40
      IJM=IJ-NXP                                                        REZONE41
      DO 80 I=1,NXP                                                     REZONE42
      IPJ=IJ+1                                                          REZONE43
      IMJ=IJ-1                                                          REZONE44
      UG(IJ)=UL(IJ)                                                     REZONE45
      VG(IJ)=VL(IJ)                                                     REZONE46
      IF(IJ.EQ.1 .OR. IJ.EQ.NXP .OR. IJ.EQ.NXP*NY+1                     REZONE47
     1 .OR. IJ.EQ.NXP*NYP) GO TO 78                                     REZONE48
      IF(I.EQ.1 .OR. I.EQ.NXP) GO TO 72                                 REZONE49
```

```
      IF(J.EQ.1 .OR. J.EQ.NYP) GO TO 74
      XN=.25*(X(IPJ)+X(IJP)+X(IMJ)+X(IJM))
      YN=.25*(Y(IPJ)+Y(IJP)+Y(IMJ)+Y(IJM))
      GO TO 76
   72 XN=.5*(X(IJP)+X(IJM))
      YN=.5*(Y(IJP)+Y(IJM))
      GO TO 76
   74 XN=.5*(X(IPJ)+X(IMJ))
      YN=.5*(Y(IPJ)+Y(IMJ))
   76 UG(IJ)=UL(IJ)+RFODT*(XN-X(IJ))
      VG(IJ)=VL(IJ)+RFODT*(YN-Y(IJ))
   78 IJ=IJ+1
      IJP=IJP+1
   80 IJM=IJM+1
   90 CONTINUE
      RETURN
C +++
C +++ GENERAL REZONE - ROLL-YOUR-OWN HERE . . .
C +++
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RINPUT                                                 RINPUT 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,CI,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,DI,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
      DIMENSION HOUT(42)                                                RINPUT 4
C +++                                                                   RINPUT 5
C +++ READ DATA DECK, COMPUTE DERIVED AND SCALAR QUANTITIES             RINPUT 6
C +++                                                                   RINPUT 7
      READ(5,210) HOUT(1),NX,HOUT(2),NY,HOUT(3),IMP,HOUT(4),INC         RINPUT 8
      IF(NX.EQ.0) RETURN                                                RINPUT 9
      READ(5,210) HOUT(5),IREZ,HOUT(6),LPR                              RINPUT10
      READ(5,210) HOUT(7),WB,HOUT(8),WL,HOUT(9),WR,HOUT(10),WT          RINPUT11
      READ(5,220) HOUT(11),DX,HOUT(12),DY,HOUT(13),CYL                  RINPUT12
      READ(5,220) HOUT(14),DT,HOUT(15),DTMAX,HOUT(16),TLIMD             RINPUT13
      READ(5,220) HOUT(17),TWFILM,HOUT(18),TWPRTR,HOUT(19),TWFIN        RINPUT14
      READ(5,220) HOUT(20),OM,HOUT(21),PEPS,HOUT(22),EPS,HOUT(23),RF    RINPUT15
      READ(5,220) HOUT(24),ARTVIS,HOUT(25),LAMBDA,HOUT(26),MU           RINPUT16
      READ(5,220) HOUT(27),ANC,HOUT(28),XI,HOUT(29),GX                  RINPUT17
      READ(5,220) HOUT(30),GY,HOUT(31),A0,HOUT(32),B0                   RINPUT18
      READ(5,220) HOUT(33),ASQ,HOUT(34),RON,HOUT(35),GM1                RINPUT19
      READ(5,220) HOUT(36),ROI,HOUT(37),SIEI                            RINPUT20
      READ(5,220) HOUT(38),UIN,HOUT(39),VIN                             RINPUT21
      READ(5,220) HOUT(40),ROIN,HOUT(41),SIEIN,HOUT(42),PAP             RINPUT22
      WRITE(12,210) HOUT(1),NX,HOUT(2),NY,HOUT(3),IMP,HOUT(4),INC       RINPUT23
      WRITE(12,210) HOUT(5),IREZ,HOUT(6),LPR                            RINPUT24
      WRITE(12,210) HOUT(7),WB,HOUT(8),WL,HOUT(9),WR,HOUT(10),WT        RINPUT25
      WRITE(12,230) HOUT(11),DX,HOUT(12),DY,HOUT(13),CYL                RINPUT26
      WRITE(12,230) HOUT(14),DT,HOUT(15),DTMAX,HOUT(16),TLIMD           RINPUT27
      WRITE(12,230) HOUT(17),TWFILM,HOUT(18),TWPRTR,HOUT(19),TWFIN      RINPUT28
      WRITE(12,230) HOUT(20),OM,HOUT(21),PEPS,HOUT(22),EPS,HOUT(23),RF  RINPUT29
      WRITE(12,230) HOUT(24),ARTVIS,HOUT(25),LAMBDA,HOUT(26),MU         RINPUT30
      WRITE(12,230) HOUT(27),ANC,HOUT(28),XI,HOUT(29),GX                RINPUT31
      WRITE(12,230) HOUT(30),GY,HOUT(31),A0,HOUT(32),B0                 RINPUT32
      WRITE(12,230) HOUT(33),ASQ,HOUT(34),RON,HOUT(35),GM1              RINPUT33
      WRITE(12,230) HOUT(36),ROI,HOUT(37),SIEI                          RINPUT34
      WRITE(12,230) HOUT(38),UIN,HOUT(39),VIN                           RINPUT35
      WRITE(12,230) HOUT(40),ROIN,HOUT(41),SIEIN,HOUT(42),PAP           RINPUT36
      WRITE(6,210) HOUT(1),NX,HOUT(2),NY,HOUT(3),IMP,HOUT(4),INC        RINPUT37
      WRITE(6,210) HOUT(5),IREZ,HOUT(6),LPR                             RINPUT38
      WRITE(6,210) HOUT(7),WB,HOUT(8),WL,HOUT(9),WR,HOUT(10),WT         RINPUT39
      WRITE(6,230) HOUT(11),DX,HOUT(12),DY,HOUT(13),CYL                 RINPUT40
      WRITE(6,230) HOUT(14),DT,HOUT(15),DTMAX,HOUT(16),TLIMD            RINPUT41
      WRITE(6,230) HOUT(17),TWFILM,HOUT(18),TWPRTR,HOUT(19),TWFIN       RINPUT42
      WRITE(6,230) HOUT(20),OM,HOUT(21),PEPS,HOUT(22),EPS,HOUT(23),RF   RINPUT43
      WRITE(6,230) HOUT(24),ARTVIS,HOUT(25),LAMBDA,HOUT(26),MU          RINPUT44
      WRITE(6,230) HOUT(27),ANC,HOUT(28),XI,HOUT(29),GX                 RINPUT45
      WRITE(6,230) HOUT(30),GY,HOUT(31),A0,HOUT(32),B0                  RINPUT46
      WRITE(6,230) HOUT(33),ASQ,HOUT(34),RON,HOUT(35),GM1               RINPUT47
      WRITE(6,230) HOUT(36),ROI,HOUT(37),SIEI                           RINPUT48
      WRITE(6,230) HOUT(38),UIN,HOUT(39),VIN                            RINPUT49
```

```
      WRITE(6,230) HOUT(40),ROIN,HOUT(41),SIEIN,HOUT(42),PAP             RINPUT50
      OMCYL=1.-CYL                                                       RINPUT51
      T=GRIND=DTMIN=0.                                                   RINPUT52
      NCYC=NUMIT=IPRES=NDUMP=0                                           RINPUT53
C +++                                                                    RINPUT54
C +++ ENSURE E.O.S. WILL BE CALCULATED PROPERLY FOR EXPLICIT RUNS...     RINPUT55
C +++                                                                    RINPUT56
      IF(IMP.EQ.0) INC=0                                                 RINPUT57
      DTF=0.2                                                            RINPUT58
      MAXIT=1000 + INC*500                                               RINPUT59
      LOOPMX=6                                                           RINPUT60
      THIRD=1./3.                                                        RINPUT61
      TWLFTH=.25*THIRD                                                   RINPUT62
      ANCO=.25*ANC                                                       RINPUT63
      XICOF=.5*(1.+XI)                                                   RINPUT64
      COLAMU=LAMBDA+2.*MU                                                RINPUT65
      DPCOF=PEPS*ROI/(2.*(1./(DX*DX)+1./(DY*DY)))                        RINPUT66
      GRFAC=1000./FLOAT(NX*NY)                                           RINPUT67
      TFILM=TWFILM                                                       RINPUT68
      TPRTR=TWPRTR                                                       RINPUT69
      NYP=NY+1                                                           RINPUT70
      NXP=NX+1                                                           RINPUT71
C +++                                                                    RINPUT72
C +++ ADJUST LIMITS OF CALCULATIONAL DO-LOOPS FOR SPECIFIED INFLOW (5)   RINPUT73
C +++ OR APPLIED PRESSURE (6) BOUNDARY CONDITIONS                        RINPUT74
C +++                                                                    RINPUT75
      IFIRST=JFIRST=IFPH1=JFPH1=1                                        RINPUT76
      ILAST=ILPH1=NX                                                     RINPUT77
      JLAST=JLPH1=NY                                                     RINPUT78
      IF(WL.GE.5) IFIRST=2                                               RINPUT79
      IF(WB.GE.5) JFIRST=2                                               RINPUT80
      IF(WR.GE.5) ILAST=NX-1                                             RINPUT81
      IF(WT.GE.5) JLAST=NY-1                                             RINPUT82
C +++                                                                    RINPUT83
C +++ PHASE 1 PRESSURE GRADIENTS INCLUDE APPLIED PRESSURE BOUNDARY       RINPUT84
C +++ BUT EXCLUDE A SPECIFIED INFLOW BOUNDARY . . .                      RINPUT85
C +++                                                                    RINPUT86
      IF(WL.EQ.5) IFPH1=2                                                RINPUT87
      IF(WB.EQ.5) JFPH1=2                                                RINPUT88
      IF(WR.EQ.5) ILPH1=NX-1                                             RINPUT89
      IF(WT.EQ.5) JLPH1=NY-1                                             RINPUT90
      ILASTV=ILAST+1                                                     RINPUT91
      JLASTV=JLAST+1                                                     RINPUT92
      ILASTM=ILAST-1                                                     RINPUT93
      JLASTM=JLAST-1                                                     RINPUT94
      IF(A0.EQ.1.0 .AND. B0.EQ.0.0) GO TO 100                            RINPUT95
      IF(WL.EQ.4 .OR. (WL.EQ.5.AND.UIN.LT.0.)) WRITE(59,240)             RINPUT96
      IF(WB.EQ.4 .OR. (WB.EQ.5.AND.VIN.LT.0.)) WRITE(59,240)             RINPUT97
      IF(WR.EQ.4 .OR. (WR.EQ.5.AND.UIN.GT.0.)) WRITE(59,240)             RINPUT98
      IF(WT.EQ.4 .OR. (WT.EQ.5.AND.VIN.GT.0.)) WRITE(59,240)             RINPUT99
  100 RETURN                                                             RINPU100
  210 FORMAT(A10,I5)                                                     RINPU101
  220 FORMAT(A10,F10.5)                                                  RINPU102
  230 FORMAT(A10,2X,1PE12.5)                                             RINPU103
  240 FORMAT(52H WARNING - OUTFLOW BOUNDARY BUT NOT FULL DONOR CELL./    RINPU104
     1 42H ARE OUTSIDE DENSITY AND ENERGY SPECIFIED?)                    RINPU105
      END                                                                RINPU106
```

```
      SUBROUTINE STRESD                                                 STRESD 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
      REAL LAMD                                                         STRESD 4
C +++                                                                   STRESD 5
C +++ THE STRESS DEVIATOR SUBROUTINE, IN WHICH WE CALCULATE THE         STRESD 6
C +++ SHEAR (MU) AND BULK (LAMBDA) VISCOSITY CONTRIBUTIONS TO UL AND VL.STRESD 7
C +++ THE 4 STRESS TERMS ARE SAVED FOR LATER USE IN SUBR. ENERGY . . .  STRESD 8
C +++ (MATERIAL STRENGTH EFFECTS COULD BE ADDED TO SUBR. STRESD.)       STRESD 9
C +++                                                                   STRESD10
      DTO2=.5*DT                                                        STRESD11
      DO 20 J=JFIRST,JLAST                                              STRESD12
      IJ=(J-1)*NXP+IFIRST                                               STRESD13
      IJP=IJ+NXP                                                        STRESD14
      DO 10 I=IFIRST,ILAST                                              STRESD15
      IPJ=IJ+1                                                          STRESD16
      IPJP=IJP+1                                                        STRESD17
      X1=X(IPJ)                                                         STRESD18
      R1=R(IPJ)                                                         STRESD19
      Y1=Y(IPJ)                                                         STRESD20
      U1=U(IPJ)                                                         STRESD21
      V1=V(IPJ)                                                         STRESD22
      X2=X(IPJP)                                                        STRESD23
      R2=R(IPJP)                                                        STRESD24
      Y2=Y(IPJP)                                                        STRESD25
      U2=U(IPJP)                                                        STRESD26
      V2=V(IPJP)                                                        STRESD27
      X3=X(IJP)                                                         STRESD28
      R3=R(IJP)                                                         STRESD29
      Y3=Y(IJP)                                                         STRESD30
      U3=U(IJP)                                                         STRESD31
      V3=V(IJP)                                                         STRESD32
      X4=X(IJ)                                                          STRESD33
      R4=R(IJ)                                                          STRESD34
      Y4=Y(IJ)                                                          STRESD35
      U4=U(IJ)                                                          STRESD36
      V4=V(IJ)                                                          STRESD37
      X24=X2-X4                                                         STRESD38
      Y24=Y2-Y4                                                         STRESD39
      X31=X3-X1                                                         STRESD40
      Y31=Y3-Y1                                                         STRESD41
      UOR=(U1+U2+U3+U4)*RRSUM(IJ)*CYL                                   STRESD42
      HR13=.5*(R1+R3)                                                   STRESD43
      HR24=.5*(R2+R4)                                                   STRESD44
      DTO2M1=DTO2*RMV(IPJ)                                              STRESD45
      DTO2M2=DTO2*RMV(IPJP)                                             STRESD46
      DTO2M3=DTO2*RMV(IJP)                                              STRESD47
      DTO2M4=DTO2*RMV(IJ)                                               STRESD48
      AREA=0.5*(X24*Y31-X31*Y24)                                        STRESD49
```

```
      AREA2=AREA*2.0                                                    STRESD50
      RAREA2=1./AREA2                                                   STRESD51
      U24=U2-U4                                                         STRESD52
      U31=U3-U1                                                         STRESD53
      V24=V2-V4                                                         STRESD54
      V31=V3-V1                                                         STRESD55
      DUDX=RAREA2*(U24*Y31-U31*Y24)                                     STRESD56
      DUDY=RAREA2*(U31*X24-U24*X31)                                     STRESD57
      DVDX=RAREA2*(V24*Y31-V31*Y24)                                     STRESD58
      DVDY=RAREA2*(V31*X24-V24*X31)                                     STRESD59
      LAMD=LAMBDA*(DUDX+DVDY+UOR)                                       STRESD60
      PIXX(IJ)=2.*MU*DUDX+LAMD                                          STRESD61
      PIYY(IJ)=2.*MU*DVDY+LAMD                                          STRESD62
      PIXY(IJ)=MU*(DUDY+DVDX)                                           STRESD63
      PITH(IJ)=CYL*(2.0*MU*UOR+LAMD)                                    STRESD64
      ZZ=0.5*AREA*PITH(IJ)                                              STRESD65
      XX=HR24*(PIXY(IJ)*X24-PIXX(IJ)*Y24)                               STRESD66
      UL(IPJ)  =UL(IPJ)  +DTO2M1*(XX-ZZ)                                STRESD67
      UL(IJP)  =UL(IJP)  -DTO2M3*(XX+ZZ)                                STRESD68
      XX=HR13*(PIXY(IJ)*X31-PIXX(IJ)*Y31)                               STRESD69
      UL(IPJP)=UL(IPJP)+DTO2M2*(XX-ZZ)                                  STRESD70
      UL(IJ)   =UL(IJ)   -DTO2M4*(XX+ZZ)                                STRESD71
      XX=HR24*(PIYY(IJ)*X24-PIXY(IJ)*Y24)                               STRESD72
      VL(IPJ)  =VL(IPJ)  +DTO2M1*XX                                     STRESD73
      VL(IJP)  =VL(IJP)  -DTO2M3*XX                                     STRESD74
      XX=HR13*(PIYY(IJ)*X31-PIXY(IJ)*Y31)                               STRESD75
      VL(IPJP)=VL(IPJP)+DTO2M2*XX                                       STRESD76
      VL(IJ)   =VL(IJ)   -DTO2M4*XX                                     STRESD77
      IJ=IPJ                                                            STRESD78
   10 IJP=IPJP                                                          STRESD79
   20 CONTINUE                                                          STRESD80
      CALL BC(UL,VL)                                                    STRESD81
      RETURN                                                            STRESD82
      END                                                               STRESD83
```

```
      SUBROUTINE TAPERD                                                 TAPERD 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,X1,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   TAPERD 4
C +++ RESTART PROBLEM FROM A TAPE DUMP                                  TAPERD 5
C +++                                                                   TAPERD 6
      NTD=NY                                                            TAPERD 7
      NWCOMD=LOCF(ZZ)-LOCF(AA)+1                                        TAPERD 8
      READ(7) (AA(N),N=1,NWCOMD)                                        TAPERD 9
      IF(NTD.NE.NDUMP) GO TO 10                                         TAPERD10
      WRITE(6,100)  NDUMP,T,NCYC                                        TAPERD11
      WRITE(59,100) NDUMP,T,NCYC                                        TAPERD12
      WRITE(12,100) NDUMP,T,NCYC                                        TAPERD13
      NDUMP=NDUMP+1                                                     TAPERD14
      CALL GETJTL(TIMLMT)                                               TAPERD15
      RETURN                                                            TAPERD16
   10 WRITE(6,110)  NDUMP,NTD                                           TAPERD17
      WRITE(59,110) NDUMP,NTD                                           TAPERD18
      WRITE(12,110) NDUMP,NTD                                           TAPERD19
      CALL EXITA(4)                                                     TAPERD20
  100 FORMAT(20H  RESTARTING FROM TD,I3,3H T=,1PE12.5,6H CYCLE,I5)      TAPERD21
  110 FORMAT(20H WRONG DUMP NUMBER -,2I6)                               TAPERD22
      END                                                               TAPERD23
```

```
      SUBROUTINE TAPEWR                                                 TAPEWR 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),   COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,X1,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
      DATA NDUMP/0/                                                     TAPEWR 4
C +++                                                                   TAPEWR 5
C +++ WRITE A DUMP TAPE AND EXIT                                        TAPEWR 6
C +++                                                                   TAPEWR 7
      NWCOMD=LOCF(ZZ)-LOCF(AA)+1                                        TAPEWR 8
      WRITE(8) (AA(N),N=1,NWCOMD)                                       TAPEWR 9
      WRITE(6,100)  NDUMP,T,NCYC                                        TAPEWR10
      WRITE(59,100) NDUMP,T,NCYC                                        TAPEWR11
      WRITE(12,100) NDUMP,T,NCYC                                        TAPEWR12
      CALL EXITA(3)                                                     TAPEWR13
  100 FORMAT(11H  TAPE DUMP,I3,6H AT T=,1PE12.5,6H CYCLE,I5)            TAPEWR14
      END                                                               TAPEWR15
```

```
      SUBROUTINE TIMSTP                                                 TIMSTP 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   TIMSTP 4
C +++ COMPUTE THE NEW TIME STEP, DT                                     TIMSTP 5
C +++                                                                   TIMSTP 6
      DTCON=DTVIS=1.E+20                                                TIMSTP 7
      UTMAX=-1.E+20                                                     TIMSTP 8
      DO 40 J=1,NY                                                      TIMSTP 9
      IJ=(J-1)*NXP+1                                                    TIMSTP10
      IJP=IJ+NXP                                                        TIMSTP11
      DO 30 I=1,NX                                                      TIMSTP12
      IPJ=IJ+1                                                          TIMSTP13
      DX1=(X(IPJ)-X(IJ))**2                                             TIMSTP14
      DY1=(Y(IPJ)-Y(IJ))**2                                             TIMSTP15
      DX3=(X(IJP)-X(IJ))**2                                             TIMSTP16
      DY3=(Y(IJP)-Y(IJ))**2                                             TIMSTP17
      RD1=1./(DX1+DY1)                                                  TIMSTP18
      RD3=1./(DX3+DY3)                                                  TIMSTP19
C +++                                                                   TIMSTP20
C +++ IF SALE IS RUN IN THE LAGRANGIAN MODE (IREZ=1), THERE IS NO       TIMSTP21
C +++ STABILITY LIMIT DUE TO CONVECTION.  NONETHELESS, DT SHOULD BE     TIMSTP22
C +++ LIMITED AS FOLLOWS FOR REASONS OF ACCURACY AND TO ENSURE          TIMSTP23
C +++ POSITIVE CELL VOLUMES AT THE END OF PHASE 1 . . .                 TIMSTP24
C +++                                                                   TIMSTP25
      UV14=(U(IPJ)-U(IJ))**2+(V(IPJ)-V(IJ))**2                          TIMSTP26
      CMAX=SQRT(AMAX1(UV14*RD1,UV14*RD3))                               TIMSTP27
      UTMAX=AMAX1(UTMAX,CMAX)                                           TIMSTP28
      IF(IREZ.EQ.1) GO TO 10                                            TIMSTP29
      UVR4=UREL(IJ)**2+VREL(IJ)**2                                      TIMSTP30
      CMAX=SQRT(AMAX1(UVR4*RD1,UVR4*RD3))                               TIMSTP31
      UTMAX=AMAX1(UTMAX,CMAX)                                           TIMSTP32
   10 IF(COLAMU.EQ.0.) GO TO 20                                         TIMSTP33
      XY1=DX1+DY1                                                       TIMSTP34
      XY3=DX3+DY3                                                       TIMSTP35
C +++                                                                   TIMSTP36
C +++ BYPASS CELLS W/ RO=0, E.G. SPECIAL BOUNDARIES OR OBSTACLES . . .  TIMSTP37
C +++                                                                   TIMSTP38
      IF(RO(IJ).EQ.0.) GO TO 20                                         TIMSTP39
      DTVIS=AMIN1(DTVIS,RO(IJ)*XY1*XY3/(XY1+XY3))                       TIMSTP40
   20 IJ=IPJ                                                            TIMSTP41
   30 IJP=IJP+1                                                         TIMSTP42
   40 CONTINUE                                                          TIMSTP43
      DTGROW=1.05*DT                                                    TIMSTP44
      IF(NCYC.EQ.0) DTGROW=DT                                           TIMSTP45
      IF(UTMAX.NE.0.) DTCON=DTF/UTMAX                                   TIMSTP46
      IF(COLAMU.NE.0.) DTVIS=.5*DTVIS/COLAMU                            TIMSTP47
      DT=AMIN1(DTGROW,DTCON,DTVIS,DTMAX)                                TIMSTP48
      IF(DT.EQ.DTGROW) IDDT=1HG                                         TIMSTP49
```

```
      IF(DT.EQ.DTCON) IDOT=1HC                                          TIMSTP50
      IF(DT.EQ.DTVIS) IDOT=1HV                                          TIMSTP51
      IF(DT.EQ.DTMAX) IDOT=1HM                                          TIMSTP52
      IF(NCYC.EQ.1) DTMIN=DT*1.E-10                                     TIMSTP53
      IF(DT.GT.DTMIN) RETURN                                            TIMSTP54
      WRITE(6,100)  DT,NCYC,IDOT                                        TIMSTP55
      WRITE(59,100) DT,NCYC,IDOT                                        TIMSTP56
      WRITE(12,100) DT,NCYC,IDOT                                        TIMSTP57
      CALL EXITA(6)                                                     TIMSTP58
  100 FORMAT(4H DT=,1PE12.5,9H AT CYCLE,I5,11H, CAUSE IS ,A1)           TIMSTP59
      END                                                               TIMSTP60
```

```
      SUBROUTINE ULTOU                                                  ULTOU  2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RCSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDOT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOGPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   ULTOU  4
C +++ LAGRANGIAN CALCULATIONS BYPASS ALL CONVECTIVE FLUXING, THUS THE   ULTOU  5
C +++ LAGRANGIAN VELOCITIES ARE THE FINAL VELOCITIES FOR THE CYCLE      ULTOU  6
C +++ AND NEED TO BE TRANSFERRED TO THE N-TIME ARRAY.                   ULTOU  7
C +++                                                                   ULTOU  8
      DO 20 J=JFIRST,JLASTV                                             ULTOU  9
      IJ=(J-1)*NXP+IFIRST                                               ULTOU 10
      DO 10 I=IFIRST,ILASTV                                             ULTOU 11
      U(IJ)=UL(IJ)                                                      ULTOU 12
      V(IJ)=VL(IJ)                                                      ULTOU 13
   10 IJ=IJ+1                                                           ULTOU 14
   20 CONTINUE                                                          ULTOU 15
      RETURN                                                            ULTOU 16
      END                                                               ULTOU 17
```

```
      SUBROUTINE VELPLT                                                 VELPLT 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,XI,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   VELPLT 4
C +++ THE VELOCITY VECTOR PLOT. (CALLED FROM SUBR. FULOUT)              VELPLT 5
C +++                                                                   VELPLT 6
      IF(VMAX.EQ.0) GO TO 30                                            VELPLT 7
      CALL ADV(1)                                                       VELPLT 8
      DO 20 J=1,NYP                                                     VELPLT 9
      IJ=(J-1)*NXP+1                                                    VELPLT10
      DO 10 I=1,NXP                                                     VELPLT11
      IX1=FIXL+(X(IJ)-XL)*XCONV                                         VELPLT12
      IY1=FIYB-(Y(IJ)-YB)*YCONV                                         VELPLT13
      IX2=FIXL+(X(IJ)+U(IJ)*DROU-XL)*XCONV                              VELPLT14
      IY2=FIYB-(Y(IJ)+V(IJ)*D[illegible])-YB)*YCONV                     VELPLT15
      IF(IY2.GE.1) GO TO 5                                              VELPLT16
      IF(IY1.EQ.IY2) GO TO 5                                            VELPLT17
      IX2=IX1+(IX2-IX1)*(IY1-1)/(IY1-IY2)                               VELPLT18
      IY2=1                                                             VELPLT19
    5 CALL DRV(IX1,IY1,IX2,IY2)                                         VELPLT20
      CALL PLT(IX1,IY1,16)                                              VELPLT21
   10 IJ=IJ+1                                                           VELPLT22
   20 CONTINUE                                                          VELPLT23
      CALL LINCNT(60)                                                   VELPLT24
      WRITE(12,100) JNM,D1,C1,NAME,T,NCYC,VMAX                          VELPLT25
   30 CONTINUE                                                          VELPLT26
      RETURN                                                            VELPLT27
  100 FORMAT(2X,A10,2(2X,A8),2X,8A10/40X,3H T=,1PE12.5,6H CYCLE,I5,     VELPLT28
     1 6H VMAX=,E12.5)                                                  VELPLT29
      END                                                               VELPLT30
```

```
      SUBROUTINE VINIT                                                  VINIT  2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSDP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,X1,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   VINIT  4
C +++ INITIALIZE RO, ROL, P, PL, UL AND VL TO BEGIN THE CYCLE.          VINIT  5
C +++ MAINTAIN PAP AT APPLIED PRESSURE BDRY, IF USED                    VINIT  6
C +++                                                                   VINIT  7
      I1=J1=1                                                           VINIT  8
      I2=NX                                                             VINIT  9
      J2=NY                                                             VINIT 10
      IF(WL.EQ.6) I1=2                                                  VINIT 11
      IF(WB.EQ.6) J1=2                                                  VINIT 12
      IF(WR.EQ.6) I2=I2+1                                               VINIT 13
      IF(WT.EQ.6) J2=J2+1                                               VINIT 14
      DO 20 J=J1,J2                                                     VINIT 15
      IJ=(J-1)*NXP+I1                                                   VINIT 16
      DO 10 I=I1,I2                                                     VINIT 17
      RO(IJ)=MC(IJ)/VOL(IJ)                                             VINIT 18
C +++                                                                   VINIT 19
C +++ E.O.S. IS BYPASSED FOR IMPLICIT CALCULATIONS, BECAUSE PL FROM THE VINIT 20
C +++ PREVIOUS CYCLE IS PROBABLY THE BEST GUESS FOR THE NEXT CYCLE.     VINIT 21
C +++                                                                   VINIT 22
      IF(IMP.EQ.0) CALL EOS(P(IJ),RO(IJ),SIE(IJ),0.,0.)                 VINIT 23
   10 IJ=IJ+1                                                           VINIT 24
   20 CONTINUE                                                          VINIT 25
      PMAX=0.                                                           VINIT 26
      DO 40 J=1,NYP                                                     VINIT 27
      IJ=(J-1)*NXP+1                                                    VINIT 28
      DO 30 I=1,NXP                                                     VINIT 29
      ROL(IJ)=RO(IJ)                                                    VINIT 30
      PL(IJ)=P(IJ                                                       VINIT 31
      PMAX=AMAX1(PMAX,ABS(P(IJ)))                                       VINIT 32
      UL(IJ)=U(IJ)                                                      VINIT 33
      VL(IJ)=V(IJ)                                                      VINIT 34
   30 IJ=IJ+1                                                           VINIT 35
   40 CONTINUE                                                          VINIT 36
      RETURN                                                            VINIT 37
      END                                                               VINIT 38
```

```
      SUBROUTINE VOLUME                                                 VOLUME 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),RDSOP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,ROI,ROIN,RON,SIEI,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,X1,XICOF,XL,YB,YCONV,ZZ           COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       COMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C +++                                                                   VOLUME 4
C +++ CALCULATE VOLUMES OF ALL CELLS IN THE MESH, USING PAPPAS THEOREM  VOLUME 5
C +++                                                                   VOLUME 6
      DO 20 J=1,NY                                                      VOLUME 7
      IJ=(J-1)*NXP+1                                                    VOLUME 8
      IJP=IJ+NXP                                                        VOLUME 9
      DO 10 I=1,NX                                                      VOLUME10
      IPJ=IJ+1                                                          VOLUME11
      IPJP=IJP+1                                                        VOLUME12
      X1=X(IPJ)                                                         VOLUME13
      Y1=Y(IPJ)                                                         VOLUME14
      R1=R(IPJ)                                                         VOLUME15
      X2=X(IPJP)                                                        VOLUME16
      Y2=Y(IPJP)                                                        VOLUME17
      R2=R(IPJP)                                                        VOLUME18
      X3=X(IJP)                                                         VOLUME19
      Y3=Y(IJP)                                                         VOLUME20
      R3=R(IJP)                                                         VOLUME21
      X4=X(IJ)                                                          VOLUME22
      Y4=Y(IJ)                                                          VOLUME23
      R4=R(IJ)                                                          VOLUME24
      ATR=.5*((X3-X2)*(Y1-Y2)-(X1-X2)*(Y3-Y2))                          VOLUME25
      ABL=.5*((X1-X4)*(Y3-Y4)-(X3-X4)*(Y1-Y4))                          VOLUME26
      VOL(IJ)=THIRD*((R1+R2+R3)*ATR+(R3+R4+R1)*ABL)                     VOLUME27
      IJ=IPJ                                                            VOLUME28
   10 IJP=IPJP                                                          VOLUME29
   20 CONTINUE                                                          VOLUME30
      RETURN                                                            VOLUME31
      END                                                               VOLUME32
```

```
      SUBROUTINE ZONPLT                                                 ZONPLT 2
      COMMON /SC1/ AA(1),X(800),R(800),Y(800),U(800),V(800),MC(800),    COMD   2
     1 MV(800),RMV(800),RO(800),VOL(800),P(800),SIE(800),UL(800),       COMD   3
     2 VL(800),ROL(800),PL(800),D(800),Q(800),RRSUM(800),PIXX(800),     COMD   4
     3 PIXY(800),PIYY(800),PITH(800),ROSOP(800),UG(800),VG(800),        COMD   5
     4 UREL(800),VREL(800),MP(800),MVP(800),SIEP(800),UMOM(800),        COMD   6
     5 VMOM(800),UMOMP(800),VMOMP(800),ZZ1                              COMD   7
      COMMON /SC2/ ANC,ANCO,ARTVIS,ASQ,A0,B0,COLAMU,CYL,C1,DPCOF,       COMD   8
     1 DROU,DT,DTF,DTMAX,DTMIN,DX,DY,D1,EPS,FIXL,FIYB,GM1,GRFAC,GRIND,  COMD   9
     2 GX,GY,IDDT,IFIRST,IFPHI,ILAST,ILASTM,ILASTV,ILPHI,IMP,INC,IPRES, COMD  10
     3 IREZ,JFIRST,JFPHI,JLAST,JLASTM,JLASTV,JLPHI,JNM,LAMBDA,LOOPS,    COMD  11
     4 LOOPMX,LPR,MAXIT,MU,NAME(8),NCYC,NDUMP,NUMIT,NX,NXP,             COMD  12
     5 NY,NYP,OM,OMCYL,PAP,PEPS,PMAX,RF,RO1,ROIN,RON,SIE1,SIEIN,        COMD  13
     6 T,TFILM,THIRD,TIMLMT,TLIMD,TPRTR,TWFILM,TWFIN,TWLFTH,TWPRTR,     COMD  14
     7 UIN,VIN,VMAX,WB,WL,WR,WT,XCONV,X1,XICOF,XL,YB,YCONV,ZZ          COMD  15
      REAL LAMBDA,MC,MP,MU,MV,MVP                                       CUMD  16
      INTEGER WB,WL,WR,WT                                               COMD  17
C ***                                                                   ZONPLT 4
C *** THE ZONE PLOT  (CALLED FROM SUBR  FULOUT)                         ZONPLT 5
C ***                                                                   ZONPLT 6
      IF(IREZ.EQ.0 .AND. NCYC.GT.0) RETURN                              ZONPLT 7
      CALL ADV(1)                                                       ZONPLT 8
      DO 20 J=1,NY                                                      ZONPLT 9
      IJ=(J-1)*NXP+1                                                    ZONPLT10
      IJP=IJ+NXP                                                        ZONPLT11
      DO 10 I=1,NX                                                      ZONPLT12
      IPJ=IJ+1                                                          ZONPLT13
      IPJP=IJP+1                                                        ZONPLT14
      IX1=FIXL+(X(IPJ)-XL)*XCONV                                        ZONPLT15
      IX2=FIXL+(X(IPJP)-XL)*XCONV                                       ZONPLT16
      IX3=FIXL+(X(IJP)-XL)*XCONV                                        ZONPLT17
      IX4=FIXL+(X(IJ)-XL)*XCONV                                         ZONPLT18
      IY1=FIYB-(Y(IPJ)-YB)*YCONV                                        ZONPLT19
      IY2=FIYB-(Y(IPJP)-YB)*YCONV                                       ZONPLT20
      IY3=FIYB-(Y(IJP)-YB)*YCONV                                        ZONPLT21
      IY4=FIYB-(Y(IJ)-YB)*YCONV                                         ZONPLT22
      CALL DRV(IX4,IY4,IX3,IY3)                                         ZONPLT23
      CALL DRV(IX4,IY4,IX1,IY1)                                         ZONPLT24
      IF(I.EQ.NX) CALL DRV(IX1,IY1,IX2,IY2)                             ZONPLT25
      IF(J.EQ.NY) CALL DRV(IX2,IY2,IX3,IY3)                             ZONPLT26
      IJP=IPJP                                                          ZONPLT27
   10 IJ=IPJ                                                            ZONPLT28
   20 CONTINUE                                                          ZONPLT29
      CALL LINCNT(60)                                                   ZONPLT30
      WRITE(12,100) JNM,D1,C1,NAME,T,NCYC                               ZONPLT31
      RETURN                                                            ZONPLT32
  100 FORMAT(2X,A10,2(2X,A8),2X,8A10/40X,3H T=,1PE12.5,6H CYCLE,I5)     ZONPLT33
      END                                                               ZONPLT34
```

APPENDIX B

SYSTEM SUBROUTINE CALLS IN SALE

SALE calls a number of system subroutines to display graphic or numeric information on microfiche. The original microfilm recording CRT device at LASL was the SC 4020, and although it has been supplanted, the coordinate system in the particular software system we are using is that of the SC 4020. The CRT face has a matrix of 1024 × 1024 raster points, where (0,0) is the coordinate of the upper left corner and (1023,1023) that of the lower right corner. Because this coordinate system is different from that of SALE, our code must convert physical mesh coordinates to locate their positions on the 4020 frame. This scaling process is performed in SALE subroutine FULOUT. Numerical information is displayed in the typing mode, in which the film frame consists of 64 lines of 128 characters each. The system subroutines called by SALE are:

CALL ADV (nf) advances the film by nf frames.

CALL PLT (IX,IY,ch) plots the 4020 character identified by ch at frame coordinates (IX,IY).

CALL DRV (IX1,IY1,IX2,IY2) draws a straight line vector segment connecting the 4020 points (IX1,IY1) and (IX2,IY2).

CALL LINCNT (LN) locates the first column of line LN. Accessible lines range between 2 and 61. Frame advancement is automatic.

In addition to the above calls that are used continuously during program execution, initialization of the film file at the beginning of the run is handled by calls to GFR80, GRPHLUN, LIB4020, GRPHFTN and SETFLSH. These communicate with the graphics system at LASL and need not concern the outside user.

Other system calls not related to film usage appear in SALE:

CALL GETJTL (TL) returns the job CPU time limit in seconds.

CALL DATEH(DI) returns the date as a Hollerith constant of the form MM/DD/YY.

CALL TIMEH(TI) returns the wall clock time as a Hollerith constant of the form HH:MM:SS.

CALL SECOND(time) returns the CPU time used by the job up to this point.

CALL EXITA(n) terminates the code. Each call to EXITA in SALE has a different value for n, which is retained by the operating system and available for determining the cause of the exit, if necessary.

APPENDIX C

SAMPLE OUTPUT FROM BROKEN DAM CALCULATION

The broken dam calculation described in Sec. IV.A is chosen as a sample calculation for aiding in code verification at other installations. It is solved using the SALE program exactly as listed in App. A and the input file listed in Sec. IV.A.

The frames on the following pages provide the cell data as created at time t = 0, again after one cycle (t = 0.1), and again much later at cycle 71 (t = 5.0). Also provided are two neighboring frames showing system totals of mass, momentum, and energy, and a sampling of cycle-by-cycle monitor prints of iteration number, grind time, and time step history.

Exact agreement with our calculated values may be impossible to attain because of different word lengths on other computers and differences in various FORTRAN compilers.

```
 [illegible] SALE     04/08/79     14 28 13     T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASQ=100 040879-3
                                      T= 0              CYCLE    0

 I   J      X            Y            U            V            SIE          RHO         MASS         VOL         P
 1   1   0.           0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 2   1   1.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 3   1   2.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 4   1   3.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 5   1   4.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 6   1   5.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 7   1   6.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 8   1   7.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 9   1   8.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
10   1   9.000E+00    0.           0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
11   1   1.000E+01    0.           0.           0.           0.           0.          0.          0.          0.
 1   2   0.           1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 2   2   1.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 3   2   2.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 4   2   3.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 5   2   4.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 6   2   5.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 7   2   6.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 8   2   7.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 9   2   8.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
10   2   9.000E+00    1.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
11   2   1.000E+01    1.000E+00    0.           0.           0.           0.          0.          0.          0.
 1   3   0.           2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 2   3   1.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 3   3   2.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 4   3   3.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 5   3   4.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
     3   5.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
     3   6.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 8   3   7.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 9   3   8.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
10   3   9.000E+00    2.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
11   3   1.000E+01    2.000E+00    0.           0.           0.           0.          0.          0.          0.
 1   4   0.           3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 2   4   1.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 3   4   2.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 4   4   3.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 5   4   4.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 6   4   5.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 7   4   6.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 8   4   7.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 9   4   8.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
10   4   9.000E+00    3.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
11   4   1.000E+01    3.000E+00    0.           0.           0.           0.          0.          0.          0.
 1   5   0.           4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 2   5   1.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 3   5   2.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 4   5   3.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 5   5   4.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 6   5   5.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 7   5   6.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 8   5   7.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
 9   5   8.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
10   5   9.000E+00    4.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
11   5   1.000E+01    4.000E+00    0.           0.           0.           0.          0.          0.          0.
 1   6   0.           5.000E+00    0.           0.           0.           1.000E+00   1.000E+00   1.000E+00   0.
```

```
     XPORT SALE     04/06/79     14 28 13     T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASQ=100 040679 3
                                          T= 0           CYCLE     0

  I   J      X          Y          U          V          SIE        RHO        MASS       VOL        P
  2   6  1.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  3   6  2.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  4   6  3.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  5   6  4.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  6   6  5.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  7   6  6.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  8   6  7.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  9   6  8.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 10   6  9.000E+00  5.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 11   6  1.000E+01  5.000E+00  0.         0.         0.         0.         0.         0.         0.
  1   7  0.         6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  2   7  1.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  3   7  2.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  4   7  3.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  5   7  4.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  6   7  5.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  7   7  6.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  8   7  7.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  9   7  8.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 10   7  9.000E+00  6.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 11   7  1.000E+01  6.000E+00  0.         0.         0.         0.         0.         0.         0.
  1   8  0.         7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  2   8  1.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  3   8  2.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  4   8  3.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  5   8  4.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  6   8  5.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  7   8  6.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  8   8  7.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  9   8  8.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 10   8  9.000E+00  7.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 11   8  1.000E+01  7.000E+00  0.         0.         0.         0.         0.         0.         0.
  1   9  0.         8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  2   9  1.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  3   9  2.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  4   9  3.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  5   9  4.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  6   9  5.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  7   9  6.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  8   9  7.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  9   9  8.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 10   9  9.000E+00  8.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 11   9  1.000E+01  8.000E+00  0.         0.         0.         0.         0.         0.         0.
  1  10  0.         9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  2  10  1.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  3  10  2.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  4  10  3.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  5  10  4.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  6  10  5.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  7  10  6.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  8  10  7.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
  9  10  8.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 10  10  9.000E+00  9.000E+00  0.         0.         0.         1.000E+00  1.000E+00  1.000E+00  0.
 11  10  1.000E+01  9.000E+00  0.         0.         0.         0.         0.         0.         0.
  1  11  0.         1.000E+01  0.         0.         0.         0.         0.         0.         0.
  2  11  1.000E+00  1.000E+01  0.         0.         0.         0.         0.         0.         0.
```

```
    XPORT.SALE    04/06/79    14:28:13    T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASQ=100 040679-3
                                            T= 0.            CYCLE    0

  I   J     X           Y           U           V           SIE         RHO         MASS        VOL         F
  3  11  2.000E+00   1.000E+01   0.          0.          0.          0.          0.          0.          0.
  4  11  3.000E+00   1.000E+01   0.          0           0.          0.          0.          0.          0.
  5  11  4.000E+00   1.000E+01   0.          0.          0.          0.          0.          0.          0.
  6  11  5.000E+00   1.000E+01   0.          0.          0.          0.          0.          0.          0.
  7  11  6.000E+00   1.000E+01   0           0.          0.          0.          0.          0.          0.
  8  11  7.000E+00   1.000E+01   0.          0.          0.          0.          0.          0.          0.
  9  11  8.000E+00   1.000E+01   0.          0.          0.          0.          0.          0.          0.
 10  11  9.000E+00   1.000E+01   0.          0.          0.          0.          0.          0.          0.
 11  11  1.000E+01   1.000E+01   0.          0.          0.          0.          0.          0.          0.
```

```
     XPORT-SALE      04/08/79      14:28.13      T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASQ=100 040679-3
                                          T= 1.00000E-01 CYCLE     1

  I   J      X          Y          U          V          SIE        RHO        MASS       VOL        P
  1   1  0.         0.         0.         0.         8.303E-05  1.000E+00  1.000E+00  1.000E+00  6.288E+00
  2   1  1.001E+00  0.         8.178E-03  0.         5.038E-05  1.000E+00  1.000E+00  1.000E+00  6.188E+00
  3   1  2.002E+00  0.         1.873E-02  0.         4.463E-05  1.000E+00  1.000E+00  1.000E+00  6.020E+00
  4   1  3.003E+00  0.         2.589E-02  0.         3.781E-05  1.000E+00  1.000E+00  1.000E+00  5.761E+00
  5   1  4.004E+00  0.         3.603E-02  0.         2.958E-05  1.000E+00  1.000E+00  1.000E+00  5.401E+00
  6   1  5.005E+00  0.         4.769E-02  0.         2.072E-05  1.000E+00  1.000E+00  1.000E+00  4.924E+00
  7   1  6.008E+00  0.         6.160E-02  0.         1.122E-05  1.000E+00  1.000E+00  1.000E+00  4.308E+00
  8   1  7.008E+00  0.         7.928E-02  0.         1.033E-06  1.000E+00  1.000E+00  1.000E+00  3.515E+00
  9   1  8.010E+00  0.         1.030E-01  0.        -1.255E-05  1.000E+00  1.000E+00  1.000E+00  2.485E+00
 10   1  9.014E+00  0.         1.438E-01  0.        -1.441E-05  1.000E+00  1.000E+00  1.000E+00  1.047E+00
 11   1  1.002E+01  0.         2.095E-01  0.         0.         0.         0.         0.         0.
  1   2  0.         9.992E-01  0.        -8.311E-03  3.502E-05  1.000E+00  1.000E+00  1.000E+00  5.352E+00
  2   2  1.001E+00  9.992E-01  8.048E-03 -8.442E-03  1.088E-05  1.000E+00  1.000E+00  1.000E+00  5.273E+00
  3   2  2.002E+00  9.991E-01  1.648E-02 -8.850E-03  9.237E-06  1.000E+00  1.000E+00  1.000E+00  5.111E+00
  4   2  3.003E+00  9.990E-01  2.544E-02 -9.585E-03  7.338E-06  1.000E+00  1.000E+00  1.000E+00  4.882E+00
  5   2  4.004E+00  9.989E-01  3.535E-02 -1.073E-02  5.022E-06  1.000E+00  1.000E+00  1.000E+00  4.515E+00
  6   2  5.005E+00  9.988E-01  4.605E-02 -1.245E-02  2.384E-06  1.000E+00  1.000E+00  1.000E+00  4.058E+00
  7   2  6.008E+00  9.985E-01  6.008E-02 -1.504E-02 -5.988E-07  1.000E+00  1.000E+00  1.000E+00  3.474E+00
  8   2  7.008E+00  9.981E-01  7.659E-02 -1.927E-02 -4.085E-06  1.000E+00  1.000E+00  1.000E+00  2.735E+00
  9   2  8.010E+00  9.974E-01  9.654E-02 -2.644E-02 -7.588E-06  1.000E+00  1.000E+00  1.000E+00  1.794E+00
 10   2  9.013E+00  9.955E-01  1.298E-01 -4.511E-02 -3.648E-06  1.000E+00  1.000E+00  1.000E+00  8.404E-01
 11   2  1.002E+01  9.941E-01  1.688E-01 -5.930E-02  0.         0.         0.         0.         0.
  1   3  0.         1.998E+00  0.        -1.623E-02  2.674E-05  1.000E+00  1.000E+00  1.000E+00  4.515E+00
  2   3  1.001E+00  1.998E+00  7.662E-03 -1.649E-02  8.114E-06  1.000E+00  1.000E+00  1.000E+00  4.441E+00
  3   3  2.002E+00  1.998E+00  1.564E-02 -1.728E-02  6.923E-06  1.000E+00  1.000E+00  1.000E+00  4.290E+00
  4   3  3.002E+00  1.998E+00  2.411E-02 -1.868E-02  5.438E-06  1.000E+00  1.000E+00  1.000E+00  4.057E+00
  5   3  4.003E+00  1.998E+00  3.338E-02 -2.085E-02  3.710E-06  1.000E+00  1.000E+00  1.000E+00  3.737E+00
  6   3  5.004E+00  1.998E+00  4.373E-02 -2.404E-02  1.808E-06  1.000E+00  1.000E+00  1.000E+00  3.318E+00
  7   3  6.006E+00  1.997E+00  5.564E-02 -2.880E-02 -1.818E-07  1.000E+00  1.000E+00  1.000E+00  2.791E+00
  8   3  7.007E+00  1.996E+00  6.958E-02 -3.592E-02 -2.075E-06  1.000E+00  1.000E+00  1.000E+00  2.138E+00
  9   3  8.008E+00  1.995E+00  8.818E-02 -4.820E-02 -2.888E-06  1.000E+00  1.000E+00  1.000E+00  1.355E+00
 10   3  9.010E+00  1.993E+00  1.017E-01 -6.978E-02 -7.244E-07  1.000E+00  1.000E+00  1.000E+00  4.747E-01
 11   3  1.001E+01  1.992E+00  1.115E-01 -8.343E-02  0.         0.         0.         0.         0.
  1   4  0.         2.998E+00  0.        -2.364E-02  1.987E-05  1.000E+00  1.000E+00  1.000E+00  3.751E+00
  2   4  1.001E+00  2.998E+00  7.050E-03 -2.401E-02  5.838E-06  1.000E+00  1.000E+00  1.000E+00  3.684E+00
  3   4  2.001E+00  2.997E+00  1.438E-02 -2.512E-02  4.967E-06  1.000E+00  1.000E+00  1.000E+00  3.548E+00
  4   4  3.002E+00  2.997E+00  2.205E-02 -2.708E-02  3.892E-06  1.000E+00  1.000E+00  1.000E+00  3.340E+00
  5   4  4.003E+00  2.997E+00  3.030E-02 -3.002E-02  2.688E-06  1.000E+00  1.000E+00  1.000E+00  3.054E+00
  6   4  5.004E+00  2.997E+00  3.932E-02 -3.429E-02  1.388E-06  1.000E+00  1.000E+00  1.000E+00  2.686E+00
  7   4  6.005E+00  2.996E+00  4.921E-02 -4.037E-02  1.083E-07  1.000E+00  1.000E+00  1.000E+00  2.230E+00
  8   4  7.006E+00  2.995E+00  6.001E-02 -4.917E-02 -8.343E-07  1.000E+00  1.000E+00  1.000E+00  1.682E+00
  9   4  8.007E+00  2.994E+00  7.058E-02 -6.212E-02 -1.098E-06  1.000E+00  1.000E+00  1.000E+00  1.053E+00
 10   4  9.008E+00  2.992E+00  7.858E-02 -7.929E-02 -2.409E-07  1.000E+00  1.000E+00  1.000E+00  3.624E-01
 11   4  1.001E+01  2.991E+00  8.371E-02 -8.877E-02  0.         0.         0.         0.         0.
  1   5  0.         3.997E+00  0.        -3.038E-02  1.377E-05  1.000E+00  1.000E+00  1.000E+00  3.055E+00
  2   5  1.001E+00  3.997E+00  6.258E-03 -3.081E-02  3.975E-06  1.000E+00  1.000E+00  1.000E+00  2.997E+00
  3   5  2.001E+00  3.997E+00  1.271E-02 -3.215E-02  3.374E-06  1.000E+00  1.000E+00  1.000E+00  2.879E+00
  4   5  3.002E+00  3.997E+00  1.942E-02 -3.448E-02  2.637E-06  1.000E+00  1.000E+00  1.000E+00  2.698E+00
  5   5  4.003E+00  3.996E+00  2.648E-02 -3.797E-02  1.813E-06  1.000E+00  1.000E+00  1.000E+00  2.455E+00
  6   5  5.003E+00  3.996E+00  3.398E-02 -4.288E-02  9.688E-07  1.000E+00  1.000E+00  1.000E+00  2.143E+00
  7   5  6.004E+00  3.995E+00  4.181E-02 -4.958E-02  2.079E-07  1.000E+00  1.000E+00  1.000E+00  1.784E+00
  8   5  7.005E+00  3.994E+00  4.958E-02 -5.858E-02 -3.190E-07  1.000E+00  1.000E+00  1.000E+00  1.320E+00
  9   5  8.006E+00  3.993E+00  5.648E-02 -7.015E-02 -4.914E-07  1.000E+00  1.000E+00  1.000E+00  8.188E-01
 10   5  9.006E+00  3.992E+00  6.147E-02 -8.418E-02 -1.011E-07  1.000E+00  1.000E+00  1.000E+00  2.803E-01
 11   5  1.001E+01  3.991E+00  6.427E-02 -9.179E-02  0.         0.         0.         0.         0.
  1   6  0.         4.996E+00  0.        -3.623E-02  9.000E-06  1.000E+00  1.000E+00  1.000E+00  2.417E+00
```

```
    XPORT-SALE      04/06/79      14:28:13      T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASQ=100 040679-3
                                          T= 1.00000E-01 CYCLE     1

  I   J      X          Y          U          V          SIE        RHO        MASS       VOL        P
  2   6  1.001E+00  4.996E+00  5.331E-03 -3.672E-02  2.492E-08  1.000E+00  1.000E+00  1.000E+00  2.389E+00
  3   6  2.001E+00  4.996E+00  1.080E-02 -3.822E-02  2.107E-08  1.000E+00  1.000E+00  1.000E+00  2.271E+00
  4   6  3.002E+00  4.998E+00  1.642E-02 -4.079E-02  1.637E-08  1.000E+00  1.000E+00  1.000E+00  2.123E+00
  5   6  4.002E+00  4.996E+00  2.222E-02 -4.457E-02  1.118E-08  1.000E+00  1.000E+00  1.000E+00  1.922E+00
  6   6  5.003E+00  4.995E+00  2.818E-02 -4.973E-02  8.021E-07  1.000E+00  1.000E+00  1.000E+00  1.670E+00
  7   6  6.003E+00  4.994E+00  3.413E-02 -5.649E-02  1.557E-07  1.000E+00  1.000E+00  1.000E+00  1.367E+00
  8   6  7.004E+00  4.993E+00  3.972E-02 -6.501E-02 -1.488E-07  1.000E+00  1.000E+00  1.000E+00  1.017E+00
  9   6  8.004E+00  4.992E+00  4.445E-02 -7.533E-02 -2.495E-07  1.000E+00  1.000E+00  1.000E+00  6.284E-01
 10   6  9.005E+00  4.991E+00  4.765E-02 -8.718E-02 -4.804E-08  1.000E+00  1.000E+00  1.000E+00  2.140E-01
 11   6  1.000E+01  4.991E+00  4.943E-02 -9.338E-02  0.         0.         0.         0.         0.
  1   7  0.         5.996E+00  0.        -4.112E-02  5.299E-08  1.000E+00  1.000E+00  1.000E+00  1.829E+00
  2   7  1.000E+00  5.996E+00  4.320E-03 -4.164E-02  1.364E-08  1.000E+00  1.000E+00  1.000E+00  1.791E+00
  3   7  2.001E+00  5.996E+00  8.731E-03 -4.322E-02  1.142E-08  1.000E+00  1.000E+00  1.000E+00  1.714E+00
  4   7  3.001E+00  5.995E+00  1.321E-02 -4.593E-02  8.720E-07  1.000E+00  1.000E+00  1.000E+00  1.598E+00
  5   7  4.002E+00  5.995E+00  1.776E-02 -4.982E-02  5.794E-07  1.000E+00  1.000E+00  1.000E+00  1.443E+00
  6   7  5.002E+00  5.994E+00  2.230E-02 -5.501E-02  2.938E-07  1.000E+00  1.000E+00  1.000E+00  1.249E+00
  7   7  6.003E+00  5.994E+00  2.869E-02 -6.158E-02  5.361E-08  1.000E+00  1.000E+00  1.000E+00  1.019E+00
  8   7  7.003E+00  5.993E+00  3.068E-02 -6.952E-02 -1.015E-07  1.000E+00  1.000E+00  1.000E+00  7.553E-01
  9   7  8.003E+00  5.992E+00  3.392E-02 -7.878E-02 -1.458E-07  1.000E+00  1.000E+00  1.000E+00  4.853E-01
 10   7  9.004E+00  5.991E+00  3.607E-02 -8.906E-02 -2.780E-08  1.000E+00  1.000E+00  1.000E+00  1.582E-01
 11   7  1.000E+01  5.991E+00  3.723E-02 -9.442E-02  0.         0.         0.         0.         0.
  1   8  0.         6.998E+00  0.        -4.496E-02  2.609E-06  1.000E+00  1.000E+00  1.000E+00  1.278E+00
  2   8  1.000E+00  6.995E+00  3.261E-03 -4.549E-02  5.717E-07  1.000E+00  1.000E+00  1.000E+00  1.251E+00
  3   8  2.001E+00  6.995E+00  6.576E-03 -4.712E-02  4.665E-07  1.000E+00  1.000E+00  1.000E+00  1.198E+00
  4   8  3.001E+00  6.995E+00  9.914E-03 -4.987E-02  3.391E-07  1.000E+00  1.000E+00  1.000E+00  1.113E+00
  5   8  4.001E+00  6.995E+00  1.325E-02 -5.376E-02  2.017E-07  1.000E+00  1.000E+00  1.000E+00  1.003E+00
  6   8  5.002E+00  6.994E+00  1.652E-02 -5.888E-02  7.147E-08  1.000E+00  1.000E+00  1.000E+00  8.687E-01
  7   8  6.002E+00  6.993E+00  1.962E-02 -6.518E-02 -3.207E-08  1.000E+00  1.000E+00  1.000E+00  7.051E-01
  8   8  7.002E+00  6.993E+00  2.235E-02 -7.263E-02 -9.098E-08  1.000E+00  1.000E+00  1.000E+00  5.215E-01
  9   8  8.002E+00  6.992E+00  2.453E-02 -8.109E-02 -9.420E-08  1.000E+00  1.000E+00  1.000E+00  3.208E-01
 10   8  9.003E+00  6.991E+00  2.595E-02 -9.031E-02 -1.750E-08  1.000E+00  1.000E+00  1.000E+00  1.088E-01
 11   8  1.000E+01  6.990E+00  2.671E-02 -9.507E-02  0.         0.         0.         0.         0.
  1   9  0.         7.995E+00  0.        -4.771E-02  8.801E-07  1.000E+00  1.000E+00  1.000E+00  7.553E-01
  2   9  1.000E+00  7.995E+00  2.179E-03 -4.826E-02  1.046E-07  1.000E+00  1.000E+00  1.000E+00  7.388E-01
  3   9  2.000E+00  7.995E+00  4.389E-03 -4.990E-02  7.270E-08  1.000E+00  1.000E+00  1.000E+00  7.080E-01
  4   9  3.001E+00  7.995E+00  6.587E-03 -5.266E-02  3.329E-08  1.000E+00  1.000E+00  1.000E+00  6.588E-01
  5   9  4.001E+00  7.994E+00  8.783E-03 -5.653E-02 -8.482E-09  1.000E+00  1.000E+00  1.000E+00  5.910E-01
  6   9  5.001E+00  7.994E+00  1.090E-02 -6.153E-02 -4.554E-08  1.000E+00  1.000E+00  1.000E+00  5.097E-01
  7   9  6.001E+00  7.993E+00  1.287E-02 -6.760E-02 -7.011E-08  1.000E+00  1.000E+00  1.000E+00  4.140E-01
  8   9  7.001E+00  7.993E+00  1.459E-02 -7.467E-02 -7.554E-08  1.000E+00  1.000E+00  1.000E+00  3.058E-01
  9   9  8.002E+00  7.992E+00  1.593E-02 -8.257E-02 -5.895E-08  1.000E+00  1.000E+00  1.000E+00  1.879E-01
 10   9  9.002E+00  7.991E+00  1.681E-02 -9.110E-02 -1.074E-08  1.000E+00  1.000E+00  1.000E+00  6.375E-02
 11   9  1.000E+01  7.990E+00  1.726E-02 -9.549E-02  0.         0.         0.         0.         0.
  1  10  0.         8.995E+00  0.        -4.937E-02  1.352E-07  1.000E+00  1.000E+00  1.000E+00  2.490E-01
  2  10  1.000E+00  8.995E+00  1.090E-03 -4.992E-02 -2.466E-09  1.000E+00  1.000E+00  1.000E+00  2.435E-01
  3  10  2.000E+00  8.995E+00  2.193E-03 -5.158E-02 -5.174E-09  1.000E+00  1.000E+00  1.000E+00  2.326E-01
  4  10  3.000E+00  8.995E+00  3.291E-03 -5.431E-02 -8.665E-09  1.000E+00  1.000E+00  1.000E+00  2.162E-01
  5  10  4.000E+00  8.994E+00  4.371E-03 -5.815E-02 -1.225E-08  1.000E+00  1.000E+00  1.000E+00  1.944E-01
  6  10  5.001E+00  8.994E+00  5.406E-03 -6.306E-02 -1.503E-08  1.000E+00  1.000E+00  1.000E+00  1.675E-01
  7  10  6.001E+00  8.993E+00  6.383E-03 -6.898E-02 -1.598E-08  1.000E+00  1.000E+00  1.000E+00  1.358E-01
  8  10  7.001E+00  8.992E+00  7.188E-03 -7.582E-02 -1.441E-08  1.000E+00  1.000E+00  1.000E+00  1.003E-01
  9  10  8.001E+00  8.992E+00  7.834E-03 -8.341E-02 -1.010E-08  1.000E+00  1.000E+00  1.000E+00  6.182E-02
 10  10  9.001E+00  8.991E+00  8.248E-03 -9.154E-02 -1.824E-09  1.000E+00  1.000E+00  1.000E+00  2.088E-02
 11  10  1.000E+01  8.990E+00  8.484E-03 -9.571E-02  0.         0.         0.         0.         0.
  1  11  0.         9.995E+00  0.        -5.020E-02  0.         0.         0.         0.         0.
  2  11  1.000E+00  9.995E+00  5.448E-04 -5.075E-02  0.         0.         0.         0.         0.
```

```
   XPORT SALE      04/06/79      14:28:13      T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASQ=100 040679-3
                                             T= 1.00000E-01 CYCLE     1

 I   J      X          Y          U          V            SIE         RHO         MASS         VOL          P
 3  11  2.000E+00  9.995E+00  1.095E-03 -5.239E-02  0.          0.          0.          0.          0.
 4  11  3.000E+00  9.994E+00  1.642E-03 -5.512E-02  0.          0.          0.          0.          0.
 5  11  4.000E+00  9.994E+00  2.177E-03 -5.894E-02  0.          0.          0           0.          0.
 6  11  5.000E+00  9.994E+00  2.689E 03 -6.381E-02  0.          0.          0.          0.          0.
 7  11  6.000E+00  9.993E+00  3.156E-03 -6.965E-02  0.          0.          0.          0.          0.
 8  11  7.000E+00  9.992E+00  3.561E-03 -7.637E-02  0.          0.          0           0.          0.
 9  11  8.000E+00  9.992E+00  3.872E-03 -8.380E-02  0.          0.          0           0           0.
10  11  9.000E+00  9.991E+00  4.073E-03 -9.175E-02  0.          0.          0.          0.          0.
11  11  1.000E+01  9.990E+00  4.177E-03 -9.582E-02  0.          0.          0.          0           0.
```

```
T= 1.00000E-01 CYCLE     1 TOT E= 2.45405442E-01 SIE= 4.45181099E-04
                             MASS= 1.00000000E+02 U MOM= 2.94712545E+00 V MOM=-4.90810883E+00
NCYC      1 T= 1.00000E-01 DT= 1.00000E-01 NUMIT= 193 GRIND= 11.855 M
NCYC      2 T= 2.00000E-01 DT= 1.00000E-01 NUMIT=  39 GRIND=  4.878 M
NCYC      3 T= 3.00000E-01 DT= 1.00000E-01 NUMIT=  41 GRIND=  1.808 M
NCYC      4 T= 4.00000E-01 DT= 1.00000E-01 NUMIT=  51 GRIND=  2.210 M
NCYC      5 T= 5.00000E-01 DT= 1.00000E-01 NUMIT=  59 GRIND=  2.506 M
NCYC      6 T= 6.00000E-01 DT= 1.00000E-01 NUMIT=  84 GRIND=  2.703 M
NCYC      7 T= 7.00000E-01 DT= 1.00000E-01 NUMIT=  68 GRIND=  2.860 M
NCYC      8 T= 8.00000E-01 DT= 1.00000E-01 NUMIT=  72 GRIND=  3.034 M
NCYC      9 T= 9.00000E-01 DT= 1.00000E-01 NUMIT=  74 GRIND=  3.110 M
```

```
      KPORT-SALE       04/06/79       14.28 13       T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASQ=100 040879-3
                                               T= 5.02599E+00 CYCLE    71

  I    J       X           Y           U           V          SIE        RHO         MASS        VOL          P
  1    1   0           0           0.          0.          2.787E-03  1.000E+00  1.000E+00  9.998E-01  6.530E+00
  2    1   1 672E+00   0           2 165E-01   0.          1.347E-03  1.000E+00  1.000E+00  9.998E-01  6.451E+00
  3    1   3 378E+00   0           4 440E-01   0           1.055E-03  1.000E+00  1 000E+00  9.998E-01  6.292E+00
  4    1   5 168E+00   0           7 030E-01   0.          6.185E-04  1.000E+00  1.000E+00  9.998E-01  6.007E+00
  5    1   7 095E+00   0           1.003E+00   0.         -1.211E-04  1.000E+00  1.000E+00  9.999E-01  5.507E+00
  6    1   9.300E+00   0.          1.418E+00   0.         -1.288E-03  9.999E-01  1.000E+00  1.000E+00  4.935E+00
  7    1   1 197E+01   0           2 043E+00   0.         -3.620E-03  9.996E-01  1.000E+00  1.000E+00  3.899E+00
  8    1   1.560E+01   0           3 082E+00   0.         -5.824E-03  9.988E-01  1.000E+00  1.001E+00  2.372E+00
  9    1   2.032E+01   0.          4 220E+00   0.         -3.820E-03  9.986E-01  1.000E+00  1.001E+00  1.488E+00
 10    1   2.524E+01   0           4.969E+00   0.         -5.853E-04  9.995E-01  1.000E+00  1.000E+00  4.387E-01
 11    1   2 894E+01   0.          5 355E+00   0.          0.         0.         0.         0.         0.
  1    2   0.          5.990E-01   0.         -7.841E-02   1.305E-03  1.000E+00  1.000E+00  9.998E-01  5.930E+00
  2    2   1.684E+00   5.824E-01   2.269E-01  -7.980E-02   1.918E-04  1.000E+00  1.000E+00  9.998E-01  5.808E+00
  3    2   3.405E+00   5.740E-01   4.668E-01  -8.024E-02   9.393E-05  1.000E+00  1.000E+00  9.998E-01  5.728E+00
  4    2   5.208E+00   5.387E-01   7.358E-01  -8.483E-02  -6.543E-05  1.000E+00  1.000E+00  9.998E-01  5.498E+00
  5    2   7.154E+00   4.928E-01   1.052E+00  -8.113E-02  -3.033E-04  1.000E+00  1.000E+00  1.000E+00  5.131E+00
  6    2   9.366E+00   4.108E-01   1.471E+00  -9.285E-02  -7.087E-04  1.000E+00  1.000E+00  1.000E+00  4.588E+00
  7    2   1.206E+01   3.316E-01   2 103E+00  -8.378E-02  -1.394E-03  1.000E+00  1.000E+00  1.000E+00  3.610E+00
  8    2   1.567E+01   2.187E-01   3.131E+00  -7.838E-02  -2.198E-03  9.998E-01  1.000E+00  1.000E+00  2.233E+00
  9    2   2.034E+01   2.078E-01   4.234E+00  -2.588E-02  -1.434E-03  9.997E-01  1.000E+00  1.000E+00  1.292E+00
 10    2   2 503E+01   2.091E-01   4 950E+00  -3.792E-02   1.298E-05  9.999E-01  1.000E+00  1.000E+00  4.105E-01
 11    2   2 808E+01   3.592E-01   5 284E+00  -2.672E-02   0.         0.         0.         0.         0.
  1    3   0.          1.199E+00   0.         -1.578E-01   1 019E-03  1.000E+00  1.000E+00  9.998E-01  5.321E+00
  2    3   1.662E+00   1 187E+00   2.221E-01  -1.598E-01   1.352E-04  1.000E+00  1.000E+00  9.998E-01  5.270E+00
  3    3   3 360E+00   1.150E+00   4.573E-01  -1.835E-01   5.197E-05  1.000E+00  1.000E+00  9.998E-01  5.181E+00
  4    3   5.133E+00   1 085E+00   7.191E-01  -1.881E-01  -7.148E-05  1.000E+00  1.000E+00  9.998E-01  4.975E+00
  5    3   7.045E+00   9.900E-01   1.031E+00  -1.677E-01  -2.580E-04  1.000E+00  1.000E+00  1.000E+00  4.877E+00
  6    3   9.193E+00   8.453E-01   1.428E+00  -1.772E-01  -5.977E-04  1.000E+00  1.000E+00  1.000E+00  4.238E+00
  7    3   1.182E+01   6.674E-01   2.044E+00  -1.775E-01  -1.140E-03  1.000E+00  1.000E+00  1.000E+00  3.411E+00
  8    3   1 529E+01   4.747E-01   3.042E+00  -1.505E-01  -1.845E-03  9.999E-01  1.000E+00  1.000E+00  2.163E+00
  9    3   1.978E-01   3.982E-01   4.129E+00  -7.097E-02  -1.272E-03  9.998E-01  1.000E+00  1.000E+00  1.193E+00
 10    3   2.417E+01   4.511E-01   4.841E+00  -6.174E-02   2.707E-05  1.000E+00  1.000E+00  1.000E+00  4.088E-01
 11    3   2.671E+01   6.810E-01   5.134E+00  -8.092E-02   0.         0.         0.         0.         0.
  1    4   0.          1.810E+00   0.         -2.408E-01   7.629E-04  1.000E+00  1.000E+00  9.998E-01  4.898E+00
  2    4   1 627E+00   1.793E+00   2.154E-01  -2.412E-01   8.886E-05  1.000E+00  1.000E+00  9.998E-01  4.858E+00
  3    4   3 286E+00   1.739E+00   4.412E-01  -2.462E-01   2.210E-05  1.000E+00  1.000E+00  9.998E-01  4.580E+00
  4    4   5 016E+00   1.846E+00   6.953E-01  -2.333E-01  -7.531E-05  1.000E+00  1.000E+00  1.000E+00  4.433E+00
  5    4   6.869E+00   1.509E+00   9.957E-01  -2.558E-01  -2.087E-04  1.000E+00  1.000E+00  1.000E+00  4.201E+00
  6    4   8.933E+00   1.312E+00   1.388E+00  -2.813E-01  -4.770E-04  1.000E+00  1.000E+00  1.000E+00  3.884E+00
  7    4   1.143E+01   1.038E+00   1.943E+00  -2 738E-01  -9.330E-04  1.000E+00  1.000E+00  1.000E+00  3.220E+00
  8    4   1.471E+01   7.636E 01   2.887E+00  -2.374E-01  -1.479E-03  9.998E-01  1.000E+00  1.000E+00  2.119E+00
  9    4   1.894E+01   6.070E-01   3.969E+00  -1.311E-01  -1.042E-03  9.998E-01  1.000E+00  1.000E+00  1.153E+00
 10    4   2.311E+01   6.810E-01   4.701E+00  -9.129E-02  -2.245E-05  1.000E+00  1.000E+00  1.000E+00  3.787E-01
 11    4   2.566E+01   1.008E+00   4.965E+00  -1.250E-01   0.         0.         0.         0.         0.
  1    5   0.          2.436E+00   0.         -3.222E-01   5 390E-04  1.000E+00  1.000E+00  9.998E-01  4.054E+00
  2    5   1.581E+00   2.414E+00   2.059E-01  -3.243E-01   4.855E-05  1.000E+00  1.000E+00  9.998E-01  4.025E+00
  3    5   3 188E+00   2.348E+00   4.213E-01  -3.287E-01   1.101E-06  1.000E+00  1.000E+00  1.000E+00  3.987E+00
  4    5   4.860E+00   2.228E+00   8.613E-01  -3.400E-01  -7.483E-05  1.000E+00  1.000E+00  1.000E+00  3.878E+00
  5    5   6.641E+00   2.060E+00   9.489E-01  -3.457E-01  -1.658E-04  1.000E+00  1.000E+00  1.000E+00  3.687E+00
  6    5   8.595E+00   1.818E+00   1.293E+00  -3.488E-01  -3.485E-04  1.000E+00  1.000E+00  1.000E+00  3.448E+00
  7    5   1.092E+01   1.471E+00   1.804E+00  -3.678E-01  -7.293E-04  1.000E+00  1.000E+00  1.000E+00  2.988E+00
  8    5   1.395E+01   1.094E+00   2.668E+00  -3.397E-01  -1.142E-03  9.999E-01  1.000E+00  1.000E+00  2.085E+00
  9    5   1.786E+01   8.577E-01   3.725E+00  -2.134E-01  -8.697E-04  9.998E-01  1.000E+00  1.000E+00  1.138E+00
 10    5   2.187E+01   8.888E-01   4.501E+00  -1.335E-01  -3.382E-05  1.000E+00  1.000E+00  1.000E+00  3.816E-01
 11    5   2.437E+01   1.249E+00   4.748E+00  -1.569E-01   0.         0.         0.         0.         0.
  1    6   0.          3.083E+00   0.         -4.049E-01   3.510E-04  1.000E+00  1.000E+00  9.998E-01  3.384E+00
```

```
     XPORT SALE      04/06/79      14:28:13      T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASQ=100 040679-3
                                         T= 5.02599E+00 CYCLE   71

  I   J      X          Y          U          V         SIE        RHO        MASS       VOL         P
  2   6  1.524E+00  3.056E+00  1.936E-01 -4.081E-01  1.545E-05  1.000E+00  1.000E+00  1.000E+00  3.372E+00
  3   6  3.073E+00  2.980E+00  3.986E-01 -4.134E-01 -1.411E-05  1.000E+00  1.000E+00  1.000E+00  3.322E+00
  4   6  4.670E+00  2.843E+00  6.198E-01 -4.250E-01 -6.592E-05  1.000E+00  1.000E+00  1.000E+00  3.278E+00
  5   6  6.370E+00  2.643E+00  8.894E-01 -4.403E-01 -1.343E-04  1.000E+00  1.000E+00  1.000E+00  3.151E+00
  6   6  8.205E+00  2.379E+00  1.212E+00 -4.364E-01 -2.299E-04  1.000E+00  1.000E+00  1.000E+00  2.955E+00
  7   6  1.032E+01  1.979E+00  1.646E+00 -4.578E-01 -5.158E-04  1.000E+00  1.000E+00  1.000E+00  2.698E+00
  8   6  1.308E+01  1.498E+00  2.410E+00 -4.539E-01 -8.408E-04  1.000E+00  1.000E+00  1.000E+00  2.020E+00
  9   6  1.666E+01  1.152E+00  3.423E+00 -3.220E-01 -7.173E-04  9.999E-01  1.000E+00  1.000E+00  1.134E+00
 10   6  2.055E+01  1.105E+00  4.271E+00 -1.954E-01 -3.338E-05  1.000E+00  1.000E+00  1.000E+00  3.473E-01
 11   6  2.305E+01  1.457E+00  4.537E+00 -1.916E-01  0.         0.         0.         0.         0.
  1   7  0.         3.753E+00  0.          4.906E-01  2.008E-04  1.000E+00  1.000E+00  9.999E-01  2.691E+00
  2   7  1.461E+00  3.727E+00  1.809E-01 -4.912E-01 -4.585E-08  1.000E+00  1.000E+00  1.000E+00  2.675E+00
  3   7  2.941E+00  3.640E+00  3.699E-01 -5.010E-01 -2.488E-05  1.000E+00  1.000E+00  1.000E+00  2.665E+00
  4   7  4.464E+00  3.496E+00  5.777E-01 -5.101E-01 -5.176E-05  1.000E+00  1.000E+00  1.000E+00  2.606E+00
  5   7  6.058E+00  3.272E+00  8.151E-01 -5.328E-01 -1.023E-04  1.000E+00  1.000E+00  1.000E+00  2.584E+00
  6   7  7.787E+00  2.985E+00  1.122E+00 -5.380E-01 -1.491E-04  1.000E+00  1.000E+00  1.000E+00  2.406E+00
  7   7  9.684E+00  2.581E+00  1.489E+00 -5.429E-01 -3.012E-04  1.000E+00  1.000E+00  1.000E+00  2.269E+00
  8   7  1.210E+01  1.999E+00  2.120E+00 -5.698E-01 -5.672E-04  1.000E+00  1.000E+00  1.000E+00  1.891E+00
  9   7  1.535E+01  1.513E+00  3.062E+00 -4.578E-01 -5.547E-04  1.000E+00  1.000E+00  1.000E+00  1.123E+00
 10   7  1.911E+01  1.344E+00  3.978E+00 -2.871E-01 -1.855E-05  1.000E+00  1.000E+00  1.000E+00  3.409E-01
 11   7  2.159E+01  1.643E+00  4.283E+00 -2.463E-01  0.         0.         0.         0.         0.
  1   8  0.         4.457E+00  0.         -5.748E-01  9.153E-05  1.000E+00  1.000E+00  9.988E-01  1.962E+00
  2   8  1.393E+00  4.428E+00  1.668E-01 -5.784E-01 -1.585E-05  1.000E+00  1.000E+00  1.000E+00  1.958E+00
  3   8  2.801E+00  4.339E+00  3.393E-01 -5.858E-01 -2.602E-05  1.000E+00  1.000E+00  1.000E+00  1.944E+00
  4   8  4.242E+00  4.183E+00  5.205E-01 -6.013E-01 -4.063E-05  1.000E+00  1.000E+00  1.000E+00  1.933E+00
  5   8  5.736E+00  3.959E+00  7.419E-01 -6.184E-01 -6.414E-05  1.000E+00  1.000E+00  1.000E+00  1.892E+00
  6   8  7.331E+00  3.642E+00  1.010E+00 -6.454E-01 -9.680E-05  1.000E+00  1.000E+00  1.000E+00  1.867E+00
  7   8  9.071E+00  3.256E+00  1.347E+00 -6.399E-01 -1.488E-04  1.000E+00  1.000E+00  1.000E+00  1.884E+00
  8   8  1.109E+01  2.655E+00  1.828E+00 -6.708E-01 -3.055E-04  1.000E+00  1.000E+00  1.000E+00  1.595E+00
  9   8  1.393E+01  1.978E+00  2.648E+00 -6.151E-01 -3.956E-04  1.000E+00  1.000E+00  1.000E+00  1.097E+00
 10   8  1.752E+01  1.633E+00  3.602E+00 -4.181E-01 -5.819E-06  1.000E+00  1.000E+00  1.000E+00  3.245E-01
 11   8  1.997E+01  1.838E+00  3.947E+00 -3.338E-01  0.         0.         0.         0.         0.
  1   9  0.         5.193E+00  0.         -6.815E-01  2.422E-05  1.000E+00  1.000E+00  1.000E+00  1.201E+00
  2   9  1.322E+00  5.164E+00  1.510E-01 -6.642E-01 -1.679E-05  1.000E+00  1.000E+00  1.000E+00  1.198E+00
  3   9  2.657E+00  5.073E+00  3.084E-01 -6.738E-01 -2.041E-05  1.000E+00  1.000E+00  1.000E+00  1.195E+00
  4   9  4.013E+00  4.919E+00  4.771E-01 -6.896E-01 -2.625E-05  1.000E+00  1.000E+00  1.000E+00  1.191E+00
  5   9  5.413E+00  4.692E+00  6.697E-01 -7.102E-01 -3.392E-05  1.000E+00  1.000E+00  1.000E+00  1.177E+00
  6   9  6.871E+00  4.381E+00  8.940E-01 -7.392E-01 -4.685E-05  1.000E+00  1.000E+00  1.000E+00  1.171E+00
  7   9  8.455E+00  3.977E+00  1.193E+00 -7.585E-01 -6.595E-05  1.000E+00  1.000E+00  1.000E+00  1.123E+00
  8   9  1.020E+01  3.448E+00  1.587E+00 -7.701E-01 -9.130E-05  1.000E+00  1.000E+00  1.000E+00  1.005E+00
  9   9  1.244E+01  2.660E+00  2.205E+00 -7.872E-01 -1.850E-04  1.000E+00  1.000E+00  1.000E+00  9.388E-01
 10   9  1.577E+01  1.999E+00  3.171E+00 -6.015E-01  5.473E-05  1.000E+00  1.000E+00  1.000E+00  3.687E-01
 11   9  1.818E+01  2.084E+00  3.543E+00 -4.571E-01  0.         0.         0.         0.         0.
  1  10  0.         5.971E+00  0.         -7.478E-01  8.703E-07  1.000E+00  1.000E+00  1.000E+00  4.048E-01
  2  10  1.252E+00  5.942E+00  1.351E-01 -7.510E-01 -6.488E-06  1.000E+00  1.000E+00  1.000E+00  4.042E-01
  3  10  2.511E+00  5.853E+00  2.748E-01 -7.611E-01 -7.310E-06  1.000E+00  1.000E+00  1.000E+00  4.045E-01
  4  10  3.787E+00  5.702E+00  4.245E-01 -7.776E-01 -8.565E-06  1.000E+00  1.000E+00  1.000E+00  4.021E-01
  5  10  5.088E+00  5.481E+00  5.903E-01 -8.011E-01 -1.085E-05  1.000E+00  1.000E+00  1.000E+00  4.085E-01
  6  10  6.432E+00  5.184E+00  7.838E-01 -8.298E-01 -1.323E-05  1.000E+00  1.000E+00  1.000E+00  3.899E-01
  7  10  7.837E+00  4.793E+00  1.014E+00 -8.612E-01 -2.314E-05  1.000E+00  1.000E+00  1.000E+00  4.238E-01
  8  10  9.381E+00  4.297E+00  1.346E+00 -8.892E-01 -2.177E-05  1.000E+00  1.000E+00  1.000E+00  3.257E-01
  9  10  1.106E+01  3.643E+00  1.739E+00 -8.731E-01 -8.968E-05  1.000E+00  1.000E+00  1.000E+00  3.898E-01
 10  10  1.369E+01  2.649E+00  2.580E+00 -8.245E-01  9.235E-06  1.000E+00  1.000E+00  1.000E+00  2.511E-01
 11  10  1.579E+01  2.442E+00  2.895E+00 -6.602E-01  0.         0.         0.         0.         0.
  1  11  0.         6.782E+00  0.         -8.338E-01  0.         0.         0.         0.         0.
  2  11  1.216E+00  6.753E+00  1.270E-01 -8.378E-01  0.         0.         0.         0.         0.
```

```
   XPORT SALE      04/06/79      14:28:1[illegible]      T3AAA SLUMP, PURE LAGRANGIAN W/ INC=1, ASO=100 040679-3
                                        T= 5.02599E+00 CYCLE    71

 I   J      X          Y          U          V           SIE         RHO         MASS         VOL          P
 3  11  2.437E+00  6.663E+00  2.579E-01  -8.493E-01  0.          0.          0.          0.          0.
 4  11  3.671E+00  6.510E+00  3.976E-01  -8.689E-01  0.          0.          0.          0.          0.
 5  11  4.923E+00  6.289E+00  5.501E-01  -8.951E-01  0.          0.          0.          0.          0.
 6  11  6.207E+00  5.988E+00  7.287E-01  -9.335E-01  0.          0.          0.          0.          0.
 7  11  7.530E+00  5.606E+00  9.298E-01  -9.634E-01  0.          0.          0.          0.          0.
 8  11  8.961E+00  5.079E+00  1.230E+00  -1.035E+00  0.          0.          0.          0.          0.
 9  11  1.044E+01  4.526E+00  1.553E+00  -1.007E+00  0.          0.          0.          0.          0.
10  11  1.259E+01  3.373E+00  2.322E+00  -1.042E+00  0.          0.          0.          0.          0.
11  11  1.450E+01  3.039E+00  2.636E+00  -8.142E-01  0.          0.          0.          0.          0.
```

```
T= 5.02599E+00 CYCLE    71 TOT E= 2.57294527E+02 SIE=-2.93271763E-02
                               MASS= 1.00000000E+02 U MOM= 1.85024287E+02 V MOM=-4.13769034E+01
NCYC      71 T= 5.02599E+00 DT= 4.05062E-02 NUMIT=  47 GRIND=  5.397 C
NCYC      72 T= 5.06649E+00 DT= 4.03490E-02 NUMIT=  47 GRIND=  2.063 C
NCYC      73 T= 5.10684E+00 DT= 4.02058E-02 NUMIT=  47 GRIND=  2.068 C
NCYC      74 T= 5.14705E+00 DT= 4.00764E-02 NUMIT=  46 GRIND=  2.016 C
NCYC      75 T= 5.18713E+00 DT= 3.99601E-02 NUMIT=  47 GRIND=  2.054 C
NCYC      76 T= 5.22709E+00 DT= 3.98565E-02 NUMIT=  46 GRIND=  2.026 C
NCYC      77 T= 5.26694E+00 DT= 3.97652E-02 NUMIT=  46 GRIND=  1.998 C
NCYC      78 T= 5.30871E+00 DT= 3.96857E-02 NUMIT=  48 GRIND=  2.006 C
NCYC      79 T= 5.34639E+00 DT= 3.96178E-02 NUMIT=  45 GRIND=  1.960 C
NCYC      80 T= 5.38601E+00 DT= 3.95611E-02 NUMIT=  45 GRIND=  1.995 C
NCYC      81 T= 5.42557E+00 DT= 3.95152E-02 NUMIT=  45 GRIND=  1.977 C
NCYC      82 T= 5.46509E+00 DT= 3.94800E-02 NUMIT=  44 GRIND=  1.928 C
NCYC      83 T= 5.50457E+00 DT= 3.94551E-02 NUMIT=  44 GRIND=  1.922 C
NCYC      84 T= 5.54402E+00 DT= 3.94402E-02 NUMIT=  43 GRIND=  1.878 C
NCYC      85 T= 5.58346E+00 DT= 3.94351E-02 NUMIT=  43 GRIND=  1.877 C
NCYC      86 T= 5.62290E+00 DT= 3.94398E-02 NUMIT=  42 GRIND=  1.843 C
NCYC      87 T= 5.66234E+00 DT= 3.94534E-02 NUMIT=  42 GRIND=  1.853 C
NCYC      88 T= 5.70179E+00 DT= 3.94784E-02 NUMIT=  42 GRIND=  1.856 C
NCYC      89 T= 5.74127E+00 DT= 3.95083E-02 NUMIT=  41 GRIND=  1.813 C
NCYC      90 T= 5.78077E+00 DT= 3.95490E-02 NUMIT=  40 GRIND=  1.770 C
NCYC      91 T= 5.82032E+00 DT= 3.95982E-02 NUMIT=  41 GRIND=  1.819 C
NCYC      92 T= 5.85992E+00 DT= 3.96558E-02 NUMIT=  39 GRIND=  1.723 C
NCYC      93 T= 5.89958E+00 DT= 3.97218E-02 NUMIT=  39 GRIND=  1.724 C
NCYC      94 T= 5.93930E+00 DT= 3.97958E-02 NUMIT=  39 GRIND=  1.728 C
NCYC      95 T= 5.97910E+00 DT= 3.98779E-02 NUMIT=  37 GRIND=  1.850 C
```

☆U.S. Government Printing Office: 1980 — 677-115/107